

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING  
LAPORAN PROYEK AKHIR**

**SISTEM INFORMASI PEMINJAMAN PAS BANDARA ADISUTJIPTO  
MENGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKER  
FILE PADA UBUNTU**



**DISUSUN OLEH:**

**NAMA ANGGOTA : EKO EMPU WIBOWO 123170023  
FAJAR HIKMAH RAMADHAN 123170100  
KELAS : B  
ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.Kom.  
MUHAMMAD IMAM ALFATAH**

**PROGRAM STUDI INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK INDUSTRI  
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"  
YOGYAKARTA  
2020**

## HALAMAN PENGESAHAN

### **SISTEM INFORMASI PEMINJAMAN PAS BANDARA ADISUTJIPTO MENGUNAKAN UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKER FILE PADA UBUNTU**

Disusun oleh :

Eko Empu Wibowo

123170023

Fajar Hikmah Ramadhan

123170100

Telah diperiksa dan disetujui oleh Asisten Praktikum Teknologi Cloud Computing  
pada tanggal : .....

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

Muhammad Imam Alfatah

NIM. 123160119

Mengetahui,

Ka. Lab. Sistem Digital

Mangaras Yanu Florestiyanto, S.T., M.Eng.

NIK. 2 8201 13 0425 1

## **KATA PENGANTAR**

Assalamualaikum wr.wb.

Puji Syukur atas kehadiran Allah SWT. yang telah memberikan nikmat, taufik dan hidayahnya. Sehingga Kita dapat menyelesaikan Laporan Proyek Akhir Praktikum Teknologi Cloud Computing ini dengan baik. Laporan Proyek Akhir Praktikum Teknologi Cloud Computing ini dapat tersusun dengan maksimal berkat banyak dorongan dari berbagai pihak. Oleh karena itu, kita mengucapkan banyak terimakasih kepada Dosen Pengampu yang telah memberikan ilmunya tentang Teknologi Cloud Computing ini, juga kepada Asisten Laboratorium yang sabar melayani dan membantu saat berlangsungnya praktikum. Serta kepada masing-masing kedua orang tua yang mendoakan tanpa pernah putus di setiap waktu demi kelancaran proses menimba ilmu. Tidak lupa kepada teman-teman yang berkenan membantu hingga laporan ini dapat terselesaikan.

Di luar itu juga, kami sebagai manusia biasa yang tak luput dari dosa menyadari bahwa masih terdapat kekurangan dalam menyelesaikan dalam penyusunan laporan ini. Oleh karena itu, segala bentuk masukan berupa saran dan kritik yang bersifat membangun sangat diperlukan.

Demikian laporan ini kami sampaikan, semoga laporan ini dapat bermanfaat bagi kami sendiri. Umumnya bagi masyarakat luas. Terima Kasih

Wassalamualaikum wr.wb.

Yogyakarta, 1 April 2020

Penyusun

## DAFTAR ISI

HALAMAN PENGESAHAN .....	iii
KATA PENGANTAR.....	iii
DAFTAR ISI .....	ivv
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang Masalah .....	1
1.2 Tujuan Proyek Akhir .....	2
1.3 Manfaat Proyek Akhir .....	3
1.4 Tahap Penyelesaian Proyek Akhir .....	3
<b>BAB II ISI DAN PEMBAHASAN .....</b>	<b>5</b>
2.1 Komponen yang Digunakan .....	5
2.2 Rancangan Arsitektur <i>Cloud Computing</i> .....	6
2.3 Parameter dan Konfigurasi .....	8
2.4 Tahap Implementasi.....	10
2.5 Hasil Implementasi .....	17
2.6 Pengujian Singkat .....	18
<b>BAB III JADWAL Pengerjaan dan Pembagian Tugas .....</b>	<b>19</b>
3.1 Agenda Pengerjaan .....	19
3.2 Keterangan Pembagian Tugas.....	19
<b>BAB IV KESIMPULAN DAN SARAN .....</b>	<b>20</b>
4.1 Kesimpulan .....	20
4.2 Saran .....	20
<b>DAFTAR PUSTAKA .....</b>	<b>21</b>
<b>LAMPIRAN .....</b>	<b>22</b>

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Proyek Akhir

Cloud computing adalah sebuah proses pengolahan daya komputasi melalui jaringan internet yang memiliki fungsi agar dapat menjalankan program melalui komputer yang telah terkoneksi satu sama lain pada waktu yang sama. Cloud computing berawal pada tahun 1960 yang diinisiasikan oleh John McCarthy seorang ahli komputasi yang memiliki sebuah harapan yang visioner dalam dunia komputasi. John McCarthy menyampaikan bahwa suatu hari nanti komputasi akan menjadi infrastruktur publik seperti listrik dan telepon. Hingga pada awal tahun 2000, Amazon menemukan sebuah arsitektur komputasi awan yang baru dan mengalami efisiensi dalam kinerja internal perusahaan. Pada tahun 2006 Amazon mulai mengembangkan sebuah produk yang menyediakan cloud computing untuk pelanggan eksternal yang saat ini bernama Amazon Web Service. Secara umum cloud computing terbagi menjadi tiga layanan yaitu Software as a Service yang merupakan kemampuan konsumen dalam mengakses aplikasi yang disediakan oleh penyedia layanan, yang kedua Platform as a Service yang merupakan tempat untuk konsumen bisa membuat suatu aplikasi, dan yang ketiga Infrastructure as a Service yang merupakan layanan di mana konsumen bisa mengatur infrastruktur server secara mandiri Fauziah, Y. (2015, July). Penggunaan cloud computing dalam kehidupan kita contohnya adalah Linux server yang sering kita temui di sekolah, kampus, maupun sebuah perusahaan. Salah satu pemanfaatan Linux server adalah untuk menyediakan layanan hosting untuk sebuah website. Seperti dalam proyek akhir kami yaitu Peminjaman PAS Bandara Adisutjipto yang menggunakan Ubuntu LAMPP sebagai web server. Selain itu kami membuat Docker File untuk aplikasi tersebut. Secara konsep, aplikasi yang kami kembangkan ini mirip seperti aplikasi peminjaman buku di perpustakaan. Seorang admin atau petugas perpustakaan di sini bertindak mencatat data diri peminjam dan mencatat barang yang dipinjam. Catatan – catatan tersebut secara real time terekam pada database yang ada di cloud. Tentu saja data rekam tersebut tidak hanya digunakan oleh satu petugas, melainkan oleh satu instansi perpustakaan. Sehingga distribusi data dapat disampaikan secara efektif dan efisien jika menggunakan layanan dari cloud computing. Selanjutnya adalah docker yang merupakan *container* dari sebuah aplikasi yang bisa digunakan di lingkungan operating sistem apa pun. Sehingga, jika kita ingin meningkatkan sumber daya

hardware kita hanya membutuhkan waktu dalam hitungan detik. Perusahaan besar seperti Google dan Facebook sudah menerapkan aplikasi Docker ini karena kecanggihannya.

Kecepatan dalam mengakses data dan kemudahan adalah salah satu alasan user dalam menggunakan cloud computing. Dengan semakin meningkatnya mobilitas manusia menggunakan jalur udara, perlu adanya pengawasan dan pemantauan daerah terbatas bandara. Salah satunya adalah dengan menerapkan web peminjaman PAS bandara dengan cloud computing. Tentu hal ini bisa memudahkan petugas bandara dalam melakukan monitoring pemegang PAS yang tidak sembarang orang memilikinya. Selain itu, sumber daya hardware pada server internal masih kurang memadai dalam pendistribusian data yang begitu besar ini sehingga diperlukan layanan dari cloud computing. Pencatatan dan pelaporan yang dahulu bisa membutuhkan waktu yang cukup lama, sekarang dalam hitungan detik laporan bisa langsung terunduh secara rapi.

Dalam merealisasikan judul yang telah kami susun, diperlukan beberapa software dan hardware untuk menjalankan layanan yang akan kami berikan. Langkah pertama adalah menganalisis kebutuhan hardware, kemudian menentukan versi software yang akan digunakan, kemudian pembuatan vm berdasarkan analisis konfigurasi yang telah dibuat sebelumnya. Salah satu contohnya adalah menggunakan VMware Workstation untuk membuat virtual server pada komputer pribadi kita, selain itu diperlukan juga operating sistem seperti Linux untuk penyedia layanan web server. Setelah itu baru dilakukan instalasi terhadap apache, mysql, phpmyadmin untuk menunjang layanan web server. Selanjutnya adalah instalasi aplikasi pada web server dengan menggunakan docker atau secara manual. Terakhir adalah pengujian hingga sistem berhasil digunakan.

## **1.2 Tujuan Proyek Akhir**

Berdasarkan latar belakang proyek akhir yang telah dijelaskan sebelumnya, mengenai tujuan dari pembuatan proyek akhir ini adalah sebagai berikut:

1. Mengimplementasikan arsitektur *cloud computing* untuk Sistem Informasi Peminjaman Pas Bandara Adisutjipto yang telah dibuat sebelumnya dengan menggunakan Ubuntu 20.4 dan LAMPP (Apache 8, PHP 7.1, MySQL 5.2).
2. Mengintegrasikan Sistem Informasi Peminjaman Pas Bandara Adisutjipto yang berada di Ubuntu Server (LAMPP) dengan proses pembuatan Dockerfilenya.

Pembuatan Sistem Informasi Peminjaman Pas Bandara Adisutjipto juga bertujuan untuk mengintegrasikan data yang ada pada divisi administrasi dengan divisi – divisi

lainnya secara real time, dikarenakan kecepatan informasi menjadi bagian penting dari kemajuan suatu perusahaan.

### 1.3 Manfaat Proyek Akhir

Manfaat yang dapat diperoleh dari pembuatan proyek akhir ini adalah sebagai berikut:

1. Sistem Monitoring yang telah dirancang dapat digunakan sewaktu-waktu tanpa khawatir *downtime* dikarenakan terdapatnya *primary* dan *backup server* yang menggunakan arsitektur *recovery* pada *cloud computing*.
2. Perusahaan tidak perlu memperlakukan *maintenance*, dikarenakan dengan menggunakan *cloud computing*, rutinitas *maintenance* akan dilakukan sepenuhnya oleh *vendor*.
3. Batasan memori penyimpanan multimedia menjadi tidak terbatas dikarenakan sistem telah sepenuhnya beralih menggunakan *cloud computing*.
4. Permintaan data secara *realtime* dapat dilakukan secara terpusat maupun secara terpilih dikarenakan setiap sistem terhubung satu sama lain melalui *private cloud network*.

Dengan dibuatnya Sistem Peminjaman Pas Bandara menggunakan cloud computing, maka proses peminjaman PAS Bandara hanya akan menggunakan waktu yang relatif singkat. Petugas Bandara yang melakukan pendataan peminjam tidak perlu repot untuk melakukan pencatatan secara manual. Sehingga keberadaan kartu pas dapat di *tracking* secara lengkap dan jelas, karena sudah tercatat pada cloud. Selain itu tindakan preventif atau pencegahan bisa dilakukan sejak dini terhadap oknum – oknum yang menyalahgunakan wewenangnya untuk masuk ke area terbatas bandara dengan tujuan yang tidak jelas. Tentu hal ini akan meningkatkan keamanan dan kenyamanan operasional bandara.

### 1.4 Tahap Penyelesaian Proyek Akhir

Tahapan secara singkat untuk penyelesaian proyek akhir ini adalah sebagai berikut:

1. Menganalisis kebutuhan dari Sistem Peminjaman Pas Bandara untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan basis IaaS/SaaS/PaaS/DBaaS dan XaaS/WaaS.

2. Mengintegrasikan penyimpanan data peminjam pas pada buku besar dengan Sistem Peminjaman Pas Bandara yang berada di Ubuntu Server.
3. Merancang topologi *cloud computing* untuk mengintegrasikan dua sub sistem yang berbeda sehingga dapat digunakan secara terintegrasi.
4. Melakukan konfigurasi Ubuntu Server sebagai *primary* dan *backup* untuk dapat digunakan sebagai *recovery* sehingga Sistem Peminjaman Pas Bandara ketersediaannya/*availability*-nya maksimal.
5. Menguji keandalan arsitektur *cloud computing* yang dibangun dengan beberapa pengujian yaitu kecepatan waktu akses, batasan akses sesuai konfigurasi, dsb...
6. Menghasilkan Sistem Peminjaman Pas Bandara yang berbasiskan *cloud computing* sesuai standar ISO 9001.
7. Menghasilkan Docker File Sistem Peminjaman Pas Bandara yang mudah diinstal di perangkat mana pun.



## BAB II

### ISI DAN PEMBAHASAN

#### 2.1 Komponen yang Digunakan

Komponen yang digunakan pada tugas proyek akhir ini terbagi menjadi dua bagian yang saling terintegrasi satu sama lainnya sesuai dengan judul yang telah dituliskan.

Untuk membangun “Sistem Informasi Peminjaman Pas Bandara Adisutjipto menggunakan Ubuntu LAMPP” yang berbasis konsep *cloud computing*, maka diperlukan analisis berbagai komponen. Berikut akan dijelaskan terlebih dahulu dalam bentuk poin-poin singkat:

1. Sistem yang telah dibangun menggunakan bahasa pemrograman PHP dengan versi 7.3 dan bahasa HTML dengan versi minimal 4.0.
2. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 5.2 sehingga dapat digunakan untuk menyimpan berbagai data kendaraan bermotor yang dibutuhkan oleh sistem tersebut.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunaanya ialah seluruh bandar udara yang ada di Indonesia dan kantor Angkasa Pura 1 pusat di Jakarta. Tidak ada yang dapat mengakses sistem tersebut kecuali harus terhubung melalui jaringan intranet Angkasa Pura 1. Sehingga diperlukan arsitektur *cloud computing* yang bersifat *private*, tidak dapat diakses secara bebas oleh semua orang kecuali orang yang berkepentingan.
4. Sistem tersebut juga harus dapat digunakan sewaktu-waktu dan ditargetkan memiliki nilai *uptime* SLA 99,9% sehingga diperlukan minimal dua buah Ubuntu Server, satu sebagai *primary server* dan salah satunya sebagai *backup server*. Mekanisme untuk peralihan antar *server* membutuhkan *proxy server* dan semuanya menggunakan sistem operasi Ubuntu.

Berdasarkan penjelasan poin-poin tersebut, untuk komponen utama penyusun *cloud computing* yang dibutuhkan dapat disimpulkan dalam bentuk tabel sebagai berikut:

**Tabel 2.1** Spesifikasi VM *cloud computing*

No.	Nama Parameter	Nilai	Keterangan
1.	Merek Server	Virtual Machine dengan VMWare Workstation	Tidak menggunakan <i>hardware</i> fisik secara langsung, melainkan menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	2 core @2.4Ghz	Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> .

3.	Konfigurasi Jaringan <i>Guest OS</i>	Mode Bridge	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.111.128/22	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.111.1	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.111.1	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	Versi Ubuntu	Ubuntu 18.04.3 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	RAM	1 GB	Alokasi RAM untuk <i>guest OS</i>
6.	Disk Size	20 GB	Alokasi harddisk untuk <i>guest OS</i> .

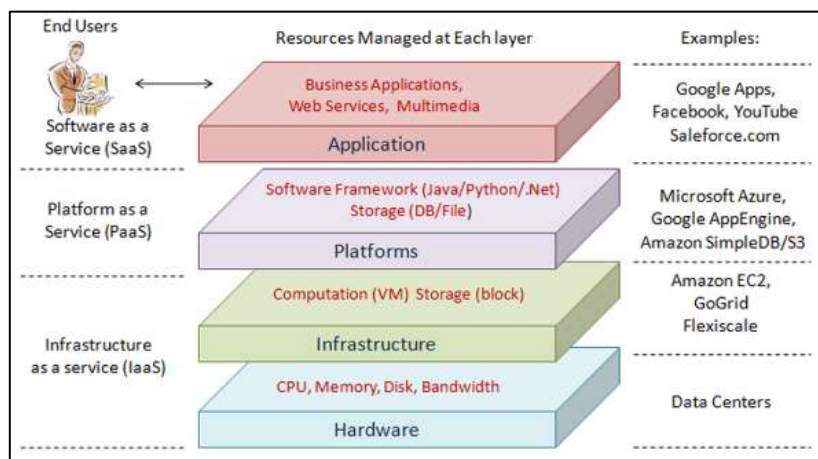
Selain spesifikasi mengenai VM *cloud computing* tersebut, untuk spesifikasi yang digunakan dalam Ubuntu OS yang telah dibuat dalam VM tersebut adalah sebagai berikut:

**Tabel 2.2** Spesifikasi Ubuntu OS

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.3	Untuk eksekusi script php.
		MySQL	Untuk penyimpanan database.
		PHP My Admin	Interface dari database MySQL
2.	Docker Engine	Docker 19.03	Aplikasi yang berbentuk wadah untuk mengepak aplikasi dan hal lainnya yang dibutuhkan agar bisa langsung digunakan
3.	Docker Compose	Docker Compose 1.24.1	Dukungan alat untuk mendefinisikan dan menjalankan multi-container

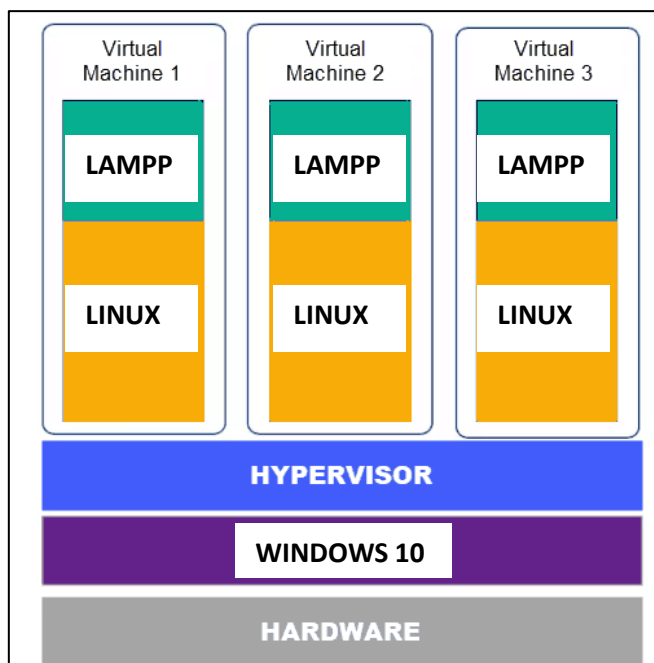
## 2.2 Rancangan Arsitektur Cloud Computing

Pada project akhir ini digunakan bentuk rancangan arsitektur IaaS di mana hardware laptop HP yang dilengkapi prosesor AMD A10 sebagai layer utama di bagian bawah, kemudian di atasnya adalah PaaS atau sistem operasi Windows dan aplikasi VMware Workstation yang berjalan. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.1** berikut ini:



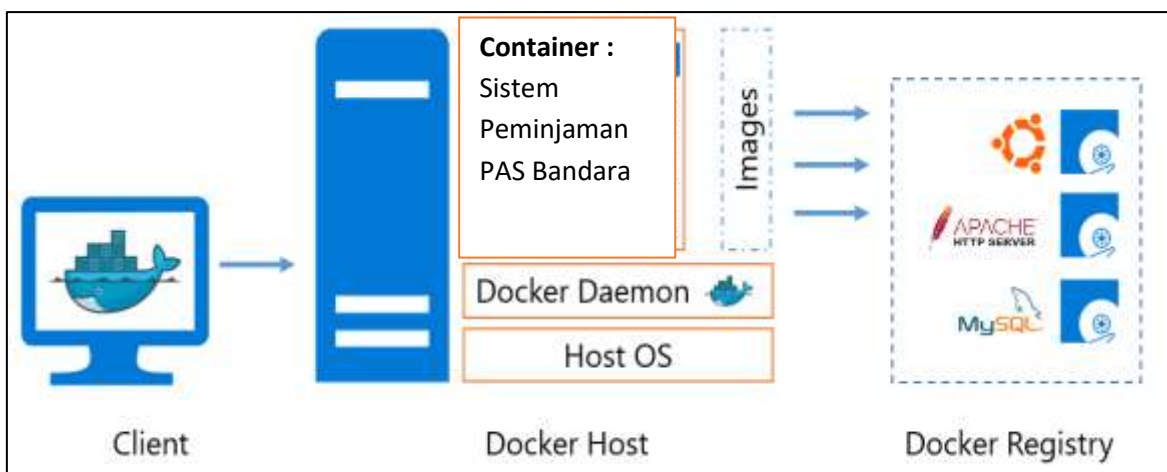
**Gambar 2.1** Penjelasan layer arsitektur terhadap komponen penyusunnya.

Dalam membuat layanan web server pada virtual machine, diperlukan sebuah operating system untuk komputer host, dan satu operating system untuk virtual machine. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.2** berikut ini:



**Gambar 2.2** Penjelasan layer arsitektur terhadap komponen Virtual Machine

Selanjutnya adalah pembuatan Dockerfile yang di dalamnya berisi penggabungan apache, php, mysql, dan project website tersebut dibuat dalam satu kesatuan. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.2** berikut ini:



**Gambar 2.2** Penjelasan layer arsitektur terhadap komponen Dockerfile

## 2.3 Parameter dan Konfigurasi

Proses pertama yang dilakukan adalah melakukan instalasi terhadap apache. Parameter yang digunakan untuk instalasi Apache dapat dilihat pada penjelasan **Modul 2.1** berikut ini:

```
$ sudo apt install apache2
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- apache2 : nama paket aplikasi untuk Apache

### Modul 2.1 Parameter instalasi Apache

Selain web server, diperlukan basis data untuk menyimpan semua file yang ada pada sebuah website. Parameter yang digunakan untuk instalasi MySQL dapat dilihat pada penjelasan **Modul 2.2** berikut ini:

```
$ sudo apt install mysql-server
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- mysql-server : nama paket aplikasi untuk database MySQL

### Modul 2.2 Parameter instalasi MySQL

Selanjutnya adalah melakukan instalasi terhadap layanan PHP. Layanan PHP ini sangat diperlukan karena pada web server Apache hanya bisa mengeksekusi file HTML, CSS, JavaScript, dan lain – lain. Parameter yang digunakan untuk instalasi PHP dapat dilihat pada penjelasan **Modul 2.3** berikut ini:

```
$ sudo apt install php libapache2-mod-php php-mysql
```

Keterangan:

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- php libapache-mod-php php-mysql : nama paket aplikasi untuk PHP

### Modul 2.3 Parameter instalasi PHP

Tahap berikutnya yaitu melakukan instalasi terhadap PHPMyAdmin untuk layanan manajemen database MySQL berbasis Web GUI. Parameter yang digunakan untuk instalasi PHP dapat dilihat pada penjelasan **Modul 2.4** berikut ini:

```
$ sudo apt install phpmyadmin php-mbstring php-gettext
```

**Keterangan:**

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- phpmyadmin php-mbstring php-gettext : nama paket aplikasi untuk PHPMyAdmin

**Modul 2.4** Parameter instalasi PHPMyAdmin

Setelah mencoba login pada PHPMyAdmin, beberapa device biasanya tidak bisa masuk ke dashboard PHPMyAdmin dengan akun root. Sehingga diperlukan beberapa konfigurasi tambahan. Parameter yang digunakan untuk konfigurasi dapat dilihat pada penjelasan **Modul 2.5** berikut ini:

```
$ sudo mysql -u root -p
Mysql> UPDATE mysql.user SET plugin = 'mysql_native_password'
      authentication_string = PASSWORD('KATA_SANDI_ROOT_USER')
      WHERE User = 'root';
Mysql> GRANT ALL PRIVILEGES ON . TO 'root'@'localhost' WITH GRANT
OPTION;
```

**Modul 2.5** Parameter instalasi konfigurasi PHPMyAdmin

Dalam melakukan pembuatan Dockerfile ada beberapa hal yang dilakukan untuk mengizinkan akses Docker Repository melalui HTTPS. Parameter yang digunakan untuk instalasi Dockerfile dapat dilihat pada penjelasan **Modul 2.6** berikut ini:

```
$ sudo apt-get update
$ sudo apt-get install apt-transport-https ca-certificates curl
software-properties-common
```

**Modul 2.6** Parameter instalasi Dockerfile

Tahap berikutnya adalah menambahkan GPG key untuk fitur keamanan. Parameter yang digunakan untuk penambahan GPG key dapat dilihat pada penjelasan **Modul 2.7** berikut ini:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-
key add -
```

**Modul 2.7** Parameter penambahan GPG key

Selanjutnya adalah melakukan instalasi terhadap Docker Repository. Parameter yang digunakan untuk instalasi Docker Repository dapat dilihat pada penjelasan **Modul 2.8** berikut ini:

```
$ sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

**Modul 2.8** Parameter instalasi Docker Repository

Kemudian proses instalasi Docker engine dapat dilakukan. Parameter yang digunakan untuk instalasi Docker engine dapat dilihat pada penjelasan **Modul 2.9** berikut ini:

```
$ sudo apt-get update
```

```
$ sudo apt-get install docker-ce
```

### Modul 2.9 Parameter instalasi Docker Engine

Untuk menjalankan Docker dengan multi container, diperlukan instalasi Docker Compose. Parameter yang digunakan untuk instalasi Docker Compose dapat dilihat pada penjelasan **Modul 2.10** berikut ini:

```
$ sudo apt install curl -y
$ sudo curl -L
https://github.com/docker/compose/releases/download/1.24.1/docker-  
compose-\$\(uname -s\)-\$\(uname -m\) -o /usr/local/bin/docker-compose
```

### Modul 2.10 Parameter instalasi Docker Compose

Setelah instalasi Docker Compose berhasil dilakukan, berikutnya adalah menjalankan docker compose tersebut. Parameter yang digunakan untuk menjalankan Docker Compose dapat dilihat pada penjelasan **Modul 2.11** berikut ini:

```
$ sudo chmod +x /usr/local/bin/docker-compose
$ docker-compose version
```

### Modul 2.11 Parameter run Docker Compose

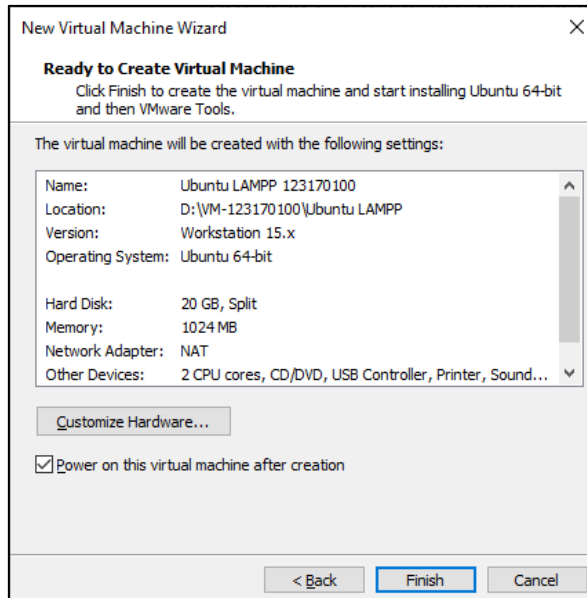
Tahap akhir adalah dengan melakukan setting docker compose untuk LAMPP server yang dibuat pada project yang terdiri dari apache, mysql, dan phpmyadmin. Parameter yang digunakan untuk menjalankan LAMPP server pada docker dapat dilihat pada penjelasan **Modul 2.11** berikut ini:

```
$ docker-compose up -d
```

### Modul 2.11 Parameter menjalankan LAMPP pada Docker

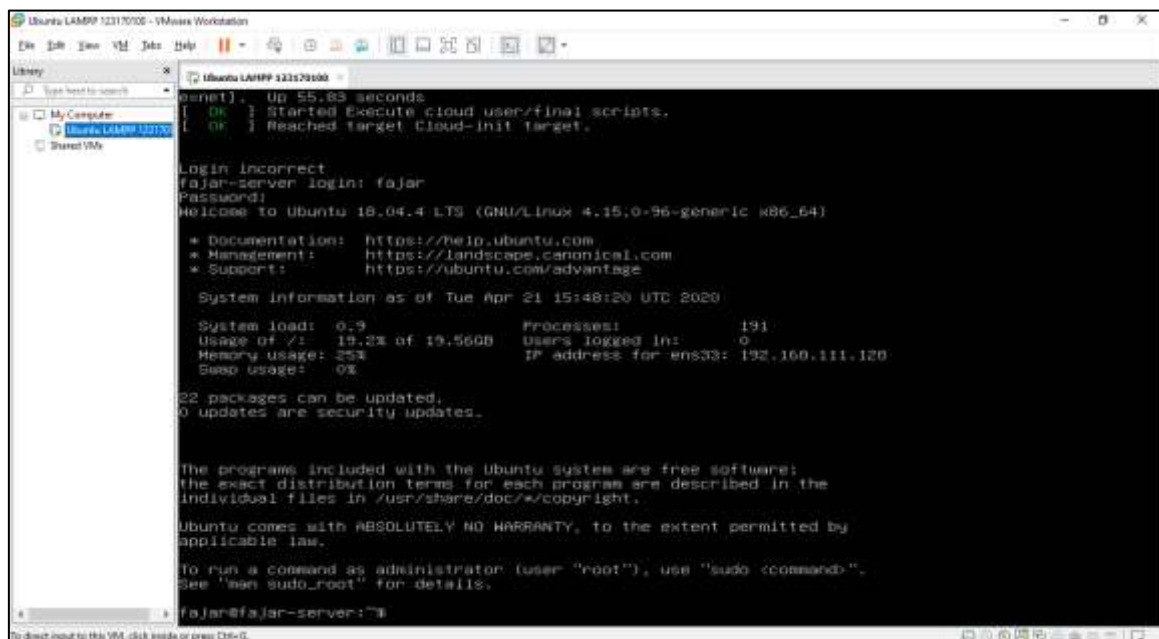
## 2.4 Tahap Implementasi

Pada tahap awal pada *wizard* pembuatan VM pada VMware Workstation digunakan opsi konfigurasi *Custom* dan disesuaikan dengan spesifikasi yang ada pada **Tabel 2.1** sehingga akan menghasilkan sebuah virtual machine seperti pada **Gambar 2.3** berikut ini:



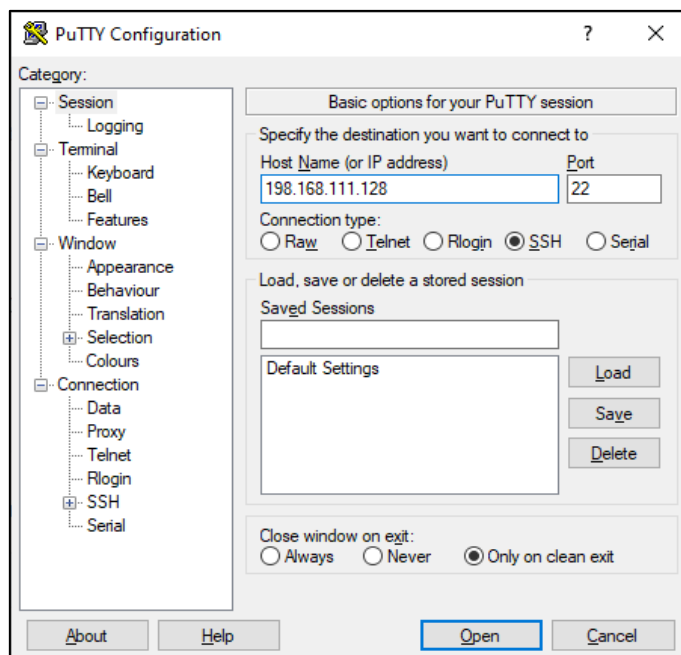
**Gambar 2.4** Tampilan opsi pemilihan *mode wizard* pembuatan VM

Setelah *virtual machine* berhasil dibuat, maka akan muncul permintaan untuk login dan tampilannya akan muncul seperti pada **Gambar 2.5** berikut:



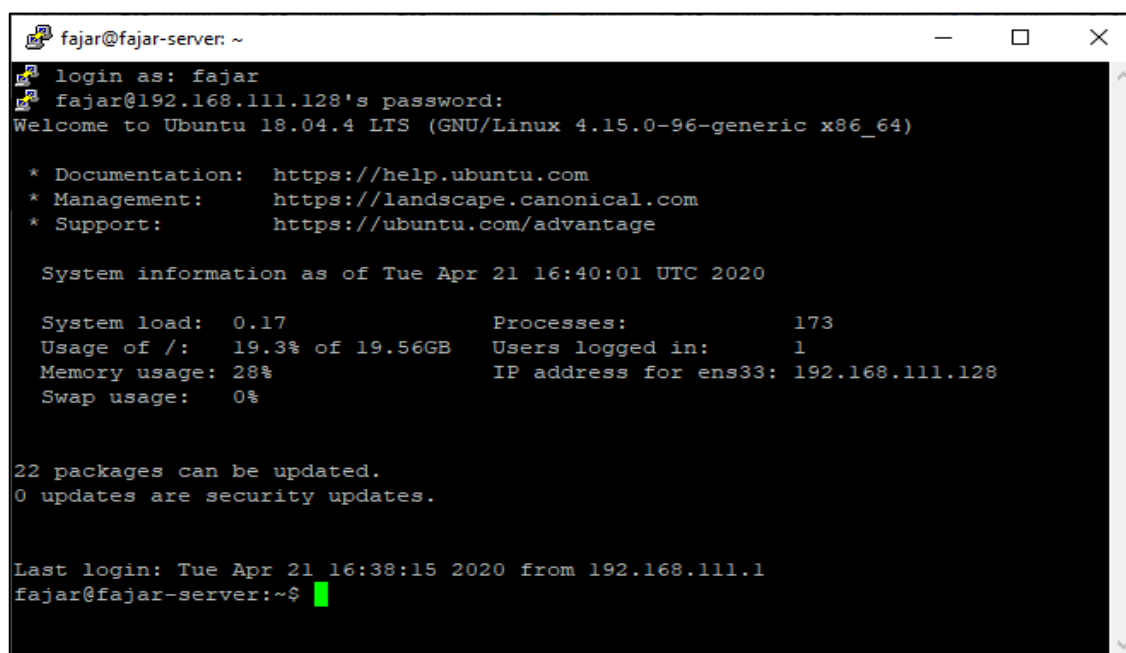
**Gambar 2.5** Halaman awal Ubuntu Server 18.04

Untuk mempermudah dalam melakukan pengoperasian Ubuntu server, maka tahap berikutnya adalah melakukan *remote* dengan menggunakan aplikasi bernama PuTTY dengan mengkonfigurasinya terlebih dahulu yakni memasukkan IP Address Ubuntu server seperti pada **Gambar 2.6** berikut:



**Gambar 2.6** Konfigurasi pada aplikasi PuTTY

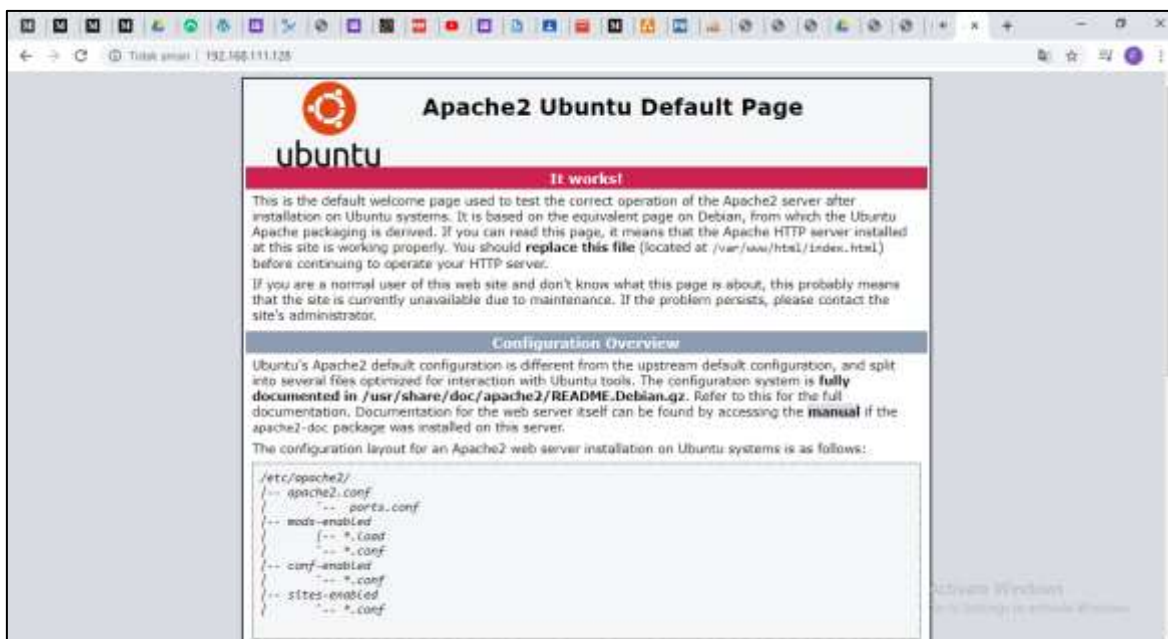
Setelah berhasil, maka akan masuk ke tampilan awal yang ada pada aplikasi PuTTY yakni pada **Gambar 2.7** berikut:



**Gambar 2.7** Tampilan awal pada aplikasi PuTTY

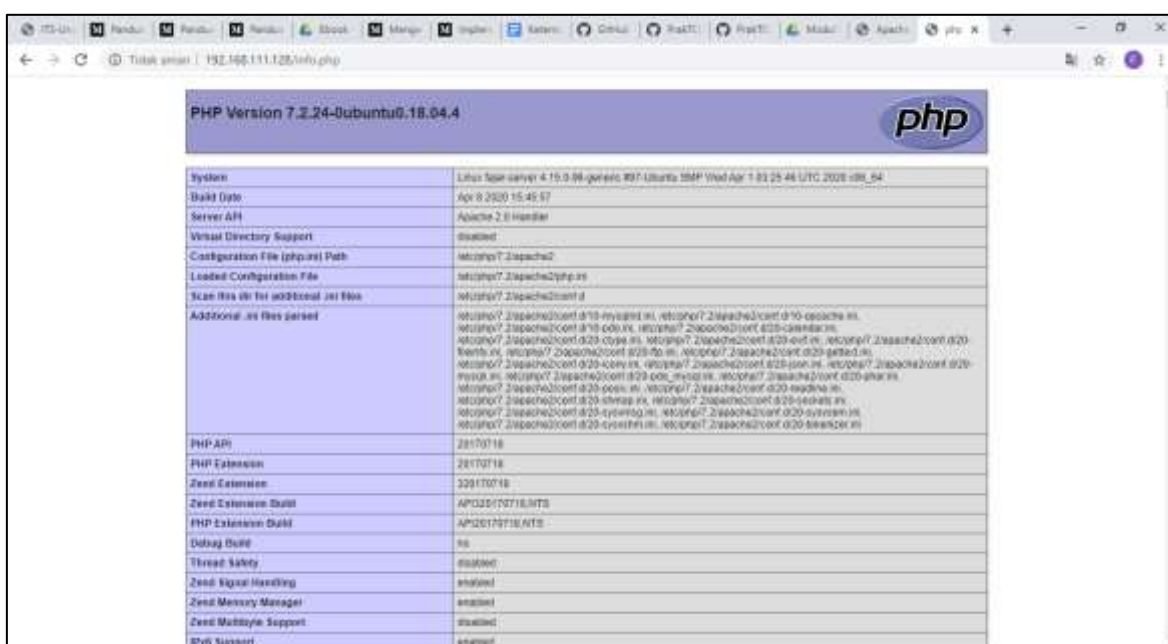
Untuk memberikan layanan web server diperlukan aplikasi bernama Apache yang bisa membaca HTML, CSS maupun JS dengan cara mengetikkan perintah sesuai dengan **Modul 2.1**, selanjutnya tunggu proses hingga selesai dan ketika IP Address kita jalankan pada web browser maka akan muncul tampilan seperti pada **Gambar 2.8** berikut:





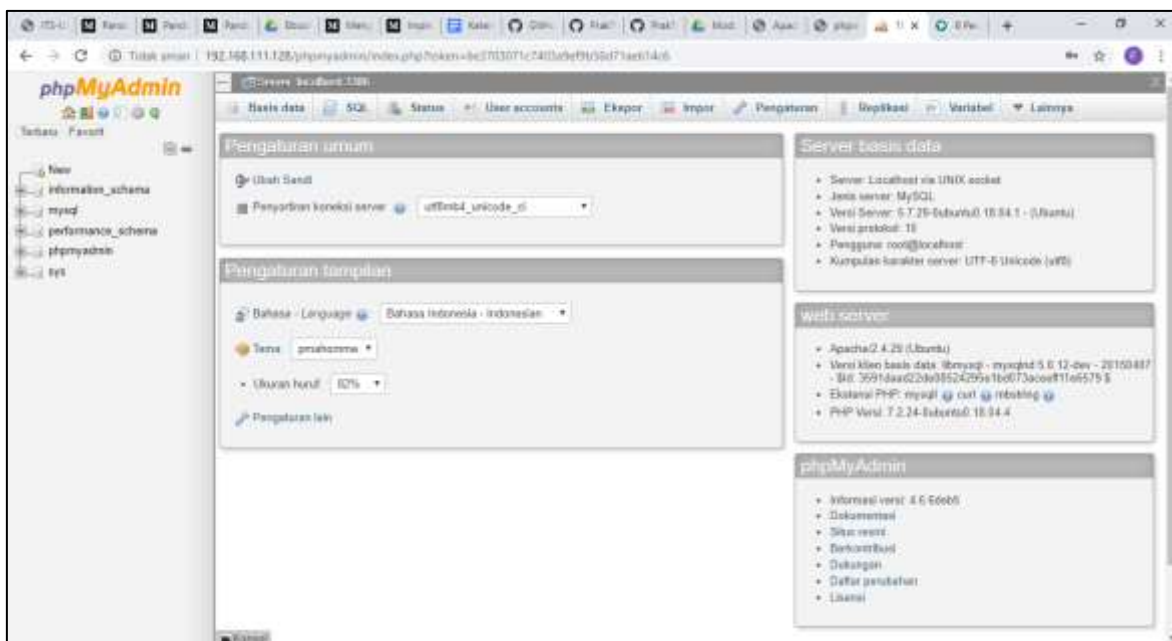
**Gambar 2.8** Apache Ubuntu Deafult Page

Tahap berikutnya adalah melakukan instalasi PHP seperti yang ada pada **Modul 2.3** sehingga web server tidak hanya membaca HTML, CSS, JS saja melainkan bisa mengeksekusi file – file berkentensi PHP seperti pada **Gambar 2.9** berikut:



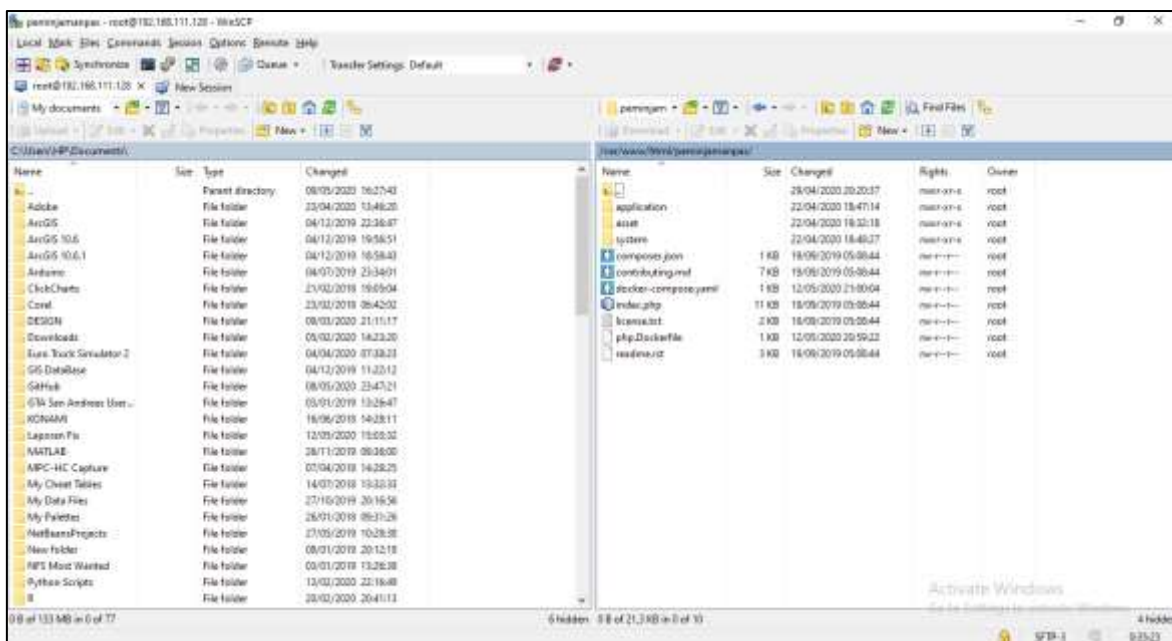
**Gambar 2.9** Halaman PHP Info

Dalam melakukan penyimpanan data diperlukan sebuah database, cara instalasi ada pada **Modul 2.2** dan **Modul 2.4** setelah proses instalasi berhasil maka akan muncul tampilan seperti pada **Gambar 2.10** berikut:



**Gambar 2.10** Halaman PHPMyAdmin

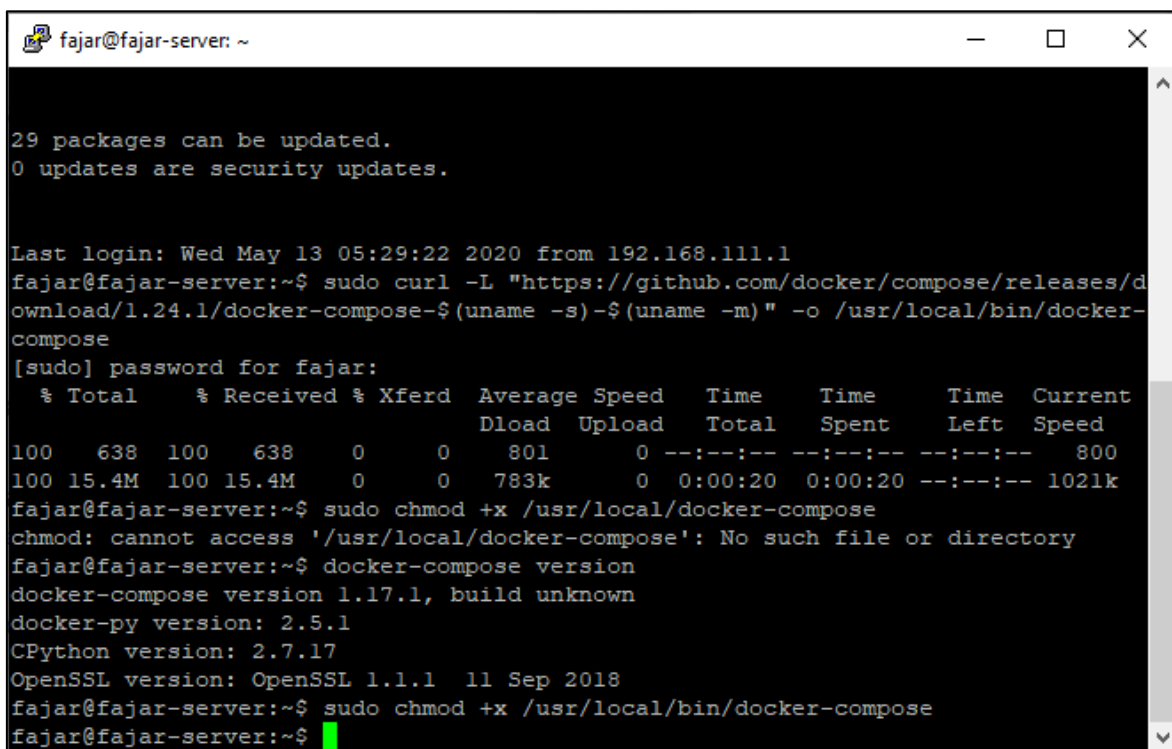
Setelah itu adalah melakukan hosting di Ubuntu server dengan menggunakan bantuan WinSCP untuk melakukan manajemen file yang ada pada Ubuntu, file project tadi di *copy* ke directory `/var/www/html/*` sehingga nantinya web tersebut bisa diakses oleh client seperti pada **Gambar 2.11** berikut:



**Gambar 2.11** Hosting web dengan bantuan WinSCP

Proses terakhir adalah pembuatan Dockerfile yang berisi container – container webserver, database service, dan tentu project web. Proses instalasi Dockerfile ada pada

**Modul 2.5** sampai dengan **Modul 2.10**. Setelah proses instalasi berhasil maka akan muncul tampilan seperti pada **Gambar 2.12** berikut:

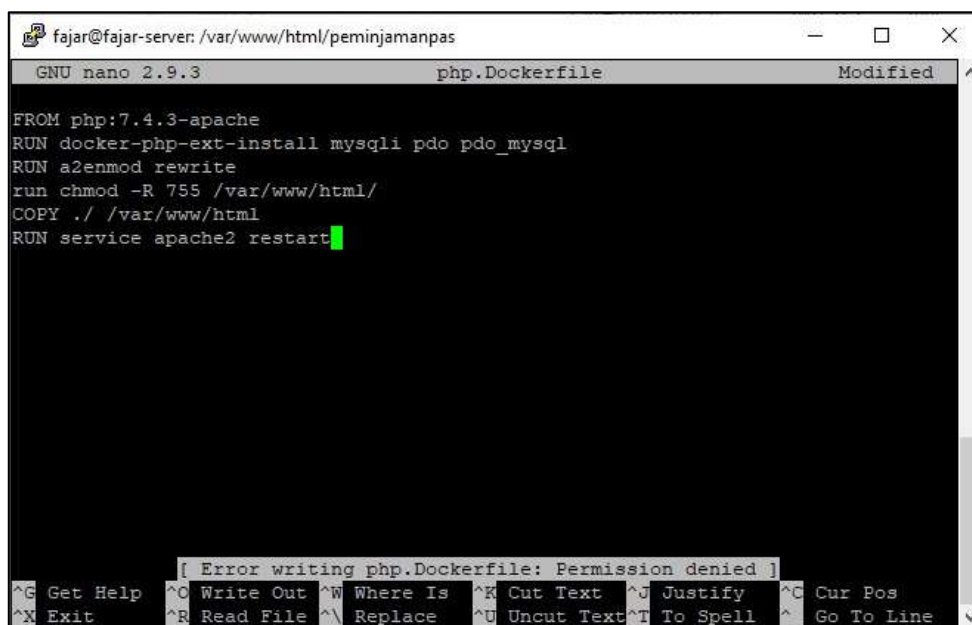


```
fajar@fajar-server: ~
29 packages can be updated.
0 updates are security updates.

Last login: Wed May 13 05:29:22 2020 from 192.168.111.1
fajar@fajar-server:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.24.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
[sudo] password for fajar:
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 638    100 638    0     0    801      0 --:--:-- --:--:-- --:--:--    800
100 15.4M  100 15.4M    0     0  783k      0  0:00:20  0:00:20 --:--:-- 1021k
fajar@fajar-server:~$ sudo chmod +x /usr/local/bin/docker-compose
chmod: cannot access '/usr/local/bin/docker-compose': No such file or directory
fajar@fajar-server:~$ docker-compose version
docker-compose version 1.17.1, build unknown
docker-py version: 2.5.1
CPython version: 2.7.17
OpenSSL version: OpenSSL 1.1.1 11 Sep 2018
fajar@fajar-server:~$ sudo chmod +x /usr/local/bin/docker-compose
fajar@fajar-server:~$
```

**Gambar 2.12** Instalasi Docker engine dan Docker Compose

Setelah itu diperlukan file `php.dockerfile` yang akan menjalankan container – container pada Dockerfile. Adapun isi dari file `php.Dockerfile` ada pada **Gambar 2.13** berikut:



```
fajar@fajar-server: /var/www/html/peminjamanpas
GNU nano 2.9.3      php.Dockerfile      Modified
FROM php:7.4.3-apache
RUN docker-php-ext-install mysqli pdo pdo_mysql
RUN a2enmod rewrite
run chmod -R 755 /var/www/html/
COPY ./ /var/www/html
RUN service apache2 restart
[ Error writing php.Dockerfile: Permission denied ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell ^_ Go To Line
```

**Gambar 2.13** Isi `php.Dockerfile`

Selain itu, diperlukan file docker-compose.yaml untuk menjalankan beberapa container secara bersamaan. Container tersebut berisi apache, php, mysql. Untuk listing program secara utuh terdapat pada **Listing Program 2.1** pada lampiran. Adapun gambaran dari file docker-compose.yaml ada pada **Gambar 2.14** berikut:



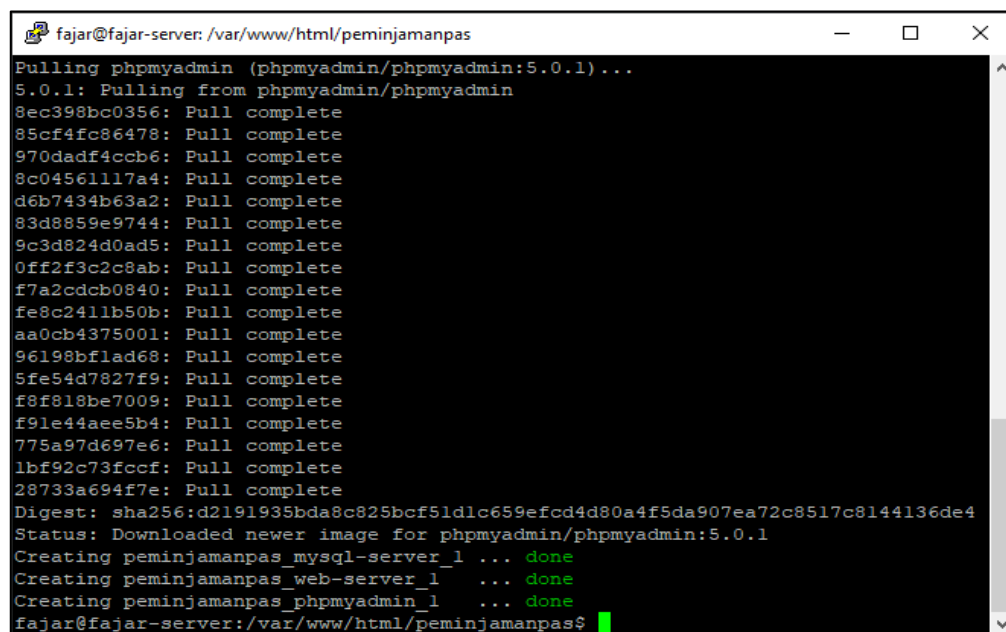
```

docker-compose - Notepad
File Edit Format View Help
version: "3.7"
services:
  web-server:
    build:
      dockerfile: php.Dockerfile
      context: .
    restart: always
    volumes:
      - "/html:/var/www/html/"
    ports:
      - "8080:80"
  mysql-server:
    image: mysql:8.0.19
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: secret
    volumes:
      - mysql-data:/var/lib/mysql
  phpmyadmin:
    image: phpmyadmin/phpmyadmin:5.0.1
    restart: always
    environment:
      PMA_HOST: mysql-server
      PMA_USER: root
      PMA_PASSWORD: secret
    ports:
      - "5000:80"
Ln 1, Col 1    100%    Windows (CRLF)    UTF-8

```

**Gambar 2.14** Isi file docker-compose.yaml

Terakhir semua Dockerfile dijalankan dengan perintah yang ada pada **Modul 2.11** . Adapun isi dari file docker-compose.yaml ada pada **Gambar 2.15** berikut:



```

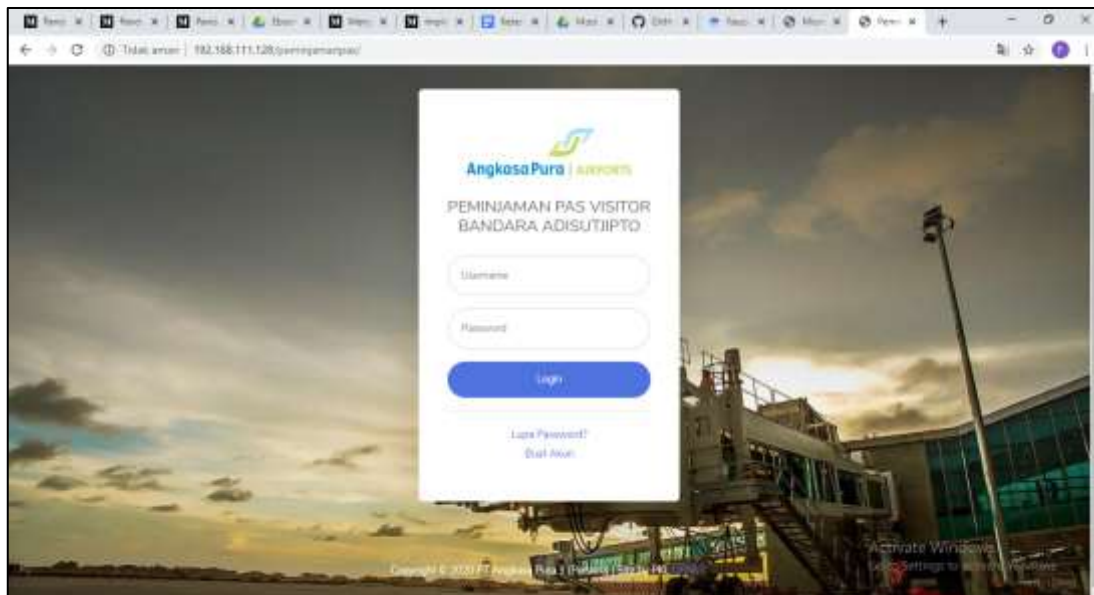
fajar@fajar-server: /var/www/html/peminjamanpas
Pulling phpmyadmin (phpmyadmin/phpmyadmin:5.0.1)...
5.0.1: Pulling from phpmyadmin/phpmyadmin
8ec398bc0356: Pull complete
85cf4fc86478: Pull complete
970dadf4ccb6: Pull complete
8c04561117a4: Pull complete
d6b7434b63a2: Pull complete
83d8859e9744: Pull complete
9c3d824d0ad5: Pull complete
0ff2f3c2c8ab: Pull complete
f7a2cdcb0840: Pull complete
fe8c2411b50b: Pull complete
aa0cb4375001: Pull complete
96198bf1ad68: Pull complete
5fe54d7827f9: Pull complete
f8f818be7009: Pull complete
f91e44aee5b4: Pull complete
775a97d697e6: Pull complete
1bf92c73fccf: Pull complete
28733a694f7e: Pull complete
Digest: sha256:d2191935bda8c825bcf51dlc659efcd4d80a4f5da907ea72c8517c8144136de4
Status: Downloaded newer image for phpmyadmin/phpmyadmin:5.0.1
Creating peminjamanpas_mysql-server_1 ... done
Creating peminjamanpas_web-server_1 ... done
Creating peminjamanpas_phpmyadmin_1 ... done
fajar@fajar-server: /var/www/html/peminjamanpas$

```

**Gambar 2.15** Isi file docker-compose.yaml

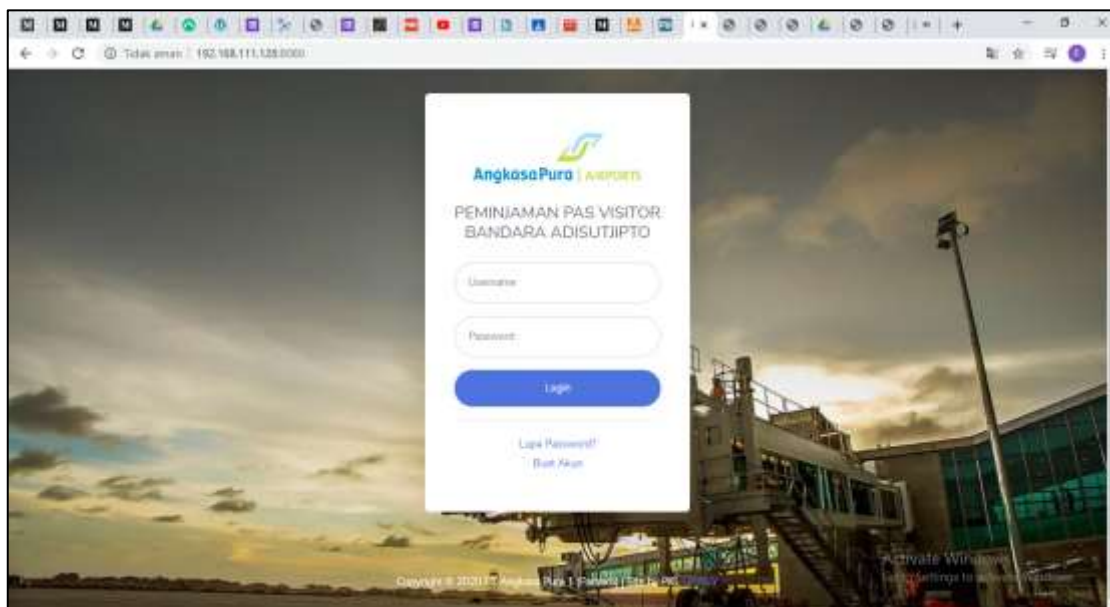
## 2.5 Hasil Implementasi

Setelah melakukan instalasi dan konfigurasi pada Ubuntu LAMPP maka ketika kita mengakses <https://192.168.111.128/peminjamanpas/> maka akan muncul halaman login dari aplikasi “Peminjaman PAS Bandara Adisutjipto” yang menandakan proses telah berhasil seperti pada **Gambar 2.16** berikut:



**Gambar 2.16** Halaman Login Peminjaman PAS Bandara Adisutjipto

Sedangkan untuk yang menggunakan Dockerfile, alamat URL nya melalui <https://192.168.111.128:8080/> dan berhasil diakses dengan normal seperti pada **Gambar 2.17** berikut:

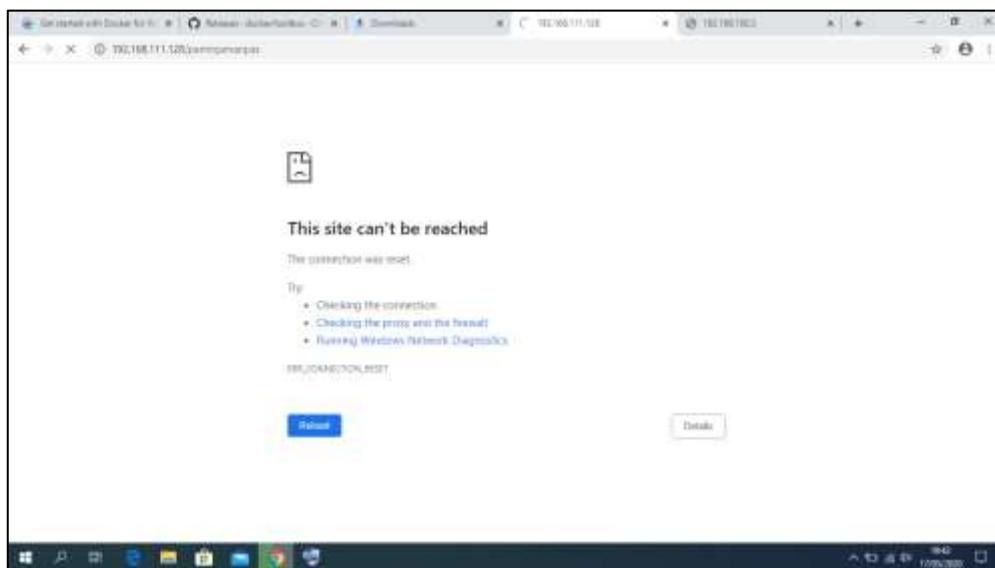


**Gambar 2.17** Halaman Login Peminjaman PAS Bandara Adisutjipto



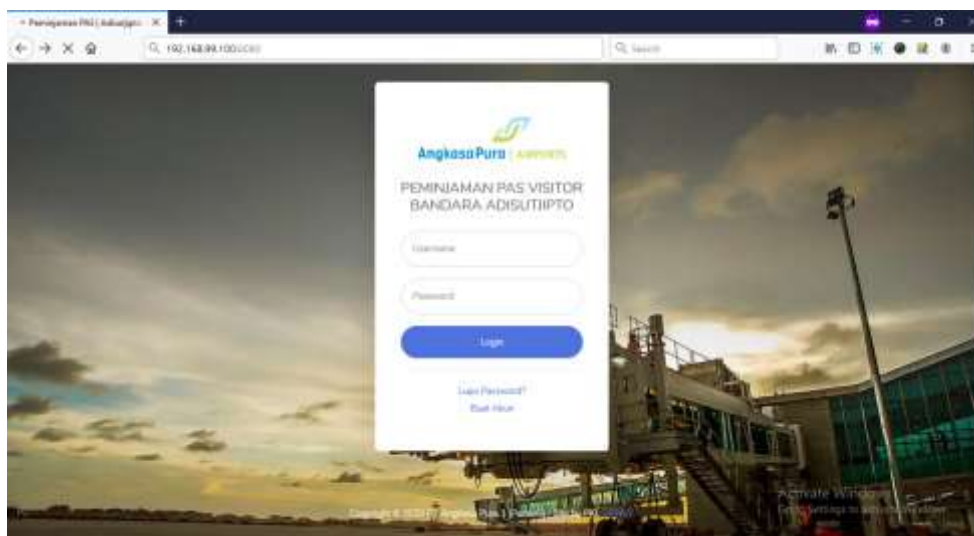
## 2.6 Pengujian Singkat

Pada tahap ini Ubuntu LAMPP akan di akses melalui Laptop yang berbeda namun dalam satu jaringan yang sama yaitu WiFi Indehome. Laptop yang pertama adalah sebagai host yang sudah terinstal VMWare dan memiliki Ubuntu Server yang akan diuji coba. Sedangkan Laptop yang kedua akan diposisikan sebagai Client. Dari hasil uji coba, Laptop kedua tidak bisa mengakses webserver pada webserver virtual machine yang ada di Laptop pertama. Kemungkinan diperlukan beberapa konfigurasi lagi agar hal tersebut bisa dilakukan seperti pada **Gambar 2.18** berikut:



**Gambar 2.18** Halaman Login tidak bisa diakses

Sedangkan untuk Dockerfile, pengujian dilakukan pada Laptop yang berbeda dengan menggunakan Docker Toolbox yang diinstal pada Windows 10 berjalan dengan baik seperti pada **Gambar 2.19** berikut:



**Gambar 2.19** Halaman Login bisa di akses

### BAB III

#### JADWAL Pengerjaan dan Pembagian Tugas

##### 3.1 Agenda Pengerjaan

Berikut pada **Tabel 3.1** merupakan pembagian jenis tugas proyek akhir terhadap alokasi waktu pengerjaan pada bulan Maret, April, Mei tahun 2020:

**Tabel 3.1** Agenda Pengerjaan Proyek

No.	Jenis Tugas	Waktu Pengerjaan							
		Maret		April				Mei	
		3	4	1	2	3	4	1	2
1.	Analisa Persoalan								
2.	Pembagian Tugas								
3.	Pembuatan Tugas 1 (Ubuntu LAMPP)								
4.	Perancangan Database								
5.	Pembuatan Laporan Submisi								
6.	Pembuatan Tugas 2 (Dockerfile)								
7.	Penyelesaian Error								
8.	Penyelesaian Laporan Akhir								
9.	Presentasi Proyek Akhir								

##### 3.2 Keterangan Pembagian Tugas

Berikut pada **Tabel 3.2** merupakan pembagian tugas-tugas pada proyek akhir terhadap anggota pada tim pembuatan proyek akhir:

**Tabel 3.2** Pembagian Tugas Proyek

No.	Keterangan Tugas	Penanggung Jawab
1.	Perancangan Arsitektur Cloud Computing	Fajar
2.	Pengujian Singkat	Eko
3.	Latar Belakang Masalah	Fajar
4.	Agenda Pengerjaan Proyek	Eko
5.	Tugas 1 (Ubuntu LAMPP)	Eko
6.	Tugas 2 (Dockerfile)	Fajar
7.	Penyusunan Laporan	Fajar & Eko

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Berawal dari hasil analisa dalam mengamati waktu yang dibutuhkan oleh petugas Bandara Adisutjipto dalam mengakses aplikasi web pada server lokal, diperlukan sebuah layanan yang bisa lebih cepat dan bisa di akses oleh pegawai di mana saja. Maka dari itu diperlukan sebuah layanan webserver berbasis cloud. Dalam perancangan Ubuntu LAMPP sebagai webserver, tidak ada kendala dalam hal layanan web server. Namun untuk layanan database terdapat kendala yakni migrasi dari database mysql ke database PostgreSQL. Ada beberapa konfigurasi yang sedikit berbeda antara kedua layanan database ini. Namun hal ini dapat teratasi dengan beberapa saran yang ada di internet. Dalam pengujian masalah yang pertama yaitu mengakses web dari VMWare menggunakan laptop yang berbeda tapi dalam satu jaringan Wifi. Web tidak bisa diakses secara langsung, dikarenakan perlu adanya settingan tambahan yang harus diterapkan dalam Ubuntu LAMPP. Untuk pembagian tugas pada **BAB III**, semua tugas bisa dikerjakan dengan baik sesuai tugas masing – masing.

#### **4.2 Saran**

Sebaiknya laptop yang digunakan berprocessor Intel, karena laptop yang kami gunakan yaitu AMD sering mengalami masalah dalam menjalankan software VMWare. Ada beberapa hal yang perlu di konfigurasi ketika menggunakan processor AMD, sehingga ada beberapa penyesuaian.



## DAFTAR PUSTAKA

Wang, Shulong., Hou, Yibin., Gao, Fang., & Ji, Xinrong. 2016. "A Novel IoT Access Architecture for Vehicle Monitoring System". 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT).

\_\_\_\_\_, <<https://idcloudhost.com/pengertian-internet-of-things-iot/>>, (23 Jun 2016, accessed 10 Mei 2019)

Erick, Jan Solem. 2012. *Programming Computer Vision with Python*.

Fauziah, Y. (2015, July). Arsitektur cloud computing pada sistem informasi desa sebagai layanan akses informasi desa. In *Seminar Nasional Informatika (SEMNASIF)* (Vol. 1, No. 1).

\_\_\_\_\_, <<https://www.ekrut.com/media/mengenal-docker-dan-penggunaannya-seperti-apa-sih>>, (9 Jan 2020, accessed 30 April 2020)

\_\_\_\_\_, <[https://linuxhint.com/lamp\\_server\\_docker/](https://linuxhint.com/lamp_server_docker/)>, (Maret 2020, accessed 10 Mei 2020)

## LAMPIRAN

Konfigurasi berikut adalah isi dari file docker-compose.yaml secara utuh seperti pada **Listing Program 2.1** berikut:

```
version: "3.3"
services:
  web-server:
    build:
      dockerfile: php.Dockerfile
      context: .
    restart: always
    volumes:
      - "./:/var/www/html/"
    ports:
      - "8080:80"
  mysql-server:
    image: mysql:8.0.19
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: secret
    volumes:
      - mysql-data:/var/lib/mysql

  phpmyadmin:
    image: phpmyadmin/phpmyadmin:5.0.1
    restart: always
    environment:
      PMA_HOST: mysql-server
      PMA_USER: root
      PMA_PASSWORD: secret
    ports:
      - "5000:80"
volumes:
  mysql-data:
```

**Listing Program 2.1** Isi konfigurasi dari file docker-compose.yaml