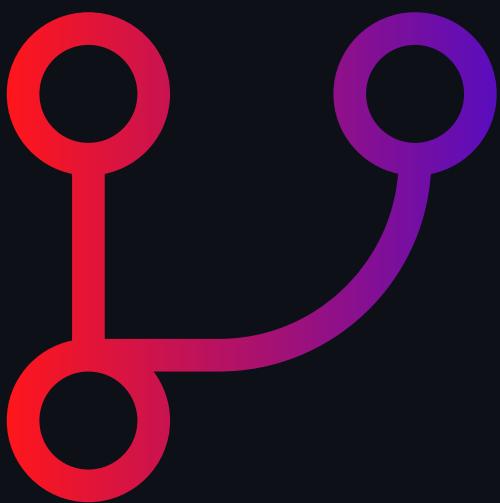


Andika Tulus Pangestu



PENGANTAR GIT & GITHUB

Panduan sederhana dalam
belajar Git dan Github untuk
pemula

Hak Cipta

Pengantar Git dan Github Dasar

Hak Cipta © 2024 oleh Andika Tulus Pangestu

Seluruh hak cipta untuk ebook ini dimiliki oleh Andika Tulus Pangestu yang diterbitkan secara independen oleh dirinya sendiri. Segala bentuk penggandaan, penyebaran, atau penggunaan untuk tujuan komersial tanpa izin tertulis dari pengarang adalah dilarang.

Pembaca dapat mengunduh salinan buku ini untuk keperluan pribadi atau pendidikan, namun tidak diizinkan untuk menjual atau mendistribusikan ulang salinan buku ini tanpa izin tertulis dari pengarang. Selain itu, juga tidak diizinkan untuk memodifikasi, memperbanyak, atau menggunakan konten dalam ebook ini untuk tujuan komersial tanpa izin tertulis dari penulis.

Penggunaan materi dalam buku ini untuk tujuan non-komersial, seperti untuk keperluan pendidikan atau penelitian, diizinkan dengan syarat mencantumkan sumber referensi yang jelas dan jangan mengubah atau memodifikasi konten asli tanpa izin tertulis dari pengarang. Pengarang buku ini tidak bertanggung jawab atas kerugian atau konsekuensi hukum yang mungkin timbul dari penggunaan buku ini.

Daftar isi

Kata Pengantar.....	2
Konvensi.....	3
Prolog.....	4
Pendahuluan.....	5
A. Version Control System.....	5
B. Mengenal GIT.....	10
C. Mengenal Github.....	12
Instalasi GIT.....	14
A. Sistem Operasi Linux.....	14
B. Sistem Operasi Windows.....	15
Konfigurasi GIT.....	24
Membuat Akun GitHub.....	25
Menghubungkan Git ke GitHub.....	28
Menghubungkan VSCode ke GitHub.....	34
Push Project ke GitHub.....	36
Membuat Perubahan Project.....	41
Mengambil Perubahan dari GitHub.....	45
Mengenal dan Membuat Branch.....	50
Berkontribusi di Project Orang Lain.....	57
Kolaborasi Project.....	66
A. Membuat Project Kolaborasi.....	67
B. Membuat Pull Request.....	71
C. Melakukan Merge.....	78
D. Mengubah Privasi Repo ke Privat.....	81
E. Mengundang Kolaborator.....	84
F. Menerima Permintaan Kolaborator.....	86
G. Membuat Perubahan sebagai Kolaborator.....	88
H. Konflik dalam Kolaborasi.....	90
Penutup.....	93
Daftar Pustaka.....	94

Kata Pengantar

Sebagai seorang seniman kode dalam dunia pemrograman. Saya senang dapat berbagi pengalaman dan pengetahuan tentang Git dan Github dalam buku ini yang berjudul "Pengantar Git dan Github untuk Pemula".

Buku ini saya tulis dengan tujuan membantu seseorang yang baru terjun atau sudah berpengalaman dalam bidang pengembangan perangkat lunak untuk memahami dasar-dasar Git dan Github, serta memberikan panduan yang jelas dan mudah dipahami tentang penggunaan Git dan Github.

Git dan Github adalah alat yang sangat berguna bagi para pengembang perangkat lunak untuk mengelola kode mereka dengan lebih efisien dan kolaboratif. Meskipun terdapat beberapa tantangan dalam menggunakan Git dan Github untuk pertama kalinya, namun dengan panduan yang tepat dan sedikit latihan, saya yakin Kamu akan mampu menguasainya.

Saya mengucapkan terima kasih kepada semua pihak yang telah berkontribusi dalam penulisan Buku ini, yang telah membantu Saya melengkapi tutorial dalam Buku ini. Terkhusus untuk Cholivia Ramadani yang telah membantu melengkapi buku ini. Lalu ada keluarga, teman, dan kolega yang turut memberikan dukungan serta masukan yang berharga, dan telah memperkaya perspektif serta kualitas Buku ini.

Akhir kata, saya sangat berharap agar Buku ini dapat membantu memperluas pengetahuan dan keterampilan. Semoga Buku ini memberikan manfaat yang besar bagi pembaca, dan saya mohon maaf bila terdapat kekurangan dalam penyusunan buku ini. Terima kasih.

Konvensi

Untuk memudahkan penggunaan Git dan Github, buku ini menggunakan beberapa konvensi dan notasi sebagai berikut:

- Direktori akan ditandai dengan tanda '/' pada akhir nama direktori, misalnya 'src/'.
- Nama branch akan ditulis dengan huruf kecil, misalnya 'main' dan 'master'.
- Penggunaan istilah asing akan di tulis tebal miring, misal *Merge*
- Teks yang ditulis dengan font Source Sans Pro akan merepresentasikan kode keseluruhan, alamat berkas/folder, bagian kode sumber, dan perintah-perintah Terminal atau CLI.
- Selain kode, penulisan teks dengan font Consolas juga diterapkan untuk merepresentasikan teks yang berada di dalam CLI.

Contoh penulisannya adalah seperti ini.

```
$ git clone https://github.com/user/repo.git  
$ cd repo/  
$ git checkout -b feature/new-feature
```

- Font dan style penulisan yang digunakan dalam buku ini adalah Times New Roman, beberapa bagian menggunakan Arial dengan ukuran font 14 dan jarak baris 1.5.

Dengan mengikuti konvensi di atas dan penggunaan font serta style penulisan yang telah ditetapkan, diharapkan lebih mudah dipahami dan lebih konsisten dalam setiap proyek yang dilakukan.

Prolog

Kita semua tahu betapa serunya mempelajari kode dan menciptakan sesuatu yang hebat, tapi apa yang terjadi ketika kita bekerja dalam tim? Seringkali, kita menemui masalah "Oh tidak, siapa yang mengubah baris kode ini?" atau "Eh, di komputer saya tidak error, kenapa di kamu tidak error?" - inilah saatnya kita memperkenalkan pahlawan tanpa jasa kita yaitu: Git!

Saya memutuskan untuk membuat "eBook Pengantar GIT & Github Dasar" ini karena, percayalah, Saya tahu betapa membingungkannya awal mula terjun ke dunia Pemrograman. Git bukan hanya sekedar alat untuk melacak perubahan dalam kode, tetapi juga merupakan kunci utama untuk kolaborasi yang sukses. Bagaimana jika kamu bekerja dengan tim yang terdiri dari berbagai keterampilan koding? Git membawa kita pada dunia di mana semua orang dapat berkontribusi tanpa merusak hasil kerja orang lain.

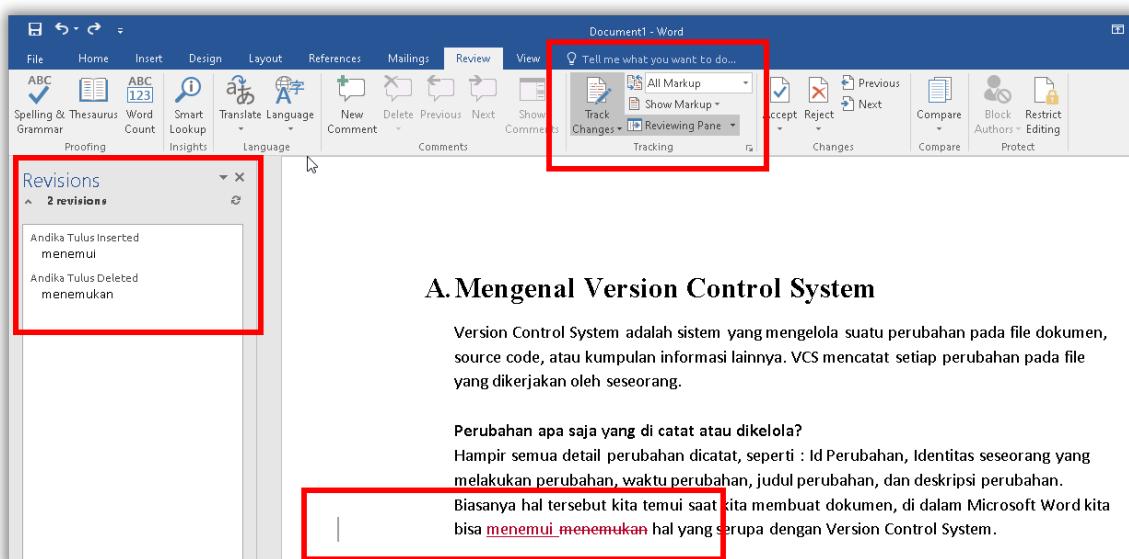
Jadi, mari kita mulai petualangan kita dalam mengenal Git dan Github. Dengan gaya santai dan bahasa yang mudah dipahami, kita akan menjelajahi dasar-dasar, dan sebelum kita sadari, kita akan menjadi ahli dalam menjalankan proyek bersama dengan harmonis.

Pendahuluan

A. Version Control System

Sistem Kontrol Versi atau dalam bahasa inggris disebut *Version Control System* adalah sebuah sistem yang digunakan untuk merekam perubahan-perubahan pada satu atau lebih file atau proyek sepanjang waktu, sehingga memungkinkan kamu untuk mengembalikan ke versi sebelumnya, melacak perubahan, dan mengkoordinasikan pekerjaan tim dengan lebih efisien.

Hampir semua rincian perubahan tercatat, seperti ID Perubahan, identitas orang yang melakukan perubahan, waktu perubahan, judul perubahan, dan deskripsi perubahan. Biasanya, kita menemui hal ini saat membuat dokumen. Dalam Microsoft Word, kita dapat menemukan fitur serupa dengan Sistem Kontrol Versi.



A. Mengenal Version Control System

Version Control System adalah sistem yang mengelola suatu perubahan pada file dokumen, source code, atau kumpulan informasi lainnya. VCS mencatat setiap perubahan pada file yang dikerjakan oleh seseorang.

Perubahan apa saja yang di catat atau dikelola?

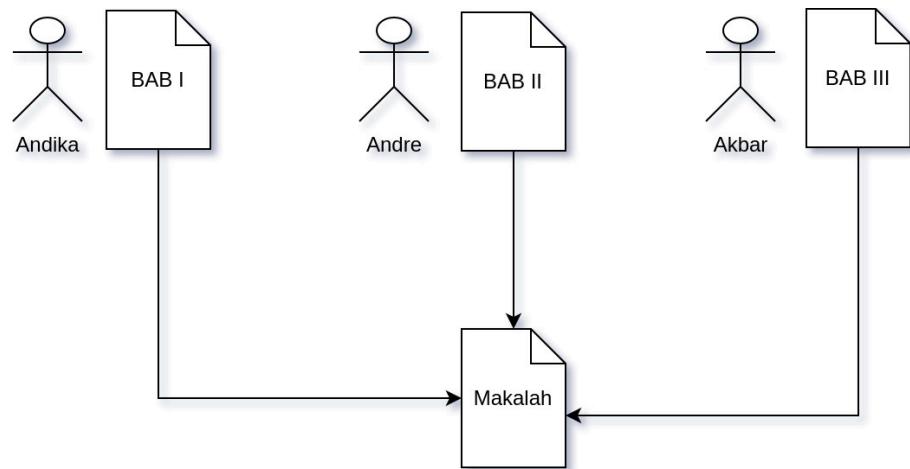
Hampir semua detail perubahan dicatat, seperti : Id Perubahan, Identitas seseorang yang melakukan perubahan, waktu perubahan, judul perubahan, dan deskripsi perubahan.

Biasanya hal tersebut kita temui saat kita membuat dokumen, di dalam Microsoft Word kita bisa [menemui menemukan](#) hal yang serupa dengan Version Control System.

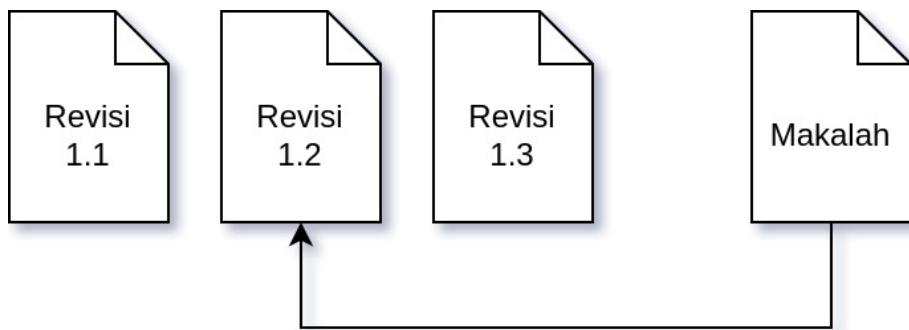
Seperti yang dapat kita lihat, sistem kontrol versi (Version Control System) melacak setiap perubahan yang kita lakukan.

Dalam gambar di atas, kita dapat melihat bahwa sistem mencatat kata '**menemukan**' dihapus (deleted) oleh pengetik, dan digantikan (insert) oleh kata '**menemui**'. Ini adalah gambaran yang jelas tentang bagaimana sistem kontrol versi bekerja.

Selain berfungsi sebagai sistem yang mengelola riwayat perubahan, sistem ini juga bermanfaat untuk mempermudah pengembangan program secara kolaboratif atau dalam kelompok. Dengan menggunakan sistem VCS, kita dapat bekerja bersama pada satu berkas secara bersamaan dengan beberapa orang tanpa harus menunggu satu per satu untuk menyelesaikan pekerjaannya secara bergantian.



Dan yang terakhir, Sistem Kontrol Versi juga memberikan kemampuan kepada kita untuk kembali ke keadaan atau versi sebelumnya dari proyek atau file. Ini berarti, jika kita melakukan perubahan yang tidak diinginkan atau mengalami kesalahan, kita dapat dengan mudah mengembalikan proyek ke kondisi yang tercatat sebelumnya. Dengan kata lain, sistem kontrol versi memungkinkan pengguna untuk melakukan perjalanan waktu pada proyek mereka, it's time traveler 😊

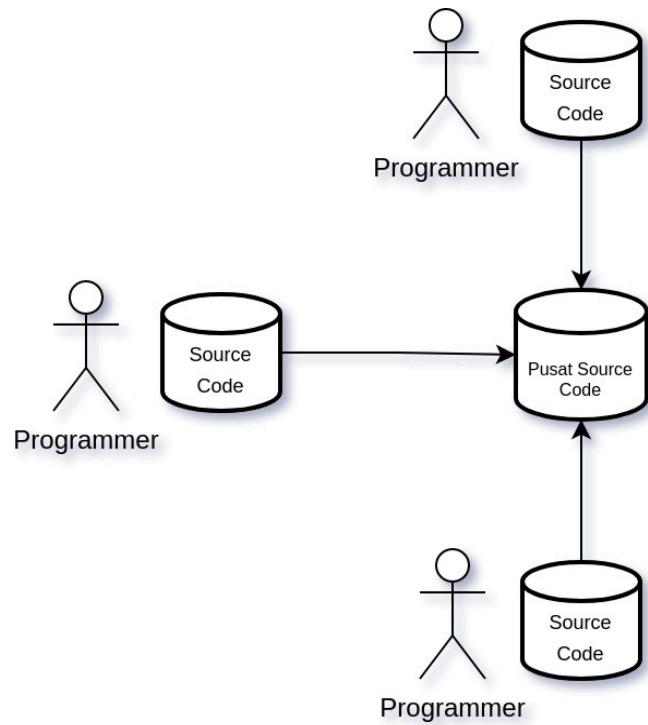


Sistem kontrol versi bekerja dengan prinsip yang mirip seperti beberapa ilustrasi di atas, tetapi dalam hal pembahasan buku ini dilakukan pada tingkat yang lebih besar dan kompleks untuk pengembangan perangkat lunak.

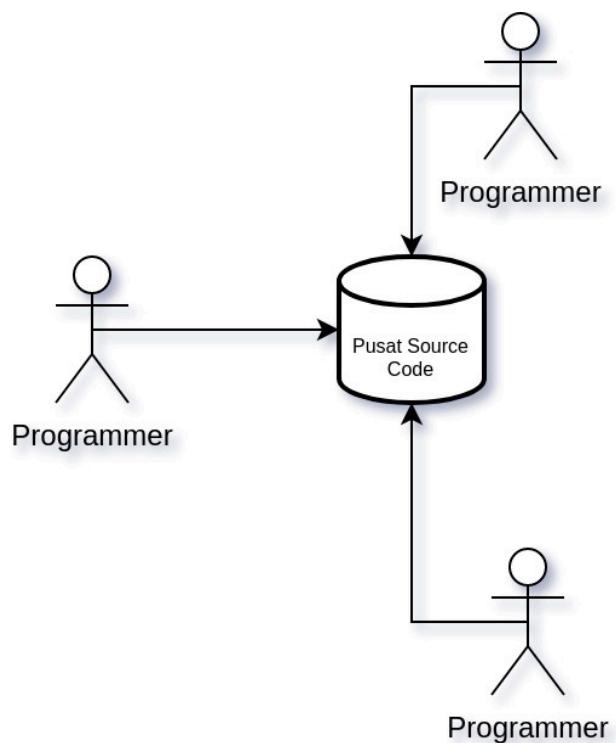
Terdapat dua jenis sistem kontrol versi yaitu terpusat dan terdistribusi, versi terpusat memiliki satu repository¹ pusat yang menyimpan riwayat perubahan dan berfungsi sebagai pusat kontrol, sehingga pengguna perlu terhubung ke repository pusat untuk mengakses sejarah atau berbagi perubahan. Sementara itu, terdistribusi seperti Git memberikan salinan penuh dari seluruh repository, termasuk catatan perubahan, kepada setiap pengguna lokal. Ini memungkinkan pengembang untuk bekerja secara mandiri tanpa bergantung pada koneksi jaringan dan memudahkan kolaborasi terdistribusi.

¹ Tempat penyimpanan untuk proyek perangkat lunak. Dalam Git, setiap proyek memiliki satu **repository** yang berisi riwayat perubahan dan file proyek.

Ilustrasi sistem kontrol versi terdistribusi :



Ilustrasi sistem kontrol versi terpusat :



Manfaat menggunakan sistem kontrol sistem atau VCS yaitu :

1. Melihat riwayat perubahan dokumen atau kode sumber
2. Menuntaskan bagian pekerjaan secara langsung tanpa menunggu bagian yang lain selesai.
3. Mengetahui siapa yang melakukan perubahan baik itu menambah, menghapus atau memodifikasi terhadap suatu dokumen atau kode sumber
4. Memungkinkan kita untuk kembali ke perubahan dokumen atau kode sumber dalam versi sebelumnya.
5. Mendeteksi konflik² yang mungkin terjadi saat penggabungan perubahan dari berbagai berkas dokumen atau kode sumber

B. Mengenal GIT



Git adalah sebuah sistem kontrol versi terdistribusi yang diciptakan oleh Linus Torvalds pada tahun 2005. Git dirancang untuk menjadi cepat, efisien, dan mendukung pengembangan perangkat lunak yang besar dan terdistribusi. Git bekerja dengan menyimpan catatan perubahan-perubahan pada suatu proyek dan memungkinkan kolaborator untuk bekerja secara bersamaan tanpa harus terhubung secara online.

² situasi ketika terdapat dua atau lebih perubahan yang bertentangan pada file atau baris kode yang sama di dalam proyek

Secara garis besar Git adalah software yang bertujuan untuk melakukan suatu pekerjaan dibidang perangkat lunak dengan menggunakan sistem kontrol versi, kalau VCS itu sistemnya sedangkan git adalah alat nya. Analogi sederhana penggunaan sistem Version Control System seperti GIT untuk kolaborasi yaitu bisa diibaratkan dengan kegiatan koding bersama dalam tugas kelompok di kuliah.

Ketika membuat sebuah project sederhana secara bersama-sama atau kelompok, maka ketua kelompok harus membagi tugas ke beberapa anggota seperti pembagian fitur yang dikerjakan, kemudian masing-masing anggota mengerjakan bagiannya masing-masing tanpa perlu menunggu bagian lain selesai terlebih dahulu, jika semua bagian telah selesai maka per-bagian itu akan dikumpulkan (*pull request*³) kepada ketua kelompok untuk digabungkan (*merge*⁴) menjadi satu project utuh.

Dalam penjelasan di kehidupan perkuliahan secara umum, git merupakan alat yang memantau setiap perubahan dalam dokumen yang kita kerjakan, kalau biasanya kita merevisi suatu dokumen itu terdapat banyak salinan/duplikasi berkas dokumen, namun dengan git kita hanya perlu memiliki satu berkas dokumen, jikapun kita ada kesalahan kita hanya perlu kembali ke versi revisi saja, bukan kembali ke berkas dokumen lain.

³ istilah permintaan penggabungan kode dalam git

⁴ istilah penggabungan kode dalam git

Sebelum ada GIT	Sesudah ada GIT
 Tugas Akhir (Baru Mulai)	• Dokumen Utama
 Tugas Akhir (Revisi 1)	• Dokumen Utama : Menambah Cover
 Tugas Akhir (Revisi 2)	• Dokumen Utama : Menambah Abstrak
 Tugas Akhir (Revisi 3)	• Dokumen Utama : Menambah Teori
 Tugas Akhir (Revisi 4)	• Dokumen Utama : Merevisi Teori dan Isi
 Tugas Akhir (Revisi 5)	• Dokumen Utama : Menambah Gambar
 Tugas Akhir (Revisi 6)	• Dokumen Utama : Menambah Daftar Isi
 Tugas Akhir Selesai	• Dokumen Utama : Selesai

Ketika kita ingin menyimpan semua perubahan pada file, biasanya kita membuat salinan baru dengan menggunakan *save as*. Kemudian, salinan tersebut akan bertumpuk di dalam folder proyek seperti yang dijelaskan sebelumnya.

Namun, setelah menggunakan Git, sekarang hanya ada satu file dokumen dalam proyek dan perubahan pada file itu disimpan dalam database. Git hanya menyimpan perbedaan atau perubahan kecil saja, tanpa harus menyimpan seluruh isi file yang bisa memakan banyak ruang penyimpanan. Ini memungkinkan kita untuk kembali ke versi tertentu dari file yang telah direvisi sebelumnya.

C. Mengenal Github



Sebelum kita mendalami berbagai perintah Git, mari kita mengenal GitHub. **GitHub** adalah platform penyimpanan dan kolaborasi proyek perangkat lunak yang menggunakan sistem kontrol versi Git.

Secara sederhana, GitHub menyediakan layanan untuk menyimpan, mengelola, dan berkolaborasi pada kode sumber proyek perangkat lunak, yang dapat diakses melalui tautan <https://www.github.com>

Pada tahun 2008 oleh Tom Preston-Werner, Chris Wanstrath, dan PJ Hyett, Tom Preston Werner pertama kali mengusulkan gagasan pembuatan sebuah situs web untuk berbagi repositori Git pada 18 Oktober 2007⁵, saat menghadiri pertemuan programmer Ruby bersama Chris Wanstrath.

Chris, seorang rekan programmer Ruby, tertarik dengan konsep tersebut dan setuju untuk bekerja sama dalam pengembangan proyek ini. Proses pengembangan awal dilakukan selama tiga bulan, di mana Tom fokus pada antarmuka pengguna dan pengembangan komponen akses repository Git menggunakan bahasa Ruby. Sementara itu, Chris membangun aplikasi menggunakan kerangka kerja Rails.

⁵ “Sejarah Github” <https://id.wikipedia.org/wiki/GitHub>

Pada pertengahan Januari 2008, versi closed beta GitHub diperkenalkan, membatasi akses hanya kepada pengguna yang menerima undangan. Pertengahan Februari 2008, PJ Hyett bergabung dengan tim pengembangan. Akhirnya, GitHub secara resmi diluncurkan untuk publik pada 10 April 2008, membuka pintu bagi pengguna yang lebih luas untuk memanfaatkan layanan tersebut.

Instalasi GIT

Sebelum dapat memulai menggunakan Git, langkah pertama yang perlu dilakukan adalah menginstalnya di sistem operasi yang digunakan. Pada distribusi Linux Ubuntu, proses instalasi Git dapat dilakukan dengan mudah melalui terminal dan di Windows dapat menggunakan berkas instalasi, dalam Buku ini akan dijelaskan secara rinci tahapan instalasi pada dua sistem operasi tersebut Linux (Ubuntu) dan Windows (Windows 10)

A. Sistem Operasi Linux

1. Periksa apakah git sudah terinstal atau belum, jalankan perintah berikut ini di terminal linux. Jika git sudah terinstal, versi git akan ditampilkan.

```
git --version
```

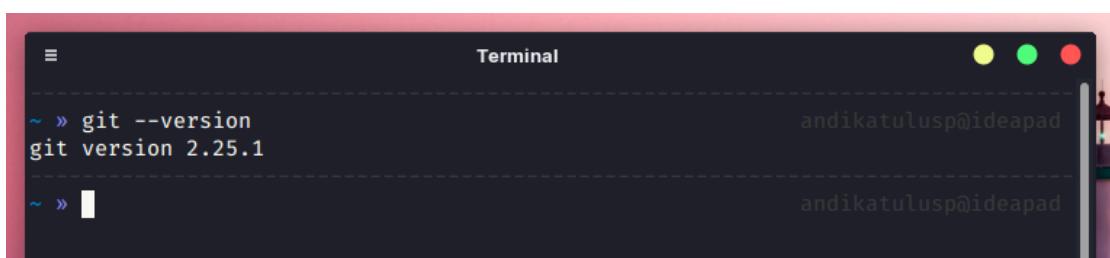
2. Pastikan paket sistem Kamu terupdate dengan menjalankan perintah

```
sudo apt update && sudo apt upgrade
```

3. Instalasi Git

```
sudo apt install git-all
```

4. Verifikasi Instalasi dengan menjalankan langka point ke-1 di atas.

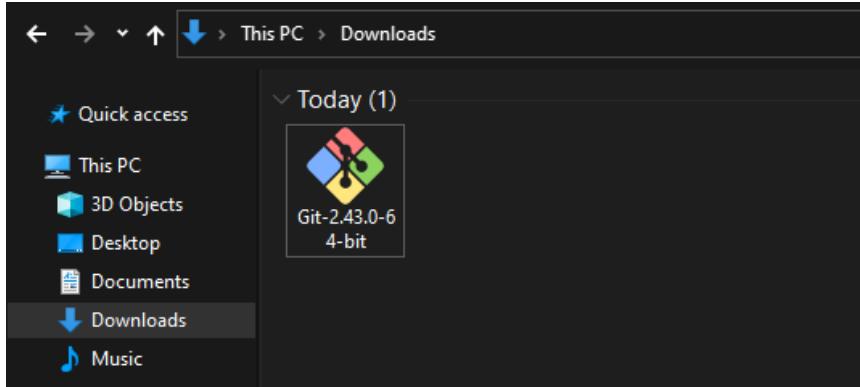


The screenshot shows a terminal window titled "Terminal". The command "git --version" is entered at the prompt (~ »), and the output "git version 2.25.1" is displayed. The terminal window has a dark background with light-colored text. The top right corner shows the user's name "andikatulusp@ideapad" and the window title "Terminal".

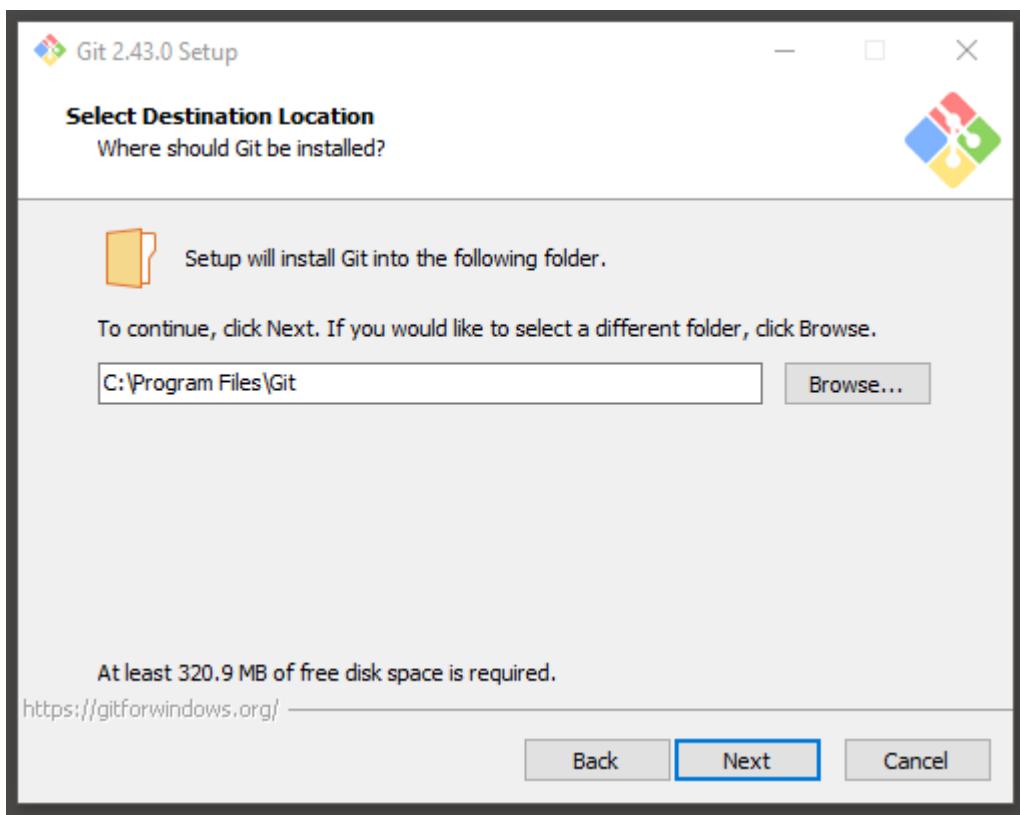
5. Selesai.

B. Sistem Operasi Windows

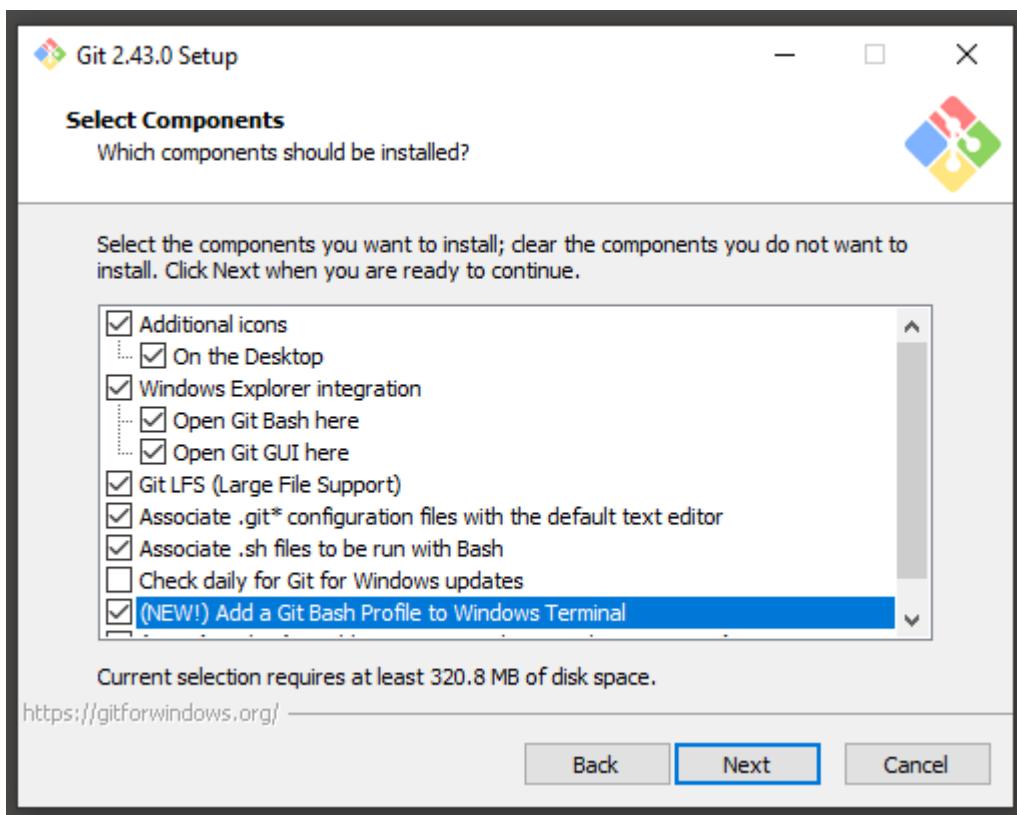
1. Unduh file instalasi git di <https://git-scm.com/downloads>
2. Kemudian taruh di folder mana saja, contoh pada folder **downloads**



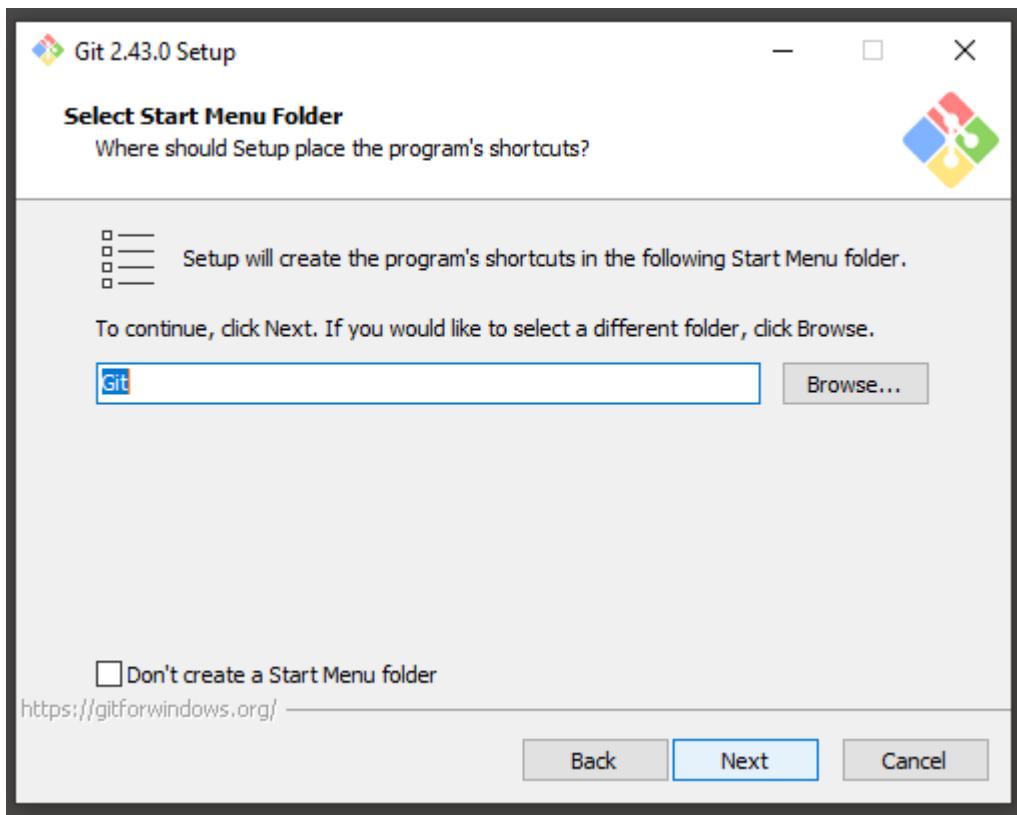
3. Klik dua kali pada file tersebut, maka akan muncul jendela instalasi.
4. Pada pemilihan lokasi instalasi, langsung saja klik “Next”



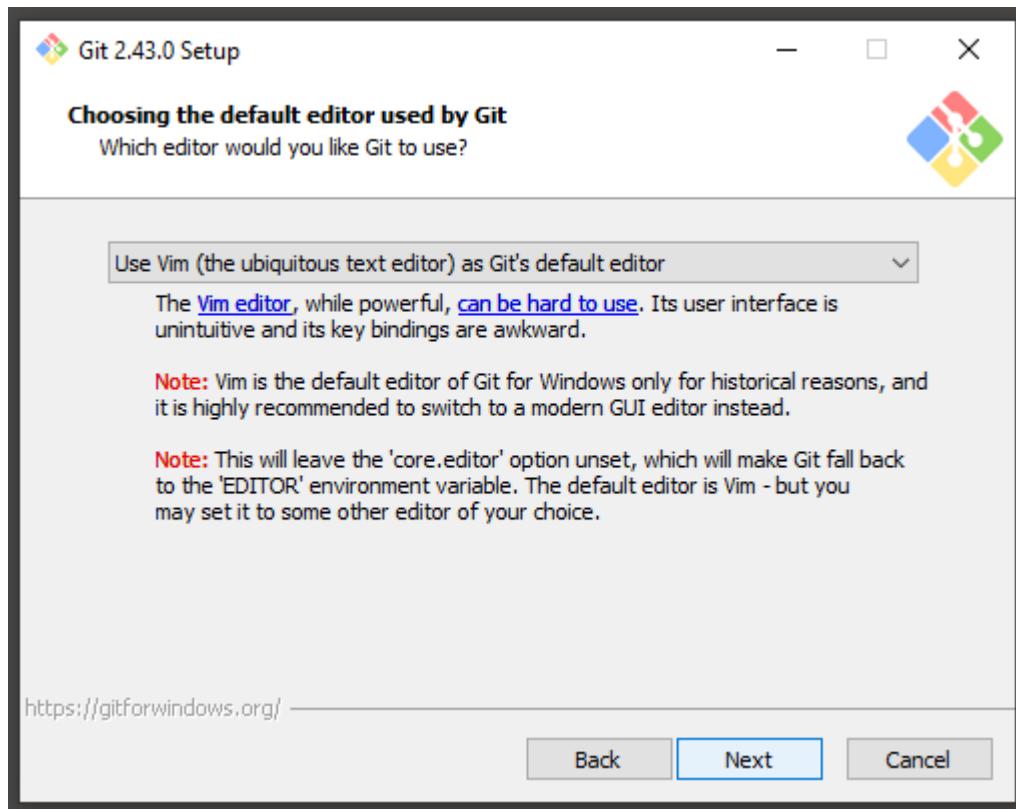
5. Kemudian, pada pemilihan komponen yang akan diinstal, sesuaikan dengan gambar di bawah ini.



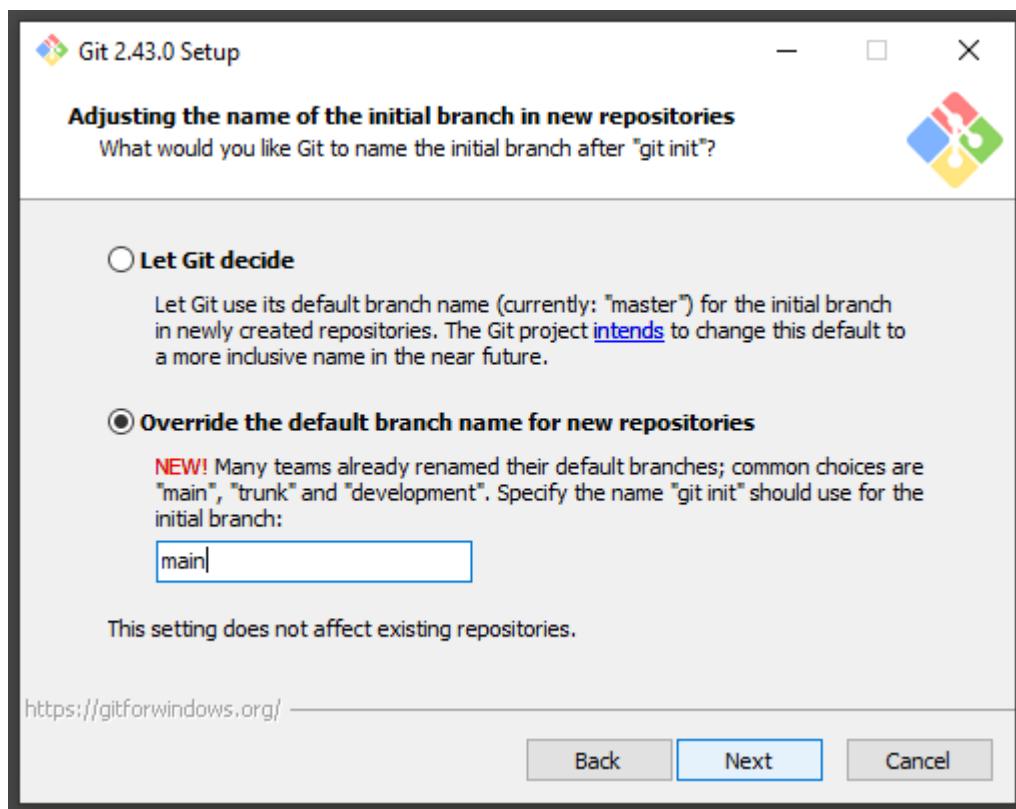
6. Beri nama shortcut icon git pada Start Menu.



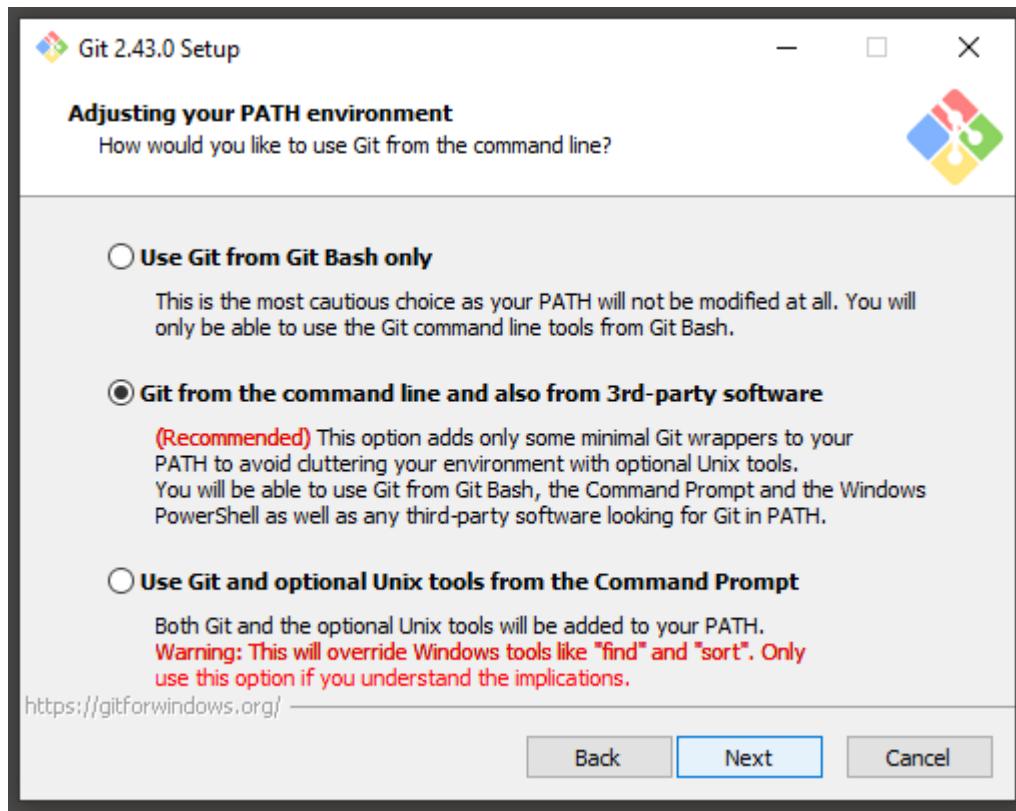
7. Pilih editor vim untuk default git editor



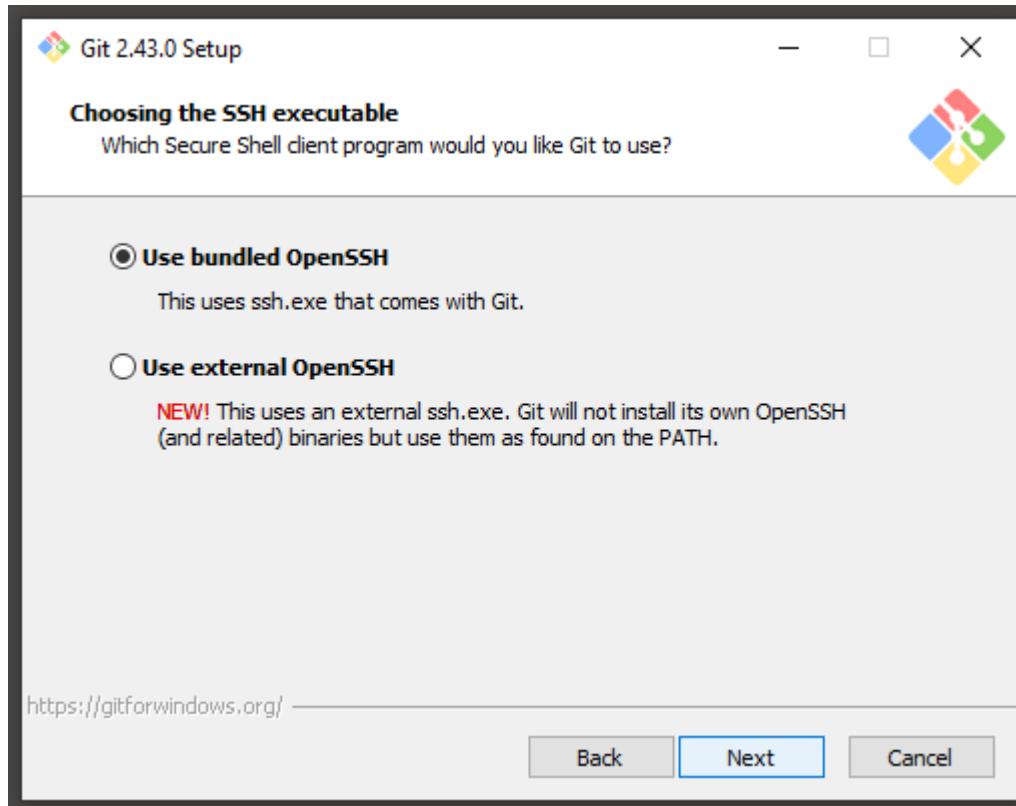
8. Ganti nama default branch ke “main” karena agar menyesuaikan default branch dari github nantinya.

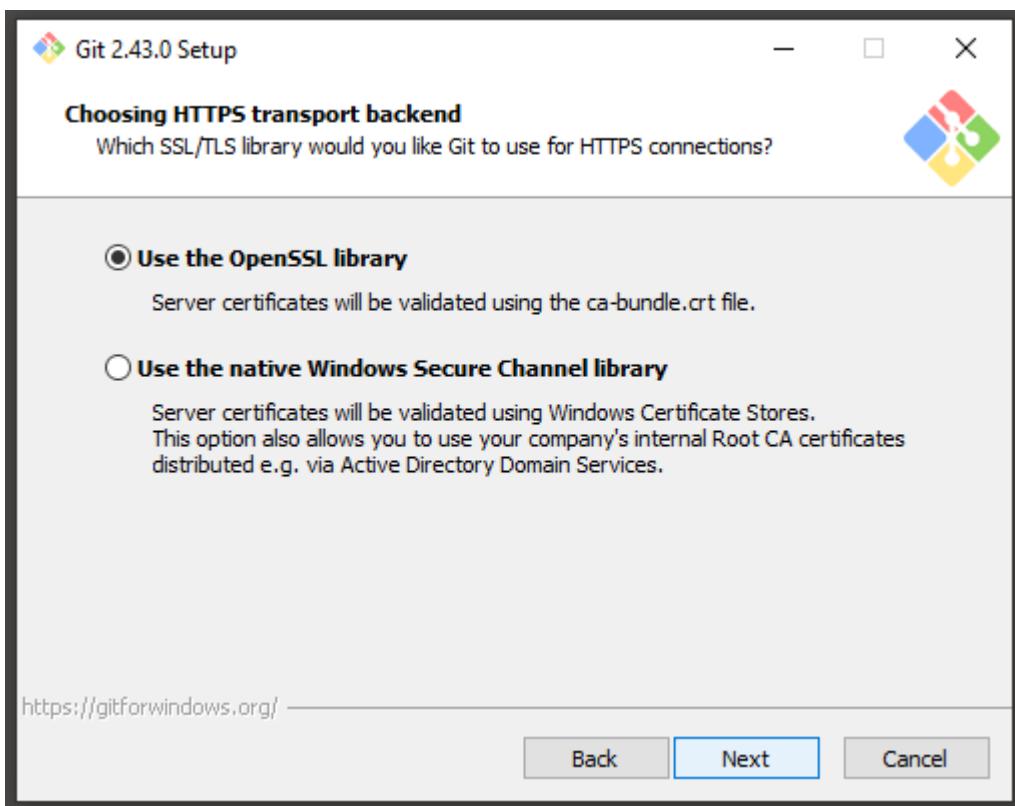


9. Izinkan perintah git dijalankan dari aplikasi pihak ketiga

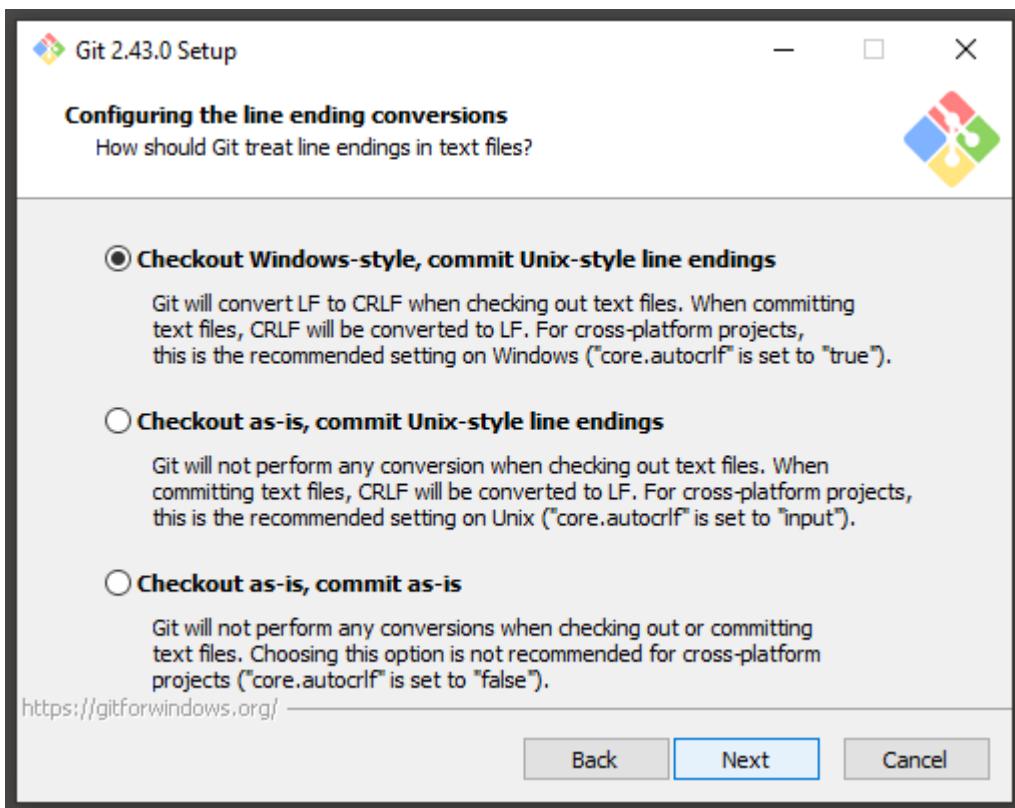


10. Pilih jenis SSH & HTTPS yang akan digunakan

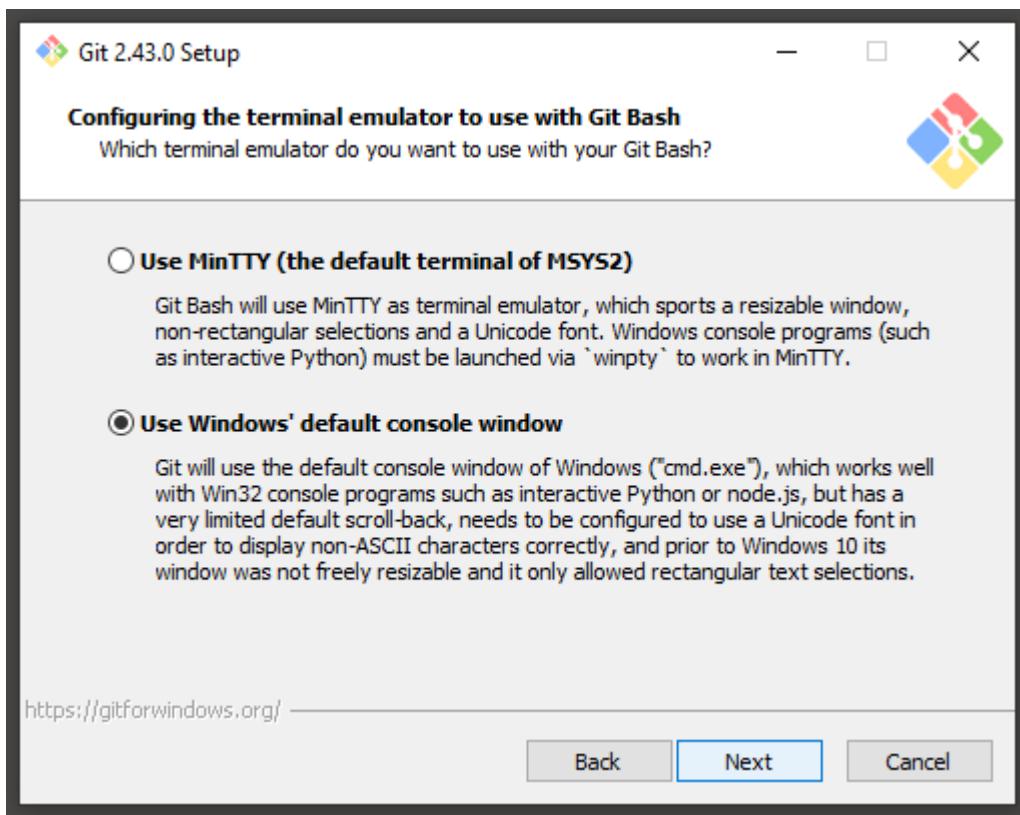




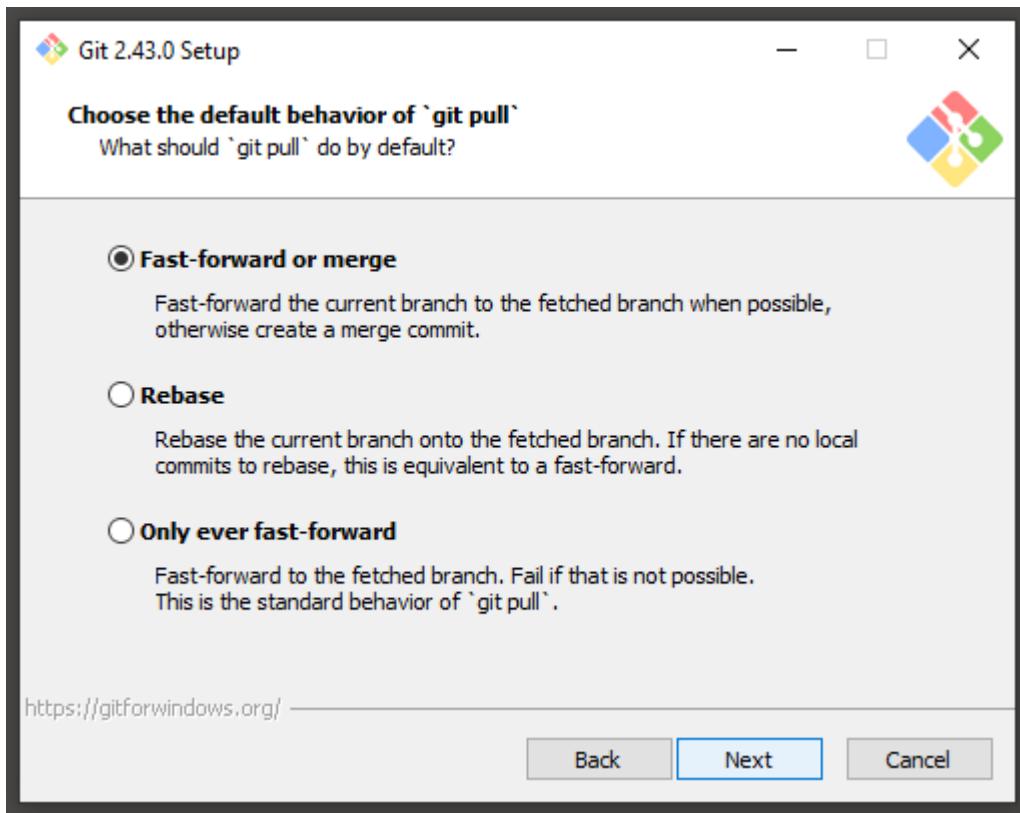
11. Konfigurasi *line-ending conversation*

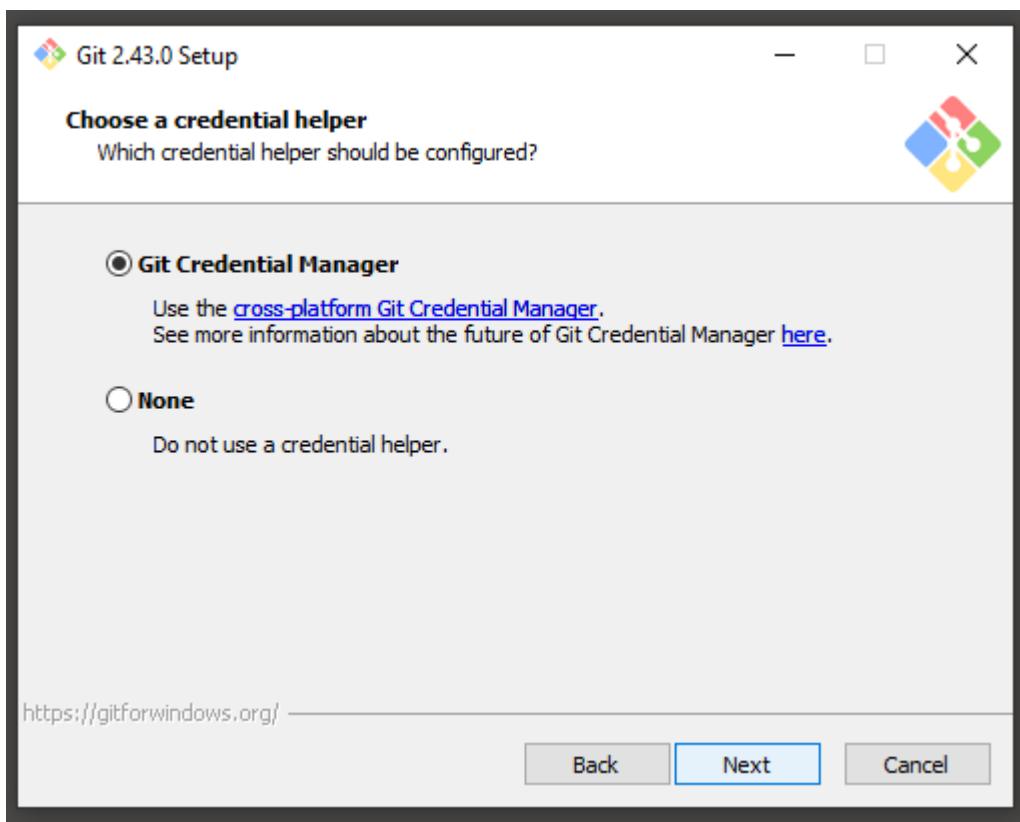


12. Jadikan *cmd* menjadi default terminal untuk menjalankan git

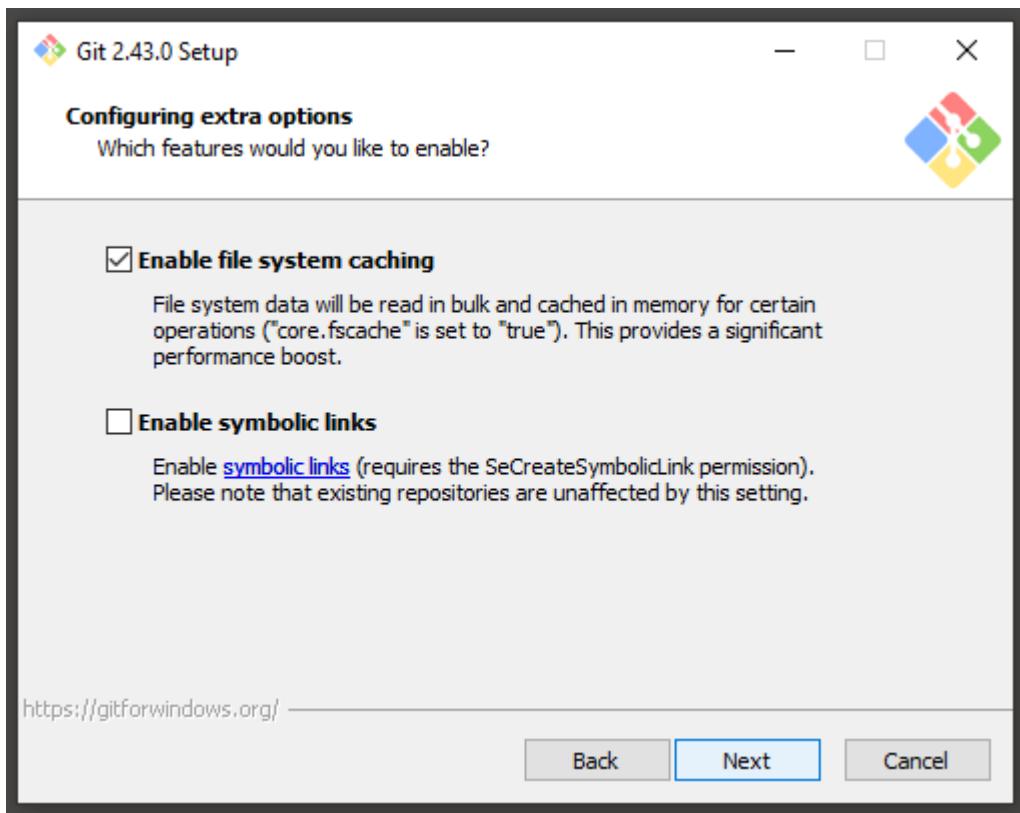


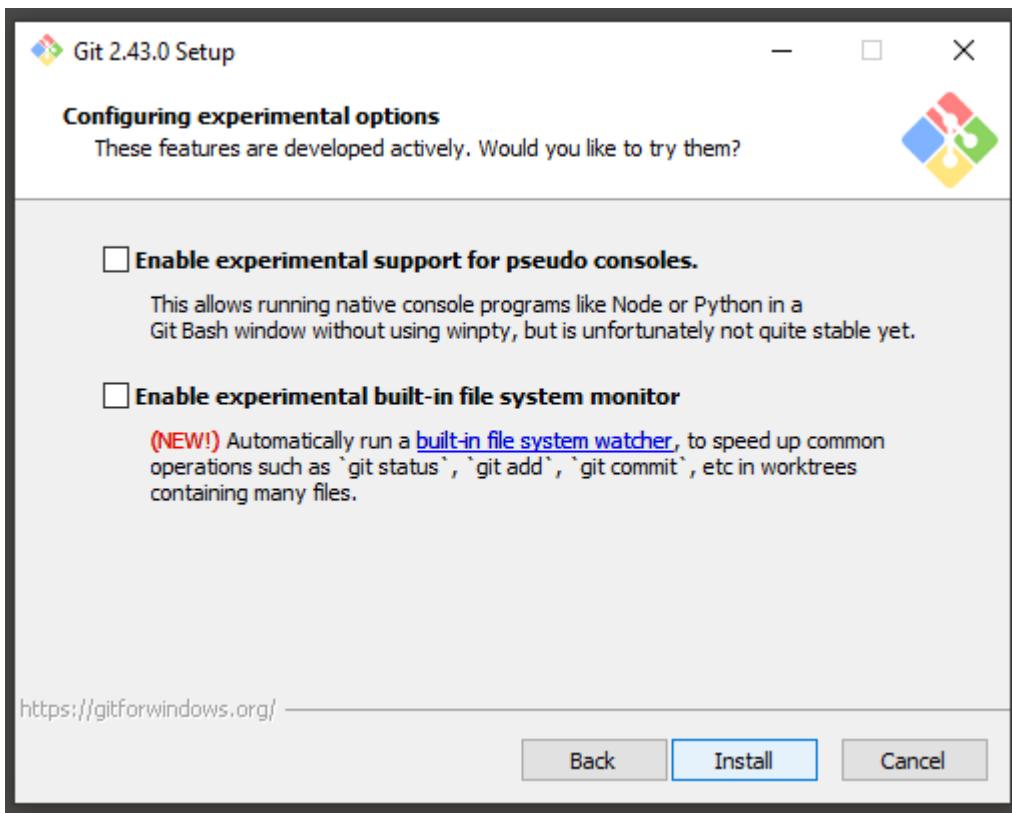
13. Konfigurasi pengaturan *git pull* dan *credential manager*



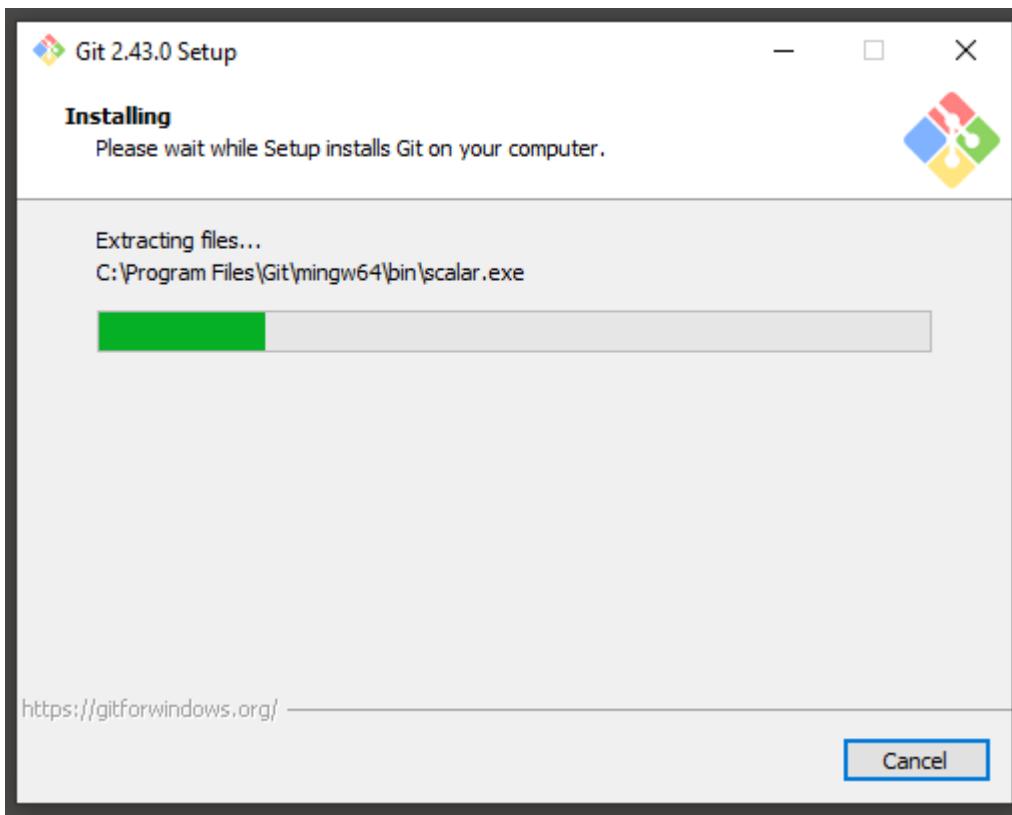


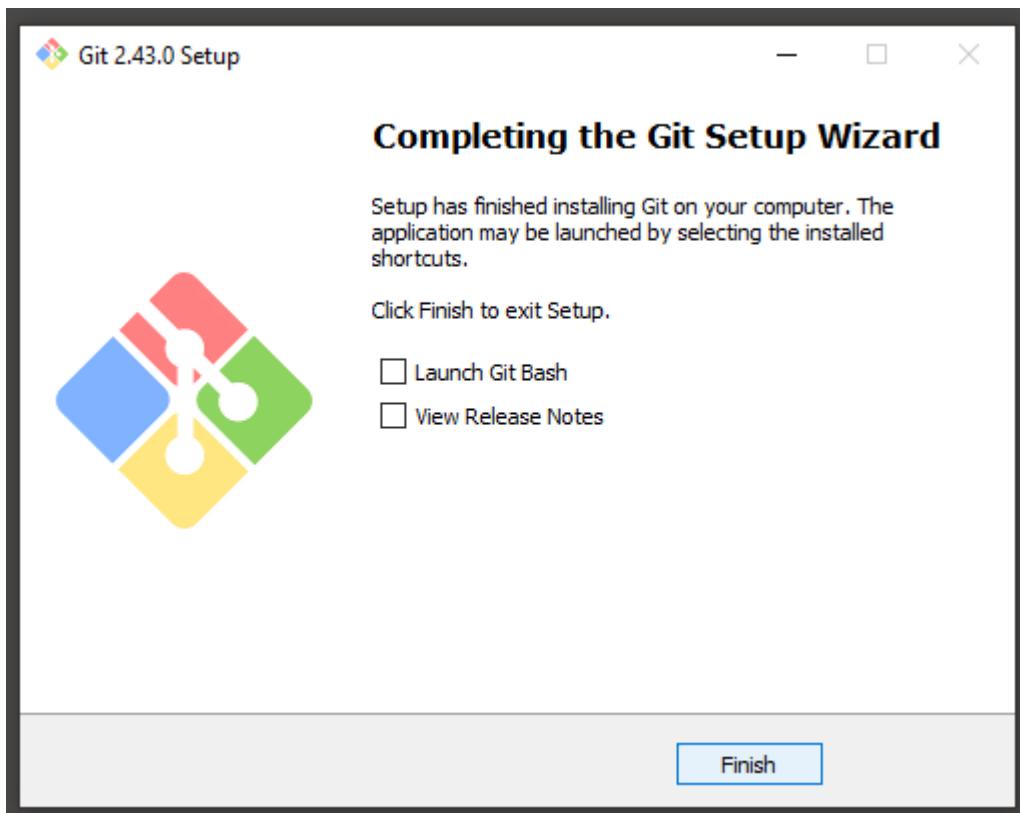
14. Atur pengaturan lainnya menjadi seperti ini.





15. Setelah itu, klik **Install** dan tunggu proses instalasi hingga selesai.





16. Cek apakah berhasil terinstall atau tidak dengan menjalankan perintah *git -version* pada command prompt atau cmd

A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following text:

```
Microsoft Windows [Version 10.0.19044.3031]
(c) Microsoft Corporation. All rights reserved.

C:\Users\andikatulusp>git --version
git version 2.43.0.windows.1

C:\Users\andikatulusp>
```

The command "git --version" was run and returned the expected output, confirming the installation of Git version 2.43.0 on a Windows system.

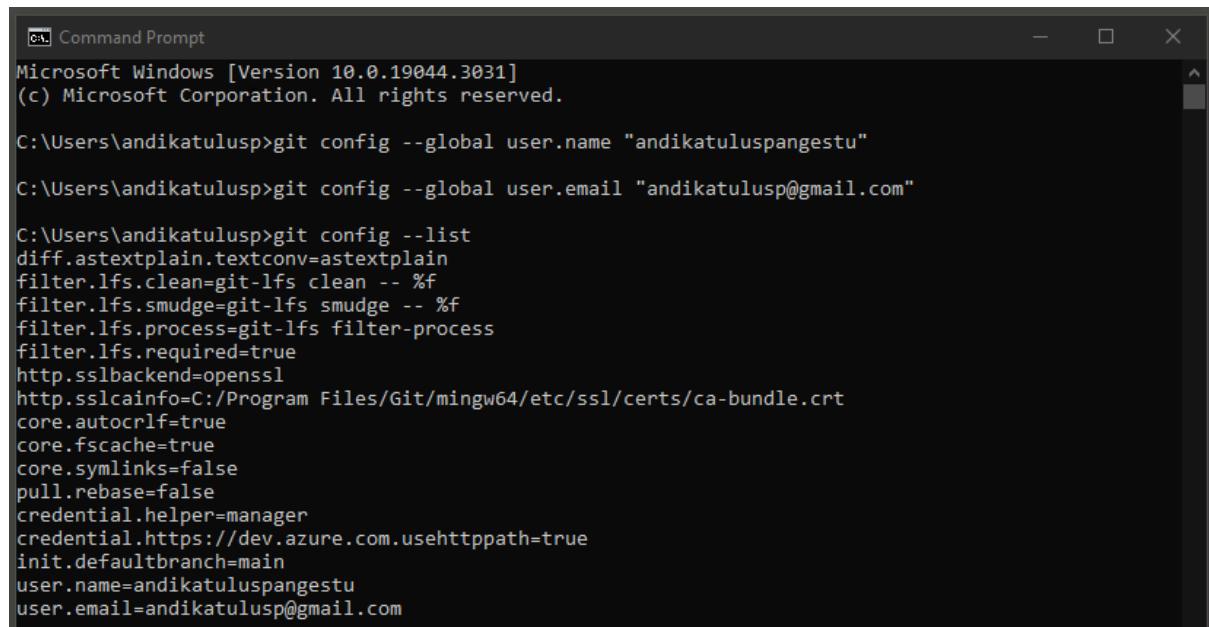
Konfigurasi GIT

Ketika kamu mengatur git untuk pertama kalinya di komputer lokal kamu, langkah pertama yang perlu dilakukan adalah melakukan konfigurasi awal. Terdapat beberapa elemen utama yang perlu yaitu tentukan nama pengguna kamu dan disarankan sesuai dengan nama pengguna yang akan terdaftar di github kemudian email aktif dan disarankan terhubung juga dengan akun github.

Jalankan perintah berikut ini melalui terminal/cmd/git bash dan diharapkan memiliki output yang sama seperti di gambar, yang menandakan kamu telah berhasil mengkonfigurasi git.

```
git config --global user.name "username-kamu"  
git config --global user.email nama_email@example.com  
git config --list
```

Lakukan seperti di gambar :

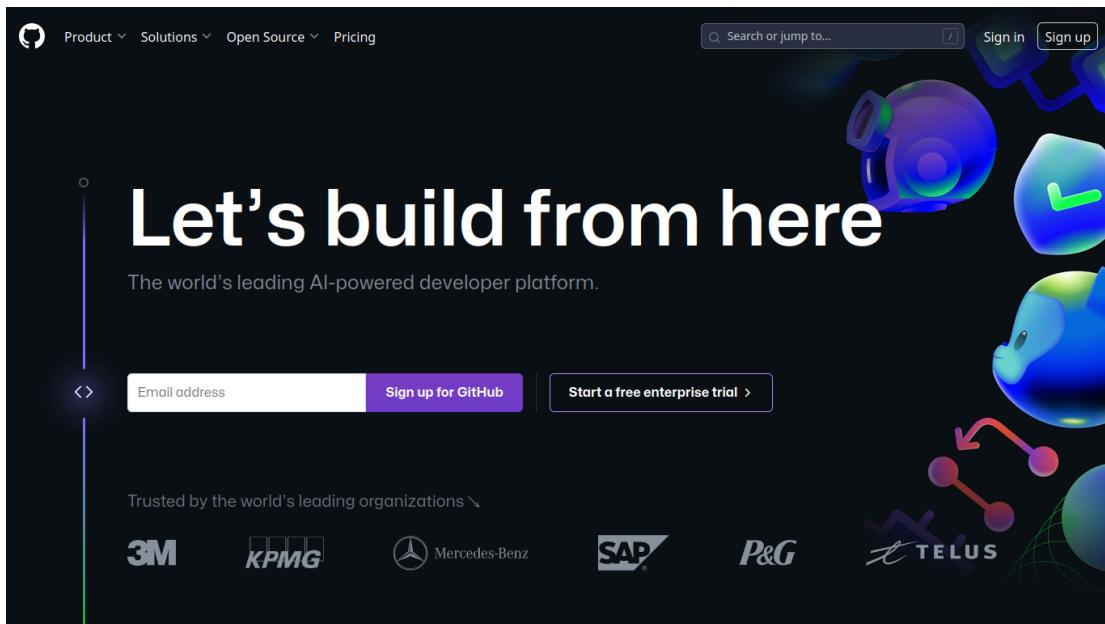


```
Command Prompt  
Microsoft Windows [Version 10.0.19044.3031]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\andikatulusp>git config --global user.name "andikatuluspangestu"  
C:\Users\andikatulusp>git config --global user.email "andikatulusp@gmail.com"  
  
C:\Users\andikatulusp>git 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/etc/ssl/certs/ca-bundle.crt  
core.autocrlf=true  
core.fscache=true  
core.symlinks=false  
pull.rebase=false  
credential.helper=manager  
credential.https://dev.azure.com.usehttppath=true  
init.defaultbranch=main  
user.name=andikatuluspangestu  
user.email=andikatulusp@gmail.com
```

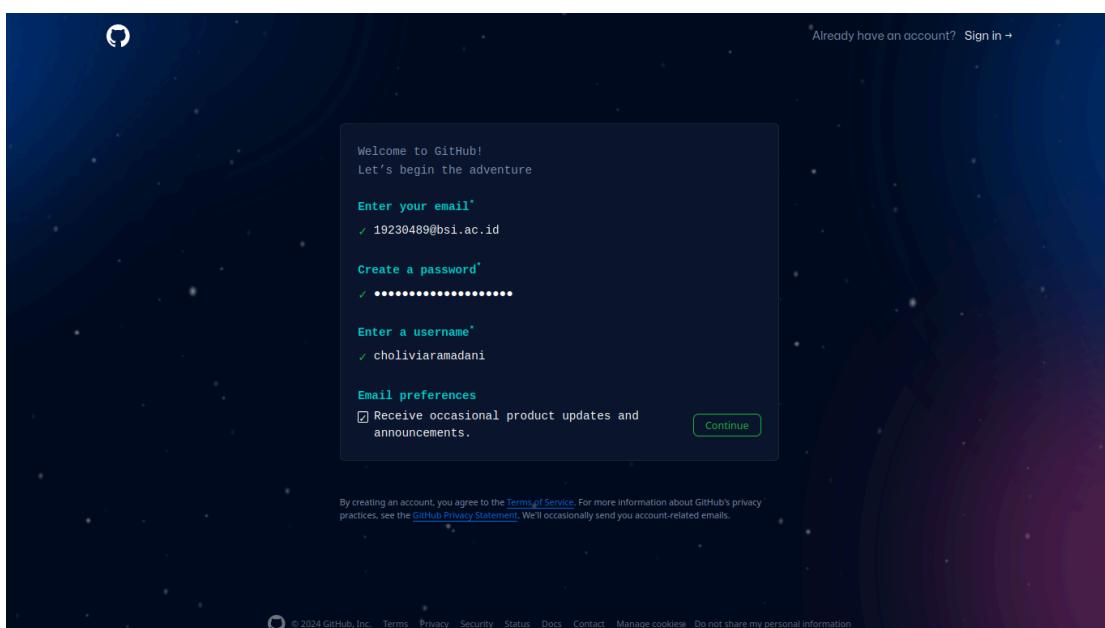
Membuat Akun GitHub

Pada bagian ini, kita akan membahas langkah-langkah yang diperlukan untuk membuat akun Github, langkah awal yang esensial sebelum memulai perjalanan Kamu dalam mengelola kode menggunakan Git dan Github.

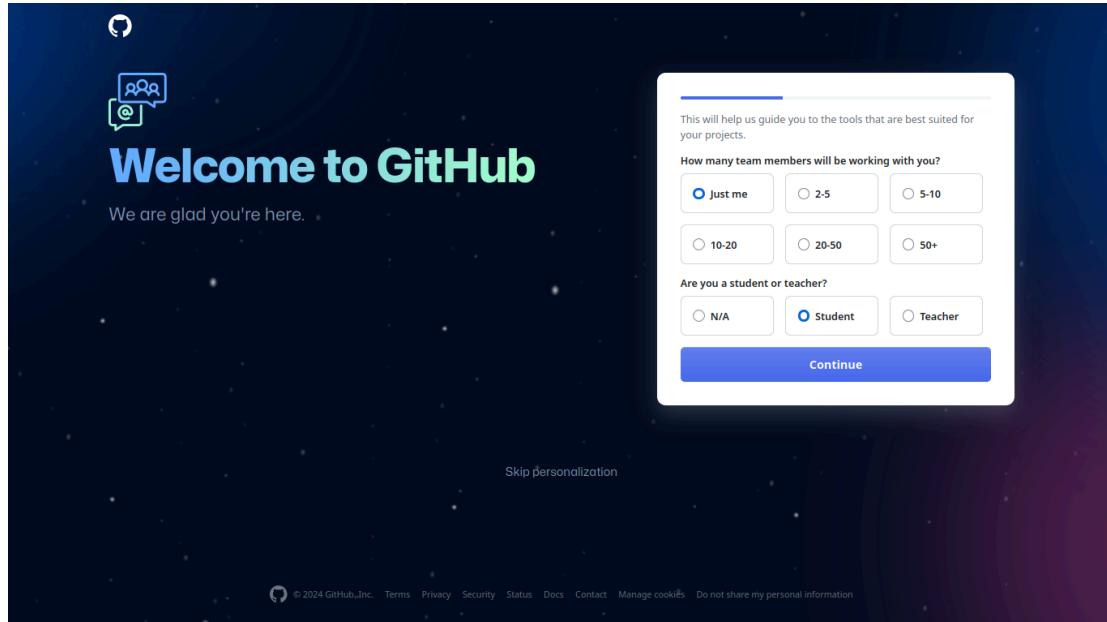
1. Mengakses situs resmi GitHub di <https://github.com>
2. Klik “Sign Up” untuk mendaftar.



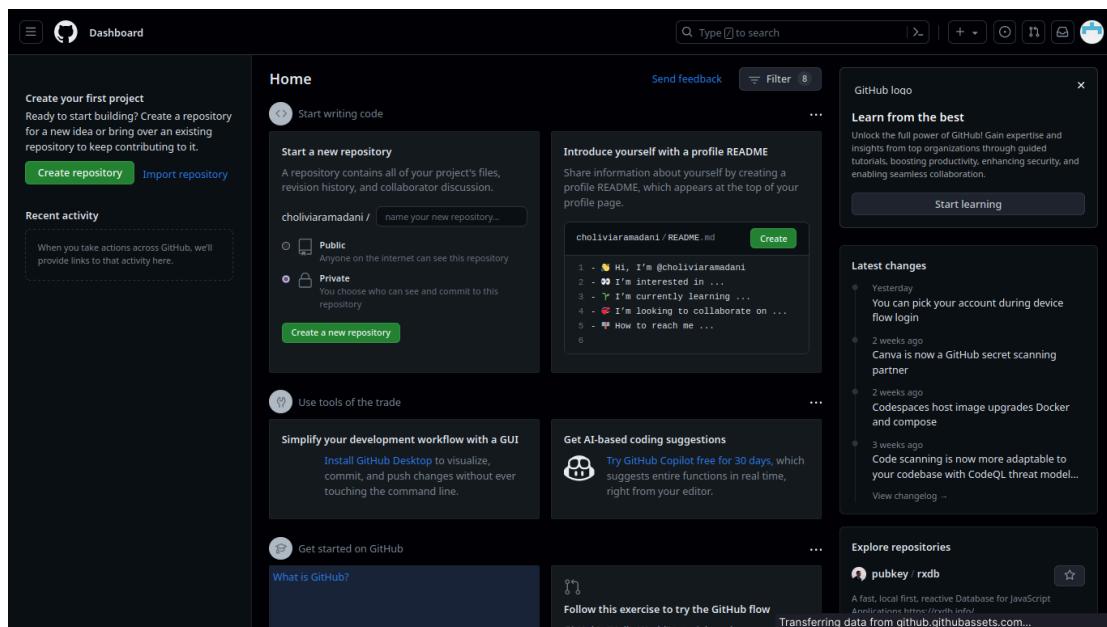
3. Masukan email aktif kamu, password, username yang mau di daftarkan!



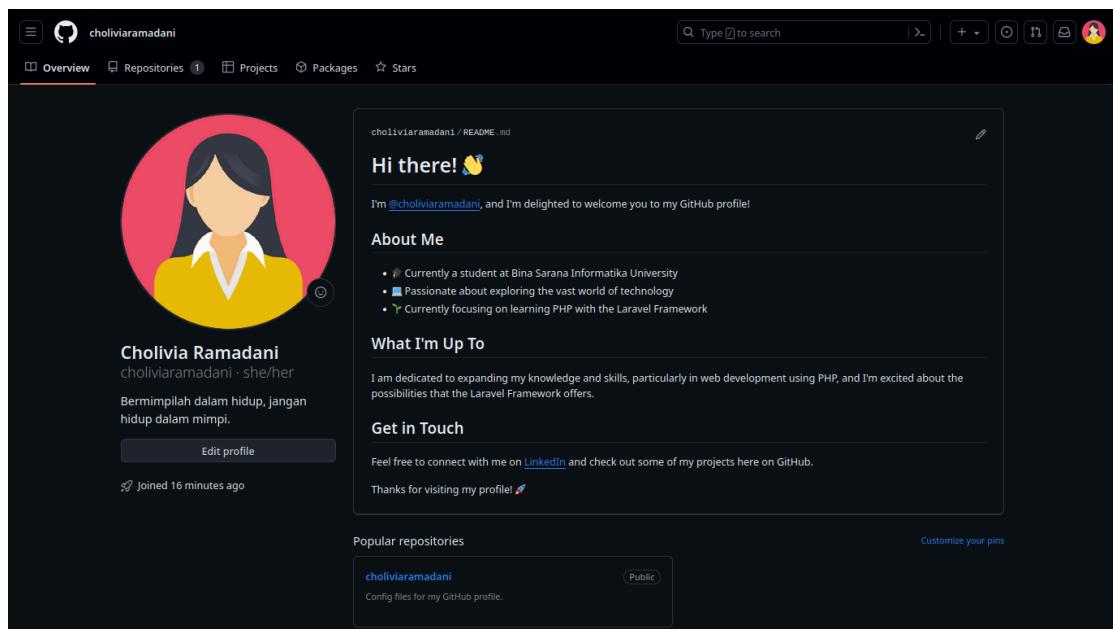
- Setelah itu kamu akan disuruh verifikasi captcha dan memasukan OTP dari email.
- Pada bagian ini kamu bisa klik “skip” kalau tidak mau melengkapi



- Jika sudah maka akan diarahkan ke halaman dashboard github kamu



7. Jangan lupa untuk melengkapi profil kamu agar terlihat lebih bagus

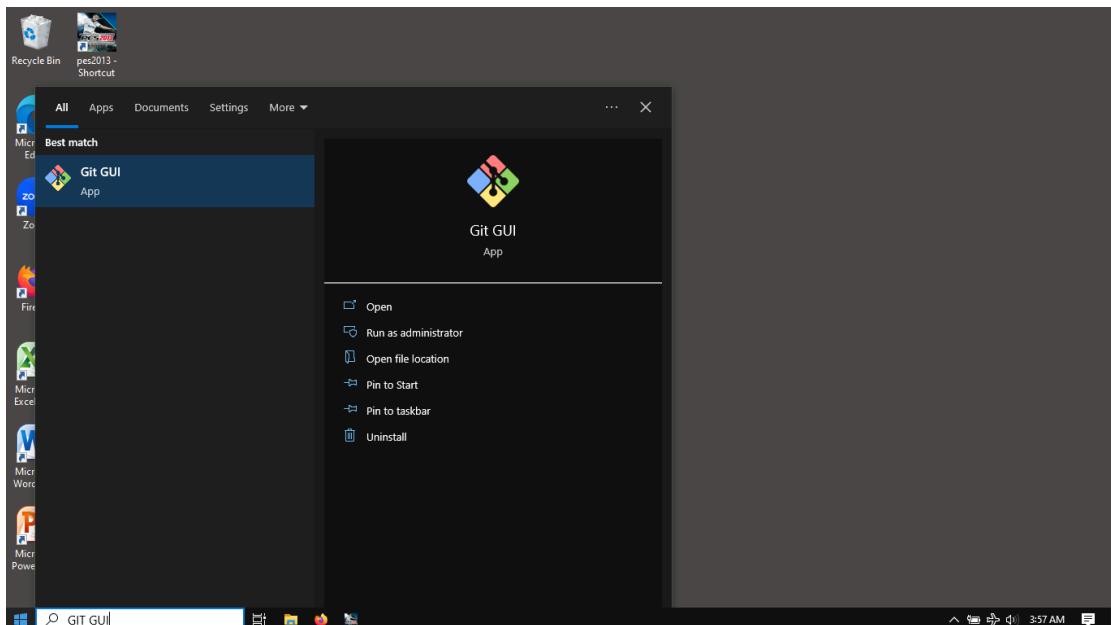


8. Setelah itu, selamat kamu telah membuat akun github baru 🎉😍

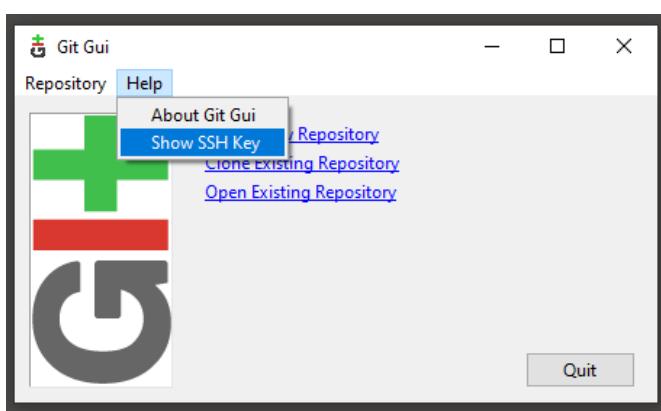
Menghubungkan Git ke GitHub

Pada bagian ini, kita akan membahas langkah-langkah menghubungkan Git dengan GitHub menggunakan metode Secure Shell (SSH). Penggunaan SSH tidak hanya meningkatkan keamanan tetapi juga mempermudah akses ke repository GitHub tanpa perlu memasukkan kata sandi setiap kali berinteraksi. Ikuti langkah-langkah di bawah untuk mengkonfigurasi koneksi Git Kamu ke GitHub menggunakan SSH.

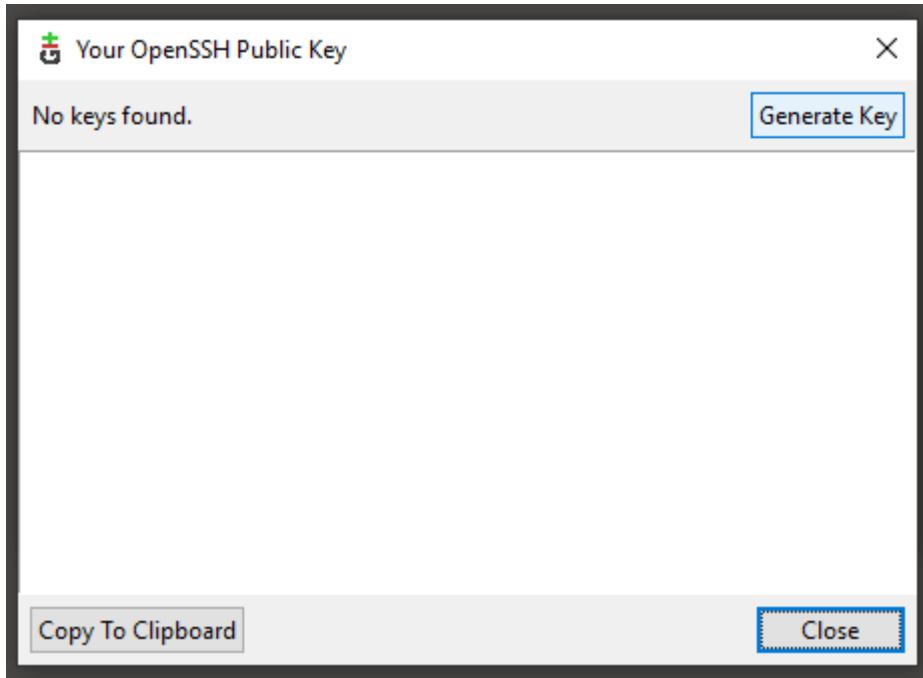
1. Buka GIT GUI di komputer kamu, bisa search aja di pencarian windows



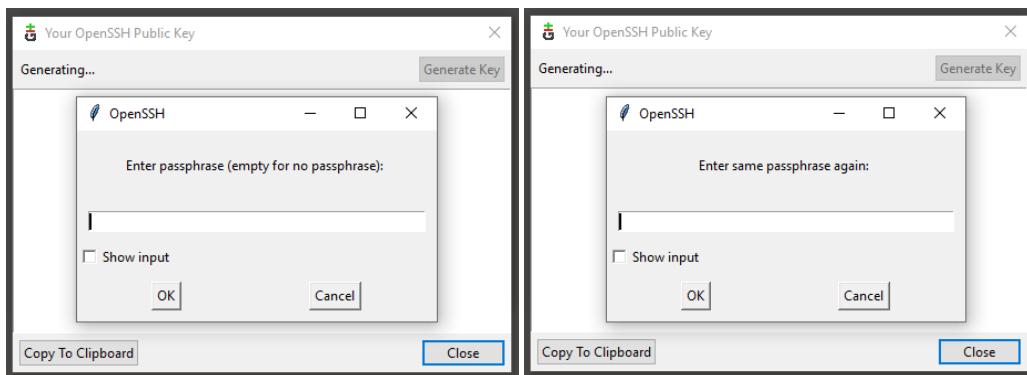
2. Setelah terbuka, klik bagian **help** lalu klik **Show SSH Key**



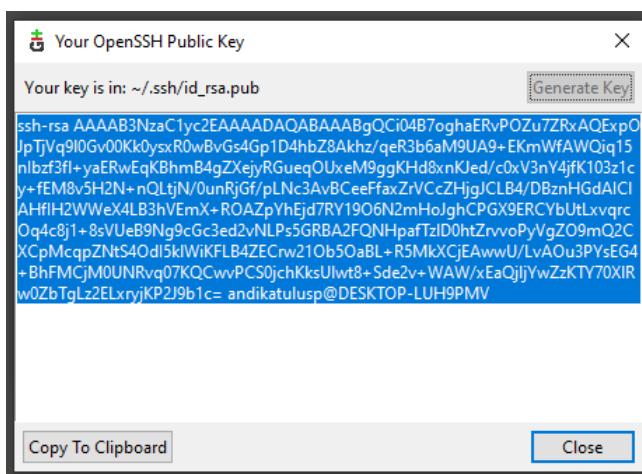
3. Klik tombol **Generate Key** untuk menghasilkan key baru



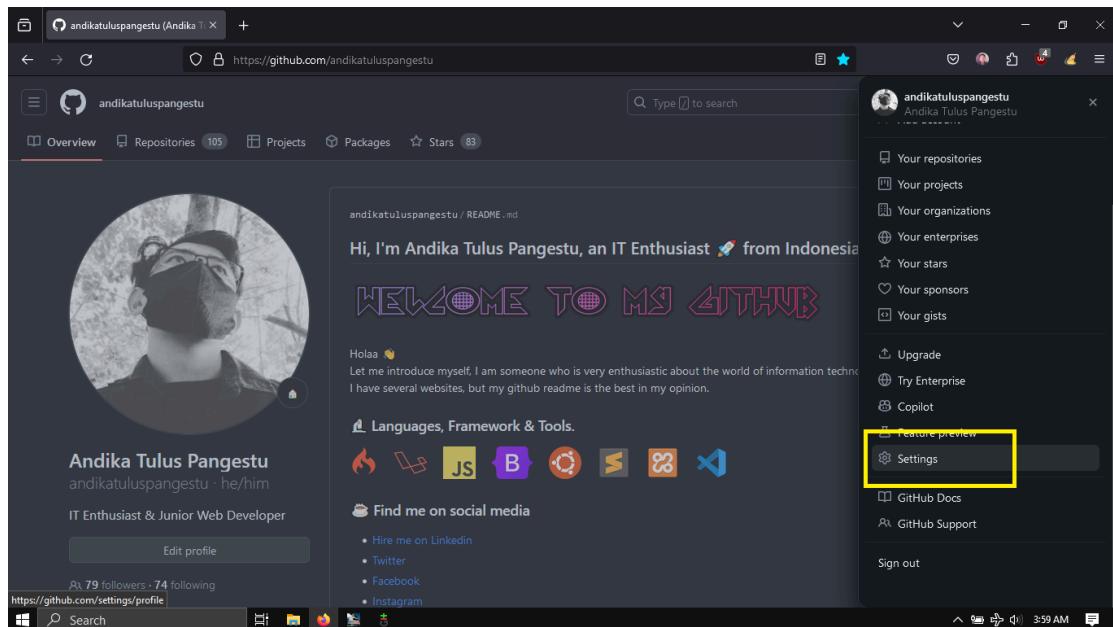
4. Jika disuruh memasukan **passphrase** kosongkan saja, langsung klik **ok**



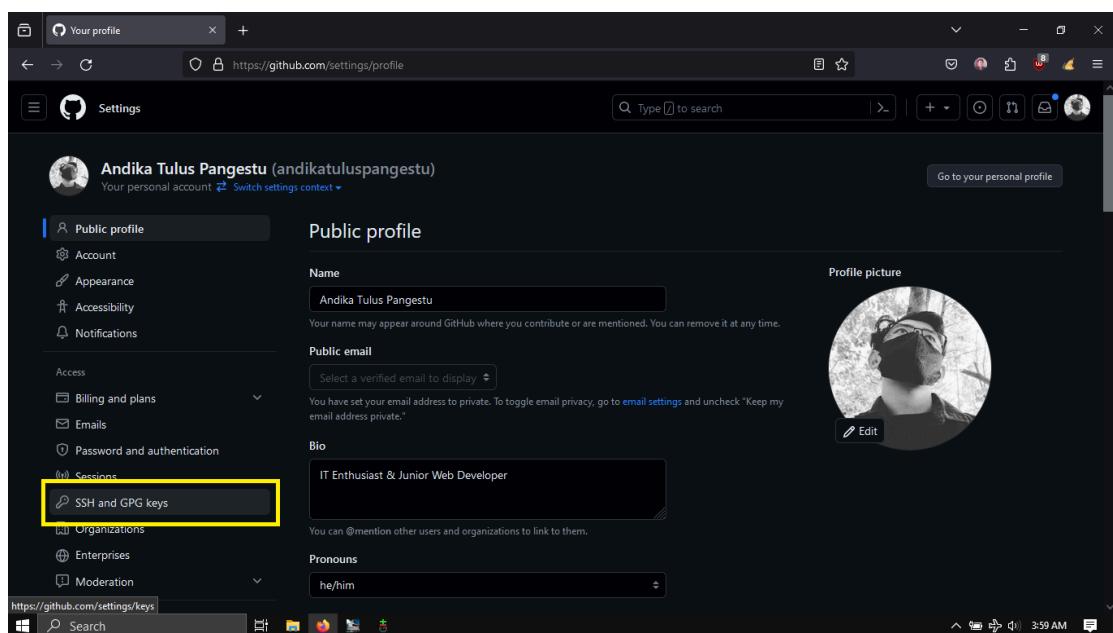
5. Setelah itu copy key (CTRL + C) atau tekan tombol **Copy To Clipboard**



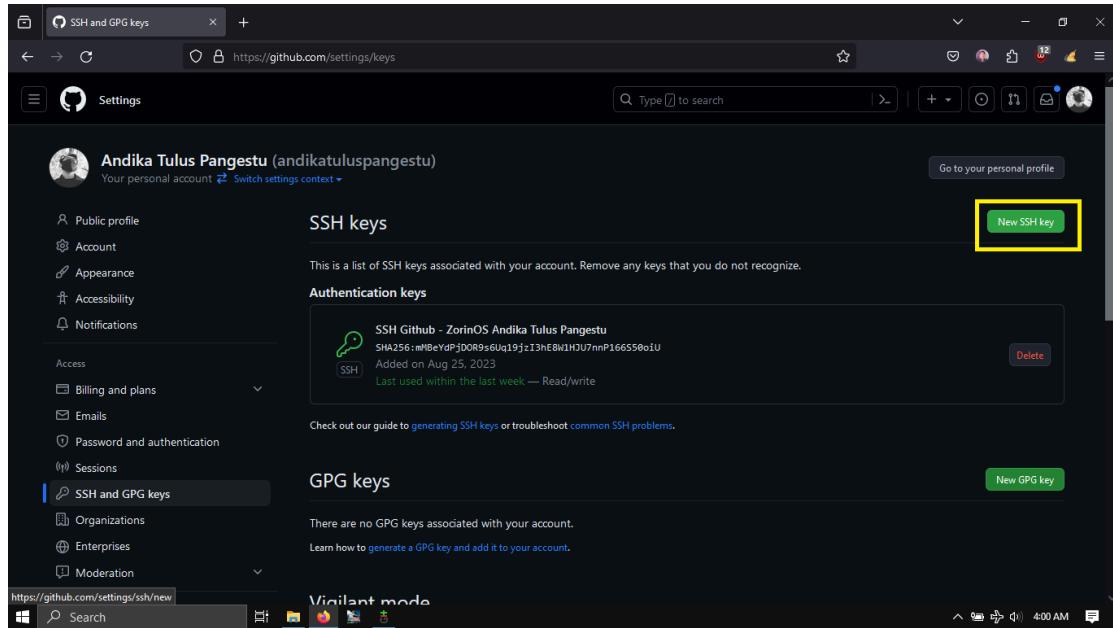
6. Setelah di copy, silahkan masuk ke akun github kamu atau akun github yang akan dihubungkan dengan repository lokal (komputer kamu)
7. Setelah masuk, klik avatar/foto profil github kamu di bagian pojok kanan atas, kemudian pilih menu settings



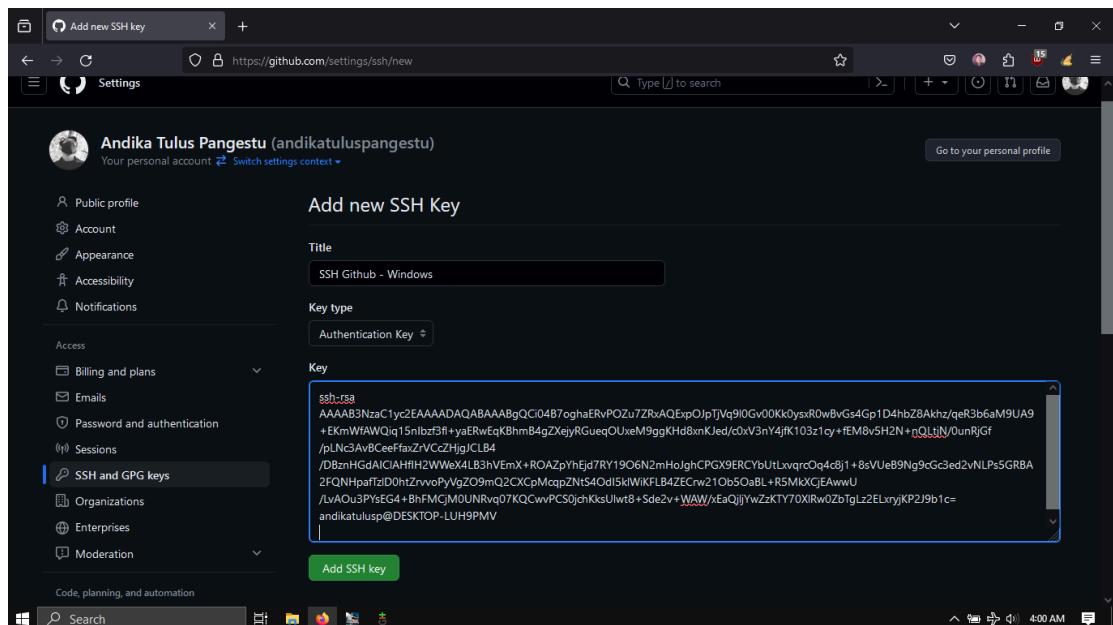
8. Klik menu ***SSH & GPG Keys***



9. Lalu, klik tombol *New SSH Key*



10. Paste *SSH Key* yang telah kamu copy sebelumnya ke halaman ini



11. Jangan lupa kita cek apakah sudah terhubung atau tidak, maka dari itu kita gunakan perintah berikut ini di CMD atau Terminal

```
ssh -T git@github.com
```

12. Jika hasilnya seperti ini, maka sudah terhubung.



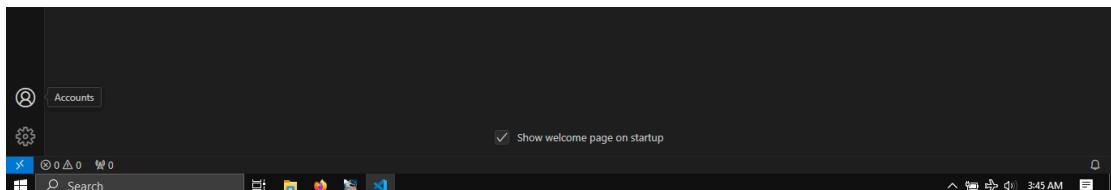
```
Command Prompt
Microsoft Windows [Version 10.0.19044.3031]
(c) Microsoft Corporation. All rights reserved.

C:\Users\andikatulusp>ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeI0trrVc98/R1BUFWu3/LiyKgUFQM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com,20.205.243.166' (ECDSA) to the list of known hosts.
Hi andikatuluspangestu! You've successfully authenticated, but GitHub does not provide shell access.
```

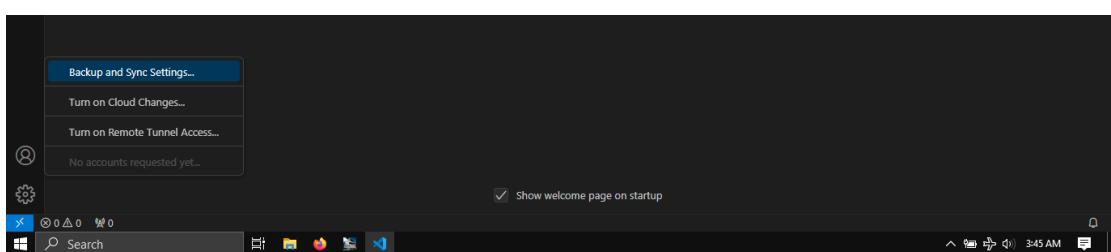
Menghubungkan VSCode ke GitHub

Pada bagian ini, saatnya bikin koneksi antara senjata andalan kita, Visual Studio Code (VSCode), dengan tempat kerennya buat nge-host proyek, yaitu GitHub! Dengan ini, kita bisa nyaman banget ngerjain proyek dan kolaborasi tanpa ribet. Jadi, di bab ini, kita bakal jelaskan cara nyambungin VS Code ke GitHub, dari konfigurasi dasar sampe trik-trik seru yang memudahkan hidup kita dalam kolaborasi.

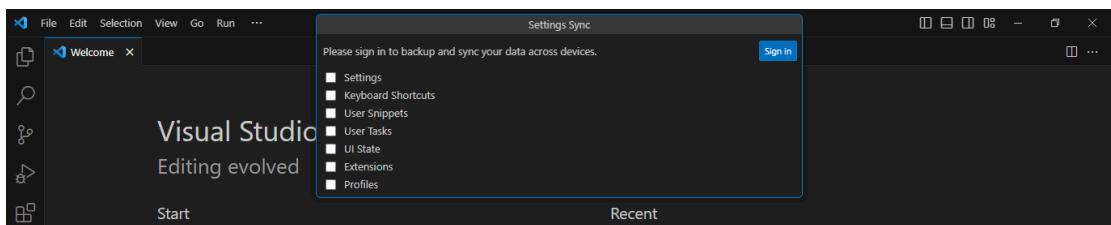
1. Pertama kita harus punya dulu akun github
2. Kedua, komputer kita harus sudah terinstall Visual Studio Code
3. Kalau kedua hal tersebut sudah terpenuhi, silahkan masuk ke akun github
4. Kemudian buka aplikasi visual studio code dan klik icon account di pojok kiri bawah layar



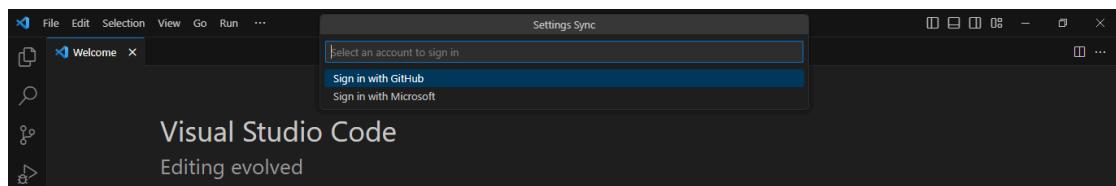
5. Klik menu ***Backup and Sync Settings***



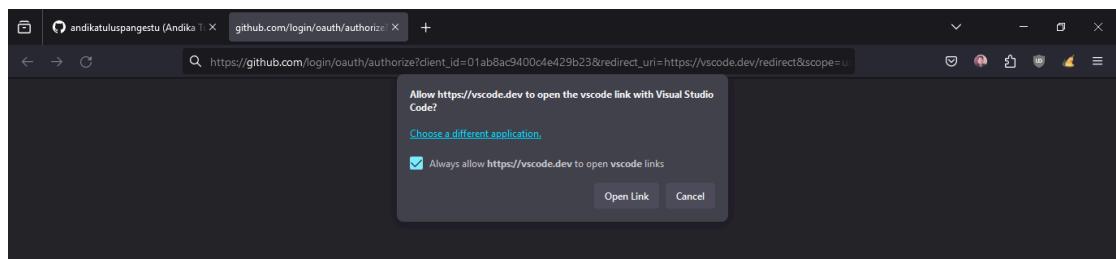
6. Lalu setelah muncul popup bisa klik tombol ***sign in***



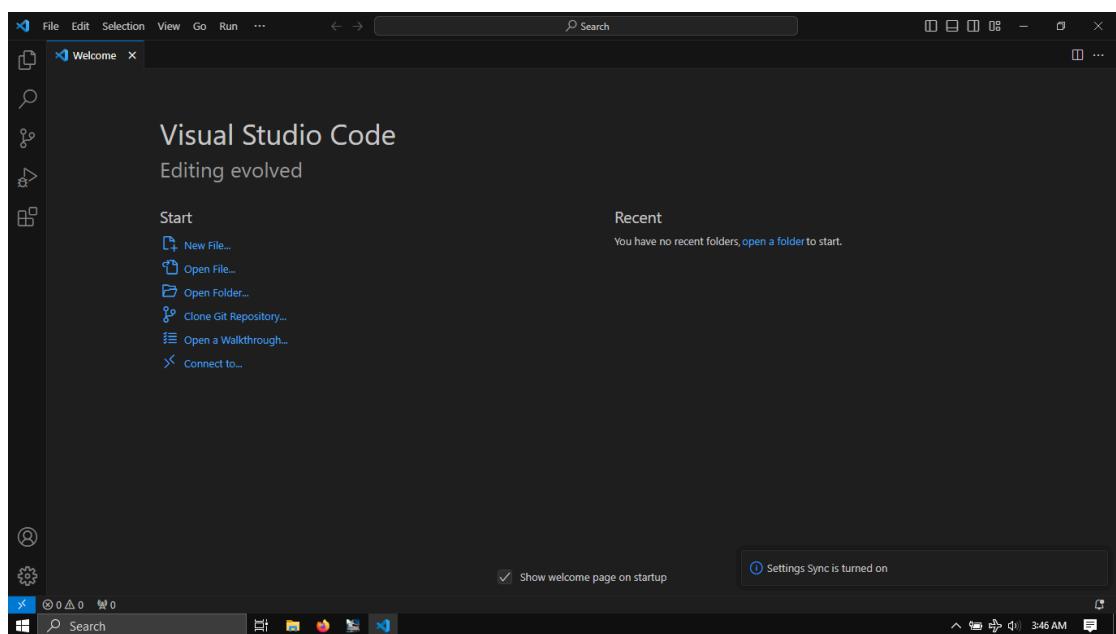
7. Kemudian pilih ***Sign in with Github***



8. Jangan lupa klik centang ***always allow*** dan klik ***open link***



9. Jika diarahkan kembali ke vscode dan terdapat notifikasi ***Settings Sync is turned on*** di pojok kanan bawah layar, maka telah terhubung.

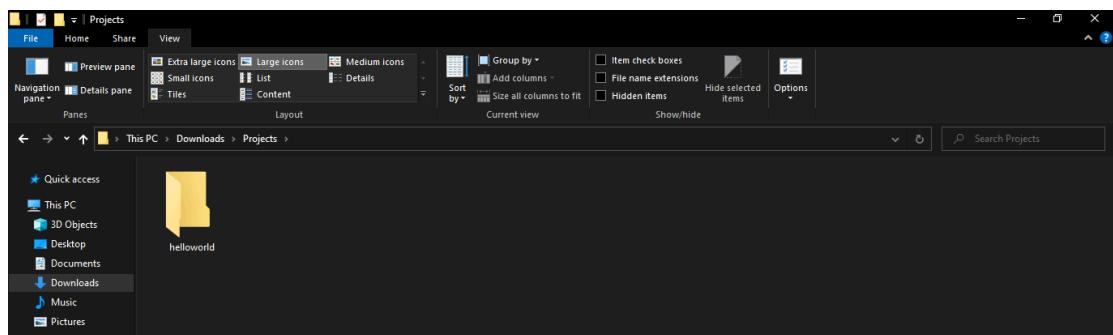


10. Selamat kamu telah berhasil menghubungkan VSCode dengan GitHub

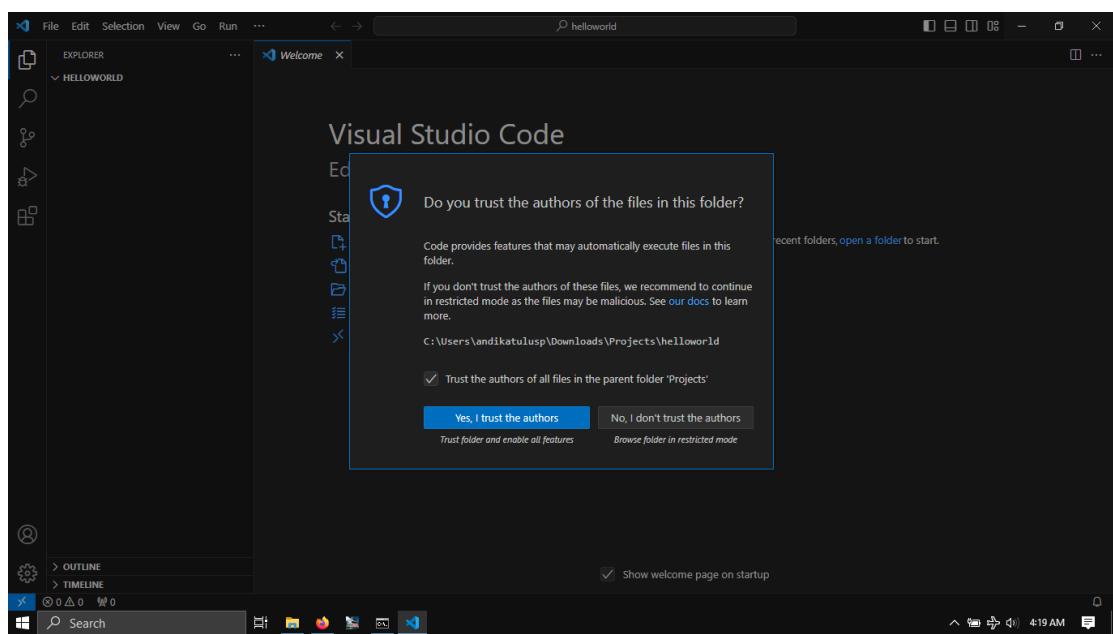
Push Project ke GitHub

Pada bagian ini, saatnya kita menggunakan kehebatan dari GIT dan Github untuk mengelola project kodingan kita, disini saya akan menuntun kamu benar-benar dari awal, intinya *push project* ialah menyimpan kodingan kita ke penyimpanan awan/cloud.

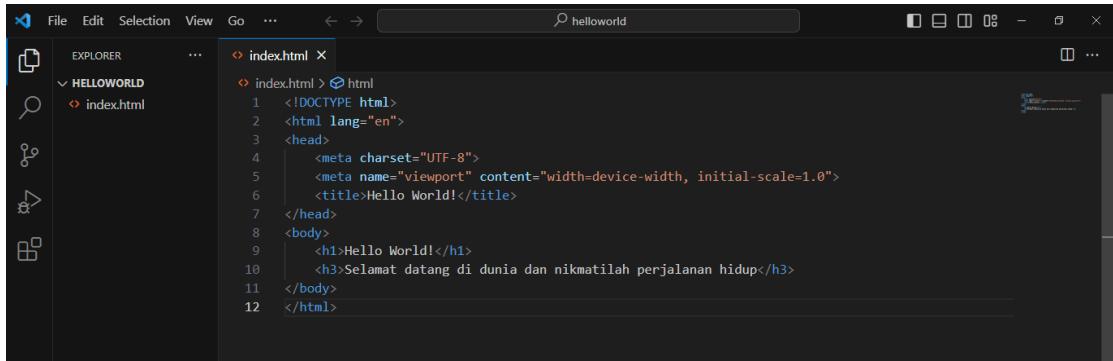
1. Buat sebuah project sederhana baru, disini saya akan membuat project **helloworld** yang berisi berkas *index.html*



2. Buka folder project di vscode dan jika ada popup **trust folder author**, maka klik saja tombol **Yes, i trust the authors**

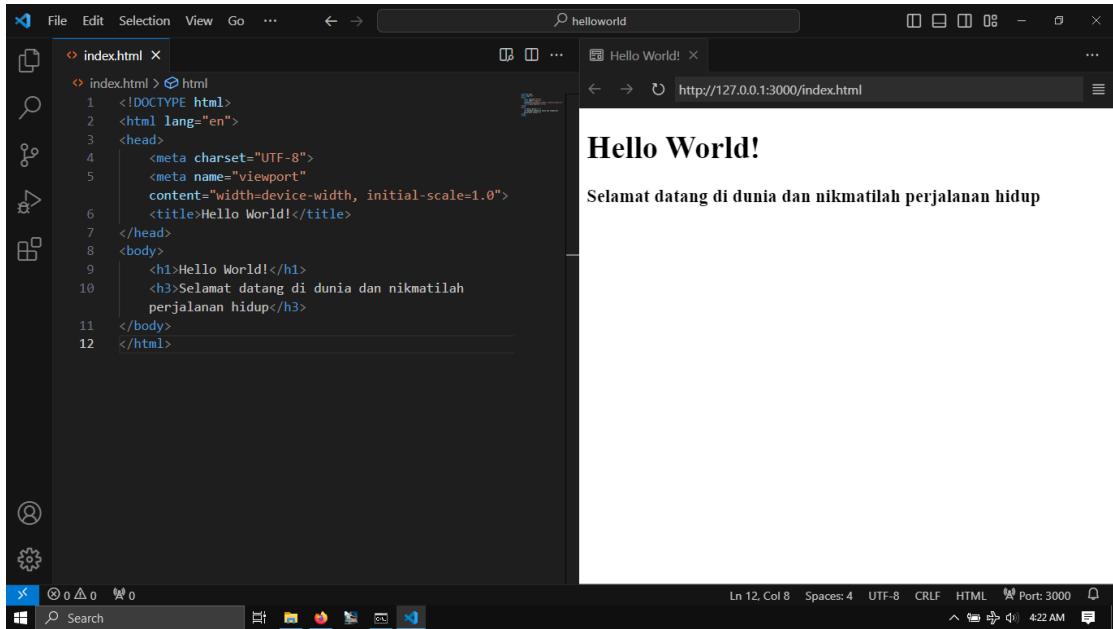


3. Kemudian, isi berkas *index.html* dengan kode sederhana seperti ini.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World!</title>
</head>
<body>
    <h1>Hello World!</h1>
    <h3>Selamat datang di dunia dan nikmatilah perjalanan hidup</h3>
</body>
</html>
```

4. Jika kita preview, maka kita hanya membuat sebuah project untuk menampilkan teks hello world saja.

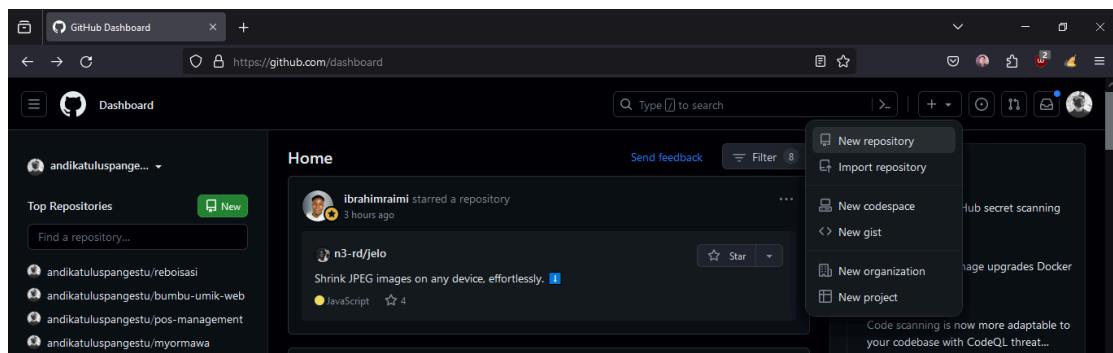


The screenshot shows the VS Code interface with the *index.html* file open in the editor. The browser preview panel on the right shows the rendered content of the HTML file:

Hello World!

Selamat datang di dunia dan nikmatilah perjalanan hidup

5. Setelah project sudah siap di push ke github, kita tinggal buat repository baru di github, dengan klik **New Repository**



6. Konfigurasi detail repository sebagai berikut.

The screenshot shows the GitHub interface for creating a new repository. At the top, there is a search bar and several navigation icons. The main section is titled "Create a new repository".

Repository template: No template selected.

Owner: andikatuluspangestu / **Repository name:** helloworld
A green checkmark indicates "helloworld is available."

Description (optional): Semua tentang yang ada di dunia, mari lakukan perjalanan hidup yang menarik.

Visibility: Public (Anyone on the internet can see this repository. You choose who can commit.)

Initialize this repository with:

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

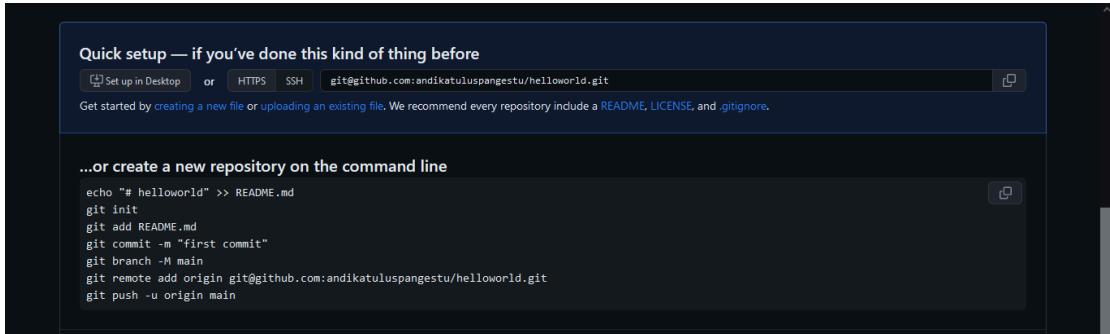
Add .gitignore: .gitignore template: None

Choose a license: License: None

A note at the bottom left says: ⓘ You are creating a public repository in your personal account.

A large green "Create repository" button is located at the bottom right.

- Setelah klik tombol **Create Repository** maka akan diarahkan ke halaman langkah-langkah push project ke github.



- Kemudian kita buka CMD/Terminal atau Git Bash, lalu jalankan perintah berikut ini.

Inisialisasi Git di Folder Project

```
git init
```

Pindahkan semua file project ke *staging area*⁶

```
git add .
```

Lakukan *commit*⁷ dengan disertai pesan commit atau pesan perubahan

```
git commit -m "First Commit"
```

Atur ke *branch*⁸ atau cabang mana project akan di push

```
git branch -M Main
```

⁶ Staging area berfungsi sebagai langkah penyaringan sebelum mengirim perubahan ke repository di GitHub. Ini memberikan kontrol tambahan terhadap perubahan yang akan di-commit.

⁷ Commit merupakan tindakan untuk menyimpan perubahan pada repositori Git. Setiap commit memiliki pesan yang menjelaskan perubahan yang dilakukan

⁸ Branch merupakan cabang yang akan menyimpan berkas project dengan fitur dan versi yang berbeda, ya istilahnya seperti folder didalam folder

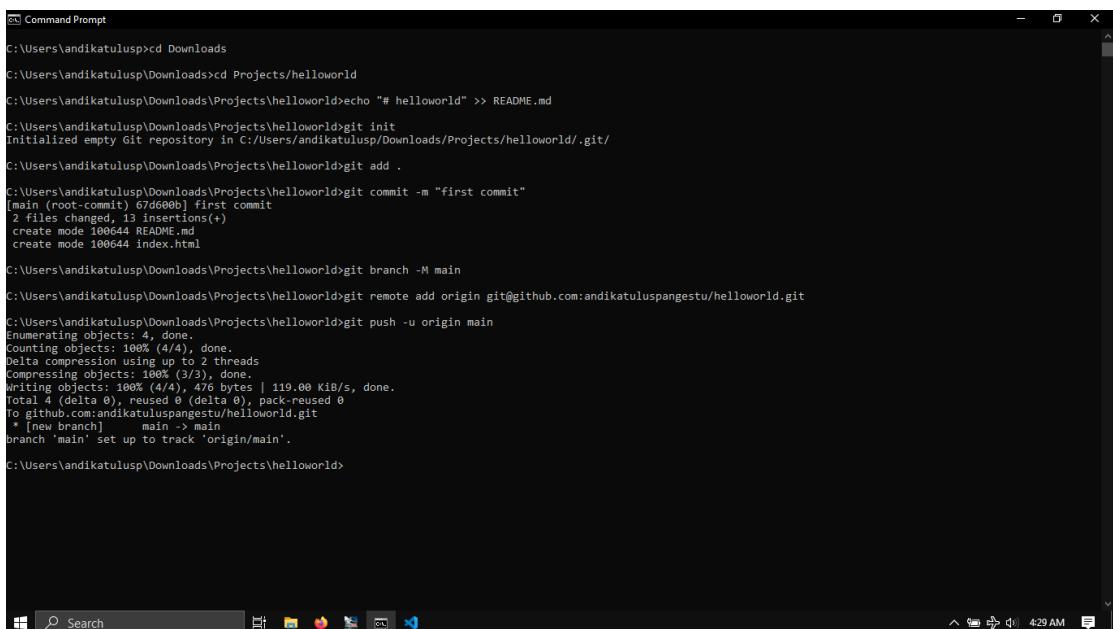
Menghubungkan folder project dengan repository github

```
git remote add origin git@github.com:link_git_kamu
```

Terakhir, lakukan push ke github

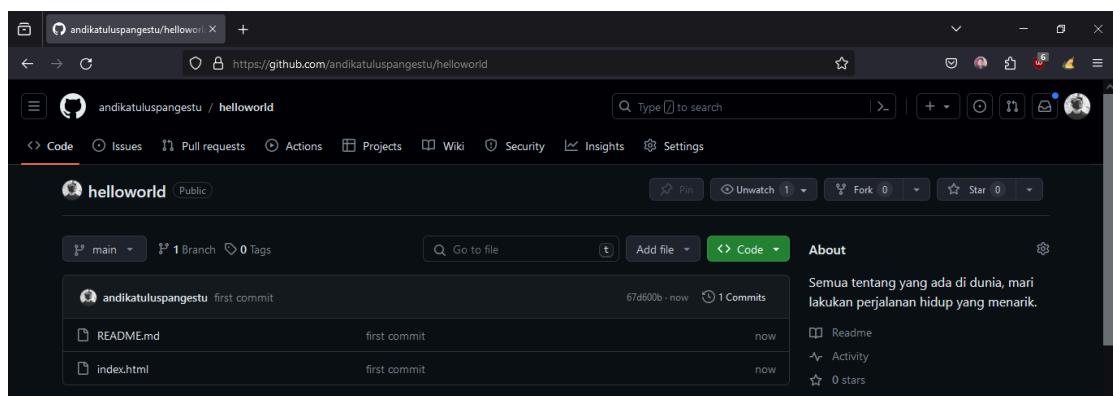
```
git push -u origin main
```

9. Kurang lebih seperti gambar berikut ini.



```
C:\Users\andikatulusp>cd Downloads
C:\Users\andikatulusp\Downloads>cd Projects/helloworld
C:\Users\andikatulusp\Downloads\Projects\helloworld>echo "# helloworld" >> README.md
C:\Users\andikatulusp\Downloads\Projects\helloworld>git init
Initialized empty Git repository in C:/Users/andikatulusp/Downloads/Projects/helloworld/.git/
C:\Users\andikatulusp\Downloads\Projects\helloworld>git add .
C:\Users\andikatulusp\Downloads\Projects\helloworld>git commit -m "first commit"
[main (root-commit) 67d600b] first commit
 2 files changed, 13 insertions(+)
 create mode 100644 README.md
 create mode 100644 index.html
C:\Users\andikatulusp\Downloads\Projects\helloworld>git branch -M main
C:\Users\andikatulusp\Downloads\Projects\helloworld>git remote add origin git@github.com:andikatuluspangestu/helloworld.git
C:\Users\andikatulusp\Downloads\Projects\helloworld>git push -u origin main
Enumerating objects: 4, done.
Counting objects: 4, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:andikatuluspangestu/helloworld.git
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.
C:\Users\andikatulusp\Downloads\Projects\helloworld>
```

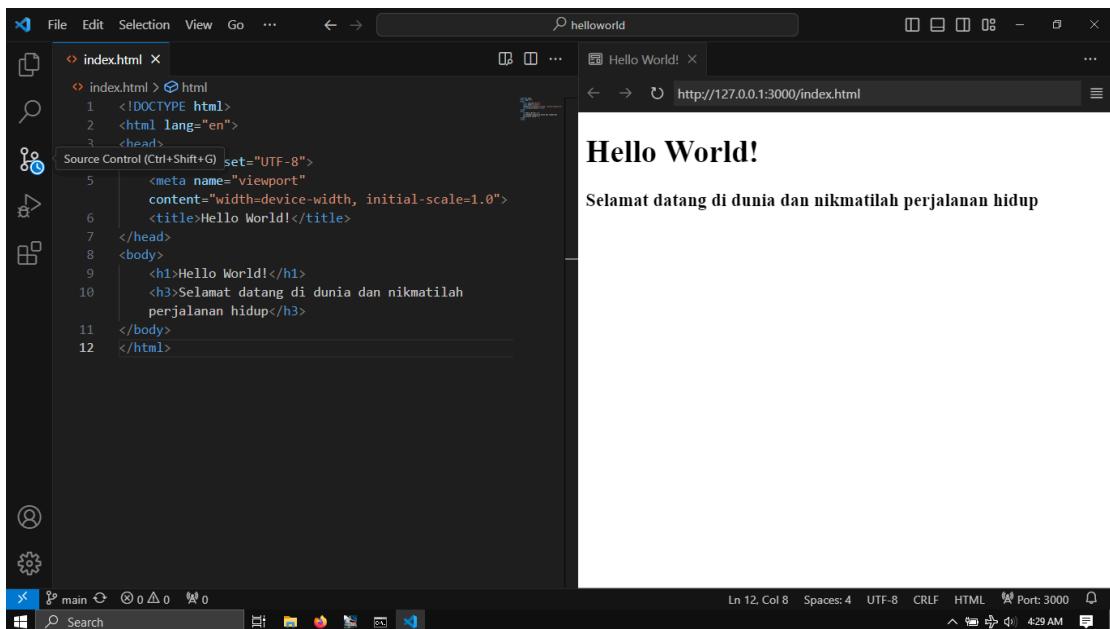
10. Kita cek, apakah kode kita sudah terupload atau belum dengan membuka link repository github kamu



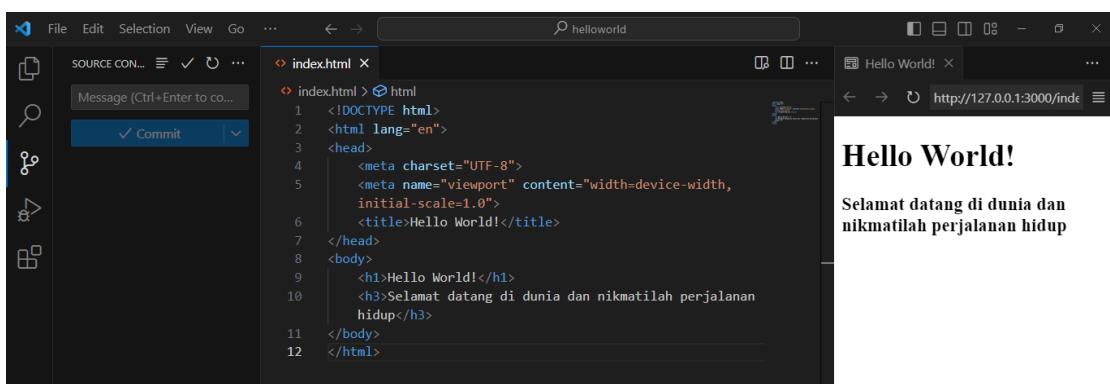
Membuat Perubahan Project

Pada bagian ini, kita akan mencoba membuat perubahan pada kodingan kita di komputer lokal, lalu menyimpan perubahan itu di Github melalui GIT.

1. Sekarang kita akan fokus untuk manajemen project dengan Git melalui VSCode, buka project kamu di VSCode dan lihat apakah di icon source control terdapat notifikasi jam?



2. Setelah kita klik icon source control, ternyata tidak terjadi apa-apa. Hal itu karena git memeriksa bahwa tidak ada perubahan apapun yang terjadi atau yang perlu disimpan.



3. Kita akan coba membuat perubahan di kodingan kita, dengan menambahkan sintaks html `<hr>` atau garis horizontal. Dan terlihat perubahannya dalam preview html.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World!</title>
</head>
<body>
    <h1>Hello World!</h1>
    <hr>
    <h3>Selamat datang di dunia dan nikmatilah perjalanan hidup</h3>
</body>
</html>

```

Lihat di sisi kiri sidebar, pada bagian **changes** sekarang sudah terisi berkas *index.html* yang artinya berkas-berkas tersebut telah mengalami perubahan.

4. Namun perubahan itu hanya disimpan di komputer kita, kodingan yang ada di github belum kita ubah seperti yang terbaru saat ini.
5. Maka dari itu, kita isi pesan perubahannya dan klik tombol **Commit** untuk menyimpan perubahan di github.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World!</title>
</head>
<body>
    <h1>Hello World!</h1>
    <hr>
    <h3>Selamat datang di dunia dan nikmatilah perjalanan hidup</h3>
</body>
</html>

```

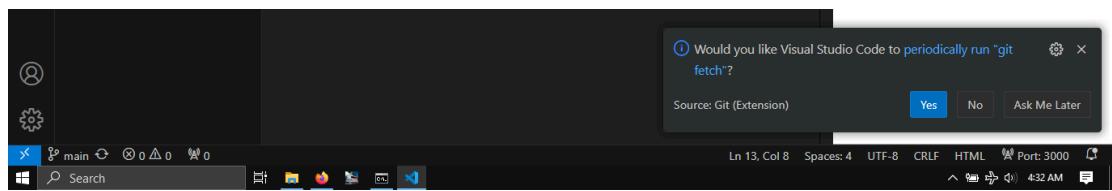
6. Jika ada popup seperti ini, klik saya “Yes”

A screenshot of Visual Studio Code showing a commit dialog. The dialog box is titled "Visual Studio Code" and contains the message: "There are no staged changes to commit. Would you like to stage all your changes and commit them directly?". It has four buttons: "Yes", "Always", "Never", and "Cancel". The "Yes" button is highlighted. The background shows an open file named "index.html" with some code and a browser preview window showing the text "Hello World!".

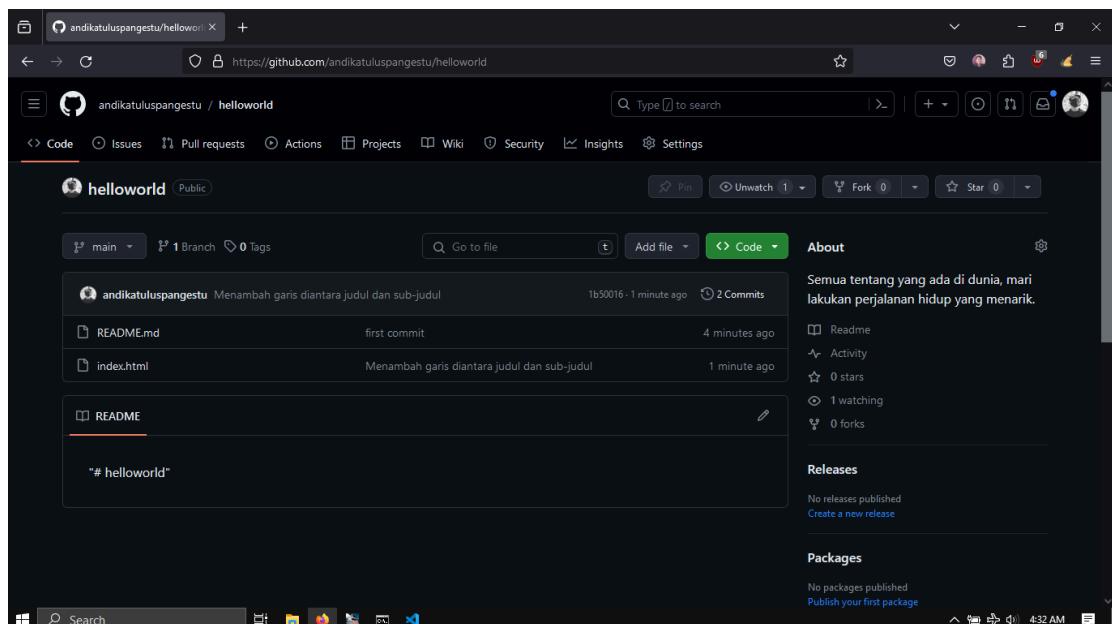
7. Setelah itu tombol **Commit** berubah menjadi **Sync Changes** artinya ada perubahan yang belum di push (↑) dan jika (↓) ada perubahan yang belum di pull atau diambil. Dan klik tombol tersebut.

A screenshot of Visual Studio Code showing a sync changes dialog. The dialog box is titled "Sync Changes" and contains the message: "1 ↑". It has three buttons: "Sync", "Always", and "Never". The "Sync" button is highlighted. The background shows an open file named "index.html" with some code and a browser preview window showing the text "Hello World!".

8. Maka akan muncul notifikasi seperti ini, klik **yes** saja.



9. Setelah itu, kita cek kembali apakah perubahan sudah di push di github atau belum dengan cara membuka link repository github kita. Jika sudah maka akan muncul pesan commit yang kita ketik tadi di VSCode.

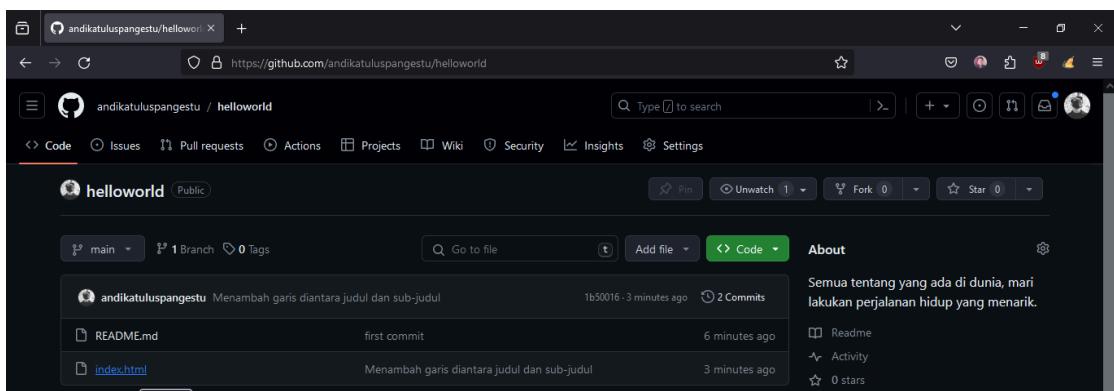


10. Selesai, kamu bisa membuat perubahan lainnya dan jangan lupa push ke Github

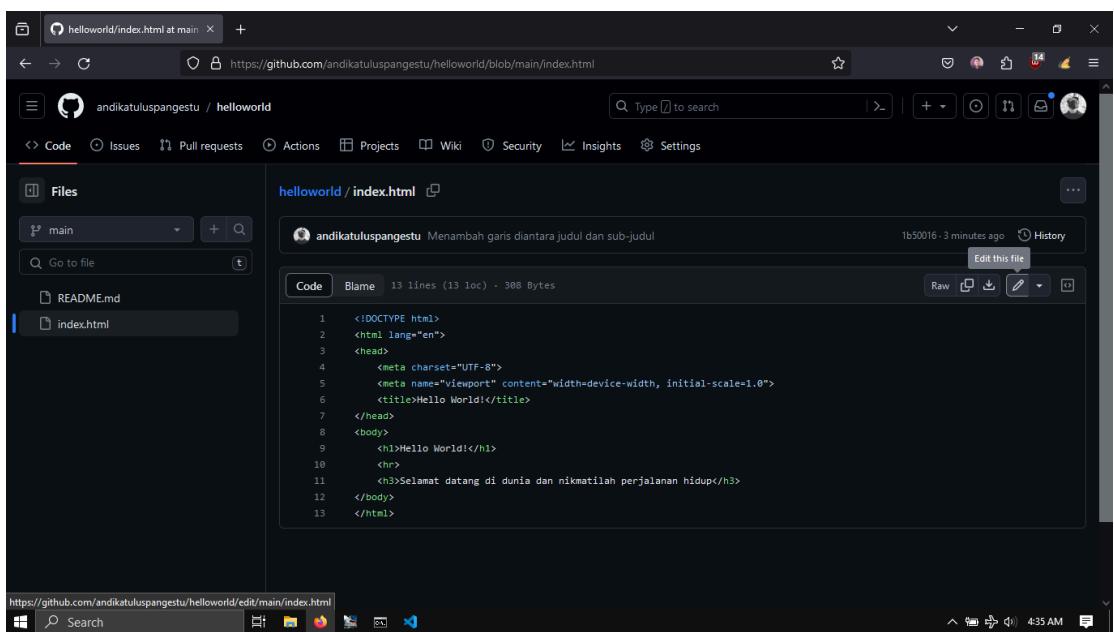
Mengambil Perubahan dari GitHub

Pada bagian ini, kita akan mencoba mengambil perubahan pada kodingan kita yang telah diubah melalui github (bukan dari komputer lokal kita), lalu menyimpan perubahan itu di Komputer lokal kita melalui GIT.

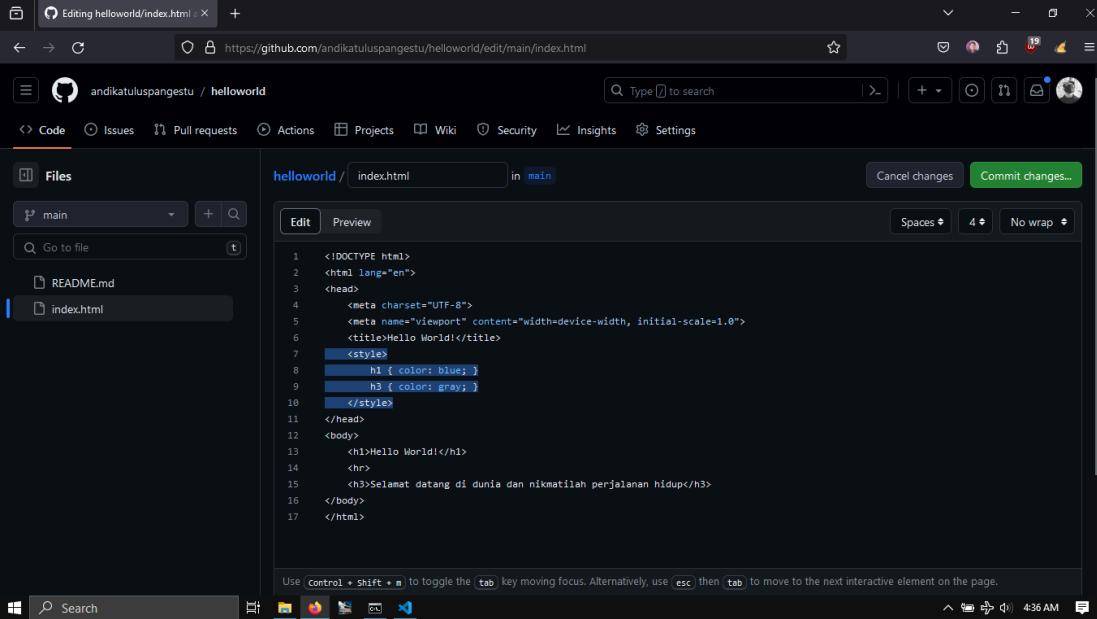
1. Membuat perubahan kodingan langsung dari Repository Github
2. Saya akan mencoba membuka *index.html*



3. Maka akan muncul editor bawaan github



4. Saya mencoba menambahkan style warna teks heading dengan CSS

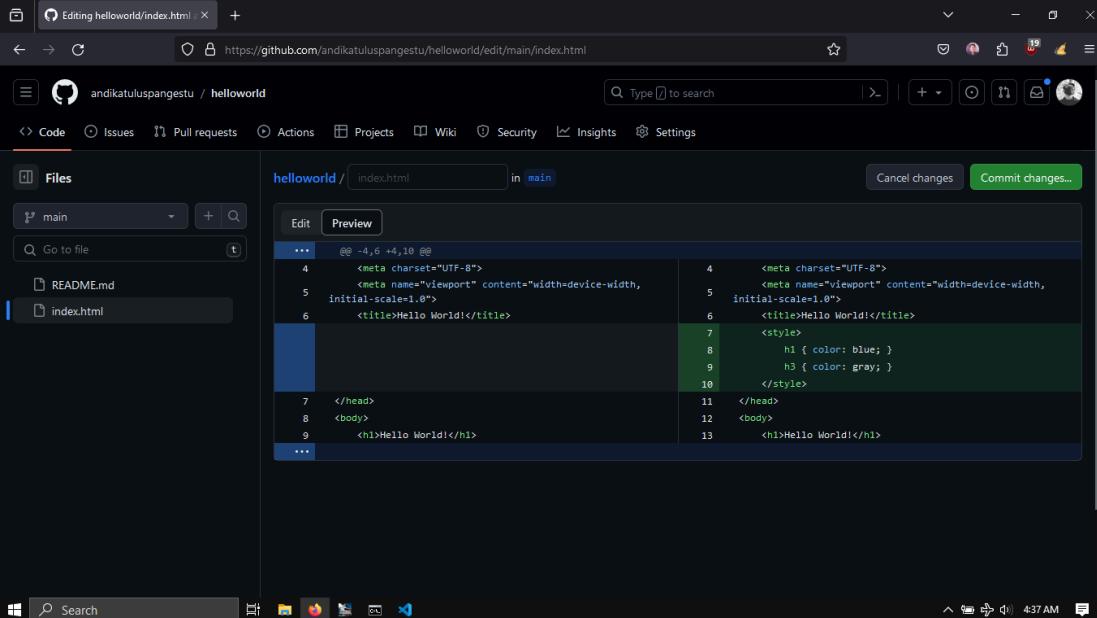


The screenshot shows a GitHub repository named "helloworld". The "index.html" file is open in the code editor. The code contains the following HTML and CSS:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Hello World!</title>
<style>
h1 { color: blue; }
h3 { color: gray; }
</style>
</head>
<body>
<h1>Hello World!</h1>
<hr>
<h3>Selamat datang di dunia dan nikmatilah perjalanan hidup</h3>
</body>
</html>
```

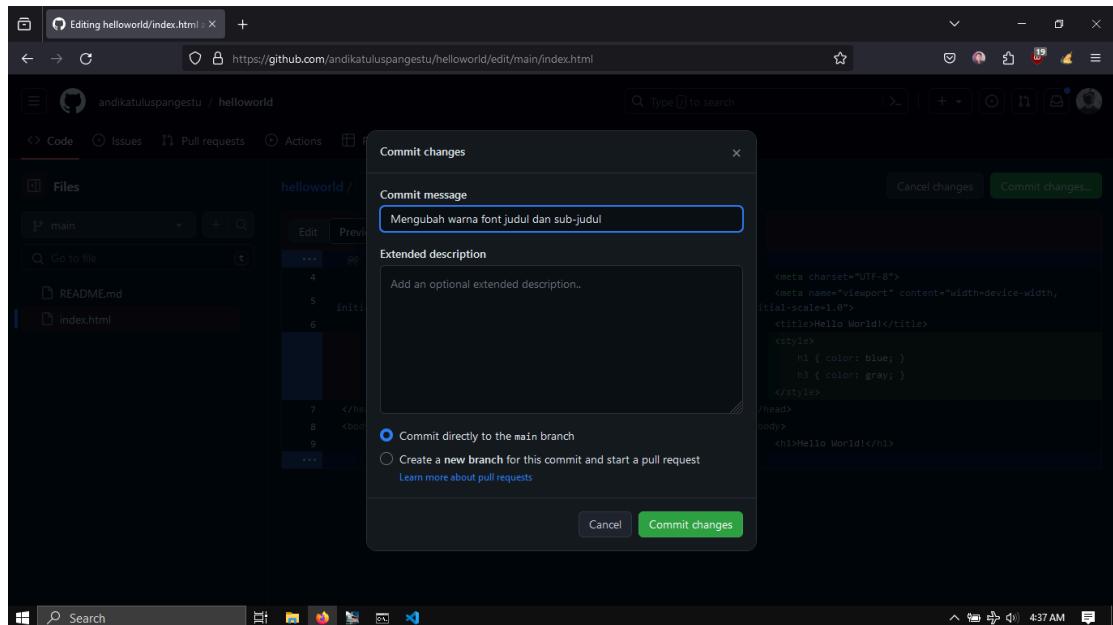
The "Preview" tab is selected, showing the rendered HTML with blue "Hello World!" and gray "Selamat datang di dunia dan nikmatilah perjalanan hidup".

5. Klik **Preview** untuk melihat perubahan kode yang lebih jelas.

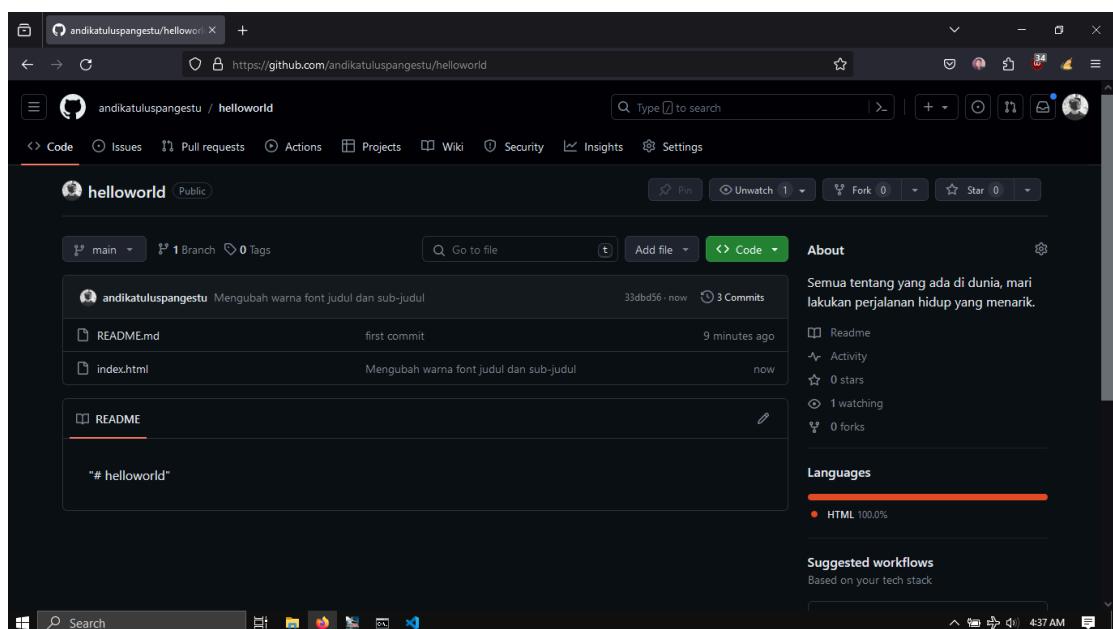


The screenshot shows the same GitHub repository and "index.html" file. The "Preview" tab is still selected, but now the CSS section is highlighted with a green background. The rendered output remains the same as in the previous screenshot.

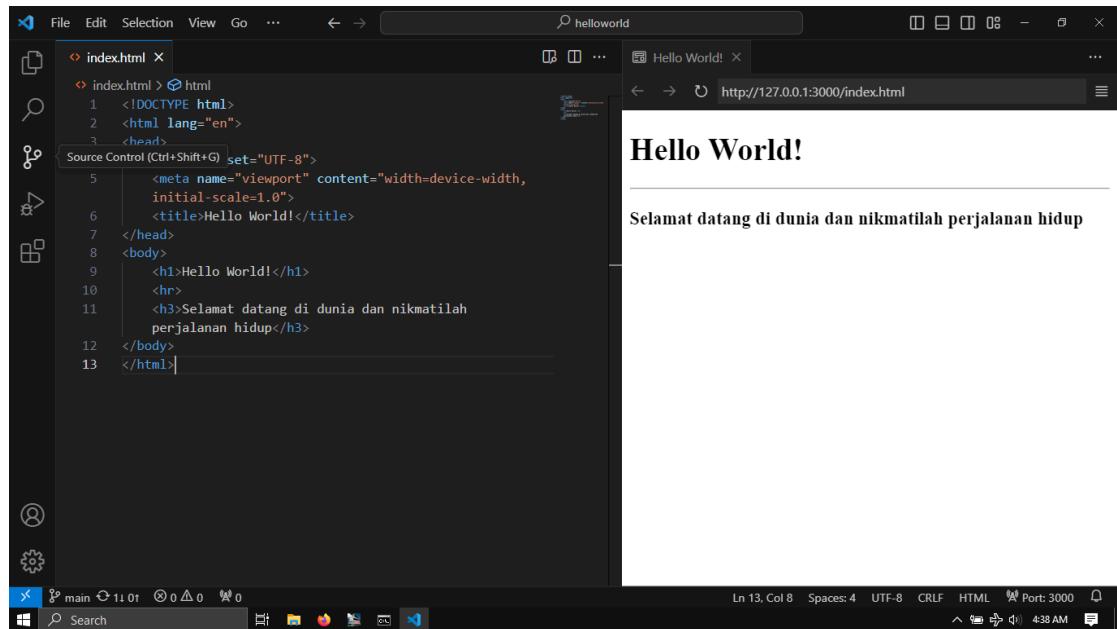
6. Kemudian klik tombol **Commit changes**, lalu akan muncul popup untuk mengisi pesan perubahan-nya.



7. Lalu, cek sudah berhasil berubah di github

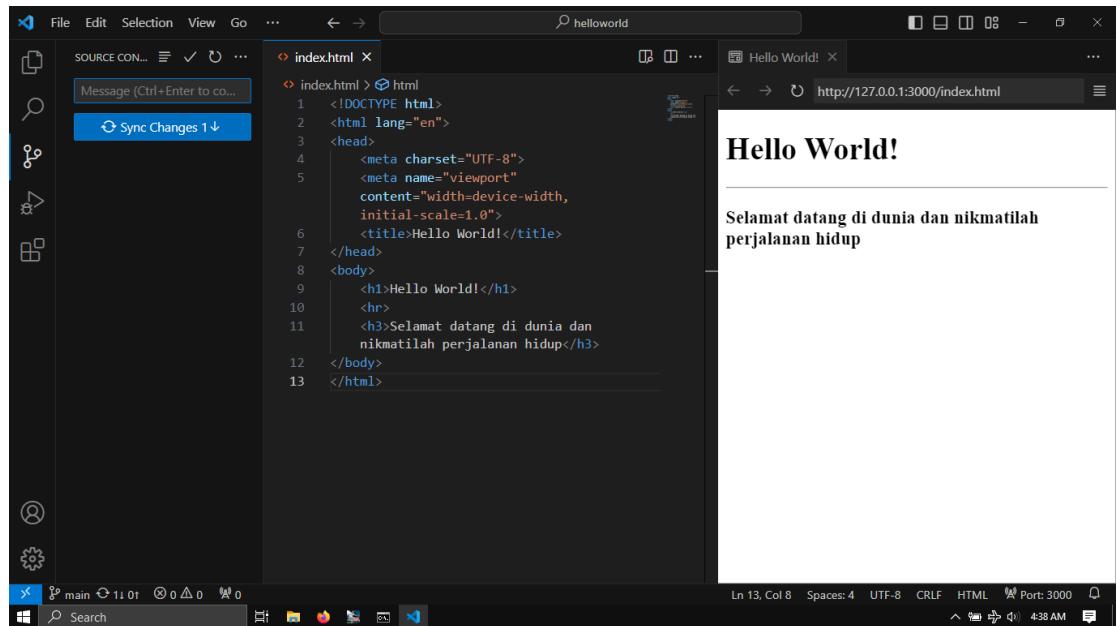


8. Seperti biasa, kita buka navbar source control di project kita melalui vscode



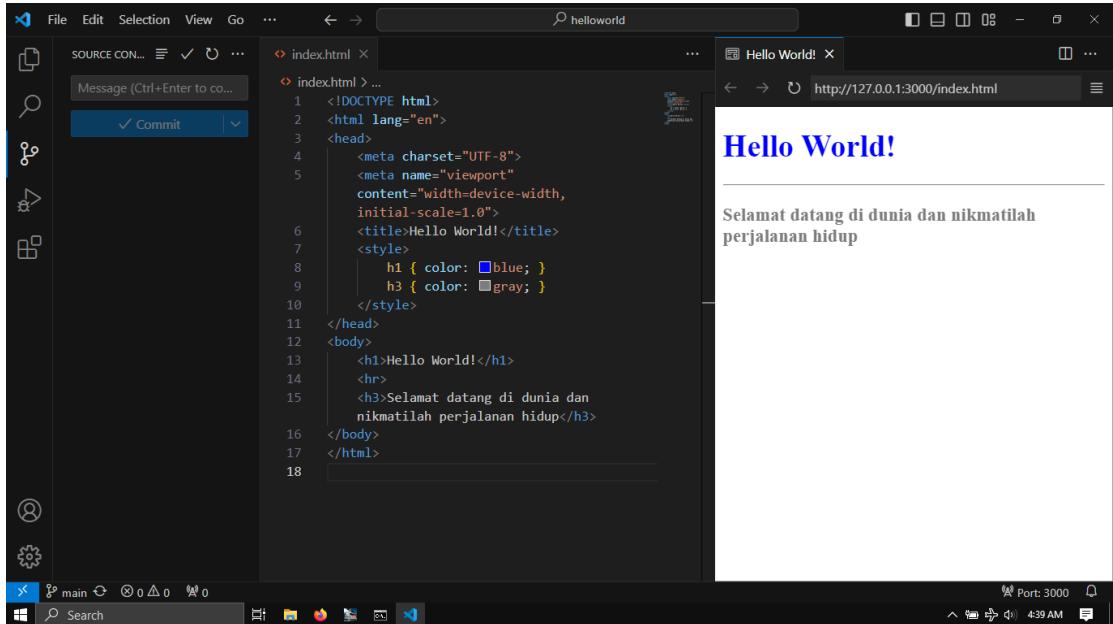
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World!</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <hr>
    <h3>Selamat datang di dunia dan nikmatilah perjalanan hidup</h3>
  </body>
</html>
```

9. Kita melihat terdapat perubahan kode masuk, yang perlu kita ambil perubahannya (pull) dan klik tombol **Sync Changes**



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World!</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <hr>
    <h3>Selamat datang di dunia dan nikmatilah perjalanan hidup</h3>
  </body>
</html>
```

10. Setelah itu lihatlah, kode yang tadi kita tambahkan di github berhasil di tambahkan atau diambil, kemudian ditaruh di kodingan project lokal kita



The screenshot shows a code editor interface with a file named 'index.html' open. The code contains HTML and CSS. In the browser preview, the page displays 'Hello World!' in blue and 'Selamat datang di dunia dan nikmatilah perjalanan hidup' in gray.

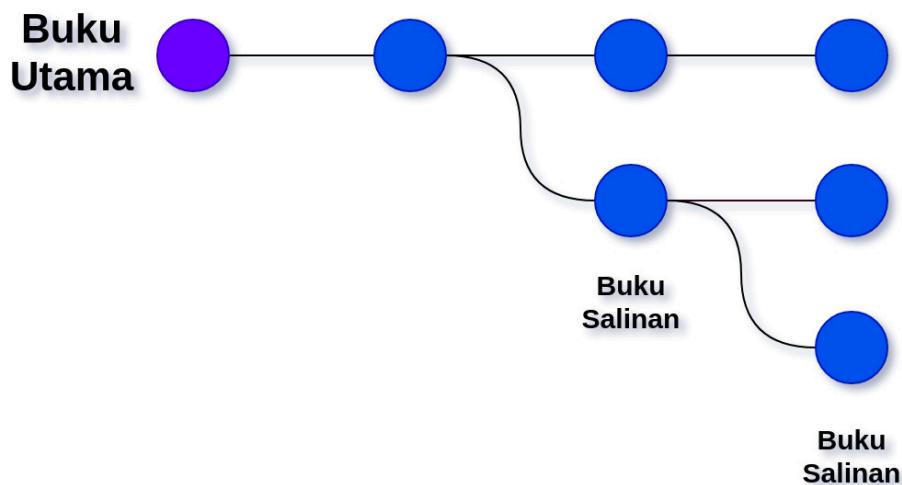
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width,
        initial-scale=1.0">
    <title>Hello World!</title>
    <style>
        h1 { color: blue; }
        h3 { color: gray; }
    </style>
</head>
<body>
    <h1>Hello World!</h1>
    <hr>
    <h3>Selamat datang di dunia dan
        nikmatilah perjalanan hidup</h3>
</body>
</html>
```

11. Selesai, akhirnya kamu telah belajar mengambil perubahan dari Github dan ini akan bermanfaat jika bekerja secara kolaborasi

Mengenal dan Membuat Branch

Dalam konteks Git dan GitHub, "branch" mengacu pada cabang pengembangan project yang dapat dibuat dari sebuah repository. Branch memungkinkan pengembang untuk bekerja pada fitur atau perbaikan bug tanpa mengubah kode di cabang utama (biasanya disebut sebagai "master" atau "main"). Ini memungkinkan pengembang untuk menjaga kestabilan cabang utama sementara mengembangkan dan menguji fitur atau perubahan lain di cabang terpisah.

Bayangkan Git repository seperti buku cerita besar. Cabang utama (main branch) adalah kisah utama yang sudah selesai ditulis. Setiap kali ada yang ingin menambahkan cerita baru, mereka bikin salinan bukunya dan menulis cerita tambahan di salinannya.

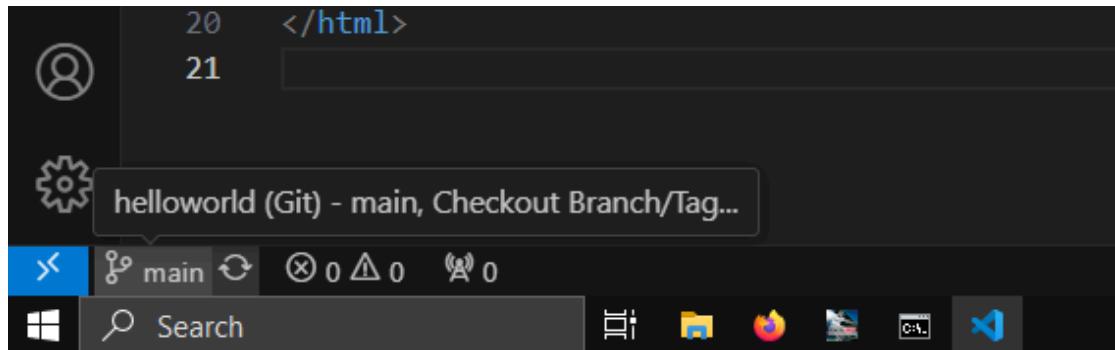


Jadi, cabang itu kayak versi baru dari buku yang sama, cuma sekarang ada cerita tambahan yang masih dalam proses. Gampangnya, ini seperti menulis kisah baru tanpa menyentuh buku utama. Nah, setelah cerita tambahan itu jadi dan bagus, kita bisa gabungin (merge) cerita itu kembali ke buku utama.

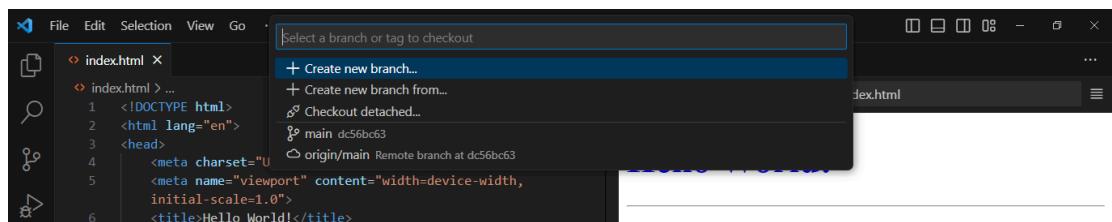
Jadi, cabang itu seperti punya salinan buku dengan cerita tambahan yang belum dimasukin ke buku utama. Proses gabungin cerita tambahan itu ke buku utama itu mirip dengan menggabungkan (merge) cabang ke cabang utama di Git.

Dalam bagian ini saya akan mengambil studi kasus sebelumnya yaitu *project hello world*, dimana sebelumnya kita telah berhasil menampilkan teks **Hello World** dan memodifikasi hal lainnya seperti warna font dll. Sekarang, saya akan membuat teks hello world dalam bahasa lain, namun saya ingin tetap tata letak atau layoutnya sama seperti teks hello world dalam bahasa inggris. Dari permasalahan dan kasus inilah kita perlu sebuah teknik **Branch** atau istilah yang lebih mudah dipahami yaitu menduplikasi. Mari kita sama-sama belajar tentang branch dalam setiap langkah per-langkah.

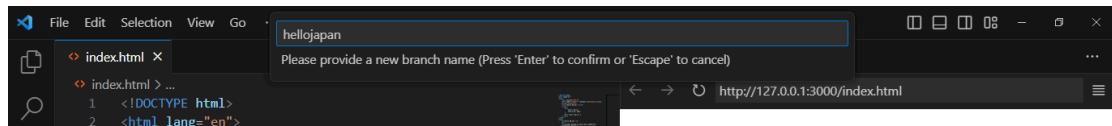
1. Buka project kita di vscode
2. Lalu klik “**switch branch**” pada pojok kiri bawah layar



3. Setelah itu akan muncul pop-up dan pilih **Create new branch**



4. Dalam kasus ini, saya akan membuat teks hello world dalam bahasa jepang, maka dari itu saya akan membuat branch/cabang bernama ***hellojapan***

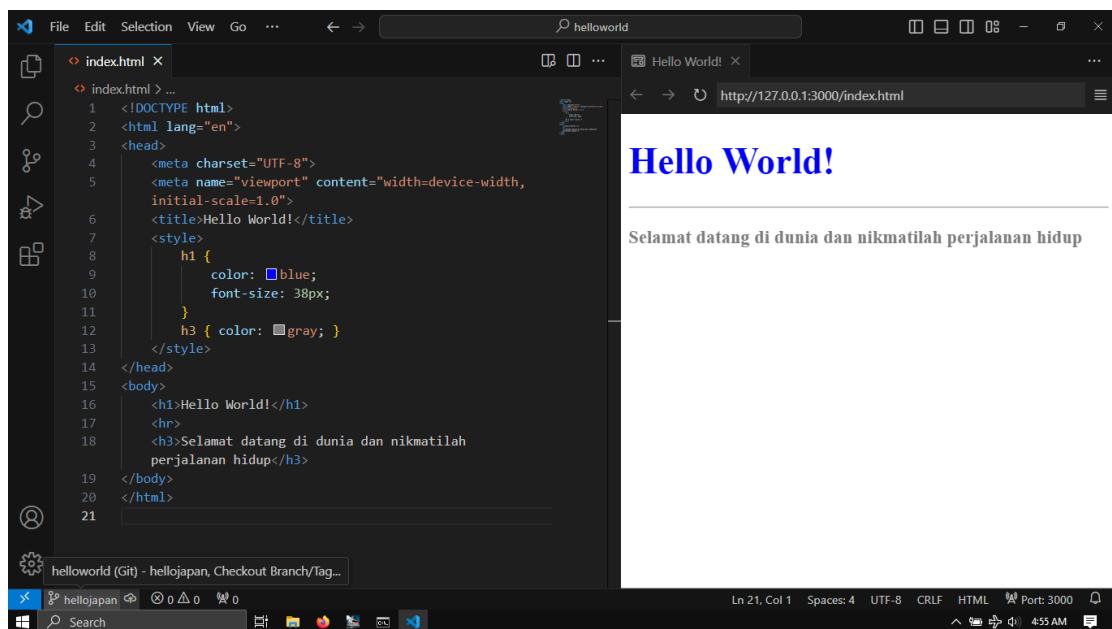


```
File Edit Selection View Go ... index.html x hellojapan
index.html > ...
<!DOCTYPE html>
<html lang="en">
```

Please provide a new branch name (Press 'Enter' to confirm or 'Escape' to cancel)

http://127.0.0.1:3000/index.html

5. Setelah itu sistem akan otomatis menduplikasi dan mengarahkan kita ke branch ***hellojapan***, kamu bisa cek di pojok kiri bawah layar dan sekarang kita sudah siap untuk mengubah kode kita ke dalam bahasa jepang, tanpa perlu khawatir akan merusak kode dalam bahasa inggris. Karena kode atau teks hello world dalam bahasa inggris ada di branch main/utama



File Edit Selection View Go ... index.html x helloworld

```
index.html > ...
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Hello World!</title>
<style>
h1 {
color: blue;
font-size: 38px;
}
h3 { color: gray; }
</style>
</head>
<body>
<h1>Hello World!</h1>
<hr>
<h3>Selamat datang di dunia dan nikmatilah perjalanan hidup</h3>
</body>
</html>
```

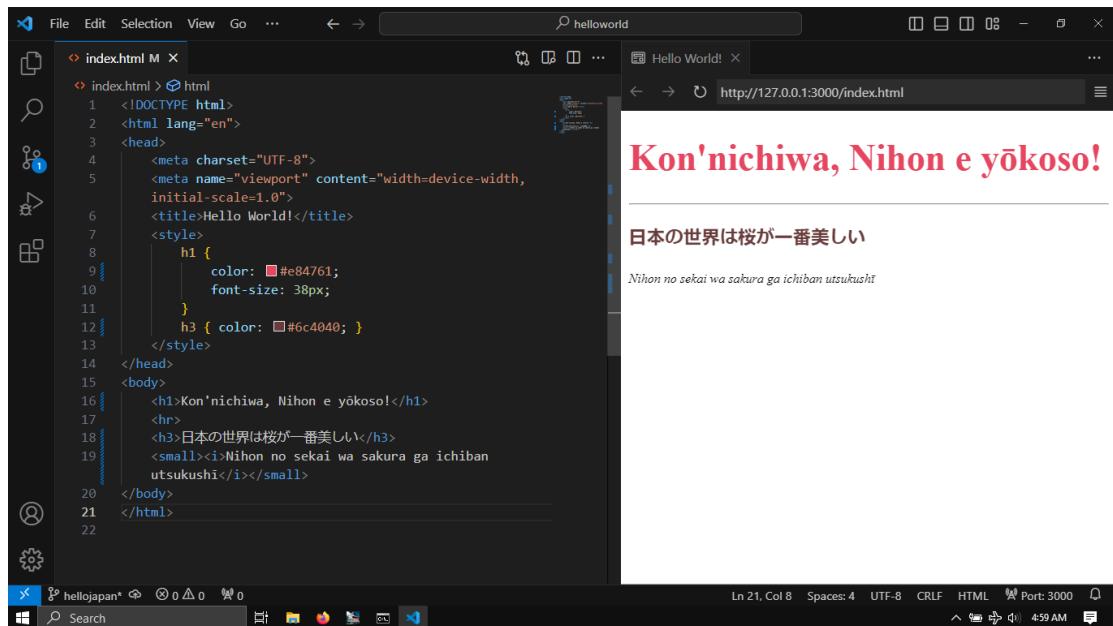
helloworld (Git) - hellojapan, Checkout Branch/Tag...

Hello World!

Selamat datang di dunia dan nikmatilah perjalanan hidup

Ln 21, Col 1 Spaces: 4 UTF-8 CRLF HTML Port: 3000

6. Sekarang, saya sudah mengubahnya ke dalam bahasa jepang

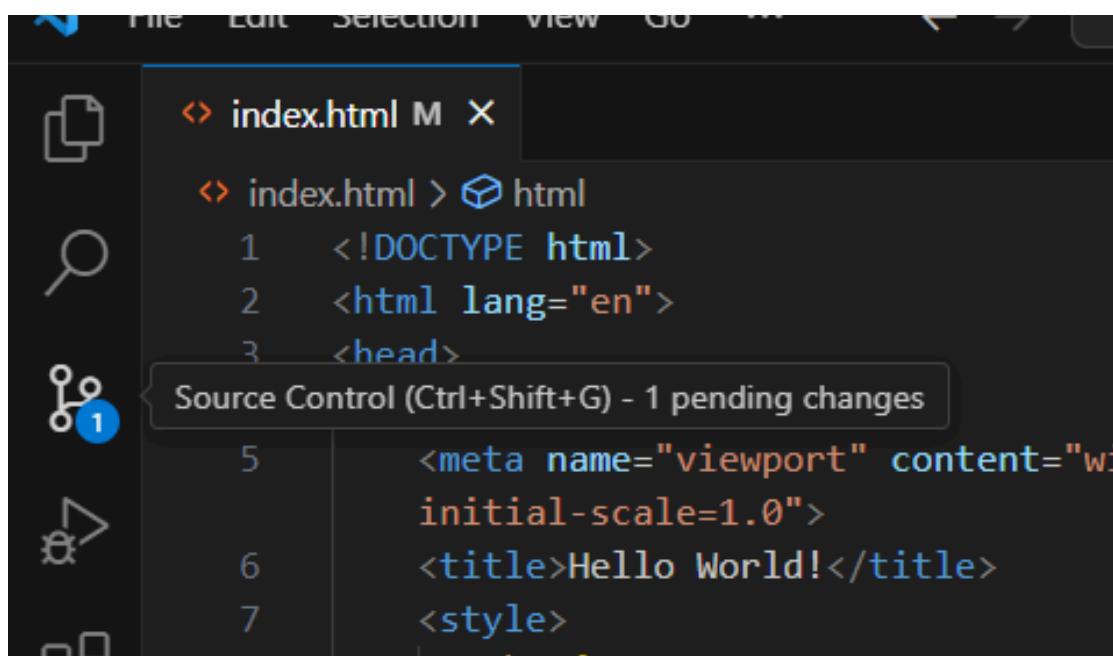


A screenshot of a code editor window titled "index.html M X". The code editor shows the following HTML content:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World!</title>
    <style>
        h1 {
            color: #e84761;
            font-size: 38px;
        }
        h3 { color: #6c4040; }
    </style>
</head>
<body>
    <h1>Kon'nichiwa, Nihon e yōkosō!</h1>
    <hr>
    <h3>日本の世界は桜が一番美しい</h3>
    <small><i>Nihon no sekai wa sakura ga ichiban utsukushī</i></small>
</body>
</html>
```

The browser tab is titled "Hello World!" and shows the rendered page with the text "Kon'nichiwa, Nihon e yōkosō!" in red and "日本の世界は桜が一番美しい" in black. Below the browser window, the status bar shows "Ln 21, Col 8" and "Port: 3000".

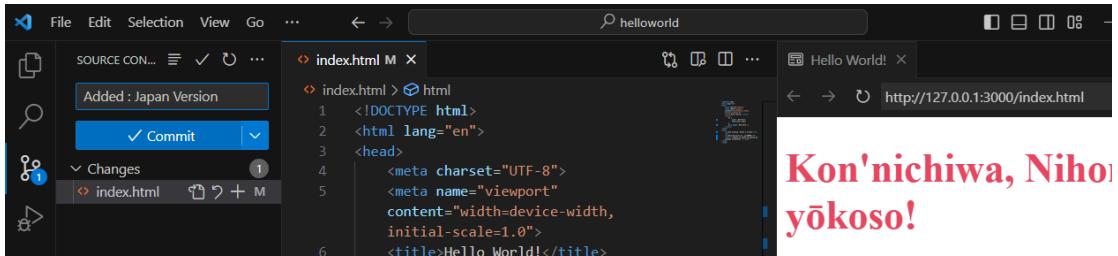
7. Di tab “Source Control” di sidebar kiri layar terdapat notifikasi sebuah perubahan, maka dari itu kita perlu push ke github agar perubahan yang kita lakukan dapat tersimpan di github.



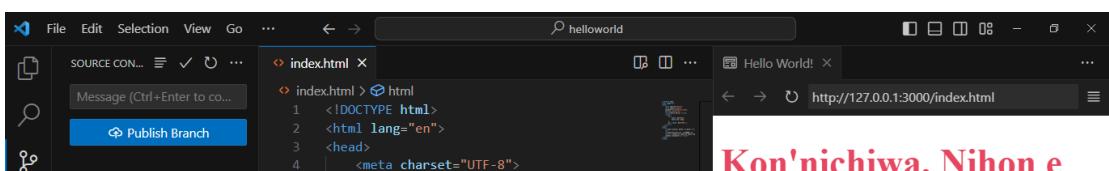
A screenshot of a code editor window showing the "Source Control" tab in the sidebar. A notification icon with the number "1" indicates one pending change. The list of pending changes shows the following additions:

Line	Text
5	<meta name="viewport" content="width=device-width, initial-scale=1.0">
6	<title>Hello World!</title>
7	<style>

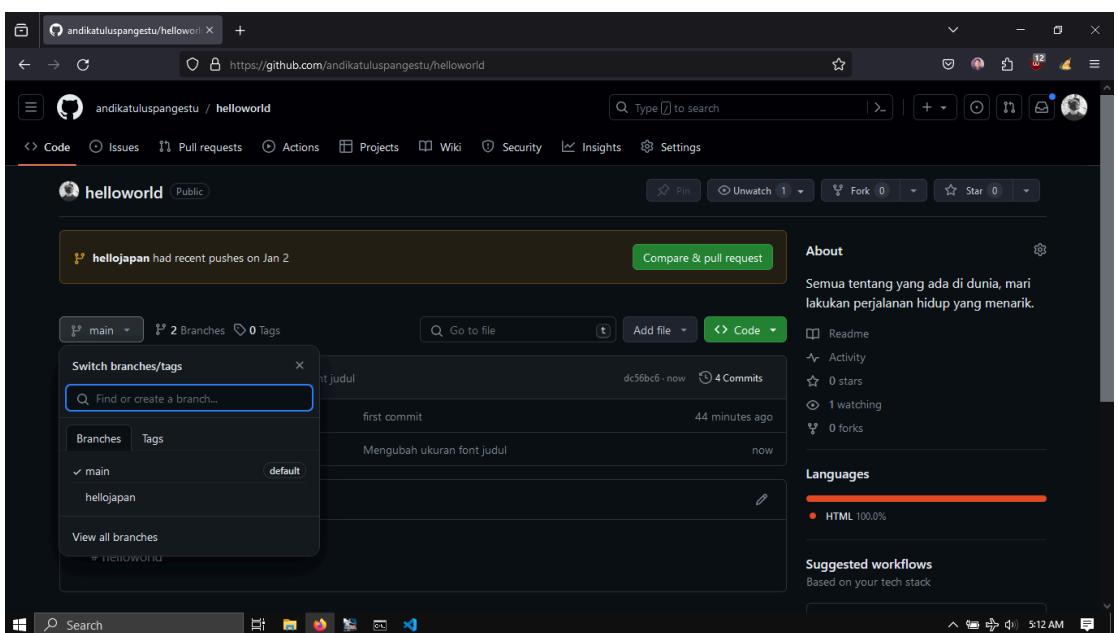
- Lalu kita lakukan commit dengan menambah pesan perubahan berupa “Added : Japan Version” artinya kita menambahkan versi dalam bahasa jepang. Setelah itu, kita klik tombol commit.



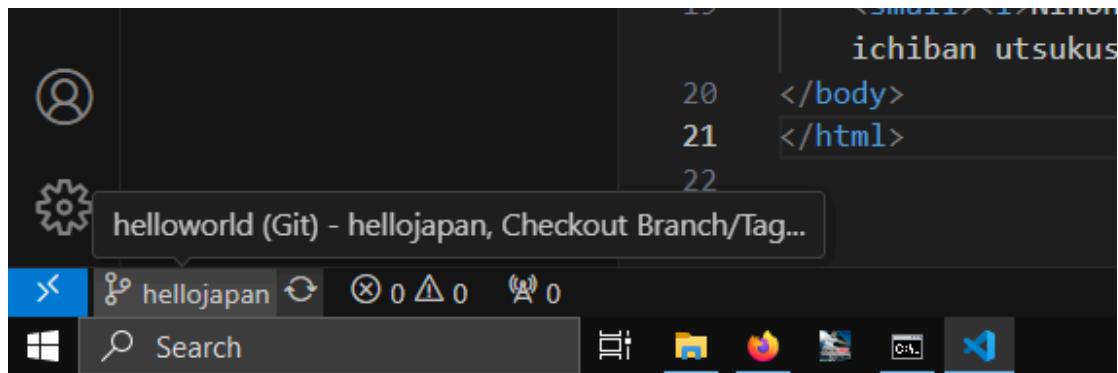
- Setelah itu, kita ***publish branch*** agar branch yang kita buat tadi, ada di github juga.



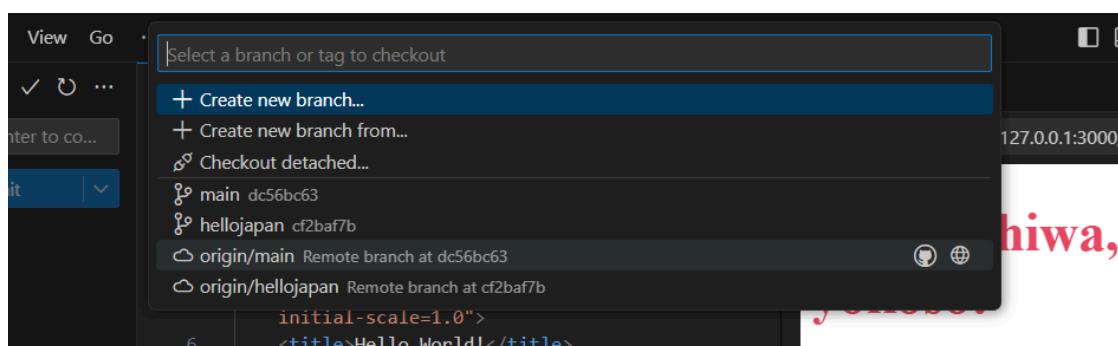
- Buka link repository github kita dan cek apakah sudah terpublish atau belum, jika sudah maka akan tampil seperti berikut ini.



11. Sekarang saya akan berpindah ke branch main/utama untuk melihat apakah hello world versi sebelumnya ikut berubah atau tidak. Dan klik switch branch di pojok kiri bawah layar.

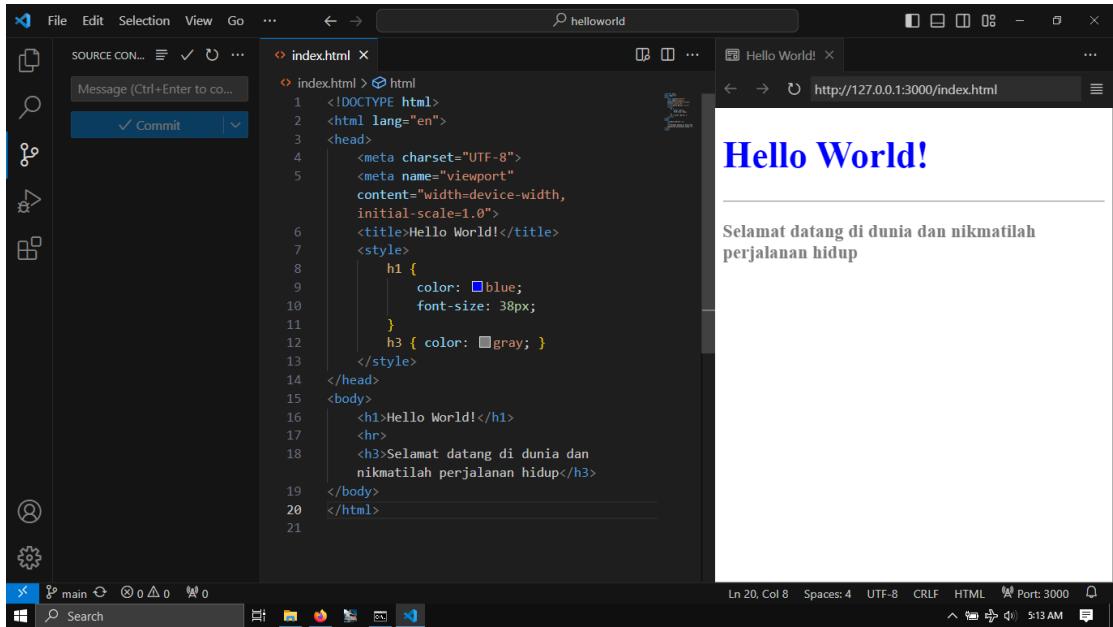


12. Kemudian pada pop-up pilih branch **main**



hiwa,

13. Dan ternyata aman, tidak merubah sedikitpun, maka dapat kita anggap telah berhasil menerapkan branching di project kita 😍🎉



```
helloworld
Hello World!
Selamat datang di dunia dan nikmatilah perjalanan hidup
```

The screenshot shows a terminal window with the title "helloworld". Inside the terminal, the command "Hello World!" is displayed in blue text, followed by the message "Selamat datang di dunia dan nikmatilah perjalanan hidup". The terminal interface includes a status bar at the bottom showing "Ln 20, Col 8" and "Port: 3000".

Berkontribusi di Project Orang Lain

GIT dan Github bisa memudahkan kita dalam berkontribusi ke dalam project milik orang lain yang bersifat project sumber terbuka atau sering disebut *Open Source Project*. Dalam proyek sumber terbuka, komunitas yang terdiri dari berbagai kontributor, termasuk pengembang perangkat lunak independen, dapat berpartisipasi. Hal ini memungkinkan adanya kerjasama di antara berbagai individu atau kelompok yang memiliki minat atau keahlian dalam pengembangan perangkat lunak tertentu.

Keuntungan utama dari proyek sumber terbuka melibatkan pemberian akses kepada banyak orang untuk berkontribusi, mengidentifikasi bug, meningkatkan fitur, atau memberikan umpan balik. Proyek sumber terbuka juga mempromosikan pertukaran pengetahuan dan inovasi di antara komunitas pengembang. Sebelum berkontribusi ke proyek sumber terbuka kita harus mematuhi beberapa hal seperti :

1. Baca aturan kontribusi
2. Memahami Lisensi Proyek
3. Meninjau Panduan Kode
4. Mengidentifikasi Masalah atau Tugas
5. Menggunakan Cabang Terpisah

Dalam kasus ini saya akan berkontribusi ke salah satu repository atau akun github milik **Sandhika Galih** yang dapat diakses di laman repository github <https://github.com/sandhikagalih/channel-youtube-programming-dan-teknologi-indonesia> dan disini saya akan menambahkan “Daftar Channel Youtube Programming & Teknologi Indonesia”.

The screenshot shows the GitHub repository page for 'channel-youtube-programming-dan-teknologi-indonesia'. The repository is public and has 8 stars, 150 forks, and 463 watchers. It contains 32 commits from 'slowy07'. The README file lists various Indonesian YouTube channels with their names, technologies, and URLs. The table includes rows for WPU, Sekolah Koding, YukCoding Media, and IDStack.

Nama Channel	Teknologi	URL
WPU	HTML, CSS, Javascript, PHP, Git, GitHub, SASS, Bootstrap, NodeJS, CodeIgniter, Laravel, Livestreaming	link
Sekolah Koding	HTML, CSS, Javascript, VueJS, ReactJS, Laravel, Golang, Python, Kotlin, Dart, Flutter, TailwindCSS	link
YukCoding Media	PHP, CodeIgniter, Laravel, Wordpress, React Native, Flutter	link
IDStack	HTML, CSS, PHP, CodeIgniter, Laravel, VueJS, Git, Livewire, MEVN, MERN, NextJS	link

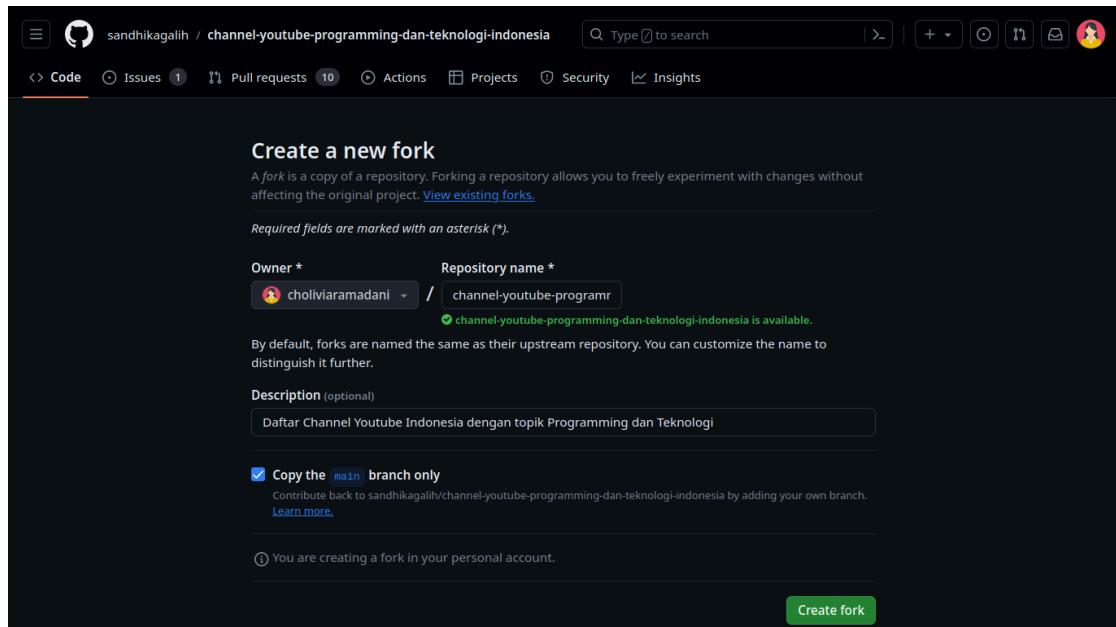
Berikut adalah langkah-langkah dalam berkontribusi :

1. Fork atau Salin repository ke akun github kita, dengan klik tombol “Fork”

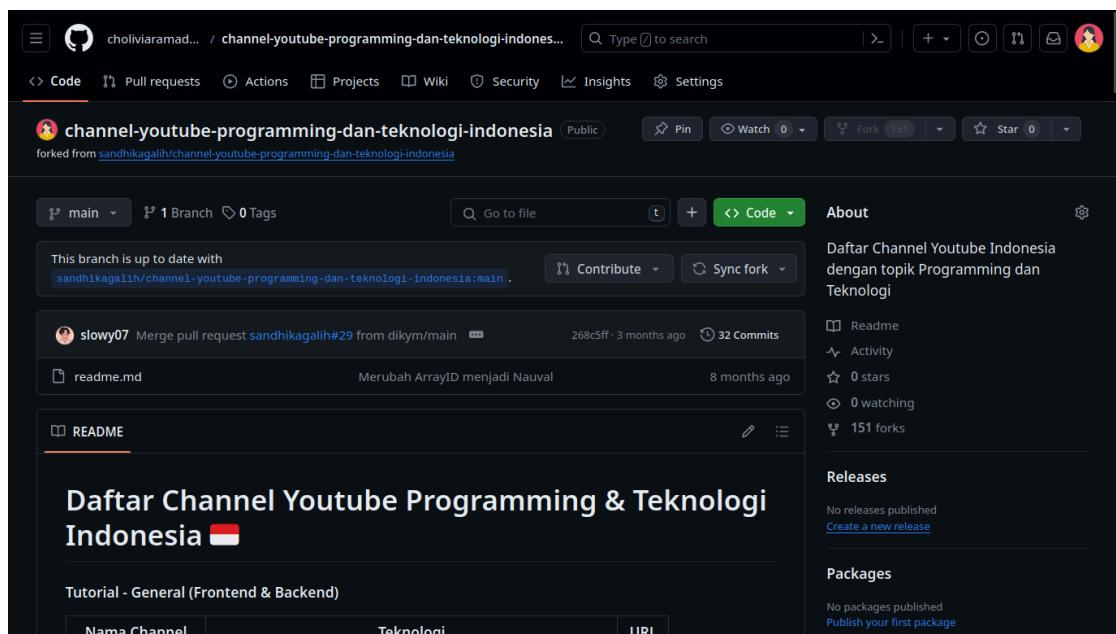
The screenshot shows the GitHub repository page for 'channel-youtube-programming-dan-teknologi-indonesia' after it has been forked. The forked repository has 8 stars, 150 forks, and 463 watchers. The 'About' section remains the same, listing the 'Daftar Channel Youtube Indonesia dengan topik Programming dan Teknologi'.

2. Biarkan detail repository nya ini seperti bawaan saja, lalu klik tombol

Create Fork

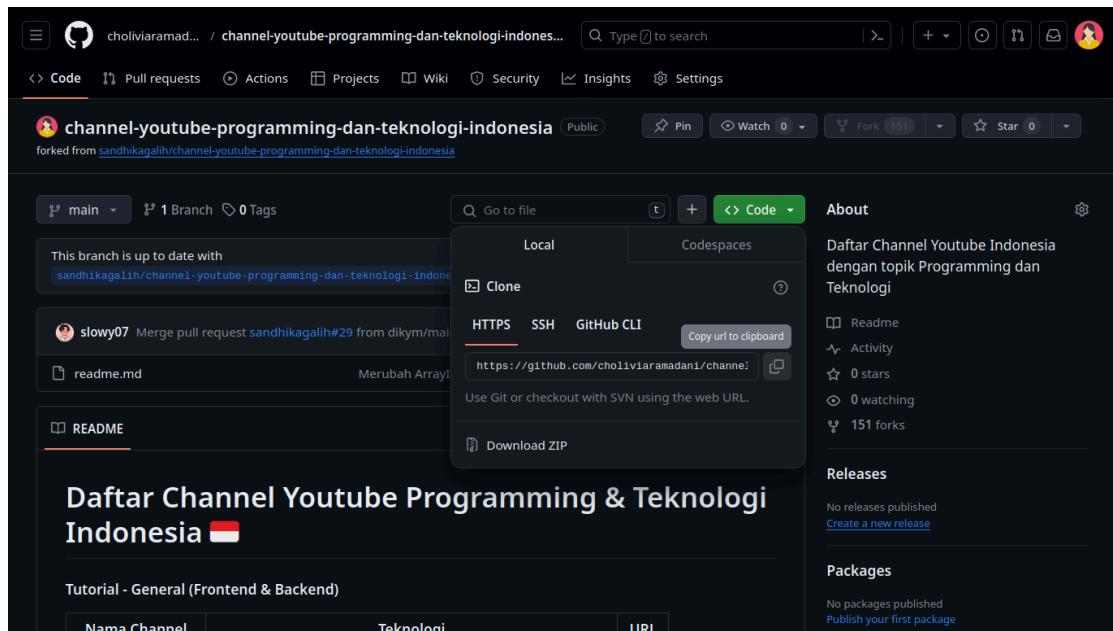


3. Setelah itu, tampilannya akan menjadi seperti ini.

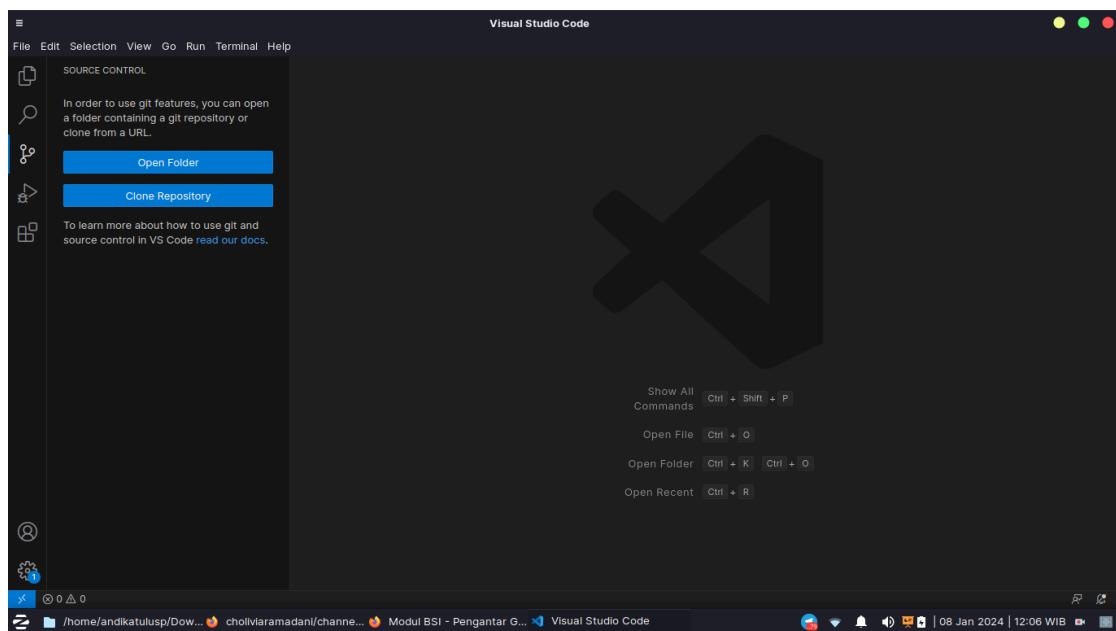


4. Selanjutnya, kita akan clone repository hasil fork tadi ke komputer kita agar bisa di edit menggunakan vscode.

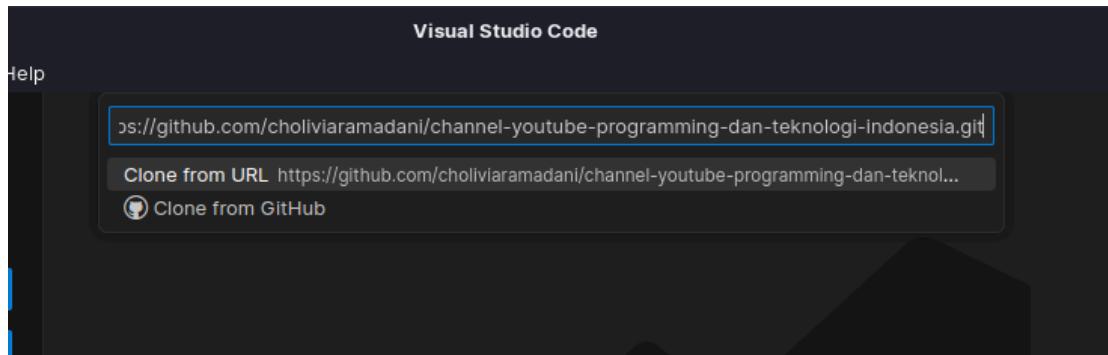
5. Klik tombol clone dan pilih yang *HTTPS*



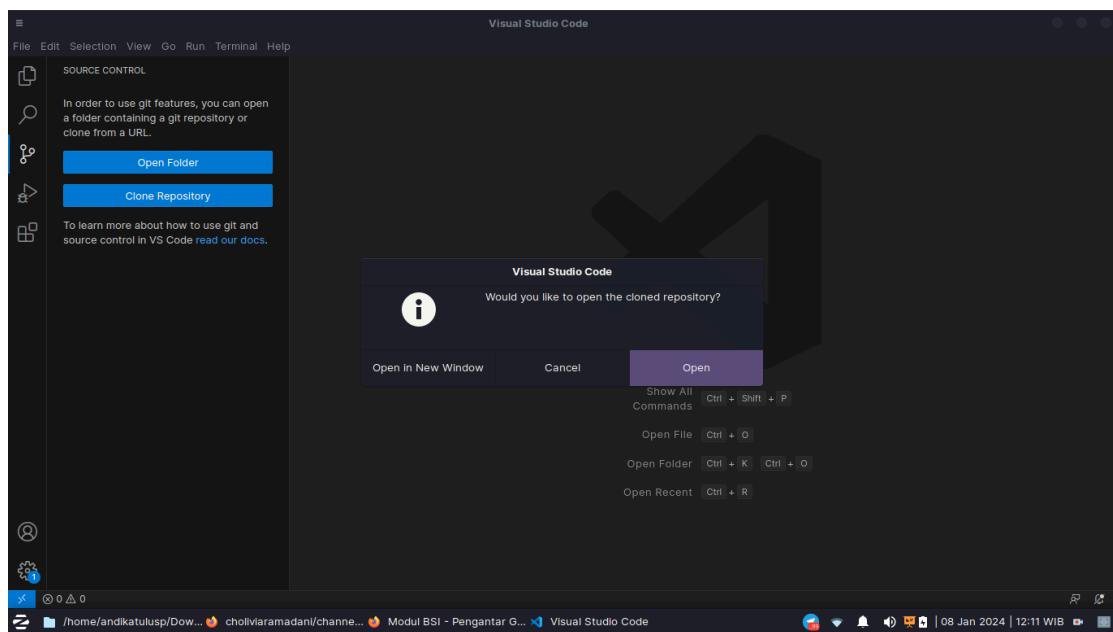
6. Sekarang kita buka visual studio code, dan klik tab *Source Control* atau tekan CTRL + SHIFT + G



7. Klik tombol **Clone Repository** dan masukan link yang dicopy dari github tadi, lalu tekan **ENTER**



8. Setelah itu, nanti akan diarahkan untuk memilih dimana project akan disimpan dalam folder mana.
9. Kemudian, klik "**Open**"



10. Maka kode yang akan kita modifikasi sudah berhasil ditampilkan

```
readme.md - channel-youtube-programming-dan-teknologi-indonesia - Visual Studio Code
File Edit Selection View Go Run Terminal Help
① readme.md x
① readme.md > # Daftar Channel Youtube Programming & Teknologi Indonesia => ##### Tutorial - General (Frontend & Backend)
1 | # Daftar Channel Youtube Programming & Teknologi Indonesia
2 |
3 | ###### Tutorial - General (Frontend & Backend)
4 |
5 | | Nama Channel | Teknologi
6 | |-----|
7 | | WPU | HTML, CSS, Javascript, PHP, Git, GitHub, SASS, <br> Bootstrap, NodeJS, CodeIgniter, Laravel, Livestreaming
8 | | Sekolah Koding | HTML, CSS, Javascript, VueJS, ReactJS, <br> Laravel, Golang, Python, Kotlin, Dart, Flutter, TailwindCSS
9 | | YukCoding Media | PHP, CodeIgniter, Laravel, Wordpress, React Native, Flutter
10 | | IDStack | HTML, CSS, PHP, CodeIgniter, Laravel, VueJS, Git, Livewire, <br> MEVN, MERN, NextJS
11 | | Aden Rahmandi | HTML, CSS, Javascript, PHP, Wordpress, Case Study
12 | | Dosen Ngoding | HTML, Laravel, Kotlin, Flutter, Meme Based Education
13 | | LampungDev | Algoritma, ReactJS, React Native, NextJS, Swift, <br> GatsbyJS, Redux, NodeJS, Wordpress, ExpressJS, <br> Golang,
14 | | M Fikri | Javascript, PHP, React, Vue, Codeigniter, Node, Express
15 | | Nusendra | Typescript, ExpressJS, VueJS, Design Pattern
16 |
17 |<br><hr>
18 |
19 | ###### Tutorial - Frontend
20 |
21 | | Nama Channel | Teknologi | URL
22 | |-----|-----|-----|
23 | | Nauval | JS, CSS, TailwindCSS, NextJS, ReactJS | [link](https://www.youtube.com/@mhdnaulavalazhar)
24 | | Balademy | Angular, Livewire, Vue | [link](https://www.youtube.com/channel/UCo2Mh0gTrEUKdL9f)
25 | | Prawito Hudoro | ReactJS, NextJS, React Native, MERN, JS, Livestreaming | [link](https://www.youtube.com/c/prawitohudoro)
26 | | Wahidev Academy | React, React Native | [link](https://www.youtube.com/c/WahidevAcademy)
27 |
28 |<br><hr>
29 |
30 | ###### Tutorial - Backend
31 |
32 | | Nama Channel | Teknologi
33 | |-----|
34 | | Tozimur Online | Struktur Data, DevOps, Rust, Ruby, Keras | [link](https://www.youtube.com/c/TozimurOnline)
Ln 17, Col 9 | Spaces: 4 | UTF-8 | LF | Markdown | ✓ Prettier | ⚡ | 📁 | 08 Jan 2024 | 12:15 WIB
```

11. Saya mencoba menambahkan referensi channel baru yaitu “Kukuh Aldy - Channel yang membahas seputar UIUX dan Tips Desain”

```
readme.md - channel-youtube-programming-dan-teknologi-indonesia - Visual Studio Code
File Edit Selection View Go Run Terminal Help
① readme.md x
① readme.md > # Daftar Channel Youtube Programming & Teknologi Indonesia => ##### UIUX Design
110 | Zain Fathoni | UI UX, Gaming, Assimilasi, Retrospective | [link](https://www.youtube.com/c/zainfathoni)
111 | Resir014 | Flutter, React | [link](https://www.youtube.com/c/Resir014) / [link](https://www.twitch.tv/resir014)
112 |
113 |<br><hr>
114 | ###### UIUX Design
115 |
116 | | Nama Channel | Topik | URL
117 | |-----|-----|-----|
118 | | Mas Kukuh Aldy | UI/UX, Tips Trick Desain | [link](https://www.youtube.com/c/MasKukuhAldy)
119 |
120 |<br><hr>
121 |
122 | ###### Podcast / Interview / Talk Show
123 |
124 | | Nama Channel | Teknologi | URL
125 | |-----|-----|-----|
126 | | Tozimur Online | Struktur Data, DevOps, Rust, Ruby, Keras | [link](https://www.youtube.com/c/TozimurOnline)
127 |
128 |
129 |
130 |
131 |
132 |
133 |
134 |
135 |
Ln 124, Col 9 | Spaces: 4 | UTF-8 | LF | Markdown | ✓ Prettier | ⚡ | 📁 | 08 Jan 2024 | 12:15 WIB
```

12. Tekan **CTRL + SHIFT + G** dan masukan pesan commit

```
readme.md - channel-youtube-pro - Visual Studio Code
File Edit Selection View Go Run Terminal Help
SOURCE CONTROL
Added : Channel Mas Kukuh Aldy
(UIUX Design)
Commit Changes on "main"
Changes
① readme.md
```

- 13.Jika ada peringatan klik “**Always**”
- 14.Kemudian klik tombol “**Sync Changes**” untuk push perubahan ke repository fork kita tadi.
- 15.Kita cek apakah sudah berubah di repository github hasil fork tadi

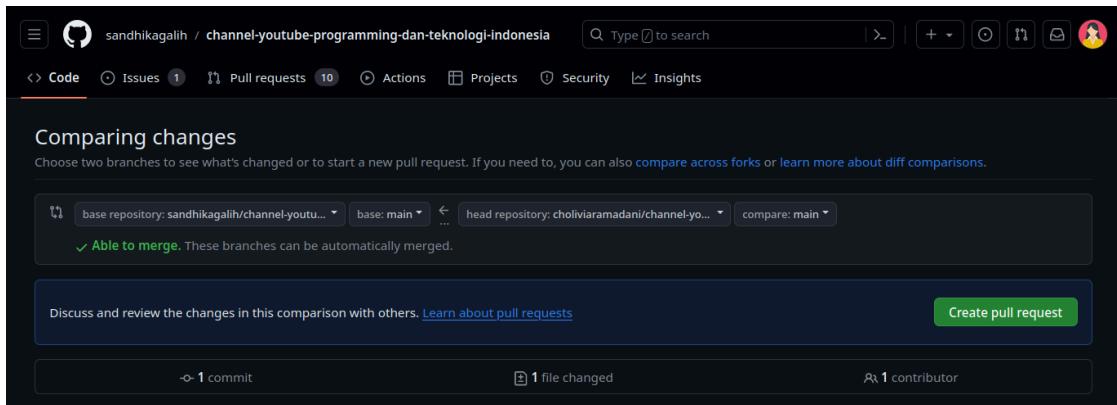
The screenshot shows a GitHub repository page. At the top, it says 'cholivaramad... / channel-youtube-programming-dan-teknologi-indonesia...'. Below that, there are tabs for 'Code', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The repository is public and has 151 forks. A commit from 'andikatuluspangestu' is shown, and a file 'readme.md' is listed. The README content is: 'Daftar Channel Youtube Programming & Teknologi Indonesia'.

- 16.Kita sudah menambahkan referensi channel youtube baru, namun baru disimpan di repository akun kita saja, belum digabungkan dengan repository di akun **Sandhika Galih**
- 17.Hal yang kita akan lakukan adalah klik tombol **Contribute** dan klik **Open Pull Request** yang artinya meminta perubahan kita diterapkan di repository di akun github sandhika galih

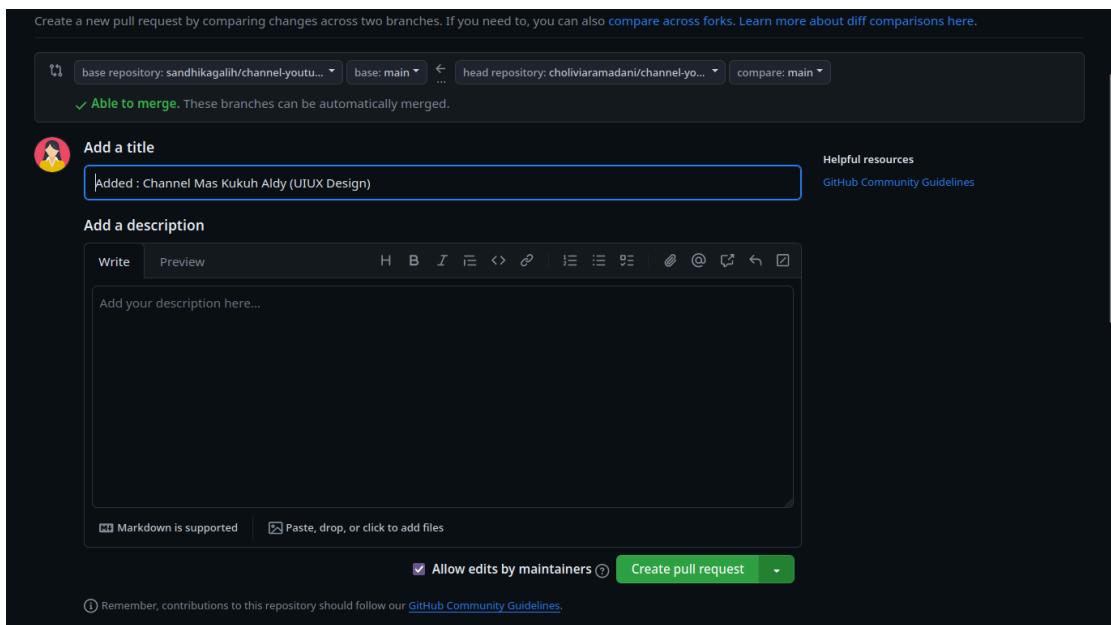
The screenshot shows the same GitHub repository page after a pull request was opened. A tooltip over the 'Contribute' button indicates that the branch is 1 commit ahead of the upstream master branch. A green 'Open pull request' button is visible at the bottom of the tooltip.

18.Jika tidak ada konflik atau error, maka kita bisa langsung klik tombol

Create Pull Request

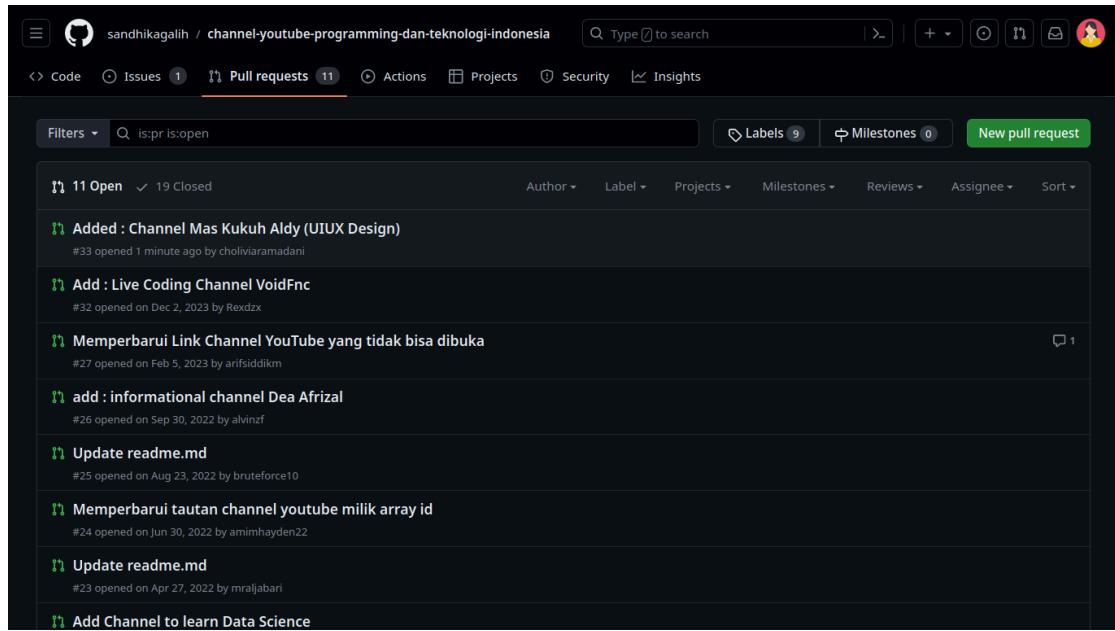


19.Kita bisa memberikan judul dan deskripsi dari perubahan yang telah dilakukan.



20. Setelah itu, kita cek apakah permintaan penggabungan sudah masuk atau belum di repository github sandika galih, bisa kita cek melalui [tautan ini](#)

21. Dan terlihat kalau permintaan kita sudah masuk di bagian paling atas dan tinggal menunggu pemilik repository utama nya yaitu pak sandhika galih menyetujui dan menggabungkan.



The screenshot shows a GitHub repository page for 'sandhikagalih / channel-youtube-programming-dan-teknologi-indonesia'. The 'Pull requests' tab is selected, showing 11 open pull requests. A search bar at the top has the query 'is:pr is:open'. Below the search bar, there are filters for 'Labels' (9), 'Milestones' (0), and a 'New pull request' button. The pull requests listed are:

- Added : Channel Mas Kukuh Aldy (UIUX Design) #33 opened 1 minute ago by chollivaramadani
- Add : Live Coding Channel VoidFnc #32 opened on Dec 2, 2023 by Rexdzx
- Memperbarui Link Channel YouTube yang tidak bisa dibuka #27 opened on Feb 5, 2023 by arifsiddikm
- add : informational channel Dea Afrizal #26 opened on Sep 30, 2022 by alvinzf
- Update readme.md #25 opened on Aug 23, 2022 by bruteforce10
- Memperbarui tautan channel youtube milik array id #24 opened on Jun 30, 2022 by amimhayden22
- Update readme.md #23 opened on Apr 27, 2022 by mraljabari
- Add Channel to learn Data Science #22 opened on Mar 27, 2022 by mraljabari

Kolaborasi Project

Dengan Git sebagai sistem kontrol versi terdistribusi dan Github sebagai platform kolaboratif, tim pengembang dapat bekerja bersama secara efisien, mengurangi konflik, efektifitas waktu dan meningkatkan kualitas perangkat lunak yang dihasilkan.

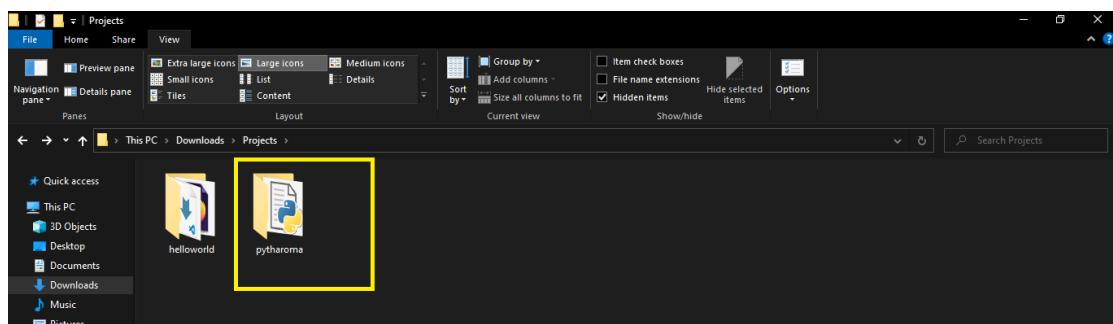
Terdapat beberapa hal yang harus diperhatikan sebelum melakukan manajemen kolaborasi menggunakan GIT & Github, yaitu :

1. Terdapat dua jenis kolaborasi, yakni kolaborasi pada proyek privat dan proyek publik. Pada proyek privat, perlu dilakukan konfigurasi terhadap daftar kolaborator yang diberikan hak untuk memodifikasi atau membuat perubahan dalam proyek. Sebaliknya, pada proyek publik, semua orang dapat berkontribusi tanpa perlu menjadi kolaborator yang terdaftar di repository. Meskipun demikian, setiap perubahan yang diusulkan akan melalui proses permintaan perubahan (pull request), yang dapat memerlukan waktu yang cukup lama, terutama jika pemilik repository utama memiliki kesibukan lain. Maka dari itu, disini akan belajar dua metode itu berkolaborasi dengan Repo Publik dan berkolaborasi dengan Repo Privat.
2. Pastikan semua anggota tim memiliki pemahaman dasar tentang Git, termasuk commit, branch, merge, dan konflik resolusi. Ini membantu mencegah kesalahan dan mempermudah manajemen proyek.
3. Tentukan aturan penamaan branch yang konsisten untuk memudahkan identifikasi dan pemahaman tentang peran setiap branch.
4. Sesuaikan pengaturan hak akses repository di Github untuk memastikan bahwa hanya anggota tim yang diperlukan yang memiliki izin untuk melakukan perubahan kritis.

A. Membuat Project Kolaborasi

Dalam kasus ini, saya **Cholivia** akan membuat sebuah project python sederhana dengan nama project **Pytharoma** yang berisi tentang program **Hitung Luas Bangun Datar** yang dibuat dengan bahasa pemrograman python, dalam project ini saya bekerja sama atau berkolaborasi dengan **Andika** untuk ikut serta menambahkan beberapa rumusan perhitungan luas bangun datar lainnya.

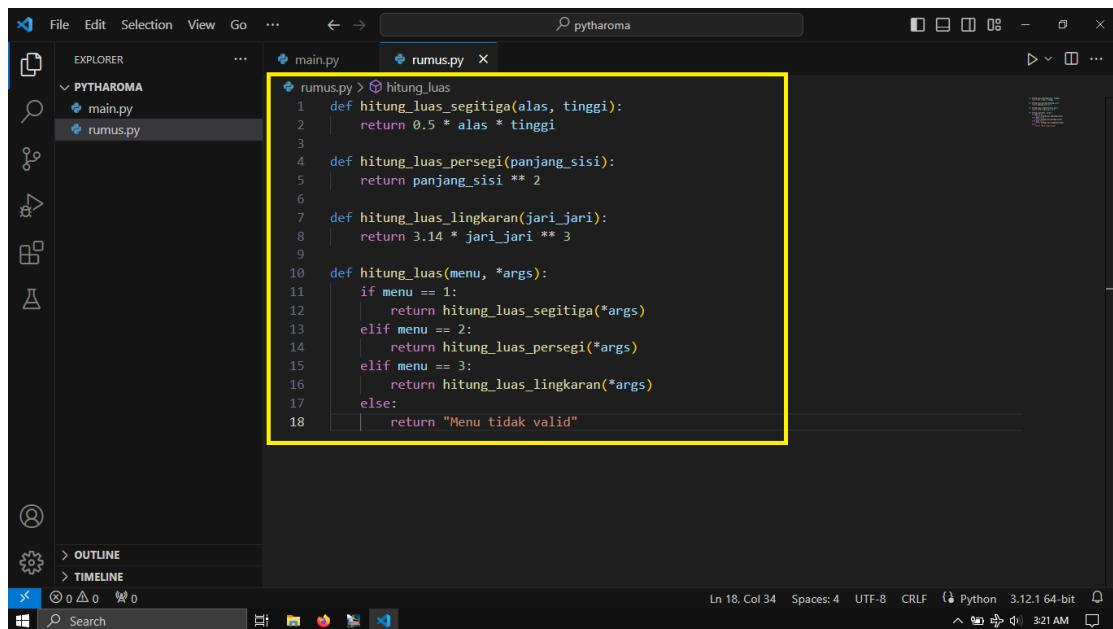
1. Membuat folder project



2. Buka folder project di VSCode dan buat file baru dengan nama **main.py** dan **rumus.py** kemudian saya isi dengan kode program dibawah ini.

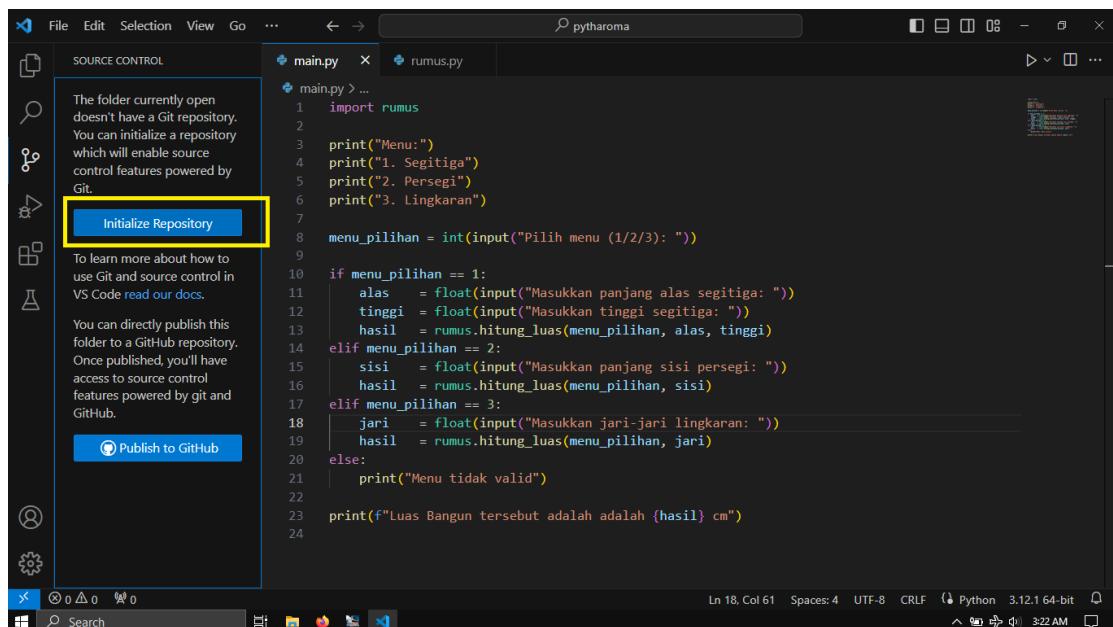
```
main.py > ...
1 import rumus
2
3 print("Menu:")
4 print("1. Segitiga")
5 print("2. Persegi")
6 print("3. Lingkaran")
7
8 menu_pilihan = int(input("Pilih menu (1/2/3): "))
9
10 if menu_pilihan == 1:
11     alas = float(input("Masukkan panjang alas segitiga: "))
12     tinggi = float(input("Masukkan tinggi segitiga: "))
13     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
14 elif menu_pilihan == 2:
15     sisi = float(input("Masukkan panjang sisi persegi: "))
16     hasil = rumus.hitung_luas(menu_pilihan, sisi)
17 elif menu_pilihan == 3:
18     jari = float(input("Masukkan jari-jari lingkaran: "))
19     hasil = rumus.hitung_luas(menu_pilihan, jari)
20 else:
21     print("Menu tidak valid")
22
23 print(f"Luas Bangun tersebut adalah adalah {hasil} cm")
```

The screenshot shows the VSCode interface with the 'PYTHAROMA' project selected in the Explorer sidebar. The 'main.py' file is open in the editor, displaying the provided Python code. The entire code block is highlighted with a yellow box. The status bar at the bottom right shows 'Ln 18, Col 61' and other file-related information.



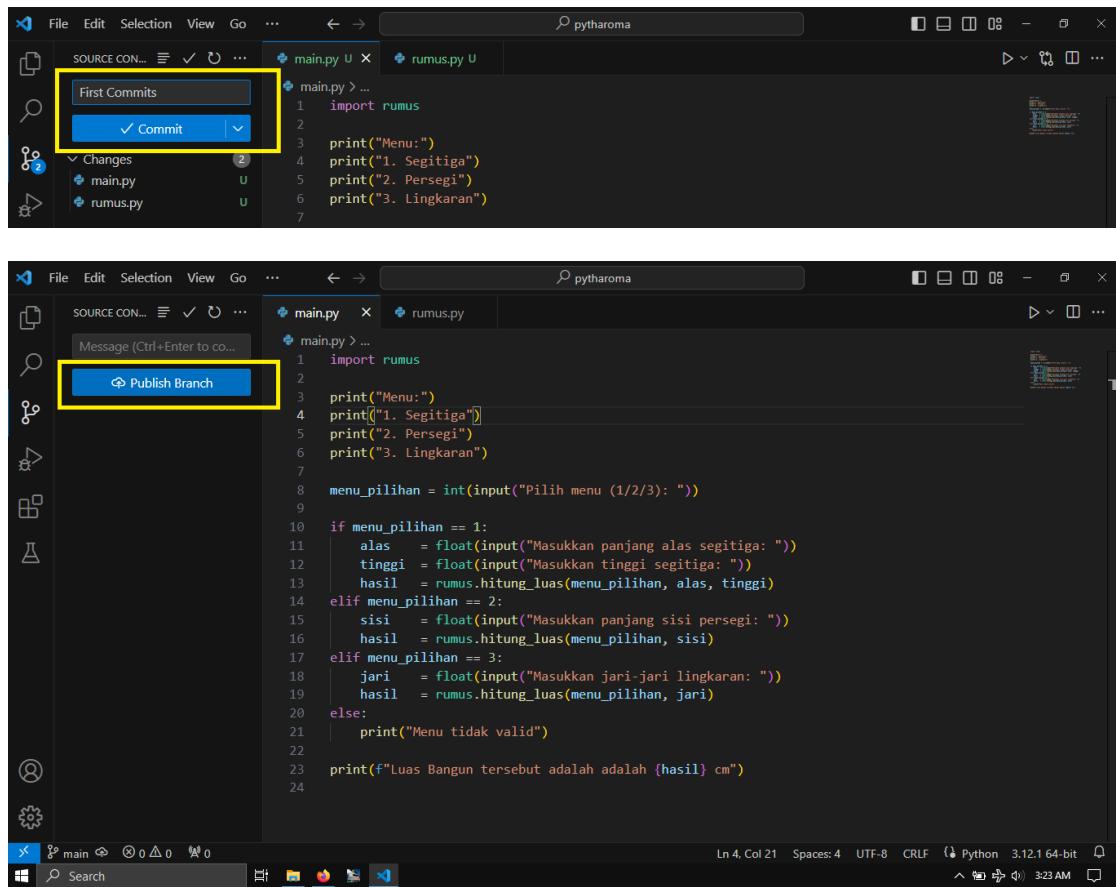
```
pytharoma
File Edit Selection View Go ...
EXPLORER PYTHAROMA main.py rumus.py
rumus.py > hitung_luas
1 def hitung_luas_segitiga(alas, tinggi):
2     return 0.5 * alas * tinggi
3
4 def hitung_luas_persegi(panjang_sisi):
5     return panjang_sisi ** 2
6
7 def hitung_luas_lingkaran(jari_jari):
8     return 3.14 * jari_jari ** 3
9
10 def hitung_luas(menu, *args):
11     if menu == 1:
12         return hitung_luas_segitiga(*args)
13     elif menu == 2:
14         return hitung_luas_persegi(*args)
15     elif menu == 3:
16         return hitung_luas_lingkaran(*args)
17     else:
18         return "Menu tidak valid"
Ln 18, Col 34 Spaces:4 UTF-8 CRLF Python 3.12.1 64-bit 3:21 AM
```

3. Jangan lupa run program python nya, agar kita mengetahui ada error atau tidak.
4. Setelah dianggap aman, maka saatnya kita push ke github.
5. Tekan **CTRL + SHIFT + G** dan klik **Initialize Repository**

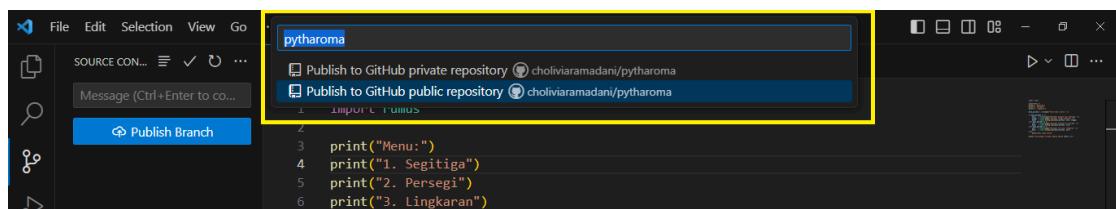


```
pytharoma
File Edit Selection View Go ...
SOURCE CONTROL main.py rumus.py
The folder currently open doesn't have a Git repository. You can initialize a repository which will enable source control features powered by Git.
Initialize Repository
To learn more about how to use Git and source control in VS Code read our docs.
You can directly publish this folder to a GitHub repository. Once published, you'll have access to source control features powered by git and GitHub.
Publish to GitHub
main.py > ...
1 import rumus
2
3 print("Menu:")
4 print("1. Segitiga")
5 print("2. Persegi")
6 print("3. Lingkaran")
7
8 menu_pilihan = int(input("Pilih menu (1/2/3): "))
9
10 if menu_pilihan == 1:
11     alas = float(input("Masukkan panjang alas segitiga: "))
12     tinggi = float(input("Masukkan tinggi segitiga: "))
13     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
14 elif menu_pilihan == 2:
15     sisi = float(input("Masukkan panjang sisi persegi: "))
16     hasil = rumus.hitung_luas(menu_pilihan, sisi)
17 elif menu_pilihan == 3:
18     jari = float(input("Masukkan jari-jari lingkaran: "))
19     hasil = rumus.hitung_luas(menu_pilihan, jari)
20 else:
21     print("Menu tidak valid")
22
23 print(f"Luas Bangun tersebut adalah adalah {hasil} cm")
Ln 18, Col 61 Spaces:4 UTF-8 CRLF Python 3.12.1 64-bit 3:22 AM
```

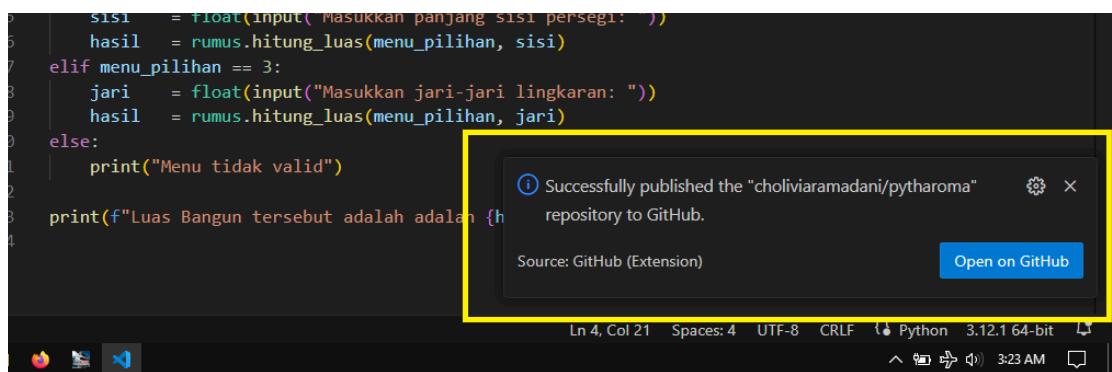
6. Lalu, klik **Commit** dan klik **Publish Branch**



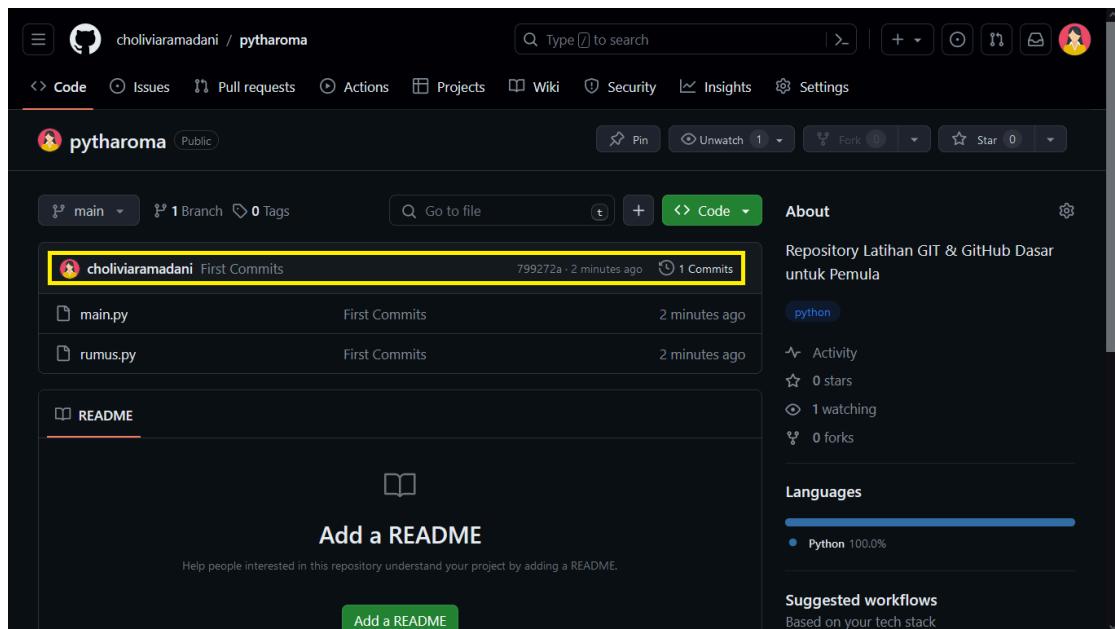
7. Pilih mau di publish secara private atau public, dan saya memilih public



8. Jika berhasil, maka terdapat notifikasi sukses



9. Saya klik tombol ***Open on Github*** dan cek apakah berhasil terpublish atau tidak. Dan ternyata berhasil.



B. Membuat Pull Request

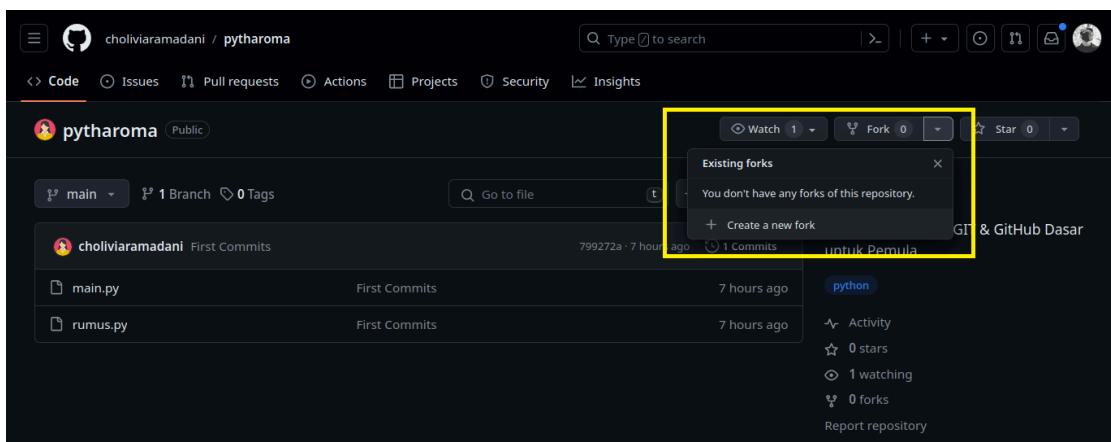
Suatu hari, Andika menemukan sebuah kode baru di Github milik Cholivia dan mengingat bahwa ia akan diundang untuk berkolaborasi dalam proyek ini. Namun, dia sepertinya lupa untuk mengundang Andika sebagai Collaborator Project di Repository.

Karena belum diundang, sebagai Collaborator Project, setiap perubahan yang ingin dilakukan di Repository tersebut harus disetujui secara manual dan di-merge secara manual oleh pemilik Repository. Permasalahan dan metode ini dalam GIT sering disebut sebagai Pembuatan Pull Request atau Pembuatan Persetujuan Penggabungan Perubahan.

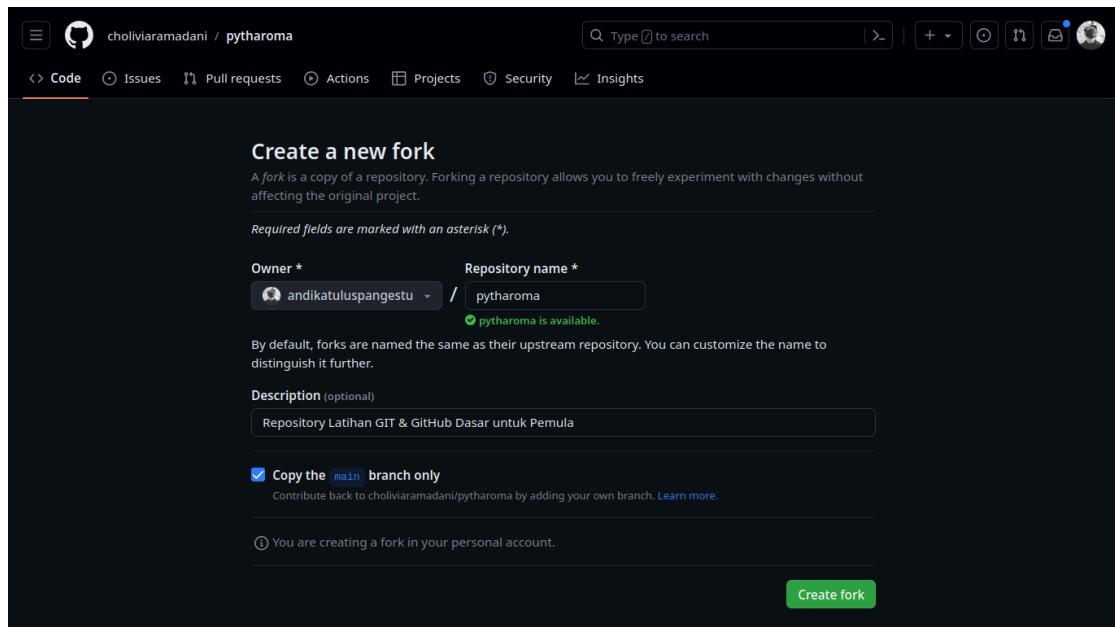
Metode ini hanya bisa dilakukan untuk berkolaborasi secara publik atau repository yang memiliki privasi publik, jika project yang kita kerjakan memiliki repository private maka kita perlu menggunakan fitur Invite Project Collaborator dan itu akan dibahas dalam bab selanjutnya.

Hal yang akan saya lakukan adalah sebagai berikut.

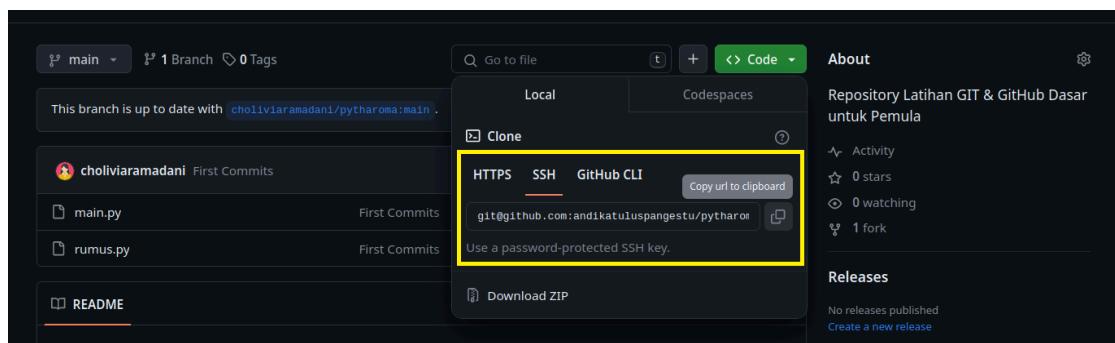
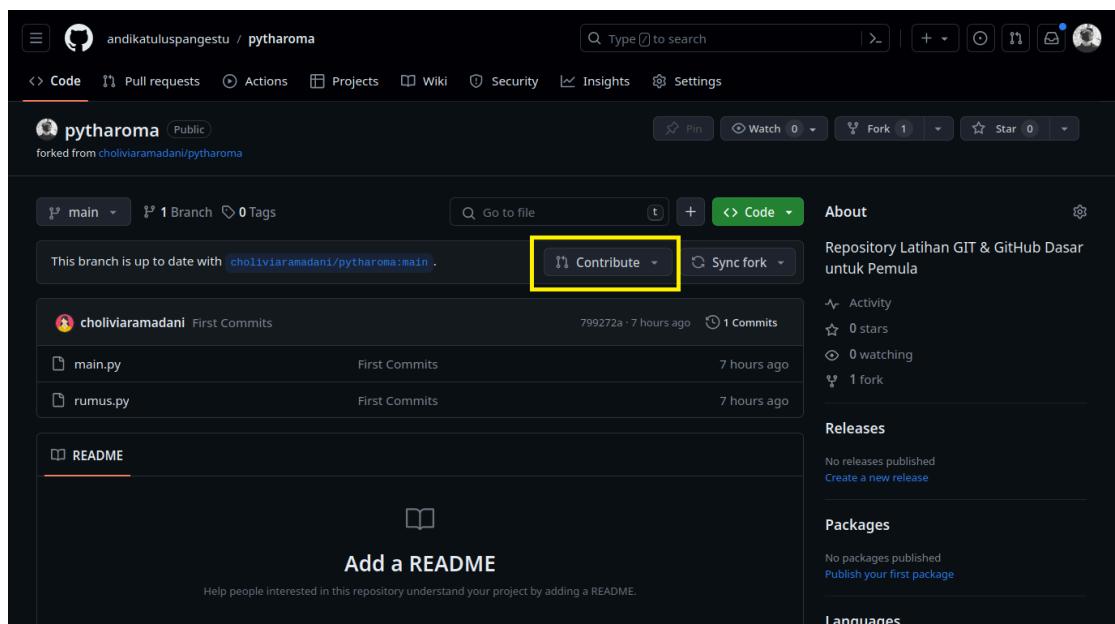
1. Membuka Repository dan buat Fork



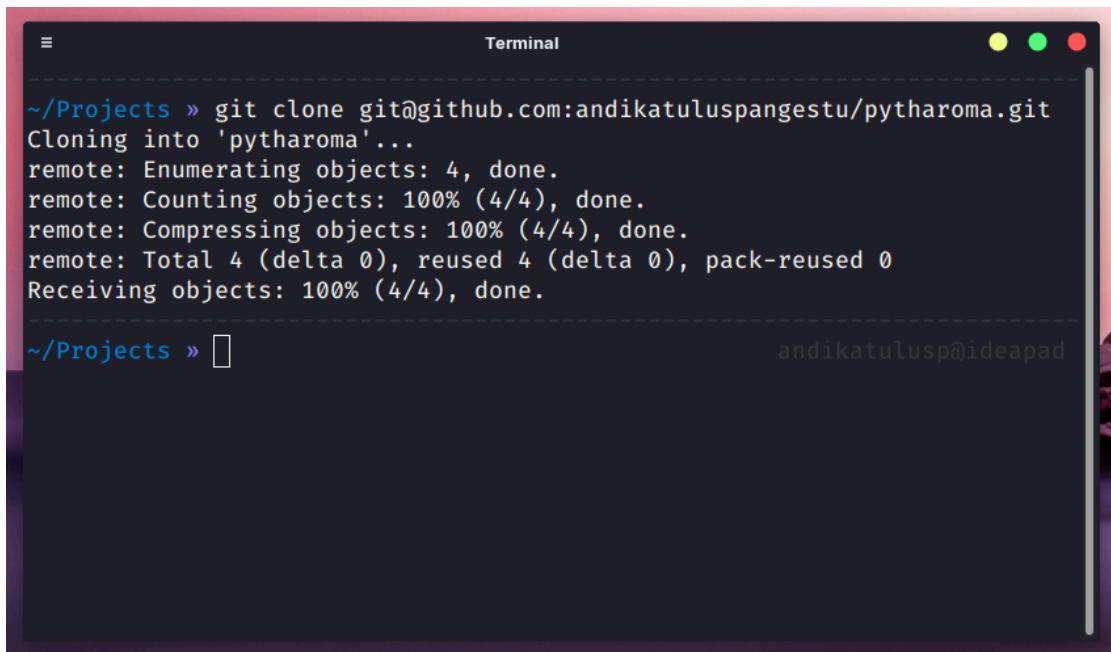
2. Mengisi detail Repo Fork



3. Setelah itu, klik **Contribute** dan Clone Repo Fork ke Lokal



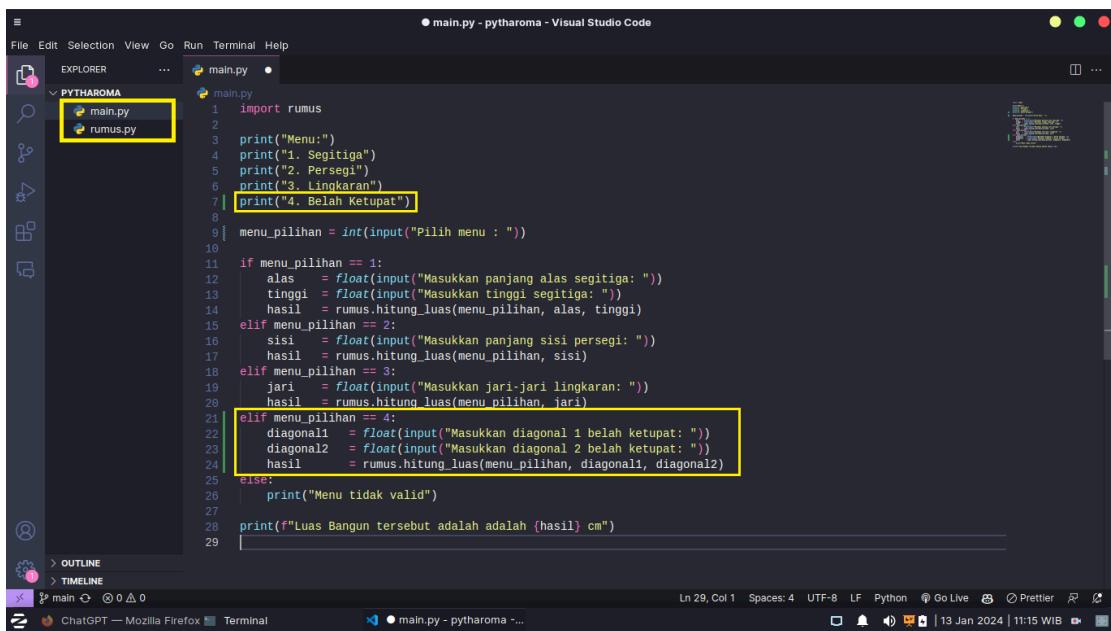
4. Lalu clone ke Komputer Lokal



```
~/Projects » git clone git@github.com:andikatuluspangestu/pytaroma.git
Cloning into 'pytaroma'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

~/Projects » 
```

5. Kemudian, saya akan membuat perubahan



```
File Edit Selection View Go Run Terminal Help
File Explorer ... main.py - pytaroma - Visual Studio Code
EXPLORER PYTHAROMA
main.py rumus.py
main.py
1 import rumus
2
3 print("Menu:")
4 print("1. Segitiga")
5 print("2. Persegi")
6 print("3. Lingkaran")
7 print("4. Belah Ketupat")
8
9 menu_pilihan = int(input("Pilih menu : "))
10
11 if menu_pilihan == 1:
12     alas = float(input("Masukkan panjang alas segitiga: "))
13     tinggi = float(input("Masukkan tinggi segitiga: "))
14     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
15 elif menu_pilihan == 2:
16     sisi = float(input("Masukkan panjang sisi persegi: "))
17     hasil = rumus.hitung_luas(menu_pilihan, sisi)
18 elif menu_pilihan == 3:
19     jari = float(input("Masukkan jari-jari lingkaran: "))
20     hasil = rumus.hitung_luas(menu_pilihan, jari)
21 elif menu_pilihan == 4:
22     diagonal1 = float(input("Masukkan diagonal 1 belah ketupat: "))
23     diagonal2 = float(input("Masukkan diagonal 2 belah ketupat: "))
24     hasil = rumus.hitung_luas(menu_pilihan, diagonal1, diagonal2)
25 else:
26     print("Menu tidak valid")
27
28 print(f"Luas Bangun tersebut adalah adalah {hasil} cm")
29 
```

```
File Edit Selection View Go Run Terminal Help
EXPLORER rumus.py
PYTHAROMA main.py rumus.py
rumus.py
1 def hitung_luas_segitiga(alas, tinggi):
2     return 0.5 * alas * tinggi
3
4 def hitung_luas_persegi(panjang_sisi):
5     return panjang_sisi ** 2
6
7 def hitung_luas_lingkaran(jari_jari):
8     return 3.14 * jari_jari ** 2
9
10 def hitung_luas_belah_ketupat(diagonal1, diagonal2):
11     return 0.5 * diagonal1 * diagonal2
12
13 def hitung_luas(menu, *args):
14     if menu == 1:
15         return hitung_luas_segitiga(*args)
16     elif menu == 2:
17         return hitung_luas_persegi(*args)
18     elif menu == 3:
19         return hitung_luas_lingkaran(*args)
20     elif menu == 4:
21         return hitung_luas_belah_ketupat(*args)
22     else:
23         return "Menu tidak valid"
Ln 23, Col 34  Spaces: 4  UTF-8  LF  Python  Go Live  Prettier
ChatGPT — Mozilla Firefox  Terminal  rumus.py - pytharoma...
```

6. Lalu push perubahan ke Github Repo Fork

```
File Edit Selection View Go Run Terminal Help
SOURCE CONTROL main.py M X rumus.py M
main.py
1 import rumus
2
3 print("Menu:")
4 print("1. Segitiga")
5 print("2. Persegi")
6 print("3. Lingkaran")
7 print("4. Belah Ketupat")
8
9 menu_pilihan = int(input("Pilih menu : "))
10
11 if menu_pilihan == 1:
12     alas = float(input("Masukkan panjang alas segitiga: "))
13     tinggi = float(input("Masukkan tinggi segitiga: "))
14     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
15 elif menu_pilihan == 2:
16     sisi = float(input("Masukkan panjang sisi persegi: "))
17     hasil = rumus.hitung_luas(menu_pilihan, sisi)
18 elif menu_pilihan == 3:
19     jari = float(input("Masukkan jari-jari lingkaran: "))
20     hasil = rumus.hitung_luas(menu_pilihan, jari)
21 elif menu_pilihan == 4:
22     diagonal1 = float(input("Masukkan diagonal 1 belah ketupat: "))
23     diagonal2 = float(input("Masukkan diagonal 2 belah ketupat: "))
24     hasil = rumus.hitung_luas(menu_pilihan, diagonal1, diagonal2)
25 else:
26     print("Menu tidak valid")
Ln 29, Col 1  Spaces: 4  UTF-8  LF  Python  Go Live  Prettier
ChatGPT — Mozilla Firefox  main.py - pytharoma - Vi...  Terminal
```

```
File Edit Selection View Go Run Terminal Help
SOURCE CONTROL main.py X rumus.py
main.py
1 import rumus
2
3 print("Menu:")
4 print("1. Segitiga")
5 print("2. Persegi")
6 print("3. Lingkaran")
7 print("4. Belah Ketupat")
8
9 menu_pilihan = int(input("Pilih menu : "))
10
11 if menu_pilihan == 1:
12     alas = float(input("Masukkan panjang alas segitiga: "))
13     tinggi = float(input("Masukkan tinggi segitiga: "))
14     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
15 elif menu_pilihan == 2:
16     sisi = float(input("Masukkan panjang sisi persegi: "))
17     hasil = rumus.hitung_luas(menu_pilihan, sisi)
18 elif menu_pilihan == 3:
19     jari = float(input("Masukkan jari-jari lingkaran: "))
20     hasil = rumus.hitung_luas(menu_pilihan, jari)
21 elif menu_pilihan == 4:
22     diagonal1 = float(input("Masukkan diagonal 1 belah ketupat: "))
23     diagonal2 = float(input("Masukkan diagonal 2 belah ketupat: "))
24     hasil = rumus.hitung_luas(menu_pilihan, diagonal1, diagonal2)
25 else:
26     print("Menu tidak valid")
Ln 29, Col 1  Spaces: 4  UTF-8  LF  Python  Go Live  Prettier
ChatGPT — Mozilla Firefox  main.py - pytharoma - Vi...  Terminal
```

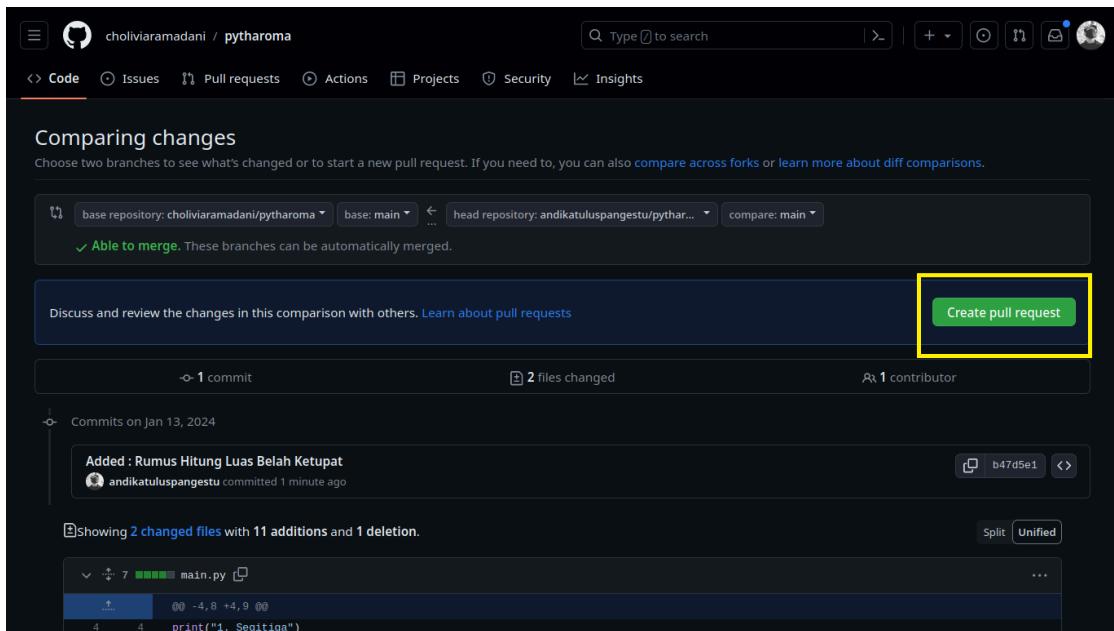
7. Kemudian kita lihat apakah sudah terpush atau tidak

The screenshot shows a GitHub repository page for 'pytharoma'. At the top, there are navigation links: Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the header, it says 'pytharoma' (Public) and 'forked from choliviaramadani/pytharoma'. The main area shows a 'main' branch with 1 Branch and 0 Tags. A commit card is highlighted with a yellow border, showing a commit from 'andikatuluspangestu' added 'Rumus Hitung Luas Belah Ketupat' at 'b47d5e1 · 1 minute ago'. The commit has 2 Commits. On the right side, there's an 'About' section with a description: 'Repository Latihan GIT & GitHub Dasar untuk Pemula'. It also shows Activity (0 stars, 0 watching, 1 fork), a 'Contribute' button, and a 'Sync fork' button.

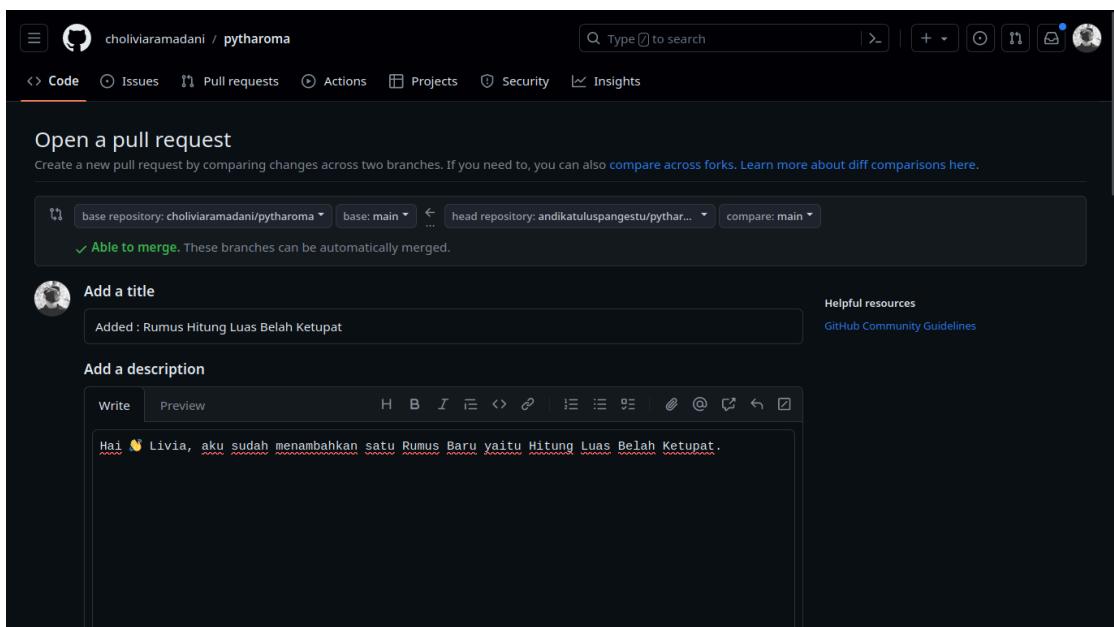
8. Lalu, saya akan mengirimkan perubahan tersebut ke Repo Utama dan mengirimkan pull request. Klik tombol **Contribute** dan klik **Open Pull Request**

The screenshot shows the same GitHub repository page for 'pytharoma'. A modal window is open over the repository details. The modal title is 'This branch is 1 commit ahead of choliviaramadani/pytharoma:main'. It contains the text 'Open a pull request to contribute your changes upstream.' and a large green 'Open pull request' button. The background repository details are visible, including the commit card from step 7 and the 'About' section. The 'About' section includes a description: 'Repository Latihan GIT & GitHub Dasar untuk Pemula', activity metrics (0 stars, 0 watching, 1 fork), and sections for Releases, Packages, and Languages.

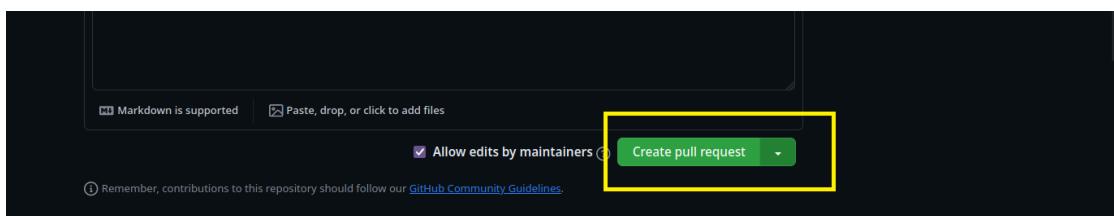
9. Cek kode dan klik *Create pull request*



10. Saya juga akan menambahkan pesan atau deskripsi pull request



11. Kemudian, klik tombol *Create pull request*



12. Setelah berhasil maka akan muncul halaman seperti ini.

choliviaramadani / pytharoma

Pull requests 1

Added : Rumus Hitung Luas Belah Ketupat #1

Open andikatuluspangestu... wants to merge 1 commit into `choliviaramadani:main` from `andikatuluspangestu:main`

Conversation 0 Commits 1 Checks 0 Files changed 2

andikatuluspangestu commented now

Hai 🐱 Livia, aku sudah menambahkan satu Rumus Baru yaitu Hitung Luas Belah Ketupat.

Added : Rumus Hitung Luas Belah Ketupat b47d5e1

Add more commits by pushing to the `main` branch on [andikatuluspangestu/pytharoma](#).

This branch has no conflicts with the base branch
Only those with [write access](#) to this repository can merge pull requests.

Reviewers
No reviews
Still in progress? Convert to draft

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

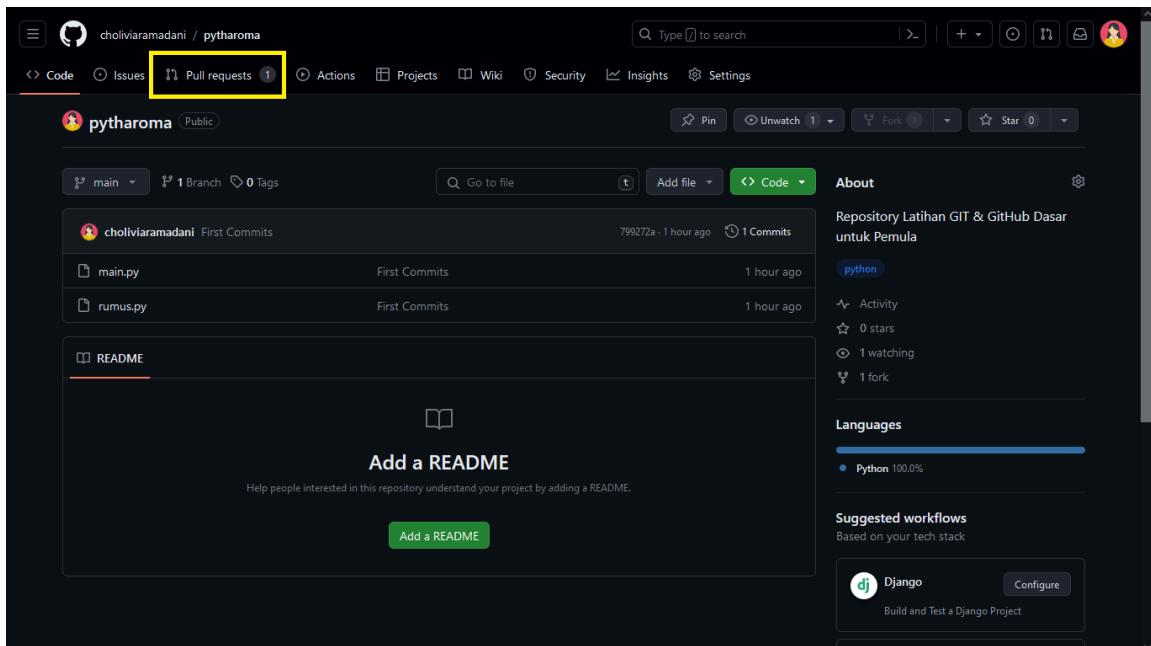
Add a comment

Write Preview

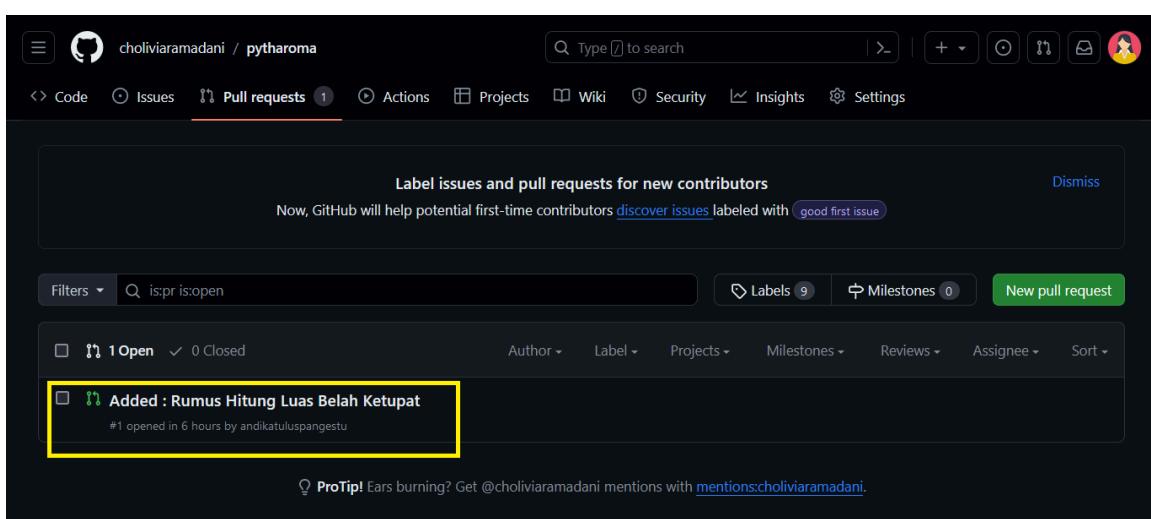
C. Melakukan Merge

Setelah Andika mengirimkan permintaan penggabungan perubahan atau pull request, maka dari itu seorang pemilik repository utama yaitu Cholivia harus melakukan pengecekan dan merge. Berikut adalah langkah-langkah yang dilakukan.

1. Membuka tautan repository project dan klik tab **pull request**



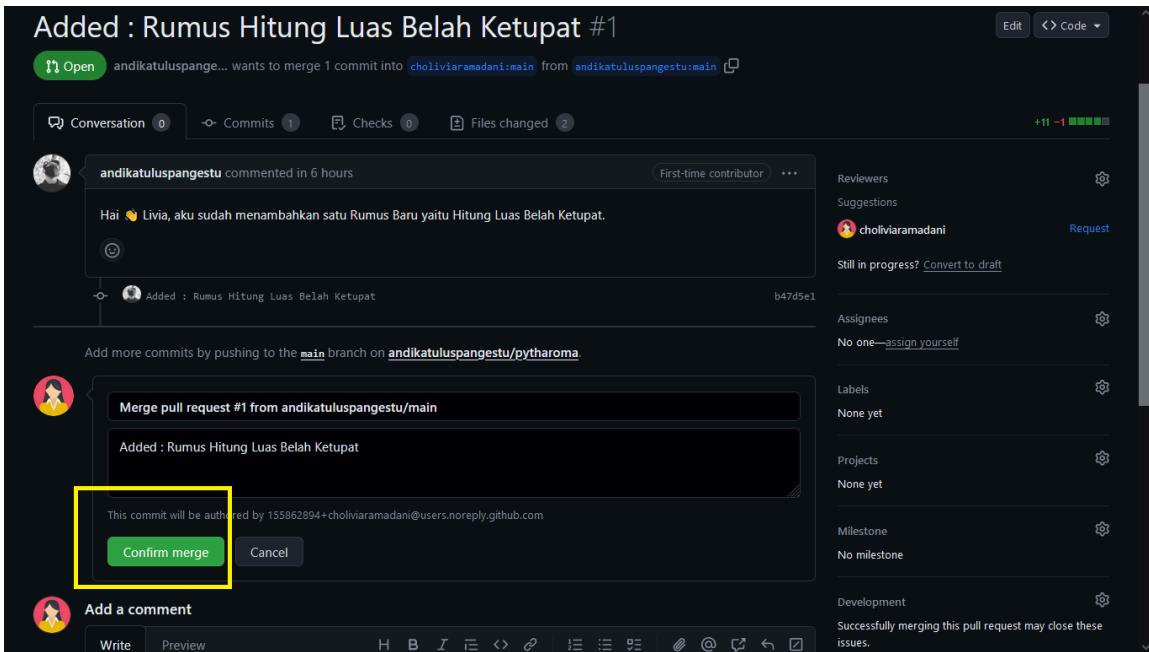
2. Lalu, saya akan membuka pesan permintaan atau pull request



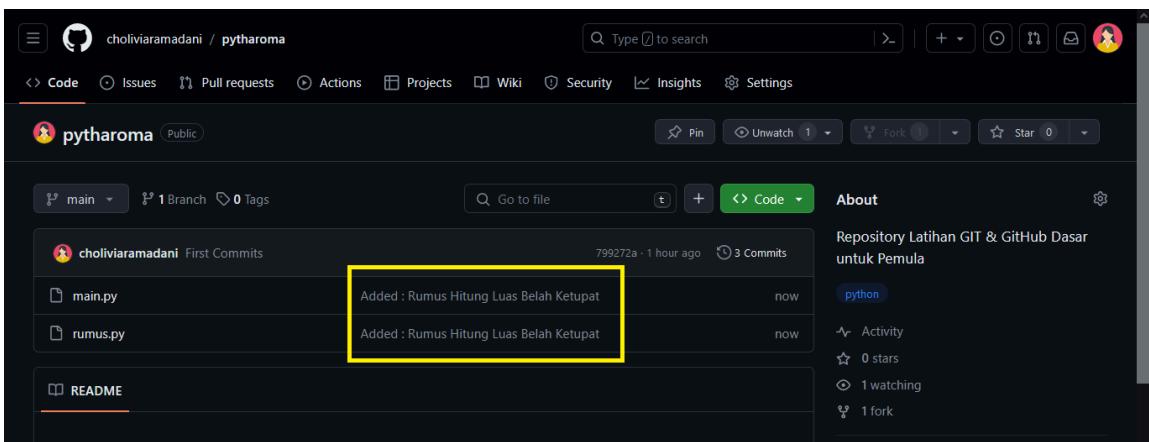
3. Kemudian klik judul pull request tersebut, untuk melihat detail

4. Setelah saya cek dan kode aman untuk digabungkan, maka akan saya tekan tombol **Merge Pull Request**

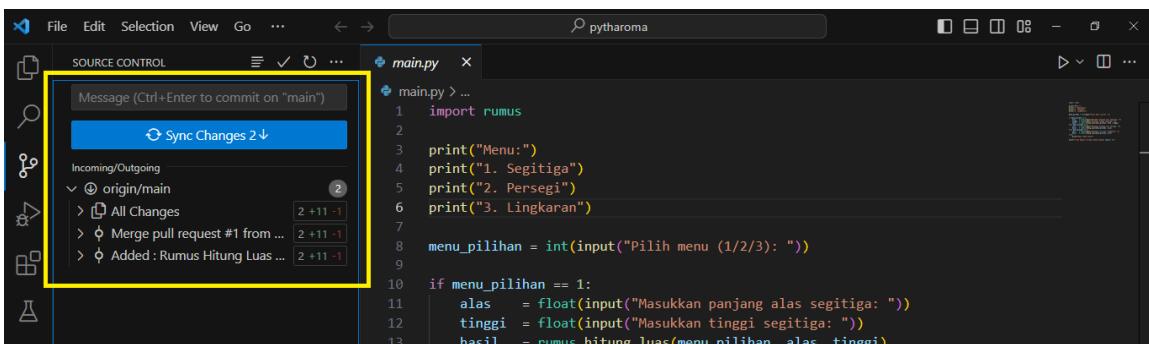
5. Di halaman ini, kita juga bisa memberikan balasan pesan, dan kalau sudah klik tombol **Confirm Merge**



6. Lalu saya akan mengecek apakah perubahan berhasil digabungkan?



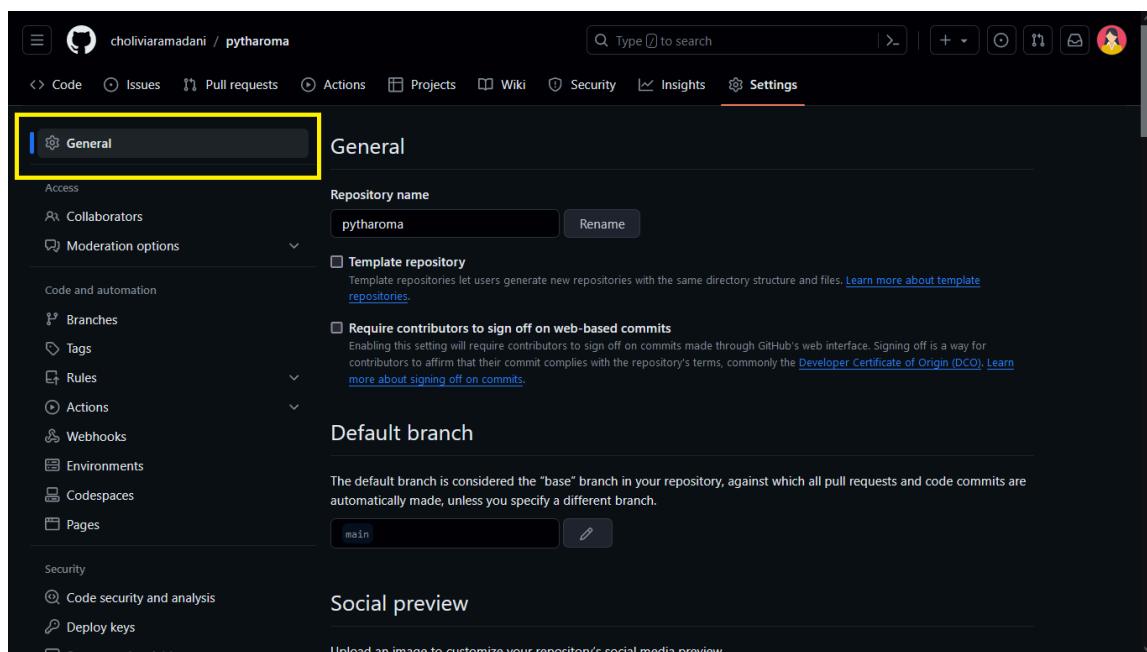
7. Terakhir, saya akan mengambil perubahan ke komputer lokal



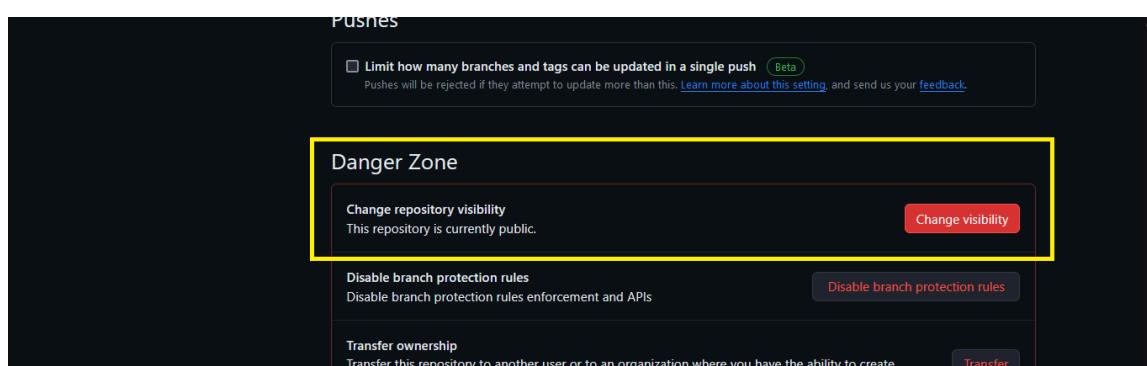
D. Mengubah Privasi Repo ke Privat

Suatu saat, ketika proyek ini terus berkembang, Cholivia berencana untuk mengubah status privasi dari repositori utama menjadi privat. Hal ini bertujuan agar tidak ada orang sembarangan yang dapat melakukan perubahan atau mengirim permintaan perubahan, seperti saat repositori dikonfigurasi sebagai publik. Untuk melakukan perubahan visibilitas repository menjadi privat, Cholivia menjalankan beberapa langkah, seperti yang dijelaskan di bawah ini.

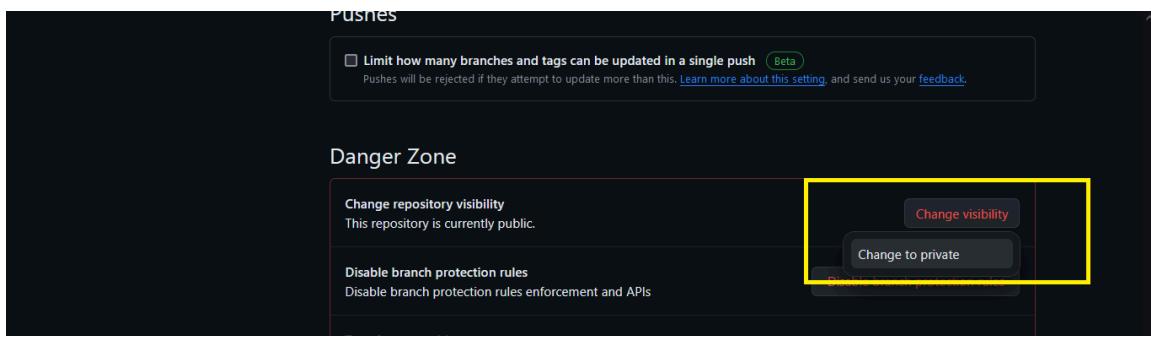
1. Klik **Settings** dan buka Tab **General**



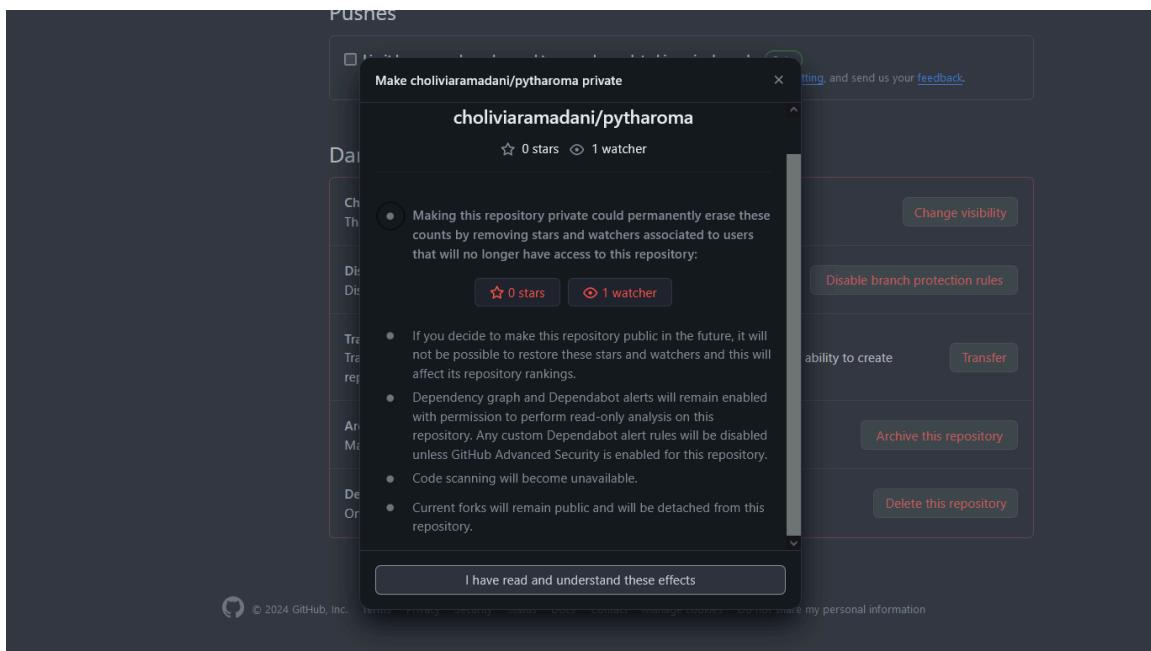
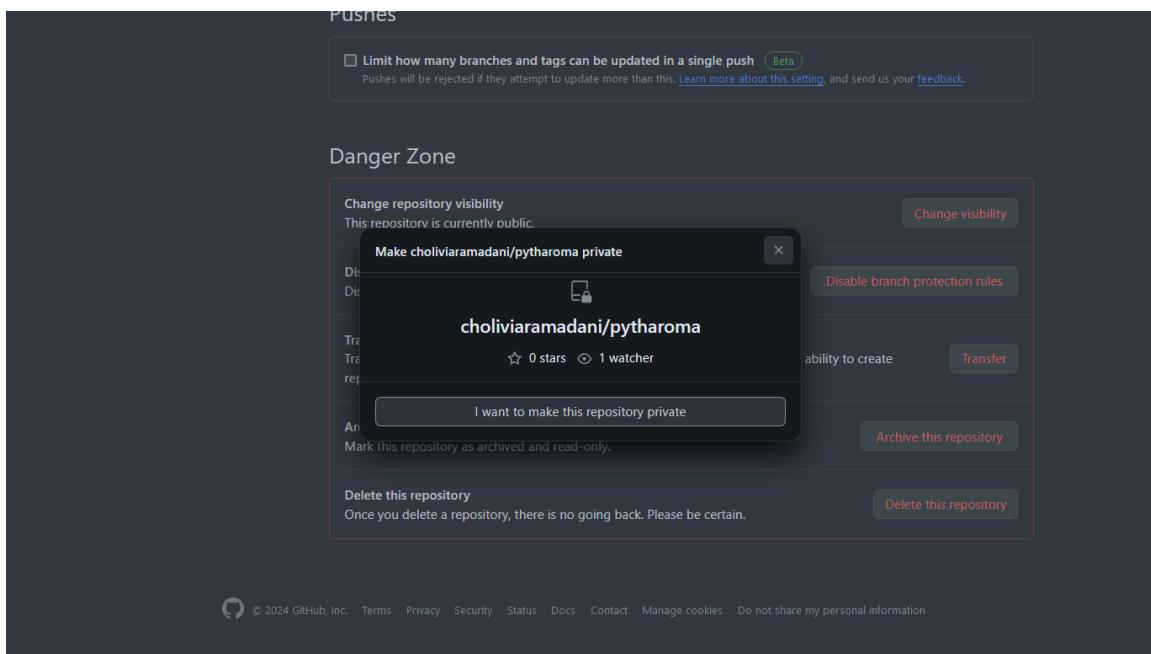
2. Lalu scroll kebawah sampai menemukan bagian **Danger Zone** dan klik tombol **Change Visibility**



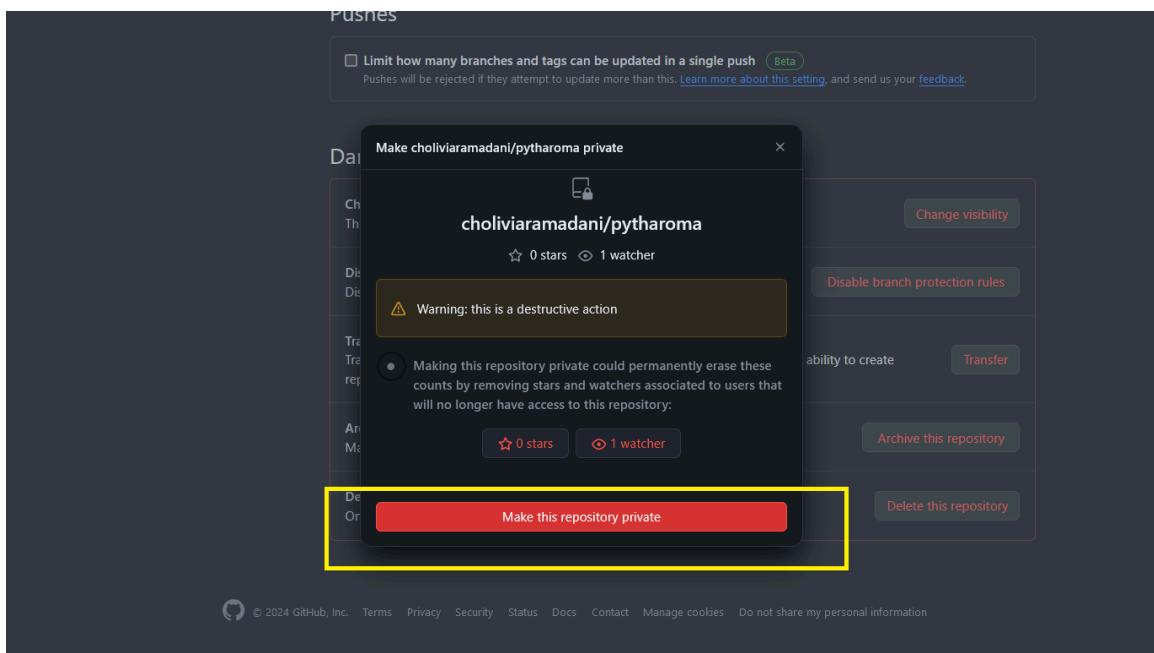
3. Kemudian klik “Change to Private”



4. Setujui beberapa pernyataan yang muncul dalam pop-up



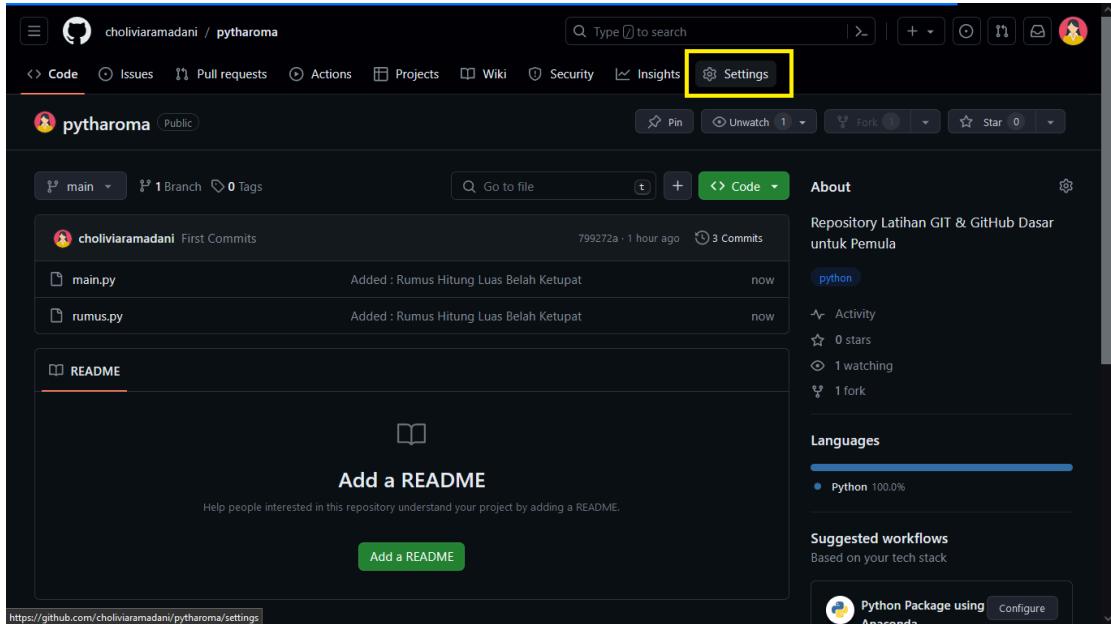
5. Lalu, klik tombol merah seperti di gambar untuk mengubah repo ke privat



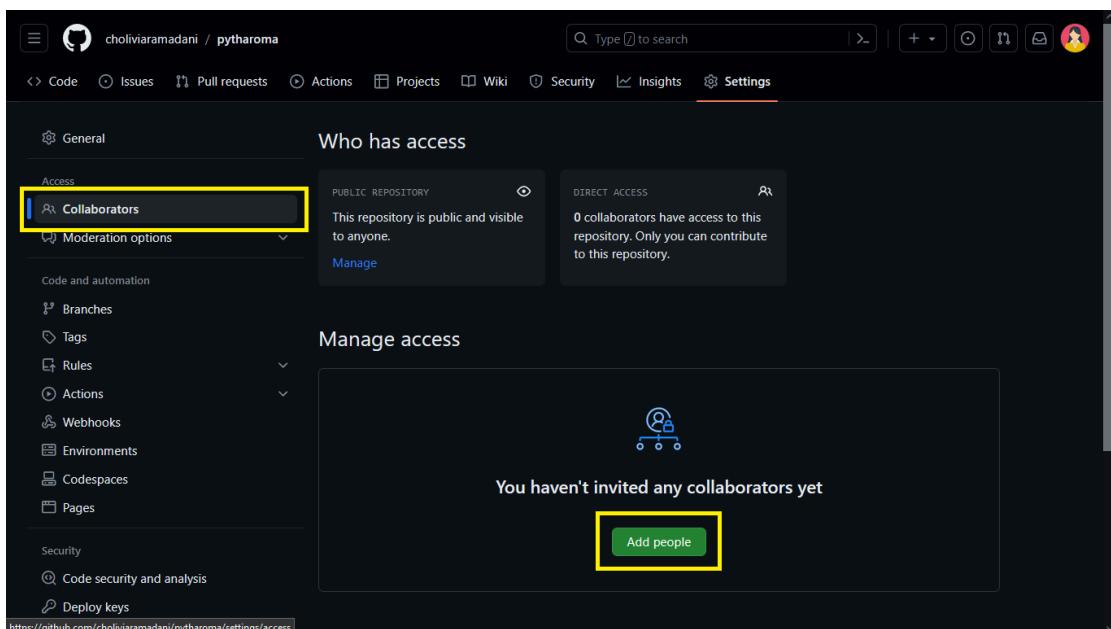
E. Mengundang Kolaborator

Dikarenakan repository telah diubah visibilitasnya menjadi privat, maka cholivia perlu mengkonfigurasi kolaborator sehingga dapat berkolaborasi dengan Andika dalam repository utama.

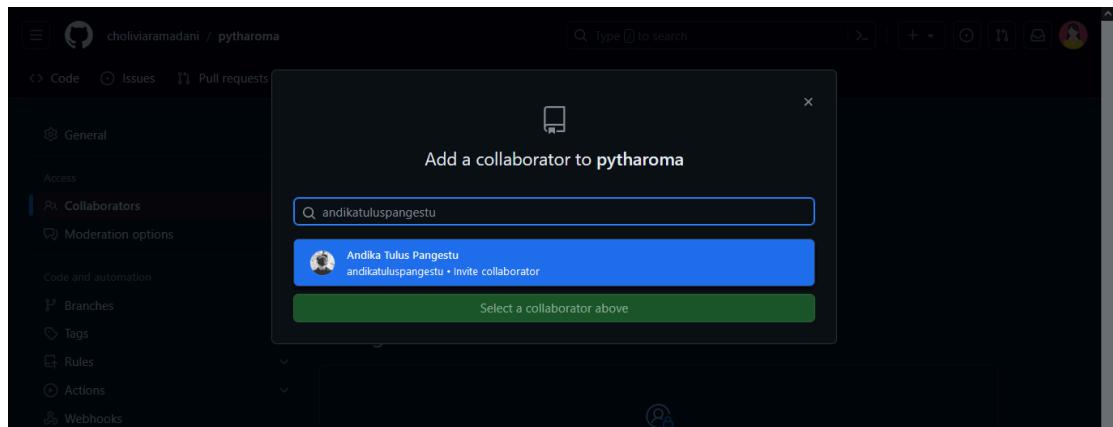
1. Klik menu **Settings** dan pilih tab **Collaborators**



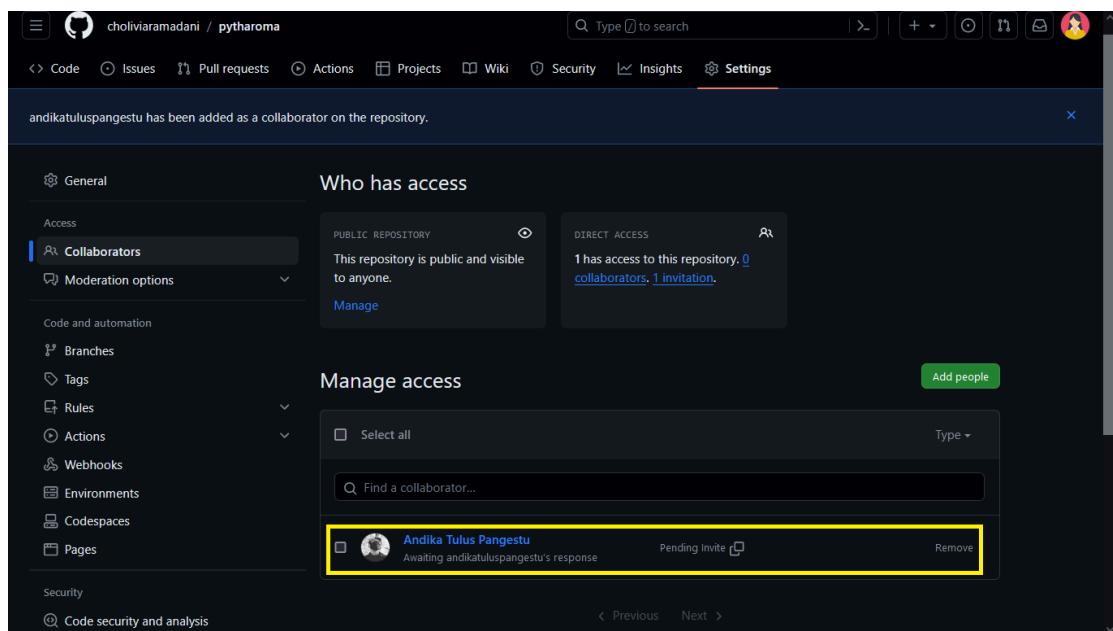
2. Lalu, klik tombol **Add People** untuk menambah kolaborator



3. Masukan username yang akan ditambahkan sebagai kolaborator



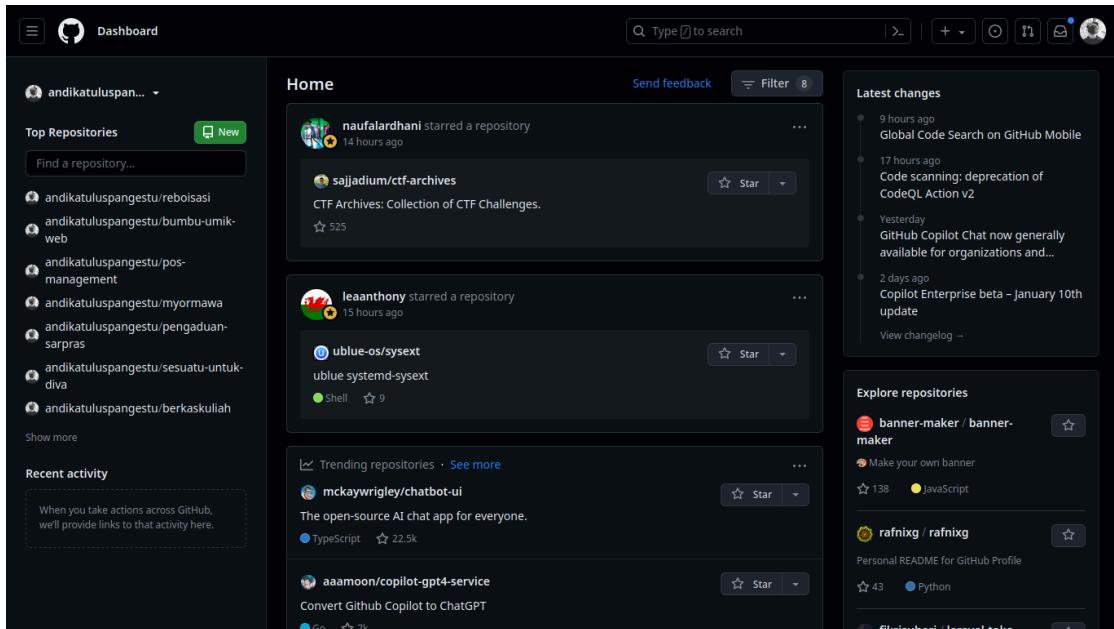
4. Setelah berhasil, maka akan terkirim permintaan kolaborasi dan akan terlihat status nya masih pending. Andika perlu menyetujui permintaan kolaborasi melalui akun githubnya.



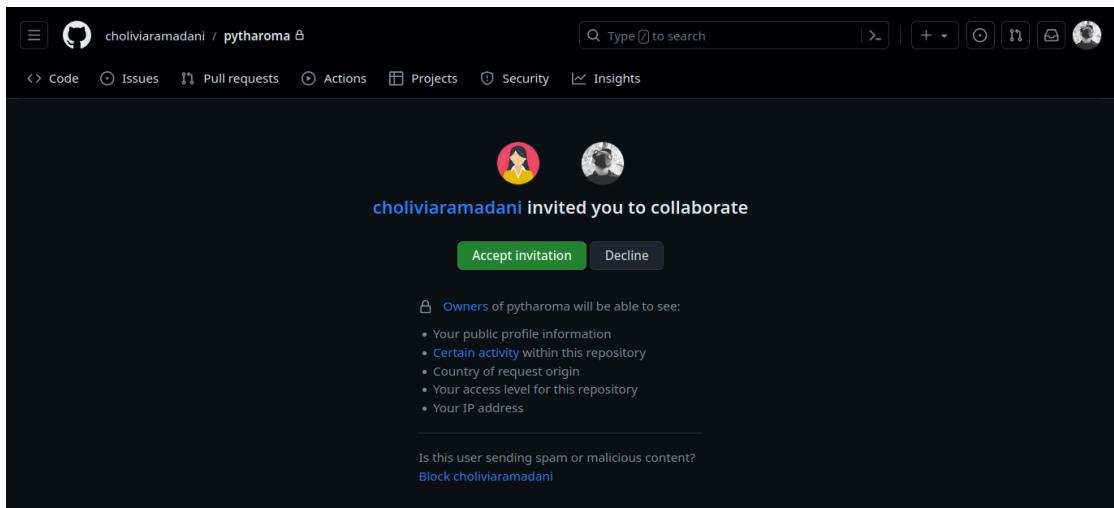
F. Menerima Permintaan Kolaborator

Sebelumnya, Cholivia telah berhasil mengubah visibilitas proyek dan mengirimkan permintaan kolaborasi kepada Andika. Kali ini, Andika akan melihat, menerima, dan berkontribusi dalam proyek kolaboratif bersama Cholivia dalam repositori privat.

1. Buka dashboard utama github



2. Lalu, buka tautan repository utama dan maka akan muncul halaman “Accept Invitation”, klik tombol **Accept Invitation** untuk menerima



3. Jika berhasil, maka akan terlihat di daftar kolaborator

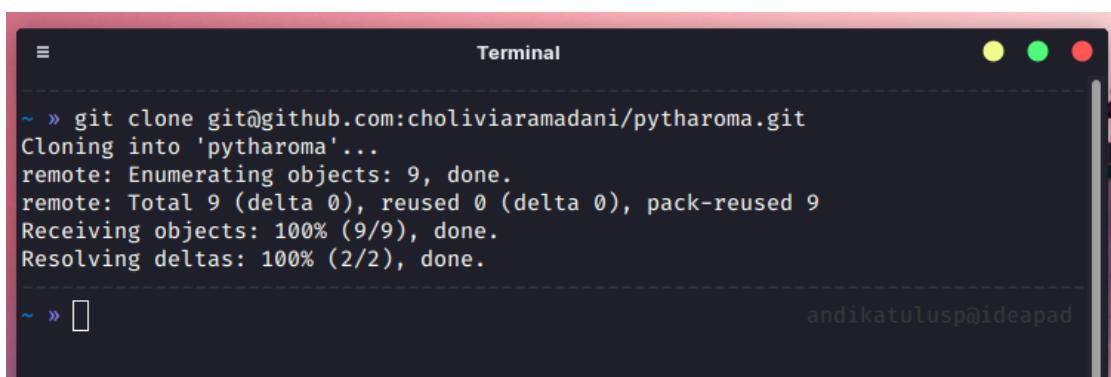
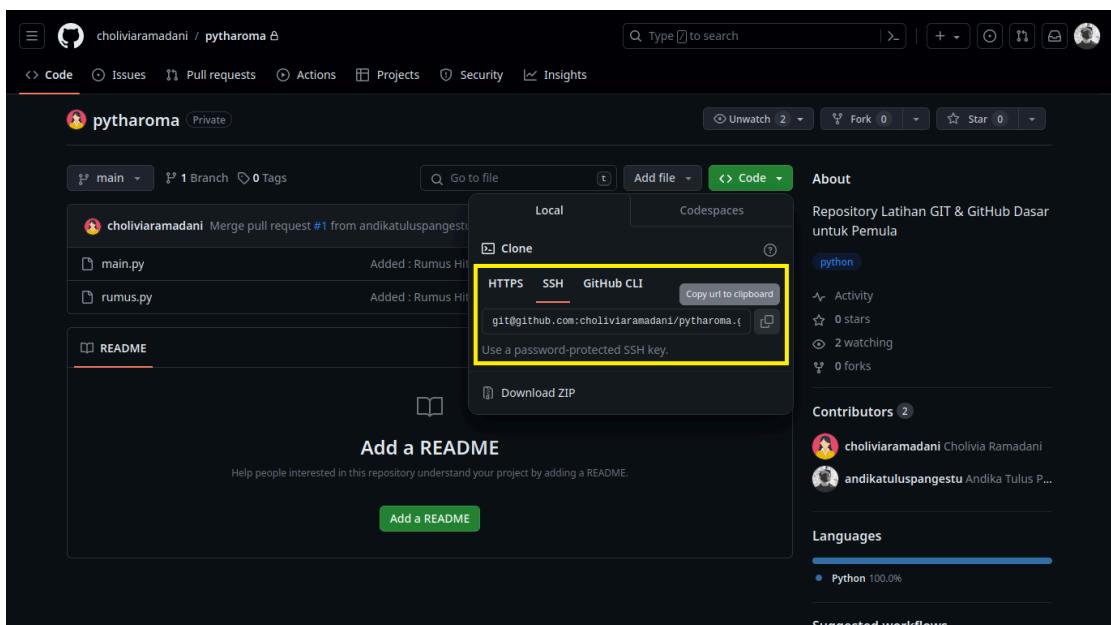
The screenshot shows a GitHub repository page for 'pytharoma' owned by 'cholivaramadani'. At the top, a message says 'You now have push access to the cholivaramadani/pytharoma repository.' The repository has 1 branch and 0 tags. It contains two files: 'main.py' and 'rumus.py', both added 1 hour ago. The 'About' section indicates it's a private repository for 'Repository Latihan GIT & GitHub Dasar untuk Pemula' and is tagged as 'Python'. The 'Activity' section shows 0 stars, 2 watching, and 0 forks. The 'Contributors' section lists 'cholivaramadani' and 'andikatuluspangestu'. The 'Languages' section shows Python at 100.0%.

G. Membuat Perubahan sebagai Kolaborator

Ketika berkolaborasi dalam repositori privat dan sudah dikonfigurasi sebagai kolaborator, prosesnya menjadi lebih mudah. Kolaborator tidak perlu melakukan pull request secara manual, melainkan pemilik utama repository dapat langsung menggabungkan perubahan yang diberikan oleh kolaborator.

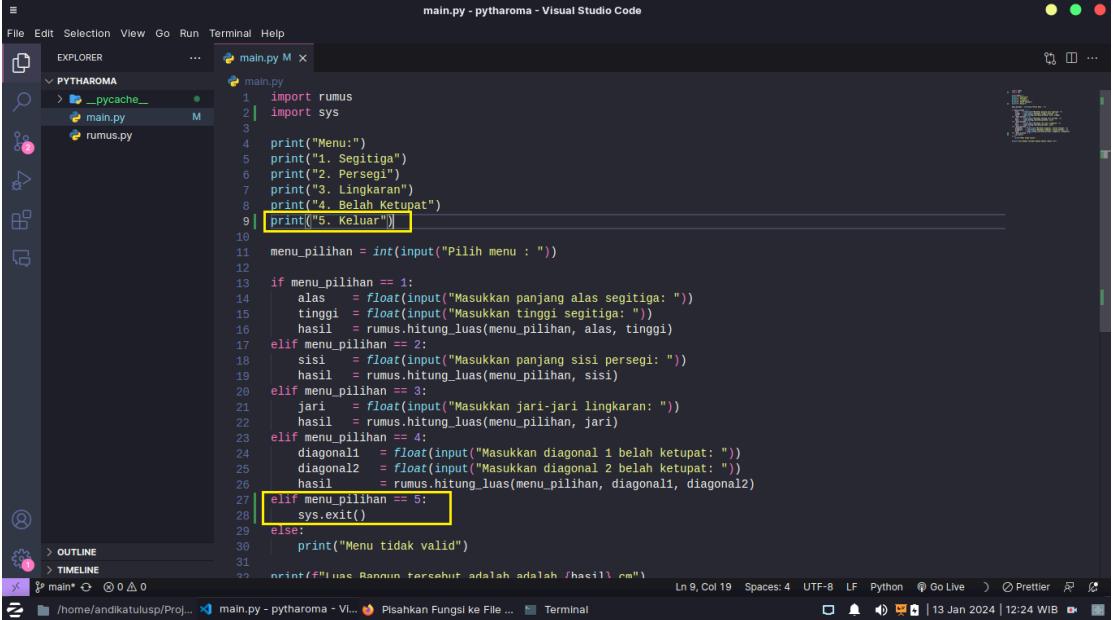
Berikut adalah langkah-langkah yang dilakukan Andika dalam berkolaborasi project sebagai kontributor.

1. Clone repository utama ke komputer lokal



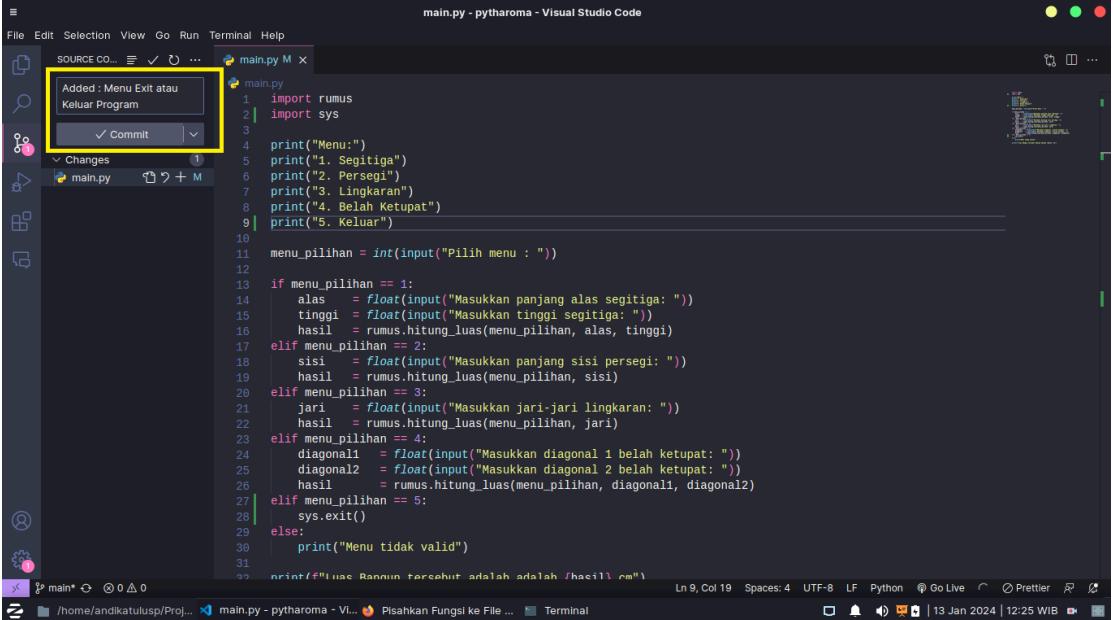
2. Kemudian, saya akan menambah perubahan yaitu menambah menu

Keluar Aplikasi



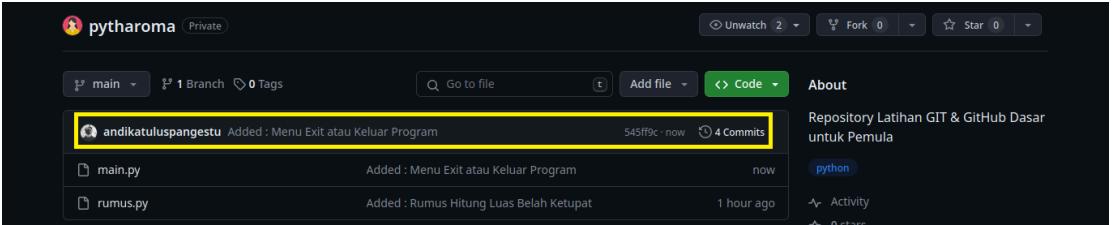
```
main.py - pytharoma - Visual Studio Code
File Edit Selection View Go Run Terminal Help
PYTHAROMA
main.py M
main.py
1 import rumus
2 import sys
3
4 print("Menu:")
5 print("1. Segitiga")
6 print("2. Persegi")
7 print("3. Lingkaran")
8 print("4. Belah Ketupat")
9 print("5. Keluar")
10
11 menu_pilihan = int(input("Pilih menu : "))
12
13 if menu_pilihan == 1:
14     alas = float(input("Masukkan panjang alas segitiga: "))
15     tinggi = float(input("Masukkan tinggi segitiga: "))
16     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
17 elif menu_pilihan == 2:
18     sisi = float(input("Masukkan panjang sisi persegi: "))
19     hasil = rumus.hitung_luas(menu_pilihan, sisi)
20 elif menu_pilihan == 3:
21     jari = float(input("Masukkan jari-jari lingkaran: "))
22     hasil = rumus.hitung_luas(menu_pilihan, jari)
23 elif menu_pilihan == 4:
24     diagonal1 = float(input("Masukkan diagonal 1 belah ketupat: "))
25     diagonal2 = float(input("Masukkan diagonal 2 belah ketupat: "))
26     hasil = rumus.hitung_luas(menu_pilihan, diagonal1, diagonal2)
27 elif menu_pilihan == 5:
28     sys.exit()
29 else:
30     print("Menu tidak valid")
31
32 print(f"Luas Ruang tersebut adalah adalah {hasil} cm²")
Ln 9, Col 19 Spaces: 4 UTF-8 LF Python Go Live Prettier
/home/andikatulusp/Proj... main.py - pytharoma - Vi... Pisahkan Fungsi ke File ... Terminal
D N M | 13 Jan 2024 | 12:24 WIB
```

3. Klik dan masukan pesan perubahan untuk push perubahan ke github



```
main.py - pytharoma - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Changes
main.py M
main.py
1 import rumus
2 import sys
3
4 print("Menu:")
5 print("1. Segitiga")
6 print("2. Persegi")
7 print("3. Lingkaran")
8 print("4. Belah Ketupat")
9 print("5. Keluar")
10
11 menu_pilihan = int(input("Pilih menu : "))
12
13 if menu_pilihan == 1:
14     alas = float(input("Masukkan panjang alas segitiga: "))
15     tinggi = float(input("Masukkan tinggi segitiga: "))
16     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
17 elif menu_pilihan == 2:
18     sisi = float(input("Masukkan panjang sisi persegi: "))
19     hasil = rumus.hitung_luas(menu_pilihan, sisi)
20 elif menu_pilihan == 3:
21     jari = float(input("Masukkan jari-jari lingkaran: "))
22     hasil = rumus.hitung_luas(menu_pilihan, jari)
23 elif menu_pilihan == 4:
24     diagonal1 = float(input("Masukkan diagonal 1 belah ketupat: "))
25     diagonal2 = float(input("Masukkan diagonal 2 belah ketupat: "))
26     hasil = rumus.hitung_luas(menu_pilihan, diagonal1, diagonal2)
27 elif menu_pilihan == 5:
28     sys.exit()
29 else:
30     print("Menu tidak valid")
31
32 print(f"Luas Ruang tersebut adalah adalah {hasil} cm²")
Ln 9, Col 19 Spaces: 4 UTF-8 LF Python Go Live Prettier
/home/andikatulusp/Proj... main.py - pytharoma - Vi... Pisahkan Fungsi ke File ... Terminal
D N M | 13 Jan 2024 | 12:25 WIB
```

4. Lalu, setelah itu kita cek di github



pytharoma · Private

main · 1 Branch · 0 Tags

Go to file Add file < Code

andikatuluspengestu Added : Menu Exit atau Keluar Program 545ff9c · now 4 Commits

main.py Added : Menu Exit atau Keluar Program now

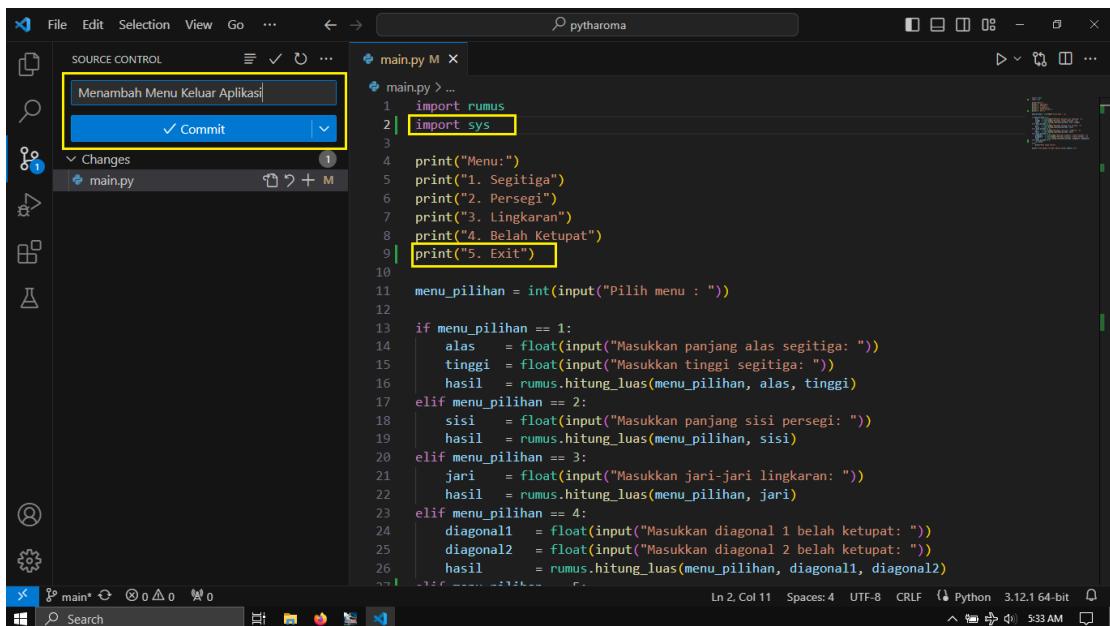
rumus.py Added : Rumus Hitung Luas Belah Ketupat 1 hour ago

About Repository Latihan GIT & GitHub Dasar untuk Pemula python Activity 0 stars

H. Konflik dalam Kolaborasi

Pada suatu waktu, terjadi ketidaksepahaman antara Andika dan Cholivia. Cholivia tidak menyadari bahwa Andika telah melakukan perubahan pada proyeknya. Akibatnya, terjadi konflik pada kode. Kejadian ini disebabkan oleh perubahan pada baris kode yang sama dalam rentang waktu yang cukup dekat. Konflik ini akan terdeteksi saat proses penggabungan perubahan.

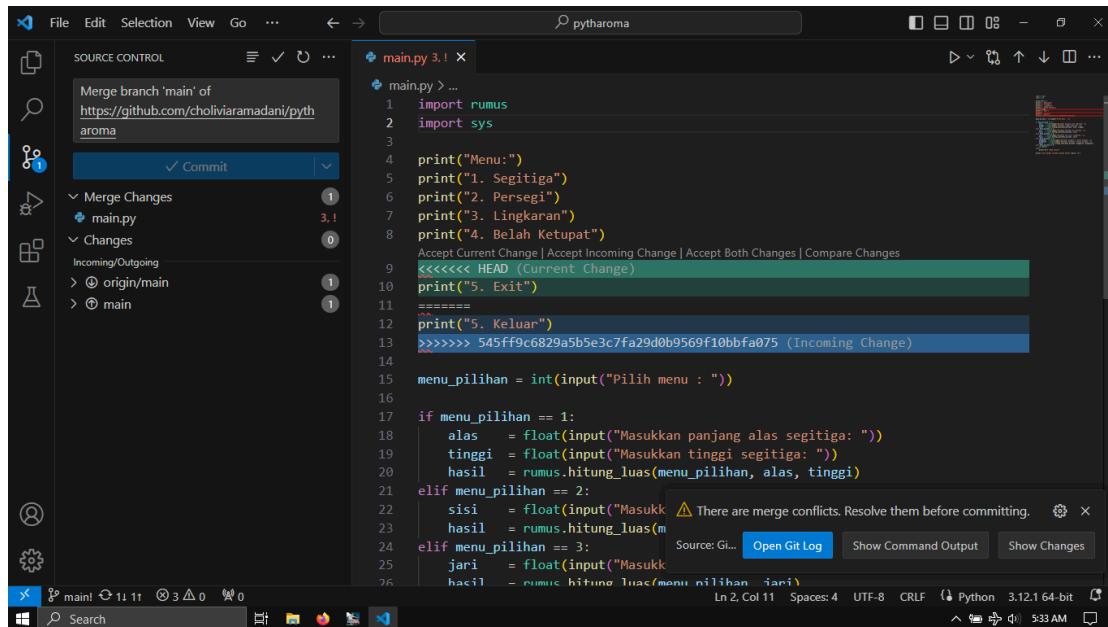
1. Cholivia membuat perubahan dengan menambahkan menu pilihan **Keluar Aplikasi**, yang secara tidak disadari telah dibuat oleh andika sebelumnya.



The screenshot shows a GitHub commit interface. In the commit message input field, the text "Menambah Menu Keluar Aplikasi" is entered. Below the message, there is a "Commit" button. The code editor shows a Python file named "main.py" with the following content:

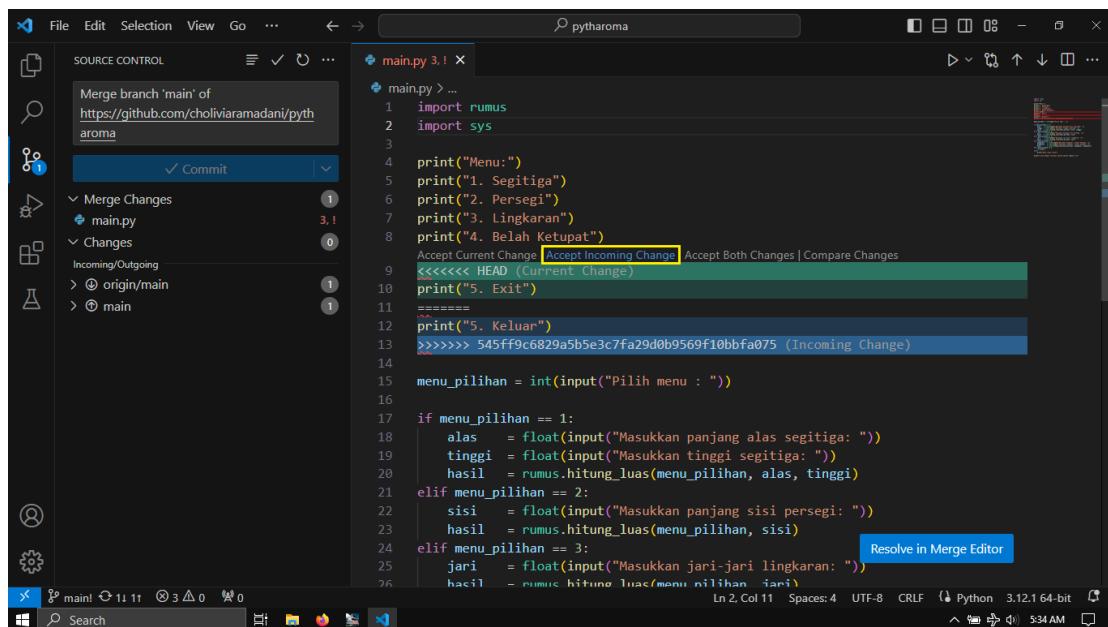
```
1 import rumus
2 import sys
3
4 print("Menu:")
5 print("1. Segitiga")
6 print("2. Persegi")
7 print("3. Lingkaran")
8 print("4. Belah Ketupat")
9 print("5. Exit")
10
11 menu_pilihan = int(input("Pilih menu : "))
12
13 if menu_pilihan == 1:
14     alas = float(input("Masukkan panjang alas segitiga: "))
15     tinggi = float(input("Masukkan tinggi segitiga: "))
16     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
17 elif menu_pilihan == 2:
18     sisi = float(input("Masukkan panjang sisi persegi: "))
19     hasil = rumus.hitung_luas(menu_pilihan, sisi)
20 elif menu_pilihan == 3:
21     jari = float(input("Masukkan jari-jari lingkaran: "))
22     hasil = rumus.hitung_luas(menu_pilihan, jari)
23 elif menu_pilihan == 4:
24     diagonal1 = float(input("Masukkan diagonal 1 belah ketupat: "))
25     diagonal2 = float(input("Masukkan diagonal 2 belah ketupat: "))
26     hasil = rumus.hitung_luas(menu_pilihan, diagonal1, diagonal2)
```

2. Ketika ia mengklik tombol commit, akan muncul pesan yang meminta ia memilih perubahan yang ingin digunakan, apakah "Perubahan Saat ini" (Current Change) untuk menggunakan perubahan yang sudah dilakukan atau "Perubahan Akan Datang" (Incoming Change) untuk menerima perubahan yang sedang dalam proses. Dalam kasus ini, ia akan menggunakan **Incoming Change**



```
main.py 3.1 x
main.py > ...
1 import rumus
2 import sys
3
4 print("Menu:")
5 print("1. Segitiga")
6 print("2. Persegi")
7 print("3. Lingkaran")
8 print("4. Belah Ketupat")
9 ===== HEAD (Current Change) =====
10 print("5. Exit")
11 =====
12 print("5. Keluar")
13 =====>>>> 545ff9c6829a5b5e3c7fa29d0b9569f10bbfa075 (Incoming Change)
14
15 menu_pilihan = int(input("Pilih menu : "))
16
17 if menu_pilihan == 1:
18     alas = float(input("Masukkan panjang alas segitiga: "))
19     tinggi = float(input("Masukkan tinggi segitiga: "))
20     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
21 elif menu_pilihan == 2:
22     sisi = float(input("Masukkan panjang sisi persegi: "))
23     hasil = rumus.hitung_luas(menu_pilihan, sisi)
24 elif menu_pilihan == 3:
25     jari = float(input("Masukkan jari-jari lingkaran: "))
26     hasil = rumus.hitung_luas(menu_pilihan, jari)

Source: Gi... Open Git Log Show Command Output Show Changes
```



```
main.py 3.1 x
main.py > ...
1 import rumus
2 import sys
3
4 print("Menu:")
5 print("1. Segitiga")
6 print("2. Persegi")
7 print("3. Lingkaran")
8 print("4. Belah Ketupat")
9 ===== HEAD (Current Change) =====
10 print("5. Exit")
11 =====
12 print("5. Keluar")
13 =====>>>> 545ff9c6829a5b5e3c7fa29d0b9569f10bbfa075 (Incoming Change)
14
15 menu_pilihan = int(input("Pilih menu : "))
16
17 if menu_pilihan == 1:
18     alas = float(input("Masukkan panjang alas segitiga: "))
19     tinggi = float(input("Masukkan tinggi segitiga: "))
20     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
21 elif menu_pilihan == 2:
22     sisi = float(input("Masukkan panjang sisi persegi: "))
23     hasil = rumus.hitung_luas(menu_pilihan, sisi)
24 elif menu_pilihan == 3:
25     jari = float(input("Masukkan jari-jari lingkaran: "))
26     hasil = rumus.hitung_luas(menu_pilihan, jari)

Resolve in Merge Editor
```

3. Setelah itu, kita lihat ternyata di tab file nya masih terdapat tanda seru merah. Ini mengisyaratkan masih ada error konflik lain, untuk melihatnya kita klik tombol “Resolve in Merge Editor”

```

1 import rumus
2 import sys
3
4 print("Menu:")
5 print("1. Segitiga")
6 print("2. Persegi")
7 print("3. Lingkaran")
8 print("4. Belah Ketupat")
9 print("5. Keluar")
10
11 menu_pilihan = int(input("Pilih menu : "))
12
13 if menu_pilihan == 1:
14     alas = float(input("Masukkan panjang alas segitiga: "))
15     tinggi = float(input("Masukkan tinggi segitiga: "))
16     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
17 elif menu_pilihan == 2:
18     sisi = float(input("Masukkan panjang sisi persegi: "))
19     hasil = rumus.hitung_luas(menu_pilihan, sisi)
20 elif menu_pilihan == 3:
21     jari = float(input("Masukkan jari-jari lingkaran: "))
22     hasil = rumus.hitung_luas(menu_pilihan, jari)
23 elif menu_pilihan == 4:
24     diagonal1 = float(input("Masukkan diagonal 1 belah ketupat: ")) Resolve in Merge Editor
25     diagonal2 = float(input("Masukkan diagonal 2 belah ketupat: "))
26     hasil = rumus.hitung_luas(menu_pilihan, diagonal1, diagonal2)

```

4. Seperti sebelumnya, disini juga disuruh memilih perubahan mana yang akan diterapkan. Setelah memilih, dan selesai klik tombol “Complete Merge”

Incoming: 545ff9c · origin/main

Result: main.py

```

1 import rumus
2 import sys
3
4 print("Menu:")
5 print("1. Segitiga")
6 print("2. Persegi")
7 print("3. Lingkaran")
8 print("4. Belah Ketupat")
9
10 menu_pilihan = int(input("Pilih menu : "))
11
12 if menu_pilihan == 1:
13     alas = float(input("Masukkan panjang alas segitiga: "))
14     tinggi = float(input("Masukkan tinggi segitiga: "))
15     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
16 elif menu_pilihan == 2:
17     sisi = float(input("Masukkan panjang sisi persegi: "))
18     hasil = rumus.hitung_luas(menu_pilihan, sisi)
19 elif menu_pilihan == 3:
20     jari = float(input("Masukkan jari-jari lingkaran: "))
21     hasil = rumus.hitung_luas(menu_pilihan, jari)
22 elif menu_pilihan == 4:
23     diagonal1 = float(input("Masukkan diagonal 1 belah ketupat: "))
24     diagonal2 = float(input("Masukkan diagonal 2 belah ketupat: "))
25     hasil = rumus.hitung_luas(menu_pilihan, diagonal1, diagonal2)

```

5. Maka konflik telah teratasi, dan bisa di Commit

SOURCE CONTROL

Staged Changes: main.py

Changes: main.py

File: main.py

```

1 import rumus
2 import sys
3
4 print("Menu:")
5 print("1. Segitiga")
6 print("2. Persegi")
7 print("3. Lingkaran")
8 print("4. Belah Ketupat")
9 print("5. Keluar")
10
11 menu_pilihan = int(input("Pilih menu : "))
12
13 if menu_pilihan == 1:
14     alas = float(input("Masukkan panjang alas segitiga: "))
15     tinggi = float(input("Masukkan tinggi segitiga: "))
16     hasil = rumus.hitung_luas(menu_pilihan, alas, tinggi)
17 elif menu_pilihan == 2:
18     sisi = float(input("Masukkan panjang sisi persegi: "))
19     hasil = rumus.hitung_luas(menu_pilihan, sisi)
20 elif menu_pilihan == 3:
21     jari = float(input("Masukkan jari-jari lingkaran: "))
22     hasil = rumus.hitung_luas(menu_pilihan, jari)
23 elif menu_pilihan == 4:
24     diagonal1 = float(input("Masukkan diagonal 1 belah ketupat: "))
25     diagonal2 = float(input("Masukkan diagonal 2 belah ketupat: "))
26     hasil = rumus.hitung_luas(menu_pilihan, diagonal1, diagonal2)

```

Penutup

Sungguh luar biasa telah memiliki Kamu sebagai teman perjalanan saya melalui setiap halaman dalam buku "Pengantar Dasar GIT & Github". Dengan tulus, terima kasih atas dedikasi Kamu dalam mengeksplorasi dunia kontrol versi dengan Git dan platform kolaborasi Github bersama kami.

Buku ini tidak hanya berfungsi sebagai panduan, tetapi juga sebagai mitra yang setia dalam membangun pemahaman yang kokoh tentang bagaimana mengelola kode sumber secara efektif. Melalui setiap bab, kami berupaya memberikan fondasi yang kuat bagi pembaca, memungkinkan mereka mengeksplorasi dan memahami intricacies dari Git dan Github.

Semoga buku ini telah membantu membuka pintu pengetahuan dan keterampilan baru dalam dunia pengembangan perangkat lunak. Dengan pemahaman yang diperoleh, diharapkan pembaca dapat dengan percaya diri mengambil langkah-langkah dalam proyek pengembangan perangkat lunak mereka sendiri, dengan dukungan penuh dari Git dan Github.

Tentu saja, perjalanan ini tak berakhir di sini. Saya berharap bahwa buku ini akan menjadi sumber daya yang andal yang akan selalu Kamu kembalikan saat menghadapi tantangan atau ketika ingin memperdalam pemahaman Kamu. Jangan ragu untuk terus menjelajahi dunia GIT & Github, dan semoga setiap langkah Kamu di dunia pengembangan perangkat lunak membawa kesuksesan dan pencapaian yang memuaskan.

Ketika memasuki dunia pengembangan perangkat lunak, ingatlah beberapa pesan penting berikut:

1. **Konsistensi adalah Kunci:** Gunakan Git dan Github secara konsisten dalam setiap proyek Kamu. Ini membantu Kamu membangun kebiasaan yang baik dan menjadikan pengelolaan kode lebih terorganisir.
2. **Pahami Konsep Dasar:** Memahami konsep dasar Git, seperti commit, branch, dan merge, akan membantumu mengatasi tantangan dengan lebih mudah. Berinvestasi waktu untuk memahaminya dengan baik.
3. **Berpartisipasi dalam Komunitas:** Komunitas pengembang sangat berharga. Bergabunglah dengan forum dan diskusi online, dan jangan ragu untuk bertanya atau berbagi pengalaman. Ini membantu Kamu tumbuh sebagai pengembang.
4. **Terus Belajar:** Dunia pengembangan terus berkembang. Selalu terbuka untuk belajar hal-hal baru, teknologi terbaru, dan praktik terbaik dalam pengelolaan kode.

Terima kasih telah meluangkan waktu untuk belajar bersama Saya. Semoga Buku ini telah memberikan nilai tambah dalam perjalanan pengembangan perangkat lunak pembaca.

Daftar Pustaka

git - panduan ringkas - gak pake ribet!,

<https://rogerdudler.github.io/git-guide/index.id.html>. Accessed 9 January 2024.

“Apa itu GitHub? Berikut Cara Menggunakannya.” *Dicoding*, 21 April 2021,

<https://www.dicoding.com/blog/apa-itu-github/>. Accessed 9 January 2024.

“Belajar Menggunakan Git dan Github Untuk Pemula.” *Jagoweb*,

<https://jagoweb.com/belajar-menggunakan-git-dan-github-untuk-pemula>. Accessed 9 January 2024.

“How to Resolve Merge Conflicts in Git Tutorial.” *DataCamp*,

<https://www.datacamp.com/tutorial/how-to-resolve-merge-conflicts-in-git-tutorial>. Accessed 14 January 2024.

“Menggunakan Git pada Visual Studio Code (Lebih praktis!).” *Petani Kode*, 16

November 2017, <https://www.petanikode.com/git-vscode/>. Accessed 9 January 2024.

“Modul Dasar GIT dan Github Untuk Mata Kuliah Praktikum.” *Repository*

Universitas BSI,

<https://repository.bsi.ac.id/repo/files/325677/download/Modul-GIT-2021.pdf>. Accessed 9 January 2024.

Biografi Penulis



Andika Tulus Pangestu atau biasa dipanggil “*dika*” atau sering juga dipanggil “*tulus*” lahir pada 01 Mei 2003 di Tegal, Jawa Tengah, Indonesia. Ia dikenal sebagai seorang penulis dan pengembang perangkat lunak. Andika mulai menunjukkan minatnya pada dunia literasi sejak SMP, dengan membaca berbagai buku dan menulis cerita pendek di waktu luangnya melalui media blog. Andika menyelesaikan pendidikannya di SMA dengan mengambil jurusan Ilmu Pengetahuan Sosial (IPS). Setelah lulus SMA, Andika melanjutkan pendidikannya di Universitas Bina Sarana Informatika, Kampus Kota Tegal, dengan mengambil jurusan Sistem Informasi.

Di sana, ia semakin mendalami pengetahuan dan keterampilannya dalam pengelolaan informasi dan teknologi. Keahliannya dalam pemrograman dan pengelolaan proyek membawanya terlibat dalam berbagai proyek dalam kampus maupun luar kampus dengan selalu ikut serta dalam pengabdian masyarakat, penelitian dan kompetisi tingkat nasional maupun daerah. Selama masa perkuliahan, Andika tidak hanya memprioritaskan mengejar akademis, tetapi juga memanfaatkan segi waktu luangnya untuk mengembangkan hobi. Hobi utamanya adalah ngoding dan sharing, di mana ia menikmati tantangan dalam memecahkan masalah, menciptakan solusi dengan bahasa pemrograman hingga membagikan apa yang telah ia lakukan di media sosial pribadinya.

Selain Itu, ia juga memiliki minat mendalam dalam menyampaikan pengetahuan dan pengalaman kepada orang lain dengan mulai menulis artikel dan panduan teknis tentang topik-topik terkait pemrograman dan pengembangan perangkat

lunak. Antusiasme dalam berbagi pengetahuan ini kemudian mendorongnya untuk menulis buku dan artikel informatif.

Buku versi Digital "Pengantar Dasar GIT & Github," merupakan buku pertama hasil dari pengalaman praktisnya dalam menggunakan alat-alat ini dalam pengembangan perangkat lunak. Andika berkomitmen untuk membuat materi teknis menjadi lebih dapat diakses, terutama bagi pembaca pemula yang ingin memahami konsep kontrol versi dan kolaborasi kode dengan mudah. Andika percaya bahwa pengetahuan harus didistribusikan secara inklusif, dan setiap orang memiliki potensi untuk mengetahui ilmu yang ada di dunia teknologi. Biografi ini mencerminkan perjalanan hidupnya yang penuh semangat dan antusiasme dalam menggali potensi diri. Andika Tulus Pangestu percaya bahwa keberhasilan bukanlah akhir dari perjalanan, tetapi merupakan awal dari petualangan yang lebih besar. Dalam setiap kode yang ditulisnya, ia temukan makna, dan dalam setiap langkahnya, ia lihat peluang untuk belajar dan tumbuh.

'Keberanian untuk mencoba adalah kunci menuju kesuksesan, semakin banyak kegagalan yang datang maka semakin dekat pula kita dengan kesuksesan'.

- *Andika Tulus Pangestu*



PENGANTAR

GIT & GITHUB DASAR

Buku ini saya tulis dengan tujuan membantu seseorang dalam bidang pengembangan perangkat lunak untuk memahami dasar-dasar Git dan Github, serta memberikan panduan yang jelas dan mudah dipahami. Git dan Github adalah alat yang sangat berguna bagi para pengembang perangkat lunak untuk mengelola kode mereka dengan lebih efisien dan kolaboratif.