

Belajar Github

Enggar Alfianto

December 13, 2017

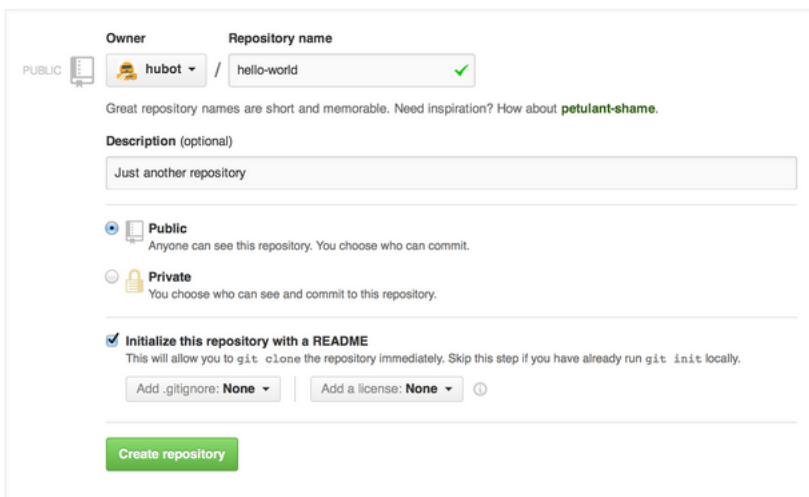
1 Membuat Repositori baru

Repository biasanya digunakan untuk mengatur proyek tunggal. Repository dapat berisi folder dan file, gambar, video, spreadsheets dan data set - atau apa saja yang dibutuhkan dalam proyek tersebut. Direkomendasikan terdapat file README atau file yang memiliki informasi apapun tentang proyek yang dibuat.

Repository "hello-world" dapat diletakkan dimanapun.

Membuat repository baru

1. Pada bagian pojok kanan atas, klik tanda "+" dan pilih **New repository**.
2. Namai repository baru dengan "hello-world"
3. Tulis deskripsi singkat
4. Centang **Initialize this repository with a README**.



The screenshot shows the GitHub interface for creating a new repository. At the top, there's a 'PUBLIC' label and a 'hubot' owner. The 'Repository name' field is filled with 'hello-world' and has a green checkmark. Below this, a hint says 'Great repository names are short and memorable. Need inspiration? How about **petulant-shame**.' The 'Description (optional)' field contains 'Just another repository'. Under 'Visibility', 'Public' is selected with a radio button. Below that, 'Private' is also an option. The 'Initialize this repository with a README' checkbox is checked. At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None'. A green 'Create repository' button is at the bottom.

5. Klik **Create repository**

2 Membuat cabang (*branch*)

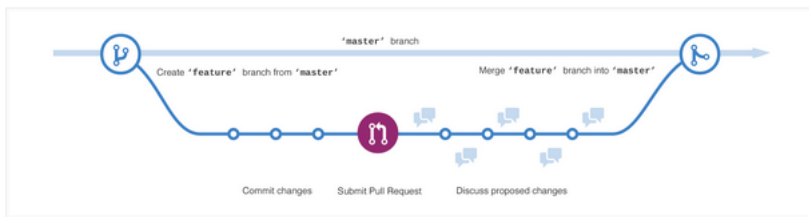
branch merupakan cara untuk bekerja dengan banyak versi repository pada saat yang sama

Semula, repository hanya memiliki satu cabang dengan nama "master" yang merupakan cabang utama. Kita menggunakan branch untuk melakukan percobaan dan melakukan perubahan sebelum memasukkannya (*committing*) ke dalam "master".

ketika kita membuat cabang dari master, artinya kita membuat duplikat dari master. Jika ada orang lain yang bekerja pada master dan kita bekerja pada branch, kita harus menarik (pull) update tersebut.

Diagram ini menunjukkan:

- cabang (*branch*) "master"
- Cabang baru yang bernama "feature" (karena kita akan bekerja pada cabang itu)
- berikutnya "feature" akan digabung kedalam cabang "master"



Sering kali kita membuat versi file berbeda saat bekerja, misalnya:

- Tugas.txt
- Tugas-enggar-edit.txt
- Tugas-enggar-edit-perbaikan.txt

Cabang (*branch*) mengerjakan hal yang sama pada repository GitHub.

Di GitHub, developer, writer, dan desainer menggunakan branch untuk menjaga perbaikan bug dan pekerjaan berikutnya dari cabang "master". Ketika perubahan telah siap dan bagus mereka menyatukannya kembali dalam cabang "master".

Membuat cabang (*branch*) baru

1. Masuk ke dalam repository "hello-world"
2. klik pilihan "branch:master"
3. ketikkan nama cabang baru "readme-edits"
4. Pilih tombol biru **Create branch** atau tekan enter pada keyboard.

Saat ini telah tersedia dua branch yaitu "master" dan "readme-edits".

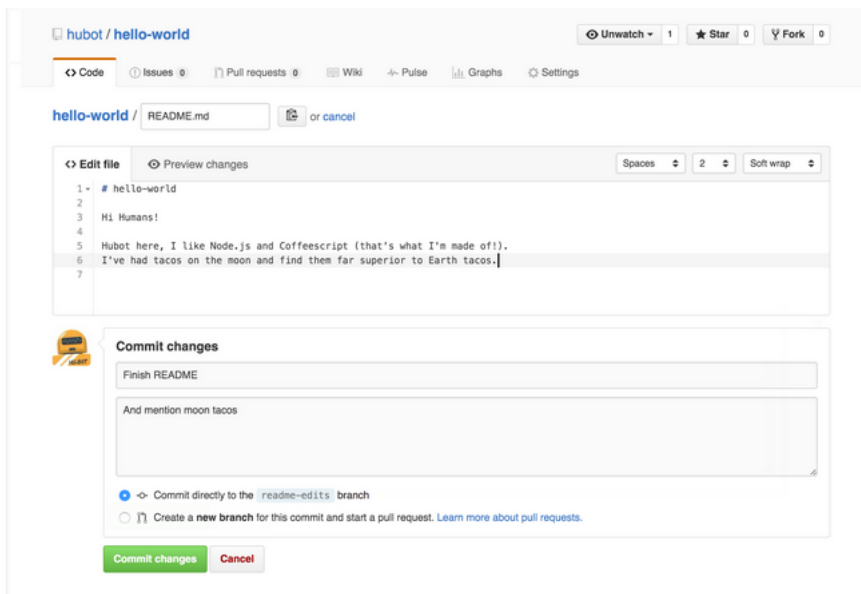
3 Membuat dan memasukkan (*commit* perubahan

Mantab, saat ini kita berada dalam tampilan cabang "readme-edits", yang mana adalah copy dari "master". Buatlah sedikit perubahan.

Pada GitHub, menyimpan perubahan disebut "commits", setiap "commit" memiliki pesan, yang merupakan deskripsi dan penjelasan mengapa ada perubahan yang dilakukan. Pesan commit menyimpan sejarah dari perubahan yang dilakukan, sehingga kontributor lain dapat memahami tentang apa yang telah diselesaikan dan mengapa hal itu dilakukan.

Membuat dan memasukkan (*commit*) perubahan

1. Klik "README.md"
2. klik gambar pensil pada bagian sudut kanan atas pada tampilan file untuk mengedit
3. Pada editor, tulis sembarang tentang dirimu.
4. tulis pesan commit yang mendeskripsikan perubahan
5. klik tombol **Commit changes**



Perubahan ini hanya dilakukan di dalam file README yang berada pada cabang "readme-edits", sehingga sekarang ada perbedaan antara cabang "master" dan "readme-edits"

4 Buka permintaan tarik (*pull*)

Sekarang kita telah berada pada cabang "master", kita dapat membuka permintaan penarikan (*pull*).

Permintaan "pull" merupakan jantung dari kerja secara kolaborasi di GitHub. Ketika kita membuka permintaan pull, itu artinya kita mengajukan perubahan dan meminta seseorang untuk mereview dan menarik kontribusi kita kemudian disatukan dengan cabang mereka. Perubahan, penambahan dan pengurangan ditunjukkan dengan warna hijau dan merah.

Segera setelah kita membuat commit, kita dapat membuka permintaan pull dan memulai diskusi, meskipun code belum terselesaikan.

Dengan menggunakan GitHub @mentionSystem pada pesan permintaan pull, kita dapat meminta timbal balik dari seorang atau tim.

Kita dapat menarik *pull* permintaan pada repositori sendiri dan menggabungkannya. Itu adalah cara terbaik untuk memahami alur Github sebelum bekerja pada proyek besar.

Membuka permintaan Pull (*Pull Request*)

1. Klik tab **Pull Request**, kemudian dari halaman Pull Request klik tombol hijau **Pull Request**.

2. Pada bagian **Example Comparisons**, pilih cabang yang telah dibuat, "readme-edits", untuk dibandingkan dengan "master" (aslinya)
3. Lihat perubahan yang telah dibuat dari laman perbandingan, pastikan itu adalah yang akan disubmit.
4. Jika sudah mantab dengan perubahan yang telah dibuat, klik tombol besar berwarna hijau bertuliskan **Pull Request**.
5. berilah judul pada *pull request yang telah dibuat*, berilah catatan kecil tentang perubahan yang dilakukan.
6. Jika sudah selesai dengan pesannya, klik **Create Pull Request**.

5 Menyatukan Pull Request

Pada bagian terakhir ini, akan digabungkan cabang "readme-edits" dan cabang "master".

1. Klik tombol hijau **Merge pull request** untuk mengubah cabang "master"
2. Klik **confirm merge**
3. Hapus cabang yang sudah digabung.

6 Git secara remote

Git merupakan aplikasi yang dapat diinstal di masing-masing PC untuk kepentingan sendiri. Namun, jika ingin bekerja sama dengan pengembang lain maka disediakan github. Dengan alamat url "https://github.com".

Untuk memulai bekerja secara remote, buatlah akun di "https://github.com". Kemudian buat repository baru dengan mengetuk tanda "+" di pojok kanan atas, kemudian pilih "New repository". Atau bisa juga memilih tombol hijau bertuliskan "New Repository" yang ada di bagian kanan tengah. Lihat Gambar 6.

Tekan tombol hijau bertuliskan create repository.

Berikut adalah tatacara untuk membuat repository baru secara remote

```
echo "# tes" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/enggaralfi/tes.git
git push -u origin master
```

Jika sebelumnya sudah memiliki repository yang sudah matang dan tinggal mengupload saja, silakan ikuti perintah berikut.

```
git remote add origin https://github.com/enggaralfi/tes.git
git push -u origin master
```

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 enggaralfi ▾

Repository name

tes 

Great repository names are short and memorable. Need inspiration? How about **studious-spork**.

Description (optional)



Public

Anyone can see this repository. You choose who can commit.




Private

You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾ 

Create repository

Figure 1: Repository