

LAPORAN PRAKTIKUM VIRTUALISASI KOMPUTER

MENJALANKAN IMAGE PADA CONTAINER DOCKER



Agus Pranata Marpaung

13323033

DIII TEKNOLOGI KOMPUTER

**INSTITUT TEKNOLOGI DEL
FAKULTAS VOKASI**

Judul Praktikum

Minggu/Sesi	:	X/3
Kode Mata Kuliah	:	1032101
Nama Mata Kuliah	:	VIRTUALISASI KOMPUTER
Setoran	:	Jawaban dalam bentuk <i>softcopy</i>
Batas Waktu Setoran	:	4 November 2024 jam 21:30
Tujuan	:	1. Mahasiswa mampu mengetahui apa itu Docker Image Registry Container

Petunjuk

Teori

1. Apa itu Dockerfile?

Dockerfile adalah sebuah file yang digunakan untuk membuat sebuah image. File ini berisi serangkaian perintah dan instruksi.

Berikut ini beberapa perintah Dockerfile yang umum digunakan, yaitu:

a) FROM

Perintah FROM digunakan untuk mendefinisikan sebuah base image yang akan digunakan untuk memulai membangun proses pada setiap docker image apakah itu di repository ataupun di host kita sendiri.

b) ADD

Perintah ADD digunakan untuk mengcopy file dari suatu direktori ke direktori tujuan. Jika direktori asal adalah sebuah URL, perintah ADD akan mendownloadnya dan menempatkannya ke direktori tujuan.

c) CMD

Perintah CMD hampir sama dengan perintah RUN. CMD digunakan untuk mengeksekusi perintah yang lebih spesifik, seperti pada saat proses pembuatan container pada image.

d) ENTRYPOINT

ENTRYPOINT adalah argumen Teori untuk menetapkan default aplikasi yang digunakan setiap sebuah container dibuat menggunakan image.

e) ENV

ENV digunakan untuk menetapkan environment variables.

f) WORKDIR

WORKDIR direktif digunakan untuk mengatur dimana perintah didefinisikan dengan CMD yang akan dieksekusi.

g) RUN

RUN adalah perintah yang digunakan untuk membangun docker images yang terpusat untuk mengeksekusi dockerfiles.

h) MAINTAINER

MAINTAINER adalah perintah yang tidak dijalankan tetapi dideklarasikan sebagai author field dari images.

i) USER

USER direktif digunakan untuk mengatur UID (atau nama pengguna) yang menjalankan sebuah container berdasarkan dari image yang sedang dibangun. Teori

j) VOLUME

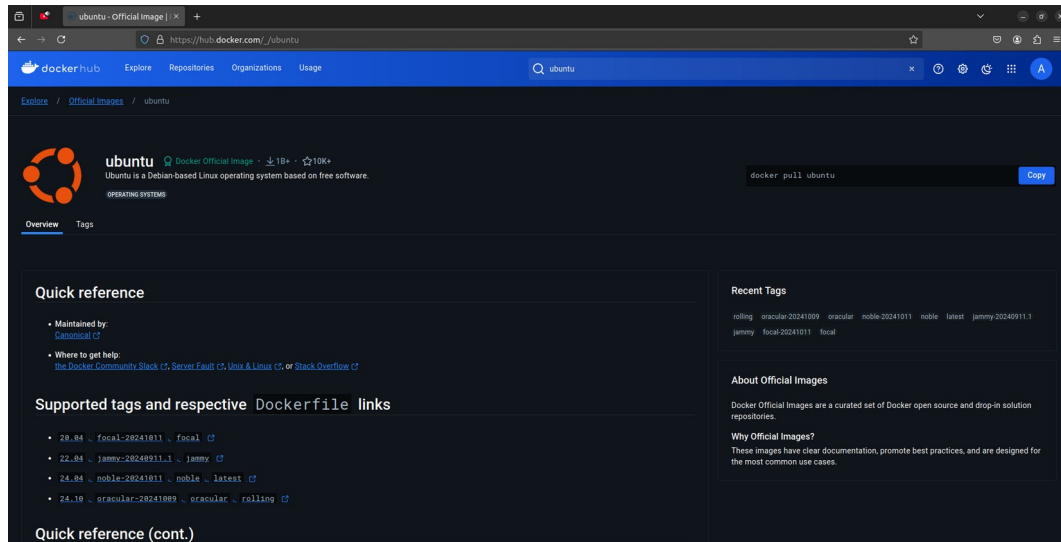
Perintah VOLUME digunakan untuk mengaktifkan akses dari container kita ke direktori pada mesin host.

k) EXPOSE

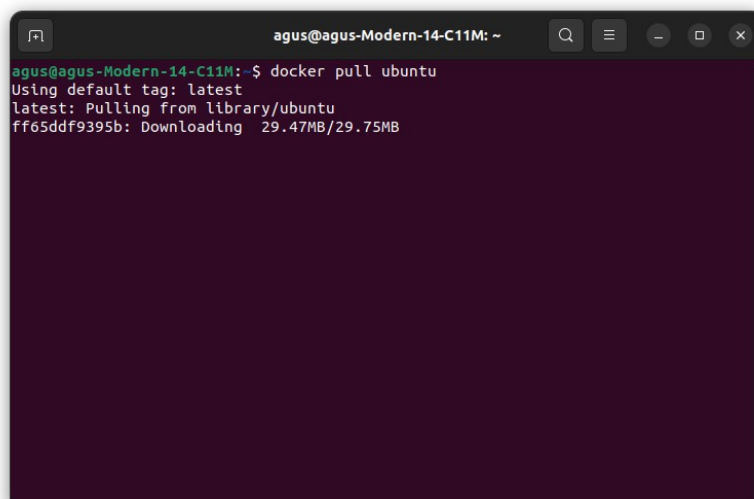
Perintah EXPOSE digunakan untuk menghubungkan port tertentu untuk mengaktifkan network antara proses yang berjalan di dalam container dan mesin host.

Praktikum

1. Sebelum Kita memulai untuk membuat sebuah Dockerfile, Kita terlebih dahulu menentukan environment yang diperlukan dalam membangun suatu image. Disini Kita akan membuat sebuah custom image yang berisi halaman website sederhana yang dijalankan pada web server Nginx menggunakan base image system operasi Ubuntu.



2. Oleh sebab itu, terlebih dahulu Kita membutuhkan image dari system operasi Ubuntu. Kita bisa pergi ke docker-hub untuk melihat versi yang tersedia pada system operasi Ubuntu.



Kita bisa mendownload image Ubuntu dengan versi yang lain, namun Kita bisa mendownload versi 22.04.

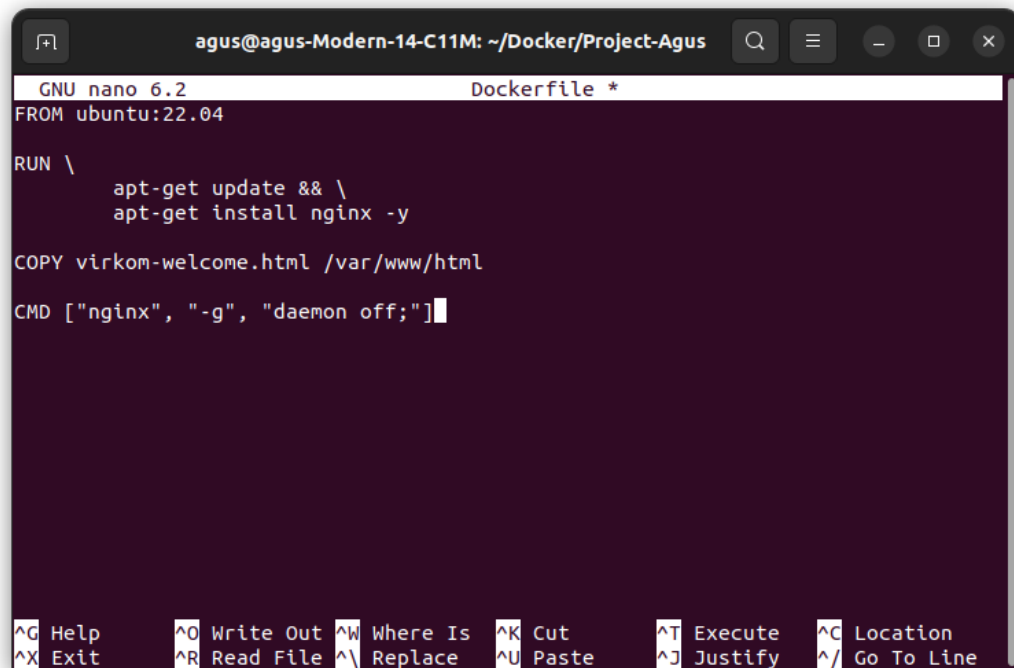
3. Lalu Kita bisa membuat sebuah direktori baru dengan nama yang Kita inginkan sebagai tempat Dockerfile dan juga file dari halaman website sederhana.

```
agus@agus-Modern-14-C11M: ~/Docker/Project-Agus
agus@agus-Modern-14-C11M:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
ff65ddf9395b: Pull complete
Digest: sha256:99c35190e22d294cdace2783ac55effc69d32896daaa265f0bbedbcde4fbe3e5
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
agus@agus-Modern-14-C11M:~$ mkdir -p Docker/Project-Agus
agus@agus-Modern-14-C11M:~$ cd Do
Docker/ Documents/ Downloads/
agus@agus-Modern-14-C11M:~$ cd Docker/Project-Agus/
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$
```

4. Setelah itu, Kita bisa membuat Dockerfile dengan nama Dockerfile pada direktori yang baru Kita buat. Kita bisa mengikuti gambar berikut untuk isi dari Dockerfile nya.
- Membuat directory Docker dan membuat file Docker

```
agus@agus-Modern-14-C11M: ~/Docker/Project-Agus
agus@agus-Modern-14-C11M:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
ff65ddf9395b: Pull complete
Digest: sha256:99c35190e22d294cdace2783ac55effc69d32896daaa265f0bbedbcde4fbe3e5
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
agus@agus-Modern-14-C11M:~$ mkdir -p Docker/Project-Agus
agus@agus-Modern-14-C11M:~$ cd Do
Docker/ Documents/ Downloads/
agus@agus-Modern-14-C11M:~$ cd Docker/Project-Agus/
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ nano Dockerfile
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ nano Dockerfile
```

- Membuat isi dari Dockerfile.



The screenshot shows a terminal window titled "agus@agus-Modern-14-C11M: ~/Docker/Project-Agus". The terminal is running the GNU nano 6.2 editor, editing a file named "Dockerfile *". The content of the Dockerfile is as follows:

```
FROM ubuntu:22.04

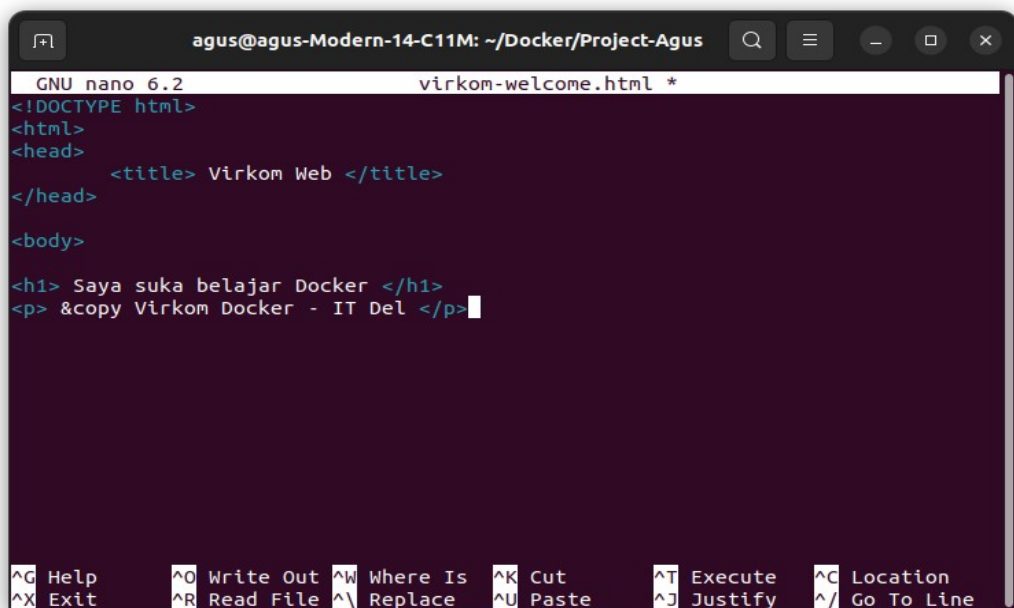
RUN \
    apt-get update && \
    apt-get install nginx -y

COPY virkom-welcome.html /var/www/html

CMD ["nginx", "-g", "daemon off;"]
```

The bottom of the terminal shows the nano editor's help menu with various shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^_ Replace, ^U Paste, ^J Justify, and ^_/ Go To Line.

- Membuat tampilan HTML Web Docker.



The screenshot shows a terminal window titled "agus@agus-Modern-14-C11M: ~/Docker/Project-Agus". The terminal is running the GNU nano 6.2 editor, editing a file named "virkom-welcome.html *". The content of the HTML file is as follows:

```
<!DOCTYPE html>
<html>
<head>
  <title> Virkom Web </title>
</head>
<body>
  <h1> Saya suka belajar Docker </h1>
  <p> &copy; Virkom Docker - IT Del </p>
```

The bottom of the terminal shows the nano editor's help menu with various shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^_ Replace, ^U Paste, ^J Justify, and ^_/ Go To Line.

Keterangan:

a) FROM ubuntu:22.04

Menjadi dasar untuk membangun image Docker Kita.

b) RUN apt-get update && apt-get install nginx -y

Menjalankan perintah untuk memperbarui paket dan menginstal web server Nginx di dalam container

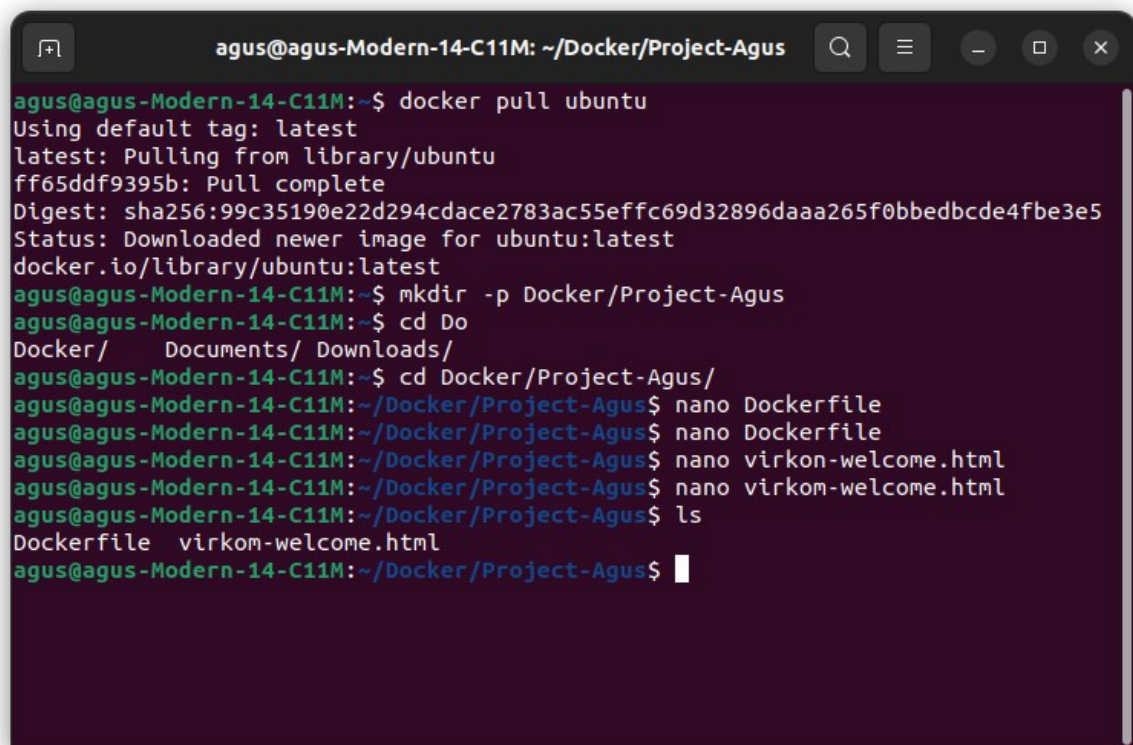
c) COPY virkom-welcome.html /var/www/html

Menyalin file virkom-welcome.html ke direktori /var/www/html dalam container.

d) CMD ["nginx", "-g", "daemon off;"]

Menjalankan server Nginx saat container berjalan.

5. Lalu Kita bisa membuat contoh tampilan website sederhana yang Kita inginkan, dan sesuaikan penamaannya dengan isi Dockerfile.



```
agus@agus-Modern-14-C11M: ~/Docker/Project-Agus
agus@agus-Modern-14-C11M:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
ff65ddf9395b: Pull complete
Digest: sha256:99c35190e22d294cdace2783ac55effc69d32896daaa265f0bbedbcde4fbe3e5
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
agus@agus-Modern-14-C11M:~$ mkdir -p Docker/Project-Agus
agus@agus-Modern-14-C11M:~$ cd Docker/Project-Agus
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ nano Dockerfile
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ nano virkon-welcome.html
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ nano virkom-welcome.html
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ ls
Dockerfile  virkom-welcome.html
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$
```

Terlihat disini bahwa hanya terdapat satu halaman website saja. Kita bisa membuat banyak tampilan website yang lain sesuai yang Kita inginkan.

- Setelah semuanya sudah siap, Kita sudah bisa menjalankan command berikut untuk membuat custom image Kita.

docker build -t image-virkom-web .

```
agus@agus-Modern-14-C11M:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
ff65d0f9395b: Pull complete
Digest: sha256:59c35190e22d94dace2783ac5effcd9d32896daa765f6bbdbcd4efb3e5
Status: Downloaded newer image for ubuntu:latest
agus@agus-Modern-14-C11M:~$ mkdir -p Docker/Project-Agus
agus@agus-Modern-14-C11M:~$ cd Docker/Project-Agus/
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ nano Dockerfile
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ nano Dockerfile
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ nano virkom-welcome.html
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ nano virkom-welcome.html
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ ls
Dockerfile  virkom-welcome.html
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ docker build -t image-virkom-web .
[+] Building 99.8s (6/0) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/ubuntu:22.04
=> [internal] load .dockerignore
=> [internal] transfer context: 2.0s
=> [1/1] FROM docker.io/library/ubuntu:22.04@sha256:8e5a57c2499248a4f1040fcd541e9a45a6b728681a39946c1187203f97
=> resolve docker.io/library/ubuntu:22.04@sha256:8e5a57c2499248a4f1040fcd541e9a45a6b728681a39946c1187203f97
=> sha256:8e5a57c2499248a4f1040fcd541e9a45a6b728681a39946c1187203f97 / 4.0MB / 4.0MB
=> sha256:8e5a57c2499248a4f1040fcd541e9a45a6b728681a39946c1187203f97 / 4.0MB / 4.0MB
=> sha256:8e5a57c2499248a4f1040fcd541e9a45a6b728681a39946c1187203f97 / 4.0MB / 4.0MB
=> sha256:8e5a57c2499248a4f1040fcd541e9a45a6b728681a39946c1187203f97 / 4.0MB / 4.0MB
=> extracting sha256:8e5a57c2499248a4f1040fcd541e9a45a6b728681a39946c1187203f97
=> [internal] load build context
=> [1/1] RUN apt-get update && apt-get install nginx -y
=> [2/1] COPY virkom-welcome.html /var/www/html
=> exporting to image
=> writing image sha256:95aed03909f322106c3915ad905810c051af1c230768d9f4d66407b320aadb12
=> naming to docker.io/library/image-virkom-web
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$
```

Keterangan:

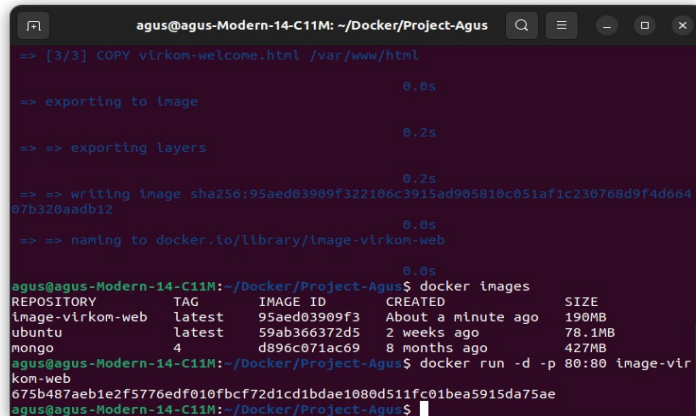
- docker build** : untuk membuat image Docker
- t** : memberikan nama dan tag pada image yang akan Kita buat
- image-virkom-web** : nama image yang Kita inginkan
- .** : lokasi direktori saat ini dari Dockerfile Kita

- Lalu Kita periksa apakah image yang baru Kita buat itu sudah jadi atau belum dengan menjalankan command berikut.

```
agus@agus-Modern-14-C11M:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
image-virkom-web latest 95aed03909f3 About a minute ago 190MB
ubuntu latest 59ab366372d5 2 weeks ago 78.1MB
mongo 4 d896c071ac69 8 months ago 427MB
agus@agus-Modern-14-C11M:~$
```

Terlihat bahwa image baru dengan nama **image-virkom-web** berhasil dibuat.

8. Kemudian Kita bisa mengaktifkan layanan yang ada pada image tersebut kedalam docker container dengan menjalankan command berikut.

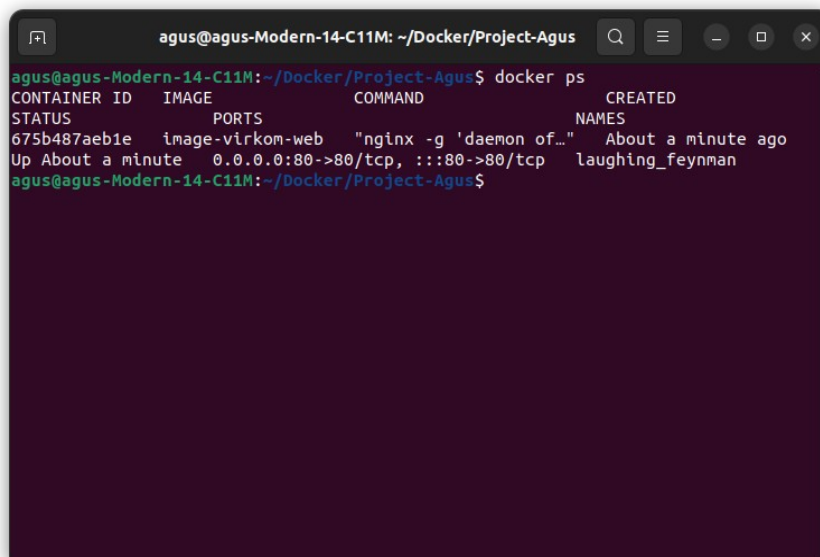


```
agus@agus-Modern-14-C11M: ~/Docker/Project-Agus
=> [3/3] COPY virkom-welcome.html /var/www/html
=> exporting to image 0.0s
=> => exporting layers 0.2s
=> => writing image sha256:95aed03909f322106c3915ad905810c051af1c230768d9f4d664
07b320aadb12 0.0s
=> => naming to docker.io/library/image-virkom-web 0.0s
agus@agus-Modern-14-C11M: ~/Docker/Project-Agus$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
image-virkom-web latest 95aed03909f3 About a minute ago 190MB
ubuntu latest 59ab366372d5 2 weeks ago 78.1MB
mongo 4 d896c071ac69 8 months ago 427MB
agus@agus-Modern-14-C11M: ~/Docker/Project-Agus$ docker run -d -p 80:80 image-vir
kom-web
675b487aeb1e2f5776edf010fbcf72d1cd1bdae1080d511fc01bea5915da75ae
agus@agus-Modern-14-C11M: ~/Docker/Project-Agus$
```

Keterangan:

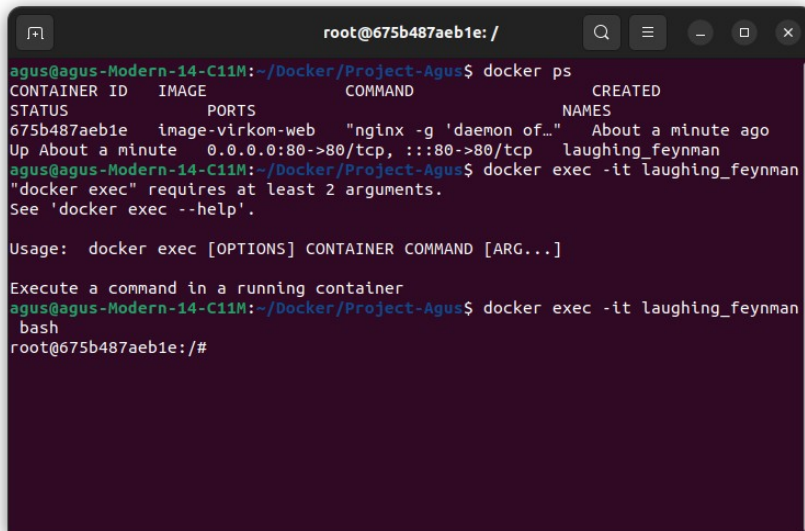
- d : menjalankan container di latar belakang (detached)
- p 80:80 : meneruskan port 80 dari host ke port 80 dalam container

9. Verifikasilah apakah docker container yang baru dijalankan benar-benar berjalan atau tidak dengan menjalankan command berikut.



```
agus@agus-Modern-14-C11M: ~/Docker/Project-Agus$ docker ps
CONTAINER ID IMAGE COMMAND CREATED
STATUS PORTS NAMES
675b487aeb1e image-virkom-web "nginx -g 'daemon of..." About a minute ago
Up About a minute 0.0.0.0:80->80/tcp, :::80->80/tcp laughing_feynman
agus@agus-Modern-14-C11M: ~/Docker/Project-Agus$
```

10. Kemudian Kita mencoba masuk kedalam docker container yang sedang berjalan tersebut dengan menjalankan command berikut.

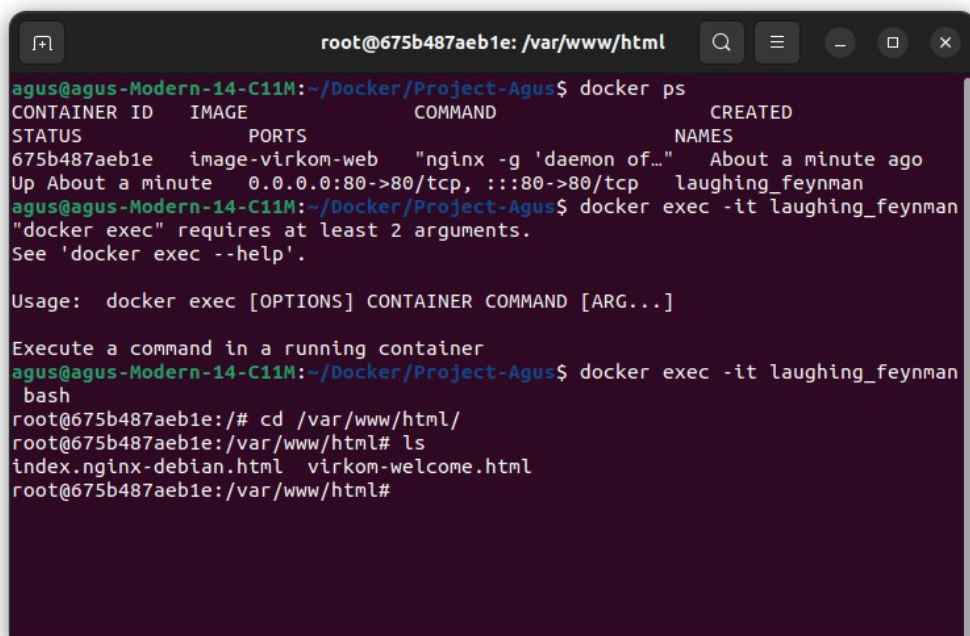


```
root@675b487aeb1e: /
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED
STATUS        PORTS         NAMES
675b487aeb1e   image-virkom-web  "nginx -g 'daemon of..." About a minute ago
Up About a minute   0.0.0.0:80->80/tcp, :::80->80/tcp   laughing_feynman
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ docker exec -it laughing_feynman
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Execute a command in a running container
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ docker exec -it laughing_feynman
bash
root@675b487aeb1e: /#
```

11. Lalu Kita periksalah dimana lokasi file halaman website yang Kita copy dari host ke container.



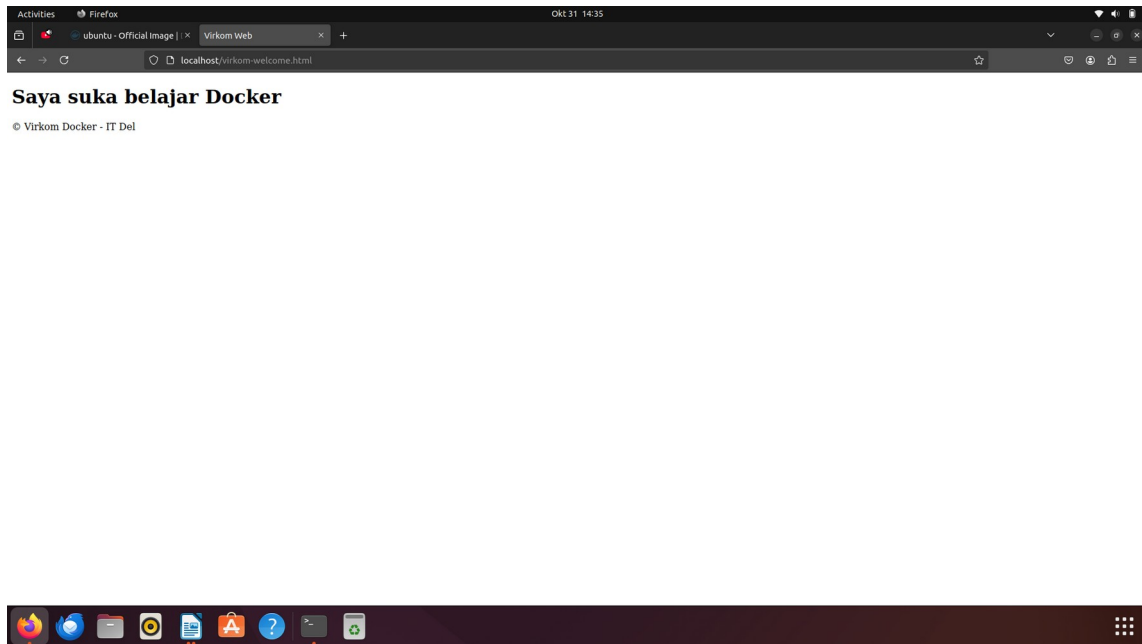
```
root@675b487aeb1e: /var/www/html
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED
STATUS        PORTS         NAMES
675b487aeb1e   image-virkom-web  "nginx -g 'daemon of..." About a minute ago
Up About a minute   0.0.0.0:80->80/tcp, :::80->80/tcp   laughing_feynman
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ docker exec -it laughing_feynman
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Execute a command in a running container
agus@agus-Modern-14-C11M:~/Docker/Project-Agus$ docker exec -it laughing_feynman
bash
root@675b487aeb1e: /# cd /var/www/html/
root@675b487aeb1e: /var/www/html# ls
index.nginx-debian.html  virkom-welcome.html
root@675b487aeb1e: /var/www/html#
```

Dan terlihat file halaman websitenya berada pada direktori /var/www/html.

12. Setelah itu Kita bisa melihat tampilan dari halaman website yang sudah Kita buat sebelumnya melalui web browser dari host dengan mengakses `http://localhost/virkom-welcome.html`.



Note: Sesuaikan dengan port yang Kita gunakan dan juga nama file html nya.

13. Selesai!