

## **2. GIT – BRANCH**

Branching merupakan fitur penting dalam GIT yang memungkinkan pengembang untuk bekerja pada versi terisolasi dari repositori. Ini memungkinkan pengembang untuk menciptakan jalur pengembangan yang berbeda, menjaga perubahan eksperimental terpisah dari kode produksi, dan memfasilitasi kolaborasi tim dengan aman.

Berikut adalah beberapa poin penting dalam deskripsi tentang GIT Branch:

1. Cabang sebagai Jalur Pengembangan: Dalam GIT, cabang adalah versi terisolasi dari repositori. Mereka memungkinkan pengembang untuk memisahkan pengembangan berbagai fitur atau perbaikan bug tanpa mempengaruhi kode di cabang utama.
2. Membuat Cabang: Pengembang dapat membuat cabang baru dengan menggunakan perintah `git branch [nama-cabang]`. Ini akan membuat salinan dari repositori saat ini dan menciptakan cabang baru dengan nama yang diberikan.
3. Beralih Antara Cabang: Pengembang dapat beralih antara cabang dengan perintah `git checkout [nama-cabang]`. Ini memungkinkan mereka untuk bekerja pada kode di cabang yang berbeda.
4. Merge Cabang: Setelah selesai dengan pengembangan di sebuah cabang, perubahan dapat digabungkan kembali ke cabang utama atau cabang lainnya menggunakan perintah `git merge [nama-cabang]`. Proses merge akan menggabungkan perubahan dari cabang yang ditentukan ke cabang yang sedang aktif.
5. Menghapus Cabang: Setelah pengembangan di sebuah cabang selesai dan perubahan telah digabungkan ke cabang utama, cabang tersebut dapat dihapus menggunakan perintah `git branch -d [nama-cabang]`.

6. Cabang Jarak Jauh (Remote Branch): Selain cabang lokal, GIT juga mendukung cabang jarak jauh yang berada di repositori jarak jauh seperti GitHub atau GitLab. Pengembang dapat mengambil cabang tersebut ke repositori lokal dengan perintah `git fetch` atau `git pull`.
7. Manajemen Cabang: GIT menyediakan sejumlah perintah untuk manajemen cabang seperti melihat daftar cabang (`git branch`), melihat perbedaan antara cabang (`git diff [cabang-1] [cabang-2]`), dan memindahkan atau mengubah nama cabang (`git branch -m [nama-cabang-lama] [nama-cabang-baru]`).

Cabang GIT memainkan peran penting dalam pengembangan perangkat lunak kolaboratif, memberikan fleksibilitas dalam mengelola kode sumber dan memungkinkan pengembang untuk bekerja secara efisien dalam tim. Dengan menggunakan cabang secara bijak, pengembang dapat mengatur dan melacak evolusi proyek dengan lebih baik.