

IMPLEMENTASI DOCKER DALAM MEMBANTU VIRTUAL LAB PADA UNIVERSITAS

Rama Aria Megantara¹, Farrikh Alzami, Ricardus Anggi P², Dwi Puji Prabowo³

^{1,2,3}Fakultas Ilmu Komputer, Universitas Dian Nuswantoro Semarang

Gedung H Lantai 1, Kampus 1 Jl. Imam Bonjol No 207, Semarang

E-mail : alzami@dsn.dinus.ac.id¹

Abstrak

Kegiatan mahasiswa praktek untuk penggunaan alat maupun melakukan pemrograman ternyata membutuhkan sumber daya yang cukup besar. Dengan berfokus pada masalah pemrograman, peneliti menggunakan pendekatan pemanfaatan linux dan docker untuk membantu mahasiswa menggunakan sumber daya Perguruan Tinggi sehingga tidak membebani mahasiswa dalam pembelian perangkat baru. Disamping itu, dosen dapat segera mengetahui hasil pekerjaan dari mahasiswa dikarenakan semua tugas pemrograman sudah diletakkan pada sumber daya Server Perguruan Tinggi.

Kata Kunci: Docker, Linux, Server, pembelajaran.

I. PENDAHULUAN

Saat ini, pembelajaran jarak jauh menjadi suatu hal yang wajar dalam kegiatan belajar mengajar di Indonesia. Namun, pembelajaran jarak jauh membuat beberapa hal menjadi cukup menantang, antara lain: 1) pembelajaran menjadi asinkron; 2) mahasiswa tidak dapat menggunakan sumber daya Perguruan Tinggi seperti Lab Pemrograman, dan 3) mahasiswa terpaksa membeli paket data.

Peneliti berfokus pada masalah kedua, yaitu mahasiswa yang tidak bisa menggunakan Lab Pemrograman karena mahasiswa tidak bisa pergi ke kampus. Alasan kenapa dibuat Lab Pemrograman di Perguruan Tinggi adalah menyamaratakan hak dan kewajiban mahasiswa dalam mengikuti kegiatan perkuliahan pemrograman. Namun, dengan kondisi pembelajaran jarak jauh, mahasiswa tidak dapat menggunakan lab dengan semestinya, dan mahasiswa harus mengeluarkan biaya yang besar untuk mengganti computer atau laptopnya. Disamping itu, dosen juga kesulitan dalam mengecek hasil dari pengerjaan program oleh mahasiswa, karena jika hanya mengumpulkan dalam bentuk laporan, maka kemungkinan duplikasi dokumen (mahasiswa mencontoh hasil jadi dari temannya) menjadi cukup besar.

Oleh karena itu, peneliti menawarkan solusi konsep virtualisasi berbasis mesin virtual (virtual machine, VM). Disini server yang ada di Perguruan Tinggi, disetting sedemikian rupa, sehingga mahasiswa dapat mengerjakan tugas pemrograman dengan cukup mengakses lewat browser.

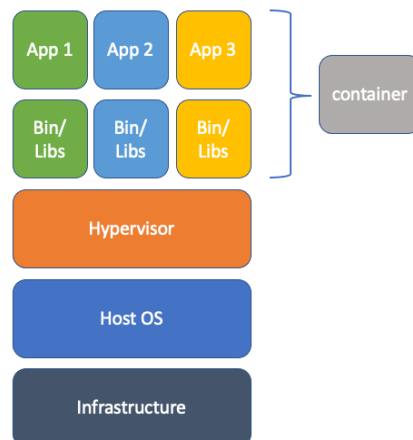
Untuk mendukung kegiatan virtualisasi server, disini peneliti menggunakan docker. Docker container merupakan salah satu aplikasi yang dapat mengelola banyak aplikasi(Wang *et al.*, 2017), kemudian Docker Container sangat cocok untuk desain arsitektur sistem dengan pendekatan microservice, karena masing-masing service memiliki lingkungan yang terisolasi namun tetap dapat berkomunikasi satu sama lain serta dapat meningkatkan efektifitas dalam penggunaan sumber daya CPU dan memori pada server (Khalida *et al.*, 2019) (Endo *et al.*, 2016).

Dengan adanya Docker ini, diharapkan dapat membantu mahasiswa dalam mengerjakan tugas pemrograman tanpa memikirkan penambahan sumber daya baru (computer baru).

II. METODOLOGI PENELITIAN

1. Metodologi Penelitian

Alur penelitian merupakan proses dari pembuatan infrastruktur docker pada server, dapat dilihat pada gambar berikut:



Gambar 1. Infrastruktur Docker yang akan dipakai

Tahapan pada Gambar 1 dapat dijelaskan pada sub-bab sebagai berikut:

a. Setting Infrastruktur

Pertama, disini peneliti menggunakan Infrastruktur berupa koneksi internet keluar masuk Server berupa bandwidth. Kebutuhan bandwidth ini bisa menyesuaikan sesuai kebutuhan dari mahasiswa. Jika banyak mahasiswa menggunakan provider A, maka peneliti akan menambah bandwidth provider A.

Server yang dipakai, menggunakan server yang ada di Perguruan Tinggi tempat peneliti bekerja. Untuk spesifikasi Hardware sebagai berikut: 1) dalam kaitan prosesor, peneliti memakai Intel Xeon 5 sebesar 56 Thread dikarenakan cukup untuk handle proses komputasi secara bersamaan oleh 100 pengguna. 2) dalam kaitan RAM, peneliti menggunakan 64GB RAM. 3) peneliti juga memasang VGA yang dapat digunakan untuk kegiatan Deep Learning dengan ukuran 10GB.

b. Setting Host OS & Hypervisor

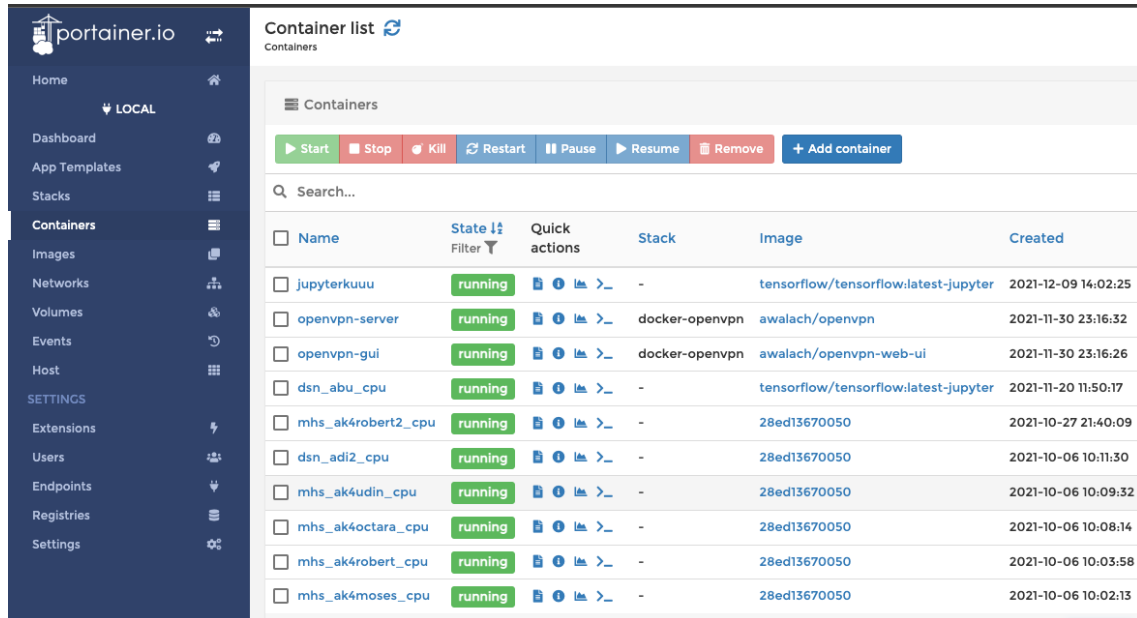
Untuk kebutuhan system operasi, peneliti menggunakan system operasi Linux pada Server. Tujuannya adalah keamanan dan kemudahan dalam manajemen Docker dibandingkan dengan Sistem Operasi lainnya. Linux yang peneliti pakai hanya menggunakan basis text. Ini dikarenakan agar peneliti dapat secara cepat mengakses system operasi dengan bandwidth minimal melalui terminal (DOS Prompt). Hypervisor merupakan hal optional pada mekanisme Docker jika menggunakan Linux. Hal ini dikarenakan jika docker berbasis linux, maka Docker dapat dijalankan pada node VM atau Baremetal untuk menjalankan aplikasi secara terisolasi pada node tersebut.

c. Container

Container merupakan unit standar perangkat lunak yang mengemas kode dan semua dependensinya sehingga aplikasi berjalan dengan cepat dan andal dari satu lingkungan komputasi ke lingkungan komputasi lainnya. Container Docker image adalah paket perangkat lunak yang ringan, mandiri, dan dapat dieksekusi yang mencakup semua yang diperlukan untuk menjalankan aplikasi: kode, runtime, alat sistem, pustaka sistem, dan pengaturan. Disini, container dapat dibuat sendiri sesuai kebutuhan, atau bisa mengambil dari marketplace seperti <https://hub.docker.com/> ataupun dari <https://ngc.nvidia.com/>

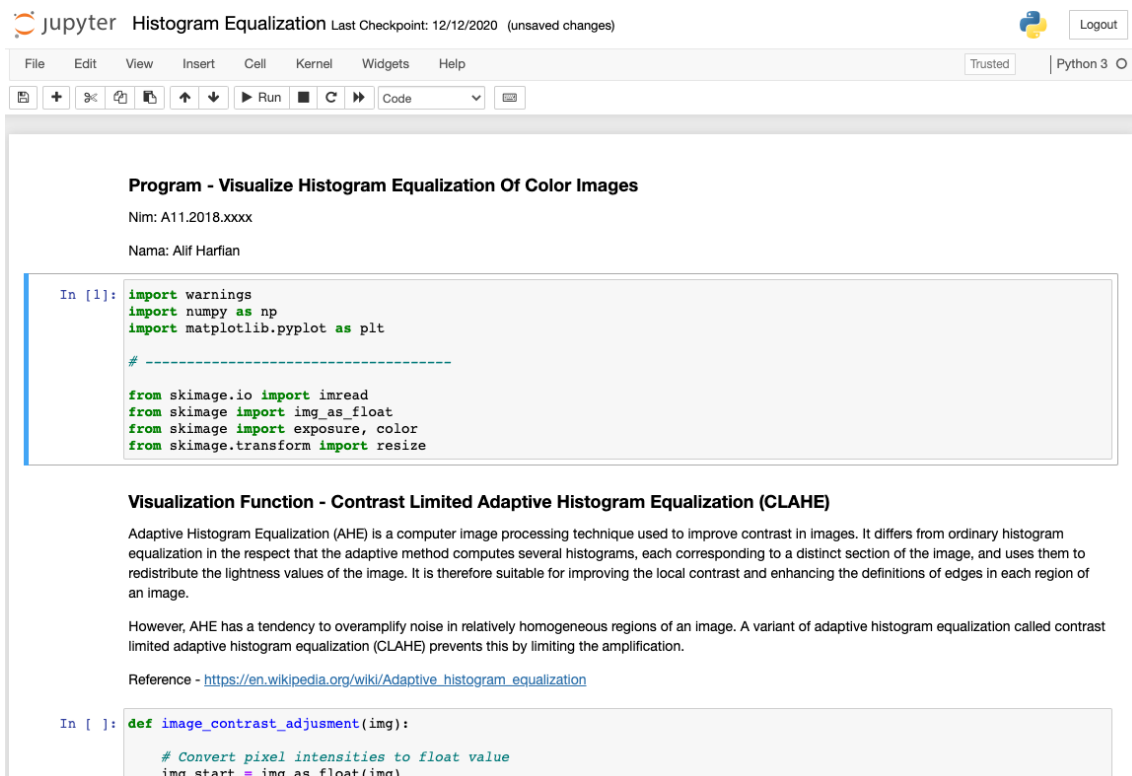
III. HASIL DAN PEMBAHASAN

Container yang peneliti gunakan dalam eksperimen ini, adalah container untuk machine learning dan data mining. Untuk mempermudah manajemen docker dan container, peneliti menggunakan Portainer. Sehingga jika ada kendala pada container, maka peneliti dapat dengan cepat untuk memperbaikinya.



Gambar 2 Manajemen Container menggunakan Portainer

Setelah mahasiswa diberikan akses ke container sesuai dengan deskripsi mata kuliah, mahasiswa mulai mengerjakan tugas dan praktek melalui browser. Dikarenakan menggunakan browser, maka mahasiswa dapat mengerjakan tugas menggunakan smartphone ataupun alat lain yang dapat terhubung dengan internet. Setelah mahasiswa mengerjakan, Dosen dapat dengan mudah mengecek hasil pengerjaan mahasiswa tersebut (bisa dilihat pada gambar 3).



Program - Visualize Histogram Equalization Of Color Images

Nim: A11.2018.xxxx
Nama: Alif Harfian

```
In [1]: import warnings
import numpy as np
import matplotlib.pyplot as plt

# -----

from skimage.io import imread
from skimage import img_as_float
from skimage import exposure, color
from skimage.transform import resize
```

Visualization Function - Contrast Limited Adaptive Histogram Equalization (CLAHE)

Adaptive Histogram Equalization (AHE) is a computer image processing technique used to improve contrast in images. It differs from ordinary histogram equalization in the respect that the adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. It is therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image.

However, AHE has a tendency to overamplify noise in relatively homogeneous regions of an image. A variant of adaptive histogram equalization called contrast limited adaptive histogram equalization (CLAHE) prevents this by limiting the amplification.

Reference - https://en.wikipedia.org/wiki/Adaptive_histogram_equalization

```
In [ ]: def image_contrast_adjusment(img):
# Convert pixel intensities to float value
img_start = img_as_float(img)
```

Gambar 3. Contoh Pengerjaan Tugas Pemrograman

IV. KESIMPULAN

Infrastruktur Docker dapat membantu mahasiswa dalam proses belajar praktikum Bahasa pemrograman dan membantu dosen dalam menilai hasil penugasan. Disamping itu, Docker dapat membantu memaksimalkan sumber daya Server dan dapat diakses secara cepat serta dapat diperbaiki dengan mudah. Kedepan, peneliti akan menggunakan kubernetes agar dapat menggabungkan beberapa server menjadi satu klaster.

V. UCAPAN TERIMA KASIH

Penelitian ini didanai oleh LPPM Udinus pada Skema Penelitian IPTEKS.

VI. REFERENSI

- [1] Endo, P. T., Rodrigues, M., Gonçalves, G. E., Kelner, J., Sadok, D. H., & Curescu, C. (2016). High availability in clouds: systematic review and research challenges. *Journal of Cloud Computing*, 5(1), 16. <https://doi.org/10.1186/s13677-016-0066-8>
- [2] Khalida, R., Muhajirin, A., & Setiawati, S. (2019). Teknis Kerja Docker Container untuk Optimalisasi Penyebaran Aplikasi. *PIKSEL : Penelitian Ilmu Komputer Sistem Embedded and Logic*, 7(2), 167–176. <https://doi.org/10.33558/piksel.v7i2.1819>
- [3] Wang, B., Song, Y., Cui, X., & Cao, J. (2017). Performance comparison between hypervisor- and container-based virtualizations for cloud users. *2017 4th International Conference on Systems and Informatics (ICSAI)*, 684–689. <https://doi.org/10.1109/ICSAI.2017.8248375>