

No.

Date .....

## Agenda

- 1 Pengenalan Version Control
- 2 Pengenalan Git
- 3 Repository
- 4 The three tree
- 5 Working Directory
- 6 ~~Staging~~ Index
- 7 Commit
- 8 Reset Commit

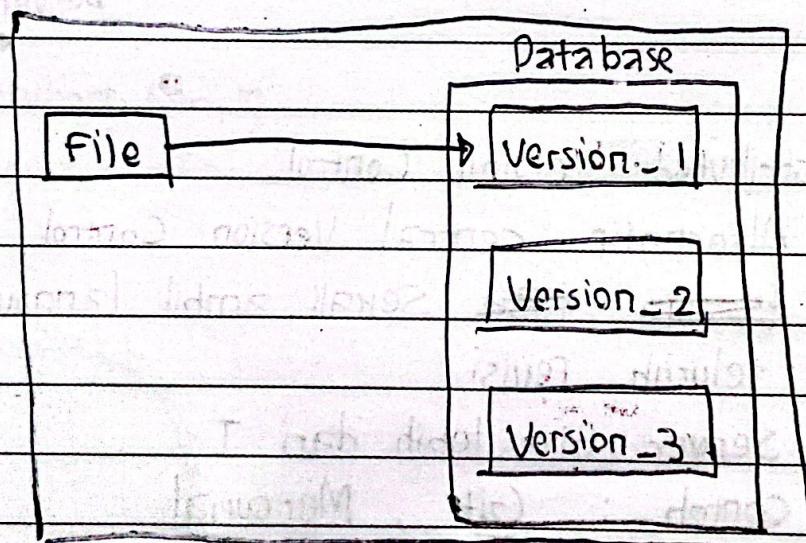
# # VERSION CONTROL

→ Controlling The record of Version, Kalo mau downgrade bisa dengan mudah roll back

→ Tipe Version Control :

## 1. Local Version Control

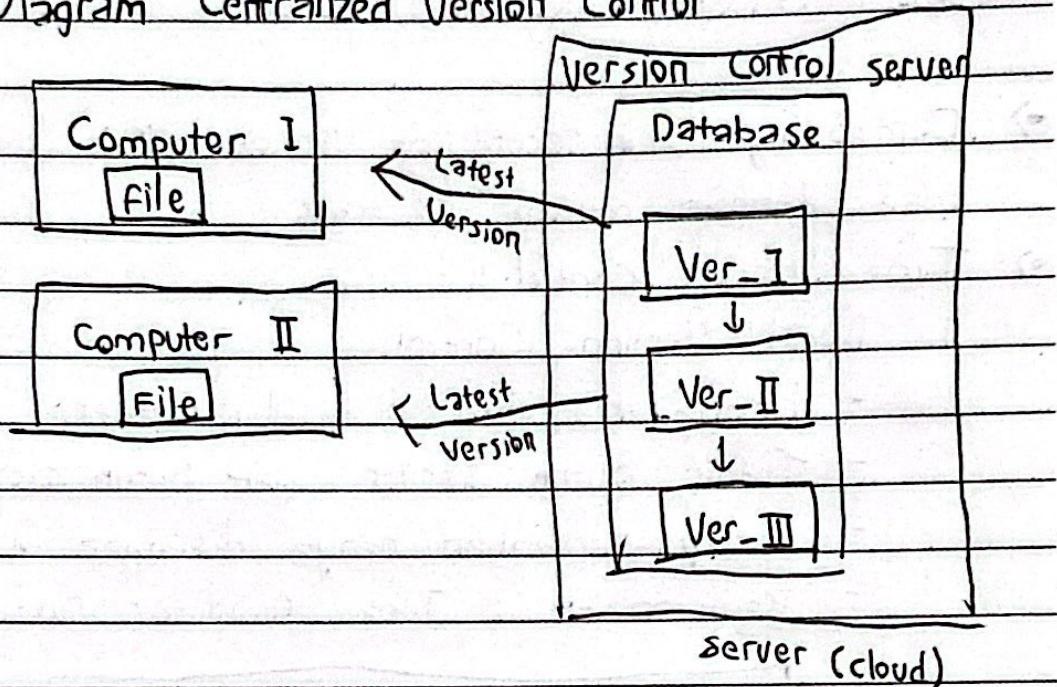
- hanya berjalan di local computer
- Tidak butuh server, gak butuh cloud
- Setiap perubahan hanya disimpan di local compi
- Kekurangan : Jika komputer rusak data hilang



## 2. Centralized Version Control

- Based on server
- Bisa collaborate → online
- Kekurangan → Cannot access when offline
- Contoh : Subversion
- hanya ambil latest Version

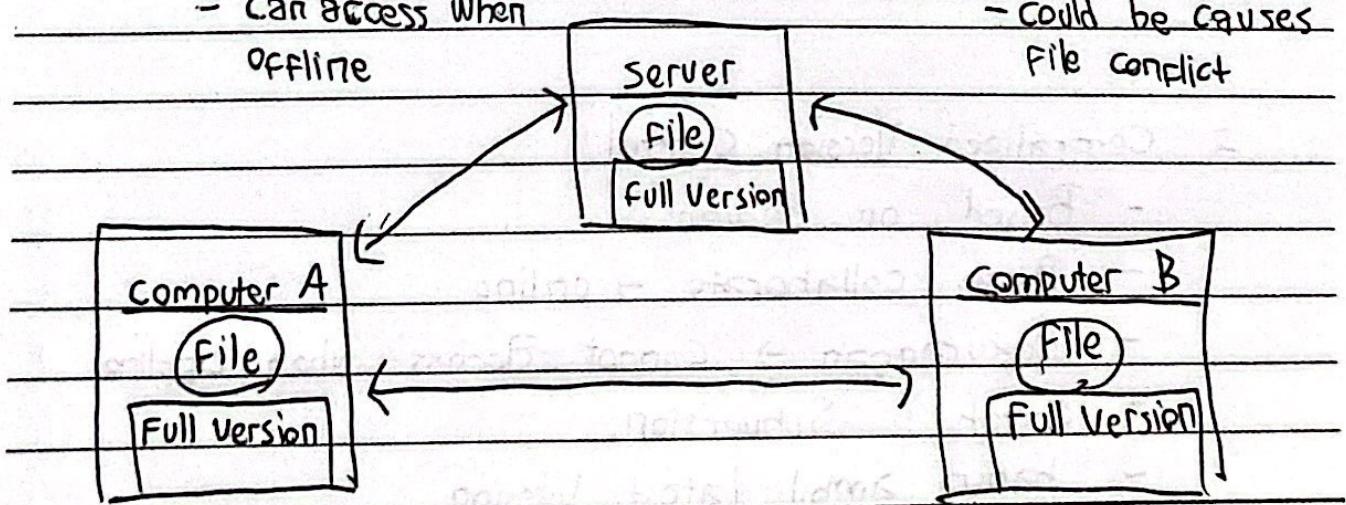
### - Diagram Centralized Version Control



### 3. Distributed Version Control

- Alternatif central Version Control
- ~~file~~ sekali ambil langsung copy seluruh revisi
- Server bisa lebih dari 1
- Contoh : Git, Mercurial
- Can access when offline

- Could be causes file conflict



# # Pengenalan Git

- Open Source Project → Linux Kernel
- Develop in 1991 - 2002 → Patch
- At 2002 → Bitkeeper
- At 2005 → Pertama Git muncul
- Sangat cepat, ringan, dapat men-manage project ukuran besar

## \* GIT

- Categorized as Distributed Version Control + local Version Control
- Gak butuh server → semua Version di-duplicate
- Setiap perubahan selalu dibuat Signatur → hashing hal sekecil apapun akan checksum → algoritma hashing
- Git → Automation recording project

## \* Installing Git → [git-scm.com/download/win](http://git-scm.com/download/win)

## \* Memastikan Git Berjalan

- Perlu membuat terminal / command line
- Check Version : git --version

## # Configuration

- User : abyannoor
- email : ariqabyan@gmail.com

### \* Integration git → VSCode

- git config --global core.editor "code --wait"
- diff tool

git config --global diff.tool "default-difftool"

git config --global difftool.default.cmd "code --wait --diff  
~~local~~ ~~remote~~ \\$LOCAL \\$REMOTE"

### → Keseluruhan Config

git config --list -- Show-Origin

No.

Date .....

## # Repository

- Repo = Project
- git init → membuat repo
- git status → melihat perubahan

## # Workflow

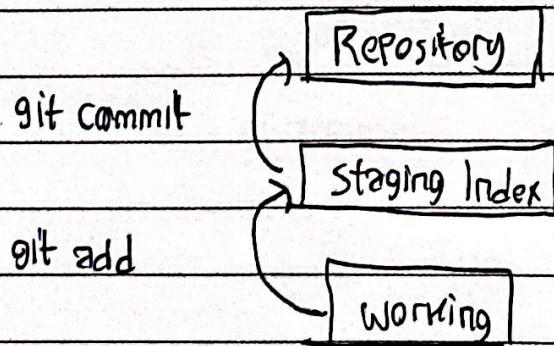
### \* The Three States

- dalam git terdapat 3 state:
  1. Modified → ada perubahan, tapi belum save permanen
  2. Staged → menandai modifikasi, dan akan save permanen
  3. Committed → data sudah aman disave di repo

### \* Three Section

1. Working directory → Modified
2. Staging Area → Staged
3. Repository → Committed

(↳ Final destination)



## # HASH

### \* Snapshot

- Versi 1 , Versi 2 → dinamakan Snapshot
- Snapshot berisi perubahan file yang sudah dicommit
- Setiap snapshot menghasilkan hash
- Hash → Checksum

### \* hash

- Primary key yang menghitung perubahan yg terjadi
- Hash bisa diubah
- menggunakan algoritma SHA - 1 untuk menghitung hash
- menjaga data integrity

### \* Head

- Pointer menuju Hash yang paling ~~akhir~~ akhir
- latest Hash

No.

Date .....

## # Adding file

- Cukup tambah file ke working directory,
- agar perubahan dpt ditrack - pindah ke staging index