Git & Github

Pertemuan 2
MK Algoritma Pemrograman II
2021/2022

Ika Qutsiati Utami, S.Kom., M.Sc.

M. N. Fakhruzzaman, S.Kom., M.Sc.

Program Studi S1 Teknologi Sains Data

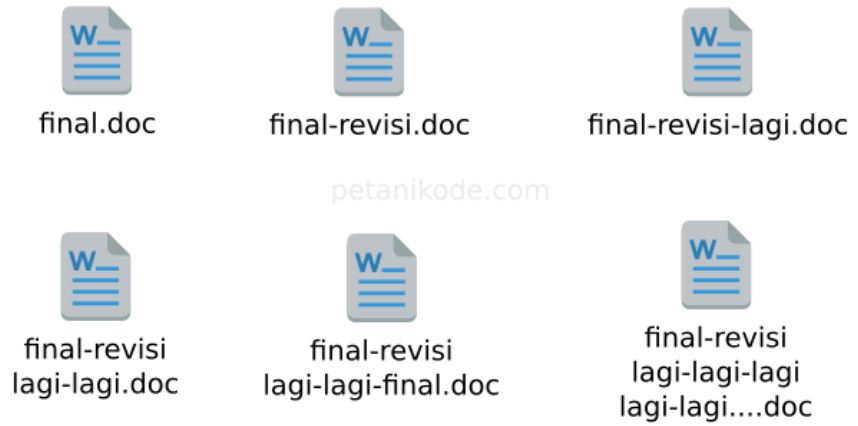
Fakultas Teknologi Maju dan Multidisiplin

Universitas Airlangga Indonesia

Git

What is Git?

- Tools proyek pengembangan software (***version control system & collaboration***)
- Termasuk ***distributed revision control*** (VCS terdistribusi)
- Dikembangkan oleh **Linus Torvald** (2005) untuk Linux



Tradisional

With Git

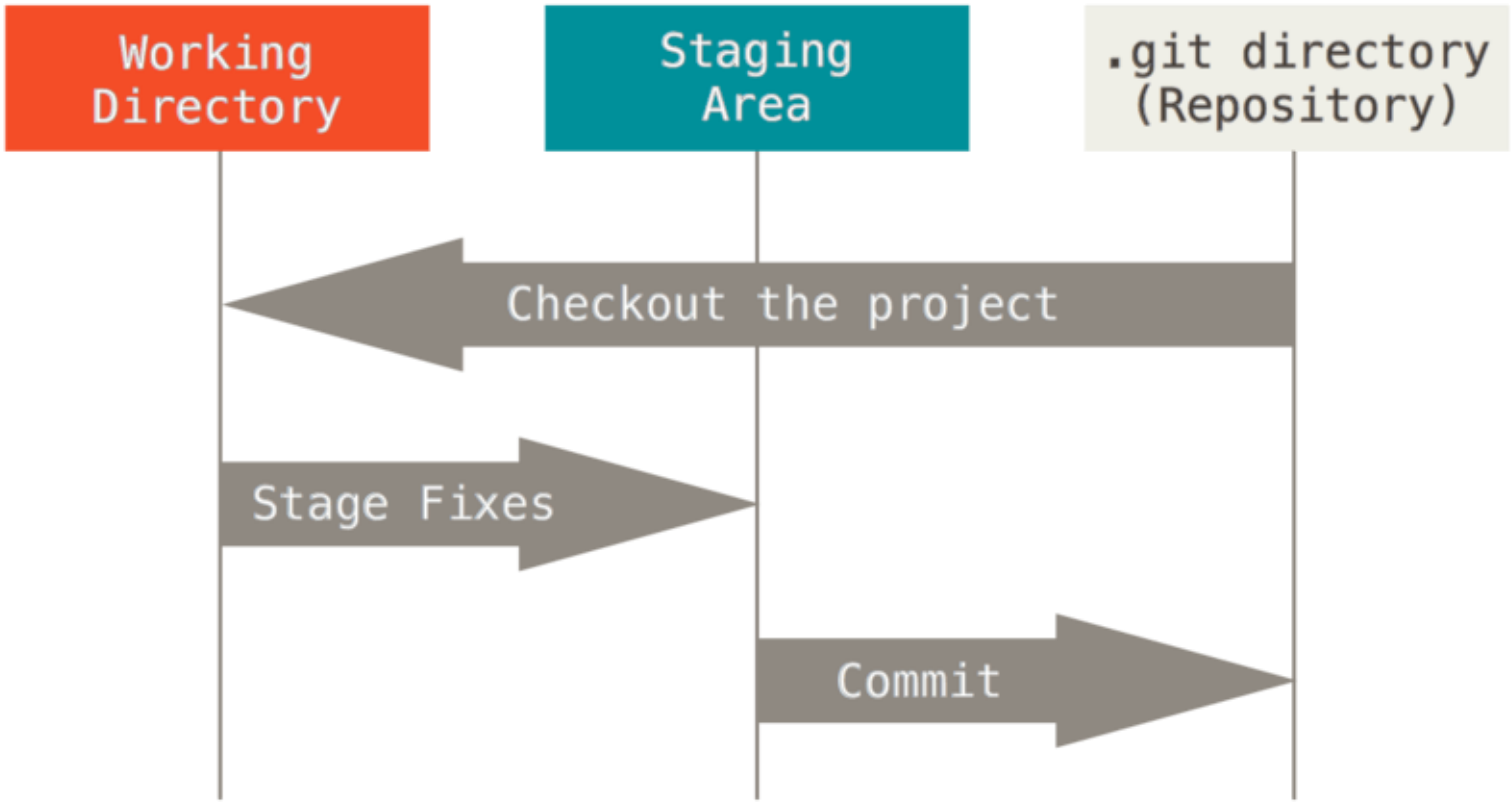
Why using Git?

- Menyimpan seluruh versi source code
- Tools untuk kolaborasi dalam proyek
- Aman untuk kolaborasi, karena dapat di cek apa yang diubah dan siapa yang mengubahnya
- Pintu masuk untuk berkontribusi pada proyek open-source
- Deploy aplikasi modern
- Membuat blog
- Dsb..

The Three States

- Git generally **only adds data**
  - **Modified:** kondisi dimana revisi atau perubahan sudah dilakukan, tetapi belum ditandai dan belum disimpan di version control.
  - **Staged**: kondisi dimana revisi sudah ditandai, tetapi belum disimpan di version control.
  - **Committed**: kondisi dimana revisi sudah disimpan di version control.

Git vs other VCS



Repositori (repository) adalah istilah yang digunakan untuk direktori proyek yang menggunakan Git.

Important Vocabulary

No	Istilah	Keterangan
1	Repo/Repository	Folder suatu project.
2	Commit	Rekaman/snapshot dari repository (Riwayat perubahan repository).
3	Hash	Penanda unik pada sebuah commit (terdiri dari angka dan huruf yang panjang).
4	Checkout	Berpindah ke sebuah perubahan tertentu.
5	Branch	Cabang dari sebuah perubahan.
6	Merge	Menggabungkan dua atau lebih branch.
7	Remote	Resource yang memiliki repository.
8	Clone	Mengambil repository dari *remote*.
9	Push	Mengirim commit ke repository.
10	Pull	Mengambil commit dari repository.

Installing Git

- Linux (https://git-scm.com/download/linux)
- MacOS (https://git-scm.com/download/mac)
- Windows (https://git-scm.com/download/win)

Using Git

Access:

- Command-line
- Graphical user interfaces (GUI)

First setting up (configuration):

- Identity:

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

- Editor:

```
$ git config --global core.editor emacs
```

- Branch name:

```
$ git config --global init.defaultBranch main
```

Getting a Git Repository

1. You can **take a local directory** that is currently not under version control, and turn it into a Git repository

```
$ cd C:/Users/user/my_project
```

and type:

```
$ git init
```

2. You can **clone** an existing Git repository from elsewhere

```
git clone <url>
```

```
$ git clone https://github.com/libgit2/libgit2
```

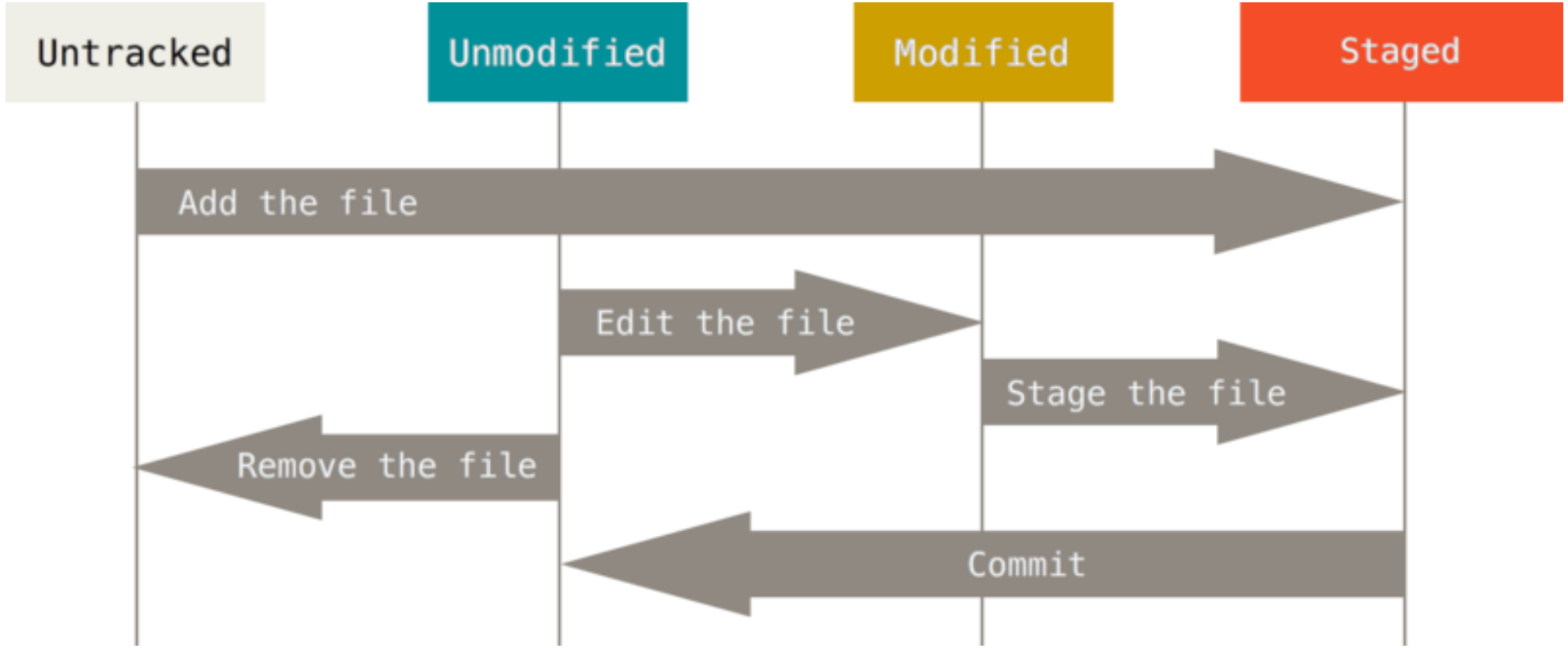
```
$ git add *.c
$ git add LICENSE
$ git commit -m 'Initial project version'
```

Recording Changes to the Repository

Each file in your working directory:

1. **Tracked**
   - Files that were in the last snapshot, as well as any newly staged files
   - They can be unmodified, modified, or staged
   - In short, tracked files are files that Git knows about
   - When you first clone a repository, all of your files will be tracked and unmodified because Git just checked them out and you haven't edited anything
2. **Untracked**
   - Any files in your working directory that were not in your last snapshot and are not in your staging area

Recording Changes to the Repository



The lifecycle of the status of your files

Checking the Status of Your Files

1. **If file exists**

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean
```

2. **If the file didn't exist before**

```
$ echo 'My Project' > README
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    README

nothing added to commit but untracked files present (use "git add" to track)
```

Git Basic Operations

1. git config
2. git init
3. git add
4. git clone
5. git commit
6. git status
7. git push
8. git checkout
9. git remote
10. git branch
11. git pull
12. git merge
13. git diff
14. git tag
15. git log
16. git restore
17. git reset
18. git remove
19. etc..

Github

What is Github?

- GitHub is a **website for hosting Git projects (layanan cloud)**.
- GitHub adds useful **web-based tools to Git** and makes it much easier to collaborate and share your projects with others.
- There is also a downloadable version of GitHub that you can install on Windows and macOS.

- Click **what is github**
- Github **website**
- Github **desktop**

What is GitHub?

- Manajemen project
- Repository online
- Sistem versioning code
- Platform jaringan sosial (untuk para developer seluruh dunia)

Git vs GitHub

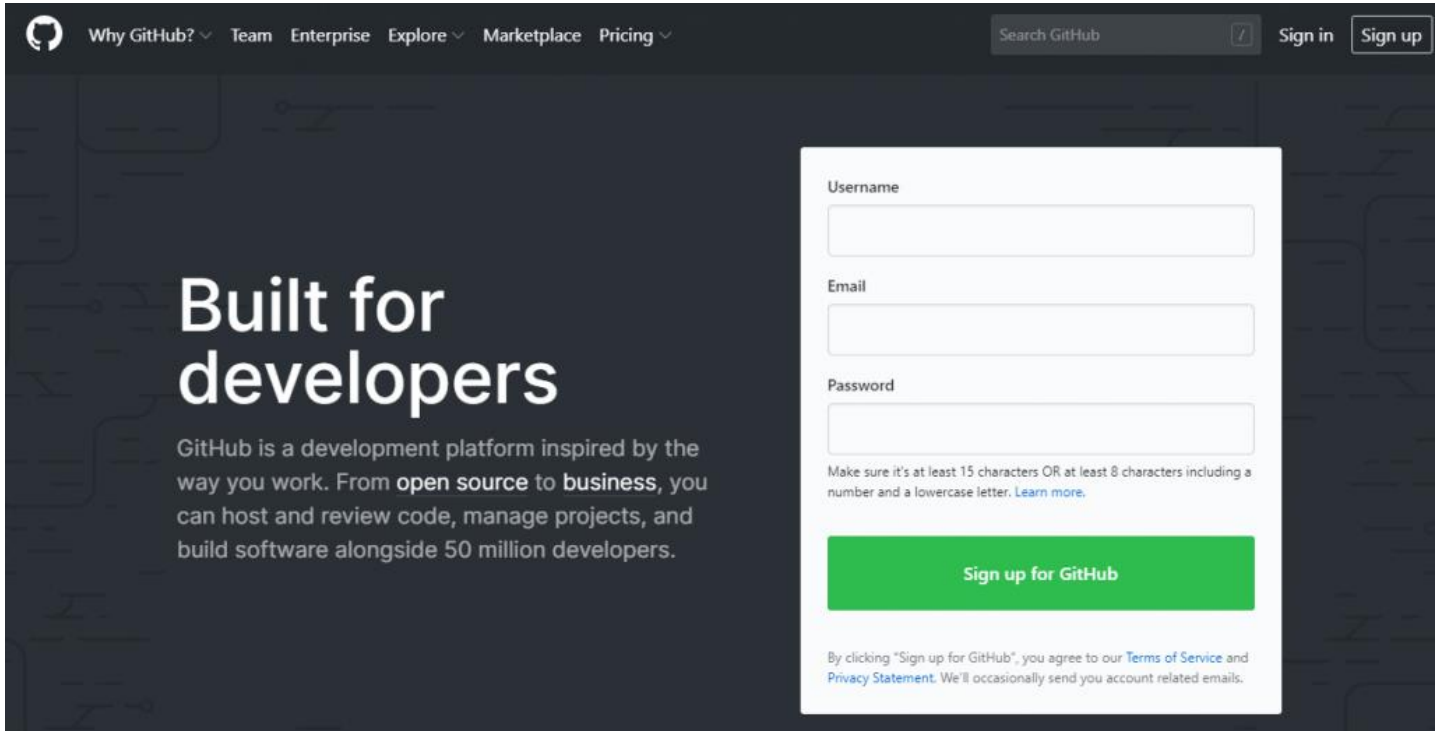
Git	GitHub
1. *Install software* di penyimpanan lokal	1. *Host* melalui layanan *cloud*
2. Dikelola oleh The Linux Foundation	2. Diakuisisi oleh Microsoft pada 2018
3. Fokus pada *version control* dan *code sharing*	3. Fokus pada *source code hosting* terpusat
4. Akses *offline*	4. Akses *online*
5. Tidak menggunakan fitur *user management*	5. Menggunakan *user management*
6. *Desktop interface* bernama “Git GUI”	6. *Desktop interface* “GitHub Desktop”
7. Pesaing: Mercurial, Subversion, IBM, Rational Team, Concert, dan ClearCase	7. Pesaing: GitLab dan Atlassian BitBucket
8. *Open sourced licensed*	8. Pilihan: pengguna gratis atau pengguna berbayar

Git vs GitHub

- **User interface** pada GitHub lebih menarik dan mudah dipahami oleh pengguna awal.
- Fitur lain GitHub: kita dapat membaca berbagai **blog dan feed** yang dibuat oleh sesama pengguna (forum diskusi para programmer)
- GitHub dan Git pada konsep kerjanya hampir sama dengan Dropbox dan Google Drive, hanya saja Git dan GitHub bekerja untuk **mengolah kode script**. Sedangkan DropBox dan Google Drive bertugas untuk mengolah kata.
- GitHub dan Git merupakan **alat version control**, bedanya github sekaligus dilengkapi **penyimpanan cloud**.

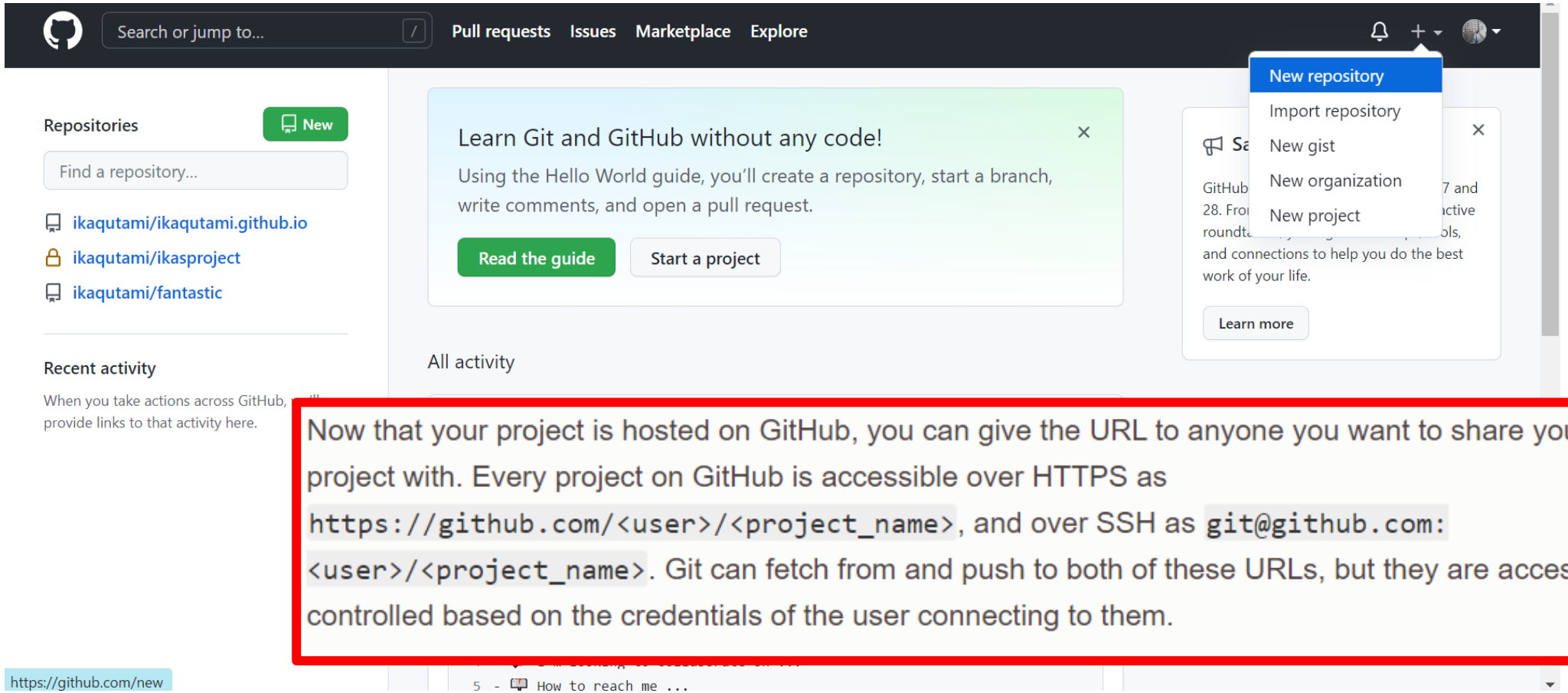
Steps

1. Join account
2. Verifikasi Alamat E-mail



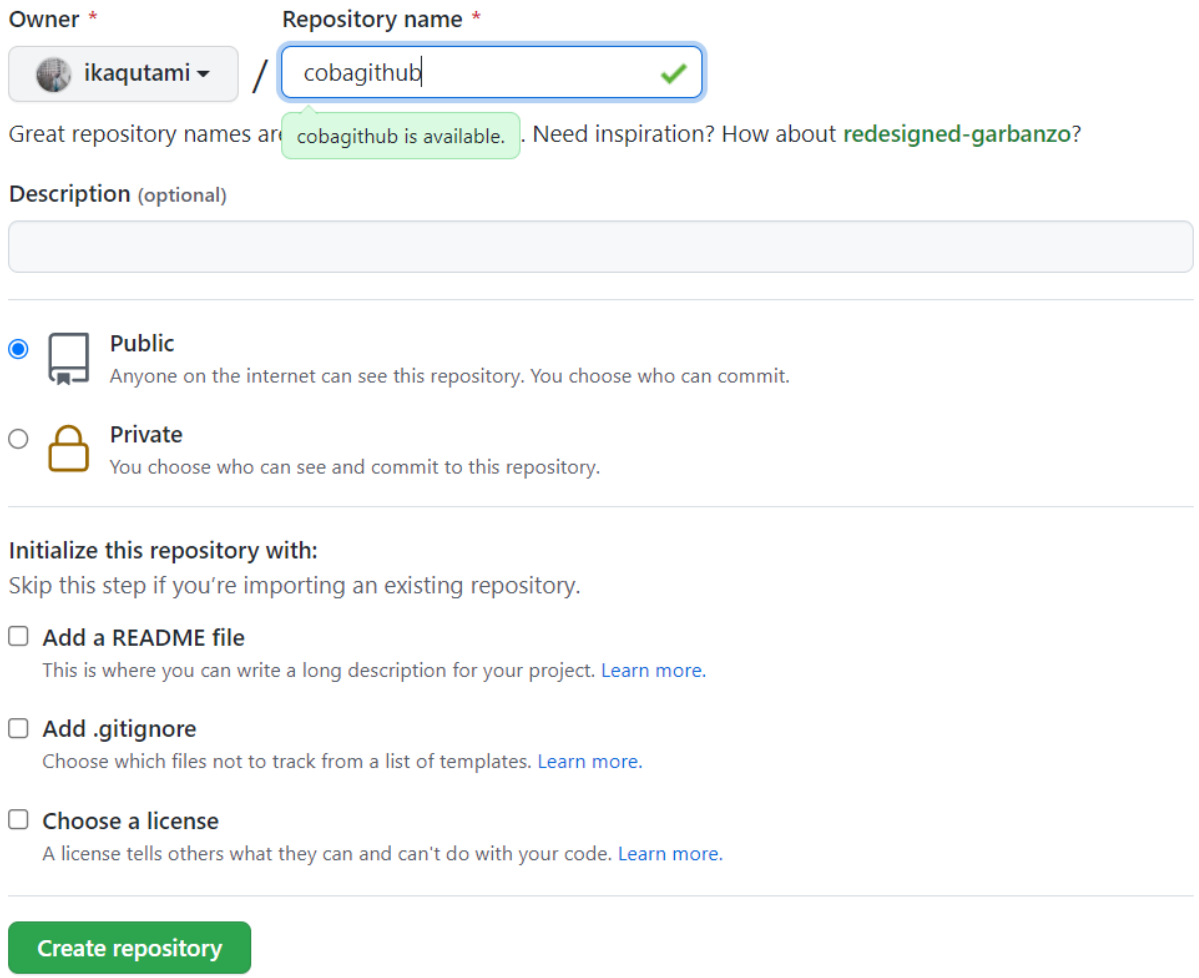
Maintaining a Project

- Creating a New Repository

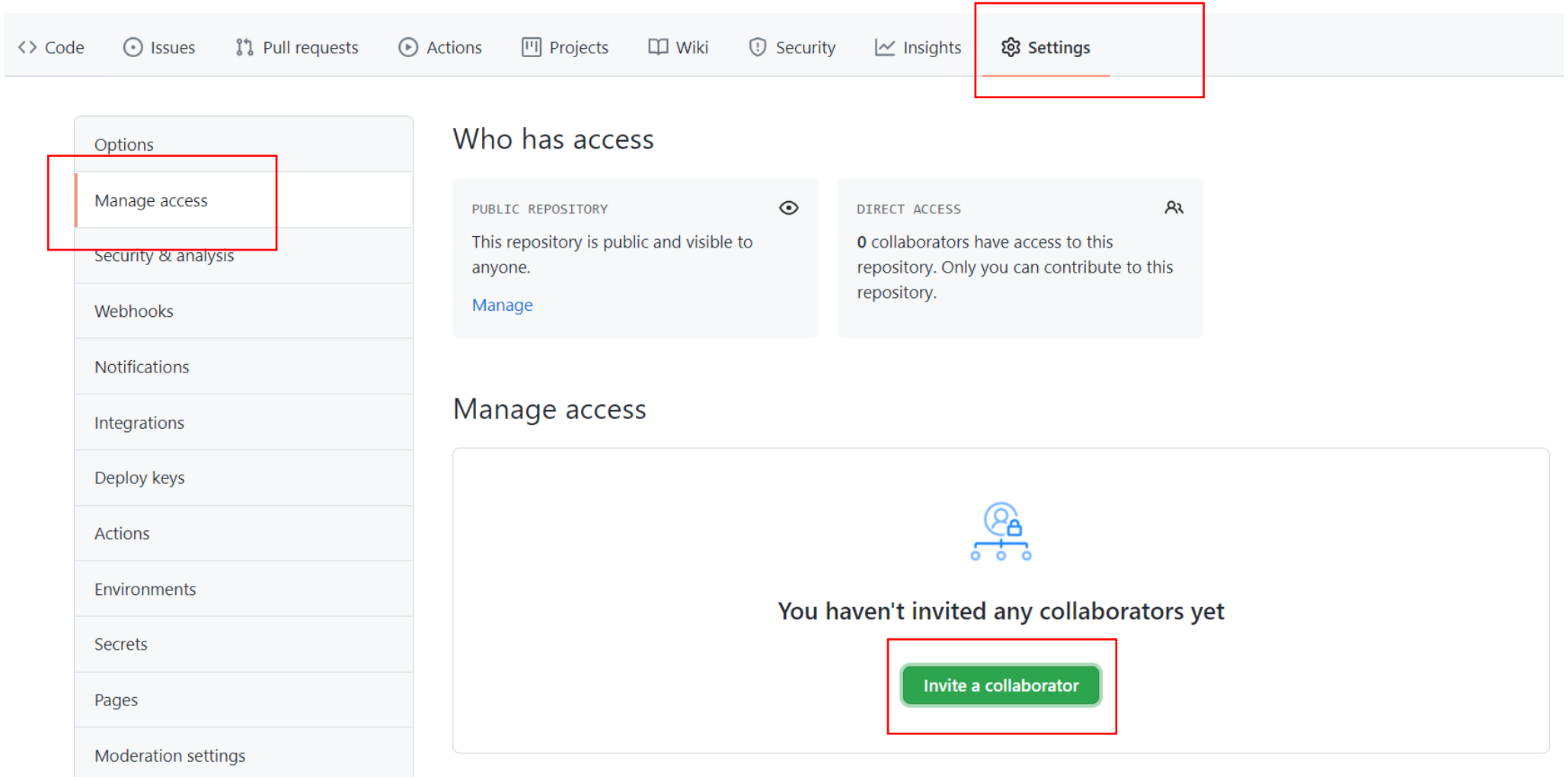


Maintaining a Project

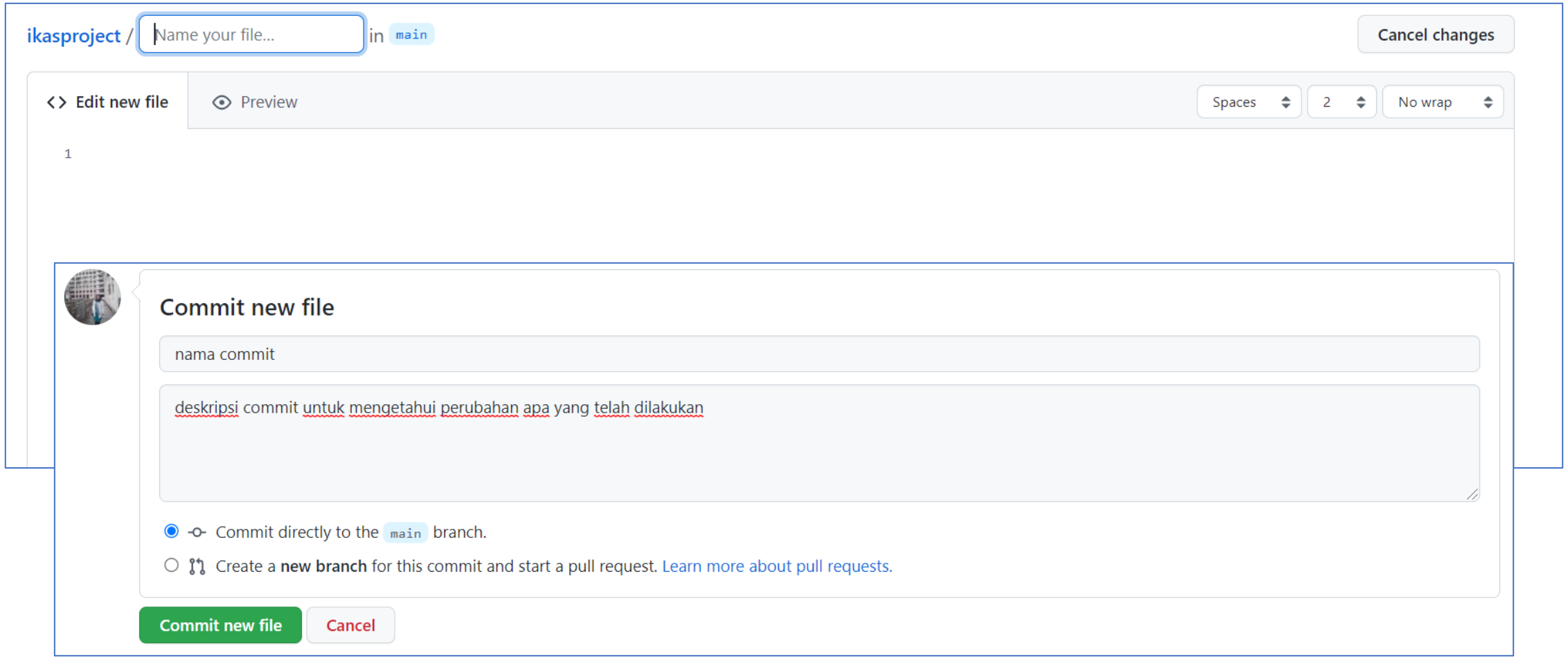
- File **“Readme”** selalu ada di setiap repository untuk menjelaskan penjelasan singkat/detail, cara penggunaan, dan sebagainya.
- Dibuat secara otomatis jika checklist tombolnya.



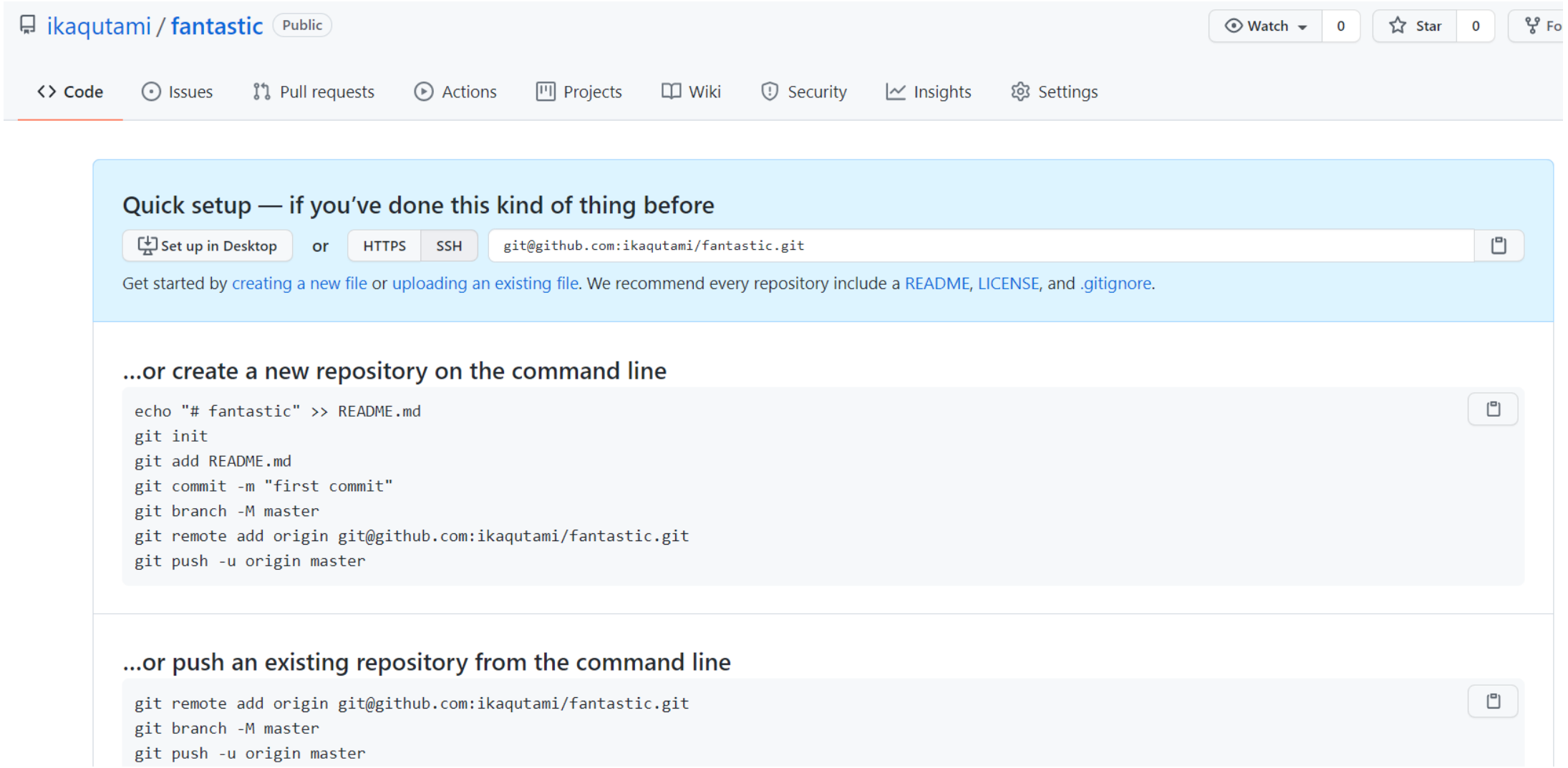
Add Collaborators



Creating new file

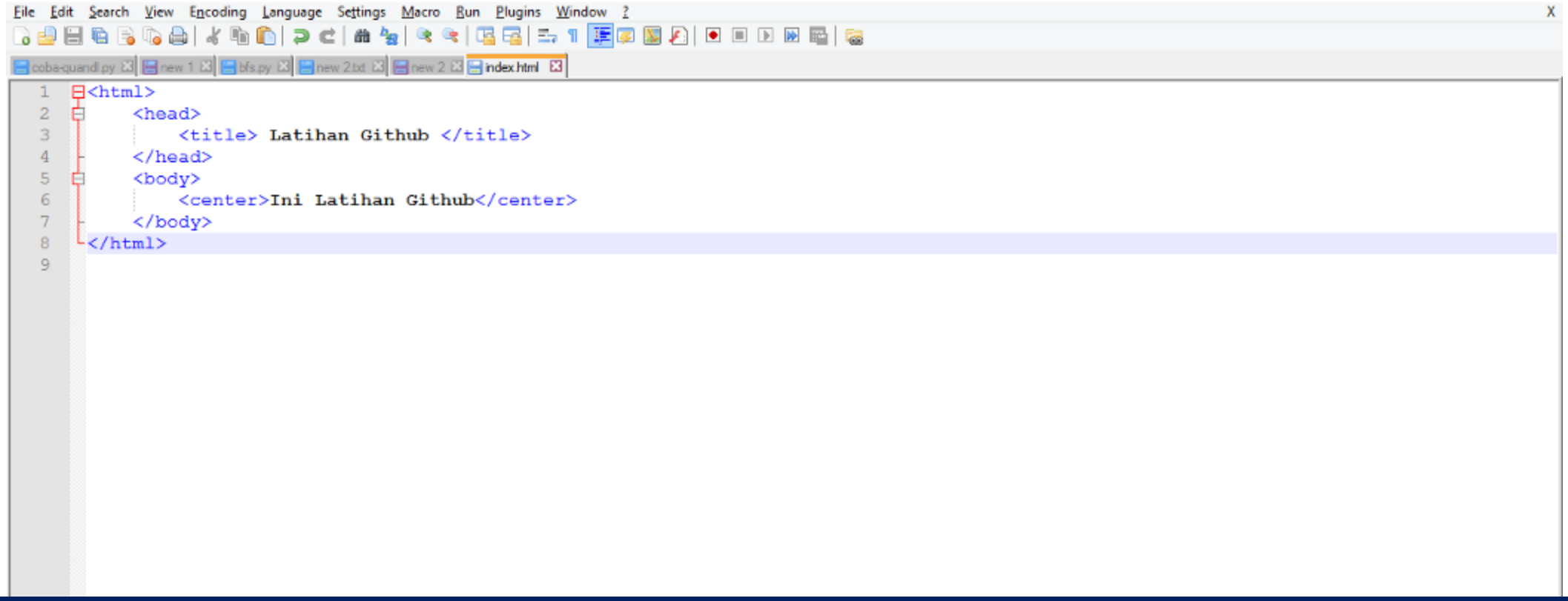


Creating new file



Push File

- Buat folder
- Buat file baru misal **index.html**
- Edit file **index.html** dan masukkan kode berikut

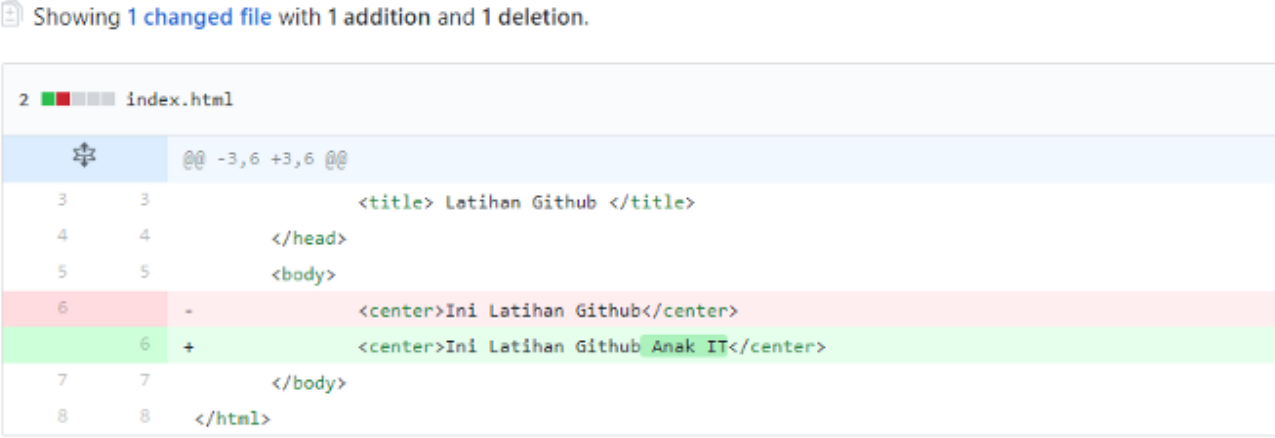


Push File

- Klik kanan dan klik "*Git Bash Here*"
- **Git init -> git add -> git commit -m "first commit"**
- Git remote add origin https://github.Com/username/latihan-github.git
- Git push origin master
- Cek file di github
- Edit file lokal, ulang proses dari atas untuk commit2 selanjutnya.

9 lines (8 sloc) | 121 Bytes — Raw | Blame | History

```
<html>
        <head>
                <title> Latihan Github </title>
        </head>
        <body>
                <center>Ini Latihan Github</center>
        </body>
</html>
```



Showing 1 changed file with 1 addition and 1 deletion.

2 index.html

```
@@ -3,6 +3,6 @@
                <title> Latihan Github </title>
        </head>
        <body>
-               <center>Ini Latihan Github</center>
+               <center>Ini Latihan Github Anak IT</center>
        </body>
</html>
```

Thank you..

References:

https://ocw.mit.edu/ans7870/6/6.005/s16/classes/05-version-control/index.html

https://git-scm.com/book/en/v2