

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING
LAPORAN PROYEK AKHIR**

**SISTEM INFORMASI PENDAFTARAN ANTRIAN PASIEN PADA JADWAL
PRAKTEK DOKTER MENGGUNAKAN UBUNTU LAMPP DAN
IMPLEMENTASI DOCKER PADA PENGELOLAAN SISTEM**



DISUSUN OLEH:

**NAMA ANGGOTA : ANISA SHOLICHAWATI 123170046
M YORA REZA ALFAJRI 123170070
KELAS : A
ASISTEN PRAKTIKUM : JALUANDA PARAMA
MUHAMMAD IMAM ALFATAH**

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

2020

HALAMAN PENGESAHAN

SISTEM INFORMASI PENDAFTARAN ANTRIAN PASIEN PADA JADWAL PRAKTEK DOKTER MENGGUNAKAN UBUNTU DAN LAMPP

Disusun oleh :

Anisa Sholichawati

123170046

M Yora Reza Alfajri

123170070

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

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

123150131

M Imam Alfatah

NIM. 123160119

Mengetahui,

Ka. Lab. Sistem Digital

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

NIK. 2 8201 13 0425 1

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan laporan tugas akhir praktikum Teknologi Cloud Computing yang berjudul **Sistem Pendaftaran Antrian Pasien pada Jadwal Praktek Dokter** ini tepat pada waktunya.

Adapun tujuan dari penulisan laporan tugas akhir ini adalah untuk memenuhi tugas asisten praktikum pada praktikum Teknologi Cloud Computing. Selain itu laporan ini juga bertujuan untuk mendokumentasikan juga menambah wawasan bagi para pembaca dan juga penulis tentang segala hal yang berhubungan dengan cloud computing salah satunya adalah Docker.

Kami ucapkan terima kasih kepada semua pihak yang telah membimbing kami dengan pengetahuan sehingga kami dapat menyusun dan menyelesaikan laporan ini dengan baik dan benar. Laporan yang kami buat masih jauh dari kata sempurna, oleh karena itu kritik dan saran sangat dibutuhkan. Guna membangun laporan menjadi lebih baik dan sempurna di masa mendatang.

Yogyakarta, 1 April 2020

Penyusun

DAFTAR ISI

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.....	2
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>	5
2.3 Parameter dan Konfigurasi	6
2.4 Tahap Implementasi.....	7
2.5 Hasil Implementasi	36
2.6 Pengujian Singkat	38
BAB III JADWAL Pengerjaan dan Pembagian Tugas	13
3.1 Agenda Pengerjaan	13
3.2 Keterangan Pembagian Tugas.....	13
BAB IV KESIMPULAN DAN SARAN	41
4.1 Kesimpulan	41
4.2 Saran	42
DAFTAR PUSTAKA	43
LAMPIRAN	44

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pada tahun 1960-an ide awal cloud computing datang dari seorang pionir intelegensi buatan yang merupakan pakar IT bernama John McCarthy menyampaikan bahwa di masa mendatang komputasi awan akan menjadi infrastruktur publik. Pada tahun 1995 Larry Ellison menawarkan ide “Network Computing” dimana pengguna tidak membutuhkan software mulai dari sistem operasi sampai dengan software lainnya, personal computer dapat digantikan oleh sebuah terminal yang tersambung langsung dengan sebuah server yang menyediakan environment berisi semua kebutuhan software yang dibutuhkan dan siap diakses oleh client. Kelanjutan dari ide-ide tersebut disambung pada tahun 2005 saat Amazon dan IBM meluncurkan produk cloud computingnya (Hechavarría, Rodney; López, 2013).

Pada era digital semua sosial media merupakan penerapan dari cloud computing, tanpa terasa sebenarnya cloud computing sudah digunakan dalam kehidupan sehari-hari. Sehingga keberadaannya memiliki peran penting dalam penerapan dan pengembangan teknologi terkini. Dengan cloud computing semua data tersimpan di server secara terpusat sehingga pengguna tidak perlu repot menyiapkan data center, atau pun media penyimpanan lain karena semua telah tersedia secara virtual maka hal ini juga dapat bermanfaat pada penghematan biaya. Manfaat lain dari cloud computing ialah dari sisi keamanan data yang dapat disimpan dengan aman pada server yang disediakan layanan cloud computing. Teknologi ini fleksibel, mudah diakses dimana saja dan kapan saja asalkan terhubung ke internet (Syaikhu, 2013).

Untuk mempermudah proses deployment aplikasi web beserta pendukungnya *Docker* diimplementasikan. Menurut Sugianto, *Docker* adalah sebuah platform terbuka bagi pengembang perangkat lunak dan pengelola sistem jaringan untuk membangun, mengirimkan dan menjalankan aplikasi-aplikasi terdistribusi. Komponen *Docker* diantaranya Docker Images, Docker Container, Docker Registry, Docker File, dan Docker Index. Docker file merupakan sebuah skrip otomatis yang membangun sebuah image, berisi semua perintah yang biasanya dilakukan manual untuk membangun sebuah image (ROMADLON BIK, 2017). Docker digunakan untuk membundel dan menjalankan aplikasi dimanapun dalam satu container yang ringan. Mirip seperti Virtual Machine namun lebih ringan karena docker tidak membawa keseluruhan sistem operasi melainkan berbagi sistem dengan host induknya. Docker menggunakan sistem aplikasi yang terisolasi. Hal yang membuat docker menjadi virtualisasi yang handal adalah antarcontainer pada docker dapat berjalan tanpa mengganggu container lainnya. Container pada docker seperti sebuah komputer yang berada di belakang router yang di NAT, sehingga container tersebut tidak langsung terhubung ke jaringan luar. Docker yang sudah menyediakan beberapa container sekaligus dan menghubungkannya disebut

Docker Compose. Docker compose adalah alat untuk mengkonfigurasi bagaimana docker image dijalankan, dengan alat ini kita dapat mengkonfigurasi port apa yang akan kita buka, environment aplikasi, dan lain-lain (Kusuma et al., 2017). Dengan adanya docker, developer dapat mendeliver aplikasi lengkap dengan environment tempat dia dijalankan sehingga sysadmin dapat langsung menjalankannya dengan mudah. *Container* adalah sebuah abstraksi pada layer aplikasi yang membungkus paket-paket perangkat lunak dan semua yang diperlukan untuk menjalankannya (Docker, 2018). *Container* adalah instance dari aplikasi berisi semua library dan komponen yang diperlukan agar aplikasi dapat berfungsi. Container docker dibentuk dari image yang merupakan hasil dari aplikasi atau layanan yang dikemas.

Dalam kegiatannya rumah sakit menyimpan begitu banyak data pasien yang sifatnya merupakan data pribadi atau rahasia karena berhubungan dengan riwayat penyakit dan kesehatan pasien. Data tersebut digunakan sebagai acuan ketika pasien membutuhkan fasilitas perawatan dan pengobatan. Data ini harus selalu dapat diakses dimana saja dan kapan saja. Namun beberapa rumah sakit masih menggunakan cara manual seperti menyimpan berkas dalam bentuk kertas. Penerapan cloud computing untuk pendaftaran antrian pasien di rumah sakit akan berguna karena dapat mempercepat pendataan dalam pendaftaran sehingga waktu yang digunakan menjadi lebih efisien. Maka dari itu diimplementasikan docker yang merupakan strategi hybrid cloud untuk mengatasi permasalahan tersebut.

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 Pendaftaran Antrian Pasien yang telah dibuat sebelumnya dengan menggunakan Ubuntu 18.04.4 dan LAMPP (Apache 8, PHP 7.2, MySQL 5.2).
2. Mengintegrasikan Sistem Pendaftaran Antrian Pasien di Rumah Sakit yang berada di Ubuntu Server (LAMPP)
3. Mengimplementasikan docker pada pengelolaan Sistem Pendaftaran Antrian Pasien di Rumah Sakit.

1.3 Manfaat Proyek Akhir

Manfaat yang dapat diperoleh dari 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 mempermasalahkan *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 terdistribusi dikarenakan setiap sistem terhubung satu sama lain melalui *private cloud network*.

1.4 Tahap Penyelesaian Proyek Akhir

Proyek akhir sudah dapat dijalankan sebagaimana mestinya. Implementasinya menggunakan virtual machine dan sudah dapat diakses melalui browser. Hasilnya adalah proyek yang dirancang sudah berfungsi dengan baik.

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Untuk menyelesaikan proyek tersebut dibutuhkan resource komputer yang mumpuni. Agar dapat menjalankan virtual machine seperti VMWare Workstation dibutuhkan spesifikasi berikut ini :

- 1) Minimum processor 64-bit x86 Intel Core 2 Duo, AMD FX Dual Core atau yang setara dengan itu.
- 2) Minimum 1.3GHz core speed processor.
- 3) Minimum 2GB RAM.
- 4) Minimum 300MB free space, pastikan komputer host tidak kehabisan resource karena virtual machine.
- 5) Sistem operasi yang dapat digunakan diantaranya Windows 10, Windows 8.x, Windows 7, Windows Server 2012, Windows Server 2008, Ubuntu 8.04 atau yang lebih baru, Red Hat Enterprise Linux 5 atau yang lebih baru, CentOS 5.0 atau yang lebih baru, Oracle Linux 5.0 atau yang lebih baru, openSUSE 10.2 atau yang lebih baru, dan SUSE Linux 10 atau yang lebih baru.

Software yang digunakan adalah VMWare Workstation Pro 15.5.2 Pro yang dapat menjalankan sistem operasi lain di dalam suatu sistem tanpa mengganggu sistem operasi utama secara virtual.

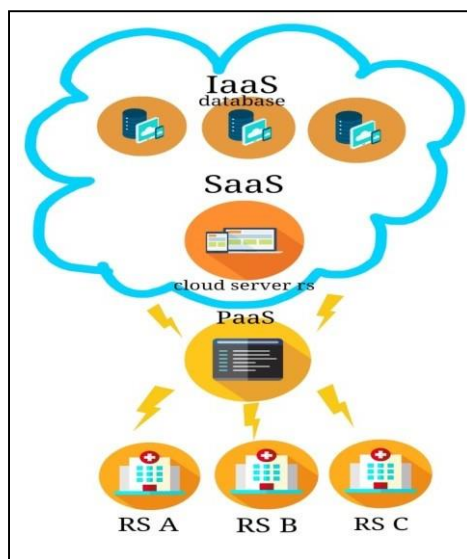
Dapat didownload dari <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html> baik dengan lisensi atau percobaan gratis selama 30 hari. Dibutuhkan juga Ubuntu 18.04.4 sebagai sistem operasinya yang dapat didownload di <https://ubuntu.com/#download> Ubuntu Server 18.04 LTS. PuTTY 64-bit-0.73 digunakan untuk memanfaatkan protokol jaringan seperti SSH dan Telnet guna mengaktifkan sesi remote pada komputer didapatkan di <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>.

Winscp adalah aplikasi yang berfungsi untuk transfer file atau copy file antara windows dengan linux. Dapat didownload di <https://winscp.net/eng/download.php>.

Sublime text adalah aplikasi text editor yang digunakan untuk membuka, menulis, bahkan mengedit file code seperti C, C++, C#, PHP, CSS, HTML, dan lain-lain. Software ini berperan penting dalam membuat web pendaftaran antrian online.

2.2 Rancangan Arsitektur *Cloud Computing*

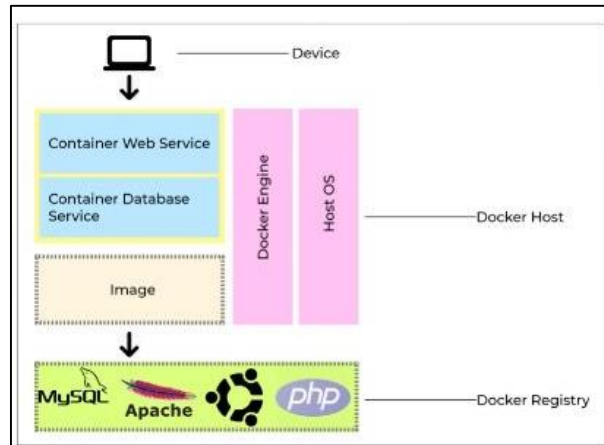
Pada projek ini terdapat sisi front end dan sisi back end. Keduanya terhubung melalui internet dan cloud. Front end mencakup komputer, web browser yang diperlukan untuk mengakses layanan SaaS (Software as a Service) berupa aplikasi Sistem Pendaftaran Antrian Pasien yang digunakan di lebih dari satu rumah sakit. Sedangkan Back end yaitu bagian penyimpanan data (storage) yang menciptakan layanan “cloud computing” pada data Rumah Sakit adalah layanan IaaS (Infrastructure as a Service).



Gambar 2.1 Arsitektur Cloud Computing SPAP

Penerapan SaaS terletak pada cloud computing Sistem Pendaftaran Antrian Pasien semua datanya disimpan dalam cloud dan hanya digunakan oleh beberapa rumah sakit. Sedangkan penerapan PaaS yaitu platform yang digunakan pada layanan SaaS yang tersedia dalam cloud computing Sistem Pendaftaran Antrian Pasien. Menyediakan layanan infrastruktur IT seperti network, memory, storage adalah peran IaaS dalam cloud computing Sistem Pendaftaran Antrian Pasien, dan komputer dikonfigurasi sesuai dengan kebutuhan rumah sakit.

Pada implementasi docker, container dibentuk dari image yang merupakan hasil dari aplikasi atau layanan yang dikemas. Ini dapat berisi sistem operasi yang lengkap atau aplikasi yang sudah diinstal sebelumnya. Artinya dari image adalah bahwa container akan mulai berfungsi. Berikut ini arsitektur virtual machine.



Gambar 2.2 Arsitektur Docker

2.3 Parameter dan Konfigurasi

Dalam pengerjaan pembuatan virtual machine dengan VMWare perlu dilakukan konfigurasi terlebih dahulu sehingga aplikasi dapat digunakan untuk memvirtualkan operating system Ubuntu.

Tabel 2.1 Konfigurasi pada VMware Workstation yang digunakan

No.	Nama Parameter	Nilai	Keterangan
1.	VMware version	15.5.2	Versi .
2.	VMWare build	15785246	Keterangan <i>build (patch) number</i> .
3.	IPv4 hypervisor	IP: 192.168.30.65 (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: 192.168.43.216	Kelas IP/ <i>subnet mask</i> yang digunakan.
		DNS :	Alamat IP untuk fungsionalitas DNS.
		GW: 192.168.43.1	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	Processor info	12 x Intel (R) Core (TM) i5-5200 CPU @ 2.2GHz	Jenis <i>processor</i> yang digunakan pada <i>hypervisor</i> .
5.	RAM info	4 GB Memory	Kapasitas RAM pada <i>hypervisor</i> .

Contoh: 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

2.4 Tahap Implementasi

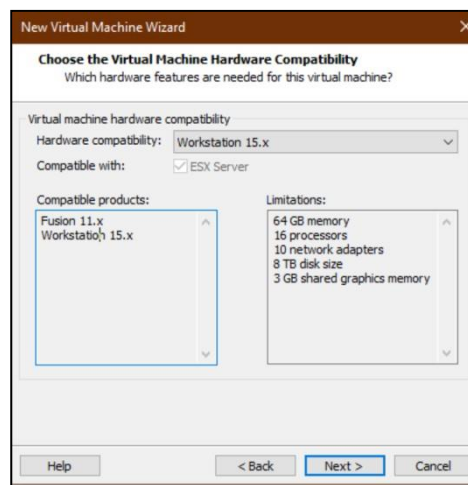
Hal pertama yang harus dilakukan adalah menginstal Ubuntu pada VMWare, berikut tahapannya :

- 1) Buat virtual machine baru dengan **Create New Virtual Machine** (CTRL+N). Pilih **Custom** dan next.



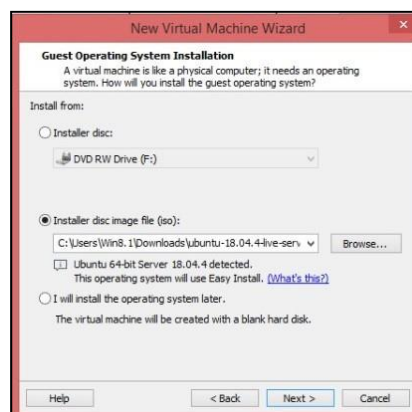
Gambar 2.2 Tampilan pilihan tipe konfigurasi

- 2) Pilih kompatibilitas hardware yang tertinggi untuk sistem operasi VM. Bila menggunakan Windows XP gunakan kompatibilitas 10.x ke bawah.



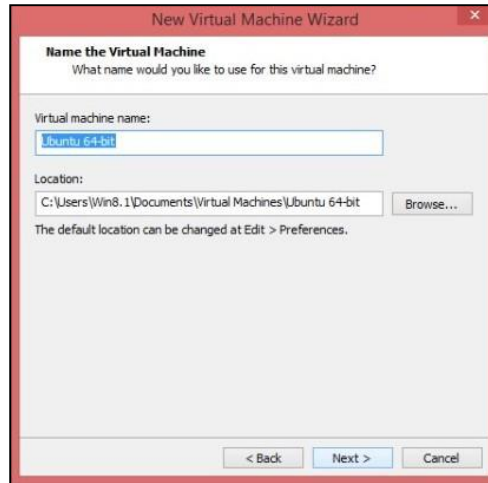
Gambar 2.3 Tampilan pilihan kompatibilitas *hardware*

- 3) Pilih berkas ISO sistem operasi untuk VM yang akan dibuat



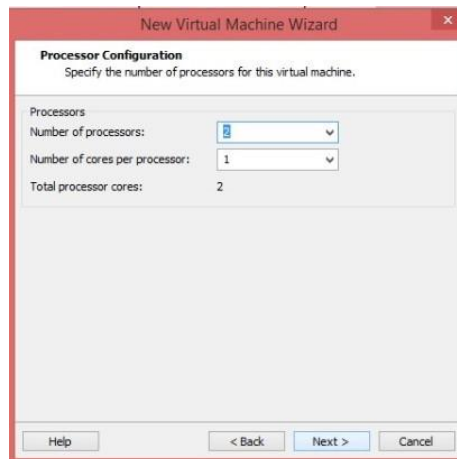
Gambar 2.4 Tampilan pilihan installer

- 4) Nama VM dapat diubah sesuai kebutuhan, simpan pada folder di C:\ atau di folder lainnya.



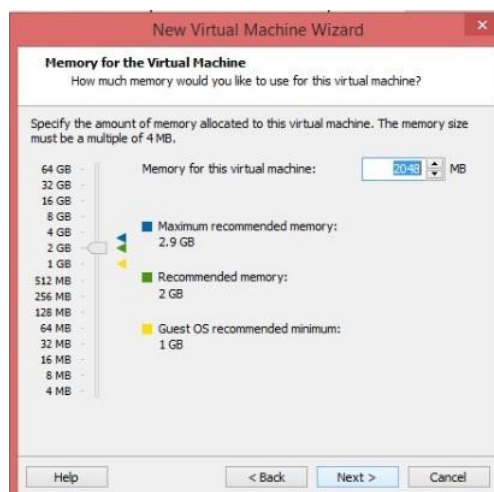
Gambar 2.5 Tampilan isi nama dan lokasi

- 5) Gunakan parameter processor sebanyak 2 dan core per processors sebanyak 1



Gambar 2.6 Tampilan isian processor dan core per processor

- 6) Gunakan parameter memori 1GB



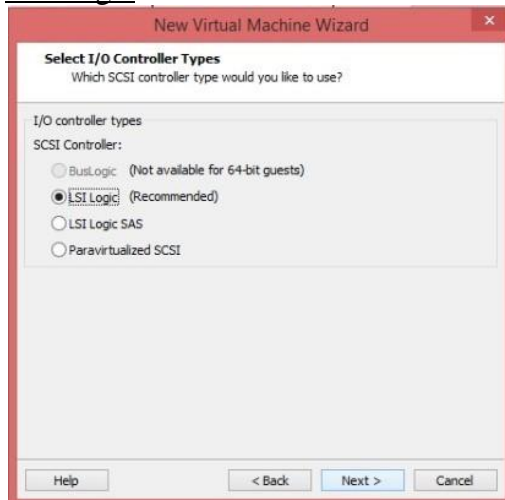
Gambar 2.7 Tampilan pilihan alokasi memori

- 7) Pada mode jaringan pilih mode NAT atau network address translation. Yang nantinya hanya dapat diakses oleh komputer host saja atau internet.



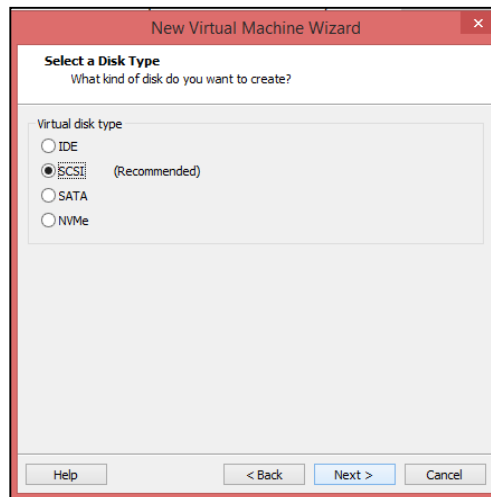
Gambar 2.8 Tampilan pilihan SCSI

- 8) Pilih controller tipe LSI Logic



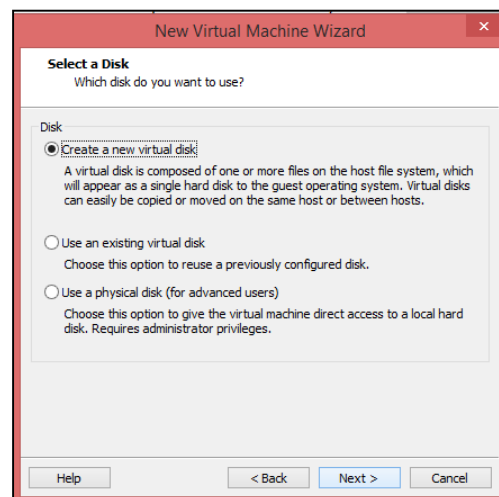
Gambar 2.9 Tampilan pilihan tipe I/O

- 9) Pilih virtual disk type SCSI (Recommended)



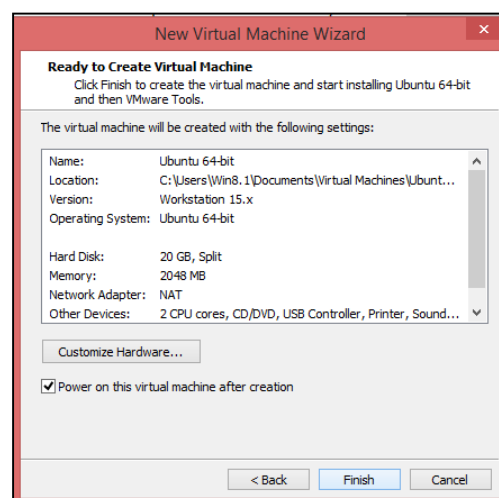
Gambar 2.10 Tampilan tipe disk

10) Pilih jenis disk, yaitu create new virtual disk



Gambar 2.11 Tampilan pilihan disk

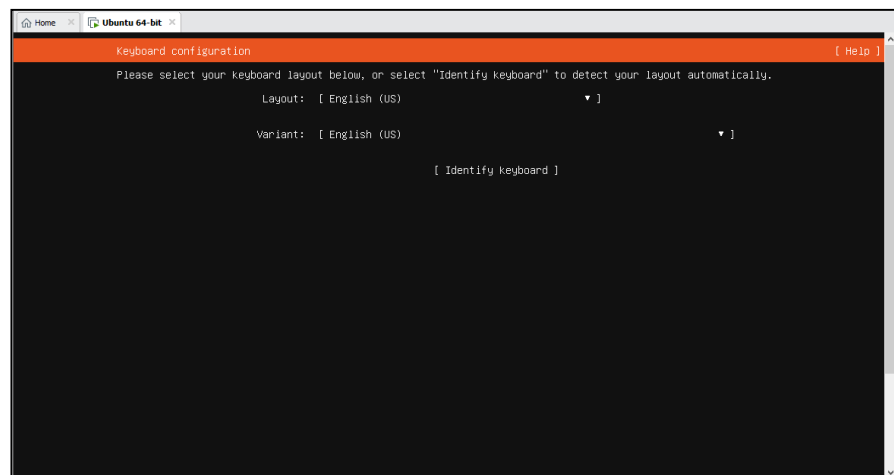
11) Klik finish jika spesifikasi sudah sesuai keinginan



Gambar 2.12 Tampilan status *create virtual machine*

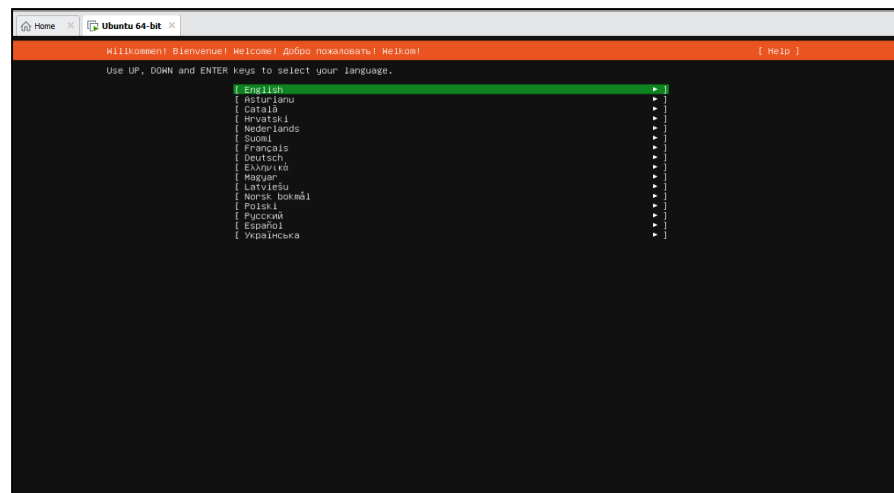
12) Setelah itu akan masuk ke proses instalasi Ubuntu. Pilih konfigurasi keyboard,

rekomendasi English agar mudah dimengerti khalayak umum.



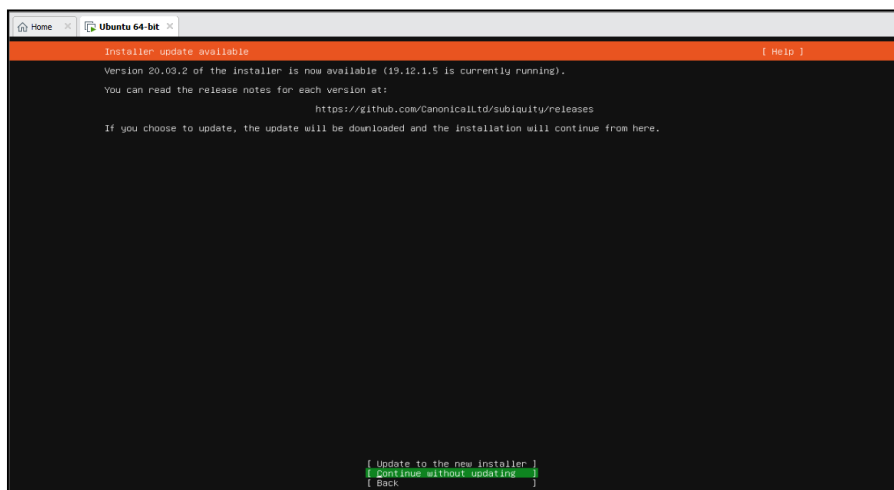
Gambar 2.13 Tampilan pilihan konfigurasi keyboard

13) Pilih bahasa, rekomendasi English agar mudah dimengerti khalayak umum



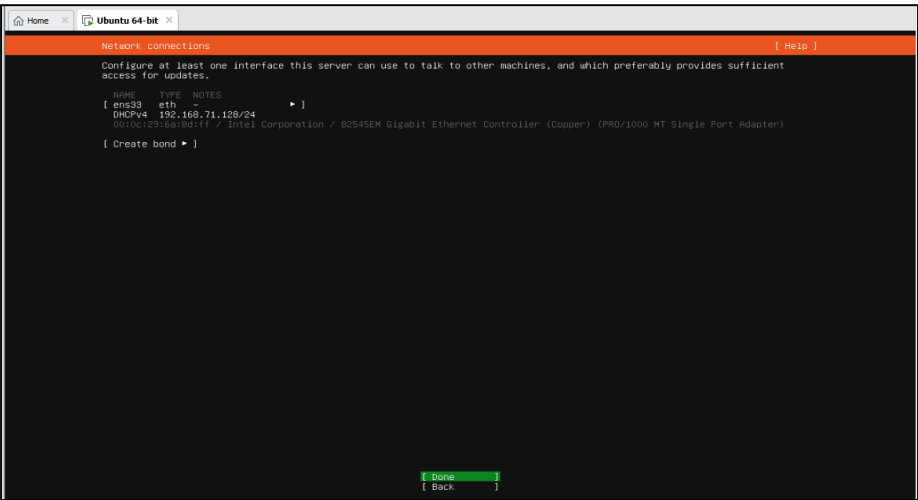
Gambar 2.14 Tampilan pilihan bahasa

14) Pilih opsi continue without update



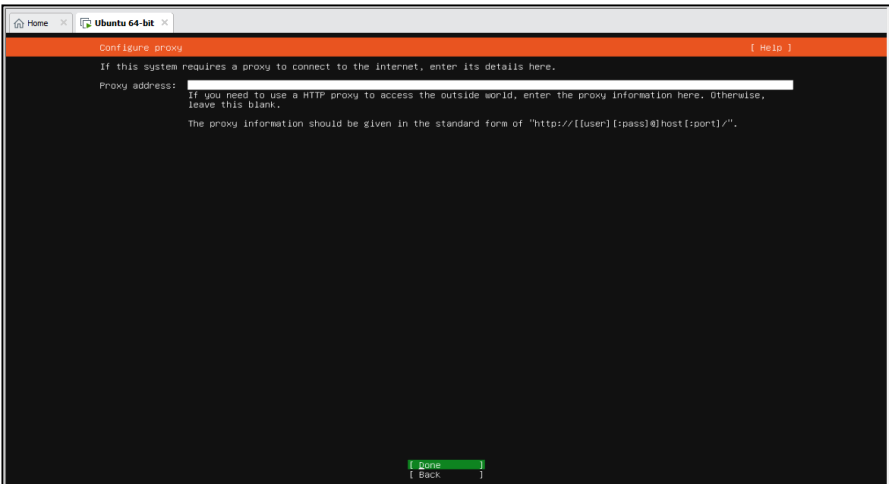
Gambar 2.15 Tampilan pilihan *update new installer*

15) Pada network connection, biarkan saja lalu pilih done



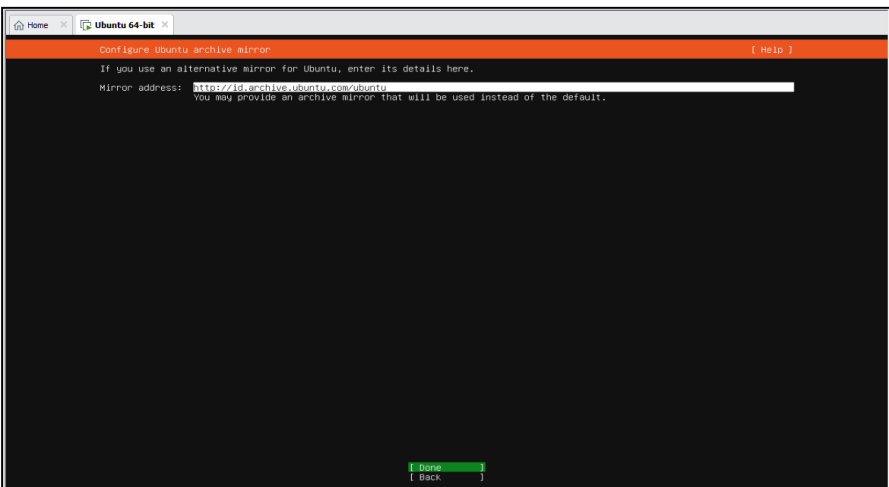
Gambar 2.16 Tampilan pilihan koneksi internet

16) Pada configure proxy, biarkan kosong lalu pilih done



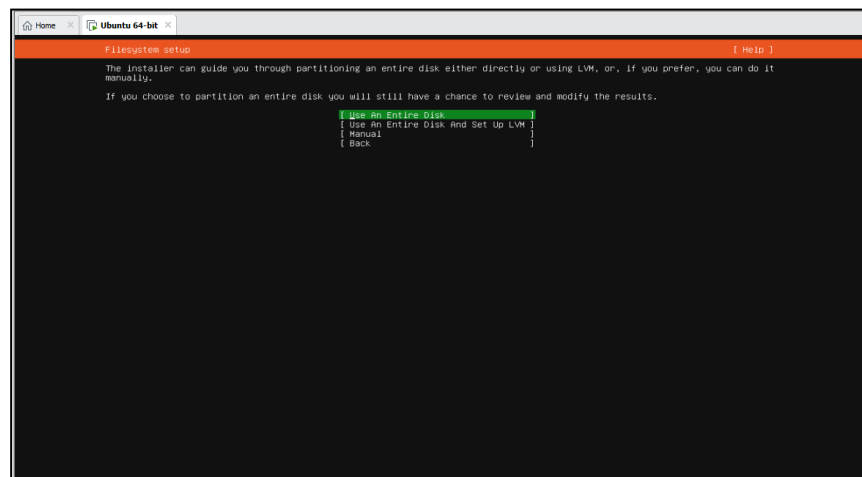
Gambar 2.17 Tampilan pilihan konfigurasi *proxy*

17) Pada Ubuntu archive mirror, biarkan saja lalu pilih done



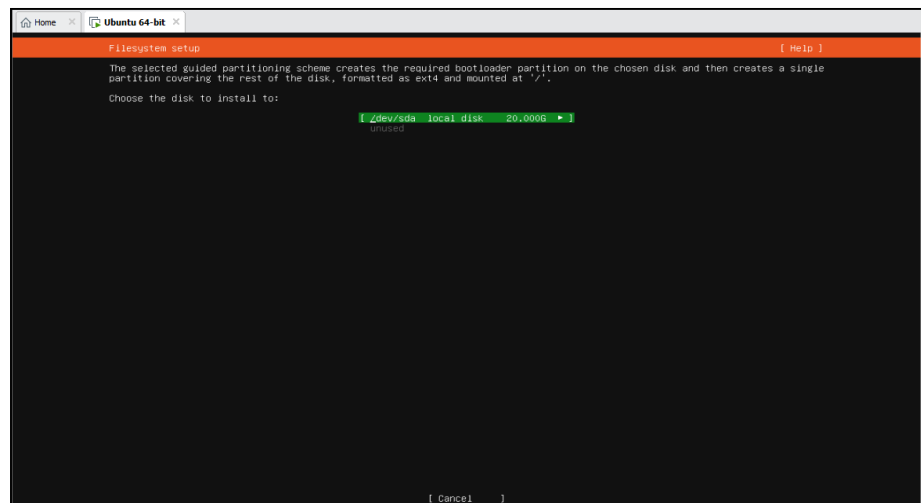
Gambar 2.18 Tampilan pilihan konfigurasi *ubuntu archive mirror*

18) Pada filesystem setup pilih use an entire disk



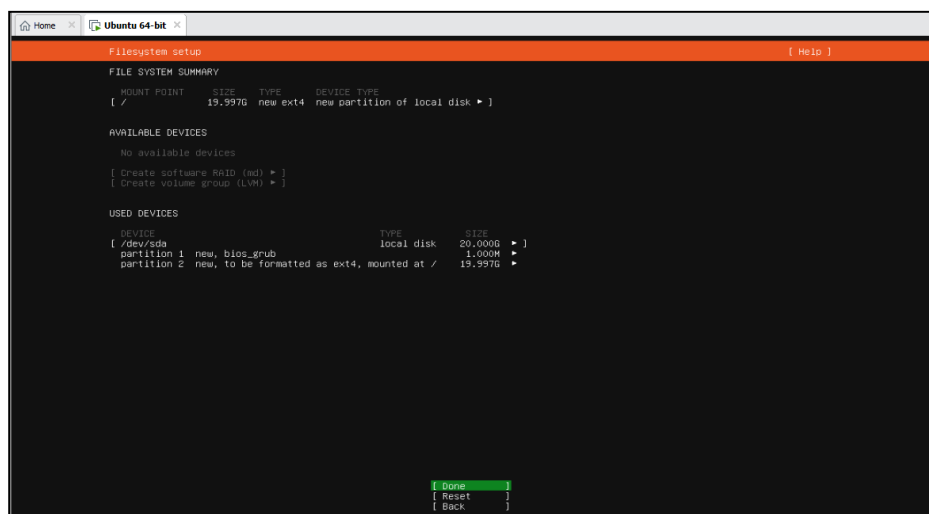
Gambar 2.19 Tampilan pilihan *Filesystem setup*

19) Lalu pilih dev/sda



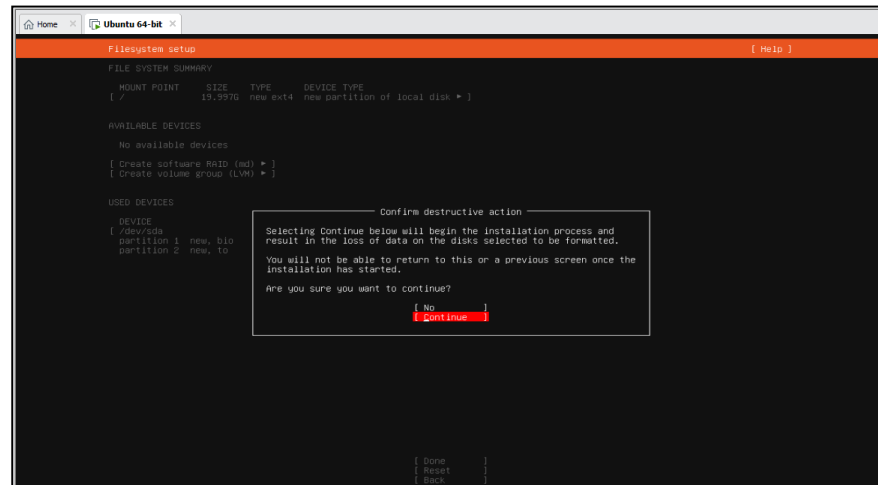
Gambar 2.20 Tampilan pilihan disk yang akan diinstal

20) Jika spesifikasi sudah sesuai yang diinginkan, pilih done



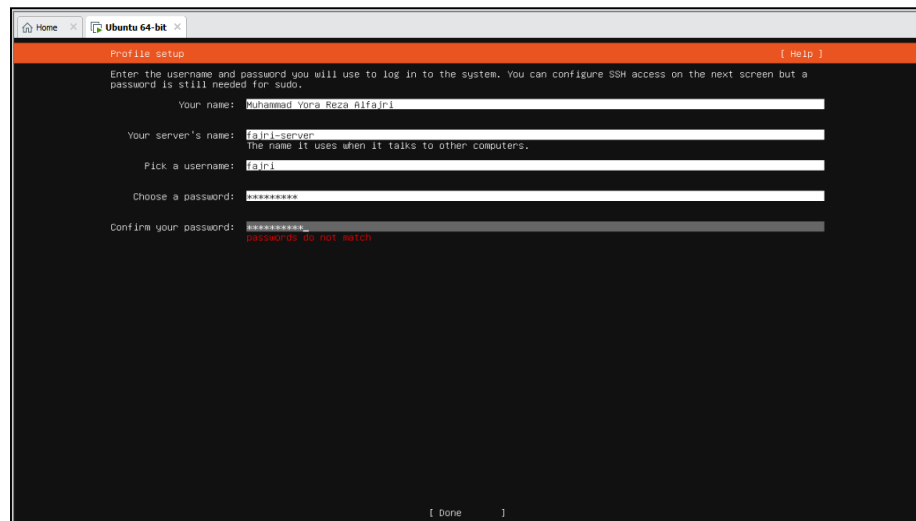
Gambar 2.21 Tampilan *Filesystem setup*

21) Pada jendela confirm destructive action, pilih continue



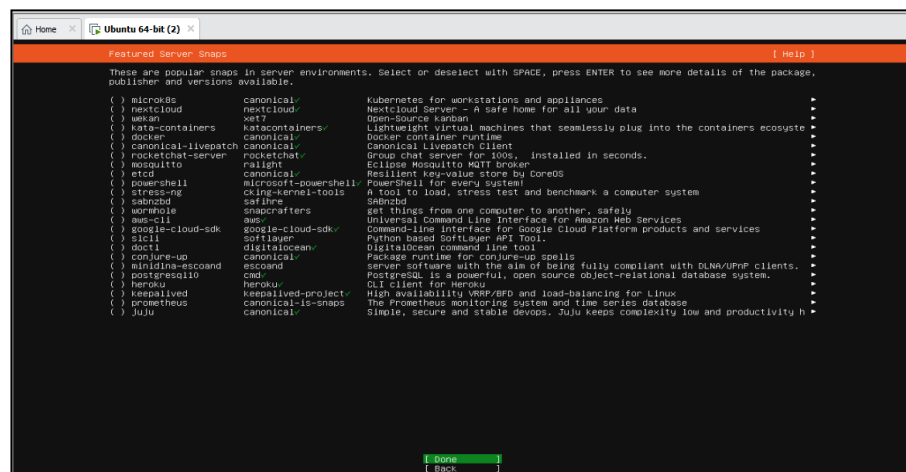
Gambar 2.22 Tampilan konfirmasi aksi

22) Isi data pada profile setup



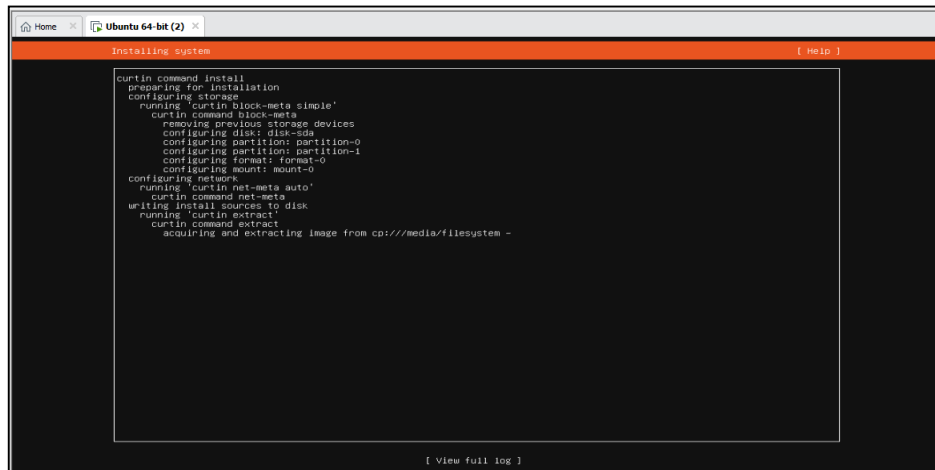
Gambar 2.23 Tampilan isi *profile setup*

23) Pada Featured server snaps langsung pilih done



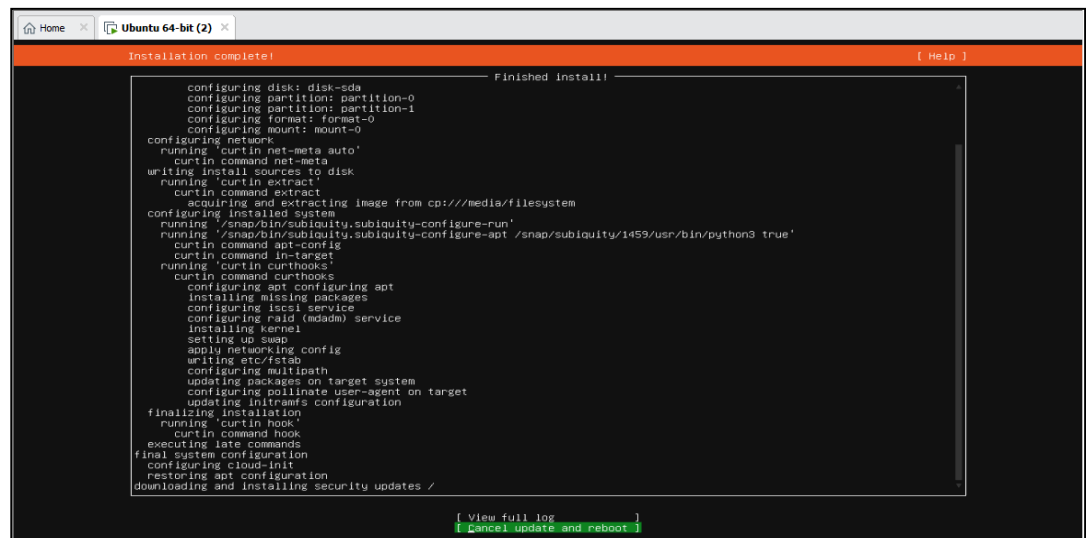
Gambar 2.24 Tampilan *featured server snaps*

24) Tunggu proses instalasi hingga selesai



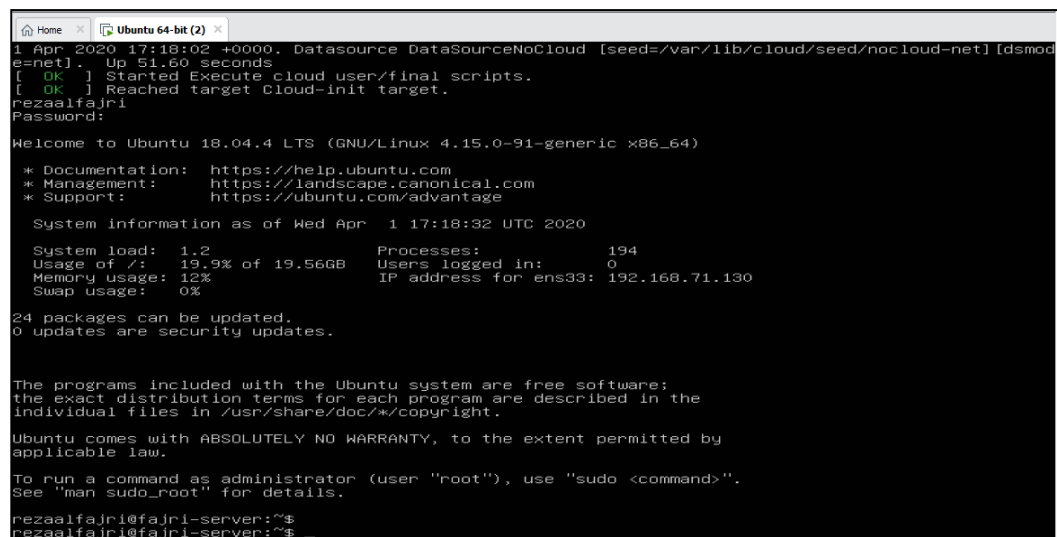
Gambar 2.25 Tampilan menunggu proses instalasi

25) Setelah instalasi selesai, pilih cancel and reboot



Gambar 2.26 Tampilan pilihan *reboot*

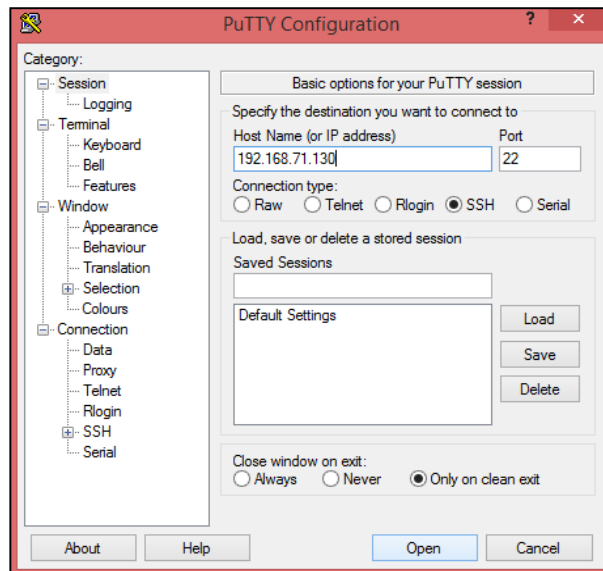
26) Setelah itu login pada Ubuntu menggunakan username dan password yang dibuat



Gambar 2.27 Tampilan setelah *login*

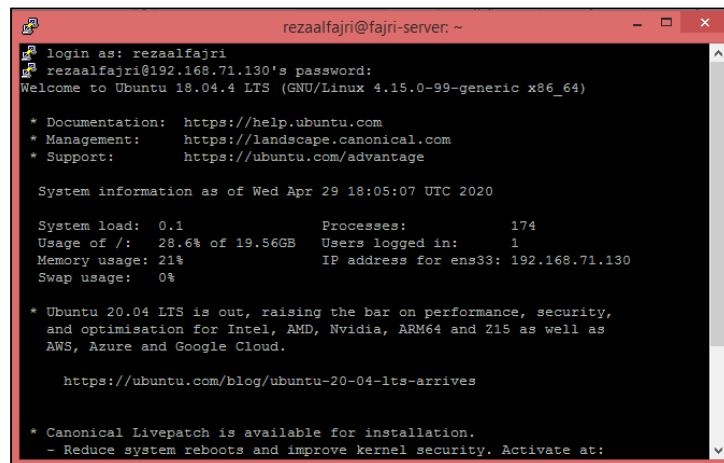
Setelah itu, buka aplikasi PUTTY di PC masing-masing. VMWare Workstation sudah dapat diminimize terlebih dahulu karena kita akan membuka virtual machine yang kita buat lewat aplikasi PUTTY. Lalu kita akan menginstall LAMPP, dengan menginstall Apache sebagai langkah pertama.

- 1) Ketik alamat IP yang dimiliki virtual machine lalu klik open



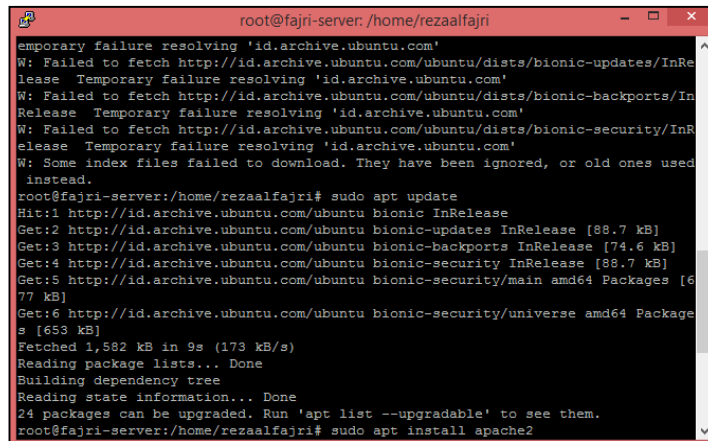
Gambar 2.28 Tampilan konfigurasi PuTTY

- 2) Setelah itu login menggunakan username dan password



Gambar 2.29 Tampilan login server pada PuTTY

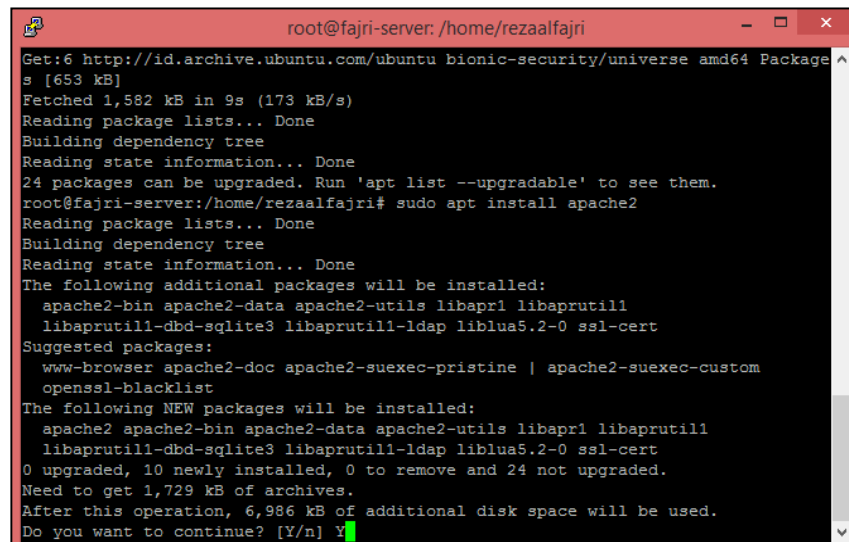
- 3) Ketikkan perintah “sudo apt install apache2” lalu tekan tombol enter



```
root@fajri-server: /home/rezaalfajri
temporary failure resolving 'id.archive.ubuntu.com'
W: Failed to fetch http://id.archive.ubuntu.com/ubuntu/dists/bionic-updates/InRelease Temporary failure resolving 'id.archive.ubuntu.com'
W: Failed to fetch http://id.archive.ubuntu.com/ubuntu/dists/bionic-backports/InRelease Temporary failure resolving 'id.archive.ubuntu.com'
W: Failed to fetch http://id.archive.ubuntu.com/ubuntu/dists/bionic-security/InRelease Temporary failure resolving 'id.archive.ubuntu.com'
W: Some index files failed to download. They have been ignored, or old ones used instead.
root@fajri-server:/home/rezaalfajri# sudo apt 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]
Get:5 http://id.archive.ubuntu.com/ubuntu bionic-security/main amd64 Packages [677 kB]
Get:6 http://id.archive.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [653 kB]
Fetched 1,582 kB in 9s (173 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
24 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@fajri-server:/home/rezaalfajri# sudo apt install apache2
```

Gambar 2.30 Tampilan PuTTY

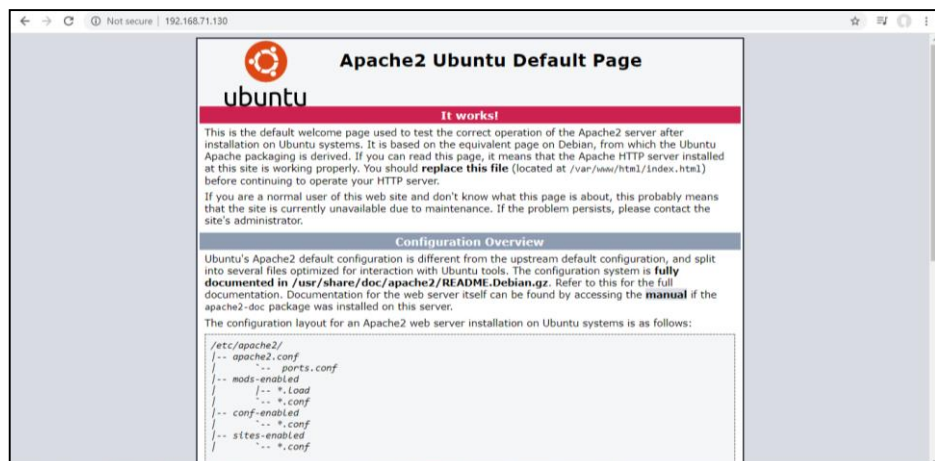
- 4) Pada pertanyaan konfirmasi ketik “Y” lalu tekan tombol enter



```
Get:6 http://id.archive.ubuntu.com/ubuntu bionic-security/universe amd64 Package s [653 kB]
Fetched 1,582 kB in 9s (173 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
24 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@fajri-server:/home/rezaalfajri# sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 ssl-cert
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom
  openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 24 not upgraded.
Need to get 1,729 kB of archives.
After this operation, 6,986 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Gambar 2.31 Tampilan PuTTY

- 5) Tunggu instalasi hingga selesai, lalu cek di browser apakah Apache sudah berjalan dengan cara ketikkan “IP” pada browser.

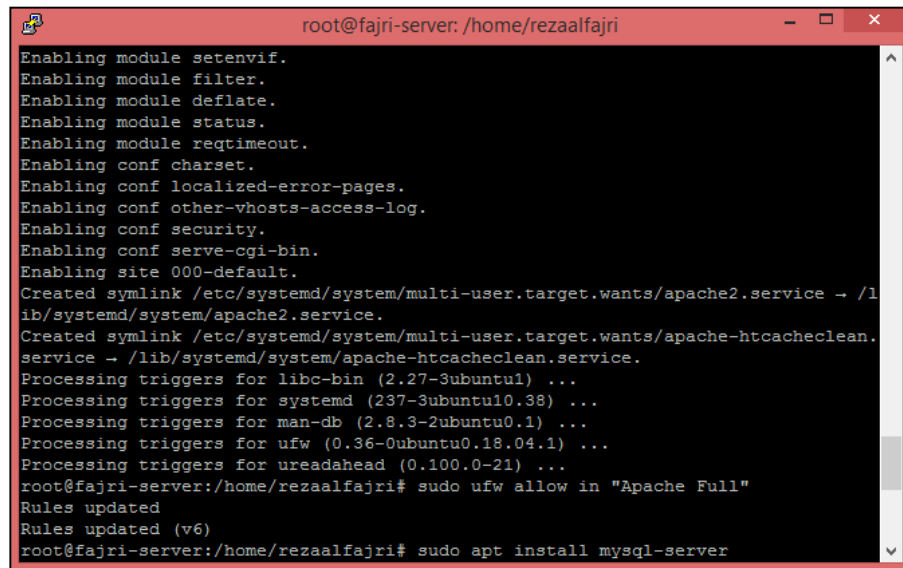


Gambar 2.32 Tampilan instalasi apache berhasil

- 6) Jika muncul tampilan seperti di atas, maka Apache sudah berjalan dengan semestinya.

Setelah kita berhasil menginstall Apache, langkah selanjutnya dalam menginstall LAMPP, kita akan menginstall mysql. Langkah-langkahnya adalah sebagai berikut :

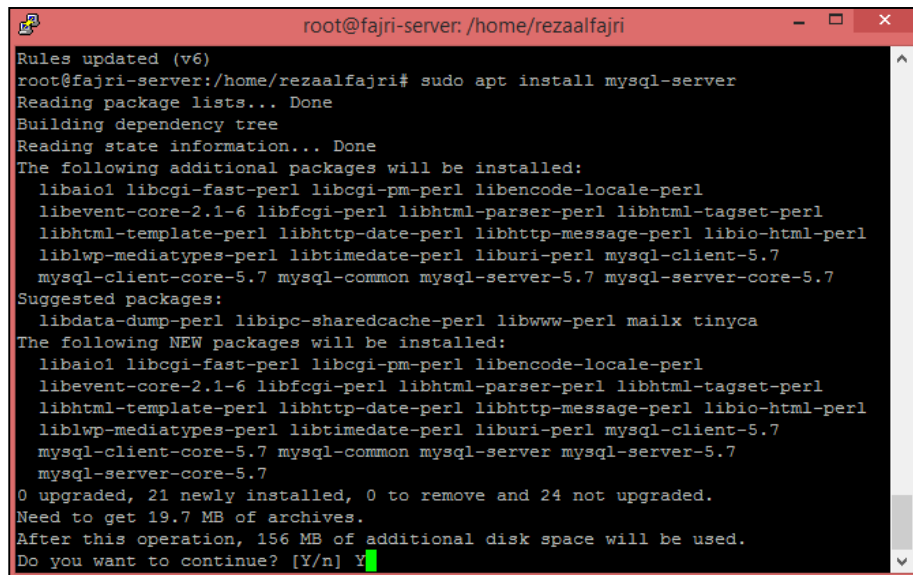
- 1) Ketikkan perintah “sudo apt install mysql-server” lalu tekan tombol enter



```
root@fajri-server: /home/rezaalfajri
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.38) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
root@fajri-server:/home/rezaalfajri# sudo ufw allow in "Apache Full"
Rules updated
Rules updated (v6)
root@fajri-server:/home/rezaalfajri# sudo apt install mysql-server
```

Gambar 2.33 Tampilan PuTTY

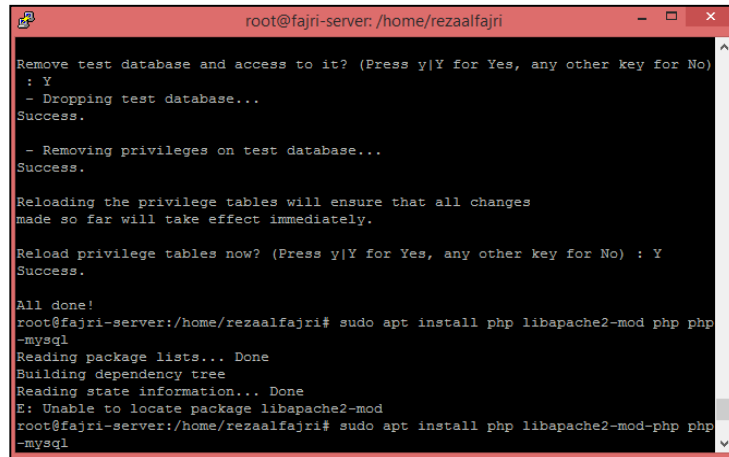
- 2) Pada jendela konfirmasi, ketik “Y” lalu tekan tombol enter



```
Rules updated (v6)
root@fajri-server:/home/rezaalfajri# sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl
  libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7
  mysql-client-core-5.7 mysql-common mysql-server-5.7 mysql-server-core-5.7
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl
  libevent-core-2.1-6 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libtimedate-perl liburi-perl mysql-client-5.7
  mysql-client-core-5.7 mysql-common mysql-server mysql-server-5.7
  mysql-server-core-5.7
0 upgraded, 21 newly installed, 0 to remove and 24 not upgraded.
Need to get 19.7 MB of archives.
After this operation, 156 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Gambar 2.33 Tampilan PuTTY

- 3) Tunggu proses instalasi hingga selesai
- 4) Setelah selesai, ketikkan perintah “sudo apt install php libapache2-mod-php php-mysql”



```
root@fajri-server: /home/rezaalfajri

Remove test database and access to it? (Press y|Y for Yes, any other key for No)
: Y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

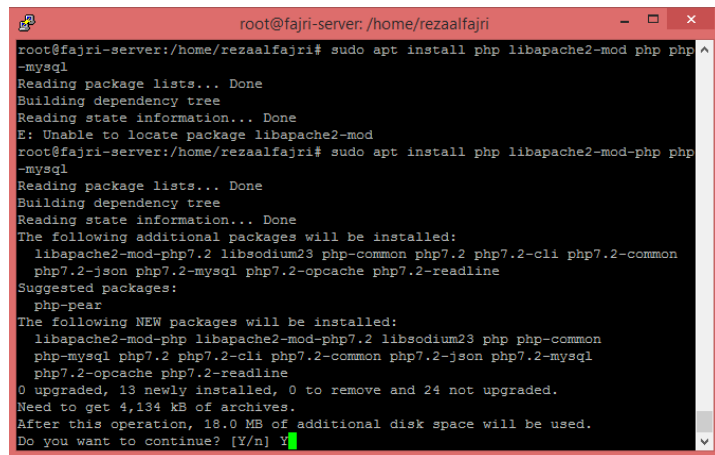
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Y
Success.

All done!
root@fajri-server:/home/rezaalfajri# sudo apt install php libapache2-mod php php
~mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package libapache2-mod
root@fajri-server:/home/rezaalfajri# sudo apt install php libapache2-mod-php php
~mysql
```

Gambar 2.34 Tampilan PuTTY

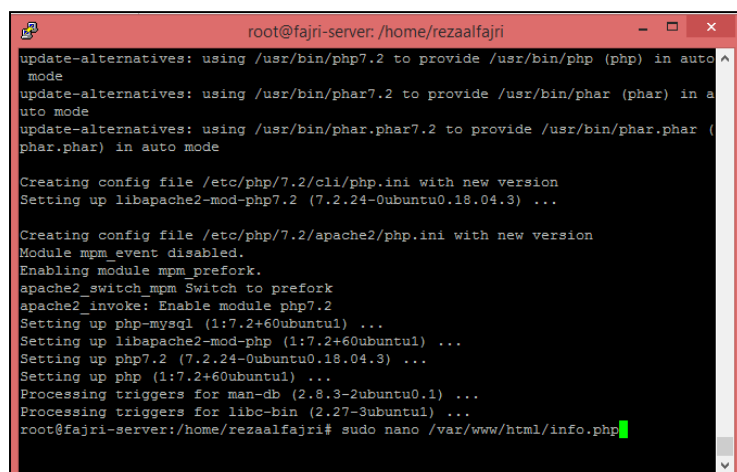
- 5) Pada jendela konfirmasi ketik “Y” lalu tekan tombol enter



```
root@fajri-server:/home/rezaalfajri# sudo apt install php libapache2-mod php php
~mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package libapache2-mod
root@fajri-server:/home/rezaalfajri# sudo apt install php libapache2-mod-php php
~mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php7.2 libsodium23 php-common php7.2 php7.2-cli php7.2-common
  php7.2-json php7.2-mysql php7.2-opcache php7.2-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php7.2 libsodium23 php php-common
  php-mysql php7.2 php7.2-cli php7.2-common php7.2-json php7.2-mysql
  php7.2-opcache php7.2-readline
0 upgraded, 13 newly installed, 0 to remove and 24 not upgraded.
Need to get 4,134 kB of archives.
After this operation, 18.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Gambar 2.34 Tampilan PuTTY

- 6) Tunggu sampai selesai instalasinya. Setelah selesai, masukkan perintah “sudo nano /var/www/html/info.php” untuk membuat file bernama “info.php” pada direktori /var/www/html/ dalam virtual machine.



```
root@fajri-server: /home/rezaalfajri

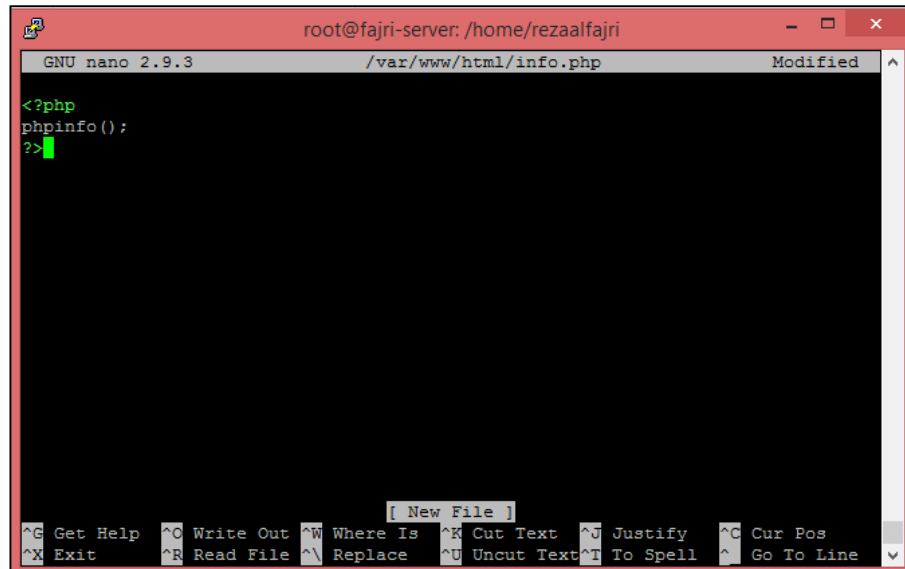
update-alternatives: using /usr/bin/php7.2 to provide /usr/bin/php (php) in auto
mode
update-alternatives: using /usr/bin/phar7.2 to provide /usr/bin/phar (phar) in a
uto mode
update-alternatives: using /usr/bin/phar.phar7.2 to provide /usr/bin/phar.phar (
phar.phar) in auto mode

Creating config file /etc/php/7.2/cli/php.ini with new version
Setting up libapache2-mod-php7.2 (7.2.24-0ubuntu0.18.04.3) ...

Creating config file /etc/php/7.2/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch mpm Switch to prefork
apache2_invoke: Enable module php7.2
Setting up php-mysql (1:7.2+60ubuntu1) ...
Setting up libapache2-mod-php (1:7.2+60ubuntu1) ...
Setting up php7.2 (7.2.24-0ubuntu0.18.04.3) ...
Setting up php (1:7.2+60ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
root@fajri-server:/home/rezaalfajri# sudo nano /var/www/html/info.php
```

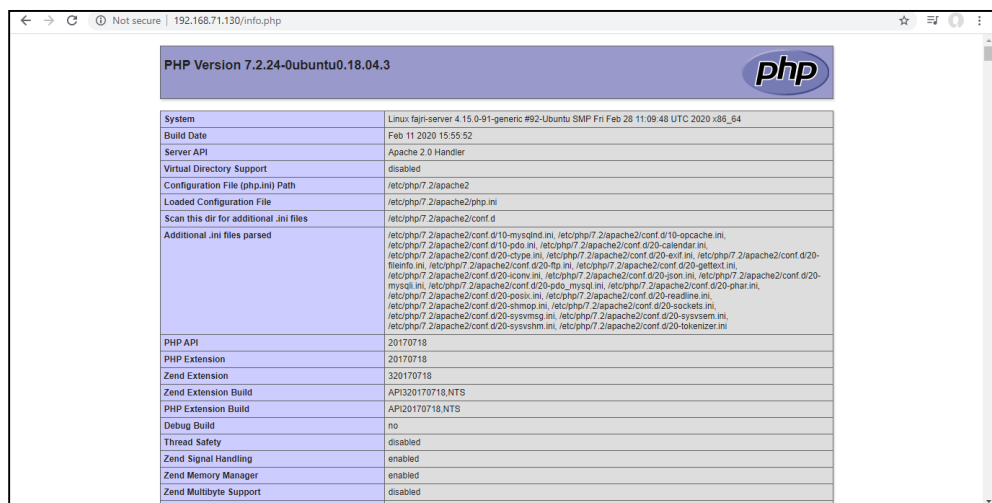
Gambar 2.35 Tampilan PuTTY

- 7) Aplikasi GNU akan terbuka, ketik “<?php phpinfo(); ?>” dalam file, lalu save.



Gambar 2.36 Tampilan PuTTY

- 8) Lalu ketikkan “IP/info.php/” pada browser untuk mengecek apakah mysql sudah berjalan atau belum.



Gambar 2.37 Tampilan phpinfo pada web server

- 9) Jika sudah muncul tampilan seperti di atas, maka mysql sudah berjalan dengan semestinya.

Langkah selanjutnya dalam menginstall LAMPP pada virtual machine yang kita miliki adalah menginstall phpmyadmin. Langkah-langkahnya adalah sebagai berikut:

- 1) Masukkan perintah “sudo apt install phpmyadmin php-mbstring php-gettext” lalu tekan enter


```
root@fajri-server: /home/rezaalfajri
update-alternatives: using /usr/bin/php7.2 to provide /usr/bin/php (php) in auto
mode
update-alternatives: using /usr/bin/phar7.2 to provide /usr/bin/phar (phar) in a
uto mode
update-alternatives: using /usr/bin/phar.phar7.2 to provide /usr/bin/phar.phar (
phar.phar) in auto mode

Creating config file /etc/php/7.2/cli/php.ini with new version
Setting up libapache2-mod-php7.2 (7.2.24-0ubuntu0.18.04.3) ...

Creating config file /etc/php/7.2/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch mpm Switch to prefork
apache2_invoke: Enable module php7.2
Setting up php-mysql (1:7.2+60ubuntu1) ...
Setting up libapache2-mod-php (1:7.2+60ubuntu1) ...
Setting up php7.2 (7.2.24-0ubuntu0.18.04.3) ...
Setting up php (1:7.2+60ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
root@fajri-server:/home/rezaalfajri# sudo nano /var/www/html/info.php
root@fajri-server:/home/rezaalfajri# sudo apt install phpmyadmin php-mbstring ph
p-gettext
```

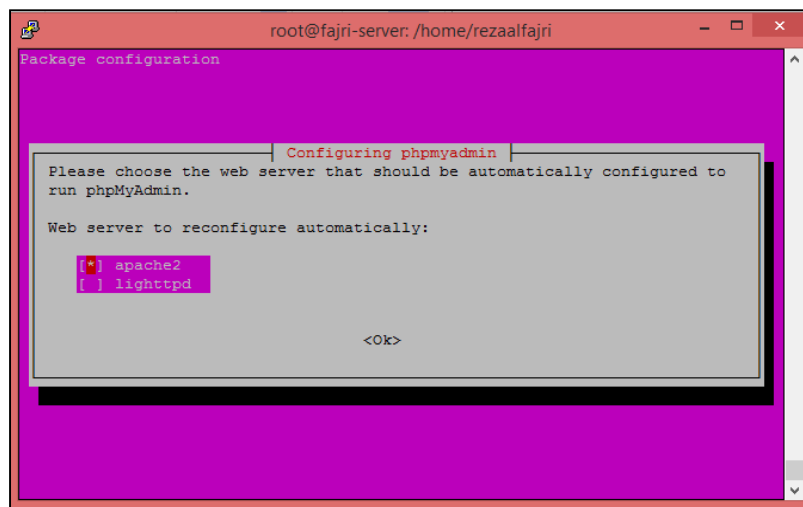
Gambar 2.38 Tampilan phpinfo pada web server

- 2) Ketikkan “Y” lalu tekan tombol enter pada jendela konfirmasi

```
root@fajri-server: /home/rezaalfajri
p-gettext
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  dbconfig-common dbconfig-mysql fontconfig-config fonts-dejavu-core
  javascript-common libfontconfig1 libgd3 libjpeg8 libjpeg-turbo8 libjpeg9
  libjs-jquery libjs-sphinxdoc libjs-underscore libtiff5 libwebp6 libxpm4
  libzip4 php-bz2 php-curl php-gd php-pear php-php-gettext php-phpseclib
  php-tcpdf php-xml php-zip php7.2-bz2 php7.2-curl php7.2-gd php7.2-mbstring
  php7.2-xml php7.2-zip
Suggested packages:
  libgd-tools php-libsodium php-mcrypt php-gmp php-imagick www-browser
The following NEW packages will be installed:
  dbconfig-common dbconfig-mysql fontconfig-config fonts-dejavu-core
  javascript-common libfontconfig1 libgd3 libjpeg8 libjpeg-turbo8 libjpeg9
  libjs-jquery libjs-sphinxdoc libjs-underscore libtiff5 libwebp6 libxpm4
  libzip4 php-bz2 php-curl php-gd php-gettext php-mbstring php-pear
  php-php-gettext php-phpseclib php-tcpdf php-xml php-zip php7.2-bz2
  php7.2-curl php7.2-gd php7.2-mbstring php7.2-xml php7.2-zip phpmyadmin
0 upgraded, 35 newly installed, 0 to remove and 24 not upgraded.
Need to get 15.6 MB of archives.
After this operation, 60.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

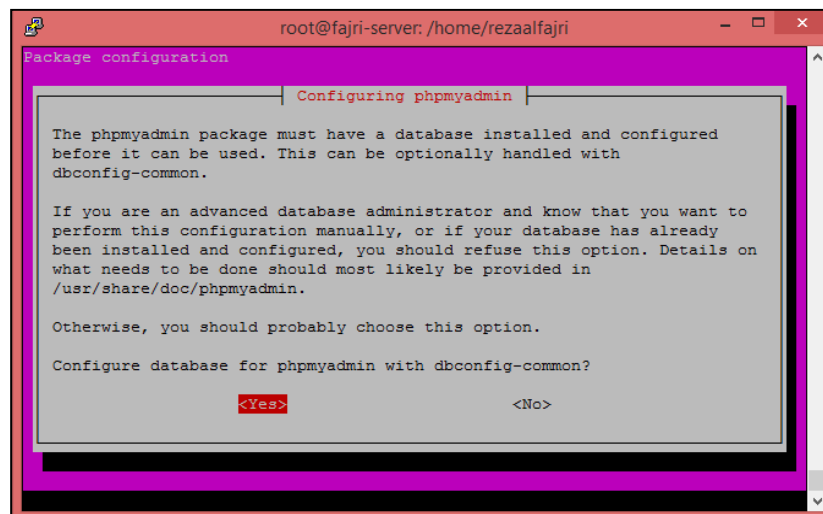
Gambar 2.39 Tampilan phpinfo pada web server

- 3) Tunggu hingga selesai instalasinya. Lalu akan muncul jendela configuring phpmyadmin, pada opsi Apache2, tekan tombol spasi, lalu tekan tombol enter.



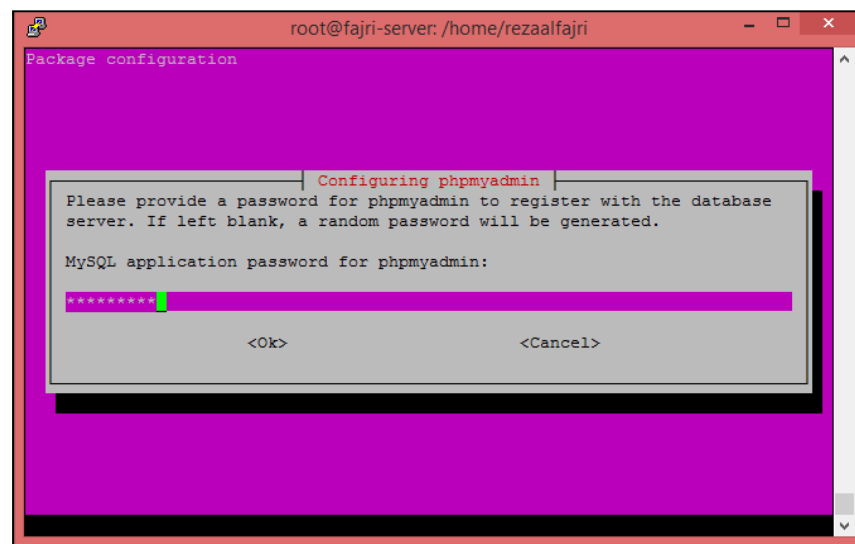
Gambar 2.40 Tampilan PuTTY

- 4) Pilih yes pada halaman selanjutnya



Gambar 2.41 Tampilan PuTTY

- 5) Buat password yang mudah diingat lalu konfirmasi



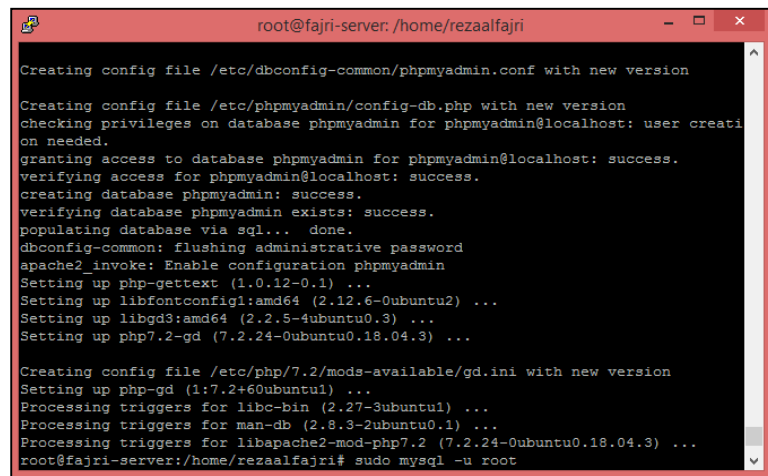
Gambar 2.42 Tampilan PuTTY

- 6) Untuk mengetahui apakah phpmyadmin sudah bisa digunakan, ketik "IP/phpmyadmin" pada browser
- 7) Jika muncul tampilan seperti di bawah, coba untuk login menggunakan password yang dibuat



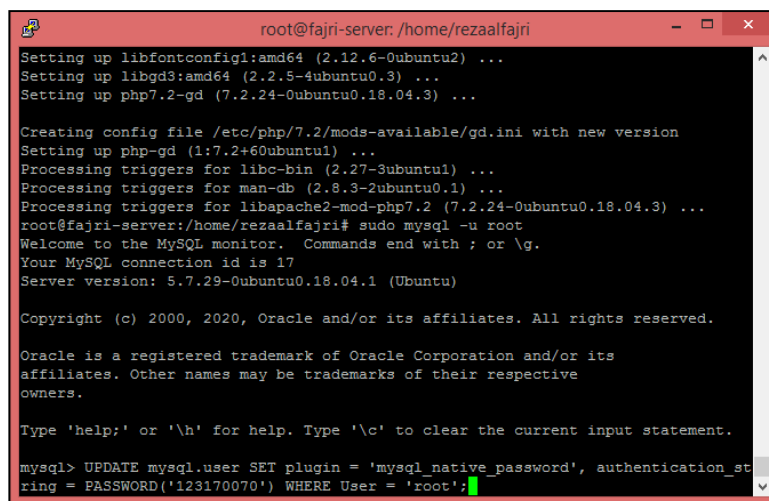
Gambar 2.43 Tampilan login phpmyadmin

- 8) Akan terjadi eror seperti di atas. Untuk memperbaikinya, kembali ke PUTTY dan masukkan perintah “sudo mysql –u root” untuk masuk ke mysql



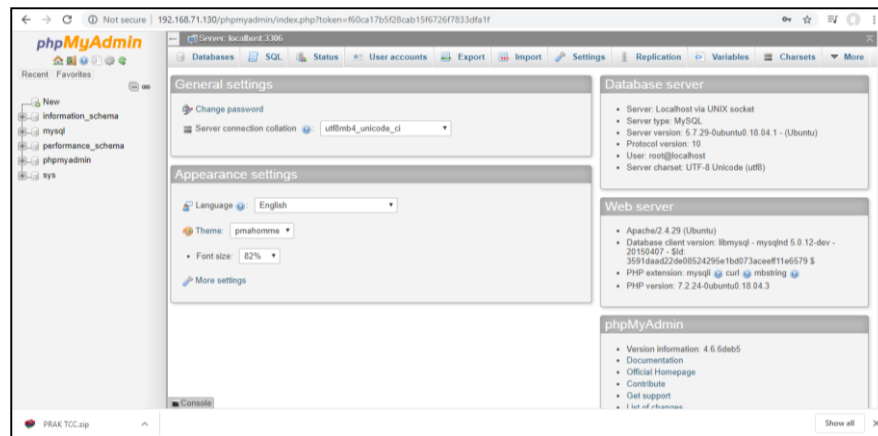
Gambar 2.44 Tampilan PuTTY

- 9) Setelah masuk ke mysql, maka masukkan perintah “UPDATE mysql.user SET plugin = ‘mysql_native_password’, authentication_string = PASSWORD(‘passwordkamu’) WHERE User = ‘root’”



Gambar 2.45 Tampilan PuTTY

- 10) Setelah itu, masukkan perintah “FLUSH PRIVILEGES;” lalu tekan enter.
Setelah itu masukkan perintah exit untuk kembali ke root
- 11) Lalu coba login menggunakan password yang dibuat



Gambar 2.46 Tampilan dashboard phpmyadmin

- 12) Jika tampilan seperti di atas sudah muncul, maka phpmyadmin sudah berjalan sebagaimana mestinya.

Langkah berikutnya mengupload berkas dengan WinSCP dengan tahapan sebagai berikut:

- 1) Instal WinSCP terlebih dahulu
- 2) Masukkan alamat IP, pilih SFTP sebagai file protocol, isi 22 pada port number, isi username dan password.
- 3) Akan muncul peringatan konfirmasi, pilih yes
- 4) WinSCP akan menampilkan folder-folder di sebelah kiri yang berarti itu adalah folder yang ada dalam perangkat yang digunakan, sedangkan sisi sebelah kanan adalah folder berkas yang di upload ke server.
- 5) Jika menggunakan code igniter sebagai framework dari proyek yang akan diupload ke server, buka folder application, lalu buka config.php, edit bagian base url dengan IP/nama_folder_projek/index.php/

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

/*
|-----|
| Base Site URL
|-----|
|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
| http://example.com/
|
| WARNING: You MUST set this value!
|
| If it is not set, then CodeIgniter will try guess the protocol and path
| your installation, but due to security concerns the hostname will be set
| to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
| The auto-detection mechanism exists only for convenience during
| development and MUST NOT be used in production!
|
| If you need to allow multiple domains, remember that this file is still
| a PHP script and you can easily do that on your own.
|-----|
*/
$config['base_url'] = 'http://192.168.71.130/eres/index.php';

/*
|-----|
| Index File
|-----|
*/

```

Gambar 2.47 Tampilan isi syntax *config.php*

- 6) Buka database.php, lalu ubah password dengan password yang telah dibuat

```

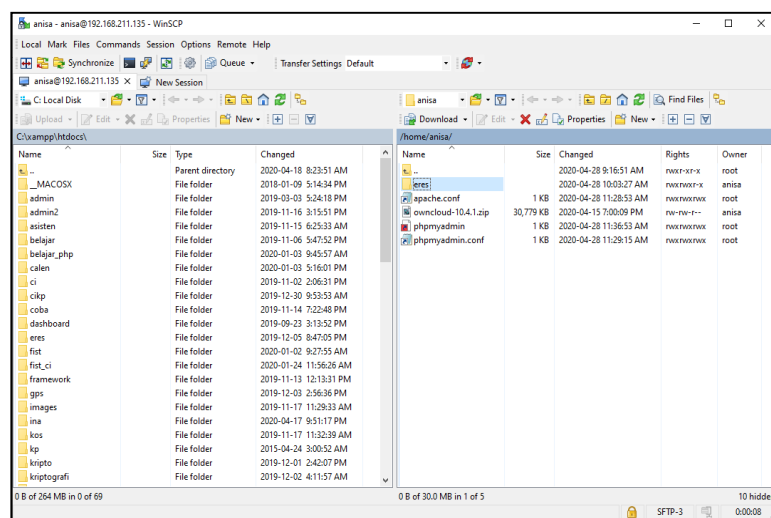
/*
| make active. By default there is only one group (the 'default' group).
|
| The $query_builder variable lets you determine whether or not to load
| the query builder class.
|-----|
*/
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '1234567890',
    'database' => 'eres',
    'dbdriver' => 'mysql',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);

```

Gambar 2.48 Tampilan isi syntax *database.php*

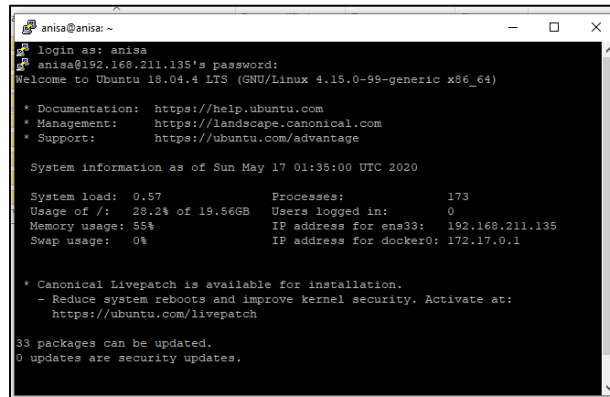
- 7) Drag and drop folder berkas yang akan diupload dari sisi sebelah kiri ke sisi sebelah kanan, maka berkas dalam folder akan terupload ke server. Proyek dapat diakses dengan IP/index.php/nama_controller/fungsi



Gambar 2.49 Tampilan WinSCP

Proses *upload* Sistem Informasi Pendaftaran Antrian Pasien pada Jadwal Praktek Dokter ke dalam Ubuntu dengan LAMPP telah selesai, selanjutnya adalah proses manajemen data dengan docker. Tahapannya sebagai berikut :

- 1) Hidupkan VM
- 2) Login dengan PuTTY

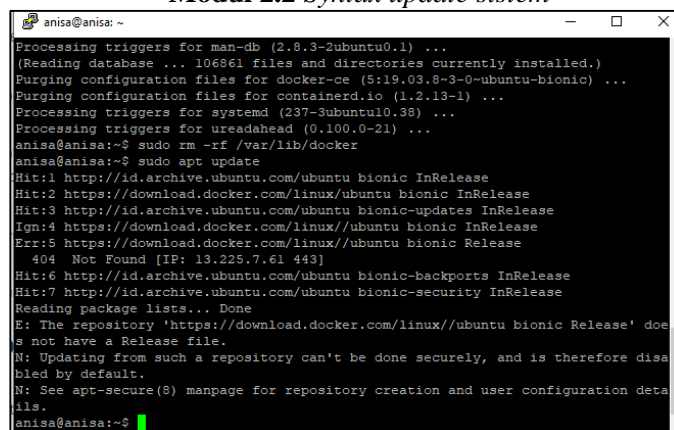


Gambar 2.50 Tampilan login PuTTY

- 3) Perbarui sistem menggunakan *syntax* dibawah ini, masukkan perintah Y untuk melanjutkan upgrade.

```
$ sudo apt update  
$ sudo apt upgrade
```

Modul 2.2 Syntax update sistem



Gambar 2.51 Tampilan proses update

- 4) Install beberapa package untuk menginstall docker dengan perintah, lanjutkan dengan Y

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

Modul 2.3 Syntax install package docker

```

rezaalfajri@fajri-server: ~
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20180409).
ca-certificates set to manually installed.
curl is already the newest version (7.58.0-2ubuntu3.8).
curl set to manually installed.
software-properties-common is already the newest version (0.96.24.32.12).
software-properties-common set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,692 B of archives.
After this operation, 153 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://id.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 apt-transport-https all 1.6.12 [1,692 B]
Fetched 1,692 B in 1s (1,405 B/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 106765 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.6.12_all.deb ...
Unpacking apt-transport-https (1.6.12) ...
Setting up apt-transport-https (1.6.12) ...
rezaalfajri@fajri-server:~$

```

Gambar 2.52 Tampilan install package docker

- 5) Tambahkan kunci GPG dengan memasukkan perintah

```

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

```

Modul 2.4 Syntax kunci GPG

```

rezaalfajri@fajri-server: ~
ca-certificates set to manually installed.
curl is already the newest version (7.58.0-2ubuntu3.8).
curl set to manually installed.
software-properties-common is already the newest version (0.96.24.32.12).
software-properties-common set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,692 B of archives.
After this operation, 153 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://id.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 apt-transport-https all 1.6.12 [1,692 B]
Fetched 1,692 B in 1s (1,405 B/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 106765 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.6.12_all.deb ...
Unpacking apt-transport-https (1.6.12) ...
Setting up apt-transport-https (1.6.12) ...
rezaalfajri@fajri-server:~$ curl -fsSL https://download.docker.com/linux/ubuntu/
gpg | sudo apt-key add -
OK
rezaalfajri@fajri-server:~$ sudo add-apt-repository "deb [arch=amd64] https://do
wnload.docker.com/linux/ubuntu $(lsb_release -cs) stable"

```

Gambar 2.53 Syntax tambah kunci GPG

- 6) Tambahkan repository Docker dengan memasukkan perintah

```

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

```

Modul 2.5 Syntax tambah repository docker

```

rezaalfajri@fajri-server: ~
ca-certificates set to manually installed.
curl is already the newest version (7.58.0-2ubuntu3.8).
curl set to manually installed.
software-properties-common is already the newest version (0.96.24.32.12).
software-properties-common set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 1,692 B of archives.
After this operation, 153 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://id.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 apt-transport-https all 1.6.12 [1,692 B]
Fetched 1,692 B in 1s (1,405 B/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 106765 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_1.6.12_all.deb ...
Unpacking apt-transport-https (1.6.12) ...
Setting up apt-transport-https (1.6.12) ...
rezaalfajri@fajri-server:~$ curl -fsSL https://download.docker.com/linux/ubuntu/
gpg | sudo apt-key add -
OK
rezaalfajri@fajri-server:~$ sudo add-apt-repository "deb [arch=amd64] https://do
wnload.docker.com/linux/ubuntu $(lsb_release -cs) stable"

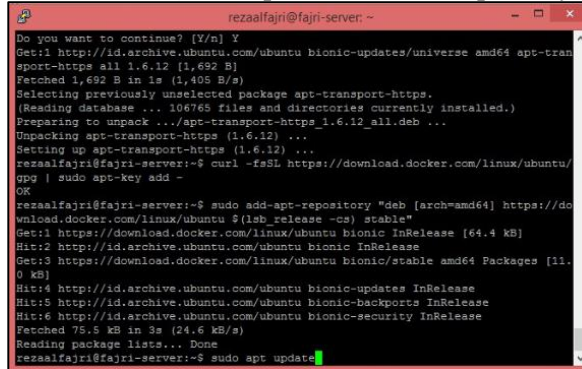
```

Gambar 2.53 Tampilan syntax tambah kunci GPG

- 7) Lalu update informasi repository dengan memasukkan perintah

```
$ sudo apt update
```

Modul 2.6 Syntax update informasi repository



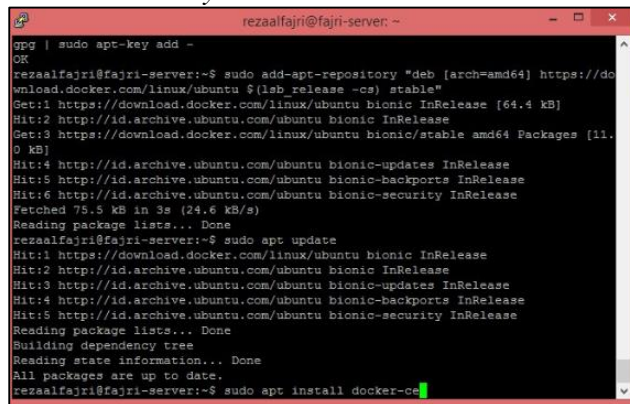
```
rezaalfajri@fajri-server: ~  
Do you want to continue? [Y/n] Y  
Get:1 http://id.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 apt-trans-  
port-https all 1.6.12 [1,692 B]  
Fetched 1,692 B in 1s (1,405 B/s)  
Selecting previously unselected package apt-transports-https.  
(Reading database ... 106765 files and directories currently installed.)  
Preparing to unpack .../apt-transports-https_1.6.12_all.deb ...  
Unpacking apt-transports-https (1.6.12) ...  
Setting up apt-transports-https (1.6.12) ...  
rezaalfajri@fajri-server:~$ curl -fsSL https://download.docker.com/linux/ubuntu/  
gpg | sudo apt-key add -  
OK  
rezaalfajri@fajri-server:~$ sudo add-apt-repository "deb [arch=amd64] https://do  
wnload.docker.com/linux/ubuntu $(lsb_release -cs) stable"  
Get:1 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]  
Hit:2 http://id.archive.ubuntu.com/ubuntu bionic InRelease  
Get:3 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [11.  
0 kB]  
Hit:4 http://id.archive.ubuntu.com/ubuntu bionic-updates InRelease  
Hit:5 http://id.archive.ubuntu.com/ubuntu bionic-backports InRelease  
Hit:6 http://id.archive.ubuntu.com/ubuntu bionic-security InRelease  
Fetched 75.5 kB in 3s (24.6 kB/s)  
Reading package lists... Done  
rezaalfajri@fajri-server:~$ sudo apt update
```

Gambar 2.54 Tampilan *syntax update repository*

- 8) Install docker dengan memasukkan perintah, lanjutkan dengan memasukkan Y pada pertanyaan konfirmasi dan tunggu proses instalasi selesai

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

Modul 2.7 Syntax install docker



```
rezaalfajri@fajri-server: ~  
gpg | sudo apt-key add -  
OK  
rezaalfajri@fajri-server:~$ sudo add-apt-repository "deb [arch=amd64] https://do  
wnload.docker.com/linux/ubuntu $(lsb_release -cs) stable"  
Get:1 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]  
Hit:2 http://id.archive.ubuntu.com/ubuntu bionic InRelease  
Get:3 https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages [11.  
0 kB]  
Hit:4 http://id.archive.ubuntu.com/ubuntu bionic-updates InRelease  
Hit:5 http://id.archive.ubuntu.com/ubuntu bionic-backports InRelease  
Hit:6 http://id.archive.ubuntu.com/ubuntu bionic-security InRelease  
Fetched 75.5 kB in 3s (24.6 kB/s)  
Reading package lists... Done  
rezaalfajri@fajri-server:~$ sudo apt update  
Hit:1 https://download.docker.com/linux/ubuntu bionic InRelease  
Hit:2 http://id.archive.ubuntu.com/ubuntu bionic InRelease  
Hit:3 http://id.archive.ubuntu.com/ubuntu bionic-updates InRelease  
Hit:4 http://id.archive.ubuntu.com/ubuntu bionic-backports InRelease  
Hit:5 http://id.archive.ubuntu.com/ubuntu bionic-security InRelease  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
All packages are up to date.  
rezaalfajri@fajri-server:~$ sudo apt install docker-ce
```

Gambar 2.55 Tampilan *syntax install docker*

- 9) Periksa status docker dengan menggunakan perintah, jika sudah seperti pada gambar di bawah ini maka docker sudah berhasil berjalan

```
$ sudo systemctl status docker
```

Modul 2.8 Syntax periksa status docker


```

rezaalfajri@fajri-server: ~
Unpacking docker-ce (5:19.03.8~3-0-ubuntu-bionic) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../6-libltdl7_2.4.6-2_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-2) ...
Setting up aufs-tools (1:4.9+20170918-1ubuntu1) ...
Setting up containerd.io (1.2.13-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service -> /lib/systemd/system/containerd.service.
Setting up cgroupfs-mount (1.4) ...
Setting up libltdl7:amd64 (2.4.6-2) ...
Setting up docker-ce-cli (5:19.03.8~3-0-ubuntu-bionic) ...
Setting up pigz (2.4-1) ...
Setting up docker-ce (5:19.03.8~3-0-ubuntu-bionic) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service -> /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket -> /lib/systemd/system/docker.socket.
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu1.0.39) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
rezaalfajri@fajri-server:~$ sudo systemctl status docker

```

Gambar 2.56 Tampilan *syntax* periksa status docker

10) Selanjutnya adalah proses install docker compose dengan perintah

```

$ sudo curl -L
"https://github.com/docker/compose/releases/download/1.24.1/
docker-compose-$ (uname -s) - $ (uname -m) " -o

```

Modul 2.9 Syntax install docker compose

```

rezaalfajri@fajri-server: ~
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/

rezaalfajri@fajri-server:~$ sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/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    133      0  0:00:04  0:00:04 --:--:--  133
3 10.3M   3 372k    0     0 10520      0  0:17:12  0:00:36  0:16:36 13855

```

Gambar 2.57 Tampilan proses *install docker compose*

11) Jika proses instalasi telah selesai, lanjutkan dengan memasukkan perintah

```

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

```

Modul 2.10 Syntax pada PuTTY

```

rezaalfajri@fajri-server: ~
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/

rezaalfajri@fajri-server:~$ sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/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    133      0  0:00:04  0:00:04 --:--:--  133
100 10.3M 100 10.3M    0     0 20297      0  0:08:54  0:08:54 --:--:-- 46581
rezaalfajri@fajri-server:~$ sudo chmod +x /usr/local/bin/docker-compose
rezaalfajri@fajri-server:~$ docker-compose --version
docker-compose version 1.21.2, build a133471
rezaalfajri@fajri-server:~$

```

Gambar 2.58 Tampilan proses PuTTY

12) Cek versi dari Docker Compose dengan memasukkan perintah

```

$ docker-compose version

```

Modul 2.11 Syntax cek versi Docker Compose

```
rezaalfajri@fajri-server: ~  
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/  
  
rezaalfajri@fajri-server:~$ sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose -u $USER -o /usr/local/bin/docker-compose  
rezaalfajri@fajri-server:~$ docker-compose --version  
docker-compose version 1.21.2, build a133471  
rezaalfajri@fajri-server:~$
```

Gambar 2.59 Tampilan cek versi docker compose

- 13) Buat direktori untuk penguploadan berkas proyek dengan perintah

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

Modul 2.12 Syntax pembuatan direktori

```
rezaalfajri@fajri-server: ~  
and optimisation for Intel, AMD, Nvidia, ARM64 and ZIS as well as  
AWS, Azure and Google Cloud.  
  
https://ubuntu.com/blog/ubuntu-20-04-lts-arrives  
  
* Canonical Livepatch is available for installation.  
- Reduce system reboots and improve kernel security. Activate at:  
https://ubuntu.com/livepatch  
  
9 packages can be updated.  
0 updates are security updates.  
  
Last login: Thu May 14 13:36:57 2020  
rezaalfajri@fajri-server:~$ docker -compose version  
unable to resolve docker endpoint: context "compose" does not exist  
rezaalfajri@fajri-server:~$ docker-compose version  
docker-compose version 1.21.2, build a133471  
docker-py version: 3.3.0  
CPython version: 3.6.5  
OpenSSL version: OpenSSL 1.0.1t  3 May 2016  
rezaalfajri@fajri-server:~$ mkdir -p ~/docker/lamp/html  
rezaalfajri@fajri-server:~$
```

Gambar 2.60 Tampilan buat direktori

- 14) Arahkan ke direktori tersebut dengan perintah

```
$ cd ~/docker/lamp
```

Modul 2.13 Syntax mengarahkan direktori

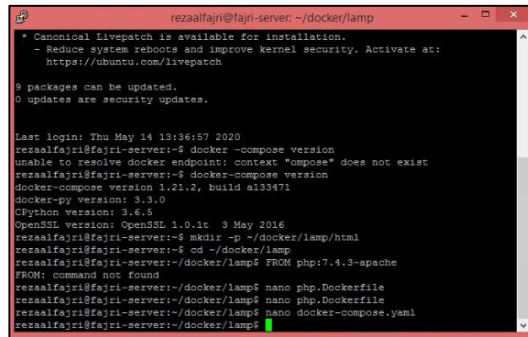
```
rezaalfajri@fajri-server: ~/docker/lamp  
* Canonical Livepatch is available for installation.  
- Reduce system reboots and improve kernel security. Activate at:  
https://ubuntu.com/livepatch  
  
9 packages can be updated.  
0 updates are security updates.  
  
Last login: Thu May 14 13:36:57 2020  
rezaalfajri@fajri-server:~/docker/lamp$ docker -compose version  
unable to resolve docker endpoint: context "compose" does not exist  
rezaalfajri@fajri-server:~/docker/lamp$ docker-compose version  
docker-compose version 1.21.2, build a133471  
docker-py version: 3.3.0  
CPython version: 3.6.5  
OpenSSL version: OpenSSL 1.0.1t  3 May 2016  
rezaalfajri@fajri-server:~/docker/lamp$ mkdir -p ~/docker/lamp/html  
rezaalfajri@fajri-server:~/docker/lamp$ cd ~/docker/lamp  
rezaalfajri@fajri-server:~/docker/lamp$ FROM php:7.4.3-apache  
FROM: command not found  
rezaalfajri@fajri-server:~/docker/lamp$ nano php.Dockerfile  
rezaalfajri@fajri-server:~/docker/lamp$ nano php.Dockerfile  
rezaalfajri@fajri-server:~/docker/lamp$ nano docker-compose.yaml  
rezaalfajri@fajri-server:~/docker/lamp$
```

Gambar 2.61 Tampilan arah direktori

- 15) Buat file php.Dockerfile menggunakan perintah

```
$ nano php.Dockerfile
```

Modul 2.14 Syntax pembuatan php.Dockerfile

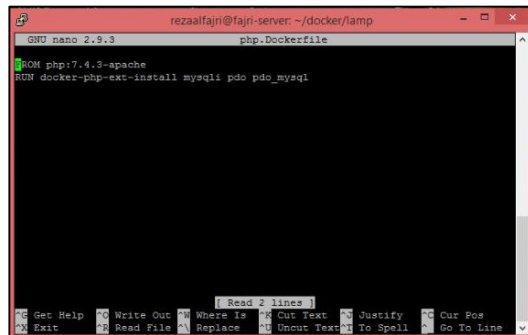


```
rezaalfajri@fajri-server: ~/docker/lamp
* Canonical livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch
9 packages can be updated.
0 updates are security updates.

Last login: Thu May 14 13:36:57 2020
rezaalfajri@fajri-server:~$ docker -compose version
Unable to resolve docker endpoint: context "compose" does not exist
rezaalfajri@fajri-server:~$ docker-compose version
docker-compose version 1.21.2, build a133471
docker-py version: 3.3.0
CPython version: 3.6.5
OpenSSL version: OpenSSL 1.0.1t  3 May 2016
rezaalfajri@fajri-server:~$ mkdir -p ~/docker/lamp/html
rezaalfajri@fajri-server:~$ cd ~/docker/lamp
rezaalfajri@fajri-server:~/docker/lamp$ FROM php:7.4.3-apache
FROM: command not found
rezaalfajri@fajri-server:~/docker/lamp$ nano php.Dockerfile
rezaalfajri@fajri-server:~/docker/lamp$ nano php.Dockerfile
rezaalfajri@fajri-server:~/docker/lamp$ nano docker-compose.yml
rezaalfajri@fajri-server:~/docker/lamp$
```

Gambar 2.62 Tampilan pembuatan dockerfile

16) Masukkan source code seperti pada gambar

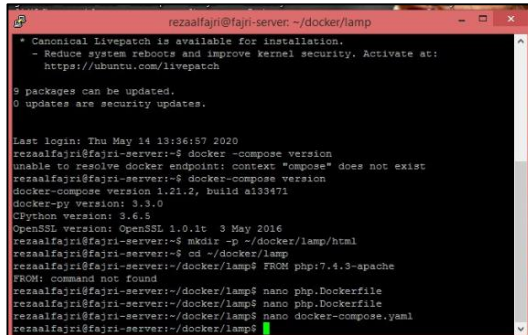


```
GNU nano 2.9.3      php.Dockerfile
FROM php:7.4.3-apache
RUN docker-php-ext-install mysqli pdo pdo_mysql

# Create default virtual host
RUN mkdir -p /etc/apache2/sites-available/
RUN echo <?php echo 'Hello World!';> > /etc/apache2/sites-available/default.conf
RUN a2enconf default
```

Gambar 2.63 Tampilan source code php.DockerFile

17) Buat file docker-compose.yml dengan perintah

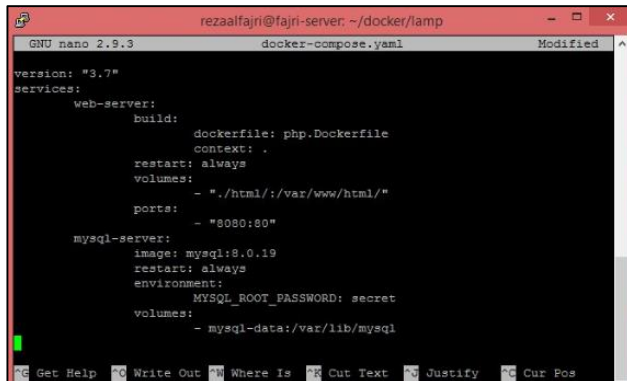


```
rezaalfajri@fajri-server: ~/docker/lamp
* Canonical livepatch is available for installation.
- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch
9 packages can be updated.
0 updates are security updates.

Last login: Thu May 14 13:36:57 2020
rezaalfajri@fajri-server:~$ docker -compose version
Unable to resolve docker endpoint: context "compose" does not exist
rezaalfajri@fajri-server:~$ docker-compose version
docker-compose version 1.21.2, build a133471
docker-py version: 3.3.0
CPython version: 3.6.5
OpenSSL version: OpenSSL 1.0.1t  3 May 2016
rezaalfajri@fajri-server:~$ mkdir -p ~/docker/lamp/html
rezaalfajri@fajri-server:~$ cd ~/docker/lamp
rezaalfajri@fajri-server:~/docker/lamp$ FROM php:7.4.3-apache
FROM: command not found
rezaalfajri@fajri-server:~/docker/lamp$ nano php.Dockerfile
rezaalfajri@fajri-server:~/docker/lamp$ nano php.Dockerfile
rezaalfajri@fajri-server:~/docker/lamp$ nano docker-compose.yml
rezaalfajri@fajri-server:~/docker/lamp$
```

Gambar 2.64 Tampilan membuat file docker-compose

18) Masukkan source code seperti pada gambar



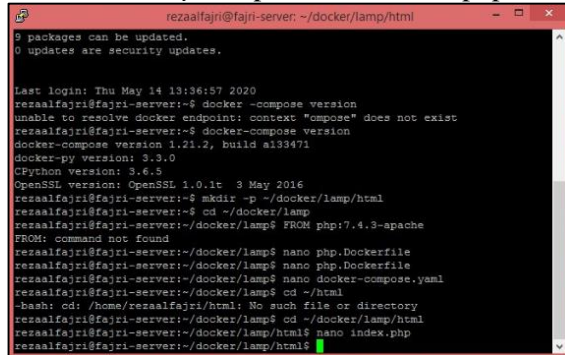
```
GNU nano 2.9.3      docker-compose.yml      Modified
version: "3.7"
services:
  web-server:
    build:
      dockerfile: php.Dockerfile
      context: .
    restart: always
    volumes:
      - ./html:/var/www/html/
    ports:
      - "8080:80"
  mysql-server:
    image: mysql:8.0.19
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: secret
    volumes:
      - mysql-data:/var/lib/mysql
```

Gambar 2.65 Tampilan source code docker-compose.yml

- 19) Buat file index.php dengan perintah

```
$ nano index.php
```

Modul 2.15 Syntax pembuatan index.php

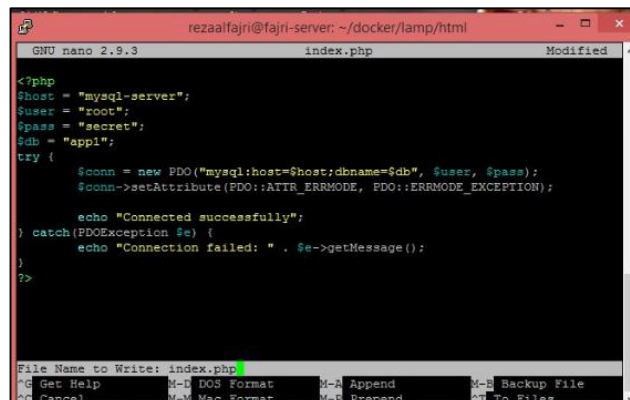
A terminal window titled 'rezaalfajri@fajri-server: ~/docker/lamp/html'. It shows a series of commands and their outputs. The user runs 'docker -compose version' which fails. Then 'docker-compose version' which shows version 1.21.2. Then 'FROM php:7.4.3-apache' which fails. Then 'nano php.Dockerfile' and 'nano docker-compose.yml'. Finally, 'cd /html' and 'nano index.php' are executed, with the cursor visible at the end of the command.

```
rezaalfajri@fajri-server: ~/docker/lamp/html
9 packages can be updated.
0 updates are security updates.

Last login: Thu May 14 13:36:57 2020
rezaalfajri@fajri-server:~$ docker -compose version
unable to resolve docker endpoint: context "compose" does not exist
rezaalfajri@fajri-server:~$ docker-compose version
docker-compose version 1.21.2, build a133471
docker-py version: 3.3.0
CPython version: 3.6.5
OpenSSL version: OpenSSL 1.0.1t  3 May 2016
rezaalfajri@fajri-server:~$ mkdir -p ~/docker/lamp/html
rezaalfajri@fajri-server:~$ cd ~/docker/lamp
rezaalfajri@fajri-server:~/docker/lamp$ FROM php:7.4.3-apache
FROM: command not found
rezaalfajri@fajri-server:~/docker/lamp$ nano php.Dockerfile
rezaalfajri@fajri-server:~/docker/lamp$ nano php.Dockerfile
rezaalfajri@fajri-server:~/docker/lamp$ nano docker-compose.yml
rezaalfajri@fajri-server:~/docker/lamp$ cd /html
-bash: cd: /home/rezaalfajri/html: No such file or directory
rezaalfajri@fajri-server:~/docker/lamp$ cd ~/docker/lamp/html
rezaalfajri@fajri-server:~/docker/lamp/html$ nano index.php
rezaalfajri@fajri-server:~/docker/lamp/html$
```

Gambar 2.64 Tampilan pembuatan index.php

- 20) Masukkan source code seperti pada gambar

A terminal window titled 'rezaalfajri@fajri-server: ~/docker/lamp/html' showing the source code of index.php in nano editor. The code is a PHP script that connects to a MySQL database. The terminal shows the file name 'index.php' and the command 'File Name to Write: index.php'.

```
rezaalfajri@fajri-server: ~/docker/lamp/html
GNU nano 2.9.3 index.php Modified
<?php
$host = "mysql-server";
$user = "root";
$pass = "secret";
$db = "app1";
try {
    $conn = new PDO("mysql:host=$host;dbname=$db", $user, $pass);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
} catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
?>

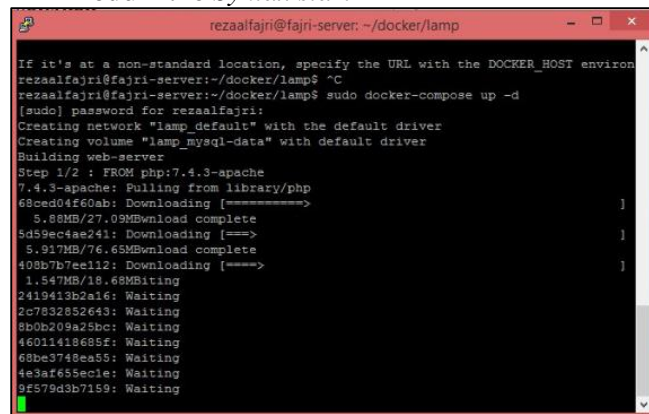
File Name to Write: index.php
^C Get Help      M-D DOS Format  M-A Append      M-S Backup File
^C Cancel       M-M Mac Format  M-P Prepend     ^T To Files
```

Gambar 2.65 Tampilan source code index.php

- 21) Start LAMP pada docker dengan perintah

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

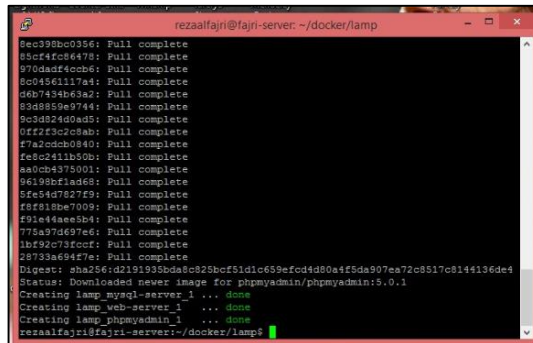
Modul 2.16 Syntax start LAMP

A terminal window titled 'rezaalfajri@fajri-server: ~/docker/lamp' showing the execution of 'sudo docker-compose up -d'. The output shows the creation of a network and volume, and the downloading of images for the LAMP stack.

```
rezaalfajri@fajri-server: ~/docker/lamp
If it's at a non-standard location, specify the URL with the DOCKER_HOST environ
rezaalfajri@fajri-server:~/docker/lamp$ ^C
rezaalfajri@fajri-server:~/docker/lamp$ sudo docker-compose up -d
[sudo] password for rezaalfajri:
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
7.4.3-apache: Pulling from library/php
68ced04f60ab: Downloading [=====>]
  5.88MB/27.09MBDownload complete
5d59ec4ae241: Downloading [====>]
  5.917MB/76.65MBDownload complete
408b7b7ee112: Downloading [=====>]
  1.547MB/18.68MBBiting
2419413b2a16: Waiting
2c7832852643: Waiting
8b0b209a25bc: Waiting
46011418e85f: Waiting
68be3748ea55: Waiting
4e3af655ec1e: Waiting
9f579d3b7159: Waiting
```

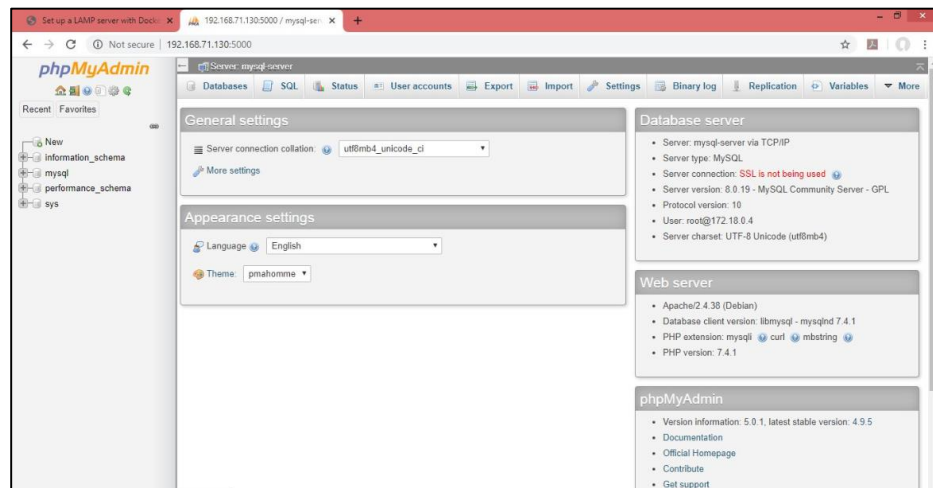
Gambar 2.66 Tampilan *start* LAMP

- 22) Jika tampilan sudah seperti pada gambar, maka LAMP pada docker sudah berhasil berjalan



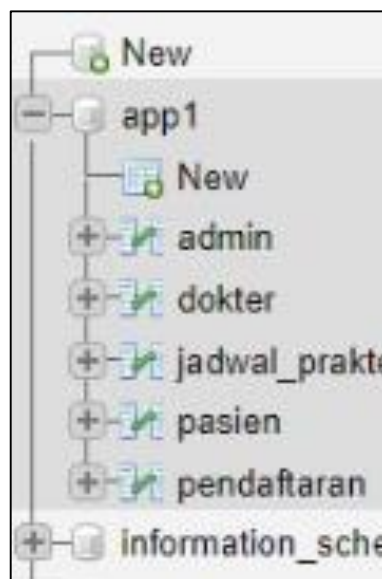
Gambar 2.67 Tampilan selesai LAMP

- 23) Buka web browser, masukkan IP:5000 untuk mengakses phpmyadmin



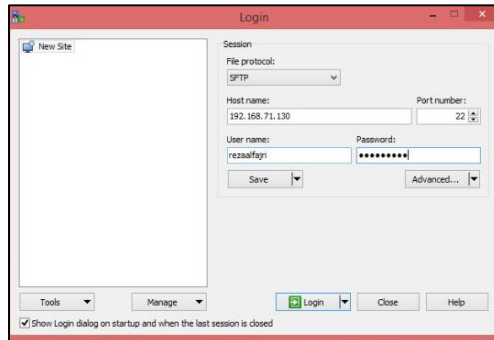
Gambar 2.68 Tampilan akses phpmyadmin

- 24) Buat database sesuai database yang ada di dalam proyek



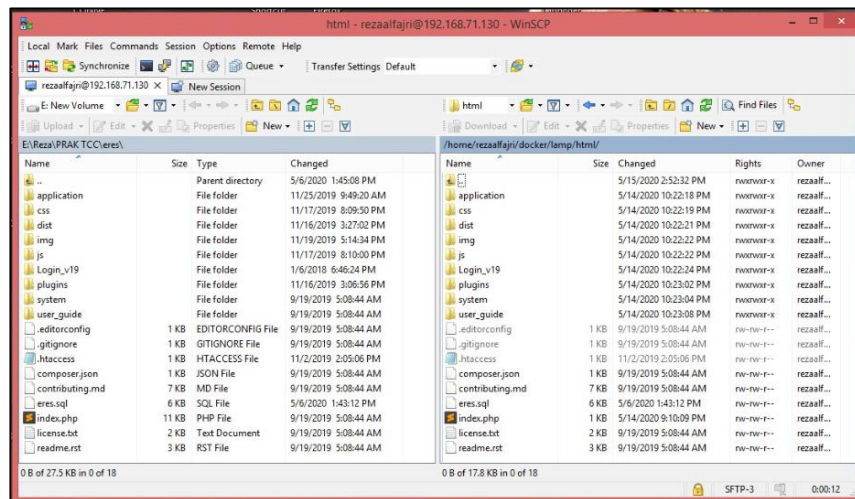
Gambar 2.69 Tampilan pembuatan database proyek

25) Login ke WinSCP



Gambar 2.70 Tampilan login WinSCP

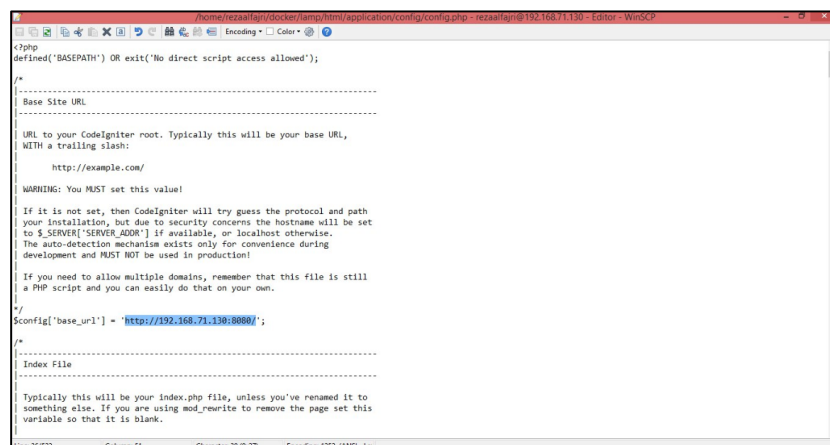
26) Upload berkas proyek ke direktori “docker/lamp/html/”



Gambar 2.71 Tampilan upload berkas ke direktori

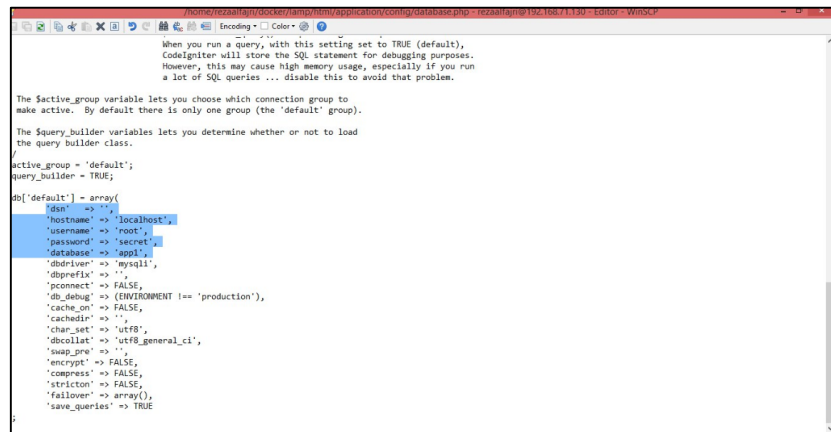
27) Buka file “config.php” pada proyek di dalam direktori

“docker/lamp/html/application/config/”, sesuaikan base-url seperti pada gambar



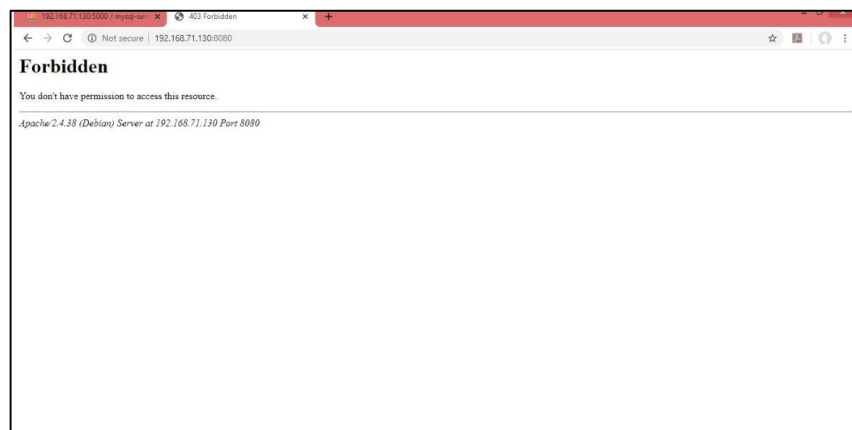
Gambar 2.72 Tampilan edit file config.php

- 28) Buka file “database.php” pada proyek dalam direktori
“docker/lamp/html/application/config/”, sesuaikan database seperti pada
gambar



Gambar 2.73 Tampilan edit database.php

- 29) Buka proyek web pada web browser dengan memasukkan “IP:8080”

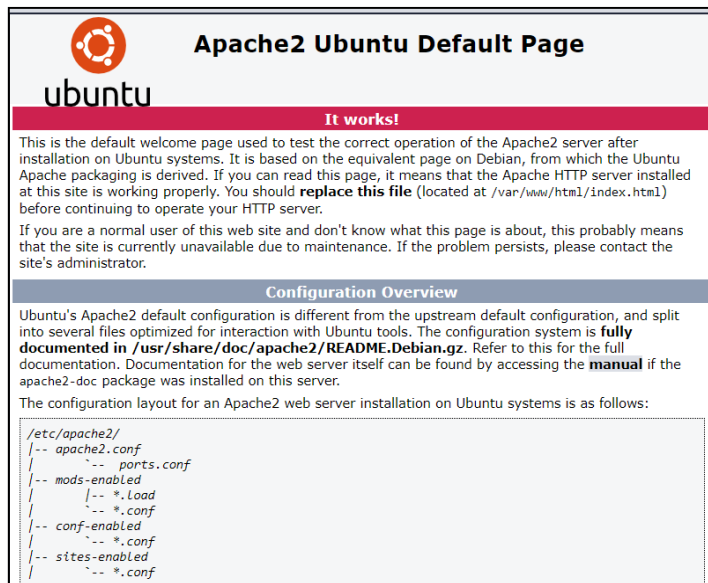


Gambar 2.74 Tampilan akses proyek

- 30) Proses upload proyek ke docker selesai


2.5 Hasil Implementasi

Penginstalan LAMPP yang meliputi Apache, MySQL, PHP, dan PHPMyAdmin. Saat Apache berhasil diinstal, perlu dilakukan pengecekan dengan mengakses alamat IP pada web browser. Jika tampilannya seperti gambar dibawah ini berarti Apache berhasil diinstal.



Gambar 2.75 Tampilan Apache berhasil diinstal

Informasi mengenai PHP yang diinstal dapat dilihat dengan cara mengakses IP/info.php pada web browser. Dengan itu maka akan ditampilkan informasi secara rinci seperti versi PHP yang digunakan.

PHP Version 7.2.24-0ubuntu0.18.04.4	
	
System	Linux anisa 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-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/15-xml.ini, /etc/php/7.2/apache2/conf.d/20-bz2.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-exif.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-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-intl.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.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-simplexml.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-sysshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini, /etc/php/7.2/apache2/conf.d/20-wddx.ini, /etc/php/7.2/apache2/conf.d/20-xmlreader.ini, /etc/php/7.2/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.2/apache2/conf.d/20-xsl.ini, /etc/php/7.2/apache2/conf.d/20-zip.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no

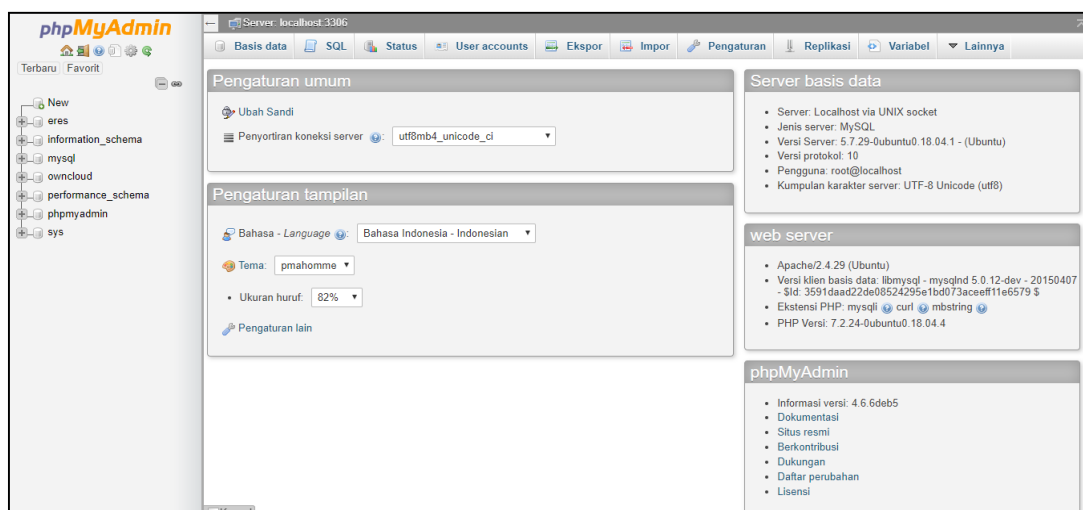
Gambar 2.76 Tampilan info php

Pada instalasi PHPMYAdmin diperlukan konfigurasi user root untuk dapat melakukan operasi login. Di bawah ini adalah contoh gambar error login gagal karena belum dilakukan konfigurasi pada root.



Gambar 2.77 Tampilan gagal login phpmyadmin

Konfigurasi dilakukan dengan mengetikkan sintak sql pada terminal PuTTY, setelah dilakukan konfigurasi coba login kembali dengan username root dan password. Gambar dibawah ini merupakan tampilan ketika berhasil login.



Gambar 2.78 Tampilan dashboard phpmyadmin setelah login

Jika berkas telah berhasil diupload ke server, maka sudah dapat diakses dengan IP/nama_folder_projek/index.php/ dan dapat berjalan sesuai fungsinya.

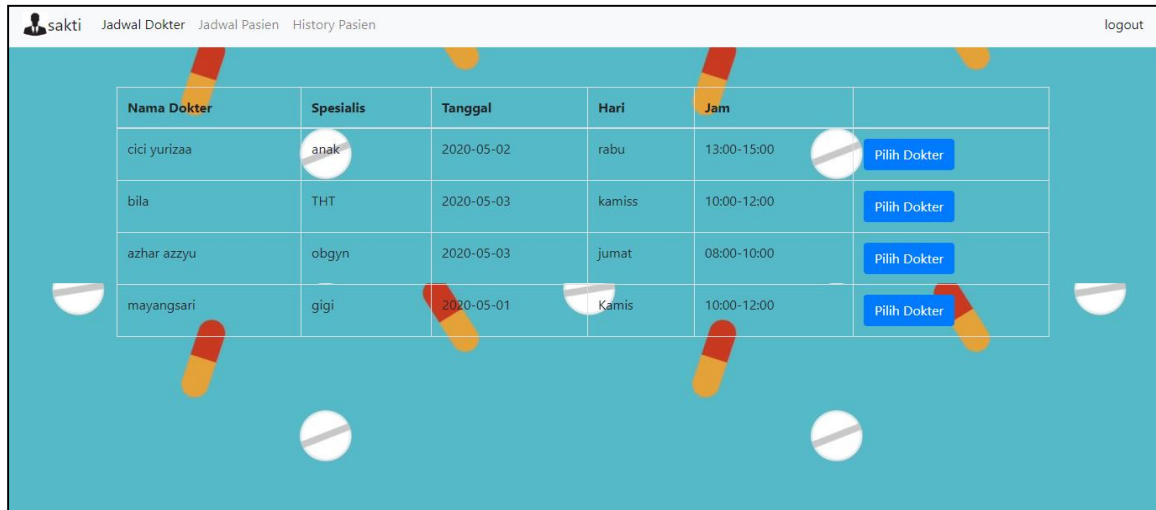
Gambar 2.79 Tampilan web login pasien

2.6 Pengujian Singkat

Pengujian dilakukan dalam proses pendaftaran pasien, data pasien diisi secara online oleh pasien yang dapat mengefesiensi waktu dan media kertas.

Gambar 2.80 Tampilan Registrasi Pasien

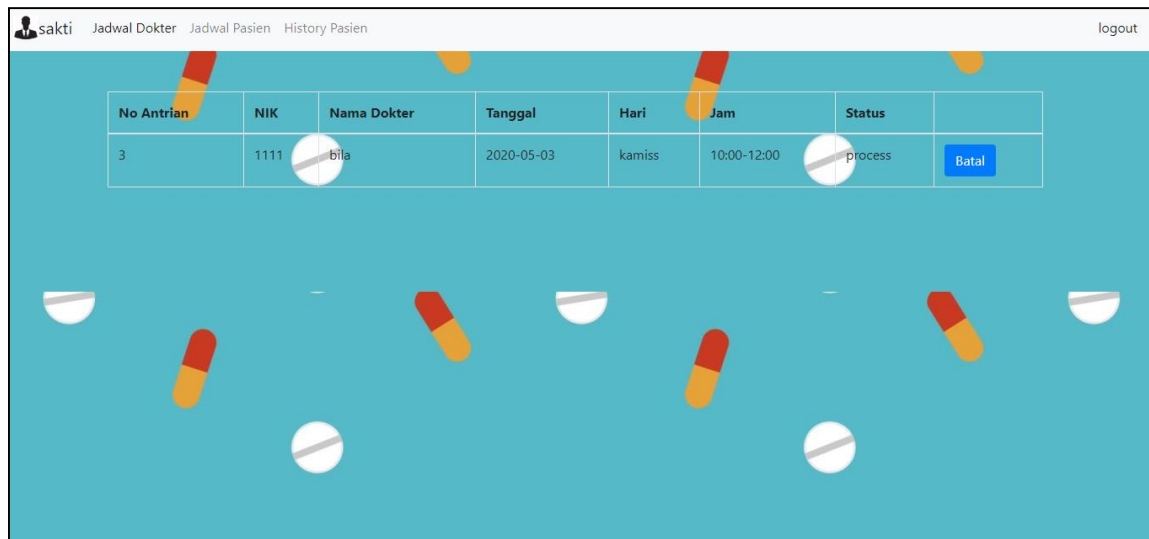
Setelah melakukan pendaftaran, pasien dapat melakukan login dengan NIK dan password yang telah didaftarkan. Pasien dapat memilih dan mendaftar antrian pada jadwal dokter yang dibutuhkan.



Nama Dokter	Spesialis	Tanggal	Hari	Jam	
cici yuriza	anak	2020-05-02	rabu	13:00-15:00	Pilih Dokter
bila	THT	2020-05-03	kamiss	10:00-12:00	Pilih Dokter
azhar azzyu	obgyn	2020-05-03	jumat	08:00-10:00	Pilih Dokter
mayangsari	gigi	2020-05-01	Kamis	10:00-12:00	Pilih Dokter

Gambar 2.81 Tampilan dashboard pasien

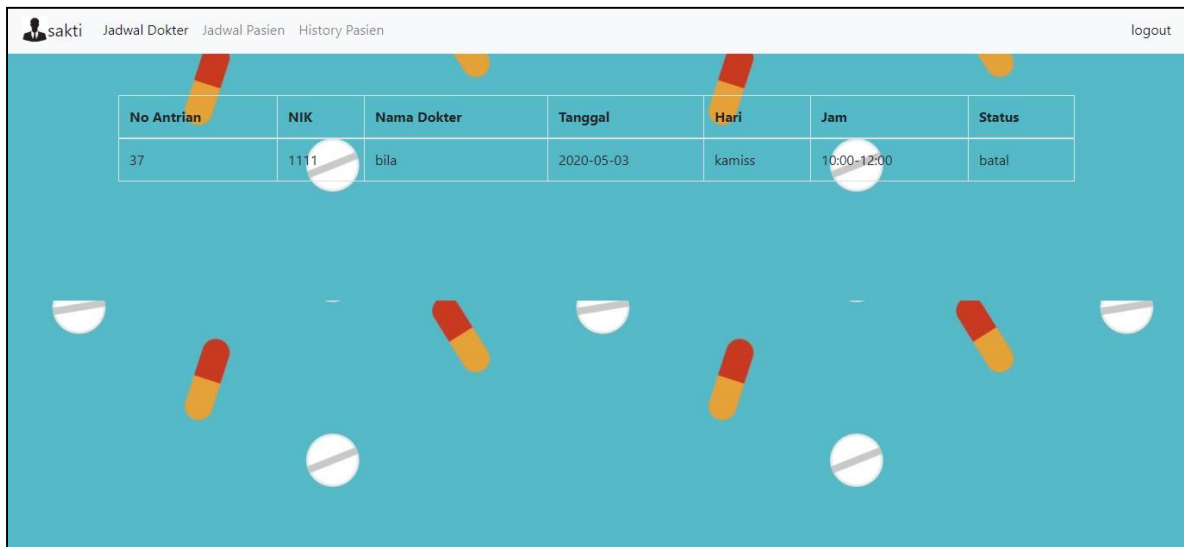
Setelah memilih jadwal dokter, pasien dapat melihat antrian dalam Jadwal Pasien.



No Antrian	NIK	Nama Dokter	Tanggal	Hari	Jam	Status	
3	1111	bila	2020-05-03	kamiss	10:00-12:00	process	Batal

Gambar 2.82 Tampilan Jadwal Pasien

Jadwal dokter yang sudah penuh, tidak dapat lagi diisi oleh pasien, dan akan ditampilkan keterangan bahwa Antrian Penuh. Jika pasien ingin membatalkan antrian, maka antrian pada Jadwal Pasien akan terhapus dan data History Pasien akan bertambah.

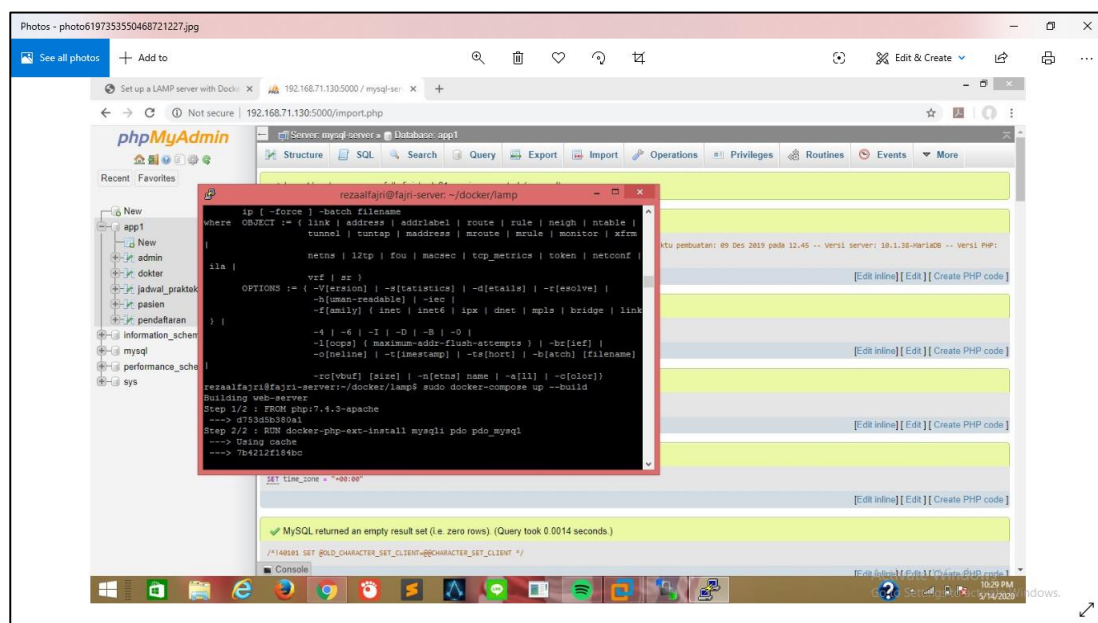


No Antrian	NIK	Nama Dokter	Tanggal	Hari	Jam	Status
37	1111	bila	2020-05-03	kamiss	10:00-12:00	batal

Gambar 2.83 Tampilan History Pasien

Dengan cara seperti ini, maka data riwayat berobat pasien pun dapat didokumentasikan dengan baik. Selain itu, pendaftaran dapat dilakukan kapan pun dan dimana pun, tidak perlu lagi menelepon ke bagian customer service untuk mendaftar antrian. Customer service tidak tersedia 24 jam, jadi pendaftaran hanya dapat dilakukan pada jam kerja. Dengan adanya sistem ini, pendaftaran dapat dilakukan 24 jam, dan memotong waktu antrian pendaftaran untuk pasien baru dalam pengisian berkas yang dilanjutkan antrian jadwal dokter.

Pada tugas dua, ketika mengakses proyek yang telah diupload masih terdapat error yang tidak diketahui, tetapi telah berhasil mengakses phpmyadmin dan mengimpor database proyek.



Gambar 2.84 Sukses impor database

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:

No.	Jenis Tugas	Waktu Pengerjaan							
		Maret	April				Mei		
		4	1	2	3	4	1	2	3
1.	Analisa Persoalan dan Penentuan Judul								
2.	Pembagian Tugas								
3.	Pengerjaan instal Ubuntu								
4.	Pengerjaan LAMPP								
5.	Pengerjaan upload projek								
6.	Pengerjaan laporan tahap 1								
7.	Pengerjaan revisi laporan								
6.	Pengerjaan instal docker								
7.	Pengerjaan instal dockerfile								
8.	Pengerjaan instal docker compose								
9.	Pengerjaan instal LAMP								
10.	Pengerjaan upload projek								
11.	Pengerjaan laporan tahap akhir								

Tabel 3.1 Agenda Pengerjaan Proyek

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	Anisa
2.	Pengujian Singkat	Fajri
3.	Pengerjaan Laporan	Anisa, Fajri
4.	Pengerjaan Ubuntu, LAMPP, dan upload projek	Anisa
5.	Pengerjaan Dockerfile, LAMP, dan upload projek	Fajri

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Sistem pendaftaran antrian pasien pada rumah sakit masih sangat terbatas oleh waktu, karena rumah sakit hanya menerima pendaftaran pada jam kerja. Selebihnya pasien tidak bisa mendaftar, hal ini merupakan sebuah masalah yang dapat diselesaikan dengan cloud computing. Dimulai dari mengintegrasikan sistem dalam cloud computing agar sistem dapat diakses dengan mudah, kapan saja, dan dimana saja dengan internet. Sistem dan data akan jauh lebih mudah dikelola dan dimaintenance menyesuaikan kebutuhan. Projek sistem berdiri dalam sistem operasi Ubuntu dengan LAMPP yang diinstal secara manual secara bertahap, agar projek web ini dapat diupload ke server dan terhubung dengan database PHPMyAdmin. Hasil akhirnya sistem dapat berjalan sesuai fungsi dan tujuannya

Pada bagian tugas dua pengimplementasian docker dilakukan karena docker dapat mengoptimalkan infrastruktur IT rumah sakit, dan dapat meningkatkan kinerja pada programmer serta sistem administrator. Dengan sistem kontainer, seluruh pekerjaan pembuatan aplikasi, modul, dan monitoring, serta manajemen infrastruktur dapat lebih cepat, efisien, dan lebih aman. Hasil akhir pada implementasinya masih terdapat kesalahan pada peng-upload-an projek yang menyebabkan sistem belum dapat diakses.

4.2 Saran

Diperlukan kajian literatur dan pengecekan yang lebih untuk mengetahui penyebab pada tugas dua sistem belum dapat diakses, sehingga nantinya penggunaan docker ini benar-benar dapat digunakan dan mempermudah dalam manajemen sistemnya.

DAFTAR PUSTAKA

- Hechavarría, Rodney; López, G. (2013). 濟無No Title No Title. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699.
<https://doi.org/10.1017/CBO9781107415324.004>
- Kusuma, T. P., Munadi, R., & Sanjoyo, D. D. (2017). Implementasi dan Analisis Computer Clustering System dengan Menggunakan Virtualisasi Docker. *E-Proceeding of Engineering*, 4(3), 1–6.
<https://libraryeproceeding.telkomuniversity.ac.id/index.php/engineering/article/download/4963/4930>
- ROMADLON BIK, M. (2017). IMPLEMENTASI DOCKER UNTUK PENGELOLAAN BANYAK APLIKASI WEB (Studi Kasus : Jurusan Teknik Informatika UNESA). *Jurnal Manajemen Informatika*, 7(2), 46–50.
- Syaikhu, A. (2013). 1927-4108-1-Pb. *Jurnal Pustakawan Indonesia Volume 10 No. 1*, 10(1), 1–12.
- Agatha, N. (2020). Cara Install Docker di Ubuntu 18.04 dan Cara Menggunakannya. Hostinger Tutorial. Retrieved 14 May 2020, from <https://www.hostinger.co.id/tutorial/cara-install-docker-ubuntu/>.

LAMPIRAN

