# Version Control System

# Version Control System (VCS)

Easily management collaboration on a project by avoiding frustration of swapping file

Ability to have unliminted number of developers working on the same code base

Easily revert back your file if something happened

# Type of VCS

1. Bazaar
2. Subversion (SVN)
3. CVS
4. Perforce
5. GIT
6. ect

# GIT

Meaning

Git is one of the version control systems used by developers to develop software simultaneously.

Function

The main function of git is to manage the version of the source code of your program by giving a dash and which code is added or replaced.

# The Advantages Of Using Git

1. Git is free and Open source distribution system
2. No centralized connectivity is needed
3. Powerfull
4. Design that is easy to understand
5. etc

# Start Using GIT

Pre- requisites

1. Run on Ubuntu 19.04
2. Git latest version (2.20.1)
3. Have a Github account

# Instalations

The first step that must be done is the instalations

1. The command used is $ sudo apt update && sudo apt install git -y
2. After that check your git version used $ git --version
3. The final  step is to add your identity such as  username and email that will become  your  author’s name

   $ git config --global user.email "youremailaddress"

   And

   $ git config --global user.name "yourusername"

# Usage

When using Git, we have 3 parts

1. Basic
2. Medium
3. Advance

# Basic

Follow every step

1. The first step is to initialize the directory to be used

   $ git init namaDirectory

2. Move to that directory and find the .git / subdirectory that git will use to save each change

   $ cd namaDirektory && ls -la

3. File with name .gitignore will contain files that will be ignored by git. Membuat file .gitignore harus dilakukan di awal

   $ touch .gitignore

   $ cat .gitignore

4. Add some file and directory
5. Check the status of directory

   $ git status
6. Next, changes the file be staged with command

   $ git add filename or $ git add .gitignore file.html index.html or $ git add .
7. Check status directory again

   $ git status
8. After that, change the condition of the file to be committed so that all changes are saved

   $ git commit -m “Message”

9. for cases when there are changes in the index.html file and then cek the status

   $ cat index.html

   $ git status

10. When you want to check which part has changed, use this command

    $ git diff

11. Committed again to add changes to git.

    $ git add index.html && git commit -m “Message

12. do the following steps to view the change log in the repository.

    $ git log

    For more spesific log for each file, using

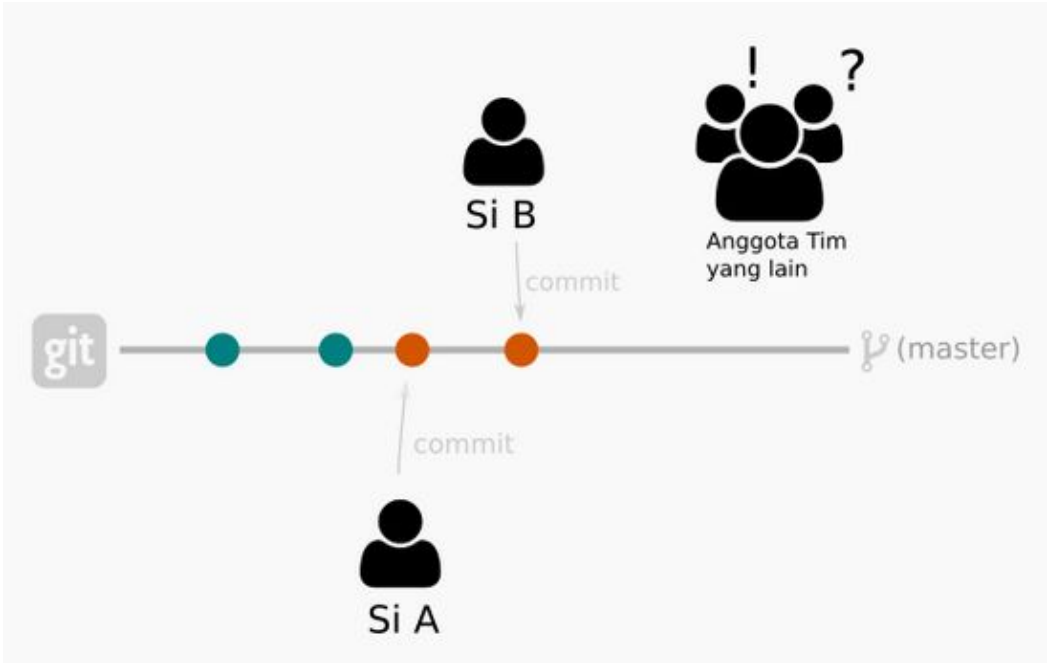
    $ git log namafile

    Or check the log by author

    $ git log --author=’namaauthor’

# Medium

Or branching model. The purpose of using branches is to avoid conflicts when collaborating while working on a project in groups.



# Follow the command

1. Make a branch

   $ git branch namabranch

2. Check the brach is in the repository

   $ git branch *

3. Move to the branch

   $ git checkout namabranch

4. Add file and do commit

   $ touch register.html; git add register.html; git commit -m “Message”

5. Move to the master branch and check that register.html already exists

   $ git chackout master && tree

6. Back to the previous branch to check whether the register file exists or not.

   $ git checkout namabranck && tree

7. First, move to the master branch and Join the branch and master branch

   $ git checkout master

   $ git merge namabranch

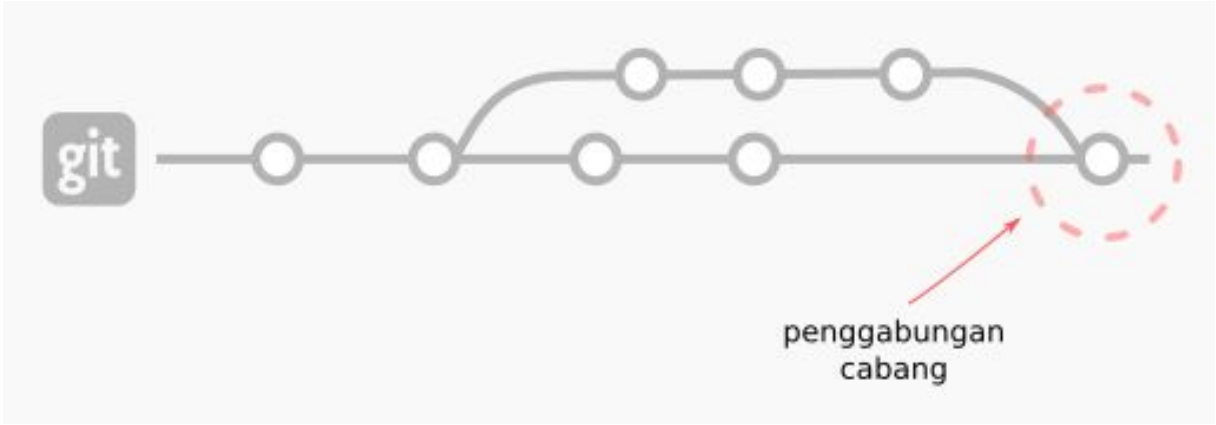
8. Check the branch file in the master branch

   $ tree

9. Delete the branch that you no longer use

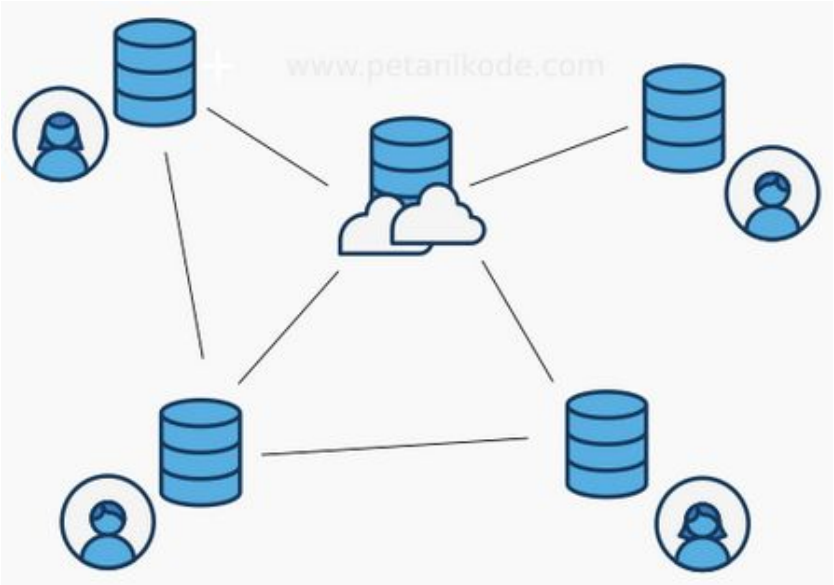
   $ git branch -d namabranch

10. And finally, the branch looks like this

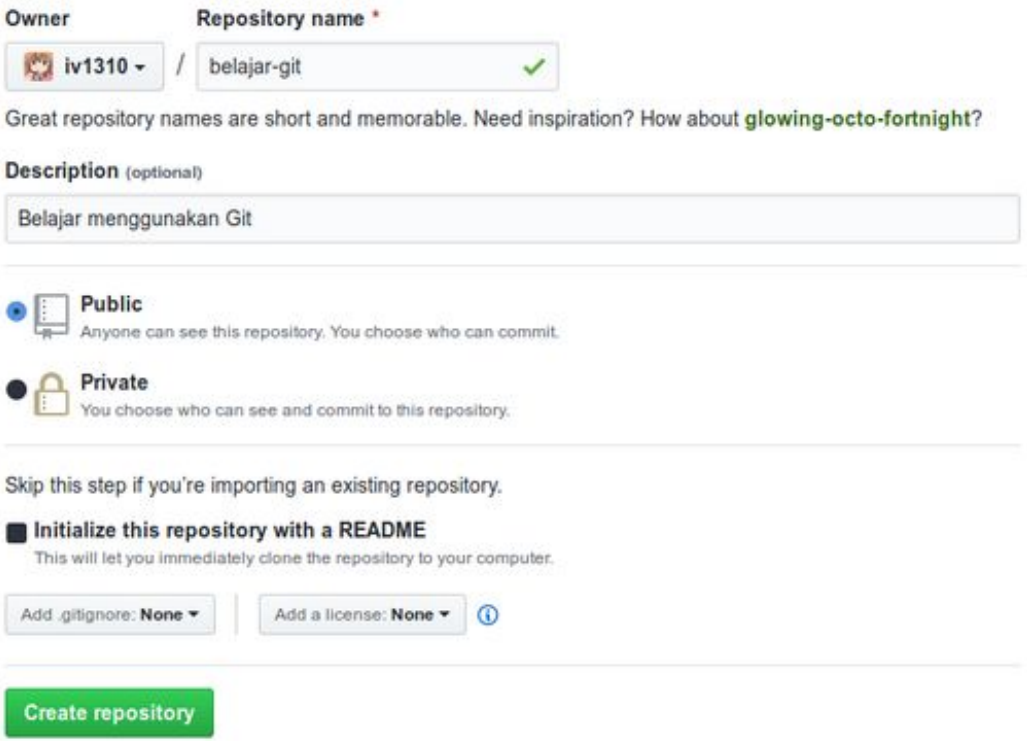


# Advance

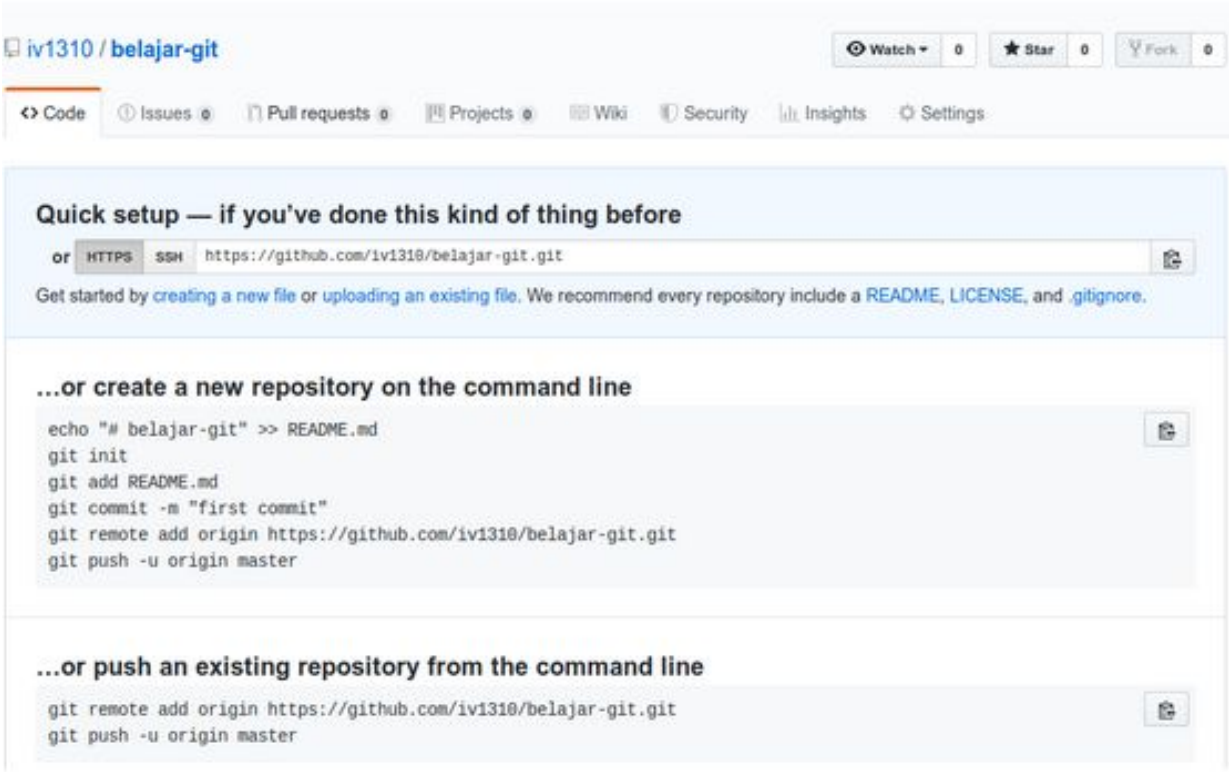
When collaborating, it is not possible to save files in a local repository. Then to combine it requires a remote that will store all the results of each person's work



1. Create a repository on the github account



2. Create remote repository



3. Add remote to connect the local repository with the remote repository

   $ git remote add nameremote urlrepository

4. check what remotes are available for the repository

   $ git remote -v

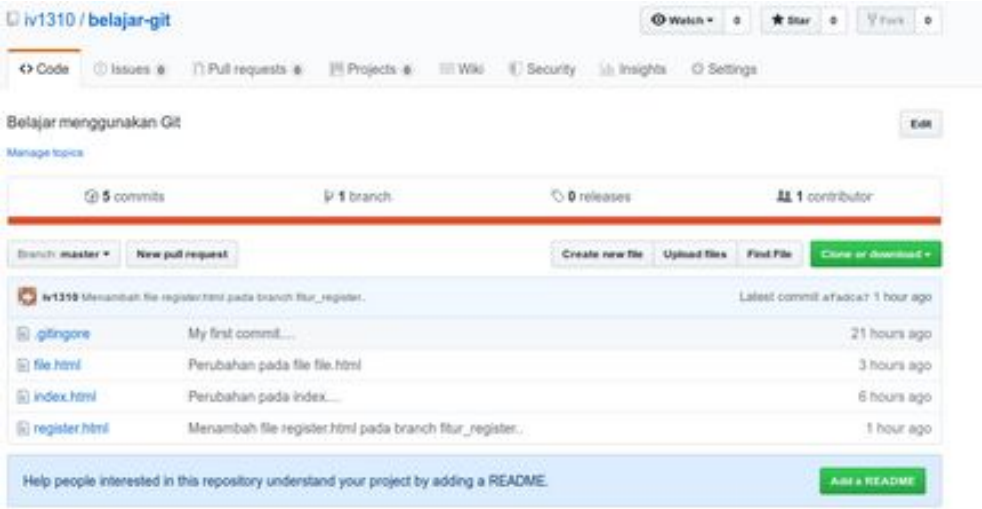
5. Check remote name

   $ git remote

6. How to send or add projects in local repositories to remote repositories

   $ git push namaremote master

8. Check the github page or the remote repository that was created




9. Try changing a file, then see the changes

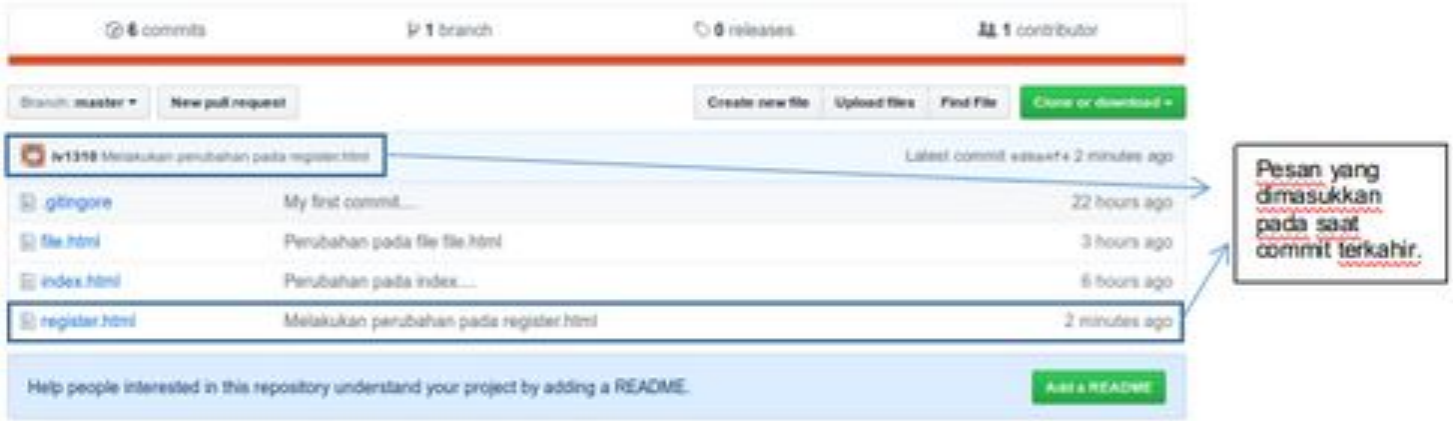
   $ git diff

10. Do the commit and push to remote rep

    $ git add namafile && git commit -m “Message”

    $ git push namaremote master

11. Cek github repository

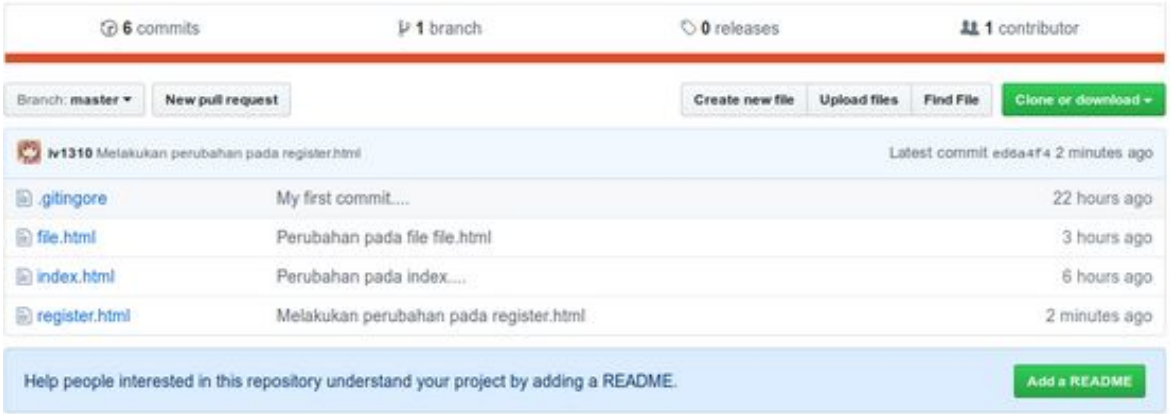


Next to take changes or source code from a remote repository. When working with a repository in collaboration or having many contributors, it should first take changes in the remote repository so there is no clash on the remote / core repository.

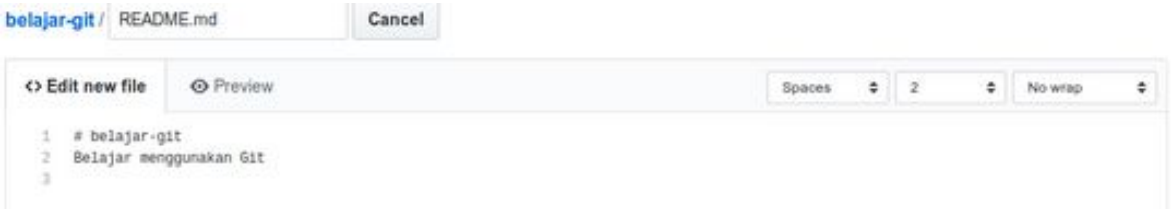
There is 2 command:

1. Git fetch namaremote, will only take revisions (commit) and not immediately merge (merge) of the local repository
2. Git pull namaremote, will take a revision (commit) and immediately merge (merge) on the local repository.

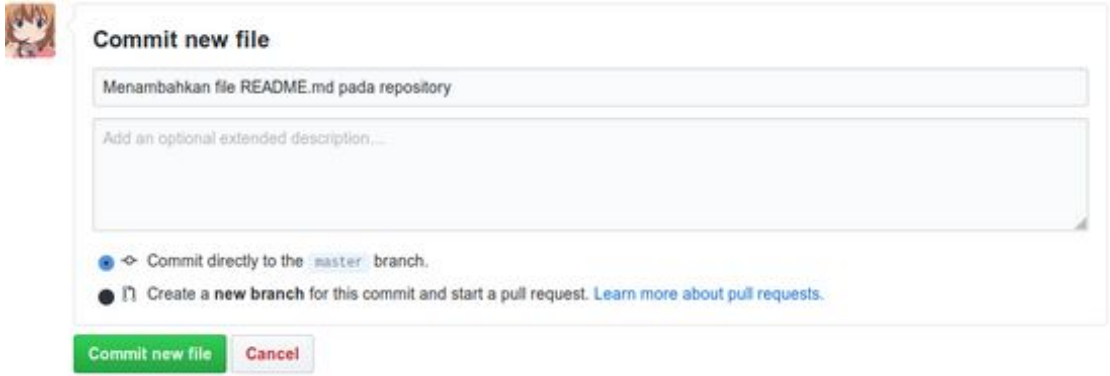
12. Next, make a revision with the git fetch command. First go to the github page, then add the README.md file via github.




13. Fill in the readme.md file section



14. Commit




15. Take the revision

    $ git fetch namaremote master

16. Check log repository

    $ git log --online

17. If the commit made on github has not been previously recorded in the log in the local repository. Then try checking the logs in the remote repository

    $ git log --online

18. Or you can chaeck it using comand

    $ git diff master namaremote/master

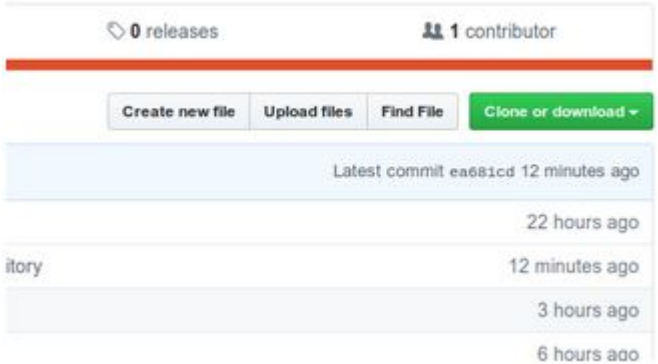
19. Then to combine it you can use

    $ git merge master namaremote/master

20. Then check log local repository

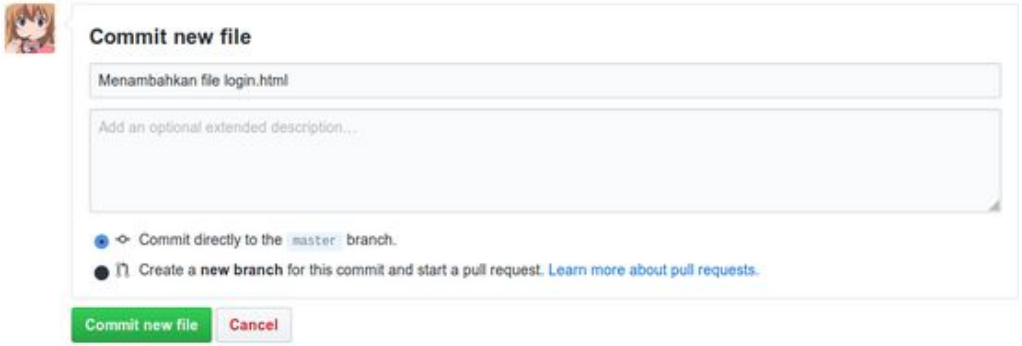
    $ git log --online

21. Next take the revision using the help of the git pull command. Add new file at github remote repository page.




22. Make the name and contents of the new file to be added.

22. And then commit




23. Take the changes in the remote repository

    $ git pull namaremote master

24. Check log repository

    $ git log --online

25. Git can also copy source code or projects in a remote repository to local

    $ git clone url

26. Then try going into the cloned remote repository directory

    $ git remote -v

27. Check log repository

    $ git log --online


Thank
you

# Version Control System

# Version Control System (VCS)

Easily management collaboration on a project by avoiding frustration of swapping file

Ability to have unliminted number of developers working on the same code base

Easily revert back your file if something happened

# Type of VCS

1. Bazaar
2. Subversion (SVN)
3. CVS
4. Perforce
5. GIT
6. ect

# GIT

Meaning

Git is one of the version control systems used by developers to develop software simultaneously.

Function

The main function of git is to manage the version of the source code of your program by giving a dash and which code is added or replaced.

# The Advantages Of Using Git

1. Git is free and Open source distribution system
2. No centralized connectivity is needed
3. Powerfull
4. Design that is easy to understand
5. etc

# Start Using GIT

Pre- requisites

1. Run on Ubuntu 19.04
2. Git latest version (2.20.1)
3. Have a Github account

# Instalations

The first step that must be done is the instalations

1. The command used is $ sudo apt update && sudo apt install git -y
2. After that check your git version used $ git --version
3. The final  step is to add your identity such as  username and email that will become  your  author’s name

   $ git config --global user.email "youremailaddress"

   And

   $ git config --global user.name "yourusername"

# Usage

When using Git, we have 3 parts

1. Basic
2. Medium
3. Advance

# Basic

Follow every step

1. The first step is to initialize the directory to be used

   $ git init namaDirectory

2. Move to that directory and find the .git / subdirectory that git will use to save each change

   $ cd namaDirektory && ls -la

3. File with name .gitignore will contain files that will be ignored by git. Membuat file .gitignore harus dilakukan di awal

   $ touch .gitignore

   $ cat .gitignore

4. Add some file and directory
5. Check the status of directory

   $ git status
6. Next, changes the file be staged with command

   $ git add filename or $ git add .gitignore file.html index.html or $ git add .
7. Check status directory again

   $ git status
8. After that, change the condition of the file to be committed so that all changes are saved

   $ git commit -m “Message”

9. for cases when there are changes in the index.html file and then cek the status

   $ cat index.html

   $ git status

10. When you want to check which part has changed, use this command

    $ git diff

11. Committed again to add changes to git.

    $ git add index.html && git commit -m “Message

12. do the following steps to view the change log in the repository.

    $ git log

    For more spesific log for each file, using

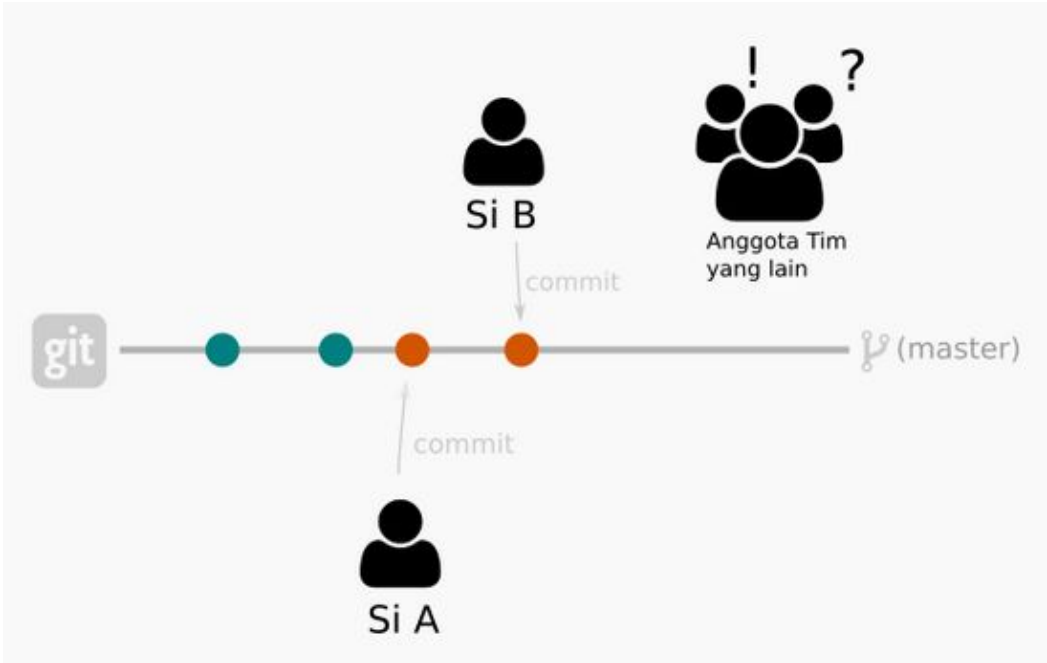
    $ git log namafile

    Or check the log by author

    $ git log --author=’namaauthor’

# Medium

Or branching model. The purpose of using branches is to avoid conflicts when collaborating while working on a project in groups.



# Follow the command

1. Make a branch

   $ git branch namabranch

2. Check the brach is in the repository

   $ git branch *

3. Move to the branch

   $ git checkout namabranch

4. Add file and do commit

   $ touch register.html; git add register.html; git commit -m “Message”

5. Move to the master branch and check that register.html already exists

   $ git chackout master && tree

6. Back to the previous branch to check whether the register file exists or not.

   $ git checkout namabranck && tree

7. First, move to the master branch and Join the branch and master branch

   $ git checkout master

   $ git merge namabranch

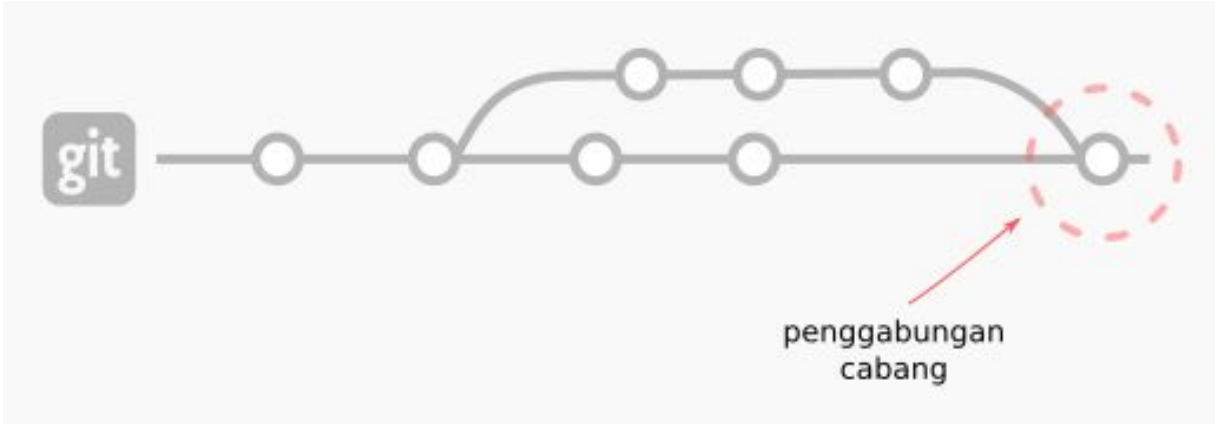
8. Check the branch file in the master branch

   $ tree

9. Delete the branch that you no longer use

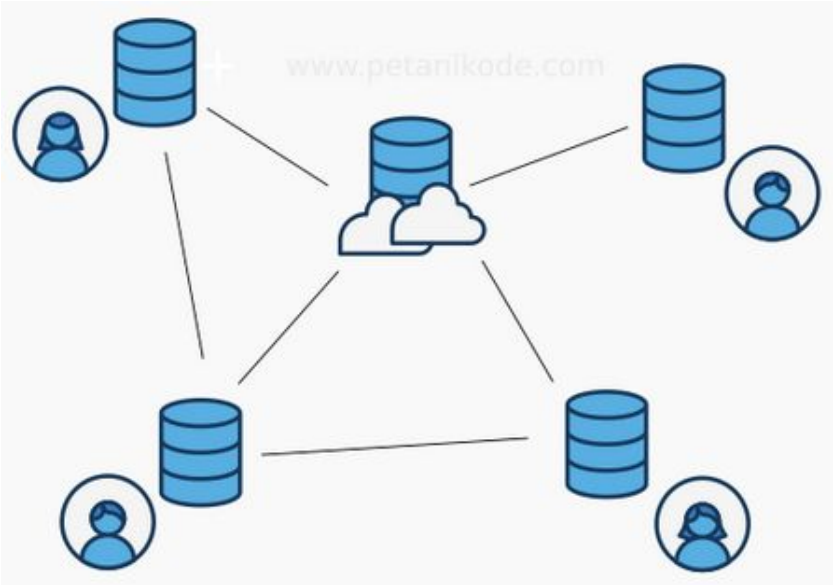
   $ git branch -d namabranch

10. And finally, the branch looks like this

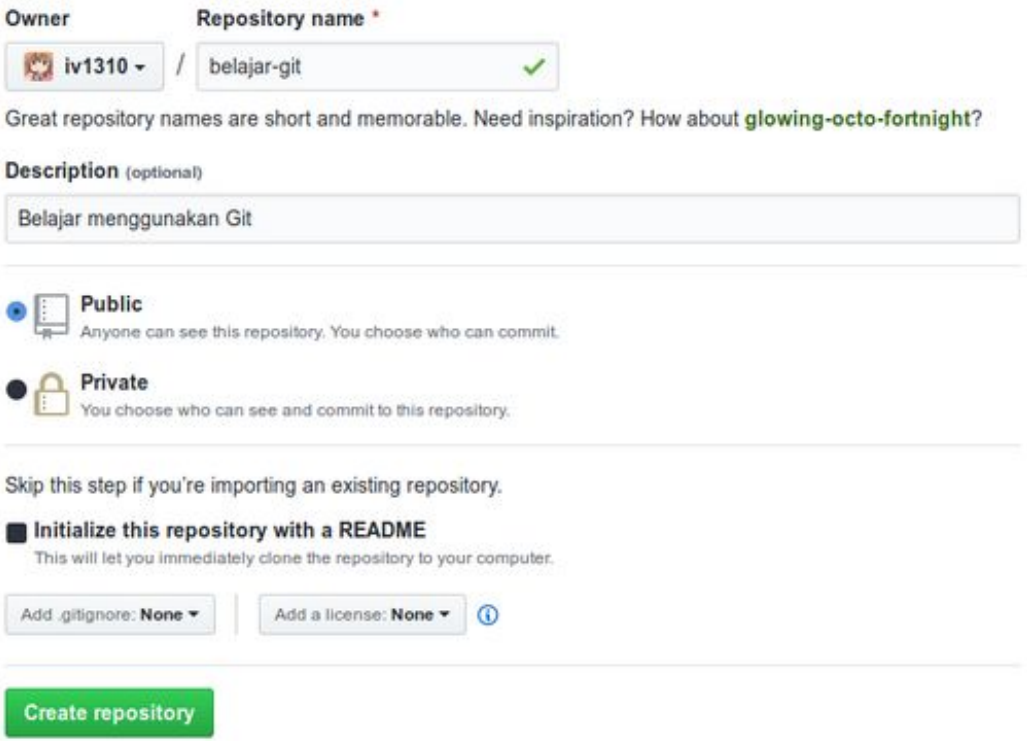


# Advance

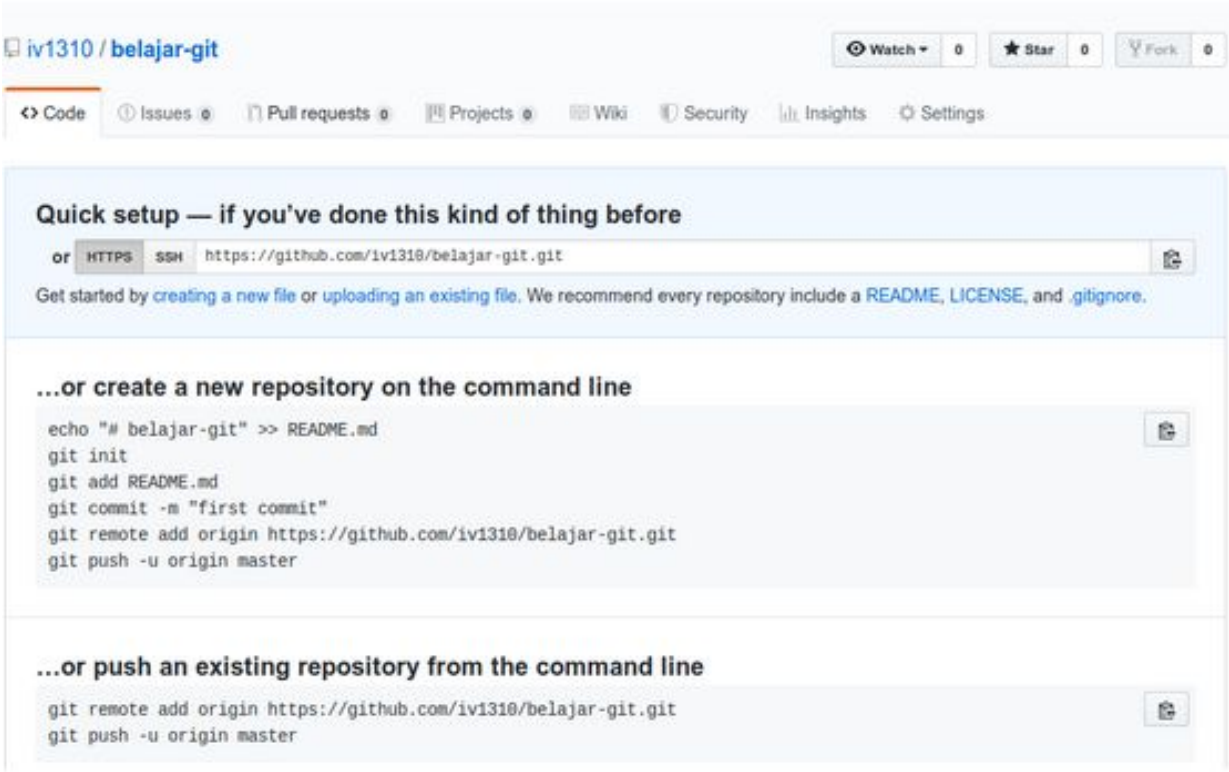
When collaborating, it is not possible to save files in a local repository. Then to combine it requires a remote that will store all the results of each person's work



1. Create a repository on the github account



2. Create remote repository



3. Add remote to connect the local repository with the remote repository

   $ git remote add nameremote urlrepository

4. check what remotes are available for the repository

   $ git remote -v

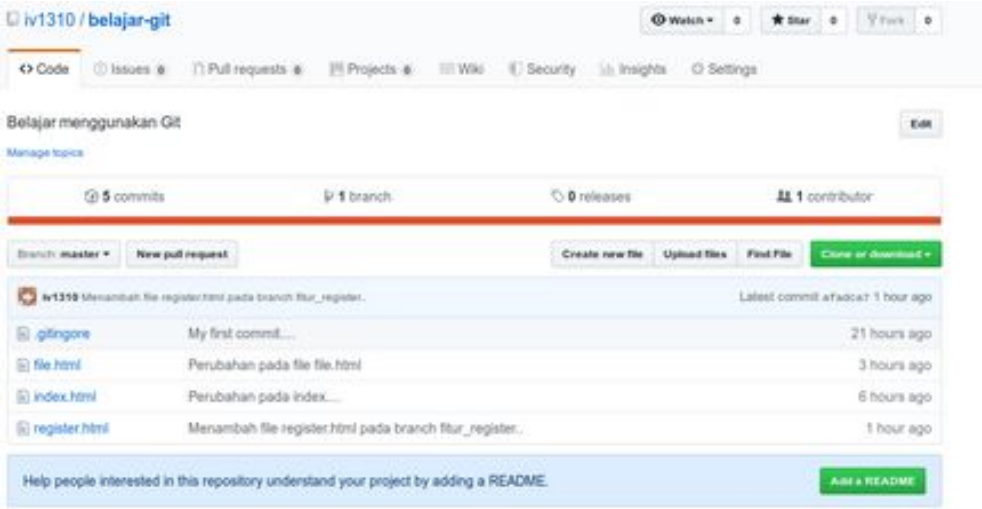
5. Check remote name

   $ git remote

6. How to send or add projects in local repositories to remote repositories

   $ git push namaremote master

8. Check the github page or the remote repository that was created




9. Try changing a file, then see the changes

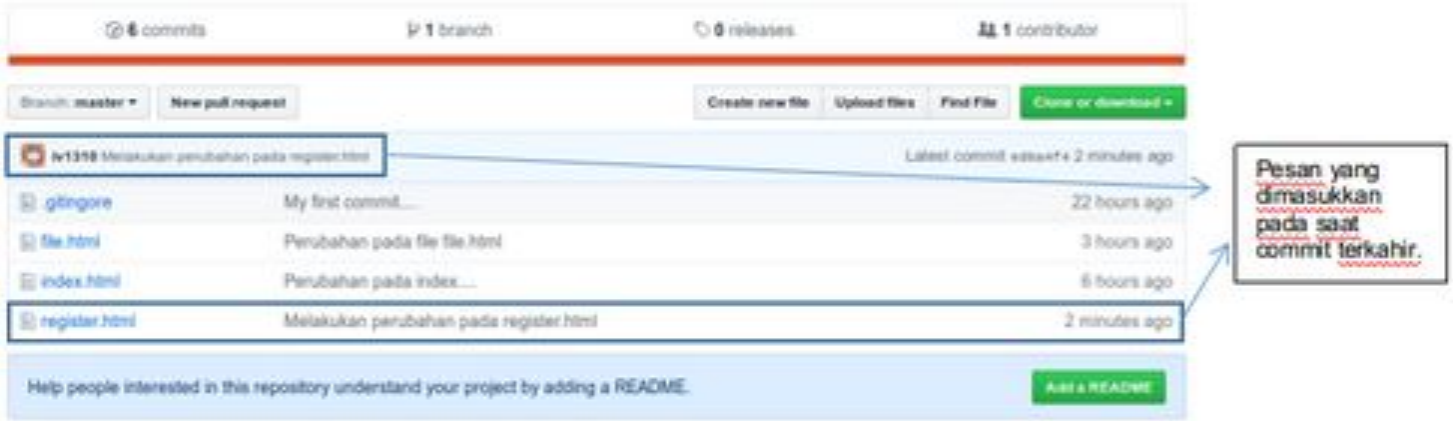
   $ git diff

10. Do the commit and push to remote rep

    $ git add namafile && git commit -m “Message”

    $ git push namaremote master

11. Cek github repository

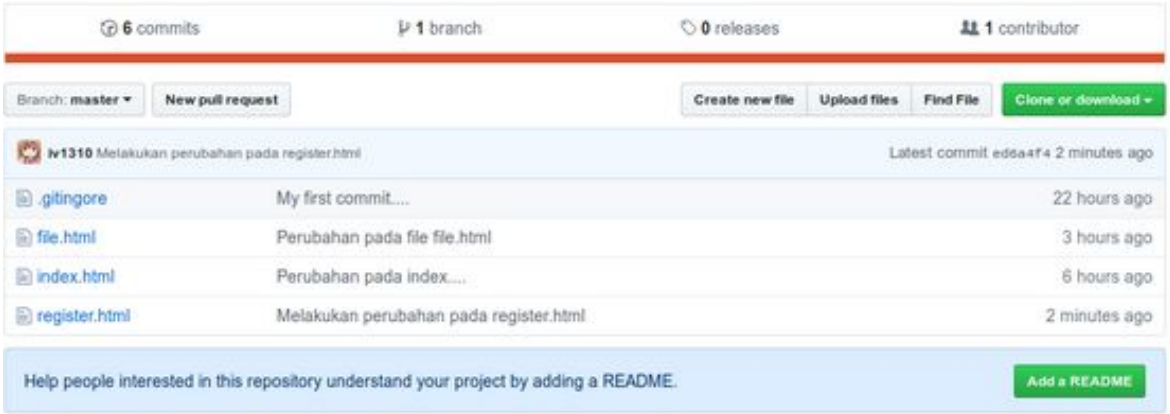


Next to take changes or source code from a remote repository. When working with a repository in collaboration or having many contributors, it should first take changes in the remote repository so there is no clash on the remote / core repository.

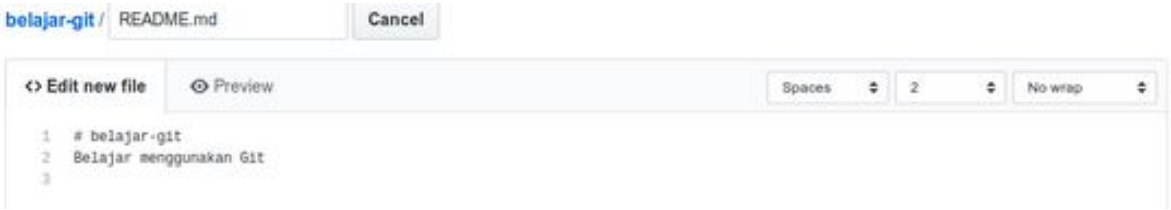
There is 2 command:

1. Git fetch namaremote, will only take revisions (commit) and not immediately merge (merge) of the local repository
2. Git pull namaremote, will take a revision (commit) and immediately merge (merge) on the local repository.

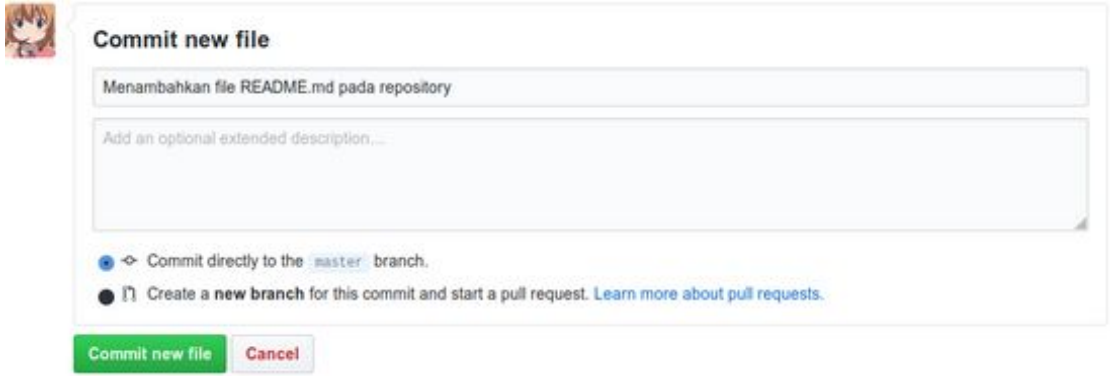
12. Next, make a revision with the git fetch command. First go to the github page, then add the README.md file via github.




13. Fill in the readme.md file section



14. Commit




15. Take the revision

    $ git fetch namaremote master

16. Check log repository

    $ git log --online

17. If the commit made on github has not been previously recorded in the log in the local repository. Then try checking the logs in the remote repository

    $ git log --online

18. Or you can chaeck it using comand

    $ git diff master namaremote/master

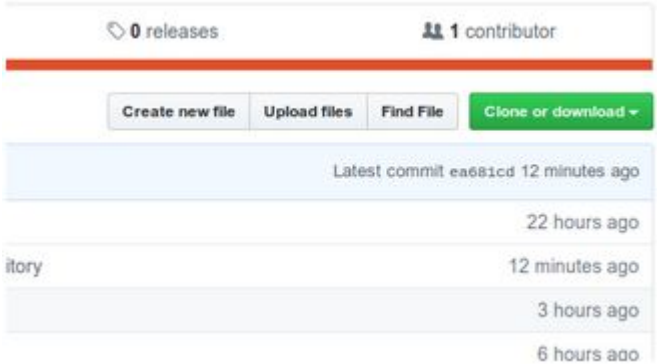
19. Then to combine it you can use

    $ git merge master namaremote/master

20. Then check log local repository

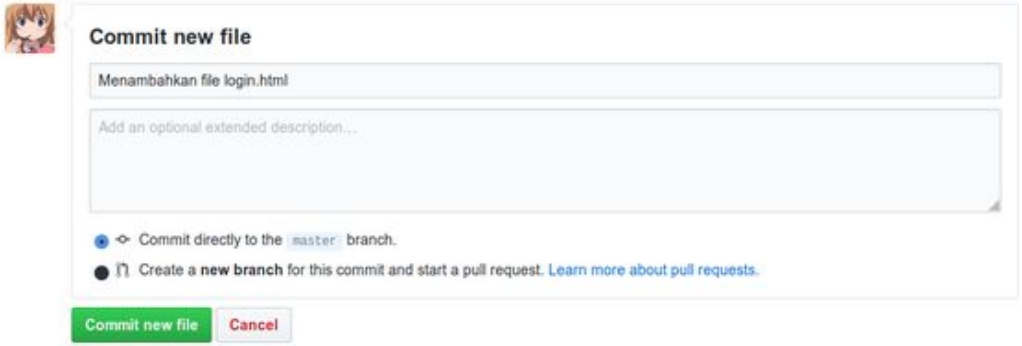
    $ git log --online

21. Next take the revision using the help of the git pull command. Add new file at github remote repository page.




22. Make the name and contents of the new file to be added.

22. And then commit




23. Take the changes in the remote repository

    $ git pull namaremote master

24. Check log repository

    $ git log --online

25. Git can also copy source code or projects in a remote repository to local

    $ git clone url

26. Then try going into the cloned remote repository directory

    $ git remote -v

27. Check log repository

    $ git log --online


Thank
you