

D-02 Training

Version Control

Git



1. Git

Version Control



Memahami Version Control Git

Kontrol versi adalah metode yang digunakan untuk **melacak dan mengelola perubahan** dalam kode sumber atau berkas proyek.

Git merupakan salah satu sistem kontrol versi terdistribusi yang paling populer dan kuat. Berikut adalah langkah-langkah untuk memahami kontrol versi dan Git

● Sistem Kontrol Versi Terpusat (Centralized Version Control System)

Dalam sistem kontrol versi terpusat, ada satu repositori sentral yang berfungsi sebagai "**master**" untuk menyimpan seluruh sejarah proyek. Setiap pengembang melakukan perubahan pada salinan lokal, kemudian mengirimkan perubahan tersebut ke repositori sentral. Contoh sistem kontrol versi terpusat adalah **Subversion** (SVN).

● Sistem Kontrol Versi Terdistribusi (Distributed Version Control System)

Dalam sistem kontrol versi terdistribusi, setiap anggota tim memiliki salinan lengkap dari seluruh repositori. Ini berarti setiap pengembang memiliki salinan lengkap sejarah perubahan, tidak hanya salinan terbaru. Contoh sistem kontrol versi terdistribusi adalah **Git**, **Mercurial**, dan **Bazaar**.

Dasar Dasar Command GIT

01 git init

Menginisialisasi direktori sebagai repositori Git kosong

```
git init
```

02 git clone

Menduplikasi repositori Git yang sudah ada ke direktori lokal.

```
git clone https://github.com/username/repo.git
```

03 git status

Menampilkan status perubahan yang belum dikomit di repositori lokal.

```
git status
```

04 git add

Menambahkan perubahan ke area persiapan (staging area) untuk disiapkan menjadi commit.

```
git add file1.txt  
git add .
```

Continue...

05 **git commit**

Membuat commit dari perubahan yang sudah di-staging dan menambahkan pesan commit.

```
git commit -m "Menambahkan fitur login"
```

06 **git push**

Mengirimkan commit ke repositori jarak jauh (remote repository).

```
git push origin main
```

07 **git pull**

Mengambil commit terbaru dari repositori jarak jauh dan menggabungkannya ke repositori lokal.

```
git pull origin main
```

08 **git branch**

Menampilkan daftar cabang (branch) yang ada di repositori dan menunjukkan cabang aktif.

```
git branch
```


Continue...

09 git checkout

Beralih ke cabang lain atau ke commit tertentu.

```
git checkout feature-branch  
git checkout abc1234 (commit hash)
```

10 git merge

Menggabungkan perubahan dari satu cabang ke cabang aktif.

```
git merge feature-branch
```

11 git log

Menampilkan daftar commit beserta riwayatnya dalam repositori.

```
git log
```

12 git remote

Menampilkan daftar repositori jarak jauh yang terhubung dengan repositori lokal.

```
git remote -v
```

13 git fetch

Mengambil informasi terbaru dari repositori jarak jauh tanpa menggabungkan perubahan.

```
git fetch origin
```

Continue...

14 **git diff**

Menampilkan perbedaan antara versi yang sudah di-staging dengan versi sebelumnya.

```
git diff
```

15 **git reset**

Mengembalikan file yang sudah di-staging ke direktori kerja sebelumnya.

```
git reset file.txt
```

Simulasi Kolaborasi Menggunakan GIT

Simulasi kolaborasi Git dengan **anggota kelompok** menggunakan GitHub melibatkan **repository, pembuatan branch, membuat perubahan, mengelola konflik**, dan menyelesaikannya dengan **permintaan tarik (pull request)** dan **penggabungan (merge)**.

Simulasi Kolaborasi Menggunakan GIT

Silahkan clone repository berikut:
`repo_url`

Buat branch sesuai dengan nama kelompok dengan format eg. **kelompok_1**

Thank You

Version Control

Git

