



Program Studi Sarjana

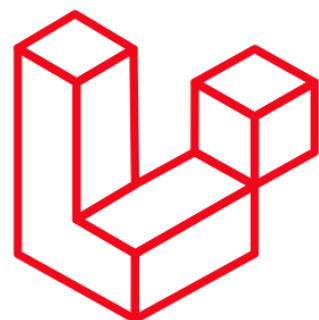
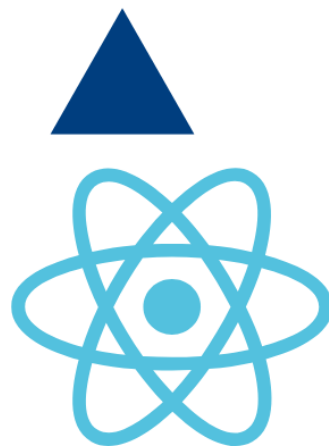
Informatika

UNIVERSITAS ATMA JAYA YOGYAKARTA

MODUL PEMROGRAMAN APLIKASI WEB T.A. 25/26

MODUL 01

GITHUB 1



REQUIREMENT

1. Web Browser (Google Chrome/Chromium (Ms Edge, Opera) atau Firefox)
2. Sistem Operasi: Windows (versi 7 atau lebih baru), macOS, atau Linux.

PENJELASAN

TUJUAN

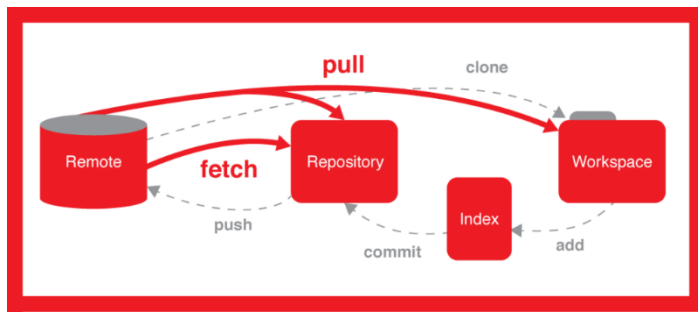
Setelah menyelesaikan modul ini, praktikan diharapkan mampu :

1. Mengetahui cara instalasi Version Control System (VCS) serta Cara menggunakannya
2. Menggabungkan project yang dikerjakan secara berkelompok

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. 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 menemukan hal yang serupa dengan Version Control System.

Selain sebagai system yang mengelola riwayat perubahan, system ini juga berguna untuk mempermudah pengembangan sebuah program secara kolaboratif atau berkelompok. Jadi dengan system VCS kita bisa mengerjakan satu berkas secara bersamaan dengan beberapa orang, tanpa menunggu satu persatu selesai dan bergantian mengerjakan.

Dan yang terakhir, Version Control System juga memungkinkan kita untuk kembali ke perubahan sebelumnya. Pada intinya, version control system bisa melakukan tugas seperti melihat riwayat perubahan, mengelola setiap perubahan (commit, push, checkout, pull), menuntaskan bagian-bagian berkas secara langsung tanpa menunggu pengerjaan yang lain selesai, dan memungkinkan kita kembali ke perubahan sebelumnya. Git termasuk dalam jenis Version Control System, dan Github adalah layanan Version Control System berbasis git secara online.



Gambar 1

A. Git

Git adalah sebuah piranti perangkat lunak berbasis Version Control System, dengan git kita mampu mengelola sebuah proyek atau berkas secara terdistribusi, karena perubahan yang tercatat oleh git disimpan didalam database git dan database git tersebut tidak hanya berada pada satu tempat, namun terdapat di berbagai tempat. Artinya bahwa semua orang yang berkontribusi pada proyek tersebut bisa menyimpan database perubahannya, sehingga akan memudahkan dalam mengelola proyek baik online maupun offline.

Dengan git kita bisa melakukan penggabungan beberapa berkas atau bagian pada berkas kedalam satu proyek atau berkas final, hal ini sangat mempermudah developer ketika melakukan pengembangan aplikasi secara berkolaborasi.

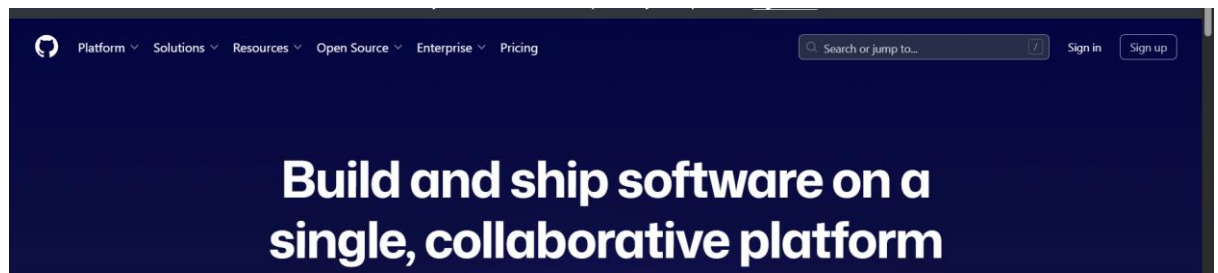
B. Github

Jika dengan git kita hanya mampu mengelola setiap perubahan secara lokal, maka dengan github kita bisa melakukannya secara online. Github adalah layanan git berbasis cloud, selain itu dengan github kita dapat dengan mudah bekerja dengan git tanpa melakukannya dengan baris perintah (CLI), karena github merupakan layanan git berbasis GUI.

C. Membuat Akun Github

1. Masuk pada situs resmi github <https://github.com/>

2. Pilih “Sign Up” Pada pojok kanan atas,



Gambar 2

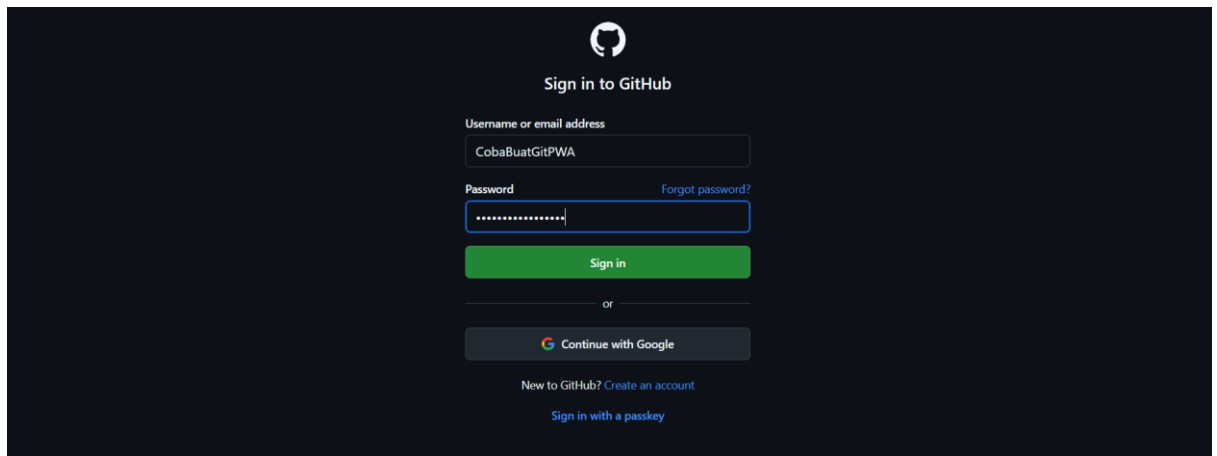
3. Lengkapi form pendaftaran yang dibutuhkan

Gambar 3

4. Masukkan kode verifikasi yang telah terkirim pada email

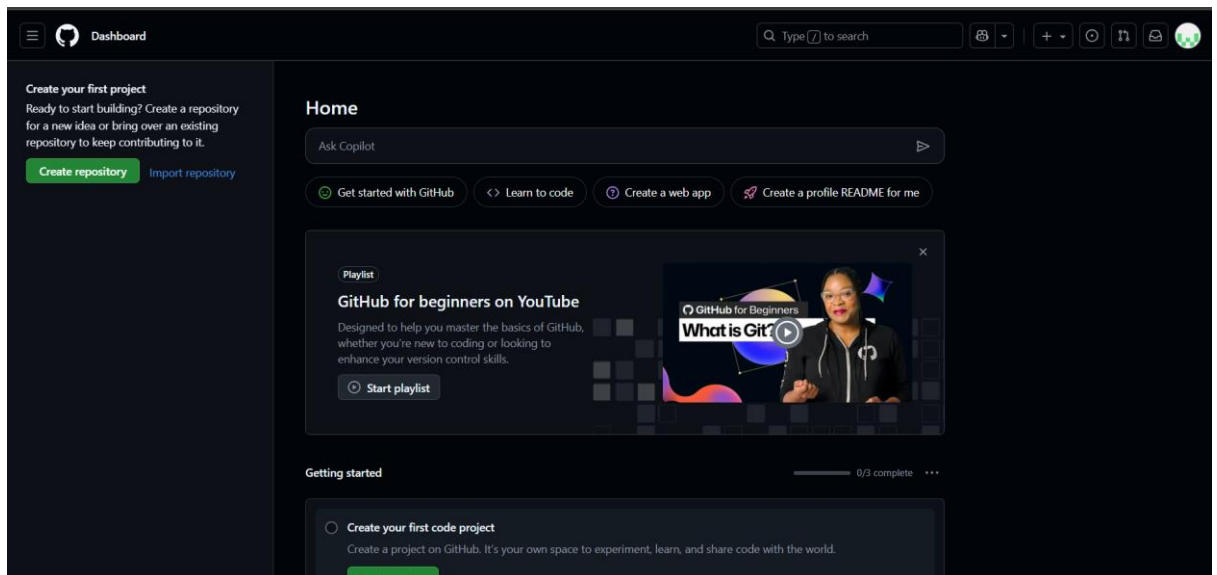
Gambar 4

5. Ketika Github sudah terbuat, login menggunakan username/email dan password yang sudah dibuat.



Gambar 5

6. Jika berhasil maka akan diarahkan ke dashboard github



Gambar 6

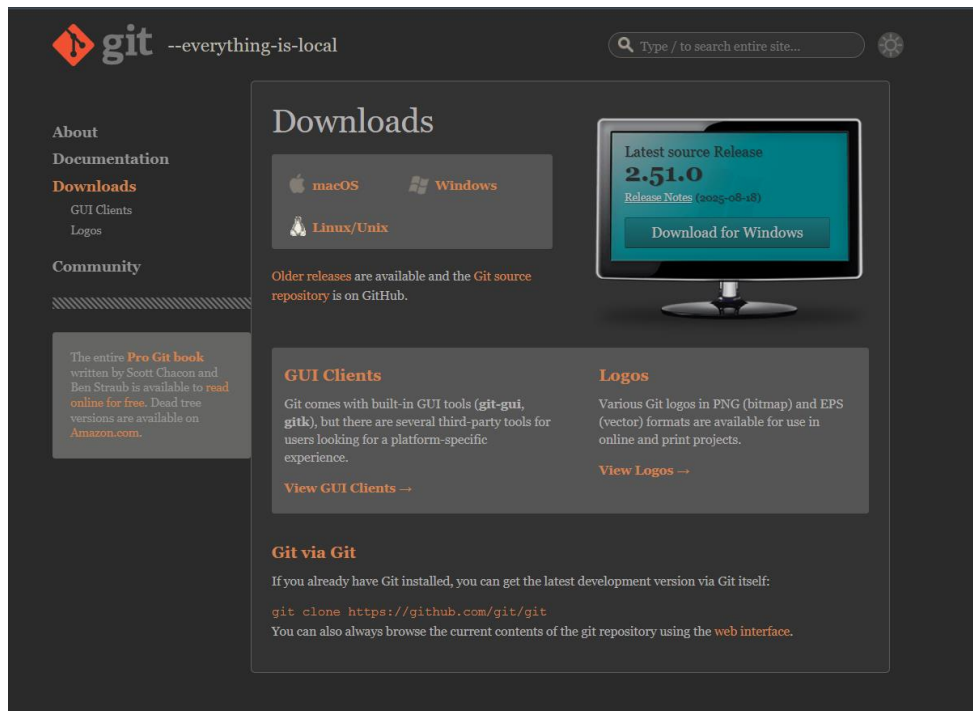
D. Instalasi Git

Dalam instalasi Git, diperlukan beberapa persyaratan system yang harus dipehuni, seperti

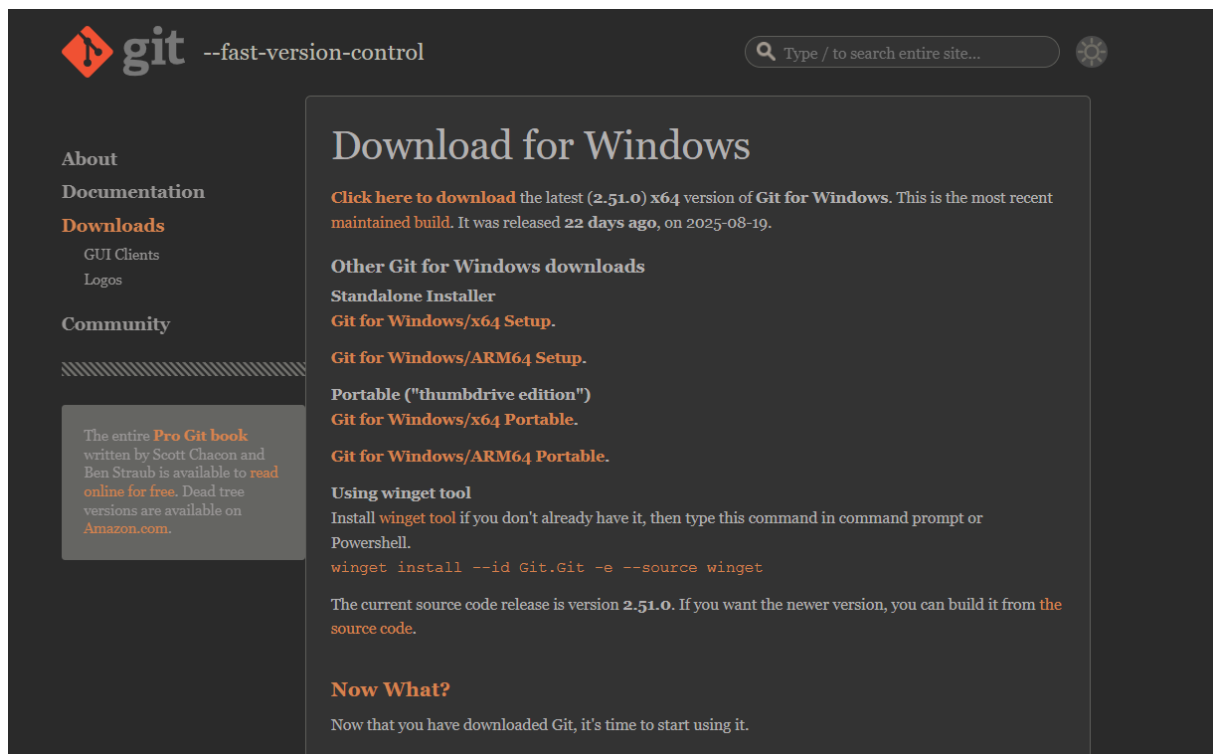
1. Perangkat dengan OS Minimal Windows 7 atau setara
2. Web Browser (Firefox, Chrome, Edge, dll)

Langkah Pemasangan

1. Unduh Git pada <https://git-scm.com/> (Atau sesuai instalasi modul sebelumnya).
2. Pilih OS sesuai yang digunakan, Karena saya menggunakan windows, maka pilih “Windows”



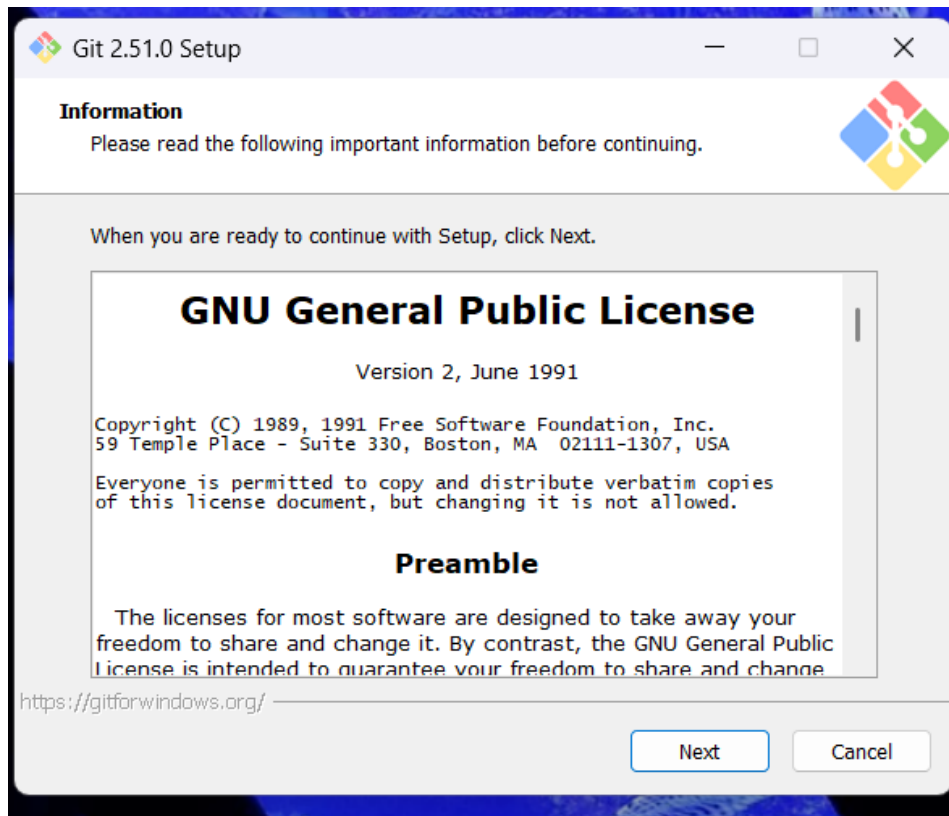
Gambar 7



Gambar 8

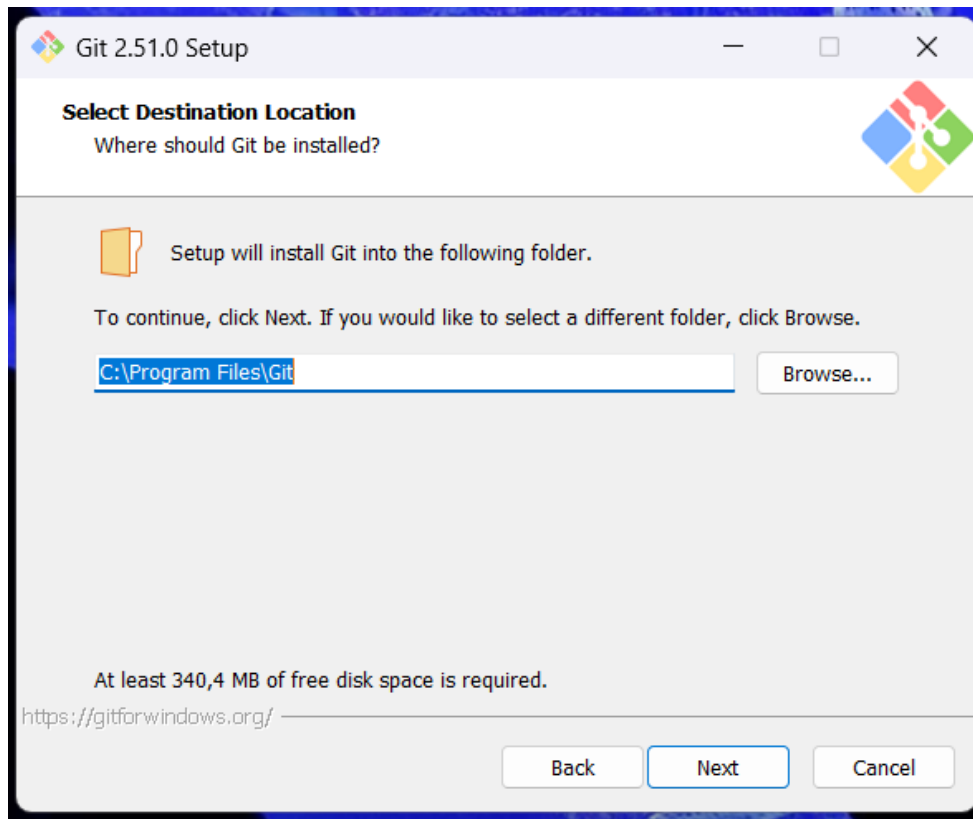
3. Lalu Pilih sesuai dengan arsitektur perangkat yang digunakan. Jika menggunakan 64 bit, pilih 64, jika menggunakan 32, pilih 32. Lalu tunggu hingga download selesai.

4. Setelah selesai mengunduh, klik 2x pada file Git yang telah terunduh.
5. Tinjau Lisensi GNU, dan jika setuju klik Next



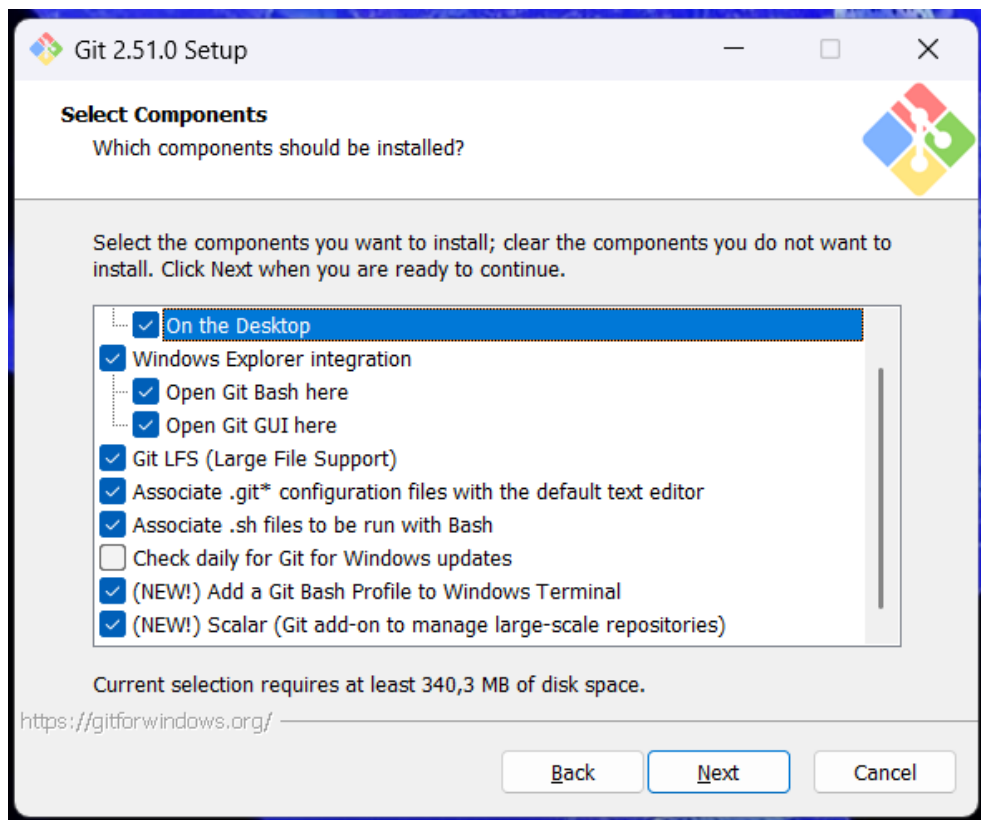
Gambar 9

6. Selanjutnya menentukan lokasi instalasi. Usahakan untuk tempat instalasinya tidak dirubah, Lalu Next



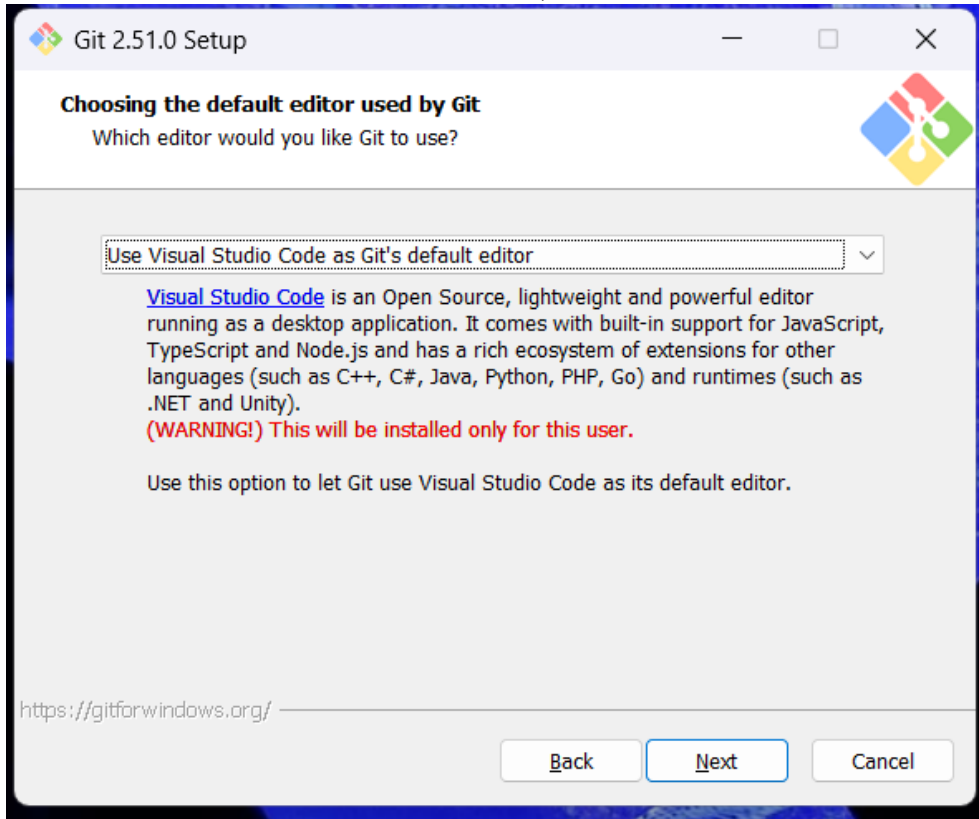
Gambar 10

7. Lalu pemilihan Komponen, Atur saja sesuai dengan gambar dibawah, Lalu Next



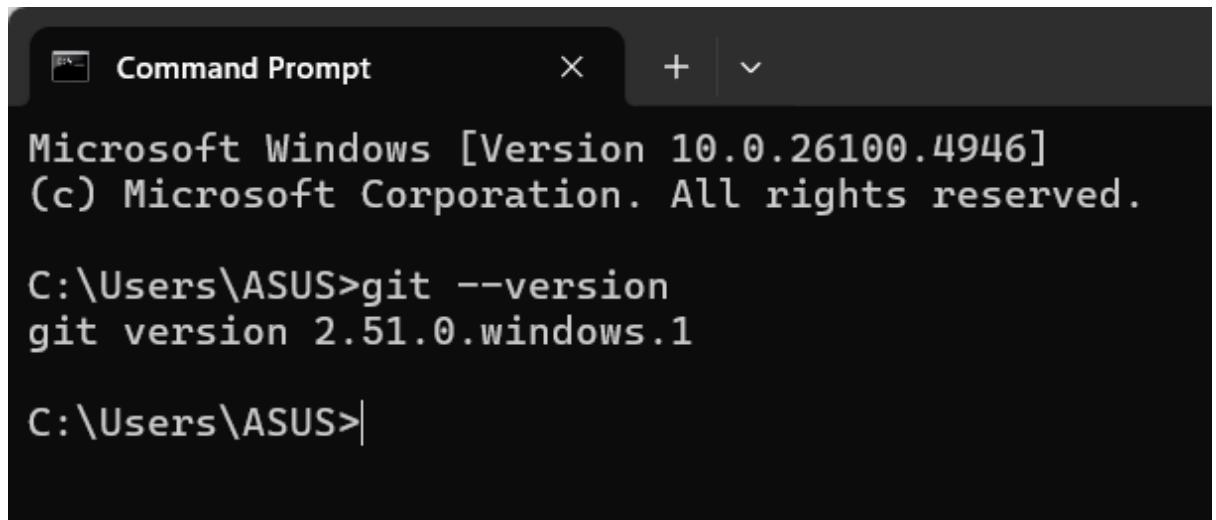
Gambar 11

8. Pilih editor teks sebagai default untuk menggunakan git (Karena kita menggunakan VSCode, maka kita memilih VSCode)



Gambar 12

9. Untuk selanjutnya, Next Saja karena kita hanya menggunakan VSCode untuk upload repository. Tidak usah diubah settingannya dan biarkan saja default.
10. Untuk finally, klik install saja jika sudah sampai di final setting, tunggu hingga selesai. Jika sudah selesai, maka git sudah berhasil di install.
11. Jika sudah selesai melakukan instalasi, maka bisa ke *Command Prompt* atau sejenisnya. Ketik `git -version` untuk mengecek apakah instalasi sudah berhasil atau tidak. Jika berhasil, maka akan muncul seperti ini



```
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>git --version
git version 2.51.0.windows.1

C:\Users\ASUS>
```

Gambar 13

12. Karena di PC/Laptop, github tidak mengenali device anda, maka anda harus melakukan setting agar device anda gunakan untuk *version control system*. Oleh karena itu, anda dapat melakukan cara berikut pada *Command Prompt* atau terminal:
 - a. `git config --global user.name <nama username Github anda>`
 - b. `git config --global user.email <email yang didaftarkan pada github>`
 - c. `git config --list` (untuk melihat apakah sudah terdaftar atau tidak username dan email anda)

Ket: Hapus tanda < >

13. Hasilnya akan seperti ini

```
Command Prompt
(c) Microsoft Corporation. All rights reserved.
C:\Users\ASUS>git config --global user.name [REDACTED]
C:\Users\ASUS>git config --global user.email [REDACTED]
C:\Users\ASUS>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=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=true
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
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
[REDACTED]
core.editor="D:\DASPRO\Microsoft VS Code\bin\code" --wait
C:\Users\ASUS>
```

Gambar 14

E. Command List GitHub dan Alur Kerja

Berikut ini beberapa perintah dan atau istilah yang umum digunakan dalam GitHub.

Perintah Umum

Perintah	Artinya	Contoh Penggunaan
git init	Mulai proyek baru - Buat folder Git kosong di komputer kamu	git init
git clone	Salin proyek orang - Download proyek dari GitHub ke komputer	git clone https://github.com/nama-user/nama-project.git
git status	Cek kondisi - Lihat file apa saja yang berubah	git status
git add	Siapkan file - Pilih file yang mau disimpan	git add . (semua file) git add index.html (file tertentu)
git commit	Simpan perubahan - Simpan dengan pesan	git commit -m "Menambah fitur login"
git push	Upload ke GitHub - Kirim perubahan ke git origin (internet)	git push origin main
git pull	Download update - Ambil perubahan terbaru dari git origin (internet)	git pull origin main

Mengelola Branch

Perintah	Artinya	Contoh
git branch	Lihat cabang - Daftar semua cabang yang ada	git branch
git branch nama	Buat cabang baru - Seperti membuat folder baru	git branch fitur-baru
git checkout	Pindah cabang - Pindah ke cabang lain	git checkout main
git merge	Gabungkan cabang - Satukan pekerjaan dari cabang lain	git merge fitur-baru

Alur Kerja GitHub

1. git clone - Download proyek dari GitHub (jika sudah ada proyek di GitHub)
2. git checkout -b fitur-baru - Buat cabang untuk fitur baru

3. Bekerja - Edit file-file yang diperlukan
4. `git add .` - Pilih semua file yang diubah
5. `git commit -m "Pesan"` - Simpan perubahan
6. `git push origin fitur-baru` - Upload ke GitHub
7. Buat Pull Request di website GitHub
8. `git pull origin main` - Update proyek local

Tips Penting

1. Selalu `git pull` sebelum mulai bekerja
2. Selalu buat cabang baru untuk fitur baru
3. Commit sesering mungkin dengan pesan yang jelas
4. Push ke GitHub secara berkala

Istilah Penting

1. Repository = Proyek / folder proyek
2. Commit = Simpanan / checkpoint
3. Branch = Cabang / versi berbeda
4. Push = Upload ke internet
5. Pull = Download dari internet

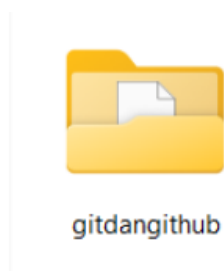
F. Bekerja dengan Git dan Github

Persiapan:

1. Device yang sudah terinstall Git
2. Telah memiliki Akun Github
3. Memiliki Koneksi Internet
4. VsCode
5. Program atau code yang akan di push pada Github

Langkah-Langkah:

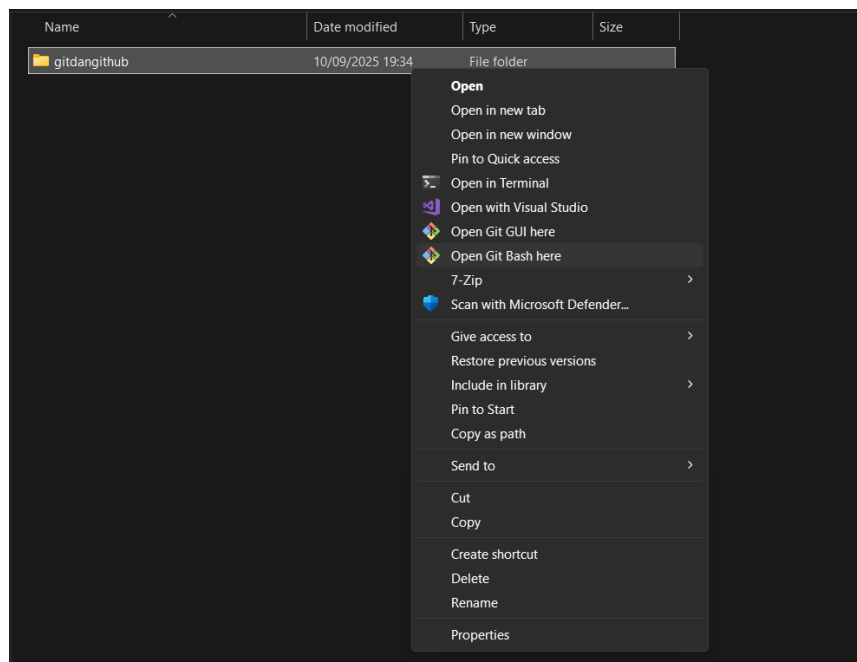
1. Siapkan folder yang akan dipush ke github



Gambar 15

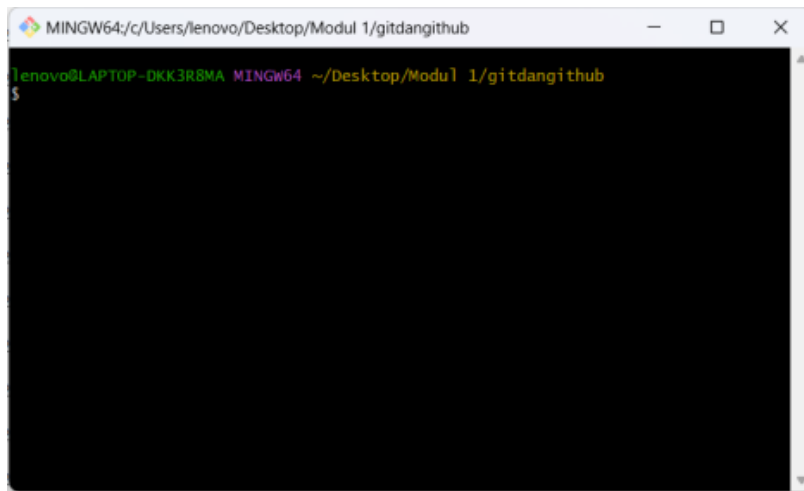
Saya akan melakukan push dan commit pada github untuk file tersebut.

2. Masuk pada file tersebut, lalu klik kanan didalam file tersebut dan pilih “Git Bash Here”



Gambar 16

Maka akan tampil seperti ini



Gambar 17

3. Lalu ketik pada git bash

-git init

-git status

```
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub
$ git init
Initialized empty Git repository in C:/Users/lenovo/Desktop/Modul 1/gitdangithub
/.git/

lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        .metadata
        README.md
        analysis_options.yaml
        android/
        ios/
        lib/
        linux/
        macos/
        pubspec.lock
        pubspec.yaml
        test/
        web/
        windows/

nothing added to commit but untracked files present (use "git add" to track)
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (master)
$
```

Gambar 18

4. Kemudian buat sebuah repository baru pada akun github

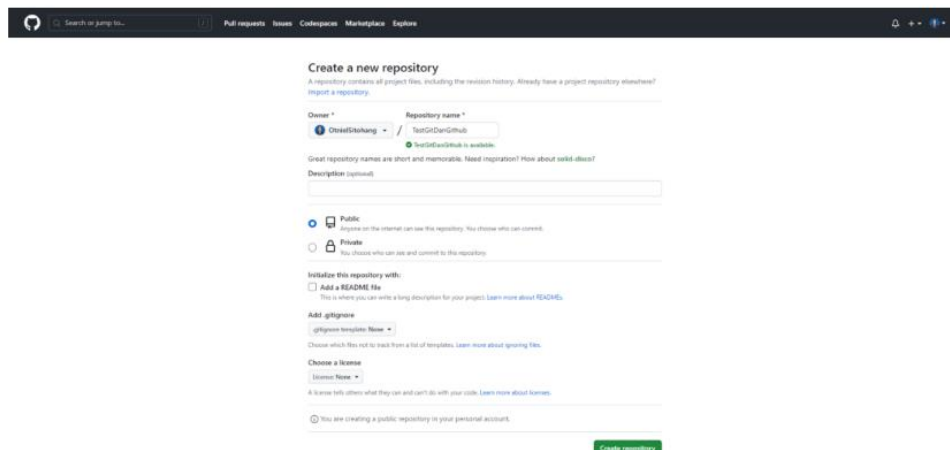
Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Create repository

Import repository

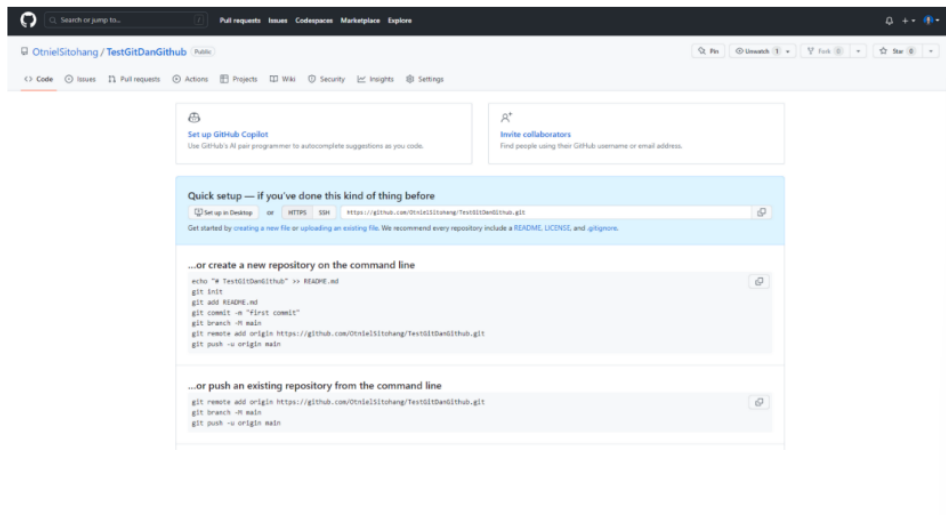
Gambar 19

The screenshot shows the GitHub 'Create a new repository' interface. At the top, there's a navigation bar with links for Pull requests, Issues, Codepaces, Marketplace, and Explore. The main heading is 'Create a new repository' with a subtext explaining that a repository contains project files and revision history. Below this, there's a form with several sections: 'Owner' (set to 'DhealShahang'), 'Repository name' (with a dropdown showing 'TestGitDarGithub' and a note that 'TestGitDarGithub' is available), a 'Description' field, and radio buttons for 'Public' (selected) and 'Private'. There's also an 'Initialize this repository with' section with a checkbox for 'Add a README file'. At the bottom, there's a 'Choose a license' dropdown and a note about creating a public repository on a personal account. A green 'Create repository' button is at the bottom right.

Gambar 20

Lalu Masukkan nama Repository yang diinginkan, untuk jenis dari repository nya tergantung kebutuhan, boleh di public atau di private, lalu klik “Create repository”.

5. Lalu akan muncul seperti ini

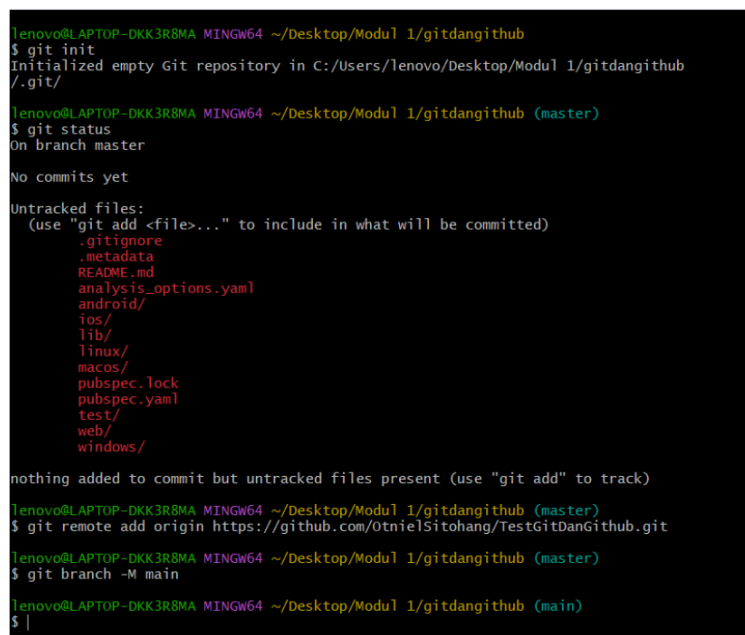


Gambar 21

6. Lalu Kembali pada git base, setelah melakukan init dan status serta membuat repository baru, langkah selanjutnya adalah memasukkan code baris 1 dan 2 yang muncul pada github secara berurutan pada git base.



Gambar 22



Gambar 23

7. Kemudian ketikkan

- git add
- git commit -m “Kata commit yang ingin Anda masukkan”
- git push -u origin main

```
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (main)
$ git add .

lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (main)
$ git commit -m "first commit"
[main (root-commit) 236a4ad] first commit
128 files changed, 4873 insertions(+)
create mode 100644 .gitignore
create mode 100644 .metadata
create mode 100644 README.md
create mode 100644 analysis_options.yaml
create mode 100644 android/.gitignore
create mode 100644 android/app/build.gradle
create mode 100644 android/app/src/debug/AndroidManifest.xml
create mode 100644 android/app/src/main/AndroidManifest.xml
create mode 100644 android/app/src/main/kotlin/com/example/gitdangithub/MainActivity.kt
create mode 100644 android/app/src/main/res/drawable-v21/launch_background.xml
create mode 100644 android/app/src/main/res/drawable/launch_background.xml
create mode 100644 android/app/src/main/res/mipmap-hdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/mipmap-mdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/mipmap-xhdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/mipmap-xxhdpi/ic_launcher.png
create mode 100644 android/app/src/main/res/values-night/styles.xml
create mode 100644 android/app/src/main/res/values/styles.xml
create mode 100644 android/app/src/profile/AndroidManifest.xml
create mode 100644 android/build.gradle

lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (main)
$ git push -u origin main
Enumerating objects: 177, done.
Counting objects: 100% (177/177), done.
Delta compression using up to 16 threads
Compressing objects: 100% (146/146), done.
Writing objects: 100% (177/177), 266.92 KiB | 8.61 MiB/s, done.
Total 177 (delta 20), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (20/20), done.
To https://github.com/OtnielSitoHang/TestGitDanGithub.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

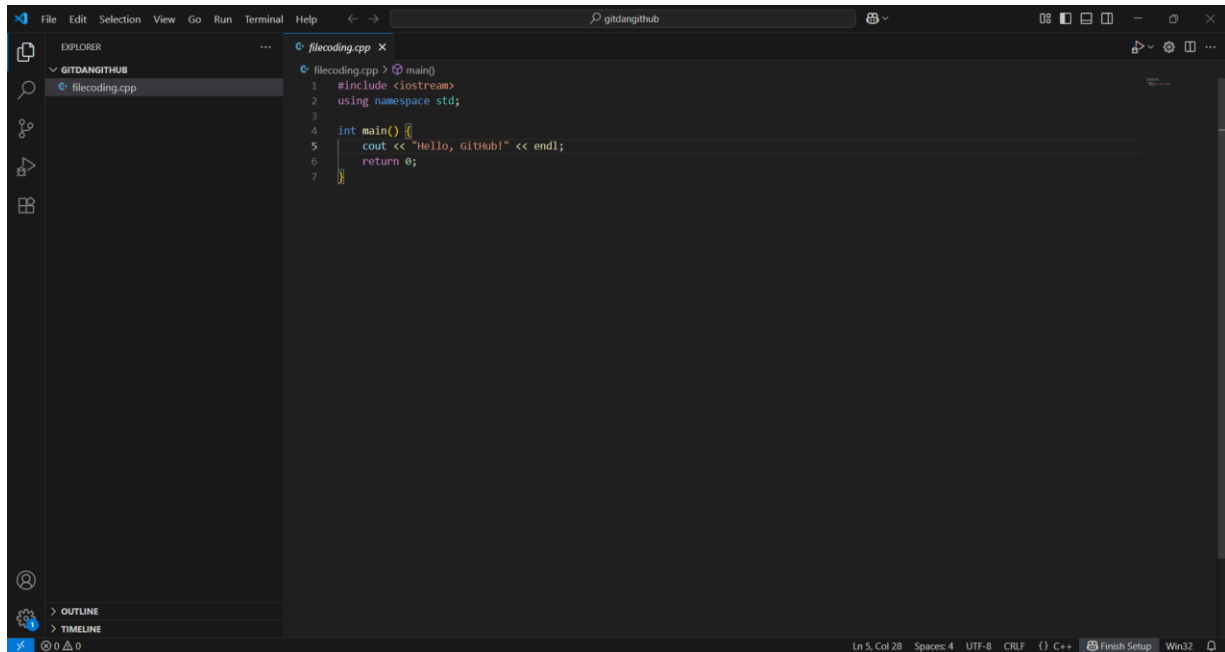
lenovo@LAPTOP-DKK3R8MA MINGW64 ~/Desktop/Modul 1/gitdangithub (main)
$ |
```

Gambar 24

Code sudah berhasil di push pada repository yang sudah di buat.

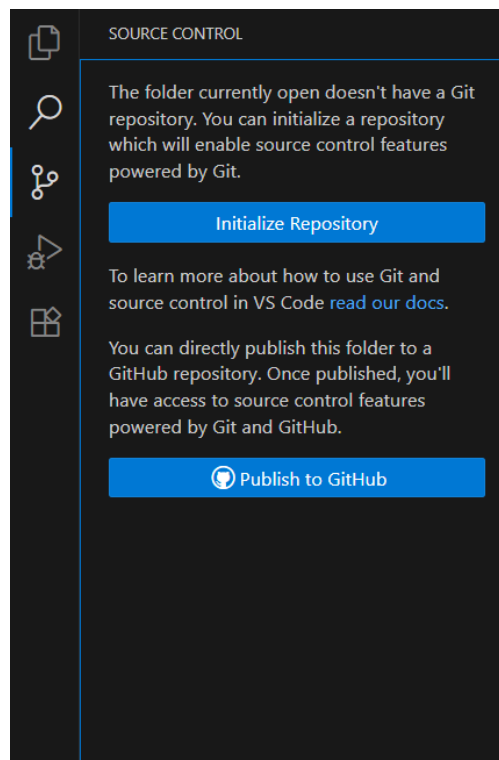
G. Cara lain melakukan push ke github menggunakan VSCode

1. Buka Folder yang ingin di push pada Vscode



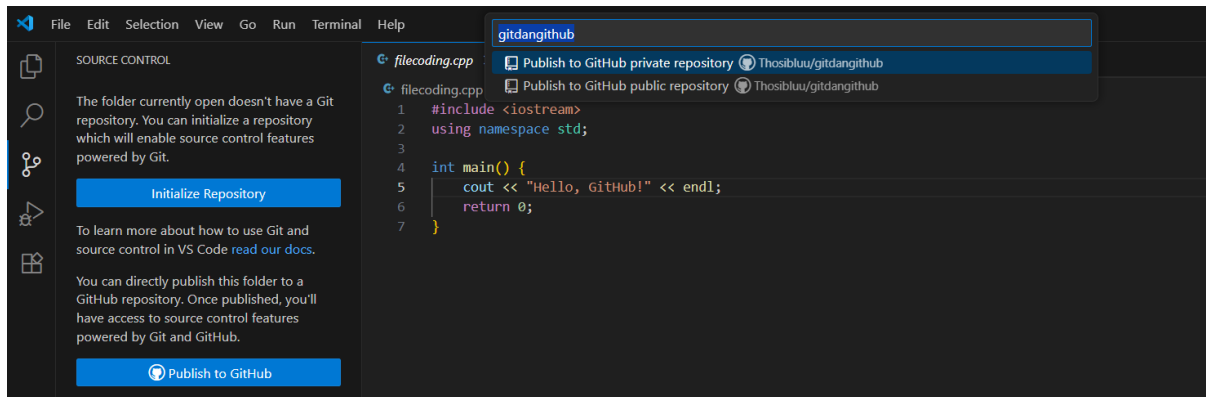
Gambar 25

2. Pilih bagian putih lalu pilih bagian “push to Github”. Jika baru pertama kali membuka folder (Tidak clone atau belum pernah di push di github tampilannya akan seperti ini).



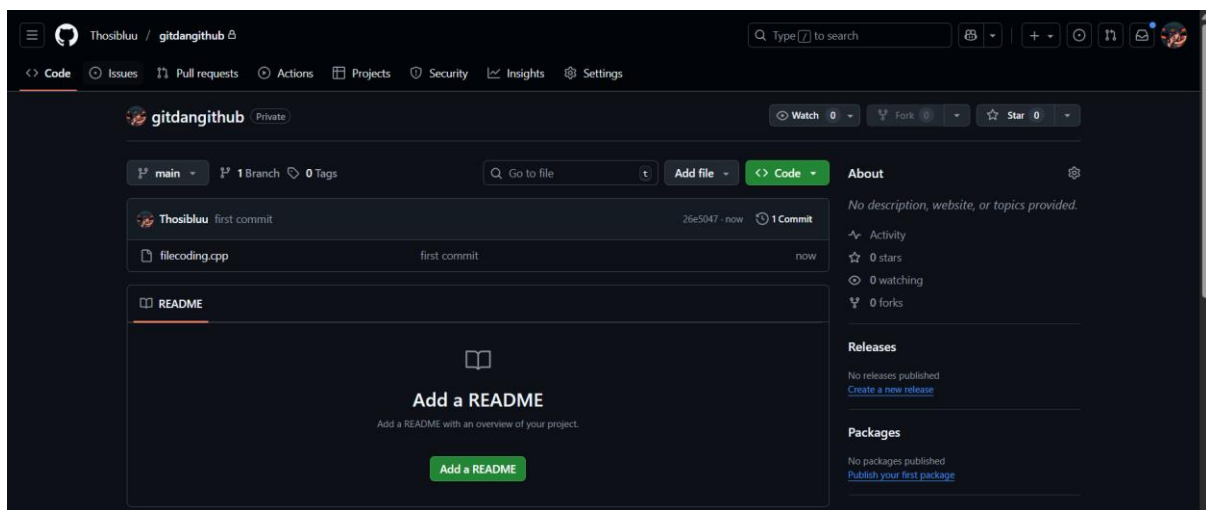
Gambar 26

3. Pada bagian atas VSCode akan muncul tampilan seperti ini. Bagian yang berwarna biru adalah nama dari repository yang akan dibuat nantinya. Anda bisa memilih jenis repository yang akan dibuat (Berupa Public atau Private) sesuai dengan kebutuhan. Lalu tunggu hingga proses selesai.



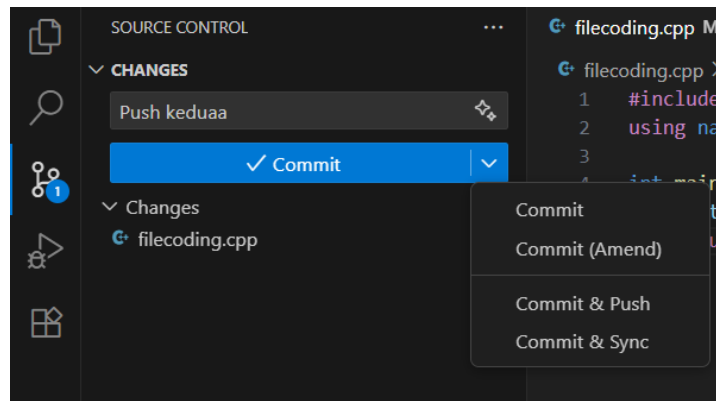
Gambar 27

4. Maka repository akan otomatis di create dan code sudah berhasil di push ke repository tujuan.



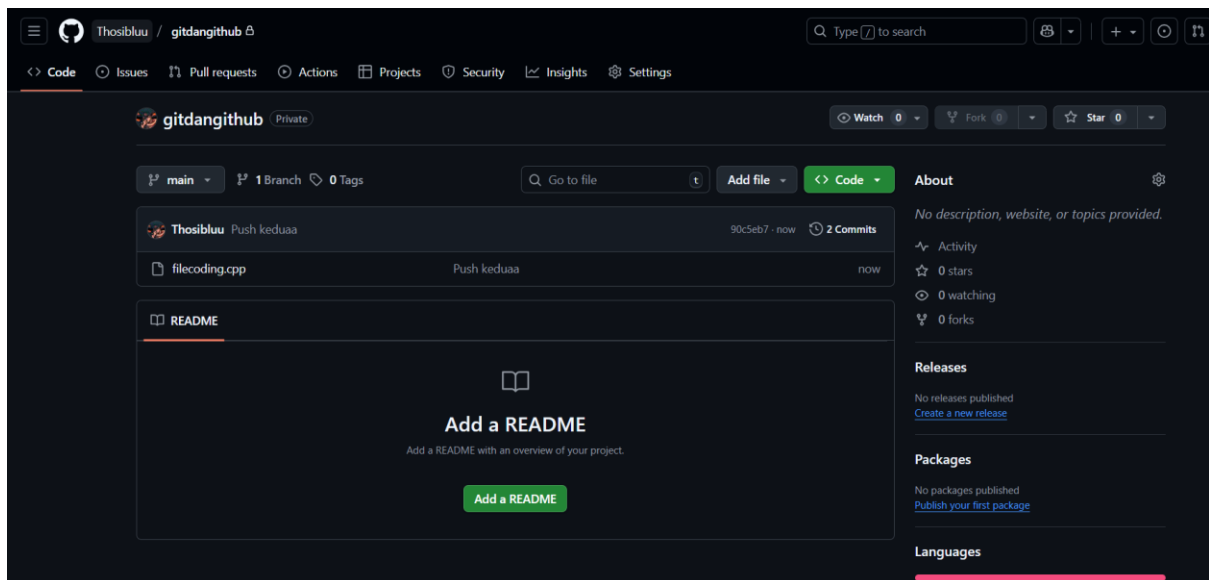
Gambar 28

5. Jika ingin melakukan push Kembali dengan perubahan code, anda hanya perlu melakukan hal yang sama, namun pada source Control tidak perlu memasukkan repository, hanya perlu memasukkan message untuk commit, lalu push dan sync.



Gambar 29

Pada Repository juga akan terjadi perubahan



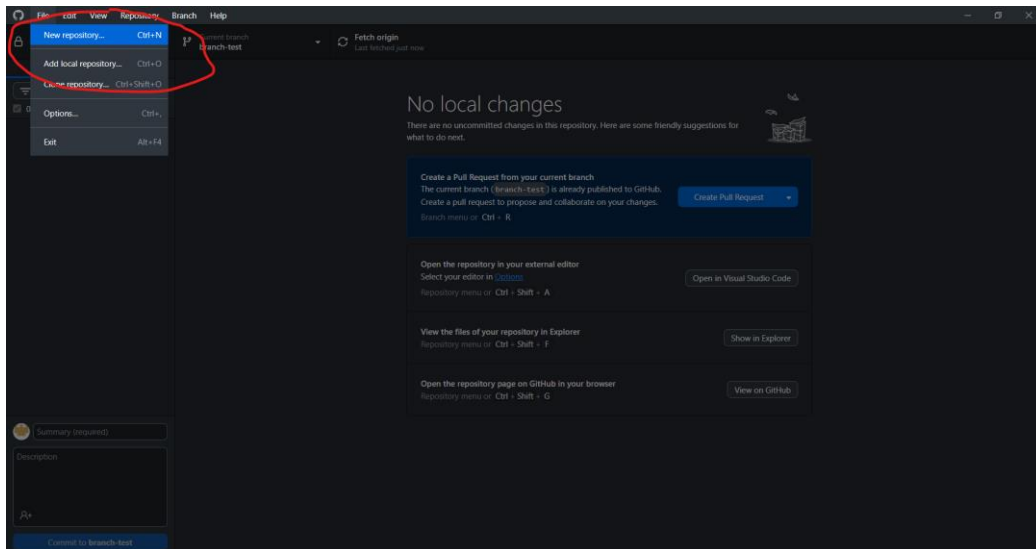
Gambar 30

G. Cara melakukan push menggunakan github desktop

Metode ini menggunakan aplikasi GitHub Desktop yang berbasis GUI (Graphical User Interface), sehingga lebih mudah bagi pemula karena tidak perlu menghafal command.

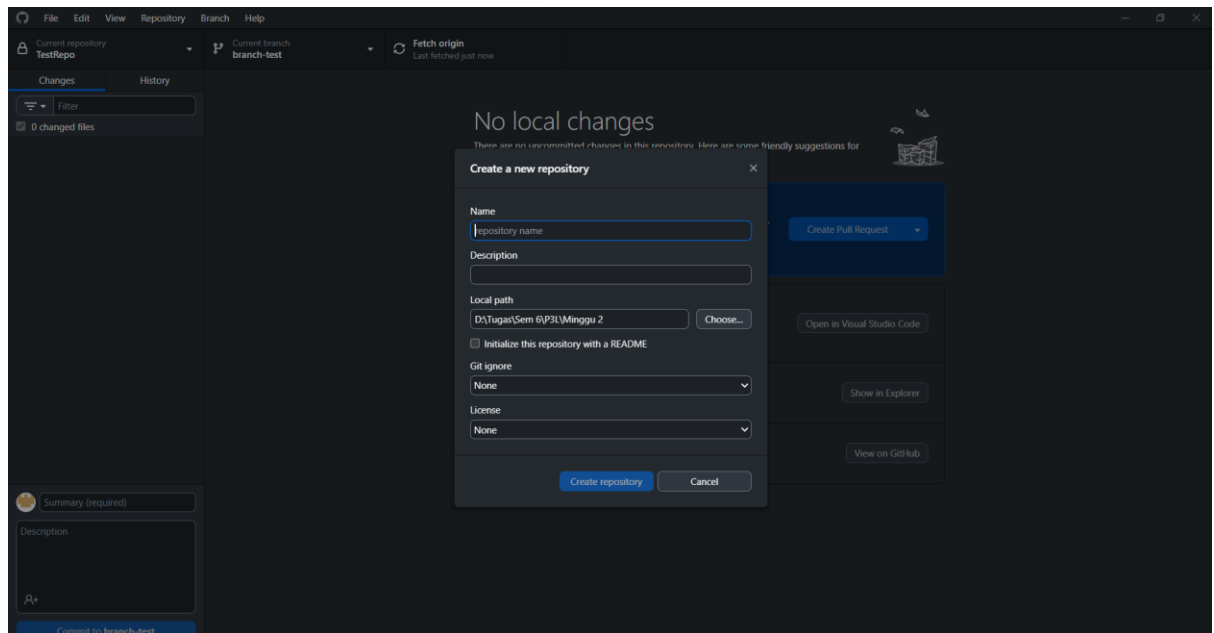
Langkah-Langkah:

1. Buka Aplikasi GitHub Desktop dari halaman utama, pilih "Create a New Repository from your Hard Drive...". Jika Anda sudah memiliki repository lain, Anda bisa pergi ke File > New repository.



Gambar 31

2. Isi Detail Repository Baru Sebuah jendela akan muncul untuk mengisi detail repository lokal Anda.
 - Name: Isi dengan nama repository yang Anda inginkan. Nama ini akan sama dengan nama folder proyek Anda.
 - Description: Berikan deskripsi singkat tentang proyek Anda (opsional).
 - Local Path: Arahkan ke lokasi folder proyek yang sudah Anda siapkan.
 - Initialize this repository with a README: Sebaiknya dicentang untuk membuat file README.
 - Klik tombol "Create repository".



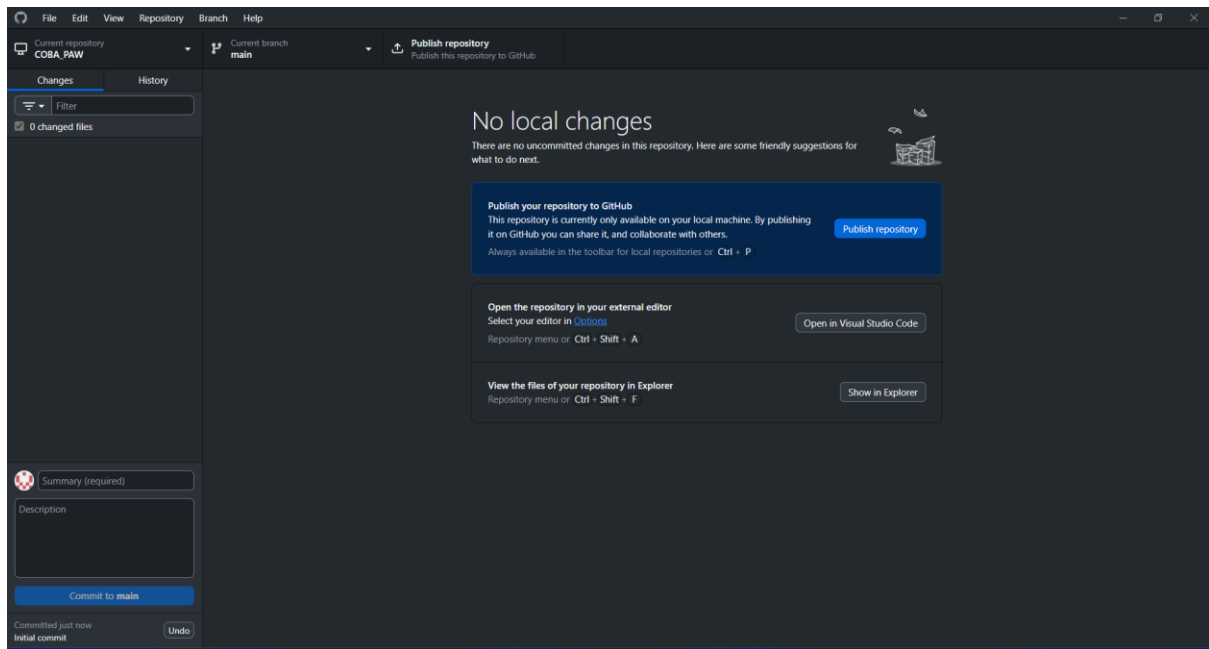
Gambar 32

3. Publikasikan Repository ke GitHub Setelah repository lokal dibuat, GitHub Desktop akan mendeteksi bahwa repository ini belum ada di akun GitHub online Anda.

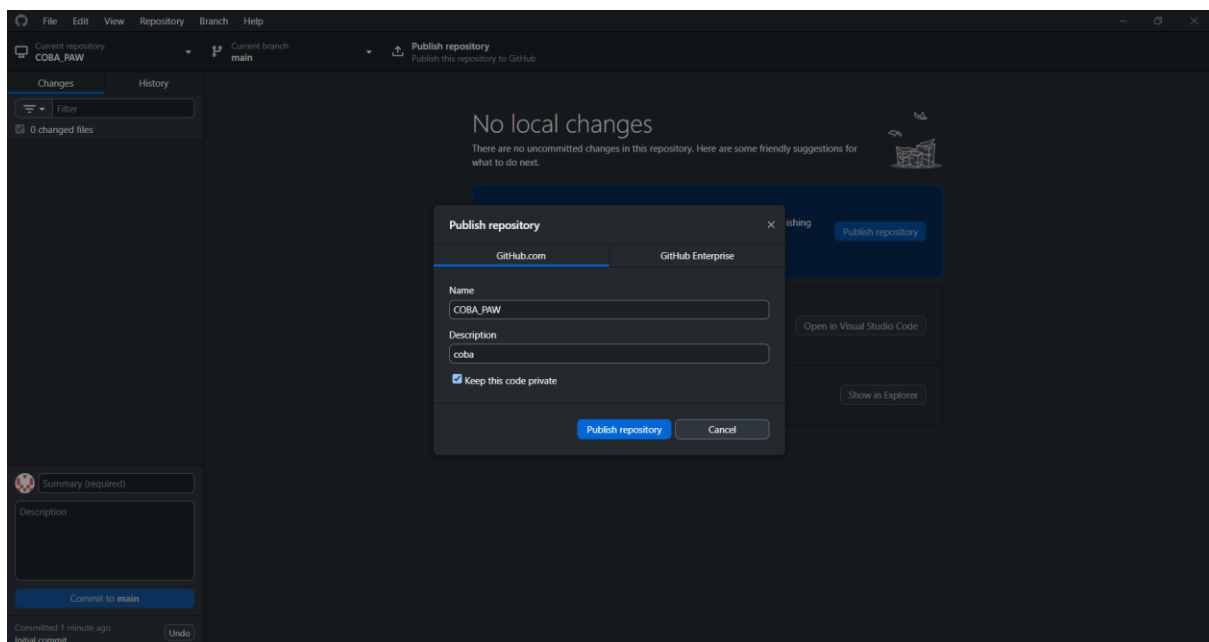
- Anda akan melihat tombol "Publish repository" di bagian tengah layar. Klik tombol tersebut.
- Sebuah jendela baru akan muncul. Pastikan nama dan deskripsi sudah benar.
- Anda dapat memilih apakah repository ini bersifat

Public atau Private. Hilangkan centang pada "Keep this code private" untuk menjadikannya Public.

- a. Klik "Publish repository". Aplikasi akan otomatis membuat repository baru di akun GitHub Anda dan melakukan push pertama kali.

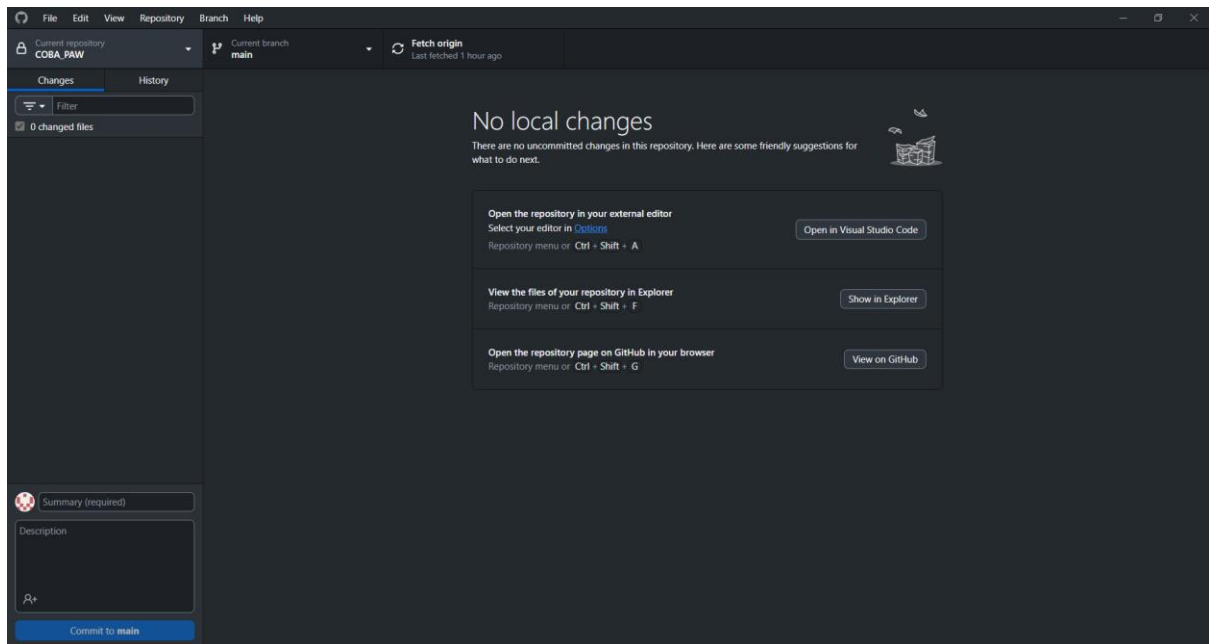


Gambar 33



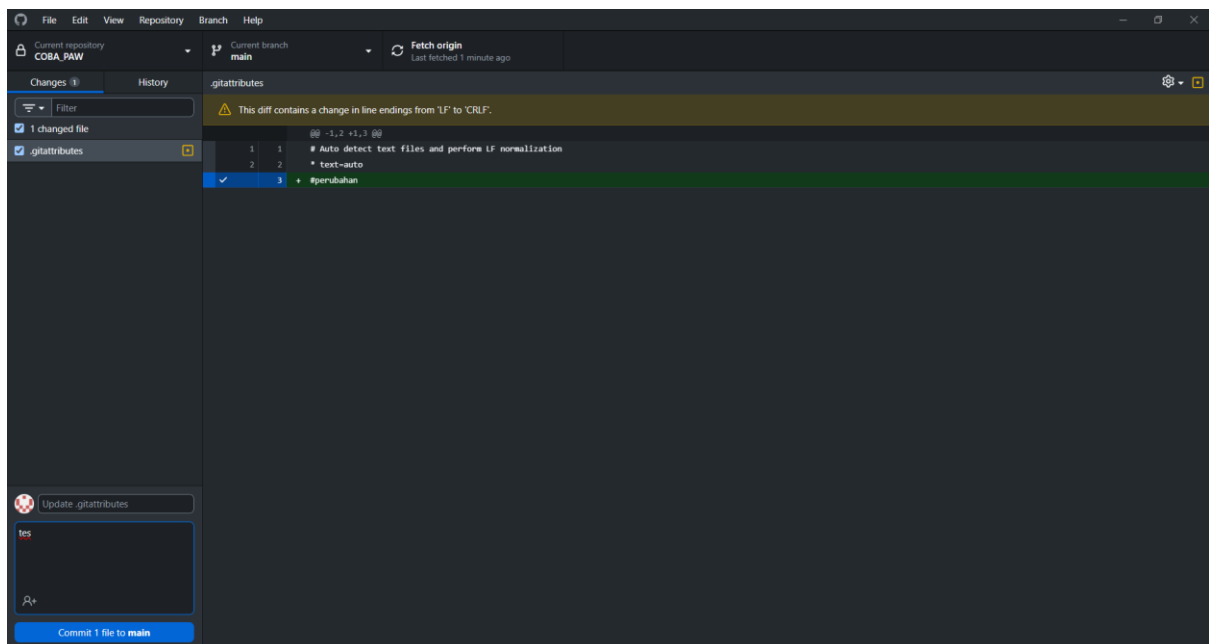
Gambar 34

4. Melakukan Commit dan Push untuk perubahan baru setiap kali anda membuat perubahan pada file di dalam folder proyek Anda (menambah, mengubah, atau menghapus kode), GitHub Desktop akan mendeteksinya secara otomatis.
 - a. Buka GitHub Desktop lalu Open in Visual Studio Code.



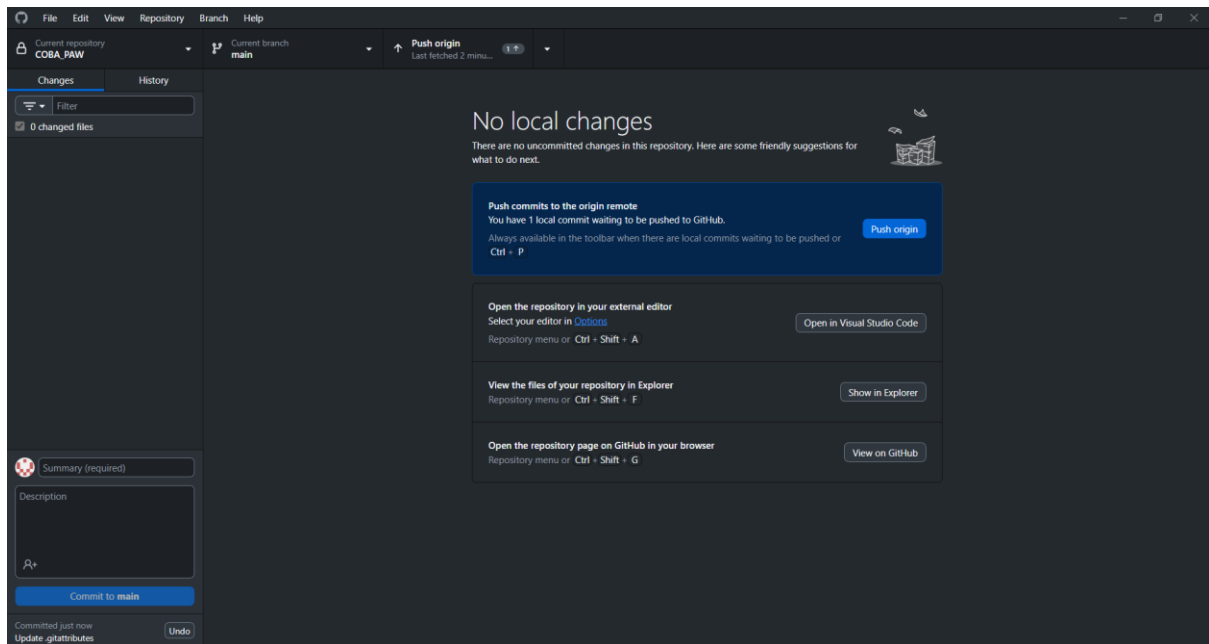
Gambar 35

- b. Kemudian jika ada yang berubah maka, anda akan melihat daftar file yang berubah di kolom "Changes" sebelah kiri.
- c. Di bagian bawah kiri, tuliskan pesan commit Anda pada kolom "Summary" (wajib diisi) dan deskripsi tambahan jika perlu.
 - a. Klik tombol "Commit to main".



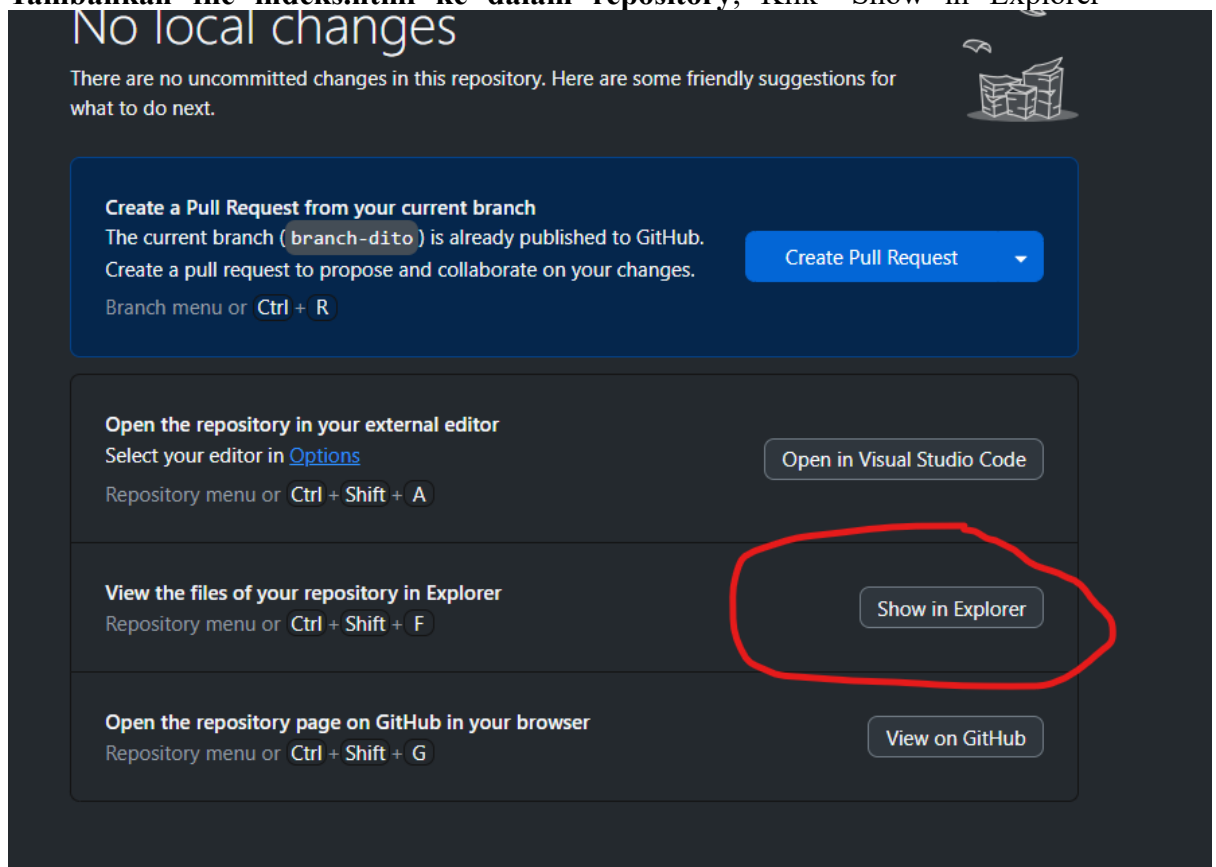
Gambar 36

- b. Setelah commit berhasil, tombol di bagian atas akan berubah menjadi "Push origin" dengan notifikasi yang menunjukkan jumlah commit yang siap diunggah. Klik tombol tersebut untuk mengirim semua perubahan ke repository GitHub Anda.



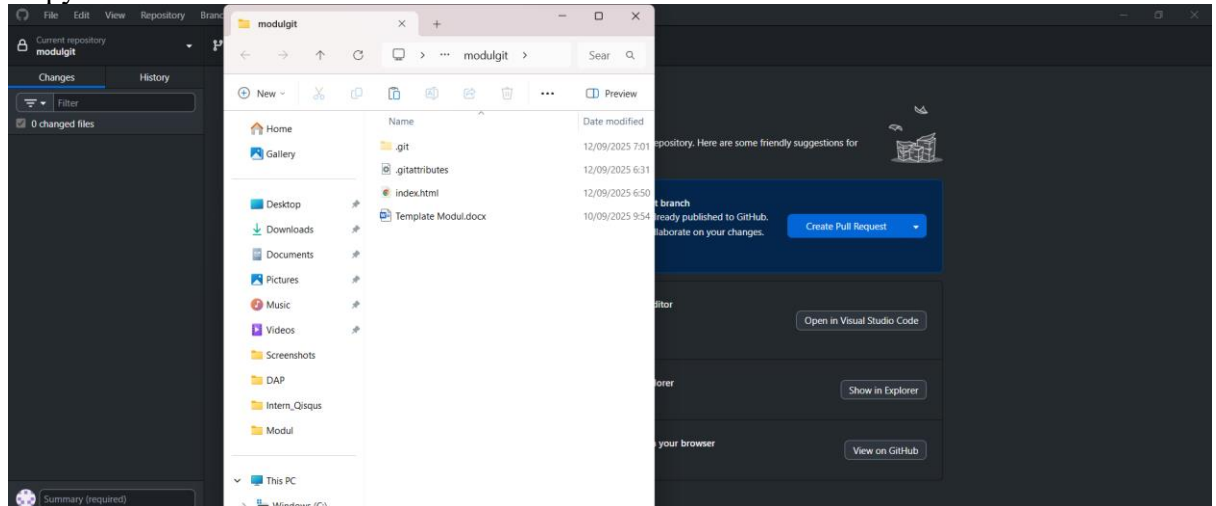
Gambar 37

5. Tambahkan file indeks.html ke dalam repository, Klik “Show in Explorer”



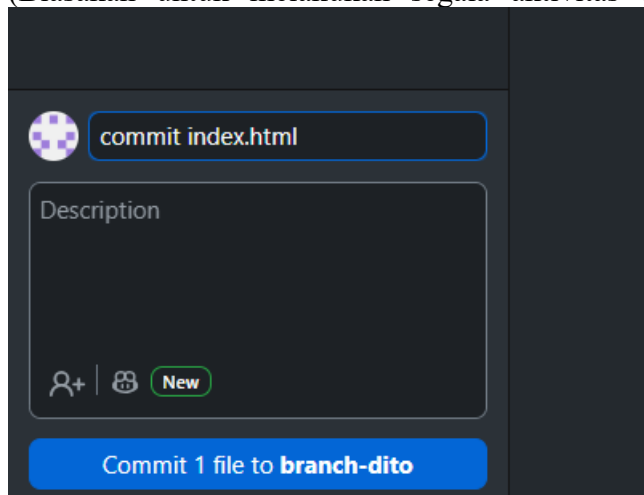
Gambar 38

6. Copy file indeks.html ke dalam folder

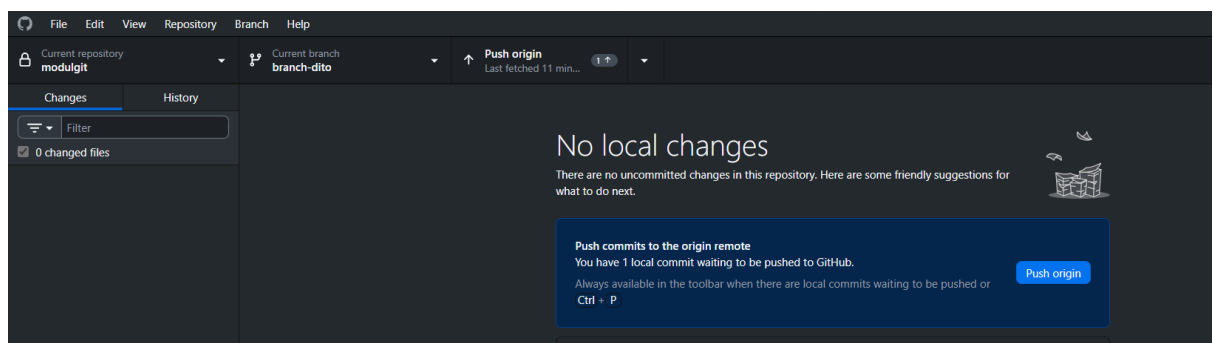


Gambar 39

7. Lakukan commit dengan menyertakan pesan yang jelas dan lakukan push origin. (Biasakan untuk melakukan segala aktivitas di dalam branch terlebih dahulu).

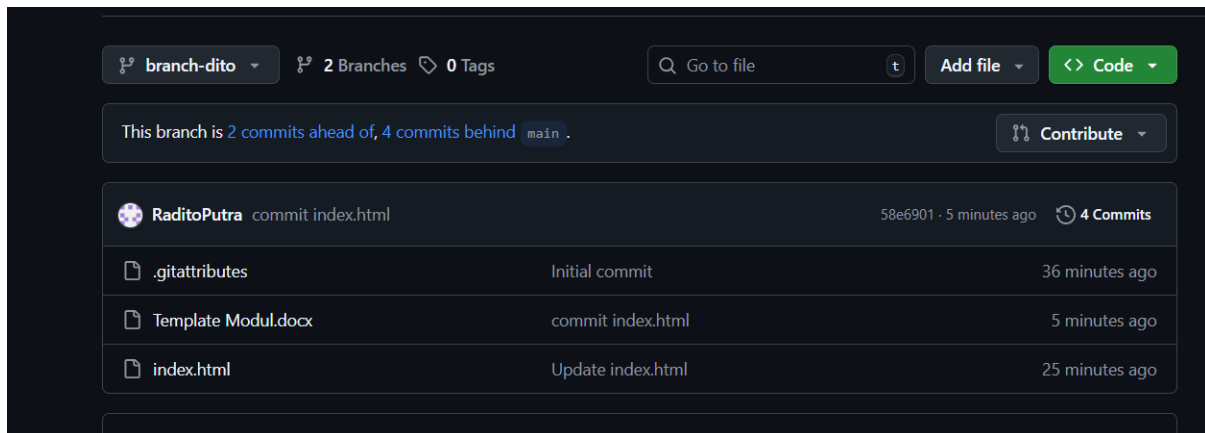


Gambar 40



Gambar 41

8. Verifikasi di Situs GitHub Buka repository Anda di situs github.com untuk memastikan semua file dan perubahan terakhir sudah berhasil diunggah.



Gambar 42

Dengan cara ini, seluruh proses mulai dari init, add, commit, hingga push dapat dilakukan dengan beberapa klik saja melalui antarmuka visual GitHub Desktop.

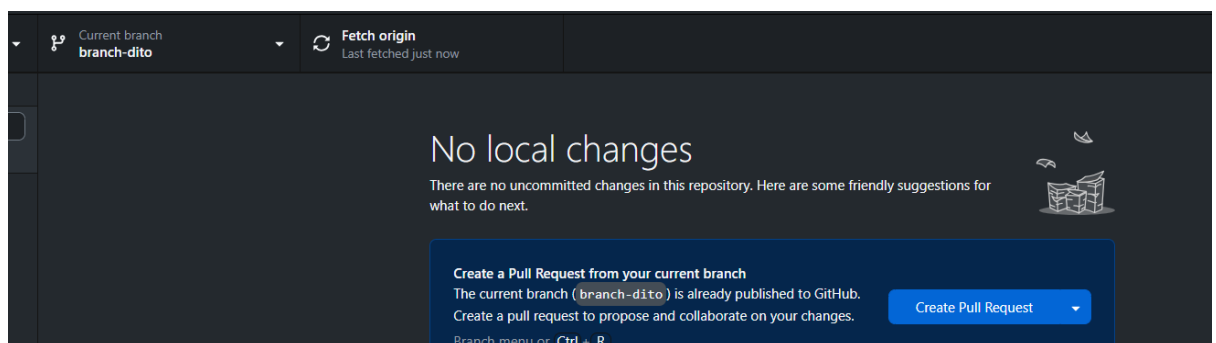
H. Cara Melakukan Pull dari Remote Repository

Operasi pull digunakan untuk mengambil dan mengintegrasikan perubahan terbaru dari repositori Github ke repositori lokal di komputer Anda.

1. Cara Melakukan Pull Menggunakan GitHub Desktop

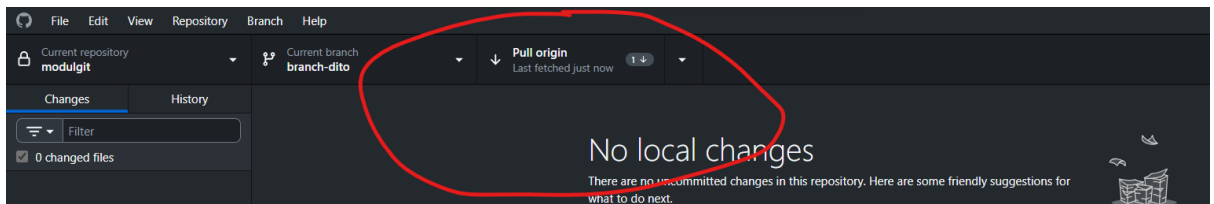
GitHub Desktop menyederhanakan proses pull menjadi beberapa klik saja.

1. **Fetch Origin:** Pertama, buka aplikasi GitHub Desktop dan pastikan Anda berada di repositori yang benar. Aplikasi secara berkala akan memeriksa perubahan dari remote. Untuk memeriksa secara manual, klik tombol Fetch origin di bagian atas. Tombol ini akan mengambil data terbaru dari GitHub tanpa mengubah file lokal Anda.



Gambar 43

2. **Pull Changes:** Jika ada perubahan baru di repositori remote yang belum ada di lokal Anda, tombol "Fetch origin" akan berubah menjadi Pull origin. Tombol ini juga akan menunjukkan jumlah commit baru yang siap ditarik.
3. **Klik Tombol Pull Origin:** Klik tombol Pull origin untuk mengunduh dan menggabungkan semua perubahan tersebut ke branch lokal Anda. Setelah selesai, repositori lokal Anda akan sepenuhnya sinkron dengan versi terbaru di GitHub.

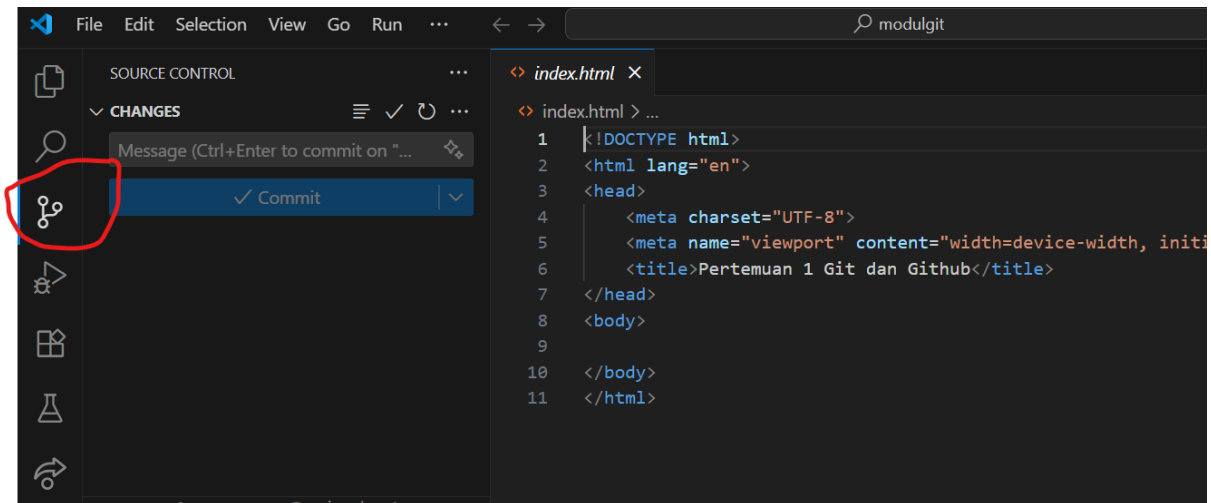


Gambar 44

2. Cara Melakukan Pull Menggunakan Visual Studio Code (VSCode)

VSCode terintegrasi langsung dengan Git, sehingga Anda bisa melakukan pull tanpa meninggalkan editor.

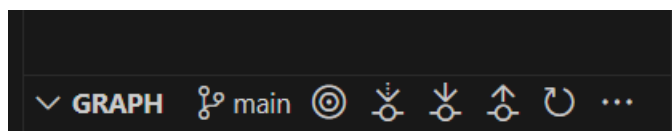
1. Buka Panel Source Control: Buka folder proyek Anda di VSCode, lalu klik ikon Source di menu sebelah kiri.



Gambar 45

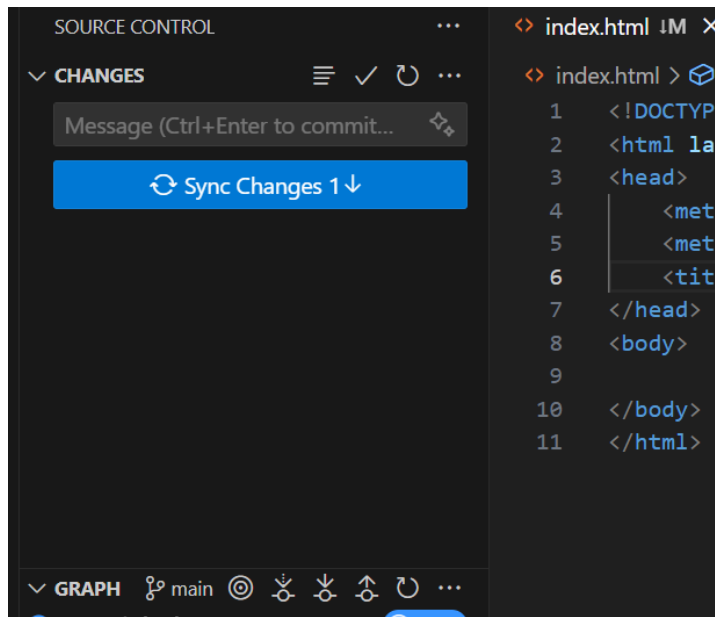
2. Gunakan Opsi "Sync Changes":

- Di bagian bawah status bar VSCode, Anda akan melihat ikon panah melingkar dengan angka di sebelahnya jika ada perubahan di remote. Angka tersebut menunjukkan jumlah commit yang perlu di-pull (panah ke bawah) dan di-push (panah ke atas).



Gambar 46

- Klik tombol Sync Changes tersebut. VSCode akan otomatis melakukan pull (menarik perubahan dari remote) dan kemudian push (mengirim perubahan lokal Anda). Ini adalah cara paling umum dan efisien.



Gambar 47