

Hari-1-Git & Command Line

Mengenal version control Git dan version control platform Gitlab

Membuat project Git dan mengelolanya

Pastikan sudah install git di komputer anda. Jika belum terinstall silakan ikuti langkah-langkahnya di sini: <https://www.digitalocean.com/community/tutorials/how-to-contribute-to-open-source-getting-started-with-git>

Materi Video Hari Pertama

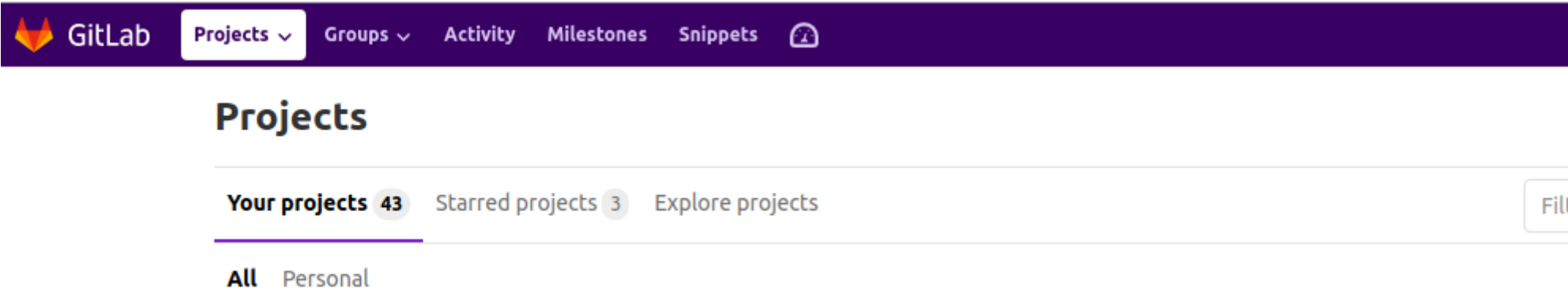


bagi teman teman yang membutuhkan materi video silahkan akses video diatas

Membuat Repository Baru

Repository atau repo di Gitlab biasa disebut dengan project. Berikut langkah-langkah untuk membuat repository baru:

- Setelah login anda akan menuju halaman Projects yang berisi semua repository yang anda miliki. Jika Anda user baru maka belum ada repository yang tersedia. Di halaman tersebut anda akan melihat tombol New Project di sebelah kanan atas. Klik tombol tersebut.



- Pada halaman membuat project baru, isi kolom project-name dengan nama yang Anda inginkan, misalkan: my first repo.
- Pada kolom visibility level Anda akan melihat tiga pilihan yaitu Private, Internal, dan Public. Private artinya repo yang akan kita buat bersifat rahasia, hanya Anda dan orang yang Anda beri akses saja yang bisa membukanya. Sedangkan pilihan Internal berarti repo Anda akan dapat dilihat oleh semua user Gitlab. Pilihan “Public” berarti repo Anda akan terlihat oleh orang dari mana saja walaupun dia tidak memiliki akun gitlab. Pilih yang sesuai kebutuhan misalkan : Public.
- Di bagian bawahnya terdapat pilihan checkbox “Initialize repository with a README”. Centang saja jika anda ingin langsung clone repository nya setelah dibuat.

New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), [among other things](#).

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

To only use CI/CD features for an external repository, choose **CI/CD for external repo**.

Information about additional Pages templates and how to install them can be found in our [Pages getting started guide](#).

Tip: You can also create a project from the command line. [Show command](#)

Blank project

Create from template

Import project

Project name

my-first-repo

Project URL

https://gitlab.com/

mabdvh

▼

Project slug

my-first-repo

Want to house several dependent projects under the same namespace?

Create a group.

Project description (optional)

Description format

Visibility Level ⓘ

☒ Private

Project access must be granted explicitly to each user.

☐ Internal

The project can be accessed by any logged in user.

☐ Public

The project can be accessed without any authentication.

☒ Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an exist

Create project

- Jika berhasil maka Anda akan diarahkan menuju halaman project atau repo yang telah dibuat. Jika sebelumnya kita centang “Initialize repository with a README” maka repository kita akan berisi file README.md.
- Pada Bagian kanan atas terdapat tombol clone. Ketika diklik maka akan keluar dua pilihan “Clone with SSH” dan “Clone with HTTPS”. Pilih saja clone dengan HTTPS. Klik tombol salin/copy supaya langsung otomatis tercopy pada clipboard. Jika ingin clone dengan SSH maka anda harus setting dulu akun anda agar terhubung dengan komputer Anda.

Muhamad Abduh > my-first-repo > Details

Project 'my-first-repo' was successfully created.

M

my-first-repo ⓘ

Project ID: 13167128

Add license

0 Commits

1 Branch

0 Tags

0 Bytes Files

Auto DevOps

It will automatically build, test, and deploy your application base

Learn more in the [Auto DevOps documentation](#)

Enable in settings

master ▼

my-first-repo /

+ ▼

History

Find file

Initial commit

Muhamad Abduh authored just now

Clone with SSH

git@gitlab.com:mabdvh/

Clone with HTTPS

https://gitlab.com/mab

- Setelah itu menuju ke terminal atau cmd. Arahkan menuju folder dimana anda ingin menyimpan repository yang sudah Anda buat. Berikan perintah git clone <link_anda_yang_barusan_dicopy> (tanda <> hanyalah sebagai penanda template tidak usah ditulis di

- Jika sudah berhasil diclone maka akan ditemukan folder dengan nama repository di Gitlab. Cara mengeceknya dengan perintah “ls” atau “dir”.

```
$ ls
my-first-repo
```

Mengupload perubahan project dengan git push

Ketika kita sudah berhasil clone repository kita di Gitlab maka tentu kita ingin melakukan update atau penambahan code pada project kita. Agar perbaruan yang kita simpan di local (komputer) kita dapat kita perbaharui juga di repository Gitlab maka kita bisa lakukan dengan git push.

Berikut langkah-langkahnya:

- Ketika kita sudah memberikan perubahan terhadap suatu file atau membuat folder-folder baru pada folder project kita maka cara mengecek status perbaruan adalah dengan git status

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

- Setelah git status maka akan ditampilkan list file yang “modified” atau “untracked”. “modified” artinya adalah kita mengubah file yang sebelumnya ada sedangkan “untracked” artinya file tersebut sama sekali baru di repository. Dalam contoh di atas terdapat perubahan pada file “README.md”.
- Untuk menaikkan perubahan (stage) yang kita buat maka berikan perintah git add

```
$ git add README.md

// Jika ada terdapat banyak file dan kita ingin naikan semuanya maka perintahnya:
$ git add .
```

- Sekarang perubahan tadi sudah pada tahap “staged”. jika kita cek status nya lagi maka akan didapati berbeda:

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   README.md
```

- Selanjutnya adalah memberikan laporan atau commit terhadap perubahan yang kita buat dengan git commit.

```
$ git commit -m "Pesan saya: Mengubah README.md"
```

- tanda “-m” pada perintah commit artinya kita ingin memberikan pesan atau laporan terhadap perubahan yang sudah dibuat. Dalam hal ini laporannya adalah “Pesan saya: mengubah README.md”
- Selanjutnya adalah push untuk mengirim /upload perubahan yang ada di local computer kita menuju repository yang ada di Gitlab. Jika kita berada di branch bernama master maka perintahnya adalah:

```
$ git push origin master
```

Jika branch yang sedang kita miliki di local adalah branch lain selain master maka cara push nya adalah

```
$ git push origin <nama_branch>
```

Jika langkah tersebut sudah maka kita sudah berhasil mengupload perubahan yang kita buat di repository kita. 🥂🎉

Git branch dan merge request (optional)

Membuat branch baru dan melakukan merge request

Clone repository

Terlebih dahulu lakukan clone terhadap repository project.

```
$ git clone https://gitlab.com/some-project
$ cd some-project
```

Catatan: Link https di atas hanyalah contoh link dari sebuah repository. Link https tersebut bisa diperoleh di halaman gitlab pada bagian clone.

nama yang diinginkan)

```
$ git branch myname
$ git checkout myname
```

maka sekarang kita sudah berada di branch baru dan dapat mulai melakukan pekerjaan kita.

Catatan:

myname hanya contoh , silakan sesuaikan sendiri nama branch masing-masing.

Push ke branch

Setelah pekerjaan selesai, kita akan melakukan upload atau push ke branch kita sendiri.

```
$ git add .
$ git commit -m "some message"// push ke branch
$ git push origin myname
```

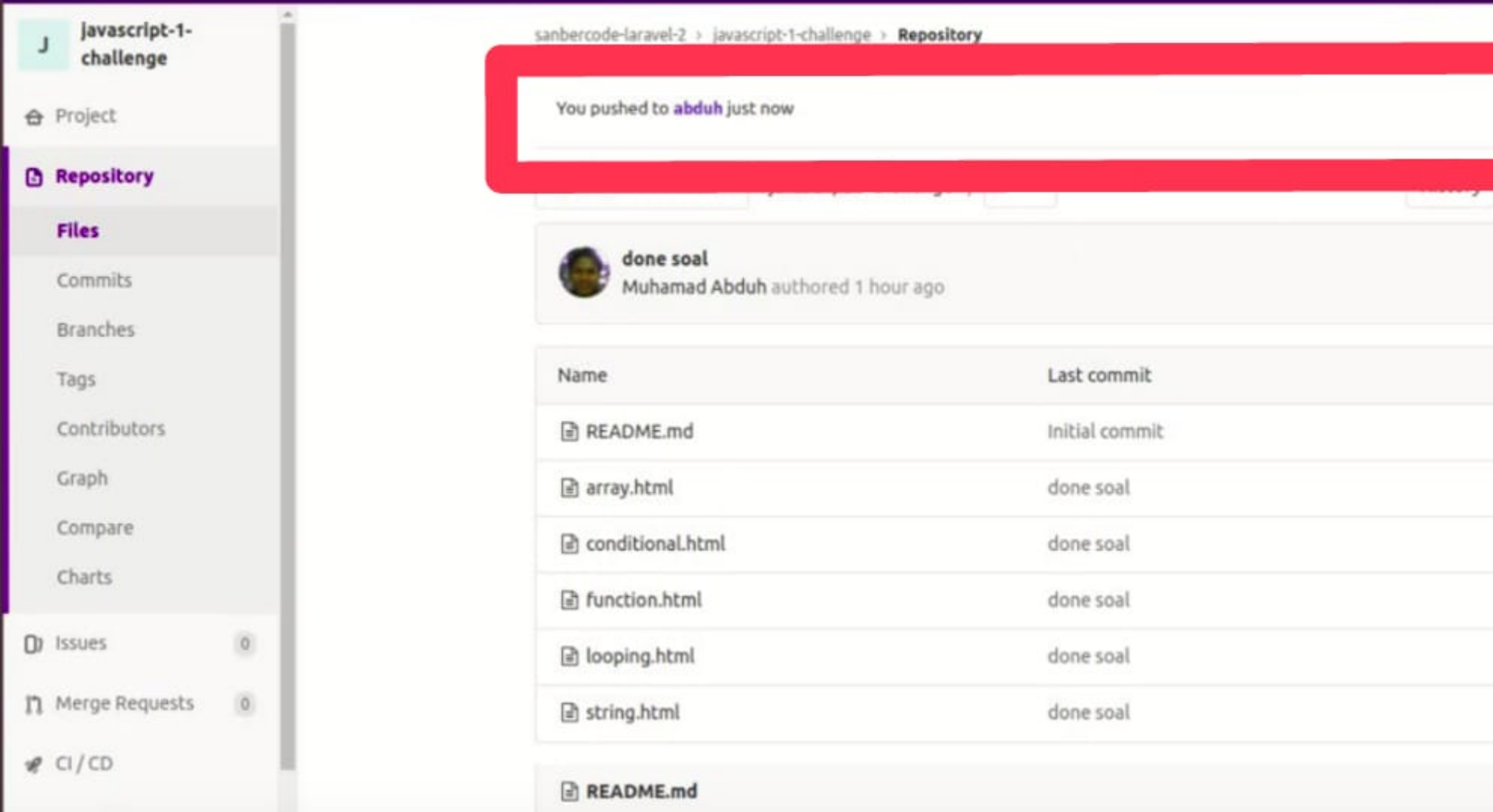
Catatan:

"some message" hanyalah contoh pesan atau komentar pada sebuah commit. Silahkan isi dengan jenis update yang dikerjakan.

Membuat merge request

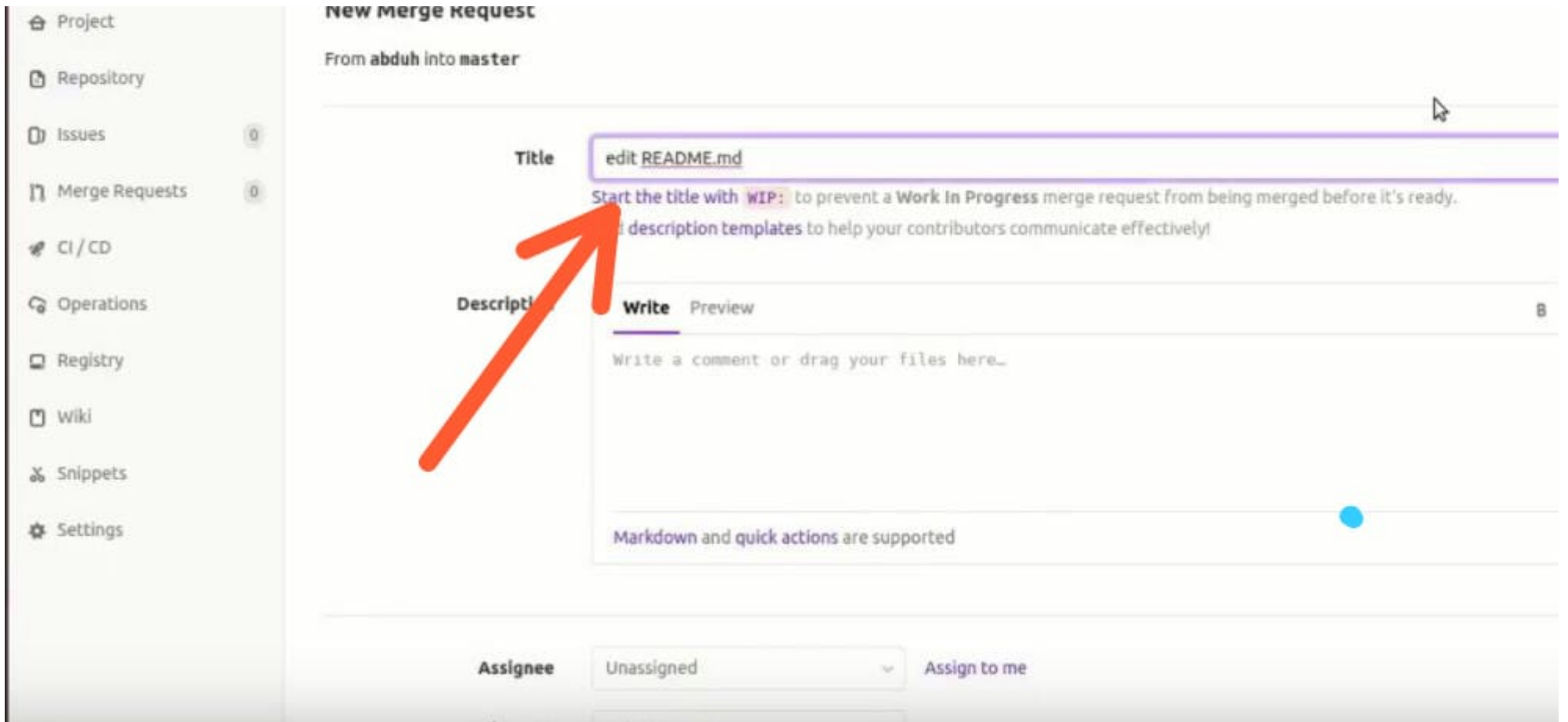
Setelah melakukan push ke branch sendiri maka selanjutnya melakukan merge request.

Secara default pada halaman project akan tersedia tombol “create merge request”.



Jika tombol tersebut tersedia, klik tombol tersebut untuk menuju halaman merge request.

Pada halaman merge request isilah kolom title dengan pesan commit atau laporan progress pekerjaan misalkan : done, work in progress atau WIP, dsb.



Setelah itu, klik submit merge request maka pekerjaan kita sudah disubmit ke merge request dan akan direview oleh pengelola project.

Selengkapnya kamu bisa ikuti juga video tutorial berikut ini: [tutorial merge request](#).

- vscode , link download : <https://code.visualstudio.com/download>
- Cmder, *link download*: <https://cmder.net/cmder>
- Git bash, link download: <https://git-scm.com/downloads>, Git bash biasanya ter-*install* jika kita meng-*install* git di pc/laptop kita.

2. Command Line

Mengenal perintah-perintah dasar pada CLI (Command Line Interface)

Apa itu command line interface?

Command line interface atau yang biasa kita kenal cmd atau *command prompt* atau *terminal* adalah *software* atau antar muka untuk memberikan perintah-perintah dasar pada komputer kita. Biasanya kita mengoperasikan komputer seperti *browsing media*, mencari *file*, membuat *file* atau *folder*, dan lain-lain dengan GUI (*Graphical User Interface*) yang sudah disediakan pengembang Sistem Operasi seperti *Windows*, *Linux*, atau *OSX*. Namun pada dasarnya kita juga bisa melakukan operasi-operasi tersebut dalam suatu perintah yang kita kirimkan melalui cmd/ *terminal*. Berikut ini beberapa contohnya:

Menuju *folder*/direktori tertentu

Untuk menuju *folder* atau direktori tertentu bisa gunakan perintah *cd* atau kepanjangan dari *change directory*

```
// Menuju Desktop
$ cd Desktop
// Menuju ke direktori parent (sebelumnya)
$ cd ..
```

Mengetahui alamat dari direktori yang sedang dibuka

Terkadang kita butuh tahu sedang dimana posisi terminal kita berjalan. Perintah untuk fungsi ini yaitu *pwd* atau kependekan dari path of working directory

```
$ pwd
/home/users
// sedang berada di folder home/users
```

Mengetahui daftar isi dari sebuah direktori/ *folder*

Kamu bisa gunakan perintah *ls* atau *dir* . contohnya sebagai berikut

```
$ ls
Document Desktop Image index.html
$ dir
Document Desktop Image index.html
```

```
// membuat folder baru bernama new-folder
$ mkdir new-folder
// mengecek apakah sudah berhasil buat folder baru
$ ls
new-folder ... ..
```

Membuat *file* baru

Gunakan perintah **touch** untuk membuat file baru

```
// membuat file index.html
$ touch index.html
// mengecek apakah sudah berhasil buat file baru
$ ls
index.html ... ..
```

Software *terminal* / *command line* yang direkomendasikan

Untuk OS Windows, *terminal* bawaannya adalah *command prompt* atau cmd yang relatif kurang nyaman untuk dipakai dalam tahap *development*. Oleh karena itu direkomendasikan untuk *install terminal* di bawah ini agar lebih optimal dalam mengerjakan *project*.

- Cmder, *link download*: <https://cmdr.net/cmdr>
- Git bash, *link download*: <https://git-scm.com/downloads>, Git bash biasanya ter-*install* jika kita meng-*install* git di pc/laptop kita.
- vscode , *link download* : <https://code.visualstudio.com/download>

Rating - Feedback

Berikan Rating pada posting ini:



Berikan kritik dan saran..

Submit