**Inisialisasi dan Kloning Repository**

1. **git init**  
   Digunakan untuk menginisialisasi repository Git baru di folder yang ada. Artinya, kamu mulai memanfaatkan Git untuk menyimpan dan mengelola versi kode di folder tersebut.
2. **git clone <url>**  
   Digunakan untuk mengunduh (kloning) repository yang ada dari GitHub (atau platform lainnya) ke komputer lokal. Misalnya, jika kamu ingin bekerja dengan project orang lain yang ada di GitHub, kamu akan menggunakan perintah ini.

**Melakukan Perubahan**

1. **git add <file>**  
   Digunakan untuk menambahkan file tertentu ke dalam staging area, yaitu tempat di mana Git menyimpan perubahan sebelum commit.
2. **git add .**  
   Ini berfungsi sama seperti git add <file>, tetapi untuk menambahkan semua file yang telah berubah (baik yang baru dibuat maupun yang sudah dimodifikasi) ke dalam staging area.
3. **git commit -m "<message>"**  
   Setelah file ditambahkan ke staging area, perintah ini digunakan untuk menyimpan perubahan ke dalam repository, dengan memberi pesan untuk menjelaskan perubahan yang dilakukan.
4. **git reset <file>**  
   Menghapus file tertentu dari staging area tanpa mengubah perubahan yang ada pada file tersebut. Artinya, file tersebut tetap dimodifikasi, tetapi tidak akan masuk ke commit berikutnya.
5. **git reset --hard**  
   Mengembalikan seluruh file ke kondisi semula, yaitu menghapus semua perubahan yang belum di-commit. Semua perubahan yang belum disimpan akan hilang.

**Memantau Status dan Perubahan**

1. **git status**  
   Menampilkan status terbaru dari repository, seperti file mana yang sudah diubah, mana yang siap di-commit, dan mana yang belum ditambahkan ke staging area.
2. **git diff**  
   Menampilkan perbedaan antara file yang sudah diubah dengan versi terakhir yang di-commit. Ini membantu kamu melihat apa saja yang telah berubah.
3. **git show**  
   Menampilkan detail dari commit tertentu, termasuk perubahan apa saja yang dilakukan dalam commit tersebut.
4. **git log**  
   Menampilkan daftar commit yang ada dalam repository beserta informasi terkait, seperti siapa yang melakukan commit, pesan commit, dan kapan commit tersebut dilakukan.

**Menyinkronkan Repository**

1. **git push**  
   Mengirim perubahan (commit) yang ada di lokal ke repository remote (misalnya GitHub). Ini memungkinkan orang lain untuk melihat perubahan yang sudah kamu buat.
2. **git pull**  
   Mengambil (download) perubahan terbaru dari repository remote dan menggabungkannya dengan pekerjaan yang ada di lokal kamu.
3. **git fetch**  
   Mengambil perubahan terbaru dari remote tanpa menggabungkannya dengan kode lokal kamu. Perubahan hanya diunduh tanpa di-merge.
4. **git push**  
   Sama dengan penjelasan sebelumnya, perintah ini mengirim perubahan lokal ke repository remote.

**Bekerja dengan Branch**

1. **git branch**  
   Menampilkan daftar branch yang ada, termasuk branch yang sedang aktif.
2. **git checkout <branch>**  
   Berpindah ke branch lain. Misalnya, jika kamu ingin bekerja pada fitur tertentu, kamu akan checkout ke branch yang sesuai.
3. **git merge <branch>**  
   Menggabungkan perubahan dari branch lain ke branch yang sedang aktif. Misalnya, jika kamu sudah selesai dengan pekerjaan di branch "fitur" dan ingin menggabungkannya ke branch utama, kamu akan menggunakan perintah ini.
4. **git tag <tag\_name>**  
   Memberikan tanda atau label pada commit tertentu. Biasanya digunakan untuk menandai versi tertentu dari proyek, misalnya v1.0 atau release-2025.

Dengan memahami dan menggunakan perintah-perintah Git di atas, kamu bisa lebih mudah mengelola proyek kode sumber, baik itu sendirian ataupun dalam tim.