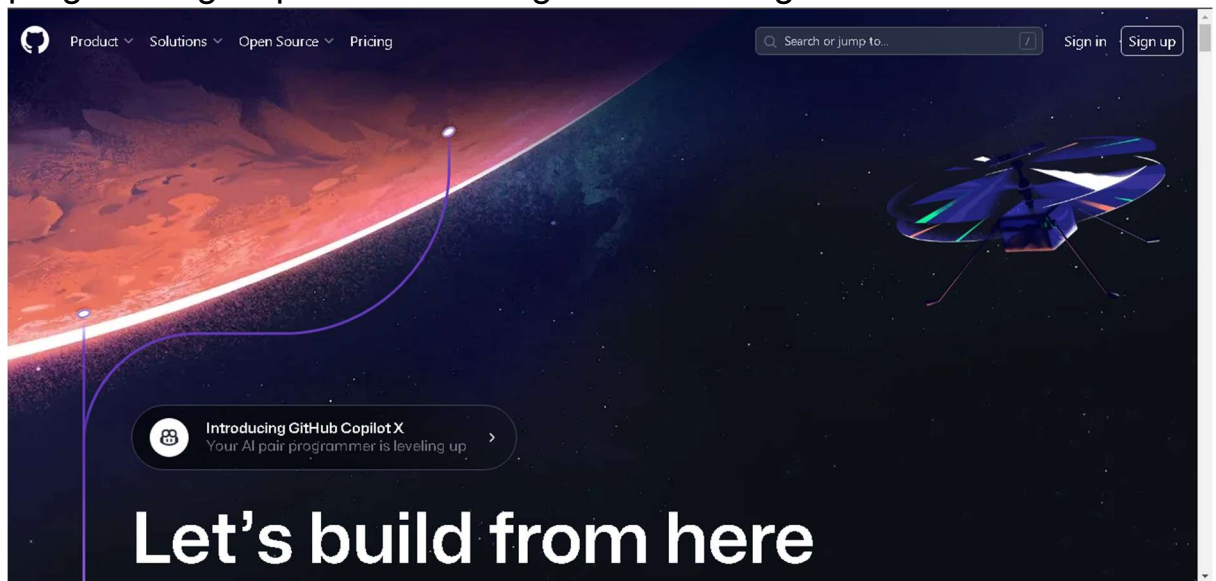


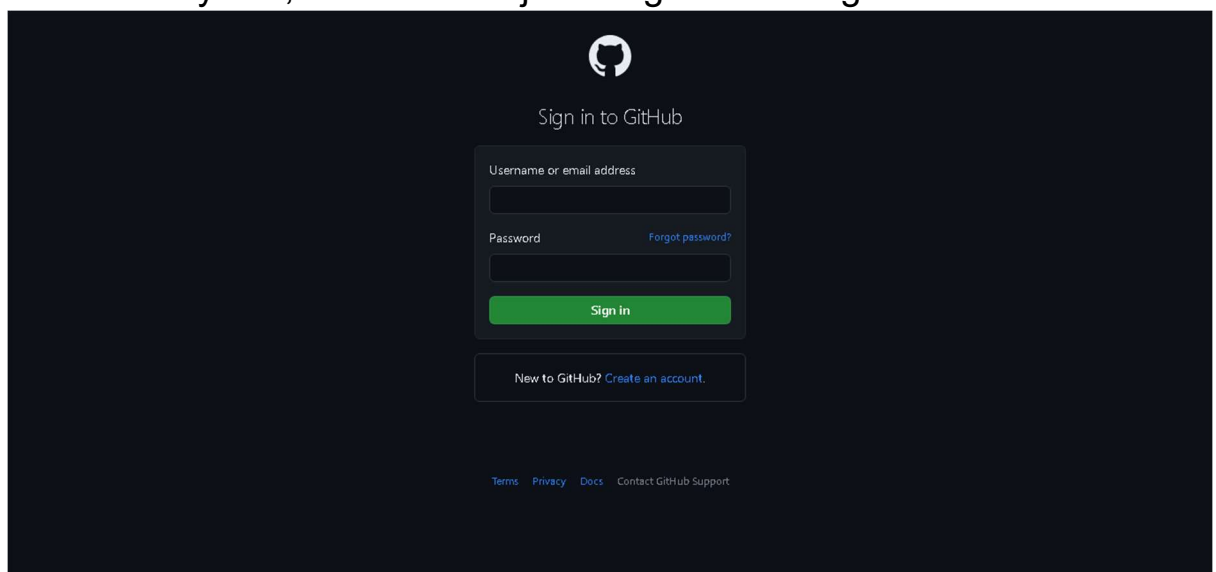
Tutorial Github

Cara Login di Github

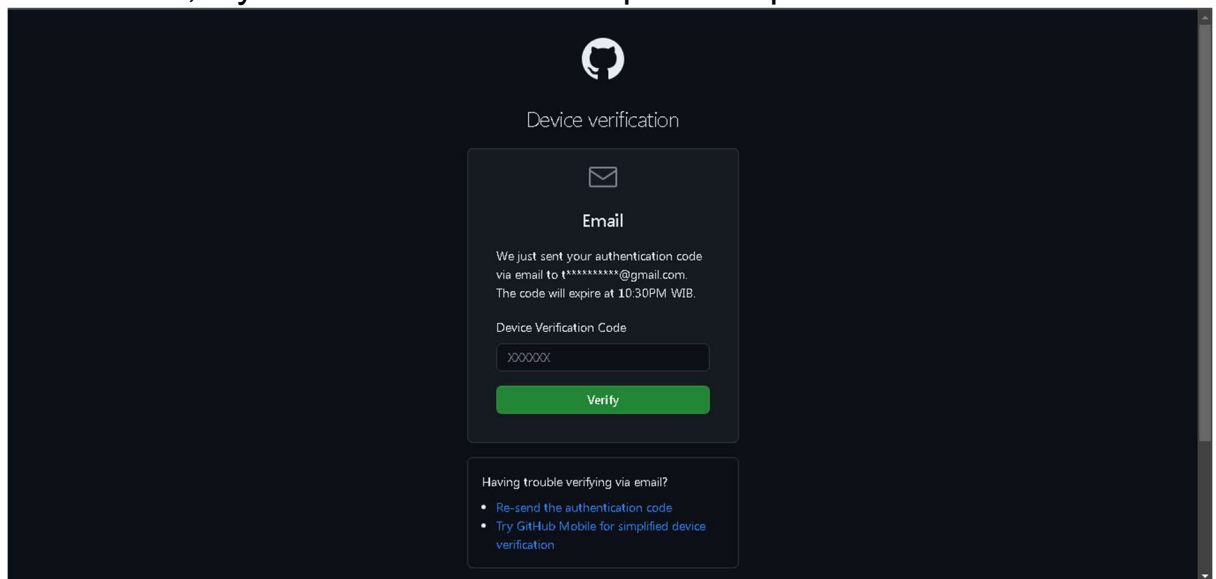
1. Buka website <https://github.com> di browser, kemudian.
2. Kalian klik “Sign In” untuk login, jika tidak memiliki akun kalian bisa pergi ke “Sign Up” dan ikuti Langkah “Cara Registrasi di Github”.



3. Silahkan isi Username atau Alamat Email dan Password anda. Jika sudah yakin, klik tombol hijau dengan teks “Sign In”



4. Jika sudah, layar kalian akan menampilkan seperti ini.

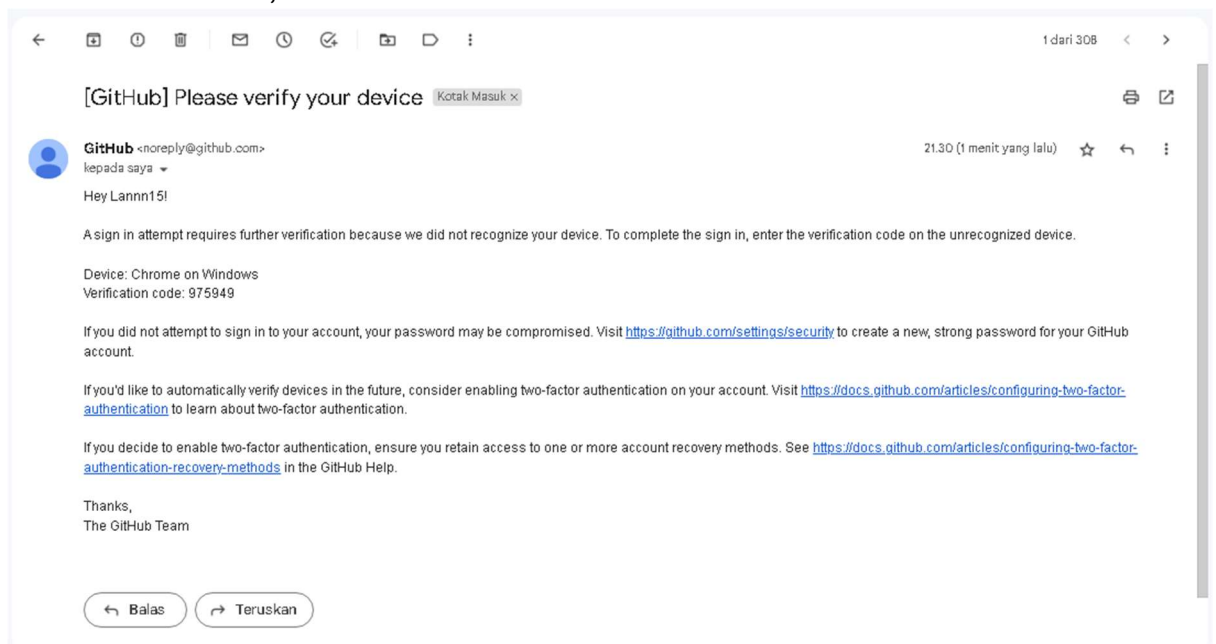


Jika ini tidak tertampil dianda, itu berarti kalian memasuki Username, Alamat Email atau Password yang salah. Jika lupa kalian bisa pergi ke-step 3 untuk melakukan Forgot Password.

Kalian buka <https://gmail.com> di browser kalian dan kemudian cari email ini untuk mengambil kodenya.

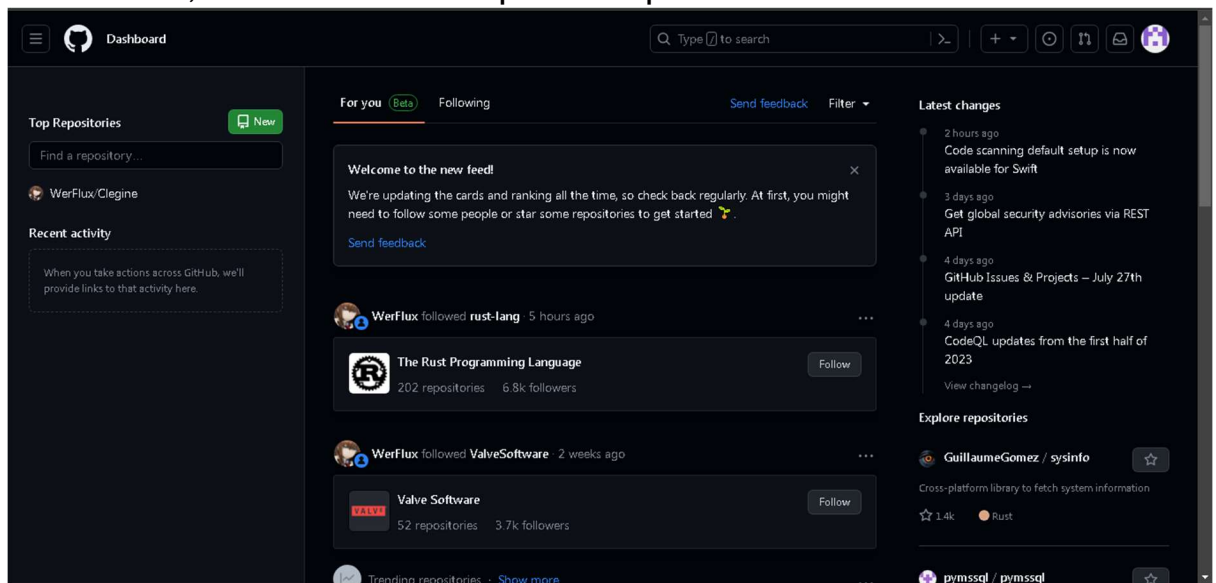
GitHub [GitHub] Please verify your device - Hey Lannn15! A sign in attempt requires further verification because we did not re... 21.30

Kemudian buka,



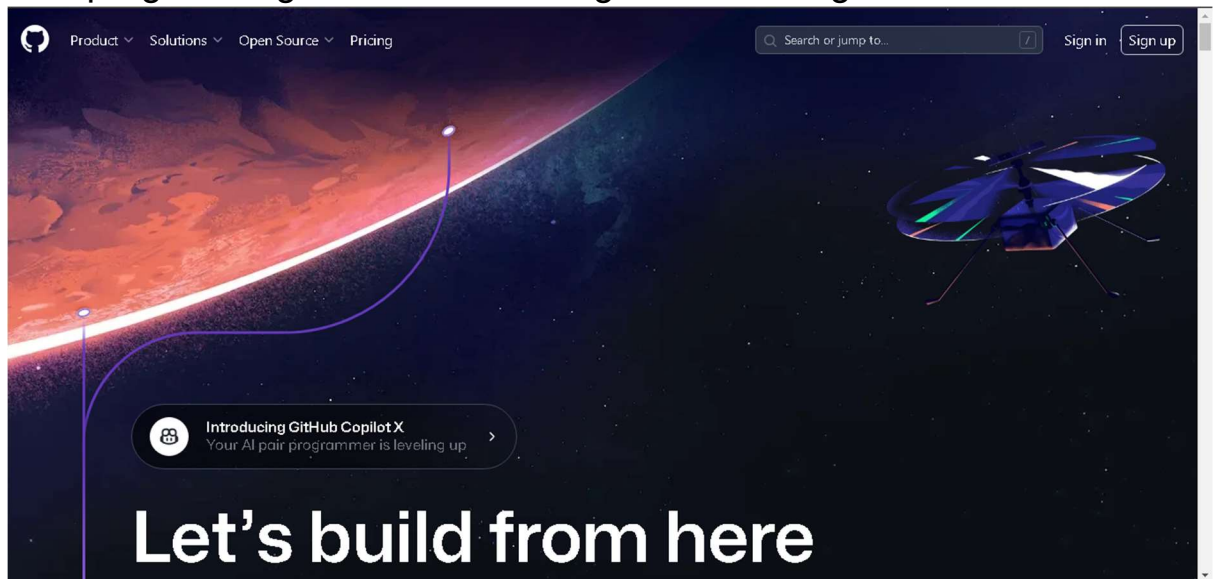
Copy Verification Code tersebut kemudian taruh dikolom Device Verification.

5. Jika sudah, kalian akan ditampilkan seperti ini.

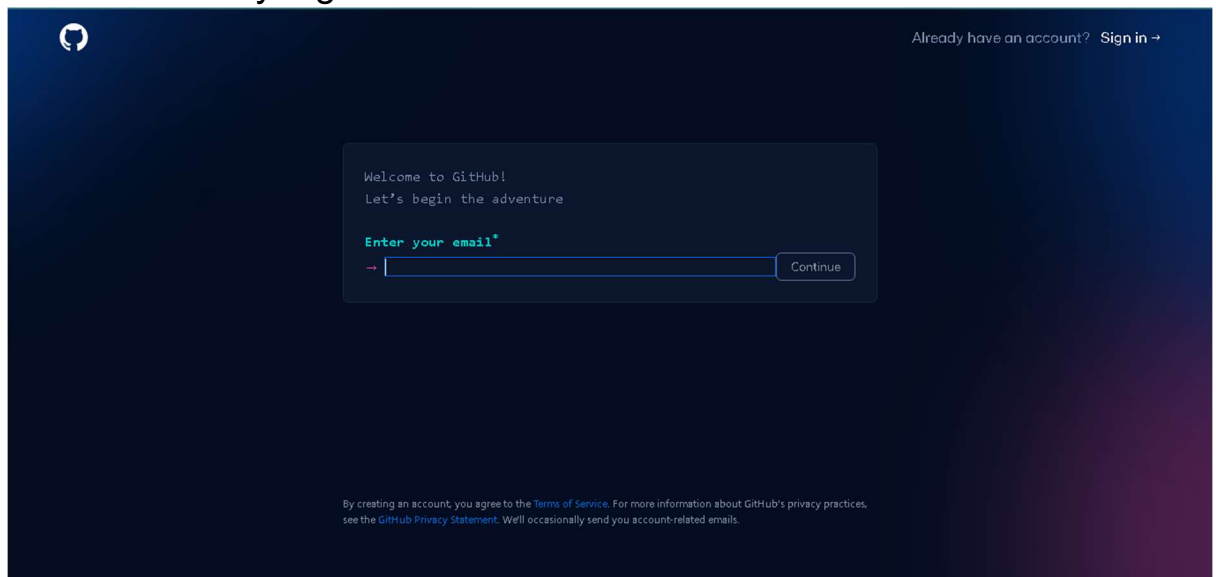


Cara Registrasi di Github

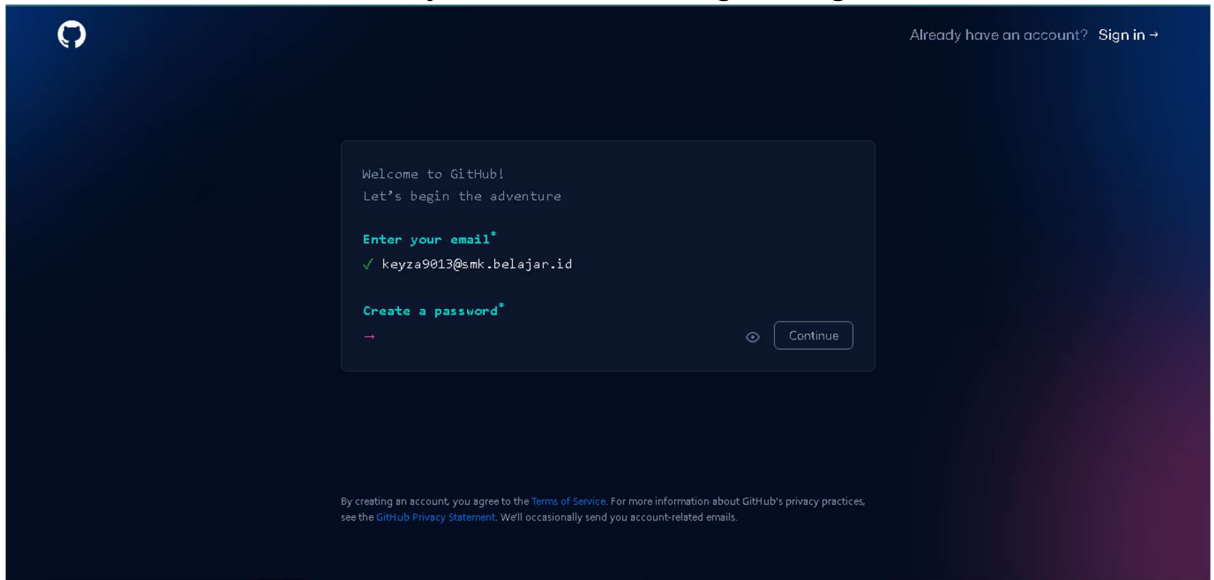
1. Buka website <https://github.com> di browser anda
2. Kalian klik “Sign Up” untuk register, jika sudah memiliki akun kalian bisa pergi ke “Sign In” dan ikuti Langkah “Cara Login di Github”



3. Silahkan isi Alamat Email anda yang valid. Nantinya akan ada kode verifikasi yang dikirim ke Alamat Email anda.

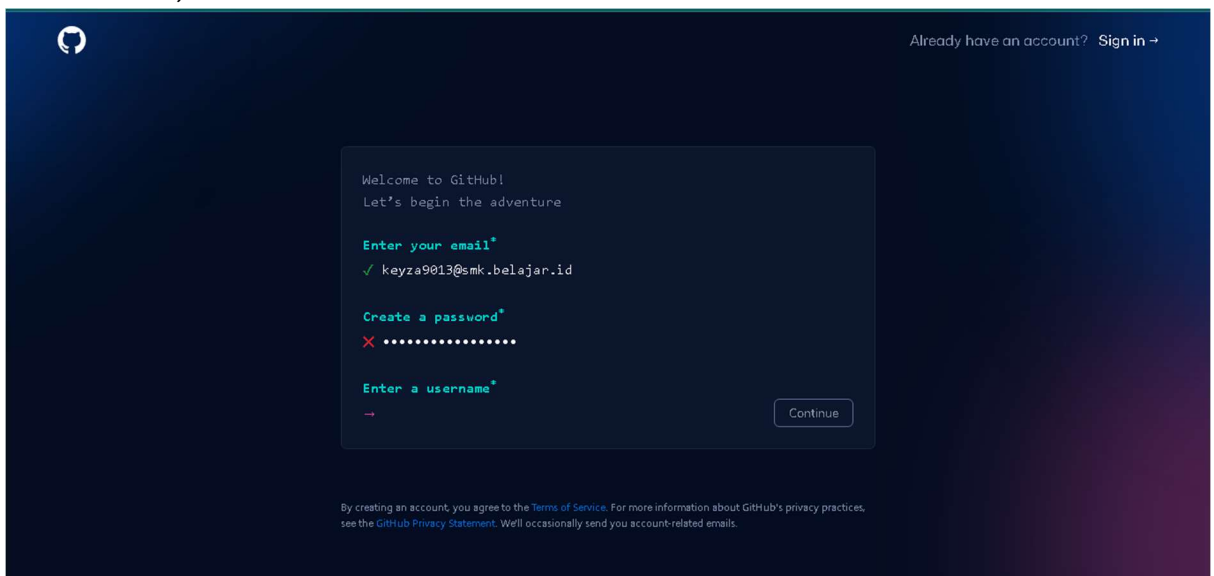


Kemudian kalian isi Password, pastikan Password harus lebih dari 15 karakter atau setidaknya 8 karakter dengan angka dan alfabet.



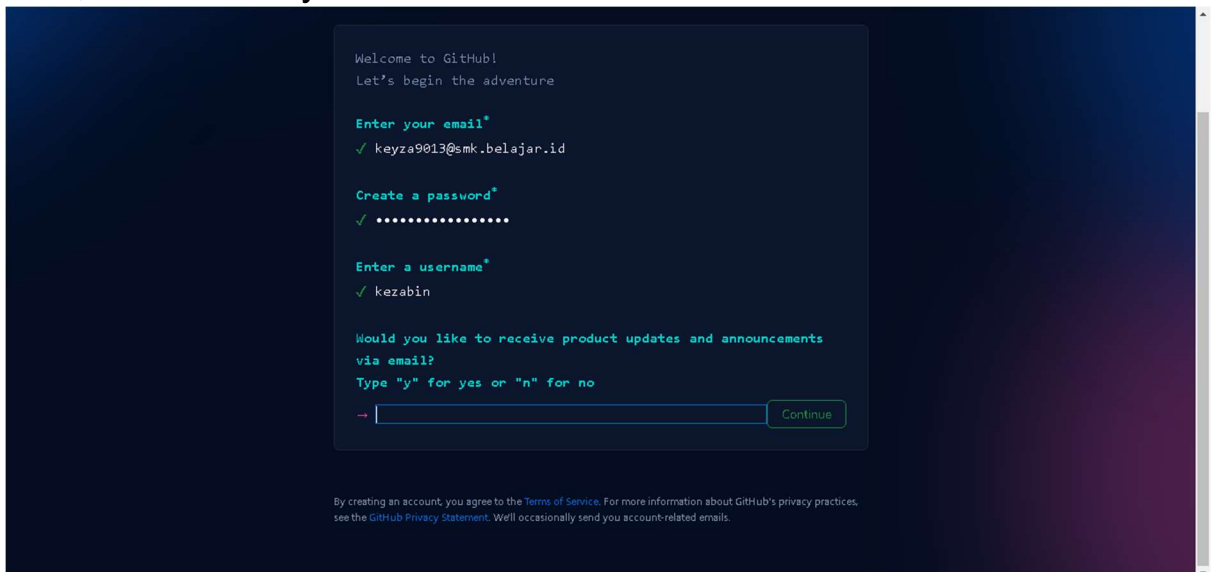
The screenshot shows the GitHub account creation interface. At the top left is the GitHub logo, and at the top right is the text "Already have an account? Sign in →". The main content area has a dark blue background with a white box containing the following text: "Welcome to GitHub!", "Let's begin the adventure", "Enter your email*", "✓ keyza9013@smk.belajar.id", "Create a password*", and a "Continue" button. Below the white box, there is a small line of text: "By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails."

Kemudian, kamu isi Username dibawah.

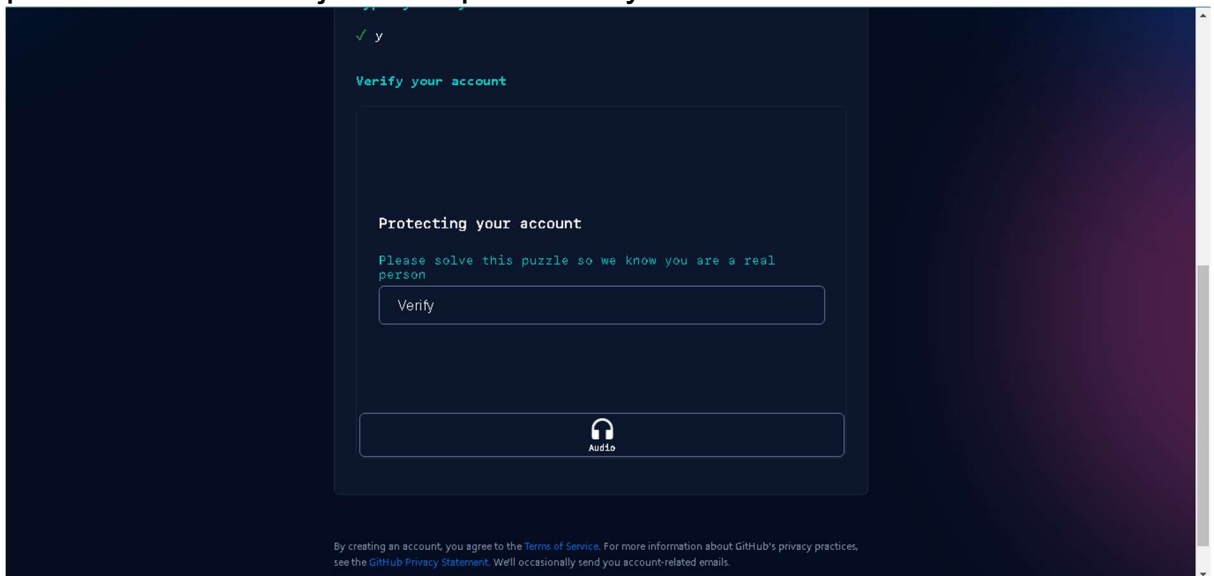


The screenshot shows the GitHub account creation interface. At the top left is the GitHub logo, and at the top right is the text "Already have an account? Sign in →". The main content area has a dark blue background with a white box containing the following text: "Welcome to GitHub!", "Let's begin the adventure", "Enter your email*", "✓ keyza9013@smk.belajar.id", "Create a password*", "X", "Enter a username*", and a "Continue" button. Below the white box, there is a small line of text: "By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails."

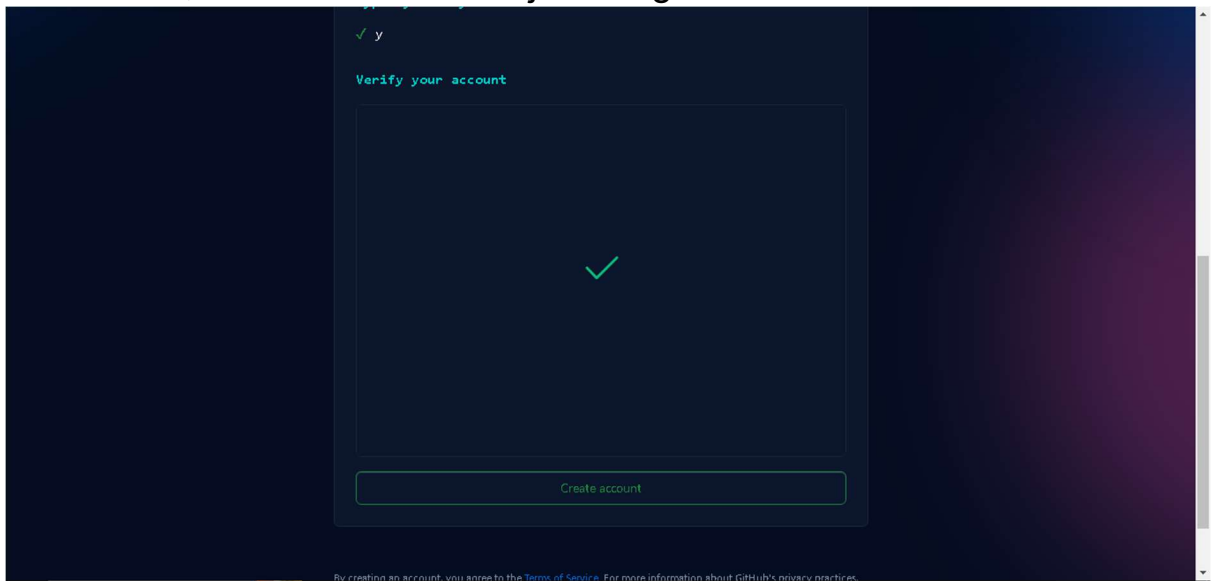
Lalu, kalian ketik 'y' dibawah.

A screenshot of the GitHub account creation interface. The background is a dark blue gradient. A central white box contains the following text: "Welcome to GitHub! Let's begin the adventure". Below this are three input fields, each with a green checkmark and a label: "Enter your email*" with the value "keyza9013@smk.belajar.id", "Create a password*" with a masked password ".....", and "Enter a username*" with the value "kezabin". Below these is a question: "Would you like to receive product updates and announcements via email? Type 'y' for yes or 'n' for no". There is a text input field with a red arrow icon on the left and a green "Continue" button on the right. At the bottom of the white box, there is a small line of text: "By creating an account, you agree to the Terms of Service. For more information about GitHub's privacy practices, see the GitHub Privacy Statement. We'll occasionally send you account-related emails."

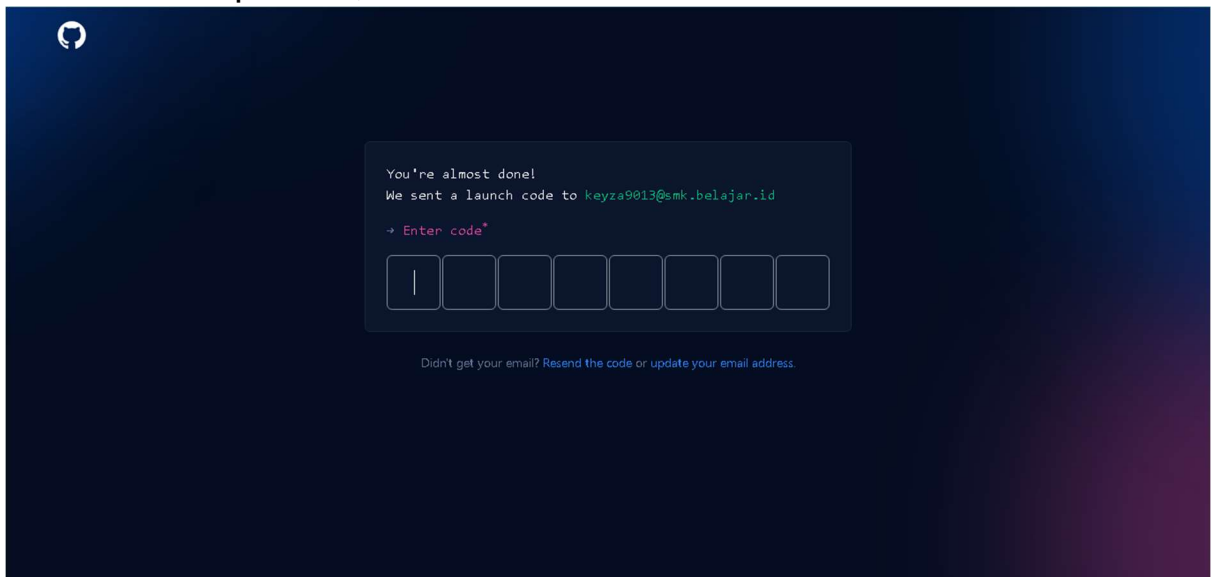
Lalu, kalian akan diminta untuk verifikasi akun kita dengan menjawab beberapa pertanyaan atau puzzle. Kamu bisa memulainya dengan mengklik "Verify", selesaikan pertanyaan atau puzzle untuk menuju ke-step berikutnya.

A screenshot of the GitHub account verification interface. The background is a dark blue gradient. A central white box contains the following text: "Verify your account". Below this is a section titled "Protecting your account" with the text "Please solve this puzzle so we know you are a real person". There is a button labeled "Verify". Below the button is a section with a headphones icon and the word "Audio". At the bottom of the white box, there is a small line of text: "By creating an account, you agree to the Terms of Service. For more information about GitHub's privacy practices, see the GitHub Privacy Statement. We'll occasionally send you account-related emails."

Jika sudah, kalian klik tombol hijau dengan teks “Create Account”



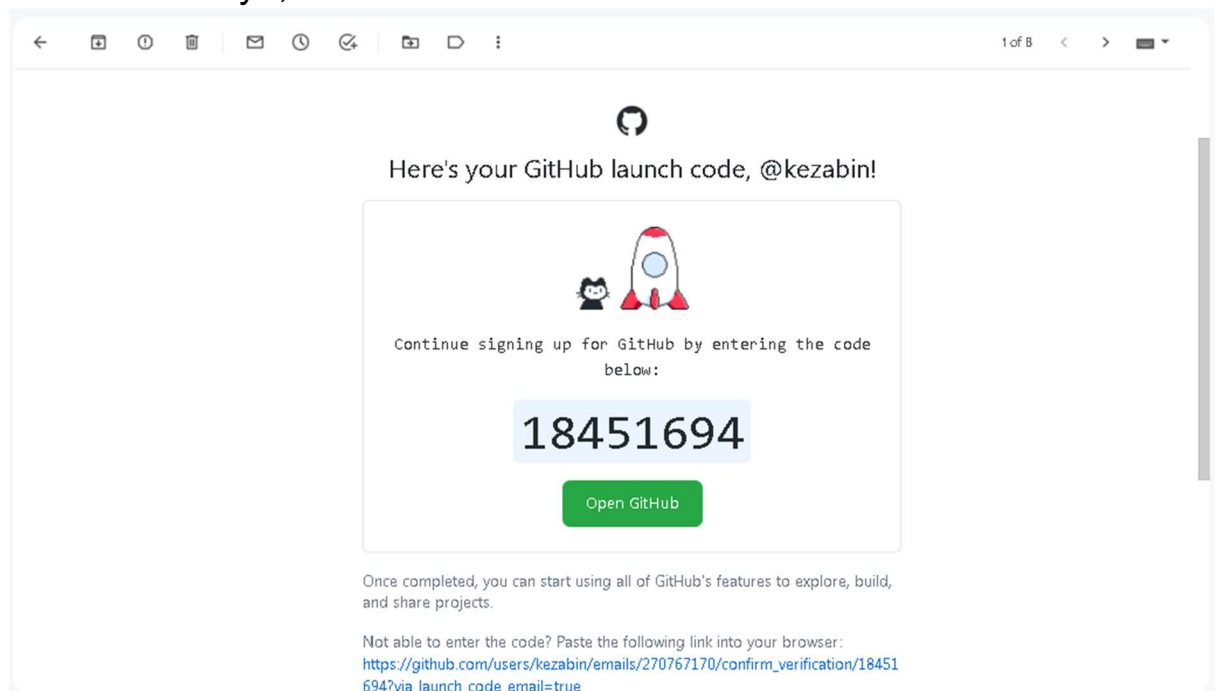
4. Jika sudah seperti ini,



Kalian buka <https://gmail.com> lalu cari Email dari Github, seperti ini.

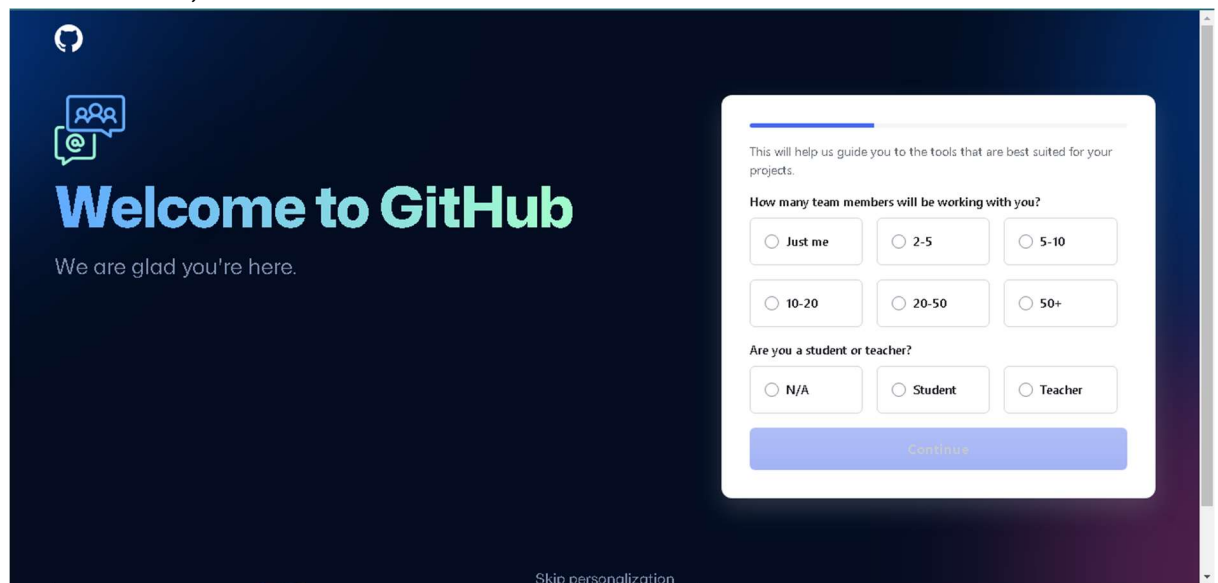


Lalu buka isinya,



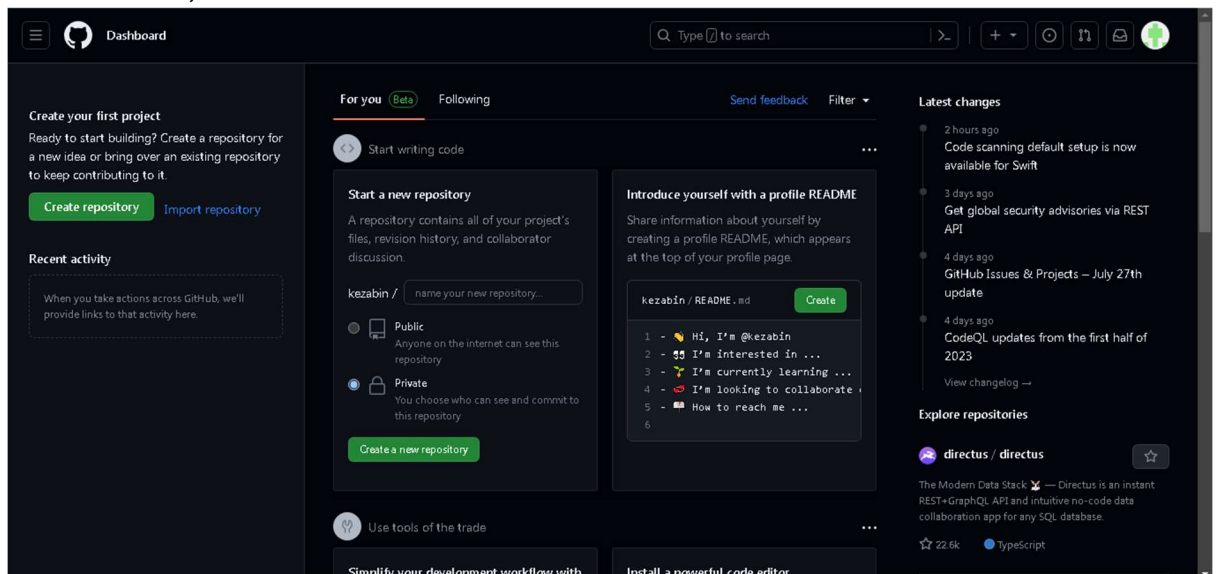
Itu adalah kode verifikasinya, disini kode saya adalah "18451694" lalu isi kolom-kolom di website Github dengan kode tersebut.

5. Jika sudah, maka kalian akan dibawa ke Personalization.



Kalian bisa skip personalization ini atau kalian bisa tetap lanjut, disini saya akan skip personalization

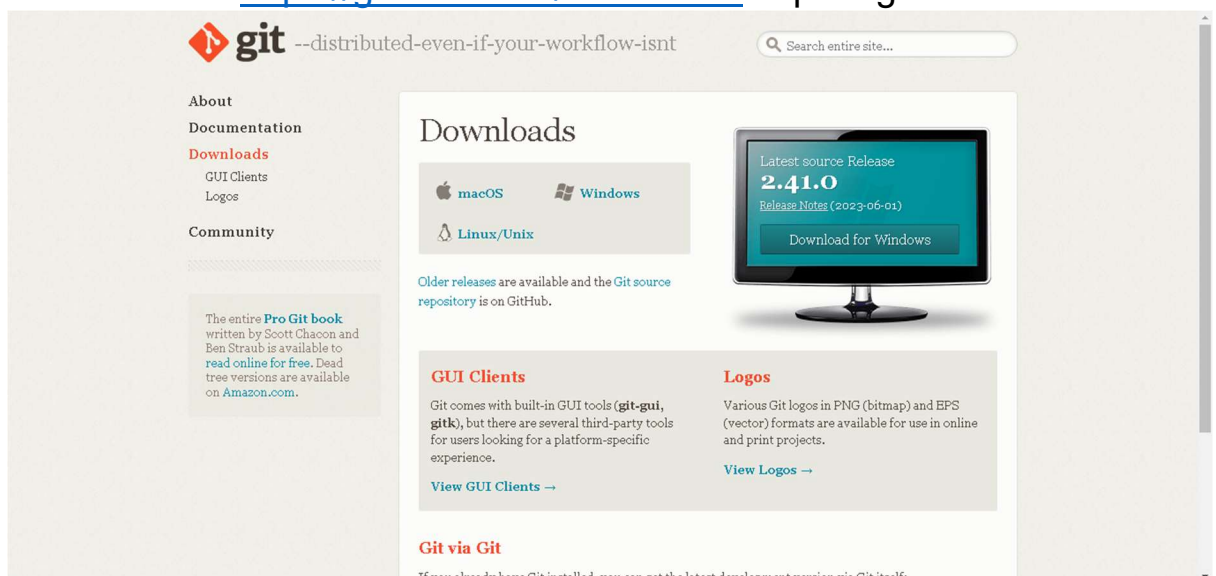
6. Jika sudah, kalian akan dibawa kesini.



Dan selamat, kalian sudah berhasil membuat akun Github. Kalian bisa memulai dengan membuat repository dengan mengklik tombol “Create repository”

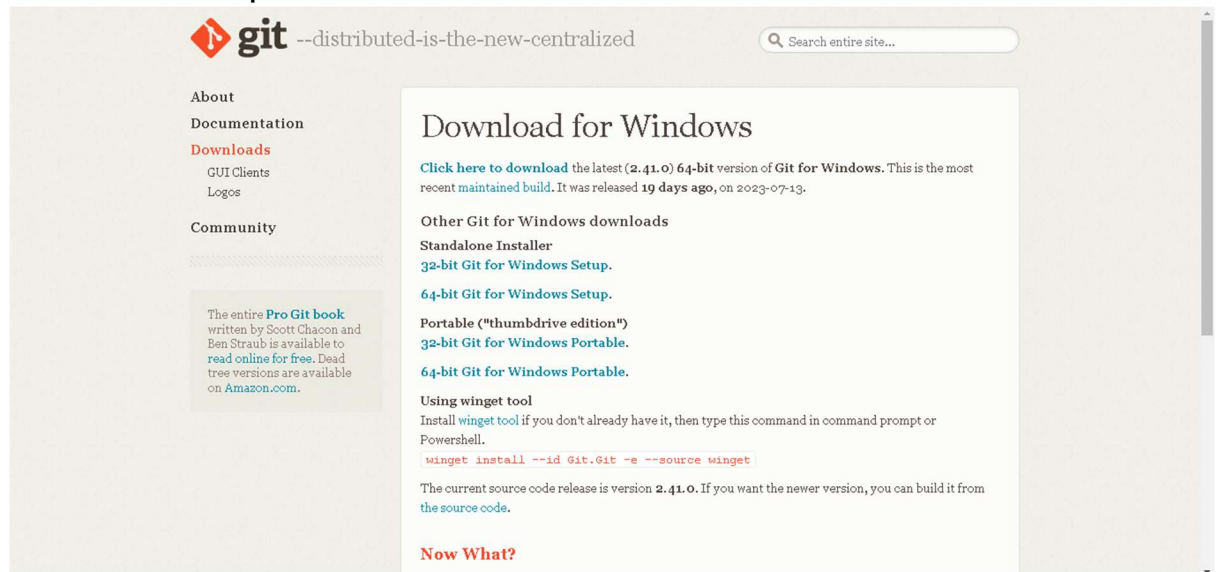
Cara download Git dan menggunakan Git Bash

1. Buka website <https://git-scm.com/downloads> seperti gambar ini.

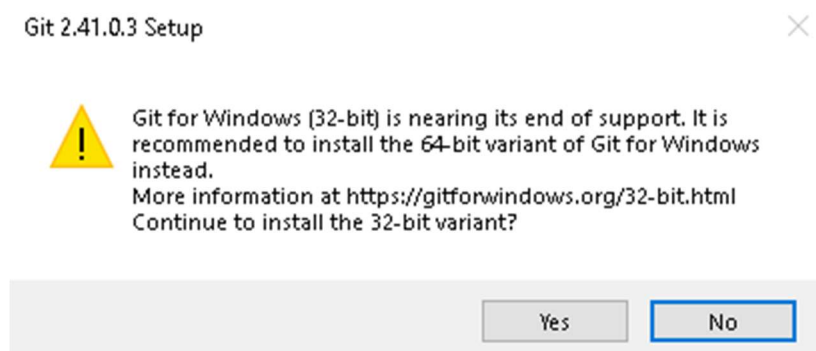


2. Kalian pilih system operasi apa yang kalian gunakan, disini saya akan gunakan Windows.

3. Jika sudah, kalian harus cek bit kalian apakah 32 bit atau 64 bit. Jika PC kalian 64 bit, kalian bisa klik yang 64-bit Git for Windows Setup. Jika PC kalian 32 bit, kalian bisa klik yang 32-bit Git for Windows Setup



Jika kalian download 32 bit kalian akan melihat box ini.

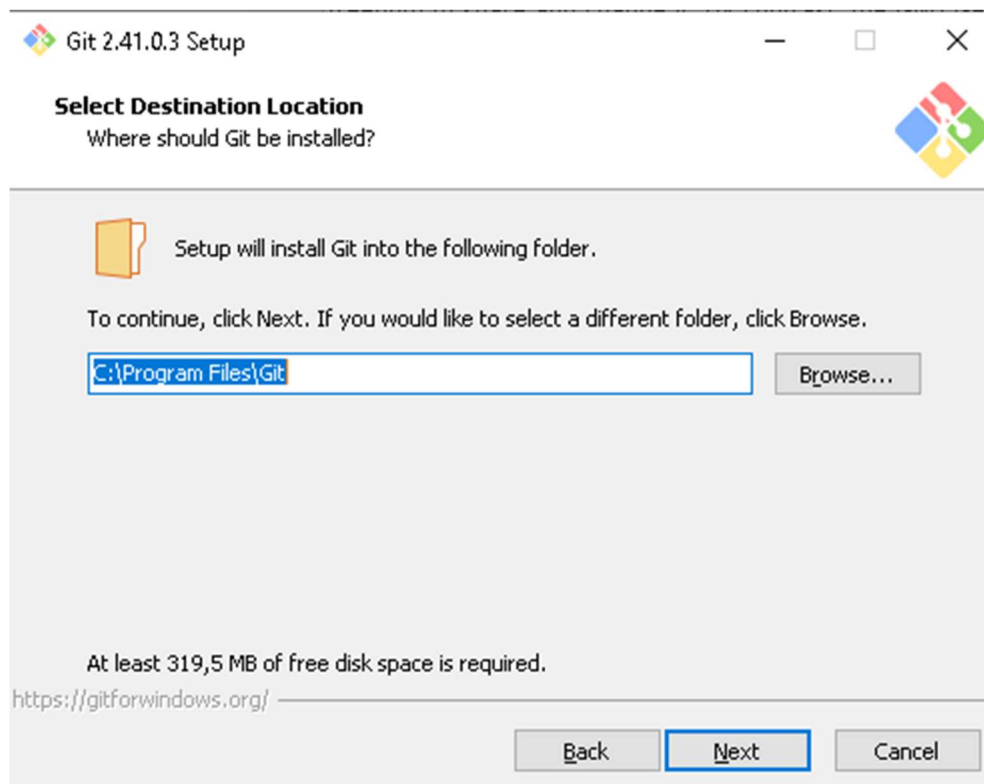


Kalian pilih “Yes” saja dan lanjut ke-step 4, ini dikarenakan Git akan mengakhiri dukungan untuk 32 bit dan diharuskan menggunakan 64 bit.

4. Jika sudah, kalian ikuti saja dibawah ini.
Pilih “Next”



Pilih “Next”



Pilih “Next” lagi.

Select Components

Which components should be installed?



Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

☐ Additional icons

☐ On the Desktop

☒ Windows Explorer integration

☒ Open Git Bash here

☒ Open Git GUI here

☒ Git LFS (Large File Support)

☒ Associate .git* configuration files with the default text editor

☒ Associate .sh files to be run with Bash

☐ Check daily for Git for Windows updates

☐ (NEW!) Add a Git Bash Profile to Windows Terminal

Current selection requires at least 319,4 MB of disk space.

<https://gitforwindows.org/>

Back

Next

Cancel

Pilih "Next"

Select Start Menu Folder

Where should Setup place the program's shortcuts?



□ —
□ — Setup will create the program's shortcuts in the following Start Menu folder.
□ —

To continue, click Next. If you would like to select a different folder, click Browse.

Git

Browse...

☐ Don't create a Start Menu folder

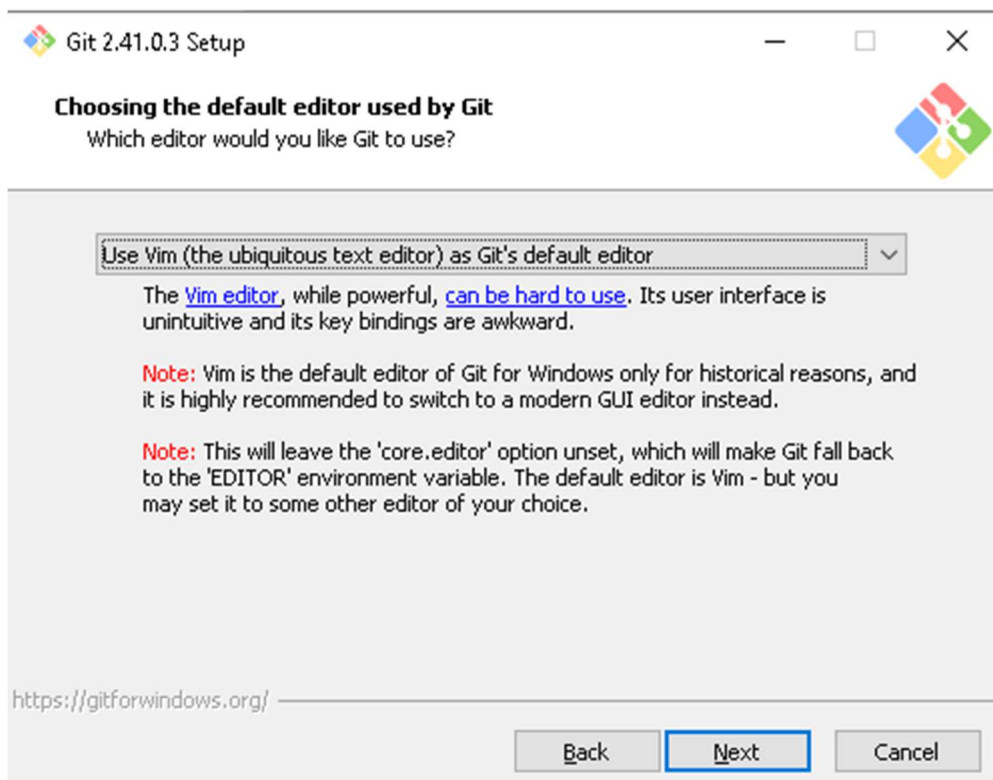
<https://gitforwindows.org/>

Back

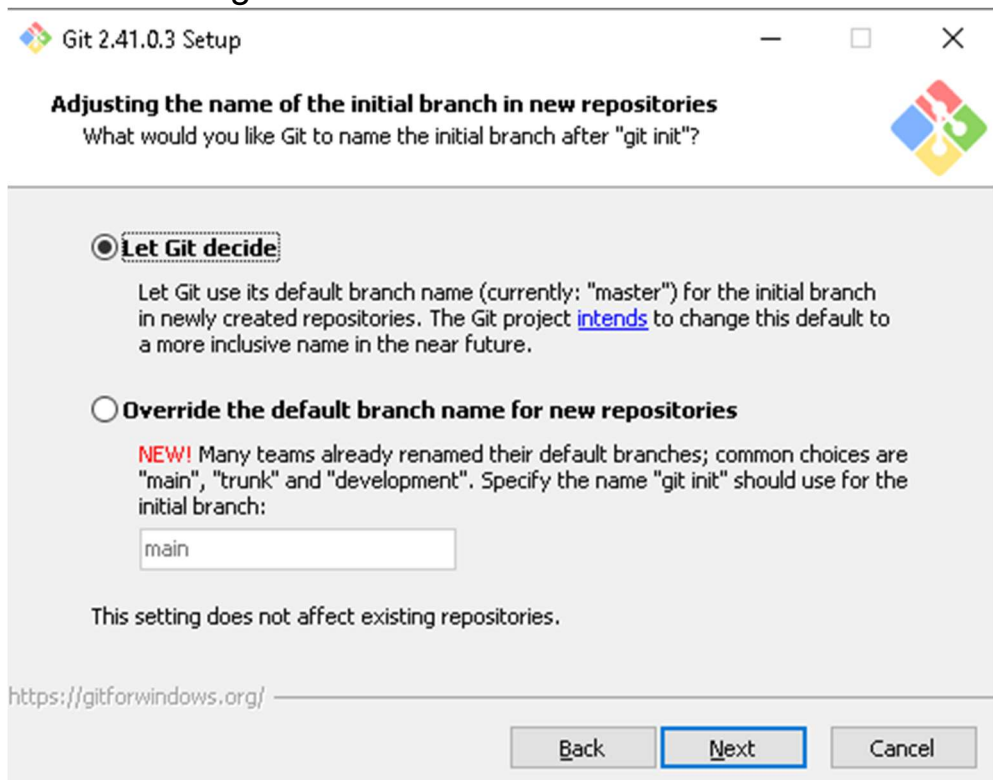
Next

Cancel

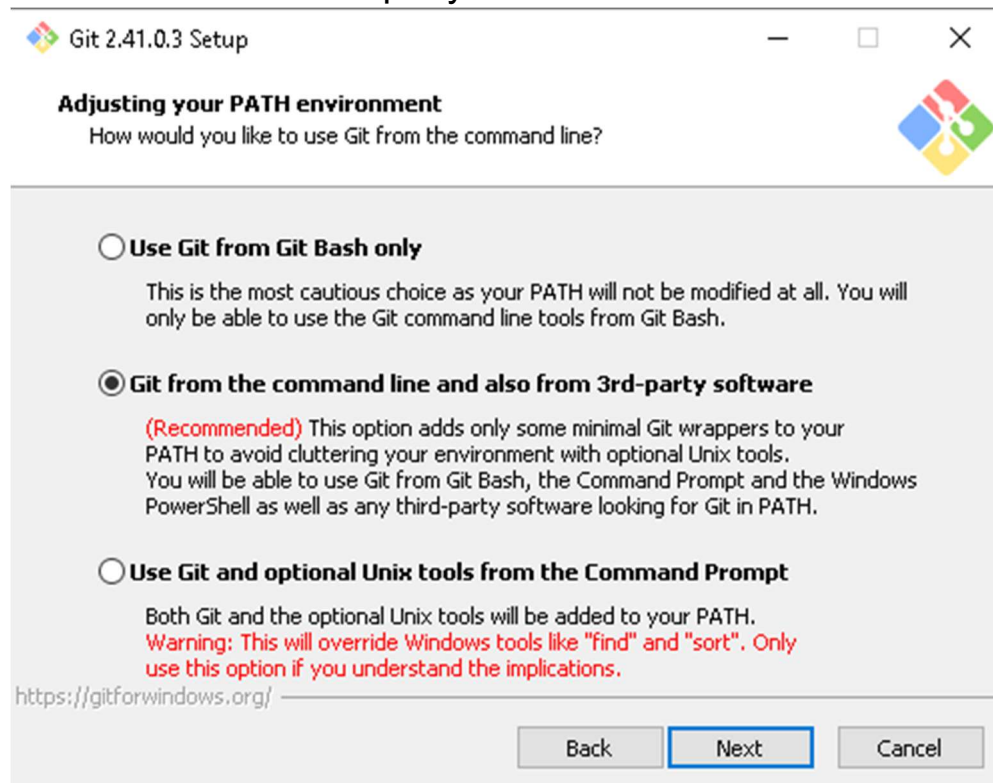
Pilih "Next" lagi.



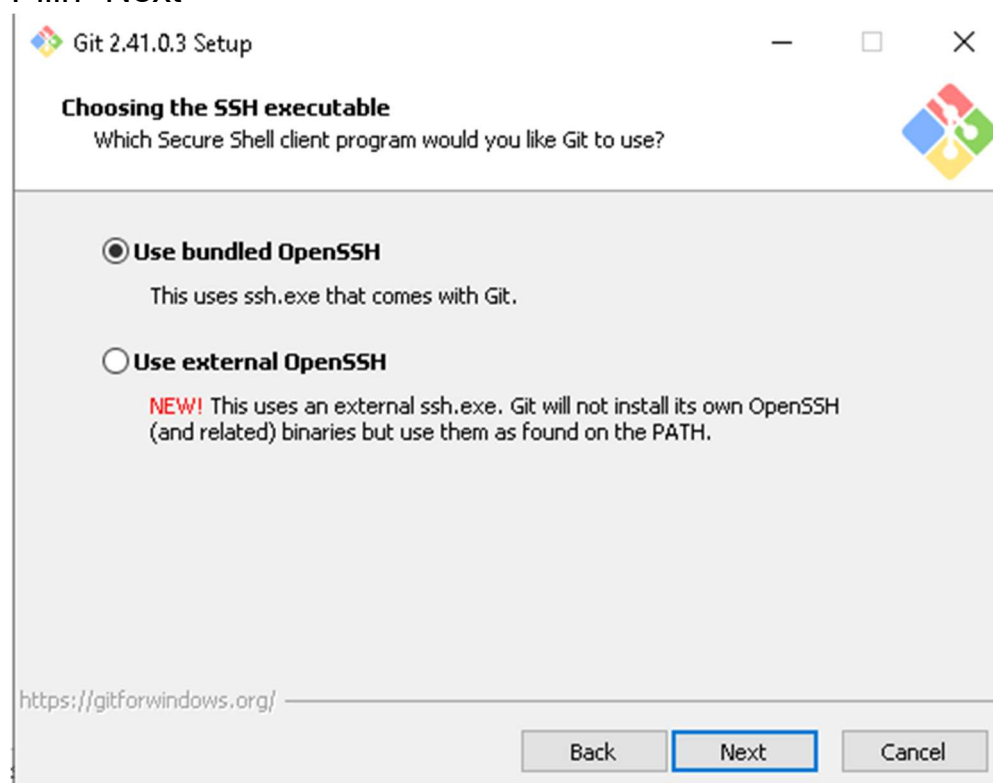
Pilih "Next" lagi.



Pilih “Next” dan pastikan kalian mengecek Git from the command line and also from 3rd-party-software



Pilih “Next”



Pilih “Next”

Choosing HTTPS transport backend

Which SSL/TLS library would you like Git to use for HTTPS connections?

☒ **Use the OpenSSL library**

Server certificates will be validated using the ca-bundle.crt file.

☐ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

<https://gitforwindows.org/>

Back

Next

Cancel

Pilih "Next" lagi.

Configuring the line ending conversions

How should Git treat line endings in text files?

☒ **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

☐ **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

☐ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

<https://gitforwindows.org/>

Back

Next

Cancel

Pilih "Next"

Configuring the terminal emulator to use with Git Bash

Which terminal emulator do you want to use with your Git Bash?

☒ **Use MinTTY (the default terminal of MSYS2)**

Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `wintpy` to work in MinTTY.

☐ **Use Windows' default console window**

Git will use the default console window of Windows ("`cmd.exe`"), which works well with Win32 console programs such as interactive Python or `node.js`, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

<https://gitforwindows.org/>

Back

Next

Cancel

Pilih "Next" lagi.

Choose the default behavior of `git pull`What should `git pull` do by default?☒ **Default (fast-forward or merge)**

This is the standard behavior of `git pull`: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.

☐ **Rebase**

Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

☐ **Only ever fast-forward**

Fast-forward to the fetched branch. Fail if that is not possible.

<https://gitforwindows.org/>

Back

Next

Cancel

Pilih "Next" lagi.

Choose a credential helper

Which credential helper should be configured?



☒ **Git Credential Manager**

Use the [cross-platform Git Credential Manager](#).

See more information about the future of Git Credential Manager [here](#).

☐ **None**

Do not use a credential helper.

<https://gitforwindows.org/>

Back

Next

Cancel

Pilih "Next" lagi.

Configuring extra options

Which features would you like to enable?



☒ **Enable file system caching**

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

☐ **Enable symbolic links**

Enable [symbolic links](#) (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

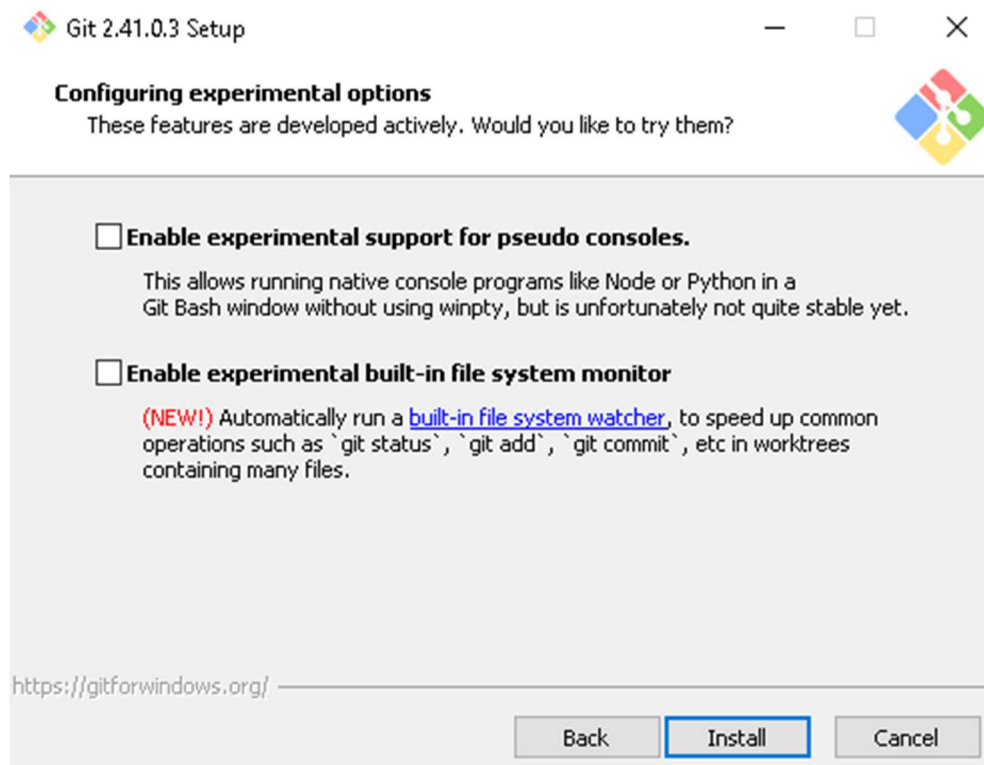
<https://gitforwindows.org/>

Back

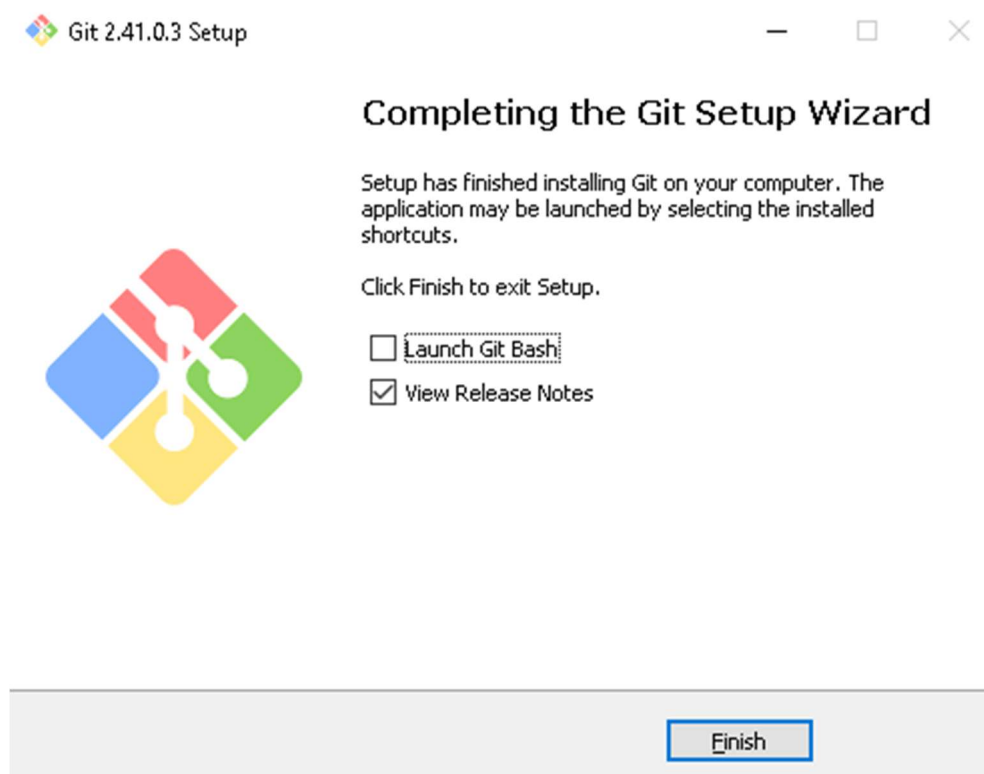
Next

Cancel

Kemudian diakhiri dengan mengklik “Install”



5. Jika sudah selesai, kalian uncek saja “View Release Notes” dan cek “Launch Git Bash”



6. Jika sudah maka kalian sudah berhasil menginstall Git Bash.
Kalian bisa langsung menggunakannya dengan mengetik “git init” atau “git clone”. Jika ingin tau lebih banyak command, kalian bisa mengetik “git help”. Kalian juga bisa mengupdate Git Bash dengan mengetik “git update-git-for-windows”.

Beberapa command umum dari Git Bash:

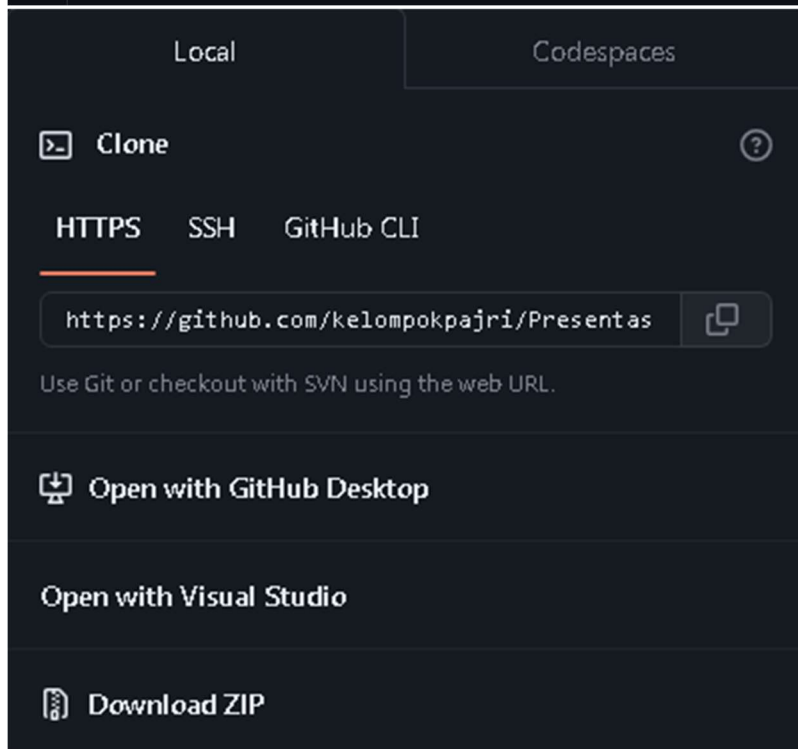
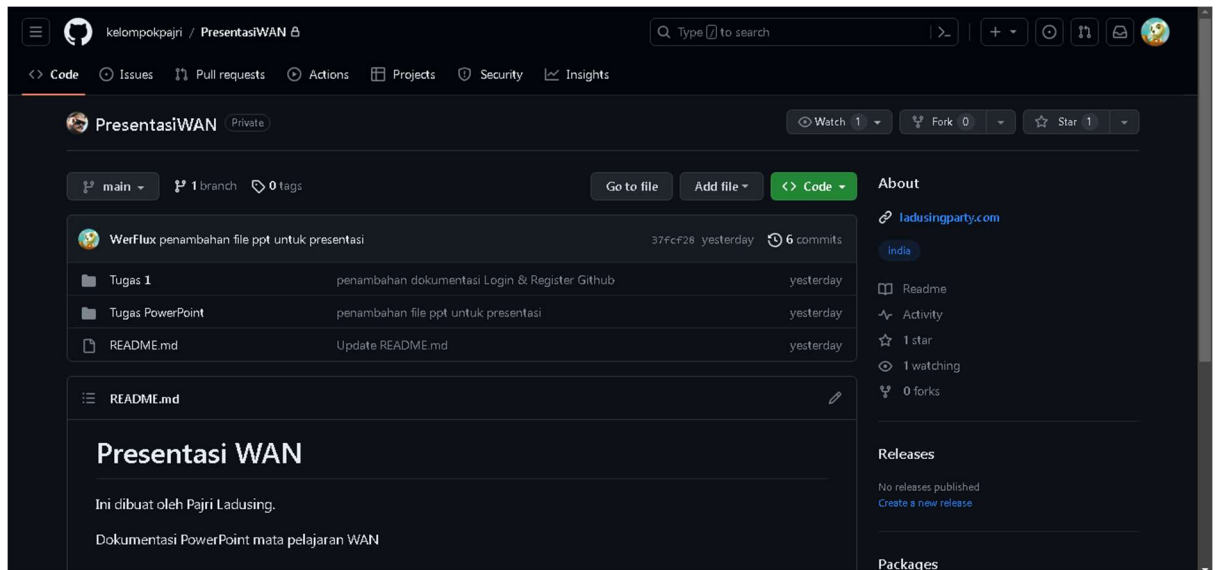
1. git init
2. git clone
3. git push
4. git commit
5. git fetch
6. git pull
7. git branch

Contoh untuk melakukan git clone pada repository dan melakukan commit push:

1. git clone <https://github.com/kelompokpajri/PresentasiWAN.git>
2. git add .
3. git commit -m 'pesan commit'
4. git push

Contoh Gambar:

Kalian bisa mengambil link repository disini melalui tombol “Code” lalu disini.



Contoh Git Bashnya:

```
MINGW64/c/Users/Bintang/Desktop/Projects/Tugas/temp/PresentasiWAN
Bintang@DESKTOP-NFG80GK MINGW64 ~/Desktop/Projects/Tugas/temp
$ git clone https://github.com/kelompokpajri/PresentasiWAN.git
Cloning into 'PresentasiWAN'...
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 20 (delta 1), reused 7 (delta 0), pack-reused 0
Receiving objects: 100% (20/20), 1.68 MiB | 1.12 MiB/s, done.
Resolving deltas: 100% (1/1), done.

Bintang@DESKTOP-NFG80GK MINGW64 ~/Desktop/Projects/Tugas/temp
$ cd PresentasiWAN
Bintang@DESKTOP-NFG80GK MINGW64 ~/Desktop/Projects/Tugas/temp/PresentasiWAN (main)
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 529.58 KiB | 3.11 MiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:kelompokpajri/PresentasiWAN.git
   37fcf28..2657597  main -> main

Bintang@DESKTOP-NFG80GK MINGW64 ~/Desktop/Projects/Tugas/temp/PresentasiWAN (main)
$ |
```

Berikut penjelasannya:

1. git clone command berfungsi untuk melakukan kloning, atau ibaratnya kita melakukan download repository tersebut kedalam penyimpanan kita.
2. git add command berfungsi untuk menambahkan file kedalam repository, titik (.) tersebut menandakan bahwa kita ingin menambahkan seluruh file kedalam repository. Pastikan ada file yang dirubah dalam penyimpanan repository sebelum menggunakan git add.
3. git commit -m command berfungsi untuk membuat sebuah commit messages kedalam repository kita.
4. git push command berfungsi untuk memasukkan repository dalam penyimpanan kita kedalam Github.

Jika sudah, silahkan cek repository kalian di Github.

The screenshot shows a GitHub repository page for 'PresentasiWAN' by user 'kelompokpajri'. The repository is private and has 1 watch, 0 forks, and 1 star. The main branch is 'main' with 1 branch and 0 tags. The repository contains a file named 'README.md' which is displayed in the main content area. The README content includes the title 'Presentasi WAN' and a description: 'Ini dibuat oleh Pajri Ladusing.' and 'Dokumentasi PowerPoint mata pelajaran WAN'. The commit history shows a commit by 'WerFlux' titled 'Penambahan dokumentasi Git dalam bentuk Word' made 1 minute ago, with 7 commits in total. The right sidebar shows the repository's about section, including the website 'ladusingparty.com', location 'India', and statistics: 1 star, 1 watching, and 0 forks. The bottom of the screenshot shows the commit details for the selected commit, including the commit message and the commit hash '2657597'.

kelompokpajri / PresentasiWAN

Code Issues Pull requests Actions Projects Security Insights

PresentasiWAN Private

main 1 branch 0 tags

Go to file Add file Code

WerFlux Penambahan dokumentasi Git dalam bentuk Word 2657597 1 minute ago 7 commits

File	Commit Message	Time
Tugas 1	Penambahan dokumentasi Git dalam bentuk Word	1 minute ago
Tugas PowerPoint	penambahan file ppt untuk presentasi	yesterday
README.md	Update README.md	yesterday

README.md

Presentasi WAN

Ini dibuat oleh Pajri Ladusing.

Dokumentasi PowerPoint mata pelajaran WAN

About

ladusingparty.com

India

Readme

Activity

1 star

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

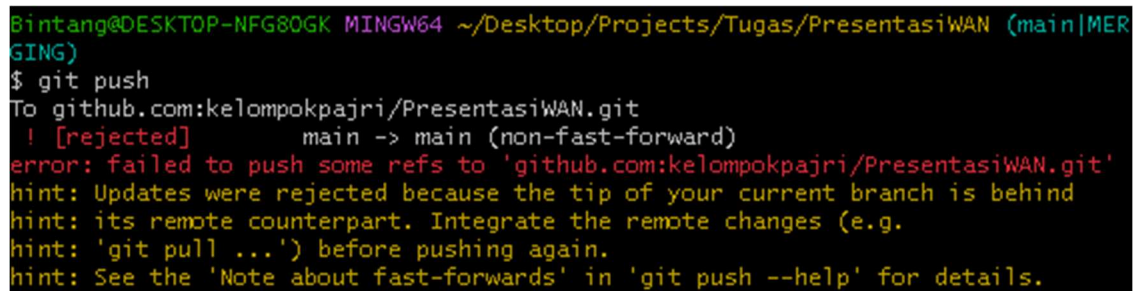
WerFlux Penambahan dokumentasi Git dalam bentuk Word 2657597 1 minute ago 7 commits

Jika ada teks “1 minutes ago” atau “1 menit yang lalu” itu tandanya kamu telah berhasil.

Jika kamu memiliki 2 orang yang bekerja dalam repository tersebut. Sebagai contoh:

1. Developer 1 melakukan git clone
 2. Developer 2 melakukan git clone
- Developer 2 melakukan git push dan Developer 1 jika melakukan git push secara bersamaan.

Ini akan menjadi tabrakan (conflict) bisa dilihat dari gambar berikut:



```
Bintang@DESKTOP-NFG8QGK MINGW64 ~/Desktop/Projects/Tugas/PresentasiWAN (main|MERGING)
$ git push
To github.com:kelompokpajri/PresentasiWAN.git
 ! [rejected]        main -> main (non-fast-forward)
error: failed to push some refs to 'github.com:kelompokpajri/PresentasiWAN.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Developer 1 akan melihat sama seperti gambar diatas. Bisa dilihat dari “hint” mengatakan untuk melakukan git pull terlebih dahulu sebelum menggunakan git push.