

SISTEM INFORMASI PENYEWAAN RUMAH KONTRAKAN & KOST BERBASIS WEB MENGGUNAKAN UBUNTU LAMPP DAN DOCKER



NAMA ANGGOTA : ARMAN RIZKY MAULANA 123170016
DZAKY MUHAMMAD IQBAL 123170085
KELAS : C
ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.Kom.
WAHYU AJI NUGROHO, S.Kom.

HALAMAN PENGESAHAN

SISTEM INFORMASI PENYEWAAN RUMAH KONTRAKAN & KOST BERBASIS WEB MENGGUNAKAN UBUNTU LAMPP DAN DOCKER

Disusun oleh :

Arman Rizky Maulana

123170016

Dzaky Muhammad Iqbal

123170085

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 Penyewaan Rumah Kontrakan & Kost Berbasis Web Menggunakan Ubuntu Lampp dan Docker. Adapun laporan ini berisi tentang proyek akhir yang salah satunya bersifat wajib dan pilihan yang kami pilih dari hasil pembelajaran selama praktikum berlangsung.

Tidak lupa ucapan terimakasih kepada asisten laboratorium yang selalu membimbing dan mengajari kami dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun kami harapkan untuk menyempurnakan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan ini, kami ucapkan terimakasih. Semoga laporan ini dapat dipergunakan seperlunya.

Yogyakarta, 31 Maret 2020

Penyusun

DAFTAR ISI

HALAMAN PENGESAHAN	i
KATA PENGANTAR	ii
DAFTAR ISI	i
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Tujuan Proyek Akhir.....	3
1.3. Manfaat Proyek Akhir.....	3
1.4. Tahap Penyelesaian Proyek Akhir.....	4
BAB II ISI DAN PEMBAHASAN.....	5
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.....	11
2.5. Hasil Implementasi	18
2.6. Pengujian Singkat.....	19
BAB III JADWAL Pengerjaan dan Pembagian Tugas.....	22
3.1. Agenda Pengerjaan	22
3.2. Keterangan Pembagian Tugas	22
BAB IV KESIMPULAN DAN SARAN.....	23
4.1. Kesimpulan.....	23
4.2. Saran	23
DAFTAR PUSTAKA	24
LAMPIRAN	25

BAB I

PENDAHULUAN

1.1. Latar Belakang Proyek Akhir

Komputasi awan memungkinkan pengguna untuk menggunakan aplikasi tanpa instalasi dan mengakses file pribadi mereka di setiap device dengan akses Internet. Penggunaan internet yang begitu pesat, membuat internet kurang dioptimalkan oleh pengguna karena melalui internet juga dapat digunakan oleh pengguna untuk menyimpan data pribadi yang dengan mudah dapat diakses kapan saja dan di mana saja oleh pengguna tersebut. Hal inilah yang menjadi salah satu latar belakang lahirnya teknologi cloud computing.

Dengan adanya teknologi ini maka pengguna dapat bebas bergerak (mobile) dengan data yang dapat diakses kapan saja dan di mana saja. Segala komputasi terjadi di awan (jaringan internet) dan menjadi infrastruktur public. Teknologi Komputasi Awan ini juga tidak memerlukan berbagai software, mulai dari Sistem Operasi dan berbagai software lain dalam device pengguna. Device yang digunakan oleh pengguna pun tidak perlu memiliki spesifikasi yang bagus seperti memory maupun media penyimpanan (harddisk) yang besar serta processor yang berkecepatan tinggi. Pada device yang digunakan oleh pengguna hanya memerlukan suatu web browser (misalnya, Internet explorer, Opera, Mozilla, dll) dengan suatu koneksi akses ke jaringan internet untuk dapat menggunakan teknologi ini. Dengan adanya cloud computing juga meminimalisasi spesifikasi system pada device (komputer/notebook) pengguna. Cloud computing menawarkan peluang dan tantangan baru bagi praktisi ICT. Dibeberapa kasus, keahlian yang dimiliki praktisi ICT dapat langsung digunakan dalam penerapan teknologi cloud, dan dalam beberapa kasus yang lain perlu dikembangkan [10]. Dituntut peningkatkan kemampuan beradaptasi sesuai standar di lingkungan cloud computing. Beberapa perusahaan besar telah mengeksplorasi teknologi cloud computing untuk dijadikan bisnis seperti Microsoft, IBM, Google dan Amazon dengan menginvestasikan jutaan dolar untuk hal tersebut. Perusahaan tersebut menawarkan layanan cloud, sehingga sebagian besar tugas pemantauan, konfigurasi, integrasi, pembelian infrastruktur perangkat keras dan perangkat lunak tidak lagi dibutuhkan karena akan ditangani oleh penyedia cloud. Dengan cloud computing, tidak ada lagi koneksi point-to-point antara pengguna dan infrastruktur komputasi (Desa & Lawang, 2009). Pendekatan ini memudahkan software developer untuk membuat aplikasi yang berkomunikasi dengan berbagai library tanpa harus memikirkan kembali strategi yang digunakan selama semua

library mengikut API yang sama. Kelebihan lain dari metode ini menunjukkan betapa mudahnya menggunakan library yang sama dengan Bahasa pemrograman yang berbeda. Sistem informasi penyewaan rumah kontrakan dan kost adalah sebuah sistem informasi berbasis website yang berfungsi untuk menyediakan informasi tentang kontrakan/kostan buat orang-orang yang merantau dan masyarakat umum seperti mahasiswa, pasutri dan lain-lain. Selain menyediakan informasi, website ini juga dapat mempromosikan rumah kontakn atau kost jika ada orang yang ingin rumah kontrakan dan kostnya dipromosikan. Website ini sangat lengkap infomasinya seperti : nama pemilik, harga, fasilitas, alamat dan masih banyak lagi .

Dari beberapa penelitian yang telah dijabarkan, masih terdapat beberapa kekurangan, diantaranya adalah penggunaan algoritma yang memiliki keterbatasan pada proses deteksi, yaitu hanya mampu mendeteksi satu objek pada satu proses deteksi. Kekurangan lainnya adalah penggunaan teknik *preprocessing* yang masih sederhana, sehingga sistem belum mampu menghasilkan representasi objek dengan baik yang berpengaruh pada ketepatan proses deteksi. Sehingga dari permasalahan yang telah dijabarkan, solusi yang diusulkan pada penelitian ini adalah penggunaan *multi-frame detection* untuk proses deteksi kendaraan sehingga mampu mendeteksi kendaraan dengan jumlah yang lebih banyak serta dapat mengetahui kecepatan yang dihasilkan dari kendaraan yang melintasi area pemantauan, dan penggunaan *cascaded mask* untuk melakukan proses *filtering* citra sehingga mampu menghasilkan data citra yang baik untuk berbagai kondisi pencahayaan yang berbeda.

Pada penelitian sebelumnya yaitu mengenai penerapan teknologi *cloud computing* di universitas. Banyak manfaat dari penerapan cloud computing di Universitas. Manfaat ini dirasakan langsung oleh semua pihak yang terlibat dalam lingkungan universitas seperti mahasiswa, dosen, staf IT, staf administratif, hingga jajaran direksi universitas. Namun tidak semua pihak mendapatkan manfaat yang sama dari *cloud computing*, ada yang merasakan manfaat dari sisi peningkatan kualitas media belajar dan mengajar, ada juga yang mendapatkan manfaat dari sisi ekonomi seperti penghematan sumber daya, dan ada yang mendapatkan manfaat dari sisi kemudahan perawatan aplikasi dan infrastruktur(Kurniawan, 2015). Kemudian untuk system kerja dari *cloud computing* sendiri terdapat 2 macam . Sistem cloud computing terbagi menjadi dua bagian: ujung depan dan ujung belakang. Mereka terhubung satu sama lain melalui jaringan, biasanya adalah internet. Ujung depan adalah sisi pengguna komputer (user), atau klien (client), dan bagian belakang adalah

“cloud” bagian dari sistem. Ujung depan termasuk komputer klien (atau jaringan komputer) dan aplikasi yang diperlukan untuk mengakses sistem komputasi awan. Tidak semua sistem komputasi awan memiliki antarmuka pengguna yang sama. Layanan seperti Web-based e- mail program memanfaatkan browser Web yang ada seperti Internet Explorer atau Firefox. Sistem lain memiliki aplikasi unik yang menyediakan akses jaringan untuk klien. Di ujung belakang sistem adalah berbagai komputer, server dan sistem penyimpanan data yang menciptakan “cloud” dari layanan komputasi. Biasanya setiap aplikasi akan memiliki server khususnya sendiri. Sebuah server pusat mengelola sistem, memantau lalu lintas dan permintaan client untuk memastikan semuanya berjalan lancar. Sistem ini mengikuti seperangkat aturan yang disebut protokol dan menggunakan jenis khusus dari perangkat lunak yang disebut middleware. Middleware network memungkinkan komputer untuk berkomunikasi satu sama lain. Sebagian besar, server tidak berjalan pada kapasitas penuh. Itu berarti ada kekuatan pemrosesan yang hasil buangnya tidak terpakai. Maka akan memerlukan sebuah cara. Teknik ini disebut virtualisasi server. Dengan memaksimalkan output dari setiap server, virtualisasi server mengurangi kebutuhan pada mesin dalam bekerja (Mutia, 2016).

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 Penjualan Ikan Koi yang telah dibuat sebelumnya dengan menggunakan Ubuntu 20.4 dan LAMPP (Apache 8, PHP 7.1, MySQL 5.2).
2. Membuat Dockerfile yang berada di Ubuntu Server (LAMPP) untuk mempermudah dalam Sistem Informasi Kost & Kontrakan..

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. Penyedia sistem 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. Pemasangan sistem ke server lain akan lebih mudah karena menggunakan Docker yang membungkus sistem beserta *dependency* yang diperlukan dalam sistem.

1.4 Tahap Penyelesaian Proyek Akhir

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

1. Menganalisis kebutuhan dari sistem informasi kost dan kontrakan untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan basis SaaS.
2. Menentukan konfigurasi yang tepat untuk pengaturan Dockerfile sehingga dapat digunakan sesuai *requirement* yang berupa:
 - a. Dapat diakses oleh jaringan/IP tertentu saja.
 - b. Dapat melakukan import database dengan mudah.
 - c. Menggunakan Ubuntu dengan versi 18.04.3 LTS.
3. Menguji keandalan arsitektur *cloud computing* yang dibangun dengan beberapa pengujian yaitu kecepatan waktu akses, batasan akses sesuai konfigurasi Docker

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Untuk membangun “Sistem Penyewaan Rumah Kontrakan & Kost Berbasis Web Menggunakan Ubuntu Lamp dan Dockerfile” yang berbasiskan 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 rumah yang dibutuhkan oleh sistem tersebut.
3. Untuk target pengguna dengan konsep cloud computing, maka penggunanya adalah para masyarakat yang merantau ke luar kota ataupun mahasiswa/pelajar. Tidak ada yang dapat mengakses sistem tersebut kecuali admin dari sistem tersebut. Sehingga diperlukan arsitektur cloud computing yang bersifat private, tidak dapat diakses secara bebas oleh semua orang kecuali orang yang berkepentingan.

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 untuk proyek pertama dan kedua

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 NAT	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.117.130/24	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.117.1	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.117.255	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	Versi Ubuntu	Ubuntu 18.04.4 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	<i>Harddisk</i>	20 GB	Alokasi <i>harddisk</i>
6.	<i>Memory</i>	1 GB	Alokasi <i>memory</i> untuk <i>guest OS</i>

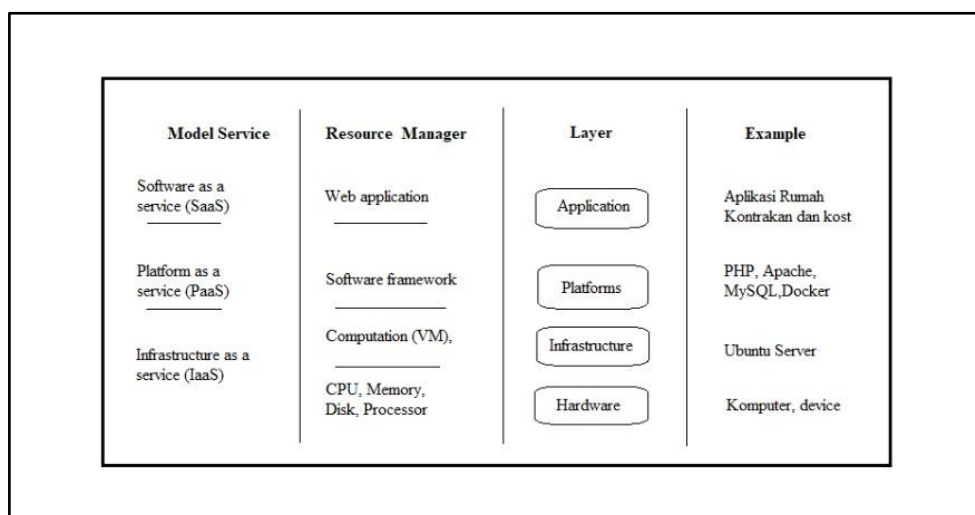
Selain spesifikasi mengenai VM *cloud computing* tersebut, untuk spesifikasi yang digunakan dalam Ubuntu OS yang telah dibuat dalam VM tersebut adalah sebagai berikut:

Tabel 2.2 Spesifikasi Ubuntu OS untuk proyek pertama

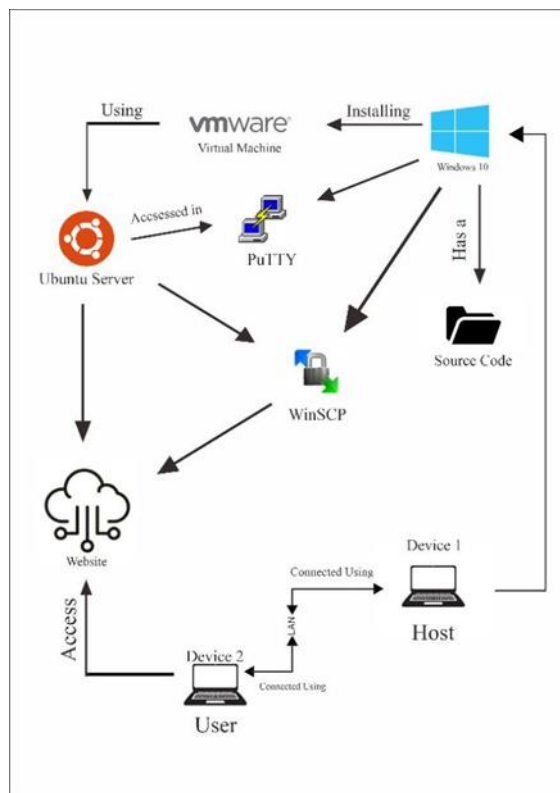
No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.3	Bahasa Pemrograman yang digunakan dalam membangun sistem informasi kost
		MySQL	Basis Data yang digunakan dalam menyimpan data sistem informasi kost
		PHPMyAdmin	Perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam MySQL
2.	PuTTY	PuTTY 0.73	Dukungan file sharing yang digunakan untuk memindahkan proyek manajemen password.
3.	SSH	Open SSH	Dukungan untuk melakukan remote server.
4.	Docker	Docker 19.03.8	Platform terbuka agar dapat membangun dan menjalankan aplikasi di berbagai lokasi sebagai sebuah <i>container</i> yang ringan
5.	Docker Compose	Docker Compose 1.24.0	Dukungan alat untuk mendefinisikan dan menjalankan multi-container.

2.2 Rancangan Arsitektur *Cloud Computing*

Pada proyek akhir ini digunakan arsitektur SaaS untuk Sistem Informasi Rumah Kontrakan & Kost, selanjutnya di bawahnya terdapat layer PaaS dimana sistem operasi Ubuntu berjalan. Kemudian pada layer paling bawah yaitu IaaS terdapat hardware laptop ASUS ROG yang melakukan virtualisasi pada hardware yang diperlukan. Arsitekturnya dapat dilihat pada gambar 2.1 dibawah ini:

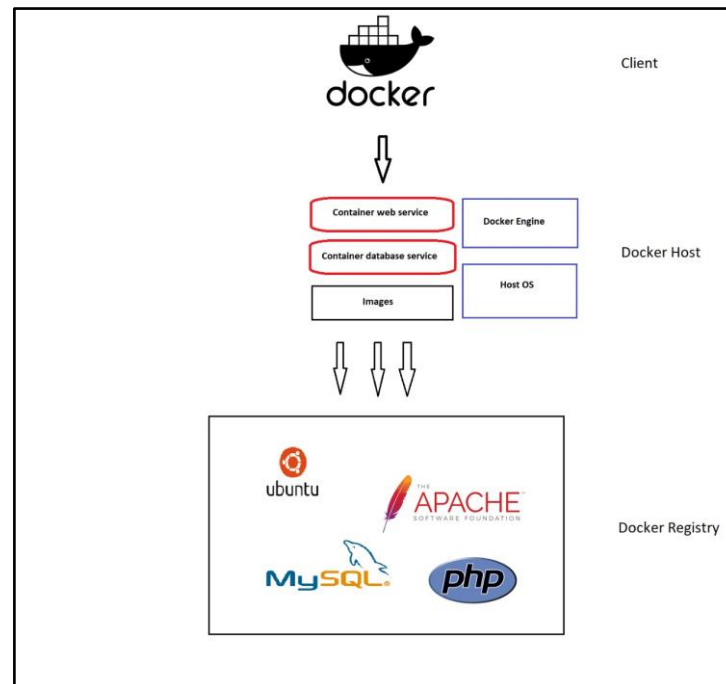
**Gambar 2.1** Penjelasan layer arsitektur terhadap komponen penyusunnya

Pada proyek pertama koneksi LAN (Indihome) digunakan untuk menghubungkan Device 1 yang bertugas sebagai host dan Device 2 yang bertugas sebagai User. Tujuan menghubungkan kedua device dengan LAN adalah agar website juga dapat diakses oleh Device 2 tidak hanya Device 1 saja. Device 1 yang bertugas sebagai host yang memakai OS Windows kemudian dilakukan instalasi menggunakan VMware Workstation, kemudian menghubungkannya dengan PuTTY dan mengoneksikannya dengan WinSCP. Kemudian pada VMWare Workstation dilakukan instalasi Ubuntu Server yang akan dilakukan remote oleh software PuTTY. Kemudian dalam Ubuntu Server akan dilakukan instalasi Apache, MySQL, PHPMyAdmin sehingga dapat menjalankan website. Source Code dari website yang akan digunakan kemudian di copy ke dalam directory yang terdapat pada Ubuntu Server dengan menggunakan WinSCP yang sudah tersambung dengan IP Address dari arsitekturnya dapat dilihat pada **Gambar 2.2** dibawah ini:



Gambar 2.2 Topologi Proyek Pertama

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



Gambar 2.3 Penjelasan arsitektur Docker terhadap komponen penyusunnya

2.3 Parameter dan Konfigurasi

Dalam pembuatan proyek akhir ini, pertama yang perlu dilakukan adalah menginstall web server 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

Dalam proyek ini database yang digunakan adalah MySQL. Parameter yang digunakan untuk instalasi MySQL dapat dilihat pada penjelasan **Modul 2.2** berikut ini:

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

```
$ sudo mysql_secure_installation
```

Keterangan:

- mysql-server : nama paket aplikasi untuk MySQL
- mysql_secure_installation : command untuk menjalankan proses instalasi MySQL

Modul 2.2 Parameter instalasi MySQL

Kemudian diperlukan suatu bahasa pemrograman agar aplikasi web ini dapat berjalan. Bahasa PHP dipilih dalam pengembangan proyek Manajemen Password ini. Parameter yang digunakan untuk instalasi PHP dapat dilihat pada **Modul 2.3** berikut :

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

Keterangan: suatu nama paket aplikasi

Modul 2.3 Parameter instalasi bahasa pemrograman PHP

Kemudian memerlukan instalasi PHP Myadmin. untuk memepmudahkan manajemen *database*.

Untuk parameter yang digunakan pada **Modul 2.4** sebagai berikut :

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

Keterangan: - phpmyadmin, php-mbstring, php-gettext : nama paket aplikasi

Modul 2.3 Parameter instalasi PHP MyAdmin

Sebelum melakukan instalasi Docker, maka diperlukan set up repository Docker, setelah itu instalasi Docker dapat dilakukan melalui repository tersebut. Parameter yang digunakan untuk melakukan set up repository pada **Modul 2.5** berikut:

```
$ sudo apt update
$ sudo apt install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common
```

Keterangan:

- update : parameter tambahan pada apt yang berfungsi untuk memperbarui dependency pada repository system
- apt-transport-https, ca-certificates, curl, gnupg-agent, software-properties-common : nama paket aplikasi

Modul 2.5 Parameter *setup repository* Docker

Setelah package index terbaru, maka dilanjut dengan menambahkan key GPG Docker. Parameter yang digunakan dalam penambahan key GPG dapat dilihat pada penjelasan **Modul 2.6** berikut ini:

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

Keterangan:

- curl : command line tool untuk mentransfer data menggunakan protokol jaringan.

- apt-key : digunakan untuk manajemen daftar key yang digunakan oleh apt untuk melakukan autentikasi package.
- add - : parameter tambahan pada key package manager untuk menambahkan key ke daftar trusted key.

Modul 2.6 Parameter penambahan key GPG Docker

Kemudian mengatur *repository* mana yang ingin digunakan. Docker memiliki tiga jenis versi repository yaitu: stable, nightly, dan test. Dalam implementasi proyek ini, repository yang digunakan yaitu versi stable. Parameter yang digunakan untuk memilih repository dapat dilihat pada Modul 2.7 berikut :

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

Keterangan:

- add-apt-repository : menambahkan PPA ke daftar sumber, sehingga Ubuntu tahu untuk mencari pembaruan dari PPA tersebut berasal dari sumber yang resmi.
- lsb_release : command untuk menampilkan LSB (Linux Standard Base) dan Distribution Information
- stable : versi repository yang akan diinstall.

Modul 2.7 Parameter pemilihan versi repository Docker

Apabila setup *repository* sudah selesai, maka dilanjutkan dengan instalasi Docker Engine. Parameter instalasi Docker Engine dapat dilihat pada **Modul 2.8** berikut:

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

Keterangan:

- docker-ce : paket aplikasi untuk docker engine community
- docker-ce-cli : paket aplikasi untuk docker engine command line
- containerd.io : paket aplikasi untuk container

Modul 2.8 Parameter instalasi Docker Engine

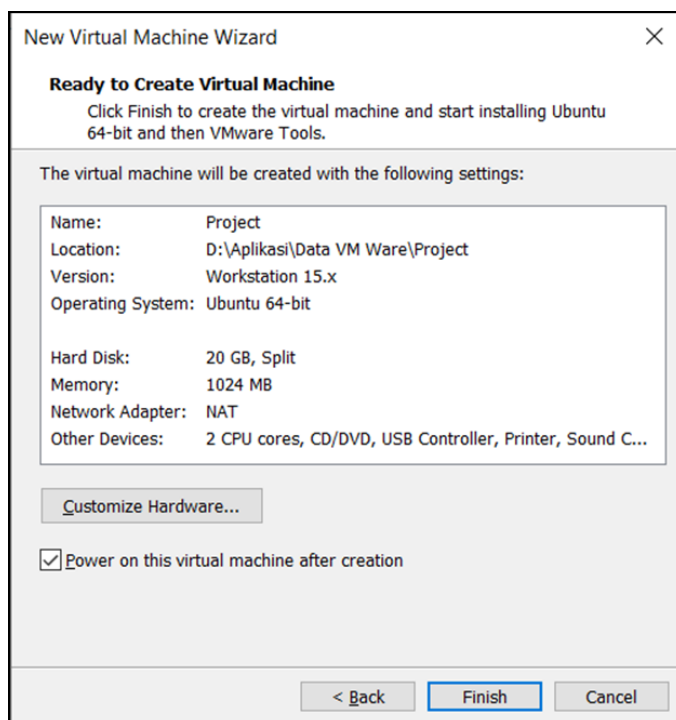
Kemudian diperlukan instalasi Docker Compose. Parameter instalasi Docker Compose dapat dilihat pada penjelasan **Modul 2.9** berikut:

```
$ sudo curl -L \
    "https://github.com/docker/compose/releases/download/1.25.5/dockercompose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Modul 2.9 Parameter instalasi Docker Compose

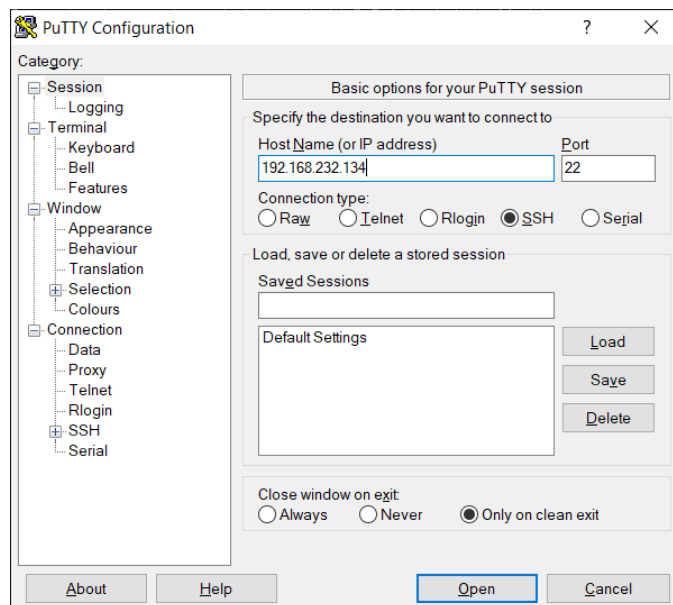
2.4 Tahap Implementasi

Pada tahap ini terdiri dari beberapa tahapan yang perlu dilakukan dalam penerapan *hosting* dengan Ubuntu LAMPP untuk Sistem Informasi Rumah Kontrakan dan Kos. Langkah awalnya adalah membuat *virtual machine* dulu. Langkah pembuatan *virtual machine* dengan VMWare Workstation dilakukan opsi konfigurasi seperti pada **Gambar 2.4** berikut ini:



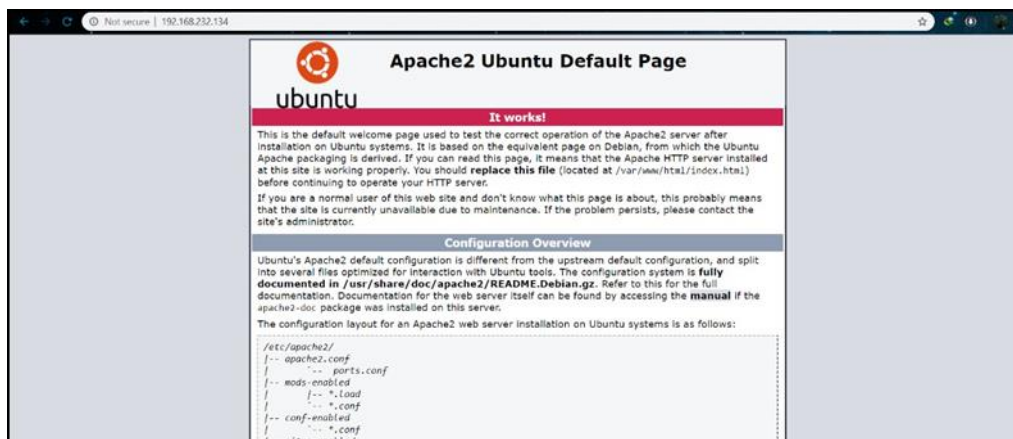
Gambar 2.4 Hasil Pembuatan Virtual Machine dengan OS Ubuntu

Apabila proses instalasi sudah selesai maka dilanjutkan dengan login pada server yang telah dibuat. Kemudian tahap selanjutnya yaitu melakukan remote server dengan menggunakan PuTTY. Ketika pertama kali membuka PuTTY dibutuhkan beberapa konfigurasi, isikan IP pada PuTTY dengan IP pada server Ubuntu. Apabila sudah, tampilan konfigurasi PuTTY dapat dilihat pada **Gambar 2.5** berikut:



Gambar 2.5 Tampilan konfigurasi pada PuTTY

Tahapan selanjutnya yaitu melakukan instalasi pada Apache, MySQL, PHP, dan PHP MyAdmin untuk mempermudah dalam manajemen database. Adapun proses instalasi dari Apache, MySQL, PHP, dan PHP MyAdmin dapat dilihat pada **Modul 2.1**, **Modul 2.2**, **Modul 2.3** dan **Modul 2.4**. Yang pertama yaitu merupakan konfigurasi untuk menginstal Ubuntu Server. Untuk hasil dari installasi modul tersebut dapat dilihat pada gambar berikut:



Gambar 2.6 Hasil instalasi Ubuntu

Kemudian untuk konfigurasi selanjutnya yaitu yang terdapat pada **Modul 2.2** dan **Modul 2.3** yaitu installasi PHP. Untuk hasil dari konfigurasi Modul tersebut terdapat pada gambar berikut :



PHP Version 7.2.24-0ubuntu0.18.04.4

System	Linux project 4.15.0-96-generic #97-Ubuntu SMP Wed Apr 1 03:25:46 UTC 2020 x86_64
Build Date	Apr 8 2020 15:45:57
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-mysqlnd.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-curl.ini, /etc/php/7.2/apache2/conf.d/20-dom.ini, /etc/php/7.2/apache2/conf.d/20-gd.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-ldap.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.ini, /etc/php/7.2/apache2/conf.d/20-mcrypt.ini, /etc/php/7.2/apache2/conf.d/20-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-xml.ini, /etc/php/7.2/apache2/conf.d/20-xmlrpc.ini, /etc/php/7.2/apache2/conf.d/20-zip.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API20170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled

Gambar 2.7 Hasil Instalasi PHP

Setelah proses selanjutnya yaitu instalasi phpMyAdmin yang terdapat pada **Modul 2.4**. Dimana setelah menginstal phpMyAdmin masih terdapat konfigurasi yang perlu dilakukan lagi. Untuk hasil instalasi phpMyAdmin dapat dilihat pada gambar berikut:

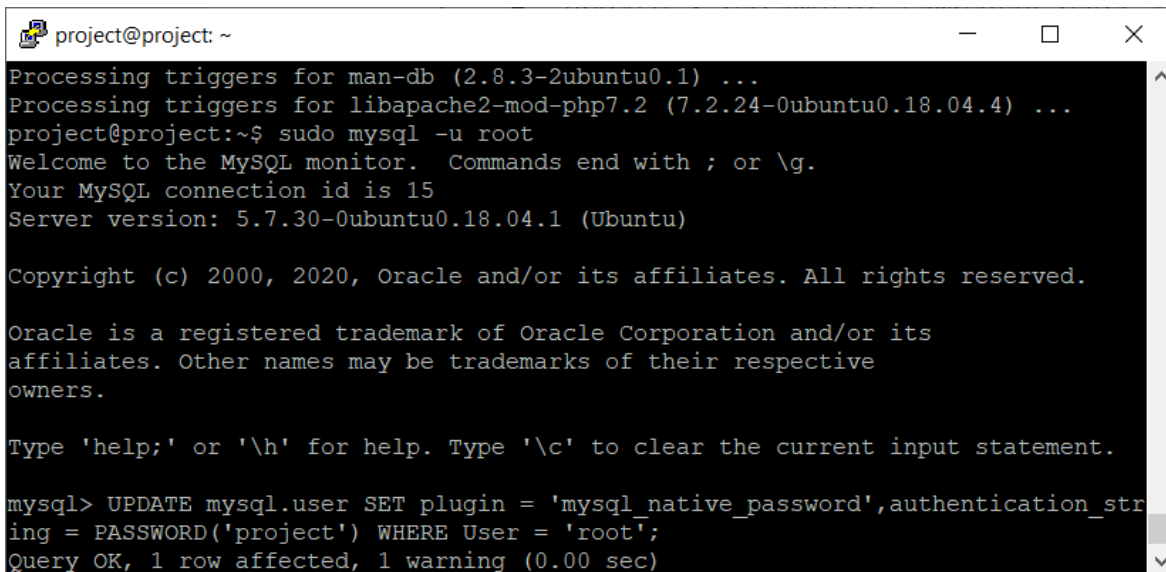


Gambar 2.8 Hasil Instalasi phpMyAdmin

Kemudian masih perlu dilakukan konfigurasi untuk mengatur mysql native. Agar kita dapat login ke dalam phpMyAdmin tersebut dengan mengubah konfigurasi passwordnya. Untuk konfigurasi nya sebagai berikut :

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

Modul 3.0 Konfigurasi password phpMyAdmin



```

project@project: ~
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libapache2-mod-php7.2 (7.2.24-0ubuntu0.18.04.4) ...
project@project:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.7.30-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

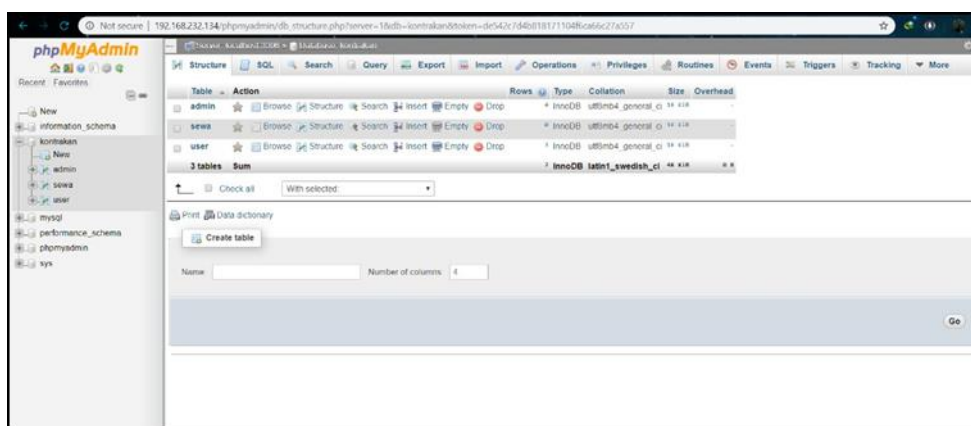
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> UPDATE mysql.user SET plugin = 'mysql_native_password',authentication_string = PASSWORD('project') WHERE User = 'root';
Query OK, 1 row affected, 1 warning (0.00 sec)

```

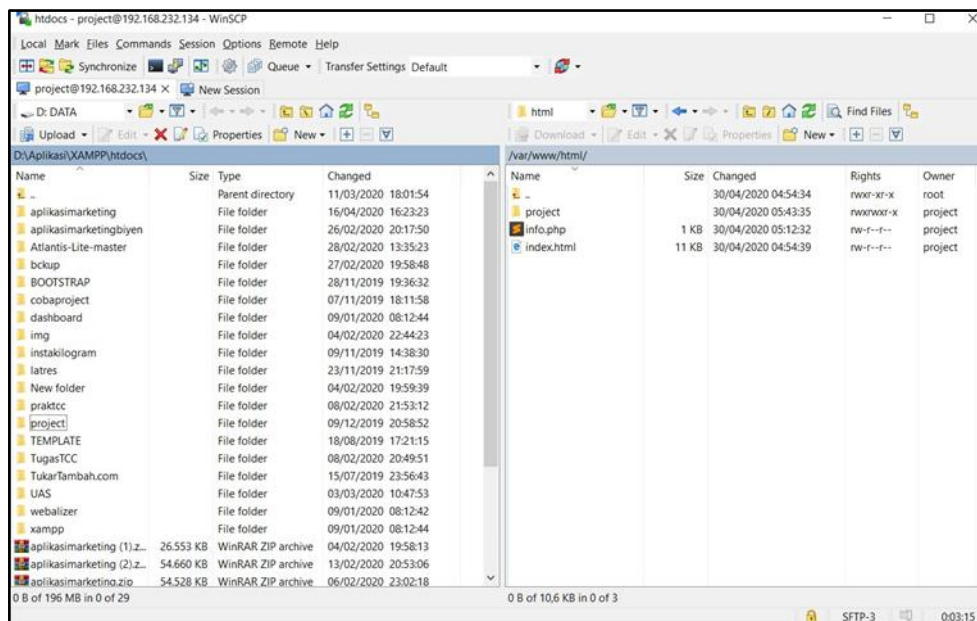
Gambar 2.9 Konfigurasi password phpMyAdmin pada PuTTY

Kalau sudah bisa login ke dalam phpMyAdmin kemudian melakukan import database tentang Sistem Informasi Rumah Kontrakan & Kos. Untuk hasil import database ke dalam phpMyAdmin dapat dilihat dalam gambar berikut :



Gambar 3.0 Hasil Import Database

Apabila proses import database sudah berhasil, maka proyek Sistem Informasi Rumah Kontrakan & Kos dapat digunakan dengan mengakses IP dari VM dan masuk ke direktori project/password. Namun, sebelum itu perlu pemindahan file proyek ke dalam server dengan menggunakan WinSCP. Adapun file proyek akan dipindah ke direktori /var/www/html/project/ pada server. Untuk gambar file yang sudah dipindahkan ke directory yang baru dapat dilihat pada gambar di bawah ini :



Gambar 3.1 Hasil Pemindahan Source Code

Apabila semua proses sudah sukses, maka proyek Sistem Informasi Rumah Kontrakan dan Kos dapat digunakan dengan mengakses IP dari VM dan masuk ke dalam system tersebut. Selanjutnya dalam pembuatan Dockerfile dilakukan dengan konfigurasi VM yang sama akan tetapi berada pada host yang berbeda. Pertama yang harus dilakukan adalah melakukan proses instalasi Docker seperti yang tertera pada **Modul 2.5** sampai dengan **Modul 2.9**. Dalam implementasi pembuatan Dockerfile dilakukan secara remote dengan memanfaatkan terminal dan SSH. **Gambar 3.2** berikut ini menampilkan docker sudah terinstall dalam server.

```

project@project: ~
Removing lamp_phpmyadmin:1 ... done
Removing network lamp_default
project@project:~/docker/lamp$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
cd
latest: Pulling from library/hello-world
0e03bdc26d7: Pull complete
Digest: sha256:6a65f928fb91fcfb963f7aa6d57c8eeb426ad9a20c7ee045538ef34847f44f1
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

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

project@project:~/docker/lamp$ cd
project@project:~$

```

Gambar 3.2 Hasil Instalasi Docker

Kemudian setelah menginstall docker selesai dilanjutkan dengan menginstall docker compose. Langkah selanjutnya yaitu dengan menginstall parameter konektivitas yang akan digunakan yaitu dengan memasukkan konfigurasinya sebagai berikut :

```
$ 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
Keterangan :
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah
instalasi paket aplikasi
- curl : parameter yang digunakan untuk mengecek konektivitas URL dan
sebagai tool transfer data dari URL tersebut
```

Modul 3.1 Konfigurasi Docker Compose

Kemudian setelah mengkonfigurasi docker compose kemudian kita akan membuat *directory* file. Dimana *file directory* tersebut nantinya merupakan tempat menyimpan file docker yang nantinya kita buat. Untuk konfigurasi pembuatan *directory* file tersebut menggunakan konfigurasi sebagai berikut :

```
$ mkdir -p ~/docker/lamp/html
```

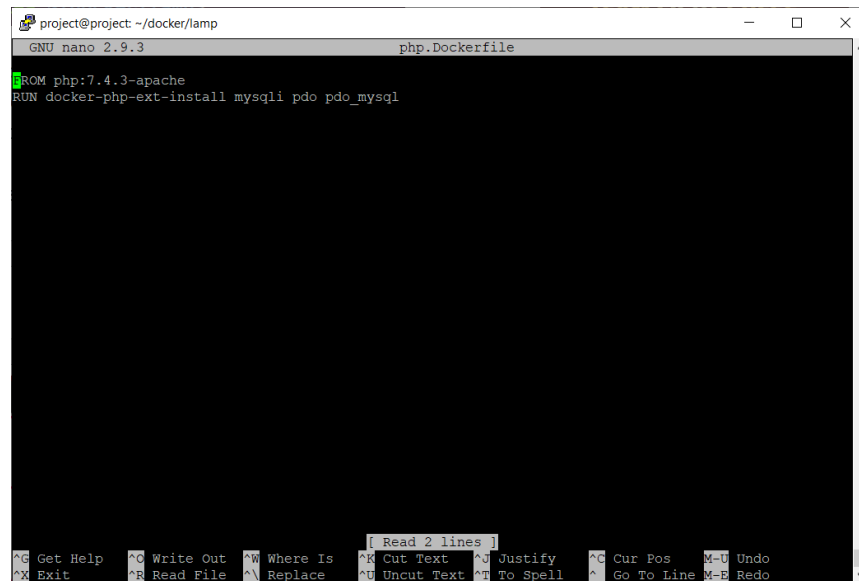
Modul 3.2 Konfigurasi pembuatan lokasi file

Setelah pembuatan file *directory* selesai, maka proses selanjutnya adalah pembuatan dockerfile. File *directory* tadi berada pada file /docker/lamp/html. Kemudian untuk membuat docker file dengan cara keluar dari *directory* tersebut menuju ~/docker/lamp/. Kemudian menggunakan konfigurasi pada **Modul 3.3** dibawah ini untuk membuat dockerfile.

```
nano php.Dockerfile
```

Modul 3.3 Konfigurasi pembuatan dockerfile

Di dalam docker file terdapat konfigurasi untuk Bahasa pemrograman yang nantinya digunakan dalam docker. Untuk isi dari docker file nya terdapat pada gambar di bawah ini:



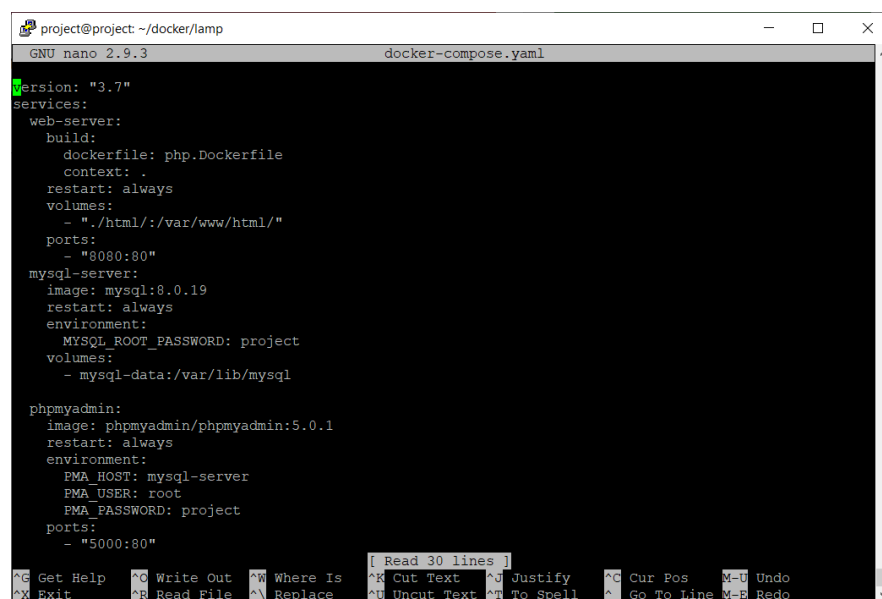
```

project@project ~/docker/lamp
GNU nano 2.9.3      php.Dockerfile
FROM php:7.4.3-apache
RUN docker-php-ext-install mysqli pdo pdo_mysql

```

Gambar 3.3 Isi dari dockerfile

Proses selanjutnya yaitu pembuatan konfigurasi database dan docker file dengan sistem yang dijalankan dalam docker. Untuk isi dalam konfigurasi system di dalam docker compose sebagai berikut yang terdapat dalam **Gambar 3.4** :



```

project@project ~/docker/lamp
GNU nano 2.9.3      docker-compose.yaml
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: project
    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: project
    ports:
      - "5000:80"

```

Gambar 3.4 Isi dari dockerfile

Jika semua service yang diperlukan sudah selesai diinisialisasi pada dockercompose.yml, tahap selanjutnya adalah menjalankan service tersebut. Adapun parameter untuk menjalankan service melalui Docker compose dapat dilihat pada **Modul 3.4** berikut ini:

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

- docker-compose : command untuk mengeksekusi perintah-perintah Docker Compose.
- up : parameter dalam docker-compose untuk menjalankan container.
- -d : parameter dalam docker-compose agar proses berjalan di background.

Modul 3.4 Parameter eksekusi docker compose

Untuk menjalankan program tersebut masuk ke dalam directory file ~/docker/lamp/. Kemudian untuk mengeliminasi proses docker compose menggunakan parameter yang terdapat dalam **Modul 3.5** sebagai berikut :

```
$ sudo docker-compose down
```

Modul 3.5 Parameter eliminasi docker compose

Apabila proses sudah selesai maka proyek Sistem Informasi Rumah Kontrakan & Kos ini dapat dibuka melalui browser dengan mengakses `http://[IP_SERVER]:[PORT]`.

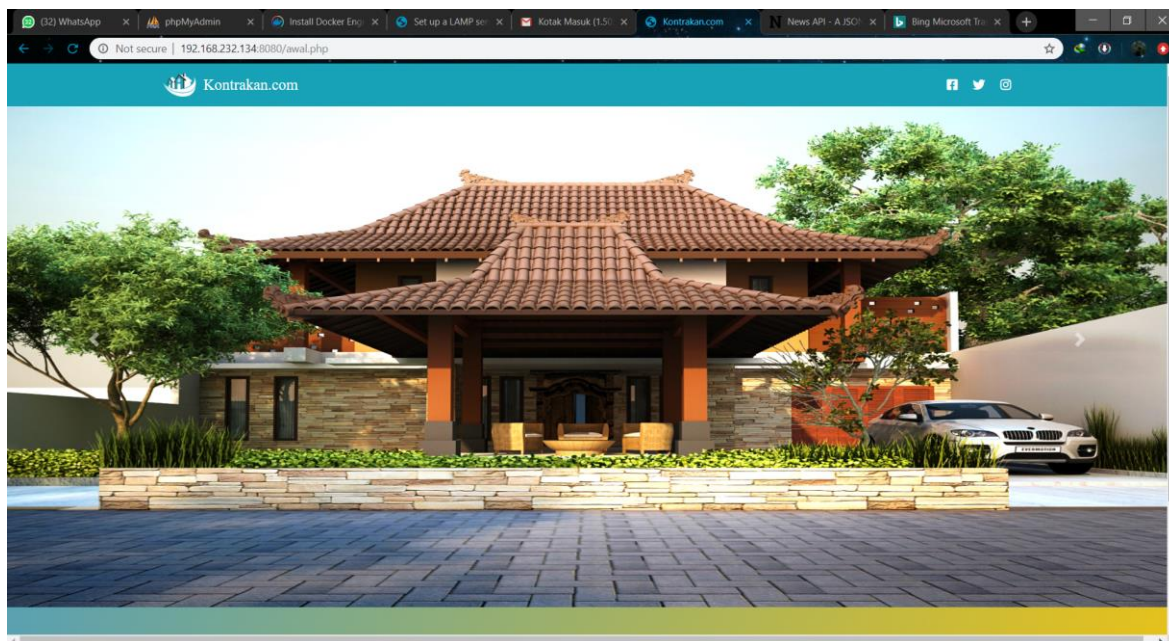
2.5 Hasil Implementasi

Setelah melakukan konfigurasi Ubuntu LAMPP dan integrasi sistem ke server, hasil proses hosting local dapat dilihat pada **Gambar 3.5** berikut ini:



Gambar 3.5 Tampilan proyek Rumah Kontrakan & Kos

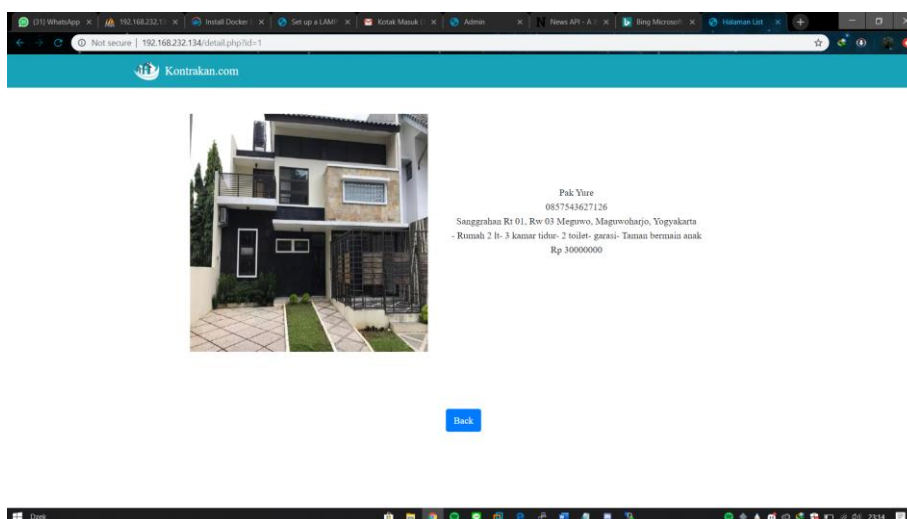
Kemudian untuk proyek yang berjalan diatas container, hasilnya dapat dilihat pada **Gambar 3.6** berikut ini:



Gambar 3.6 Tampilan proyek Rumah Kontrakan & Kos

2.6 Pengujian Singkat

Untuk pengujian singkat ini akan dijelaskan hasil dari apa yang sudah dikerjakan dalam tahap tahap sebelumnya, untuk mengetahui apakah sistem yang dihosting sudah dapat berjalan normal, atau malah terdapat kendala atau sistem yang dihosting dapat dijalankan namun, terdapat beberapa fitur yang tidak berjalan secara normal. Pengujian pertama yang dilakukan adalah, pengujian melihat informasi yang tersedia dalam menu lihat list. Di mana menu tersebut berisikan list rumah yang di kontrakkan dan di kostkan. Kemudian untuk melihat informasi rumah yang diinginkan dapat melihat info detail dari rumah tersebut dengan klik *button* detail, seperti dapat dilihat pada **Gambar 3.7** seperti dibawah ini.



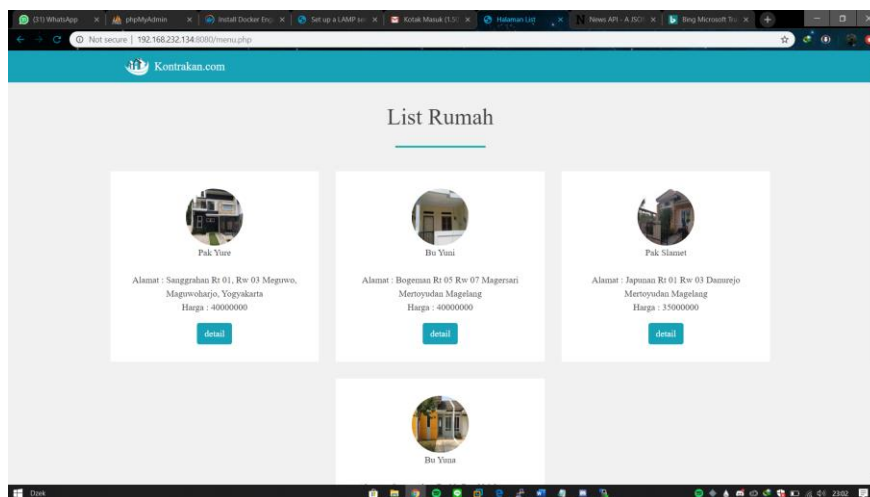
Gambar 3.7 Tampilan detail Rumah Kontrakan & Kos

Pengujian selanjutnya adalah pengujian yang ditujukan untuk mengedit data informasi rumah yang di kontrakkan maupun di koskan. Seperti yang terlihat dalam **Gambar 3.8**, terdapat Nama dari pemilik rumah yang diubah menjadi Ibu Sholehah.

Form 1: Pak Yure	Form 2: Ibu Sholehah	Form 3: Pak Slamet
Nama Pemilik: Pak Yure	Nama Pemilik: Ibu Sholehah	Nama Pemilik: Pak Slamet
Alamat Rumah: Saugrahan Rt 01, Rw 03 Megurwo, Magurovoharjo	Alamat Rumah: Japunan Rt 05 Rw 07 Magerani Mertoyudan Mag	Alamat Rumah: Japunan Rt 01 Rw 03 Damurejo Mertoyudan Mage
Harga Rumah: 30000000	Harga Rumah: 40000000	Harga Rumah: 35000000
No Hp: 0857543627126	No Hp: 09999999	No Hp: 08583252625
Fasilitas Rumah: - Rumah 2 lt- 3 kamar tidur- 2 toilet- garasi- Taman	Fasilitas Rumah: -Kamar mandi dalam - 4 kamar tidur - 3 toilet	Fasilitas Rumah: - 3 Kamar Tidur - 2 Kamar Mandi- Garasi
Gambar: [Image of Pak Yure's house]	Gambar: [Image of Ibu Sholehah's house]	Gambar: [Image of Pak Slamet's house]
Buttons: Update, Hapus	Buttons: Update, Hapus	Buttons: Update, Hapus

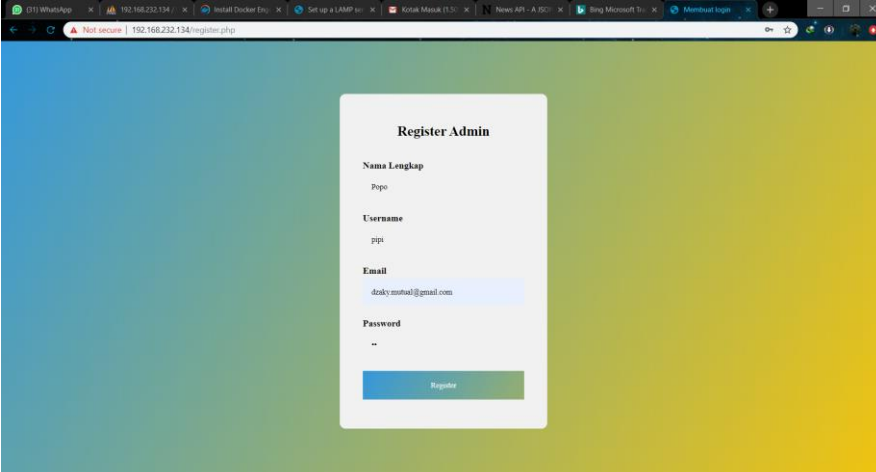
Gambar 3.8 Tampilan edit data Rumah Kontrakan & Kos

Kemudian untuk pengujian selanjutnya adalah pengujian melihat list dari data informasi rumah yang di kontrakkan maupun di koskan. Berapa jumlah rumah, berbagai bentuk dan tipe rumah yang di kontrakkan. Seperti yang terlihat dalam **Gambar 3.9** berikut:



Gambar 3.9 Tampilan list data Rumah Kontrakan & Kos

Kemudian untuk pengujian yang terakhir selanjutnya adalah pengujian untuk membuat admin baru yang akan mengelola sistem ini. Dimana calon admin tersebut di harus kan mengisi Nama Lengkap, Username, Email dan Password. Seperti yang terlihat dalam **Gambar 4.0** berikut:



The image shows a web browser window with a 'Register Admin' form. The form is white and centered on a blue and yellow gradient background. The form fields are:

- Nama Lengkap**: Popo
- Username**: pipi
- Email**: dzaky.mutual@gmail.com
- Password**: Two empty fields with a strength indicator (two dots).

A green 'Register' button is located at the bottom of the form.

Gambar 4.0 Tampilan Registrasi Admin Rumah Kontrakan & Kos

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	
		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	Dzaky
2.	Pengujian Singkat	Dzaky
3.	Latar Belakang Masalah	Arman, Dzaky
4.	Agenda Pengerjaan Proyek	Dzaky
5.	Pembuatan Tugas 1 (LAMPP)	Dzaky
6.	Pembuatan Laporan Sementara	Arman, Dzaky
7.	Revisi Laporan	Arman
8.	Pembuatan Tugas 2 (Docker)	Dzaky
9.	Pembuatan Laporan Akhir	Arman, Dzaky

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan masalah yang ada, yaitu membuat aplikasi sistem informasi rumah kontrakan & kost diperoleh hasil yang memuaskan. Sistem tersebut dapat berjalan dengan baik dengan dijalankan melalui hosting lokal maupun melalui Docker container. Dari pengujian sistem, semua pengujian yang dibuat dapat berjalan dengan baik sesuai dengan yang diharapkan. Pembagian tugas terlaksana dengan kurang baik, dikarenakan terdapat kendala pada Arman sehingga pembagian tugas menjadi kurang merata.

4.2 Saran

Pembagian tugas seharusnya lebih merata, untuk penjelasan sebaiknya lebih di perjelas lagi agar semakin mudah dipahami dalam pembelajaran.

DAFTAR PUSTAKA

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

Kurniawan, E. (2015). PENERAPAN TEKNOLOGI CLOUD COMPUTING DI UNIVERSITAS Studi Kasus : Fakultas Teknologi Informasi UKDW. *Eksis*, 08(01), 29–36.

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>

Ashari, A., & Setiawan, H. (2011). Cloud Computing: Solusi ICT?. *Jurnal Sistem Informasi*, 3(2).

LAMPIRAN

Lampiran pada bagian ini dapat berupa screenshoot, listing program yang terlalu panjang, dan sebagainya, atau dapat juga tugas bilamana diminta oleh asisten praktikum.

```
project@project: ~
Query OK, 1 row affected, 1 warning (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 1

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
project@project:~$ sudo apt-get update
Hit:1 http://id.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://id.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://id.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://id.archive.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Fetched 252 kB in 6s (39.4 kB/s)
Reading package lists... Done
project@project:~$ sudo apt-get install \
> apt-transport-https \
> ca-certificates \
> curl \
> gnupg-agent \
> software-properties-common
```

```
project@project: ~
Preparing to unpack .../gnupg-agent_2.2.4-1ubuntu1.2_all.deb ...
Unpacking gnupg-agent (2.2.4-1ubuntu1.2) ...
Setting up apt-transport-https (1.6.12ubuntu0.1) ...
Setting up gnupg-agent (2.2.4-1ubuntu1.2) ...
project@project:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
project@project:~$ sudo apt-key fingerprint 0EBFCD88
pub  rsa4096 2017-02-22 [SCEA]
     9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid          [ unknown] Docker Release (CE deb) <docker@docker.com>
sub  rsa4096 2017-02-22 [S]

project@project:~$ sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) \
> stable"
Get:1 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
```

```

project@project: ~
;
curl: (3) Illegal characters found in URL
-bash: syntax error near unexpected token `;'
project@project:~$ sudo apt install curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.8).
0 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
project@project:~$ 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
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 638    100 638    0     0    712      0 --:--:-- --:--:-- --:--:--    711
100 15.4M  100 15.4M    0     0 38414      0 0:07:00 0:07:00 --:--:-- 37472
project@project:~$

```

```

project@project: ~
curl: (3) Illegal characters found in URL
-bash: syntax error near unexpected token `;'
project@project:~$ sudo apt install curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.8).
0 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
project@project:~$ 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
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 638    100 638    0     0    712      0 --:--:-- --:--:-- --:--:--    711
100 15.4M  100 15.4M    0     0 38414      0 0:07:00 0:07:00 --:--:-- 37472
project@project:~$ sudo chmod +x /usr/local/bin/docker-compose
project@project:~$ mkdir -p ~/docker/lamp/html
project@project:~$

```

```

project@project: ~/docker/lamp
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
project@project:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.24.1/"
>
>
> ;
> sudo curl -L "https://github.com/docker/compose/releases/download/1.24.1/"
;
curl: (3) Illegal characters found in URL
-bash: syntax error near unexpected token `;'
project@project:~$ sudo apt install curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
curl is already the newest version (7.58.0-2ubuntu3.8).
0 upgraded, 0 newly installed, 0 to remove and 31 not upgraded.
project@project:~$ 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
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 638    100 638    0     0    712      0 --:--:-- --:--:-- --:--:--    711
100 15.4M  100 15.4M    0     0 38414      0 0:07:00 0:07:00 --:--:-- 37472
project@project:~$ sudo chmod +x /usr/local/bin/docker-compose
project@project:~$ mkdir -p ~/docker/lamp/html
project@project:~$ cd ~/docker/lamp
project@project:~/docker/lamp$ nano php.Dockerfile
project@project:~/docker/lamp$ nano docker-compose.yaml
project@project:~/docker/lamp$ cd html
project@project:~/docker/lamp/html$ nano index.php
project@project:~/docker/lamp/html$ cd ~/docker/lamp
project@project:~/docker/lamp$ sudo apt install tree

```

```

project@project: ~/docker/lamp
project@project:~/docker/lamp/html$ cd ~/docker/lamp
project@project:~/docker/lamp$ sudo apt install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 31 not upgraded.
Need to get 40.7 kB of archives.
After this operation, 105 kB of additional disk space will be used.
Get:1 http://id.archive.ubuntu.com/ubuntu bionic/universe amd64 tree amd64 1.7.0-5 [40.7 kB]
Fetched 40.7 kB in 22s (1,868 B/s)
Selecting previously unselected package tree.
(Reading database ... 71540 files and directories currently installed.)
Preparing to unpack .../tree_1.7.0-5_amd64.deb ...
Unpacking tree (1.7.0-5) ...
Setting up tree (1.7.0-5) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
project@project:~/docker/lamp$ tree
.
├── docker-compose.yaml
├── html
│   └── index.php
└── php.Dockerfile

1 directory, 3 files
project@project:~/docker/lamp$ sudo docker-compose up -d
Creating network "lamp_default" with the default driver
Creating volume "lamp_mysql-data" with default driver
Building web-server
Step 1/2 : FROM php:7.4.3-apache

```

