Version control system:

* **Version control**, also known as source control, is the practice of tracking and managing changes to software code. Version control systems are software tools that help software teams manage changes to source code over time atau sebuah kumpulan perangkat lunak yang sudah terintegrasi dan digunakan untuk membantu software engineer mengelola perubahan dalam source code dari waktu ke waktu. -> padahal 1.
  + Skripsi1, skripsi\_final, skripsi\_finalbanget -> can see before version, rollback
  + **Vcs** **-> 3 versi**
  + **Local version**: tersimpan/berjalan dilokal kita, **masalahnya**: kalo komputer kita rusak jadi gabisa diakses, file <- version\_1 , version\_2, version\_3
  + **Centralize version control**: mengakses server (pakai jaringan), computer 1 (file 1 <- vcs (version\_1,2,3)), kekurangannya: gabisa lihat versi sebelumnya & gabisa melakukan perubahan
  + **Distributed version control**: mendistribusi data (dibagikan) file dan datanya di distribusikan juga beda sama centralize, computer a & b terpusat ke server (terintegrasi). branching -> mirroring

Git

* **Git** adalah singkatan dari Group Inclusive Tour, tools ini digunakan untuk mengelola versi source code program dengan cara menentukan baris serta kode yang perlu diganti atau ditambahkan. Lebih lanjut, Git adalah perangkat lunak buatan Linus Torvald yang saat ini digunakan untuk mengelola source code dan dia menggunakan hashing sha-1 (membaca kode biner kalo ada perubahan dia akan menginformasikan)
* **Three tree:** *modified* (memodifikasi -> update, add, delete), *staged* (proses tandai & bisa menentukan siapa yg akan ditandakan), *commited* (sudah dilakukan).
* **Three section:** working dir (lokasi perubahan data), staging area (sudah commit), repository (akan muncul).
* **Snapshot:** save game, kita sendiri yg menentukan siapa yg mau save (1 snapshot bisa banyak file bahkan 1) setiap kali snapshot dibuat pasti ada “**hash**”.
* **Hash:** integritas data (sesuai dengan apa yg benar) kalo di database namanya primary key.
* **Head:** isinya hash dari data yg terbaru (Cuma 1).
* **Repository:** nama dari sebuah project.

Init, add (file), masuk ke stagging area, commit

Git add . (menambahkan semuanya)

Cara hapus: git rm . atau (nama file)

Gajadi: git clean -f (ke commit sebelumnya)

Kalo mau adain lagi: git restore . atau (file)

Kalo mau modif lagi (diworking dir): git restore - -staged (nama file)

Kalo keynya gagal: eval `ssh-agent -s`,

ssh-add (pathnya/nama keynya)