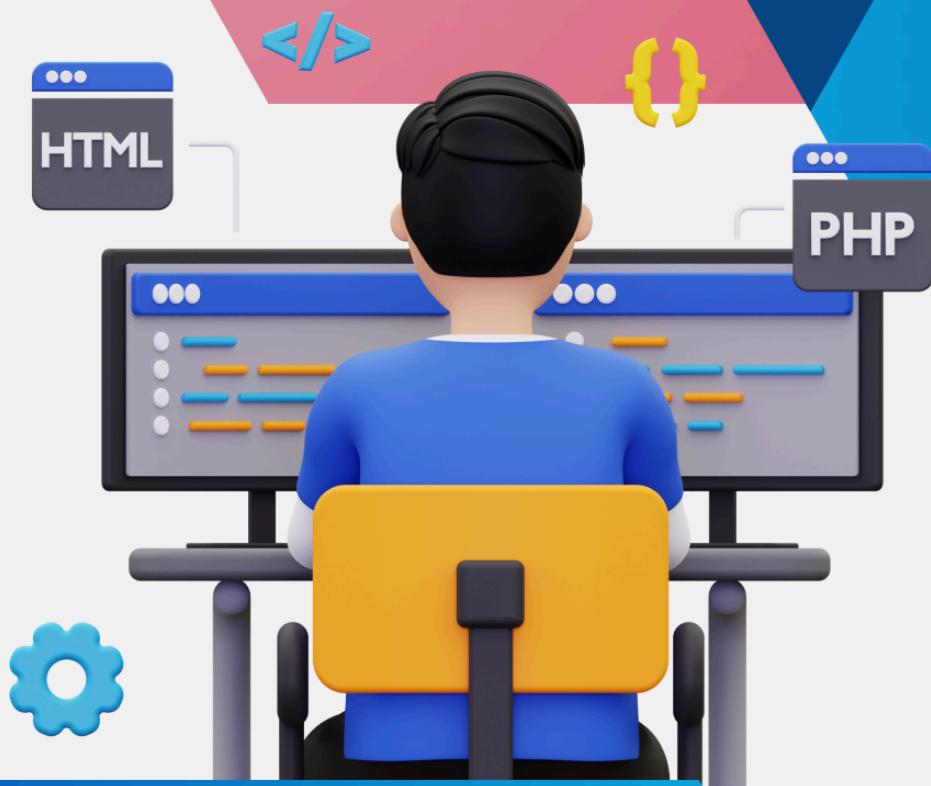


# Praktikum **Rekayasa** **Perangkat** **Lunak** 2024

Modul 1  
**Scenario**  
**Testing**



## Asisten

**Kode****Nama**

NWJN Alif Zaky Muhana

ICAM Ikram Zaidan Wicaksono

ANUG Muhammad Adam Nugraha

PUTU Putu Wisnu Wirayuda P.

RDND Rodia Andini

ARDS Ari Dwi Setyawan

DEYA Demitria Mayana A.

MHAP M. Hafidz Alkhair Pasaribu



## Daftar Isi

Peraturan Praktikum	2
Pengenalan Github	3
1. Repository pribadi	3
a. Persyaratan	3
1. Instalasi Sourcetree	3
d. Commit dan Push pada GitHub menggunakan Sourcetree	7
2. Repository bersama	15
a. Prasyarat	15
b. Create Repository & Invite Member	15
c. Create Branch, Commit, Push dan Merge	20
e. Membuat Pull Request	32
Referensi	34



## Peraturan Praktikum

1. Setiap peserta praktikum wajib hadir tepat waktu sesuai dengan jadwal. Jika melebihi batas waktu tersebut dan tidak dapat memberikan bukti alasan keterlambatan maka tetap dapat mengikuti praktikum tanpa tambahan waktu. Namun apabila **tidak bisa memberikan keterangan yang jelas** maka nilai akhir akan **dipotong 50%**
2. Perizinan Praktikum :
  - Izin berkaitan dengan Sakit atau Kemalangan, maka praktikan dapat memberikan surat perizinan kepada pihak Komisi Disiplin maksimal 3 hari setelah jadwal (Shift) praktikum.
  - Izin lomba atau penugasan institusi berlaku dengan memberikan alasan/bukti yang jelas dan dapat mengikuti susulan.
3. Seragam Praktikum :
  - Mahasiswa wajib menggunakan celana bahan hitam (bukan jeans atau chino) pada saat praktikum
  - Mahasiswi dapat menggunakan rok warna hitam yang tidak ketat pada saat praktikum
  - Dresscode Praktikum (Mengikuti Peraturan Telkom)
  - Senin : Menggunakan kemeja merah telkom atau kemeja putih polos telkom
  - Selasa s/d Rabu : Mengenakan kemeja putih polos telkom
  - Kamis & Sabtu : Menggunakan kemeja formal berkerah (bukan kerah sanghai dan bukan polo)
  - Jumat : Menggunakan kemeja batik
4. Konsekuensi :
  - Jika Praktikan tidak mengikuti dresscode yang berlaku maka **harus mengganti dengan dresscode yang berlaku** tanpa ada penambahan waktu asistensi.

## Pengenalan Github

Github merupakan platform *Version Control System* (VCS) dan kolaborasi dalam mendekvelop perangkat lunak dengan basis web. Github memungkinkan pengembang berkolaborasi secara lebih efektif dengan menyediakan alat untuk mengelola perubahan yang mungkin saling bertentangan dari beberapa pengembang. Github memungkinkan pengembang untuk mengubah, mengadaptasi, dan meningkatkan perangkat lunak dari repositori. Pada Modul 1 - Pengenalan GitHub akan dijelaskan langkah dasar dalam penggunaan *repository* pribadi dan *repository* bersama pada GitHub.

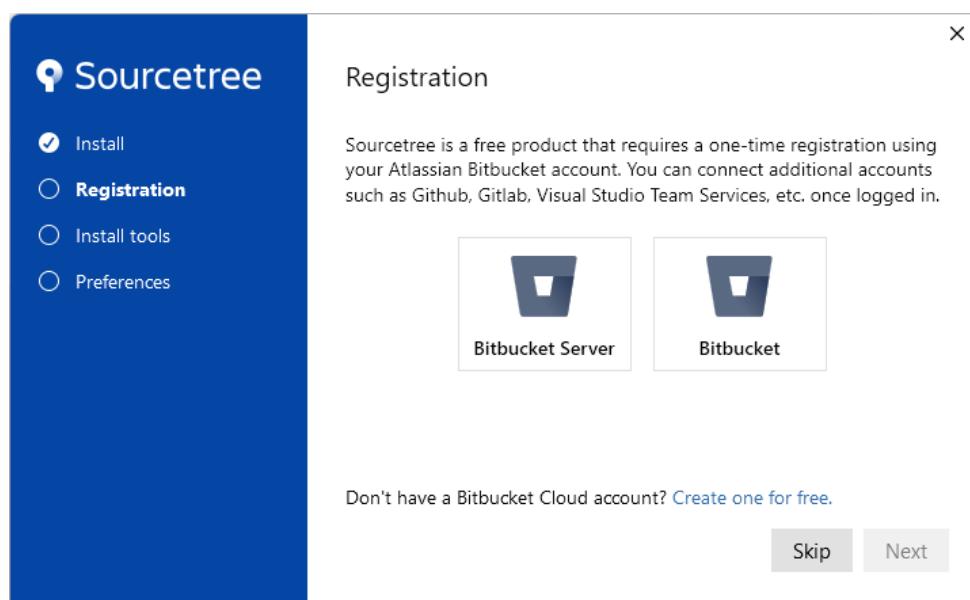
### 1. Repository pribadi

#### a. Persyaratan

1. Memiliki akun GitHub.
2. Download Sourcetree melalui link <https://www.sourcetreeapp.com/>.
3. Melakukan instalasi Sourcetree.

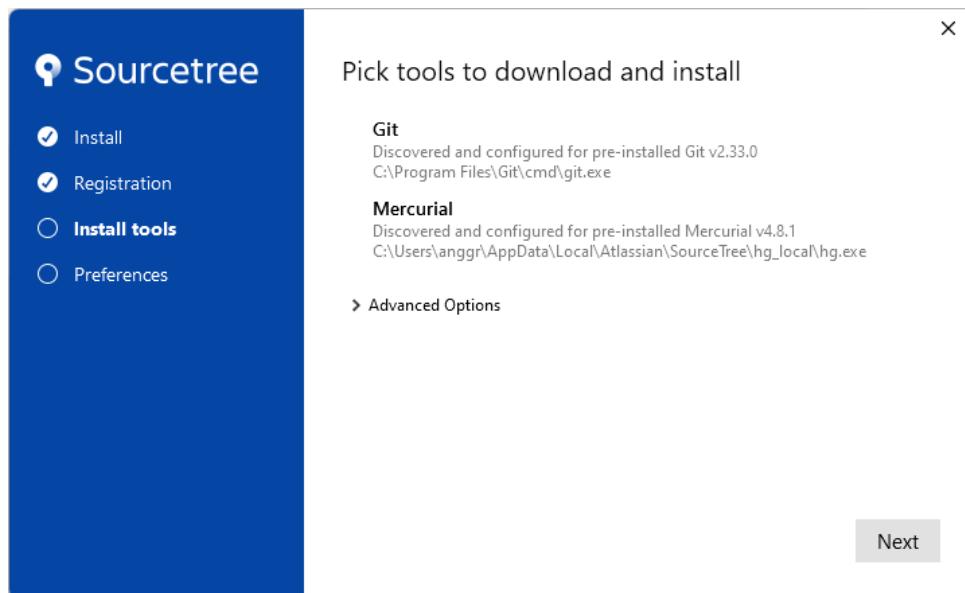
#### 1. Instalasi Sourcetree

1. Jalankan file Sourcetree yang sudah di-download pada link di atas,
2. Pada tab *Registration* pilih 'Skip'

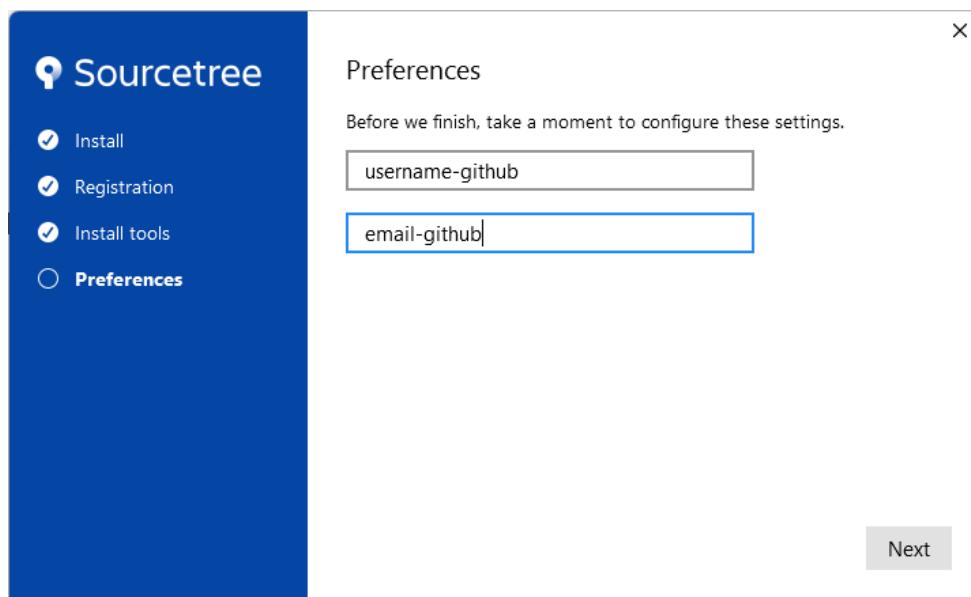




3. Pada tab *Install tools* pilih ‘Next’, uncheck mercurial

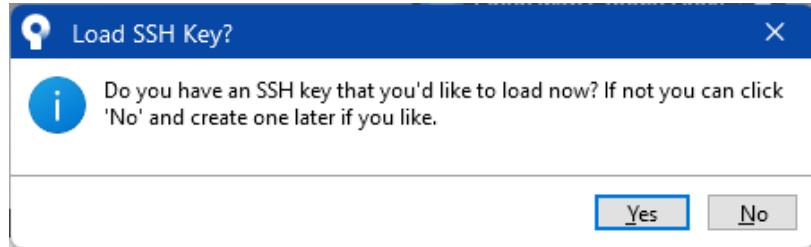


4. Pada tab *Preferences*, masukkan username dan email yang digunakan pada akun Github

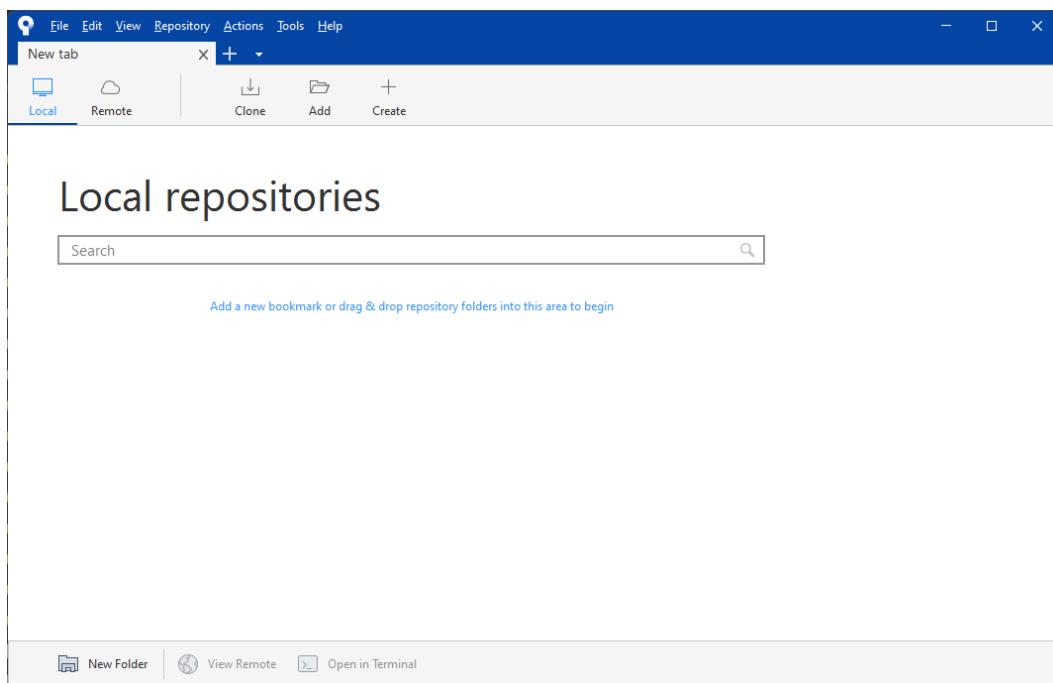




5. Selanjutnya pilih ‘No’

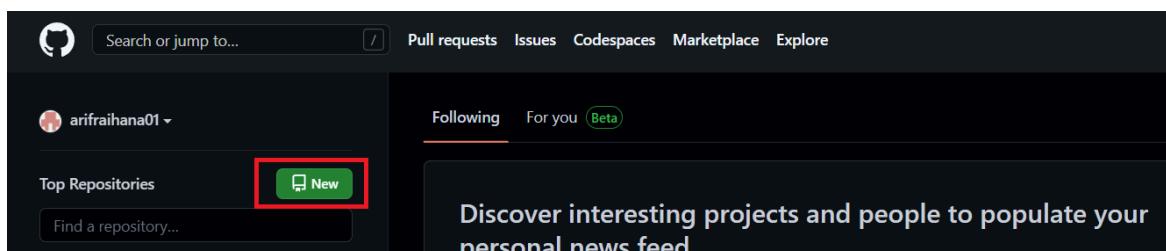


6. Berikut tampilan utama Sourcetree



## 2. Membuat Repository pada GitHub

i. Login GitHub lalu klik tombol “New” pada menu Repositories



ii. Kemudian akan diarahkan ke halaman untuk membuat repository baru sesuai dengan gambar ini



## Create a new repository

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

**Repository template**  
Start your repository with a template repository's contents.

No template ▾

**Owner \*** arifraihana01 / **Repository name \***

Great repository names are short and memorable. Need inspiration? How about [stunning-guacamole](#)?

**Description (optional)**

**Public**  
Anyone on the internet can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

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

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more](#).

.gitignore template: None ▾

**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more](#).

License: None ▾

ⓘ You are creating a public repository in your personal account.

**Create repository**

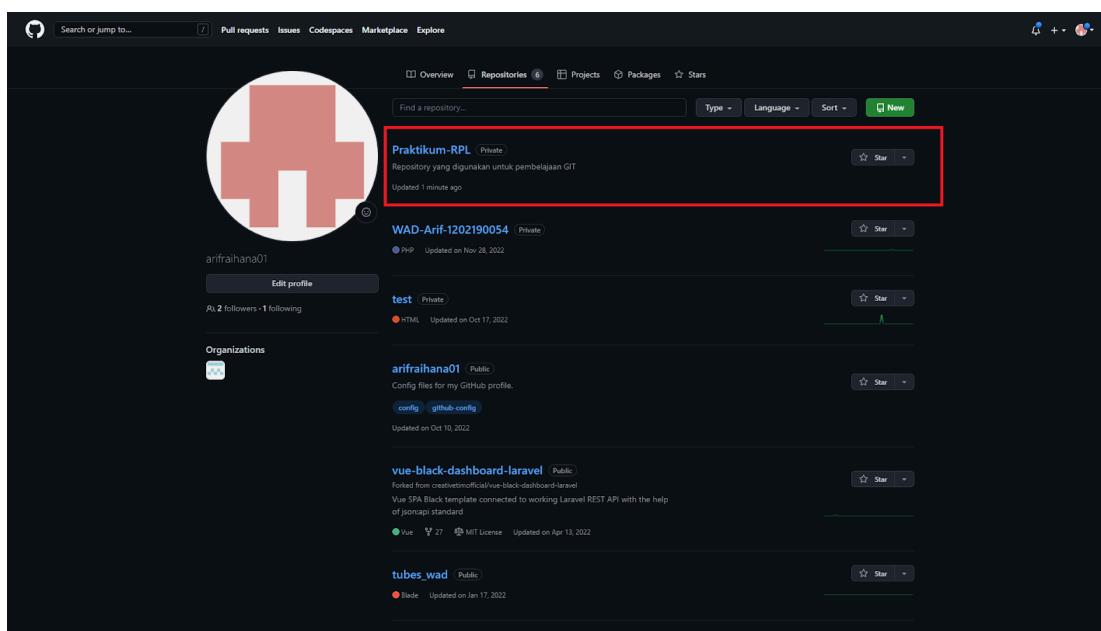
Dalam membuat repository baru ada beberapa hal yang perlu diperhatikan antara lain:

1. Nama Repository : berfungsi untuk memberikan identitas pada repository yang akan dibuat.
2. Deskripsi Repository : berfungsi untuk memberikan penjelasan terkait dengan repository yang dibuat.

3. Jenis Repository : terbagi menjadi **public** dan **private**. Ketika mengatur repository menjadi Public maka orang lain dapat melihat isi dari repository yang sudah dibuat. sebaliknya. ketika mengganti repository menjadi Private maka hanya kita dan orang yang memiliki akses yang dapat melihat isi dari repository yang sudah kita buat.

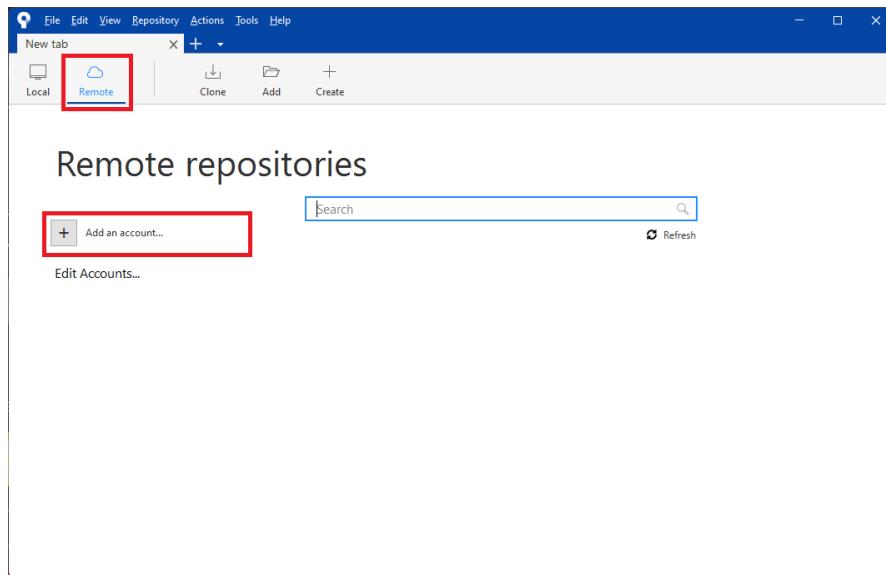
Setelah mengisi semua data yang dibutuhkan dalam membuat repository dilanjutkan dengan meng-klik “Create Repository”.

- iii. Berikut merupakan hasil ketika repository berhasil dibuat.

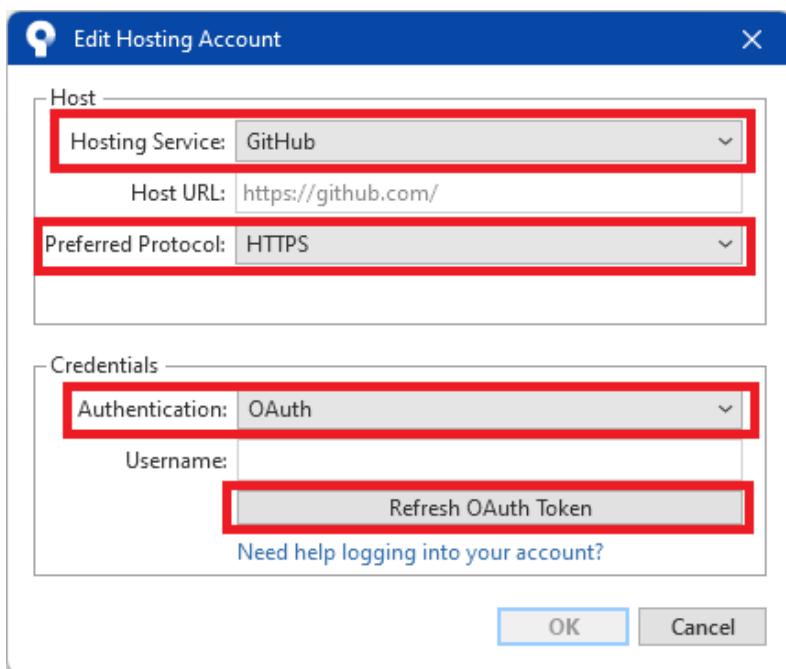


#### d. Commit dan Push pada GitHub menggunakan Sourcetree

1. Sebelum melakukan commit dan push di Sourcetree, kita harus masuk ke dalam akun Github dengan cara buka tab *Remote* pada Sourcetree, lalu pilih ‘Add an account’.



2. Menghubungkan akun Github dengan memperhatikan beberapa hal berikut.

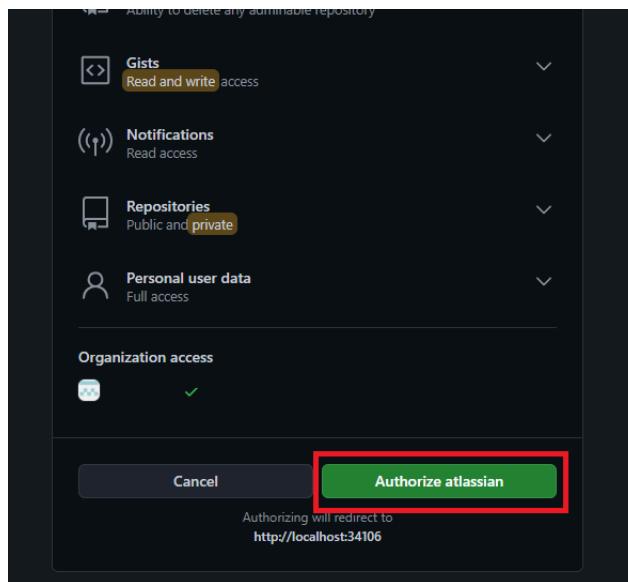


- Hosting Service : **GitHub**
- Preferred Protocol : **HTTPS**
- Authentication : **OAuth**

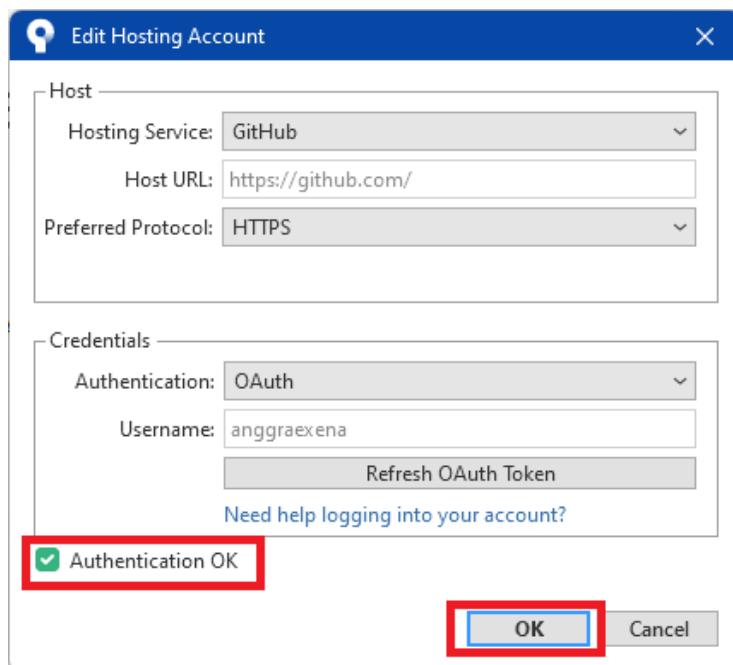
Setelah memastikan hal di atas, klik ‘*Refresh OAuth Token*’ yang akan mengarahkan pada tampilan konfirmasi *authorize* (jika sudah masuk akun GitHub pada website), namun jika belum masuk akun GitHub pada website akan diarahkan ke halaman *Sign in* GitHub untuk masuk akun GitHub terlebih dahulu.



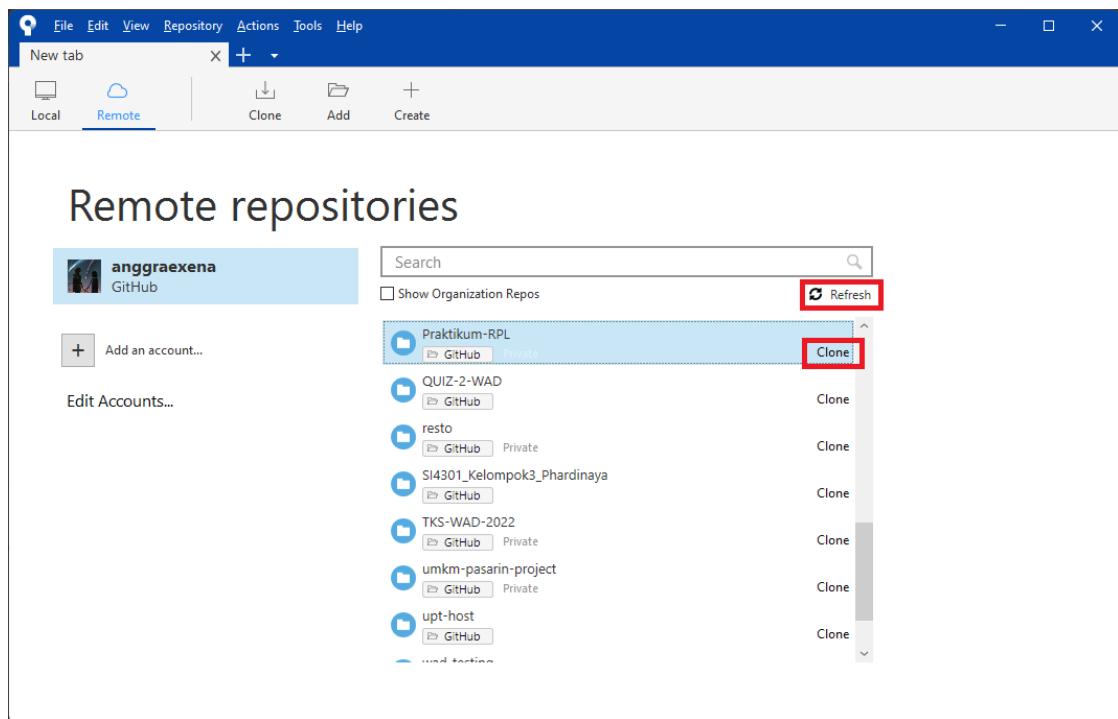
3. Berikut tampilan untuk mengonfirmasi *authorize* (perizinan) untuk menghubungkan Sourcetree dengan GitHub, klik ‘*Authorize atlassian*’.



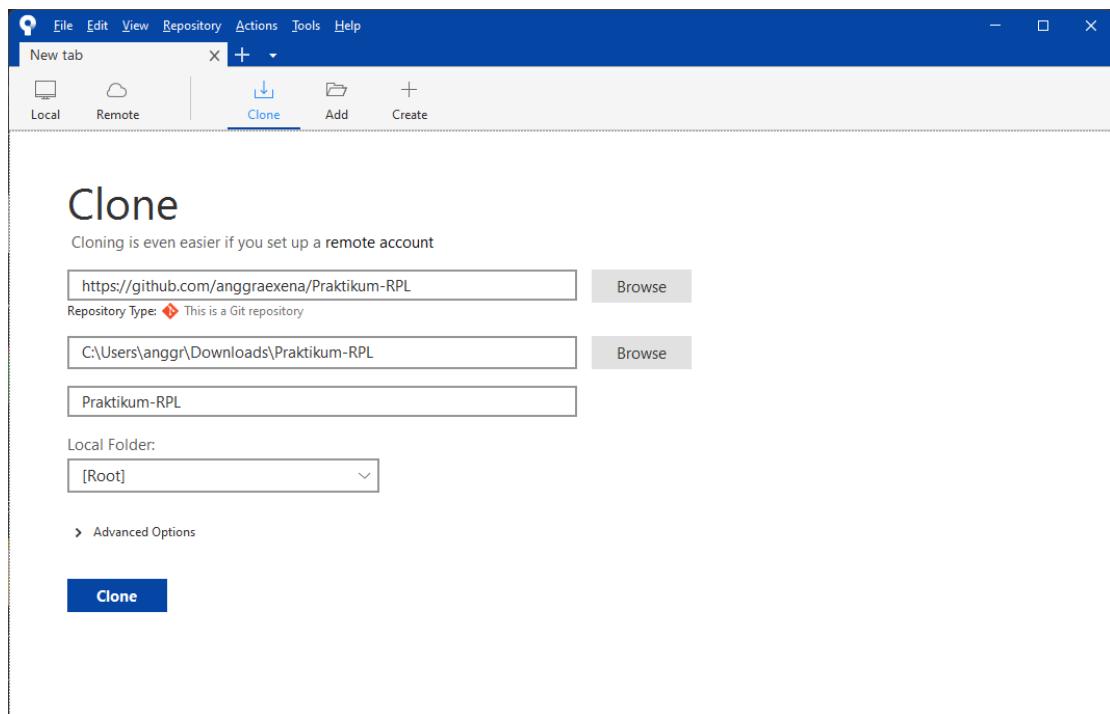
Jika sudah berhasil terhubung, pada Sourcetree akan tercentang Authentication-nya, lalu klik ‘Ok’.



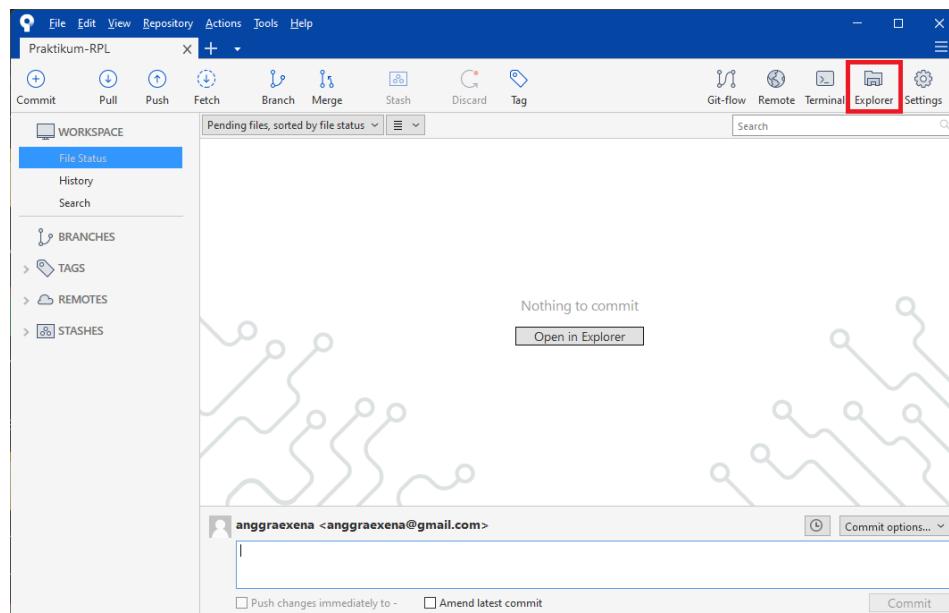
4. Ketika akun GitHub sudah terhubung dengan Sourcetree, maka kita dapat melihat semua repository yang kita miliki. Klik ‘*Refresh*’ untuk melihat repository yang baru dibuat. Klik ‘*Clone*’ pada repository yang akan kita gunakan.



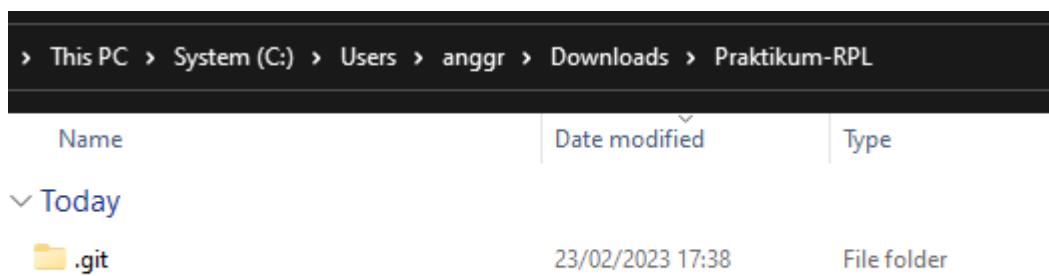
5. Fungsi clone adalah untuk menduplikasi repository yang sudah ada ke *local disk* komputer kita. Ada beberapa hal yang perlu diperhatikan yaitu.
  - Source Path Remote Repository : Link repository GitHub yang akan digunakan
  - Destination Path : Path **lokal** yang akan menyimpan repository
  - Nama Local RepositoryJika sudah selesai, klik ‘Clone’ dan tunggu hingga proses selesai.



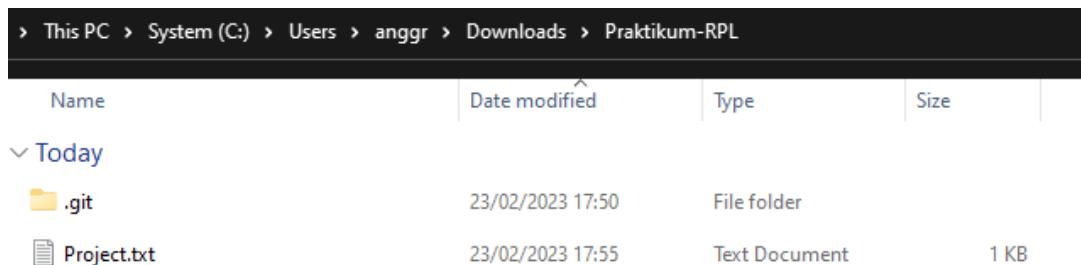
6. Setelah proses clone selesai, maka *local repository* berhasil dibuat. Berikut adalah tampilan dari *local repository* yang kita miliki. Untuk melihat isi repository tersebut, klik ‘*Explorer*’ agar diarahkan pada folder repository lokal.



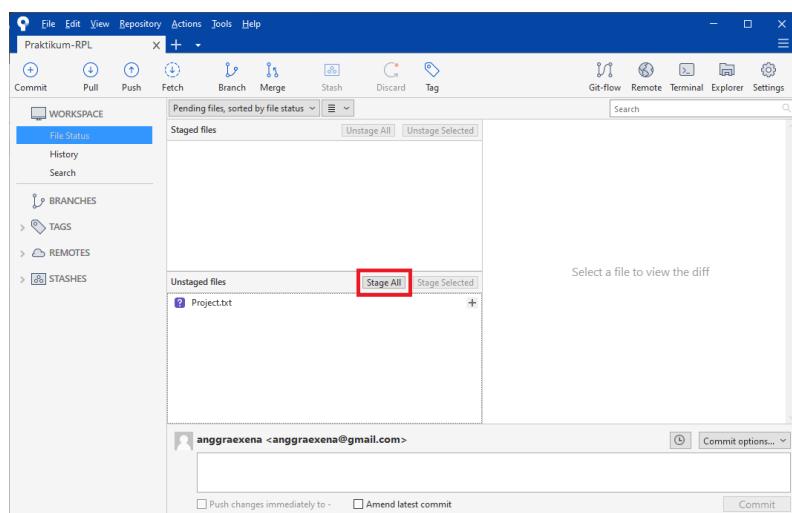
Berikut tampilan folder lokal hasil *clone repository*, file .git menandakan bahwa folder tersebut sudah terinisialisasi dengan remote repository GitHub.



7. Untuk melakukan commit dan push, tambahkan file proyek ke dalam repository lokal..

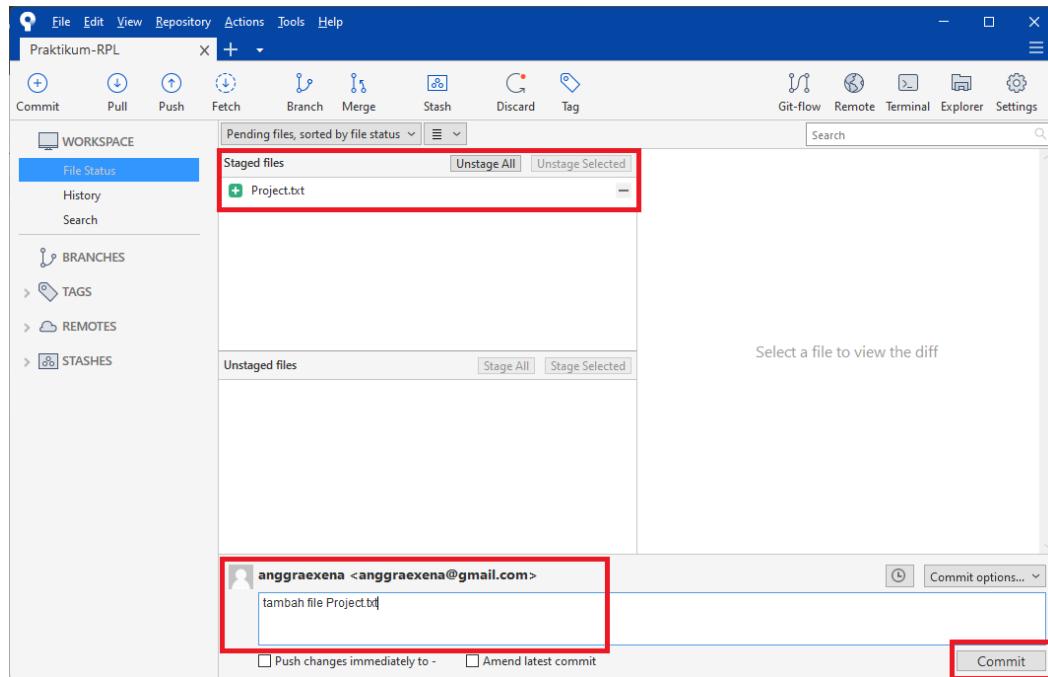


8. Setelah menambahkan file pada repository lokal, maka kita akan melihat file tersebut dalam posisi "*Unstaged File*" di Sourcetree. Semua perubahan apapun (menambah, mengubah, menghapus) yang terjadi pada proyek akan tercatat dan terlihat pada Sourcetree. Selanjutnya, untuk melakukan commit seluruh file yang berubah pilih '*Stage All*' atau jika hanya beberapa file saja yang akan di commit, kita dapat memilih '*Stage Selected*'.

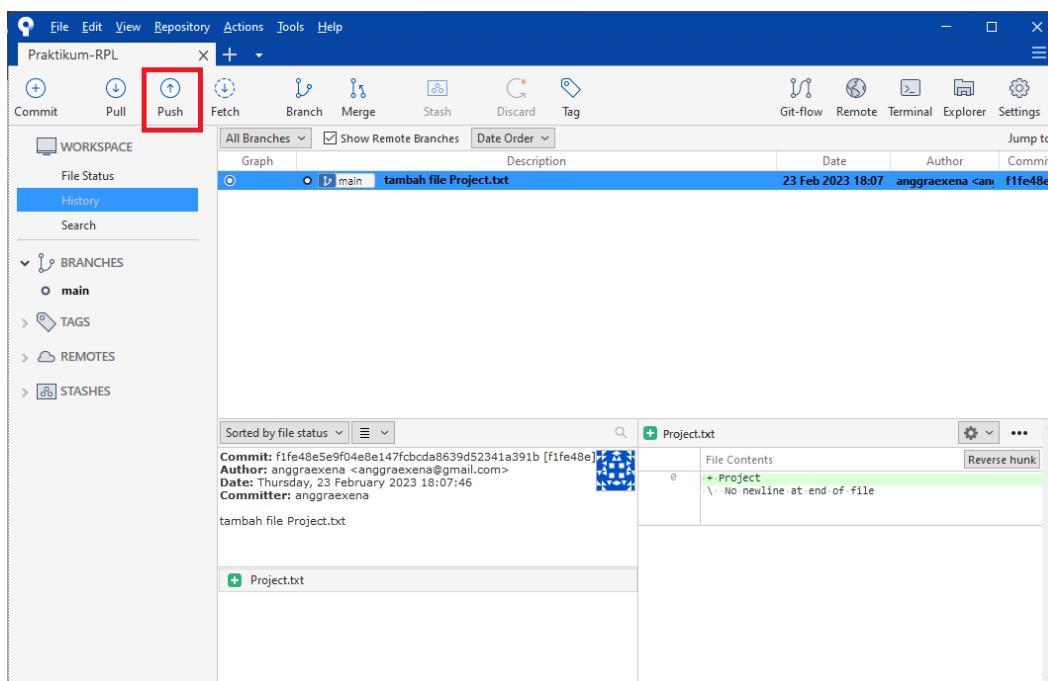




9. Sebelum melakukan commit, pastikan file yang sudah dibuat sudah masuk ke dalam Staged Files, setiap melakukan commit berikan komen sesuai dengan kebutuhan, jika sudah selesai maka klik ‘Commit’.

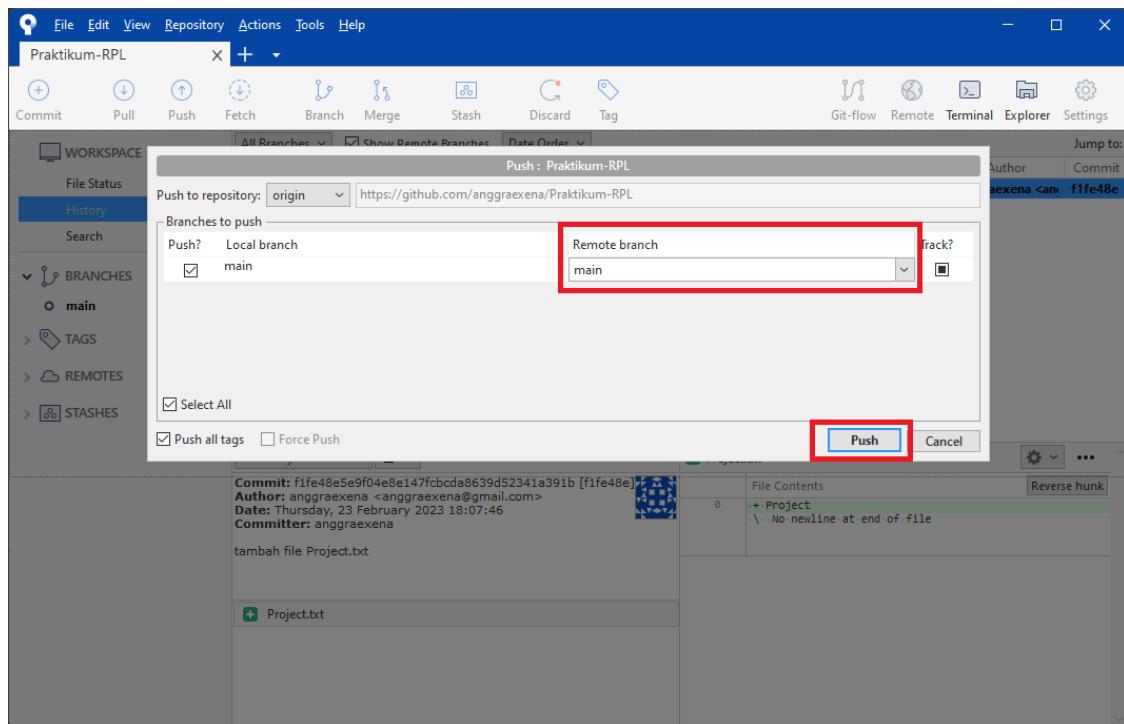


10. Setelah melakukan commit, maka kita akan melakukan push dengan klik ‘Push’ pada Sourcetree. Push digunakan untuk mengunggah file yang sudah di-commit sebelumnya ke repository GitHub.

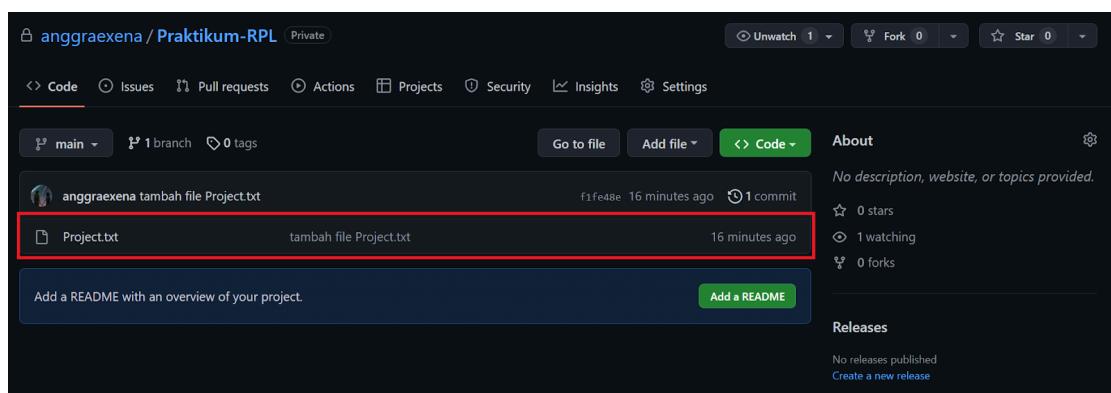




11. Sourcetree akan menampilkan pilih remote branch yang ada, karena pada repository ini kita belum membuat branch atau percabangan, maka pilih ‘main’ sebagai remote branch nya, lalu klik ‘Push’ untuk melanjutkan proses upload file ke repository GitHub dan tunggu hingga proses push selesai.



12. Jika sudah berhasil push, maka di repository GitHub sudah terlihat file yang kita tambahkan.





## 2. Repository bersama

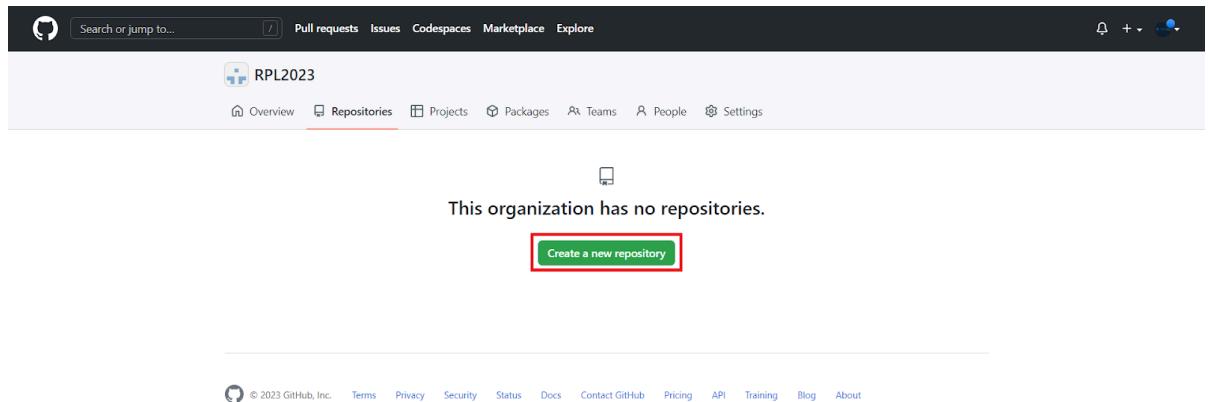
### a. Prasyarat

Mendapatkan invitation untuk join Repository **SI-RPL-2024** (Nama di sesuaikan kembali) di GitHub.

### b. Create Repository & Invite Member

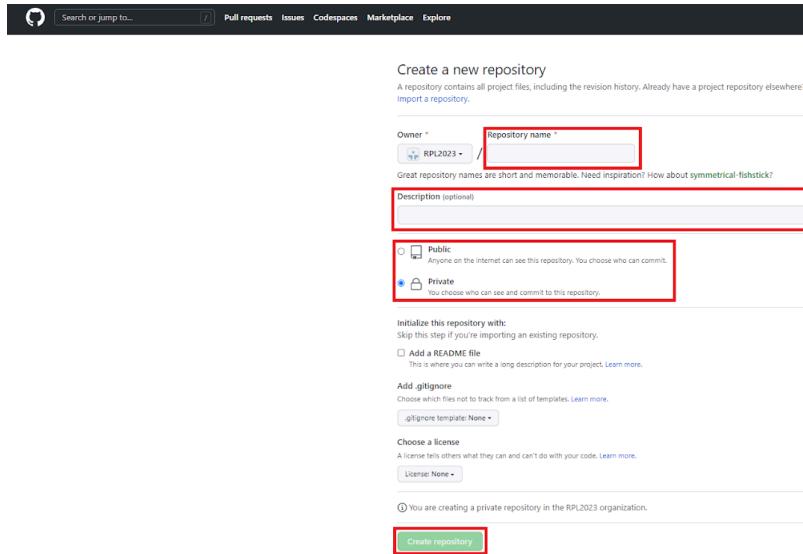
Setelah menerima invitation untuk join repository di GitHub, selanjutnya PM masing-masing tim dapat membuat repository bersama.

1. Pada tab Repositories, pilih tombol '*Create a new repository*' untuk membuat repository baru.



2. Kemudian akan diarahkan pada halaman untuk membuat repository baru seperti gambar di bawah ini. Untuk mengisi, ada beberapa hal yang perlu diperhatikan:
  - **Nama Repository:** digunakan untuk identitas repository yang dibuat.
  - **Deskripsi Repository:** berfungsi sebagai deskripsi dari repository yang dibuat.
  - **Repository:** jenis repository dibagi menjadi Public dan Private. Ketika teman-teman mengatur repository menjadi Public, orang lain dapat melihat repository yang teman-teman buat. Sebaliknya, jika teman-teman mengaturnya sebagai Private, maka repository tersebut hanya bisa diakses oleh pemilik repository dan orang yang diinvite ke dalam repository tersebut.

Setelah mengisi detail informasi di atas, klik Create Repository.



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 \*

Great repository names are short and memorable. Need inspiration? How about [symmetrical-fishstick](#)?

Description (optional)

Public Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

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

Add .gitignore Choose which files not to track from a list of templates. [Learn more](#).

.gitignore template: None

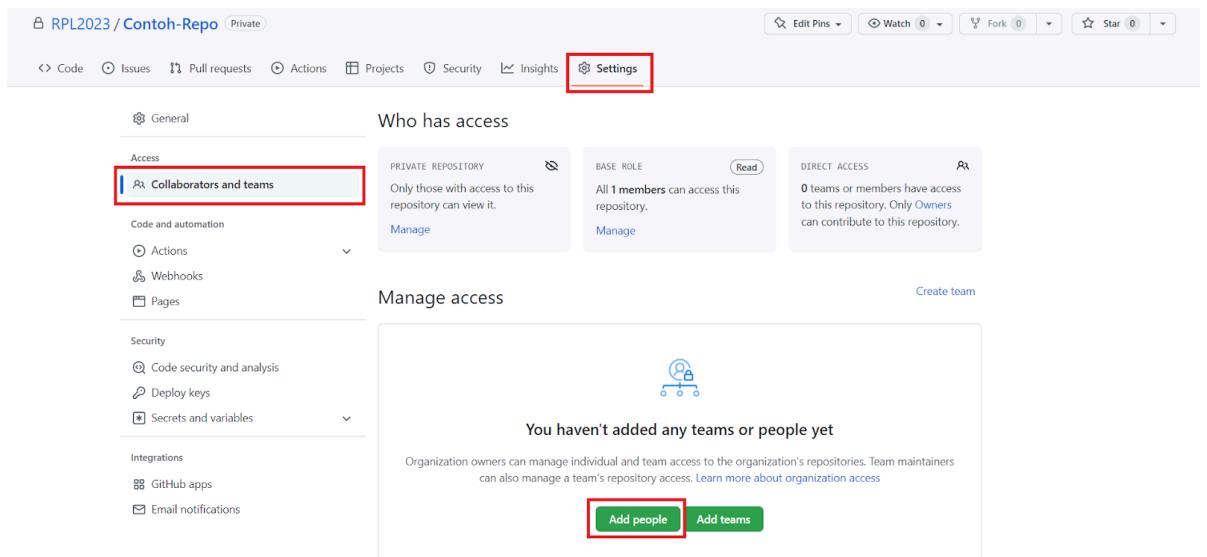
Choose a license A license tells others what they can and can't do with your code. [Learn more](#).

License: None

You are creating a private repository in the RPL2023 organization.

**Create repository**

- Setelah berhasil membuat repository, selanjutnya PM dapat mengundang anggota tim ke repository tersebut. Untuk melakukan invitation, Pilih tab ‘Settings’, lalu Pilih menu ‘Collaborators and teams’ dan Klik Tombol ‘Add people’.



RPL2023 / Contoh-Repo · Private

Code Issues Pull requests Actions Projects Security Insights Settings

General

Access

**Collaborators and teams**

Code and automation

Actions Webhooks Pages

Security

Code security and analysis Deploy keys Secrets and variables

Integrations GitHub apps Email notifications

Who has access

PRIVATE REPOSITORY

Only those with access to this repository can view it.

Manage

BASE ROLE

All 1 members can access this repository.

Manage

DIRECT ACCESS

0 teams or members have access to this repository. Only Owners can contribute to this repository.

Manage

Manage access

Create team

You haven't added any teams or people yet

Organization owners can manage individual and team access to the organization's repositories. Team maintainers can also manage a team's repository access. [Learn more about organization access](#)

**Add people** **Add teams**

Maka selanjutnya akan diarahkan ke pop-up untuk memilih anggota tim yang akan diundang berdasarkan username, full name atau email anggota tim



The screenshot shows the GitHub repository settings for 'RPL2023 / Contoh-Repo'. The 'Collaborators and teams' tab is selected. A modal window titled 'Add people to Contoh-Repo' is open, containing a search bar with placeholder text 'Search by username, full name, or email' and a green button at the bottom labeled 'Select a member above'.

Lalu inputkan username atau email teman anda, kemudian akan muncul profil teman anda, lalu klik pada profil teman anda.

The screenshot shows the GitHub repository settings for 'RPL2023 / Contoh-Repo'. The 'Collaborators and teams' tab is selected. A modal window titled 'Add people to Contoh-Repo' is open, showing a search bar with the text 'affadhlurr-19' and a result card for 'affadhlurr-19' with the option 'Invite outside collaborator'.

Lalu akan diarahkan untuk memilih role anggota tim, pilih salah satu role. Selanjutnya klik tombol ‘Add (username) to this repository’. Jika sudah, ulangi langkah sebelumnya untuk mengundang anggota tim lainnya.

The screenshot shows the GitHub repository settings for 'RPL2023 / Contoh-Repo'. The 'Collaborators and teams' tab is selected. A modal window titled 'Add people to Contoh-Repo' is open, showing a list of roles: 'Triage', 'Write', 'Maintain', and 'Admin'. Below the roles is a green button labeled 'Add affadhlurr-19 to this repository'.

Selanjutnya anggota tim akan menerima notifikasi invitations melalui email yang digunakan pada GitHub. Lalu klik ‘Accept invitation’.



RPL2023 / Contoh-Repo · Private

Code Issues Pull requests Actions Projects Security Insights

affadhlur19 invited you to collaborate

**Accept invitation** Decline

Owners of Contoh-Repo will be able to see:

- Your public profile information
- Certain activity within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

Is this user sending spam or malicious content?  
Block affadhlur19

Maka akan diarahkan ke halaman repository.

Search or jump to...

Pull requests Issues Codespaces Marketplace Explore

You now have admin access to the RPL2023/Contoh-Repo repository.

RPL2023 / Contoh-Repo · Private

Code Issues Pull requests Actions Projects Security Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/RPL2023/Contoh-Repo.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

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

```
echo "# Contoh-Repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/RPL2023/Contoh-Repo.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/RPL2023/Contoh-Repo.git
```

4. Jika semua anggota tim sudah menerima invitation, selanjutnya seluruh anggota tim melakukan clone repository dari github ke local dengan menggunakan SourceTree. Jalankan SourceTree dan pastikan akun github anda sudah terhubung dengan SourceTree. Jika belum ikuti langkah 1-3 pada poin “Commit & Push pada Github” di atas. Jika sudah, pilih tab clone pada halaman SourceTree.

File Edit View Repository Actions Tools Help

New tab +

Local Remote **Clone** Add Create

## Remote repositories

affadhlur-19 GitHub

Add an account...

Edit Accounts...

Search

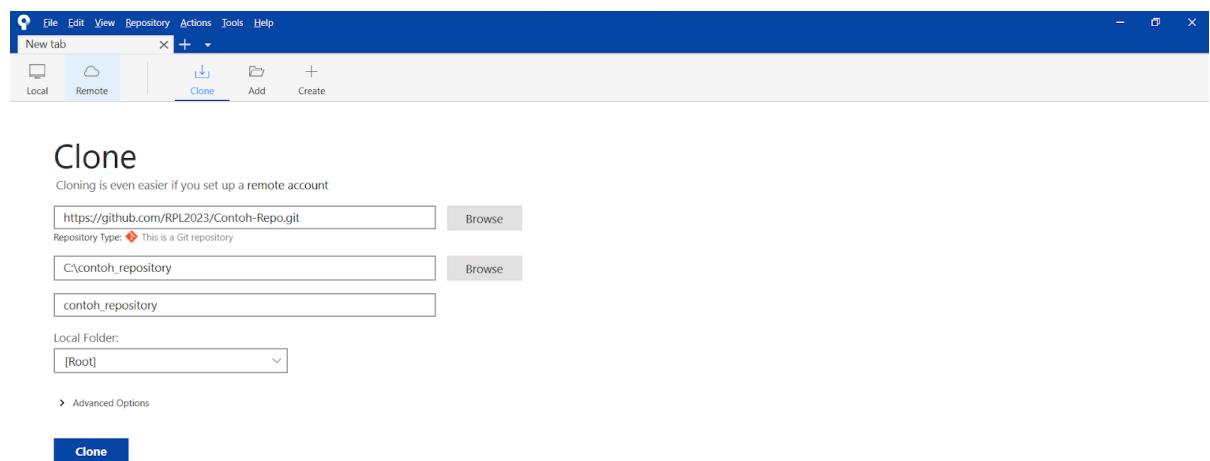
Show Organization Repos Refresh

5. Pada tab clone, ada beberapa yang harus di inputkan.

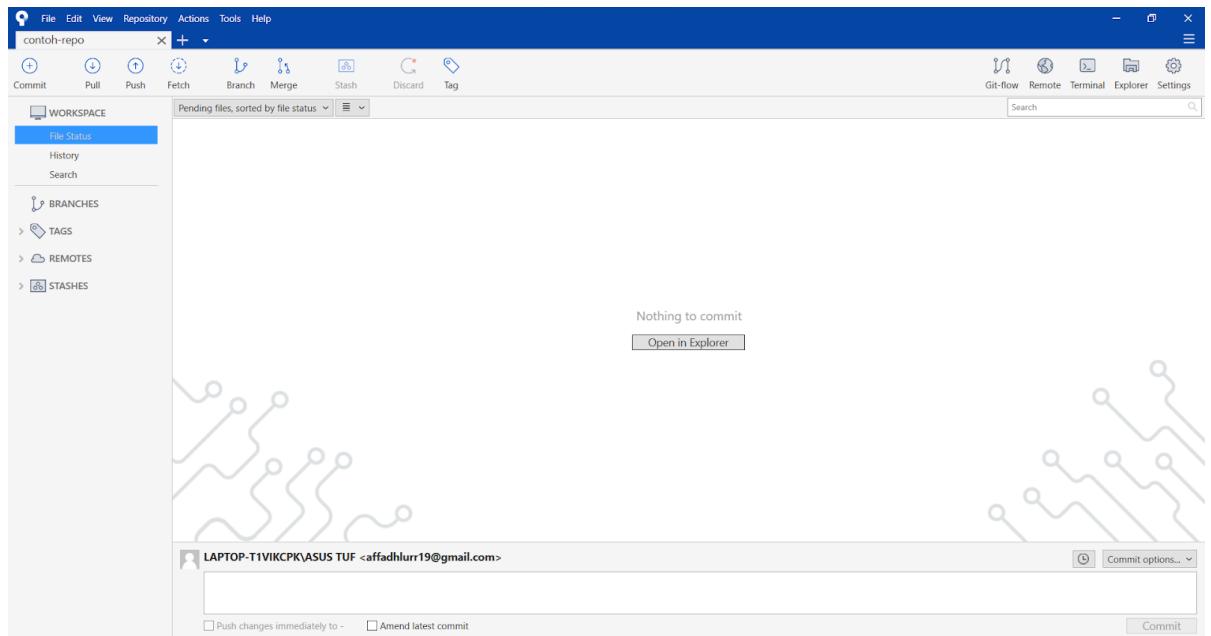
- Masukan Source Path/ URL repository github yang diperoleh dengan cara copy url https git dan paste pada kolom Source Path/URL.



- Masukan Destination Path untuk menentukan local path repository local.
- Masukan Nama Project sesuai kebutuhan.
- Pilih Local Folder : **Root**.
- Klik clone



6. Berikut merupakan tampilan SourceTree setelah berhasil melakukan cloning repository GitHub.



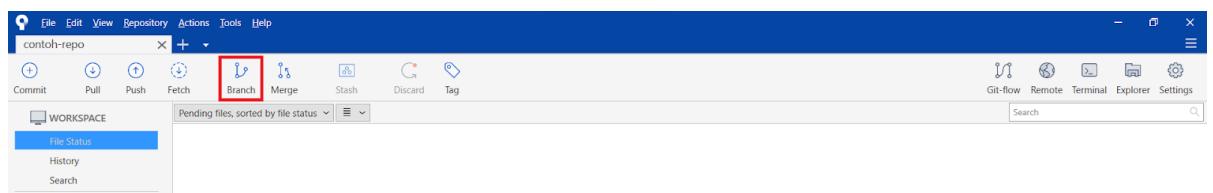
### c. Create Branch, Commit, Push dan Merge

Branch digunakan untuk mencegah konflik. Bayangkan teman-teman sedang bekerja dengan tim pada suatu repository git yang sama. Repository ini dikerjakan secara bersama-sama. Kemungkinan akan terjadi konflik, karena kode yang kita tulis berbeda dengan yang lain. Misalnya, Si A menulis kode untuk fitur X dengan algoritma yang ia ketahui. Sedangkan Si B menulis dengan algoritma yang berbeda. Lalu mereka melakukan commit, dan kode sumber akan jadi berantakan. Anggota tim lain menjadi pusing.

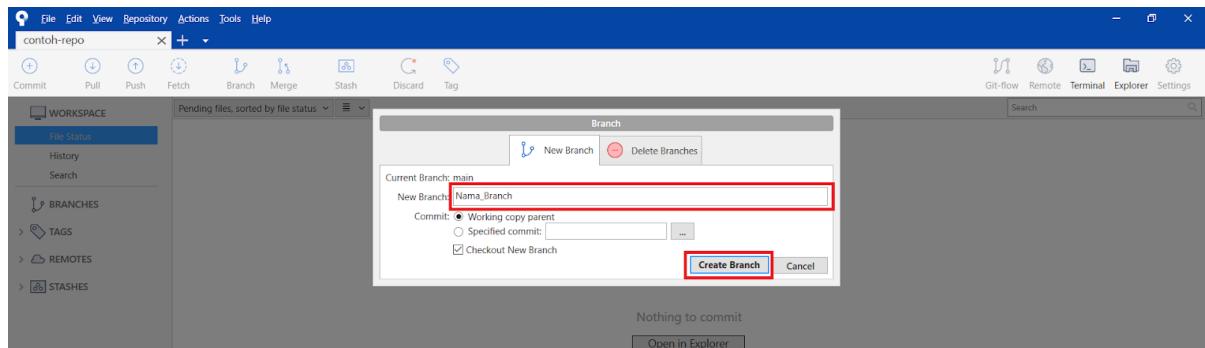
Agar tidak terjadi hal yang seperti ini, teman-teman harus membuat cabang (branch) tersendiri. Misalnya, Si A akan mengerjakan fitur X, maka dia harus membuat cabang sendiri. Si A akan bebas melakukan apapun di branchnya tanpa mengganggu cabang utama (master). Berikut merupakan cara membuat branch baru pada SourceTree.

#### **Langkah 1: Membuat Branch, Commit dan Push dari Branch**

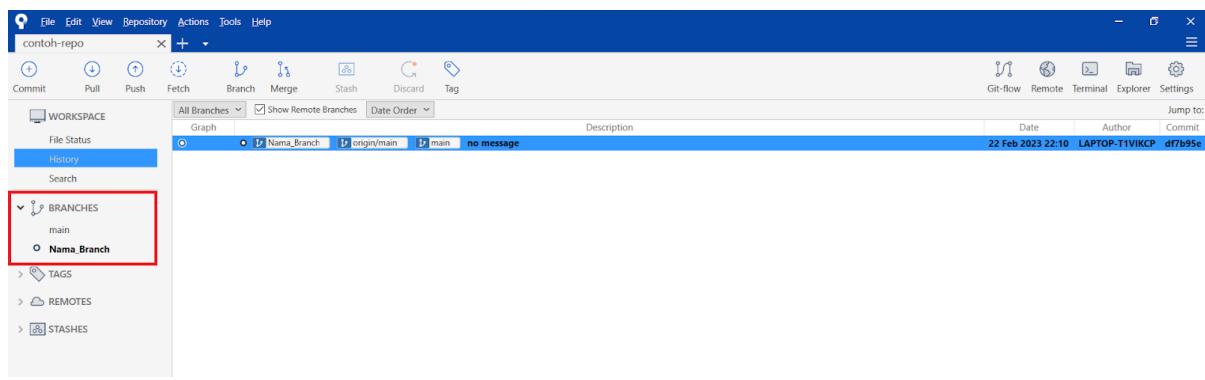
1. Untuk membuat branch baru, klik “Branch” pada menu atas.



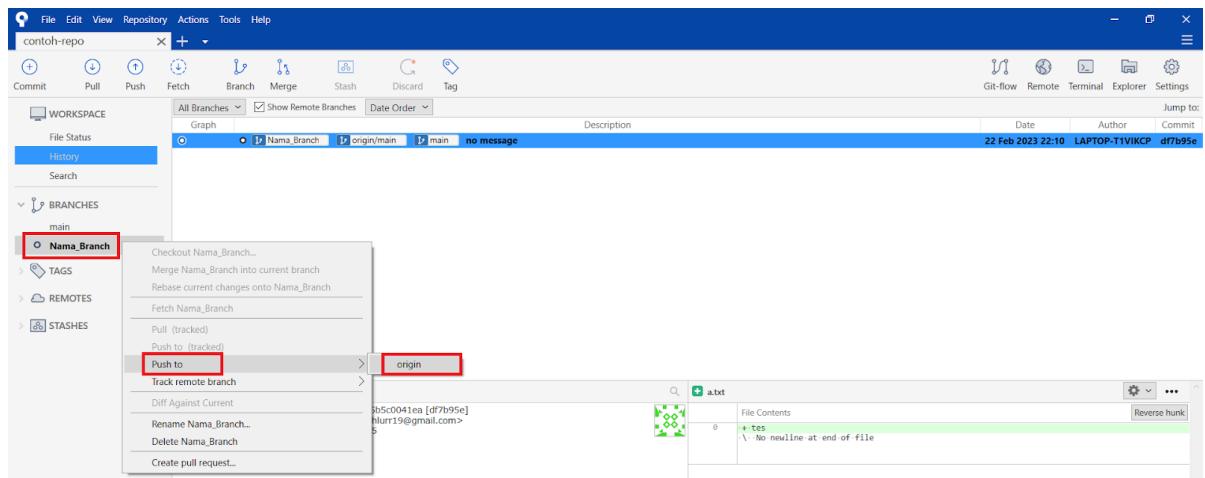
2. Kemudian akan muncul dialog box seperti dibawah ini. Kemudian inputkan nama branch yang ingin di buat, lalu klik “Create Branch”.



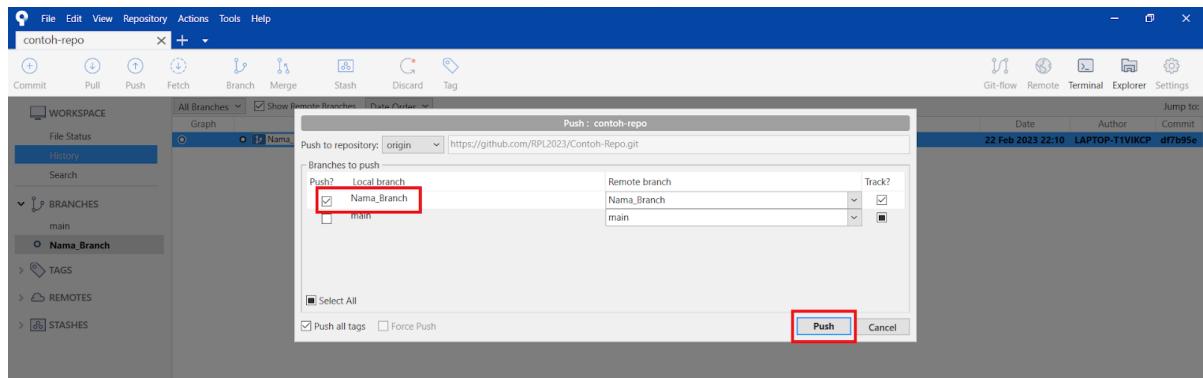
3. Berikut merupakan tampilan branch yang baru dibuat. Terdapat 2 branch yaitu branch master dan branch yang baru saja dibuat.



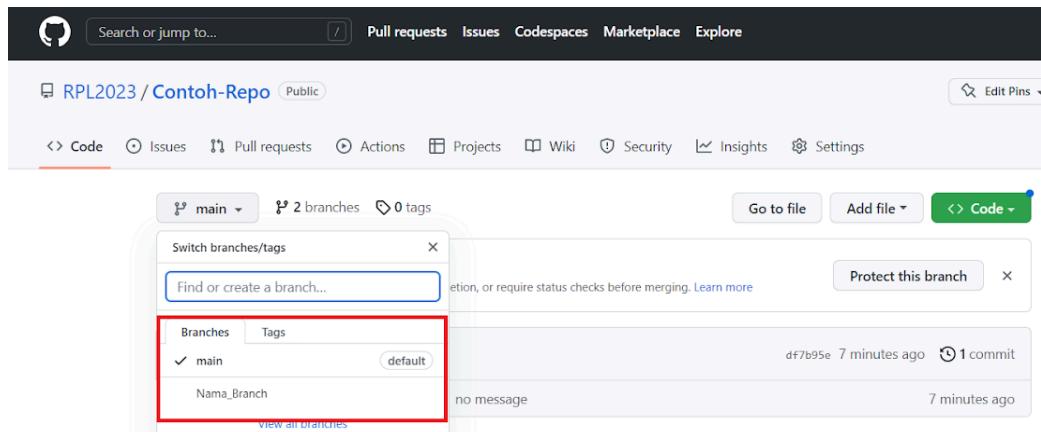
4. Kemudian push branch baru ke github. Klik kanan pada nama branchnya, lalu pilih “push to”, lalu pilih “origin”.



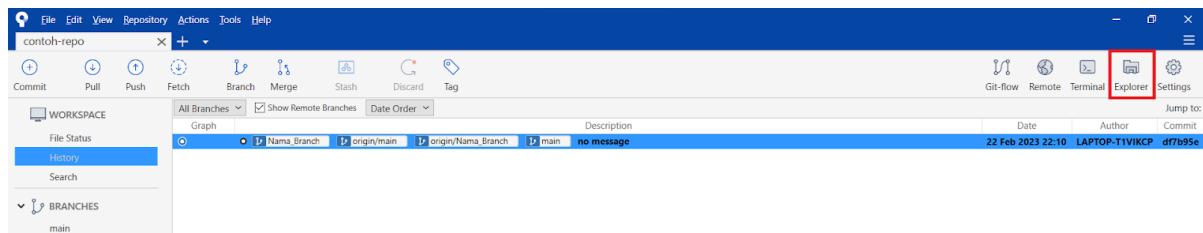
5. Kemudian akan tampil dialog box seperti dibawah ini. Lalu checklist branch yang ingin di push, lalu klik tombol push.



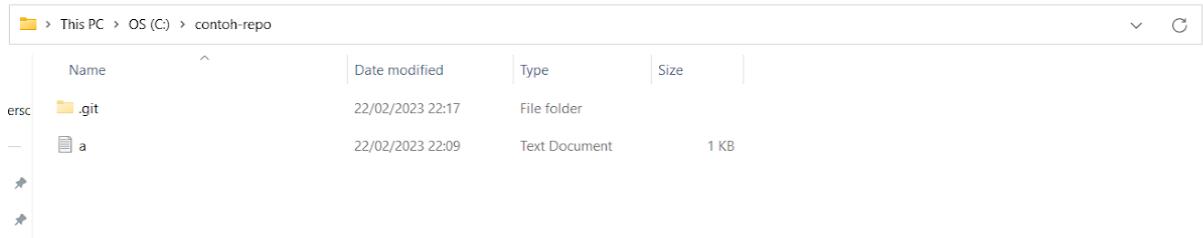
6. Kemudian check pada github, dan sekarang terdapat dua branch yaitu branch master dan branch yang baru saja di push



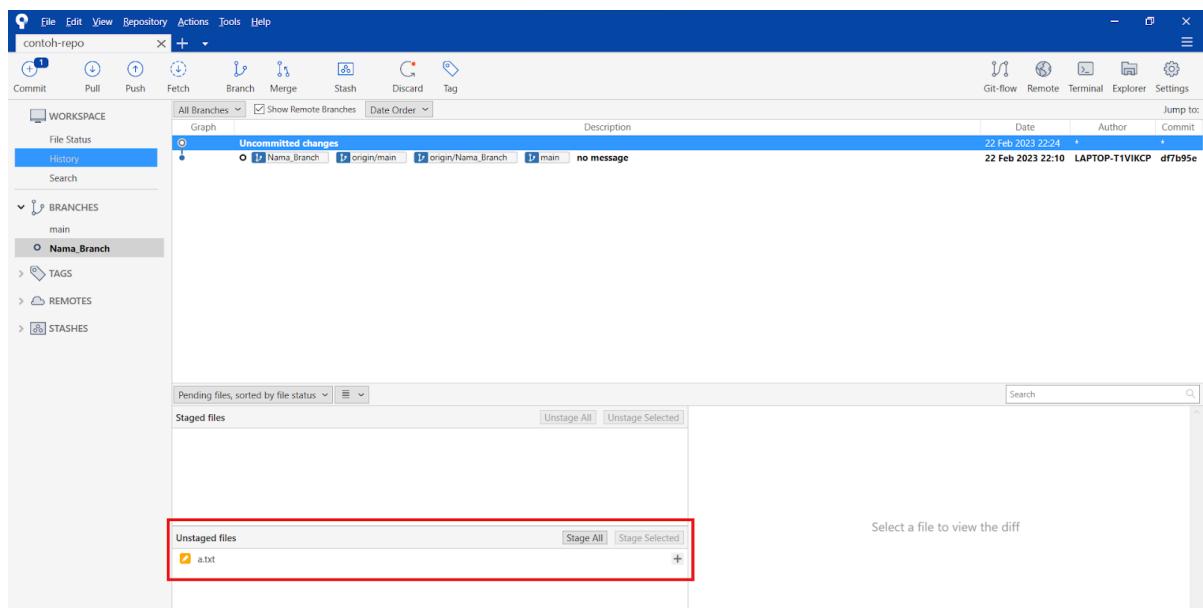
7. Kembali ke SourceTree. Pastikan current branch aktif adalah branch yang baru saja dibuat. Klik “Explorer” pada menu kanan atas untuk ke direktori local.



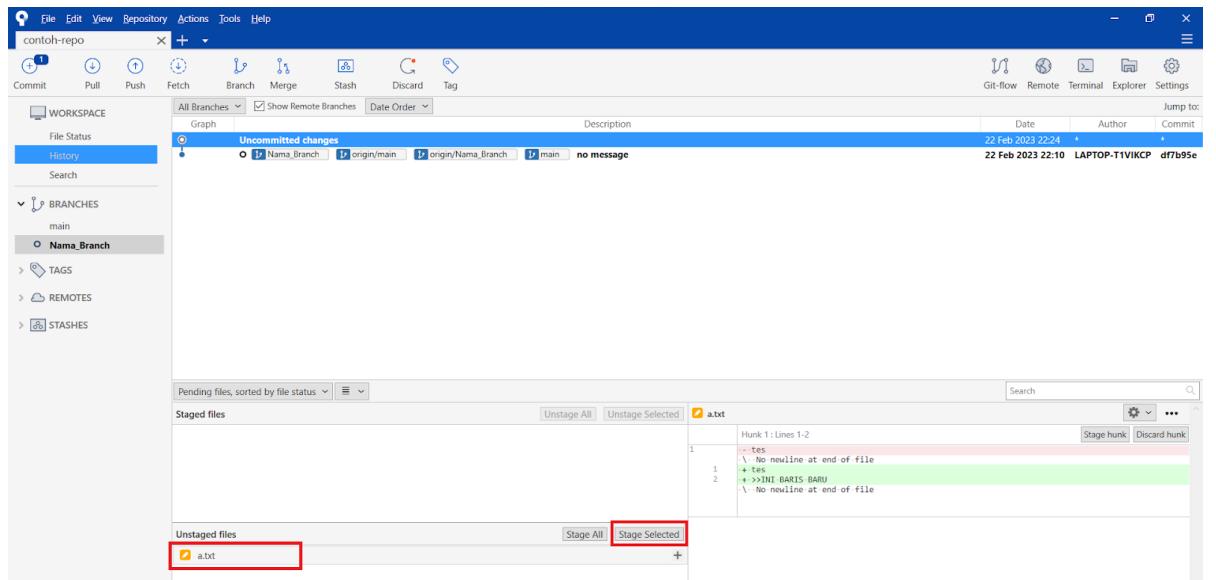
8. Lalu akan diarahkan ke direktori repository local. Sudah terdapat file a.txt (asumsikan file tersebut sudah dibuat dan di push oleh PM sebelumnya). Kemudian buka file tersebut dan lakukan perubahan (penambahan text atau paragraf). Lalu save file tersebut.



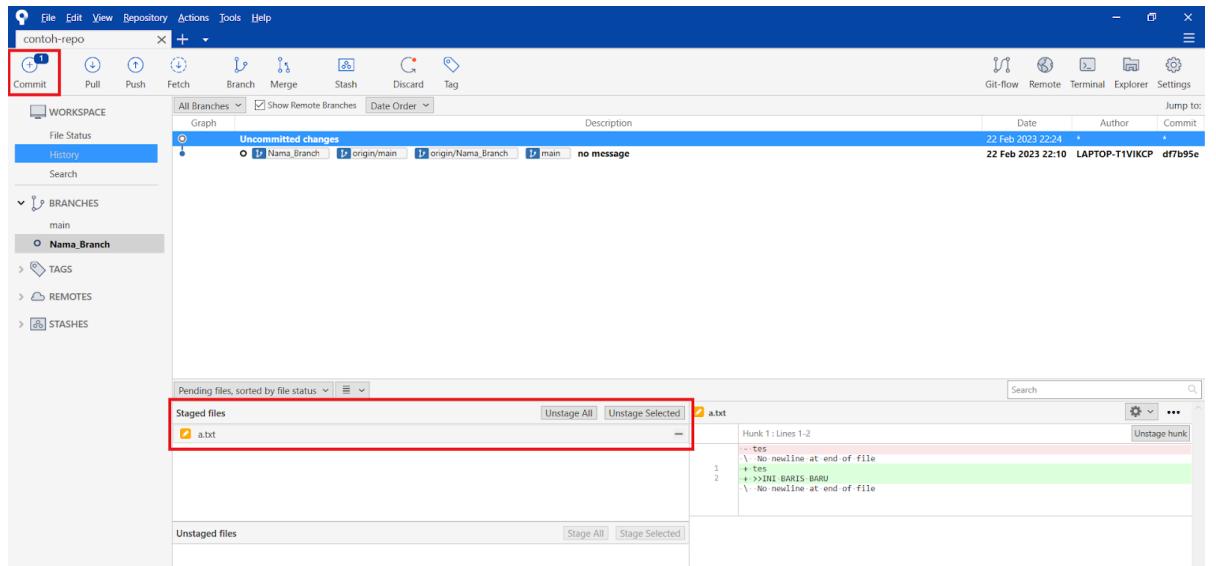
9. Lalu kembali ke SourceTree dan tekan F5 untuk melakukan *refresh*, maka akan menampilkan *notice* “**Uncommitted changes**” pada history graph, dan terdapat file apa saja yang mengalami perubahan pada bagian “**Unstaged Files**”.



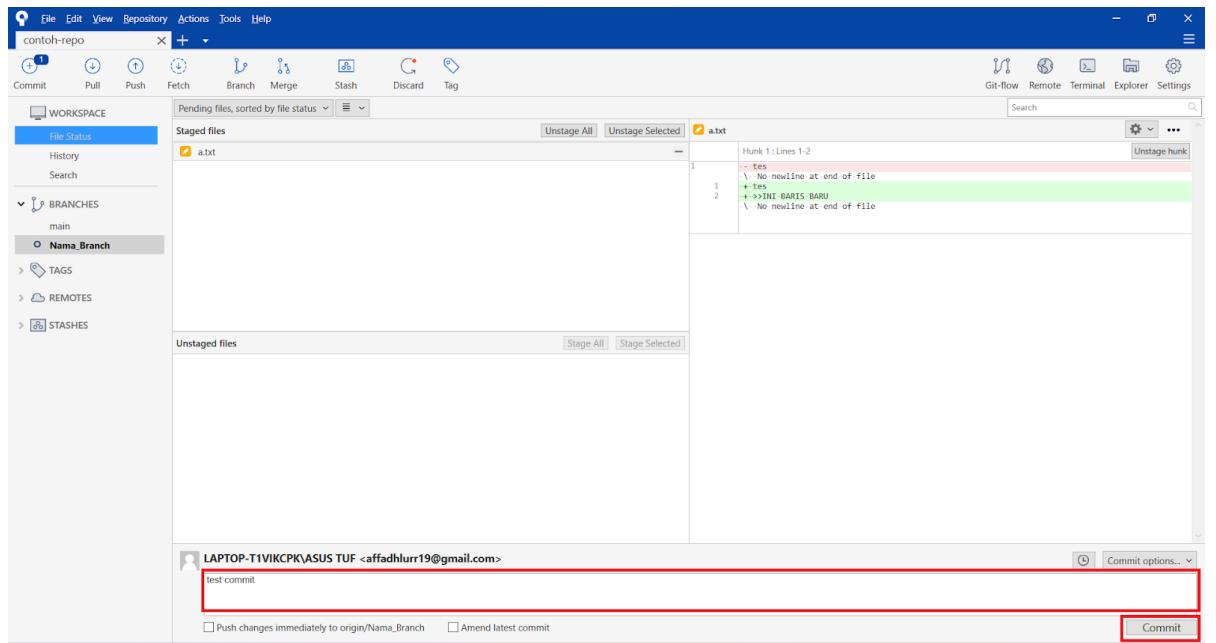
10. Untuk melakukan *commit file* yang mengalami perubahan ke *branch*, pada bagian “**Unstaged Files**” klik file yang akan di *commit*. Lalu klik “**Stage Selected**”



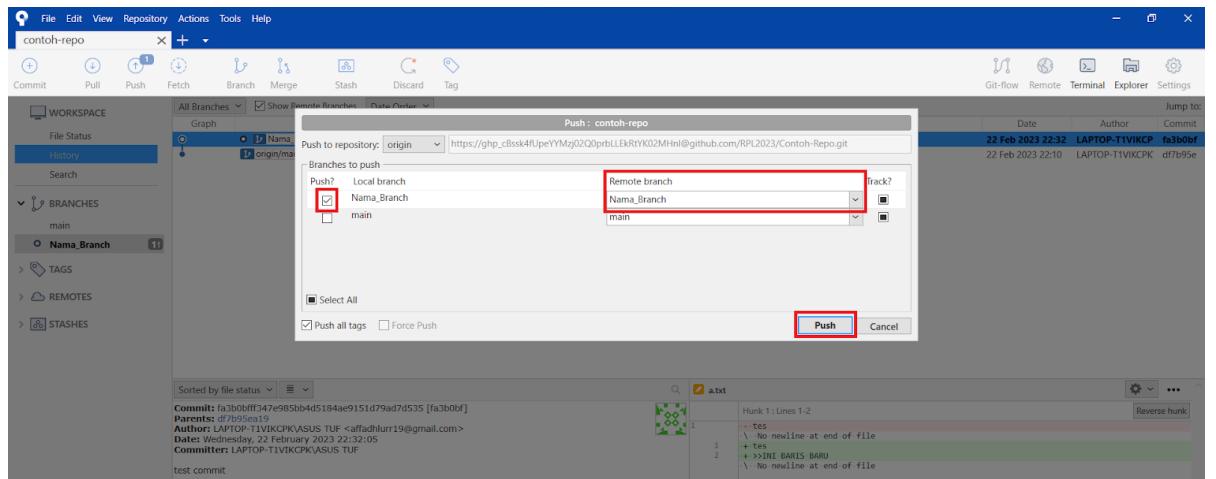
11. Kemudian file tersebut akan pindah ke bagian “**Staged Files**”. Lalu klik tombol “**Commit**” pada menu kiri atas.



Maka akan muncul sebuah *box* inputan untuk memasukan *comment*. Kemudian masukkan *comment*, lalu klik *button commit*.



12. Selanjutnya melakukan *push* untuk mengupload file tersebut ke *remote repository* github. Klik tombol “**Push**” pada menu atas, lalu pilih *remote branch* tujuan, lalu klik tombol *push*.



13. Kemudian *check* di *remote repository* github , lalu pilih *branch*, akan tampil seperti di bawah ini.



The screenshot shows a GitHub repository page for 'RPL2023 / Contoh-Repo'. At the top, there's a search bar and navigation links for Pull requests, Issues, Codespaces, Marketplace, and Explore. Below the header, there are tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A red box highlights the 'Code' tab. Underneath, a summary box shows '2 branches' and '0 tags'. A message indicates 'This branch is 1 commit ahead of main.' Below this, a commit card for 'affadhlur-19 test commit' is shown, with a red box highlighting it. The commit details include 'fa3b0bf 2 minutes ago' and '2 commits'. A file 'a.txt' is listed under the commit. The right sidebar shows repository statistics: 0 stars, 0 watching, 0 forks, and releases information.

Perubahan file yang di push tadi hanya terjadi pada branch yang baru dibuat yaitu Nama\_Branch. Hal itu terjadi karena kita membuat perubahan, commit, dan push di branch yang aktif yaitu Nama\_Branch.

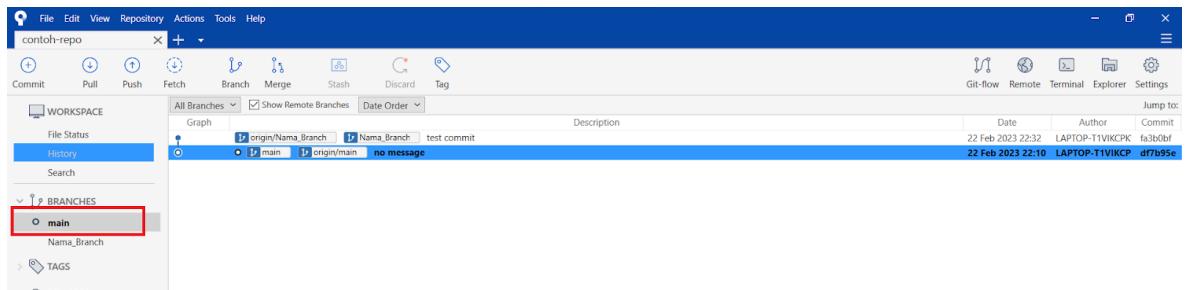
14. Kembali ke SourceTree, teman-teman dapat memilih current branch (yang aktif) dengan cara mengklik dua kali nama branch di bawah menu samping “Branches”. Untuk mengetahui branch mana yang aktif, teman-teman bisa tahu dari tampilan fontnya yang tebal.

The screenshot shows the SourceTree application interface. The left sidebar has sections for WORKSPACE, BRANCHES, TAGS, REMOTES, and STASHES. The BRANCHES section is expanded, and 'Nama\_Branch' is highlighted with a red box. The main pane displays a branch graph with 'origin/Nama\_Branch' and 'test commit' at the top, and 'origin/main' and 'main' below. The right pane shows commit details for 'test commit' on 'origin/Nama\_Branch' dated 22 Feb 2023 22:32 by LAPTOP-T1VIKCPK with commit fa3b0bf. Other commit details for 'main' are also visible.

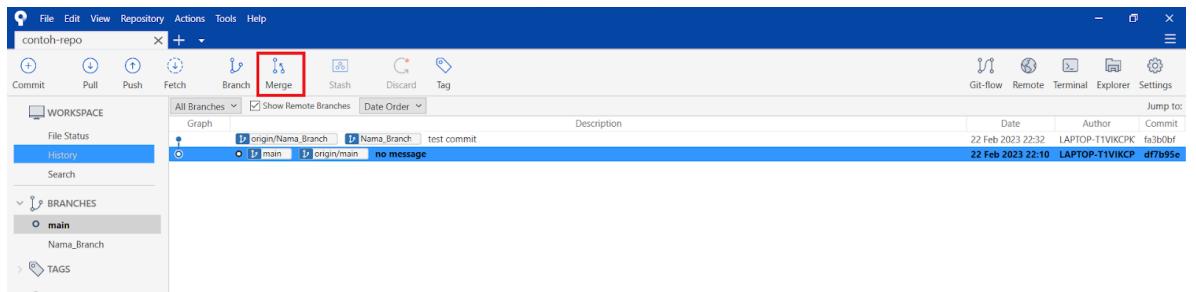
## Langkah 2: Merging File Perubahan dari Branch

Merging adalah proses penggabungan sebuah branch dengan branch yang lain. Kita coba berpindah ke branch master/main lalu menggabungkan branch Nama\_Branch ke dalamnya.

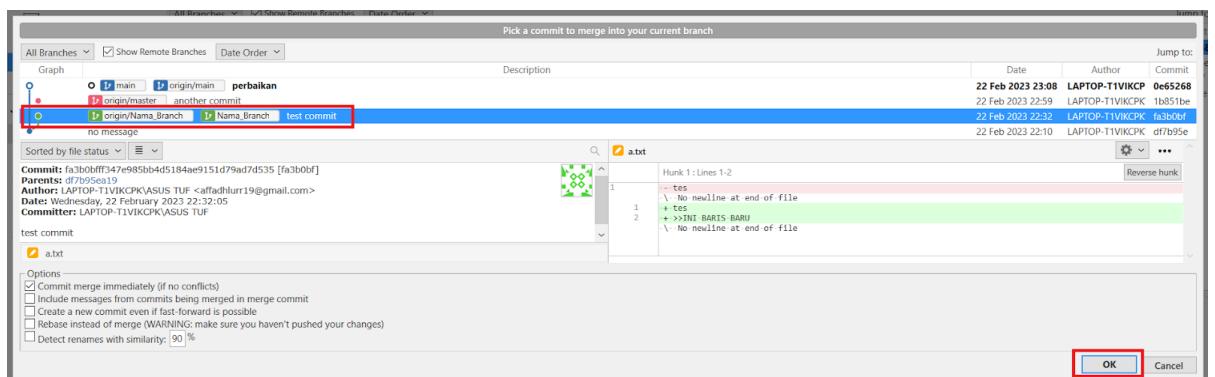
1. Pertama, switch ke branch master/main.



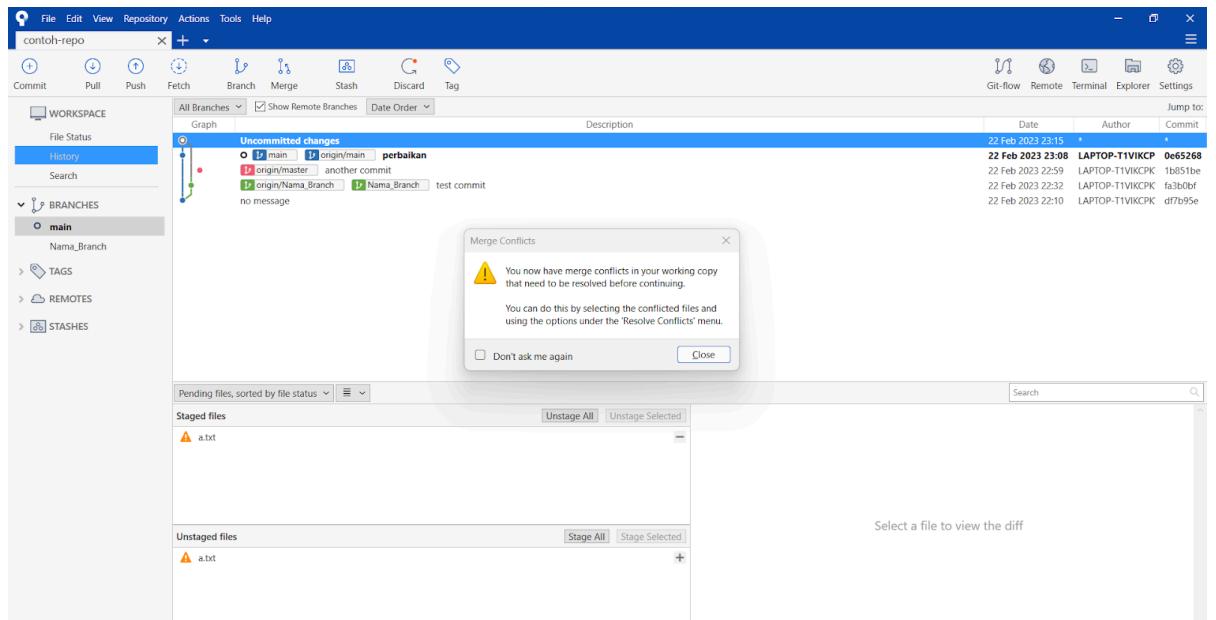
## 2. Klik tombol “Merge”



## 3. Dari popup yang muncul, pastikan di daftar commit, branch yang ingin digabungkan disorot. Itu menunjukkan bahwa kita ingin menambahkan commit dari branch ini ke branch utama /master atau /main. Kemudian klik OK.

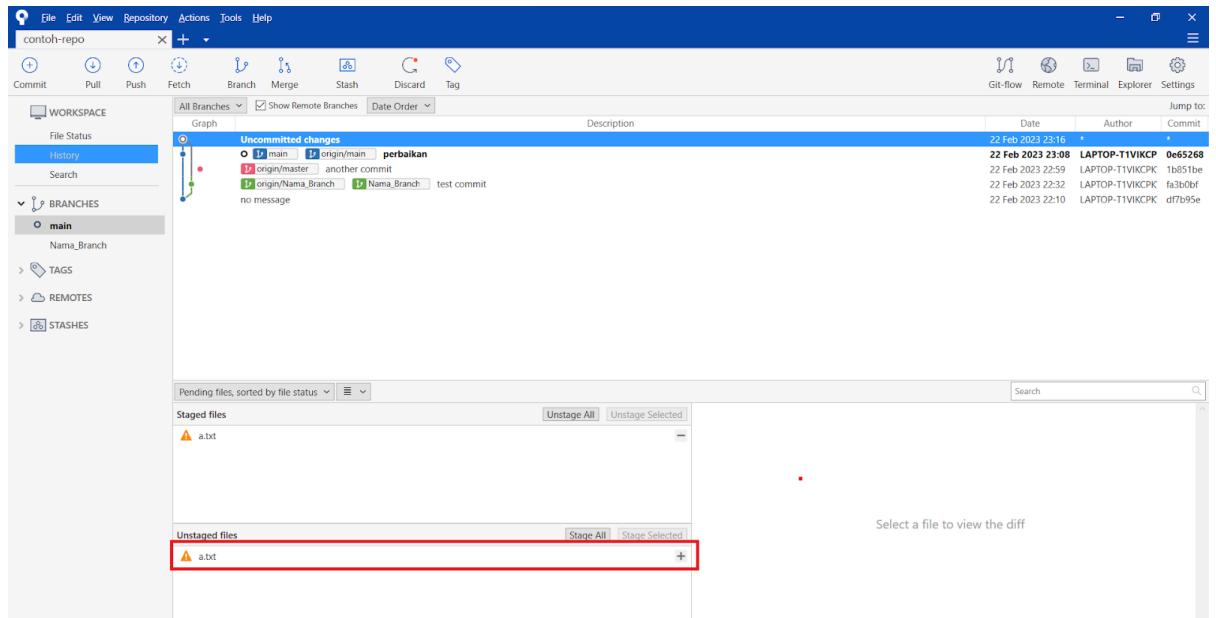


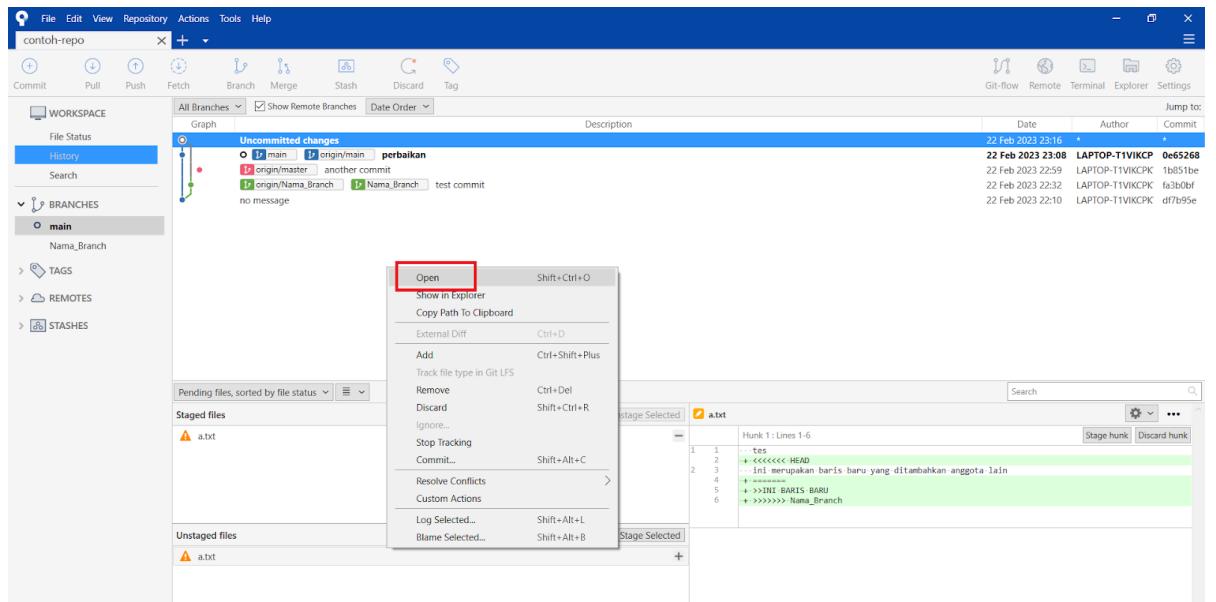
## 4. Jika muncul pesan seperti dibawah, itu menandakan terjadinya conflict. Tidak selamanya proses merging berjalan mulus tanpa masalah. Kadang kita mengalami conflict karena dalam kedua branch yang akan kita gabungkan ada satu atau beberapa file yang isinya berbeda di branch yang satu dengan yang lain. Hal ini akan lebih sering kita alami ketika kita berkolaborasi dengan orang lain menggunakan shared repository. Di sini kita berada dalam situasi yang disebut merge conflict karena test.txt di branch master dan test.txt di branch Nama\_Branch memiliki perbedaan isi tepat pada baris yang sama. Klik Close.



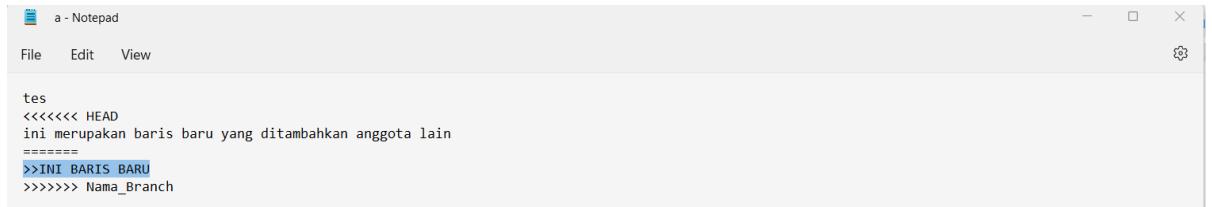
Note: Conflict terjadi jika sebuah file dalam dua buah branch yang akan digabungkan memiliki perbedaan tepat pada baris yang sama.

- Untuk melihat baris dan file yang menyebabkan konflik, klik kanan pada file yang terjadi konflik pada bagian Unstaged Files, lalu klik Open.





- Maka akan terbuka file yang terjadi konflik. Baris yang mengakibatkan konflik ditandai dengan deretan karakter <<<< dan >>>> di dalam file yang mengalami konflik, dalam contoh di atas, file a.txt.



```
a - Notepad
File Edit View

tes
<<<< HEAD
ini merupakan baris baru yang ditambahkan anggota lain
=====
>>INI BARIS BARU
>>>> Nama_Branch
```

Dalam contoh di atas, baris yang bermasalah adalah baris kedua yang berisi teks "ini merupakan baris baru yang ditambahkan anggota lain" (main) dan ">>INI BARIS BARU" (branch Nama\_Branch).

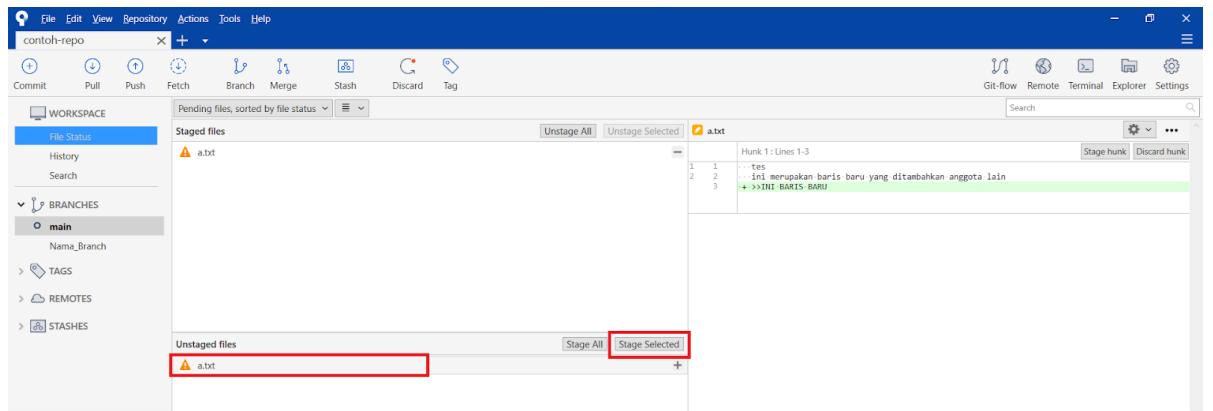
Kita gabungkan kedua teks tersebut dengan mengubah isi test.txt menjadi seperti berikut:



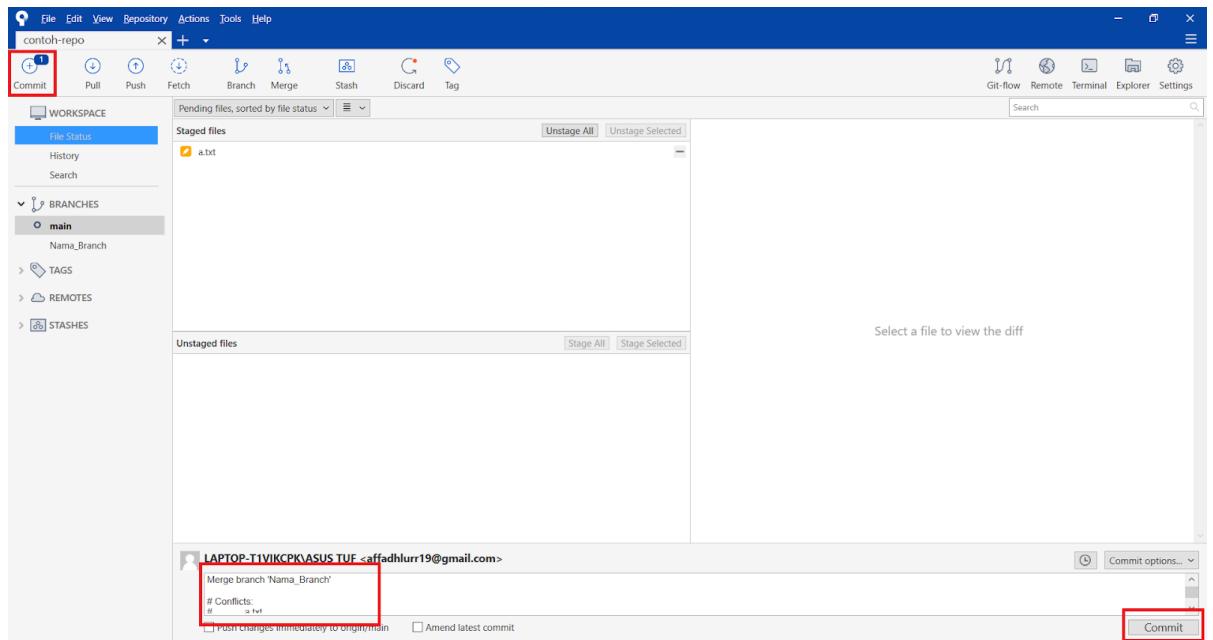
```
a - Notepad
File Edit View

tes
ini merupakan baris baru yang ditambahkan anggota lain
>>INI BARIS BARU
```

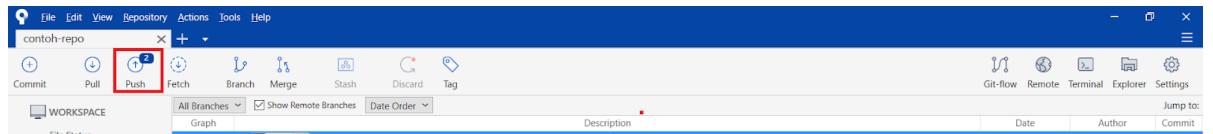
- Kemudian lakukan commit. Kembali ke SourceTree, pada bagian Unstaged Files, klik file a.txt, lalu klik tombol Stage Selected.



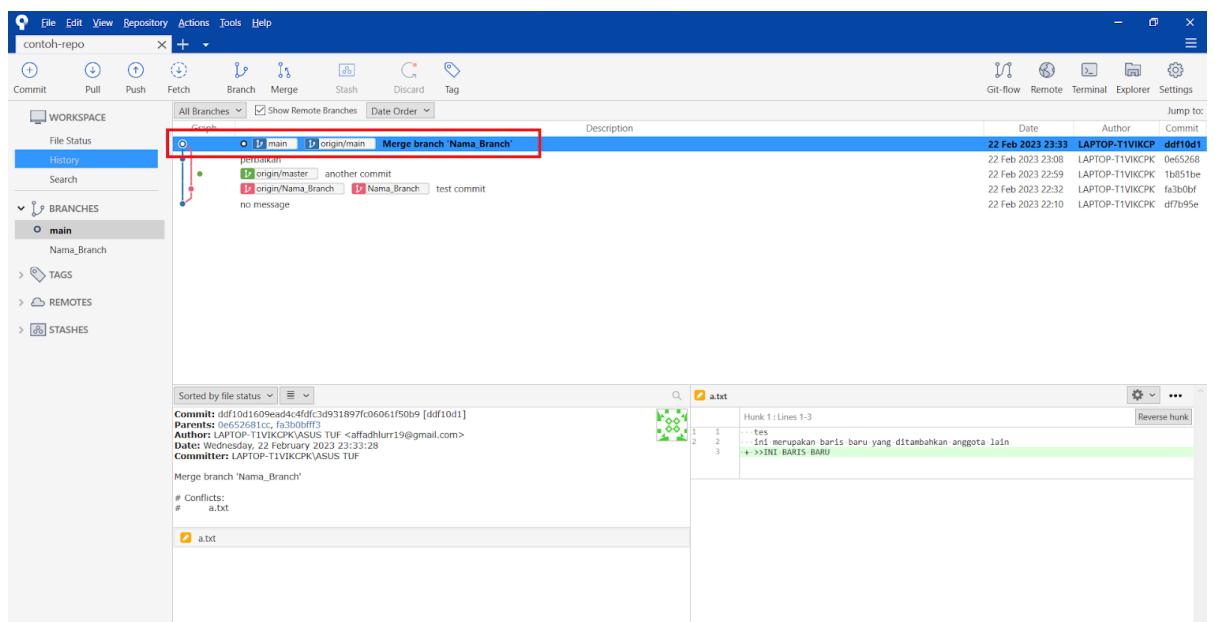
8. Klik tombol Commit pada menu kiri atas. Maka akan otomatis commentnya sudah dibuatkan, lalu klik tombol commit.



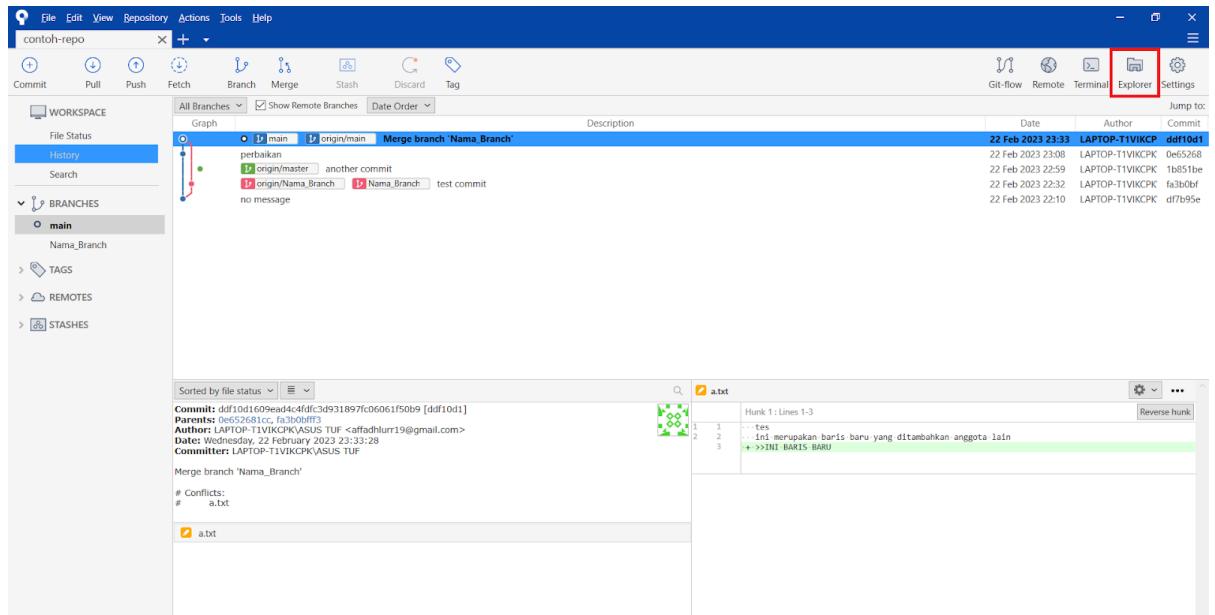
9. Kemudian melakukan push. Klik tombol push pada menu atas.



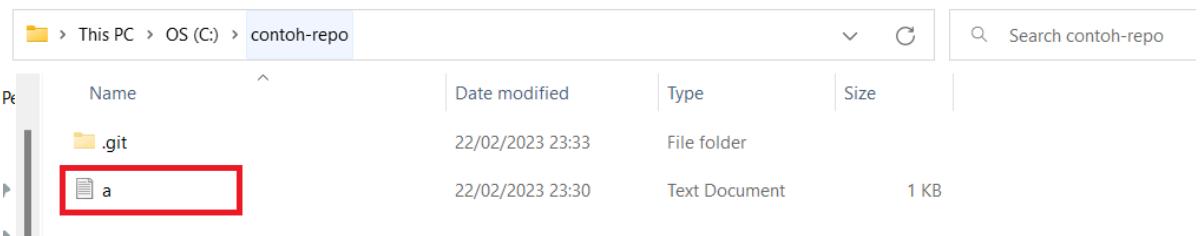
10. Jika proses push berhasil, maka akan menampilkan notice di history graph "Merge branch 'Nama\_Branch'".



11. Untuk mengecek apakah berhasil melakukan merge antara branch Nama\_Branch dan branch master, klik Explorer.



12. Maka di arahkan ke direktori lokal, buka file a.txt.



File a.txt yang dikerjakan secara bersamaan pada dua branch yang berbeda,



berhasil digabungkan.

```
a - Notepad
File Edit View
tes
ini merupakan baris baru yang ditambahkan anggota lain
>>INI BARIS BARU
```

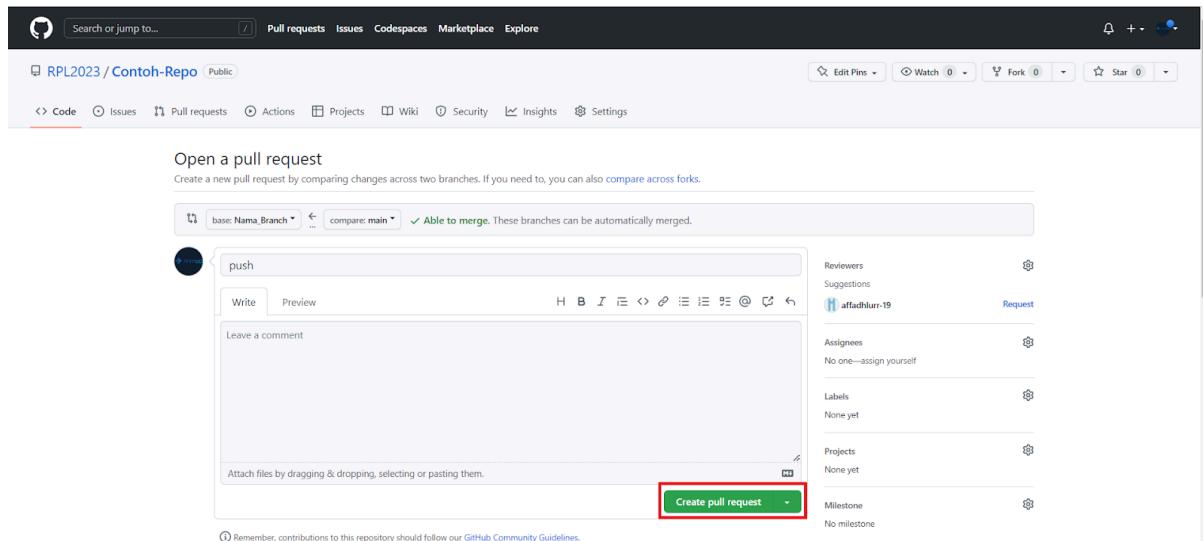
#### e. Membuat Pull Request

Pull Request adalah istilah yang bisa kita artikan sebagai permintaan untuk menggabungkan kode. Setelah membuat branch, teman-teman dapat pindah (checkout) ke branch yang baru dibuat lalu melakukan kontribusi seperti commit & push file yang dikerjakan ke branch masing-masing. Jika teman-teman sudah melakukan kontribusi di branch masing-masing, lalu ingin menggabungkan dengan branch utama (master). Maka teman-teman dapat membuat Pull Request.

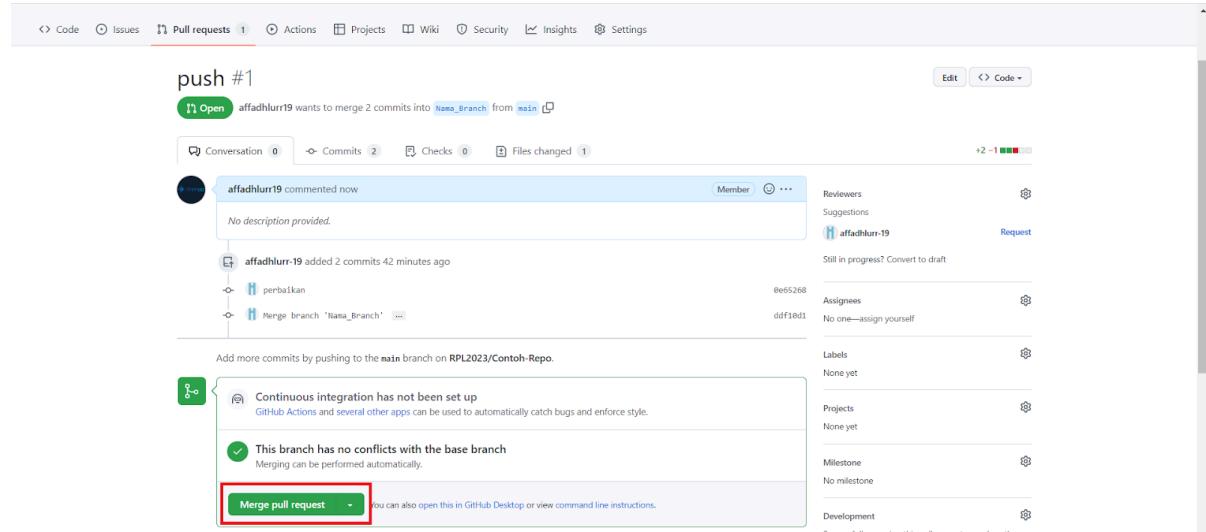
1. Setelah push dilakukan, buka kembali repository GitHub di browser. Maka akan terdapat tombol seperti di bawah ini.



2. Silakan klik tombol Compare & pull request tersebut, lalu akan di redirect ke halaman Open a pull request. Pada halaman ini silakan masukkan judul pull request dan deskripsi (opsional), setelah itu silahkan klik Create pull request.

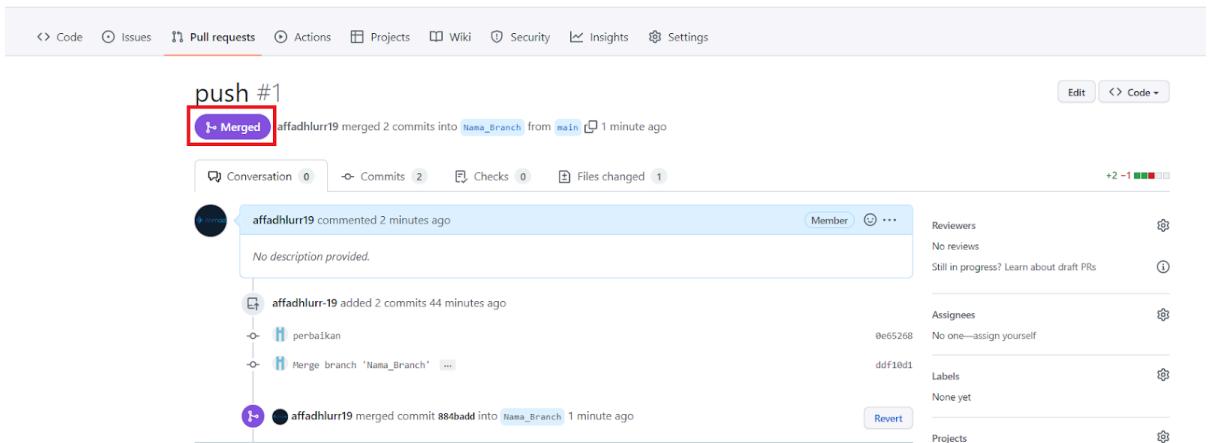


3. Setelah itu, GitHub akan melakukan komparasi. Apakah ada kode yang bentrok atau tidak? Kalau tidak ada yang bentrok biasanya akan muncul tulisan “This branch has no conflicts with the base branch”. Kemudian klik tombol “Merge pull request”.



The screenshot shows a GitHub pull request page for a repository named 'RPL2023/Contoh-Repo'. The pull request is titled 'push #1' and is from the user 'affadhlur19' into the 'main' branch. The status bar indicates 'affadhlur19 wants to merge 2 commits into Nama\_Branch from main'. Below the title, there's a comment from 'affadhlur19' stating 'No description provided.' Under the commit list, it says 'affadhlur-19 added 2 commits 42 minutes ago' with two commits: 'perbaikan' and 'Merge branch "Nama\_Branch"'. A note at the bottom left says 'Add more commits by pushing to the main branch on RPL2023/Contoh-Repo.' On the right side, there are sections for 'Reviewers', 'Suggestions', 'Assignees', 'Labels', 'Projects', 'Milestone', and 'Development'. At the bottom, there's a large green button labeled 'Merge pull request' with a dropdown arrow, which is highlighted with a red box. Below it, a smaller note says 'You can also open this in GitHub Desktop or view command line instructions.'

4. Lalu melakukan merge, klik tombol “**confirm merge**” dan jika berhasil akan ada tulisan “**Merged**” berwarna ungu.



The screenshot shows the same GitHub pull request page after the merge. The title now includes a purple box with the text '+ Merged'. The status bar now says 'affadhlur19 merged 2 commits into Nama\_Branch from main 1 minute ago'. The commit list shows the same two commits: 'perbaikan' and 'Merge branch "Nama\_Branch"'. A new commit is listed at the bottom: 'affadhlur19 merged commit 884bad1 into Nama\_Branch 1 minute ago'. The right sidebar remains the same with various project management options.

Maka pengeraan yang dilakukan di branch telah berhasil digabungkan ke branch utama (master/main).

## Referensi

Modul 1 GIT. 2022. Praktikum Rekayasa Perangkat Lunak, Laboratorium Enterprise Application Development

Modul 1 Version Control System. 2023. Praktikum Rekayasa Perangkat Lunak, Laboratorium Enterprise Application Development