JOBSHEET 2
Version Control System dan Kanban Board

1. Learning Outcome

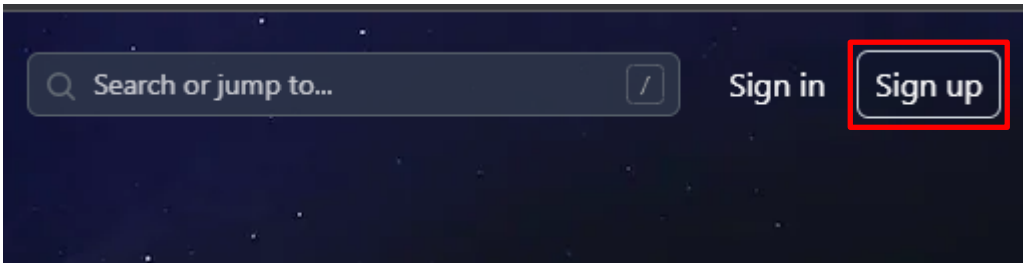
- Students must be able to create a repository account.
- Students must be able to use basic command of GitHub.
- Students can collaborate in a team using Github.
- Students could perform task management using Kanban Board

2. Labs Activity

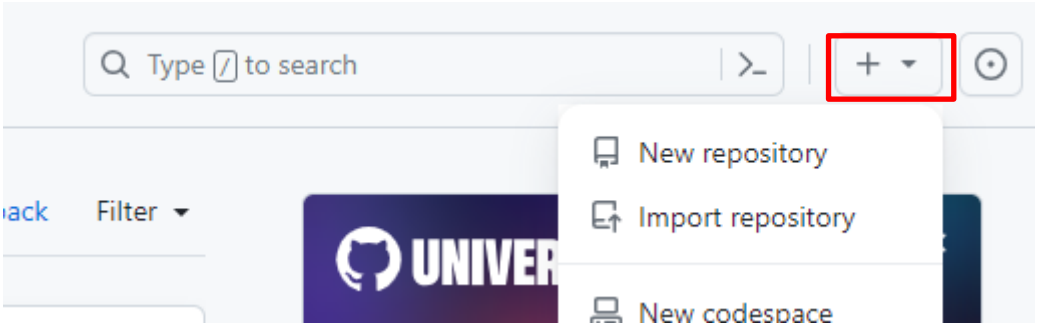
2.1 Experiment 1: Getting Started with Github

Experiment Time: 120 minutes

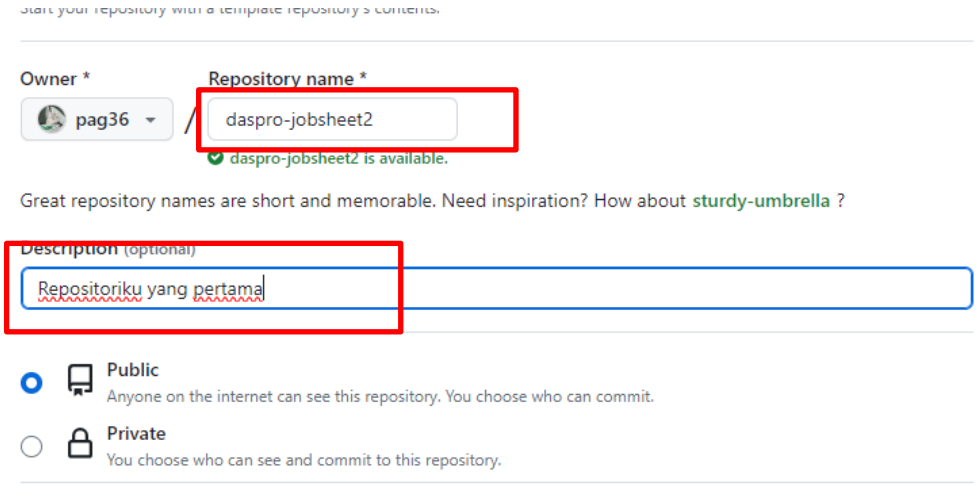
1. Open GitHub at **https://github.com**.
2. Proceed the registration by using **"Sign up"** button.



3. Follow the registration steps by providing information needed and continue with email verification.
4. After finishing the registration, log in to your GitHub account, and now you are ready to create a new repository by clicking **"+"** and **"New repository"**



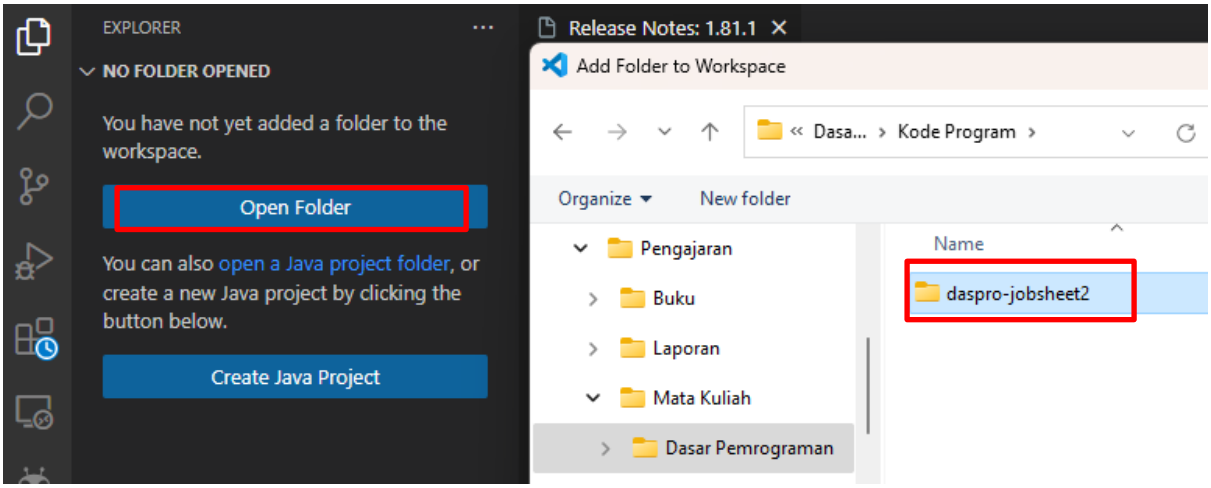
5. Fill in **repository name, description (optional),** and **other needed configurations**.
6. You can set up your repository to be public or private based on your needs. To complete the you can click "**Create repository"**.



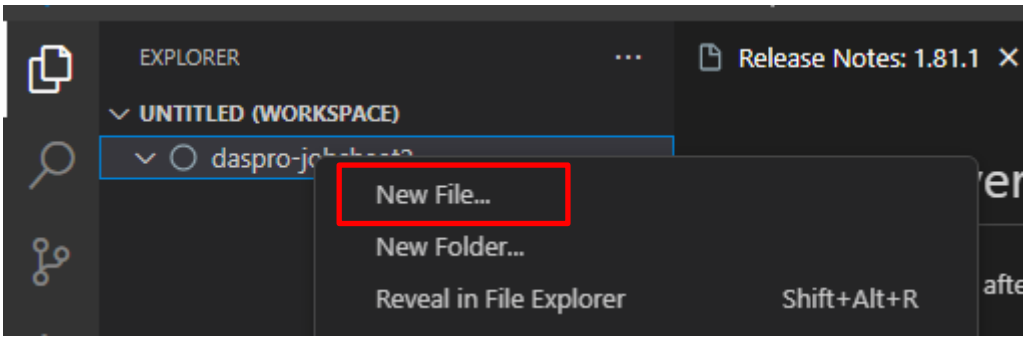
7. The repository that is created in the previous steps, is saved in the Github server. You can manage your repository locally as well. To do so, then you will need to clone the repository into your local computer. First, the Github Client must be installed on your local computer, the installer is provided here https://git-scm.com/downloads, Follow the steps to finish the installation.
8. Use **git clone** command (from the terminal or command line) to clone the repository. It will be **git clone https://github.com/username/nama-repositori.git**

```
POLINEMA@LAPTOP-COBHS463 MINGW64 ~/OneDrive/Tridharma/Pengajaran/Mata Kuliah/Das
ar Pemrograman/Kode Program
$ git clone https://github.com/pag36/daspro-jobsheet2.git
Cloning into 'daspro-jobsheet2'...
warning: You appear to have cloned an empty repository.

POLINEMA@LAPTOP-COBHS463 MINGW64 ~/OneDrive/Tridharma/Pengajaran/Mata Kuliah/Das
ar Pemrograman/Kode Program
$ dir
daspro-jobsheet2

POLINEMA@LAPTOP-COBHS463 MINGW64 ~/OneDrive/Tridharma/Pengajaran/Mata Kuliah/Das
ar Pemrograman/Kode Program
$ cd daspro-jobsheet2/

POLINEMA@LAPTOP-COBHS463 MINGW64 ~/OneDrive/Tridharma/Pengajaran/Mata Kuliah/Das
ar Pemrograman/Kode Program/daspro-jobsheet2 (main)
$
```

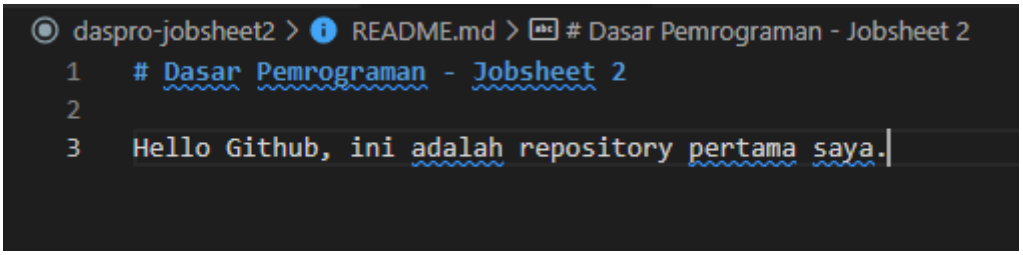
9. Create or **edit files in repository** based on your needs. Open folder repository by using Visual Studio Code.



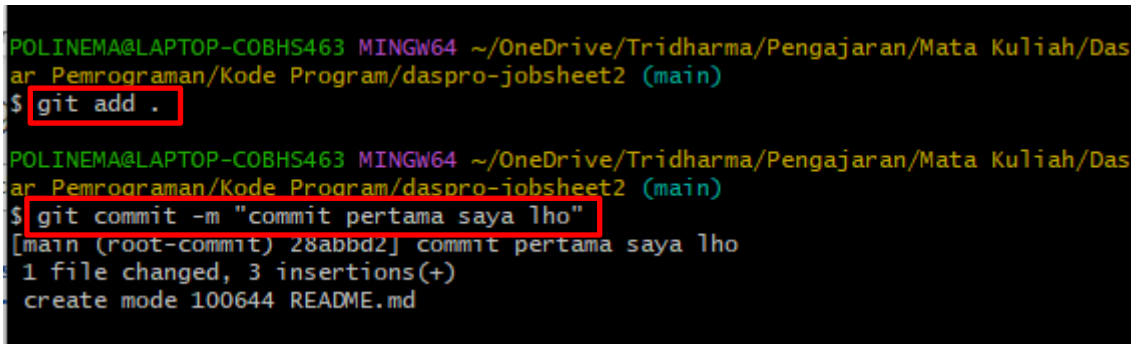
10. Create a new file by **right click – New File**, save the new file as **"README.md"**



11. Open fil **"README.md"** and fill it with the following contents



12. Save the changes and commit it with the command **git commit**. You will be asked to **write the commit message** that indicates the changes made by you.



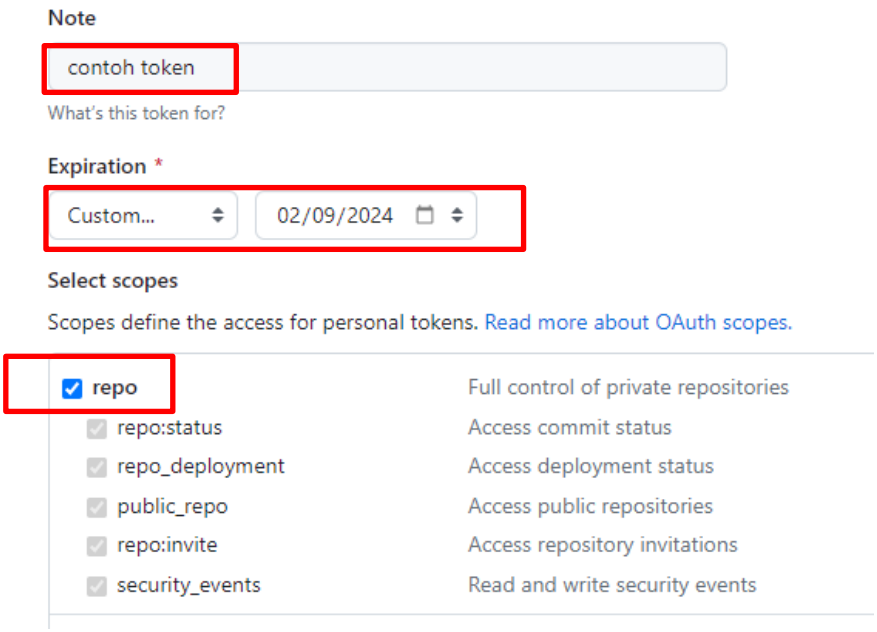
13. To update the repository on GitHub with the changes you have made locally, use the command **git push**.

14. For example, **git push origin nama-branch** will save the changes on your main branch in GitHub.

```
POLINEMA@LAPTOP-COBHS463 MINGW64 ~/OneDrive/Tridharma/Pengajaran/Mata Kuliah/Das
ar Pemrograman/Kode Program/daspro-jobsheet2 (main)
$ git push origin main
remote: Permission to pag36/daspro-jobsheet2.git denied to 0d3ng.
fatal: unable to access 'https://github.com/pag36/daspro-jobsheet2.git/': The re
quested URL returned error: 403
```

Normally, to push to the repository you will be asked for a user or password. But if you experience something like the above, what you need to do is, **create** a **token** for the **push repository**.

15. Click your Account – Settings – Developer Settings – Tokens (classic) – Genereate new token (classic). Fill in Note, Expiration, and Select scopes. Then click **Generate token**.

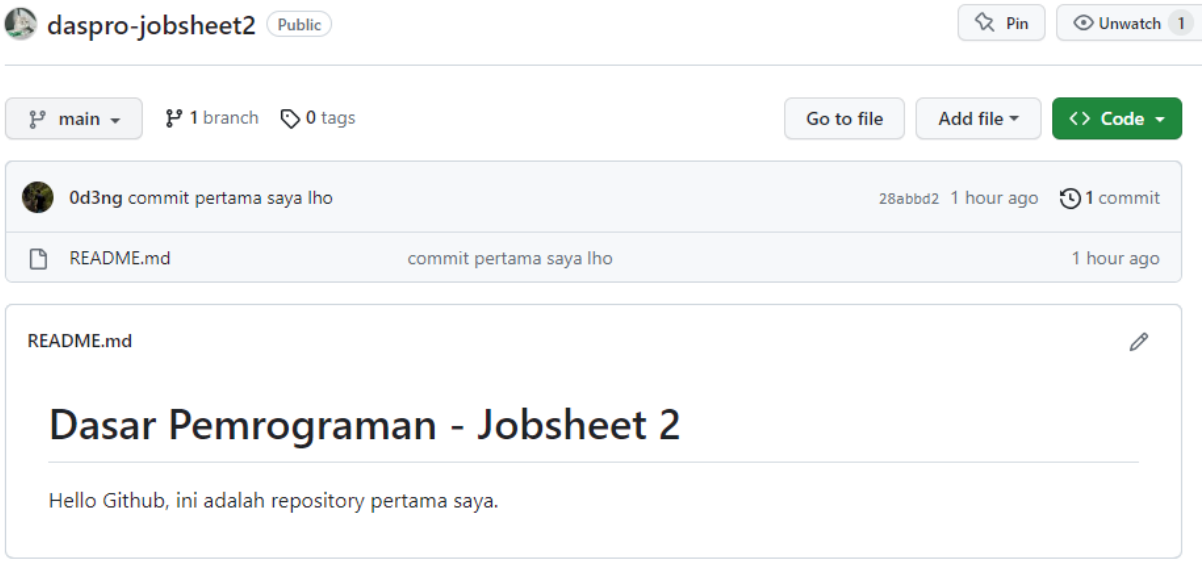


Save the token since it will be no longer visible for the next push

16. Run the command git push https://[token]@github.com/username/nama-repository.git

```
POLINEMA@LAPTOP-COBHS463 MINGW64 ~/OneDrive/Tridharma/Pengajaran/Mata Kuliah/Das
ar Pemrograman/Kode Program/daspro-jobsheet2 (main)
$ git push https://ghp_[redacted]uw19@github.com/pag36/das
pro-jobsheet2.git
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 294 bytes | 294.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/pag36/daspro-jobsheet2.git
 * [new branch]      main -> main
```

17. Then check your Github account (refresh it)



18. In case step no.14 is successfully done, then you do not have to create token, and you can directly go to step no. 17 (check your Github account)

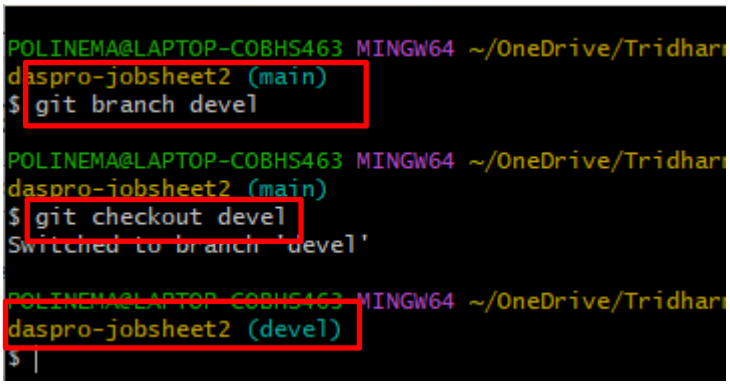
Question

1. Explain the difference between **git commit** and **git push**?
2. In the above steps, we firstly create a repository in Github then clone it to the local computer to be modified. Is it possible to create the project folder and then push (upload) it to Github? Proove it!

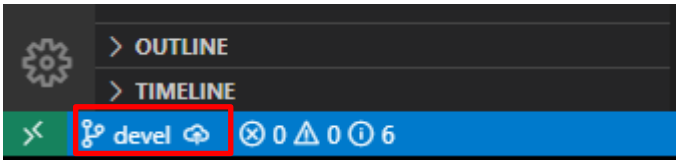
2.2 Experiment 2: Basic Collaboration in Github

Time Experiment: 60 minutes

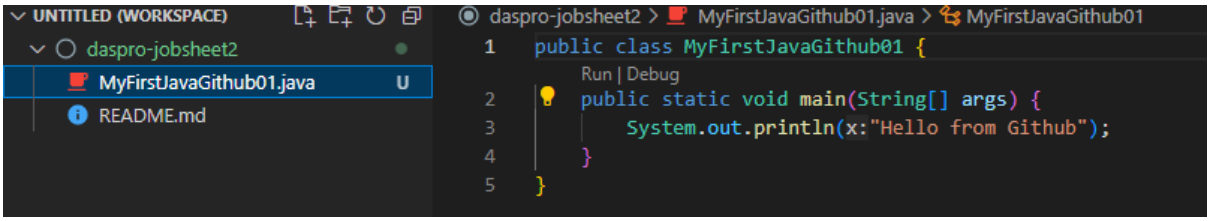
1. Before creating a project, it is recommended to create a branch first to isolate your changes from the main branch (usually "main" or "master").
2. Use command **git branch *branch-name*** to create a new branch and **git checkout *branch-name*** to move to the branch.



In Visual Studio Code it should change to the "devel" branch, otherwise, please click then select the "**devel**" branch.

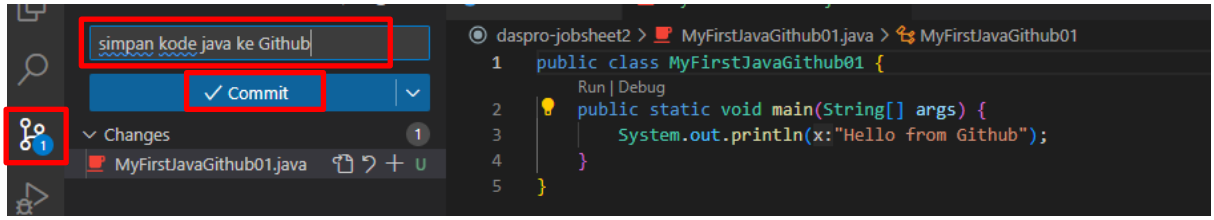


3. Create a new file **MyFirstJavaGithubNoAbsen.java** (use your own number).



Create the basic java program as you already did at **Jobsheet 1**.

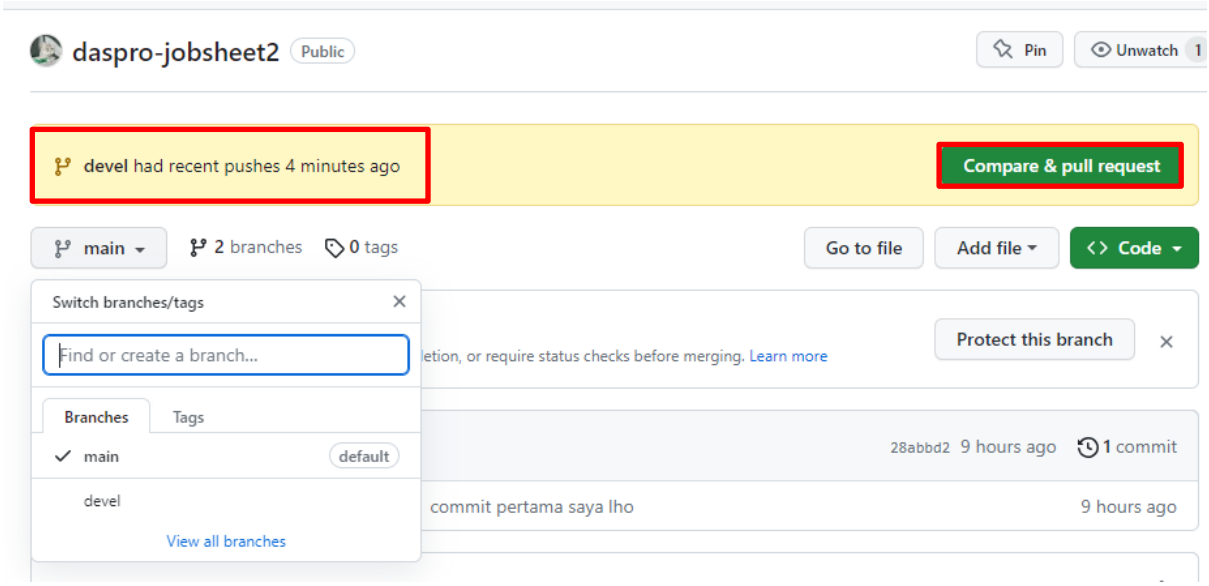
4. Save the changes locally by using command **commit** then **push** it to Github using Visual Studio Code. Don't forget to **set the message** when calling **commit**. The way we do it is, **click branch icon** – **fill in the commit message** – click **Commit button – click Publish Branch** button



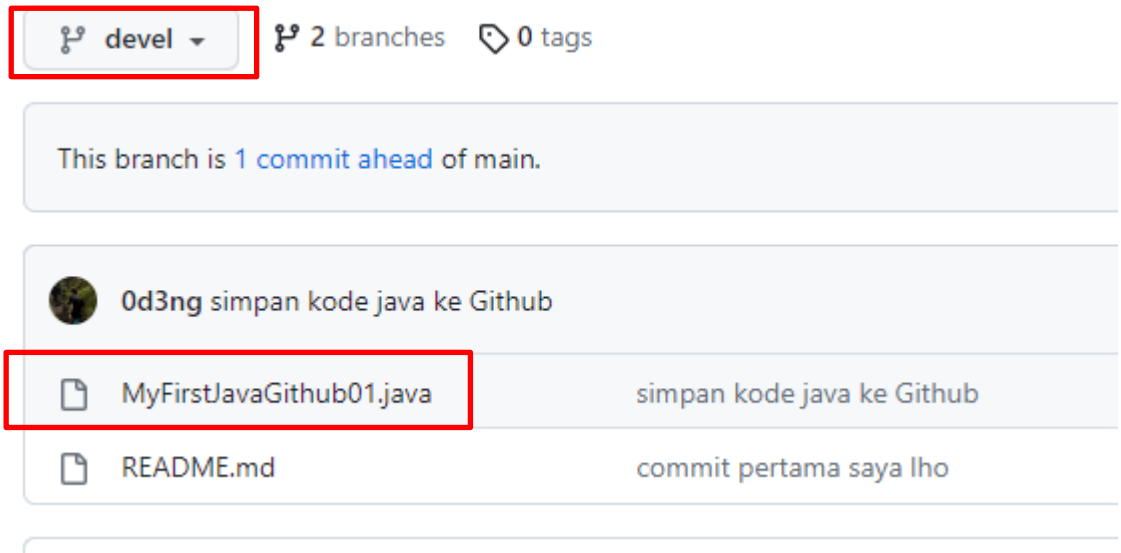
Or you can use command at **git bash** as in the previous session, while the result will be like this.

```
$ git push https://ghp_[redacted]uw19@github.com/pag36/daspro-jobsheet2.git
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 413 bytes | 413.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'devel' on GitHub by visiting:
remote:      https://github.com/pag36/daspro-jobsheet2/pull/new/devel
remote:
To https://github.com/pag36/daspro-jobsheet2.git
 * [new branch]      devel -> devel
```

5. Please go to the Github page, the **devel branch** should appear after pushed successfully in the previous step

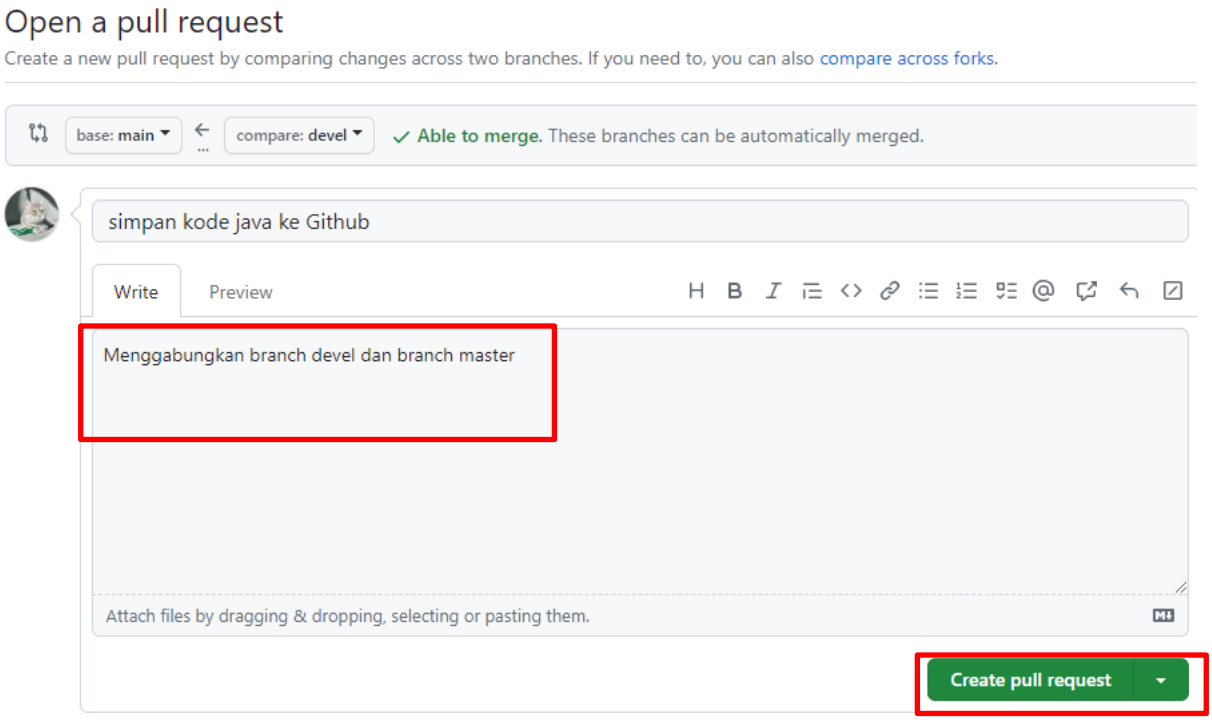


Next you will be able to identify the **main branch** and **devel branch**.

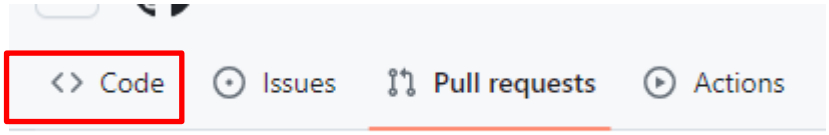


6. Click **Compare & pull request** button, you can choose which branch to merge (**devel** to **master**). Fill in the **message** and click the **Create pull request** button, wait for a few

moments then click the **Merge pull request** button. Finally, click the **Confirm merge** button.



7. Go to **tab Code**, then observe the result at **main branch** dan **devel branch**.



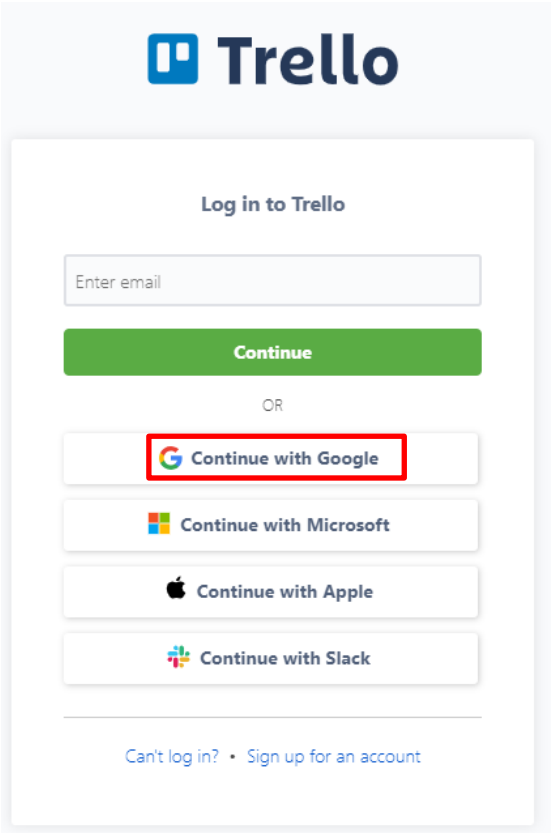
Question!

1. What is the use of **Pull requests**?
2. Why do we need to create **branch**?

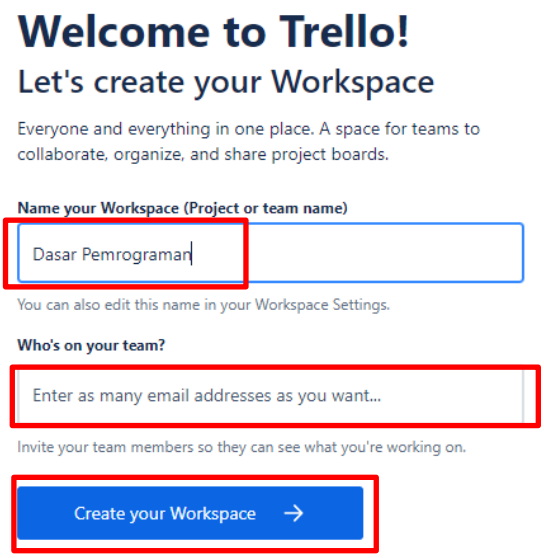
2.3 Experiment 3: Trello

Time Experiment: 45 minutes

1. Open **Trello (https://trello.com/)** and click **"Sign Up"** to start the registration.
2. You can register using your email address or link it with your Google account.

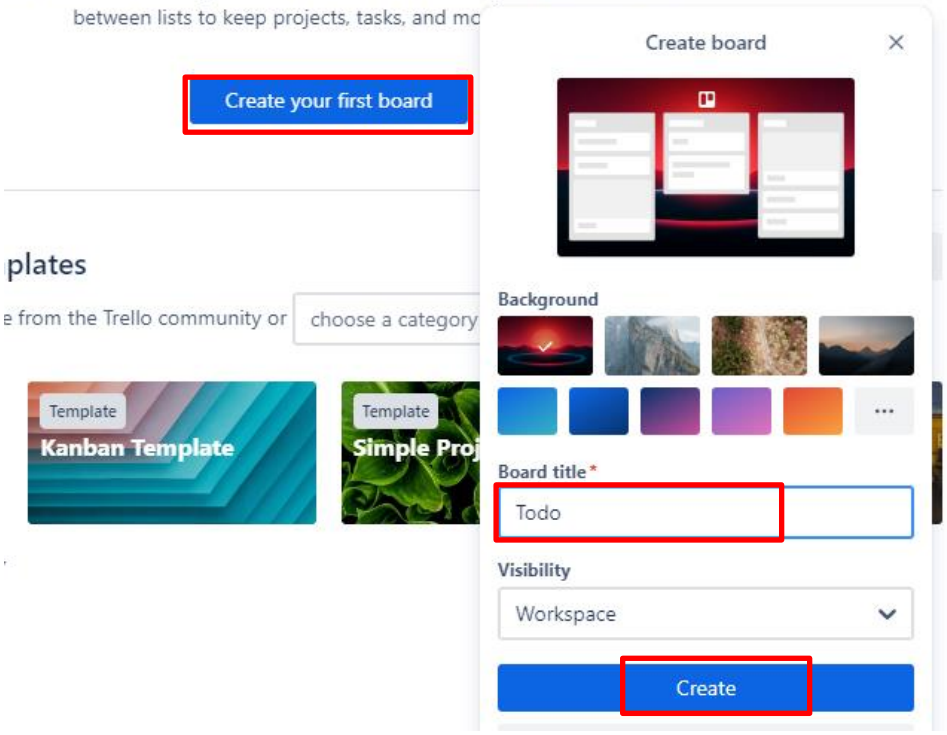


If the registration is already completed, login to your Trello account and open **Workspaces**. Start to create your **workspace** while the team is optional, then click **Create your Workspace**.

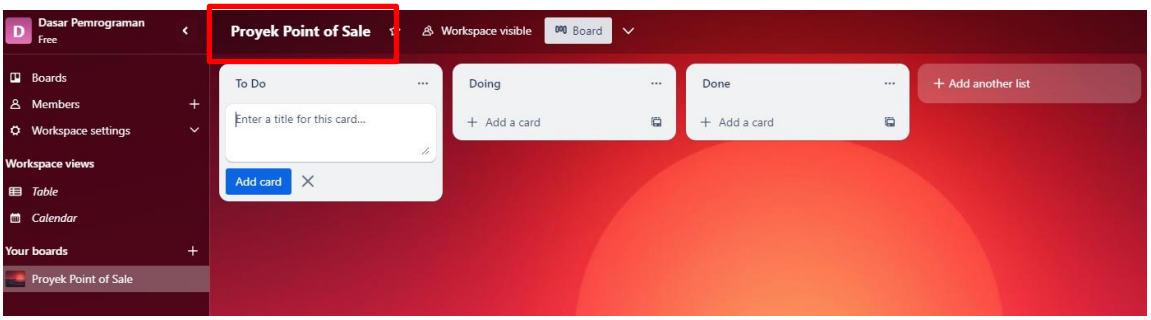


3. Once you are logged in, you will be on the Trello homepage. To create a new Kanban board, click the button **"Create your first board"**.

4. Give your board a title and set the visibility (Public, Private, or Team).



Board title can be the name of the project to be worked on, you can change the board title according to your needs



5. On your board, you will have one initial list called "To Do". You can add additional lists according to the stage of your workflow. For example, "In Progress", "Review", "Testing", and "Done".

6. Click "Add a list" to create a new list.

Question!

1. What steps do you take to invite team members to join your board?
2. Is it possible to make a board not from scratch (adding a list one by one), if so, how we can do that?

1. Assignment

Time: 60 minutes

1. Make groups/teams, and there are 3 students in each team!
2. Create a repository based on the topic chosen by your group, invite all team members to the repository.
3. Create a Kanban Board with a board title complete with cards according to the project topic your group has chosen, invite all team members to the board.