



# Day 1 Training ARC 2024

Introduction to Git and Github



# Konten Materi

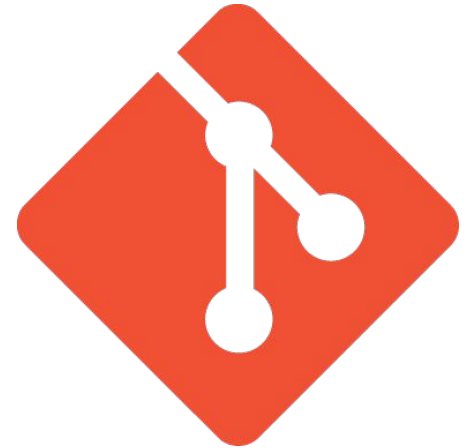
1. Apa itu Git?
2. Git workflow
3. Hands-on git
4. Istirahat
5. Apa itu Github dan remote repository?
6. Hands-on Github

# Apa sih itu git?

Git adalah sebuah software *version control*, banyak manfaat yang dapat didapatkan dari software ini. Kurang lebih:

1. Tracking history revisian
2. Workflow kolaborasi yang baik untuk programming

**Fun fact:** “bisa pake git” itu merupakan sebuah keharusan/asumsi standar untuk software engineer



# Siapa sih penemu/pembuat git?

---

**Linus Benedict Torvalds** adalah penemu dari git, dia merupakan software engineer yang terkenal karena juga merupakan:

1. Perintis Linux
2. Programmer yang cukup sensasional



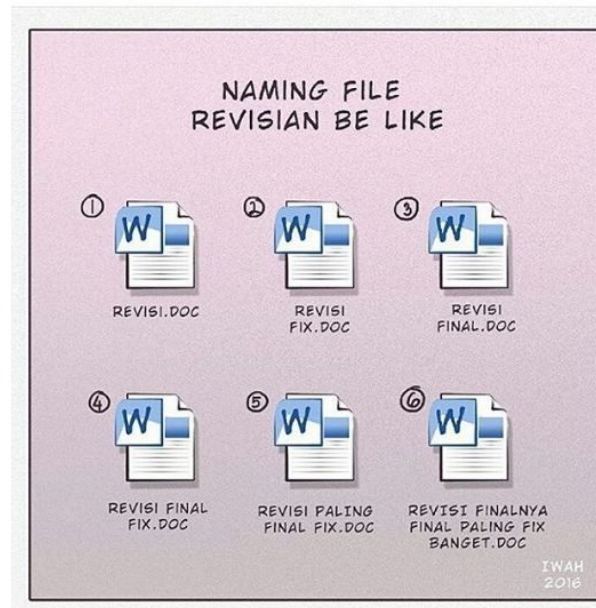
# Siapa sih penemu/pembuat git?

---



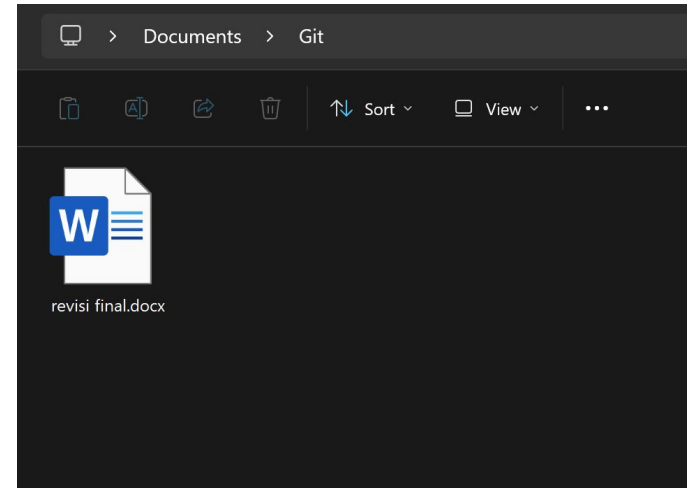
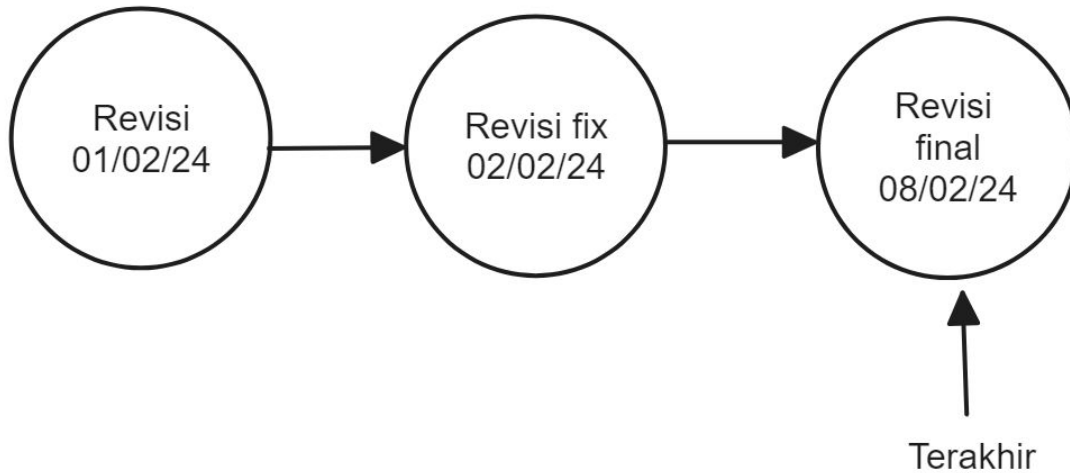
# Permasalahan apa sih yang git selesaikan?

## Version control



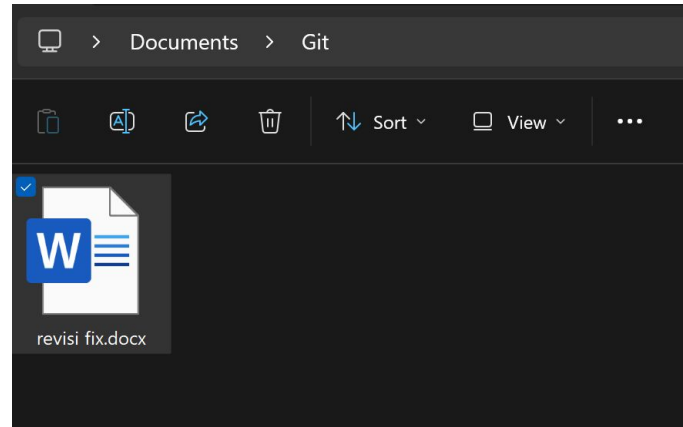
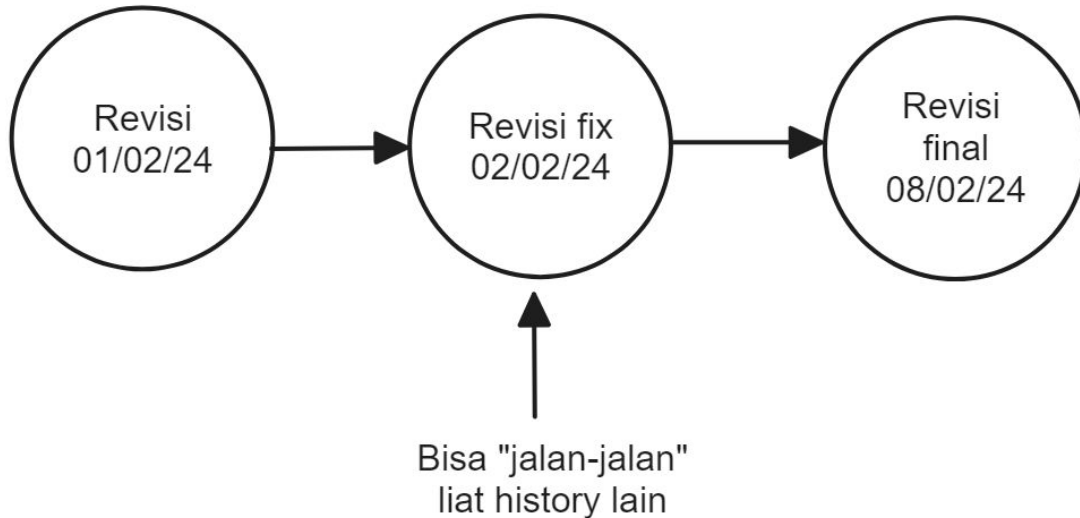
# Permasalahan apa sih yang git selesaikan?

History dalam 1 folder



# Permasalahan apa sih yang git selesaikan?

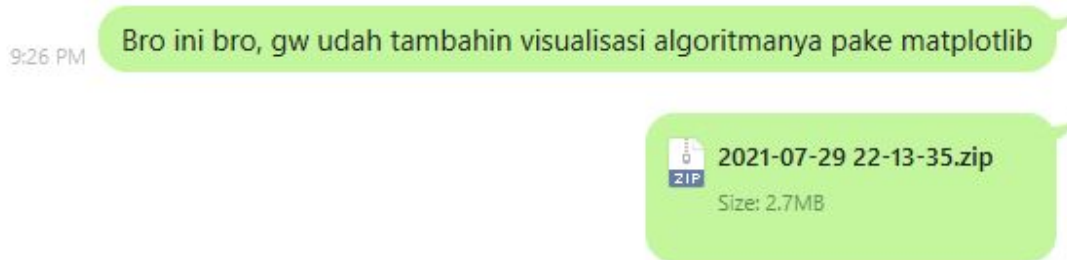
History dalam 1 folder



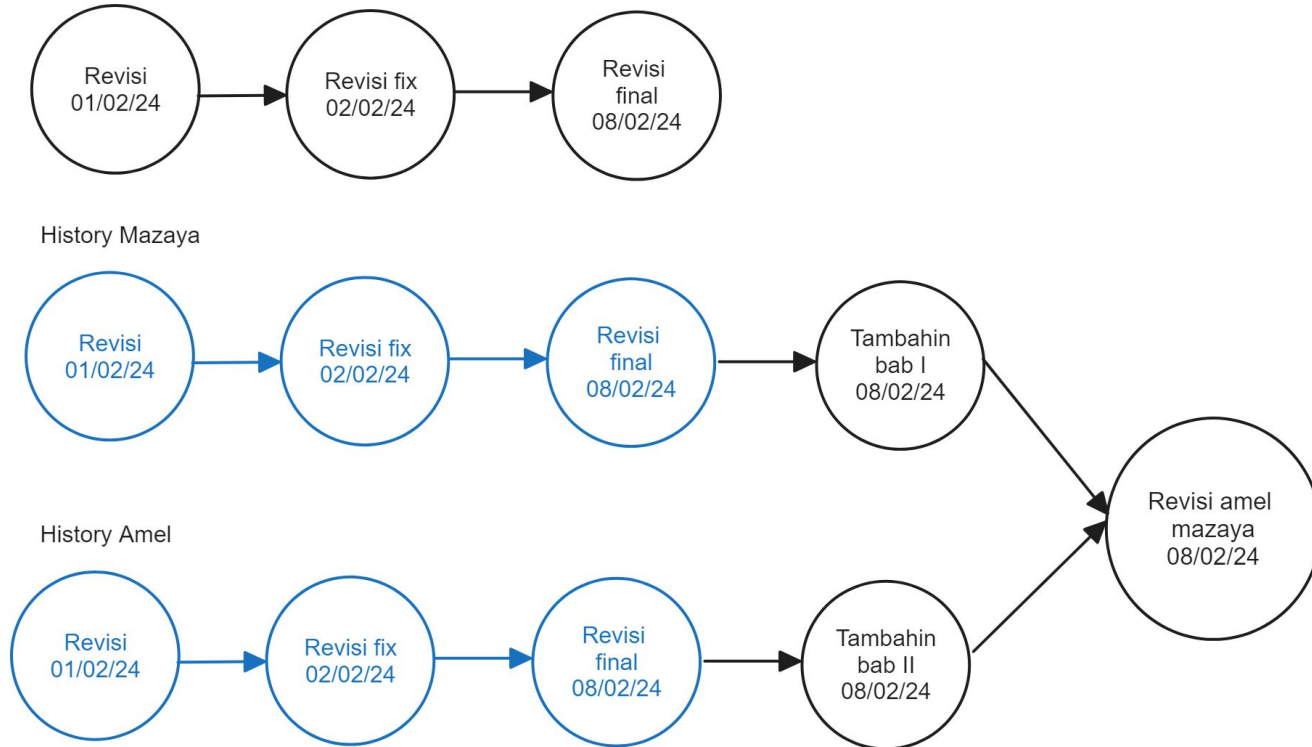


# Permasalahan apa sih yang git selesaikan?

## Collaboration



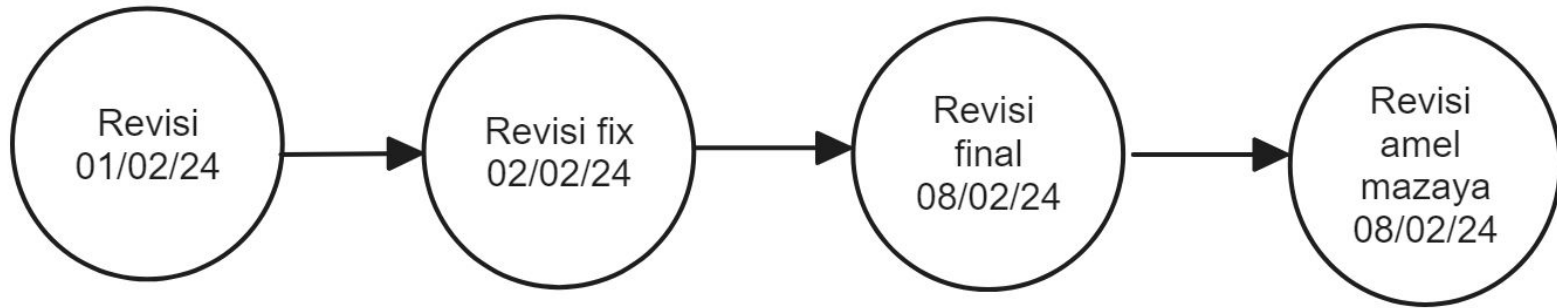
# Permasalahan apa sih yang git selesaikan?



# Permasalahan apa sih yang git selesaikan?



History



# Git Workflow

---



## 3 Git stages

Ada 3 tahap/stages di git:

1. Perubahan
2. Pemilihan (staging)
3. Penyimpanan (committed)

Akan dibahas persatu selagi jalan hands-on.

# Hands on

---

# Gimana sih cara pake git?



Git merupakan software yang ada **pada terminal**. Untuk mengecek apakah git sudah terinstall secara baik, ketik hal berikut pada terminal: **git**

# Gimana sih cara pake git?

```
PS C:\Users\msult> git
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>]
          [--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.
```

Output bila sukses



# Mengenal git repository



Sebuah repository merupakan sebuah folder yang sudah diinisiasi oleh git. Ciri-ciri dari folder yang merupakan repository adalah adanya folder `.git`.

Note: bila membuat folder `.git` secara manual, tentunya tidak akan otomatis menjadikan folder tersebut sebagai repository. Namun, harus diinisiasi oleh git menggunakan *command* tertentu.

# Mengenal git repository



Hands-on contoh melihat salah satu git repository.

# Mengenal git repository



Hands-on contoh melihat salah satu git repository.

# Apa sih itu command?

Sebelum mencoba membuat sebuah repository, baiknya mengenal apa itu yang dimaksud oleh *command*. *Command* adalah sebuah perintah kepada git untuk menjalankan/melakukan sesuatu.

formatnya: `git [command]`



# Membuat sebuah repository



Untuk mengubah sebuah **folder biasa menjadi repository**, dapat digunakan command **init**.

formatnya: `git init`

# Membuah sebuah repository



Bila sukses:

```
PS C:\Users\msult\arc-git> git init  
Initialized empty Git repository in C:/Users/msult/arc-git/.git/
```

# Status git



Revisit, git ada 3 tahapan:

1. Perubahan
2. Pemilihan (staging)
3. Penyimpanan (committed)

Kita dapat melihat status dari sebuah repository, dengan menggunakan command **status**.

formatnya: `git status`

## Status perubahan (untracked)

```
PS C:\Users\msult\arc-git> git status  
On branch master
```

```
No commits yet
```

```
nothing to commit (create/copy files and use "git add" to track)
```

Kita dapat mencoba untuk menambahkan sebuah file pada repository. Lalu check apakah terdapat perubahan status dari git status sebelumnya. Misalnya, membuat sebuah file **arc-gokil.txt**



# Status perubahan (untracked)

```
PS C:\Users\msult\arc-git> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    arc-gokil.txt

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

Ini adalah status yang disebut status perubahan (untracked). Status ini, menandakan terdapat sebuah perubahan yang belum kita simpan. Bila, misalnya, ingin menyimpan perubahan maka dapat menggunakan command **add**.

formatnya: `git add [nama file/folder]`

# Status pemilihan

```
PS C:\Users\msult\arc-git> git add .\arc-gokil.txt
PS C:\Users\msult\arc-git> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   arc-gokil.txt
```

Sekarang terlihat bahwa terdapat file yang disebut “changes to be committed”. Berarti, filenya sudah siap untuk disimpan. Untuk menyimpan file/perubahan, dapat dilakukan command **commit**.

formatnya: `git commit -m “[judul penyimpanan]”`

## Status pemilihan



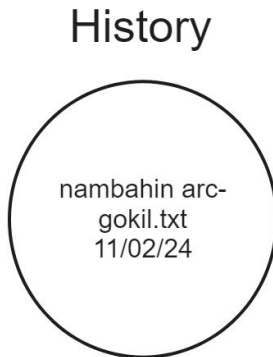
```
PS C:\Users\msult\arc-git> git commit -m "nambahin arc-gokil.txt"
[master (root-commit) 9ba8a62] nambahin arc-gokil.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 arc-gokil.txt
```

Sekarang, terlihat bahwa file sudah “disimpan”, bila dijalankan status lagi maka yang muncul seperti awal:

```
PS C:\Users\msult\arc-git> git status
On branch master
nothing to commit, working tree clean
```

# Apa yang sebenarnya terjadi?

Sebenarnya, yang terjadi pada saat kita melakukan command-command tadi, itu membuat sebuah history seperti berikut:



History ini, bisa dilihat dari command **log**.

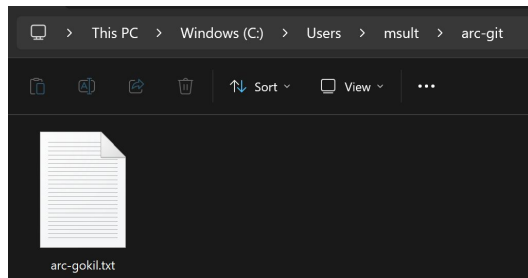
formatnya: `git log`

# Apa yang sebenarnya terjadi?

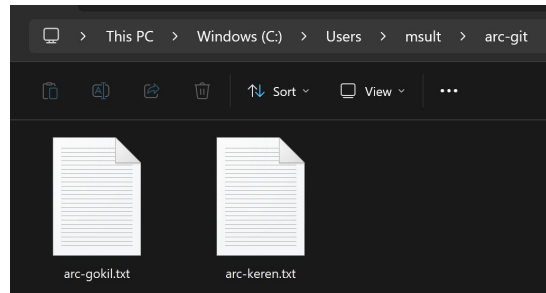
```
PS C:\Users\msult\arc-git> git log
commit 9ba8a621aa2b057e992aeb5085a9cafada67d98 (HEAD -> master)
Author: Muhammad Sulthan Mazaya <msulthanmazaya@gmail.com>
Date:   Sun Feb 11 11:04:39 2024 +0700

    nambahin arc-gokil.txt
```

Tentunya, karena history-nya masih satu, git ini masih belum terlalu berguna. Sekarang, kita coba untuk menambahkan 1 file lagi yaitu **arc-keren.txt**. Sehingga, kita punya 2 history:



History pertama hanya ada 1 file



History pertama ada 2 file

# Tambah arc-keren.txt



Caranya, sama kayak yang awal:

1. Bikin file arc-keren.txt
2. `git add arc-keren.txt`
3. `git commit -m "tambahin arc-keren.txt"`
4. Lihat history dengan `git log`

# Hasil git log setelah tambah arc-keren.txt

```
PS C:\Users\msult\arc-git> git add .\arc-keren.txt
PS C:\Users\msult\arc-git> git commit -m "nambahin arc-keren.txt"
[master 96b318c] nambahin arc-keren.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 arc-keren.txt
PS C:\Users\msult\arc-git> git log
commit 96b318c1b6ba0ca2156b91a0bc048ec021921948 (HEAD -> master)
Author: Muhammad Sulthan Mazaya <msulthanmazaya@gmail.com>
Date:    Sun Feb 11 11:17:27 2024 +0700

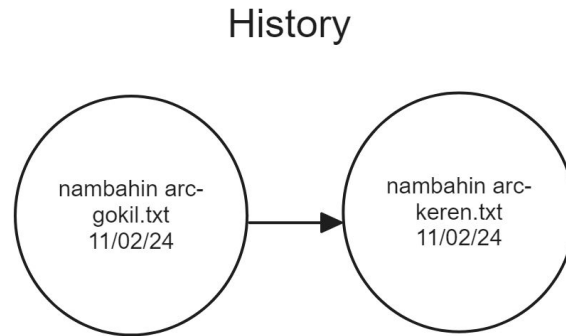
    nambahin arc-keren.txt

commit 9ba8a621aa2b057e992aeb5085a9cafadaf67d98
Author: Muhammad Sulthan Mazaya <msulthanmazaya@gmail.com>
Date:    Sun Feb 11 11:04:39 2024 +0700

    nambahin arc-gokil.txt
```

# History terbaru

Nah, sekarang history-nya udah ada dua:



Terus, cara “pindah” dan melihat history yang sebelum-sebelumnya bagaimana? Dapat dilakukan dengan menggunakan command checkout.

formatnya: `git checkout [id history]`



# Checkout

Apa sih id itu? Dapat dilihat di git log:

```
PS C:\Users\msult\arc-git> git log
commit 96b318c1b6ba0ca2156b91a0bc048ec021921948 (HEAD -> master)
Author: Muhammad Sulthan Mazaya <msulthanmazaya@gmail.com>
Date:    Sun Feb 11 11:17:27 2024 +0700

    nambahin arc-keren.txt

commit 9ba8a621aa2b057e992aeb5085a9cafadaaf67d98
Author: Muhammad Sulthan Mazaya <msulthanmazaya@gmail.com>
Date:    Sun Feb 11 11:04:39 2024 +0700

    nambahin arc-gokil.txt
```

# Checkout



Misalnya, ingin checkout/pindah ke history nambahin arc-gokil, maka idnya:

```
commit 9ba8a621aa2b057e992aeb5085a9cafadaf67d98
Author: Muhammad Sultnan Mazaya <msultnanmazaya@gmail.com>
Date:   Sun Feb 11 11:04:39 2024 +0700

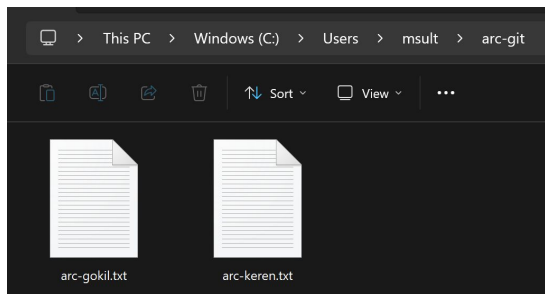
    nambahin arc-gokil.txt
```

Sehingga, command untuk checkoutnya adalah:

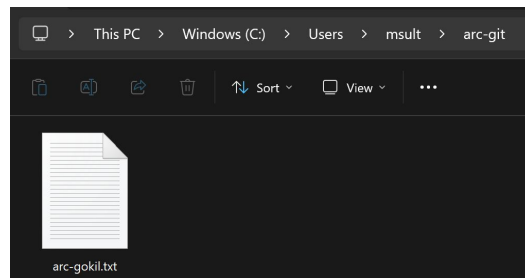
git checkout 9ba8a621aa2b057e992aeb5085a9cafadaf67d98

# Checkout

Setelah run command tersebut, harusnya, foldernya terdapat perubahan.



Sebelumnya



Sesudahnya

Bila ingin kembali, maka bisa menggunakan “git checkout -” tanpa kutip.

# Disclaimer



Sebenarnya:

1. Yang daritadi disebut history itu namanya **commits**, jadi tadi kita bikin 2 commit.
2. Add file, itu ngga harus 1 per 1, bisa langsung multiple:  
`git add [file 1] [file 2] [...]`
3. Bisa juga seluruh file baru:  
`git add .`
4. Bisa juga sebenarnya perubahannya itu konten file-nya

# Recap commands



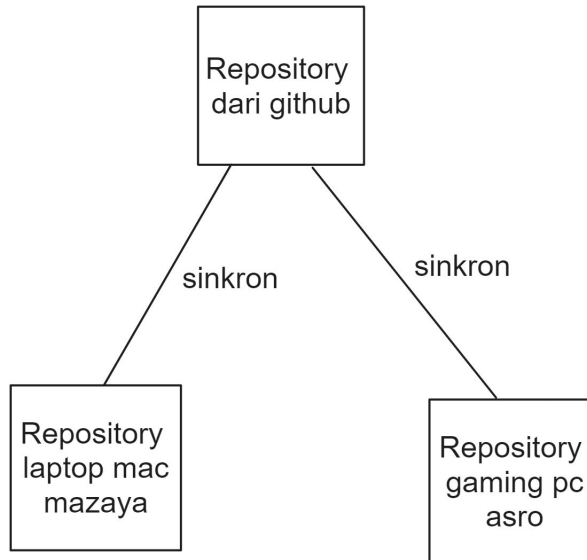
1. `init`: membuat sebuah repository
2. `add`: menambahkan sebuah file/folder untuk disimpan
3. `commit`: membuat/menyimpan sebuah commit baru
4. `status`: mengecek status repository
5. `log`: melihat history commits
6. `checkout`: pindah commit

**Istirahat dulu 15 menit :D**

---

# Apa itu Github dan remote repository?

Github merupakan, semacam, provider remote repository. Maksudnya, Github itu merupakan perusahaan yang punya folder-folder repository di cloud. Jadi, kita bisa memiliki repository yang ada di cloud dan dapat disinkronisasikan antar device maupun antar user lain.



# Daftar Github

---





**Nunggu 3 menit pada daftar  
github**

---

# Janlup follow saiah EHEHEHE




**Muhammad Sulthan**


**Mazaya**


MSMazaya

Follow

Also known as Mazaya/Jay

 **51** followers · **37** following

 Indonesia

 mazaya.id

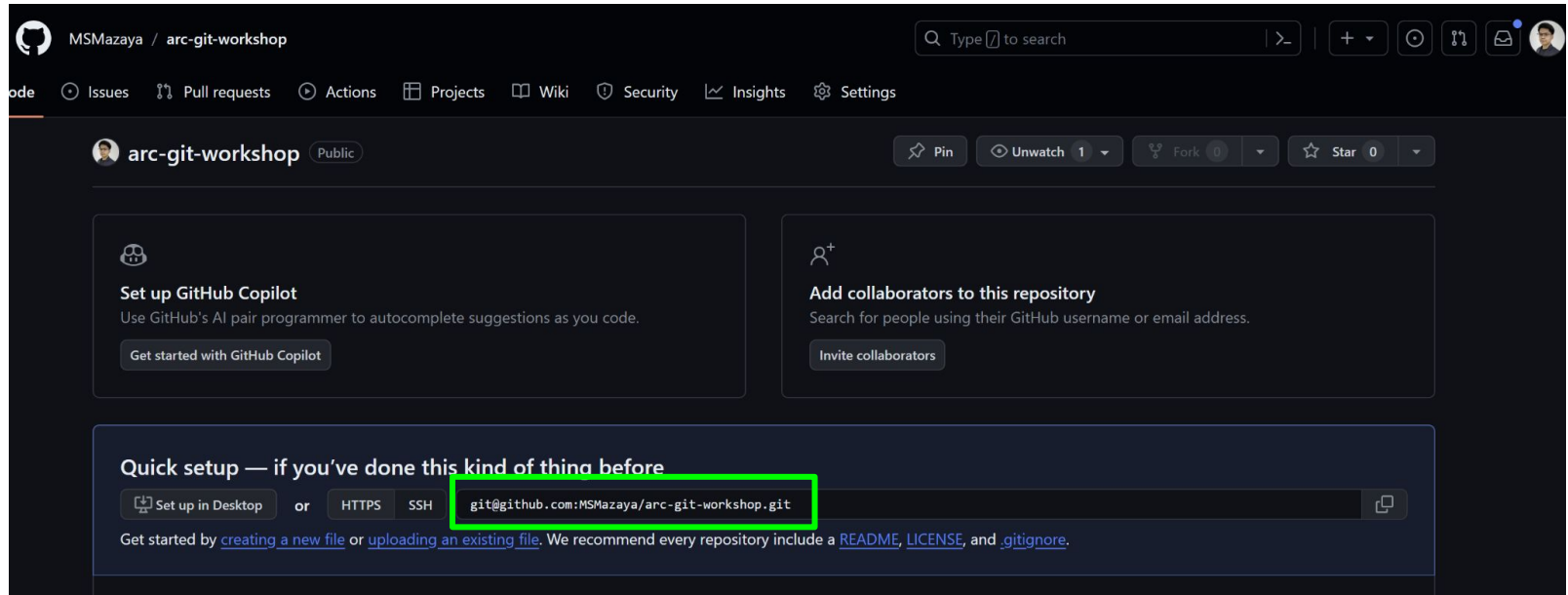
# Hands-on bikin repository di github

---

# Cara mengunduh remote repository github

Untuk mengunduh remote repository github, dapat digunakan command `clone`.

formatnya: `git clone [alamat url remote repository]`



# Menambahkan file baru



Tambahkan sebuah commit baru yang membuat sebuah file bernama **github-gokil.txt**.

1. Buat file
2. `git add github-gokil.txt`
3. `git commit -m "commit pertama"`

Lalu, bagaimana bila ingin melakukan sinkronisasi antara lokal (laptop sendiri) dengan github? Dapat menggunakan command `git push`.

formatnya: `git push [nama remote] [nama branch]`

# Menambahkan file baru

Cara checking branch, dapat dilakukan command `git branch -v`

```
PS C:\Users\msult\arc-git> git branch -v
* master 96b318c nambahin arc-keren.txt
```

Berarti, branch kita adalah master. Lalu, untuk melakukan checking nama remote dapat menggunakan `git remote -v`.

```
PS C:\Users\msult\arc-git> git remote -v
origin git@github.com:MSMazaya/arc-git-workshop.git (fetch)
origin git@github.com:MSMazaya/arc-git-workshop.git (push)
```

# Menambahkan file baru



Sehingga, untuk lakukan sinkronisasi, commandnya: **git push origin master**

Namun, sinkronisasi **dari** dan **ke** github, itu berbeda commandnya. Untuk sinkronisasi **dari** github, itu menggunakan **git pull** dengan format yang sama dengan git push.

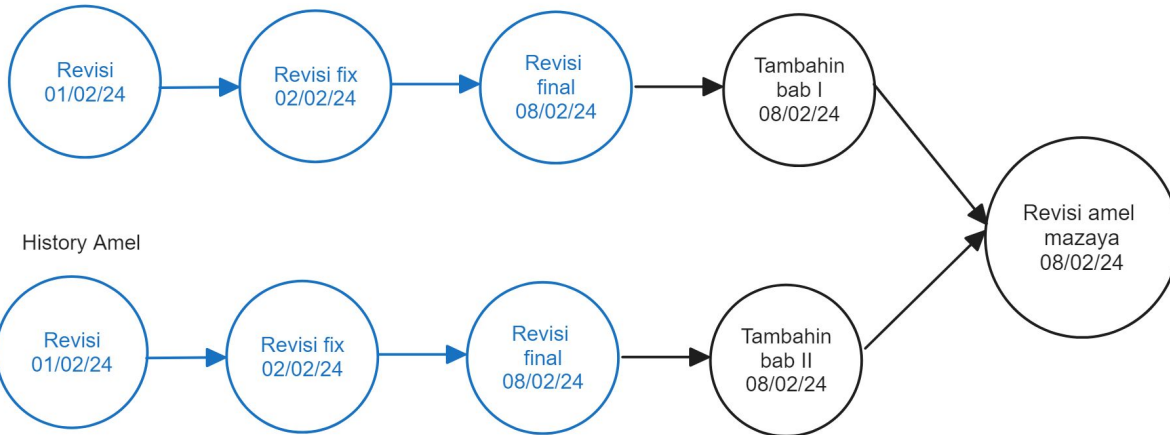
formatnya: `git pull [nama remote] [nama branch]`

# Apa sih itu branch?

Pada kasus di bawah, history mazaya dan history amel adalah contoh branch



History Mazaya





# Apa sih itu branch?



Untuk membuat branch baru, dapat menggunakan command checkout.

formatnya: `git checkout -b [nama branch]`

Lalu, untuk menggabungkan 2 branch, dapat dengan command merge.

formatnya: `git merge [nama branch]`

# FYI



Ada beberapa aplikasi pembantu untuk ngelakuin command-command git tersebut :D

1. Github desktop
2. Langsung di VSCode
3. LazyGit <- yang gw pake di ubuntu



# Thanks!

Ada pertanyaan?

# Tugasnya



Pelajari:

1. Apa itu merge conflict dan cara menanggulangnya
2. Apa itu git rebase
3. Apa itu fork dan pull request github

Buatlah jawaban dari 3 pertanyaan diatas dalam sebuah file .txt, beri nama: jawaban\_soal\_1.txt, jawaban\_soal\_2.txt, dan jawaban\_soal\_3.txt.

Lalu, jawaban tersebut upload ke github repository masing-masing yang **dibuat public**. Nanti akan ada link pengumpulan tugas.