

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING
LAPORAN PROYEK AKHIR**

**SISTEM INFORMASI TOKO PENJUALAN BAJU MENGGUNAKAN UBUNTU
LAMPP DAN DALAM BENTUK DOCKERFILE**



DISUSUN OLEH:

**NAMA ANGGOTA : VANNYA YURE PUTRA S. 123170082
OKTAVIANI ROOSDIAWATI 123170087**
KELAS : C
**ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.Kom.
WAHYU AJI NUGROHO, S.Kom.**

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

HALAMAN PENGESAHAN

SISTEM INFORMASI TOKO PENJUALAN BAJU MENGGUNAKAN UBUNTU LAMPP DAN DALAM BENTUK DOCKERFILE

Disusun oleh :

Vannya Yure Putra Swandjana

123170082

Oktaviani Roosdiawati

123170087

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

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

Wahyu Aji Nugroho, S.Kom.

Mengetahui,

Ka. Lab. Sistem Digital

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

NIK. 2 8201 13 0425 1

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan praktikum Teknologi Cloud Computing serta laporan proyek akhir praktikum yang berjudul Sistem Informasi Toko Penjualan Baju Menggunakan Ubuntu Lampp dan Dalam Bentuk Dockerfile. Adapun laporan ini berisi tentang proyek akhir yang salah satunya bersifat wajib dan pilihan yang kami pilih dari hasil pembelajaran selama praktikum berlangsung.

Tak lupa pula kami mengucapkan banyak terima kasih kepada asisten praktikum yang selalu membimbing dan mengajari kami dalam melaksanakan praktikum dan dalam penyusunan laporan akhir ini. Serta semua pihak yang telah membantu kami dalam penyusunan laporan akhir ini. Karena laporan akhir ini masih jauh dari sempurna oleh karena itu kritik dan saran yang membangun masih kami harapkan untuk penyempurnaan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan laporan akhir ini kami ucapkan terima kasih. Semoga laporan ini dapat dipergunakan sesuai dengan keperluan.

Yogyakarta, 02 April 2020

Penyusun

DAFTAR ISI

HALAMAN JUDUL/COVER	i
HALAMAN PENGESAHAN	ii
KATA PENGANTAR.....	iii
DAFTAR ISI	iv
BAB I PENDAHULUAN	1
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
BAB II ISI DAN PEMBAHASAN	4
2.1 Komponen yang Digunakan	4
2.2 Rancangan Arsitektur <i>Cloud Computing</i>	4
2.3 Parameter dan Konfigurasi	6
2.4 Tahap Implementasi.....	10
2.4.1 Tahap Implementasi Proyek Pertama	10
2.4.2 Tahap Implementasi Proyek Kedua.....	16
2.5 Hasil Implementasi	21
2.6 Pengujian Singkat	22
BAB III JADWAL Pengerjaan dan Pembagian Tugas	24
3.1 Agenda Pengerjaan	24
3.2 Keterangan Pembagian Tugas.....	24
BAB IV KESIMPULAN DAN SARAN.....	25
4.1 Kesimpulan	25
4.2 Saran	25
DAFTAR PUSTAKA.....	26
LAMPIRAN	27

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Cloud computing (komputasi awan) merupakan gabungan pemanfaatan teknologi komputer (komputasi) dalam suatu jaringan dengan pengembangan berbasis internet yang mempunyai fungsi untuk menjalankan program atau aplikasi melalui komputer – komputer yang terkoneksi pada waktu yang sama, tetapi tak semua yang terkoneksi melalui internet menggunakan *cloud computing*. Teknologi komputer berbasis sistem *Cloud* ini merupakan sebuah teknologi yang menjadikan internet sebagai pusat server untuk mengelola data dan juga aplikasi pengguna. Teknologi ini mengizinkan para pengguna untuk menjalankan program tanpa instalasi dan mengizinkan pengguna untuk mengakses data pribadi mereka melalui komputer dengan akses internet.

Penggunaan *cloud computing* memiliki banyak manfaat. Selain mengurangi biaya operasional karena pelanggan hanya akan membayar jasa yang digunakan, juga pelanggan tidak perlu menyediakan infrastruktur dan perangkat lunak ketika akan menggunakan aplikasi *cloud computing*, karena semua itu telah disediakan oleh provider dari jarak jauh dengan menggunakan media internet. Selain itu karena sifatnya sangat *mobile (based on internet)*, maka para pelanggan dapat mengaksesnya setiap saat dan dimanapun berada sehingga akan lebih efisien. Beberapa contoh aktifitas keseharian yang menggunakan *cloud computing* dewasa ini antara lain *web-based* email seperti selama ini yang dilakukan melalui Yahoo dan Microsoft Hotmail, menyimpan foto

Revolusi 4.0 berupa digitalisasi segala lini jelas memaksa industri retail untuk mendigitalisasi bisnis mereka sambil memperkuat bisnis mereka di ranah *offline*. Bisnis retail tidak hanya sekedar membuat digitalisasi toko mereka, namun juga mengadopsi *cloud computing* bagi toko *offline* mereka. Sedangkan *e-commerce* seperti tokopedia, blibli, bukalapak, JD.id yang berfokus pada penjualan *online* meningkat pesat secara pendapatan maupun skala pasar. Ekspansi pasar mereka yang diikuti dengan inovasi berupa digitalisasi membuat *e-commerce* ini diterima cepat oleh pasar. Kemudahan, kepraktisan toko retail online ini dimulai dari memilih barang, memesan, membayar hingga menerima barang tidak lagi mengikuti kaidah-kaidah retail tradisional.

Sistem informasi penjualan baju tersebut merupakan suatu sistem informasi yang berbasis web yang berfungsi untuk membantu *administrator* dapat menginput barang tanpa

memikirkan ruang memori dikarenakan banyaknya barang menyebabkan penggunaan memori yang cukup banyak. Jika hal tersebut tidak dilakukan maka *administrator* perlu mengeluarkan untuk melakukan pembaruan memori. Selain hal tersebut, penggunaan teknologi *cloud computing* juga untuk menghindari rusaknya memori yang dimana data *administrator* akan ikut rusak. Oleh Sebab itu, untuk mengurangi biaya perawatan memori serta menghindari adanya kerusakan memori diperlukan teknologi *cloud computing*. Kedua hal tersebut dapat dihindari karena, data data yang telah di *upload* ke server akan terjaga keamanannya apabila terjadi kerusakan pada memori local dan tidak akan mempengaruhi hilangnya data yang telah diupload ke *server*. Penggunaan teknologi *cloud computing* ini juga dapat *administrator* dapat meng *update* barang dimana saja karena konsep *cloud computing* yaitu dapat diakses oleh siapa saja yang memiliki hak untuk mengakses *cloud* tersebut.

Dalam pengembangan website tersebut, ada beberapa tahap yang dilakukan untuk meyelesaikannya yaitu dengan membuat website toko penjualan baju yang pastinya sudah bisa berjalan dengan baik tanpa ada kendala apapun. Kemudian menyiapkan server Ubuntu sebagai tempat untuk melakukan *hosting* website tersebut. Jika server Ubuntu sudah berjalan dengan normal maka artinya layanan untuk *hosting* website pun sudah dapat digunakan. Sehingga website tersebut sudah dapat diakses oleh khalayak umum, tidak hanya menjadi website yang dapat diakses oleh perseorangan saja. Pada project akhir aplikasi yang digunakan yaitu VM Workstation yang digunakan untuk menginstall Ubuntu dan LAMPP yang digunakan untuk *hosting* website.

1.2 Tujuan Proyek Akhir

Karena judul project akhir ini judulnya Sistem Informasi Toko Penjualan Baju Menggunakan Ubuntu Lampp dan Manajemen Data Penjualan Barang maka laporan ini akan menjelaskan tentang bagaimana cara kerja dari sistem ini. Dalam implementasinya sistem ini memiliki tujuan untuk mempermudah kegiatan baik input data maupun akses data, dari tujuan tersebut ada kelebihan dari sistem ini yaitu dapat mempermudah *administrator* untuk menginputkan semua data yang bersangkutan tentang jual beli baju, kemudian dari hasil *hosting* website tersebut *user* dapat mengakses tampilan dari data yang telah diinputkan sehingga mereka dapat meilihatnya kapan saja dan dimana saja.

1.3 Manfaat Proyek Akhir

Dengan dibuatnya Sistem Informasi Toko Penjualan Baju berbasis web dan menggunakan teknologi *cloud*, maka *administrator* mudah dalam menginputkan data barang serta data barang tersebut dapat terjaga keamanannya apabila data tersebut sudah di *cloud/hosting* karena keamanan data tersebut sudah tidak dipengaruhi oleh *local memory* dengan ancaman kerusakan pada hardisk. Selain itu manfaat terpenting adalah memudahkan *administrator* dalam pembuatan serta pembaharuan data barang dimana saja dan kapan saja.

Sehingga dengan adanya manfaat diatas dapat disimpulkan manfaat yang diperoleh adalah ruang dan waktu *administrator* akan lebih efisien ketika menggunakan teknologi *Cloud Computing*.

1.4 Tahap Penyelesaian Proyek Akhir

Tahapan-tahapan yang dilakukan untuk mengerjakan proyek akhir ini yaitu :

1. Melakukan analisis persoalan dan kebutuhan sistem yang akan digunakan dalam web ini.
2. Membuat website sedemikian rupa menggunakan *PHP*, *HTML*, *Mysql* dan *Source Code*.
3. Mengintegrasikan *database* yang menggunakan *xampp* dengan *database* yang menggunakan *Ubuntu server*.
4. Melakukan konfigurasi yang tepat untuk melakukan *hosting* web tersebut.
5. Melakukan *testing* web tersebut dengan *Ubuntu server* untuk mengetahui kendala web yang telah di *hosting*.

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Untuk menyelesaikan proyek ini ada beberapa alat dan bahan (komponen) yang dapat menunjang keberhasilan project ini. Beberapa komponen tersebut diantaranya yaitu lapotop dengan spesifikasi prosessor *Intel Core i5*, software seperti *VMware Workstation Pro*, *OS Ubuntu*, *LAMPP*, *Putty*, dan *WinSCP*.

Berikut adalah tabel alat dan bahan (komponen) dibutuhkan untuk mengerjakan proyek ini.

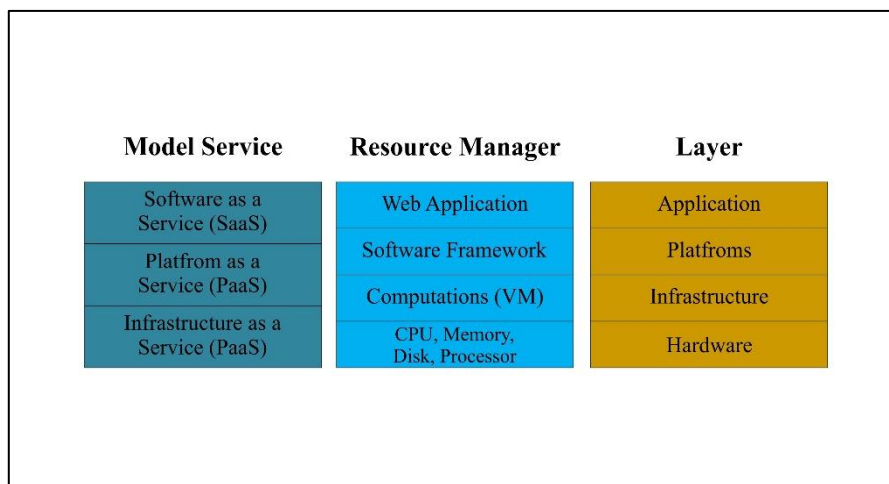
Tabel 2.1 Spesifikasi laptop yang digunakan untuk proyek pertama dan kedua

No.	Komponen	Spesifikasi
1.	Laptop	ASUS Vivobook A442UR
2.	Processor	Intel Core i5 generasi 8
3.	RAM	4 GB
4.	Harddisk	1 TB
5.	Sistem Operasi	Windows 10
6.	VMWare	VMware Workstation Pro 15.5

Penjelasan dari tabel komponen tersebut adalah sebagai berikut. Laptop yang digunakan untuk proyek ini adalah laptop merk ASUS seri *Vivobook A442UR* dengan spesifikasi *processor Intel Core i5-8250U*, RAM yang digunakan sebesar 4GB, sistem operasi yang digunakan adalah *Windows 10*, dengan *harddisk* sebesar 1 TB, dan menggunakan software *VMware Workstation Pro 15.5*.

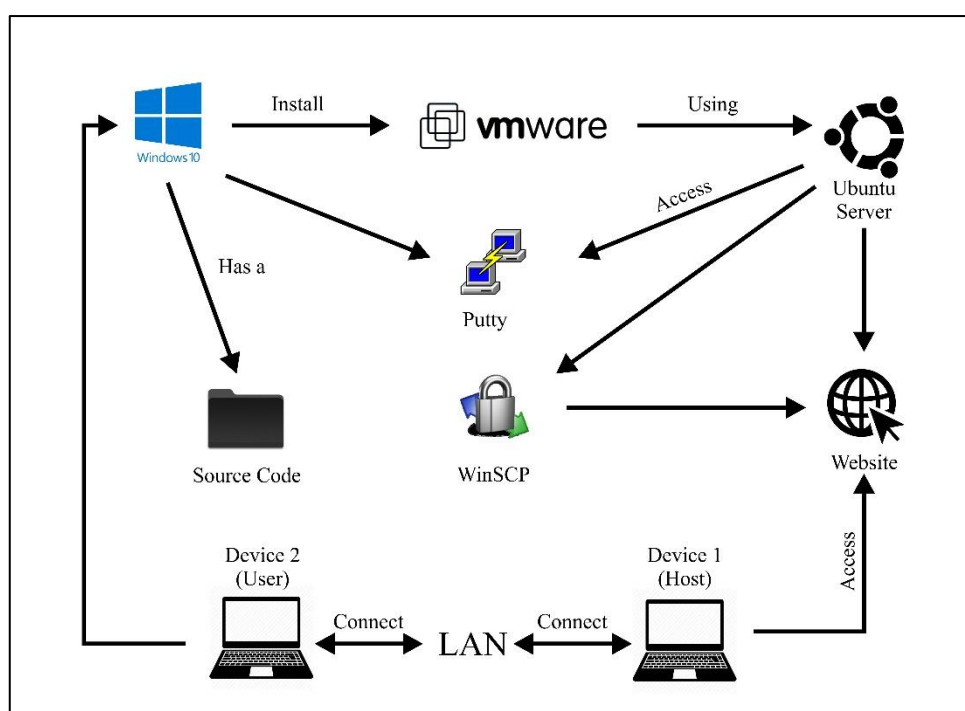
2.2 Rancangan Arsitektur Cloud Computing

Pada project akhir ini digunakan bentuk rancangan arsitektur IaaS dimana *hardware* laptop ASUS Vivobook A442UR sebagai layer utama di bagian bawah, kemudian diatasnya adalah Paas atau sistem operasi *Windows 10* dan aplikasi *VMware Workstation* yang berjalan ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.1** berikut ini:



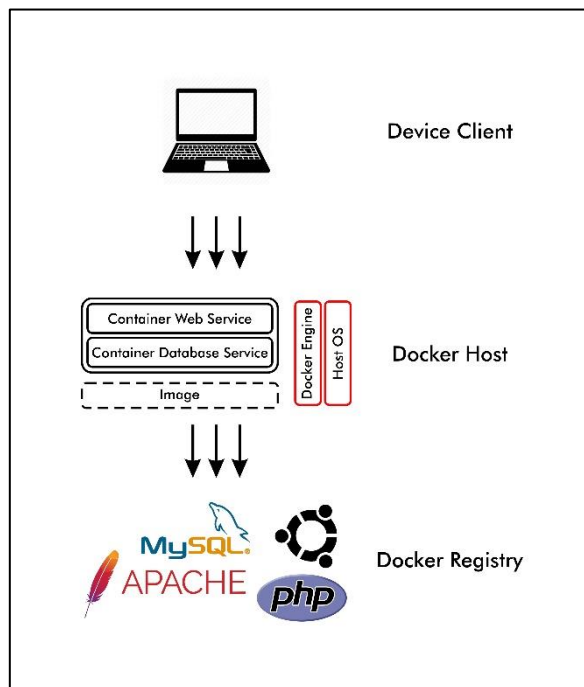
Gambar 2.1 Layer arsitektur

Pada project pertama penggunaan LAN dibutuhkan untuk menghubungkan *device 1* sebagai *host* dan *device 2* sebagai *user* agar keduanya dapat saling mengakses. *Device 1* menggunakan Windows 10 untuk melakukan instalasi *VMware Workstation*, *Putty*, dan *WinSCP*. Pada *VMware Workstation* diinstall *Ubuntu Server* yang akan digunakan oleh *Putty*. *Ubuntu Server* juga perlu menginstall *Apache*, *MySQL* dan *PHPMyadmin* yang nantinya digunakan untuk menjalankan website. *Source Code* yang digunakan dicopy ke directory yang terdapat pada *Ubuntu Server* melalui *WinSCP* yang sudah sesuai dengan *IP Address* dari *Ubuntu Server* tersebut. Ilustrasi rancangan arsitekturnya tersebut dapat dilihat pada **Gambar 2.2** berikut ini:



Gambar 2.2 Topologi Project Pertama

Pada proyek kedua menerapkan teknologi *Docker* yang akan membungkus project pertama berupa web yang sudah dibuat beserta *dependency* yang diperlukan web tersebut. Arsitekturnya dapat dilihat pada **gambar 2.3** dibawah ini:



Gambar 2.2 Topologi Project Kedua

2.3 Parameter dan Konfigurasi

Pada bagian ini, tuliskan dalam bentuk tabel dan penjelasannya mengenai isian parameter dan konfigurasi terhadap komponen alat dan bahan yang Anda gunakan. Jelaskan juga bilamana alat dan bahan yang digunakan tersebut perlu diolah (dikonfigurasi) terlebih dahulu sehingga siap pakai untuk digunakan pada tahap berikutnya (misal dikonfigurasi). Format tabel yakni: penjelasan terlebih dahulu mengenai jenis konfigurasi atau parameter, kemudian buat tabel, lalu berikan tulisan penjelasan mengenai keterangan tabel di bawah tabel bilamana diperlukan.

Agar dapat digunakan, maka *VMware Workstation* perlu dikonfigurasi terlebih dahulu dengan konfigurasi seperti pada **Tabel 2.3** berikut ini:

Tabel 2.3 Konfigurasi pada *VMware Workstation* yang digunakan

No.	Nama Parameter	Nilai	Keterangan
1.	VMware ESXi <i>version</i>	15.5.2	Keterangan versi yang digunakan.
2.	VMware ESXi <i>build</i>	15785264	Keterangan <i>build (patch) number</i> .
3.	IPv4 <i>hypervisor</i>	IP: 192.168.52.128 (Static)	IP <i>number</i> untuk akses manajemen <i>hypervisor</i> melalui <i>web based</i> .

			Pengaturan IP <i>address</i> menggunakan mode statik.
		SM: 255.255.255.0	Kelas IP/ <i>subnet mask</i> yang digunakan.
		DNS: 192.168.52.1	Alamat IP untuk fungsionalitas DNS.
		GW: 192.168.52.255	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	<i>Processor info</i>	4 x Intel(R) Core(TM) i5-8250 CPU @ 3.4GHz	Jenis <i>processor</i> yang digunakan pada <i>hypervisor</i> .
5.	<i>RAM info</i>	8 GiB Memory	Kapasitas RAM pada <i>hypervisor</i> .
6.	<i>Network info</i>	NAT	Modul jaringan pada <i>hypervisor</i> .

Parameter yang digunakan untuk instalasi *Apache* dapat dilihat pada penjelasan **Modul 2.1** berikut ini:

<pre>\$ sudo apt install apache2</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - <code>sudo</code> : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root) - <code>apt</code> : merupakan package manager pada Ubuntu - <code>install</code> : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi - <code>apache2</code> : nama paket aplikasi untuk Apache <pre>\$ sudo ufw allow in "Apache Full"</pre> <p>Keterangan: Untuk mengatur Firewall agar port 80 dan port 443 diizinkan oleh sistem Ubuntu.</p>
--

Modul 2.1 Parameter instalasi *Apache*

Parameter yang digunakan untuk instalasi *MySQL* dapat dilihat pada penjelasan **Modul 2.2** berikut ini:

<pre>\$ sudo apt install mysql-server</pre> <p>Keterangan:</p> <ul style="list-style-type: none"> - <code>sudo</code> : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root) - <code>apt</code> : merupakan package manager pada Ubuntu - <code>install</code> : parameter pada apt untuk mengeksekusi perintah instalasi paket - <code>mysql-server</code> : nama paket untuk MySQL

```
$ sudo mysql_secure_installation
```

Keterangan: Untuk mengatur keamanan pada MySQL contohnya username dan password

Modul 2.2 Parameter instalasi *MySQL*

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
- php libapache2-mod-php php-mysql : nama paket untuk PHP

```
$ sudo nano /var/www/html/info.php
```

Keterangan: Untuk membuat file baru bernama info.php yang berada di direktori var/www/html

Modul 2.3 Parameter instalasi PHP

Parameter yang digunakan untuk instalasi *PHPMyAdmin* 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
- phpMyAdmin php-mbstring php-gettext: nama paket untuk PHPMyAdmin

```
$ sudo mysql -u root
```

Keterangan: Untuk masuk ke MySQL sebagai user root.

```
mysql> UPDATE mysql.user SET plugin = 'mysql_native_password',
authentication_string = PASSWORD('123') WHERE User = 'root';
```

Keterangan: Untuk mengubah password dari user 'root' menjadi '123'

```
mysql> FLUSH PRIVILEGES;
```

Keterangan: Untuk merefresh akun yang terkoneksi dengan phpmyadmin

```
$ sudo chown vannya /var/www/html
```

Keterangan: Untuk memberikan akses ke User vannya agar dapat mengakses direktori var/www/html

Modul 2.4 Parameter instalasi *PHPMYAdmin*

Parameter yang digunakan untuk instalasi *Docker* dapat dilihat pada penjelasan

Modul 2.5 berikut ini:

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

```
$ sudo apt-get install \
```

```
apt-transport-https \
```

```
ca-certificates \
```

```
curl \
```

```
gnupg-agent \
```

```
software-properties-common
```

Keterangan: Untuk memperbarui indeks paket apt dan instal paket untuk memungkinkan apt menggunakan repositori melalui HTTPS:

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

```
sudo apt-key add -
```

```
$ sudo apt-key fingerprint 0EBFCD88
```

Keterangan: Untuk menambahkan Docker Official GPG Key

```
$ sudo add-apt-repository \
```

```
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \
```

```
$(lsb_release -cs) \
```

```
stable"
```

Keterangan: Untuk mengatur repository menjadi stable

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

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Keterangan: Untuk memperbarui indeks paket apt dan install versi terbaru Docker Engine

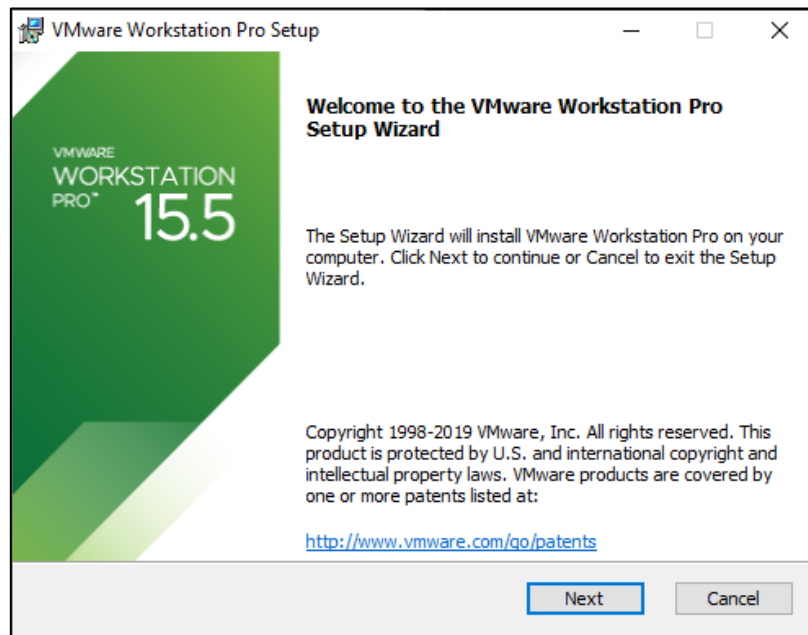
Modul 2.5 Parameter instalasi *Docker*

2.4 Tahap Implementasi

2.4.1 Tahap Implementasi Proyek Pertama

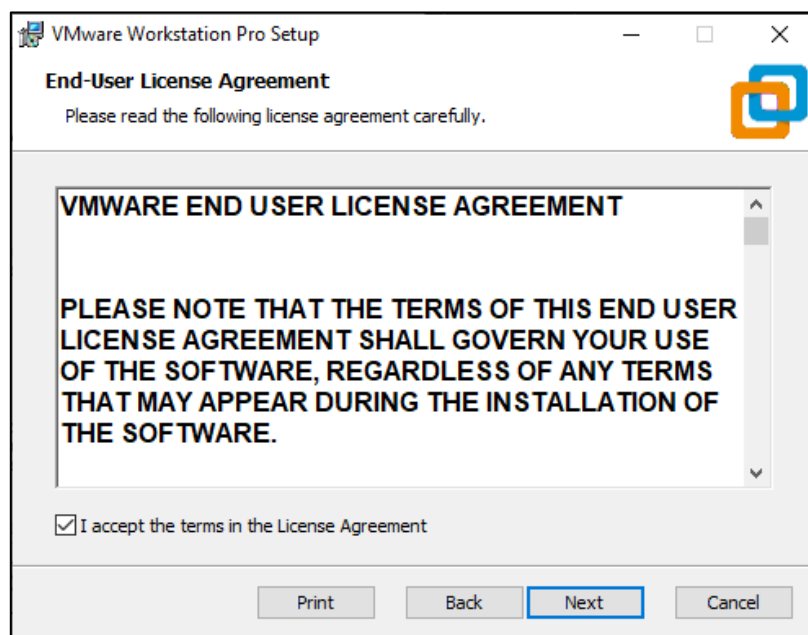
Pada bagian ini akan dijelaskan mengenai implementasi yang dilakukan untuk penyelesaian proyek ini. Ada beberapa yang harus diinstal seperti *VMware Workstation*. Untuk lebih lengkapnya, dijelaskan melalui langkah-langkah berikut :

1. Install file *setup VMware Workstation Pro 15.5* dengan cara klik *next*



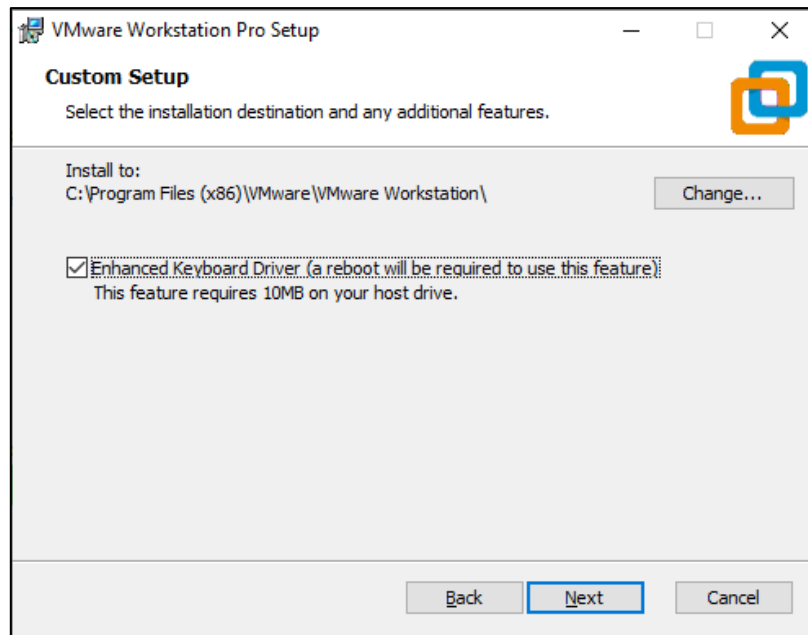
Gambar 2.4 Tampilan *setup VMware Workstation Pro*

2. Klik *next* pada tampilan *license agreement*



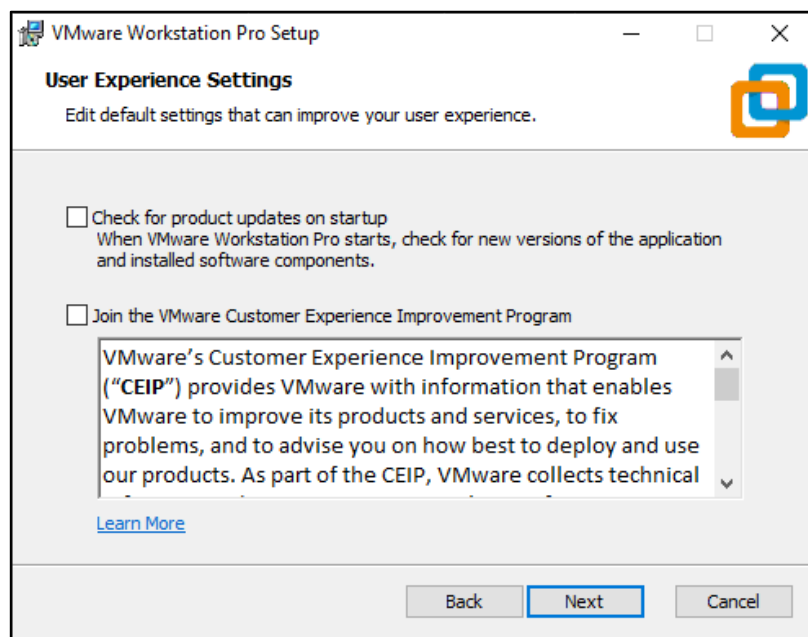
Gambar 2.5 Tampilan *licence agreement*

3. Klik *next* pada tampilan *custom setup*



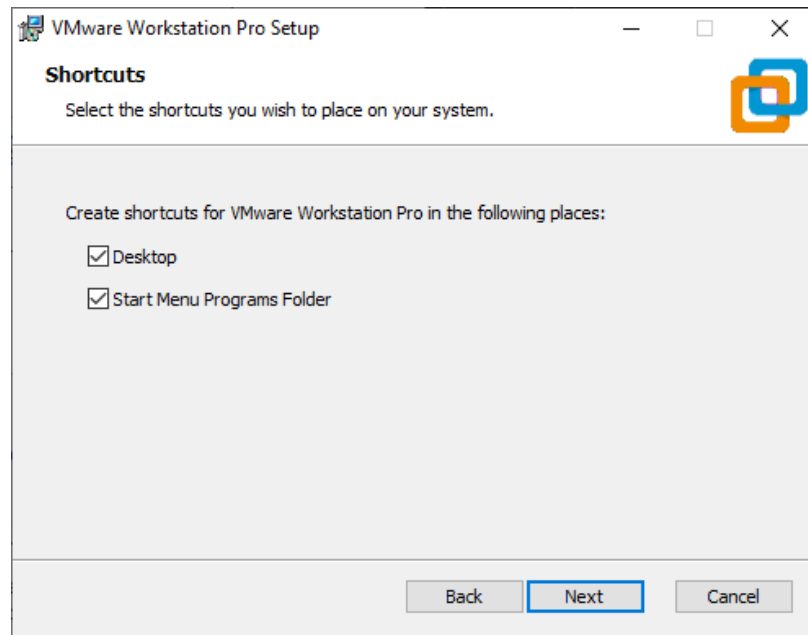
Gambar 2.6 Tampilan *custom setup*

4. Klik *next* pada tampilan *user experience settings*



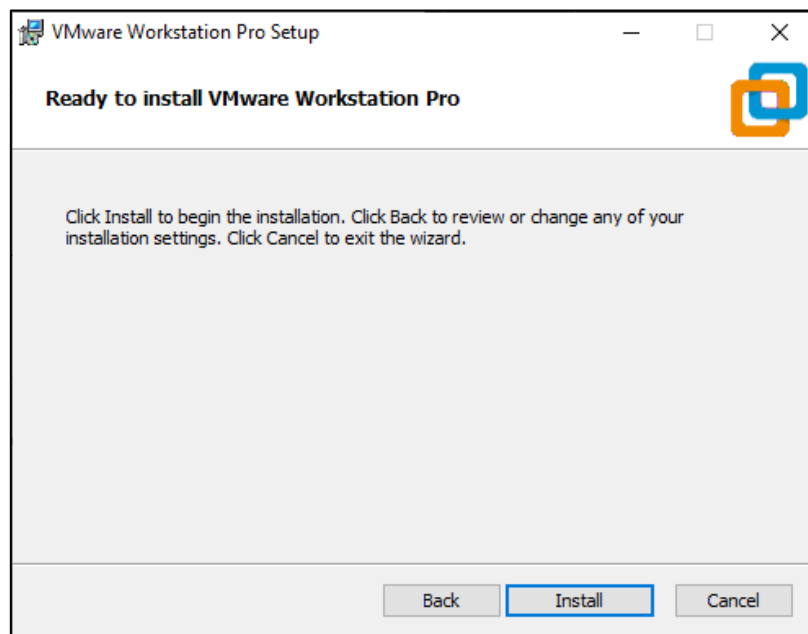
Gambar 2.7 Tampilan *user experience settings*

5. Klik *next* pada tampilan *shortcuts*



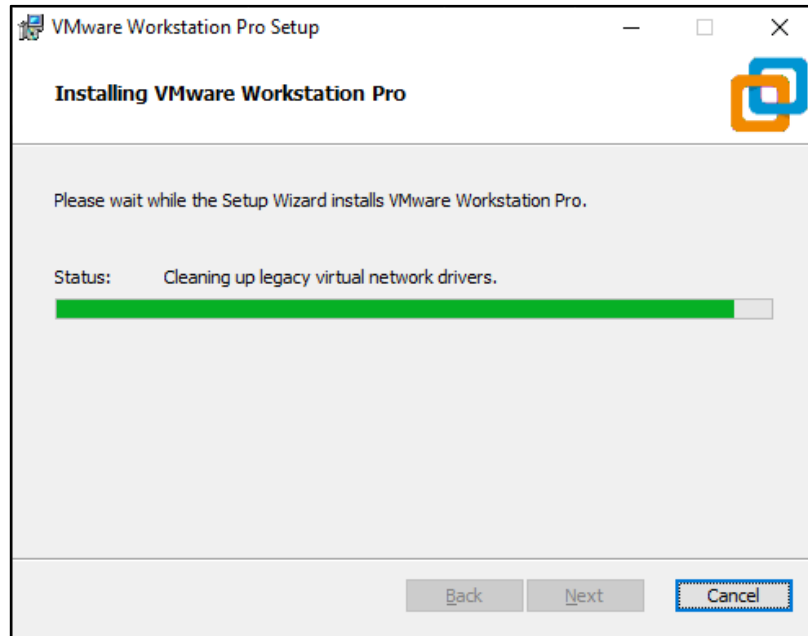
Gambar 2.8 Tampilan *shortcut*

6. Klik *next* pada tampilan *ready to install VMware Workstation Pro*



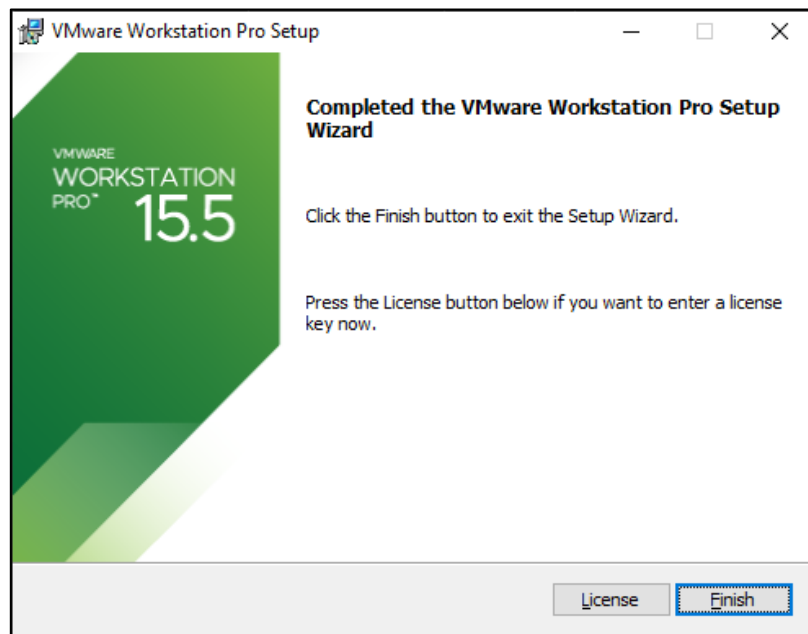
Gambar 2.9 Tampilan *ready to install VMware Workstation Pro*

7. Tunggu sampai selesai *install VMware Workstation Pro*



Gambar 2.10 Tampilan proses *install VMware Workstation Pro*

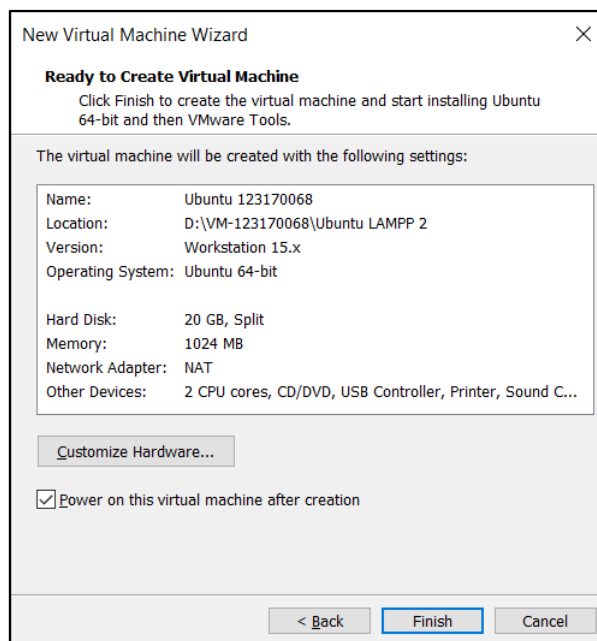
8. Klik *finish* pada tampilan *completed the VMware Workstation Pro setup*



Gambar 2.11 Tampilan ketika proses *install VMware Workstation Pro* selesai

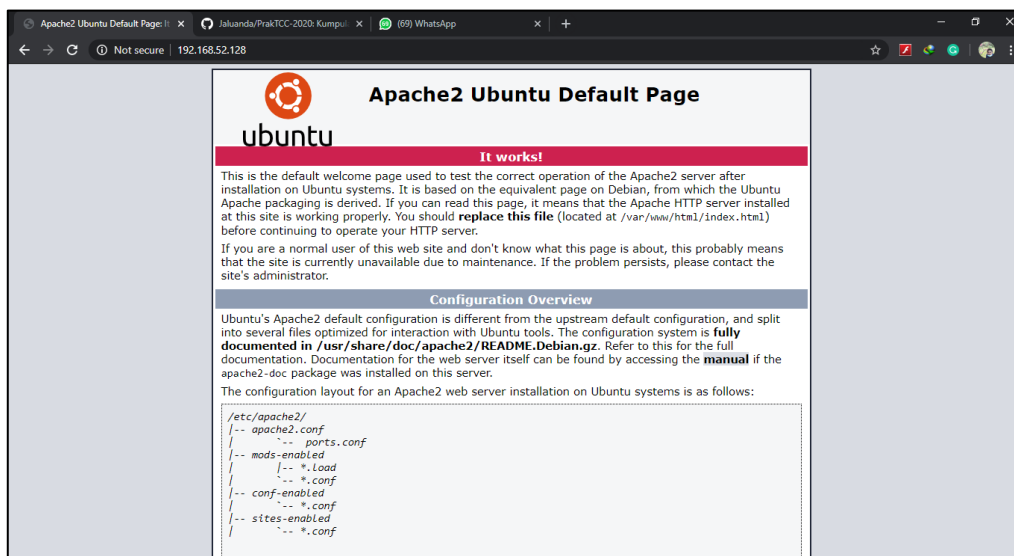
Lalu ini adalah hasil dari konfigurasi dari beberapa modul yang sudah tersedia dibagian sebelumnya.

Hasil dari implementasi **Tabel 2.3** mengenai konfigurasi untuk *Vmware Workstation* dapat dilihat pada **Gambar 2.12** berikut ini:



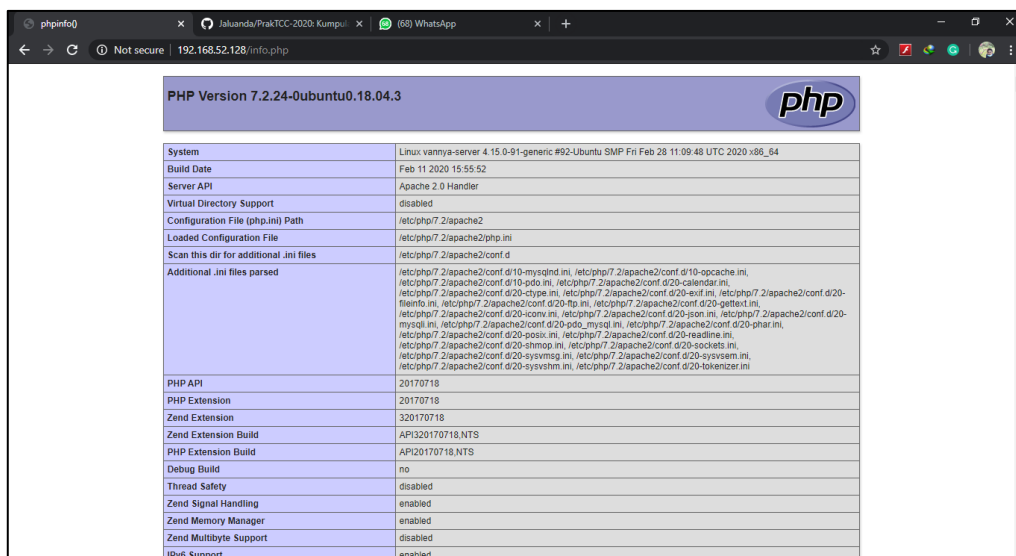
Gambar 2.12 Hasil Pembuatan *Virtual Machine* dengan *Ubuntu*

Hasil dari implementasi **Modul 2.1** mengenai cara instalasi *Apache* dapat dilihat pada **Gambar 2.13** berikut ini:



Gambar 1.13 Hasil Instalasi *Apache*

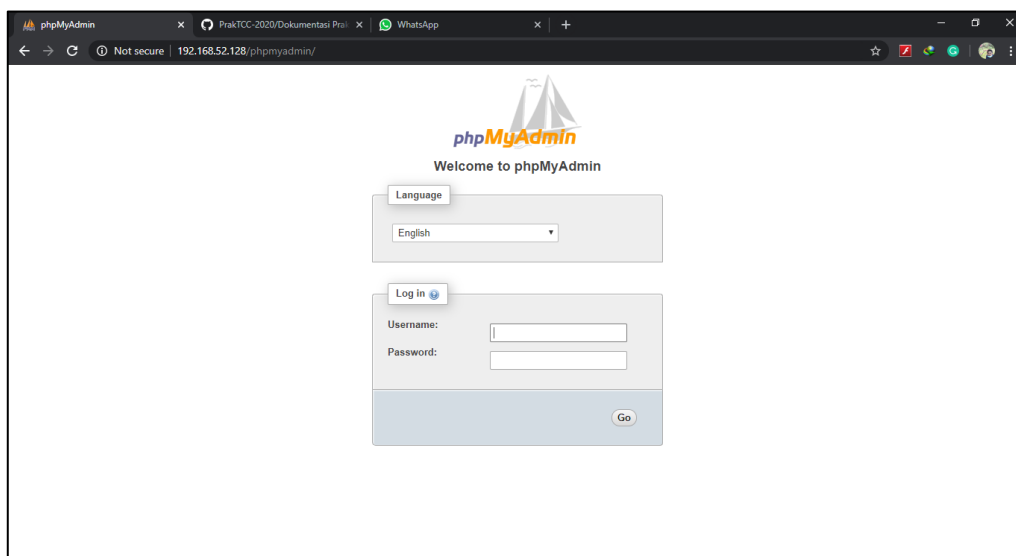
Hasil dari implementasi **Modul 2.3** mengenai cara instalasi PHP dapat dilihat pada **Gambar 2.14** berikut ini:



PHP Version 7.2.24-0ubuntu0.18.04.3	
System	Linux vmnya-server 4.15.0-91-generic #92-Ubuntu SMP Fri Feb 28 11:09:48 UTC 2020 x86_64
Build Date	Feb 11 2020 15:55:52
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqld.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-enf.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysql.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysmsg.ini, /etc/php/7.2/apache2/conf.d/20-syssem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled

Gambar 2.14 Hasil Instalasi PHP

Hasil dari implementasi **Modul 2.4** mengenai cara instalasi *PHPMYAdmin* dapat dilihat pada **Gambar 2.15** berikut ini:



phpMyAdmin

Welcome to phpMyAdmin

Language

English

Log in

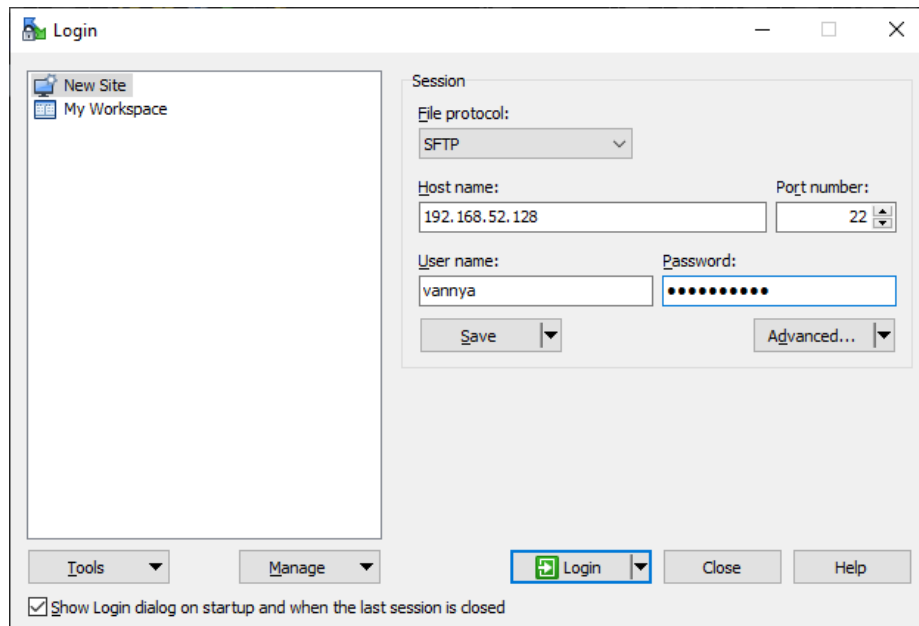
Username:

Password:

Go

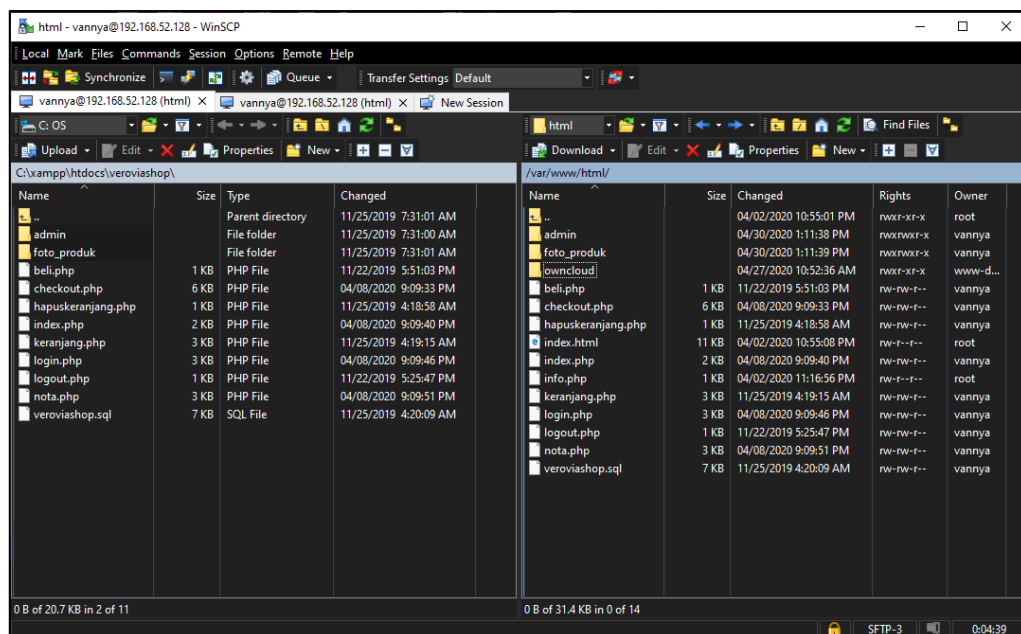
Gambar 2.15 Hasil Instalasi *PHPMYAdmin*

Setelah melakukan instalasi *LAMPP* maka langkah selanjutnya adalah melakukan *hosting* web yang telah kita buat sebelumnya. Caranya dengan membuka aplikasi *WinSCP* lalu *login* dengan memasukkan *IP address* yang kita miliki ke dalam *hostname* dan memasukan *username* dan *password* yang sesuai seperti **Gambar 2.16** dibawah ini.



Gambar 2.16 Proses login ke WinSCP

Kemudian lakukan pemindahan *source code* web yang telah kita buat sebelumnya ke direktori `var/www/html` seperti **Gambar 2.17** dibawah ini



Gambar 2.17 Proses pemindahan *Source Code*

2.4.2 Tahap Implementasi Proyek Kedua

Hasil dari implementasi **Modul 2.5** tentang instalasi *Docker* dapat dilihat pada **Gambar 2.18** berikut ini:

```
vannya@vannya-server:~$ docker -v
Docker version 19.03.8, build afacb8b7f0
```

Gambar 2.18 Hasil instalasi *Docker*

Kemudian kita implementasi *source code* berikut dapat dilihat pada **Gambar 2.19** berikut ini:

```
vannya@vannya-server: ~
vannya@vannya-server:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:8e3114318a995a1ee497790535e7b8836522a21771ae7e53687ad76563e8e76
Status: Downloaded newer image for hello-world:latest

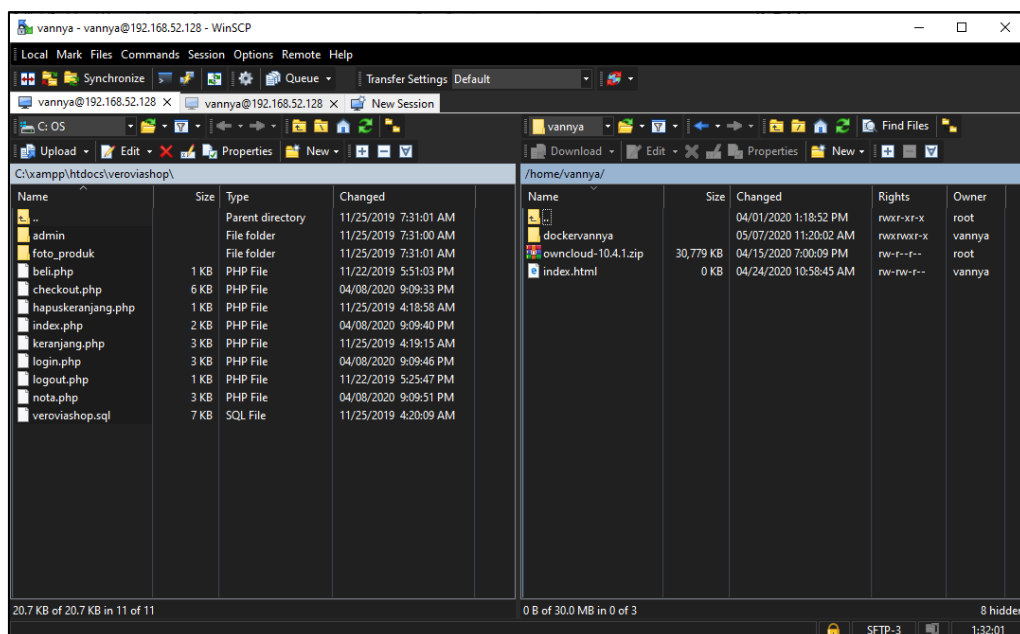
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

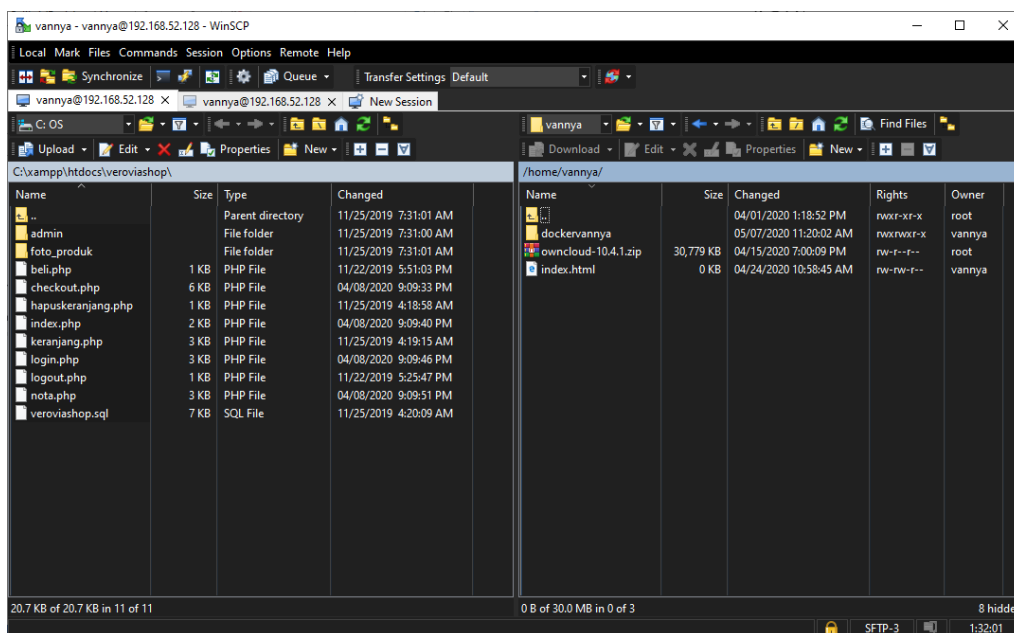
To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash
```

Gambar 2.19 Hasil *run sourcecode hello-world*

Langkah selanjutnya yaitu membuat *folder* baru bernama *dockervannya* dibagian */home/vannya* untuk menampung *source code* web yang akan digunakan pada *docker* ini dengan menggunakan *WinSCP* dapat dilihat pada **Gambar 2.20** dan **Gambar 2.21** berikut ini:

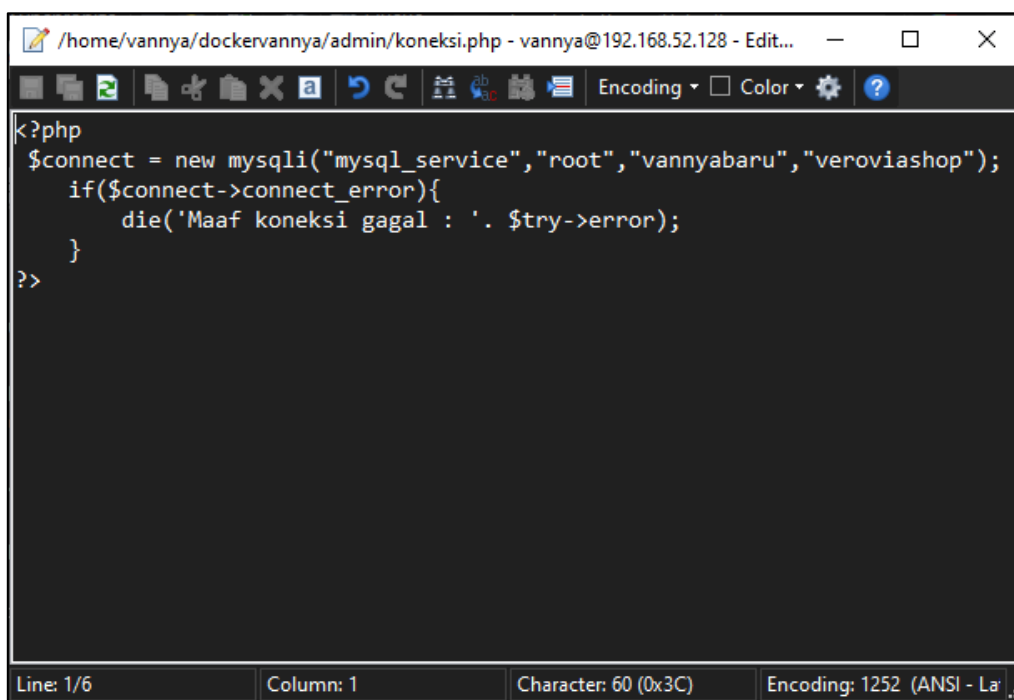


Gambar 2.20 Pembuatan folder baru dibagian /home/vannya



Gambar 2.21 Menambahkan *source code* pada folder baru

Langkah selanjutnya yaitu mengubah *source code* koneksi.php dapat dilihat pada **Gambar 2.22** berikut ini:



Gambar 2.22 Mengubah *source code* koneksi.php

Langkah selanjutnya yaitu masuk ke direktori dockervannya dan lakukan *source code* yang dapat dilihat pada **Gambar 2.23** berikut ini:

```
vannya@vannya-server:~$ cd dockervannya
vannya@vannya-server:~/dockervannya$ touch Dockerfile
vannya@vannya-server:~/dockervannya$
```

Gambar 2.23 Masuk ke direktori dockervannya dan lakukan *touch Dockerfile*

Langkah selanjutnya yaitu menjalankan *source code* yang dapat dilihat pada **Gambar 2.24** dan **Gambar 2.25** berikut ini:

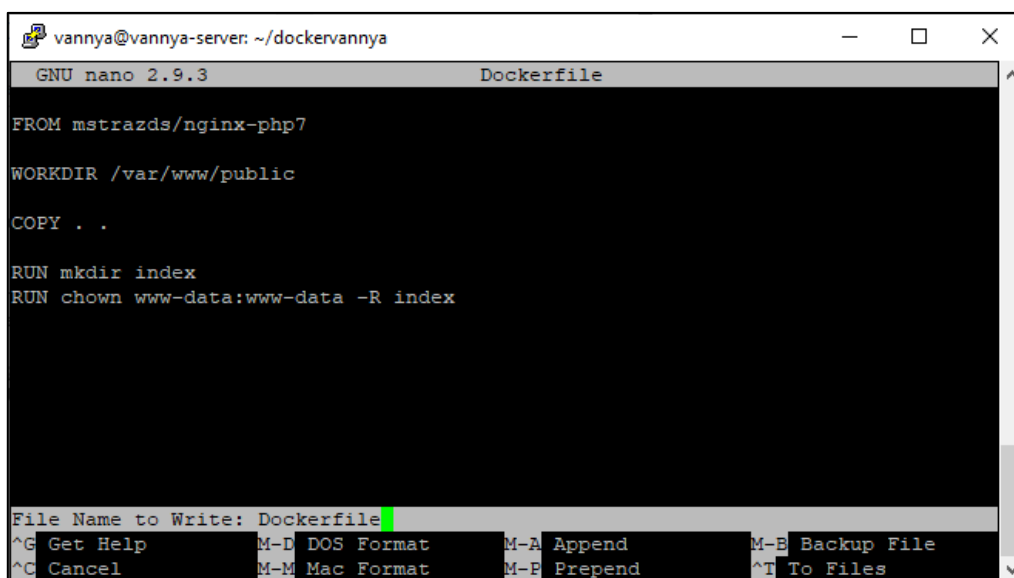
```
vannya@vannya-server:~/dockervannya$ ls -la
total 60
drwxrwxr-x 4 vannya vannya 4096 May  7 06:00 .
drwxr-xr-x 5 vannya vannya 4096 May  7 04:58 ..
drwxrwxr-x 3 vannya vannya 4096 May  7 04:20 admin
-rw-rw-r-- 1 vannya vannya  513 Nov 22 10:51 beli.php
-rw-rw-r-- 1 vannya vannya 5125 Apr  8 14:09 checkout.php
-rw-rw-r-- 1 vannya vannya    0 May  7 06:00 Dockerfile
drwxrwxr-x 2 vannya vannya 4096 May  7 04:20 foto_produk
-rw-rw-r-- 1 vannya vannya  302 Nov 24 21:18 hapuskeranjang.php
-rw-rw-r-- 1 vannya vannya 1547 Apr  8 14:09 index.php
-rw-rw-r-- 1 vannya vannya 2423 Nov 24 21:19 keranjang.php
-rw-rw-r-- 1 vannya vannya 2397 Apr  8 14:09 login.php
-rw-rw-r-- 1 vannya vannya  194 Nov 22 10:25 logout.php
-rw-rw-r-- 1 vannya vannya 2612 Apr  8 14:09 nota.php
-rw-rw-r-- 1 vannya vannya 6169 Nov 24 21:20 veroviashop.sql
vannya@vannya-server:~/dockervannya$
```

Gambar 2.24 Menjalankan *source code ls -la*

```
vannya@vannya-server:~/dockervannya$ nano Dockerfile
vannya@vannya-server:~/dockervannya$
```

Gambar 2.25 Menjalankan *nano Dockerfile*

Langkah selanjutnya yaitu menambahkan *source code* ke dalam *nano Dockerfile* yang dapat dilihat pada **Gambar 2.26** berikut ini:



```
vannya@vannya-server: ~/dockervannya
GNU nano 2.9.3 Dockerfile

FROM mstrazds/nginx-php7

WORKDIR /var/www/public

COPY . .

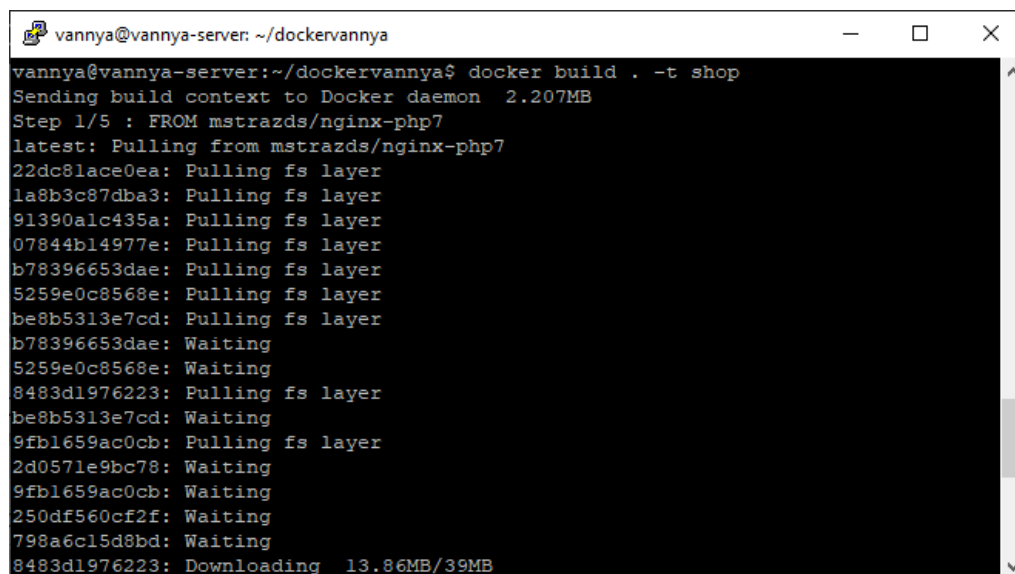
RUN mkdir index
RUN chown www-data:www-data -R index

File Name to Write: Dockerfile
^G Get Help      M-D DOS Format  M-A Append      M-B Backup File
^C Cancel        M-M Mac Format  M-P Prepend     ^T To Files
```

Gambar 2.26 Menambahkan *source code*

Setelah mengetikkan *source code* tersebut maka kita simpan dengan `ctrl+O` lalu tekan `enter` lalu kita keluar dari *nano Dockerfile* dengan `ctrl+X`. Kemudian kita membuat

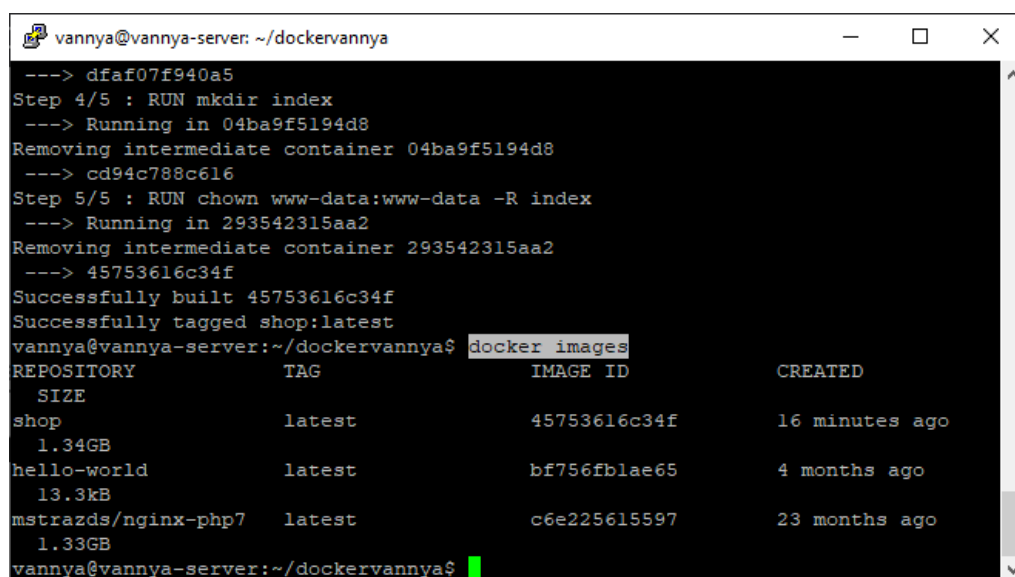
repository baru ke dalam *docker image* dengan nama *shop* dapat dilihat pada **Gambar 2.27** berikut ini:



```
vannya@vannya-server: ~/dockervannya
vannya@vannya-server:~/dockervannya$ docker build . -t shop
Sending build context to Docker daemon 2.207MB
Step 1/5 : FROM mstrazds/nginx-php7
latest: Pulling from mstrazds/nginx-php7
22dc81ace0ea: Pulling fs layer
1a8b3c87dba3: Pulling fs layer
91390alc435a: Pulling fs layer
07844b14977e: Pulling fs layer
b78396653dae: Pulling fs layer
5259e0c8568e: Pulling fs layer
be8b5313e7cd: Pulling fs layer
b78396653dae: Waiting
5259e0c8568e: Waiting
8483d1976223: Pulling fs layer
be8b5313e7cd: Waiting
9fb1659ac0cb: Pulling fs layer
2d0571e9bc78: Waiting
9fb1659ac0cb: Waiting
250df560cf2f: Waiting
798a6c15d8bd: Waiting
8483d1976223: Downloading 13.86MB/39MB
```

Gambar 2.27 Membuat *repository* baru

Hasil dari pembuatan *repository* baru dapat dilihat dengan mengetikkan *source code docker image* dapat dilihat pada **Gambar 2.28** berikut ini:



```
vannya@vannya-server: ~/dockervannya
---> dfaf07f940a5
Step 4/5 : RUN mkdir index
---> Running in 04ba9f5194d8
Removing intermediate container 04ba9f5194d8
---> cd94c788c616
Step 5/5 : RUN chown www-data:www-data -R index
---> Running in 293542315aa2
Removing intermediate container 293542315aa2
---> 45753616c34f
Successfully built 45753616c34f
Successfully tagged shop:latest
vannya@vannya-server:~/dockervannya$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
shop                 latest             45753616c34f       16 minutes ago
hello-world          latest             bf756fblae65       4 months ago
mstrazds/nginx-php7 latest             c6e225615597       23 months ago
vannya@vannya-server:~/dockervannya$
```

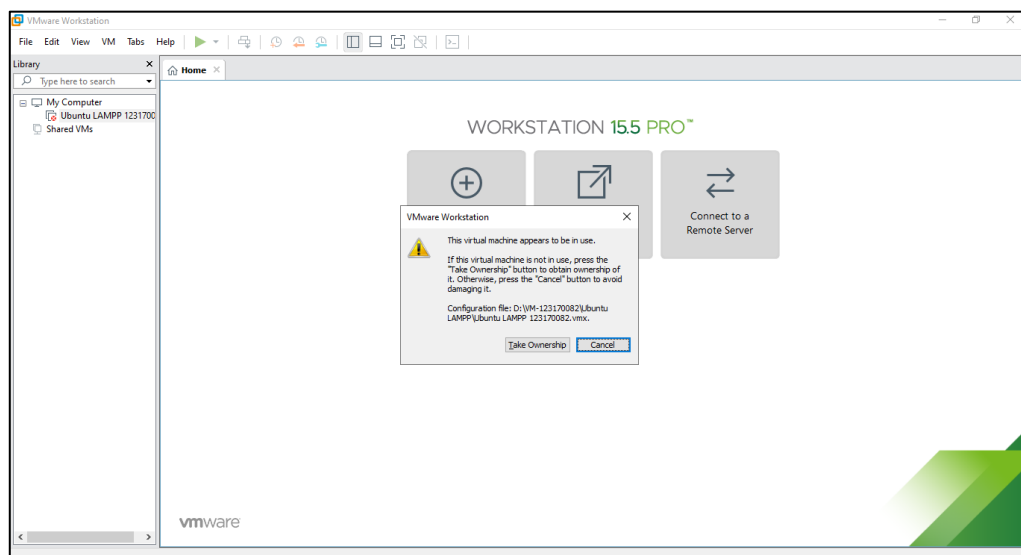
Gambar 2.28 Membuat *repository* baru

Selanjutnya yaitu memasukkan dan menyetarakan *MySQL* yang akan digunakan. dapat dilihat pada **Gambar 2.29** berikut ini:


```
vannya@vannya-server:~/dockervannya$ docker pull mysql:8.0
8.0: Pulling from library/mysql
afb6ec6fdbc1c: Downloading 22.35MB/27.1MB
0bdc5971ba40: Download complete
97ae94a2c729: Download complete
f777521d340e: Download complete
1393ff7fc871: Download complete
a499b89994d9: Download complete
7ebe8eefbaf6: Download complete
597069368ef1: Download complete
ce39a5501878: Downloading 3.75MB/111.5MB
7d545bca14bf: Download complete
0f5f78cccacb: Waiting
623a5dae2b42: Waiting
```

Gambar 2.29 Penyetaraan *MySQL* yang akan digunakan

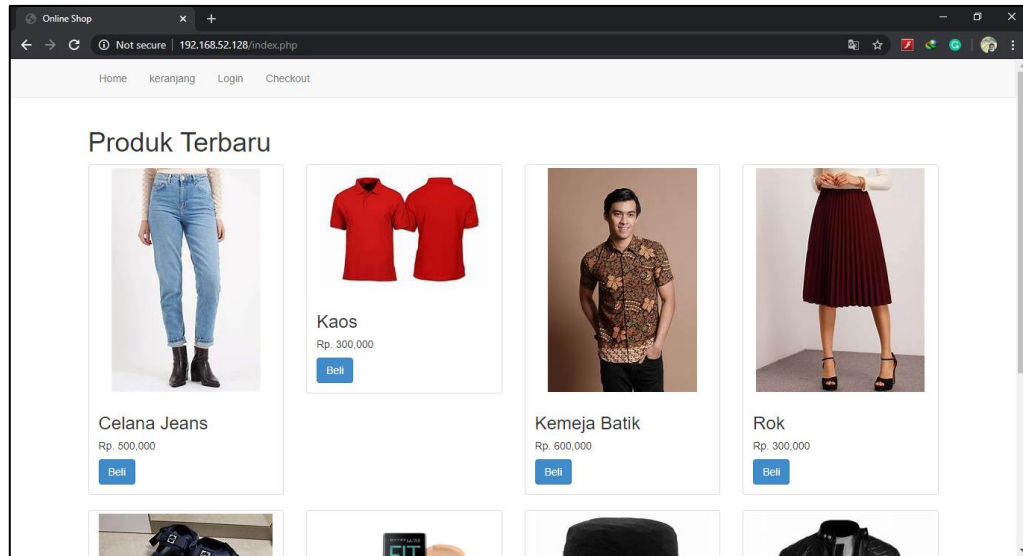
Seharusnya masih ada beberapa langkah lagi namun terdapat kendala pada *VMware Workstation* yang digunakan dan kami tidak menemukan cara megatasinya sehingga pengerjaan project akhir terhenti sampai disini. Kendala yang ada dapat dilihat pada **Gambar 2.30** berikut ini:



Gambar 2.30 Kendala pada *VMware Workststion* yang digunakan

2.5 Hasil Implementasi

Setelah konfigurasi dan instalasi selesai, berikut merupakan hasil dari *hosting* website menggunakan *LAMP* pada *Ubuntu* :



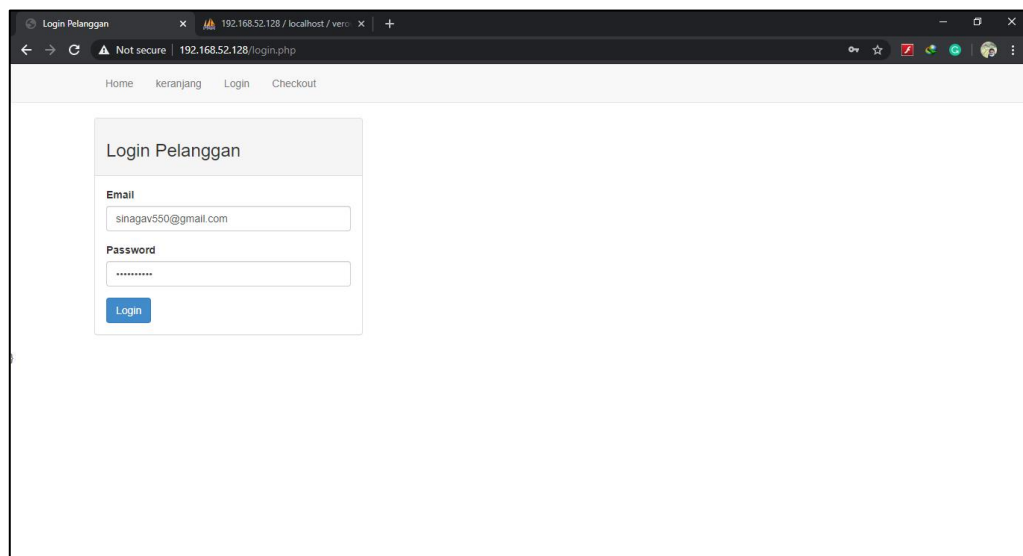
Gambar 2.28 Halaman awal web yang di *hosting*

Dari tampilan diatas dapat dilihat bahwa proses *hosting* berhasil dilakukan.

2.6 Pengujian Singkat

Pada bagian ini akan dijelaskan hasil dari apa yang sudah dikerjakan dalam tahap-tahap sebelumnya, untuk mengetahui apakah file yang di *hosting* sudah dapat berjalan normal, atau malah memiliki Kendala atau file yang di *hosting* dapat dijalankan namun, terdapat beberapa fitur yang tidak berjalan secara normal. Pada **Gambar 2.29** dan **Gambar 2.30** berikut ini merupakan tampilan untuk *login* pelanggan dan tampilan keranjang belanja.

Untuk proses penaksesan web, *user* akan diminta untuk melakukan *login* untuk dapat melakukan proses selanjutnya yaitu proses *checkout*.



Gambar 2.29 Halaman web sudah dapat diakses

Berikut merupakan tampilan dari proses *checkout* dimana tampilan ini merupakan proses yang telah diakumulasi pada *database*.

No	Produk	Harga	Jumlah	Subharga
1	Jaket Kulit	Rp. 2.000.000	1	Rp. 2.000.000
2	Kaos	Rp. 300.000	1	Rp. 300.000
3	Rok	Rp. 300.000	1	Rp. 300.000
Total Belanja				Rp. 2.600.000

vero 08528022 bandung - Rp. 12,000

Checkout

```
Array
(
    [id_pelanggan] => 2
    [email_pelanggan] => sinagav55@gmail.com
    [password_pelanggan] => veronika25
    [nama_pelanggan] => vero
    [telepon_pelanggan] => 08528022
)
```

Gambar 2.30 Halaman web sudah dapat melakukan proses *checkout*

Kedua tampilan gambar diatas merupakan hasil dari pengujian *hosting* website menggunakan *LAMP* pada *Ubuntu* dan dapat dilihat bahwa proses tersebut berhasil dilakukan.

Kesimpulan dari percobaan diatas adalah semakin banyaknya layanan berbasis *cloud* membuat banyak sistem beralih dari *offline* ke *online*. Sistem *offline* membutuhkan bantuan untuk dapat diakses secara *online*, demikian juga dengan pengujian diatas. Dimana web tersebut dapat diakses dimana saja namun harus terdapat koneksi internet. Hal tersebut bertujuan untuk mempermudah pengaksesan sehingga dapat web tersebut diakses dimana saja dan kapan saja.

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 dan April tahun 2020:

Tabel 3.1 Agenda Pengerjaan Proyek

No.	Jenis Tugas	Waktu Pengerjaan									
		Maret				April				Mei	
		1	2	3	4	1	2	3	4	1	2
1.	Analisa Persoalan										
2.	Pembagian Tugas										
3.	Pengerjaan Tugas 1 (LAMPP)										
4.	Pembuatan Laporan Submisi										
5.	Pengerjaan Tugas 2 (Docker)										
6.	Melanjutkan pembuatan laporan										
7.	Revisi										
8.	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	Via
2.	Latar Belakang Masalah	Via
3.	Agenda Pengerjaan Proyek	Via
4.	Pembuatan Tugas 1 (LAMPP)	Vannya
5.	Pembuatan Tugas 2 (Docker)	Vannya
6.	Pengujian Singkat	Vannya
7.	Pembuatan Laporan	Vannya dan Via

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan pada project akhir ini terdapat dua tugas. Pada tugas pertama yaitu melakukan hosting menggunakan Ubuntu LAMPP berjalan dengan baik, yaitu berupa tampilan maupun *database*. Keduanya dapat dikerjakan dengan hasil sesuai yang diharapkan yaitu pengerjaan sukses tanpa adanya *error*. Pada tugas kedua yaitu pembuatan *dockerfile*, proses penginstalan *docker* berhasil dilakukan namun pada saat pengerjaan selanjutnya terdapat *error* (gambar terlampir pada lampiran) dan ketika diketikan kodingan apapun terdapat pesan *error* kembali berupa “*structure need cleaning*”. Pada saat program dicoba untuk dijalankan kembali juga terdapat *error* berupa laptop yang tiba-tiba mati sendiri. Kejadian tersebut terjadi terus menerus.

4.2 Saran

Saran yang sebaiknya pemberian contoh pengerjaan dilakukan secara virtualisasi, karena kami rasa akan lebih mudah dipahami ketika diberikan contoh berupa video sehingga pengimplementasian pada masing-masing program dapat lebih mudah dan efektif. Pada saat pemilihan project, praktikan dibebaskan untuk memilih project nya masing-masing sehingga kami rasa untuk pemerataan tingkat kesulitan dari masing-masing program kurang sama.

DAFTAR PUSTAKA

Adiputra, F. (2015). Container dan Docker: Teknik Virtualisasi dalam Pengelolaan Banyak Aplikasi Web. *Jurnal SimanteC*, 4(3).

Mutia, I. (2016). Pemanfaatan Komputasi Awan (Cloud Computing) Bagi Pembelajaran Mahasiswa Perguruan Tinggi. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 1(1), 1–9. <https://doi.org/10.30998/string.v1i1.963>

<https://lawavedesign.com/berita/33/pengertian-manfaat-cara-kerja-dan-contoh-cloud-computing.html>

<https://blog.wowrack.co.id/2018/12/bagaimana-cloud-computing-membantu.html>

<https://docs.docker.com/engine/install/ubuntu/>

<https://medium.com/@denmasyarikin/dasar-penggunaan-docker-f80d9602186c>

LAMPIRAN

```
vannya@vannya-server:~/dockervannya$ docker run --name=mysql_service -e MYSQL_ROOT_PASSWORD=mysql -d mysql:8.0
319bc875e70713764d3febf15d01d9ed381a97a144f53318588d5f30bb692aad
docker: Error response from daemon: structure needs cleaning: unknown.
vannya@vannya-server:~/dockervannya$
```