



JOBSHEET 4

Version Control System

1. Tujuan

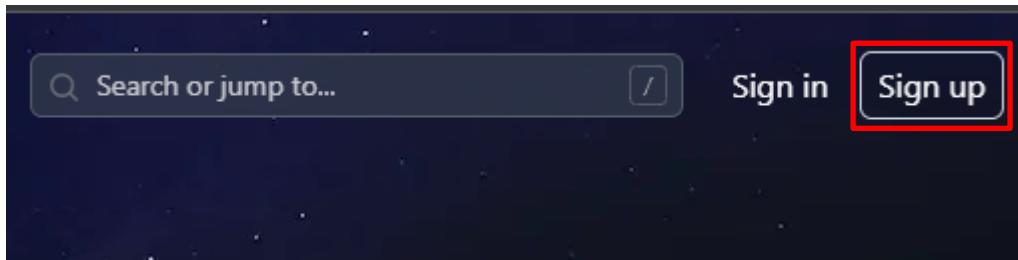
- Mahasiswa mampu menggunakan Git
- Mahasiswa mampu berkolaborasi dengan menggunakan Github

2. Praktikum

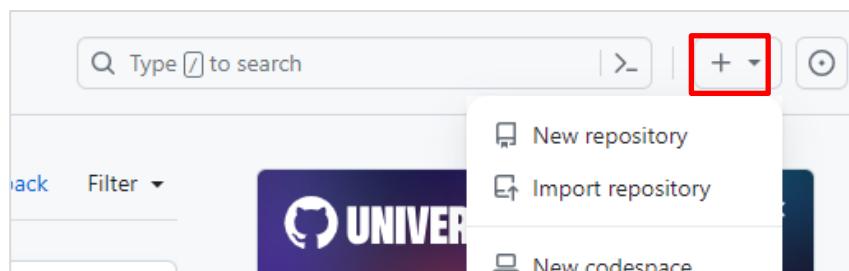
2.1 Percobaan 1: Menggunakan Github

Waktu percobaan : 60 menit

1. Buka situs web GitHub di <https://github.com>.
2. Klik tombol "Sign up" (Daftar) untuk membuat akun GitHub.



3. Ikuti petunjuk untuk mengisi informasi yang diperlukan dan lakukan verifikasi via email.
4. Setelah masuk ke akun GitHub Anda, klik tombol "+", lalu pilih "New repository" (Repositori baru).

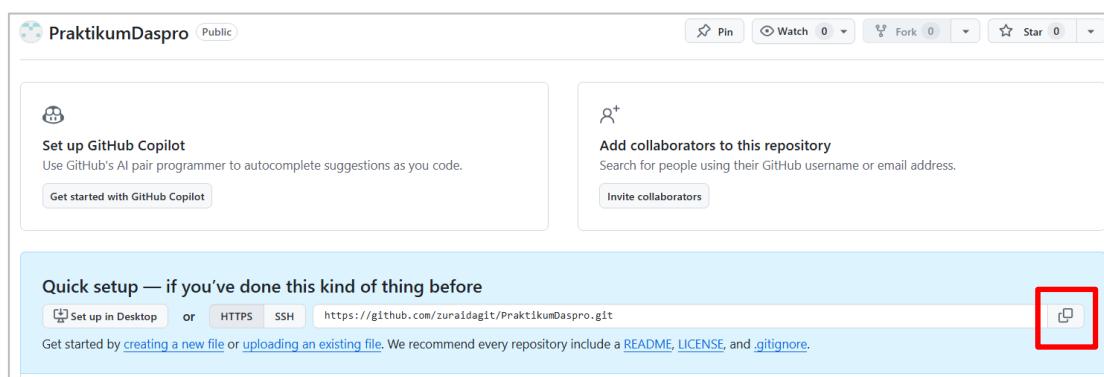




5. Isi **nama repositori, deskripsi (opsional), dan konfigurasi lainnya**. Anda dapat membuat repository public atau private sesuai kebutuhan. Pada praktikum ini, pilih visibility **Public**

The screenshot shows the GitHub repository creation interface. In the General section, the owner is set to 'zuraidagit' and the repository name is 'PraktikumDaspro'. A note says 'PraktikumDaspro is available.' Below it, there's a description field containing 'Repository pertama' with 18 characters out of 350. In the Configuration section, the visibility is set to 'Public'. Other options like 'Add README', 'Add .gitignore', and 'Add license' are shown with their respective settings.

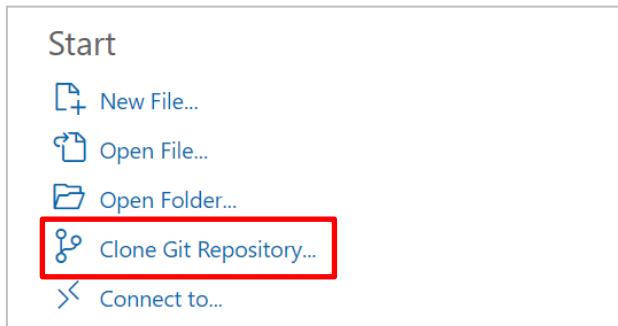
6. Klik tombol “Create repository”.
7. Repository baru telah dibuat di Github. Copy Url repository dengan click icon **Copy** yang nantinya akan digunakan dalam proses cloning.



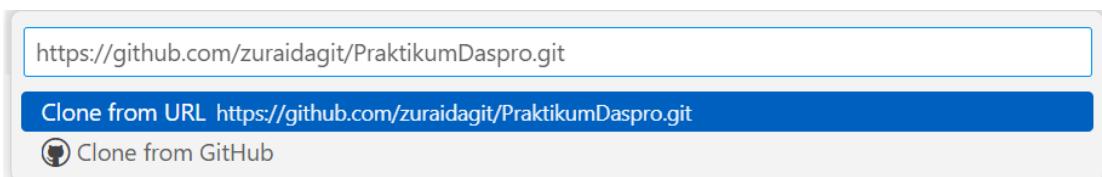
8. Untuk mengelola repositori secara lokal, Anda perlu meng-clone repository yang ada di Github ke komputer Anda. Lakukan installasi **git client** terlebih dahulu. Git client dapat didownload di <https://git-scm.com/downloads>



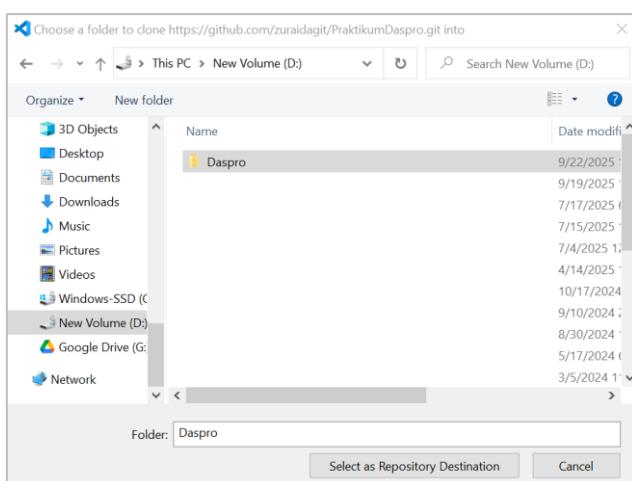
9. Untuk melakukan clone, buka VS Code. Pilih **Clone Git Repository**



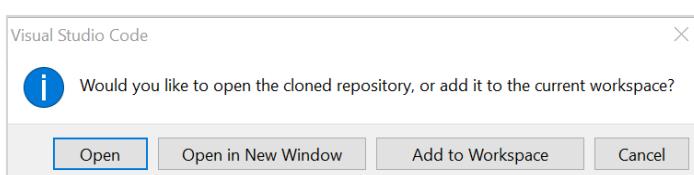
10. Paste repository Url dari Github, lalu enter



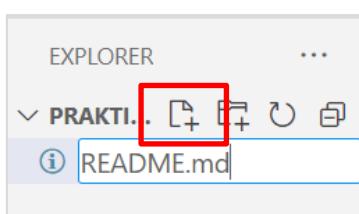
11. Pilih "Repository Destination"



12. Pilih Open atau Open in New Window



13. Anda dapat membuat file baru atau memodifikasi file pada repository sesuai kebutuhan. Lakukan percobaan dengan menambahkan file baru "**README.md**".



14. Isikan file “**README.md**” dengan kalimat seperti berikut. Perhatikan bahwa icon **U** menunjukkan bahwa file tersebut “Untracked” atau belum di-track oleh Git.

The screenshot shows the VS Code interface with the Explorer sidebar open. In the main editor area, there is a file named "README.md" with the content "Hello Github, ini adalah repository pertama saya.". To the left of the file name, there is a small blue icon with a white letter "U", indicating that the file is untracked. A red box highlights this "U" icon.

15. Untuk menyimpan perubahan di local repository, langkah pertama adalah melakukan **Add**. Proses Add berarti menambahkan file ke **staging area** yaitu tempate sementara di mana perubahan disiapkan sebelum disimpan permanen ke local repository melalui **Commit**. Dengan kata lain, **Add** berarti Anda sedang memilih perubahan mana saja yang akan di-commit.
16. Tambahkan file Test.txt sebagai perbandingan.

The screenshot shows the VS Code interface with the Explorer sidebar open. There are two files listed: "README.md" and "Test.txt". Both files have a small blue icon with a white letter "U" to their left, indicating they are untracked. The content of "Test.txt" is "Hello World, ini file random saja." A red box highlights the "U" icon next to "Test.txt".

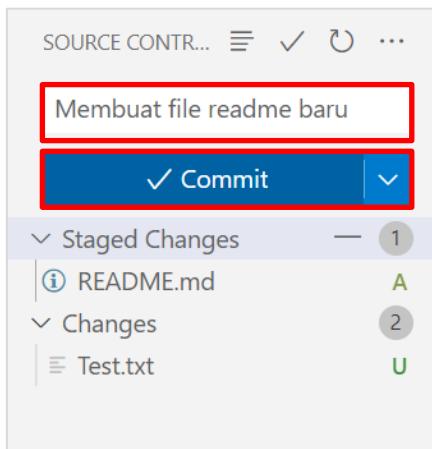
17. Pilih icon Source Control di sebelah kiri lalu klik icon + untuk **Add** file **README.md**. Biarkan file **Test.txt** tetap Untracked.

The screenshot shows the VS Code interface with the Source Control sidebar open. In the main editor area, there is a "Changes" section. Under "Changes", there are two items: "README.md" and "Test.txt". The "README.md" item has a small blue icon with a white plus sign "+" to its right, indicating it is ready to be added. A red box highlights this plus sign icon. The "Test.txt" item also has a similar icon but is not highlighted.

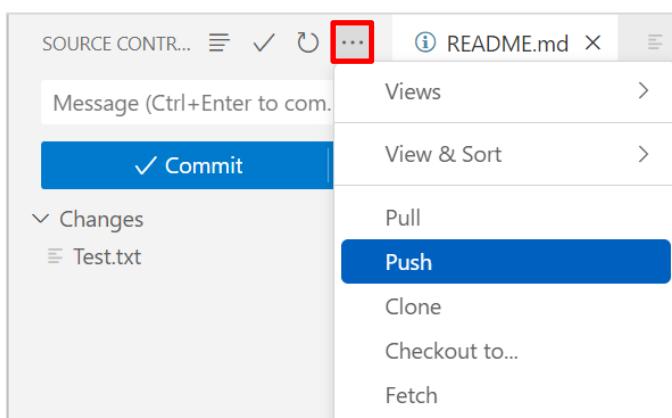
18. Status file **README.md** akan berubah menjadi **A** atau **Added**

The screenshot shows the VS Code interface with the Source Control sidebar open. In the main editor area, there is a "Staged Changes" section. Under "Staged Changes", there is one item: "README.md", which is marked with a small blue icon containing a white letter "A", indicating it is now staged for commit. The "Test.txt" item is still present but not highlighted.

19. Untuk menyimpan perubahan (membuat version baru) di local repository, isi **Commit Message** kemudian klik **Commit**. Biasakan mengisi commit message dengan catatan yang jelas, misalnya mengenai perubahan yang dilakukan.



20. Pada tahap ini, version baru telah dibuat tetapi hanya di local repository Anda. Untuk memperbarui repository di GitHub dengan versions yang sudah Anda simpan di local repository, lakukan **Push**.



21. Saat melakukan push ke repository, jika Git meminta username atau password, silakan gunakan username dan password Git Anda.

Jika muncul error seperti berikut:

```
remote: Permission to <username>/<nama-repository>.git denied
fatal: unable to access 'https://github/<username>/<nama-repository>.git': The requested URL returned
error: 403
```

maka yang perlu dilakukan adalah membuat token untuk push ke repository. **Jika error tidak muncul, silakan loncat ke langkah 25.**

22. Untuk membuat token, klik akun Anda di Github → Settings → Developer Settings → Personal access tokens → Tokens (classic) → Generate a personal access token. Isi Note, Expiration, dan Select scopes. Jika sudah, klik tombol **Generate token**. **Simpan token tersebut karena tidak bisa dilihat kembali untuk digunakan pada push yang selanjutnya.**



Note

What's this token for?

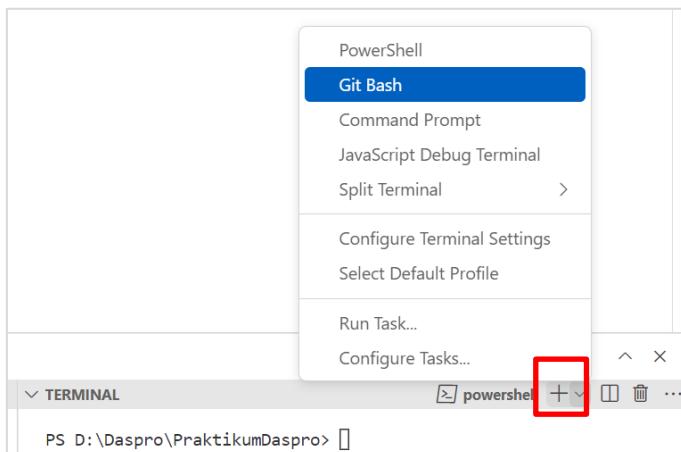
Expiration *

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

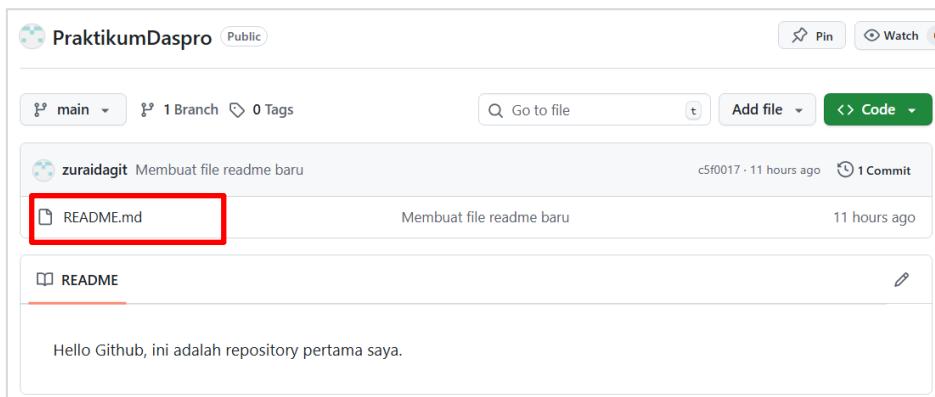
<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events

23. Buka terminal Git Bash

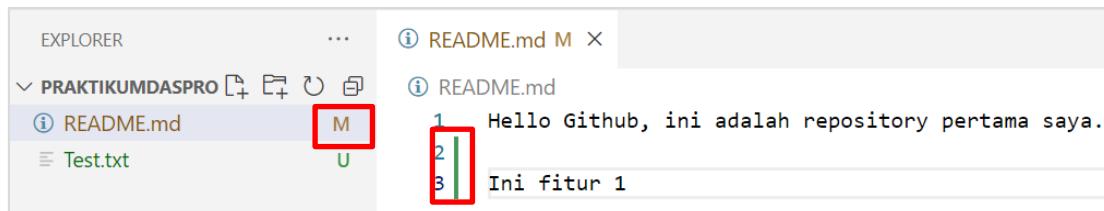


24. Di terminal git bash, jalankan command:

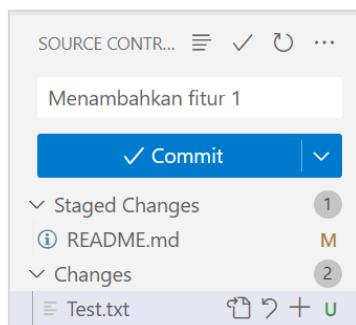
**git push [### 25. Cek halaman Github Anda. File README.md yang dibuat di local sekarang telah tersedia di repository Github. Klik filename untuk menampilkan isi file.](https://<token>@github.com/<username>/<nama-repository>.git</p>
</div>
<div data-bbox=)**



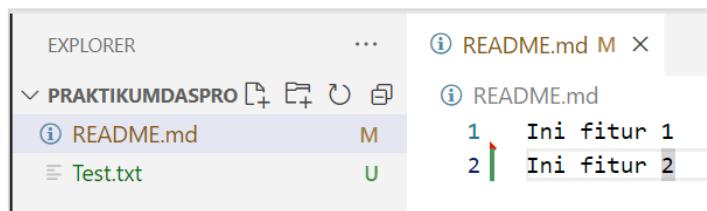
26. Jika Git meminta Anda untuk melakukan login setiap kali **push**, gunakan command:
- ```
git config --global user.email "<email@example.com>"
git config --global user.name "<username>"
```
27. Untuk memperjelas Add, Commit, dan Push. Coba lakukan perubahan pada file README.md lagi. Jika file di-Save, maka status dari file README.md akan berubah jadi **M** atau **Modified** artinya terdapat perubahan pada file, tapi belum masuk ke staging Area (belum di-Add). Baris yang baru ditambahkan ditandai dengan line berwarna hijau pada line number.



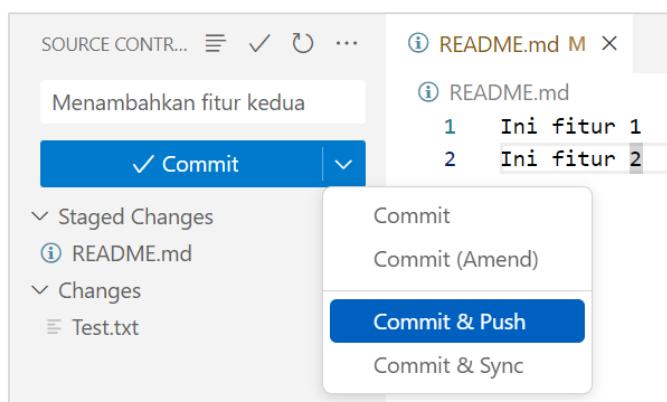
28. Lakukan Add dan Commit file README.md dengan cara yang sama seperti sebelumnya.



29. Lakukan perubahan lagi pada file README.md.



30. Lakukan Add dan Commit kembali. Kali ini, coba lakukan **Commit dan Push** sekaligus. Commit sekaligus Push sebaiknya hanya dilakukan jika Anda mengelola repository pribadi (tidak berkolaborasi dengan orang lain)



31. Cek Github repository, pilih link pada kotak merah untuk menampilkan riwayat commit.

The screenshot shows a GitHub repository named 'PraktikumDaspro' which is public. It has one branch ('main') and no tags. There is one file named 'README.md'. A commit was made by 'zuraidagit' titled 'Menambahkan fitur kedua' at 3491638, 2 minutes ago. A red box highlights the '3 Commits' link in the top right corner of the commit list.

32. Anda dapat melihat setiap version yang dibuat setiap kali Commit dilakukan.

The screenshot shows the commit history for the 'main' branch. It lists three commits:

- Commits on Sep 23, 2025:
  - Menambahkan fitur kedua** (zuraidagit committed 4 minutes ago) - Hash: 3491638
  - Menambahkan fitur 1** (zuraidagit committed 8 minutes ago) - Hash: d4745bc
- Commits on Sep 22, 2025:
  - Membuat file readme baru** (zuraidagit committed 12 hours ago) - Hash: c5f0017

33. Silakan kembali ke VS Code dan coba lakukan perubahan lagi pada file README.md seperti contoh berikut:

The screenshot shows the 'README.md' file in VS Code. The content of the file is:

```
1 Ini fitur 1
2
3 Ini fitur 3
```

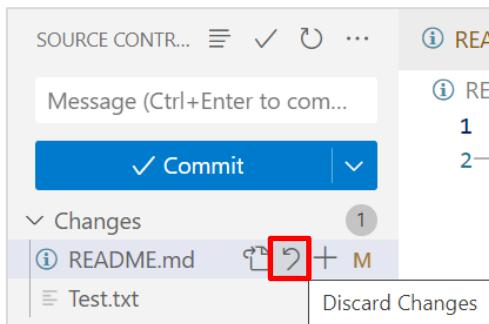
The second line '2' is highlighted in blue, indicating it is currently selected or being edited.

34. Untuk membandingkan perubahan yang baru dilakukan dengan commit terakhir, klik nama file pada Source Control maka akan ditampilkan perbandingannya. Baris berwarna merah adalah baris yang dihapus sementara baris berwarna hijau adalah baris yang ditambahkan.

The screenshot shows the Source Control view in VS Code. The 'Changes' tab is selected, showing a diff view for the 'README.md' file. The diff highlights changes made to the file:

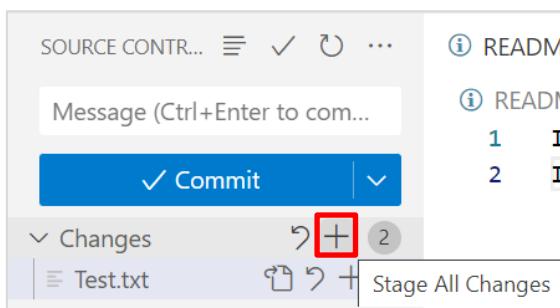
- Line 2 is marked with a red background, indicating it was deleted.
- Line 3 is marked with a green background, indicating it was added.

35. Misalnya Anda ingin membatalkan perubahan ke versi commit terakhir, klik icon Undo. Pilih **Discard Changes** jika muncul confirmation pop up.



## Pertanyaan

1. Cek apakah file Test.txt ada di Github. Jika tidak, mengapa?
2. Jelaskan fungsi dari Add, Commit, dan Push
3. Add Test.txt. Untuk Add semua perubahan yang dilakukan, Anda bisa menggunakan icon + di sisi kanan **Changes**. Lakukan **Stage All Changes** kemudian **Commit & Push**.



## 2.2 Percobaan 2 : Dasar Kolaborasi di Github

### Waktu percobaan : 30 menit

Selain sebagai repository pribadi, Github sering dimanfaatkan sebagai sarana kolaborasi untuk mengerjakan project bersama dengan tim. Lakukan percobaan berikut untuk melihat bagaimana kolaborasi dilakukan dengan Github.

1. Perlu diperhatikan bahwa ketika Anda berkolaborasi dengan tim pada sebuah project, lakukan **Pull** untuk memperbarui versi project pada local repository Anda dengan versi terbaru di Github. Best practice nya, lakukan Pull setiap Anda mau mulai bekerja dan setiap kali akan melakukan Push.



2. Sebagai contoh, lakukan perubahan pada file README.md di Github. Klik pada nama file.

The screenshot shows a GitHub repository interface. At the top, it says 'PraktikumDaspro' (Public). Below that, there are buttons for 'main' (with 1 Branch and 0 Tags), 'Go to file', 'Add file', and 'Code'. A search bar is also present. The file list includes 'README.md' (by zuraidagit, 3 minutes ago), 'Test.txt' (3 minutes ago), and another 'README.md' entry (Menambahkan file Test, 6 hours ago).

3. Klik icon Edit

The screenshot shows the GitHub editor for the 'README.md' file. It has tabs for 'Preview', 'Code', and 'Blame'. On the right, there are buttons for 'Raw', 'Copy', 'Delete', and 'Edit' (which is highlighted with a red box). The content of the file is shown as:

```
Ini fitur 1 Ini fitur 2
```

4. Ubah file kemudian **Commit changes**

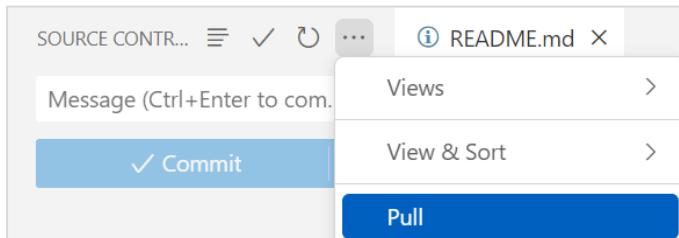
The screenshot shows the 'Commit changes' dialog for the 'README.md' file. It includes fields for 'Commit message' (containing 'Update README.md') and 'Extended description'. At the bottom, there are radio buttons for committing directly to the 'main' branch or creating a new branch, with the first option selected. There are also 'Cancel changes' and 'Commit changes...' buttons.

5. Klik **Commit changes**

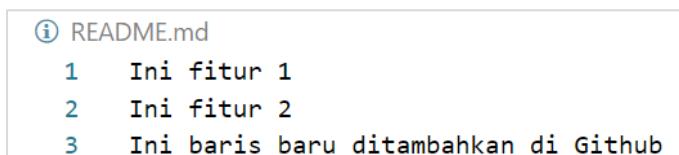
The screenshot shows the 'Commit changes' dialog again. The 'Commit message' field contains 'Update README.md'. The 'Commit changes...' button at the bottom is highlighted with a red box.



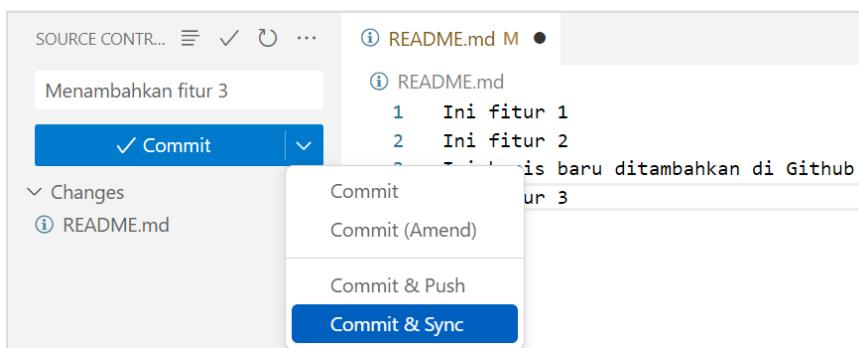
6. Untuk mengambil/mengunduh perubahan di Github, klik More (...) → **Pull**.



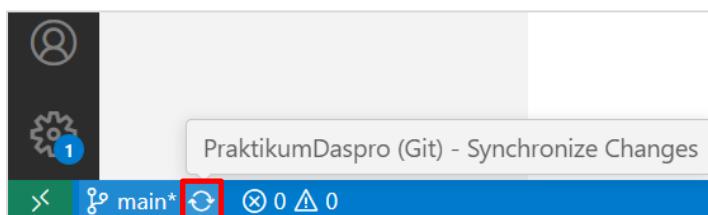
7. Setelah dilakukan Pull maka file README.md di local repository juga akan diupdate.



8. Anda juga dapat memilih opsi Commit & Sync. Sync berarti melakukan sinkronisasi local repository dan Github, jadi akan dilakukan Pull dari Github kemudian dilakukan Push ke Github secara langsung. Untuk mencobanya, tambahkan baris baru di README.md kemudian lakukan **Add** lalu **Commit & Sync**.



9. Opsi lainnya adalah menggunakan icon **Sync** di status bar.



10. Sync sebaiknya juga hanya dilakukan jika repository dikelola secara pribadi. Jika bekerja dengan tim, lakukan **Pull** dan **Push** secara terpisah untuk meng-handle konflik yang mungkin terjadi dengan lebih baik.



## Tugas

### Waktu pengerjaan : 60 menit

- Untuk mencoba melakukan kolaborasi, buat repository baru dengan nama FinalProject<NoAbsen>.
- Lakukan cloning ke local repository seperti pada Percobaan 1.
- Cobalah tambahkan file baru misalnya Test.java yang berisi fungsi main().

```

EXPLORER ... Welcome
FINALP... J Test.java U
J Test.java U
1 public class Test {
2 Run | Debug
3 public static void main(String[] args) {
4 System.out.println("Hi, ini baris kode pertama");
5 }

```

- Lakukan Add, Commit, dan Push.
- Anda dapat berkolaborasi dengan anggota tim dengan pilih Settings → Collaborators → Add people

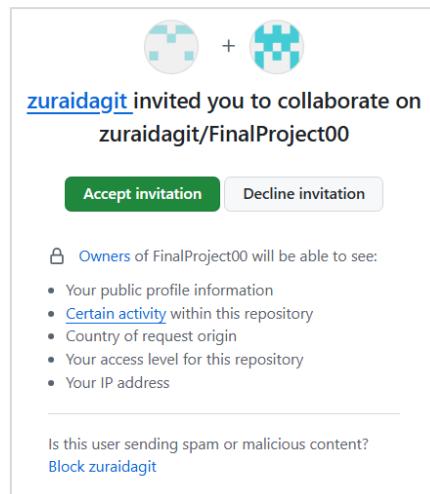
The screenshot shows the GitHub repository settings page for 'zuraidagit / FinalProject00'. The 'Settings' tab is active. In the left sidebar, the 'Collaborators' tab is highlighted with a red box. At the bottom right of the main area, there is a button labeled 'Add people' with a red box around it.

- Sebagai contoh, tambahkan 1 teman Anda berdasarkan username Github mereka. Invite collaborator kemudian Add.

The screenshot shows the 'Add people to FinalProject00' dialog box. It includes a search bar with 'dasproclass' typed in, a result card for 'dasproclass' with the 'Invite collaborator' button, and 'Cancel' and 'Add to repository' buttons at the bottom.



7. Minta teman Anda untuk membuka email yang dikirimkan Github. Klik link “Accept or decline this invitation”. Pilih “Accept invitation”.



8. Minta teman Anda untuk meng-clone repository tersebut ke laptop/PC nya kemudian lakukan perubahan pada file Test.java atau menambahkan file baru. Lakukan Add, Commit, dan Push agar perubahan yang dia lakukan tersimpan di repository Github.
9. Lakukan Pull agar local repository Anda juga menyimpan versi project terbaru.
10. Bekerjalah bergantian dengan meng-clone repository teman Anda kemudian lakukan beberapa percobaan lagi dengan github (add, commit, push, pull) agar Anda terbiasa. Screenshot riwayat commit dari Github.

### Catatan: Merge Conflict

Merge Conflict adalah masalah yang terjadi saat Git tidak bisa menggabungkan (merge) dua version secara otomatis, misalnya saat dua orang melakukan perubahan pada baris yang sama (atau berdekatan) pada sebuah file. Di samping itu, merge conflict juga bisa terjadi saat perubahan dilakukan pada file yang ternyata dihapus oleh orang lain.

1. Sebagai contoh, misalnya Test.java diubah oleh User1 sebagai berikut.

```
public class Test {
 Run | Debug
 public static void main(String[] args) {
 System.out.println("Hi, ini baris kode pertama yang dimodifikasi User 1");
 System.out.println("User 1 menambahkan baris baru");
 }
}
```

2. User2 juga mengubah file Test.java (terhadap version yang sama) dan sudah di-push ke Github.

```
public class Test {
 Run | Debug
 public static void main(String[] args) {
 System.out.println("Hi, ini baris kode pertama yang dimodifikasi User 2");
 System.out.println("User 2 menambahkan baris baru");
 }
}
```

3. Saat dilakukan pull akan terjadi merge conflict. Current Change berwarna hijau adalah version di local repository Anda sementara Incoming Change berwarna biru adalah version yang ada di Github.

```

public class Test {
 Run | Debug
 public static void main(String[] args) {
 <<<<< HEAD (Current Change)
 System.out.println("Hi, ini baris kode pertama yang dimodifikasi User 1");
 System.out.println("User 1 menambahkan baris baru");
 =====
 =====
 System.out.println("Hi, ini baris kode pertama yang dimodifikasi User 2");
 System.out.println("User 2 menambahkan baris baru");
 >>>>> bb5cfee23542af8fa934cf0345b9ae6a0f2ebf98 (Incoming Change)
 }
}

```

Resolve in Merge Editor

4. Anda dapat memilih beberapa opsi, yaitu:

- **Accept Current Change:** memilih untuk menyimpan version local repository Anda dan mengabaikan version dari Github
- **Accept Incoming Change:** memilih version yang di Github dan mengabaikan version local repository Anda
- **Accept Both Changes:** menggabungkan kedua versi. Opsi ini biasanya akan menyebabkan error sehingga perlu diedit kembali secara manual
- **Compare Changes:** membandingkan kedua version sebelum memilih.

5. Opsi lainnya adalah **Resolve in Merge Editor**.

```

J TestJava > Test > main(String[])
Incoming φ bb5cfee · origin/main ...
1 class Test {
2 public static void main(String[] args) {
3 System.out.println("Hi, ini baris kode pertama yang dimo
4 System.out.println("User 2 menambahkan baris baru");
5
Current φ a6970ba · main ...
1 class Test {
2 public static void main(String[] args) {
3 Accept Current | Accept Combination (Current First) | Ignore
4 System.out.println("Hi, ini baris kode pertama yang di
5 System.out.println("User 1 menambahkan baris baru");
5
Result Test.java
1 public class Test {
2 public static void main(String[] args) {
No Changes Accepted
3 System.out.println("Hi, ini baris kode pertama");
4 }
5
1 Conflict Remaining ...

```

6. Anda dapat memilih version yang diinginkan pada bagian **Incoming** (klik Accept Incoming | Accept Combination(Incoming First) | Ignore) atau bagian **Current** (klik Accept Current | Accept Combination(Current First) | Ignore) kemudian modifikasi secara manual di bagian **Result** jika diperlukan.



7. Jika merge sudah sesuai, klik **Complete Merge**.

The screenshot shows a code merge interface for a Java file named Test.java. It displays two branches: 'Incoming' (bb5cfee) and 'Current' (a6970ba). The 'Incoming' branch has four lines of code: 'System.out.println("Hi, ini baris kode pertama yang dimodifikasi User 1");', 'System.out.println("User 1 menambahkan baris baru");', 'System.out.println("User 2 menambahkan baris baru");', and a closing brace. The 'Current' branch has three lines of code: 'System.out.println("Hi, ini baris kode pertama yang dimodifikasi User 1");', 'System.out.println("User 1 menambahkan baris baru");', and 'System.out.println("User 2 menambahkan baris baru");'. A 'Result' section shows the merged code with all four lines from the incoming branch. Below the code, there are buttons for 'Manual Resolution' and 'Reset to base'. A red box highlights the 'Complete Merge' button at the bottom right.

```
J Test.java > 🛡 Test > ⚙ main(String[])
Incoming Ⓜ bb5cfee • origin/main ... Current Ⓜ a6970ba • main ...
1 class Test {
2 lic static void main(String[] args) {
3 System.out.println("Hi, ini baris kode pertama yang dimodifikasi User 1");
4 System.out.println("User 1 menambahkan baris baru");
5 System.out.println("User 2 menambahkan baris baru");
6 }
}
Result Test.java
1 public class Test {
2 public static void main(String[] args) {
3 System.out.println("Hi, ini baris kode pertama yang dimodifikasi User 1");
4 System.out.println("User 1 menambahkan baris baru");
5 System.out.println("User 2 menambahkan baris baru");
6 }
}
0 Conflicts Remaining ...
```

**Complete Merge**

8. Lakukan **Commit** kemudian **Push**.