

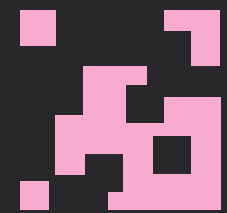
***PERTEMUAN 1***

***VERSION***

***CONTROL SYSTEM***

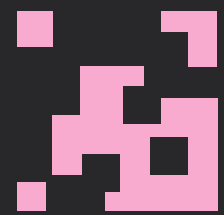
MULAI PRESENTASI





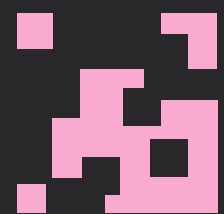
***PENGAJAR***

**ISMAIL ROSID**



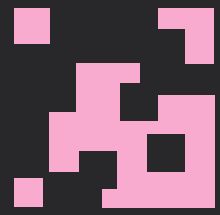
## APA ITU VERSION CONTROL SYSTEM?

**Version control system** adalah sebuah sistem yang memungkinkan penggunaanya untuk **mengelola** perubahan pada source code atau berkas proyek secara efisien. Tidak hanya itu, Version control system juga dapat **mencatat** setiap perubahan pada file yang dikerjakan baik secara individu maupun team.



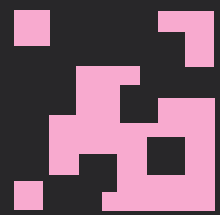
## PERUBAHAN APA SAJA YANG DI CATAT ATAU DIKELOLA?

Hampir semua detail perubahan dicatat, seperti : Id Perubahan, Identitas seseorang yang melakukan perubahan, waktu perubahan, judul perubahan, dan deskripsi perubahan



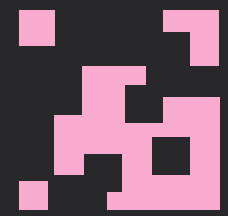
## MANFAAT PENGGUNAAN VERSION CONTROL SYSTEM

- Menandai setiap perubahan pada proyek
- Melihat perbandingan perubahan terbaru dengan perubahan-perubahan yang sebelumnya
- Memudahkan pekerjaan bersama dengan orang lain (collaborative)
- Mengembalikan perubahan terbaru ke perubahan yang sebelumnya sesuai dengan yang kita inginkan



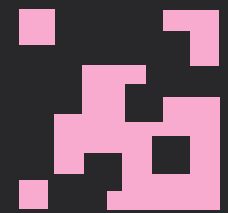
## JENIS-JENIS VERSION CONTROL SYSTEM

- Mercurial, **Git**, Subversion, CVS, dll.



## PERKENALAN GIT

Git adalah salah satu Version Control System yang populer dan open-source yang dikembangkan oleh Linus Torvalds pada tahun 2005. dengan git kita mampu mengelola sebuah proyek atau berkas secara terdistribusi.

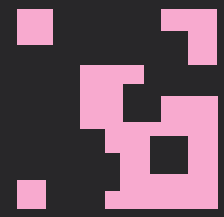


## INSTALASI GIT

Untuk memasang aplikasi git, ada beberapa persyaratan sistem yang harus dipenuhi, yaitu seperti berikut ini :

- Laptop atau Komputer dengan OS Windows 7, 8, 10
- Web Browser ( Firefox, Google Chrome, Brave, Edge )

Unduh Git dengan membuka tautan: <https://git-scm.com/downloads>



# KONFIGURASI GIT DASAR

Git memerlukan beberapa informasi dari kita sebagai author / pengarang dari perubahan yang nantinya kita buat. Informasi yang dibutuhkan adalah nama dan email.

## **Perintah Untuk Konfigurasi Git :**

Untuk melakukan penyetelan nama dan email yang akan digunakan, kita dapat menggunakan perintah git config

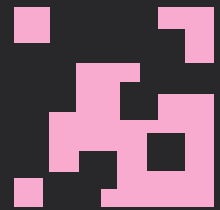
- Konfigurasi nama:
  - > *git config --global user.name "YOUR\_NAME"*
- Konfigurasi email:
  - > *git config --global user.email "YOUR\_EMAIL"*

Kemudian periksa konfigurasinya dengan perintah:

> *git config --list*

## **Konfigurasi Text Editor**

> *git config --global core.editor "code -w"*



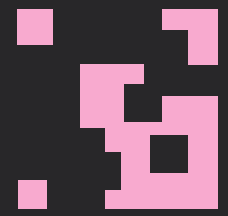
## REPOSITORY

Agar Git dapat melacak segala perubahan yang ada pada folder project, maka kita perlu mengubah folder tersebut menjadi Git repository dengan cara.

- > `git init`

Untuk mengecek kondisi saat ini pada repository, kita dapat menggunakan perintah

- > `git status`



# WORK SPACE

## ***Working Directory***

Semua file dan folder yang kita ubah akan masuk ke dalam working directory. Dalam working directory ini terdapat 2 jenis perubahan file yaitu:

### a. Untracked files

File yang perubahannya belum direkam oleh git. Biasanya untracked file ini merupakan file yang baru dibuat dan belum masuk ke dalam riwayat git.

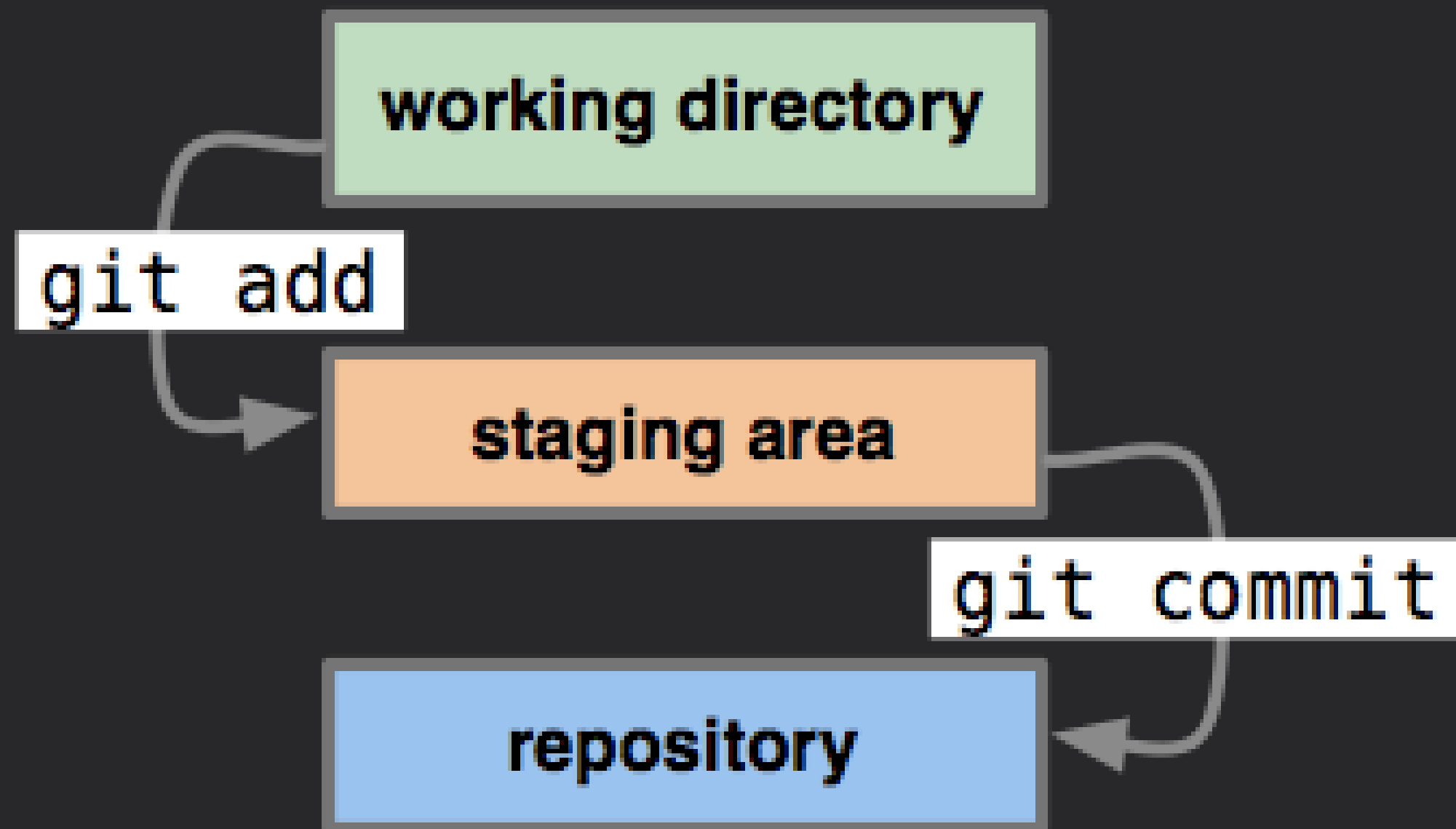
### b. Tracked files

File yang segala perubahannya di rekam oleh git. Terdapat 3 status dalam tracked files yaitu: staged, committed dan modified. Staged adalah perubahan yang siap untuk di-commit, committed adalah perubahan yang sudah disimpan secara permanen di repositori Git dan modified adalah status di mana file telah comited lalu diubah tetapi belum ditambahkan ke staged area untuk di commit agar menjadi comited kembali.



## --WORK SPACE

Berikut ini adalah gambar simulasi workspace. Silakan perhatikan dengan seksama untuk memahami konsepnya.



## --WORK SPACE

Untuk menyimpan perubahan yang telah teman-teman lakukan ke dalam staging area dan mengubah statusn filenya menjadi staged dapat menggunakan command:

```
> git add <nama_file>
```

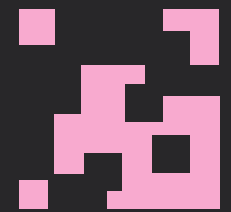
Atau juga bisa ditulis

```
> git add .
```

Digunakan untuk memindahkan semua perubahan file yang ada di dalam working directory ke staging index.

Kemudian dari staging area jika semuanya udah oke tidak ada masalah teman teman bisa langsung menyimpan perubahan yang telah dilakukan ke dalam repository git secara permanent dan merubah setatusn filenya menjadi committed dengan cara:

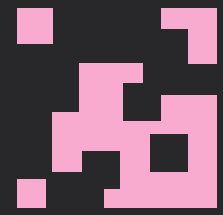
```
> git commit -m "YOUR_MESSAGE"
```



## LOGGING

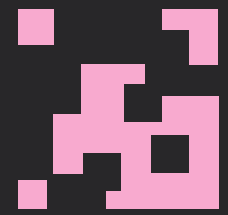
git log adalah perintah yang sudah disediakan oleh Git untuk melihat riwayat commit atau perubahan yang ada pada repository. Menggunakan command.

> git log



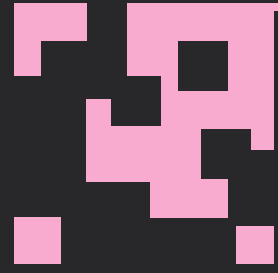
## CHECKOUT

Git checkout adalah perintah yang digunakan untuk berpindah ke branch atau commit tertentu dalam repository Git. Menggunakan command > **git checkout <id\_commit>**. Dengan perintah tersebut, Kita dapat langsung berpindah ke keadaan project pada branch atau commit tertentu untuk melihat perubahan apa saja yang saat itu terjadi terjadi.



## CATATAN PENTING

- **git config** untuk melakukan konfigurasi
- **git init** untuk mengubah folder kita menjadi repository
- **git add .** atau **git add <nama\_file>** , digunakan untuk memindahkan semua perubahan file yang ada di working directory ke staging area
- **git commit -m "YOUR\_MESSAGE"** , untuk melakukan commit
- **git log** , untuk melihat history commit kita
- **git checkout <id\_commit>** , untuk melihat kondisi file commit sebelumnya berdasarkan id



***TERIMA KASIH***  
***PERHATIANNYA***

PRESENTASI SELESAI

