1. Apa itu distributed versioning control?

<https://about.gitlab.com/topics/version-control/benefits-distributed-version-control-system/>

Sistem kontrol versi terdistribusi (DVCS) membawa salinan lokal dari repositori lengkap ke komputer setiap anggota tim, sehingga mereka dapat melakukan, mencabangkan, dan menggabungkan secara lokal. Server tidak harus menyimpan file fisik untuk setiap cabang — server hanya memerlukan perbedaan antara setiap penerapan.

Sistem manajemen kode sumber terdistribusi, seperti Git, Mercurial, dan Bazaar, mencerminkan repositori dan seluruh riwayatnya sebagai salinan lokal pada hard drive individual.

Sistem kontrol versi terdistribusi membantu tim pengembangan perangkat lunak menciptakan alur kerja dan hierarki yang kuat, dengan masing-masing pengembang mendorong perubahan kode ke repositori mereka sendiri dan pengelola mengatur proses peninjauan kode untuk memastikan hanya kode berkualitas yang digabungkan ke dalam repositori utama.

DVCS dapat membingungkan, terutama jika anggota tim terbiasa dengan sistem kode sumber terpusat , karena kontributor tidak dapat lagi mengandalkan server untuk menyelesaikan konflik saat menggabungkan dan harus menyelesaikannya secara lokal, yang dapat mengakibatkan komitmen penggabungan yang membingungkan. Namun, meskipun ada ketidaknyamanan awal, sistem kontrol sumber terdistribusi dapat memastikan pengembangan kode yang stabil ketika banyak pengembang berkontribusi pada proyek pengembangan perangkat lunak.

1. Apa itu git?

<https://www.petanikode.com/git-untuk-pemula/>

Git adalah salah satu sistem pengontrol versi (Version Control System) pada proyek perangkat lunak yang diciptakan oleh Linus Torvalds.

Pengontrol versi bertugas mencatat setiap perubahan pada file proyek yang dikerjakan oleh banyak orang maupun sendiri.

Git dikenal juga dengan distributed revision control (VCS terdistribusi), artinya penyimpanan database Git tidak hanya berada dalam satu tempat saja.

1. Cara membuat akun di github!

<https://learn.microsoft.com/id-id/visualstudio/version-control/git-create-github-account?view=vs-2022>

1. Buka [https://github.com](https://github.com/) di browser web, lalu pilih **Daftar**.
2. **Masukkan alamat email** Anda.
3. **Buat kata sandi** untuk akun GitHub baru Anda, dan **Masukkan nama** pengguna juga. Selanjutnya, pilih apakah Anda ingin menerima pembaruan dan pengumuman melalui email, lalu pilih **Lanjutkan**.
4. **Verifikasi akun** Anda dengan memecahkan teka-teki. Pilih tombol **Mulai Teka-teki** untuk melakukannya, lalu ikuti perintah.
5. Setelah Anda memverifikasi akun Anda, pilih tombol **Buat akun** .
6. Selanjutnya, GitHub mengirimkan kode peluncuran ke alamat email Anda. Ketik kode peluncuran tersebut **dalam dialog Masukkan kode** , lalu tekan **Enter**.
7. GitHub mengajukan beberapa pertanyaan untuk membantu menyesuaikan pengalaman Anda. Pilih jawaban yang berlaku untuk Anda dalam dialog berikut:

* Berapa banyak anggota tim yang akan bekerja dengan Anda?
* Fitur spesifik apa yang Anda minati untuk digunakan?

1. Di layar Tempat tim berkolaborasi dan mengirim, Anda dapat memilih apakah Anda ingin menggunakan akun Gratis atau akun Tim. Untuk memilih akun Gratis , pilih tombol Lewati personalisasi .
2. Bagaimana cara untuk membuat repository? Public

<https://learn.microsoft.com/id-id/visualstudio/version-control/git-create-repository?view=vs-2022>

1. Buka Visual Studio, lalu pilih **Buat Proyek** baru.
2. Dari **menu Git**, pilih **Buat Repositori** Git.
3. Dalam dialog **Buat Repositori** Git, di bawah Bagian **Dorong ke jarak jauh** baru, pilih **Azure DevOps.**
4. Di bagian **Buat Repositori** Azure DevOps baru, masuk ke akun Azure Anda, lalu pilih proyek dari **daftar drop-down Proyek.**
5. Pilih tombol **Buat dan Dorong** untuk membuat repositori dan menambahkan aplikasi anda.
6. Bagaimana cara upload file ke repository?

<https://docs.github.com/en/repositories/working-with-files/managing-files/adding-a-file-to-a-repository>

1. Menambahkan File Ke Repositori Di GitHub
2. Di GitHub.com, navigasikan ke halaman utama repositori.
3. Di atas daftar file, pilih menu tarik-turun Tambahkan file dan klik Unggah file . Alternatifnya, Anda dapat menarik dan melepas file ke browser Anda.
4. Untuk memilih file yang ingin Anda upload, drag dan drop file atau folder tersebut, atau klik pilih file Anda .
5. Di bidang "Pesan komit", ketikkan pesan komit singkat dan bermakna yang menjelaskan perubahan yang Anda buat pada file. Anda dapat mengatribusikan komit ke lebih dari satu penulis dalam pesan komit. Untuk informasi lebih lanjut, lihat " [Membuat komit dengan banyak penulis](https://docs.github.com/en/pull-requests/committing-changes-to-your-project/creating-and-editing-commits/creating-a-commit-with-multiple-authors) ."
6. Di bawah bidang pesan penerapan, putuskan apakah akan menambahkan penerapan Anda ke cabang saat ini atau ke cabang baru. Jika cabang Anda saat ini adalah cabang default, Anda harus memilih untuk membuat cabang baru untuk komit Anda dan kemudian membuat permintaan tarik. Untuk informasi lebih lanjut, lihat " [Membuat permintaan tarik](https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/creating-a-pull-request) ."
7. Klik Usulkan perubahan .

Menambahkan File Ke Repositori Menggunakan Baris Perintah

1. Di komputer Anda, pindahkan file yang ingin Anda unggah ke GitHub ke direktori lokal yang dibuat saat Anda mengkloning repositori.
2. Buka Git Bash .
3. Ubah direktori kerja saat ini ke repositori lokal Anda.
4. Tahapkan file untuk dikomit ke repositori lokal Anda.
5. $ git add .
6. # Adds the file to your local repository and stages it for commit. To unstage a file, use 'git reset HEAD YOUR-FILE'.
7. Komit file yang telah Anda tempatkan di repositori lokal Anda.
8. $ git commit -m "Add existing file"
9. # Commits the tracked changes and prepares them to be pushed to a remote repository. To remove this commit and modify the file, use 'git reset --soft HEAD~1' and commit and add the file again.
10. [Dorong perubahan](https://docs.github.com/en/get-started/using-git/pushing-commits-to-a-remote-repository) di repositori lokal Anda ke GitHub.com.
11. $ git push origin YOUR\_BRANCH
12. # Pushes the changes in your local repository up to the remote repository you specified as the origin

DAFTAR PUSTAKA

* <https://about.gitlab.com/topics/version-control/benefits-distributed-version-control-system/>
* <https://www.petanikode.com/git-untuk-pemula/>
* <https://learn.microsoft.com/id-id/visualstudio/version-control/git-create-github-account?view=vs-2022>
* <https://learn.microsoft.com/id-id/visualstudio/version-control/git-create-repository?view=vs-2022>
* <https://docs.github.com/en/repositories/working-with-files/managing-files/adding-a-file-to-a-repository>