

Nama : Tritia Mutiara

Nim : 191402048

Kom : C

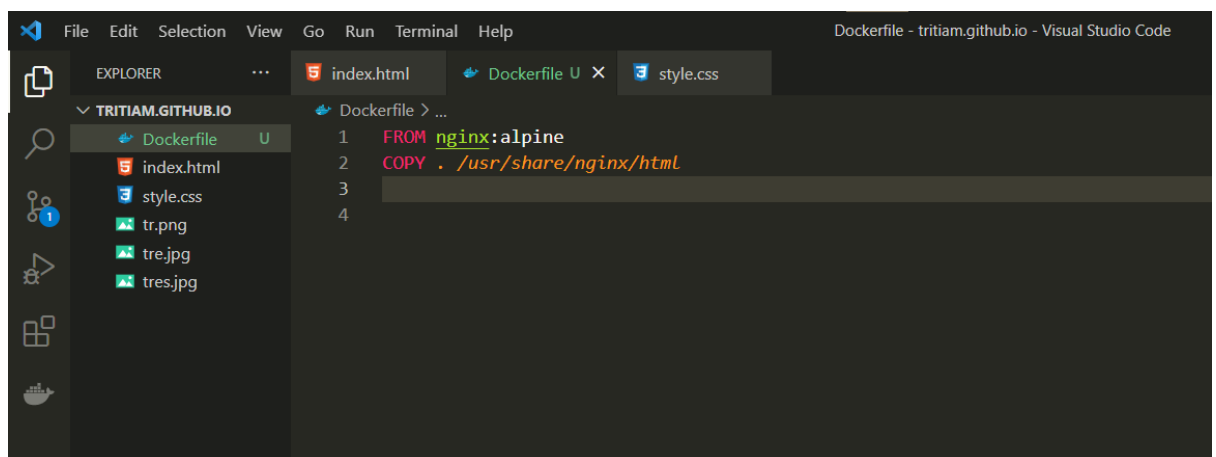
Github: <https://github.com/tritiam>

<https://github.com/users/tritiam/packages/container/package/html-tritia-mutiara>

Tugas

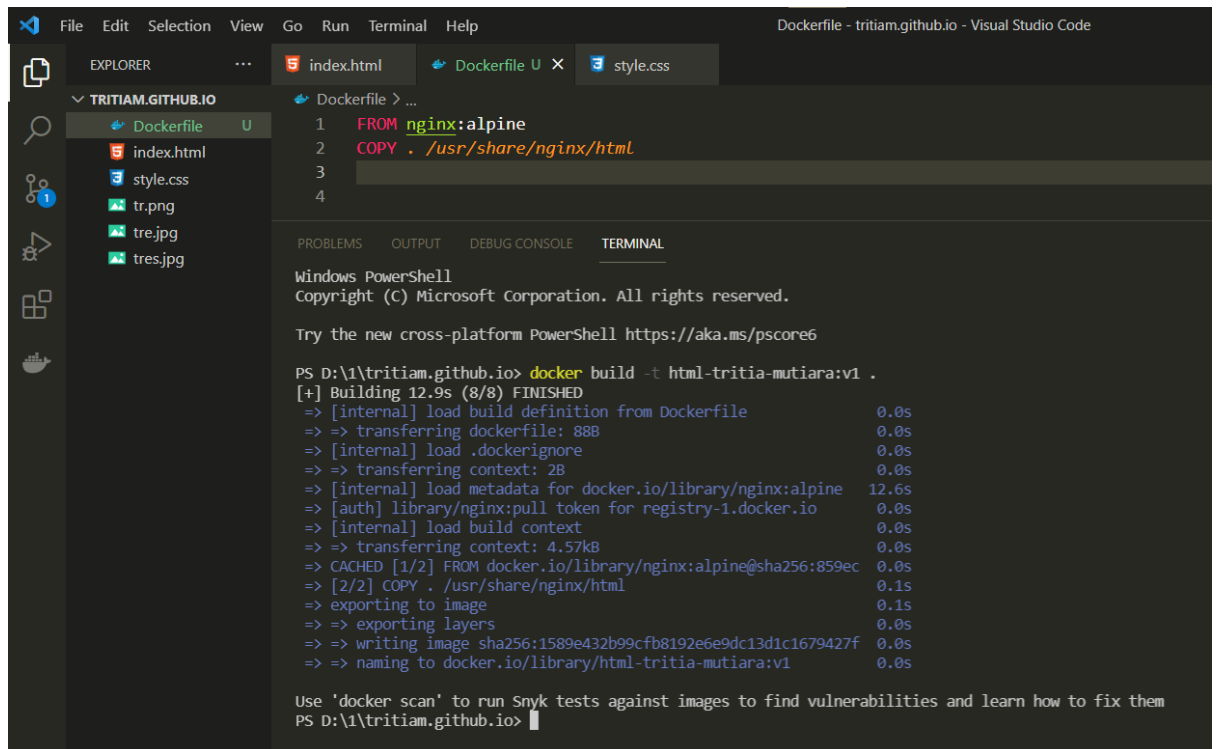
Cara men-deploy website statis dengan menggunakan docker.

1. Disini saya menggunakan sistem operasi windows, jadi kita harus mendownload wsl (windows subsystem for linux) terlebih dahulu disini.
<https://docs.microsoft.com/en-us/windows/wsl/install-win10>
2. Kemudian kita mendownload docker desktop untuk windows disini.
<https://www.docker.com/products/docker-desktop>
3. Melakukan registrasi didocker hub agar kita bisa sign in diaplikasi docker desktop yang sudah kita download.
<https://hub.docker.com/>
4. Pertama kita harus membuat dockerfile dalam folder website statis kita seperti ini.



Code ini mewakili image yang akan kita gunakan sekaligus dengan menyalin file file dalam folder website kita ke container.

5. Kemudian kita akan membangun docker image untuk HTML server dengan menjalankan perintah sebagai berikut dan memasukkan nama imagenya.
`docker build -t html-tritia-mutiara:v1 .`



The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Dockerfile contains the following content:

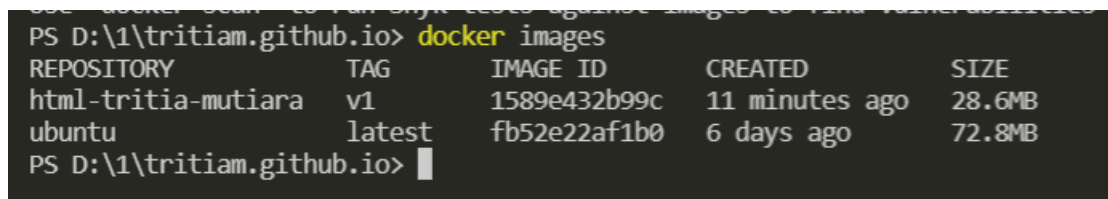
```
1 FROM nginx:alpine
2 COPY . /usr/share/nginx/html
3
4
```

The terminal window shows the output of the command `docker build -t html-tritia-mutiara:v1 .`:

```
PS D:\1\tritiam.github.io> docker build -t html-tritia-mutiara:v1 .
[+] Building 12.9s (8/8) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 888 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/nginx:alpine 12.6s
=> [auth] library/nginx:pull token for registry-1.docker.io 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 4.57kB 0.0s
=> CACHED [1/2] FROM docker.io/library/nginx:alpine@sha256:859ec 0.0s
=> [2/2] COPY . /usr/share/nginx/html 0.1s
=> exporting to image 0.1s
=> => exporting layers 0.0s
=> => writing image sha256:1589e432b99cfb8192e6e9dc13d1c1679427f 0.0s
=> => naming to docker.io/library/html-tritia-mutiara:v1 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
PS D:\1\tritiam.github.io>
```

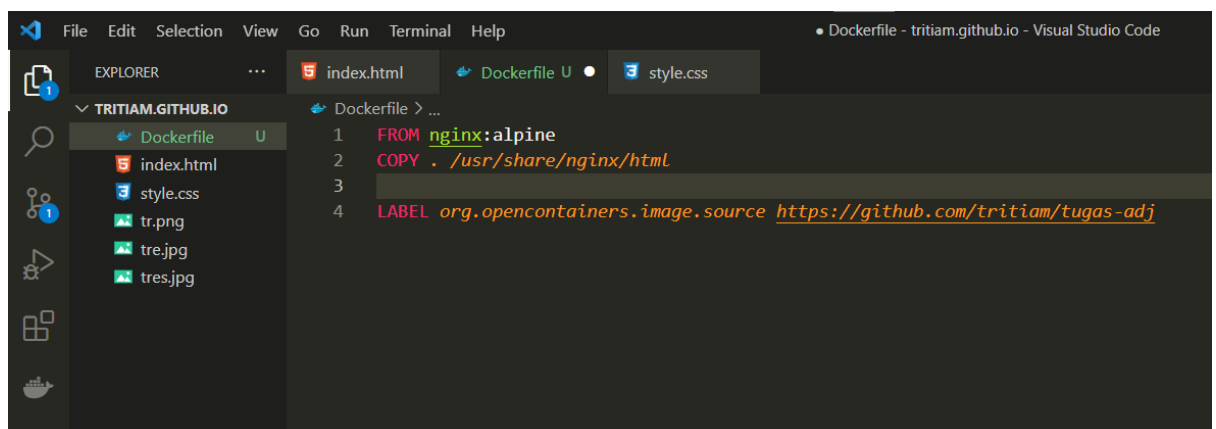
6. Untuk memastikan apakah image berhasil kita buat kita dapat menjalankan perintah berikut.
- docker images



The screenshot shows the output of the `docker images` command in a terminal window:

```
PS D:\1\tritiam.github.io> docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
html-tritia-mutiara v1             1589e432b99c   11 minutes ago 28.6MB
ubuntu              latest         fb52e22af1b0   6 days ago    72.8MB
PS D:\1\tritiam.github.io>
```

7. Untuk ke langkah selanjutnya kita harus membuat repository baru terlebih dahulu digithub untuk menghubungkan docker image kita ke repository digithub kita dan membuat personal access token pada developer settings di pengaturan github.



The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Dockerfile contains the following content:

```
1 FROM nginx:alpine
2 COPY . /usr/share/nginx/html
3
4 LABEL org.opencontainers.image.source https://github.com/tritiam/tugas-adj
```

GitHub Apps

OAuth Apps

Personal access tokens

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

tugas-adj

What's this token for?

Expiration *

30 days The token will expire on Wed, Oct 6 2021

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

- ☒ **repo** Full control of private repositories
 - ☒ repo:status Access commit status
 - ☒ repo_deployment Access deployment status
 - ☒ public_repo Access public repositories
 - ☒ repo:invite Access repository invitations
 - ☒ security_events Read and write security events
- ☐ **workflow** Update GitHub Action workflows
- ☒ **write:packages** Upload packages to GitHub Package Registry
 - ☒ read:packages Download packages from GitHub Package Registry
- ☒ **delete:packages** Delete packages from GitHub Package Registry
- ☐ **admin:org** Full control of orgs and teams, read and write org projects
 - ☐ write:org Read and write org and team membership, read and write org projects

Disini kita hanya menceklis write:packages dan delete:packages saja.

- Kemudian kita akan login dan memasukkan password dari token yang sudah kita buat tadi.

```
PS D:\1\tritiam.github.io> docker login ghcr.io --username tritiam
Password:
Login Succeeded
PS D:\1\tritiam.github.io>
```

Search or jump to...

Pull requests Issues Marketplace Explore

Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

Settings / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

Personal access tokens

Generate new token Revoke all

Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_D1sQXhK4ppMF7hXP9mpUvLC9jY1u44QWjjz Delete

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

- Kemudian kita menjalankan command docker tag seperti berikut dengan memasukkan image ID dari tag v1.
docker tag 1589e432b99c ghcr.io/tritiam/html-tritia-mutiara:v1

```
PS D:\1\tritiam.github.io> docker images
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
html-tritia-mutiara v1          1589e432b99c  28 minutes ago 28.6MB
ubuntu              latest      fb52e22af1b0   6 days ago    72.8MB
```

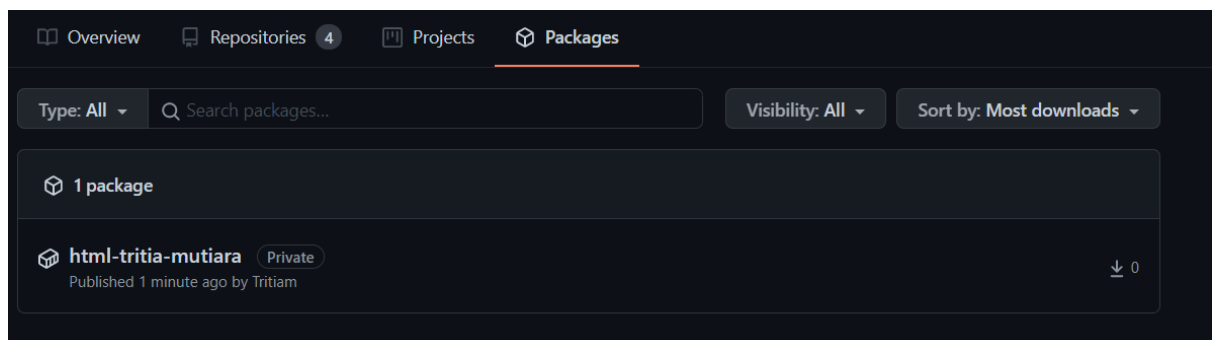
```
PS D:\1\tritiam.github.io> docker tag 1589e432b99c ghcr.io/tritiam/html-tritia-mutiara:v1
```

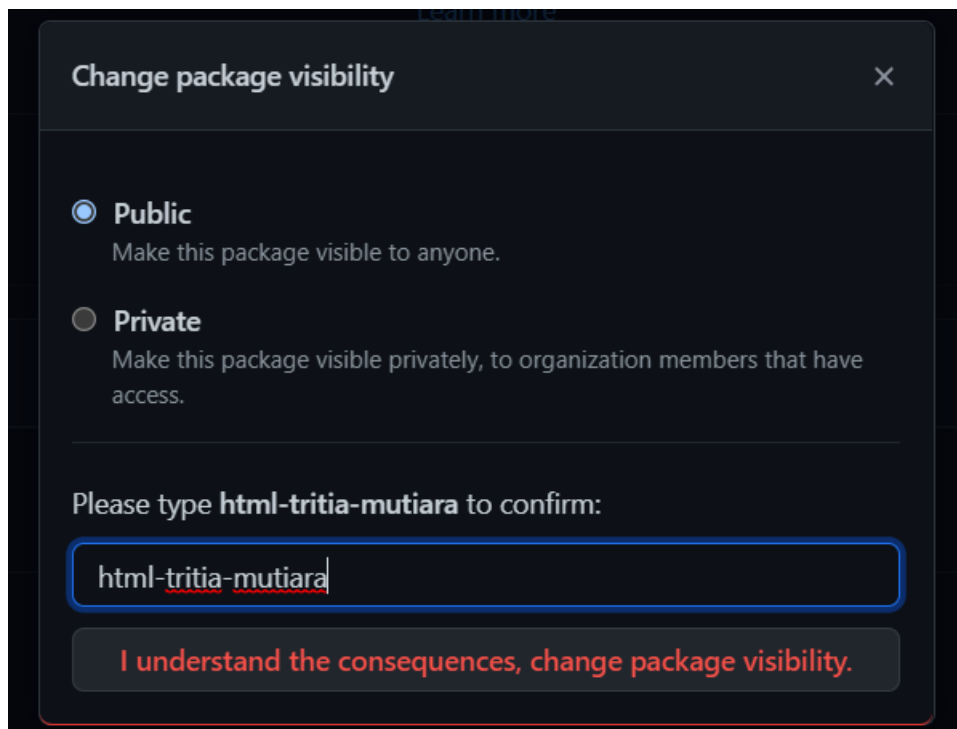
10. Kemudian kita akan memasukkan docker image yang sudah kita buat ke github yang sudah kita buat tadi dengan memasukkan command sebagai berikut.

```
docker push ghcr.io/tritiam/html-tritia-mutiara:v1
```

```
PS D:\1\tritiam.github.io> docker push ghcr.io/tritiam/html-tritia-mutiara:v1
The push refers to repository [ghcr.io/tritiam/html-tritia-mutiara]
f626fc730582: Pushed
45d993692050: Layer already exists
1ea998b95474: Layer already exists
95b99a5c3767: Layer already exists
fc03e3cb8568: Layer already exists
24934e5e6c61: Layer already exists
e2eb06d8af82: Layer already exists
v1: digest: sha256:2bccfe36e100a45d8295baf5ad3622c633c98648106481a2f2720f6288aa72e5 size: 1779
```

11. Image sudah ada dipackages github kita tetapi karena statusnya masih private kita harus mengubahnya terlebih dahulu ke public agar dapat diakses semua orang.





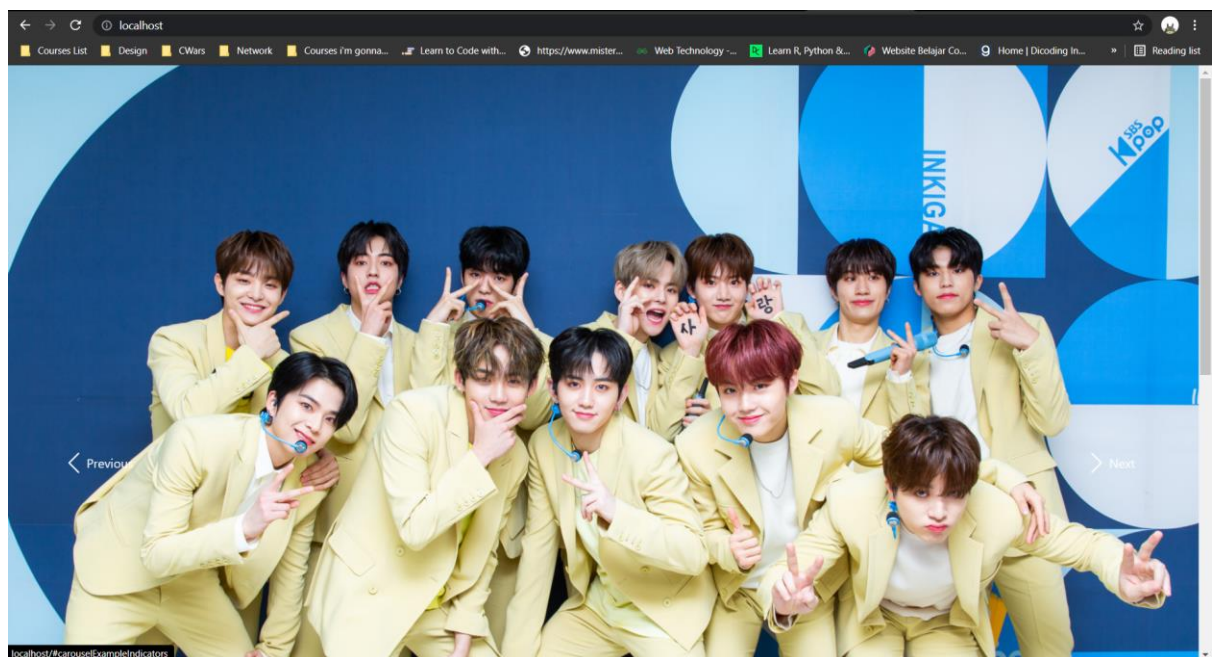
12. Kita akan menjalankan docker kita dengan perintah berikut.

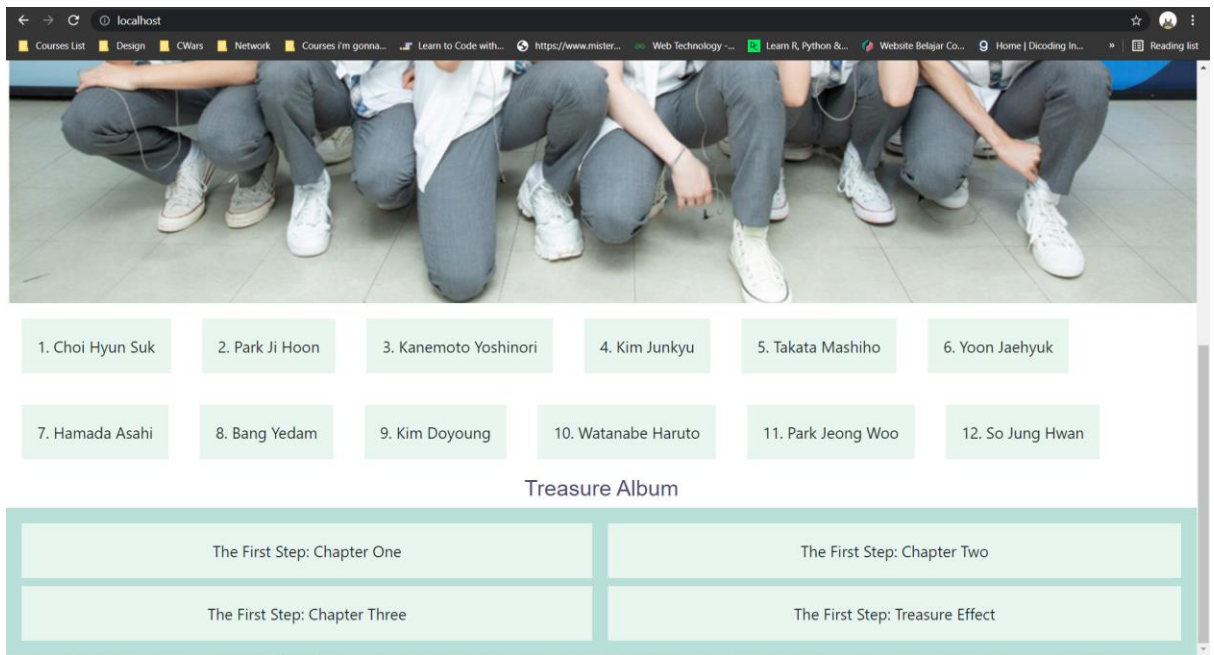
```
docker run -d -p 80:80 html-tritia-mutiara:v1
```

```
PS D:\1\tritiam.github.io> docker run -d -p 80:80 html-tritia-mutiara:v1
c6d5f2f823aa542809507d8d8c4799933ac13dfef02a0ded63ac4a065fc97b40
PS D:\1\tritiam.github.io> |
```

13. Untuk melihat hasilnya kita harus menjalankan perintah sebagai berikut di browser kita.

localhost:80





Berikut adalah hasil dari website yang sudah saya buat.