

perintah, dan aliran keseluruhan yang memungkinkan pengembang dan pengguna bekerja sama. GitHub bertindak sebagai "repositori jarak jauh" yang disebutkan sebelumnya di bagian **Git**.

Fitur utama yang disediakan oleh GitHub meliputi:

- Masalah
- Diskusi
- Tarik permintaan
- Notifikasi
- Label
- Tindakan
- Garpu
- Proyek

Untuk mempelajari lebih lanjut tentang **GitHub**, lihat [Memulai dengan GitHub](#).

## Masalah

**Masalah** adalah di mana sebagian besar komunikasi antara konsumen proyek dan tim pengembangan terjadi. Sebuah *masalah* dapat dibuat untuk membahas kumpulan topik yang luas, termasuk laporan bug, permintaan fitur, klarifikasi dokumentasi, dan banyak lagi. Setelah masalah dibuat, masalah tersebut dapat ditetapkan ke pemilik, label, proyek, dan pencapaian. Anda juga dapat mengaitkan masalah dengan permintaan penarikan dan item GitHub lainnya untuk menyediakan ketertelusuran di masa mendatang.

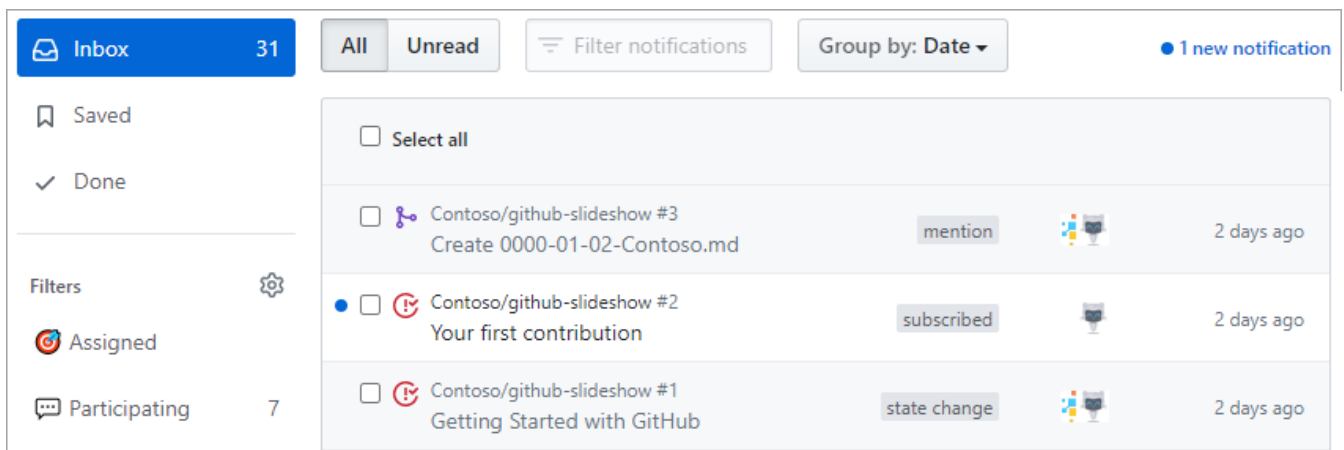


Untuk mempelajari lebih lanjut tentang Masalah GitHub, lihat [Menguasai Masalah](#).

# Notifikasi

Sebagai platform kolaboratif, GitHub menawarkan **notifikasi** untuk hampir setiap peristiwa yang terjadi dalam alur kerja tertentu. Notifikasi ini dapat disetel dengan halus untuk memenuhi preferensi Anda. Misalnya, Anda dapat berlangganan semua pembuatan dan

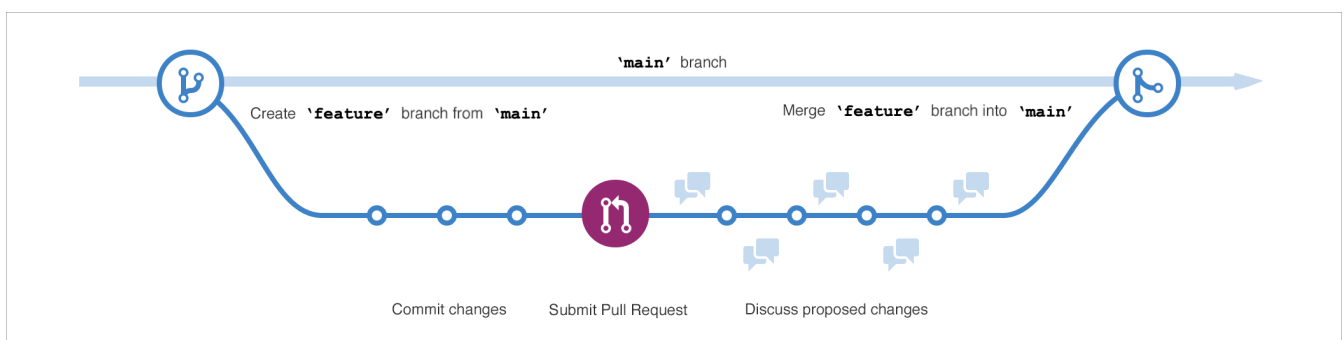
pengeditan masalah pada proyek, atau Anda hanya dapat menerima pemberitahuan untuk masalah yang menyebutkan Anda. Anda juga dapat memutuskan apakah Anda menerima notifikasi melalui email, web & seluler, atau keduanya. Untuk melacak semua notifikasi Anda di berbagai proyek, gunakan [dasbor GitHub Notifications](#).



Untuk mempelajari lebih lanjut tentang notifikasi GitHub, lihat [Mengonfigurasi notifikasi](#).

# Ranting

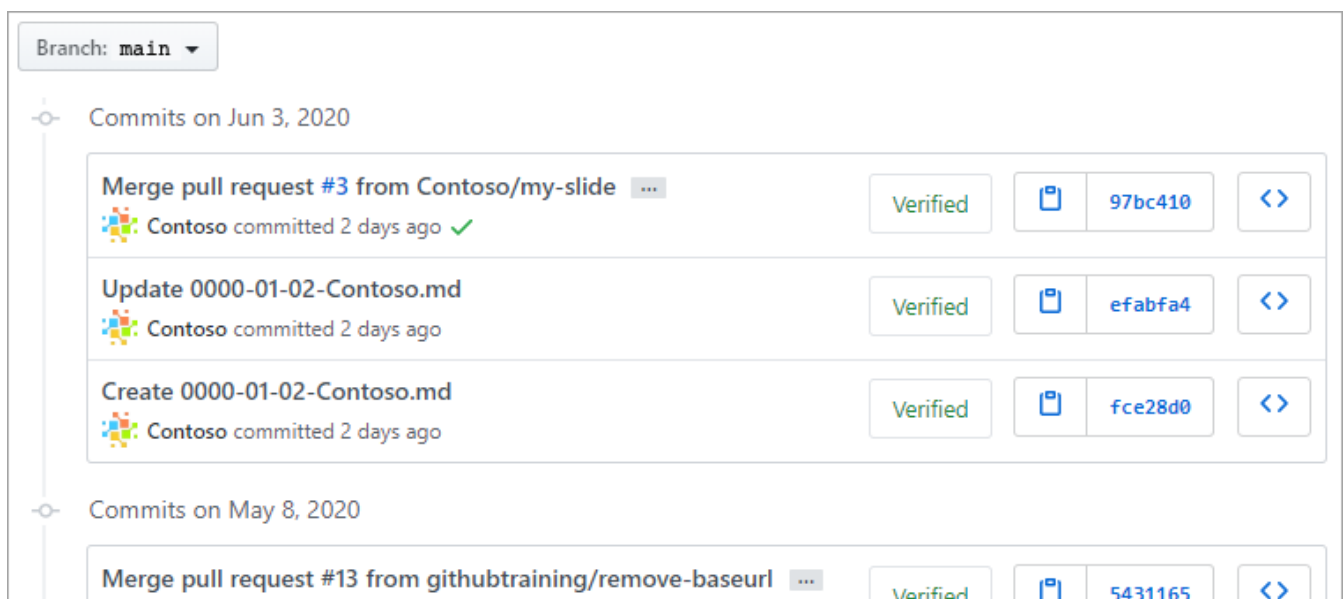
**Cabang** adalah cara yang disukai untuk membuat perubahan dalam [alur GitHub](#). Mereka menyediakan isolasi sehingga banyak orang dapat bekerja secara bersamaan pada kode yang sama dengan cara yang terkontrol. Model ini memungkinkan stabilitas di antara cabang-cabang penting, seperti `main`, sekaligus memungkinkan kebebasan penuh bagi pengembang untuk melakukan perubahan apa pun yang diperlukan untuk memenuhi tujuan mereka. Setelah kode dari cabang siap menjadi bagian dari `main` cabang, kode tersebut dapat digabungkan melalui permintaan tarik.



Untuk mempelajari lebih lanjut tentang cabang GitHub, lihat [Tentang cabang](#) .

## Komitmen

Komit adalah perubahan pada satu atau lebih file di cabang . Setiap kali komit dibuat, itu diberikan ID unik dan dilacak, bersama dengan waktu dan kontributor. Ini memberikan jejak audit yang jelas bagi siapa pun yang meninjau riwayat file atau item tertaut, seperti masalah atau permintaan penarikan.



Untuk mempelajari lebih lanjut tentang komitmen GitHub, lihat [Melakukan dan meninjau perubahan pada proyek Anda](#) .

## Tarik Permintaan

Permintaan **tarik** adalah mekanisme yang digunakan untuk memberi sinyal bahwa komit dari satu cabang siap digabungkan ke cabang lain. Pengembang yang mengirimkan **permintaan penarikan** akan sering meminta satu atau lebih peninjau untuk memverifikasi kode dan menyetujui penggabungan. Peninjau ini memiliki kesempatan untuk mengomentari perubahan, menambahkannya sendiri, atau menggunakan permintaan tarik itu sendiri untuk diskusi lebih lanjut. Setelah perubahan disetujui (jika persetujuan diperlukan), cabang sumber permintaan tarik (cabang pembanding) dapat digabungkan ke dalam cabang dasar.

The screenshot shows a GitHub pull request titled "Create 0000-01-02-Contoso.md #3". At the top, it indicates that "Contoso merged 2 commits into main from my-slide" 2 days ago. Below this, there are tabs for "Conversation" (4), "Commits" (2), "Checks" (0), and "Files changed" (1). The "Conversation" tab is active, showing a comment from "Contoso" (Owner) stating "Adding a much-needed file." Below this, a commit "Create 0000-01-02-Contoso.md" is shown as "Verified" with hash "fce28d0". A comment from "github-learning-lab" (bot) requests changes, stating: "Good pull requests have a body description that tells other contributors about the change you're suggesting, so they understand the context. Let's edit this pull request to add a body description." Below the comment is an "Activity: Fixing your pull request" section with a numbered list: "1. The first comment on your pull request will have the default text of **No description provided**. Click on the ... icon located at the top right corner of the comment box, then click on **Edit** to make an edit". On the right side, there are sections for "Reviewers" (github-learning-lab with a green checkmark), "Assignees" (No one—assign yourself), "Labels" (None yet), "Projects" (None yet), "Milestone" (No milestone), and "Linked issues" (Successfully merging this pull request may close these issues. None yet).

Untuk mempelajari lebih lanjut tentang permintaan tarik GitHub, lihat [Tentang permintaan tarik](#).

## Label

Labels provide a way to categorize and organize **issues** and **pull requests** in a repository. As you create a GitHub repository several labels will automatically be added for you and new ones can also be created.

Examples of Labels include:

- bug
- documentation
- duplicate
- help wanted
- enhancement
- question

## Getting Started with GitHub #1

Edit

New issue

🕒 Open github-learning-lab bot opened this issue now · 1 comment



github-learning-lab bot commented now

😊 ...



### Welcome to GitHub Learning Lab's "Introduction to GitHub"

To get started, I'll guide you through some important first steps in coding and collaborating on GitHub.

👉 This arrow means you can expand the window! Click on them throughout the course to find more information.

► What is GitHub?

► Exploring a GitHub repository

#### Using issues

This is an issue 🗨️: a place where you can have conversations about bugs in your code, code review, and just about anything else.

Issue titles are like email subject lines. They tell your collaborators what the issue is about at a glance. For example, the title of this issue is Getting Started with GitHub.

► Using GitHub Issues

► Managing notifications

Keep reading below to find your first task



github-learning-lab bot commented now

Author 😊 ...

### Step 1: Assign yourself

Unassigned issues don't have owners to look after them. When you're assigned to an issue or pull request, it tells repository visitors and contributors that you'll be facilitating the conversation or task 🙋.

Assignees

No one—assign yourself



Labels



Apply labels to this issue

Filter labels



bug  
Something isn't working



documentation  
Improvements or additions to documentation



duplicate  
This issue or pull request already exists



enhancement  
New feature or request



good first issue  
Good for newcomers



help wanted  
Extra attention is needed



invalid  
This doesn't seem right



question  
Further information is requested



Edit labels

📌 Pin issue ⓘ

➔ Transfer issue

🗑️ Delete issue

To learn more about GitHub labels see [About labels](#).

## Actions

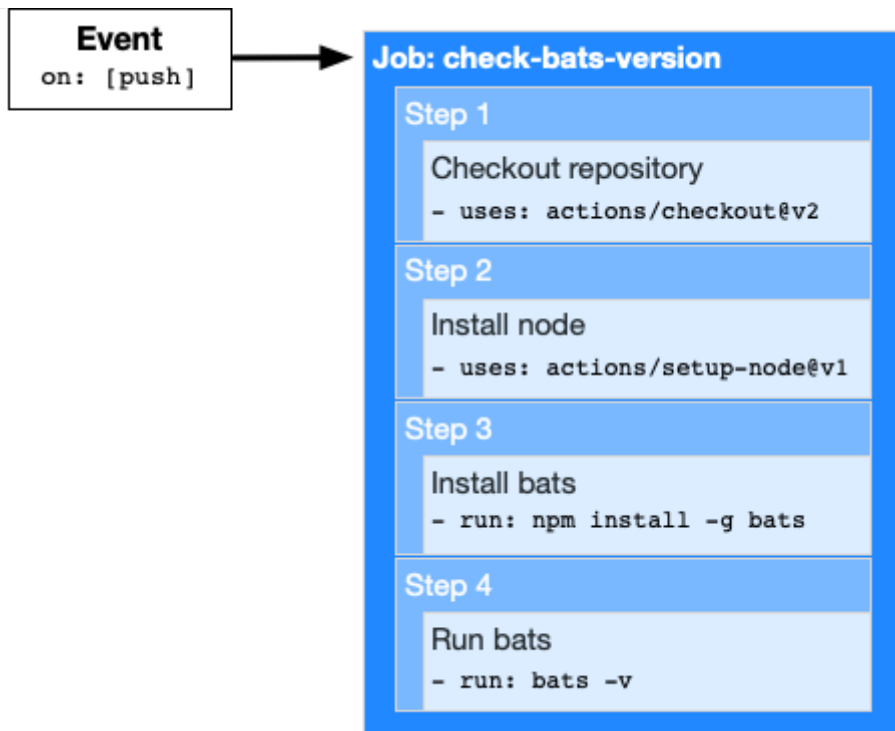
**GitHub actions** provide task automation and workflow functionality in a repository. Actions can be used to streamline processes in your software development lifecycle and implement continuous integration and continuous deployment (CI/CD).

GitHub Actions are composed of the following components:

- **Workflows:** Automated processes added to your repository.
- **Events:** An activity that triggers a workflow.
- **Jobs:** A set of steps that execute on a runner.
- **Steps:** A task that can run one or more commands (actions).
- **Actions:** Standalone commands that can be combined into steps. Multiple steps can be

combined to create a job.

- **Runners:** Server that has the GitHub Actions runner application installed.



To learn more about GitHub actions see [Introduction to GitHub Actions](#) .

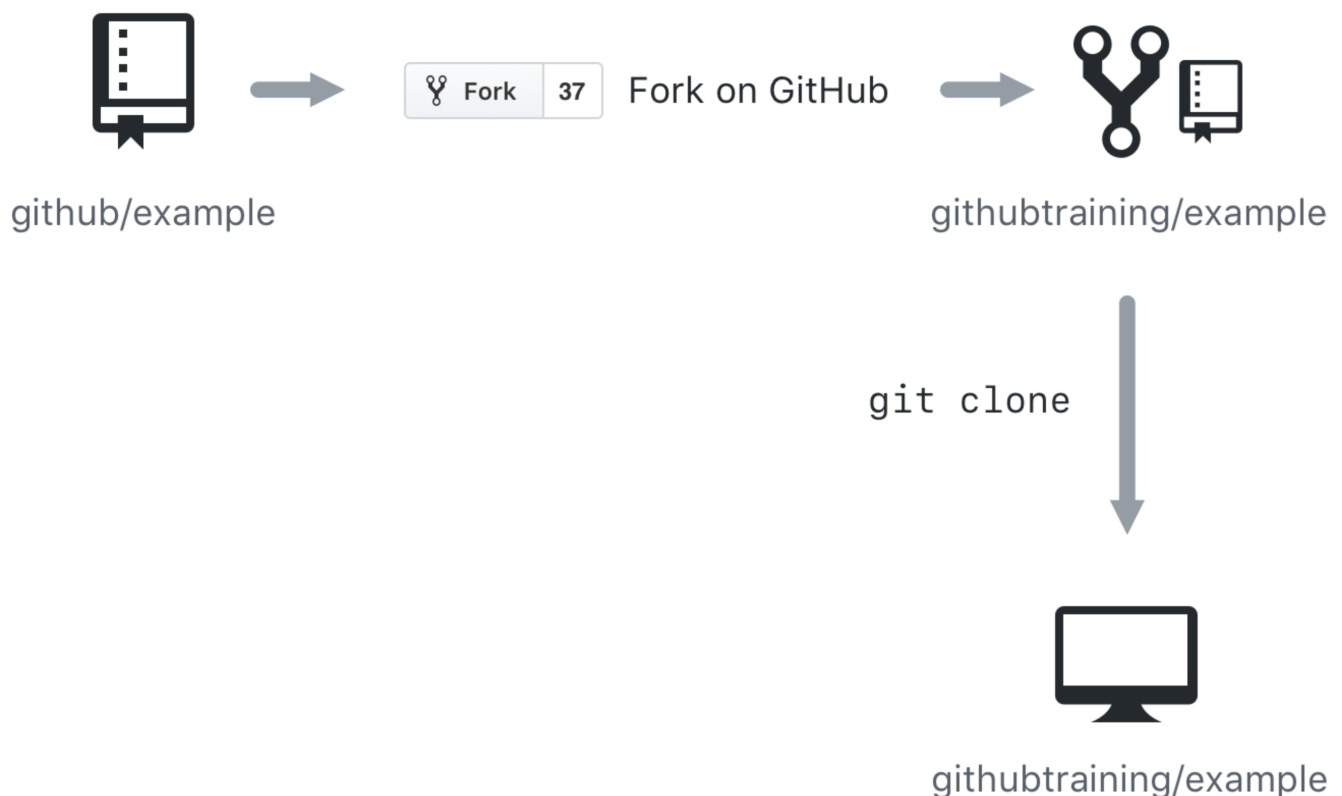
## Cloning and forking

GitHub provides multiple ways to copy a repository so that you can work on it.

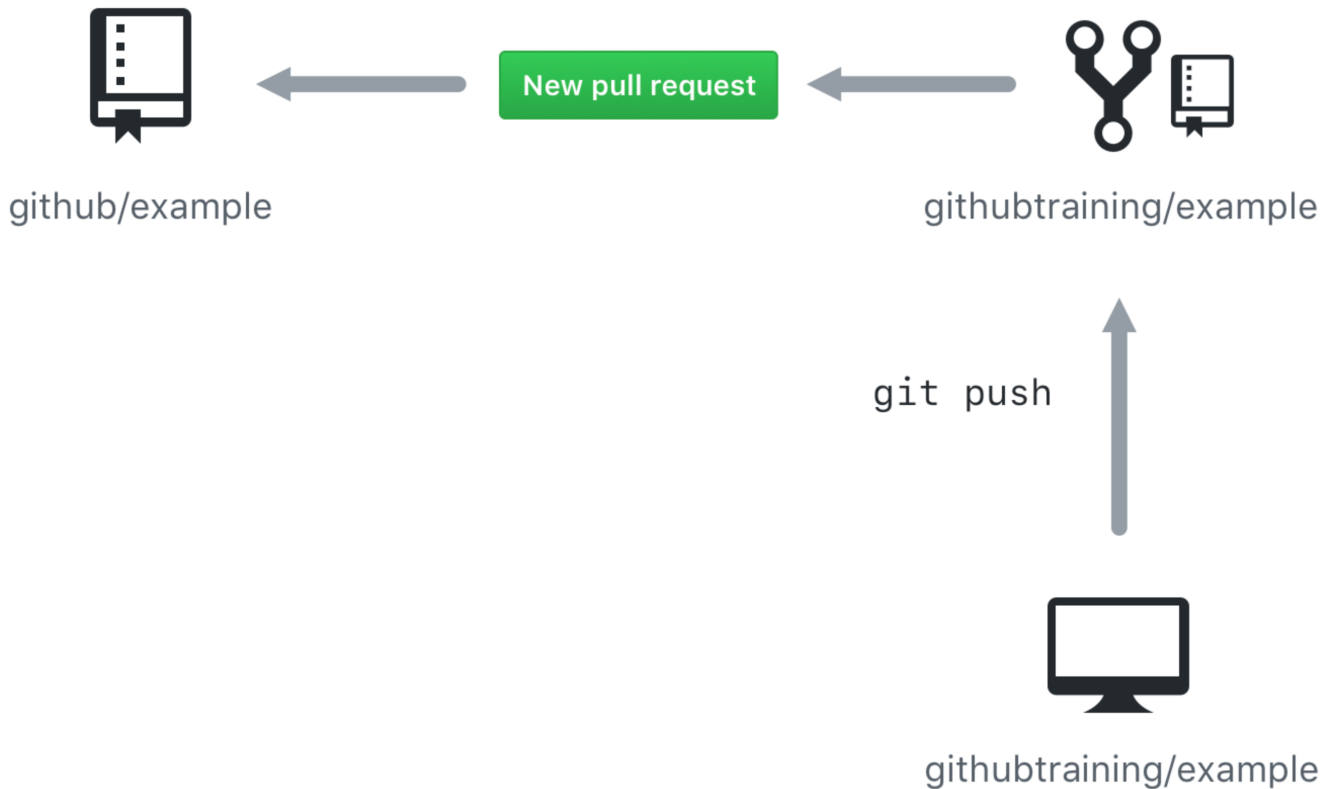
- **Cloning a Repository** - Cloning a repository will make a copy of the repository and its history on your local machine. If you have write access to the repository, you can push changes from your local machine to the remote repository (called the **origin**) as they're completed. To clone a repository, you can use the `git clone [url]` command or the GitHub CLI's `gh repo clone [url]` command.
- **Forking a Repository** - Forking a repository makes a copy of the repository in your GitHub account. The parent repository is referred to as the **upstream** while your forked copy is referred to as the **origin**. Once you've forked a repository into your GitHub account, you can **clone** it to your local machine. Forking allows you to freely make changes to a project without affecting the original **upstream** repository. To contribute changes back to the upstream repository, you create a **pull request** from your forked repository. You can also run `git` commands to ensure that your local copy stays synced with the **upstream** repository.

When would you clone a repository versus fork a repository? If you're working with a repository and have write access, you can clone it to your local machine. From there you can make modifications and push your changes directly to the **origin** repository.

If you need to work with a repository created by another owner such as `github/example` and don't have write access, you can fork the repository into your GitHub account, and then clone the fork to your local machine. To see this visually, let's assume that your GitHub account is called `githubtraining`. Using the GitHub website you can fork `github/example` or any other repository into your account. From there, you can clone the forked version of the repository to your local machine. These steps are shown in the following image.



Perubahan dapat dilakukan pada salinan lokal Anda `githubtraining/example` dan kemudian didorong kembali ke repositori **asal** `githubtraining/example` jarak jauh Anda ( ). Perubahan kemudian dapat dikirimkan ke repositori `github/example` **upstream** menggunakan **pull request** seperti yang ditunjukkan berikut ini.

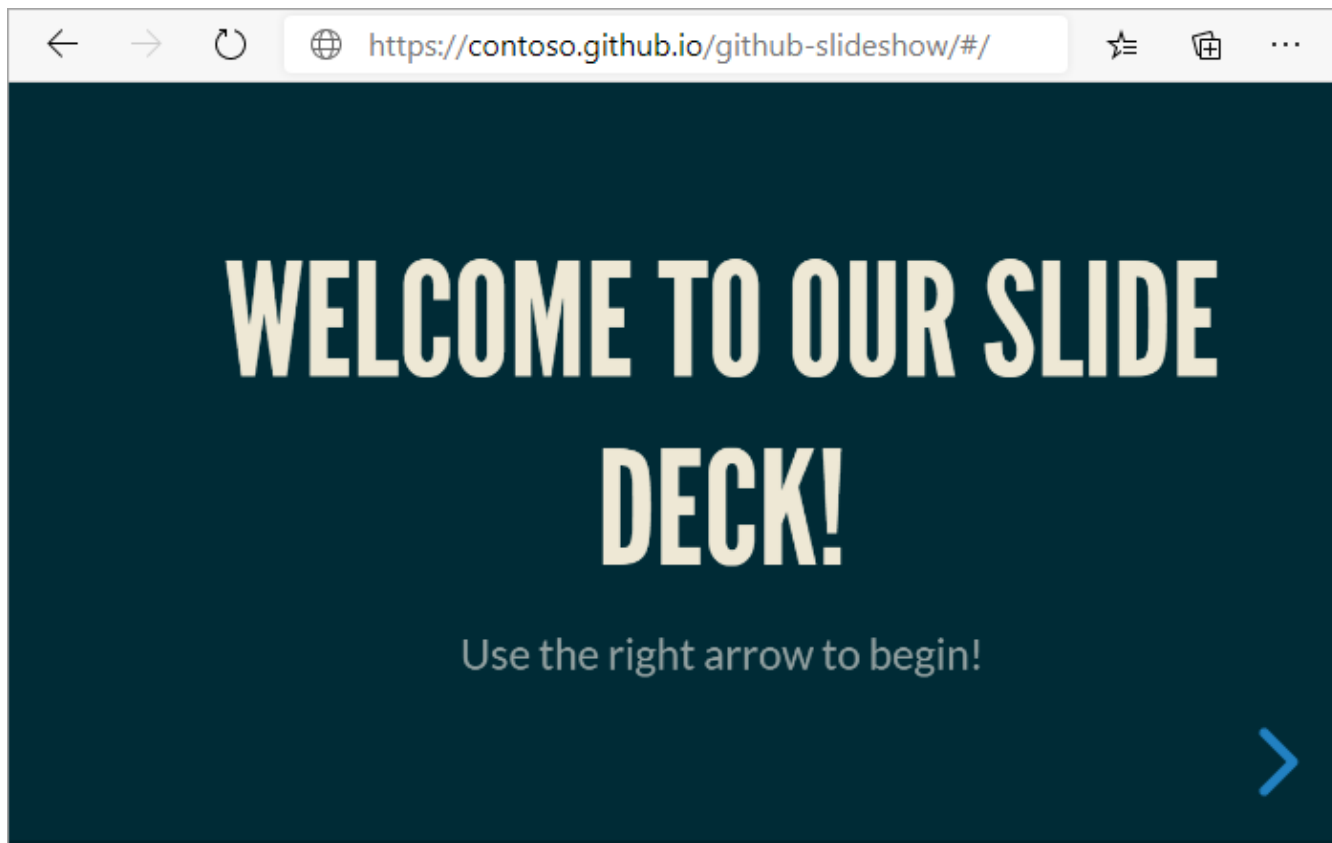


Untuk mempelajari lebih lanjut, lihat [Fork a repo](#) .

## Halaman GitHub

**Halaman GitHub** adalah mesin hosting yang dibangun langsung ke akun GitHub Anda. Dengan mengikuti beberapa konvensi, dan mengaktifkan fitur tersebut, Anda dapat membangun situs statis Anda sendiri yang dihasilkan dari HTML dan kode penurunan harga yang diambil langsung dari repositori Anda.





Untuk mempelajari lebih lanjut, lihat [Halaman GitHub](#) .

## Next unit: Exercise - A guided tour of GitHub

[Continue >](#)

How are we doing? ☆ ☆ ☆ ☆ ☆

