usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]

[--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]

[-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]

[--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]

[--super-prefix=<path>] [--config-env=<name>=<envvar>]

<command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

restore Restore working tree files

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

diff Show changes between commits, commit and working tree, etc

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

commit Record changes to the repository

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

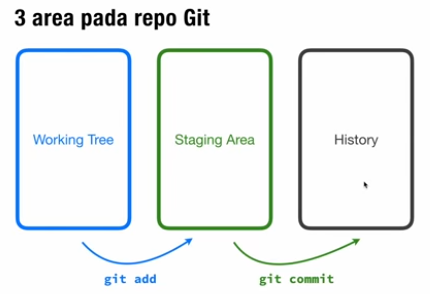
'git help -a' and 'git help -g' list available subcommands and some

concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

See 'git help git' for an overview of the system.

Git –version versi git



pwd : print working directory

ls : list directory

cd : change directory

clear : clear directory

git init : menjadikan folder sbg repo

$ git init

Initialized empty Git repository in D:/repo-git/.git/

User@DESKTOP-0GRTEHQ MINGW64 /d/repo-git (master)

Git status

git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

GIT & GITHUB.docx

git.docx

~$git.docx

nothing added to commit but untracked files present (use "git add" to track)

git add git.docx masukkan file ke staging area

git commit

Author identity unknown

\*\*\* Please tell me who you are.

Run

git config --global user.email "you@example.com"

git config --global user.name "Your Name"

to set your account's default identity.

Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'User@DESKTOP-0GRTEHQ.(none)')

perlu isi author

User@DESKTOP-0GRTEHQ MINGW64 /d/repo-git (master)

$ git config --global user.email "husniorbek@gmail.com"

User@DESKTOP-0GRTEHQ MINGW64 /d/repo-git (master)

$ git config --global user.name "husniokbah"

git commit -m "update manual"

[master (root-commit) 2c9c3fd] update manual

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 git.docx

git add . memindah semua untracked files ke staging area

git log

commit 2c9c3fd6aa69804c3da7ac6a05f3de58ca1969be (HEAD -> master)

Author: husniokbah <husniorbek@gmail.com>

Date: Thu Oct 26 20:09:55 2023 +0700

update manual

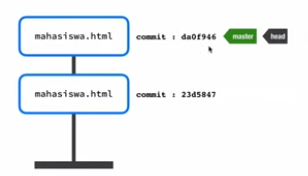
mengembalikan file yg telah dihapus

git checkout 5 digit hash –namafile

git checkout 2c9c3 –git.docx

git commit -am "menambah 1 list"

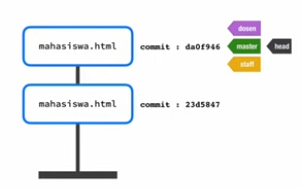
memasukkan ke staging area sekaligus commit



Buat branch

Git branch <nama\_branch>

Buat 2 branch git branch dosen & git branch staf



Git branch menampilkan daftar branch dmn warna hijau sbg current branch

git log --all --decorate --oneline --graph

\* 665ce89 (HEAD -> master, staf, dosen) menambah 1 list

\* 29180b9 menambahkan item list

\* 2c9c3fd update manual

HEAD itu pointer branch, ada 3 branch dan branch aktif adalah master

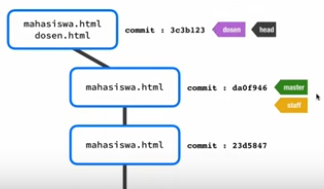
Telah melakukan 3 commit

Buat alias utk mengganti perintah

alias graph="git log --all --decorate --oneline --graph"

pindah ke branch lainnya

git checkout <nama\_branch>





Lalu melakukan merge

Untuk merge

Lihat posisi aktif branch

Git branch

git branch

dosen

master

\* staf

Ternyata di staf

Pindah dulu ke master

git checkout master

git merge dosen

menghapus branch yg udah dihapus

git branch –d dosen

git branch merged : ingin tahu branch mana yg udah di merge