



Git & Github

Pertemuan 2

*MK Algoritma Pemrograman II
2021/2022*

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

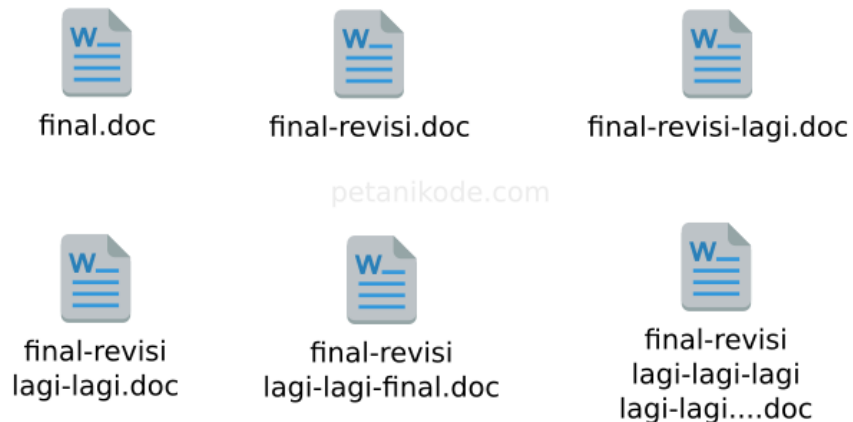
M. N. Fakhruzzaman, S.Kom., M.Sc.

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

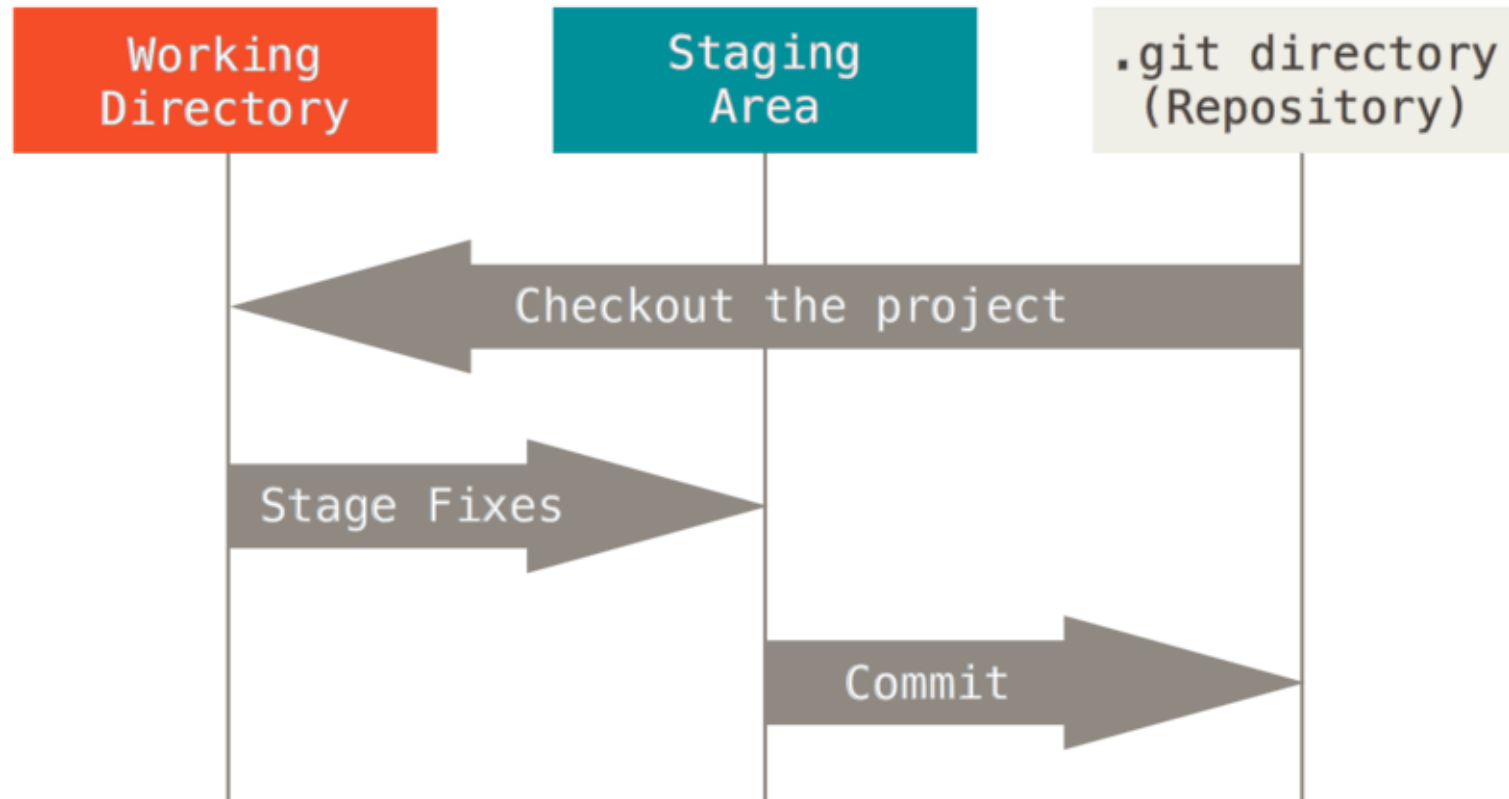
Why using Git?

- 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



Repository (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 <i>remote</i> .
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:

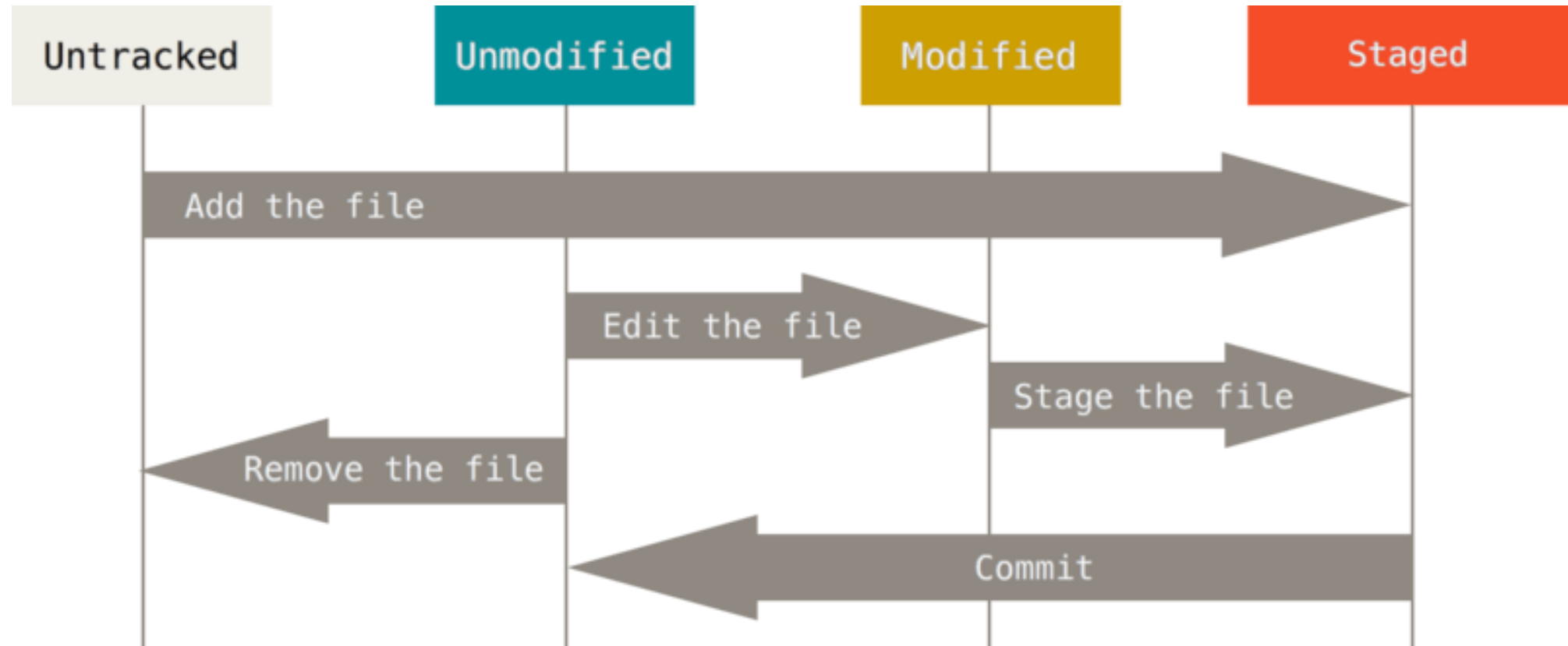
1. Tracked

- 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

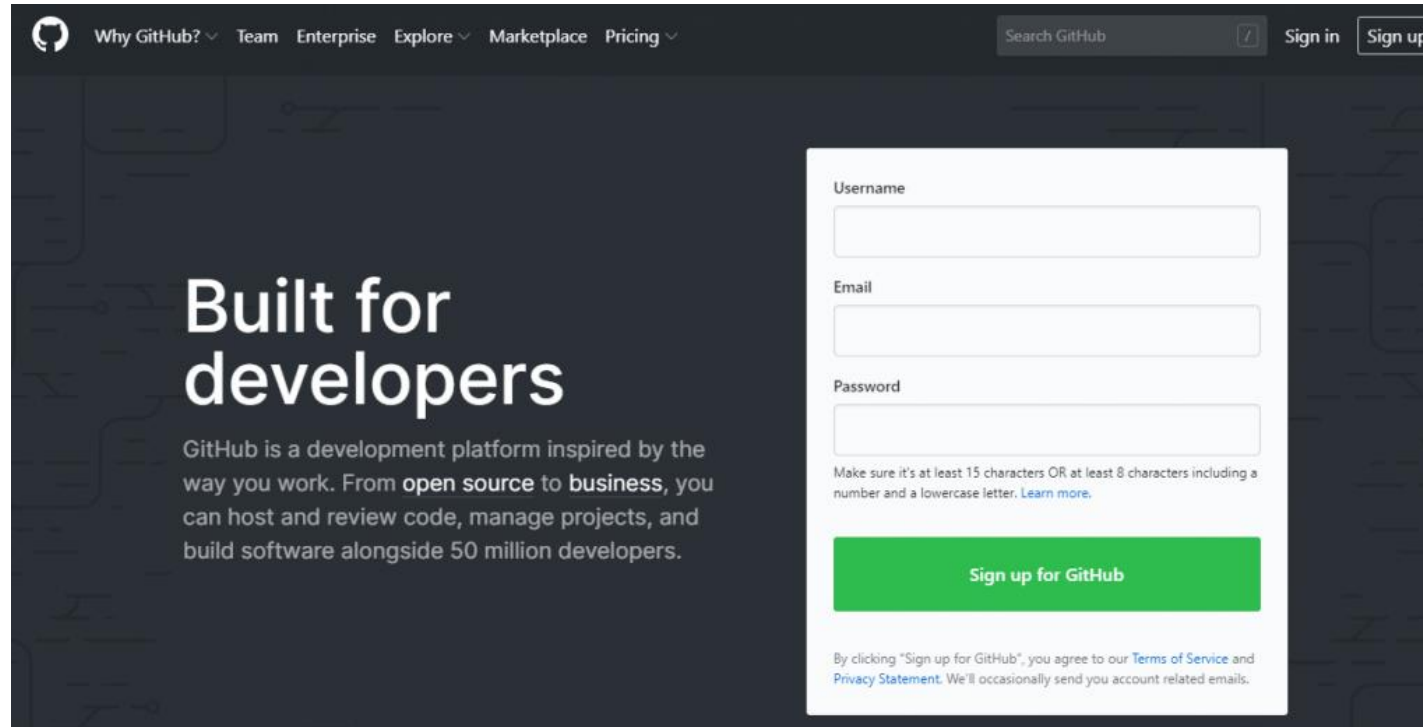
Git	GitHub
<ol style="list-style-type: none">1. <i>Install software</i> di penyimpanan lokal2. Dikelola oleh The Linux Foundation3. Fokus pada <i>version control</i> dan <i>code sharing</i>4. Akses <i>offline</i>5. Tidak menggunakan fitur <i>user management</i>6. <i>Desktop interface</i> bernama "Git GUI"7. Pesaing: Mercurial, Subversion, IBM, Rational Team, Concert, dan ClearCase8. <i>Open sourced licensed</i>	<ol style="list-style-type: none">1. <i>Host</i> melalui layanan <i>cloud</i>2. Diakuisisi oleh Microsoft pada 20183. Fokus pada <i>source code hosting</i> terpusat4. Akses <i>online</i>5. Menggunakan <i>user management</i>6. <i>Desktop interface</i> "GitHub Desktop"7. Pesaing: GitLab dan Atlassian BitBucket8. 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

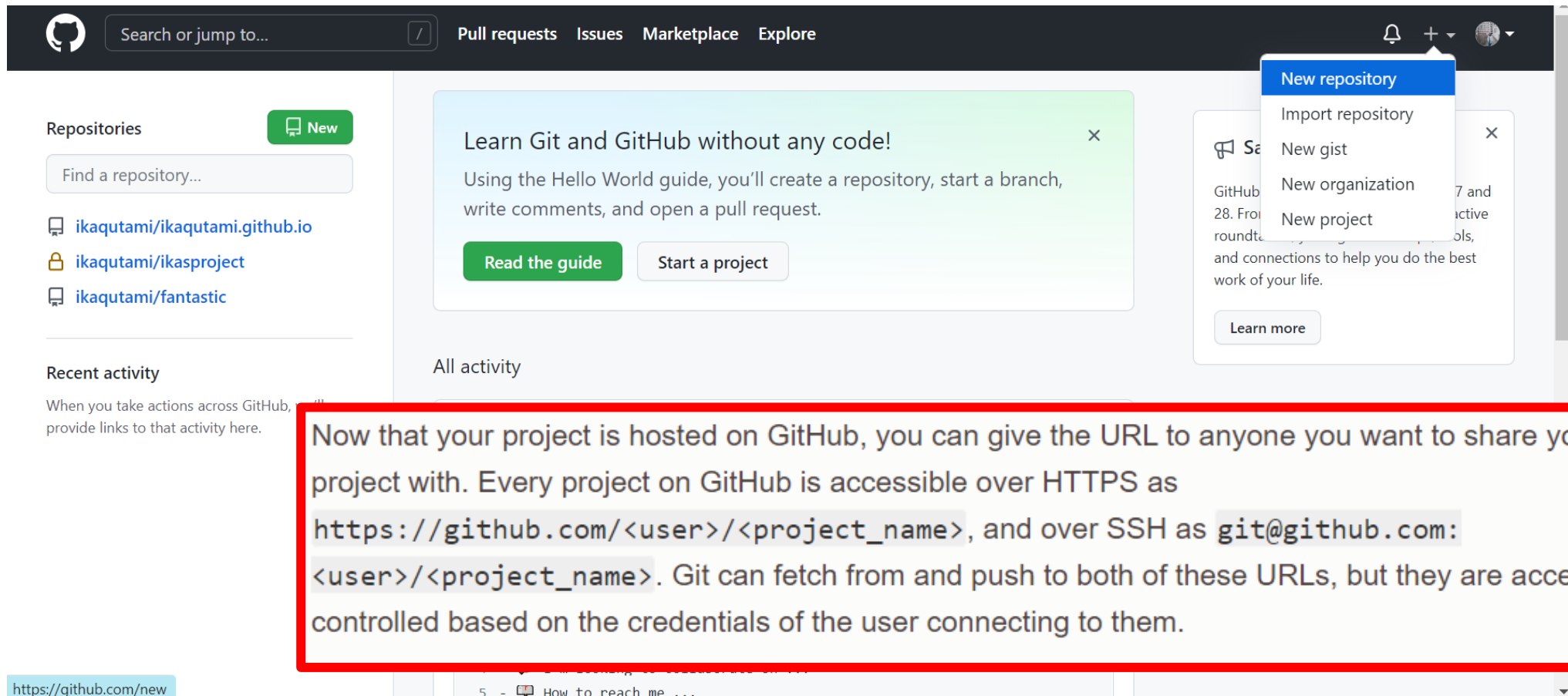


The screenshot shows the GitHub homepage with a sign-up overlay. The overlay contains the following elements:

- Navigation bar:** GitHub logo, links for Why GitHub?, Team, Enterprise, Explore, Marketplace, Pricing, a search bar, and Sign in / Sign up buttons.
- Main heading:** "Built for developers".
- Description:** "GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside 50 million developers."
- Form fields:**
 - Username:** A text input field.
 - Email:** A text input field.
 - Password:** A text input field with a note: "Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)"
- Sign up button:** A large green button labeled "Sign up for GitHub".
- Footer text:** "By clicking 'Sign up for GitHub', you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails."

Maintaining a Project

- Creating a New Repository



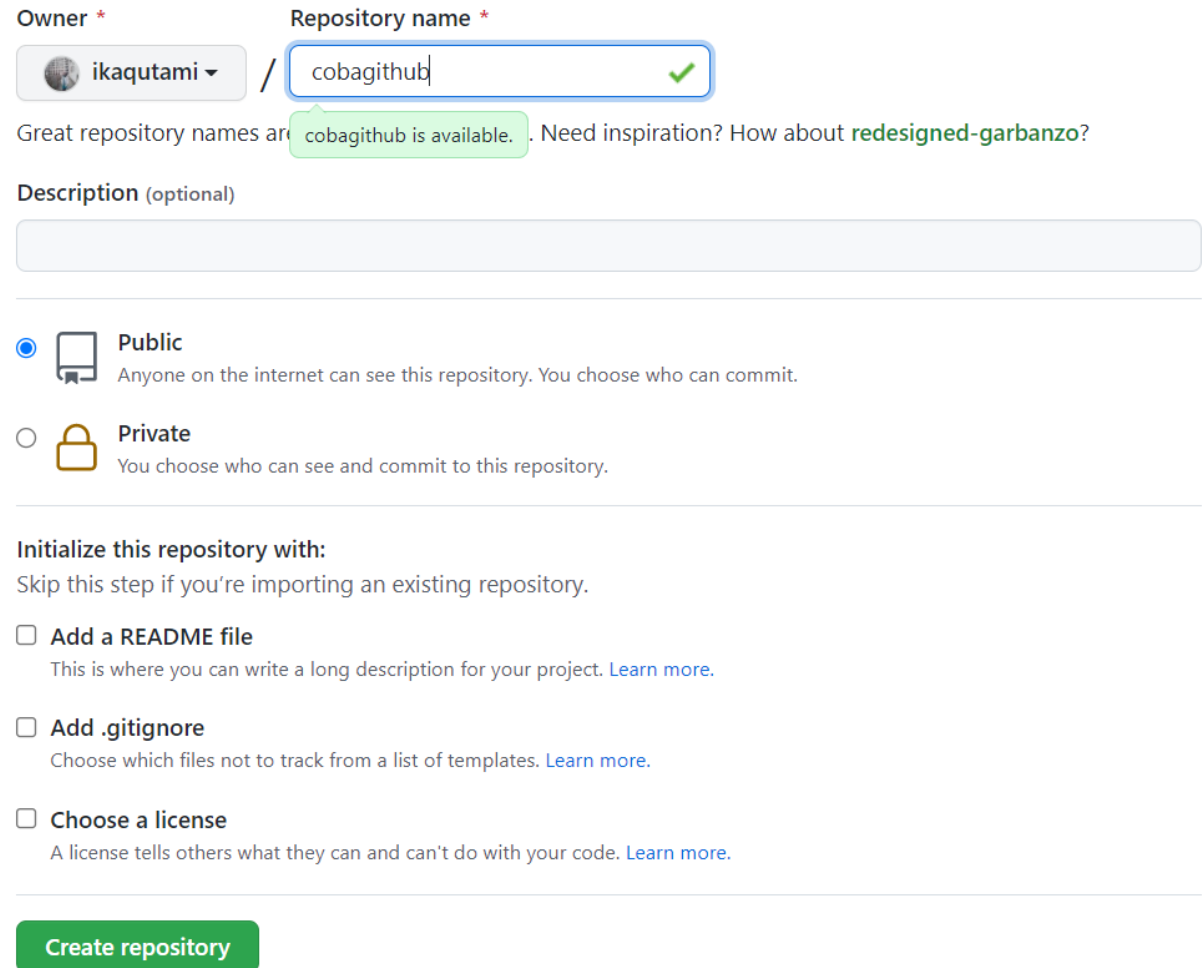
The screenshot shows the GitHub 'new' page. On the left, there's a sidebar with 'Repositories' and 'Recent activity'. The main content area has a green banner for 'Learn Git and GitHub without any code!' with buttons for 'Read the guide' and 'Start a project'. On the right, a dropdown menu is open from the '+' icon, showing options: 'New repository', 'Import repository', 'New gist', 'New organization', and 'New project'. Below this, there's a text box with a red border containing the following text:

Now that your project is hosted on GitHub, you can give the URL to anyone you want to share your project with. Every project on GitHub is accessible over HTTPS as `https://github.com/<user>/<project_name>`, and over SSH as `git@github.com:<user>/<project_name>`. Git can fetch from and push to both of these URLs, but they are access-controlled based on the credentials of the user connecting to them.

At the bottom left, the URL `https://github.com/new` is visible.

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.



The screenshot shows the GitHub repository creation interface. At the top, there are two fields: 'Owner' with a dropdown menu showing 'ikaqutami' and 'Repository name' with a text input containing 'cobagithub'. A green checkmark is visible next to the repository name. Below these fields, a message states: 'Great repository names are available. cobagithub is available. Need inspiration? How about redesigned-garbanzo?'. There is a 'Description (optional)' text area below this. The 'Public' option is selected with a radio button, and the 'Private' option is unselected. Below the visibility options, there is a section titled 'Initialize this repository with:' with the instruction 'Skip this step if you're importing an existing repository.' and three checkboxes: 'Add a README file', 'Add .gitignore', and 'Choose a license'. Each checkbox has a brief description and a 'Learn more' link. At the bottom, there is a green 'Create repository' button.

Owner * / Repository name *

ikaqutami / cobagithub ✓

Great repository names are available. cobagithub is available. Need inspiration? How about redesigned-garbanzo?

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.](#)

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

Create repository

Add Collaborators

The screenshot shows the GitHub repository settings page. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The Settings link is highlighted with a red box. On the left sidebar, the 'Options' menu is expanded, and the 'Manage access' option is highlighted with a red box. The main content area is titled 'Who has access' and shows two sections: 'PUBLIC REPOSITORY' and 'DIRECT ACCESS'. The 'DIRECT ACCESS' section indicates that 0 collaborators have access. Below this, the 'Manage access' section is titled, and a large box contains the message 'You haven't invited any collaborators yet' with an icon of a person and a lock. The 'Invite a collaborator' button is highlighted with a red box.

<> Code Issues Pull requests Actions Projects Wiki Security Insights **Settings**

Options

Manage access

Security & analysis

Webhooks

Notifications

Integrations

Deploy keys

Actions

Environments

Secrets

Pages

Moderation settings

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

[Manage](#)

DIRECT ACCESS

0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

You haven't invited any collaborators yet

Invite a collaborator

Creating new file

ikasproject /

in main


Cancel changes

<> Edit new file

Preview

Spaces 2 No wrap

1



Commit new file

deskripsi commit untuk mengetahui perubahan apa yang telah dilakukan

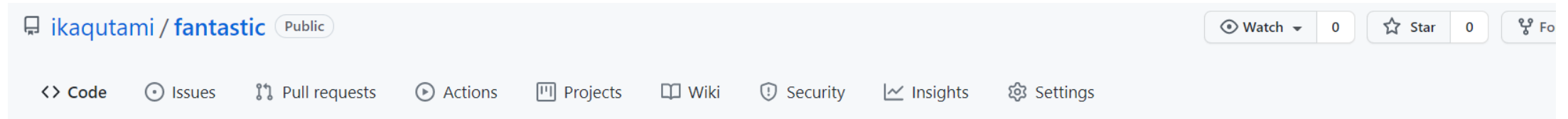
☒ Commit directly to the main branch.

☐ Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)



Commit new file

Cancel

Creating new file



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

 Set up in Desktop or HTTPS SSH `git@github.com:ikaqutami/fantastic.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 "# fantastic" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin git@github.com:ikaqutami/fantastic.git
git push -u origin master
```



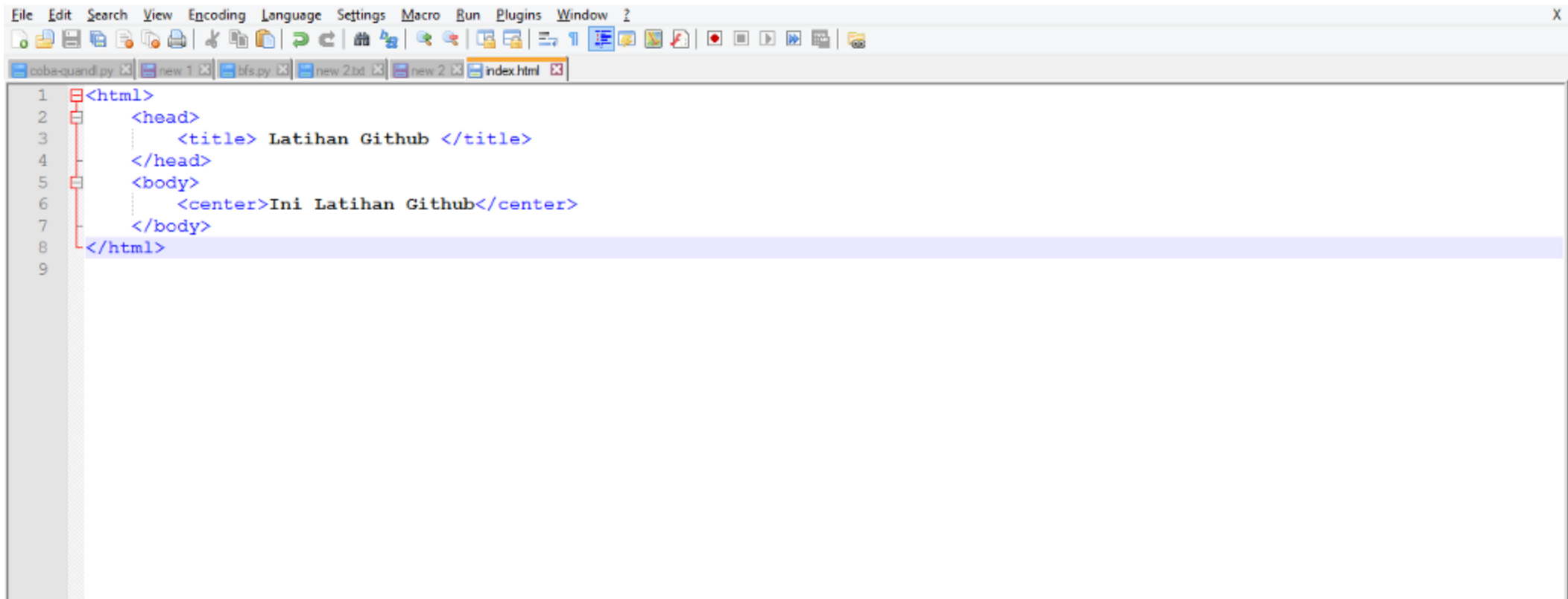
...or push an existing repository from the command line

```
git remote add origin git@github.com:ikaqutami/fantastic.git
git branch -M master
git push -u origin master
```



Push File

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

A screenshot of a code editor window. The title bar shows 'File Edit Search View Encoding Language Settings Macro Run Plugins Window ?'. The tab bar shows 'coba-quand.py', 'new 1', 'bfs.py', 'new 2bd', 'new 2', and 'index.html'. The editor area shows the following HTML code:

```
1 <html>
2   <head>
3     <title> Latihan Github </title>
4   </head>
5   <body>
6     <center>Ini Latihan Github</center>
7   </body>
8 </html>
```

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

```
1 <html>
2     <head>
3         <title> Latihan Github </title>
4     </head>
5     <body>
6         <center>Ini Latihan Github</center>
7     </body>
8 </html>
```

Showing 1 changed file with 1 addition and 1 deletion.

2 index.html

	@@ -3,6 +3,6 @@
3	3 <title> Latihan Github </title>
4	4 </head>
5	5 <body>
6	- <center>Ini Latihan Github</center>
6	+ <center>Ini Latihan Github Anak IT</center>
7	7 </body>
8	8 </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>