

Apa itu Git?

Git adalah sistem kontrol versi yang sangat populer yang digunakan oleh pengembang perangkat lunak untuk melacak perubahan dalam kode mereka. Berikut adalah beberapa konsep dasar tentang Git:

1. **Version Control System (VCS):** Git adalah jenis sistem kontrol versi yang mengelola perubahan dalam kode sumber selama pengembangan perangkat lunak. Ini memungkinkan pengembang untuk melacak setiap perubahan yang dilakukan pada kode.
2. **Distributed Version Control System (DVCS):** Git merupakan DVCS yang berarti setiap klon dari repositori Git adalah repositori lengkap dengan semua riwayat perubahan. Ini memungkinkan kolaborasi yang lebih mudah di antara tim pengembang tanpa memerlukan akses terus menerus ke server sentral.
3. **Commit:** Tindakan menyimpan perubahan dalam repositori Git disebut commit. Setiap commit memiliki pesan yang menjelaskan perubahan yang dilakukan.
4. **Branch:** Git memungkinkan pengembang untuk bekerja di cabang (branch) yang terisolasi dari kode utama (biasanya cabang utama disebut `master` atau `main`). Ini memungkinkan pengembangan fitur baru tanpa mempengaruhi kode yang sudah stabil.
5. **Merge:** Proses menggabungkan perubahan dari satu cabang ke cabang lainnya disebut merge. Ini umumnya digunakan untuk menggabungkan fitur yang sudah selesai dikembangkan ke cabang utama.
6. **Pull Request (PR):** Ketika seorang pengembang telah selesai dengan perubahan di cabangnya, mereka dapat membuat pull request untuk menggabungkan perubahan tersebut ke cabang utama. Ini adalah proses umum dalam kolaborasi tim.
7. **Remote Repository:** Repositori Git yang berada di server eksternal (seperti GitHub, GitLab, atau Bitbucket) disebut remote repository. Ini digunakan untuk berbagi kode dengan tim atau publik.
8. **Workflow:** Git mendukung berbagai workflow pengembangan, termasuk Gitflow, GitHub Flow, dan sebagainya. Setiap workflow memiliki aturan dan praktik terbaik sendiri untuk penggunaan Git dalam pengembangan perangkat lunak.

Git sangat penting dalam pengembangan perangkat lunak modern karena memungkinkan kolaborasi yang efisien, manajemen versi yang baik, dan pengelolaan kode yang terorganisir.

Revision #1

Created 13 December 2024 14:58:45 by Admin

Updated 13 December 2024 15:08:57 by Admin