GIT DASAR

Rahmat Sunjani

Daftar yang akan dibahas :

- Version Control
- Git
- Repository
- The Three Tree
- Working Directory
- Staging Index
- Commit
- Reset Commit
- Dan lain-lain

Source Code Yang Akan Dibahas

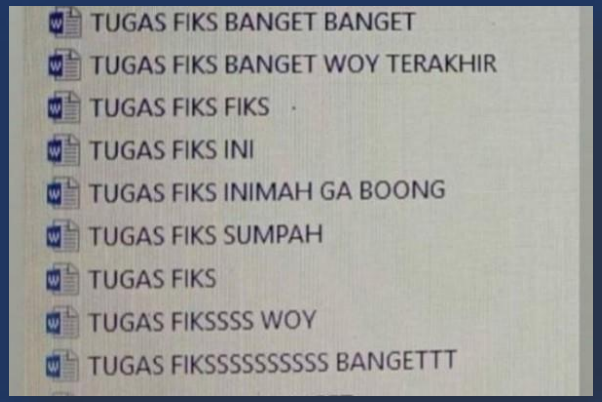
- git –version
- git config --global user.name “Rahmat Sunjani”
- git config --global user.email “email.example@gmail.com”
- git init
- git add <name-file>
- git commit -m “your-message”
- git diff
- git clean –f
- git restore <name-file>
- git restore --staged namafile
- git revert HEAD
- git log
- git log --oneline
- git log
- git log –oneline
- git show <hash>
- git reset --<mode> hash
- git commit --amend -m “Eh, ada baris lupa”
- git checkout hash -- <namafile>
- git checkout <hash>
- git checkout <branch>
- git branch --show-current
- git blame <namafile>
- git config --global alias.<name-alias> “<your-command>”

VERSION CONTROL

Rahmat Sunjani

Sebelum Ada Version Control System

Revisi



Banyak File

Version Control System

Version Control adalah sebuah system yang merekam perubahan pada file dari waktu ke waktu, sehingga kita bisa melihat versi sebelumnya jika diinginkan. Version Control dapat mendukung untuk berbagai jenis file.

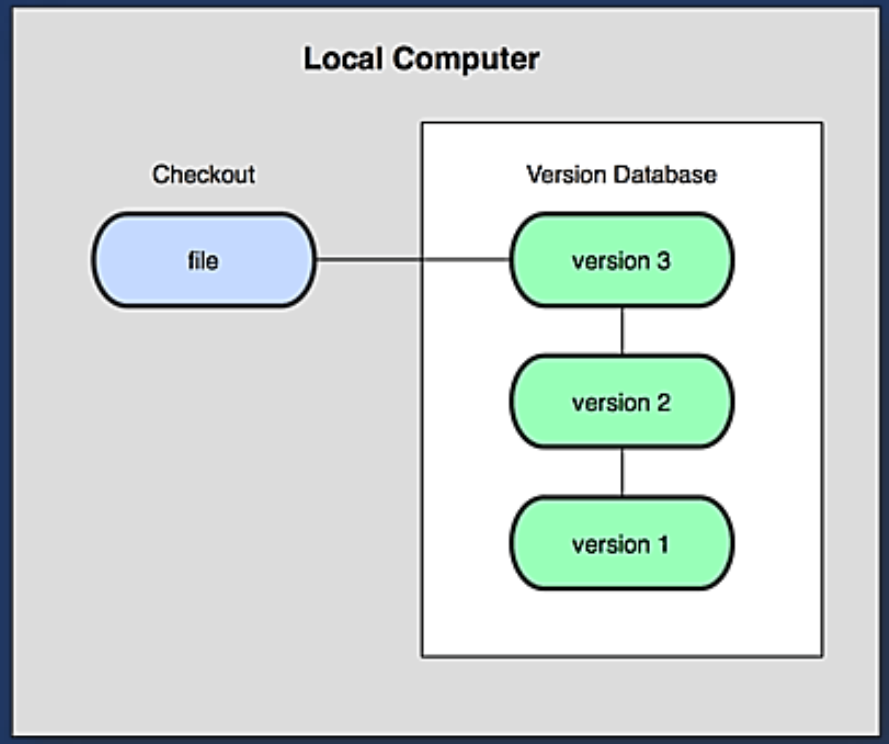
Tipe Version Control

Secara garis besar, version control dibagi menjadi 3 jenis :

- Local Version Control
- Centralized Version Control, dan
- Distributed Version Control

Local Version Control

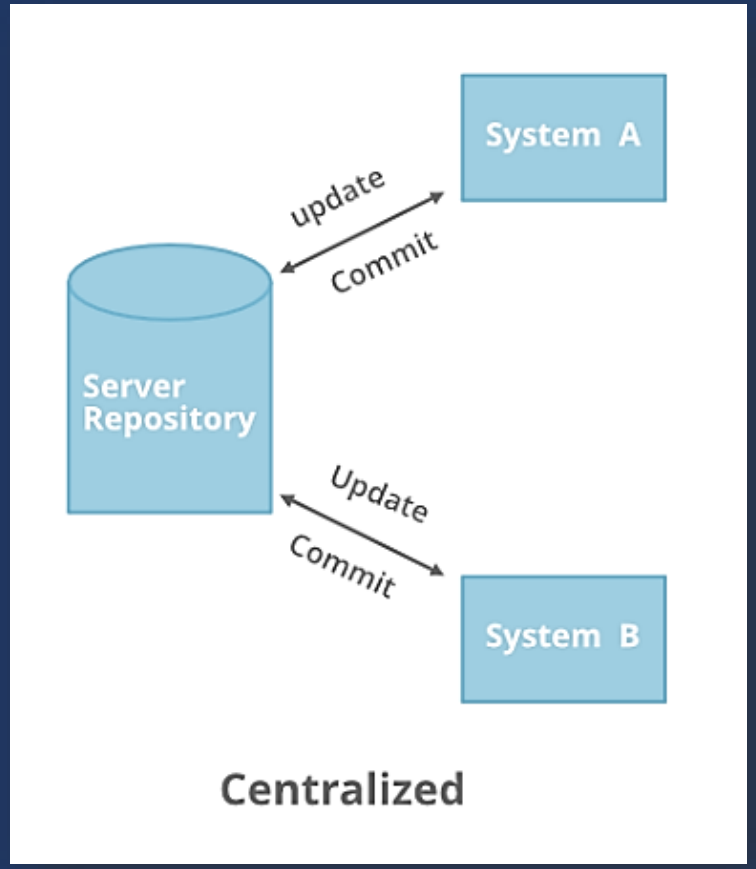
Local version control merupakan version control yang berjalan hanya di local computer.



Centralized Version Control

Centralized Version Control merupakan version control yang berjalan hanya di sebuah server.

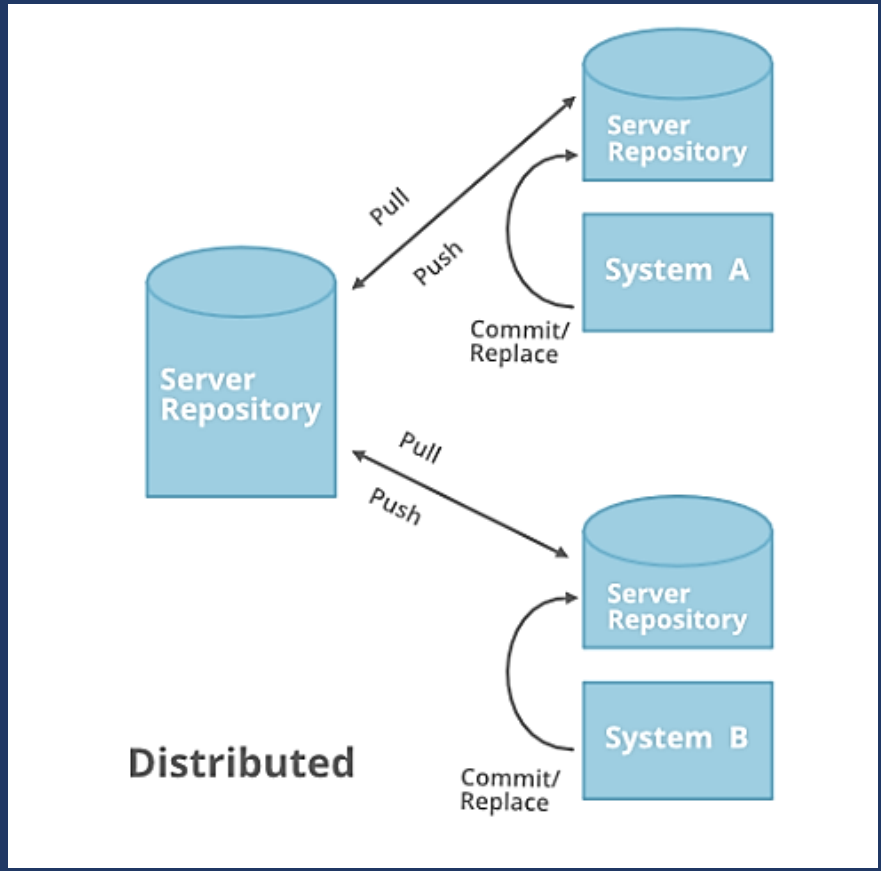
Contoh Centralized Version Control adalah Subversion.



Distributed Version Control

Distributed Version Control merupakan version control yang berjalan di sebuah server dan local host.

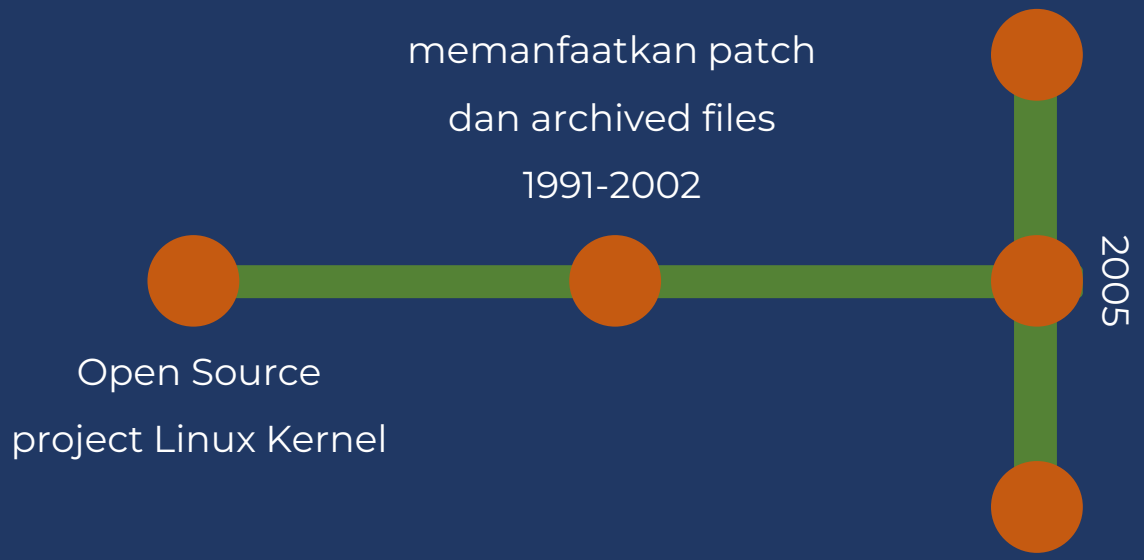
Contoh DVCs adalah Git, Mercurial dan lain-lain.



GIT

Rahmat Sunjani

Sejarah Git



BitKeeper dengan komunitas Linux Kernel kurang baik, sehingga pembuat Linux, Linus Torvalds mulai membuat DVCs sendiri

Mulai menggunakan DVCs bernama BitKeeper

semakin kesini Git semakin populer dan sekarang menjadi DVCs yang paling populer di dunia

Pengenalan Git

- **Git** adalah salah satu DVCs yang ada.
- **Git** tidak membutuhkan server untuk melakukan perubahan atau melihat riwayat revisi, hal ini dikarenakan dalam **Git**, semua riwayat project akan selalu di duplikasi, baik itu di server ataupun dilocal computer.
- Semua hal yang terjadi di git secara otomatis akan dicatat, hal ini menjadikan perubahan apapun di **Git**, pasti selalu bisa dikembalikan ke versi sebelumnya

Menginstall Git

- Git adalah aplikasi OpenSource dan Gratis,
- Git bisa untuk Windows, Mac dan Linux.
- Kita bisa download Git di : https://git-scm.com/downloads

Cek Versi Git

Masukan Perintah :

```
git --version
```

First Configuration

Masukan Perintah :

```
git config --global user.name “Rahmat Sunjani”
git config --global user.email “email.example@gmail.com”
```

First Configuration

Masukan Perintah :

```
git config --global user.name “Rahmat Sunjani”
git config --global user.email “email.example@gmail.com”
```

REPOSITORY

Rahmat Sunjani

Repository

Kosong

Repository

Setelah membuat folder maka kita akan membuat perintah untuk membuat Repository.

```
git init
```

dan akan ada file .git file tersebut adalah database dari git.

Cek Repository

Jika repository sudah ada maka kita akan coba cek status dari repository tersebut.

```
git status
```

The Three States

Git memiliki **tiga** state terhadap file kita

Modified

(menambah, mengedit, menghapus) file, namun belum disimpan secara permanen ke repository

Staged

kita menandai modifikasi yang kita lakukan terhadap file akan disimpan secara permanen ke repository.

Committed

artinya data sudah aman disimpan di repository

Diagram Three Tree

```
git add <name-file>
git commit –m “your-messege”
```

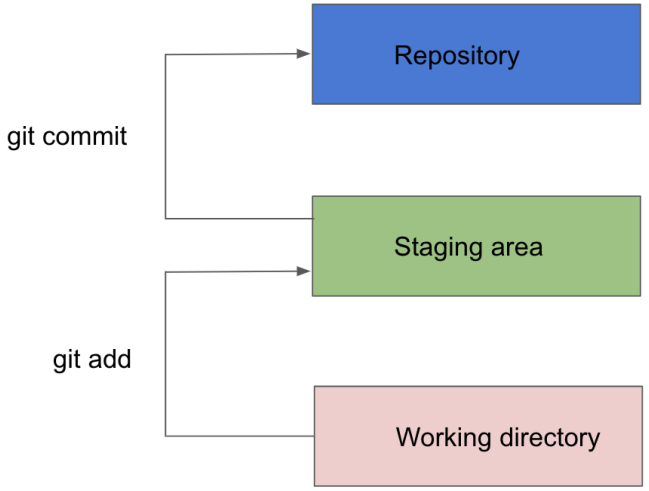


Fig: The three tree architecture of git (Basic Git Workflow)

Melihat Perubahan File

Jika kita ingin melihat perubahannya, kita juga dapat menggunakan Git untuk melihatnya.

```
git diff
```

Menghapus File

- Untuk menghapus file kita cukup hapus dari direktorinya.
- Sama seperti menambah dan menghapus, jika ingin simpan secara permanen di Repository, kita harus menambahkan operasi tersebut ke Staging Index, lalu commit ke Repository.

Membatalkan Penambahan File Di Working Direktori

```
git clean -f
```

Membatalkan Perubahan File Di Working Direktori

```
git restore <nama-file>
```

Membatalkan Perubahan dan Penghapusan File Di Working Direktori

```
git restore <nama-file>
```

Membatalkan dari Staging Index

Dalam kasus ini kita harus memindahkan dari Staging Index ke Working Directory.

```
git restore --staged namafile
```

Membatalkan Yang Sudah di Commit

- Tidak ada cara yang bisa kita lakukan jika perubahan sudah terlanjur di commit
- Yang bisa kita lakukan adalah dengan dua cara, **Revert Commit**

```
git revert HEAD
git revert <hash>
```

COMMIT LOG

Rahmat Sunjani

Commit Log

- Semua riwayat perubahan disimpan di komputer kita.
- Kekurangannya menjadi makin lama Repository akan semakin besar ukurannya, namun keuntungannya, kita bisa melihat semua riwayat commit, atau disebut Commit Log.

```
git log
git log --oneline
```

Graph

- Untuk melihat commit log dengan hubungannya dengan commit log sebelumnya atau log lainnya.

```
git log --oneline --graph
```

Melihat Detail Commit

- Untuk melihat detail yang sudah commit sesuai dari hash.

```
git show <hash>
```

RENAME FILE

Rahmat Sunjani

Rename File

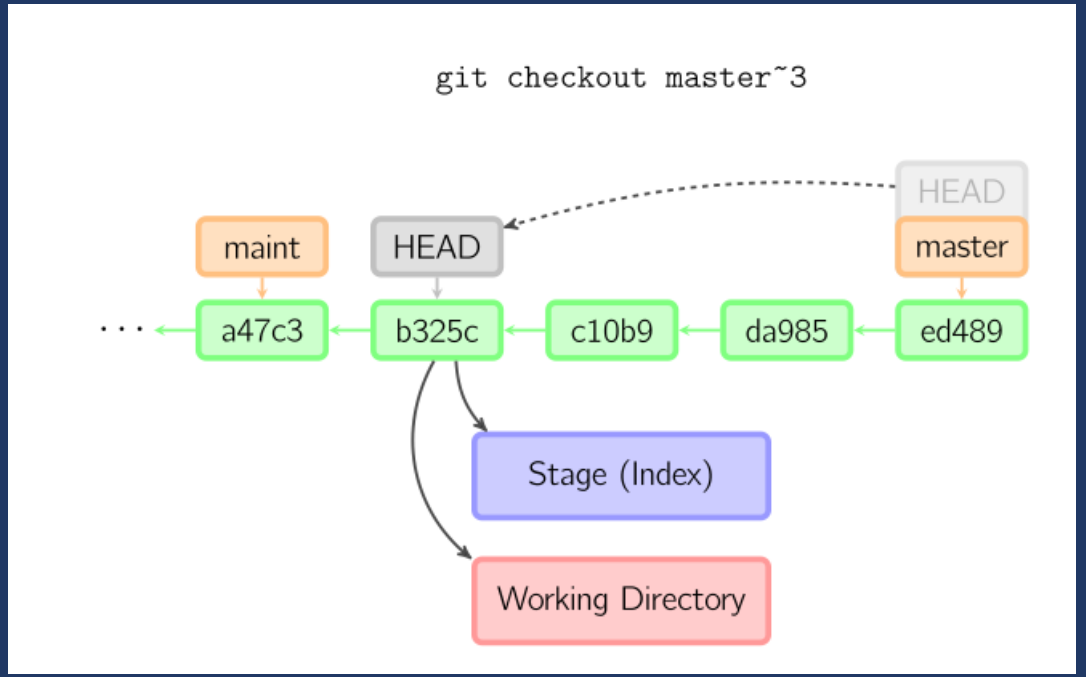
- Git bisa otomatis mendeteksi jika terjadi perubahan nama file, karena isi file sebagian besar masih sama.
- Langkahnya sama dari **Working Directory** => **Stagging Index** => Lalu **Commit**.

RESET COMMIT

Rahmat Sunjani

Reset Commit

Reset Commit merupakan mekanisme dimana kita menggeser HEAD pointer ke posisi commit yang kita mau, artinya commit selanjutnya akan dilakukan pada posisi HEAD baru.



Mode Git Reset

SOFT

Tidak mengubah Staging Index dan Working Directory

MIXED

Mengubah Staging Index menjadi sama seperti Repository, tidak mengubah Working Directory

HARD

Mengubah Staging Index dan Working Directory sehingga sama dengan Repository

```
git reset --<mode> hash
git reset --mixed hash
```

AMEND COMMIT

Rahmat Sunjani

Amend Commit

- Saat sudah melakukan commit, mungkin ada beberapa hal yang terlupakan.
- Biasanya kita akan lakukan reset soft ke commit sebelumnya, lalu tambahkan perubahan yang terlupakan, lalu kita lakukan commit ulang.
- Namun dengan **Amend Commit** kita bisa menggabungkan 2 commit.

```
git commit --amend -m “Eh, ada baris lupa”
```

PERGI VERSI SEBELUMNYA

Rahmat Sunjani

Versi Sebelumnya

- Git memiliki fitur dimana kita bisa melihat versi file pada commit sebelumnya
- Saat kita ambil versi file sebelumnya, file pada commit tersebut akan berada di Staging Index

```
git checkout hash -- <namafile>
```

SNAPSHOT SEBELUMNYA

Rahmat Sunjani

Snapshot Sebelumnya

- Git juga memiliki fitur seperti mesin waktu, dimana kita bisa kembali pada snapshot sebelumnya.
- Cara jika kita ingin menuju ke snapshot tertentu

```
git checkout <hash>
```

- Cara jika kita ingin kembali lagi di awal

```
git checkout <branch>
```

Cek Git Branch

- Materi ini akan dibahas dimateri selanjutnya

- Cara jika kita ingin menuju melihat branch saat ini

```
git branch --show-current
```

IGNORE

Rahmat Sunjani

Ignore

- Jika ada file yang tidak perlu Git eksekusi seperti file log, hasil kompilasi, kadang itu tidak butuh di track di Git.
- Git memiliki fitur ignore, dimana kita bisa meminta Git secara otomatis tidak men-track file di Git.
- Caranya kita bisa tambahkan file .gitignore di Repository
- Untuk template source codenya https://github.com/github/gitignore

BLAME

Rahmat Sunjani

Blame

- Git memiliki fitur yang bernama **blame**, ini digunakan untuk mencari tahu, siapa yang menambah perubahan pada file dan juga untuk mengetahui commit nya

```
git blame namafile
```

ALIAS

Rahmat Sunjani

Menambah Alias

- Git memiliki fitur yang dapat memberikan alias atau singkatan.

```
git config --global alias.<name-alias> “<your-command>”
git config --global alias.getBranch "branch --show-current"
```

MATERI SELANJUTNYA

Rahmat Sunjani

Materi Selanjutnya

- Git Branching
- Git Remote