

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING  
LAPORAN PROYEK AKHIR**

**SISTEM INFORMASI JUAL BELI SEAFOOD MENGGUNAKAN LAYANAN  
UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKERFILENYA**



**DISUSUN OLEH:**

**NAMA ANGGOTA : ANGGITA RAMADHANI 123170076  
SALSABILA LATIFFA FIKRI 123170084  
KELAS : A  
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 JUAL BELI SEAFOOD MENGGUNAKAN LAYANAN UBUNTU LAMPP DAN PROSES PEMBUATAN DOCKERFILENYA

Disusun oleh :

Anggita Ramadhani

123170076

Salsabila Latiffa Fikri

123170084

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

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Jaluanda Parama, S.Kom.

Muhammad Imam Alfatah

NIM. 123160119

Mengetahui,

Ka. Lab. Sistem Digital

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

NIK. 2 8201 13 0425 1

## **KATA PENGANTAR**

Assalamualaikum wr. wb.

Puji syukur dipanjatkan bagi Allah SWT karena berkat rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan praktikum Teknologi Cloud Computing serta laporan proyek akhir praktikum yang berjudul “Sistem Informasi Jual Beli Seafood Menggunakan Layanan Ubuntu LAMPP dan Proses Pembuatan Dockerfilenya”.

Laporan akhir ini merupakan tanggung jawab dan tugas kami dalam menjalani kuliah Teknik Informatika yang dimana dalam laporan ini tertera proyek akhir kami dalam menjalani setiap praktikum yang ada.

Tidak lupa ucapan terimakasih kepada asisten laboratorium yang selalu membimbing dan mengajari kami dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini tentu saja masih jauh dari yang diharapkan untuk memenuhi tugas akhir praktikum Teknologi Cloud Computing, sehingga masukan dari pihak sangat diharapkan untuk menjadikan tugas akhir kami menjadi semakin bagus dan berkualitas.

Wassalamualaikum, wr. wb.

Yogyakarta, 1 April 2020

Penyusun

## DAFTAR ISI

COVER.....	i
HALAMAN PENGESAHAN .....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI .....	iv
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang Masalah .....	1
1.2 Tujuan Proyek Akhir.....	2
1.3 Manfaat Proyek Akhir.....	3
1.4 Tahap Penyelesaian Proyek Akhir .....	4
<b>BAB II ISI DAN PEMBAHASAN .....</b>	<b>5</b>
2.1 Komponen yang Digunakan.....	5
2.2 Rancangan Arsitektur <i>Cloud Computing</i> .....	8
2.3 Parameter dan Konfigurasi.....	11
2.4 Tahap Implementasi .....	16
2.5 Hasil Implementasi.....	30
2.6 Pengujian Singkat.....	34
<b>BAB III JADWAL Pengerjaan dan Pembagian Tugas .....</b>	<b>45</b>
3.1 Agenda Pengerjaan.....	45
3.2 Keterangan Pembagian Tugas.....	45
<b>BAB IV KESIMPULAN DAN SARAN .....</b>	<b>46</b>
4.1 Kesimpulan.....	46
4.2 Saran.....	46
<b>DAFTAR PUSTAKA .....</b>	<b>47</b>
<b>LAMPIRAN .....</b>	<b>48</b>

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

*Cloud computing* ada ketika tahun 1960 dimana seorang ilmuwan komputer bernama John McCarthy meramalkan jika suatu hari nanti komputerisasi akan menjadi infrastruktur publik seperti berlangganan listrik atau telepon. Kemudian, pada akhir tahun 1990-an lahirlah konsep ASP (*Application Service Provider*) yang ditandai dengan munculnya perusahaan pengolahan *data center*. Lalu pada tahun 1995, Larry Ellison, pendiri Oracle melahirkan wacana “*Netwok Computing*” pasca penetrasi Microsoft Windows 95 dimana memiliki maksud jika PC tidak perlu dibenamkan *software* yang membuat berat kinerja dan cukup diganti sebuah terminal utama berupa server. Pada awal 2000-an, Marc Beniof, eks Vice President Oracle melansir aplikasi CRM berbentuk “*Software as a Service*” bernama Salesforce.com sebagai penanda lahirnya cloud computing. Kemudian pada tahun 2005, situs *online shopping* Amazon.com meluncurkan Amazon EC2 (*Elastic Compute Cloud*) yang diikuti Google dengan *Google App Engine*, dan IBM yang melansir *Blue Cloud Initiative* (tersedia di <https://www.slideshare.net/cahkos/sejarah-cloud-computing> pada 14 Mei 2020). Seiring dengan perkembangan zaman, teknologi berkembang sangat pesat, perubahan teknologi yang ada, bisa berubah dalam hitungan detik. Akan tetapi, semua perkembangan itu justru membuat kita mendapatkan kemudahan dan kenyamanan dalam segala hal. Banyak hal yang dulunya tidak mungkin, sekarang bisa terwujud. Salah satu perkembangan teknologi yang ada di bidang teknologi guna menyimpan data dalam jumlah yang sangat banyak dinamakan *Cloud Computing* atau Komputasi Awan. *Cloud computing* memiliki arsitektur yang dapat dibedakan dalam 4 lapisan, yaitu *hardware* dimana lapisan ini bertanggung jawab dalam pengelolaan sumber daya fisik dari *cloud*, infrastruktur yang dimana bertanggung jawab mengenai pembentukan media yang terdiri dari media penyimpanan (*storage*) dan sumber daya komputasi, platform yang terdiri dari sistem operasi dan kerangka aplikasi, dan aplikasi dimana terdiri dari aplikasi aktual dari *cloud* dimana jenis layanannya secara langsung dapat dikirimkan atau digunakan oleh pengguna (tersedia di <http://rebornofdisciple.blogspot.com/2014/11/cloud-computing.html> pada 14 Mei 2020). Pengembangan teknologi komputasi berbasis internet saat ini dibuat untuk proses sistem yang mudah, dan tidak memerlukan banyak waktu dan tenaga serta mengurangi biaya operasional.

Sekarang ini penggunaan *cloud computing* sangat luas, karena banyaknya pihak yang menginginkan sebuah data bisa diakses dimana saja tanpa sebuah media penyimpanan tertentu seperti *flashdisk*, *hardisk*, dan lainnya. Oleh karena itu, beberapa bidang telah memanfaatkan teknologi cloud computing ini seperti layanan keuangan atau transaksi (*e-payment*), layanan pendidikan (*e-learning*), layanan pemerintahan (*e-government*), layanan perdagangan atau jual-beli (*e-commerce*), dan masih banyak yang lainnya lagi. Pada penjualan online *cloud computing* digunakan untuk pengumpulan data *customer* yang nantinya akan diolah untuk mengetahui produk makanan apa saja yang laku di pasaran (Rumetna, 2018).

Proyek yang kami buat saat ini sangat dibutuhkan oleh masyarakat pada umumnya untuk membuat kemudahan dalam berbelanja. Karena sistem ini yang berjalan online, sehingga semua pengguna bisa mengaksesnya kapan saja dan dimana saja. Proyek yang kami buat terpikirkan karena banyaknya masyarakat umum yang jika pergi ke suatu pasar atau swalayan tidak mendapatkan produk yang mereka inginkan, untuk itu kami berinisiatif untuk membuat suatu program yang memudahkan masyarakat untuk melihat produk apa saja yang dijual dan harganya serta dapat membeli produk tersebut tanpa harus keluar rumah. Sehingga proses pembelian akan semakin mudah dan efisien.

Proyek ini diselesaikan dengan menggunakan dua laptop. Untuk proyek pertama menggunakan Laptop MSI GF63 9RCX – 622ID. Operating System (OS) yang digunakan adalah Windows 10 Home dengan *processor* Intel® Core™ i7-9750H. Untuk software yang digunakan yaitu Ubuntu LAMPP yang akan diinstall apache dan MySQL untuk web hostingnya. Kami juga menggunakan Virtual Machine (VMWare Pro 15.0.1). Proyek web yang digunakan adalah proyek web pada tugas semester 5, yang telah kami rancang dengan databasenya. Untuk proyek kedua menggunakan Laptop ASUS. Operating System (OS) Windows 10 Pro dengan *processor* Intel® Core™ . Untuk *software* yang digunakan yaitu Docker.

## **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 Jual Beli Seafood yang telah dibuat sebelumnya dengan menggunakan Ubuntu 18.04.3 LTS dan LAMPP (Apache 2.4.29, PHP 7.2.24, MySQL 5.7, phpMyAdmin 4.6.6).

2. Membuat Dockerfile pada Windows 10 Pro untuk mempermudah dalam instalasi Sistem Informasi Jual Beli Seafood pada host yang berbeda.

Tujuan dari proyek ini adalah sebagai bentuk implementasi hasil dari proses belajar selama praktikum agar dapat dimanfaatkan untuk kepentingan pribadi maupun bersama. Sistem Informasi Jual Beli Seafood menggunakan layanan ubuntu LAMPP dan pembuatan dockerfilenya ditujukan agar website Penjualan seafood bisa diakses dimana saja dan dapat berjalan dengan baik serta mempermudah dalam instalasi sistem pada host yang berbeda.

### 1.3 Manfaat Proyek Akhir

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

1. Sistem yang telah dirancang dapat digunakan sewaktu-waktu tanpa khawatir *downtime* dikarenakan terdapatnya *primary* dan *backup server* yang menggunakan arsitektur *recovery* pada *cloud computing*.

2. Penyedia sistem tidak perlu memperlakukan *maintenance*, dikarenakan dengan menggunakan *cloud computing*, rutinitas *maintenance* akan dilakukan sepenuhnya oleh *vendor*.

3. Batasan memori penyimpanan multimedia menjadi tidak terbatas dikarenakan sistem telah sepenuhnya beralih menggunakan *cloud computing*.

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

Dibuatnya Sistem Informasi Penjualan Seafood dikarenakan para pembeli cenderung kesulitan untuk pergi ke pasar terutama jika tidak memiliki waktu tentu saja akan sulit untuk ke pasar, belum lagi biaya perjalanan dan tidak adanya informasi stok dan harga dari seafood yang ingin dibeli sehingga menyulitkan pembeli. Dengan adanya sistem informasi seafood online, maka pengguna tidak perlu melakukan pembelian secara konvensional, hanya dengan menggunakan website ini pengguna bisa membeli seafood dimana saja dan kapan saja, serta para pengguna bisa melihat produk apa saja yang dijual beserta harganya karena semua datanya telah tersimpan di dalam *cloud* dan akan terupdate setiap waktunya sehingga memudahkan pergantian stok data jika ada data yang bertambah atau berkurang. Proses pembayaran juga menjadi salah satu faktor, dimana bisa melakukan pembayaran dengan *Cash On Delivery* (COD) ataupun via transfer, sehingga kedua belah pihak tidak disulitkan untuk ini.

#### 1.4 Tahap Penyelesaian Proyek Akhir

Tahapan secara singkat yang dilakukan untuk menyelesaikan proyek akhir ini adalah sebagai berikut:

1. Menganalisis kebutuhan dari Sistem Informasi Jual Beli Seafood untuk ditransformasikan ke dalam arsitektur cloud computing menggunakan basis SaaS.
2. Menentukan konfigurasi yang tepat untuk pengaturan Dockerfile sehingga dapat digunakan sesuai requirement yang berupa:
  - a. Dapat diakses oleh jaringan/IP tertentu saja.
  - b. Dapat di install di mana saja tanpa mempengaruhi environment sistem.
  - c. Menggunakan Windows 10 Pro 64-Bit.
  - d. Menggunakan PHP 7.4.
3. Menguji konfigurasi Docker pada host untuk mengetahui apakah konfigurasi dapat berjalan pada host lain atau hanya berjalan pada host dimana konfigurasi Docker dibuat.
4. Merancang topologi cloud computing untuk mengintegrasikan dua sub sistem yang berbeda sehingga dapat digunakan secara terintegrasi.
5. Melakukan konfigurasi Ubuntu Server sebagai primary dan backup untuk dapat digunakan sebagai recovery sehingga Sistem Informasi Jual Beli Seafood ketersediaannya/availability-nya maksimal.

Pada proses pengerjaan project ini langkah pertama adalah konfigurasi dari software dan hardware yang digunakan dimana software yang penulis gunakan adalah vmware sebagai workstation dan os yang digunakan adalah ubuntu dan docker dengan windows 10 pro menggunakan MySQL Workbench 8.0 CE sebagai database, selanjutnya setelah konfigurasi keperluan software dan hardware lanjut ke tahap instalasi Apache, MySQL, PHP dan dan kebutuhan lainnya dengan memperhatikan konfigurasi yang telah ada, kemudian masuk ke tahap hasil implementasi yang membahas bagaimana penggunaan project setelah proses instalasi, terakhir adalah pengujian singkat yang dimana setiap project di uji guna mengetahui kekurangan dan kelebihan dalam project tersebut.



## **BAB II**

### **ISI DAN PEMBAHASAN**

#### **2.1 Komponen yang Digunakan**

Komponen yang digunakan pada tugas proyek akhir ini terbagi menjadi dua sesuai dengan judul yang dituliskan sebelumnya. Untuk komponen yang pertama ialah “Sistem Informasi Jual Beli Seafood Menggunakan Layanan Ubuntu LAMPP” dan yang kedua merupakan kelanjutannya yaitu “Proses Pembuatan Dockerfilenya”. Berikut akan dibahas mengenai keduanya, masing-masing dalam sub bab terkait.

##### **2.1.1 Komponen pada Proyek Pertama**

Untuk membangun “Sistem Informasi Jual Beli Seafood Menggunakan Layanan Ubuntu LAMPP” yang berbasiskan konsep *cloud computing*, maka diperlukan analisis berbagai komponen. Berikut akan dijelaskan terlebih dahulu dalam bentuk poin-poin singkat berikut ini:

1. Proyek pertama ini menggunakan Laptop MSI GF63 9RCX – 622ID. Operating System (OS) Microsoft Windows 10 Home Single Language, tipe prosesor 64-bit, dan RAM 8GB.
2. Komponen utama penyusun dari *cloud computing*-nya yaitu VMWare Workstation 15.0.1.
3. Menggunakan emulator terminal PuTTY yang *open source* dengan versi 0.73 dan *build platform* 32-bit x86 Windows untuk mengaktifkan protocol jaringan.
4. Menggunakan WinSCP versi 5.17.3 sebagai klien SFTP, FTP, WebDAV, Amazon S3, dan SCP dimana SSH dan SCP berdasarkan PuTTY 0.73 dimana digunakan untuk transfer file atau bisa dikatakan untuk upload file yang akan digunakan sebagai website pada Tugas 1 ini.
5. Untuk Ubuntu OS yang digunakan yaitu Ubuntu 18.04.3 LTS yang diinstal Apache, PHP, MySQL, dan phpMyAdmin.
6. Sistem yang telah dibangun menggunakan Bahasa pemrograman PHP dengan versi 7.2.24 dan bahasa HMTL5.
7. Penyimpanan basis data dengan arsitektur penyimpanan MySQL versi 5.7 sehingga dapat digunakan untuk menyimpan data penjualan seafood yang dibutuhkan oleh sistem tersebut.

8. Target pengguna adalah orang-orang yang ingin membeli seafood tanpa harus pergi ketempat penjualannya secara langsung sehingga dapat melalui website ini.

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	Menggunakan aplikasi <i>virtual machine</i> .
2.	Prosesor	2 core	Prosesor dari <i>hypervisor</i> yang dialokasikan ke <i>guest</i> .
3.	Konfigurasi Jaringan Guest OS	Mode NAT	Mode adapter jaringan VM <i>guest</i> yang digunakan.
		IP : 192.168.111.133	Alamat IP yang digunakan oleh <i>guest OS</i> .
		DNS : 192.168.111.1	Alamat IP untuk DNS <i>guest OS</i> .
		GW : 192.168.111.1	Alamat untuk <i>gateway</i> atau gerbang menuju akses jaringan luar.
4.	Versi Ubuntu	Ubuntu 18.04.3 LTS	ISO Ubuntu yang digunakan untuk <i>guest OS</i> .
5.	RAM	2GB	Alokasi RAM untuk <i>guest OS</i> .
6.	Disk info	20GB	Alokasi storage untuk <i>guest OS</i> .

Untuk spesifikasi 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.	LAMP	Apache 2.4.29	Preprosesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.2.24	Bahasa pemrograman yang dapat disisipkan ke HTML.
		MySQL 5.7	DBMS untuk manajemen basis data SQL.
		phpMyAdmin 4.6.6	<i>Software</i> yang digunakan untuk mengelola server MySQL.
2.	Python	Python 3.6.9	Dukungan Bahasa pemrograman yang digunakan.

### 2.1.2 Komponen pada Proyek Kedua

Untuk membangun Sistem Informasi Jual Beli Seafood dengan *dockerfile* yang berbasiskan *cloud computing*, maka diperlukan analisis berbagai komponen yang akan dijelaskan pada poin-poin berikut:

1. Proyek pertama ini menggunakan Laptop ASUS X450LCP. Operating System (OS) Microsoft Windows 10 Pro 64-bit, tipe prosesor Intel® Core™ i5-4200U CPU @ 1.60GHz (4 CPUs), ~2.3GHz, dan RAM 8GB.
2. Penyimpanan basis data dengan arsitektur penyimpanan MySQL Workbench 8.0 CE sehingga dapat digunakan untuk menyimpan data penjualan seafood yang dibutuhkan oleh sistem tersebut.
3. Menggunakan Docker versi 2.2.0.5.
4. Untuk menjalankan perintah digunakan *file* yang bernama *dockerfile* yang berisi perintah untuk menjalankan ekstensi MySQL yaitu *mysqli* yang berguna untuk koneksi ke dalam database.
5. Dengan beberapa *images* yang akan dijalankan melalui *container*, maka akan lebih mudah untuk menjalankannya secara stack. Untuk menjalankan stack tersebut digunakan *docker-compose*.
6. Dalam pembuatan Dockerfile diperlukan Docker Engine dengan versi 19.03.8 dan Docker Compose dengan versi 1.25.4.

Untuk spesifikasi docker *desktop* dapat dilihat pada **Tabel 2.3** sebagai berikut:




**Tabel 2.3** Spesifikasi docker *desktop* untuk proyek kedua

No	Nama Parameter	Nilai	Keterangan
1.	Docker Image	Apache	Preproesor bahasa pemrograman HTML, termasuk CSS dan JS.
		PHP 7.4	Bahasa pemrograman yang dapat disisipkan ke HTML.
		MySQL	DBMS untuk manajemen basis data SQL.
		MySQL Workbench 8.0 CE	<i>Software</i> yang digunakan untuk mengelola server MySQL.
2.	Docker Desktop	Versi 2.2.0.5	Aplikasi yang bersifat <i>open source</i> yang berfungsi sebagai wadah/ <i>container</i> untuk mengepak/memasukkan sebuah <i>software</i> secara lengkap beserta semua hal lainnya yang dibutuhkan.
3	Docker Engine	Docker Client 19.03.8	<i>Docker engine client</i> yang digunakan untuk berkomunikasi dengan <i>docker server</i> .
		Docker Server 19.03.8	<i>Docker engine server</i> yang digunakan dalam menjalankan <i>container</i> .
4	Docker Compose	Docker Compose 1.25.4	Dukungan alat untuk mendefinisikan dan menjalankan <i>multi-container</i> .
5	Dockerfile	Dockerfile	File untuk menjalankan perintah seperti contohnya menjalankan ekstensi <i>mysqli</i> .

## 2.2 Rancangan Arsitektur *Cloud Computing*

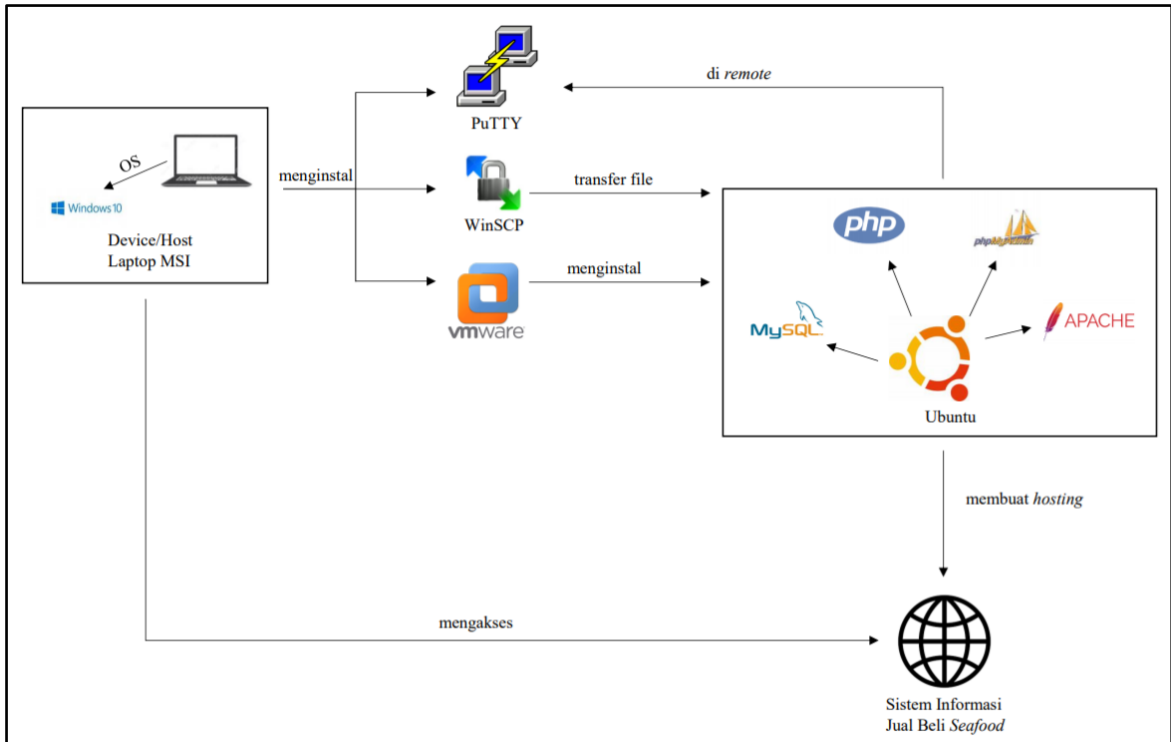
Pada proyek ini menggunakan jenis *cloud computing* SaaS dimana pada layer pertama terdapat webservice selanjutnya pada layer kedua berisi aplikasi yang digunakan seperti apache sebagai server, docker digunakan untuk run *dockerfile*, vmware sebagai computing proses, MySQL untuk database dan php sebagai bahasa yang digunakan dalam membuat layanan webservices, pada layer insfrastruktur terdapat laptop atau komputer dengan akses local yang digunakan untuk menjalankan proyek yang dibuat. Ilustrasi mengenai rancangan arsitektur tersebut dapat dilihat pada **Tabel 2.4** berikut ini :

**Tabel 2.4** Ilustrasi Rancangan Arsitektur

Model Service	Resource Manager	Layer	Example
Software as a Service (SaaS)	Web Application	Application	
Platform as a Service (PaaS)	Software Framework	Platforms	
Infrastructure as a Service (IaaS)	CPU, Memory, Disk, Processor	Hardware	

### 2.2.1 Rancangan Arsitektur *Cloud Computing* Proyek Pertama

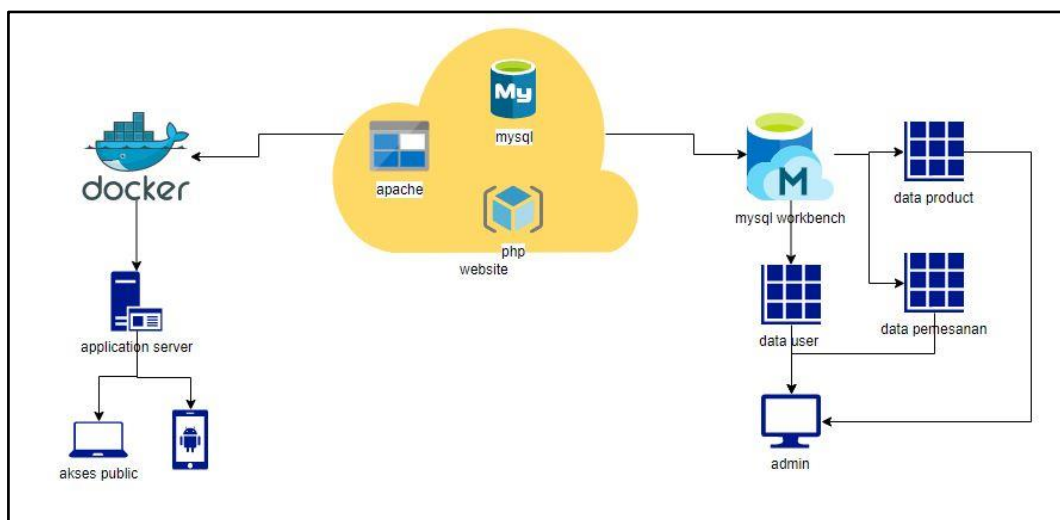
Pada proyek pertama, komputer yang digunakan menggunakan sistem operasi Windows 10 Home yang kemudian diinstal VMWare Workstation untuk membuat Virtual Machine-nya, PuTTY untuk melakukan remote server Ubuntu, dan WinSCP yang digunakan untuk transfer file Sistem Informasi Jual Beli *Seafood* yang akan di-hosting. Ubuntu yang digunakan terinstal Apache, MySQL, PHP, dan phpMyAdmin. Setelah semua terkonfigurasi maka untuk dapat mengakses website yang sudah di-hosting tadi, komputer yang digunakan harus terkoneksi dengan jaringan dan untuk topologi ilustrasi proyek pertama dapat dilihat pada **Gambar 2.1** berikut:



**Gambar 2.1** Rancangan arsitektur proyek pertama

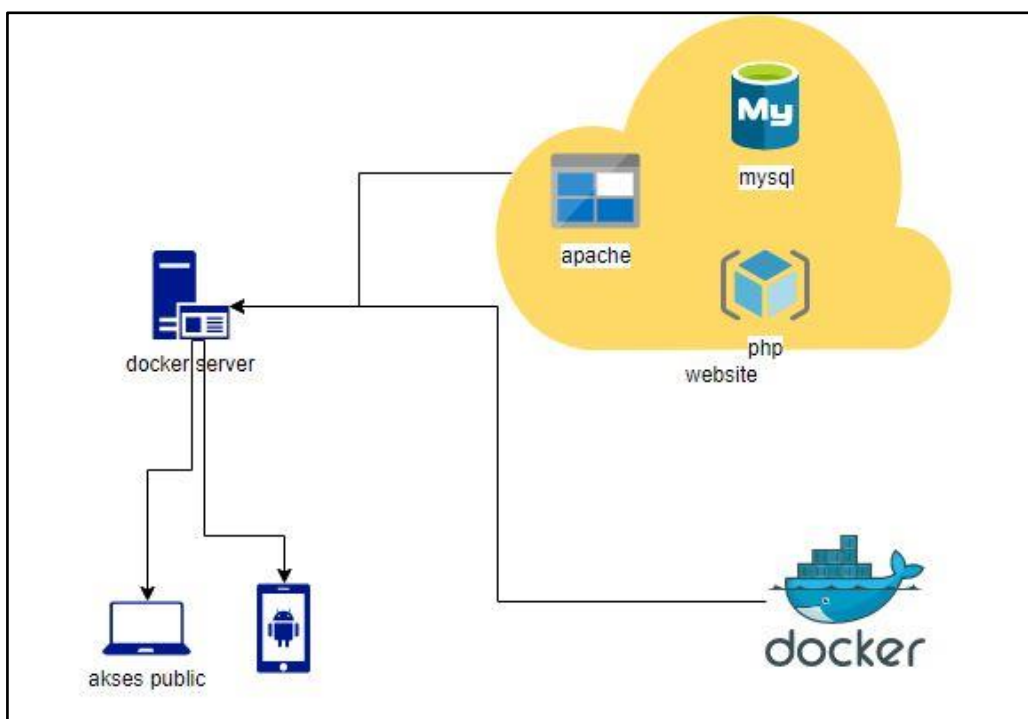
### 2.2.2 Rancangan Arsitektur *Cloud Computing* Proyek Kedua

Pada pembuatan proyek kedua ini rancangan topologi yang dibuat ada topologi jaringan dan topologi website sistem informasi jual beli seafood yang akan dijelaskan seperti pada **Gambar 2.2** dan **Gambar 2.3** berikut ini :



**Gambar 2.2** Topologi website Sistem Informasi Jual Beli Seafood menggunakan *docker*

Pada topologi sistem informasi jual beli seafood, user mengakses dengan menggunakan *application server* yang kita buat sebelumnya sesuai dengan **Gambar 2.20** setelah itu *application server* akan membuka website yang ada pada docker, kemudian docker akan mengarahkan data yang ada pada MySQL Workbench yang merupakan server dari database yang telah dibuat sesuai dengan **Gambar 2.24 (import database)** kemudian admin bisa mengakses data tersebut melalui docker yang nantinya akan diarahkan pada MySQL Workbench sehingga admin bisa melakukan proses CRUD pada data.



**Gambar 2.3** Topologi Jaringan pada *docker*

Pada topologi jaringan menggambarkan bagaimana interkoneksi antara device dengan server docker yang digunakan, pertama user masuk melalui jaringan ke docker server setelah itu docker server akan mengakses docker untuk membangun container untuk website setelah itu website di build dan user bisa mengakses website dengan jaringan yang digunakan.

## 2.3 Parameter dan Konfigurasi

Dalam pembuatan proyek akhir cloud computing dengan judul Sistem Informasi Jual Beli Seafood Menggunakan Layanan Ubuntu LAMPP dan Proses Pembuatan Dockerfilenya menggunakan parameter dan konfigurasi yang akan dibahas masing-masing pada sub bab berikut.

### 2.3.1 Parameter dan Konfigurasi Proyek Pertama

Pada pembuatan proyek pertama ini yang perlu dilakukan pertama adalah membuat *Virtual Machine* untuk *OS* Ubuntu. Konfigurasi untuk pembuatan *Virtual Machine* akan dijelaskan seperti pada **Tabel 2.5** berikut ini:

**Tabel 2.5** Konfigurasi pada pembuatan Virtual Machine

No	Nama Parameter	Nilai	Keterangan
1.	Merk Server	Virtual Machine VMWare Workstation	Aplikasi yang digunakan untuk virtualisasi.
2.	Versi	15.0.1	Versi VMWare yang digunakan.
3.	Versi Ubuntu	Ubuntu 18.04.3 LTS	ISO Ubuntu yang digunakan.
4.	Konfigurasi prosesor	2	Jumlah dari prosesor yang digunakan.
5.	RAM	1GB	Alokasi RAM yang digunakan untuk <i>Virtual Machine</i> .
6.	<i>Network Type</i>	NAT (Network Address Translation)	Tipe jaringan yang digunakan untuk virtualisasi.
7.	<i>Disk Capacity</i>	20GB	Alokasi disk yang digunakan untuk virtualisasi.

Melakukan pengecekan IP yang akan digunakan pada tahapan selanjutnya. Parameter yang digunakan untuk cek IP dapat dilihat pada penjelasan **Modul 2.1** berikut ini:

```
$ ifconfig
```

Keterangan : perintah yang dilakukan untuk menampilkan IP, netmask, broadcast, dan lain-lain.

#### **Modul 2.1** Parameter untuk cek IP dan lainnya

Selanjutnya setelah Ubuntu berhasil diinstal, kemudian melakukan instalasi PuTTY yang nantinya digunakan untuk *remote access* SSH. Untuk konfigurasi pada PuTTY akan dijelaskan pada **Tabel 2.6** berikut ini:

**Tabel 2.6** Konfigurasi pada PuTTY

No	Nama Parameter	Nilai	Keterangan
1.	PuTTY <i>version</i>	Versi 0.73	Versi dari PuTTY yang digunakan.
2.	<i>Host Name/IP Address</i>	IP : 192.168.111.133	IP yang digunakan untuk akses remote Ubuntu melalui PuTTY.
3.	<i>Port</i>	22	<i>Port</i> SSH aktif yang defaultnya di 22.
4.	<i>Connection Type</i>	SSH	Tipe koneksi yang digunakan.
5.	<i>Login as</i>	sabila	<i>User</i> admin yang akan di <i>remote</i> .

Untuk selanjutnya yaitu melakukan konfigurasi untuk instalasi Apache2 yang akan dijelaskan pada **Tabel 2.7** berikut:

**Tabel 2.7** Konfigurasi pada Apache2 yang digunakan

No	Nama Parameter	Nilai	Keterangan
1.	Apache2 <i>version</i>	2.4.29	Versi dari Apache2 yang digunakan.
2.	<i>Server Built</i>		

Parameter yang digunakan untuk instalasi Apache2 dapat dilihat pada penjelasan **Modul 2.2** berikut:

```
$ sudo apt update
```

```
$ sudo apt install apache2
```

Keterangan :

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

```
$ sudo ufw allow in "Apache Full"
```

Keterangan : perintah untuk menkonfigurasi Firewall untuk mengizinkan layanan Web server Apache dengan tipe protocol HTTP (port 80) dan HTTPS (port 443)

### **Modul 2.2** Parameter untuk install Apache2

Selanjutnya setelah Apache2 yang harus di konfigurasi yaitu instalasi MySQL yang akan dijelaskan pada **Tabel 2.8** berikut ini:



**Tabel 2.8** Konfigurasi pada MySQL yang digunakan

No	Nama Parameter	Nilai	Keterangan
1.	MySQL <i>version</i>	5.7	Versi dari MySQL yang digunakan.
2.	<i>Server</i>	localhost via UNIX socket	Keterangan server yang digunakan.
3.	<i>Server Version</i>	5.7.30-0ubuntu0.18.04.1 - (Ubuntu)	Versi server yang digunakan.
4.	Versi Protokol	10	Keterangan versi dari protocol yang digunakan.
5.	<i>Server charset</i>	UTF-8 Unicode (utf8)	Keterangan server charset yang digunakan.
6.	<i>Database</i>	projectpw	Nama <i>database</i> yang digunakan untuk proyek pertama ini.

Kemudian, untuk parameter yang digunakan dalam instalasi MySQL akan dijelaskan pada **Modul 2.3** berikut:

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

Keterangan :

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

```
$ sudo mysql_secure_installation
```

Keterangan : perintah untuk melakukan pengaturan keamanan pada MySQL contohnya username dan password

### Modul 2.3 Parameter untuk install MySQL

Setelah melakukan instalasi MySQL maka selanjutnya melakukan instalasi PHP yang konfigurasinya akan dijelaskan pada **Tabel 2.9** berikut ini:

**Tabel 2.9** Konfigurasi pada PHP yang digunakan

No	Nama Parameter	Nilai	Keterangan
1.	PHP <i>version</i>	Versi 7.2.24-0ubuntu0.18.04.3	Versi dari PHP yang digunakan.
2.	<i>System</i>	Linux sabila 4.15.0-91-generic #92-Ubuntu SMP Fri Feb 28 11:09:48 UTC 2020 x86_64	Keterangan <i>system</i> yang digunakan.
3.	<i>Build Date</i>	Feb 11 2020 15:55:52	Tanggal dan jam PHP dibuat.
4.	PHP API	20170718	Keterangan PHP API yang digunakan.
5.	PHP <i>Extension</i>	20170718	Keterangan PHP <i>Extension</i> yang digunakan.

Kemudian, untuk parameter yang digunakan untuk instalasi PHP dapat dilihat pada penjabaran **Modul 2.4** berikut ini:

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

Keterangan :

- sudo : perintah untuk eksekusi suatu command dengan hak akses tertinggi (root)
- apt : merupakan package manager pada Ubuntu
- install : parameter tambahan pada apt untuk mengeksekusi perintah instalasi paket aplikasi
- php libapache2-mod-php php-mysql : nama paket aplikasi untuk PHP

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

Keterangan : untuk membuat sebuah file dengan nama info dengan ekstensi php yang disimpan di directory var/www/html

#### **Modul 2.4** Parameter untuk install PHP

Setelah melakukan instalasi PHP maka melakukan konfigurasi pada instalasi phpMyAdmin yang akan dijelaskan pada **Tabel 2.10** berikut ini:

**Tabel 2.10** Konfigurasi pada phpMyAdmin yang digunakan

No	Nama Parameter	Nilai	Keterangan
1.	phpMyAdmin <i>version</i>	Versi 4.6.6	Versi dari PphpMyAdmin yang digunakan.
2.	<i>Username</i>	root	<i>Username</i> yang digunakan untuk login ke phpMyAdmin.
3.	Alamat	192.168.111.133/phpmyadmin	Alamat yang digunakan untuk mengakses phpMyAdmin.

Kemudian, parameter yang digunakan untuk melakukan instalasi phpMyAdmin dapat dilihat pada **Modul 2.5** berikut ini:

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

Keterangan :

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

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

Keterangan : perintah yang digunakan untuk masuk ke MySQL sebagai user root kemudian nantinya untuk melakukan konfigurasi

```
mysql> UPDATE mysql.user SET plugin = 'mysql_native_password',
authentication_string = PASSWORD('salsabila28') WHERE User = 'root';
```

Keterangan : untuk mengubah parameter plugin dari user root dan untuk mengatur password yang telah dibuat pada akun root di MySQL

```
mysql> FLUSH PRIVILEGES;
```

Keterangan : untuk merefresh settingan yang telah dibuat

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

Keterangan : untuk memberikan akses ke user sabila agar dapat mengakses direktori /var/www/html

### **Modul 2.5** Parameter untuk install phpMyAdmin

#### **2.3.2 Parameter dan Konfigurasi Proyek Kedua**

Pada pembuatan proyek kedua ini yang perlu dilakukan pertama adalah install docker *desktop* versi 2.2.0.5 pada Windows 10 Pro. Konfigurasi *docker desktop* akan dijelaskan seperti pada **Tabel 2.11** berikut ini:

**Tabel 2.11** Konfigurasi pada *Docker Desktop*

No	Nama Parameter	Nilai	Keterangan
1.	Merk Server	Docker Desktop	Aplikasi yang digunakan untuk virtualisasi.
2.	Versi	2.2.0.5	Versi <i>Docker Desktop</i> yang digunakan.
3.	Konfigurasi prosesor	2	Jumlah dari prosesor yang digunakan.
4.	RAM	2GB	Alokasi RAM yang digunakan untuk <i>Docker Desktop</i> .
5.	<i>Swap</i>	1GB	Konfigurasi ukuran file swap sesuai kebutuhan.
6.	<i>Disk Image Size</i>	64GB	Alokasi disk yang digunakan untuk virtualisasi.
7.	<i>Disk Image Location</i>	C:\ProgramData\DockerDesktop\vm-data	Lokasi volume Linux tempat penyimpanan gambar disimpan.

Setelah melakukan konfigurasi pada *docker desktop* maka melakukan konfigurasi pada *MySQL connection* yang akan dijelaskan pada **Tabel 2.12** berikut ini:

**Tabel 2.12** Konfigurasi pada MySQL yang digunakan

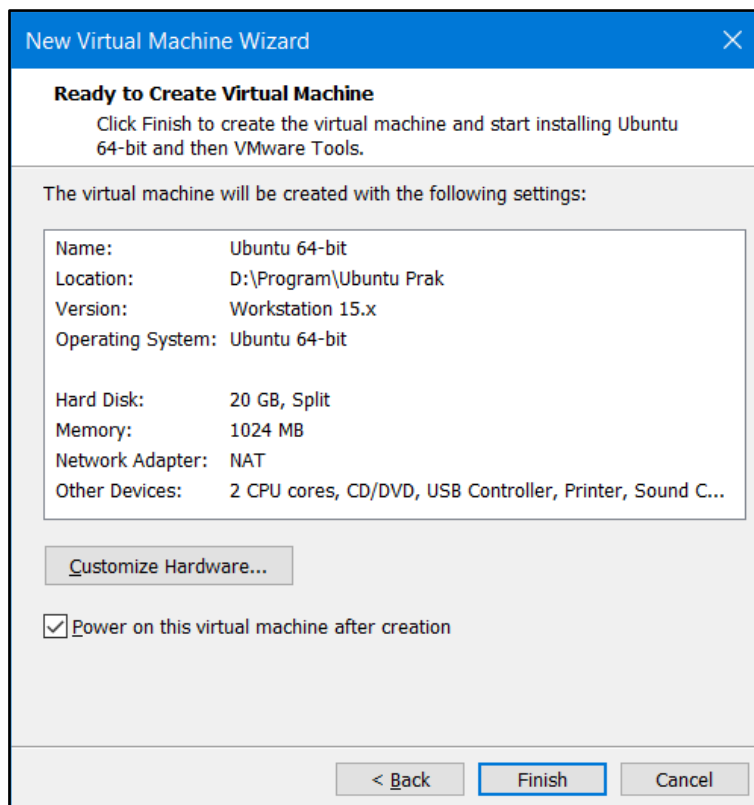
No	Nama Parameter	Nilai	Keterangan
1.	MySQL Workbench <i>version</i>	Versi 8.0.17	Versi dari MySQL yang digunakan.
2.	<i>Username</i>	root	<i>Username</i> yang digunakan untuk login ke MySQL.
3.	<i>Password</i>	12345	<i>Password</i> yang digunakan untuk login ke MySQL.
4.	<i>MySQL Connection</i>	Docker db	Nama koneksi yang dibuat pada MySQL.
5.	<i>Database</i>	docker_database	Nama <i>database</i> yang digunakan untuk proyek kedua.

## 2.4 Tahap Implementasi

Terdapat beberapa tahapan yang perlu dilakukan dalam penerapan hosting pada proyek Sistem Informasi Jual Beli Seafood dan Proses Pembuatan Dockerfilenya. Berikut akan dibahas pada sub bab terkait.

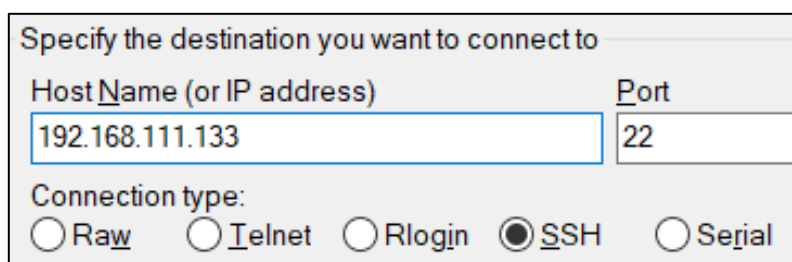
### 2.4.1 Tahap Implementasi Proyek Pertama

Pada proyek pertama, tahapan yang perlu dilakukan dalam penerapan hosting dengan Ubuntu LAMPP yaitu membuat virtual machine terlebih dahulu. Tahapan awal pada *wizard* pembuatan VM pada VMware Workstation digunakan opsi konfigurasi *Custom* seperti pada **Gambar 2.4** berikut ini:



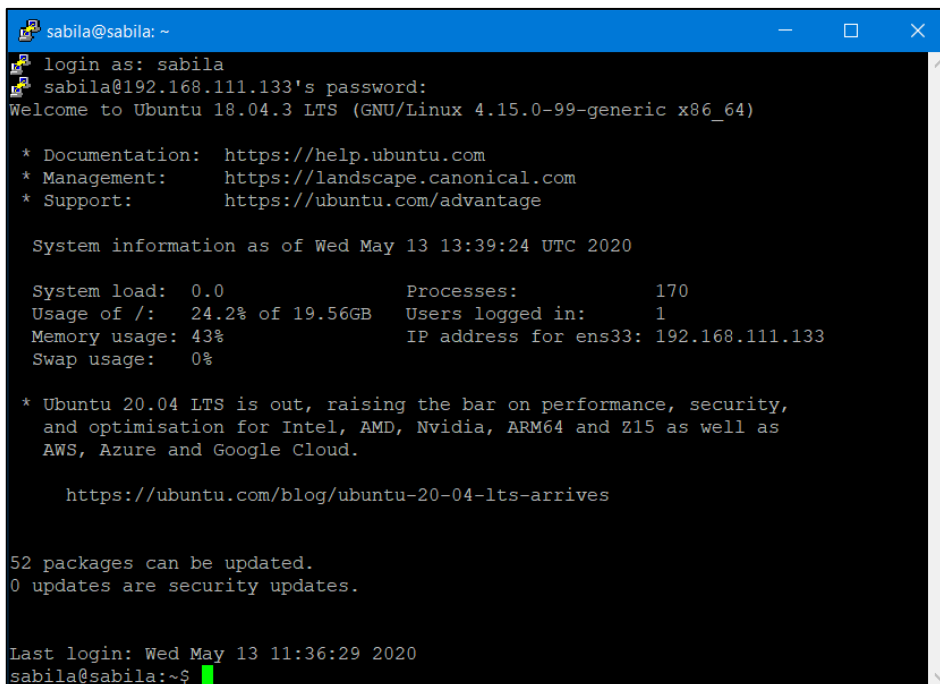
**Gambar 2.4** Tampilan opsi pemilihan *mode wizard* pembuatan VM

Selanjutnya, setelah VM selesai dibuat kemudian melakukan login pada server yang sudah dibuat tadi. Setelah melakukan login kemudian lakukan cek IP untuk nantinya digunakan pada langkah selanjutnya. Setelah IP server Ubuntu didapatkan, kemudian melakukan remote server dengan menggunakan PuTTY. Buka aplikasi PuTTY kemudian masukkan IP tadi pada kolom *Host Name* seperti pada **Gambar 2.5** berikut:



**Gambar 2.5** Memasukkan IP pada PuTTY

Setelah itu lakukan login pada command PuTTY dengan username dan password dan setelah login maka tampilan PuTTY dapat dilihat pada **Gambar 2.6** berikut ini:



```
sabila@sabila: ~
login as: sabila
sabila@192.168.111.133's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-99-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed May 13 13:39:24 UTC 2020

System load:  0.0                       Processes:    170
Usage of /:   24.2% of 19.56GB          Users logged in:  1
Memory usage: 43%                       IP address for ens33: 192.168.111.133
Swap usage:  0%

 * Ubuntu 20.04 LTS is out, raising the bar on performance, security,
   and optimisation for Intel, AMD, Nvidia, ARM64 and Z15 as well as
   AWS, Azure and Google Cloud.

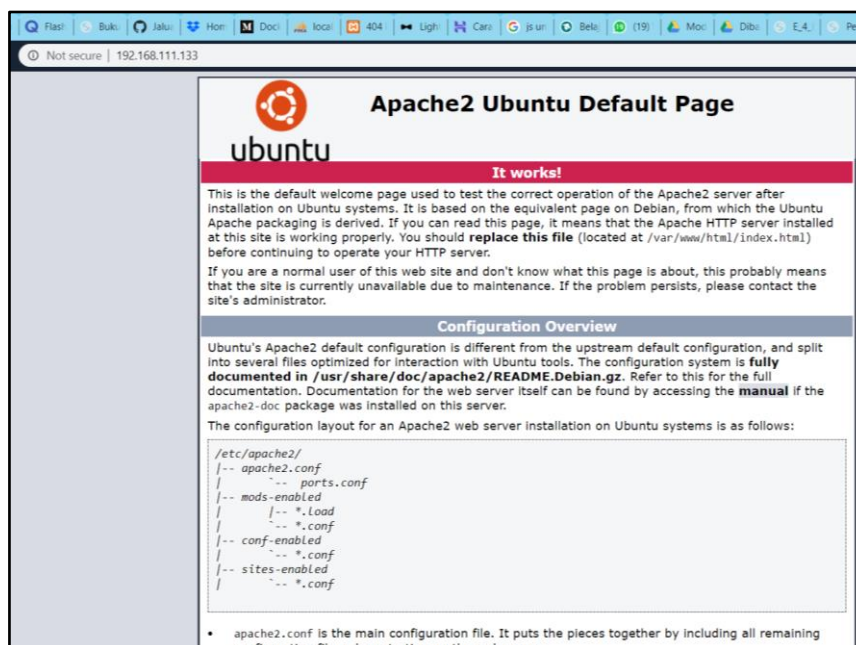
https://ubuntu.com/blog/ubuntu-20-04-lts-arrives

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

Last login: Wed May 13 11:36:29 2020
sabila@sabila:~$
```

**Gambar 2.6** Tampilan awal PuTTY setelah masuk ke server

Tahap selanjutnya yaitu melakukan instalasi Apache, MySQL, PHP, dan phpMyAdmin. Setelah semua terinstal maka akan menampilkan hasil instalasi seperti pada **Gambar 2.7**, **Gambar 2.8**, **Gambar 2.9**, dan **Gambar 2.10** berikut ini:



**Gambar 2.7** Tampilan Apache yang berhasil diinstal

```

sabila@sabila: ~
By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

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.

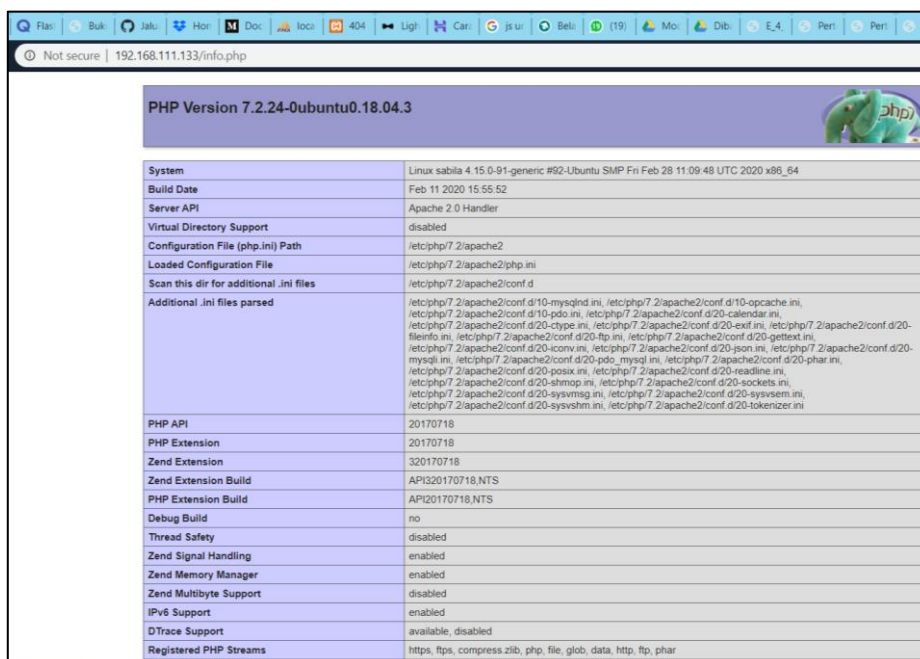
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

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

All done!
sabila@sabila:~$

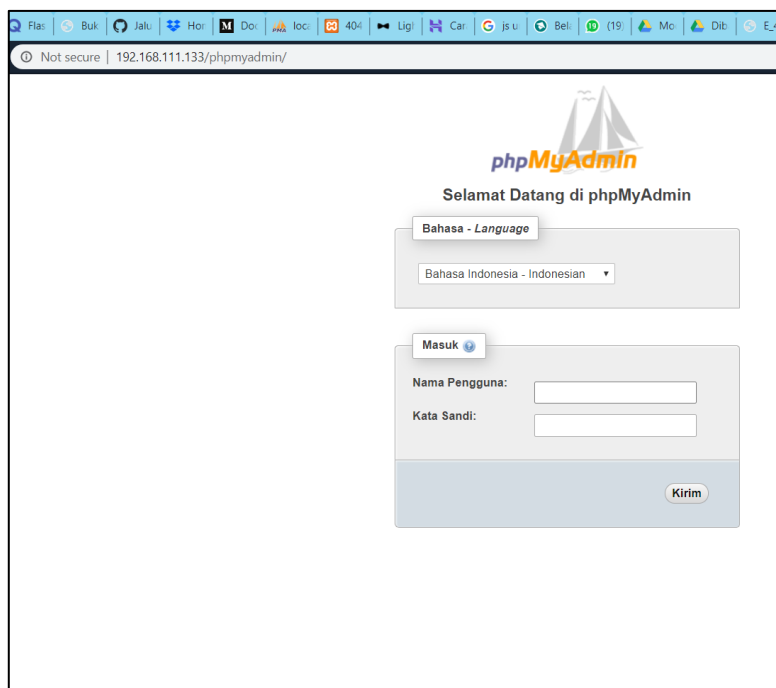
```

**Gambar 2.8** Tampilan MySQL yang berhasil diinstal



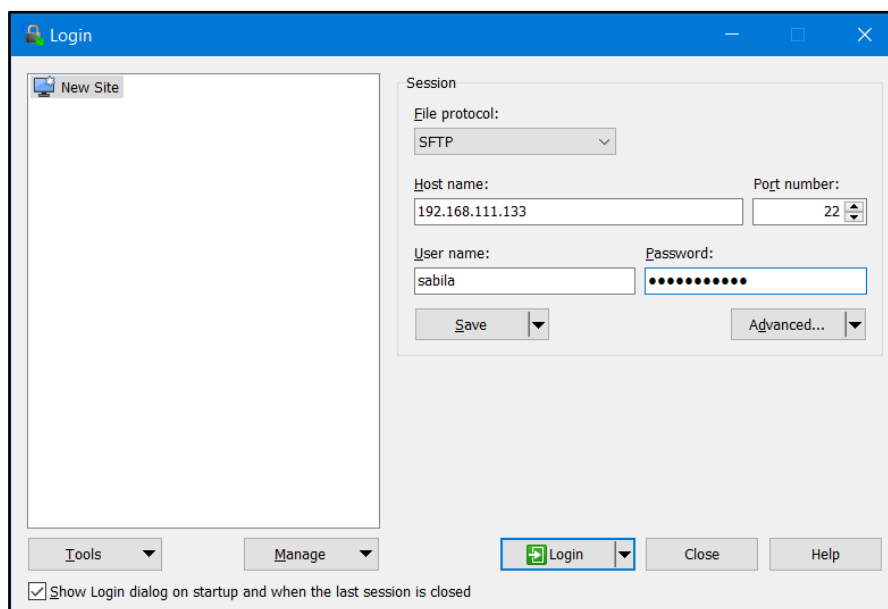
System	Linux sabila 4.15.0-91-generic #92-Ubuntu SMP Fri Feb 28 11:09:48 UTC 2020 x86_64
Build Date	Feb 11 2020 15:55:52
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqld.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-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-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-syssem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar

**Gambar 2.9** Tampilan PHP yang berhasil diinstal



**Gambar 2.10** Tampilan phpMyAdmin yang berhasil diinstall

Setelah komponen-komponen tadi berhasil diinstall, selanjutnya memindahkan file proyek yang akan di hosting ke dalam server dengan menggunakan WinSCP. Setelah WinSCP dibuka maka pertama kali akan diminta untuk memasukkan IP, username server Ubuntu, dan password server Ubuntu seperti pada **Gambar 2.11** berikut ini:

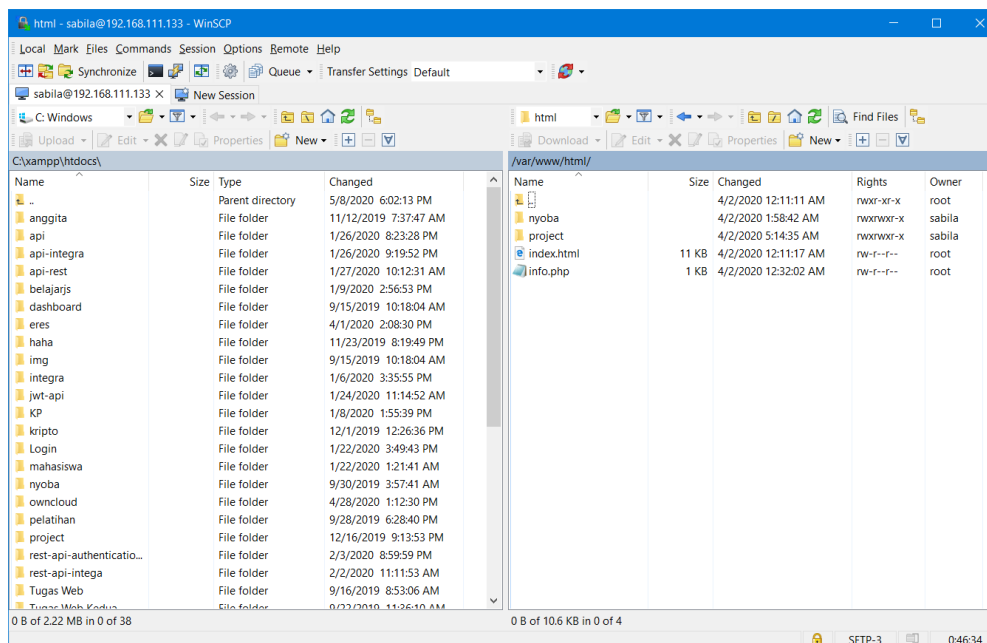


**Gambar 2.11** Login pada WinSCP

Setelah berhasil masuk, kemudian terdapat dua bagian yaitu bagian kiri dan kanan. Bagian yang kiri merupakan directory pada laptop dan untuk bagian kanan merupakan

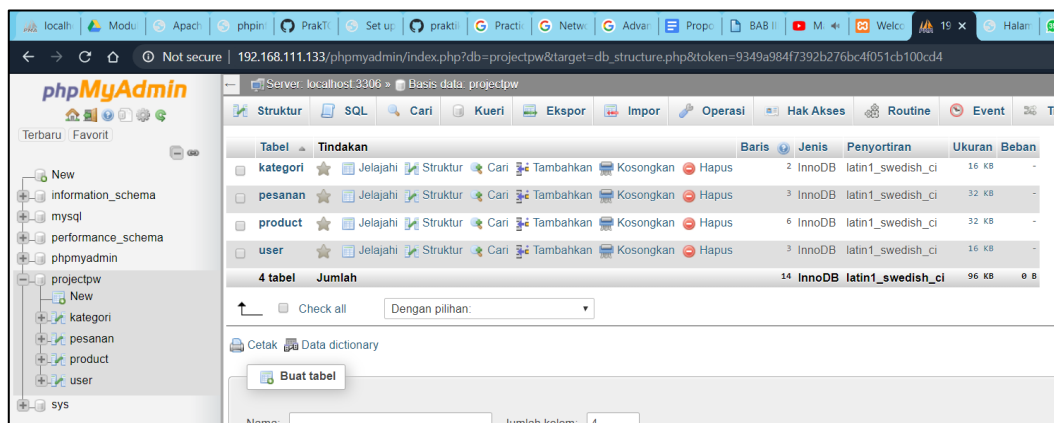


directory pada server. Untuk memindahkan file proyek yang akan di hosting dengan memindahkan dari kiri ke kanan namun sebelum memindahkan file proyek, arahkan directory pada server ke directory /var/www/html/project seperti pada **Gambar 2.12** berikut ini



**Gambar 2.12** Hasil pemindahan file proyek pada WinSCP

Setelah file berhasil dipindahkan yang perlu dilakukan selanjutnya yaitu melakukan import database melalui phpMyAdmin server. Cara melakukan import yaitu dengan menentukan database mana yang akan digunakan untuk file proyek tersebut lalu pada phpMyAdmin buat sebuah database dengan nama projectpw kemudian masuk ke fitur import dan masukkan file projectpw.sql dimana file sql tersebut merupakan database dari proyek yang digunakan. Hasil dari import database dapat dilihat dari **Gambar 2.13** berikut ini:

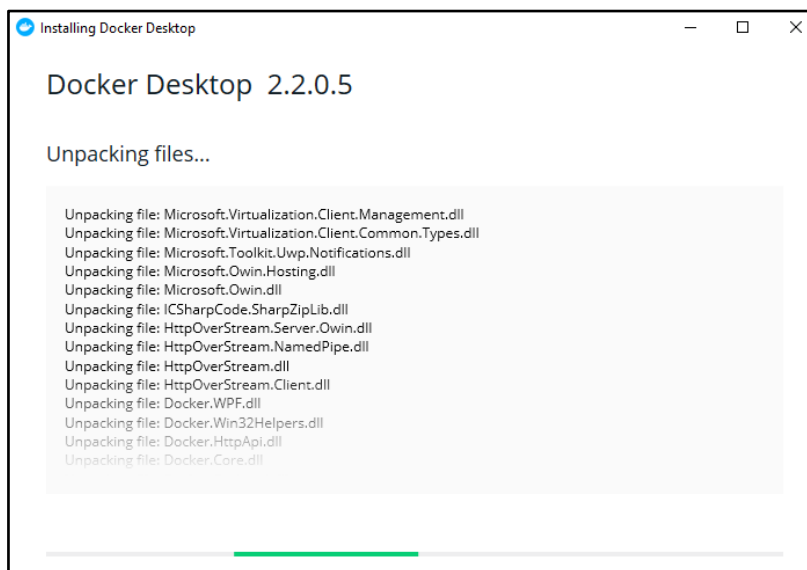


**Gambar 2.13** Tampilan database setelah di-import

Setelah langkah terakhir selesai dilakukan kemudian mencoba untuk mengakses proyek Sistem Informasi Jual Beli Seafood dengan mengakses IP yang didapatkan dari VM tadi kemudian masuk ke folder dimana nama proyek tadi ada yaitu folder project.

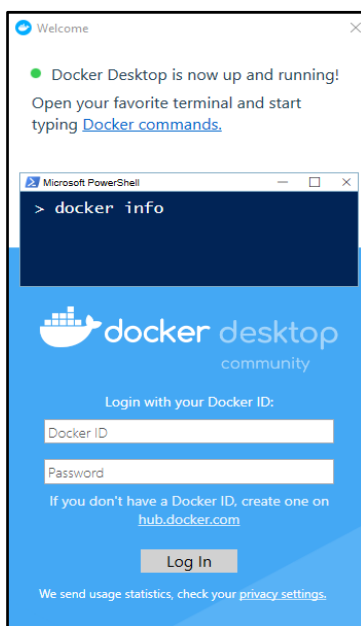
#### 2.4.2 Tahap Implementasi Proyek Kedua

Pada proyek kedua, tahapan yang pertama yang dilakukan dalam membuat *dockerfile* pada OS Windows 10 Pro adalah menginstall *docker desktop*. Tahapan awal install *docker desktop* seperti pada **Gambar 2.14** berikut ini



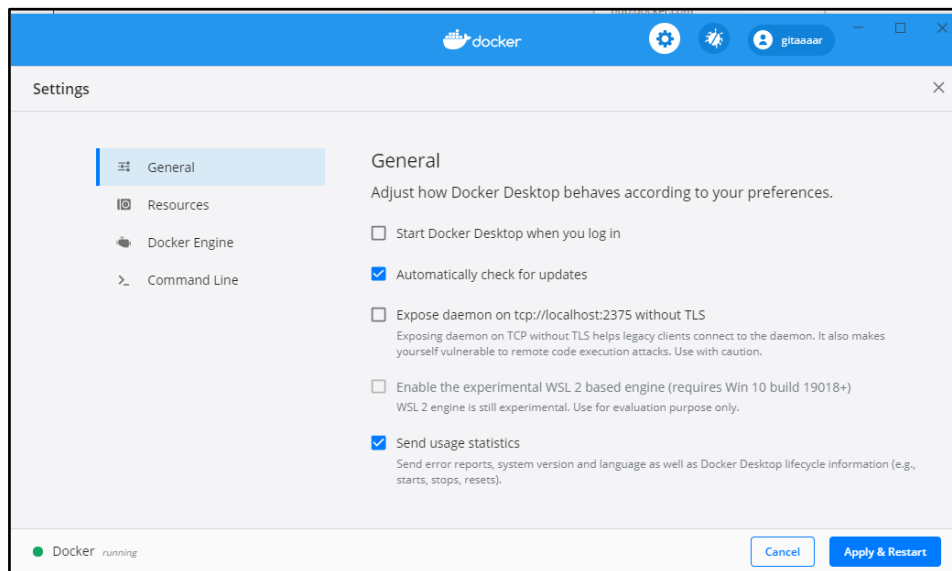
**Gambar 2.14** Proses instalasi Docker Dekstop

Selanjutnya, setelah docker desktop terinstall dengan baik, lalu login dengan tampilan awal *docker desktop* seperti yang ditunjukkan **Gambar 2.15** seperti gambar berikut :



**Gambar 2.15** Tampilan awal login pada Docker Desktop

Setelah login, buka settings *docker desktop*, dan kemudian lihat *docker* sudah *running*, akan tetapi belum ada *container* yang berjalan disini (karena belum dibuat) yang bisa dilihat pada **Gambar 2.16** seperti gambar berikut :



**Gambar 2.16** Tampilan settings pada *Docker Desktop*

Langkah selanjutnya yaitu, cek *docker version* dan *docker compose* nya untuk memastikan jika *docker* telah terinstall dengan baik seperti yang ditunjukkan oleh **Gambar 2.17** seperti berikut :

```

Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Anggita>docker version
Client: Docker Engine - Community
 Version:           19.03.8
 API version:       1.40
 Go version:        go1.12.17
 Git commit:        afacb8b
 Built:             Wed Mar 11 01:23:10 2020
 OS/Arch:           windows/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.8
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.12.17
  Git commit:       afacb8b
  Built:            Wed Mar 11 01:29:16 2020
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          v1.2.13
  GitCommit:        7ad184331fa3e55e52b890ea95e65ba581ae3429
 runc:
  Version:          1.0.0-rc10
  GitCommit:        dc9208a3303feef5b3839f4323d9beb36df0a9dd
 docker-init:
  Version:          0.18.0
  GitCommit:        fec3683

C:\Users\Anggita>docker-compose version
docker-compose version 1.25.4, build 8d51620a
docker-py version: 4.1.0
CPython version: 3.7.4
OpenSSL version: OpenSSL 1.1.1c 28 May 2019
  
```

**Gambar 2.17** Cek versi docker pada *Command Prompt*

Kemudian download image php dan apache yang ada di docker-hub dengan menggunakan syntax yang ada pada **Gambar 2.18** seperti gambar berikut :

```
C:\Users\Anggita>docker pull php:7.4-apache
7.4-apache: Pulling from library/php
afb6ec6fdc1c: Downloading
3d895574014b: Download complete
c309fdad6410: Downloading
c201f6a5d6f9: Download complete
e87f853892aa: Downloading
998b2113b400: Download complete
b3c0b4710d3b: Download complete
031d21f0c70d: Downloading [=====] 10.62MB/10.62MB
d06357e06110: Download complete
8f23fed45284: Downloading
f9d08c7d9248: Download complete
18c5c4844124: Download complete
14efef405562: Download complete
```

**Gambar 2.18** Proses *pull image php dan apache* from docker pada Command Prompt

Adapun perintah program untuk menghapus image dapat ditunjukkan seperti

**Perintah Program 2.1** seperti gambar berikut :

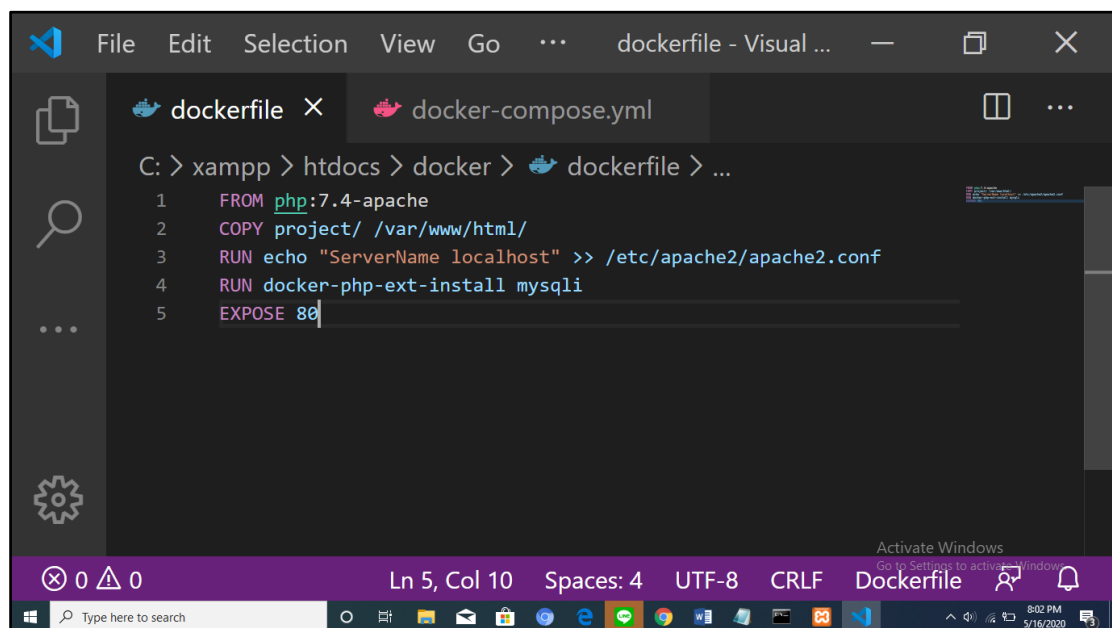
```
docker rmi [image id]
```

Keterangan :

- rmi : command untuk menghapus image.

**Perintah Program 2.1** Parameter untuk menghapus image

Kemudian buatlah dockerfile untuk project yang akan dibuat dengan syntax yang ditunjukkan pada **Gambar 2.19** seperti gambar berikut ini :



**Gambar 2.19** Dockerfile pada project

Selanjutnya penjelasan mengenai parameter pembuatan Dockerfile dapat dilihat pada

**Modul 2.6** berikut ini :

```
FROM php:7.4-apache
COPY project/ /var/www/html/
RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
RUN docker-php-ext-install mysqli
EXPOSE 80
```

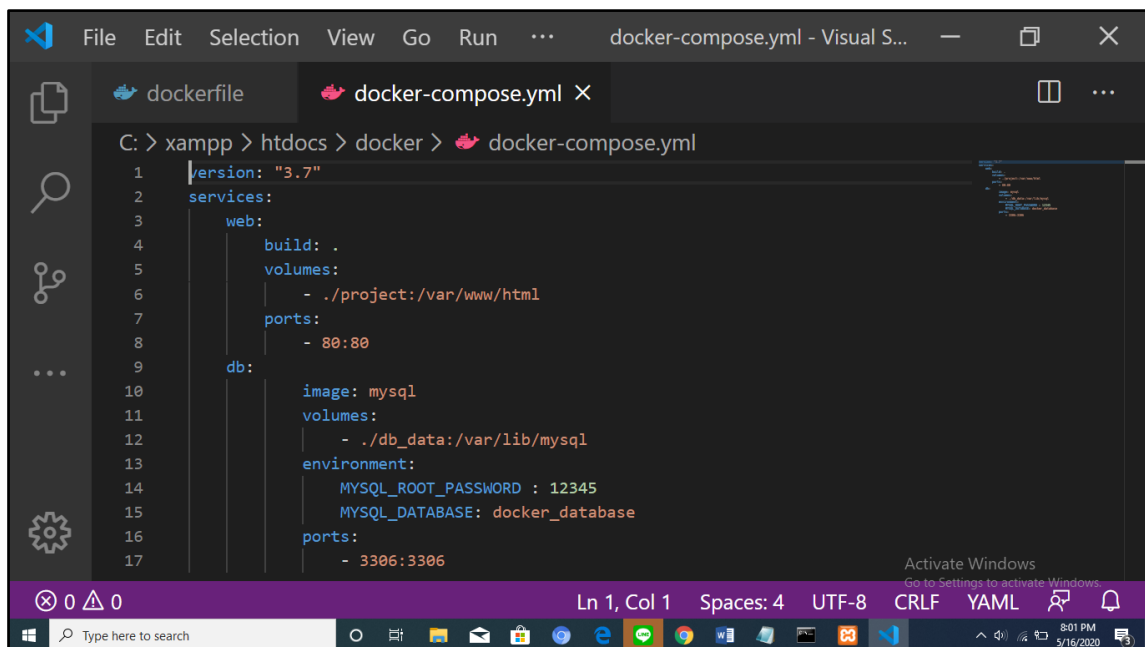
Keterangan :

- FROM php:7.4-apache : menggunakan image php:7.4-apache
- RUN : menjalankan command
- EXPOSE : meng-expose PORT yang digunakan untuk berkomunikasi
- COPY : command menyalin suatu file atau direktori atau mengcopy dari asal ke tujuan, image yang akan di build ada di folder project dan tujuannya ada di linux.
- echo "ServerName localhost" >> /etc/apache2/apache2.conf : Untuk membuat file yang berfungsi untuk menyediakan container untuk image yang dipull kemudian menampilkan server pada container di localhost kemudian mengaktifkan konfigurasi yang dibuat di service apache2
- docker-php-ext-install mysqli : menginstall extention mysqli karena untuk akses driver php membutuhkan mysqli

### Modul 2.6 Parameter pembuatan *dockerfile*

Kemudian buatlah docker-compose untuk project yang akan dibuat yang ditunjukkan oleh

**Gambar 2.20** seperti gambar berikut :



**Gambar 2.20** Docker-compose pada project

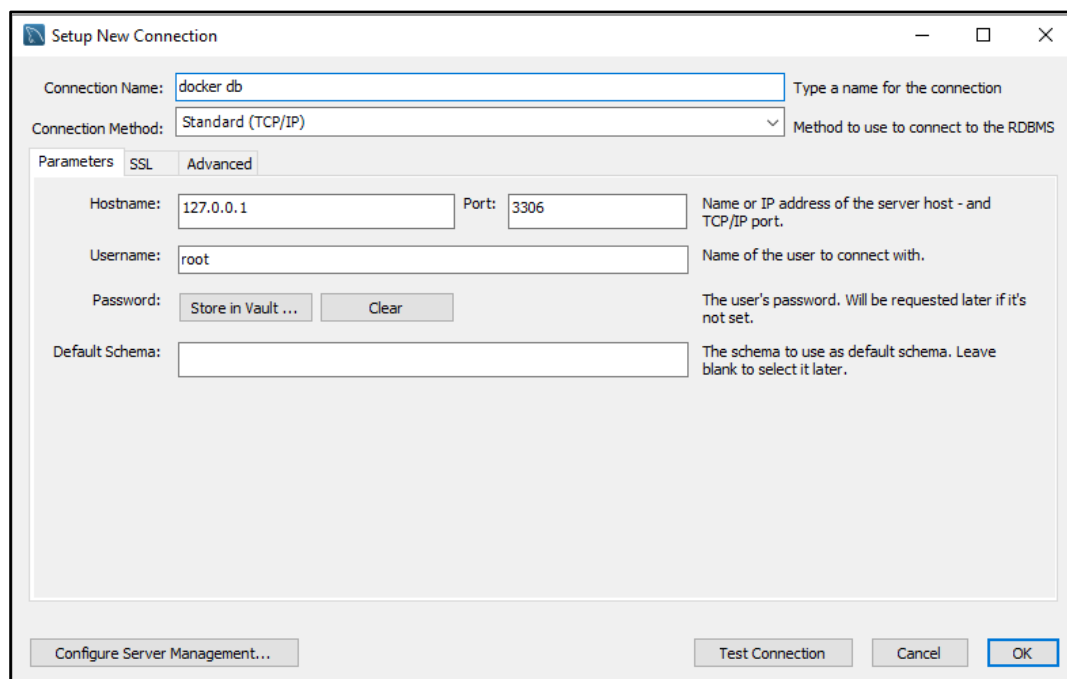
Didalam file docker-compose terdapat beberapa parameter. Terdapat dua service utama yang dijalankan, yaitu web dan db. Web berisi web server sistem informasi jual beli seafood. Untuk service db berisi image MySQL dan konfigurasinya. Untuk container name dari service web adalah *docker\_web\_1* dan service db memiliki nama container *docker\_db\_1*. Environment variabel yang ada pada MySQL adalah konfigurasi manajemen database. Untuk penjelasan lebih lanjut mengenai parameter pembuatan docker-compose dapat dilihat pada **Modul 2.7** berikut :

Keterangan :

- version : untuk menyetting versi yang digunakan.
- services : untuk penamaan servis yang berjalan.
- web : untuk nama aplikasi yang ingin digunakan.
- build : untuk build project yang digunakan.
- volumes : untuk mount dari dari volume yang digunakan.
- ports : untuk menyetting port yang akan dibangun.
- db : untuk konfigurasi database.
- image: untuk setting image yang akan digunakan.
- environment: untuk setting environment pada myaql.
- MYSQL\_ROOT\_PASSWORD : untuk setting password pada database.
- MYSQL\_DATABASE : untuk nama database yang akan digunakan.

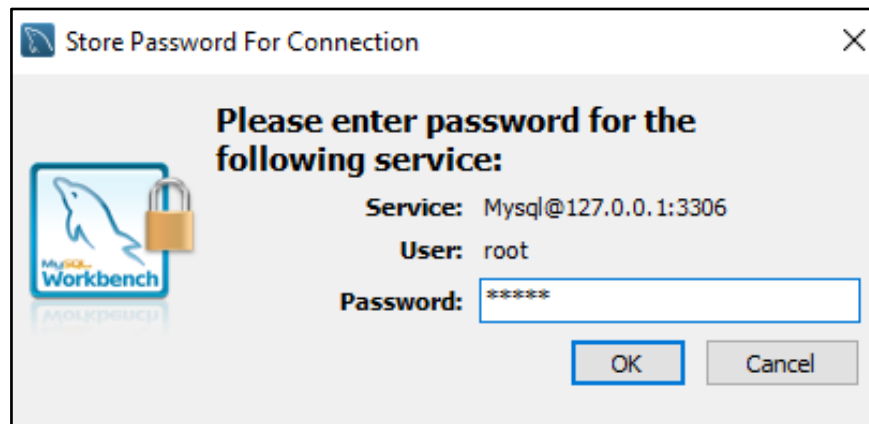
### Modul 2.7 Parameter pembuatan docker-compose.yml

Langkah selanjutnya yaitu membuat database koneksi pada MySQL Workbench yang ditunjukkan oleh **Gambar 2.21** seperti gambar berikut :

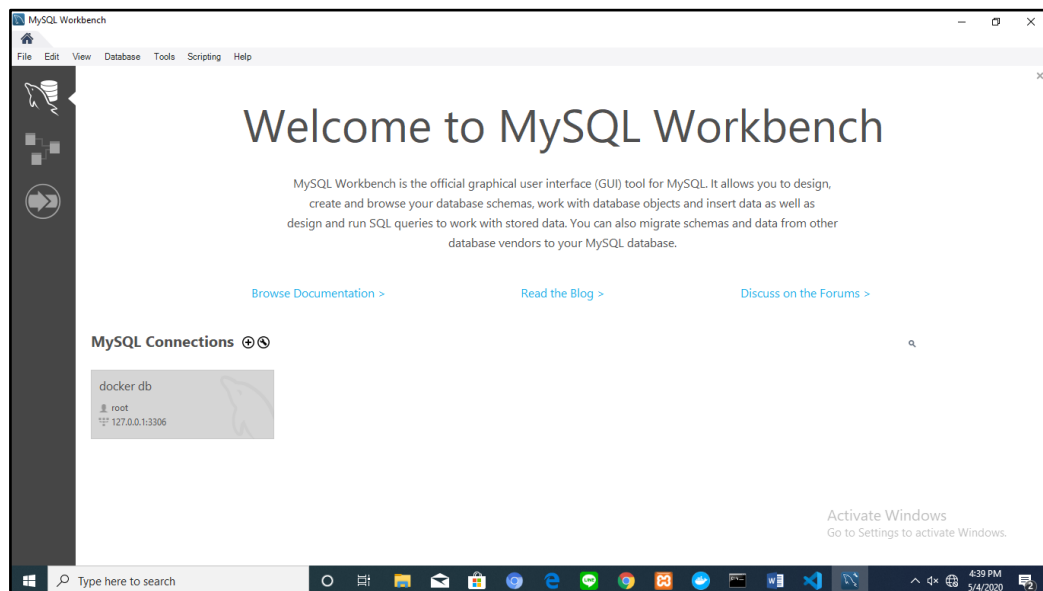


**Gambar 2.21** Create database connection pada MySQL WorkBench 8.0 CE

Kemudian masukkan password yang telah dibuat pada docker-compose.yml yang ditunjukkan oleh **Gambar 2.22** seperti gambar berikut :

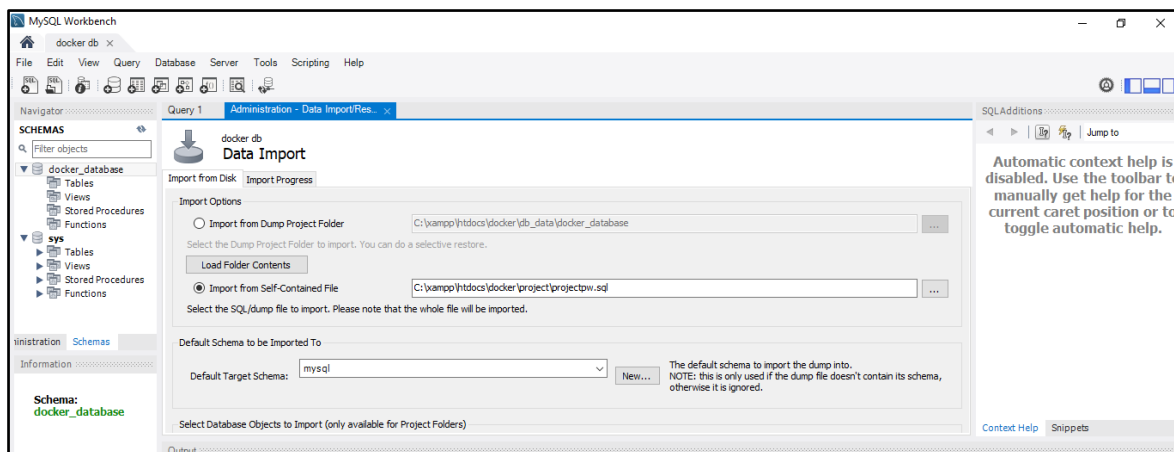


**Gambar 2.22** Masukkan password sesuai yang telah dibuat pada file docker-compose.yml

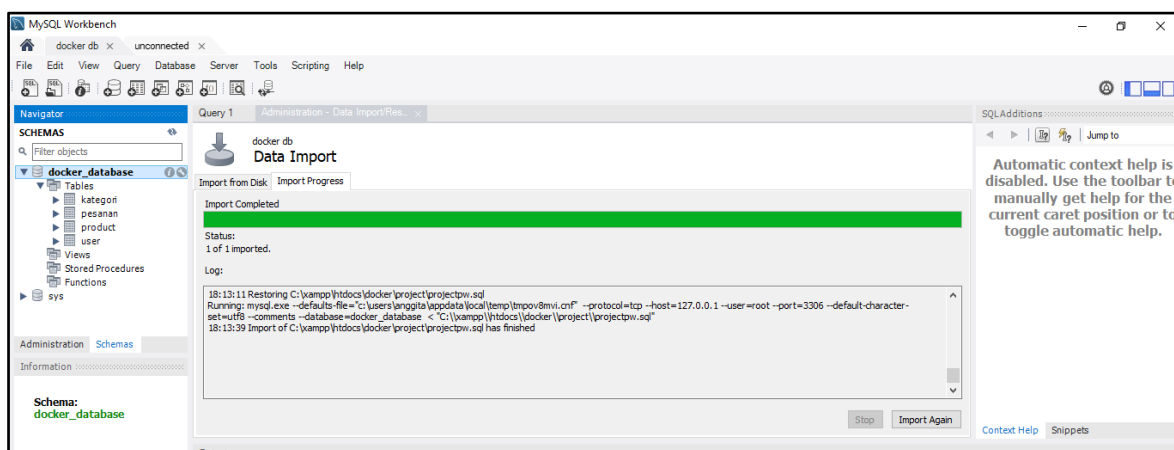


**Gambar 2.23** Tampilan jika telah berhasil membuat MySQL koneksi

Setelah berhasil melakukan login dan konfigurasi pada MySQL Workbench selanjutnya kita membuat database baru. Setelah itu kita import data dari database yang sudah ada sehingga terisi dengan data yang kita inginkan. Untuk lebih jelasnya bisa dilihat pada **Gambar 2.24** berikut ini :



**Gambar 2.24** Import database yang telah dibuat



**Gambar 2.25** Database berhasil diimport

Setelah membuat file docker-compose.yml maka selanjutnya adalah menjalankan nya dengan cara mengetikan docker-compose up yang nantinya akan menjalankan multi container yaitu container untuk php dan mysql. Sebelumnya kita tidak melakukan *pull* pada *image MySQL*, namun pada file docker-compose kita menggunakan *image MySQL*, perintah ini ditujukan untuk menginstall MySQL saat proses build *multicontainer* berjalan. Kemudian akan terbuat *container* baru yang bernama *docker\_web\_1* dan *docker\_db\_1* yang proses dan perintah programnya dapat dilihat pada **Gambar 2.26** dan **Perintah Program 2.2** seperti berikut :

```
docker-compose up
```

Keterangan :

- docker-compose : command untuk mengeksekusi perintah-perintah Docker Compose.
- up : parameter dalam docker-compose untuk menjalankan multicontainer.

### **Perintah Program 2.2** Parameter eksekusi docker compose



```

C:\xampp\htdocs\docker>docker-compose up
Creating network "docker_default" with the default driver
Building web
Step 1/5 : FROM php:7.4-apache
---> b7ec96dce970
Step 2/5 : COPY project/ /var/www/html/
---> 8fd679c48121
Step 3/5 : RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
---> Running in 5d26f1bd7c97
Removing intermediate container 5d26f1bd7c97
---> c3fb77511a42
Step 4/5 : RUN docker-php-ext-install mysqli
---> Running in 4b258bf8b39f
Configuring for:
PHP Api Version:      20190902
Zend Module Api No:   20190902
Zend Extension Api No: 320190902
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for a sed that does not truncate output... /bin/sed
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for cc... cc
checking whether the C compiler works... yes

```

**Gambar 2.26** Proses build project

Untuk melihat container yang telah dibuat dan sedang berjalan bisa menggunakan

**Perintah Program 2.3** seperti berikut :

```

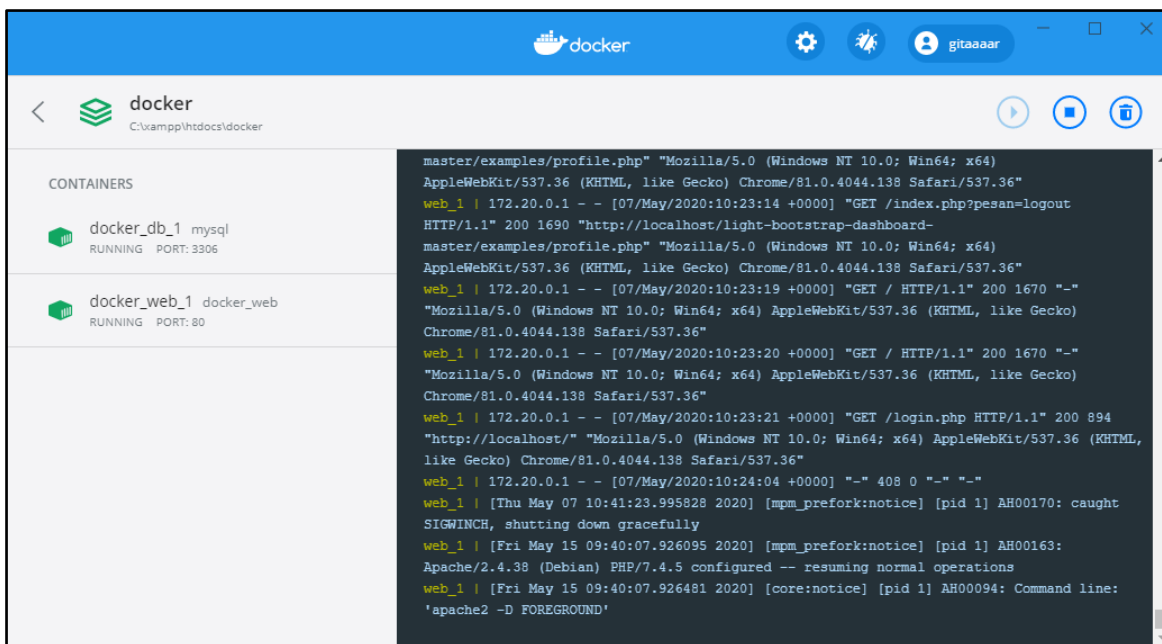
docker ps
docker ps -a

Keterangan :
- ps: command untuk menampilkan container yang sedang
  running.
- ps -a : command untuk menampilkan container yang tersedia.

```

**Perintah Program 2.3** Parameter untuk melihat container

Jika ingin melihat prosesnya pada *docker desktop* bisa dilihat pada **Gambar 2.27** seperti berikut :



**Gambar 2.27** Proses *build* telah berhasil dan *container* sedang berjalan

Jika ingin memberhentikan container, maka bisa menggunakan **Perintah Program**

**2.4** seperti berikut :

```
docker stop [container id]
docker-compose down
```

Keterangan :

- stop : command untuk memberhentikan satu container.
- down : parameter dalam docker-compose untuk memberhentikan multicontainer.

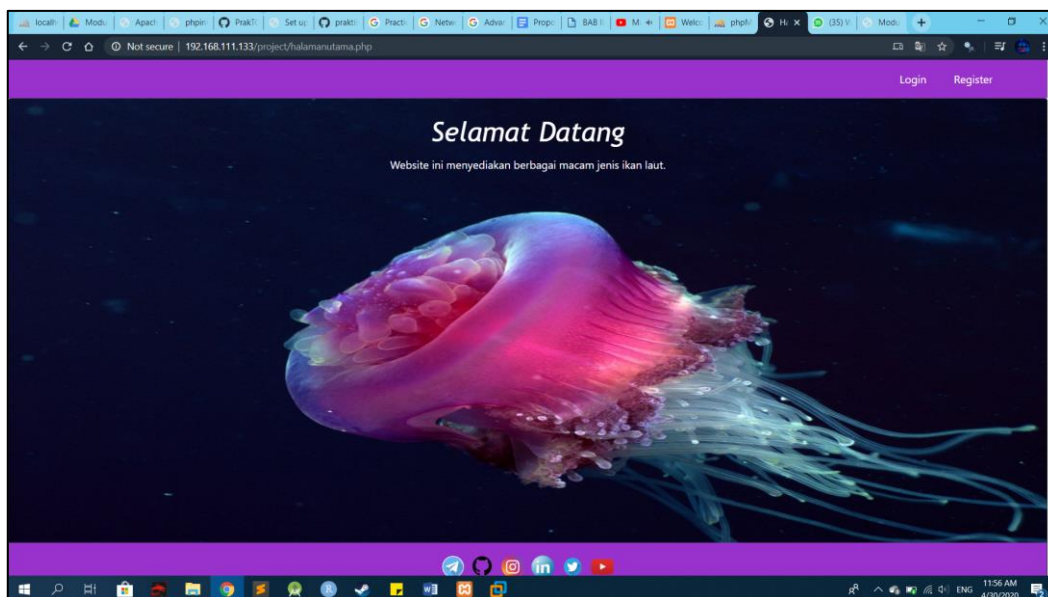
#### **Perintah Program 2.4** Parameter untuk memberhentikan container

### **2.5 Hasil Implementasi**

Hasil implementasi pada proyek Sistem Informasi Jual Beli Seafood dan Proses Pembuatan Dockerfilenya akan dibahas pada sub bab terkait.

#### **2.5.1 Hasil Implementasi Proyek Pertama**

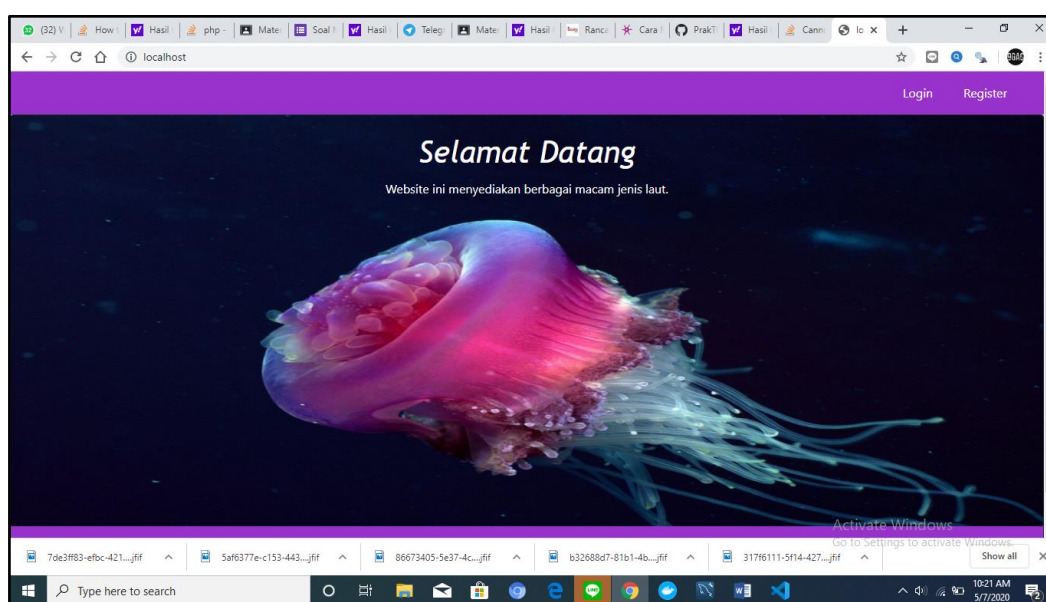
Pada proyek pertama yaitu membuat layanan LAMPP yang berisikan tugas kuliah pada VM Ubuntu yang berisikan LAMPP dengan judul dari kelompok kami yaitu “Sistem Informasi Jual Beli Seafood Menggunakan Layanan Ubuntu LAMPP dan Pembuatan Dockerfilenya” dan proyek satu kelompok kami ini sudah bisa diakses dari lokal dengan mengetikkan format url <http://IPdariservercloud/namafolderfile> dan untuk tugas kelompok kami memiliki url <http://192.168.111.133/project/halamanutama.php>. Hasil dari proyek pertama kelompok kami dapat dilihat pada **Gambar 2.28** sebagai berikut:



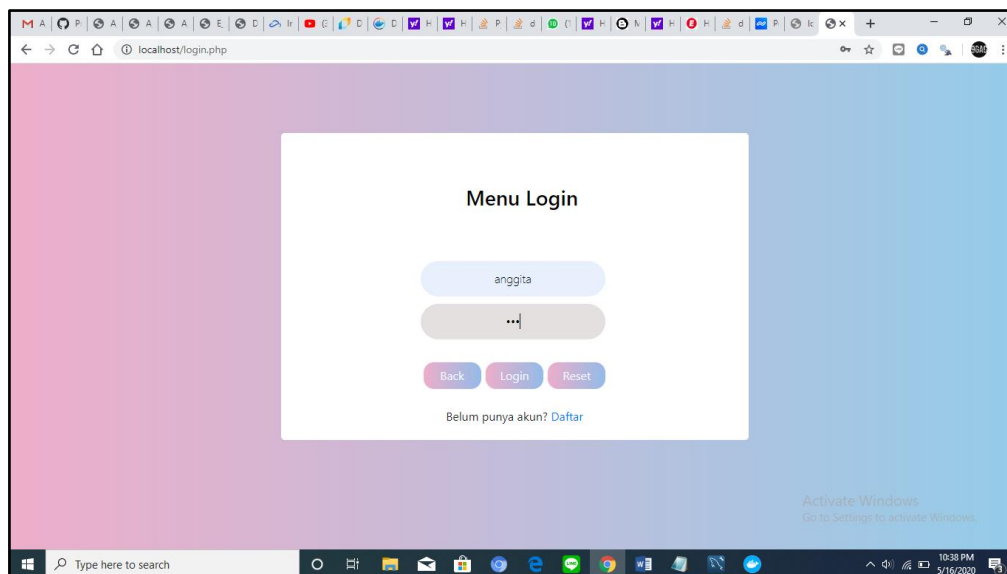
**Gambar 2.28** Halaman Awal Sistem Informasi Penjualan *Seafood*

### 2.5.2 Hasil Implementasi Proyek Kedua

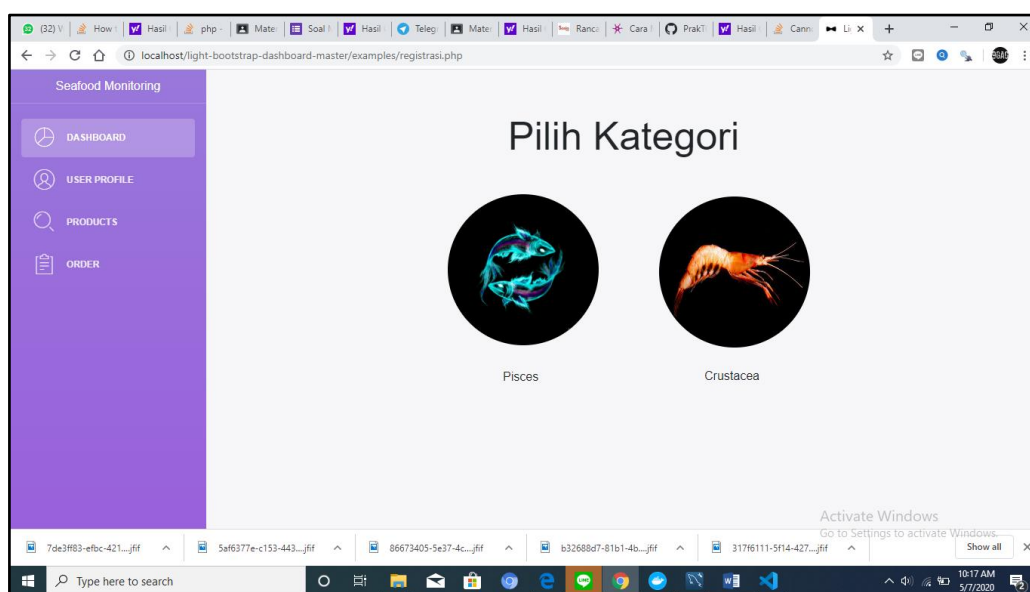
Pada proyek kedua yaitu membuat *dockerfile* yang berisikan tugas kuliah pada *Docker Desktop* yang berisikan *Docker Images* dengan judul dari kelompok kami yaitu “Sistem Informasi Jual Beli Seafood Menggunakan Layanan Ubuntu LAMPP dan Pembuatan Dockerfilenya” dan proyek kedua kelompok kami ini sudah bisa diakses dengan mengetikkan format url <http://localhost:80> . Hasil dari proyek pertama kelompok kami yang merupakan halaman awal sistem informasi jual beli seafood, dapat dilihat pada **Gambar 2.29** sebagai berikut:



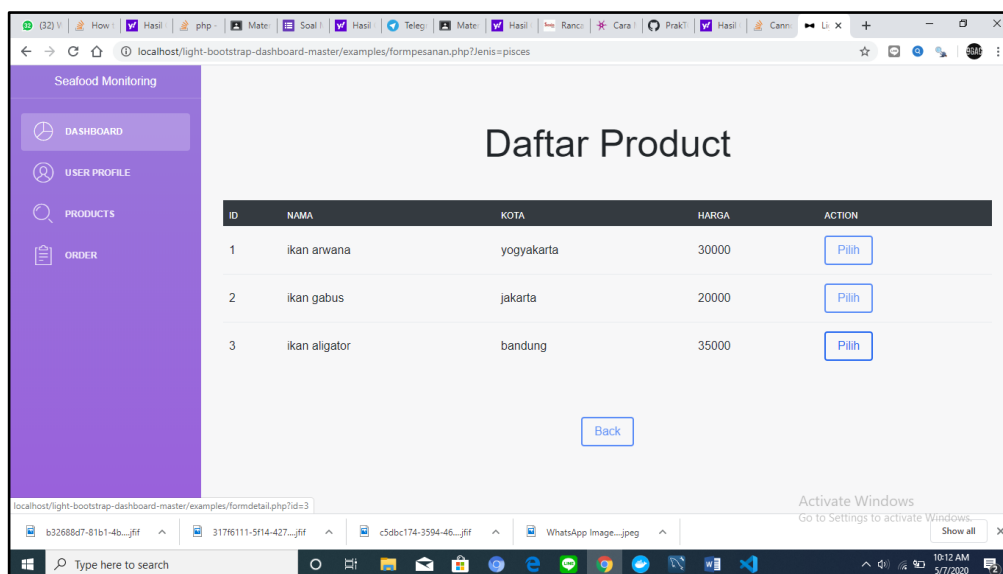
**Gambar 2.29** Halaman Awal Sistem Informasi Penjualan *Seafood*



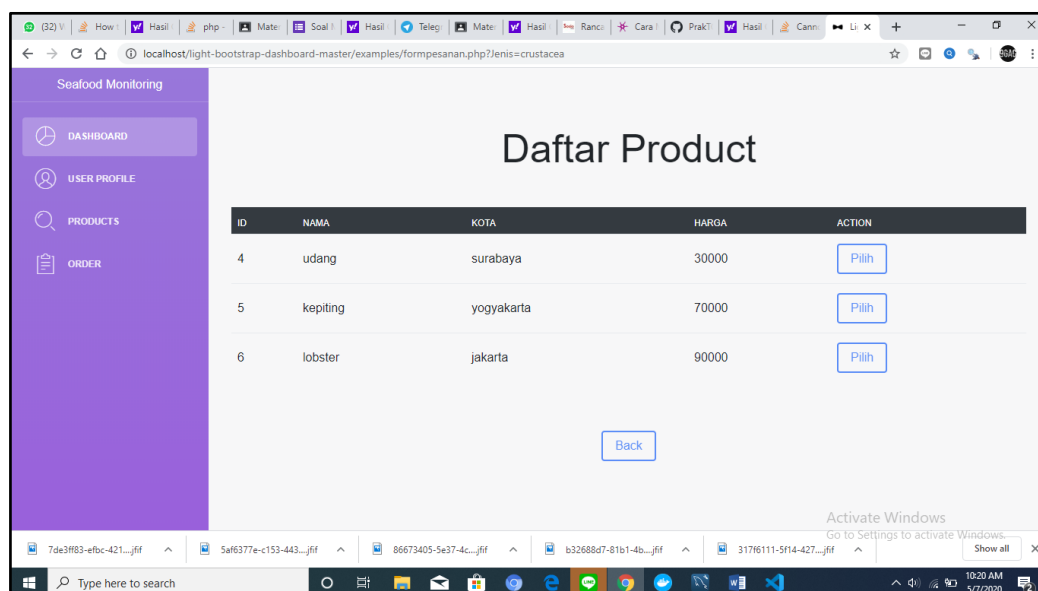
**Gambar 2.30** Tampilan Login Sistem Informasi Penjualan *Seafood*



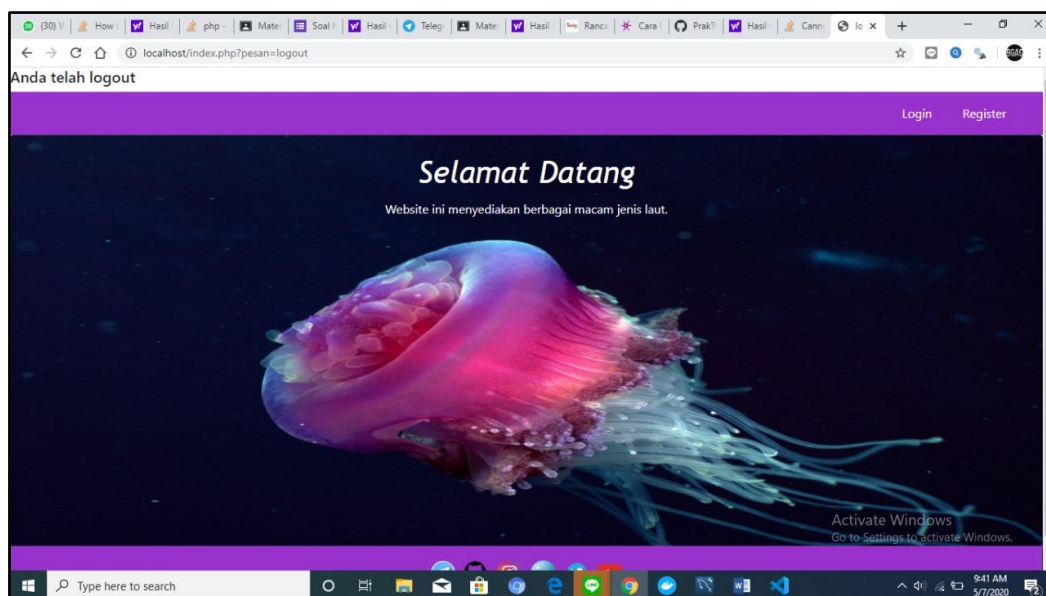
**Gambar 2.31** Tampilan halaman *products*



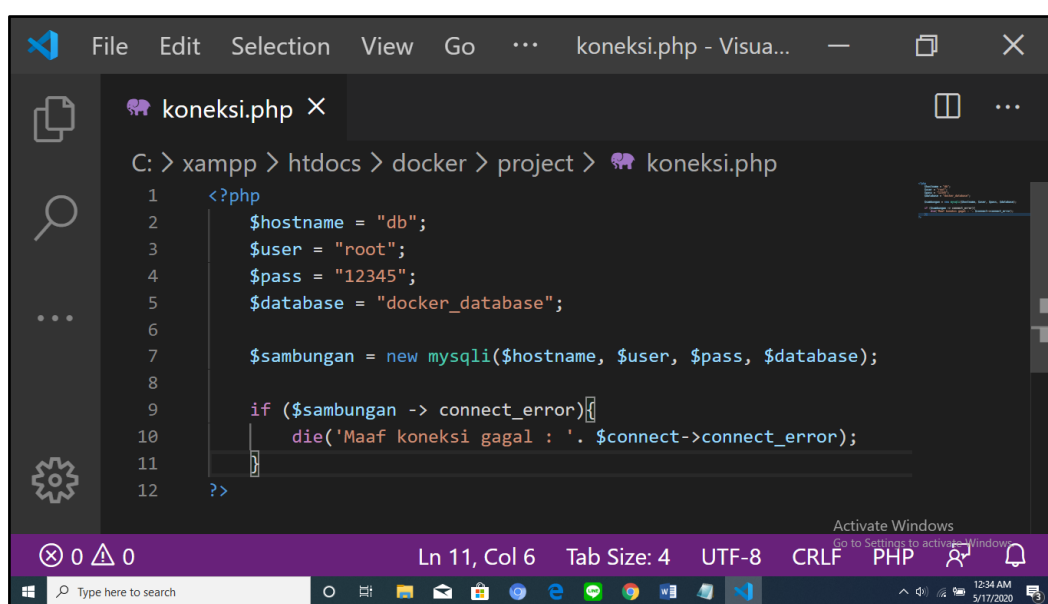
**Gambar 2.32** Tampilan halaman daftar *products* jenis *pisces*



**Gambar 2.33** Tampilan halaman daftar *products* jenis *crustacean*



**Gambar 2.34** Tampilan berhasil *logout*



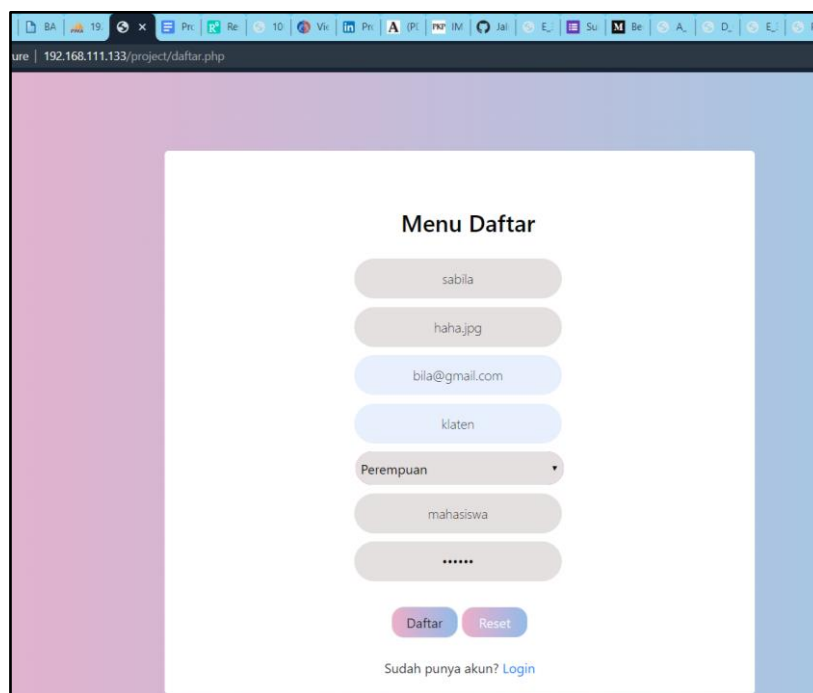
**Gambar 2.35** File koneksi yang menghubungkan antara *project*, *database*, dan *docker*

## 2.6 Pengujian Singkat

Dalam pengujian singkat ini, terdapat tiga hal yang akan diuji dalam masing – masing proyek yang akan dibahas pada subbab terkait.

### 2.6.1 Pengujian Singkat Proyek Pertama

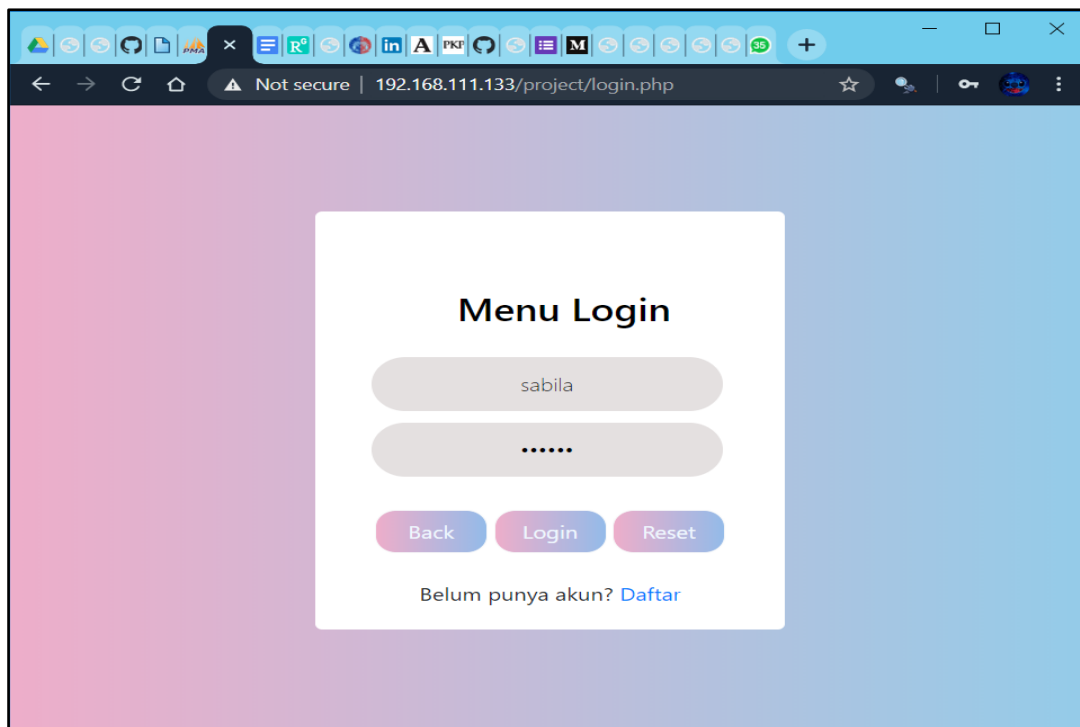
Pada pengujian untuk proyek pertama terdapat tiga skenario, yaitu membuat akun pengguna pada Sistem Informasi Jual Beli Seafood, edit data pengguna, dan melakukan pemesanan *seafood*. Proses yang pertama kali yaitu melakukan registrasi akun pengguna dimana datanya dapat dilihat pada **Gambar 2.36** berikut ini:



The image shows a web browser window with the address bar displaying '192.168.111.133/project/daftar.php'. The main content area features a registration form titled 'Menu Daftar'. The form includes several input fields: a text field with 'sabila', a file upload field with 'haha.jpg', an email field with 'bila@gmail.com', a location field with 'klaten', a gender dropdown menu currently set to 'Perempuan', a role field with 'mahasiswa', and a password field with masked characters '\*\*\*\*\*'. Below these fields are two buttons: 'Daftar' (pink) and 'Reset' (blue). At the bottom of the form, there is a link that says 'Sudah punya akun? Login'.

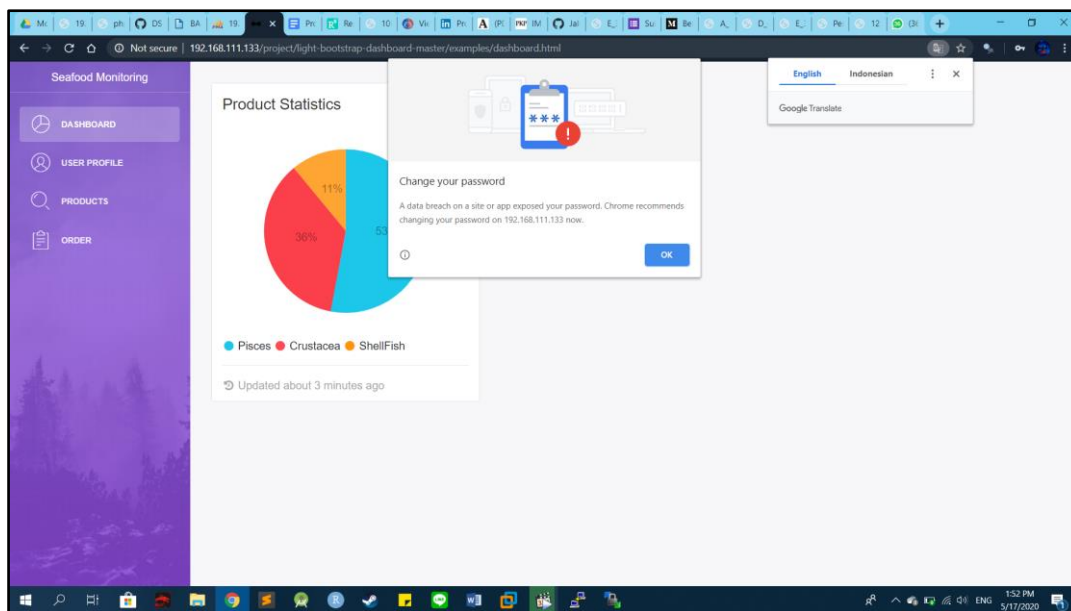
**Gambar 2.36** Proses pembuatan akun

Selanjutnya, untuk membuktikan jika pembuatan akun berhasil maka dicoba *login* dengan menggunakan *username* dan *password* yang tadi dibuat. Proses login dilakukan dengan klik pada *Login* yang berada pada halaman utama *website* pada bagian pojok kanan atas bersebelahan dengan *Register*. Data yang dimasukkan ketika *login* data dilihat pada **Gambar 2.37** berikut ini:



**Gambar 2.37** Proses login pada Sistem Informasi Jual Beli Seafood

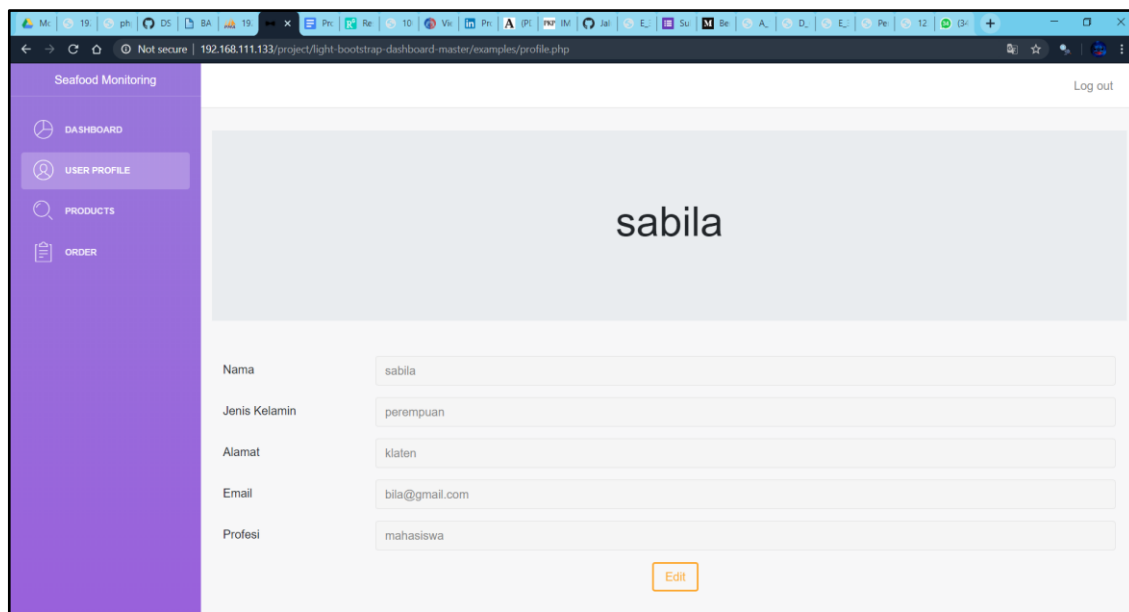
Proses login dinyatakan berhasil jika masuk ke halaman utamanya. Halaman awal setelah melakukan login dapat dilihat pada **Gambar 2.38** berikut:



**Gambar 2.38** Halaman utama pada Sistem Informasi Jual Beli Seafood

Skenario pertama sudah berhasil dilakukan selanjutnya melakukan skenario kedua yaitu melakukan *edit* data pengguna. Pertama yang dilakukan yaitu klik pada tab *User Profile* kemudian akan tampil seperti **Gambar 2.39** berikut:





Seafood Monitoring Log out

**sabila**

Nama: sabila

Jenis Kelamin: perempuan

Alamat: klaten

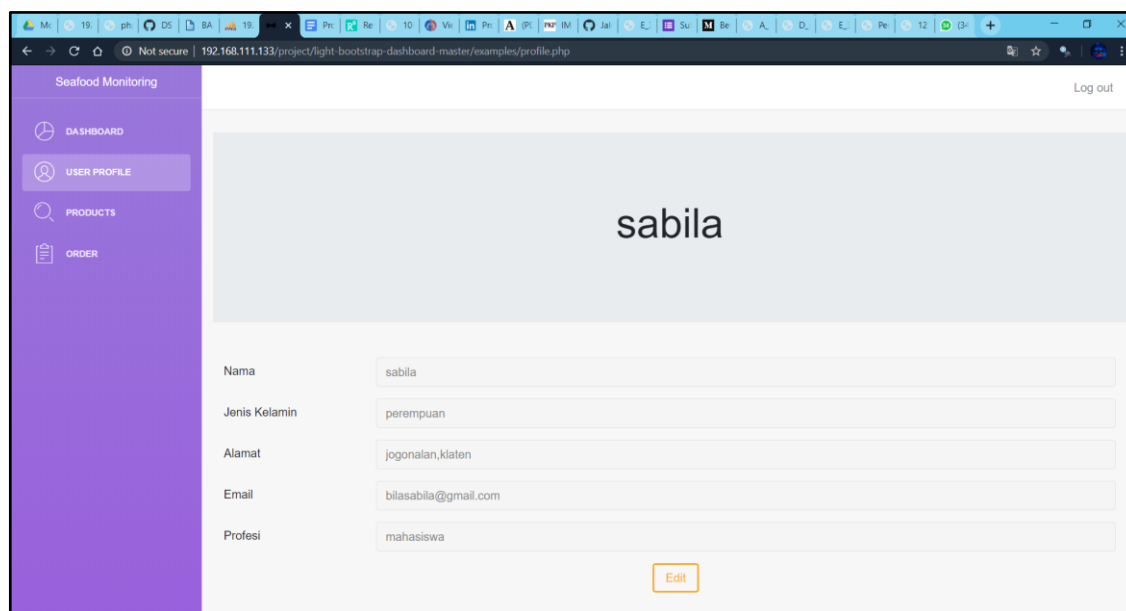
Email: bila@gmail.com

Profesi: mahasiswa

[Edit](#)

**Gambar 2.39** Data pengguna sebelum di-*edit*

Pada skenario kedua, setelah klik pada tombol “*Edit*” selanjutnya akan masuk ke halaman *edit*. Sebagai contoh bagian yang di-*edit* adalah pada bagian email dan pada alamat. Setelah klik pada tombol “*Update*” maka program akan mengalihkan pada halaman *User Profile* yang menampilkan data baru yang setelah di-*edit* dan data baru setelah di-*edit* bisa dilihat pada **Gambar 2.40** berikut:



Seafood Monitoring Log out

**sabila**

Nama: sabila

Jenis Kelamin: perempuan

Alamat: jogonalan,klaten

Email: bilasabila@gmail.com

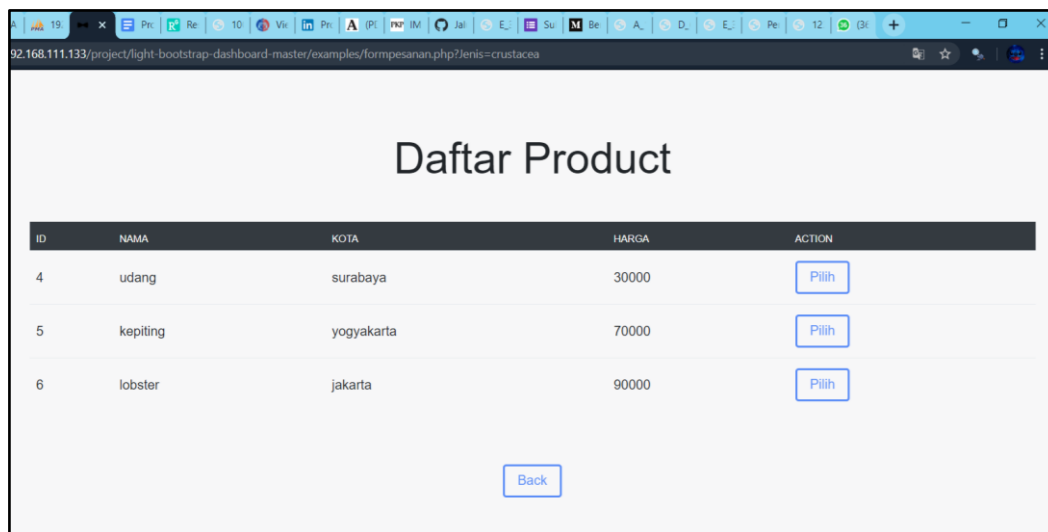
Profesi: mahasiswa

[Edit](#)

**Gambar 2.40** Data pengguna setelah di-*edit*

Setelah skenario kedua berhasil, lanjut pada skenario ketiga yaitu melakukan pemesanan *seafood*. Pengguna dapat melakukan pemesanan pada tab *Products*. Pada

halaman ini pengguna akan diminta untuk memilih antara *Pisces* atau *Crustacea* dimana *Pisces* berisi ikan-ikanan dan *Crustacea* berisi udang-udangan dan untuk skenario ini menggunakan yang *Crustacea*. Halaman awal setelah klik pada icon *Crustacea* dapat dilihat pada **Gambar 2.41** berikut:

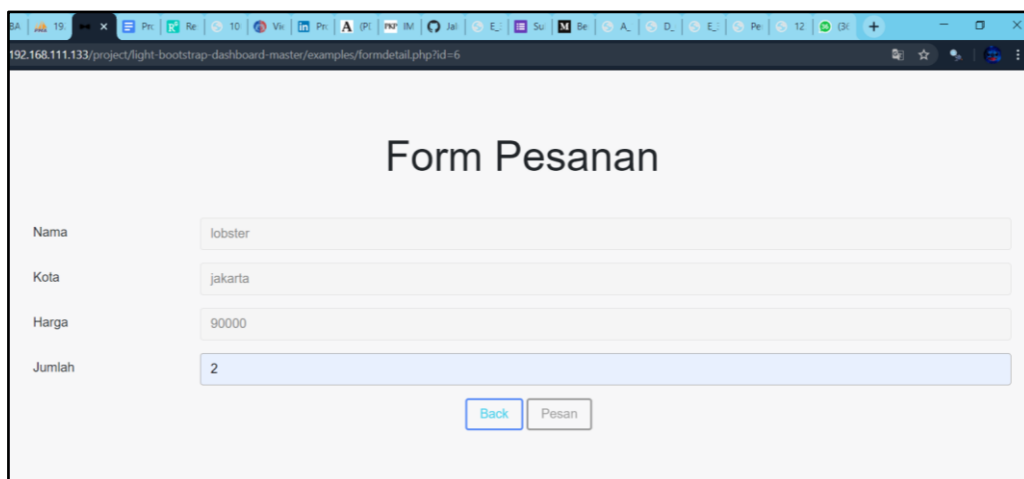


ID	NAMA	KOTA	HARGA	ACTION
4	udang	surabaya	30000	<a href="#">Pilih</a>
5	kepiting	yogyakarta	70000	<a href="#">Pilih</a>
6	lobster	jakarta	90000	<a href="#">Pilih</a>

[Back](#)

**Gambar 2.41** Data pengguna setelah di-edit

Kemudian, dipilih yang tombol pada lobster setelah itu akan masuk ke halaman Form Pesanan lalu masukkan jumlah banyaknya lobster yang akan dipesan kemudian akan dihitung jumlah total yang harus dibayarkan seperti pada **Gambar 2.42** dan ketika klik tombol pesan kemudian akan ditampilkan Form Detail Pesanan dimana terdapat tampilan jumlahnya seperti pada **Gambar 2.43** berikut:



**Form Pesanan**

Nama: lobster

Kota: jakarta

Harga: 90000

Jumlah: 2

[Back](#) [Pesan](#)

**Gambar 2.42** Form Pesanan

The screenshot shows a web browser window with the URL `92.168.111.133/project/light-bootstrap-dashboard-master/examples/formdetail2.php?id=6`. The page title is "Form Detail Pesanan". The form has the following fields:

Field	Value
Nama	lobster
Kota	jakarta
Harga	90000
Jumlah	2
Total	180000

At the bottom of the form, there are two buttons: "Back" (blue) and "Pesanan" (grey).

**Gambar 2.43** Form Detail Pesanan

Form Detail Pesanan menunjukkan jika proses inputan jumlah lobster yang akan dipesan dapat diproses sehingga akan menampilkan jumlah yang harus dibayarkan yang menandakan skenario ketiga berhasil dijalankan.

### 2.6.2 Pengujian Singkat Proyek Kedua

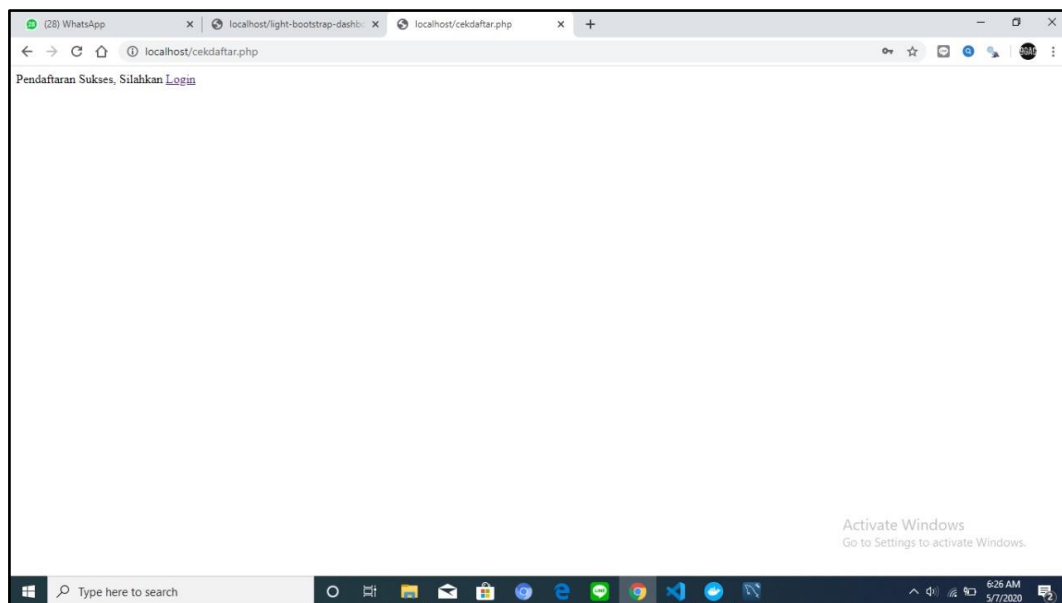
Pada pengujian untuk proyek kedua terdapat tiga skenario, yaitu membuat akun pengguna pada Sistem Informasi Jual Beli Seafood, edit data pengguna, dan melakukan pemesanan *seafood*. Proses yang pertama kali yaitu melakukan registrasi akun pengguna dimana datanya dapat dilihat pada **Gambar 2.44** dan **Gambar 2.45** berikut ini:

The screenshot shows a web browser window with the URL `localhost/daftar.php`. The page title is "Menu Daftar". The form has the following fields:

Field	Value
Username	lutfah
Password	oceanjog
Email	lutfahbariana@gmail.com
Address	debongwetan
Gender	Perempuan
Role	kepala sekolah

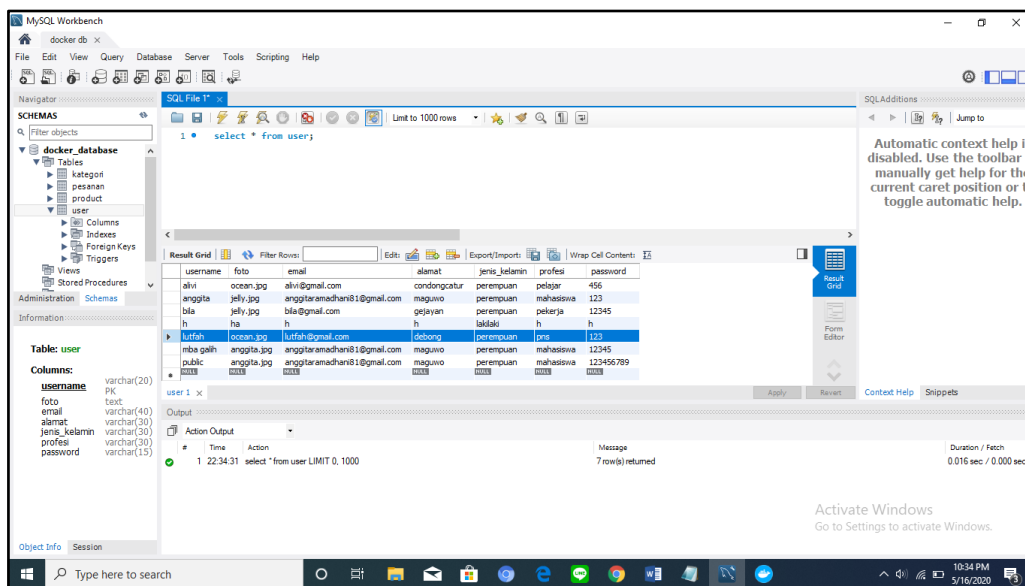
At the bottom of the form, there are two buttons: "Daftar" (blue) and "Reset" (grey). Below the buttons, there is a message: "Sudah punya akun? Login".

**Gambar 2.44** Menu Daftar Sistem Informasi Penjualan *Seafood*



**Gambar 2.45** Proses pendaftaran berhasil dilakukan

Setelah melakukan pendaftaran maka tentunya akan berpengaruh pada database yang kita gunakan semisal pada salah satu tabel. Kita bisa melihat perubahan database tersebut pada MySQL Workbench pada bagian tabel user karena mendapat tambahan data dari hasil proses pendaftaran pada **Gambar 2.46** seperti berikut :



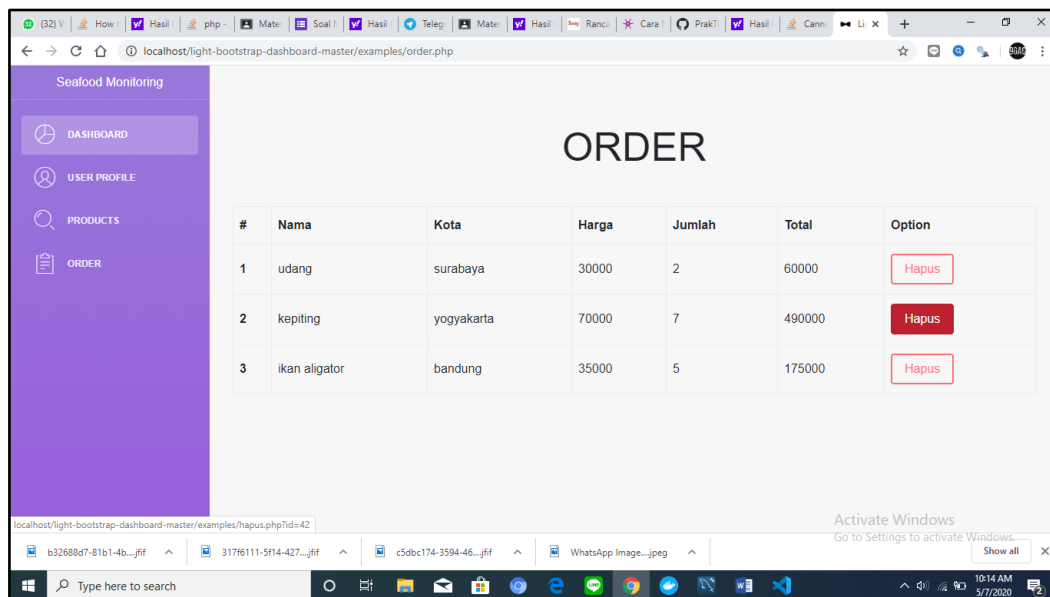
**Gambar 2.46** Isi database dari tabel user

Selanjutnya pengujian pada halaman pemesanan, jika di halaman *products* memilih jenis pisces dengan jenis ikan alligator akan muncul tampilan seperti **Gambar 2.47** lalu masukkan jumlah banyaknya ikan aligator yang akan dipesan dan ketika klik tombol pesan kemudian akan ditampilkan Form Detail Pesanan dimana terdapat tampilan jumlahnya seperti pada **Gambar 2.48** berikut:

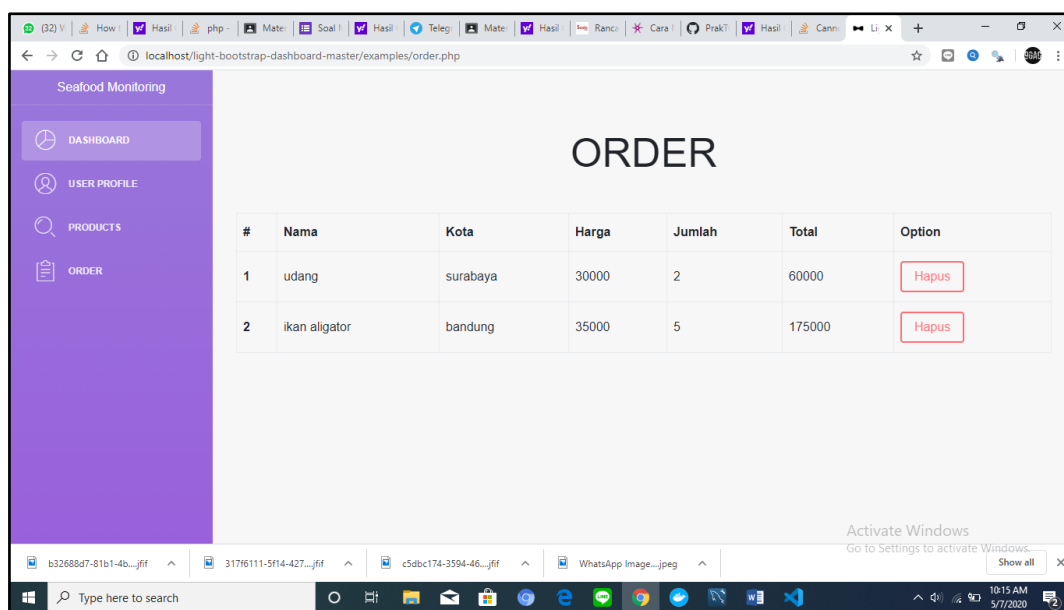
**Gambar 2.47** Tampilan halaman *form* pesanan jenis ikan aligator

**Gambar 2.48** Tampilan halaman *form detail* pesanan jenis ikan alligator

Setelah pesanan berhasil di proses akan masuk ke halaman order yang berarti pesanan telah berhasil masuk ke database yang ditunjukkan oleh **Gambar 2.49** yang kemudian kita akan mencoba proses penghapusan pesanan kepiting pada halaman order yang hasilnya ditampilkan pada **Gambar 2.50** seperti berikut :

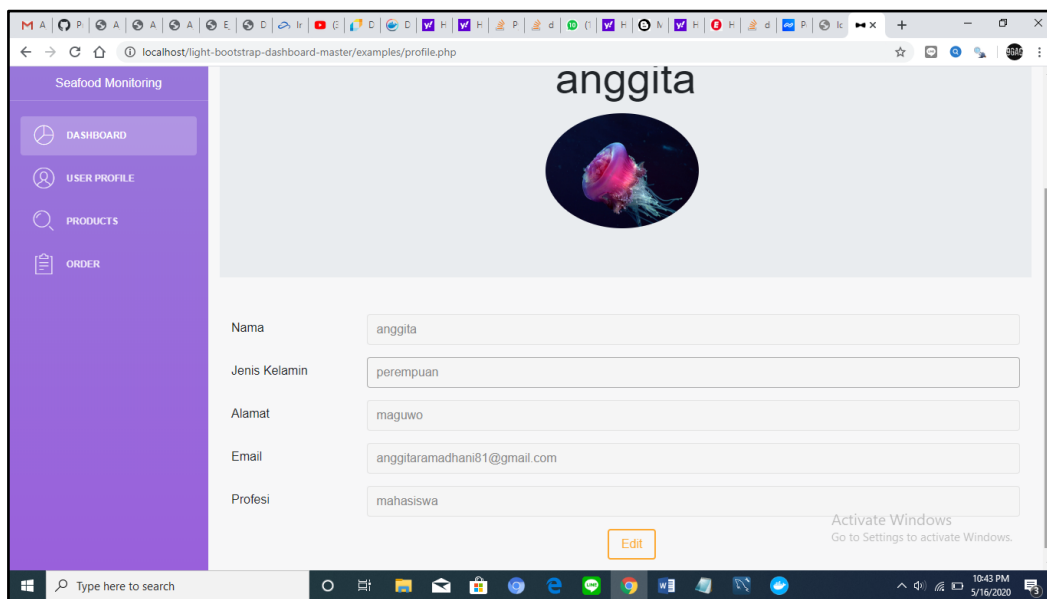


**Gambar 2.49** *Input* pesanan berhasil dan masuk ke halaman *order*

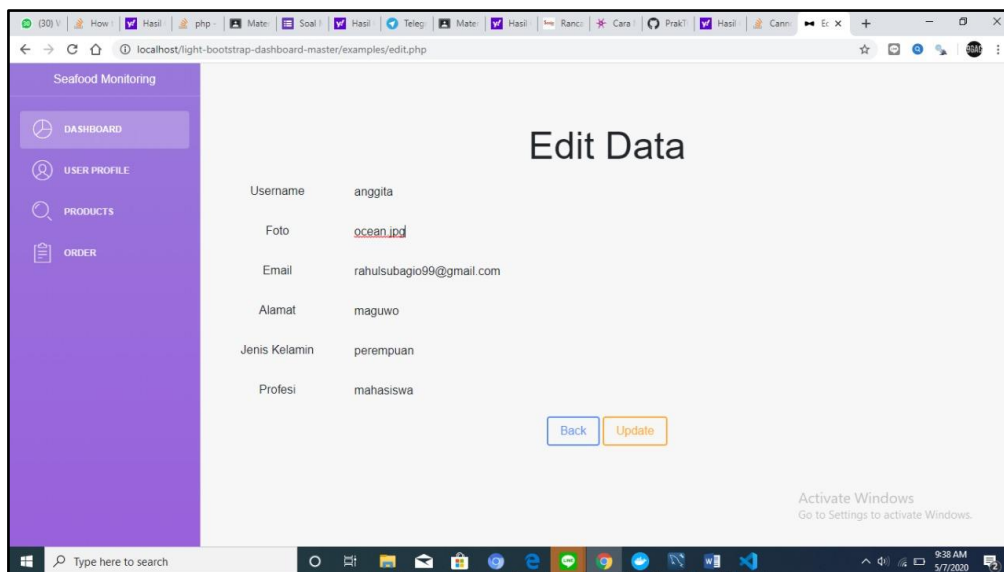


**Gambar 2.50** Halaman *order* setelah dihapus pesanan kepiting

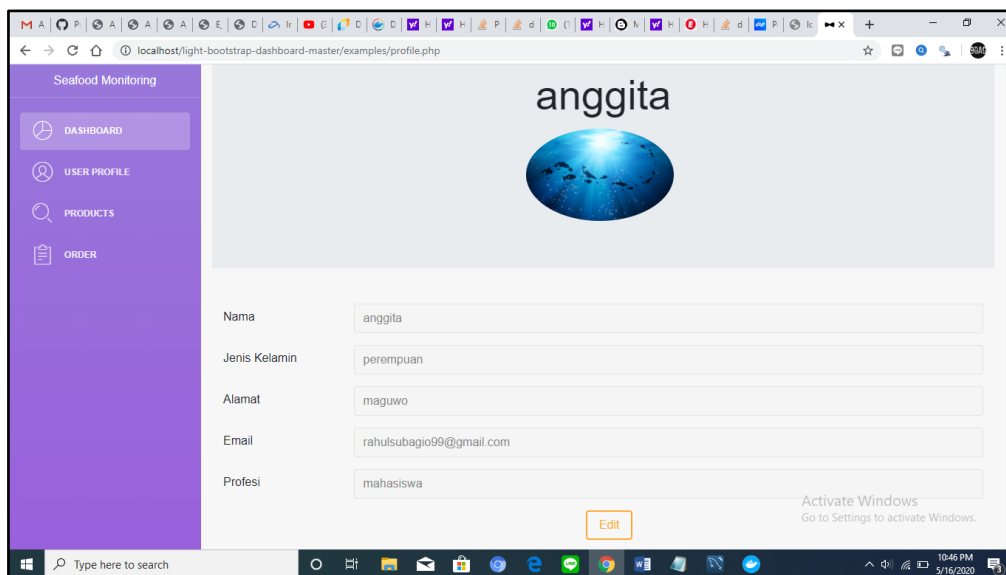
Terakhir adalah pengujian *editing* akun yang dimana user akan diarahkan pada menu profil dan diberikan pilihan untuk mengedit akun, jika memilih edit user akan diarahkan ketampilan editing akun, dan user mengisi sejumlah keterangan yang akan diedit, pada tahap ini berhasil sehingga data masuk ke database dan bisa dilihat pada halaman profil hasil perubahan data yang ditunjukkan pada **Gambar 2.51** dan proses pengeditannya dapat dilihat pada **Gambar 2.52** serta hasil perubahan datanya ditunjukkan pada **Gambar 2.53** seperti berikut :



**Gambar 2.51** Tampilan *User Profile* Sistem Informasi Penjualan *Seafood*



**Gambar 2.52** Tampilan *Edit Data User* Sistem Informasi Penjualan *Seafood*



**Gambar 2.53** Data berhasil di *edit*



### BAB III

#### JADWAL Pengerjaan dan Pembagian Tugas

##### 3.1 Agenda Pengerjaan

Berikut pada **Tabel 3.1** merupakan pembagian jenis tugas proyek akhir “Sistem Informasi Jual Beli Seafood Menggunakan Layanan Ubuntu LAMPP dan Proses Pembuatan Dockerfilenya” terhadap alokasi waktu pengerjaan pada bulan Februari, Maret, April, dan Mei tahun 2020:

**Tabel 3.1** Agenda Pengerjaan Proyek

No.	Jenis Tugas	Waktu Pengerjaan											
		Feb	Maret				April				Mei		
		4	1	2	3	4	1	2	3	4	1	2	3
1.	Analisa Persoalan												
2.	Pembagian Tugas												
3.	Penentuan Proyek Tugas 1 (LAMPP)												
4.	Install VMWare dan Ubuntu												
5.	Install Apache2, MySQL, PHP, phpMyAdmin												
6.	Testing Tugas 1												
7.	Pembuatan Laporan Progress 1												
8.	Revisi Laporan 1												
9.	Penentuan Proyek Tugas 2 (Docker)												
10.	Install Docker												
11.	Pembuatan Laporan Progress 2												
12.	Testing Tugas 2												
13.	Revisi Laporan 2												
14.	Presentasi Proyek Akhir												

##### 3.2 Keterangan Pembagian Tugas

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

**Tabel 3.2** Pembagian Tugas Proyek

No.	Keterangan Tugas	Penanggung Jawab
1.	Perancangan Arsitektur Cloud Computing	Anggita & Bila
2.	Pengujian Singkat	Anggita & Bila
3.	Agenda Pengerjaan Proyek	Anggita & Bila
4.	Pembuatan Tugas 1 (LAMPP)	Bila
5.	Pembuatan Tugas 2 (Docker)	Anggita
6.	Pembuatan Laporan Bab 1	Anggita & Bila
7.	Pembuatan Laporan Bab 2	Anggita & Bila
8.	Pembuatan Laporan Bab 3	Bila
9.	Pembuatan Laporan Bab 4	Anggita & Bila
10.	Merapikan Laporan	Anggita & Bila

## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Dari permasalahan yang telah ada dan setelah dikerjakannya proyek akhir ini maka dapat diambil kesimpulan bahwa pada *hosting* dengan menggunakan Ubuntu LAMPP memperoleh hasil yang memuaskan dan untuk *hosting* menggunakan Docker juga memperoleh hasil yang memuaskan dimana sama-sama berhasil. Dalam pengerjaan *hosting* dengan Ubuntu LAMPP dan pembuatan *dockerfile*nya tidak terdapat kendala. Dari pengujian yang telah dilakukan pada *hosting* dengan Ubuntu LAMPP dengan menggunakan tiga skenario yang telah dibuat, semuanya dapat berjalan sesuai dengan yang diharapkan. Untuk pengujian yang dilakukan pada Docker dengan menggunakan tiga skenario yang telah dibuat, semuanya dapat berjalan sesuai dengan yang diharapkan. Pembagian tugas dapat terlaksana dengan baik, sudah merata dan sesuai dengan waktu yang telah ditentukan dimana saling membagi ketika salah satu mengerjakan proyek pertama dan lainnya mengerjakan laporan, begitupun sebaliknya ketika proyek kedua dikerjakan dan untuk beberapa bab dikerjakan secara bersama-sama sesuai bagaimana hasil yang sudah didapatkan.

#### **4.2 Saran**

Dari hasil pengerjaan proyek dimana terdapat kendala dalam pengaplikasian Docker dikarenakan kurangnya pemahaman mengenai Docker sehingga membutuhkan waktu yang lebih banyak dalam pemahaman sebelum melakukan pengaplikasiannya. Saran untuk kedepannya agar lebih baik sebaiknya dari Praktikum Teknologi Cloud Computing setidaknya memperkenalkan terlebih dahulu Docker bagaimana dan bagaimanakah pengimplementasiannya sehingga akan mempermudah dan membantu dalam menyelesaikan proyek akhir yang menggunakan Docker.

## DAFTAR PUSTAKA

Rumetna, M. S. (2018). Title Case. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 5(3), 305. <https://doi.org/10.25126/jtiik.201853595>

<https://www.slideshare.net/cahkos/sejarah-cloud-computing>

<http://rebornofdisciple.blogspot.com/2014/11/cloud-computing.html>

## **LAMPIRAN**