**LAPORAN**

**PRAKTIKUM TEKNOLOGI CLOUD COMPUTING**

**Build aplikasi web simple menggunakan Docker, Apache, dan MySQL pada Virtual BOX Ubuntu Server 18.04**

****

**Disusun oleh :**

|  |  |
| --- | --- |
| **Nama** | **: Fitra Muriyanto** |
| **NIM** | **: 165410230** |
| **Jurusan** | **: Teknik Informatika / S1** |

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

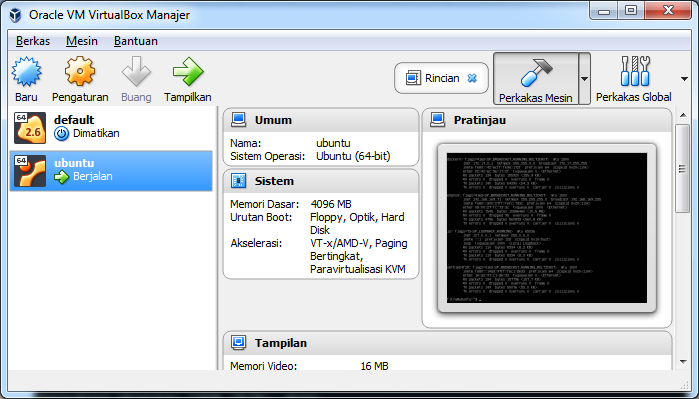
**STMIK AKAKOM**

**YOGYAKARTA**

**2018**

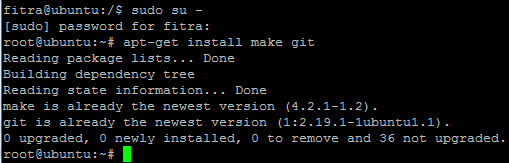
**Build aplikasi web simple menggunakan Docker, Apache, dan MySQL pada Virtual BOX Ubuntu Server 18.04**

1. Untuk step awal pertama yang lakukan adalah dengan menginstall ubuntu server pada Virtual box yang nantinya akan kita install docker dan portainer di dalamnya

****

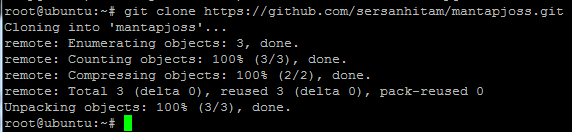
1. Apabila Virtual box sudah terinstall ubuntu server, maka selanjutnya adalah menginstall Docker dan Portainer. Untuk menyingkat waktu penginstall saya mencoba merangkum semua command install docker dan portainer ke dalam suatu dockerfile yang bernama ***Makefile*** dan dapat di copy untuk clone pada ubuntu server pada link github saya sbb*:* [*https://github.com/sersanhitam/mantapjoss.git*](https://github.com/sersanhitam/mantapjoss.git)
2. langkah install docker melalui VM ubuntu server

* apt-get install make git



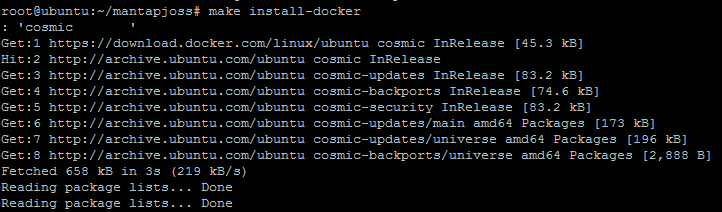
* Pull file dari repository github saya

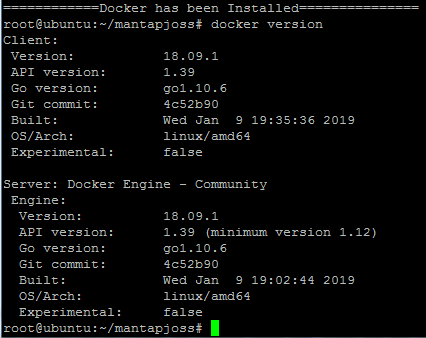
***git clone*** [***https://github.com/sersanhitam/mantapjoss.git***](https://github.com/sersanhitam/mantapjoss.git)

****

* Masuk pada direktori yang sudah di pull dan running file dengan menggunakan perintah ***make install-docker***

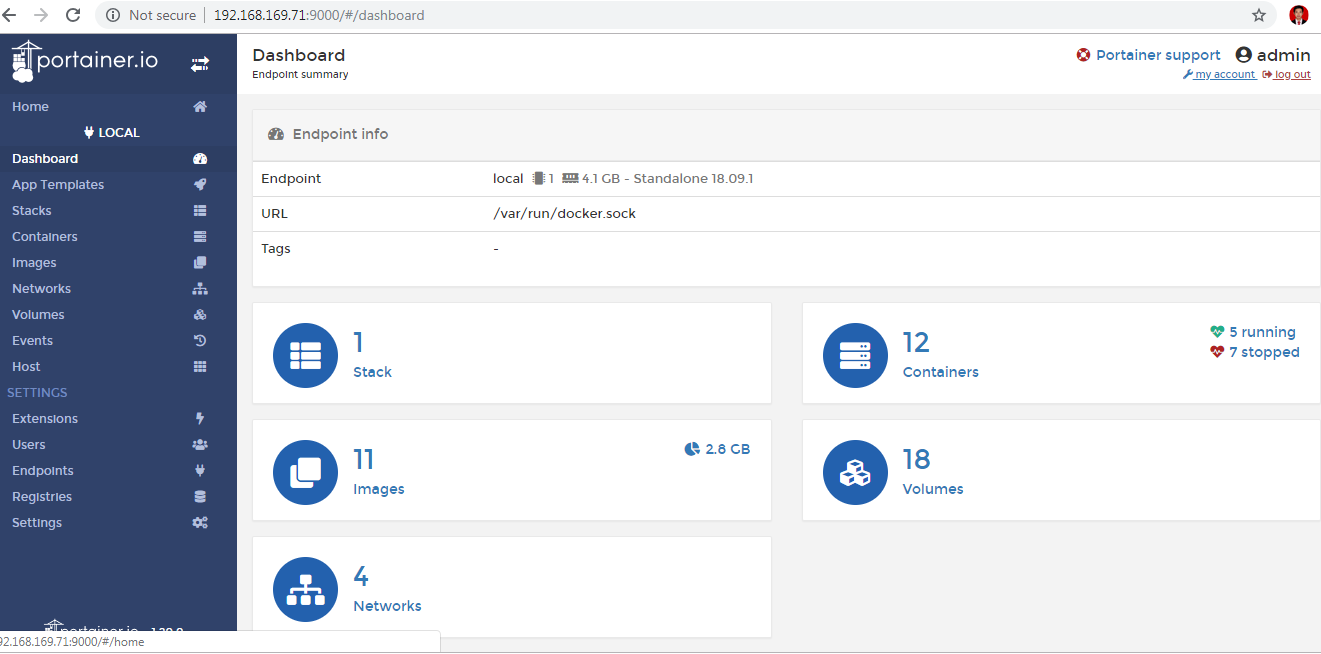




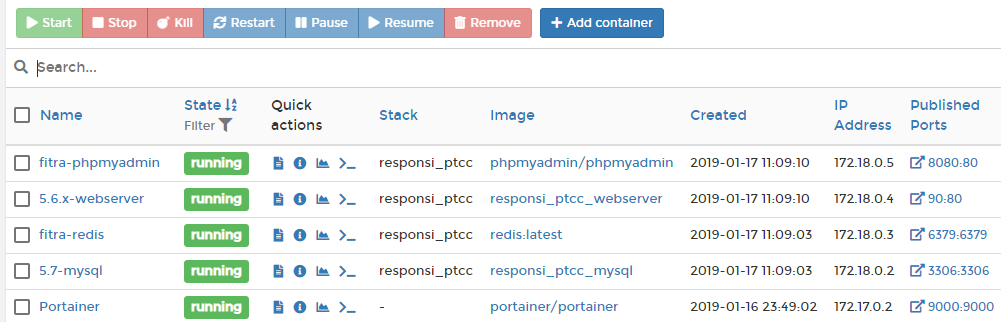


🡪 Docker has been installed pertanda docker sudah terinstall pada ubuntu server

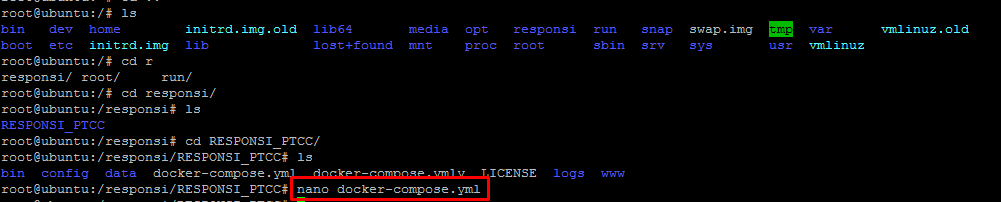
1. Install portainer untuk manage docker & container secara GUI dengan syntax ***make install-portainer***, apabila sudah terinstall maka dapat di akses dengan ip docker dengan ***port :9000***

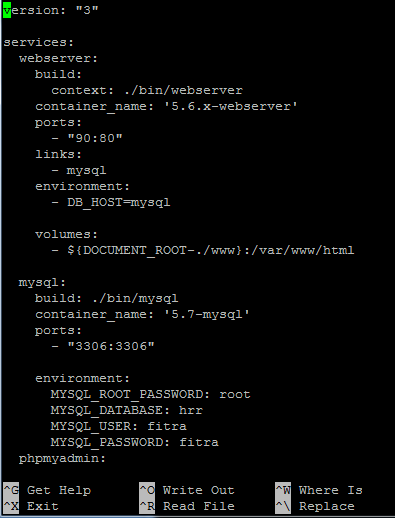


1. Install dan deploy aplikasi yang di butuhkan secara singkat melalui portainer, pada tugas responsi ini saya menginstall **apache** untuk web server, **mysql** untuk database dan tentunya **PHP**.



1. Setelah image sudah ready maka selanjutnya adalah mengkonfigurasi user, password, serta port pada ***docker-compose.yml*** 🡪 yang nantinya berguna untuk menjalankan container secara bersamaan

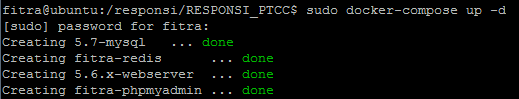




1. Baiklah pada tahap terakhir sebelum build image, perlu di ketahui bahwa sebelum image container di deploy, service container harus dalam posisi stop/off.

***Running docker compose 🡪 docker-compose up –d***

Prosesi akan memakan waktu yang lama dalam rebuild image



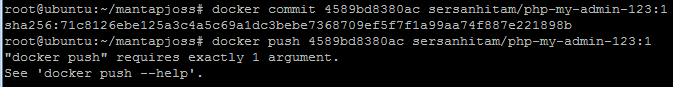
1. **Push image** ke docker hub melalui command line

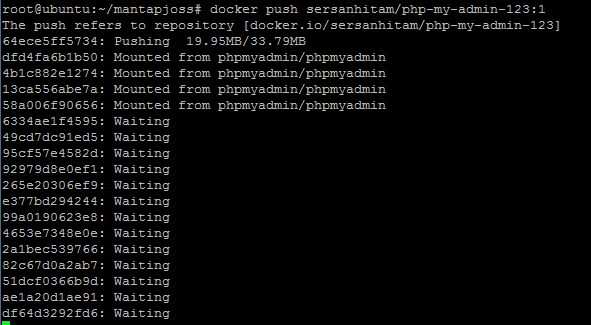
***docker commit (container id) (akun-dockerhub/test-image-123:2)***

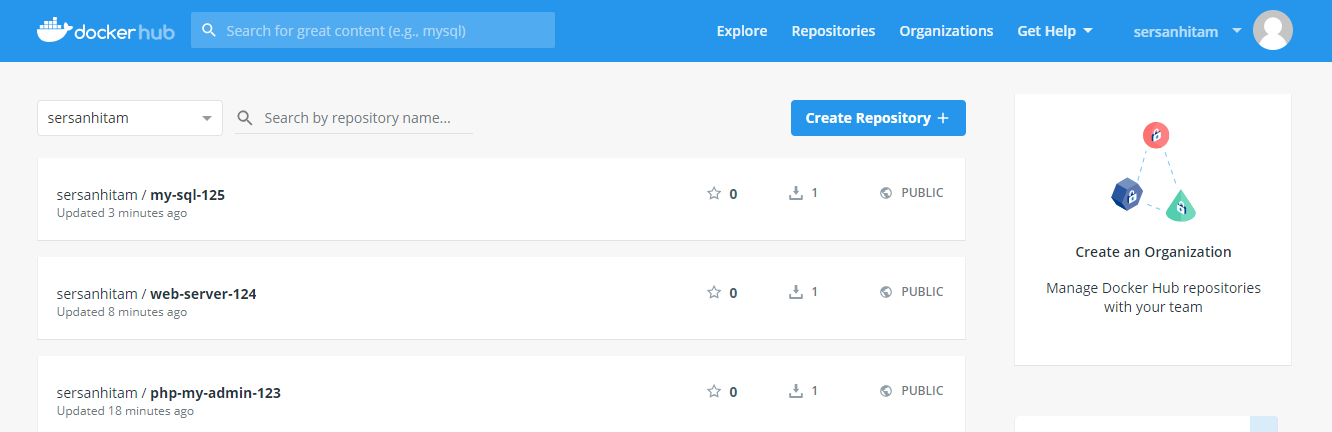
***docker push (akun-dockerhub/test-image-123:2)***

pada tugas responsi ini saya mengupload image ke docker hub sebanyak 3 image saja, yaitu

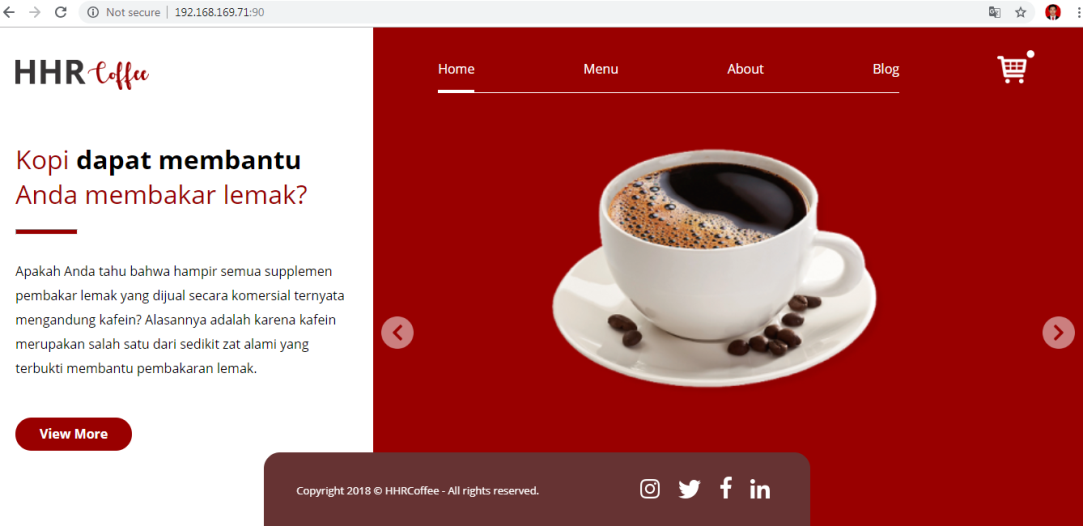
1. **Phpmyadmin**
2. **Webserver**
3. **MySQL**



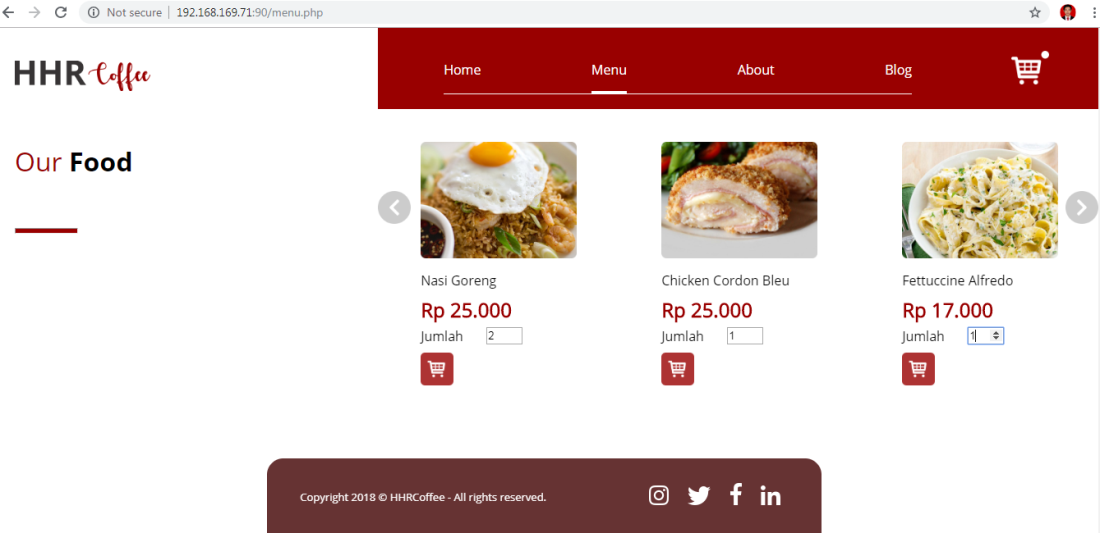


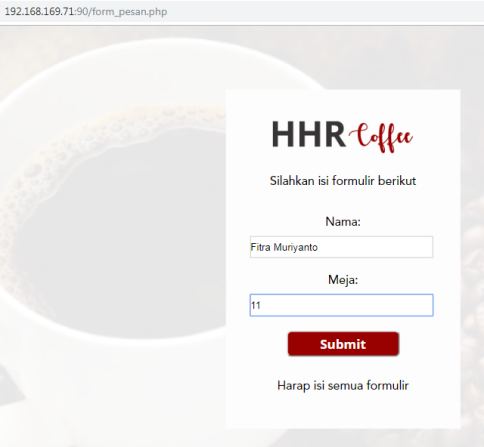


1. Docker yang sudah di runing dengan ***docker-compose up –d*** akan bisa di akses pada url [***http://192.168.169.71:90/***](http://192.168.169.71:90/)



1. Mencoba menambahkan jumlah menu makanan yang disediakan pada web HRR coffe, di lanjutkan klik pada troli/ keranjang

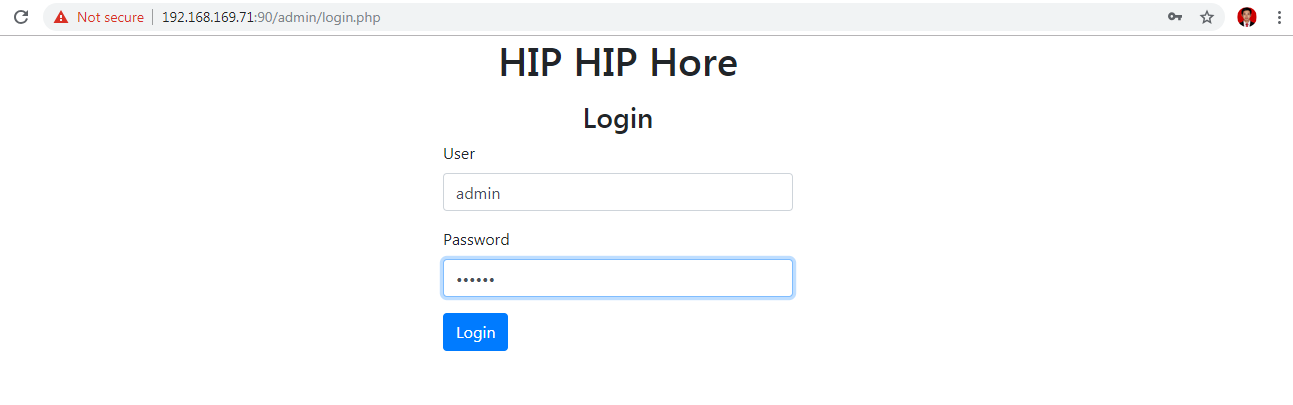


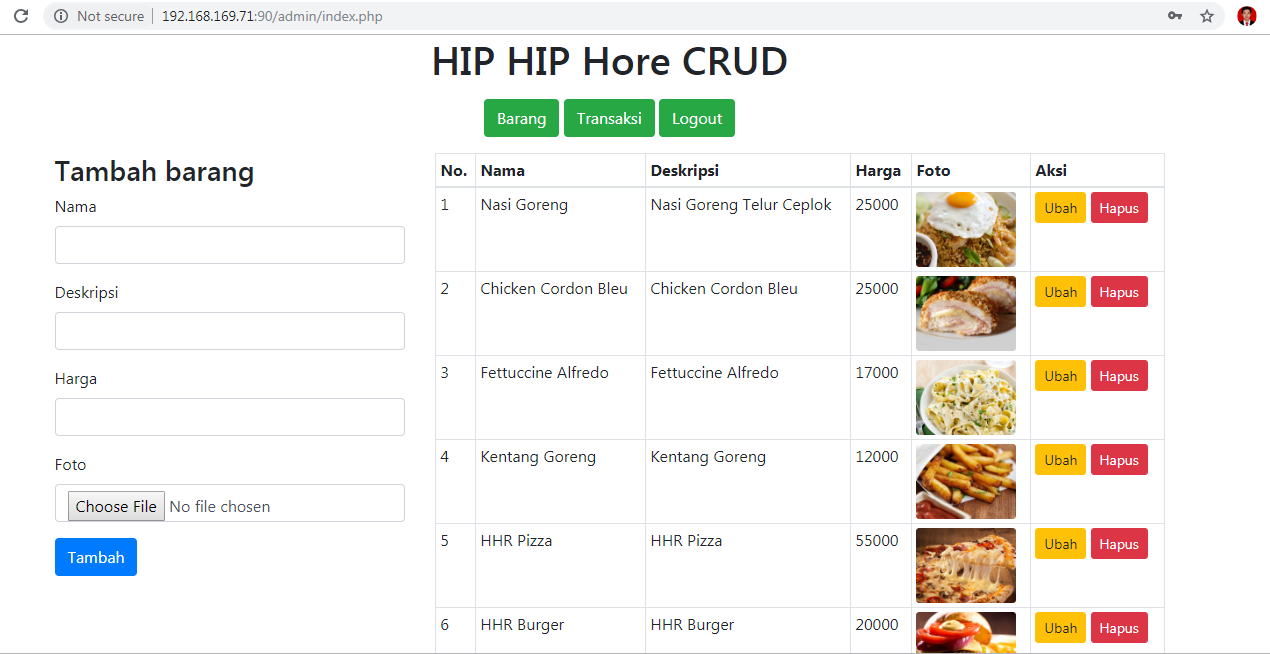
 

1. Untuk login pada halaman **KASIR (admin)** dapat akses url : [***http://192.168.169.71:90/admin***](http://192.168.169.71:90/admin)

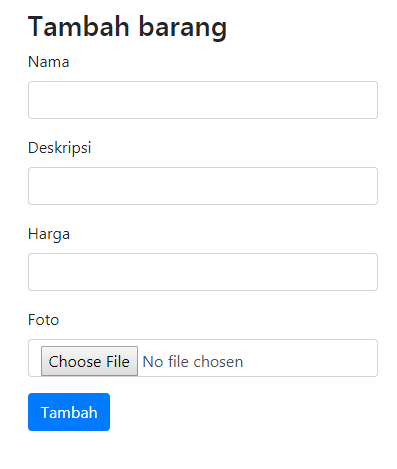
***username : admin***

***password : admin***





1. Menu untuk menambahkan barang berada pada sidebar sebelah kiri, bertujuan untuk menginputkan secara langsung nama barang, deskripsi barang, harga barang maupun foto barang yang akan di tambahkan.



1. Untuk menu yang sudah dapat di tambahkan juga dapat dengan simple di ubah ataupun di hapus.

