



UPN "VETERAN" JAKARTA
FAKULTAS ILMU KOMPUTER

MODUL PELATIHAN

Collaborative Coding: GIT Untuk Meningkatkan Keterampilan dan Kolaborasi Siswa Kejuruan



Pondok Labu, Jakarta



(021) 7656971



upnvj@upnvj.ac.id

new-fik.upnvj.ac.id

Penyusun
Tim Abdimas PKM-T FIK UPNVJ

Collaborative Coding : GIT

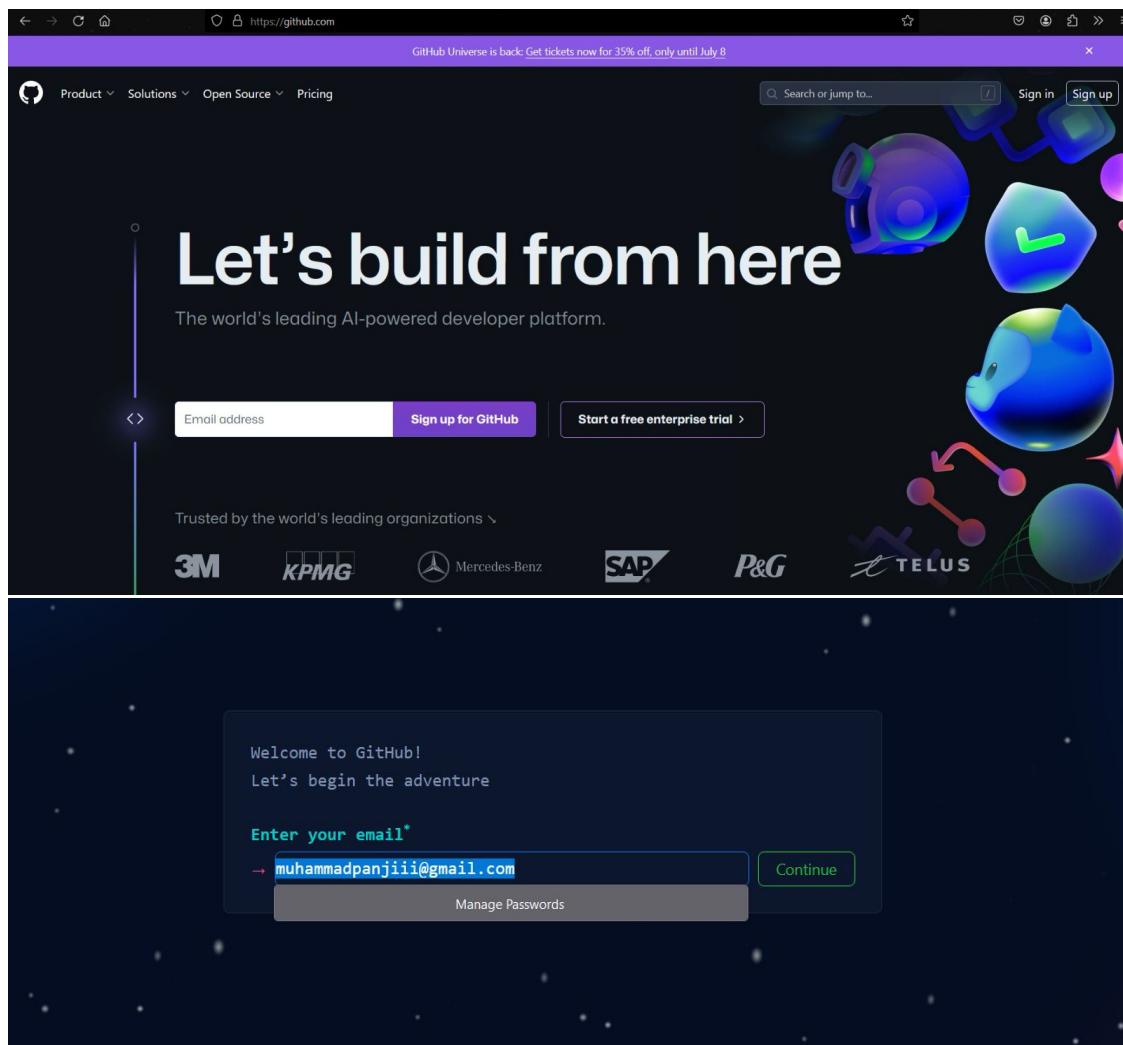
Kompetensi:

Siswa diharapkan mampu

- Mengimplementasikan *Collaborative Coding* untuk mengerjakan tugas besar atau kelompok
- Menerapkan Perintah Eksekusi GIT dalam implementasi di Github

A. Membuat Akun Github

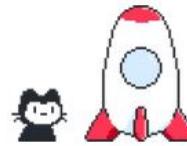
1. Membuat akun github dengan membuat situs web GitHub pada <https://github.com/>. Klik Tombol “Sign Up” dan ikuti panduan untuk pembuatan akun baru dengan email personal.



2. Jika sudah berhasil, silahkan rekan rekan untuk memverifikasi dan di email yang di daftarkan.



Here's your GitHub launch code,
@muhammadpanjimuslim!



Continue signing up for GitHub by entering the code below:



Open GitHub

3. Setelah pembuatan akun selesai silahkan ke halaman Awal GitHub Kembali.
4. Dan Akun siap untuk dicoba dengan klik “Sign In”.

B. Instalasi Aplikasi Git

Jika anda belum memiliki GIT di komputer rekan rekan, unduh dan instal GIT dari <https://git-scm.com/>. Ikuti panduan instalasi yang sesuai dengan sistem operasi rekan rekan.

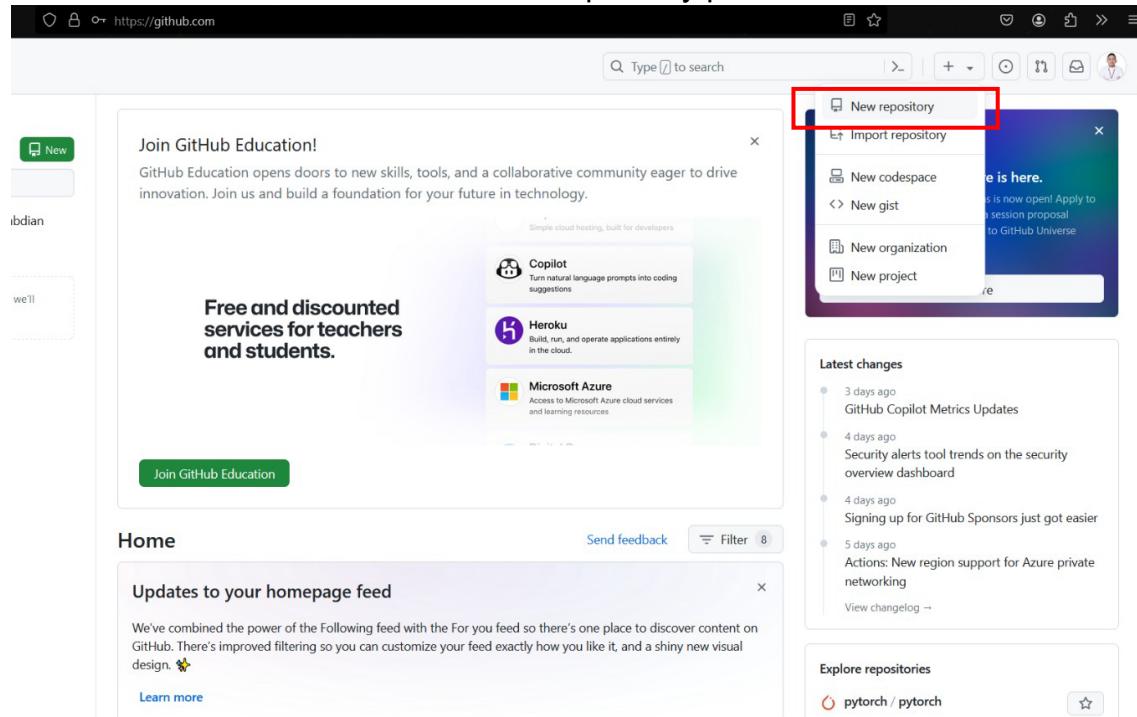
C. Github

1. Lakukan Sign In Akun yang telah di buat
2. Jika berhasil akan muncul tampilan seperti berikut:

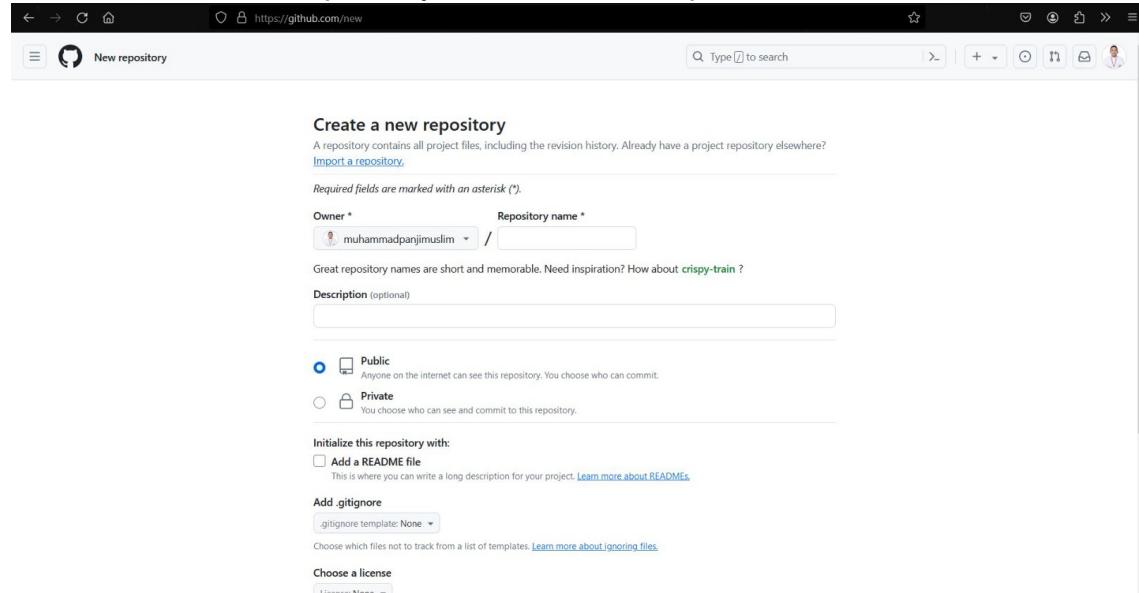
The screenshot shows the GitHub Home page with several overlays. At the top right is a dark blue banner for "UNIVERSE24" with the text "The world's fair of software is here." and a "Learn more" button. Below it is a "Latest changes" sidebar with items from 3 days ago to 5 days ago. The main content area has a "Join GitHub Education!" overlay with a green "Join GitHub Education" button. To the left is a "Top Repositories" sidebar with a search bar and a repository for "muhammadpanjimuslim/pengabdian". The bottom left shows a "Recent activity" sidebar with a message about providing links to activity here. The bottom right shows an "Updates to your homepage feed" sidebar with a message about combining the power of the Following and For you feeds.

Repository pada GitHub adalah “tempat penyimpanan untuk proyek perangkat lunak yang menggunakan sistem kontrol versi GIT. Repository ini dapat berisi berbagai file, seperti kode program, gambar, dokumen, dll. Repository ini memungkinkan para pengembang untuk bekerja secara kolaboratif, melacak perubahan yang terjadi pada proyek dan mengelola versi dari proyek tersebut”.

3. Buatlah Repository baru pada GIT, dengan klik tombol “+” atau New di sudut kanan atas halaman dan klik New Repository pada GitHub.

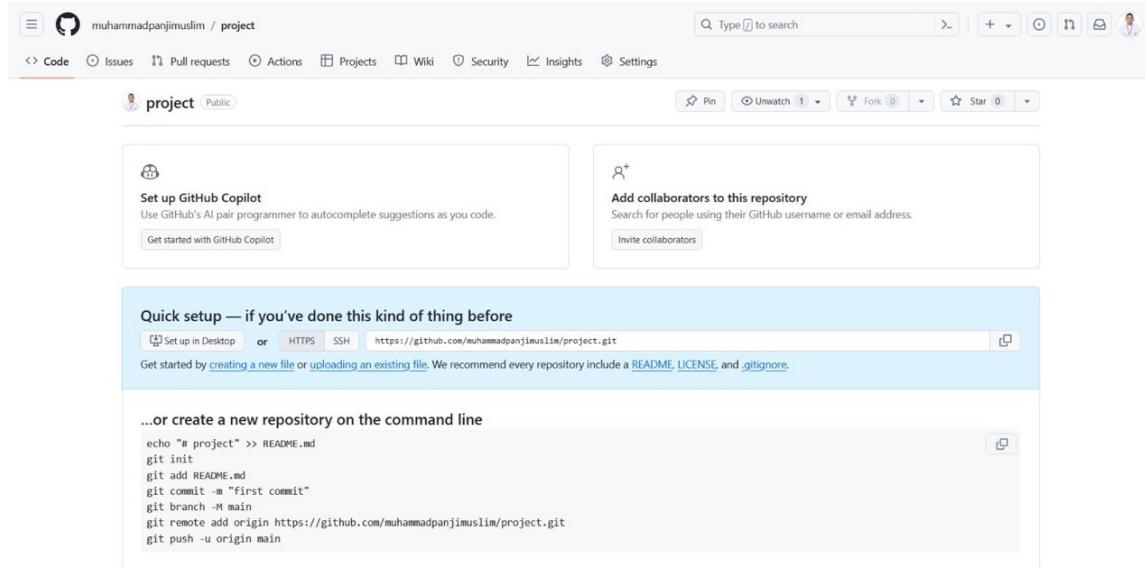


4. Jika telah klik New Repository akan muncul tampilan berikut:



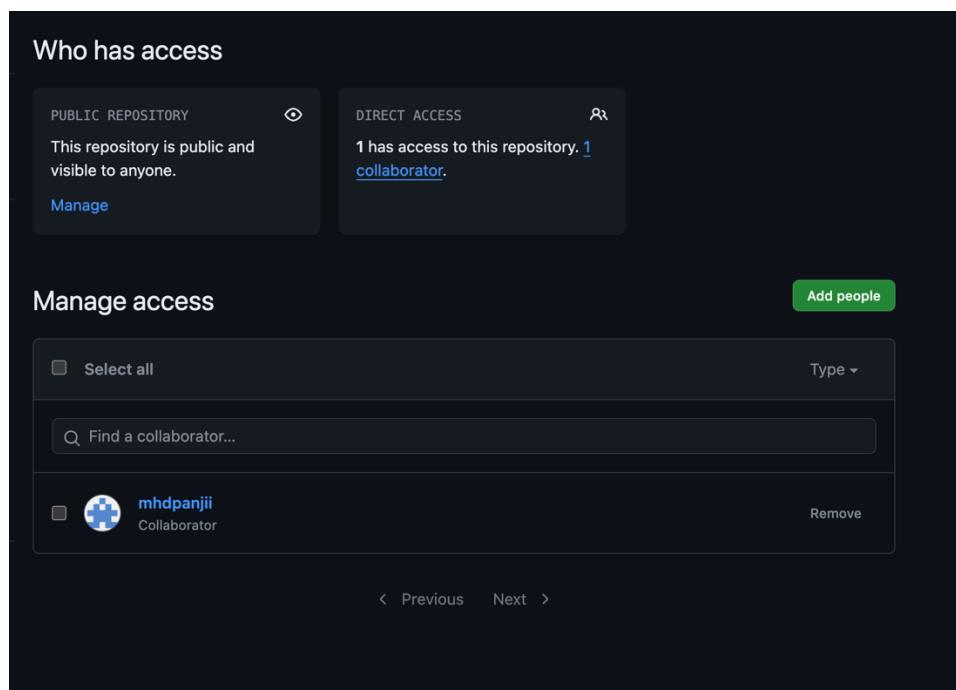
5. Berikan nama untuk repository rekan-rekan sesuai dengan nama projek atau tugasnya.

6. Pilih apakah repository akan menjadi public (visible to everyone) atau pribadi (visible to you and collaborators only).
7. Jika diperlukan, tambahkan deskripsi untuk repository.
8. Lalu Klik Tombol “Create Repository”.
9. Lalu akan muncul tampilan sebagai berikut:



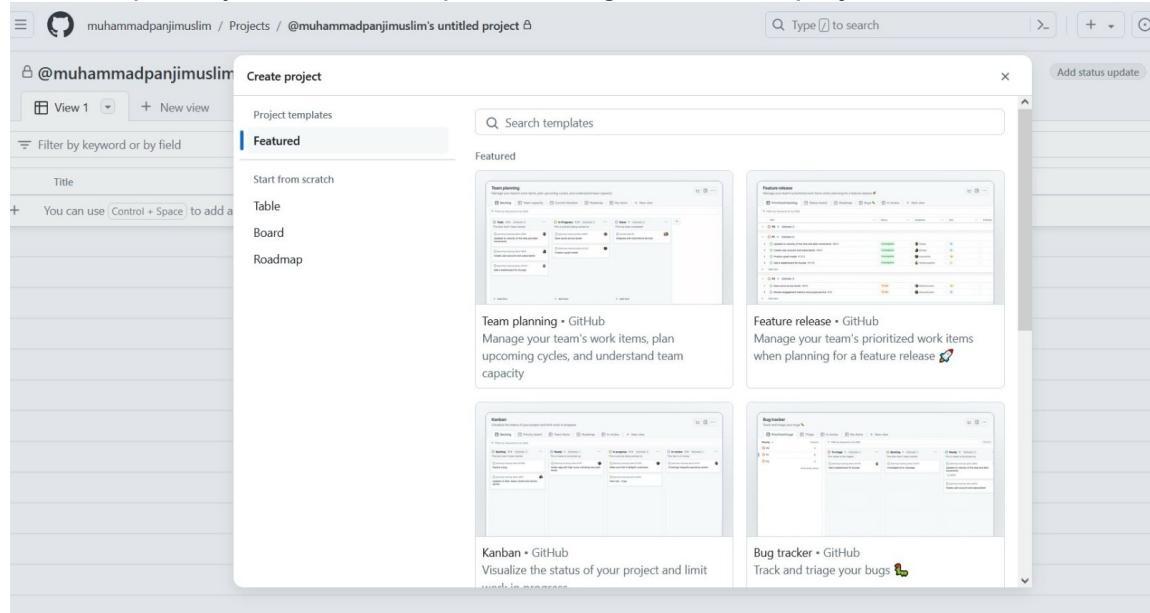
D. Menambahkan Anggota (Collaborator)

1. Buka repository yang baru rekan rekan buat
2. Klik Tab “Setting” di bagian atas kanan layer
3. Di panel sebelah kiri, pilih “Manage Access”
4. Klik tombol “Invite a collaborator”
5. Ketik username atau alamat email dari orang yang ingin Anda tambahkan sebagai collaborator
6. Pilih nama pengguna atau alamat email yang sesuai dari daftar yang muncul
7. Klik tombol “Add Collaborator”

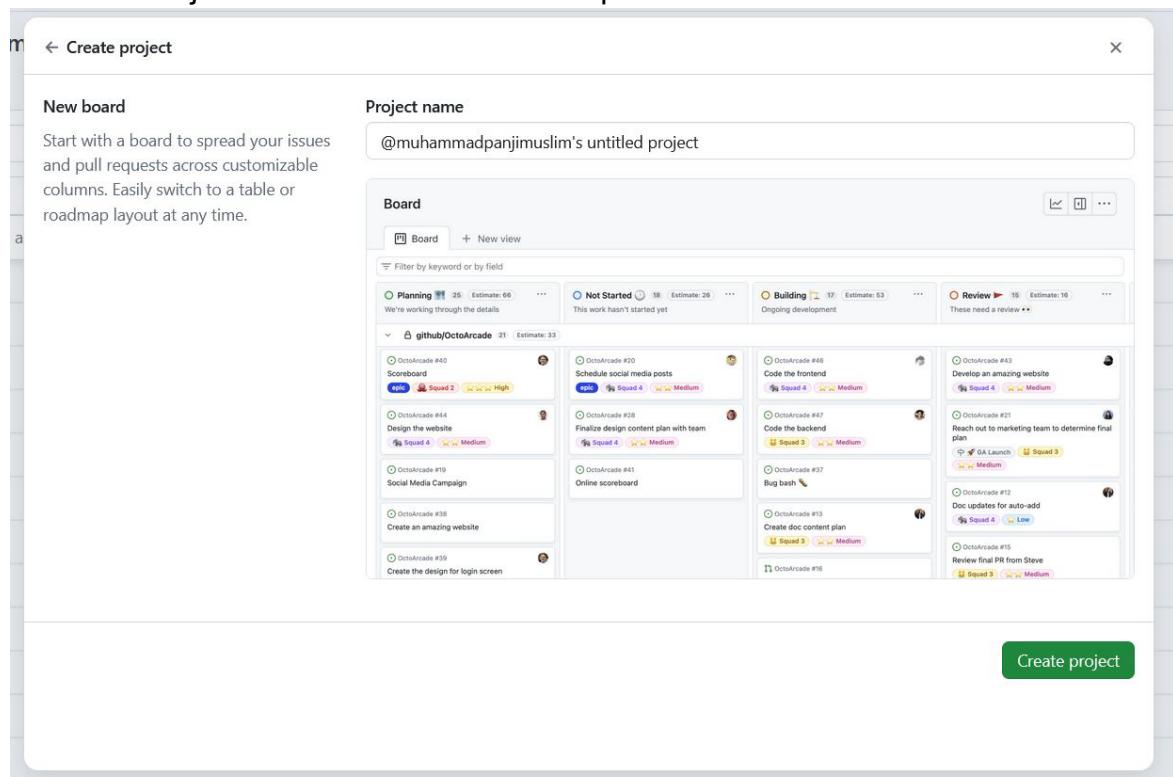


E. Membuat Project Board di GitHub dan Anggota

1. Buka Repository di GitHub tempat anda ingin membuat project board.



2. Pilih Tab "Projects". Di atas daftar file dalam repository, ada beberapa tab, seperti "Code", "Issues", "Pull request", dll.
3. Klik Tab "Project". Maka akan muncul tampilan berikut:



4. Buat Project Board Baru. Pada halaman "Projects", klik tombol "Create a project" (jika belum ada project board) atau "New Project" (Jika sudah ada project board sebelumnya)
5. Pilih jenis project board yang anda inginkan. GitHub menyediakan beberapa template seperti "Basic Kanban", "Automated Kanban", atau "Automated Kanban with Review"

6. Berikan nama untuk project board rekan rekan dan klik tombol Create Project
7. Kelola Proyek, Kita dapat menyeret dan melepaskan catatan (cards) antar kolom untuk menunjukkan status kemajuan.
8. Klik pada catatan untuk melihat atau mengedit detailnya.
9. Gunakan fitur komentar untuk berkomunikasi dengan tim rekan rekan didalam catatan. Dan menambahkan collaborator seperti tampilan berikut:

The screenshot shows the 'Manage access' section of a GitHub repository settings page. It includes sections for 'Who has access' (private project), 'Invite collaborators' (with a search bar for 'mhdpanjji'), and 'Manage access' (showing 0 members). A message at the bottom states 'You don't have any collaborators yet. Add a collaborator to see them here.'

F. Mengunggah Proyek ke Repository

1. Buka terminal/commad prompt atau Git Bash Terminal yang telah diinstall
2. Lalu ikuti command yang ada pada tampilan new repository yang telah di buat seperti berikut:

...or create a new repository on the command line

```
echo "# project" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/muhammadpanjimuslim/project.git
git push -u origin main
```

//Inisialisasi repository Git Lokal jika belum dilakukan

git init

//Tambahkan File Proyek ke repository local

git add .

//Lakukan Komit (commit) perubahan ke repository local

git commit -m "Initial commit"

//Tautkan repository lokal dengan repository di GitHub

git remote add origin <URL_Repository_GitHub>

//Dorong (push) proyek rekan rekan ke GitHub

git push -u origin master

3. Lalu tuliskan di Git Bash nya seperti berikut:

```
Last login: Sun May 12 20:03:58 on ttys000
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[Muhammads-MacBook-Pro:~ muhammadpanji]$ cd Public
[Muhammads-MacBook-Pro:Public muhammadpanji]$ cd Experiment
[Muhammads-MacBook-Pro:Experiment muhammadpanji]$ cd ExperimentC
[Muhammads-MacBook-Pro:ExperimentC muhammadpanji]$ cd Git
[Muhammads-MacBook-Pro:Git muhammadpanji]$ git init
Reinitialized existing Git repository in /Users/muhammadpanji/Public/Experiment/ExperimentC/Git/.git/
[Muhammads-MacBook-Pro:Git muhammadpanji]$ git add .
[Muhammads-MacBook-Pro:Git muhammadpanji]$ git commit -m "first commit"
On branch main
nothing to commit, working tree clean
[Muhammads-MacBook-Pro:Git muhammadpanji]$ git branch -M main
[Muhammads-MacBook-Pro:Git muhammadpanji]$ git remote add origin https://github.com/muhammadpanjimuslim/project.git
fatal: remote origin already exists.
[Muhammads-MacBook-Pro:Git muhammadpanji]$ git remote 0v
error: Unknown subcommand: 0v
usage: git remote [-v | --verbose]
   or: git remote add [-t <branch>] [-m <master>] [-f] [--tags | --no-tags] [--mirror=<fetch|push>] <name> <url>
   or: git remote rename <old> <new>
   or: git remote remove <name>
   or: git remote set-head <name> (-a | --auto | -d | --delete | <branch>)
   or: git remote [-v | --verbose] show [-n] <name>
   or: git remote prune [-n | --dry-run] <name>
   or: git remote [-v | --verbose] update [-p | --prune] [(<group> | <remote>)...]
   or: git remote set-branches [--add] <name> <branch>...
   or: git remote get-url [--push] [--all] <name>
   or: git remote set-url [--push] <name> <newurl> [<oldurl>]
   or: git remote set-url --add <name> <newurl>
   or: git remote set-url --delete <name> <url>

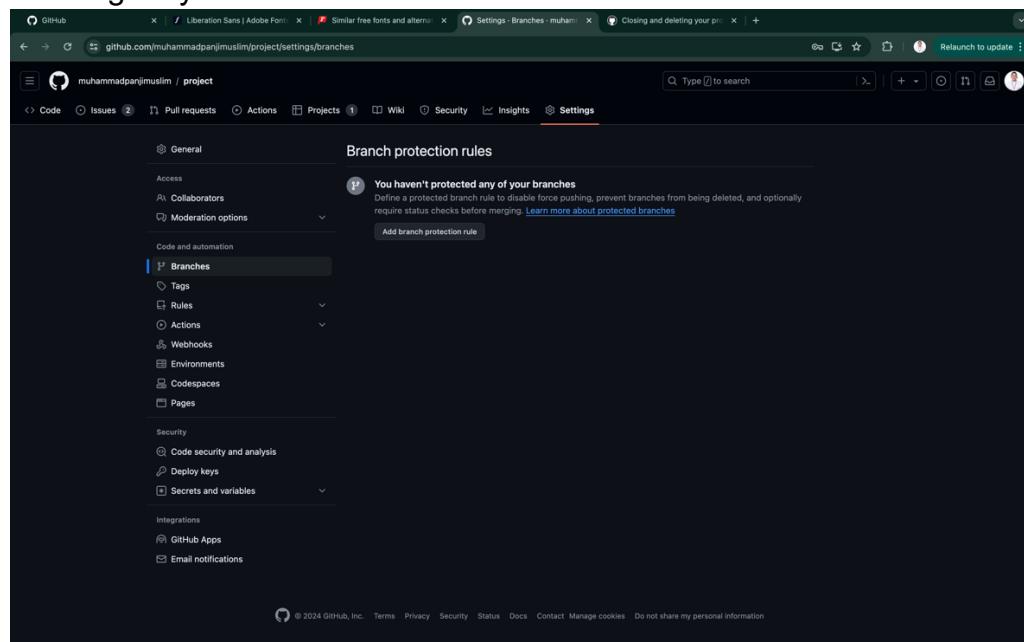
-v, --verbose      be verbose; must be placed before a subcommand

[Muhammads-MacBook-Pro:Git muhammadpanji]$ git remote -v
origin https://github.com/muhammadpanjimuslim/project.git (fetch)
origin https://github.com/muhammadpanjimuslim/project.git (push)
[Muhammads-MacBook-Pro:Git muhammadpanji]$ git push -u origin main
Username for 'https://github.com': muhammadpanjimuslim
Password for 'https://muhammadpanjimuslim@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 256 bytes | 256.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/muhammadpanjimuslim/project.git
 * [new branch]    main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
[Muhammads-MacBook-Pro:Git muhammadpanji$ ]
```

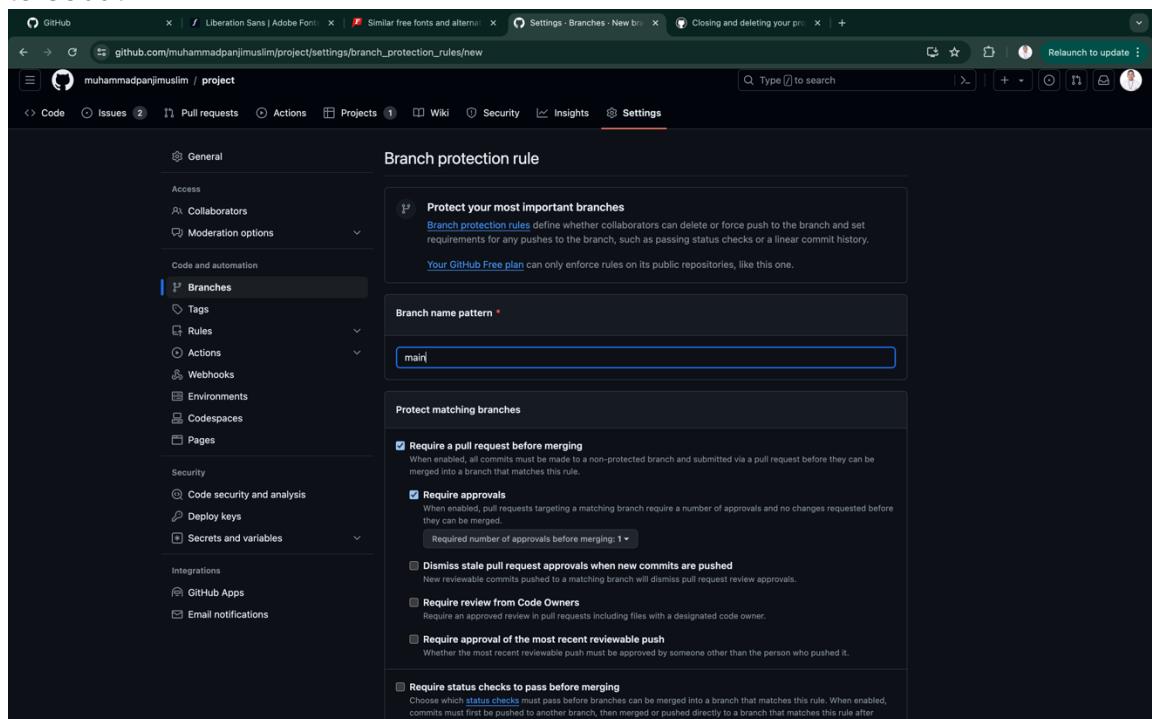
4. Jika sudah selesai maka Proyek telah ada di Repository GitHub .

G. Membuat Pengaturan Branch Protection Rule

1. Buka Repository, Buka repository di GitHub yang ingin rekan atur aturan perlindungannya.



2. Pilih Tab "**Settings**", Di atas daftar file dalam repository, ada beberapa tab, seperti "Code", "Issues", "Pull requests", dll.
3. Klik tab "Settings". Pilih "Branches" (Cabang)
4. Di panel sebelah kiri, pilih opsi "Branches" atau "Branches" (tergantung pada tata letak baru atau lama dari antarmuka pengguna GitHub).
5. Tambahkan Perlindungan untuk Cabang
6. Pada halaman "Branches", gulir ke bawah hingga rekan rekan menemukan bagian yang berjudul "Protected branches".
7. Klik tombol "Add rule" atau "Edit" di sebelah nama cabang yang ingin rekan rekan lindungi.
8. Pilih nama cabang yang ingin rekan lindungi dari daftar yang tersedia. Aktifkan opsi "Protect this branch" untuk mengaktifkan perlindungan untuk cabang tersebut.



9. Atur Aturan Perlindungan, Pilih opsi yang ingin rekan terapkan untuk cabang tersebut, seperti: "**Require pull request reviews before merging**": Memastikan bahwa setidaknya satu tinjauan telah dilakukan sebelum penggabungan perubahan ke cabang tersebut. "**Require status checks to pass before merging**": Memastikan bahwa semua pemeriksaan status yang ditentukan (seperti pengujian otomatis) telah berhasil sebelum penggabungan perubahan. "**Include administrators**": Menyertakan administrator repository dalam aturan perlindungan.
10. Rekan juga dapat menambahkan pemeriksaan status kustom atau pemeriksaan status cabang yang diperlukan.
11. Simpan Perubahan, Klik tombol "Save changes" atau "Save" untuk menyimpan aturan perlindungan yang baru saja Anda atur.

H. Clone Repository

1. Collaborator perlu mengkloning (clone) repository ke komputer lokalnya menggunakan perintah ‘**git clone URL_Repository**’

```
Muhammad Panji@DESKTOP-046FQIC MINGW64 /d
$ git clone https://github.com/muhammadpanjimuslim/project.git
Cloning into 'project'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

I. Buat Cabang (Branch)

1. Collaborator dapat membuat cabang baru untuk mengerjakan fitur atau perbaikan bug tertentu. Dengan perintah “**git checkout -b nama_cabang**”

```
Muhammad Panji@DESKTOP-046FQIC MINGW64 /d
$ cd Project

Muhammad Panji@DESKTOP-046FQIC MINGW64 /d/Project (main)
$ git branch selamat_datang

Muhammad Panji@DESKTOP-046FQIC MINGW64 /d/Project (main)
$ git checkout selamat_datang
Switched to branch 'selamat_datang'
```

J. Lakukan Perubahan Proyek

1. Collaborator melakukan perubahan pada proyek di komputer lokalnya menggunakan editor atau IDE.

K. Tambahkan dan Komit Perubahan

1. Collaborator menambahkan perubahan ke area staging menggunakan “**git add .**”
2. Kemudian, commit perubahan ke branch local menggunakan “**git commit -m “Deskripsi Perubahan”**”

```
Muhammad Panji@DESKTOP-046FQIC MINGW64 /d/Project (selamat_datang)
$ git add .

Muhammad Panji@DESKTOP-046FQIC MINGW64 /d/Project (selamat_datang)
$ git commit -m "tambah ucapan selamat datang"
[selamat_datang 505114e] tambah ucapan selamat datang
 1 file changed, 1 insertion(+)
```

L. Dorong perubahan ke Repository Pusat

1. Collaborator mendorong (push) perubahan ke repository di GitHub.



```
Muhammad Panji@DESKTOP-046FQIC MINGW64 /d/Project (selamat_datang)
$ git push -u origin selamat_datang
fatal: unable to connect to cache daemon: Unknown error
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 319 bytes | 319.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'selamat_datang' on GitHub by visiting:
remote:     https://github.com/muhammadpanjimuslim/project/pull/new/selamat_datang
remote:
To https://github.com/muhammadpanjimuslim/project.git
 * [new branch]      selamat_datang -> selamat_datang
branch 'selamat_datang' set up to track 'origin/selamat_datang'.
```

M. Buat Permintaan Tarik (Pull Request)

1. Collaborator membuka GitHub, navigaris ke repository, dan klik tombol “Pull Request”
2. Collaborator memilih cabang yang baru saja dikerjakan dan membuat permintaan Tarik
3. Collaborator memberikan deskripsi yang jelas tentang perubahan yang dilakukan

N. Tindak Lanjut

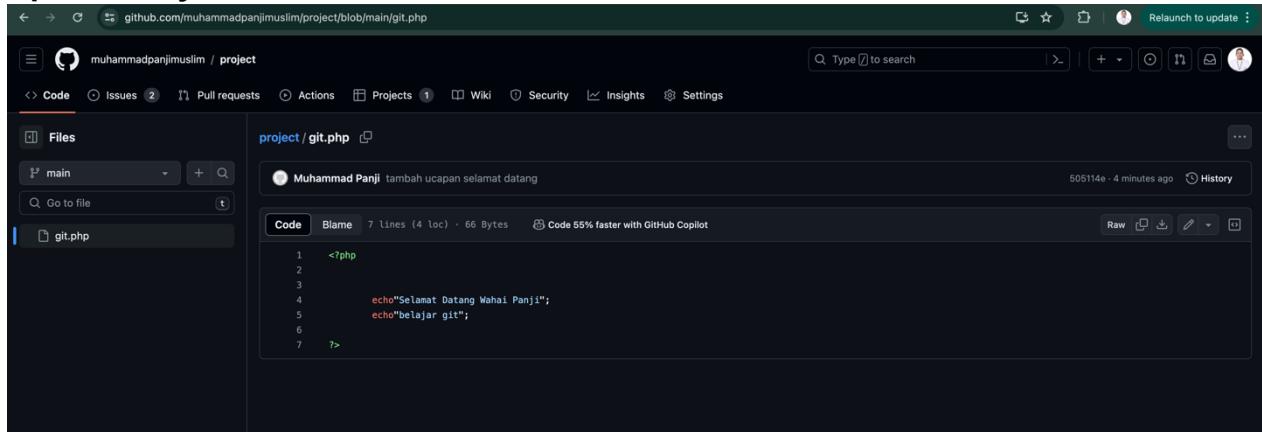
1. Pemilik repository atau reviewer akan meninjau permintaan tarik (pull request), memberikan masukan, dan melakukan diskusi jika diperlukan
2. Jika permintaan Tarik disetujui, pemilik repository akan menyatukan (merge) cabang ke cabang utama (misalnya “master”)

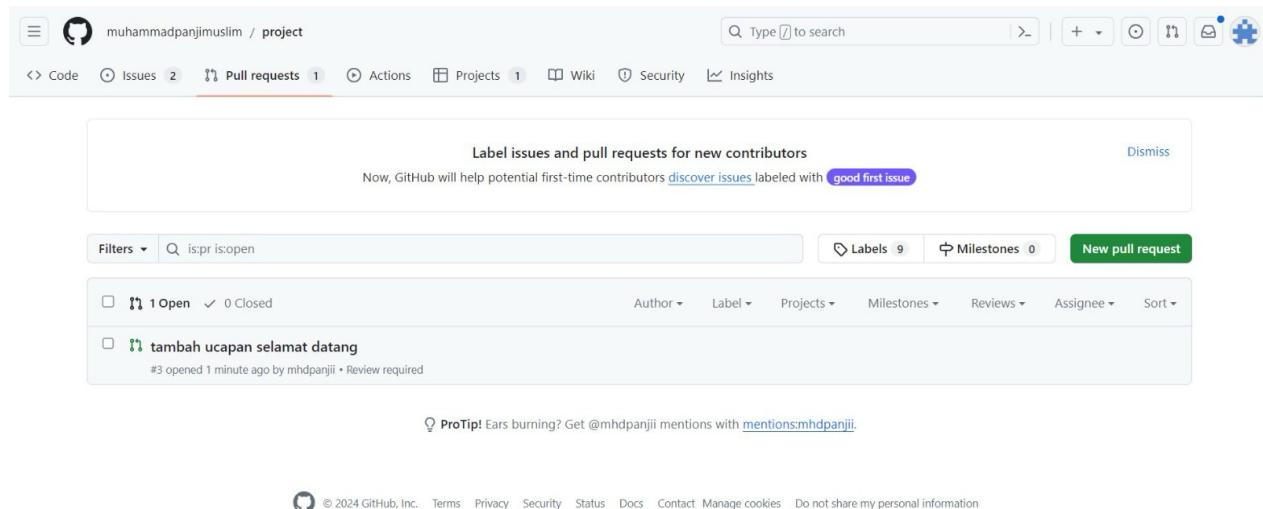
O. Sinkronisasi dengan Cabang Utama

Collaborator melakukan sinkronisasi dengan cabang utama (misalnya ‘master’) dengan menarik (pull) perubahan terbaru dari GitHub

Command : “Git pull origin master”

P. Update Proyek di GitHub





The screenshot shows a GitHub repository page for 'muhammadpanjimuslim / project'. The 'Pull requests' tab is selected, showing 1 open pull request titled 'tambah ucapan selamat datang' by 'mhdpanjii'. The pull request was opened 1 minute ago and requires review. The page includes standard GitHub navigation and search bars, and a 'ProTip!' message about mentions.

- Selamat Mencoba. -

