



CLI, Git, dan GitHub



Command Line Interface (CLI)

adalah mekanisme perangkat lunak yang Anda gunakan untuk berinteraksi dengan sistem operasi menggunakan keyboard. Mekanisme lainnya adalah UI grafis (GUI), yang saat ini populer di semua aplikasi dan sistem perangkat lunak.

Kita dapat menggunakan GUI guna menavigasi secara visual serta mengeklik ikon dan gambar untuk menjalankannya. Namun, GUI tidak efisien untuk tugas-tugas administrasi sistem, terutama jika lingkungannya virtual atau jarak jauh. Dengan CLI, Anda dapat memasukkan perintah teks untuk mengonfigurasi, menavigasi, atau menjalankan program di server atau sistem komputer apa pun. Semua sistem operasi, termasuk Linux, macOS, dan Windows menyediakan CLI untuk interaksi sistem yang lebih cepat.



Manfaat Command Line Interface

- Efisiensi (Daripada menghabiskan waktu untuk mencari dan mengeklik setiap file, Anda dapat menggunakan CLI untuk melakukan tindakan pada beberapa file dalam satu perintah teks.)
- Remote Akses (CLI lebih sedikit menggunakan resource dibandingkan GUI yang menjadikannya lebih cocok untuk remote akses seperti server, mini pc (raspberry pi, NVIDIA Jetson, dll), dan pc lainnya.)
- Belajar Dasar Sistem Operasi (Kita dapat belajar OS lebih dalam seperti konfigurasi, manajemen file, hierarki struktur sistem, akses fitur yang tidak ada di gui, dll)
- Lebih powerful untuk developer (banyak tool developer seperti git, npm, pip, dan command untuk mengatur server yang hanya bisa dijalankan lewat CLI)



Cara Kerja CLI

Antarmuka baris perintah (CLI) adalah antarmuka berbasis teks di mana Anda dapat memasukkan perintah yang berinteraksi dengan sistem operasi komputer. CLI beroperasi dengan bantuan shell default, yang berada di antara sistem operasi dan pengguna.

Shell mengacu pada program yang merupakan perantara antara pengguna dan sistem operasi. Contohnya termasuk Bash, cmd.exe, PowerShell, dan lainnya untuk fungsi yang lebih spesifik.



Command

Command adalah instruksi yang kamu berikan ke sistem operasi melalui shell (terminal), untuk melakukan suatu tugas tertentu.

misal :

```
ls -l home|
```

ls → nama command

-l → opsi/flag

/home → argumen



Macam-macam command

command berasal dari berbagai macam sumber, diantaranya :

- Built-in shell command (cd, echo, dll)
- GNU core utilities (ls, cp, mv, dll)
- Tools tambahan (misal curl, git, ping, nano, dll)
- Package yang kamu install sendiri



Apa saja command yang umum digunakan?

1. Navigasi File System

Command	Fungsi
ls(bash), dir(cmd)	Melihat isi direktori
cd	Pindah direktori
pwd(bash), cd(cmd)	Tampilkan direktori saat ini
mkdir	Buat direktori
rmdir	Hapus direktori kosong

2. Manajemen File

Command	Fungsi
touch(bash), echo.>(cmd)	Buat file kosong
cp(bash), copy(cmd)	Salin file atau folder
mv(bash), move(cmd)	Pindah/ubah nama file
rm(bash), del(cmd)	Hapus file atau folder
cat(bash), type(cmd)	Tampilkan isi file




3. Edit File (Teks Editor di CLI)

Command	Fungsi
nano	Editor teks sederhana
vim / vi	Editor teks powerful
gedit	GUI editor (Linux desktop)
code	vscode

4. Manajemen User dan Permission

Command	Fungsi
whoami	Lihat siapa user sekarang
id	Info user dan group
adduser	Tambah user baru
passwd	Ganti password
chmod	Ubah permission file
groups	Lihat grup user
su, sudo	Akses root/admin sementara



5. Proses dan Sistem

Command	Fungsi
ps	Lihat proses aktif
top / htop	Lihat aktivitas sistem
kill	Hentikan proses
sleep	Tunda proses
time	Hitung waktu eksekusi command
uptime	Lihat durasi sistem menyala

6. Jaringan

Command	Fungsi
ping	Cek koneksi ke alamat IP/host
ifconfig / ip a	Lihat info jaringan
curl	Akses URL via CLI
wget	Download file dari web
netstat / ss	Lihat koneksi dan port



7. Package Management (ubuntu)

Command	Fungsi
<code>apt update</code>	Update info repo
<code>apt upgrade</code>	Upgrade semua paket
<code>apt install</code>	Install software
<code>apt remove</code>	Hapus software
<code>dpkg -i</code>	Install file .deb manual

8. Utilities Lain

Command	Fungsi
<code>history</code>	Lihat riwayat perintah
<code>alias</code>	Buat nama pendek custom
<code>echo</code>	Cetak teks ke output
<code>date</code>	Lihat waktu sekarang
<code>clear</code>	Bersihkan layar terminal



Operator CLI

Operator	Fungsi	Contoh Penggunaan
>	Output redirection: Menulis output ke file	<code>echo "Hello" > output.txt</code>
>>	Append output redirection: Menambahkan output ke file	<code>echo "Hello again" >> output.txt</code>
<	Input redirection: Mengambil input dari file	<code>sort < daftar.txt</code>
	Pipe: Mengalirkan output dari perintah ke perintah lain	<code>ls grep "data"</code>
&&	Menjalankan perintah kedua hanya jika perintah pertama berhasil	<code>command1 && command2</code>
*	Wildcard: Mencocokkan berbagai karakter dalam nama file	<code>ls *.txt</code> (menampilkan semua file dengan ekstensi .txt)
?	Wildcard: Mencocokkan satu karakter dalam nama file	<code>ls file?.txt</code> (mencocokkan file1.txt, fileA.txt, dsb.)



Pengenalan Git

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

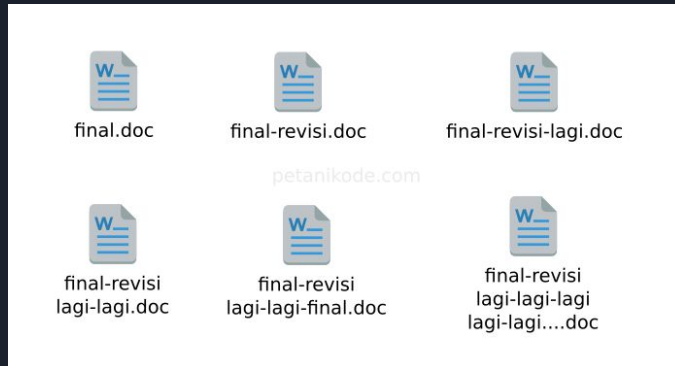
Pengontrol versi bertugas mencatat setiap perubahan pada file proyek yang dikerjakan oleh banyak orang maupun sendiri.

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

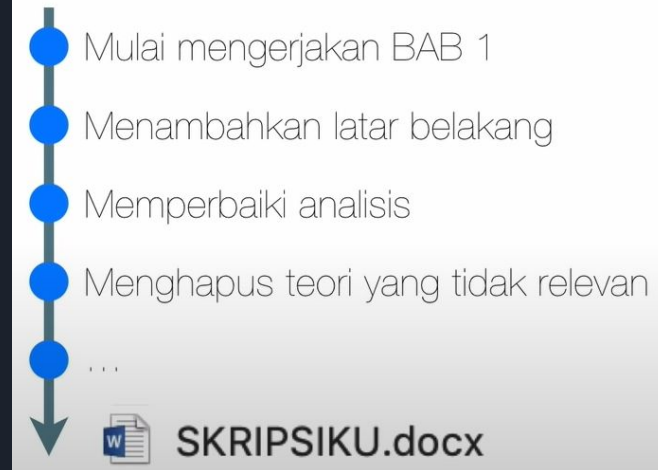
Apa yang dilakukan git?

Git sebenarnya akan memantau semua perubahan yang terjadi pada file proyek. Lalu menyimpannya ke dalam database.

sebelum menggunakan git :



Setelah menggunakan git :





Cara Kerja Git

Git punya 3 area utama:

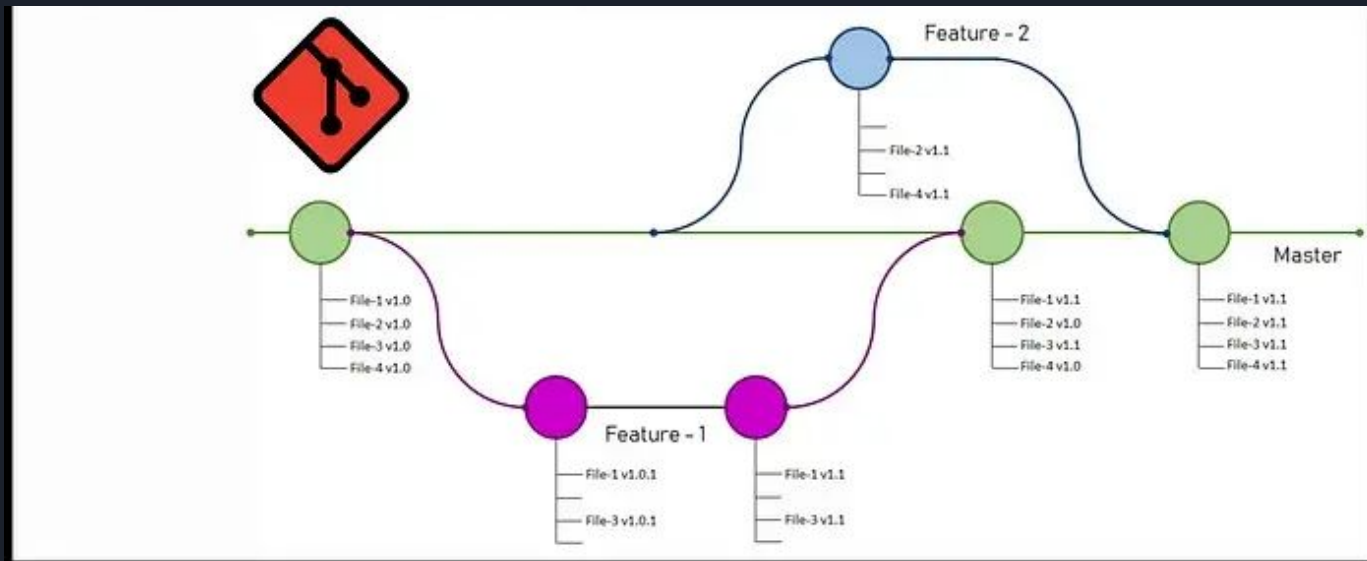
1. **Working Directory**
→ Tempat kamu ngoding atau ngedit file
2. **Staging Area**
→ Tempat buat "nyiapin" file yang mau kamu commit
3. **Repository (Local Repo)**
→ Tempat simpan riwayat commit kamu

Alur umumnya:

1. **git init**
Buat repo Git baru di folder proyek kamu
2. **Edit file**
Misal kamu bikin `main.py`
3. **git add main.py**
Masukkan file ke staging area
4. **git commit -m "nambah file main.py"**
Simpan perubahan ke riwayat Git

git branch dan merge

kita bisa membuat cabang dari project yang ingin dikerjakan dengan (git branch <nama branch>) dan kemudian menyatukan cabangnya kembali dengan git merge.





GitHub

GitHub merupakan manajemen project, sistem versioning code, sekaligus platform jaringan sosial bagi para developer seluruh dunia. Banyak sekali fungsi yang bisa digunakan oleh para developer, sehingga melalui platform ini memudahkan developer dalam mengembangkan sebuah karya.

GitHub juga memberikan layanan cloud untuk menyimpan dan mengelola project/repository git. Karena bersifat online, kita meng-edit sebuah repository/project secara bersamaan dengan orang lain di tempat yang berbeda. Oleh karena itu, platform ini sangat membantu tim project dalam menyusun suatu folder yang berisikan files terkait pemrograman.



remote

git remote adalah command untuk mengkoneksikan repository git lokal kita ke repository github, yang kemudian dapat kita lakukan push (kirim update dari lokal ke github) ataupun pull (ambil update dari github ke lokal).

untuk menambahkan remote dapat dengan command : “git remote add <nama remote> <url repository>”



pull request

Pull Request (PR) adalah permintaan resmi untuk menggabungkan perubahan dari satu branch ke branch lain di repository GitHub (atau GitLab, Bitbucket, dll).

alur pull request :

- Clone repo ke lokal → `git clone`
- Buat branch baru → `git checkout -b fitur-xyz`
- Ngerjain perubahan → coding, add, commit
- Push branch ke GitHub → `git push origin fitur-xyz`
- Buka GitHub → ada tombol "Compare & Pull Request"