

Mengenal Git dan GitHub

Adityo Pratomo (@kotakmakan)

Git dan GitHub

Git adalah version control system, digunakan untuk memantau proses perkembangan kode dalam sebuah lumbung (repository)

Keberadaan Git, membuat para pembuat software, bisa berkolaborasi, dengan perubahan kode yang selalu tersinkronisasi, sehingga masing-masing bisa selalu mendapat kode terbaru

GitHub adalah layanan Git berbasis web

GitHub dan Open Source

Keberadaan layanan Git seperti GitHub, Gitlab dan BitBucket, sangat penting bagi eksistensi berbagai proyek open source.

Di sinilah, kumpulan kode terkumpul, dikerjakan secara kolaboratif dan bisa dibaca serta diulas oleh publik.

Pada dasarnya, saat sebuah proyek sudah diumumkan berbentuk open source, semua orang di dunia berhak dan mampu untuk berkontribusi.

Open Source

"In open source, you don't wait. You contribute"

Berkontribusi di Proyek Open Source

- Kreator / Inisiator
- Maintainer
- Kontributor

Berkontribusi di Proyek Open Source

- Amati issue yang ada di repo upstream (opsional)
- Buat fork
- Buat branch untuk perubahan baru (fitur baru, bug fix, dsb.)
- Tambahkan perubahan di branch tersebut
- Pull Request dari branch di fork kita, ke repo upstream

Amati Issue

- Issue tertulis di tab Issue di sebuah repo GitHub
- Gunakan fitur ini untuk berdiskusi lebih detail
- Jika issue sudah di-assign ke dirimu, berarti kamu bertanggung jawab untuk menyelesaikannya

Membuat Fork

- Di setiap repository GitHub, ada tombol fork di kanan atas, klik saja tombol ini
- Fork repository akan muncul sebagai repository milik akunmu
- Kita akan bekerja di repository fork ini, supaya repository upstream tidak terganggu
- Clone fork ini ke komputer kita, dengan perintah

```
git clone {url repository}
```


Membuat Branch

- Pada dasarnya, ada 1 branch utama yang akan selalu memiliki versi kode terbaik, ini disebut branch `master` / `main`
- Supaya branch ini tidak terganggu, maka setiap penambahan fitur dikerjakan di branch baru
- Branch bisa dibuat dengan perintah `git checkout -b {nama branch}`

Menambahkan Perubahan

- Selanjutnya kita bisa bekerja dan menambahkan perubahan sesuai yang diinginkan
- Kita bisa menyimpan perubahan ini dengan melakukan stage
`git add {nama file yang berubah}`
- Setelah semua perubahan tersimpan, kita bisa mengupdate branch dengan melakukan commit `git commit`
- Jika sudah yakin, kita bisa mengupdate perubahan di branch lokal supaya masuk ke repository remote dengan push
`git push {nama repository remote} {nama branch remote}`
- Misal: `git push my-remote new-feature`

Menambahkan Perubahan

- Saat sedang menambahkan perubahan, seringkali kita harus mensinkronkan kode dengan kode terbaru di repository upstream atau bahkan dengan branch lain
- Untuk itu kita bisa melakukan git pull untuk mensinkronkan perubahan ini
- ``git pull {nama repository remote} {nama branch remote}`

Membuat Pull Request

- Setelah kita menambahkan kode ke branch baru di repository fork kita, maka sebuah banner akan muncul
- Kita bisa mengklik tombol Pull Request untuk mengajukan Pull Request ke repository upstream
- Jangan lupa tambahkan detail mengapa Pull Request ini diperlukan, apakah ada issue spesifik yang diselesaikan, atau apakah ada penyebab lain

Code Review

- Saat menerima sebuah Pull Request, biasanya seorang maintainer akan memeriksa kode yang masuk, proses ini lazim disebut **Code Review**
- Selama proses ini, diskusi akan berlangsung untuk memperjelas penambahan kode
- Ini dilakukan untuk memastikan agar kode yang masuk tidak menimbulkan dampak yang tak diinginkan
- Dari sini, kode bisa diterima, atau harus diubah untuk diterima, atau bahkan ditolak

Hands On

- Cek repository <https://github.com/lunchboxav/git-practice>
- Cek issue, tambahkan issue baru
- Buat fork
- Kumpulkan pull request untuk menjawab issue