

PRAKTIKUM TEKNOLOGI CLOUD COMPUTING
LAPORAN PROYEK AKHIR

SISTEM INFORMASI DINAS KEARSIPAN DAN PERPUSTAKAAN
MENGUNAKAN LAMPP DAN PROSES PEMBUATAN DOCKERFILENYA



DISUSUN OLEH:

NAMA ANGGOTA : MOH EKA SAPUTRA KIAY DEMAK 123140052
FIKI SYIHAB IRAWAN 123170050

KELAS : B

ASISTEN PRAKTIKUM : JALUANDA PARAMA, S.Kom.
MUHAMMAD IMAM ALFATAH

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

HALAMAN PENGESAHAN

SISTEM INFORMASI DINAS KEARSIPAN DAN PERPUSTAKAAN MENGUNAKAN LAMPP DAN PROSES PEMBUATAN DOCKERFILENYA

Disusun oleh :

Moh Eka Saputra Kiay Demak

123140052

Fiki Syihab Irawan

123170050

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

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

Muhammad Imam Alfatah

NIM. 123160119

Mengetahui,

Ka. Lab. Digital

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

NIK. 2 8201 13 0425 1

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan praktikum Teknologi Cloud Computing serta laporan proyek akhir praktikum yang berjudul Sistem Informasi Dinas Kearsipan Dan Perpustakaan Menggunakan LAMPP Dan Proses Pembuatan Dockerfilenya. Adapun laporan ini berisi tentang proyek akhir yang kami pilih dari hasil pembelajaran selama praktikum berlangsung.

Tidak lupa ucapan terimakasih kepada asisten dosen yang selalu membimbing dan mengajari kami dalam melaksanakan praktikum dan dalam menyusun laporan ini. Dalam penyusunan laporan kami menyadari bahwa ini masih sangat jauh dari kesempurnaan dikarenakan keterbatasan pengalaman dan pengetahuan, oleh karena itu kritik serta saran yang membangun kami harapkan untuk menyempurnakan laporan akhir ini.

Semoga laporan ini dapat menambah wawasan para pembaca dan bisa bermanfaat untuk perkembangan ilmu dan pengetahuan. Atas perhatian dari semua pihak yang membantu penulisan laporan ini baik secara langsung maupun tidak langsung, kami ucapkan terimakasih. Semoga laporan ini dapat dipergunakan dengan sebaik baiknya dan bermanfaat bagi semua pihak.

Yogyakarta, 16 Mei 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.....	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>	7
2.3 Parameter dan Konfigurasi.....	9
2.4 Tahap Implementasi.....	13
2.5 Hasil Implementasi	19
2.6 Pengujian Singkat	21
BAB III JADWAL Pengerjaan dan Pembagian Tugas.....	29
3.1 Agenda Pengerjaan	29
3.2 Keterangan Pembagian Tugas	29
BAB IV KESIMPULAN DAN SARAN.....	30
4.1 Kesimpulan.....	30
4.2 Saran	30
DAFTAR PUSTAKA.....	31
LAMPIRAN	32

BAB I

PENDAHULUAN

1.1. Latar Belakang Proyek Akhir

Sejarah *Cloud Computing* tercatat mulai pada tahun 1960-an. Dimana pada saat itu seorang ahli komputer dari MIT memberikan gagasan bahwa kelak pada suatu hari akan lahir infrastruktur publik untuk sebuah komputasi layaknya layanan listrik dan telepon (Awan, 2011). Lalu pada tahun 1990-an ide itu mulai direalisasikan dengan hadirnya konsep ASP atau *Application Service Provider* yang memungkinkan pengelolaan data secara terpusat oleh suatu perusahaan. Ide-ide tersebut terus berkembang hingga pada tahun 1995 lewat Larry Ellison sang pendiri Oracle menggagas ide yang diberi nama *Network Computing* (Arutyunov, 2012). Larry Ellison mengatakan bahwa aplikasi seharusnya tidak dipasang dalam sebuah komputer personal yang berefek b

uruk pada beban berat yang ditanggung pc tersebut, sudah seharusnya konsep tersebut diganti dengan penggunaan sebuah terminal utama berupa komputer *server*. *Cloud computing* merupakan akses layanan *on-demand* ke sekumpulan sumber daya komputasi seperti jaringan, server, penyimpanan, aplikasi dan layanan (Ernawati, 2013). *Cloud computing* terdiri dari tiga jenis, tingkatan pada layanan berbasis *cloud*, yaitu SaaS (*Software as a Service*), IaaS (*Infrastructure as a Service*) dan PaaS (*Platform as a Service*). Terdapat satu lagi layanan yaitu XaaS. XaaS (*Anything as a Service*) merujuk pada semakin beragamnya layanan yang tersedia melalui internet melalui *cloud computing* yang bertentangan dengan yang disediakan secara lokal, atau di tempat. Sistem *Cloud* bekerja menggunakan internet sebagai *server* dalam mengolah data. Sistem *cloud computing* ini memungkinkan pengguna untuk *login* ke internet yang tersambung ke program untuk menjalankan aplikasi yang dibutuhkan tanpa melakukan instalasi (Dillon et al., 2010). Sehingga aplikasi dapat diakses dimanapun.

Salah satu produk *cloud computing* adalah Open Library. Open Library adalah katalog. Proyek ini dimulai pada November 2007 dan telah membuat katalog catatan dari beberapa perpustakaan terbesar di dunia sejak itu (openlibrary.org). Katalog memiliki lebih dari 20 juta catatan edisi *online*, menyediakan akses ke 1,7 juta versi buku yang dipindai, dan tautan ke sumber-sumber eksternal seperti WorldCat dan Amazon jika memungkinkan. Sasaran kedua adalah membuat pengguna sedekat mungkin dengan dokumen aktual yang Anda cari sebaik baiknya, apakah itu versi pindaian dari *Internet Archive*, atau tautan ke Powell's di mana Anda dapat membeli salinan Anda sendiri (McGibbon et al., 2015). Open

Library memungkinkan pengguna mengakses tanpa perlu koneksi internet. Buku-bukunya tersedia dalam format PDF, ePub, MOBI, hingga Plain text yang memudahkan pengguna untuk membaca.

Alasan untuk membuat sistem informasi dinas kearsipan dan perpustakaan kebumen adalah proses-proses pengolahan data yang dilakukan di perpustakaan pada umumnya masih bersifat manual, seperti pencatatan pada laporan buku, peminjaman, pengembalian serta pengunjung yang setiap hari yang datang, yang mengakibatkannya kesulitan sering timbul dan dirasakan oleh perpustakaan dalam mengolah data karena jumlah data yang harus diolah cukup banyak serta proses pengarsipan yang membutuhkan ruang untuk penyimpanan. Karena begitu banyaknya data yang masuk maka data yang akan dicari begitu lama untuk mendapatkan data. Pada saat ini petugas dinas kearsipan dan perpustakaan yang menangani pengolahan perpustakaan masih banyak kekurangan yang harus diperbaiki lagi. Sebelumnya pengarsipan dilakukan secara konvensional yang mengharuskan kontak fisik secara langsung. Penggunaan internet untuk pengarsipan perpustakaan sangat memudahkan pengguna dalam mengatur buku. Dikarenakan proses penggunaan yang berubah dari konvensional menjadi menggunakan internet, maka diperlukan sosialisasi cara penggunaan sistem informasi ini kepada pengguna dan pengelola terkait. Sistem informasi akan di *hosting* ke internet sehingga dapat diakses dimana pun oleh pengguna.

Dalam pengerjaan *project* akan menggunakan Ubuntu LAMPP dan Docker. Untuk itu langkah yang akan diambil adalah pertama, mengidentifikasi kebutuhan *hardware* yang diperlukan agar dapat menjalankan Sistem Informasi Perpustakaan ini dengan baik. Selanjutnya, mengumpulkan bahan dan perangkat lunak yang dibutuhkan agar aplikasi dapat terbangun dengan baik. Dikarenakan sistem operasi menggunakan Ubuntu maka perlu untuk membuat *Virtual Machine* untuk menjalankan *operating sistem* tersebut didalam *operating sistem* Windows. Setelah Ubuntu berhasil terinstal maka selanjutnya adalah penginstalan dan konfigurasi LAMPP (Linux, Apache, Mysql, PhpMyadmin, PHP). Dan proses terakhir adalah menghosting web Sistem Informasi Perpustakaan menggunakan prinsip FTP ke dalam ubuntu lalu pembuatan Dockerfile sehingga aplikasi dapat berjalan di dalam Docker *environment*.

1.2 Tujuan Proyek Akhir

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

2. Mengimplementasikan arsitektur *cloud computing* untuk Sistem Informasi Perpustakaan yang telah dibuat sebelumnya dengan menggunakan Ubuntu 18.04 LTS dan LAMPP (Apache 8, PHP 7.1, MySQL 5.2).
3. Mengintegrasikan Sistem Informasi Perpustakaan yang berada di Ubuntu *Server* (LAMPP).
4. Mengintegrasikan web Sistem Informasi Dinas Pengarsipan dan Perpustakaan dengan Docker untuk mengepak/memasukkan aplikasi secara lengkap beserta semua hal lainnya yang dibutuhkan sehingga dapat dideploy dan berfungsi secara baik.
5. Megimplementasikan arsitektur Docker untuk Sistem Informasi Dinas Pengarsipan dan Perpustakaan menggunakan layanan docker yang berisi *container* LAMPP (Apache 8, PHP 7.1, MySQL 5.2) untuk menjalankan web secara baik.

1.3 Manfaat Proyek Akhir

Manfaat yang dapat diperoleh dari pembuatan proyek Sistem Informasi Dinas Pengarsipan dan Perpustakaan dengan LAMPP dan Dockerfilenya adalah sebagai berikut:

1. Sistem Informasi Perpustakaan yang telah dirancang dapat digunakan sewaktu-waktu tanpa khawatir *downtime* dikarenakan terdapatnya *primary* dan *backup server* yang menggunakan arsitektur *recovery* pada *cloud computing*.
2. Perusahaan tidak perlu memperlakukan *maintenance*, dikarenakan dengan menggunakan *cloud computing*, rutinitas *maintenance* akan dilakukan sepenuhnya oleh *vendor*.
3. Tidak perlu mengkhawatirkan batasan memori penyimpanan data perpustakaan dikarenakan sistem telah sepenuhnya beralih menggunakan *cloud computing* sehingga penyimpanan menjadi tidak terbatas.
4. Permintaan data buku dalam perpustakaan dan manajemen data buku secara *realtime* dapat dilakukan secara terpusat maupun secara terdistribusi dikarenakan setiap sistem terhubung satu sama lain melalui *private cloud network*.
5. *Deployment* aplikasi Sistem Informasi Dinas Pengarsipan dan Perpustakaan menjadi lebih mudah dan dapat di luncurkan pada berbagai sistem operasi yang digunakan.
6. Manajemen data aplikasi menggunakan Docker menjadi lebih efisien dilakukan dikarenakan menggunakan sistem *container* yang bersifat isolasi.
7. Pengembangan aplikasi Sistem Informasi Dinas Pengarsipan dan Perpustakaan menggunakan Docker yang dapat mendukung kemampuan konfigurasi, memisahkan kebutuhan infrastruktur dari lingkungan aplikasi.

1.4 Tahap Penyelesaian Proyek Akhir

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

1. Menganalisis kebutuhan dari Sistem Informasi Dinas Pengarsipan dan Perpustakaan untuk ditransformasikan ke dalam arsitektur *cloud computing* menggunakan basis IaaS/SaaS/PaaS/DBaaS dan XaaS/WaaS.
2. Mengintegrasikan aplikasi Sistem Informasi Dinas Pengarsipan dan Perpustakaan dengan Docker
3. Menentukan konfigurasi yang tepat untuk pengaturan Docker sehingga dapat digunakan sesuai *requirement* yang berupa:
 - a. Dapat diakses oleh jaringan/IP tertentu saja.
 - b. Dapat menjalankan container AMPP (Apache2, Mysql, PHP, PHPMyAdmin)
 - c. Menggunakan Ubuntu dengan versi 18.04.4 LTS.
4. Merancang topologi *cloud computing* untuk mengintegrasikan dua sub sistem yang berbeda sehingga dapat digunakan secara terintegrasi.
5. Menguji keandalan arsitektur *cloud computing* yang dibangun dengan beberapa pengujian yaitu kecepatan waktu akses, batasan akses sesuai konfigurasi, dan Area akses.
6. Menghasilkan Sistem Informasi Dinas Pengarsipan dan Perpustakaan yang berbasis *cloud computing* sesuai standar ISO 9001.

BAB II

ISI DAN PEMBAHASAN

2.1 Komponen yang Digunakan

Untuk membangun “Sistem Informasi Dinas Kearsipan Dan Perpustakaan Menggunakan LAMPP dan Dockerfilenya” yang berbasis konsep *cloud computing*, maka diperlukan analisis berbagai komponen. Berikut akan dijelaskan terlebih dahulu dalam bentuk poin-poin singkat:

1. Sistem yang telah dibangun menggunakan bahasa pemrograman PHP dengan versi 7.3 dan bahasa HTML dengan versi minimal 4.0.
2. Selain itu juga diperlukan penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 5.2 sehingga dapat digunakan untuk menyimpan berbagai data perpustakaan yang dibutuhkan oleh sistem tersebut.
3. Untuk target pengguna dengan konsep *cloud computing*, maka penggunaanya ialah seluruh pengunjung dan peminjam buku perpustakaan yang ada di Dinas. Tidak ada yang dapat mengakses sistem tersebut kecuali harus terhubung melalui jaringan intranet Dinas. Sehingga diperlukan arsitektur *cloud computing* yang bersifat *private*, tidak dapat diakses secara bebas oleh semua orang kecuali orang yang berkepentingan.
4. Dalam pembangunan Sistem Informasi Perpustakaan menggunakan IDE Netbeans untuk merancang dan membangun aplikasi berbasis PHP dan HTML
5. Sistem operasi yang digunakan dalam *virtual* adalah Ubuntu 18.04 LTS
6. Penggunaan Ubuntu *server* secara *remote* menggunakan Putty Versi 0.73 melalui *terminal* Linux yang menggunakan IP Ubuntu *server*.
7. Dalam pemindahan data Web Sistem Informasi Dinas Pengarsipan dan Perpustakaan dari sistem operasi *host* (Windows) ke dalam Ubuntu *server* menggunakan WinSCP Versi 15.17.3 dengan prinsip FTP (*File Transfer Protokol*)
8. Dalam mengepak aplikasi Sistem Informasi Dinas Pengarsipan dan Perpustakaan menggunakan Docker untuk membangun lingkungan aplikasi yang dapat dijalankan diberbagai sistem operasi
9. Untuk menjalankan Sistem Informasi Dinas Pengarsipan dan Perpustakaan dalam lingkungan Docker, dibutuhkan Docker Image Apache, Mysql, PHP, PHPMyAdmin sebagai pendukung

10. Untuk menjalankan Docker Image secara bersamaan digunakan Docker Compose yang Terdiri dari Web Server, Apache, Mysql, Phpmyadmin.

11. Spesifikasi laptop yang digunakan untuk menjalankan project ini adalah sebagai berikut : Dell G3 15, Intel Core i7 8750H, 8GB RAM, Nvidia Geforce 1050TI, LAN, WLAN.

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

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 @2.2Ghz	Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> .
3.	Konfigurasi Jaringan <i>Guest OS</i>	Mode Bridge	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP: 192.168.43.192/30	Alamat IP dan <i>network</i> yang digunakan oleh <i>guest OS</i> .
		DNS: 192.168.43.1	Alamat IP untuk DNS <i>guest OS</i> .
		GW: 192.168.43.1	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.	RAM	4GB	Alokasi RAM untuk <i>guest OS</i>
6.	<i>Operating System</i>	Ubuntu 64bit	Sistem operasi yang digunakan <i>guest OS</i>
7.	Nama Virtual Machine	VMKelompok9	Nama yang digunakan untuk <i>Virtual Machine</i>

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

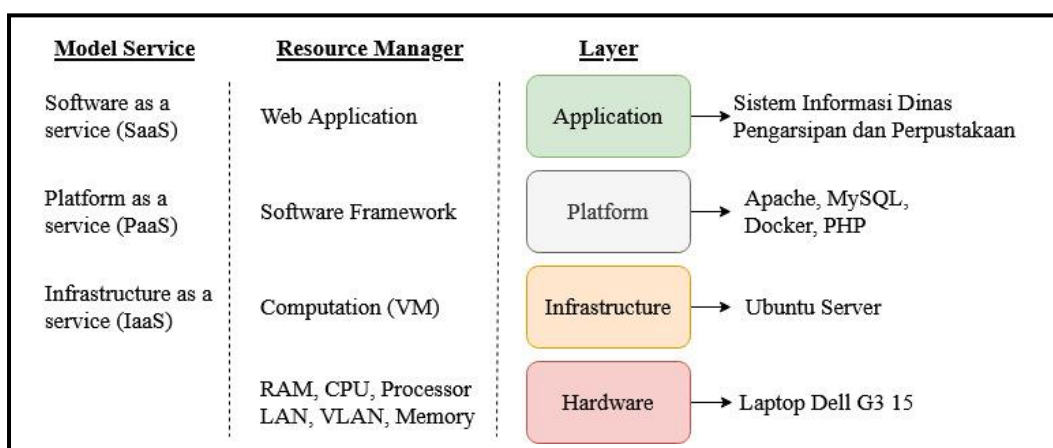
Tabel 2.2 Spesifikasi Ubuntu OS untuk proyek pertama

No.	Nama Parameter	Nilai	Keterangan
1.	LAMPP	Apache 2.4	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.3	dsb
		MySQL	dsb
		PHPMyAdmin	dsb
2.	PHP + HTML	Version 7	Bahasa pemrograman yang digunakan untuk membangun Sistem Informasi Perpustakaan
3.	MySQL	Version 5.2	DBMS yang digunakan Sistem Informasi Perpustakaan untuk akses <i>database</i>
4.	Netbeans	Netbeans 8.2	IDE yang digunakan untuk membangun Sistem Informasi
5.	PuTTY	Putty Versi 0.73	Aplikasi yang digunakan untuk menggunakan Ubuntu secara <i>remote</i>

6.	WinSCP	WinSCP Versi 15.17.3	Aplikasi yang digunakan untuk transfer file dari Windows ke Ubuntu <i>server</i>
7.	SSH	OpenSSH	Digunakan untuk dapat melakukan <i>remote server</i>
8.	Docker Engine	Docker Community Edition	aplikasi untuk membuat Docker <i>environment</i>
9.	Docker Compose	Docker Compose 1.17	Berguna untuk menjalankan <i>service</i> secara <i>multi-container</i> dalam waktu yang bersamaan

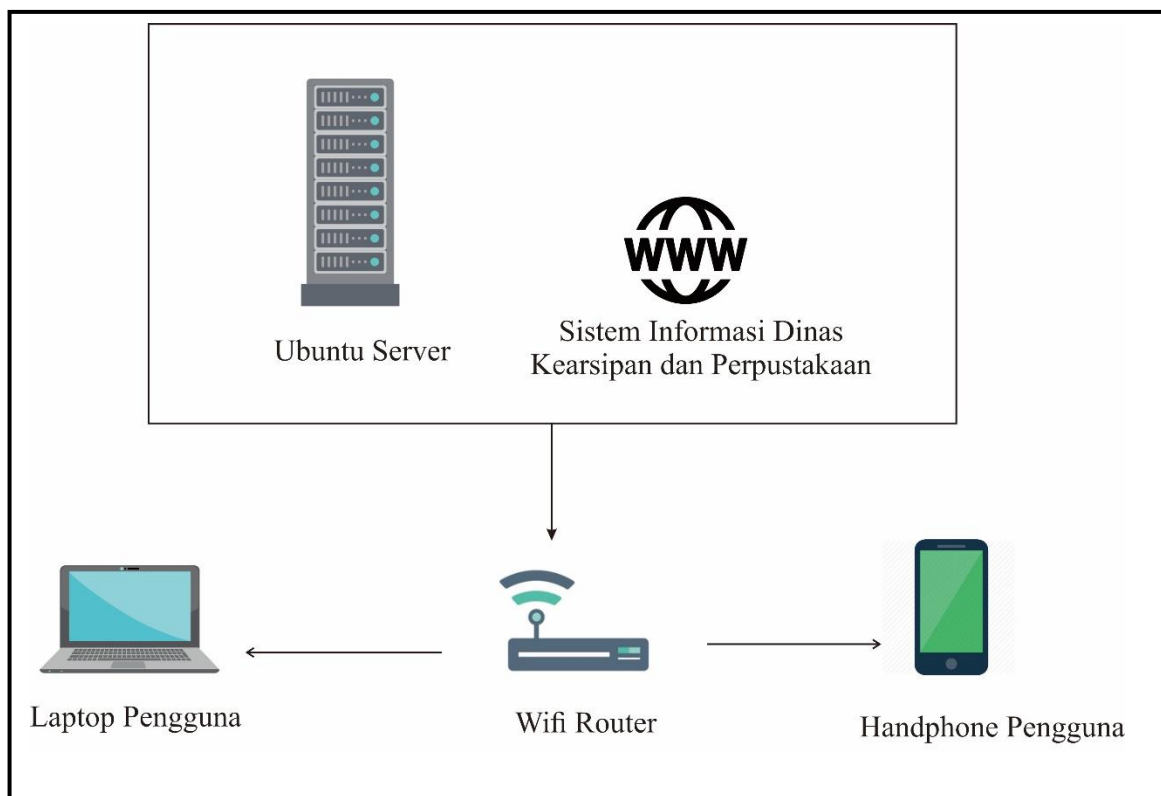
2.2 Rancangan Arsitektur *Cloud Computing*

Pada project akhir ini menggunakan rancangan Saas (*Software As a Service*) dengan laptop Dell G3 15 sebagai hardware dasar untuk menjalankan sistem. Sistem operasi yang digunakan adalah Ubuntu 18.04 Live Server yang berjalan secara virtual. Dalam melakukan virtualisasi tersebut menggunakan VMWare Workstation sebagai *Hypervisor* yang menjalankan Ubuntu *Server*. Di dalam sistem operasi Ubuntu tersebut dipasang beberapa aplikasi pendukung yaitu Apache, PHP, MySQL, Docker. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Gambar 2.1** berikut ini:



Gambar 2.1 Komponen *layer* terhadap komponen penyusunnya

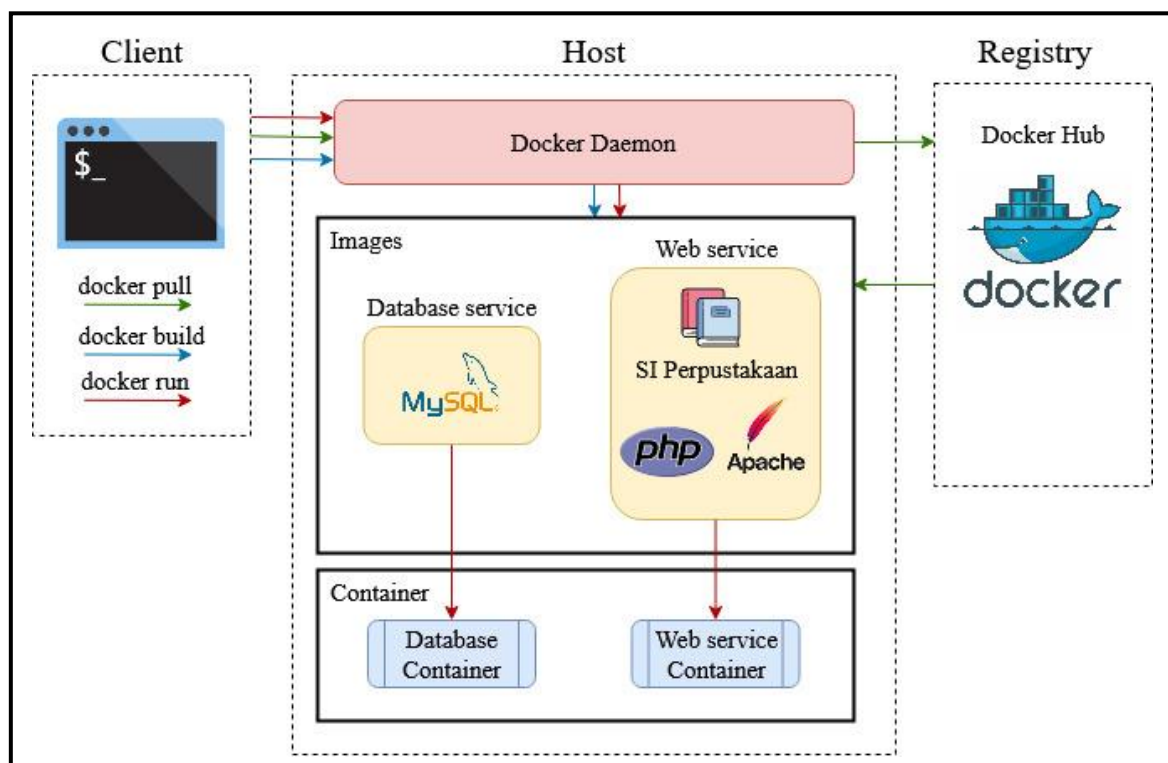
Website Sistem Informasi Dinas Kearsipan dan Perpustakaan yang telah diupload ke Ubuntu *server* dapat digunakan oleh pengguna yang berada dalam satu lingkup jaringan. Pengguna dapat mengakses *website* tersebut dengan menggunakan laptop maupun *handphone*. Pemancaran Ubuntu *server* dapat menggunakan *Router* Wifi apapun atau menggunakan fasilitas bawaan *handphone* yaitu tethering jaringan. Pengguna yang akan mengakses *Website* harus terlebih dahulu tersambung dengan jaringan yang sama yang dipancarkan oleh *Router* Wifi ataupun *Tethering Handphone*. Ilustrasi mengenai interaksi jaringan dengan *device* pengguna dapat dilihat pada **Gambar 2.2** berikut ini:



Gambar 2.2 Ilustrasi jaringan

Arsitektur Docker dibagi menjadi 3 layer yaitu, *Client*, *Docker Host*, dan *Registry*. Pada layer *Docker Host* terdapat *container* dan *images*. *Container* berfungsi untuk menjalankan *Images* yang ada. *Registry* merupakan *repository online* salah satunya Docker Hub. Di dalam *registry* pengguna dalam mengunduh *Images* yang dibutuhkan seperti Apache, Mysql, PHP, dan sebagainya. Susun Docker. Compose adalah alat untuk mendefinisikan dan menjalankan aplikasi Docker multi-kontainer. Dengan Compose, Anda menggunakan file *Compose* untuk mengonfigurasi layanan aplikasi. Kemudian, menggunakan satu perintah, Anda membuat dan memulai semua layanan dari konfigurasi. Docker-compose berisi Docker daemon adalah sebuah *service* yang dijalankan di dalam *host* dalam *Operating System* (OS). Fungsinya adalah membangun, mendistribusikan, dan menjalankan *container* docker. Pengguna tidak dapat langsung menggunakan Docker Daemon, akan tetapi untuk menggunakan Docker Daemon maka pengguna menggunakan Docker Client sebagai perantara atau CLI. Di dalam *layer client* pengguna dapat melakukan perintah yaitu *docker pull*, *docker build*, dan *docker run*. *Docker pull* berguna untuk mengambil *images* dari Docker Hub, *docker build* berguna untuk membangun *images* untuk dijalankan di Docker Container, dan *docker run* adalah perintah untuk menjalankan Docker

Images yang sudah ada ke dalam *container* Ilustrasi arsitektur Docker dapat dilihat pada **Gambar 2.2** Berikut:



Gambar 2.2 Arsitektur Docker

2.3 Parameter dan Konfigurasi

Langkah pertama dalam pembuatan proyek akhir ini adalah pemasangan Apache sebagai *web service*. Parameter yang digunakan untuk instalasi Apache dan Pengaturan Firewall dapat dilihat pada penjelasan **Modul 2.1** berikut ini:

```
$ sudo apt update
Keterangan:
- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- update : perintah untuk mngupdate paket

$ sudo apt install apache2
Keterangan:
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- apache2 : nama paket aplikasi untuk Apache

$ sudo ufw allow in "Apache Full"
Keterangan:
- ufw : merupakan singkatan dari Uncomplicated Firewall
- allow in : perintah untuk perizinan Firewall
- Apache Full : merupakan application yang tersedia dalam Uncomplicated Firewall
```

Modul 2.1 Parameter instalasi Apache dan Pengaturan Firewall

Kemudian dilanjutkan untuk pemasangan MySQL sebagai alat untuk penyimpanan basis data. Parameter yang digunakan untuk instalasi MySQL dapat dilihat pada penjelasan **Modul 2.2** berikut ini:

```
$ sudo apt install mysql-server
Keterangan:
- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket mysql
- mysql-server : berisi MySQL dan semua dependensi

$ sudo mysql_secure_installation
Keterangan:
- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- mysql_secure_installation : merupakan perintah untuk menginstal mysql secara secure
```

Modul 2.2 Parameter instalasi MySQL

Dalam membangun Sistem Informasi Dinas Kearsipan dan Perpustakaan, Bahasa yang digunakan adalah PHP. Parameter yang digunakan untuk instalasi PHP dapat dilihat pada penjelasan **Modul 2.3** berikut ini:

```
$ sudo apt install php libapache2-mod-php php-mysql
Keterangan:
- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket php
- php : singkatan dari Hypertext Preprocessor
- libapache2-mod-php: Paket ini menyediakan modul PHP untuk server web Apache 2
- php-mysql : layanan konektivitas antara web server dengan mysql

$ sudo nano /var/www/html/info.php
Keterangan:
- nano : merupakan aplikasi sejenis Notepad namun berbasis CLI
- /var/www/html/info.php : direktori yang dituju oleh nano untuk dibuka
- Info.php : merupakan file yang akan dibuka oleh nano

<?php
    phpinfo();
?>
Keterangan: merupakan kodingan yang dituliskan pada info.php yang digunakan untuk mengetes hasil instalasi php
```

Modul 2.3 Parameter instalasi PHP

Selanjutnya dilanjutkan untuk pemasangan PHPMYAdmin yang berguna untuk manajemen basis data berbasis *web* GUI. Parameter yang digunakan untuk instalasi PHPMYAdmin dapat dilihat pada penjelasan **Modul 2.4** berikut ini:

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

Keterangan:

- `sudo` : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- `apt` : merupakan package manager pada Ubuntu
- `install` : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket phpMyAdmin
- `phpMyAdmin` : merupakan paket yang akan diinstal
- `php-mbstring php-gettext` : merupakan file tujuan yang akan diinstal pada system

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

Keterangan:

- Perintah tersebut digunakan untuk memperbarui password yang terdapat pada user yang bernama root
- `Authetcation_sring` : merupakan password yang akan diperbarui
- `KATA_SANDO_ROOT_USER` : merupakan password yang baru yang akan menggantikan password lama
- `Where User = 'root'` : adalah perintah SQL yang digunakan sebagai keterangan user mana yang dituju yang akan diperbarui passwordnya. Dalam kasus tersebut nama user yang akan diperbarui adalah 'root'

Modul 2.4 Parameter instalasi PHPMYAdmin

Salah satu cara untuk memasang Docker Engine adalah dengan menggunakan *repository*. Maka dari itu diperlukan mengatur *repository* Docker. Parameter yang digunakan untuk mengatur *repository* Docker dapat dilihat pada **Modul 2.5** berikut ini:

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

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

Keterangan:

- `apt` : adalah parameter yang digunakan untuk memperbarui paket pada repository
- `apt-transport-https, ca-certificates, curl, gnupg-agent, software-property-common` : adalah nama paket pada repository

Modul 2.5 Parameter set up repository

Proses selanjutnya adalah menambahkan Docker's *Official* GPG Key. Parameter yang digunakan untuk menambahkan GPG key dapat dilihat pada **Modul 2.6** berikut ini:

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

Keterangan:

- curl : adalah perintah untuk memindahkan data dengan berbagai network protocol
- 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 GPG Key

Setelah pengaturan *repository* telah berhasil dilakukan. Proses selanjutnya adalah pemasangan Docker Engine. Tersedia 3 *repository* pada Docker yaitu : *stable*, *nightly*, dan *test*. Versi *repository* Docker yang digunakan pada proyek ini adalah versi *stable*. Parameter yang digunakan untuk memilih *repository* dapat dilihat pada penjelasan **Modul 2.7** berikut ini:

```
$ sudo add-apt-repository \
  "deb [arch=arm64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
```

Keterangan:

- add-apt-repository : command yang berguna untuk menambahkan repository
- lsb_release : command untuk membantu mengidentifikasi distribusi Linux yang digunakan dan kepatuhannya dengan Linux Standar Base
- stable : versi repository yang akan dipasang

Modul 2.7 Parameter pemasangan *stable repository*

Kemudian dilanjutkan untuk pemasangan Docker Engine versi terbaru. Parameter yang digunakan untuk pemasangan Docker Engine dapat dilihat pada **Modul 2.8** berikut ini:

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

Keterangan:

- docker-ce : versi Docker yang akan dipasang adalah community edition
- docker-ce-cli : untuk memasang paket docker engine command line
- containerd.io : untuk menambah fasilitas docker container

Modul 2.8 Parameter pemasangan Docker Engine

Agar mempermudah untuk menjalankan banyak servis, maka diperlukan docker-compose yang berguna untuk menjankan servis *multi-container*. Parameter pemasangan docker-compose dapat dilihat pada **Modul 2.9** berikut ini:

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

Keterangan:

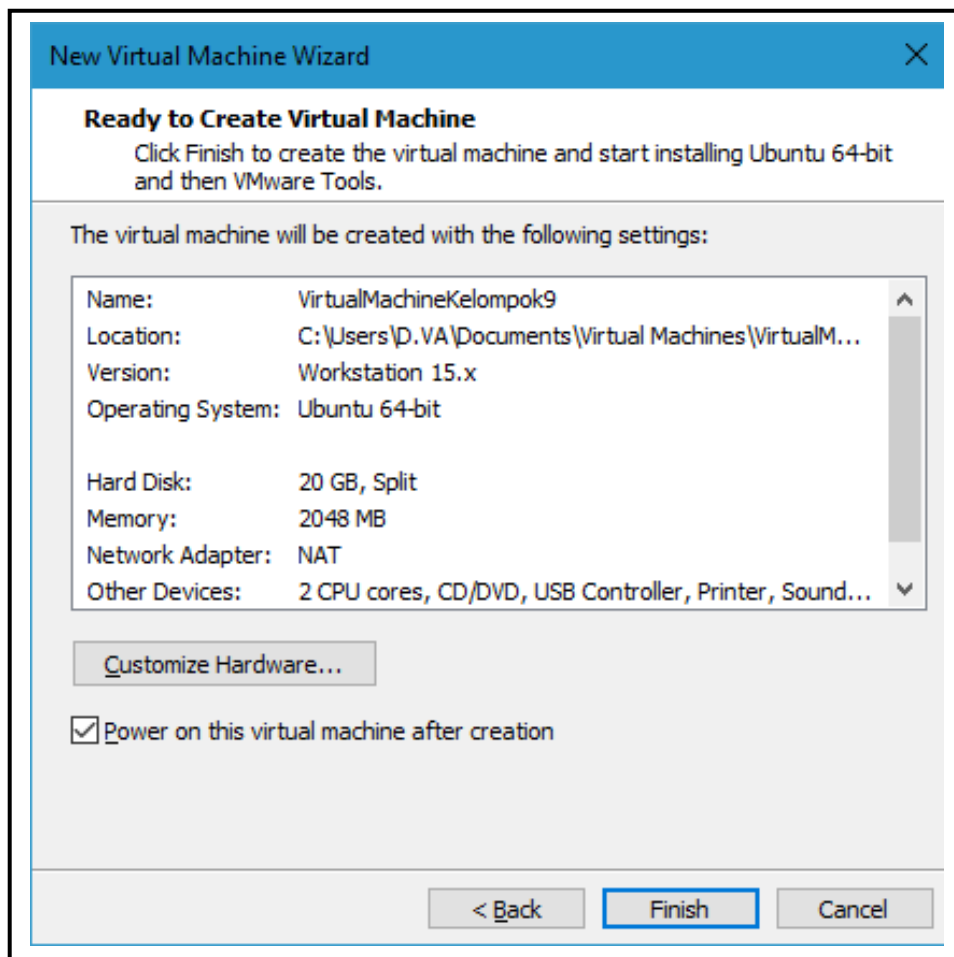
- https://github.com/docker/compose/releases/download/1.25.5/dockercompose : tautan tujuan yang berisi docker-compose

- \$(uname -s)-\$(uname -m) : parameter username dan password
- -o /usr/local/bin/docker-compose : direktori pemasangan

Modul 2.9 Parameter pemasangan docker-compose

2.4 Tahap Implementasi

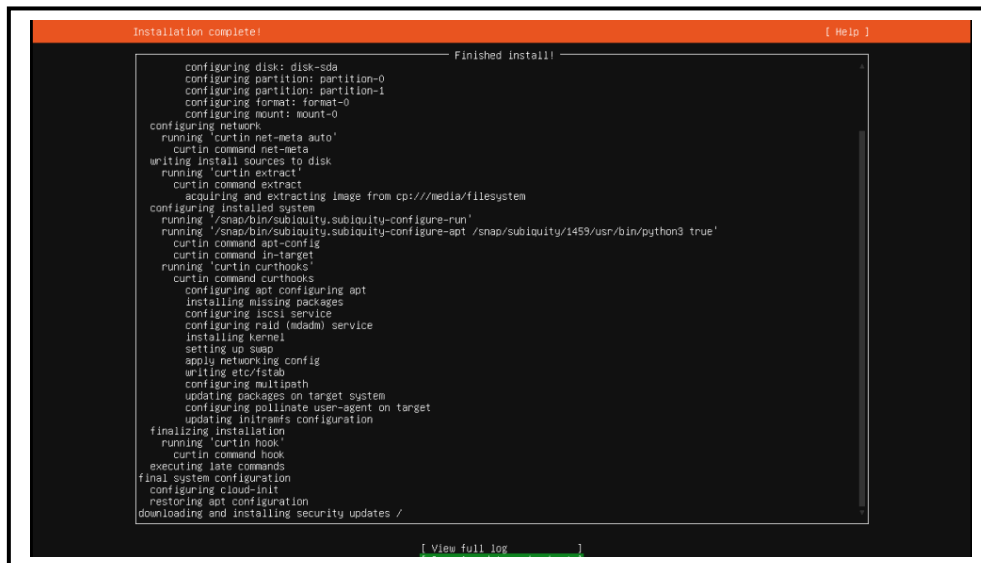
Untuk mewujudkan tujuan *hosting* “Sistem Informasi Dinas Kearsipan dan Perpustakaan pada Ubuntu LAMPP dan Dockerfilenya”, tahapan pertama yang harus dilakukan adalah membuat *virtual machine* dengan VMWare Workstation. *Virtual machine* ini berguna untuk memvirtualisasikan sistem operasi Ubuntu untuk berjalan diatas sistem operasi Windows. Pengaturan konfigurasi *virtual machine* dapat dilakukan seperti pada **Gambar 2.4** berikut:



Gambar 2.4 Ringkasan konfigurasi

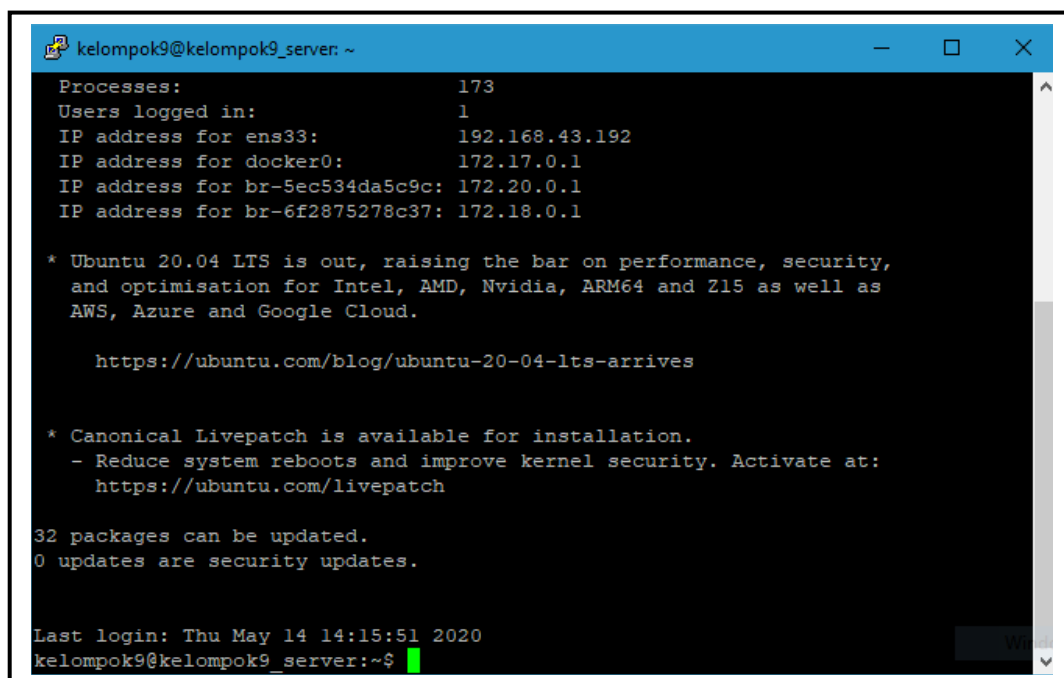
Dapat dilihat pada ringkasan konfigurasi pada **Gambar 2.4** bahwa *check box* “Power on this virtual machine after creation” yang memiliki arti bahwa setelah instalasi Ubuntu telah dilakukan maka akan otomatis menghidupkan Ubuntu untuk pertama kali. Setelah

menekan tombol *Finish* proses instalasi dan konfigurasi Ubuntu 18.04.4 LTS akan segera dilakukan. Ilustrasi proses instalasi Ubuntu dapat dilihat pada **Gambar 2.5** berikut ini:



Gambar 2.5 Proses instalasi Ubuntu

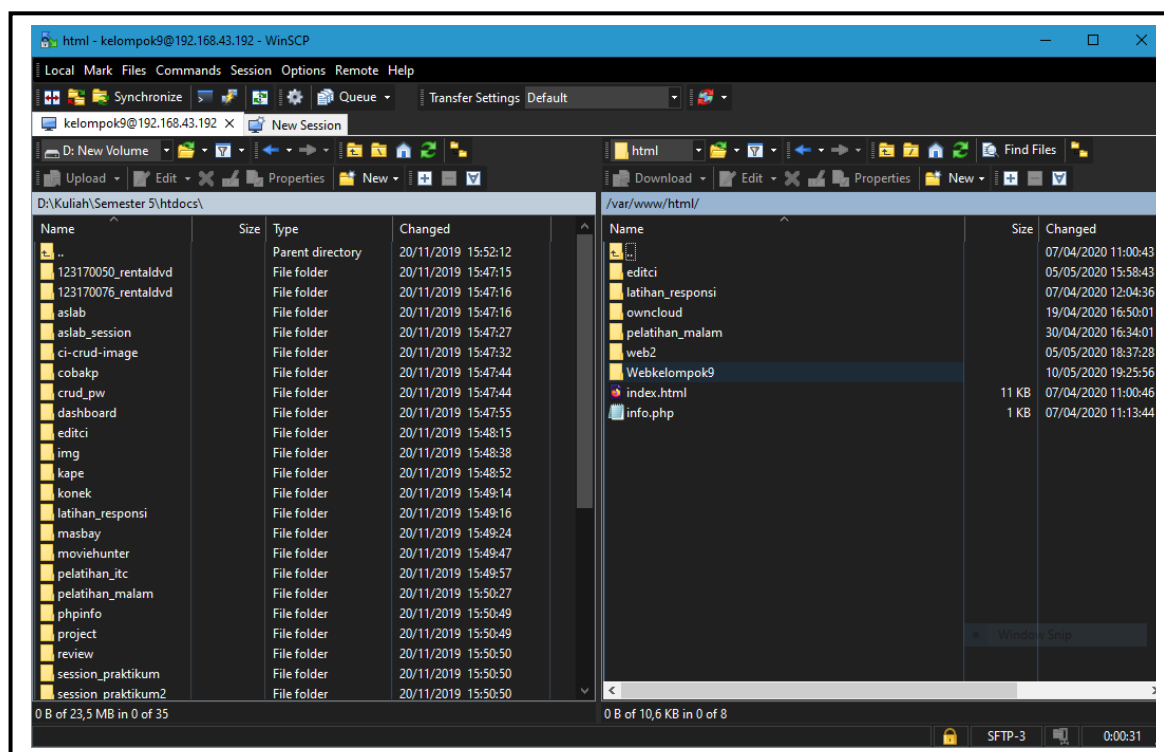
Setelah instalasi selesai, maka akan otomatis *booting* kedalam Ubuntu untuk pertama kali. Selanjutnya login dengan *username* dan *password* sesuai dengan data yang telah dimasukan pada saat instalasi Ubuntu. Setelah masuk dalam Ubuntu ketikkan perintah `ifconfig` untuk melihat IP *Server* yang digunakan oleh Ubuntu tersebut. IP tersebut akan digunakan untuk masuk PuTTY. Jika proses masuk kedalam Putty berhasil maka akan akan muncul tampilan Putty. Ilustrasi PuTTY dapat dilihat pada **Gambar** berikut:



Gambar 2.6 Remote server menggunakan PuTTY

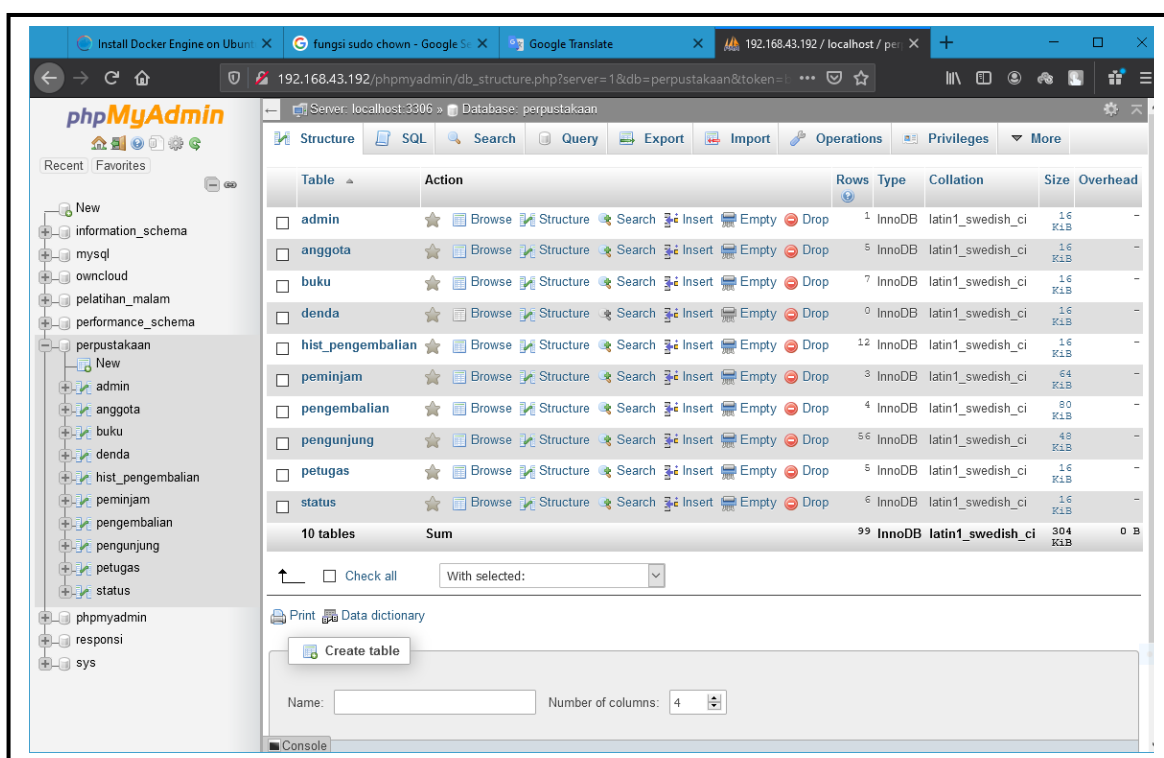
Kemudian tahapan berikutnya adalah menginstal Apache, MySQL, PHP, dan PHPMyAdmin pada Ubuntu 18.04.4 LTS menggunakan perantara *remote server* PuTTY. Parameter instalasi Apache, MySQL, PHP, dan PHPMyAdmin dapat dilihat pada **Modul 2.1, Modul 2.2, Modul 2.3, dan Modul 2.4**. Setelah semua komponen LAMPP terpasang dengan baik pada Ubuntu 18.04.4 LTS, maka artinya semua *dependency* yang diperlukan untuk menjalankan sebuah *web* sudah terpasang. Setelah itu dilanjutkan proses *hosting* dengan cara membuka aplikasi WinSCP yang berguna untuk pemindahan data dari windows ke dalam Ubuntu dengan menggunakan prinsip FTP (*File Transfer Protocol*).

Untuk membuka WinSCP diperlukan alamat *IP server*, *password*, dan *username* dari Ubuntu *server* yang akan dibuka. Setelah selesai *login* WinSCP, langkah selanjutnya adalah menekan tombol *Open Directory* pada menu WinSCP untuk membuka direktori *root* Apache pada Ubuntu. Adapun alamat direktori *root* Apache adalah `/var/www/html/`. Sebelum melakukan pemindahan folder web, terlebih dahulu melakukan pengambilan kepemilikan folder `/var/www/html/` dengan menjalankan perintah `sudo chown nama_user /var/www/html/` pada PuTTY.. Ilustrasi pemindahan folder web dengan WinSCP dapat dilihat pada **Gambar 2.7** berikut ini:



Gambar 2.7 Pemindahan data dengan WinSCP

Jika data *web* telah berhasil dipindahkan ke dalam *server*, maka agar *web* dapat digunakan dengan baik tahap selanjutnya adalah *import* basis data web dengan menggunakan PHPMyAdmin yang sebelumnya telah instal. Untuk mengakses PHPMyAdmin Ubuntu *server* dapat dilakukan dengan cara membukanya lewat *browser* menggunakan alamat `http://ip-server/phpmyadmin`. Setelah laman PHPMyAdmin terbuka kemudian dilanjutkan membuat *database* dengan nama sesuai keperluan dan lakukan proses *import* file `.sql` kedalam *database* yang telah dibuat. Ilustrasi *import database* menggunakan PHPMyAdmin dapat dilihat pada **Gambar 2.8** berikut ini:

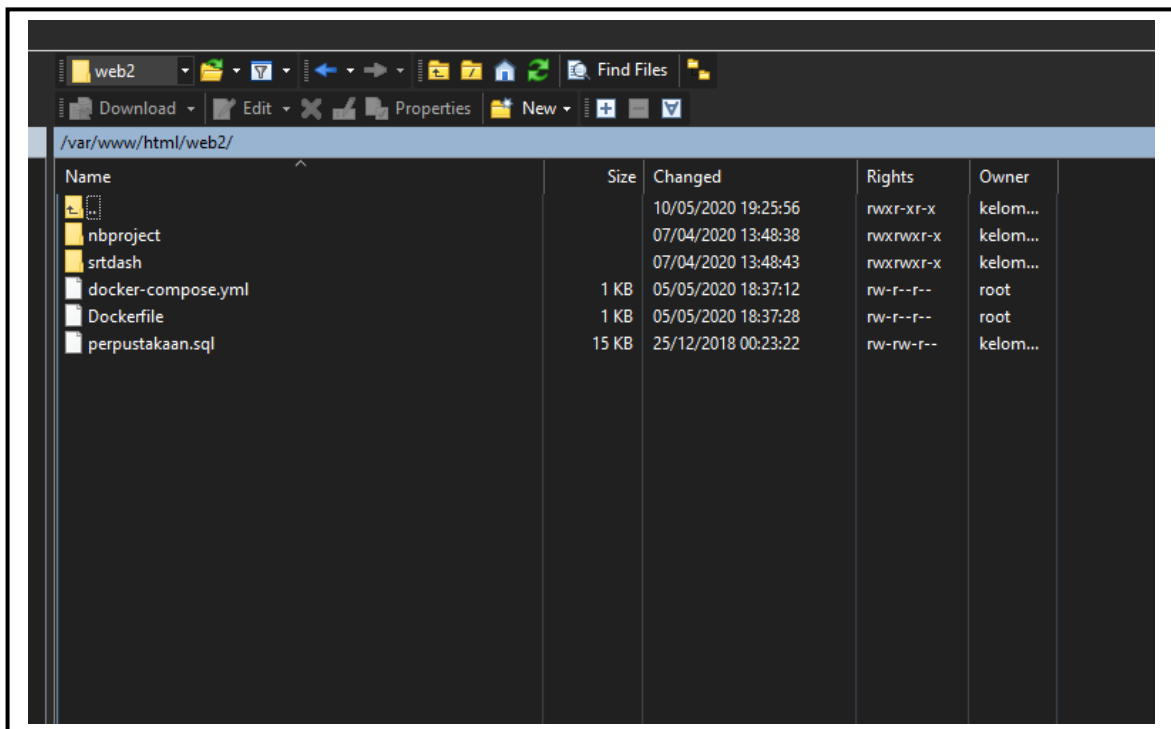


Gambar 2.8 *Import database* PHPMyAdmin

Jika *import database* berhasil dilakukan maka web Sistem Informasi Dinas Kearsipan dan Perpustakaan telah siap dilakukan dan berhasil di *hosting* ke Ubuntu *server*. Untuk mengakses web tersebut dilakukan dengan cara `http://ip-server/web-perpustakaan`.

Kemudian dilanjutkan untuk pembuatan Dockerfile dari web “Sistem Informasi Dinas Kearsipan dan Perpustakaan”. Langkah yang harus dilakukan adalah memasang Docker Engine ke dalam *server* dengan menggunakan perantara PuTTY. Parameter pemasangan Docker Engine tertera pada **Modul 2.5** sampai **Modul 2.9**. Proses pada parameter tersebut akan mengatur *repository* docker, mengunduh dan memasang Docker Engine pada *server*, dan memasang docker-compose yang berguna untuk menjalankan servis

multi-container. Apabila pemasangan Docker Engine telah dilakukan dengan baik, maka selanjutnya menuju *folder web* Sistem Informasi Dinas Kearsipan dan Perpustakaan yang berada pada direktori `/var/www/html/`. Kemudian buat *file* Dockerfile dan docker-compose.yml di dalam direktori web tersebut. Dockerfile dan docker-compose.yml berisi parameter parameter yang berguna untuk membuat dan menjalankan *environment* docker untuk *web*. Parameter Dockerfile dan docker-compose.yml tertera pada **Lampiran 2.1** dan **Lampiran 2.2**. Apabila kedua file tersebut telah berhasil dibuat, maka isi dari folder web akan seperti pada **Gambar 2.9** berikut:



Gambar 2.9 root folder web

Docker-compose berguna untuk menjalankan beberapa servis secara bersamaan. Dikarenakan untuk menjalankan *web* dengan baik diperlukan dua *container* yaitu *container web-server* dan *container database*, maka untuk mempermudah menjalankan dua servis tersebut secara bersamaan digunakanlah docker-compose. Parameter untuk menjalankan docker-compose tertera pada **Modul 3.0** berikut ini:

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

Keterangan:

- docker-compose : perintah yang digunakan untuk menjalankan parameter yang ada pada docker-compose.yml
- up : perintah untuk menjalankan docker compose dan mengeksekusi semua parameter di dalamnya

- `-d` : perintah agar proses composing dilakukan dibalik layer untuk meringkas proses di CLI

Modul 3.0 Parameter untuk menjalankan docker-compose

Pada saat pertama kali menjalankan docker-compose proses yang dilakukan adalah *pulling dependency images* dari Docker Hub seperti Apache, MySQL, dan PHP. Proses tersebut memakan waktu tergantung dengan kecepatan koneksi internet yang dimiliki. Setelah *dependency images* berhasil diunduh, maka semua *dependency* akan dijalankan secara otomatis ke dalam Docker Container. Apabila *web-server service* dan *database service* berhasil dijalankan dalam *Container*, maka dilanjutkan dengan *import* database yang web Sistem Informasi Dinas Kearsipan dan Perpustakaan ke dalam MySQL yang sebelumnya telah di *pull* dari Docker Hub. Parameter untuk *import database* tertera pada **Modul 3.1** berikut ini:

```
$ docker exec -i perpustakaan-db mysql -uroot -proot perpustakaan <
./perpustakaan.sql
```

Keterangan

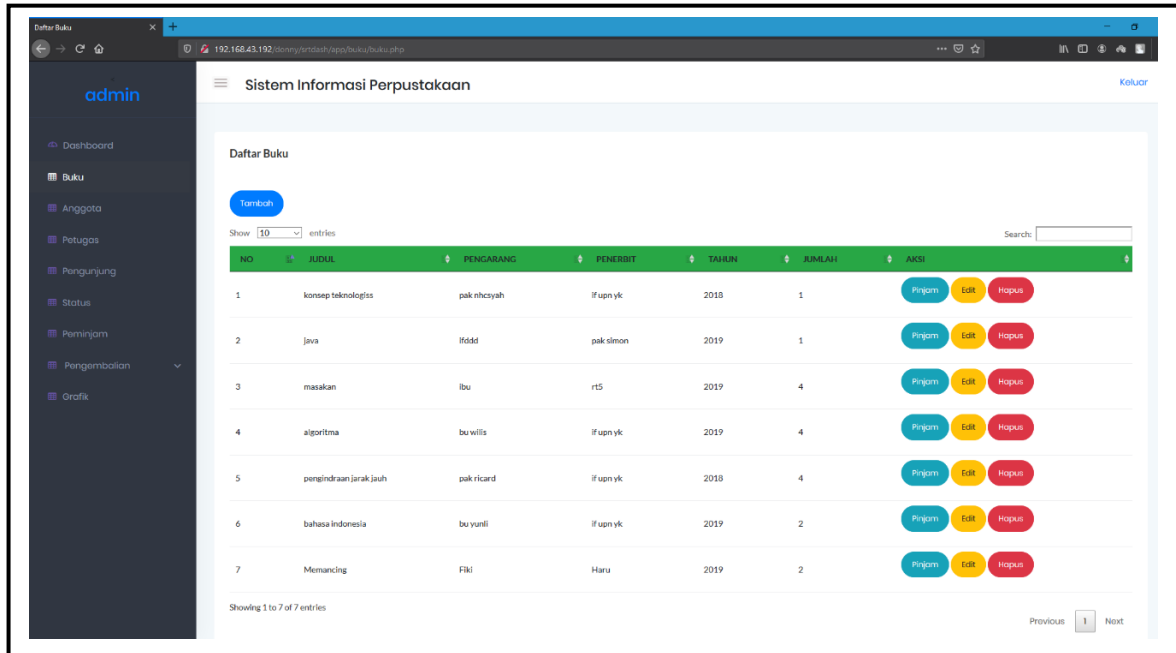
- `exec` : perintah dalam docker yang berguna untuk mengeksekusi suatu perintah pada container yang sedang berjalan
- `perpustakaan-db` : nama container database yang telah di deklarasikan di dalam docker-compose
- `mysql` : servis tujuan adalah mysql service
- `-uroot, -proot` : username dan password yang berisi root. Data tersebut telah dideklarasikan didalam docker-compose
- `perpustakaan` : nama database tujuan
- `./perpustakaan.sql` : file import tujuan. Berada didalam satu folder dengan Dockerfile dan docker-compose.

Modul 3.1 Parameter *import database*

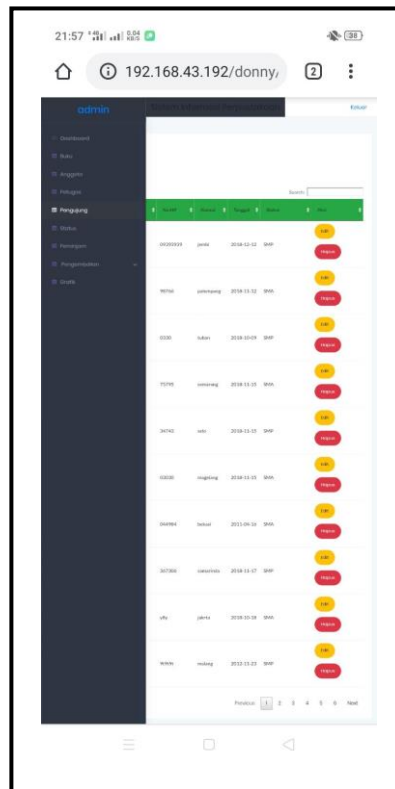
Apabila semua *dependency* telah berjalan di dalam *docker container* dan *database* berhasil diimpor, maka web Sistem Informasi Dinas Kearsipan dan Perpustakaan telah berhasil berjalan dalam *docker environment*. Untuk mengakses *web* tersebut dapat dilakukan dengan cara membukanya dari *web browser* dan megakses `http://ip-server:8002`. Dalam kasus ini untuk mengakses *web* tersebut dengan `http://192.168.43.192:8002`. 8002 merupakan *port* yang telah diekspos dari *default* Apache port Ubuntu yaitu 80. Tujuan dari mengekspos *port* tersebut adalah agar *port* tersebut dapat diakses dari sistem operasi Windows.

2.5 Hasil Implementasi

Setelah melakukan instalasi *hosting local* menggunakan Ubuntu LAMPP, Tampilan web yang diakses melalui *desktop* dapat dilihat pada **Gambar 3.0** dan tampilan *web* yang diakses melalui *handphone* dapat dilihat pada **Gambar 3.1** berikut:

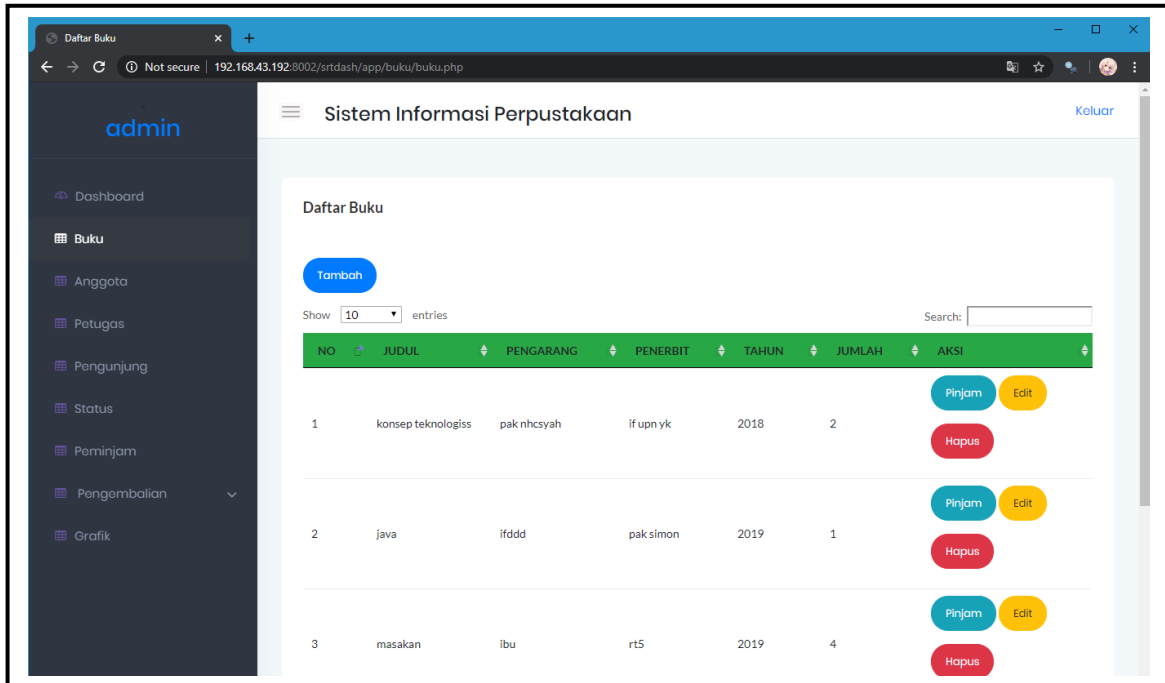


Gambar 3.0 Tampilan *web* diakses lewat *browser desktop*

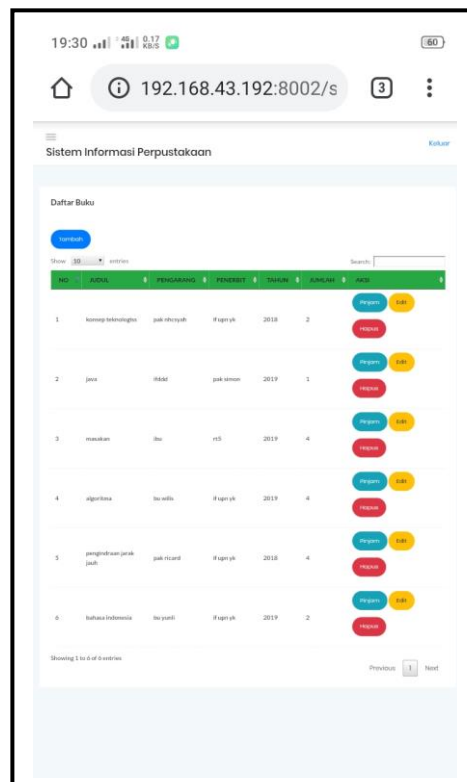


Gambar 3.1 Tampilan *web* yang diakses dari *handphone*

Kemudian proyek dilanjutkan untuk dijalankan dalam Docker *environment*. Tampilan proyek yang diakses dari *desktop* dapat dilihat pada **Gambar 3.2** dan dari *handphone* dapat dilihat pada **Gambar 3.3** berikut ini:



Gambar 3.2 Tampilan *desktop* dengan Docker *environment*

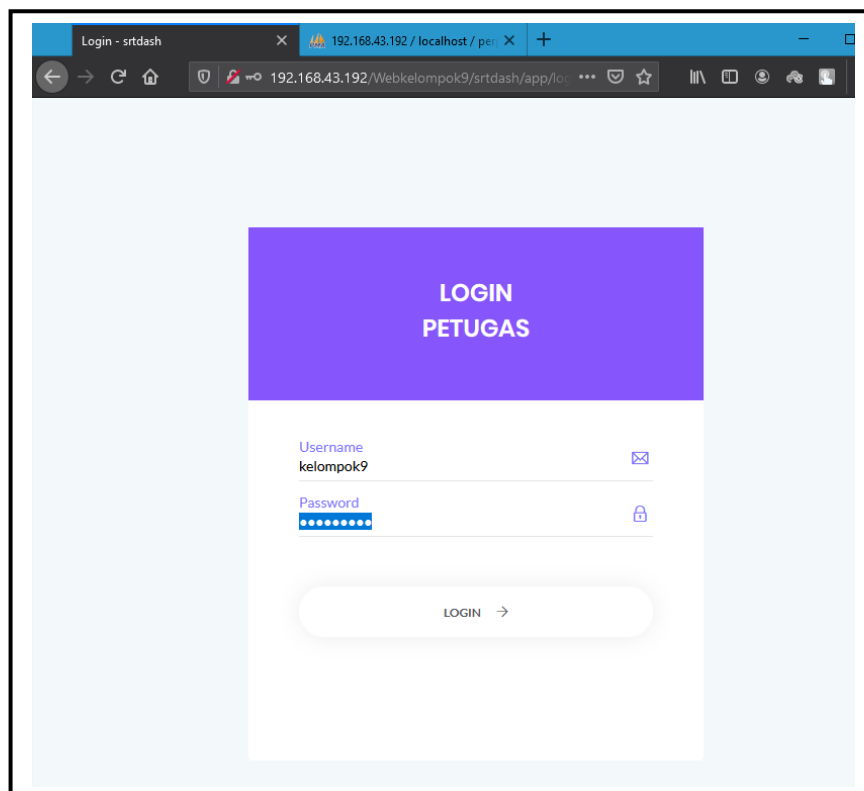


Gambar 3.3 Tampilan *handphone* dengan Docker *environment*

2.6 Pengujian Singkat

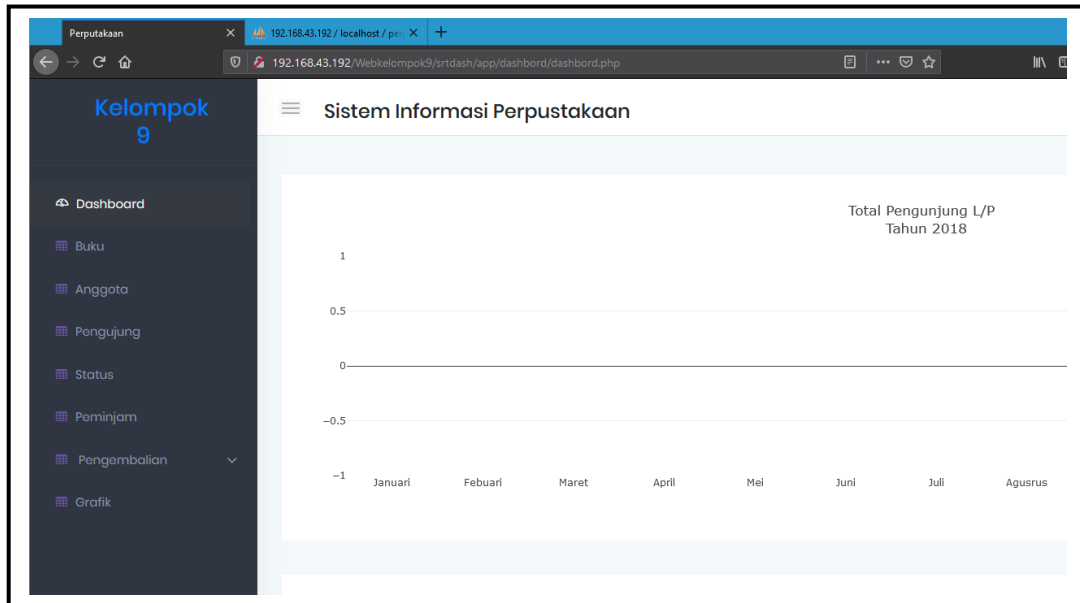
Pada pengujian singkat ini terdapat dua skenario pengujian yaitu pengujian *web* LAMPP yang di *hosting* secara *local* dan pengujian *web* yang dijalankan di dalam Docker *environment*. Skenario pertama *web* LAMPP akan diakses menggunakan *desktop* dan *handphone* untuk membuka menu “Dashboard” web sebagai petugas. Skenario kedua menjalankan *web* dalam Docker *environment* untuk menjalankan fitur CRUD buku sebagai admin. *Create* dan *read* akan dilakukan melauai *desktop* lalu *update* dan *delete* akan dilakukan melalui *handphone*. Penggunaan *desktop* dan *handphone* adalah sebagai bukti keberhasilan implementasi *cloud computing* dan kedua perangkat mengacu pada *database* yang sama.

Langkah pertama untuk melakukan skenario pertama adalah dengan membuka *browser* dan mengakses web LAMPP dengan alamat `http://[ip-server]/[nama_web]`. Tampilan antar muka *login* petugas dapat dilihat pada **Gambar 3.2** berikut ini:



Gambar 3.2 Tampilan *login* petugas di *desktop*

Setelah berhasil *login* maka akan masuk pada menu “*Dashboard*” sebagai petugas. Tampilan menu “*Dashboard*” di *desktop* dapat dilihat pada **Gambar 3.3** berikut ini:

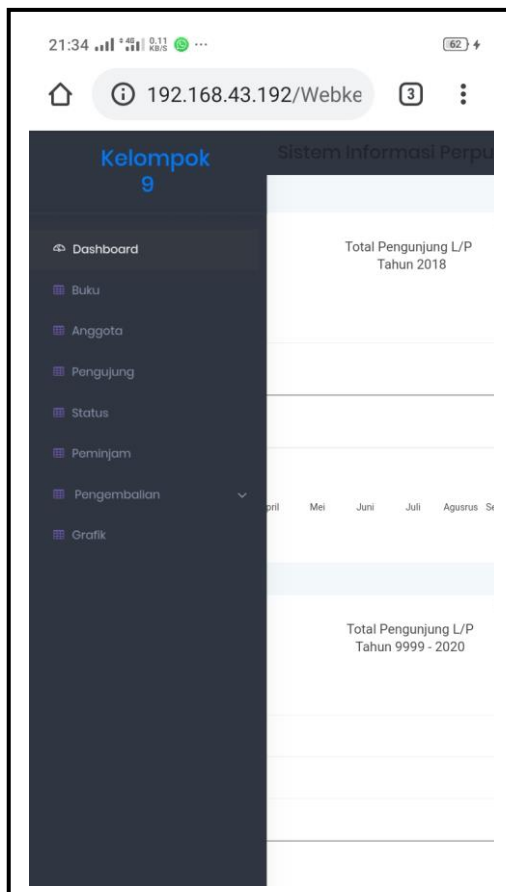


Gambar 3.3 Tampilan *dashboard* petugas pada *desktop*

Kemudian dilanjutkan dengan pengaksesan web LAMPP menggunakan *handphone*. Tampilan *login* petugas versi *handphone* dapat dilihat pada **Gambar 3.4** berikut ini:

Gambar 3.4 Antarmuka *login* petugas di *handphone*

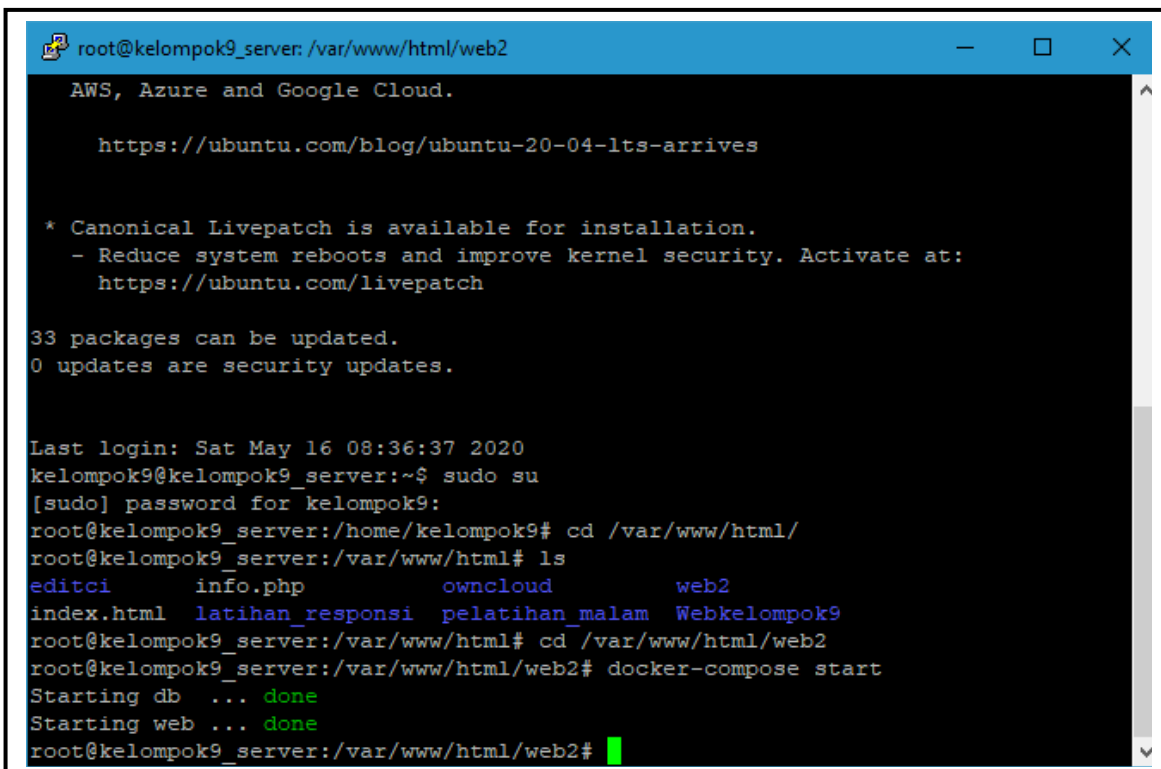
Setelah proses *login* sebagai petugas berhasil maka akan masuk kedalam menu “*Dashboard*” petugas. Tampilan manu “*Dashboard*” petugas di *handphone* dapat dilihat pada **Gambar 3.5** berikut ini:



Gambar 3.5 Antarmuka *dashboard* petugas di *handphone*

Dengan begitu skenario pertama untuk membuka web Sistem Informasi Dinas Kearsipan dan Perpustakaan dengan menggunakan *desktop* dan *handphone* dengan pengujian *login* dan masuk menu “*Dashboard*” berhasil dilakukan.

Kemudian dilanjutkan untuk melakukan skenario yang kedua. Untuk melakukan skenario yang kedua, sebelum dapat mengakses web dalam *docker environment* terlebih dahulu diperlukan *set up docker* itu sendiri. Proses tersebut dilakukan dengan membukanya lewat PuTTY lalu berpindah direktori ke dalam *folder web* Sistem Informasi Dinas Kearsipan dan Perpustakaan. Setelah berhasil berpindah direktori untuk memulai servis *docker* dapat digunakan perintah “*docker-compose start*” untuk menghidupkan *dependency* yang sebelumnya telah dipasang. Setelah servis *docker* telah hidup maka *web* dapat diakses melalui *browser*. Tampilan PuTTY untuk menghidupkan servis *dependency* dapat dilihat pada **Gambar 3.6** berikut ini:



```

root@kelompok9_server: /var/www/html/web2
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

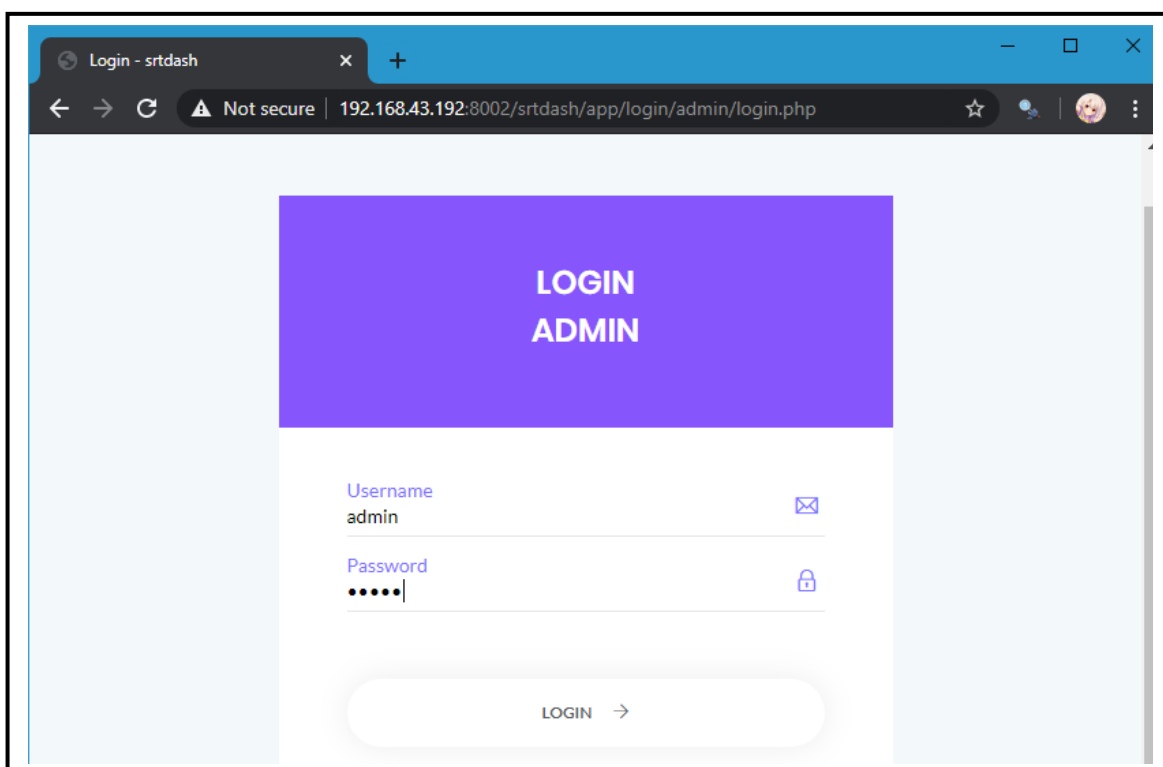
33 packages can be updated.
0 updates are security updates.

Last login: Sat May 16 08:36:37 2020
kelompok9@kelompok9_server:~$ sudo su
[sudo] password for kelompok9:
root@kelompok9_server:/home/kelompok9# cd /var/www/html/
root@kelompok9_server:/var/www/html# ls
editci      info.php    owncloud    web2
index.html  latihan_responsi  pelatihan_malam  Webkelompok9
root@kelompok9_server:/var/www/html# cd /var/www/html/web2
root@kelompok9_server:/var/www/html/web2# docker-compose start
Starting db ... done
Starting web ... done
root@kelompok9_server:/var/www/html/web2#

```

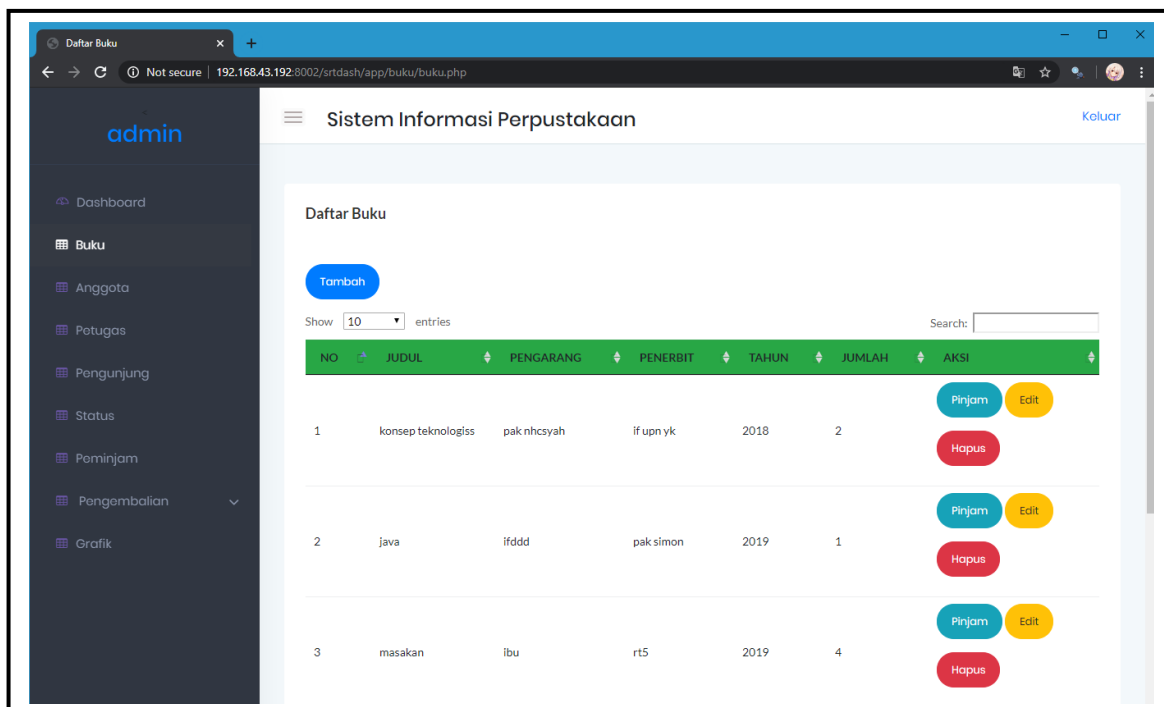
Gambar 3.6 Tampilan docker compose start

Setelah itu buka *web* dengan menggunakan *browser*. Tampilan halaman *login* dapat dilihat pada **Gambar 3.7** berikut ini:



Gambar 3.7 Halaman login admin

Selanjutnya masuk dalam menu “Daftar Buku” dengan menekan menu tersebut pada *sidebar* dibagian kiri halaman. Tampilan menu “Daftar Buku” dapat dilihat pada **Gambar 3.8** berikut ini:



Gambar 3.8 Tampilan menu Daftar Buku

Setelah itu tekan tombol “Tambah” untuk menambah data buku baru. *Form* data buku baru diisi seperti pada **Gambar 3.9** berikut ini:

Tambah Buku

Kode Register: 045

Judul: What Do You Do at the End? Are You Busy? Will You Save Us?

Pengarang: Akira Kareno

Penerbit: Kadokawa Shoten

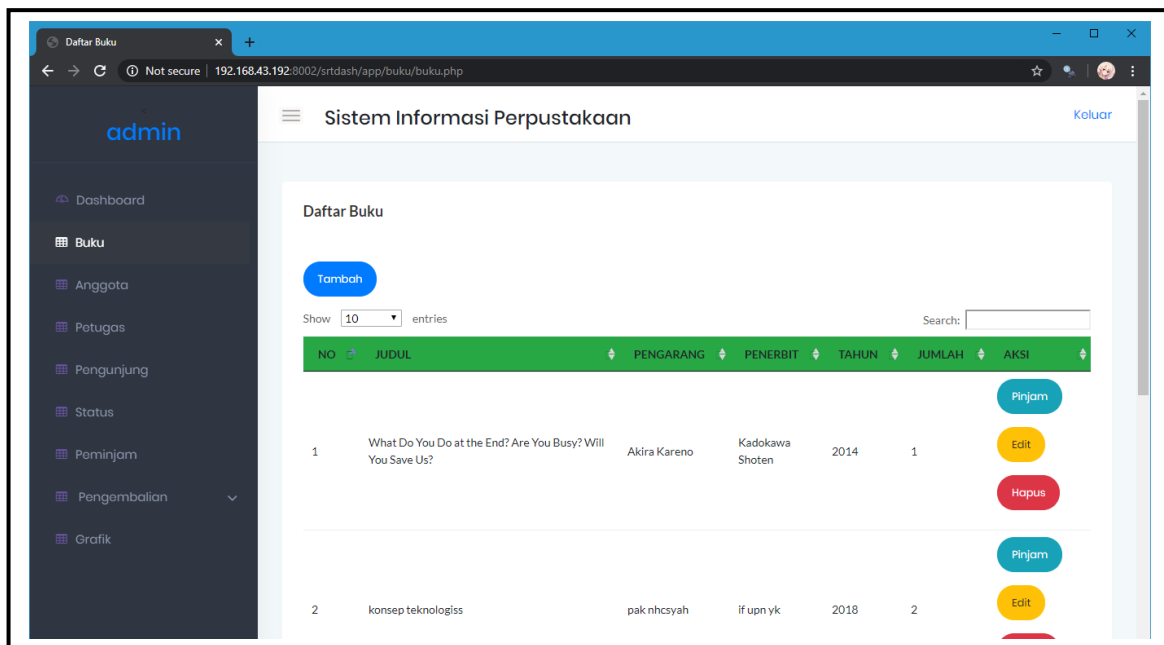
Tahun: 2014

Jumlah: 1

Tambah

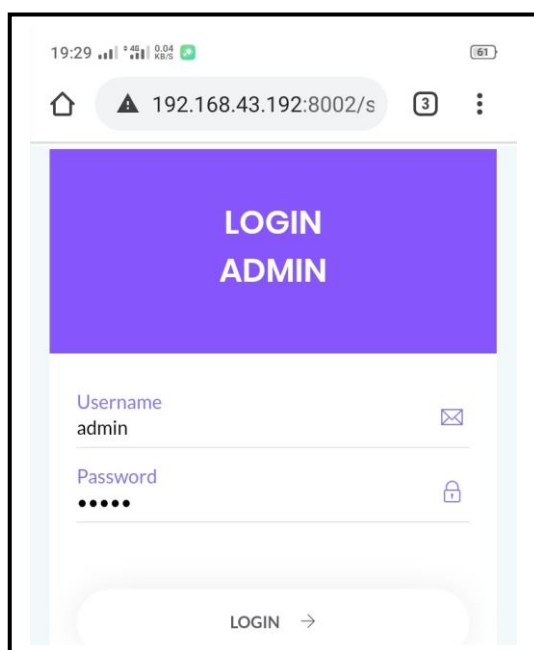
Gambar 3.9 Tampilan *Create* buku baru

Setelah menekan tombol “Tambah”, maka otomatis akan di *redirect* menuju menu Daftar Buku dan daftar buku otomatis akan bertambah dengan data buku yang baru saja ditambahkan. Tampilan menu Daftar Buku dapat dilihat pada **Gambar 4.0** berikut ini:



Gambar 4.0 Tampilan Daftar Buku

Untuk selanjutnya adalah proses *update* dan *delete*. Kedua proses tersebut akan dieksekusi menggunakan perangkat *handphone*. Tampilan login pada *handphone* dapat dilihat pada **Gambar 4.1** berikut ini:



Gambar 4.1 Tampilan *login handphone*

Setelah *login* sebagai admin menggunakan *handphone*, maka menuju data buku yang telah tambahkan sebelumnya dan pilih menu “*Edit*” untuk melakukan *update* data. Adapaun pengisian data *update* dapat dilihat pada **Gambar 4.2** berikut ini:

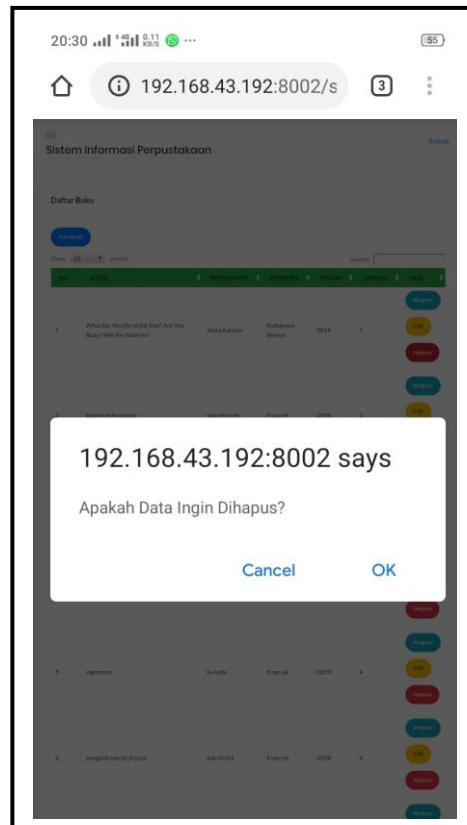
The screenshot shows a mobile application interface for a library information system. The status bar at the top displays the time 20:29, signal strength, and battery level. The address bar shows the URL 192.168.43.192:8002/s. The main header is 'Sistem Informasi Perpustakaan'. Below this is a section titled 'Edit Buku'. The form contains the following fields:

- Kode Register: 045
- Judul: What Do You Do at the End? Are You Busy? Will You Save Us?
- Pengarang: Akira Kareno
- Penerbit: Kadokawa Shoten
- Tahun: 2014
- Jumlah: 1

A yellow button labeled 'Update' is located at the bottom right of the form.

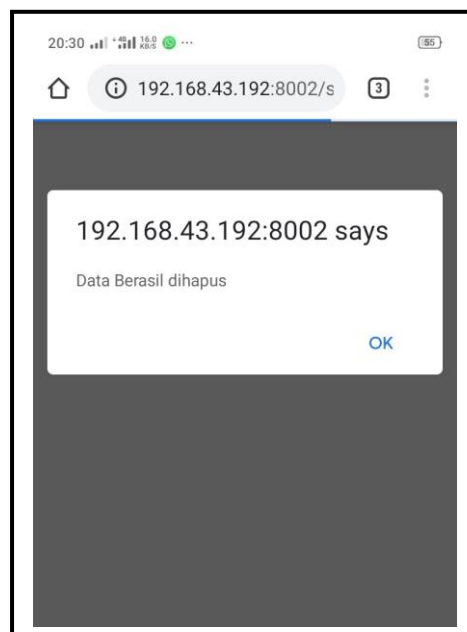
Gambar 4.2 Tampilan *update* data di *handphone*

Setelah menekan tombol “Update” maka akan otomatis dialihkan ke menu Daftar buku dapat dilihat pada **Gambar 4.0**. Daftar buku yang diakses melalui *handphone* maupun *desktop* adalah data daftar buku yang sama dikarenakan keduanya mengacu pada *database* yang sama. Jadi perubahan apapun yang terjadi pada data, maka akan saling berefek satu sama lain. Dan terakhir di dalam menu Daftar Buku tekan tombol “Hapus” pada buku yang telah dilakukan *update*. Tampilan *delete* buku dapat dilihat pada **Gambar 4.3** berikut:



Gambar 4.3 Delete buku pada handphone

Setelah “*delete confirmation dialog*” ditekan “OK”, maka data pada *database* akan otomatis terhapus. Dengan begitu pengujian CRUD Sistem Informasi Dinas Kearsipan dan Perpustakaan berhasil dilakukan. Adapun tampilan data buku terhapus dapat dilihat pada **Gambar 4.4** berikut ini:



Gambar 4.4 Tampilan berhasil dihapus di *handphon*

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 hingga Mei tahun 2020:

Tabel 3.1 Agenda Pengerjaan Proyek

No.	Jenis Tugas	Waktu Pengerjaan							
		Maret		April				Mei	
		3	4	1	2	3	4	1	2
1.	Analisa Persoalan								
2.	Pembagian Tugas								
3.	Pengerjaan Ubuntu LAMPP								
4.	Pengerjaan Laporan Proyek Pertama								
5.	Submisi Laporan dan Proyek Pertama								
6.	Revisi Laporan Proyek Pertama								
7.	Pengerjaan Docker								
8.	Submisi Laporan Proyek Kedua								
9.	Revisi Laporan Proyek Akhir dan Melanjutkan Proyek								
10.	Submisi 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	Fiki, Eka
2.	Pengujian Singkat	Fiki, Eka
3.	Latar Belakang Masalah	Eka
4.	Agenda Pengerjaan Proyek	Fiki
5.	Ubuntu LAMPP	Fiki, Eka
6.	Dockerfile	Fiki
7.	Laporan	Fiki, Eka

BAB IV

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan masalah yaitu membuat Sistem Informasi Dinas Kearsipan dan Perpustakaan untuk di *hosting* secara lokal menggunakan Ubuntu LAMPP lalu membuat Dockerfilenya telah dilakukan serangkaian pembuatan sistem dan telah dilakukan pengujian dengan skenario yang telah disiapkan membuahkan hasil yang memuaskan. Sistem yang diuji dapat berjalan dengan baik dan bekerja seperti yang diharapkan yang dibuktikan dengan pengujian *hosting local* dan Dockerfile yang keduanya dapat diakses dengan *desktop* maupun *handphone* dan setiap fasilitas *website* dapat digunakan dengan baik. Segala kendala selama pengerjaan proyek dapat diatasi dengan lancar.

Pengujian *hosting local* dilakukan dengan menggunakan *desktop* dan *handphone* untuk menguji *login* sebagai admin dan *login* sebagai petugas dan pengujian Dockerfile dilakukan menggunakan *desktop* dan *handphone* untuk menguji fasilitas CRUD (*create, read, update, delete*). Semua pengujian dapat dilakukan dengan baik sebagaimana mestinya.

Terdapat dua tugas permasalahan yaitu *hosting local* dengan menggunakan Ubuntu LAMPP dan pembuatan Dockerfilenya dan kelompok terdiri dari dua orang sehingga pembagian kelompok dibagi berdasarkan tugas 1 dan tugas 2. Setiap anggota bertanggung jawab dengan tugasnya dan membuahkan hasil yang memuaskan dan selesai pada waktu yang telah direncanakan.

4.2 Saran

Dikarenakan PSBB selama proses pengerjaan proyek akhir maka dengan terpaksa dilakukan pembagian tugas terhadap anggota kelompok yang mengakibatkan anggota kelompok lebih dominan terhadap materi apa yang dikerjakan. Pengerjaan proyek akhir seharusnya dilakukan secara bersama dan pada laptop yang sama sehingga dapat dilakukan *brainstorming* bersama untuk memecahkan masalah. Untuk menanggulangi masalah tersebut, telah dilakukan diskusi secara *online* untuk memberikan pemahaman materi tugas 1 dan 2 antar anggota kelompok lengkap dengan *flow* pengerjaan tugas.

DAFTAR PUSTAKA

Arutyunov, V. V. (2012). Cloud computing: Its history of development, modern state, and future considerations. *Scientific and Technical Information Processing*, 39(3), 173–178. <https://doi.org/10.3103/S0147688212030082>

Awan, K. (2011). *Cloud Computing k.awan*.

Dillon, T., Wu, C., & Chang, E. (2010). Cloud computing: Issues and challenges. *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, 27–33. <https://doi.org/10.1109/AINA.2010.187>

Ernawati, T. (2013). Analisis dan Pembangunan Infrastruktur Cloud Computing. *Jurnal Cybermatika*, 1(2), 17–23.

McGibbon, R. T., Beauchamp, K. A., Harrigan, M. P., Klein, C., Swails, J. M., Hernández, C. X., Schwantes, C. R., Wang, L. P., Lane, T. J., & Pande, V. S. (2015). MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. *Biophysical Journal*, 109(8), 1528–1532. <https://doi.org/10.1016/j.bpj.2015.08.015>

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

Sejarah *Cloud Computing*, <<http://www.postmedya.com/teknologi/cloud-computing-pengertian-sejarah-serta-manfaatnya/>>, (16 Nov 2018, *accessed* 16 April 2020)

LAMPIRAN

```
FROM php:7.3-apache
RUN apt-get update && apt-get upgrade -y
RUN docker-php-ext-install mysqli
EXPOSE 80
```

```
RUN a2enmod rewrite
RUN chmod -R 755 /var/www/html/
COPY ./ /var/www/html
RUN service apache2 restart
```

Keterangan:

- FROM php:7.3-apache : mengacu pada image yang akan digunakan pada Dockerfile
- RUN : mengekusi parameter
- EXPOSE : mengekspos port
- a2enmod rewrite : perintah untuk melakukan rewrite pada file tujuan dalam apache
- chmod : pengambilan hak pada suatu folder tujuan
- COPY : perintah untuk menyalin file. ./ berarti akan menyalin semua file di current directory ke tujuan yaitu /var/www/html/
- service apache2 restart : perintah untuk memulai ulang servis apache

Lampiran 2.1 Paramter Docluckerfile

```
version: '3.3'
services:
  web:
    build:
      context: ./
      dockerfile: Dockerfile
    container_name: perpustakaan-web
    depends_on:
      - db
    volumes:
      - ./:/var/www/html/
    ports:
      - 8002:80
  db:
    container_name: perpustakaan-db
    image: mysql:8.0
    command: --default-authentication-plugin=mysql_native_password
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: perpustakaan
      MYSQL_USER: kelompok9
      MYSQL_PASSWORD: kelompok9
    ports:
      - 6036:3306
```

Keterangan:

- version : versi yang digunakan docker-compose adalah 3.3
- services : terdapat dua service yang dijalankan docker-compose yaitu web-sercice dan database service
- web : nama image yang digunakan
- build : parameter untuk membangun image yang berisi

- context : direktori tujuan yang berisi proyek web. Dikarenakan docker-compose berada dalam folder proyek web itu sendiri, maka digunakan parameter ./ yang akan mereferensikan semua file yang berada pada current directory
- dockerfile : nama dockerfile yang digunakan. Pada kasus ini dockerfile memiliki nama Dockerfile
- container name : nama dari container untuk web-server SI Perpustakaan
- depends on : berarti container tersebut bergantung pada container lain yaitu container db
- volume : mount direktori ke direktori tujuan. ./ berarti semua file di current directory akan di mount ke direktori /var/www/html/
- ports : agar dapat servis dapat digunakan di sistem operasi host, maka port web-service harus di refleksikan. 8002:80 berarti port internal 80 di referensikan pada port eksternal 8002
- db : nama image yang akan dibuild
- container name : nama container dari database servis yaitu perpustakaan-db
- image : mysql:8.0 : image yang digunakan dalam docker-compose. Jika image belum tersedia, maka akan di pull dari Docker hub
- command : berguna untuk menjalankan perintah
- restart : perintah untuk restart container jika terjadi suatu masalah
- environment : adalah variable lingkungan dalam container.

Lampiran 2.2 Parameter docker-compose