

Nama : Putri Laura Latersia Br L Tobing

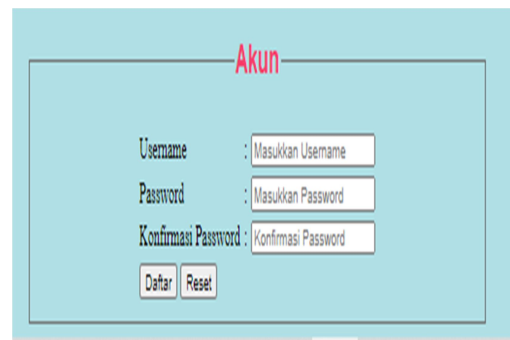
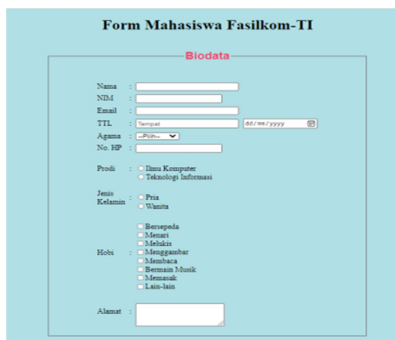
NIM : 191402042

KOM : C

TUGAS 1

Langkah-langkah Pengaplikasian Docker

1. Pertama sekali tentunya harus memiliki website. Di sini saya menggunakan website simple html, bootstrap dan juga css.



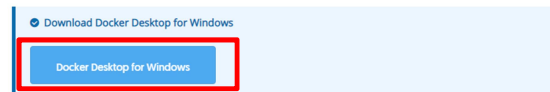
2. Download dan install Docker Windows.

Link download Docker: <https://docs.docker.com/desktop/windows/install/>

Install Docker Desktop on Windows

Estimated reading time: 8 minutes

Welcome to Docker Desktop for Windows. This page contains information about Docker Desktop for Windows system requirements, download URL, instructions to install and update Docker Desktop for Windows.



Docker digunakan bersamaan dengan WSL

Untuk itu saya diminta untuk install WSL terlebih dahulu.

Link download WSL:

<https://docs.microsoft.com/en-us/windows/wsl/install-win10#step-4--->

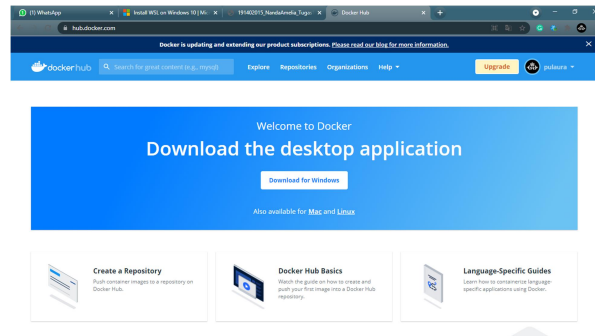
1. Download the latest package:

- [WSL2 Linux kernel update package for x64 machines](#)

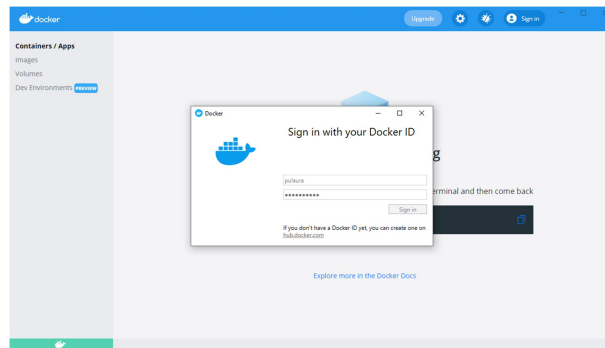
[download-the-linux-kernel-update-package](#)

3. Sign Up bagi yang belum memiliki ID Docker dengan link:

<https://hub.docker.com/>

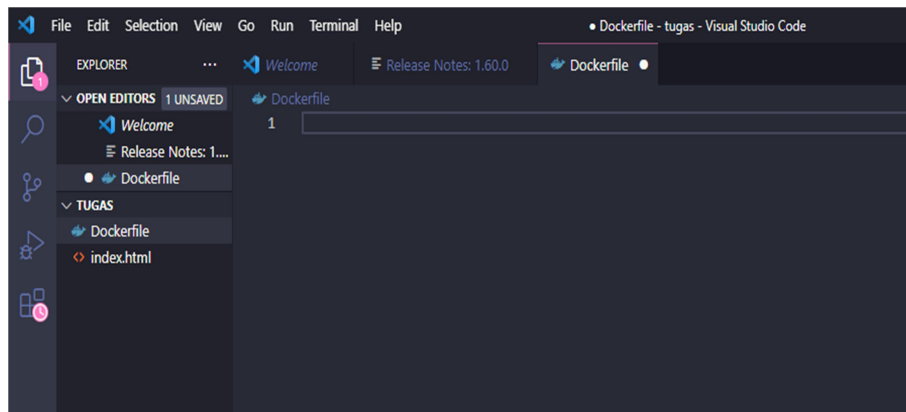


Setelah melakukan sign up dan memiliki akun, maka dapat sign in ke Docker. Docker pun aktif pada device.

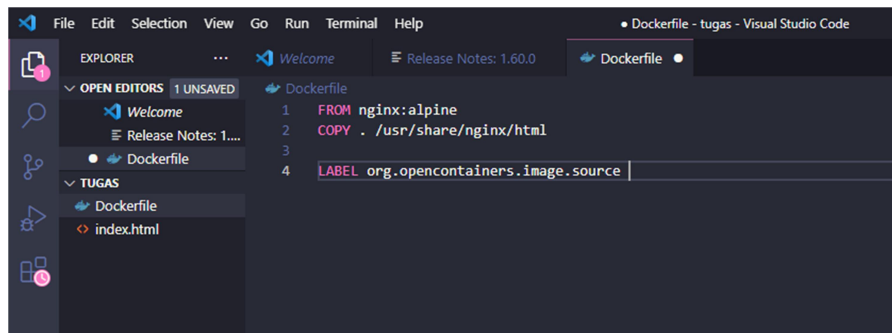


4. Membuat sebuah folder

Dimana isi folder tersebut adalah file 'index.html' yang telah saya buat dan membuka file tersebut di text editor di sini saya menggunakan Visual Studio Code. Kemudian membuat new file "Dockerfile" yang disatukan dalam folder tersebut.

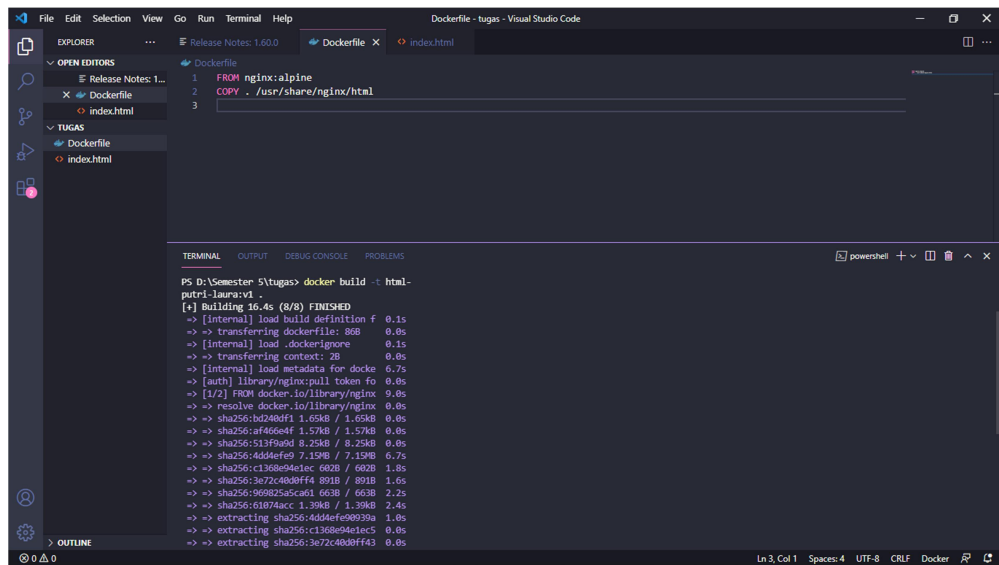


Lalu ketikkan syntax berikut



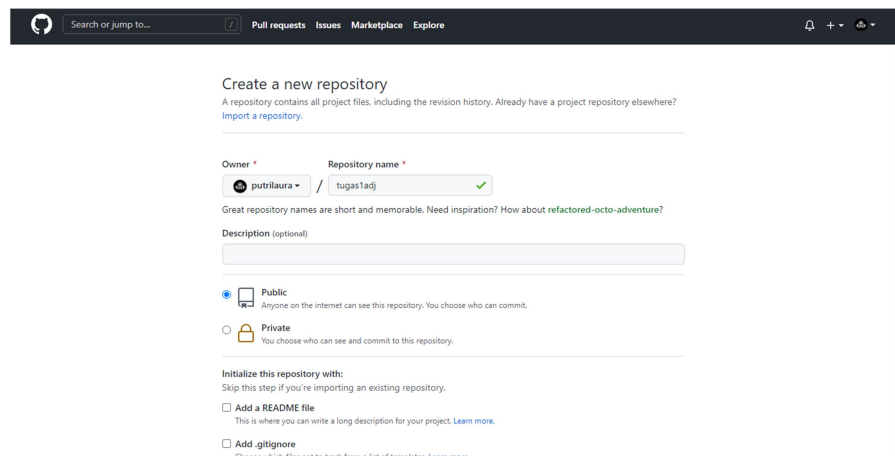
```
1 FROM nginx:alpine
2 COPY . /usr/share/nginx/html
3
4 LABEL org.opencontainers.image.source
```

5. Menjalankan perintah seperti di bawah:



```
PS D:\Semester 5\tugas> docker build -t html-putri-laura:v1 .
[+] building 16.4s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> transferring dockerfile: 868B
=> [internal] load .dockerignore
=> transferring context: 28B
=> [internal] load metadata for docker.io/library/nginx:pull token fo
=> [1/2] FROM docker.io/library/nginx:pull token fo
=> resolve docker.io/library/nginx:pull token fo
=> sha256:b2d40d1f1 1.65kB / 1.65kB
=> sha256:af466eaf 1.57kB / 1.57kB
=> sha256:513f9a9d 8.25kB / 8.25kB
=> sha256:4dd4efe9 7.19MB / 7.19MB
=> sha256:c1368e94ec 602B / 602B
=> sha256:3e72c40b1f4 893B / 893B
=> sha256:96925a5ca61 663B / 663B
=> sha256:61874acc 1.39kB / 1.39kB
=> extracting sha256:4dd4efe90939a
=> extracting sha256:c1368e94ec5
=> extracting sha256:3e72c40b1f4
```

6. Buka <https://github.com/> Kemudian login kea kun github untuk membuat repository baru.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * Repository name *

putrilaura / tugas1adj ✓

Great repository names are short and memorable. Need inspiration? How about [refactored-octo-adventure?](#)

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Initialize this repository with:

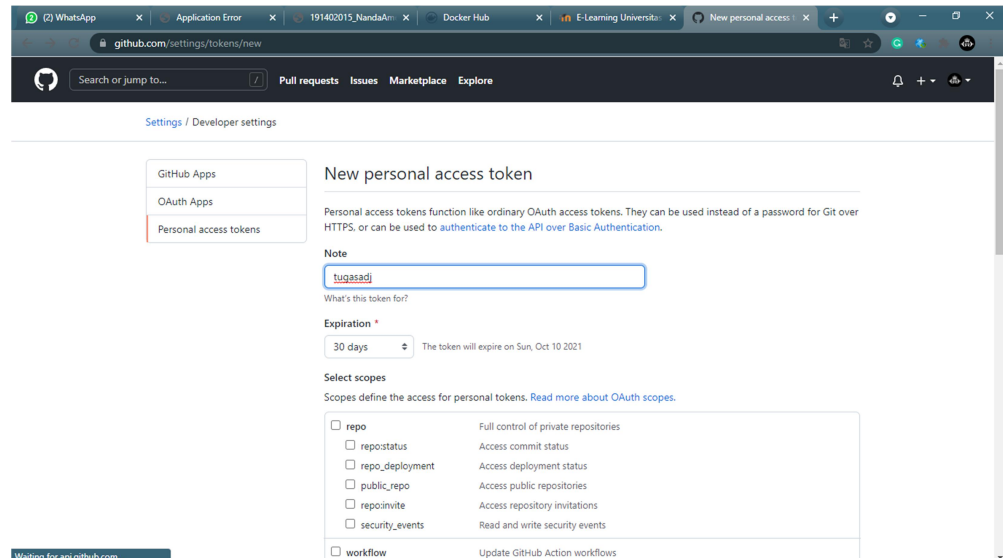
Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

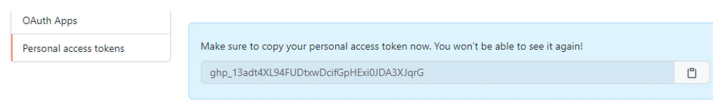
☐ Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

7. Kemudian buka membuat token dengan cara

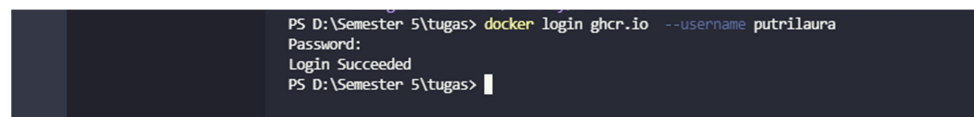
Settings → Developer setting → personal access tokens



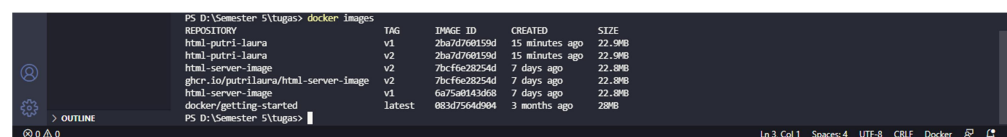
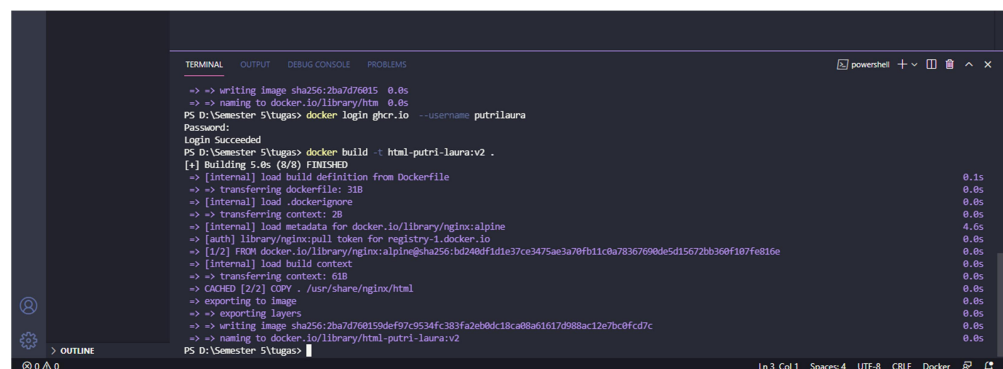
Jangan lupa untuk menyalin token.



8. Kembali ke Visual Studio Code jalan syntax dan paste Personal Token yang telah di copy tadi.



9. Masukkan syntax untuk membuat v2 seperti pada gambar



10. Terlihat ada docker tag v2 dan Image ID. Lalu gunakan Image ID V2 pada syntax berikut:

```
docker/getting-started latest 083d7564d904 3 months ago 28MBPS D:\Semester 5\tugas> docker tag dcb4639cde2d ghcr.io/putrilaura/html-putri-laura:v2
PS D:\Semester 5\tugas> docker push ghcr.io/putrilaura/html-putri-laura:v2
The push refers to repository [ghcr.io/putrilaura/html-putri-laura]
ebb12f49275f: Pushed
40403bebe4fd: Pushed
b4b4e85910ea: Pushed
311d0db33235: Pushed
20d8effdf3a2: Pushed
e6d3cea19fef: Pushed
e2eb0608af82: Layer already exists
v2: digest: sha256:e3c31c88aaa775be3e17f7b1b8e4e387fcd640dbc85c0375c1834b02d857d686 size: 1776
```

11. Menjalankan container yang telah dibuat

```
PS D:\Semester 5\tugas> docker run -d -p 80:80 html-putri-laura:v1
505f642a16d10ced15ee1115139ebfbb83d989af78cc3cf11f0d338543dfcfb5
PS D:\Semester 5\tugas>
```

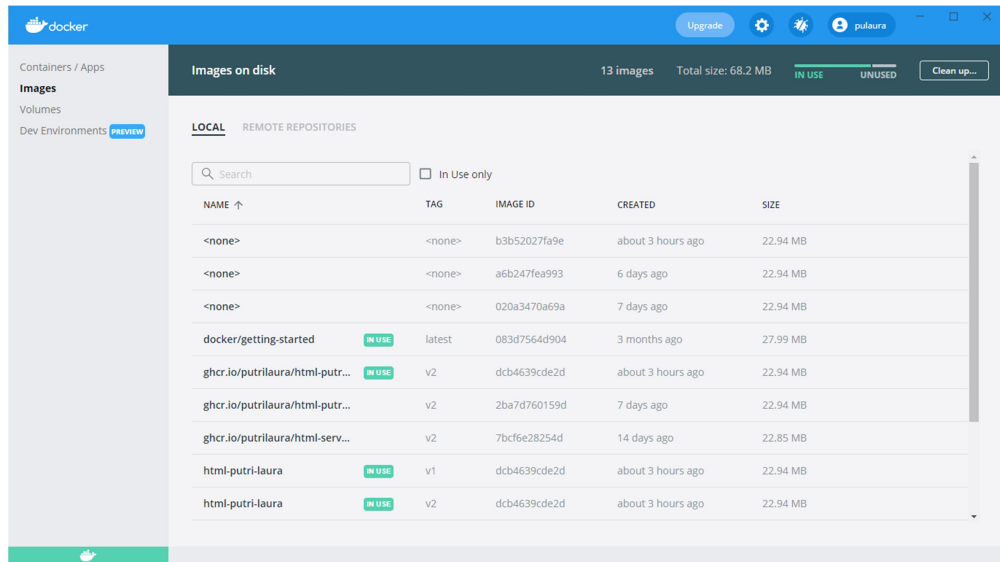
12. Melihat package di Github



13. Buka pada localhost:80

The screenshot shows a web browser window with the address bar set to 'localhost'. The page displays a form titled 'Form Mahasiswa Fasilkom-TI'. The form is divided into two main sections: 'Biodata' and 'Akun'. The 'Biodata' section contains fields for Nama, NIM, Email, TTL (with a date picker set to 16/11/2023), Agama (a dropdown menu), No. HP, Prodi (with radio buttons for 'Ilmu Komputer' and 'Teknologi Informasi'), Jenis Kelamin (with radio buttons for 'Pria' and 'Wanita'), Hobi (with checkboxes for Bersepeda, Menari, Membaca, Bermain Musik, and Lain-lain), and Alamat. The 'Akun' section contains fields for Username (with a placeholder 'Masukkan Username'), Password (with a placeholder 'Masukkan Password'), and Konfirmasi Password (with a placeholder 'Konfirmasi Password').

14. Tampilan Docker images



Link Packages: <https://github.com/putrilaura/tugasadj/pkgs/container/html-putri-laura>

Link Github : <https://github.com/putrilaura/tugasadj>