

Tugas Praktikum

Section 2 - Version Control (Git)


Soal Prioritas 1

- Buatlah sebuah repository github yang sesuai dengan namaKelas_namaKalian.

Create a new repository














A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner * Repository name *

 shintadeee / qe_shinta-dewi-kurniawati 

Great repository names are short and memorable. Need inspiration? How about [curly-octo-spoon?](#)

- Buatlah folder yang berurutan dan diberi nama sesuai dengan nama soal yang kalian kerjakan.

 shintadeee Merge pull request #1 from shintadeee/baru 	5f78c45 8 minutes ago	 71 commits
 10_RESTful API Testing with REST As...	Create README.md	1 hour ago
 11_Web Programming & Testing	Create README.md	1 hour ago
 12_Fundamental Performance Test ...	Create README.md	1 hour ago
 13_Mobile Testing with Serenity	Create README.md	1 hour ago
 14_Testing in CI CD Pipeline	Create README.md	1 hour ago
 1_Software Testing as a Career Path ...	Create README.md	1 hour ago
 2_Version Control (Git)	Update README.md	21 minutes ago
 3_Testing Documentation & Test Cas...	Create README.md	1 hour ago
 4_Agile Testing	Create README.md	1 hour ago
 5_Basic Linux Command	Create README.md	1 hour ago
 6_Basic Programming & OOP	Create README.md	1 hour ago
 7_Iterable & Map Data Structure	Create README.md	1 hour ago
 8_RESTful API Introduction & Testing	Create README.md	1 hour ago
 9_From Manual to Automation Testi...	Create README.md	1 hour ago

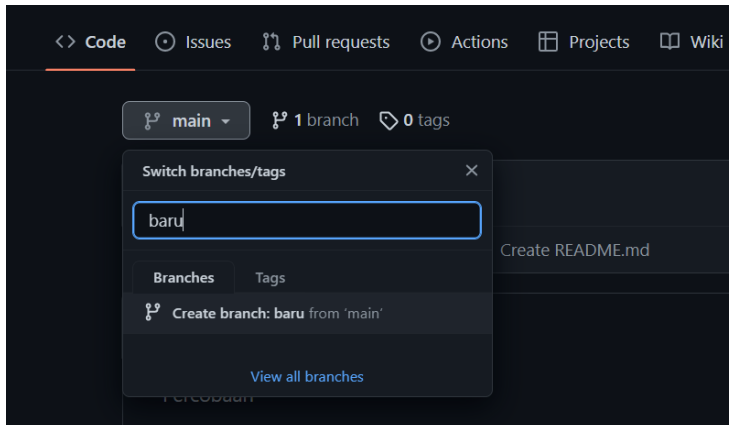
- Pada setiap folder akan berisikan sub folder lagi.

 shintadeee Update README.md	67f928c 22 minutes ago	 History
 Praktikum	Create README.md	2 hours ago
 Screenshots	Create README.md	2 hours ago
 README.md	Update README.md	22 minutes ago

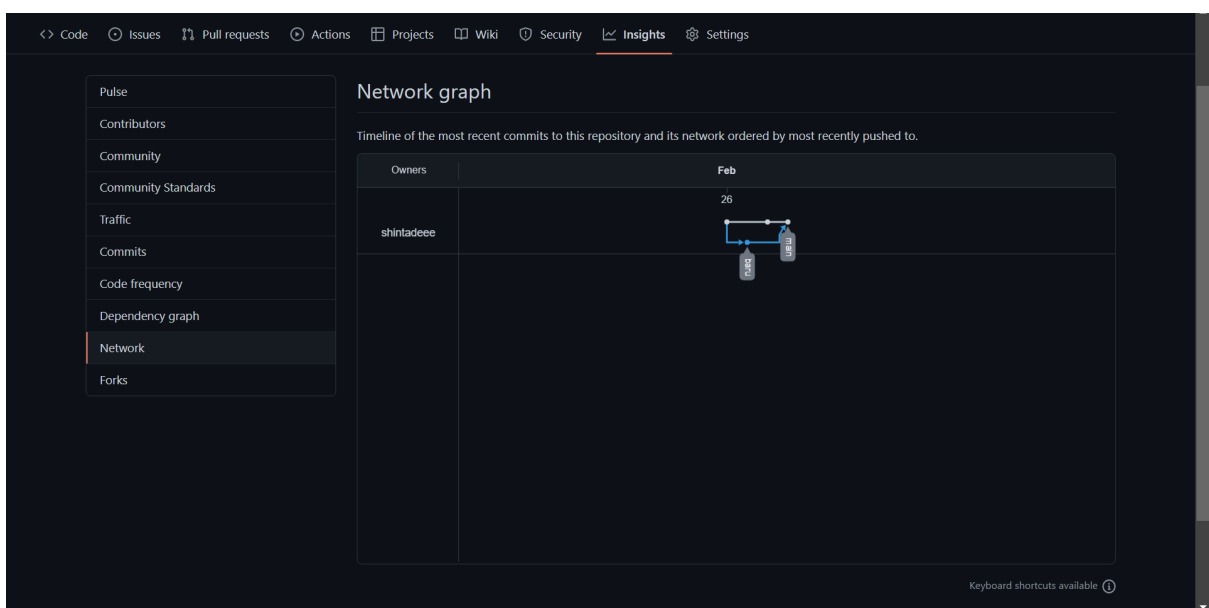
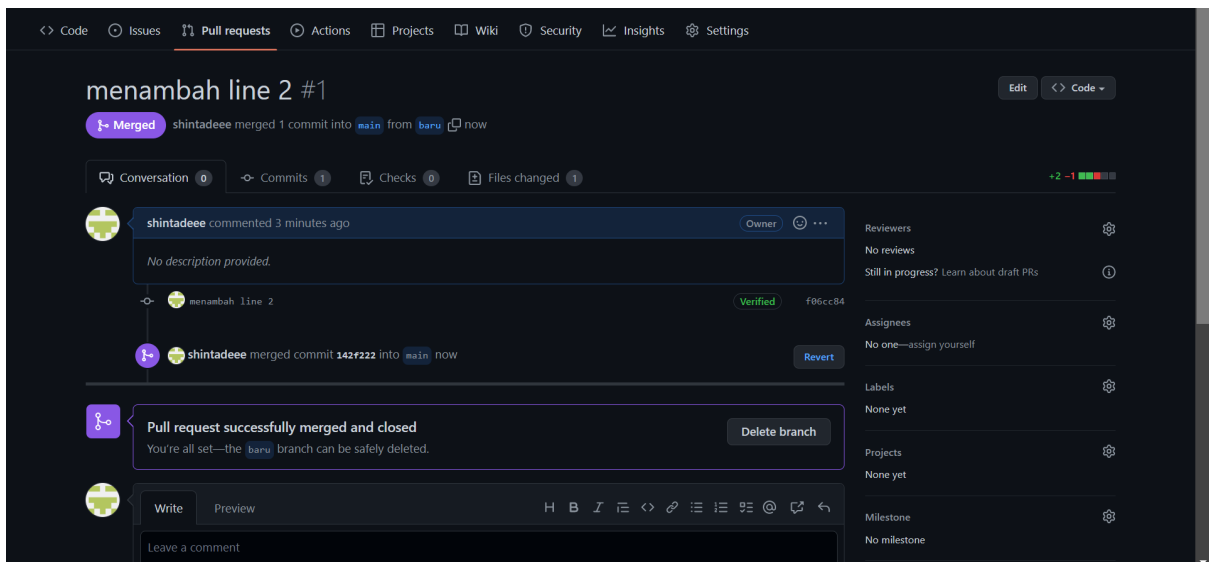
- Melakukan commit.

Soal Prioritas 2

- Buatlah branch baru pada github yang kalian buat.



- Lakukan pull request pada branch tersebut sehingga akan menambah hal baru pada branch main/master.

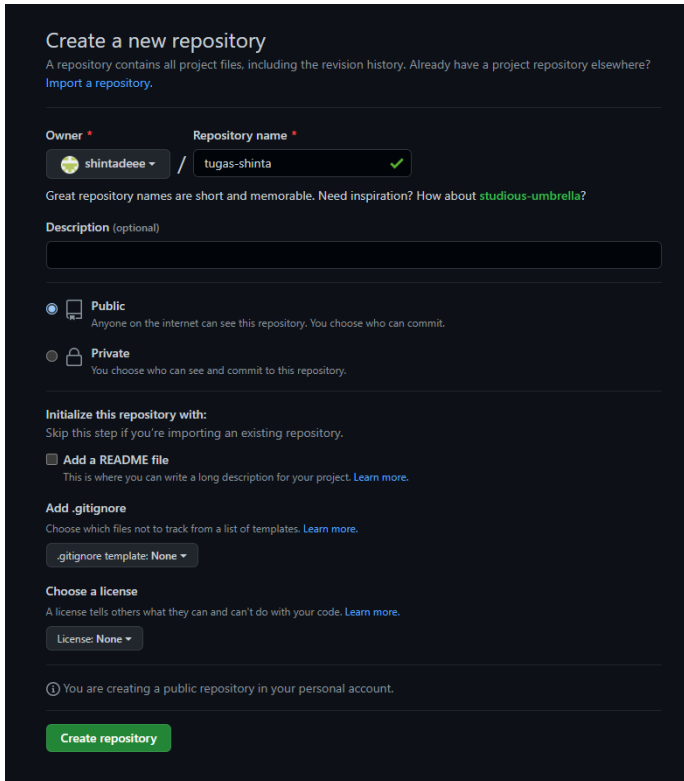


Soal Eksplorasi

- Dengan project terserah kalian, praktikkan salah satu workflow sederhana dan buktikan dengan screenshots.

Langkah-langkah untuk mempraktikkan Workflow Github Flow adalah sebagai berikut :

a. Buatlah repository di Github.



b. Clone repository tersebut ke lokal dengan menggunakan perintah git clone [url repository].

```
PS D:\Belajar\tugas-shinta> git clone https://github.com/shintadeee/tugas-shinta.git
Cloning into 'tugas-shinta'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
PS D:\Belajar\tugas-shinta>
```

c. Buat branch baru dengan menggunakan perintah git checkout -b [nama branch].

```
PS D:\Belajar\tugas-shinta> git checkout -b baru
Switched to a new branch 'baru'
PS D:\Belajar\tugas-shinta>
```

d. Lakukan perubahan kode pada branch tersebut.

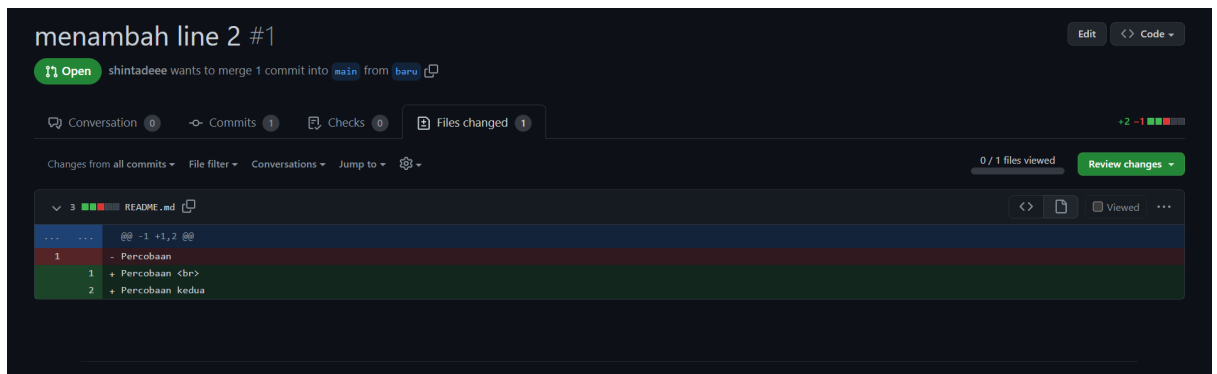
e. Setelah selesai melakukan perubahan kode, commit perubahan tersebut dengan menggunakan perintah git commit -m "[pesan commit]".

```
PS D:\Belajar\tugas-shinta> git commit -m "menambah line 2"
[baru 2626164] menambah line 2
1 file changed, 1 insertion(+)
create mode 160000 tugas-shinta
PS D:\Belajar\tugas-shinta>
```

- f. Push branch yang telah diubah ke repository di Github dengan menggunakan perintah `git push -u origin [nama branch]`.

```
PS D:\Belajar\tugas-shinta> git push -u origin baru
To https://github.com/shintadeeee/tugas-shinta.git
```

- g. Setelah melakukan push, buat pull request pada branch tersebut dengan memilih opsi “Create Pull Request” pada halaman repository di Github.
- h. Tunggu sampai tim melakukan review pada perubahan kode yang dilakukan. Dalam hal ini, dapat dilakukan code review pada pull request.



- i. Jika diperlukan, lakukan perbaikan pada perubahan kode sesuai saran dari reviewer.
- j. Setelah perubahan kode diterima, gabungkan branch dengan branch utama di repository dengan menggunakan opsi “Merge Pull Request” pada halaman repository di Github.
- k. Setelah branch utama diperbarui dengan perubahan kode, lakukan sinkronisasi branch lokal dengan branch utama di repository dengan menggunakan perintah `git pull origin [nama branch utama]`.

```
PS D:\Belajar\tugas-shinta> git pull origin main
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 1.80 KiB | 102.00 KiB/s, done.
From https://github.com/shintadeeee/tugas-shinta
* branch          main      -> FETCH_HEAD
41c97e9..142f222  main      -> origin/main
Merge made by the 'ort' strategy.
 README.md | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

- l. Kemudian setelah sinkronisasi selesai, hapus branch yang telah digunakan untuk melakukan perubahan kode dengan menggunakan perintah `git branch -d [nama branch]`.

```
PS D:\Belajar\tugas-shinta> git branch -d baru
error: Cannot delete branch 'baru' checked out at 'D:/Belajar/tugas-shinta'
PS D:\Belajar\tugas-shinta>
```

- Lakukan management konflik pada github.
Beberapa langkah yang dapat dilakukan untuk menyelesaikan konflik pada Github, yaitu :
 - a. Pull changes dari repository : sebelum melakukan perubahan pada file yang sama dengan pengembang lain, pastikan untuk selalu melakukan sinkronisasi dengan repository terbaru dengan menggunakan perintah git pull.
 - b. Membuat branch baru : jika ingin membuat perubahan pada file yang sama dengan pengembang lain buat branch baru dengan menggunakan perintah git checkout -b [nama branch baru].
 - c. Mengubah file : lakukan perubahan pada file yang ingin diubah.
 - d. Menyimpan dan melakukan commit : setelah melakukan perubahan, simpan dan lakukan commit dengan menggunakan perintah git commit -m "[pesan commit]".
 - e. Push branch baru : setelah commit, push branch baru ke repository dengan menggunakan perintah git push origin [nama branch]
 - f. Buat pull request : setelah branch baru di push ke repository, buat pull request dengan mengklik tombol "New Pull Request" pada halaman repository.
 - g. Menyelesaikan konflik : jika terjadi konflik saat menggabungkan perubahan pada pull request, Github akan menampilkan pesan konflik dan memberi tahu file yang bermasalah. Untuk menyelesaikan konflik, buka file yang bermasalah dan perbaiki konfliknya secara manual. Setelah menyelesaikan konflik, simpan file dan lakukan commit dengan menggunakan perintah git commit -m "[pesan commit]".
 - h. Merge pull request : setelah konflik diselesaikan, merge pull request dengan mengklik tombol "Merge Pull Request".
 - i. Pull changes dari repository : setelah merge pull request, lakukan sinkronisasi dengan repository terbaru dengan menggunakan perintah git pull.
- Lakukan code review pada github yang kalian kumpulkan.