"*THE PLASTIC HUNTER SENSORIC*"

APLIKASI PENDETEKSI SAMPAH PLASTIK BERBASIS KECERADASAN BUATAN

Fahmi Pradana S. W., Moch Yusuf Faisal A.A., Muhammad Nafi Udin

Ardian Yusuf Wicaksono

Rekayasa Perangkat Lunak,Fakultas Teknologi Informasi dan Industri

Institut Teknologi Telkom Surabaya

Abstrak-Indonesia merupakan negara kepulauan terbesar dan memiliki area laut yang cukup luas (maritim) dan memiliki area pantai yang luas. Dari hal tersebut banyak sekali permasalahan yang dialami oleh negara kita, salah satu permasalahan yaitu pembuangan sampah sembarangan di area pesisir. Sampah tersebut kebanyakan berupa sampah plastik yang dapat mengganggu ekosistem laut karena sulit diurai. Pada penilitian ini kami akan mengembangkan aplikasi untuk menyelesaikan masalah tersebut dengan cara memanfaatkan teknologi *Artificial Intelegence*. Aplikasi tersebut bertujuan untuk mendeteksi sampah plastik di pesisir laut. Aplikasi ini bernama bernama *The Plastic Hunter'19* yaitu aplikasi yang mendeteksi sampah plastik atau bukan dengan memanfaatkan pengolahan citra. Berhubung tingkat volume sampah di Indonesia semakin banyak, maka kami dengan inovasi tersebut ingin membantu mengurangi volume sampah yang ada dan berharap akan cepat terealisasi demi terciptanya kondisi lingkungan sehat danbersih.

Kata Kunci: aplikasi, AI, pengelolah citra, pesisir, sampah plastik

Sub Tema: Pemanfaatan Potensi Teknologi di Wilayah Pesisir dan Laut

I. PENDAHULUAN

1. Latar Belakang

Indonesia merupakan negara kepulauan terbesar dan memiliki area laut yang cukup luas (maritim) dan memiliki area pantai yangluas. Dari hal tersebut banyak sekali permasalahan yang dialami oleh negara kita, salah satu permasalahan yaitu pembuangan sampah sembarangan di area pesisir. Sampah tersebut kebanyakan berupa sampah plastik yang dapat mengganggu ekosistem laut karena sulit diurai(Adharsyah, 2019).

Seperti kita ketahui plastik merupakan salah satu masalah yang sedang dihadapi warga Indonesia. Maka dari itu kami berinisiatif membuat sebuah alat yang bisa membantu melestarikan Laut Indonesia lagi seperti dulu kala dimana biota laut masih bisa hidup tanpa adanya pecemaran sampah. Biota Laut Indonesia sudah terkenal dengan berbagai sumber daya yang sangat beragam mulai dari terumbu karang, rumput laut, penyu, dan berbagai jenis ikan hias maupun ikan konsumsi.

Indonesia harus memiliki sebuah terobosan dari pemuda pemudi yang inovatf dalam menanggapi hal yang serius seperti ini. Dengan itu kami akan ikut serta dalam membangun Indonesia lebih maju dan lebih baik. Kami akan membuat sebuah "**Aplikasi Pendeteksi Sampah Plastik Berbasis Kecerdasan Buatan"** dan memberi nama ***The Plastic Hunter Sensoric.*** Kami berharap dalam pembuatan aplikasi ini bisa membantu dalam membersihkan pencemaran sampah plastik di lautan Indonesia dan mengurangi volume sampah plastik yang bertebaran di pesisir pantai.

2. PERMASALAHAN

Adapun beberapa permasalahan/kendala yang dihadapi dalam pembuatan aplikasi ini yaitu, sebagai berikut:

a. KURANGNYA DATA YANG VALID

Data merupakan sumber penting untuk mempermudah kami dalam membuat sebuah aplikasi ini. Namun, data yang kami dapat masih kurang untuk dapat merealisasikan aplikasi kami.

b. BANYAKNYA JENIS SAMPAH KEMASAN PLASTIK

Data yang kami bukan hanya data kuantitatif tetapi, ada juga data kualitatif yang dimana berupa foto sampah plastik. Foto sampah ini memerlukan foto kualitas yang bagus dan memori penyimpanan yang besar untuk menyimpannya.

3. TUJUAN DAN MANFAAT

Tujuannya untuk menjadi mahasiswa yang tidak diam saja melihat negaranya rusak akibat kelalaian beberapa orang .Kami melakukannya dengan cara mengurangi jumlah sampah yang ada di daerah pesisir/laut dengan cara pemilahan sampah yang berupa sampah plastik yang ada di sekitar laut menggunakan aplikasi pendeteksi sampah. Maanfaat dari ide kita adalah untuk bisa menjaga ekosistem laut dan juga membantu menjaga kebersihan sekitar laut dan pesisir.

4. HIPOTESIS

Menurut dugaan jumlah sampah dipesisir pantai atau bahkan di laut Indonesia itu sangatlah banyak, jadi kami disini akan meneliti apakah itu benar adanya terutama untuk sampah plastiknya yang notabenya sulit terurai, jika benar kami akan mengembangkan Sensor berbasis kecerdasan buatan yang akan memilah sampah itu.

5. RANCANGAN PENELITIAN

Rancangan penelitian kami untuk langkah awalnya akan mengumpulkan data-data tentang sampah yang berupa foto-foto yang kemudian akan di lebeling. Setelah data terkumpul kami akan melanjutkan untuk mengembangkan Sensorik berbasis Kecerdasan Buatan yang hanya akan memilah sampah plastik di lingkungan pesisir pantai/laut.

6. TINJAUAN PUSTAKA

6.1. Robot

Robot adalah seperangkat alat mekanik yang bisa melakukan tugas fisik, baik dengan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu. Istilah robot berawal bahasa Ceko “robota” yang berarti pekerja atau kuli yang tidak mengenal lelah atau bosan(‘Steganografi - Wikipedia bahasa Indonesia, ensiklopedia bebas’, no date).

6.2. SAMPAH

Sampah merupakan material sisa yang tidak diinginkan setelah berakhirnya suatu proses. Sampah didefinisikan oleh manusia menurut derajat keterpakaiannya, dalam proses-proses alam sebenarnya tidak ada konsep sampah, yang ada hanya produk-produk yang dihasilkan setelah dan selama proses alam tersebut berlangsung. Akan tetapi karena dalam kehidupan manusia didefinisikan konsep lingkungan maka sampah dapat dibagi menurut jenis- jenisnya

6.3. PLASTIK

Plastik mencakup produk polimerisasi sintetik atau semi-sintetik. Mereka terbentuk dari kondensasi organik atau penambahan polimer dan bisa juga terdiri dari zat lain untuk meningkatkan performa atau ekonomi. Ada beberapa polimer alami yang termasuk plastik. Plastik dapat dibentuk menjadi film atau fiber sintetik. Nama ini berasal dari fakta bahwa banyak dari mereka "malleable", memiliki properti keplastikan. Plastik didesain dengan variasi yang sangat banyak dalam properti yang dapat menoleransi panas, keras, "reliency" dan lain-lain. Digabungkan dengan kemampuan adaptasinya, komposisi yang umum dan beratnya yang ringan memastikan plastik digunakan hampir di seluruh bidang industri(Adharsyah, 2019).

6.4. KAMERA

Kamera adalah sebuah alat yang di gunakan dalam kegiatan fotografi, kamera digunakan untuk membentuk atau merekam suatu bayangan ke dalam film / memory card. Sebagai fotografer, kamera merupakan alat terpenting di dunia fotografi. Jika anda ingin menjadi fotografer, yuk kita melihat lebih jauh tentang seluk beluk kamera.

6.5. PENGOLAHAN CITRA

Pengolahan citra merupakan suatu metode atau teknik yang dapat digunakan untuk memproses citra atau gambar dengan jalan mjemanipulasinjya menjadi suatu data gambar yang diisikan untuk mendapatkan suatu informasi tertentu mengenai obyek yang sedang diamati.

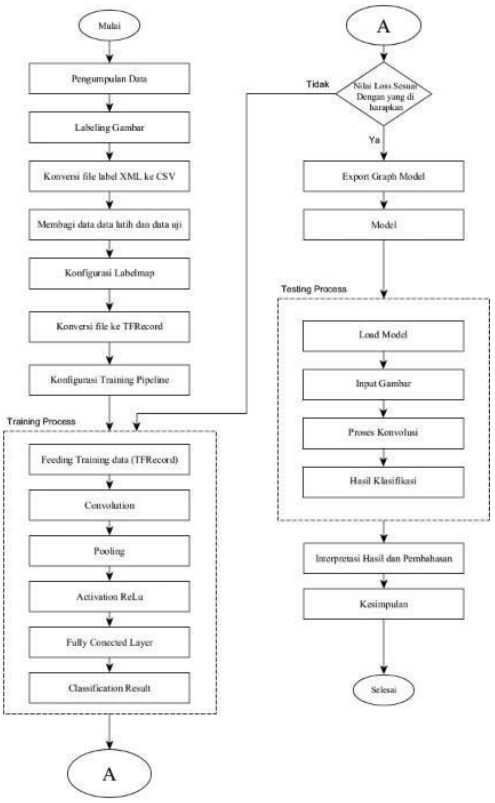
6.6. KECERDASAN BUATAN

Kecerdasan buatan (AI) memungkinkan mesin untuk belajar dari pengalaman, menyesuaikan input- input baru dan melaksanakan tugas seperti manusia. Sebagian besar contoh AI yang Anda dengar dewasa ini – mulai dari komputer yang bermain catur hingga mobil yang mengendarai sendiri – sangat mengandalkan pembelajaran mendalam dan pemrosesan bahasa alamiah. Dengan menggunakan teknologi ini, komputer dapat dilatih untuk menyelesaikan tugas-tugas tertentu dengan memproses sejumlah besar data dan mengenali pola dalam data.

6.7. Metode CNN (*Convolutional Neural Network*)

Metode CNN adalah salah satu pengembangan yang lebih lanjut dari metode MLP yang dikarenakan menggunakan metode yang mirip dengan dimensi yang lebih banyak. Di dalam metode CNN, input layer yang digunakan sebelumnya bukanlah bentuk 1 dimensi melainkan bentuk dari dua dimensi. Apabila dianalogikan dengan fitur-fitur dari wajah manusia, layer pertama adaklah

penggambaran goresan yang berbeda arah yang mana pada layer kedua fitur tersebut seperti bentuk mata, hidung, dan mulut mulai terlihat, proses tersebut dikarenakan penggabungan dari layer pertama yang masih berupa goresan-goresan, di dalam layer ketiga akan terbentuk kombinasi fitur-fitur mata hidung, dan mulut yang mana nantinya akan disimpulkan dengan wajah orang tertentu bahkan dapat kemungkina sudah dapat diidentifikasi hasilnya.



6.8. OBJECT DETECTION DENGAN TENSORFLOW-API

Dalam suatu penelitian diperlukan dukungan hasil-hasil penelitian yang telah ada sebelumnya yang berkaitan dengan penelitian tersebut. Dari penelitian Hafizhan Aliady Aifif (2018) telah dibuatnya pendeteksi/mendeteksi obyek menggunakan data yang sudah dia buat sendiri. Dia menggunakan data foto dari makanan ringan yang tersedia di minimarket. Untuk data yang digunakan itu totalnya ada 1500an gambar semuanya dia ambil menggunakan kamera HP, kemudian foto dikumpulkan dalam satu folder kemudian di lebeling setiap foldernya.

II. METODE

1. Penjelasan

Dalam pengaplikasiannya kami menggunakan *Object Detection with Tensorflow-API* untuk metodenya. *Object Detection* sendiri merupakan teknologi komputer yang terkait dengan pemrosesan

gambar yang berhubungan dengan mendeteksi ojek dari kategori tertentu didalam sebuah gambar atau video (‘Artificial Intelligence/Kecerdasan Buatan – Apa itu dan mengapa hal itu penting | SAS’, no date).

2. Rincian Alur Kerja

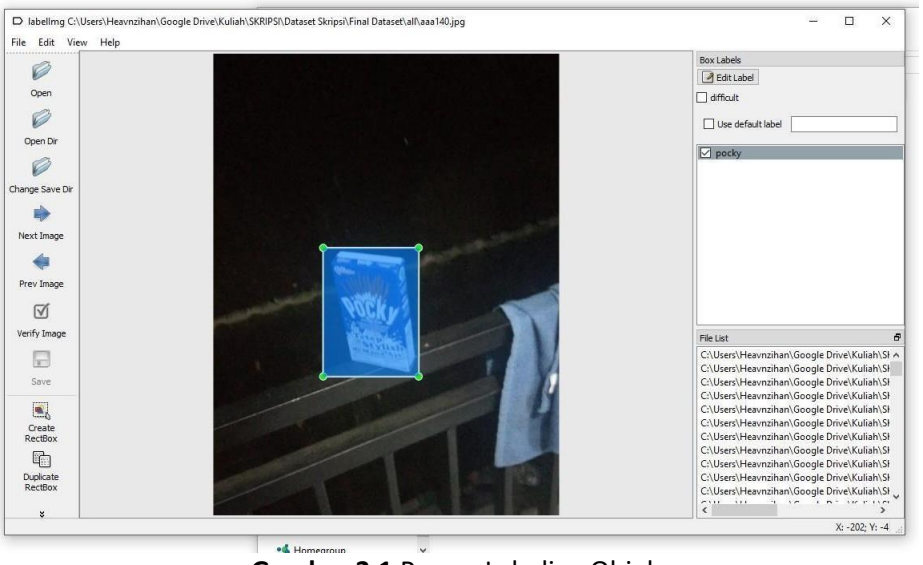
1. Langkah Awal

Siapkan data-data berupa foto yang dibutuhkan seperti berbagai jenis sampah plastic yang akan di definisikan untuk labelling.

2. LANGKAH LABELING

Labeling menggunakan pyhton, install modul Labelimg di pyhton, kemudian lakukan labelling foto/gambarnya satu persatu. Setelah selesai semua file labelnya berupa xml. Kemudian akan muncul struktur foldernya seperti ini :

```
+gambar
| +anotasi
| | -img1.xml
| | -img2.xml
| | -...
| | -imgn.xml
| -img1.jpg
| -img2.jpg
| -...
| -imgn.jpg
```



Gambar 2.1 Proses Labeling Objek

3. MENGUBAH XML MENJADI CSV

Pada tahap ini kita mengubah XML menjadi CSV agar bisa kumpulkan menjadi 1 file yang kemudia di gunakan untuk mengabil data gambar dari box/label yang telah di berikan.

```
import os
import glob
import pandas as pd
import xml.etree.ElementTree as ETdef xml_to_csv(path):
    xml_list = []
    for xml_file in glob.glob(path + '/*.xml'):
        tree = ET.parse(xml_file)
        root = tree.getroot()
        for member in root.findall('object'):
            value = (root.find('filename').text,
                     int(root.find('size')[0].text),
                     int(root.find('size')[1].text),
                     member[0].text,
                     int(member[4][0].text),
                     int(member[4][1].text),
                     int(member[4][2].text),
                    int(member[4][3].text)
                                )
            xml_list.append(value)
    column_name = ['filename', 'width', 'height', 'class', 'xmin', 'ymin', 'xmax', 'ymax']
    xml_df = pd.DataFrame(xml_list, columns=column_name)
    return xml_dfdef main():
    image_path= os.path.join(os.getcwd(),'images/anotation') #namafolder
    xml_df = xml_to_csv(image_path)
```

4. MEMBUAT DATASET (MEMISAHKAN) TRAIN DANTEST

Setelah membuat dataset csv keseluruhanya, nanti langsung run script **split_train_test.py.** Nanti datanya langsung terpisah menjadi 80:20 untuk train testnya.

train.csv

test.csv

5. MEMBUAT TFRECORD

Di bagian ini tugas untuk mengupulkan dataset hamper selesai, jika sudah melakukan pembuatan data train dan test dalam bentuk CSV maka selanjutnya adalah merubah csv menjadi dataset yang di baca oleh tensorflow. Yaitu *TFRECORD.*

6. MENGATUR FILE CONFIG

Mengatur file config, dimana config ini akan di gunakan untuk melakukan konfigurasi dari model training. Config ini menggunakan model SSD mobile Pet v1 dimana ini di sediakan oleh *tensorflow* itu sendiri.

```
model {
ssd {
num_classes: 2 //Total Objek
box_coder {
faster_rcnn_box_coder {
y_scale: 10.0
x_scale: 10.0
height_scale: 5.0
```

7. MENGATUR DAFTAR OBJEK YANG DIGUNAKAN

Memberikan label pada masing-masing objek yang telah memiliki data berupa foto.

```
item { id: 1
name: 'lays'
}
item { id: 2
name: 'goodtime'
}
item { id: 3
name: 'pocky'
```

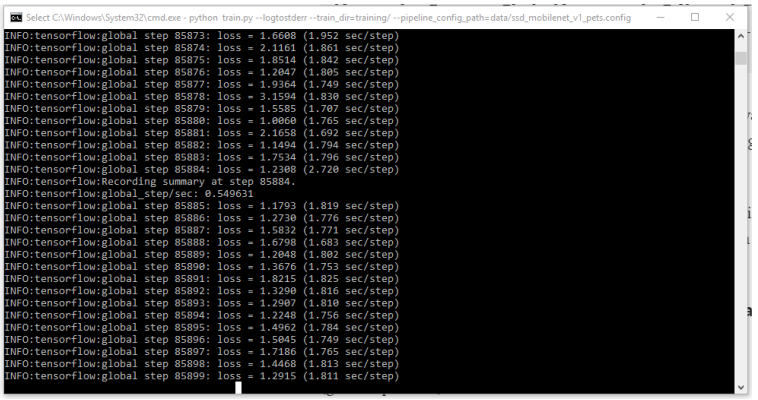
Mengatur Tata Letak File Dan Folder

Susunan folder seperti dibawah ini :

```
+data
| - ssd_mobilenet_v1_pet.config
| - object_detection.pbtext
| - train.tfrecord
| - test.tfrecord
```

8. Proses Training Model

Bagian ini adalah saat-saat penentuan, apakah sebuah PC yang di gunakan bisa untuk melakukan training object detection nya *tensorflow*. pada proses ini memakan waktu yang sangat lama jika menggunakan PC/Laptop yang biasa-biasa saja tanpa dukungan GPU.

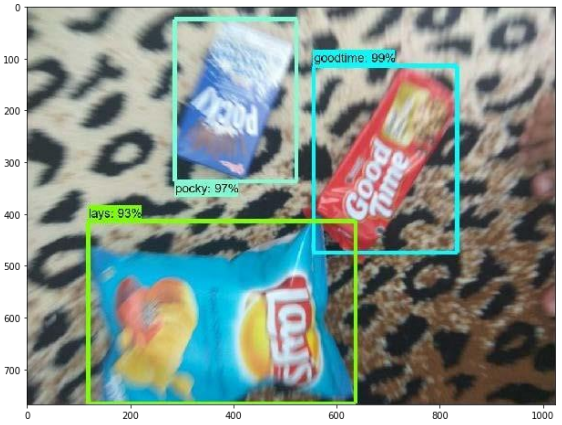


Gambar 2.2 Proses Training Model

9. Ekstrak Model Hasil Training

Meng-ekstrakmodel menjadi *frozeninference graph* sehingga bisa digunakan untuk memprediksi sebuah gambar yang kita sediakan.

10. PENGUJIAN MODEL



Gambar 2.3 Proses Pengujian Objek

III. HASIL DAN PEMBAHASAN

The Plastic Hunter Sensoric merupakan alat untuk mendeteksi sampah yang yang ada di pesisir pantai terutama sampah plastik. Dengan menggunakan metode *Convolutional Neural Network (CNN)* untuk mengidentifikasi objeknya. Selain mendeteksi sampah, kedepan akan digabungkan dengan robot untuk menciptakan teknologi robot pengambil sampah otomatis.Dengan melakukan pengembangan diharapkan nantinya alat ini bisa memudahkan masyarakat untuk membersihkan area pesisir pantai yang kebanyakan terdapat banyak sampah plastik.

IV. KESIMPULAN

The Plastic Hunter Sensoric diciptakan untuk mengembalikan kelestarian laut Indonesia yang tercemar oleh sampah-sampah sukar terurai. Namun, meskipun alat ini telah diciptakan apabila masih ada beberapa masyarakat kurang peduli dengan lingkungan maka sia-sia sajalah hal ini. Kita butuh sebuah pergerakan yang besar dari peran masyarakat dan pemerintahan untuk merealiasikan negara yang bersih dan lestari.

V. DAFTAR PUSTAKA

1. Adharsyah, T. (2019) 'Sebegini Parah Ternyata Masalah Sampah Plastik di Indonesia', 21 Juli, p. 1. Available at: https://www.cnbcindonesia.com/lifestyle/20190721140139-3386420/sebegini-parah- ternyata-masalah-sampah-plastik-di-indonesia .
2. 'Artificial Intelligence/Kecerdasan Buatan – Apa itu dan mengapa hal itu penting | SAS' (no date). Available at: https://www.sas.com/id_id/insights/analytics/what-is-artificialintelligence.html.
3. 'Pengertian Sampah Beserta Definisi, Jenis-Jenis dan Contohnya' (no date). 'Steganografi Wikipedia bahasa Indonesia, ensiklopedia bebas' (no date). Available at: https://id.wikipedia.org/wiki/Steganografi.

DOCKER SALAH SATU *PLATFORM* YANG DIBANGUN BERDASARKAN TEKNOLOGI *CONTAINER*

Mohamad Septyan Asrofil, Arda Erico Yuda, Hendrawan Widianto, Dominggo Bayu Baskara
Teknik Industri, Fakultas Teknik Informasi dan Industri,
INSTITUT TEKNOLOGI TELKOM SURABAYA

ABSTRAK- Pada saat ini *docker* dengan sangat cepat menjadi standar *tools* berbasis *container*, dan banyak di integrasikan oleh *project platform* as a *service* seperti: *dokku, deis, flynn atau vagrant*. *Docker* merupakan sebuah *project open-source* yang menyediakan *platform* terbuka untuk *developer* maupun *sysadmin* agar dapat menjalankan aplikasi dimanapun sebagai sebuah wadah (*container*) yang ringan. Menggunakan d*ocker* memungkinkan anda mengirimkan kode lebih cepat, menstandardisasi operasi aplikasi, memindahkan kode dengan lancar, dan menghemat uang dengan meningkatkan pemanfaatan sumber daya. Dengan d*ocker,* Anda mendapatkan satu objek yang dapat dijalankan di mana saja. Berbeda dengan virtualisasi yang mana aplikasi berjalan di atas *hypervisor* dan *guest operating system*, *docker* dapat menjalankan aplikasi langsung tanpa kedua hal tadi. *Docker* juga dilengkapi dengan fitur *sandbox* yang menjamin pengerjaan pengembangan dan *sysadmin* tidak terganggu. Bagi pengembang, *sandbox* menjamin aplikasinya dapat berjalan tanpa ada gangguan atas perubahan lingkungan *host*. Sedangkan bagi *sysadmin*, menjamin *host server* yang dikelola tidak terganggu dan dapat melakukan *update* tanpa takut mengganggu aplikasi. Berkat fitur *sandbox*, pengembang leluasa untuk berkreasi tanpa takut merusak programnya. *Docker* menjamin program yang kita buat akan selamanya berjalan seperti seharusnya. Pemaketan aplikasi dan seluruh kebutuhannya, memastikan aplikasi berjalan lancar pada kondisi pada lingkungan apapun*. Docker* dengan teknologi kontainernya kemudian muncul dan membuat gebrakan di tengah- tengah pengguna *hypervisor*, dan *docker* memungkinkan kini menjadi salah satu teknologi yang banyak dipakai di dalam industri *startup*.

Kata Kunci : Docker*, teknologi, container, sandbox..*

Sub Tema : Optimalisasi pengolahan aplikasi *docker* dalam teknologi kontainer(peti kemas).

BAB I

PENDAHULUAN

Perkembangan aplikasi berbasis web sangat pesat, seiring dengan perkembangan komputer dan internet. Selain itu, aplikasi berbasis web juga semakin banyak digunakan karena dapat diakses di berbagai platform komputer hanya dengan menjalankan peramban web. Sehingga, kemudahan proses deployment (penyebaran) aplikasi web beserta perangkat lunak pendukung seperti server web, server basis data, dependencies dan environment lain ke server sangat dibutuhkan. Pengguna yang memiliki layanan aplikasi berbasis server-klien (Aplikasi Web) banyak menggunakan layanan cloud computing untuk hosting aplikasinya, terutama aplikasi yang bersifat publik yang dapat diakses setiap saat. Dengan kata lain pengguna tersebut menyewa layanan cloud pada sebuah Penyedia Layanan Internet (PLI) yang berupa infrastruktur (Pusat Data) atau platform. Hal yang jadi pertimbangan untuk menggunakan cloud computing dibandingkan komputasi tradisional adalah efektivitas, keamanan, skalabilitas, kolaborasi dan lebih murah. Secara umum ada dua metode deployment aplikasi web ke dalam server. Pertama menginstal aplikasi web beserta environment yang dibutuhkan ke dalam server tunggal, kelebihannya adalah melakukan konfigurasi terhadap server lebih mudah, sederhana dan cepat dalam proses deployment. Tetapi metode tersebut memiliki kekurangan yaitu setiap aplikasi tidak terisolasi, sehingga apabila mendeploy beberapa aplikasi yang masing-masing memiliki ketergantungan dengan paket versi tertentu maka dapat menimbulkan Dependecie Hell (Cross-platform Dependencies, Conflicting Dependencies dan Custom Dependencies). Metode yang kedua yaitu dengan memanfaatkan teknologi virtualisasi berbasis hypervisor, jadi setiap aplikasi dan dependency yang dibutuhkan di-deploy ke dalam mesin virtual yang berbeda. Dengan metode ini dapat meningkatkan skalabilitas, karena 2 setiap aplikasi berjalan pada sumber daya (CPU, memori, penyimpnan data) yang berbeda sehingga dapat dengan mudah ditambahkan sesuai kebutuhan. Akan tetapi masalah klasik menjalankan mesin virtual berbasis hypervisor adalah membutuhkan sumber daya yang besar. Karena setiap mesin virtual menjalankan sistem operasi guest beserta kernel-nya sendiri terpisah dari host. Sehingga ketika menjalankan aplikasi yang mungkin besarnya hanya puluhan MB, mesin virtual juga harus menjalankan sistem operasi guest yang besarnya bisa mencapai 10 GB. Sebab itu dibutuhkan teknologi yang dapat menyediakan virtualisasi yang mengisolasi aplikasi beserta environment yang dibutuhkan dengan kebutuhan sumber daya minimal yang dapat berjalan di berbagai infrastruktur untuk memudahkan proses deployment aplikasi. Muncul sebuah perangkat lunak Platform as a Service yang dapat mengatasi permasalah tersebut yaitu Docker.

Docker merupakan perangkat lunak virtualisasi jenis Operating systemlevel virtualization berbasis LXC (Linux Container) yang memungkinkan pengembang aplikasi membuat, menguji dan menjalankan aplikasi dalam sebuah lingkungan yang berbeda, terisolasi dan fleksibel. Tujuan utama

dari virtualisasi ini adalah untuk memudahkan para pengembang dan sistem administrator aplikasi berbasis server-klien (Aplikasi Web) dalam proses pengembangan, uji coba dan pemaketan kode program dan hosting. Docker sangat ringan dan mempunyai mekanisme yang lebih maju jika dibandingkan dengan perangkat lunak virtualisasi berbasis hypervisor seperti VirtualBox atau KVM. Indikasinya adalah adanya efektivitas lebih pada Docker dalam hal penggunaan sumber daya mesin host, karena dalam proses deploymentnya Docker akan menjalankan sebuah container menggunakan base image dengan metode file system as a layer yang berarti Docker hanya akan menyalin lapisan perubahannya saja untuk dijalankan sebagai duplikasi container yang berbeda dengan base image yang sama. Berbeda dengan virtualisasi tradisional di mana kita harus menyalin seluruh container beserta kernel dan library yang berada di dalamnya, hal ini mengakibatkan adanya penghematan penyimpanan dan memori pada Docker. Kelebihan yang lain adalah Docker merupakan perangkat lunak yang berlisensi open source. Saat ini pengguna yang memiliki aplikasi berbasis web banyak menggunakan Content Management System (CMS) sebagai perangkat lunak yang dapat mempermudah pengelolaan konten dalam sebuah website. CMS sendiri adalah perangkat lunak yang digunakan untuk menambah atau memanipulasi (mengubah) isi dari suatu situs web. Diambil dari laman web perusahaan BuiltWith persentase pengguna website yang menggunakan CMS hingga tanggal 10 Oktober 2016 adalah sebagai berikut: WordPress 38%; Drupal 8%; Google search Appliance 3%; Adobe CQ 3%; DNN Software 1%; CPanel 1%; Joomla! 1%; Blooger 1%; vBulletin 1%; Lain-lain 43%.

A. Rumusan Masalah

Berdasarkan pemaparan di atas maka dapat disimpulkan beberapa rumusan masalah sebagai berikut:

a. Bagaimana implementasi penggunaan Docker pada website yang menggunakan Content Management System ?
b. Bagaimana kompatibilitas Docker untuk menjalankan aplikasi berbasis web yang telah dibangun pada platform dan lingkungan host yang berbeda ?
c. Bagaimana pengujian efektivitas Docker dalam penggunaan sumber daya penyimpanan dan memori dibandingkan dengan virtualisasi berbasis hypervisor?

B. Tujuan Penelitian

Tujuan yang ingin dicapai dari perancangan tugas akhir ini adalah sebagai berikut:

a. Implementasi penggunaan Docker pada website yang menggunakan Content Management System.
b. Melakukan pengujian kompatibilitas Docker untuk menjalankan aplikasi berbasis web yang telah dibangun pada platform dan lingkungan host yang berbeda.

c. Melakukan pengujian terhadap efektivitas Docker dalam penggunaan sumber daya penyimpanan dan memori serta membandingkannya dengan perangkat lunak virtualisasi berbasis hypervisor.

C. Penelitian Lain yang Berhubungan

Penelitian lain yang pernah dilakukan yang memiliki kemiripan dengan penelitian ini dilakukan oleh beberapa lembaga, pendidikan dan komersial. Penelitian yang juga menguji Docker adalah penelitian oleh Persada, Fahmi (2014) dari Politeknik Sukabumi. Beberapa parameter yang membedakan penelitian ini adalah:

a. Tujuan penelitian Persada membandingkan Docker dengan VirtualBox dan VMware Workstation yang memiliki hypervisor tipe 2. Sementara penelitian ini membandingkan Docker dengan KVM yang memiliki hypevisor tipe 1 dan VirtualBox yang memiliki hypervisor tipe 2.
b. Ada dua cara dalam implementasi pengembangan sebuah aplikasi web dengan menggunakan Docker, secara manual dan menggunakan skrip. Deployment secara manual artinya kita membangun aplikasi dengan langkah- langkah yang kita lakukan di dalam container melalui terminal secara langsung. Sedangkan deployment menggunakan skrip merupakan sebuah proses otomasi (auto-mate build), hal ini dilakukan dengan skrip yang disebut Dockerized. Pada penelitian Persada menggunakan Dockerized jenis Dockerfile, sedangkan pada penelitian ini menggunakan dua jenis Dockerized yaitu Dockerfile dan Docker-compose.
c. Pada pengujian Persada hanya dilakukan terhadap server web Apache dan PHP biasa. Sementara penelitian ini melakukan pengujian terhadap 5 Aplikasi yang menggunakan CMS yang berbeda, yaitu WordPress, Joomla, Drupal, OpenCart,dan PrestaShop.
d. Pengujian persada tidak ada pengujian terhadap kinerja Docker, sedangkan pengujian ini menguji kinerja dari Docker yang menggunakan aplikasi Apache JMeter berbasis Java dan aplikasi Sysstat untuk menguji rata- rata kinerja CPU. Dari kedua penelitian di atas, aspek utama yang ditonjolkan adalah performa beberapa virtualisasi yang dijalankan di atas mesin host, dengan perbandingan utama adalah performa satu sistem virtual dengan sistem virtual yang lain sebagai gambaran sebuah konsolidasi server yang sebenarnya, terhadap berbagai kondisi beban kerja.

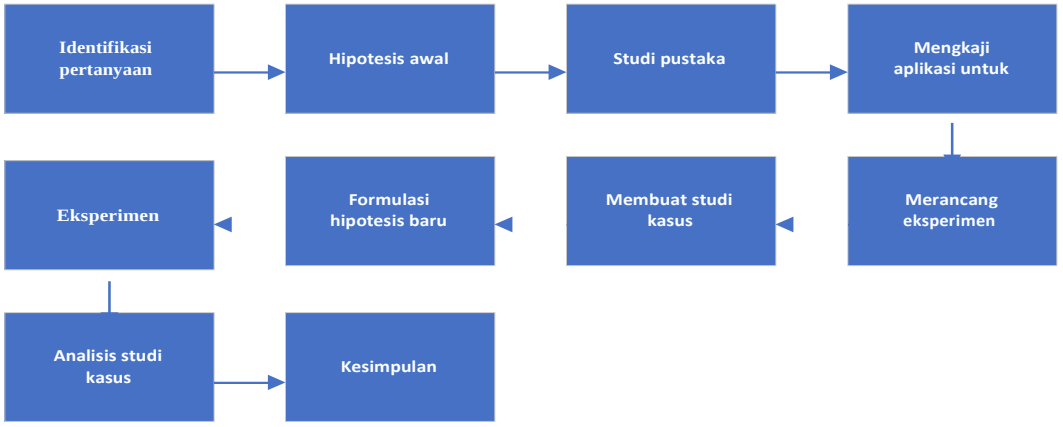
D. Batasan Masalah

Untuk memfokuskan pembahasan, maka didapat beberapa batasan masalah sebagai berikut:

a. Instalasi komputer host yang akan digunakan untuk menjalankan container Docker dan mesin virtual (VirtualBox dan KVM).
b. Uji coba dilakukan pada 5 CMS yaitu WordPress, Joomla, Drupal, OpenCart, dan PrestaShop.
c. Manajemen server web menggunakan Docker, VirtualBox, dan KVM.

BAB II

Metodologi Penelitian



Gambar 1.1 Tahapan Metodologi Eksperimen

Aberu (2008) dari Massachusetts Institute of Technology mendefinisikan dengan jelas langkah-langkah penelitian menggunakan metodologi eksperimen. Langkah-langkah itu yang dijadikan acuan dalam penelitian ini. Secara umum langkah-langkah metodologi riset seperti pada **Gambar 1.1.**Tahapan pertama melakukan identifikasi dari pertanyaan penelitian ini mengenai performa dari Docker dan membandingkannya dengan performa VirtualBox dan KVM. Tahapan kedua adalah melakukan formulasi hipotesis awal. Tahapan ketiga berisi pendalaman pengertian mengenai cloud computing, virtualisasi, Docker, dan CMS melalui studi pustaka. Studi pustaka merupakan tahapan untuk mengkaji beberapa buku, artikel ilmiah, jurnal, makalah, dan sumber dari situs internet yang terkait masalah yang penulis bahas. Tahapan keempat yaitu mengkaji aplikasi untuk eksperimen. Tahapan kelima merupakan perancangan eksperimen yang akan diimplementasikan dalam studi kasus. Tahapan keenam adalah mendefinisikan studi kasus. Tahapan ketujuh adalah kembali mendefinisikan formulasi hipotesis setelah menentukan studi kasus. Eksperimen dilakukan berdasarkan formulasi yang ditentukan pada tahap ketujuh. Tahapan kesembilan adalah menguraikan penjelasan mengenai analisis masalah yang merupakan proses percobaan implementasi pemodelan pengujian performa Docker yang akan digunakan untuk website yang menggunakan Content Management System. Kemudian hasil dari analisis ini dijadikan bahan untuk membuat kesimpulan pada tahapan kesepuluh.

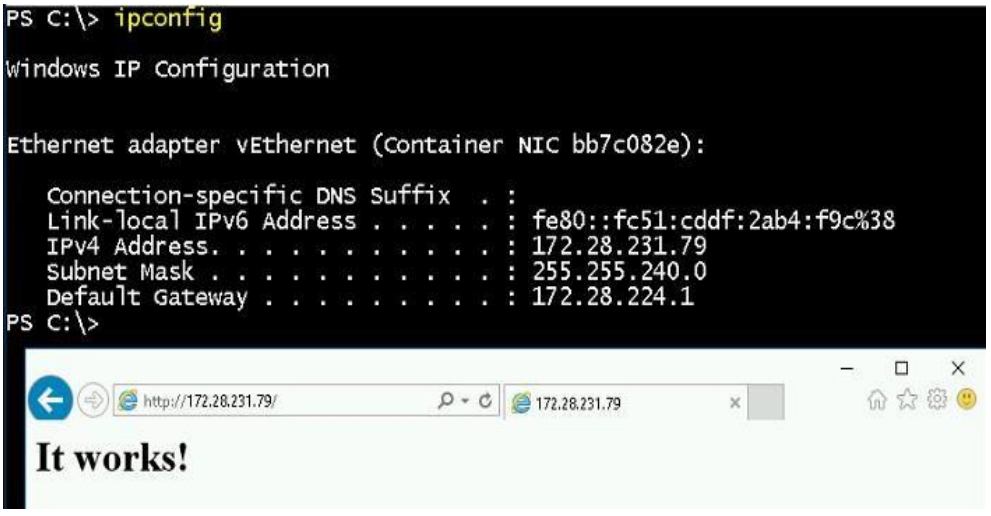
A. Metode Pengujian

Metode implementasi pengujian dilakukan dengan menguji kompatibilitas Docker untuk mengatasi permasalahan Dependency Hell serta menguji performa antara Docker, Kernel Virtual Machine (KVM), dan VirtualBox yang sudah diinstal Conten Managemen Syste

No	Pengujian Kompatibilitas, Efektivitas, dan Kinerja	
1	Pengujian Kompatibilitas Docker	Pengujian Kompatibilitas Docker Terhadap Perbedaan Platform
		Pengujian Kompatibilitas Docker Terhadap Perbedaan Environment Host
2	Pengujian Efektivitas Docker	Pengujian Efektivitas Penyimpanan (Storage) pada Docker, VirtualBox, dan KVM
		Pengujian Efektivitas Memori pada Docker, VirtualBox, dan KVM
3	Pengujian kinerja pada Docker, VirtualBox, dan KVM	

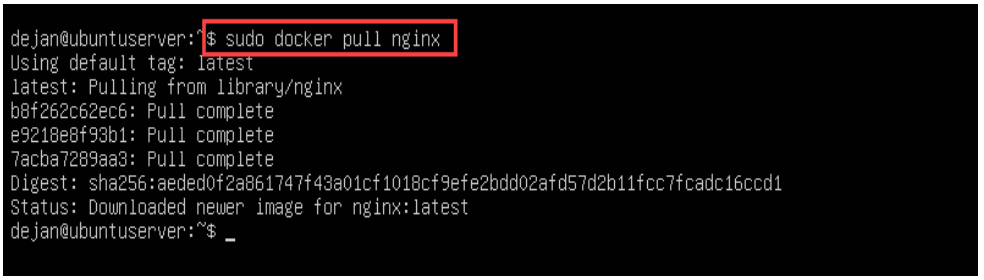
1. Web server Containers

Halaman dibawah ini merupakan halaman untuk proses akses domain ke setiap containers dengan melalui web browser dari client.



2. SSH Container

Halaman di bawah ini merupakan halaman untuk memastikan bahwa SSH server berjalan dengan baik maka akan dilakukan pengujian tersebut.

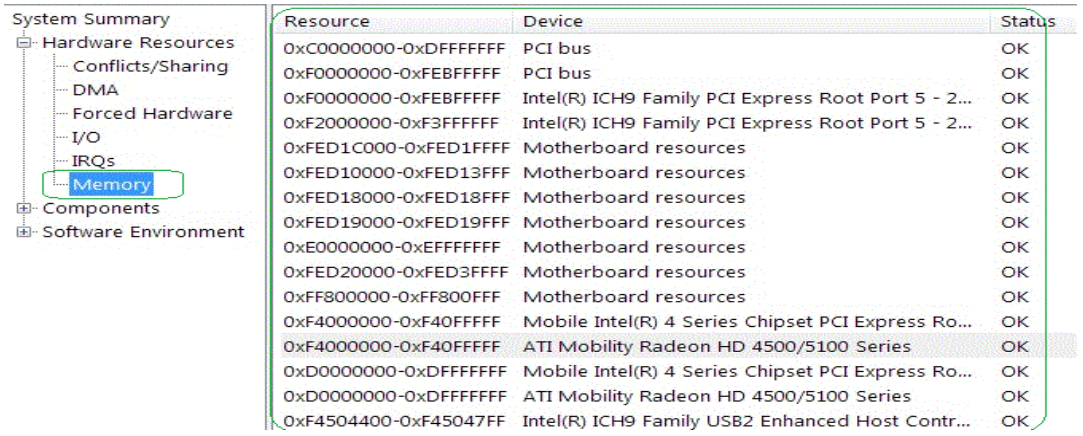


3. Info Container

Pada halaman dibawah ini merupakan untuk mengetahui informasi lengkap pada containers, dapat menggunakan command docker inspect dan menjalankan perintah docker info.

4. Resources Memory

Pada halaman dibawah ini merupakan halaman melihat kapasitas total memory dan besar memory yang sedang digunakan secara singkat dan sederhana.



5. Kesimpulan

Simpulan dari hasil kegiatan penelitian yang dilakukan adalah : 1. Telah berhasil dirancang dan diimplementasi Lightweight Virtualization dengan menggunakan Linux Containers (LXC) dan Docker deployment aplikasi web setelah dilakukan beberapa pengujian. 2. Service web pada masing-masing containers berjalan dengan baik sehingga semua aplikasi web dapat diakses oleh client. 3. Setiap aplikasi web beserta environment yang dibutuhkan berjalan pada lingkungan virtual (virtual environment) sehingga meminimalisir .

6. Daftar Pustaka

1. Adi Kurniawan, 2015, “Eksplorasi Pemanfaatan Docker untuk Mempermudah Pengelolaan Instalasi Komputer di Laboratorium Komputer Teknik Informatika Universitas Kristen Petra”,Surabaya. Arif Rahman Hakim, 2015, “Desain dan Implementasi Lightweight Virtualization berbasis linux
2. Containers dengan docker untuk deployment Aplikasi web”, Yogyakarta.
3. Docker, Installation Linux Debian. Website: https://docs.docker.com/engine/installation/linux/debian/#install-docker-ce, diakses tanggal 02 Feb 2017.
4. Firmansyah Adiputra,”Container dan Docker: Teknik Virtualisasi Dalam pengelolaan banyak Aplikasi web. Universitas Trunojoyo Madura, 2015

5. Masim vavai sugianto, Marsan susanto dkk. 2016. Virtualisasi Modern Berbasis Docker. PT Excellent Infotama Kreasindo, Bekasi Timur 17111.