

TUGAS MAKALAH

(Instalasi Docker dan Membangun Server Website E-commerce)



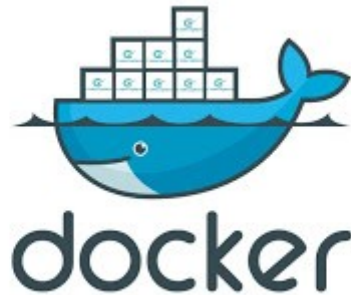
Dosen Pengampu
R. Dimas Adityo ST,MT.

Disusun Oleh

- **Bayu Eka Wibowo (1914321035)**
- **Indra Prasetyo W (1914321005)**
- **Bragas Antero (1914321036)**
- **Ilyasa Octavianto A.D (1914321027)**
- **Badrus Solihin (1914321006)**
- **Andhika Bayu D. (1914321036)**
- **Gusti Rama Dwi N. (1914321017)**

JURUSAN TEKNIK INFORMATIKA
UNIVERSITAS BHAYANGKARA
SURABAYA

2019



Apa itu Docker ??

Docker adalah aplikasi yang menyederhanakan proses, mengelola proses aplikasi dalam container. container memungkinkan Anda menjalankan aplikasi Anda dalam proses yang sumber daya yang terbatas. Container lebih portabel, lebih ramah sumber daya, dan lebih bergantung pada sistem operasi host.

"Kemampuan utama Docker yang baru adalah memungkinkan Anda untuk “mengemas” aplikasi atau serangkaian layanan dalam container. Container Docker adalah instance dari aplikasi berisi semua library dan komponen yang diperlukan agar aplikasi dapat berfungsi. Dari sudut pandang teknis, container adalah sejenis perangkat virtual tereduksi yang berfungsi secara independen dari sistem operasi tempat aplikasi atau layanan tertentu dijalankan" ([Hostinger.co.id](https://www.hostinger.co.id))

Untuk mengikuti tutorial ini anda perlu melakukan hal berikut:

- Install Ubuntu 20.04
- Membuat akun di hub.docker.com
- login menggunakan super user

Instalasi Docker

Update system anda

```
# apt update  
# apt upgrade
```

Install beberapa paket prerequisite

```
# apt install apt-transport-https ca-certificates curl software-properties-common
```

Tambahkan GPG key

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

Tambahkan Repository

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

Update system

```
# apt update
```

Install docker di Ubuntu 20.04

```
# apt install docker-ce
```

Cek docker apakah sudah terinstall

```
# systemctl status docker
root@Anubis:~# sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled;
 vendor preset>
   Active: active (running) since Fri 2020-07-03 09:40:52 WIB;
 2h 11min ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
   Main PID: 1002 (dockerd)
     Tasks: 14
    Memory: 115.6M
    CGroup: /system.slice/docker.service
           └─1002 /usr/bin/dockerd -H fd:// --containerd=/run/
containerd/cont>
```

Menggunakan Docker di Ubuntu 20.04

Setelah melakukan instalasi docker, kita akan belajar menggunakan docker tersebut. susunan perintah docker adalah

```
docker [OPTIONS] COMMAND
```

option

```
Options:
  --config string          Location of client config files (default
                           "/root/.docker")
  -c, --context string     Name of the context to use to connect to the
                           daemon (overrides DOCKER_HOST env var and
                           default context set with "docker context
                           use")
  -D, --debug              Enable debug mode
  -H, --host list          Daemon socket(s) to connect to
  -l, --log-level string   Set the logging level
                           ("debug"|"info"|"warn"|"error"|"fatal")
                           (default "info")
  --tls                    Use TLS; implied by --tlsverify
```

<code>--tlscacert string</code>	Trust certs signed only by this CA (default <code>"/root/.docker/ca.pem"</code>)
<code>--tlscert string</code>	Path to TLS certificate file (default <code>"/root/.docker/cert.pem"</code>)
<code>--tlskey string</code>	Path to TLS key file (default <code>"/root/.docker/key.pem"</code>)
<code>--tlsverify</code>	Use TLS and verify the remote
<code>-v, --version</code>	Print version information and quit

Command :

```

Commands:
  attach      Attach local standard input, output, and error streams to a
              running container
  build       Build an image from a Dockerfile
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's filesystem
  events      Get real time events from the server
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  images      List images
  import      Import the contents from a tarball to create a filesystem image
  info        Display system-wide information
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  login       Log in to a Docker registry
  logout      Log out from a Docker registry
  logs        Fetch the logs of a container
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  ps          List containers
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  rmi         Remove one or more images
  run         Run a command in a new container
  save        Save one or more images to a tar archive (streamed to STDOUT by
              default)
  search      Search the Docker Hub for images
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top         Display the running processes of a container
  unpause    Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  version     Show the Docker version information
  wait        Block until one or more containers stop, then print their exit
              codes

```

Untuk memeriksa apakah dapat mengakses dan mengunduh image dari dokcer hub, ketikan perintah berikut;

```
# docker run hello-world
```

Mencari docker images

```
# docker search debian
root@Anubis:~# docker search debian
NAME
DESCRIPTION
OFFICIAL          AUTOMATED          STARS
ubuntu
based Linux operating sys...  11068          [OK]          Ubuntu is a Debian-
debian
distribution that's compos...  3524          [OK]          Debian is a Linux
arm32v7/debian
distribution that's compos...  66
itscaro/debian-ssh
debian:jessie          28
[OK]
arm64v8/debian          Debian is a Linux
distribution that's compos...  4
```

Download Images dari Docker hub

```
# docker pull debian
```

Cek images

```
# docker images
```

Running docker container

Setelah mendownload images, selanjutnya kita akan membuat container dari images tersebut.

```
# docker run -it debian
root@462605080fa5:/#
```

Menyimpan konfigurasi Container kedalam images

Untuk menyimpan container yang kita setting, kita perlu menyimpan manual dengan mengetikan perintah susunan perintah

```
docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
```

```
# docker commit 462605080fa5 (username-akun-docker)/debian:demo
```

Push image ke Docker hub

Untuk push/mendorong image ke repository docker hub, anda perlu membuat akun di hub.docker.com

```
# docker login -u (username-akun-docker)
```

Susunan perintah untuk push image adalah

```
docker push username-docker/nama-image:tag
```

```
# docker push anubis/debian:demo
```

Tutorial Installasi docker pada Ubuntu 20.04 telah selesai, pada tutorial selanjutnya kita akan mengaplikasikan docker untuk [membuat server CMS wordpress.](#)



Introduction

Docker adalah aplikasi yang menyederhanakan proses, mengelola proses aplikasi dalam container. container memungkinkan Anda menjalankan aplikasi Anda dalam proses yang sumber daya yang terbatas. mereka mirip dengan mesin virtual, tetapi container lebih portabel, lebih ramah sumber daya, dan lebih bergantung pada sistem operasi host.

Untuk detail tentang berbagai komponen container Docker, lihat di [docker overview](#)

Wordpress adalah sebuah aplikasi sumber terbuka (*open source*) yang sangat populer digunakan sebagai mesin blog (*blog engine*). WordPress dibangun dengan bahasa pemrograman PHP dan Database mysql. saat ini wordpress menjadi CMS (Content Management System) paling populer di dunia.

Dalam tutorial ini, Anda akan menginstal wordpress dengan Docker Community Edition (CE) di Debian 10 buster. kita akan membangun lokal server wordpress dengan bind9, Nginx, mysql, Php7.3-fpm

Untuk mengikuti tutorial ini anda perlu melakukan hal berikut:

- [Install docker](#) pada operating sistem anda
- Run images Debian 10 pada docker
- login menggunakan super user

Installing Bind9

BIND9 atau Berkeley Internet Name Domain Versi 9 adalah salah satu software yang biasa digunakan untuk membuat, membangun dan mengatur sebuah DNS (Domain Name Server) pada sistem operasi Linux. pada tutorial kali ini saya akan membuat nama domain *warunkjarkom.com*

Installing bind9

```
# apt update
```

```
# apt install bind9
```

Masuk pada direktori aktif "/etc/bind "

Buat file untuk tempat konfigurasi DNS anda di db.domain (*file forward*) dan db.ip (*file reverse*)

```
# cp db.lokal db.domain
```

```
# cp db.127 db.ip
```

konfigurasi zona domain

```
# nano named.conf
```

tambahkan script seperti dibawah ini

```
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";

zone "warunkjarkom.com"{
type master;
file "/etc/bind/db.domain";
};
zone "2.in-addr.arpa"{
type master;
file "/etc/bind/db.ip";
};
```

Konfigurasi file forward

File forward berfungsi untuk konversi dari DNS ke IP address.

```
# nano db.domain
```

Edit seperti dibawah ini

```
;  
; BIND data file for local loopback interface  
;  
$TTL      604800  
@         IN      SOA      warunkjarkom.com.  
root.warunkjarkom.net. (                2           ; Serial  
                                604800        ; Refresh  
                                86400         ; Retry  
                                2419200       ; Expire  
                                604800 )      ; Negative Cache TTL  
;  
@         IN      NS       warunkjarkom.com.  
@         IN      A        172.17.0.2
```

Konfigurasi file reserve

File reserve berfungsi untuk konversi dari IP address ke DNS.


```
# nano db.ip
```

Edit seperti dibawah ini

```
;  
; BIND reverse data file for local loopback interface  
;  
$TTL      604800  
@         IN      SOA      warunkjarkom.com.  
root.warunkjarkom.com. (   
                        1           ; Serial  
                        604800      ; Refresh  
                        86400       ; Retry  
                        2419200     ; Expire  
                        604800 )    ; Negative Cache TTL  
;  
@         IN      NS       warunkjarkom.com.  
2.0.17   IN      PTR      warunkjarkom.com.
```

Membuat Forwarding DNS

DNS Forwarding adalah suatu cara yang dilakukan oleh DNS server mengubah arah pencarian dari permintaan name query (yang dia sendiri tidak bisa menjawabnya) ke DNS Server lainnya.

karena kita akan membuat lokal server tentunya akan sangat terganggu bilamana saat test server kita tapi tidak bisa Internetan sedangkan DNS lokal kita tidak bisa memetakan permintaan query keluar, untuk itu kita perlu mengalihkan query tersebut ke DNS public.

Edit file " named.conf.options " seperti dibawah ini

```
# nano named.conf.options  
acl goodclients {  
    172.17.0.0/16;  
    localhost;  
    localnets;  
};  
  
options {  
    directory "/var/cache/bind";  
  
    recursion yes;  
    allow-query { goodclients; };  
  
    forwarders {  
        8.8.8.8;  
        8.8.4.4;  
    };  
};
```

```
forward only;

dnssec-enable yes;
dnssec-validation yes;

auth-nxdomain no;      # conform to
RFC1035
listen-on-v6 { any; };
};
```

Restart bind9

```
# service bind9 restart
```

Installing Nginx, PHP7.3, Mysql, Wordpress

Install Nginx

Nginx adalah web server berbasis open source.

```
# apt install nginx
```

Install PHP7.3 beserta modul-modul yang dibutuhkan

```
# apt install php7.3 php7.3-cli php7.3-common php7.3-curl php7.3-gd php7.3-json php7.3-mbstring php7.3-mysql php7.3-xml
```

Pada Debian 10 buster ini kita tidak perlu menambahkan repository untuk install php versi 7.3, namun untuk php versi atasnya seperti 7.4 7.5 perlu menambahkan repository terlebih dahulu

Install mysql

MySQL adalah sebuah database management system (manajemen basis data) menggunakan perintah dasar SQL (*Structured Query Language*) yang cukup terkenal. Database management system (DBMS) MySQL multi pengguna dan multi alur ini sudah dipakai [lebih dari 6 juta pengguna di seluruh dunia.](#)

tambahkan repository untuk mysql-server

```
# apt install gnupg
# cd /tmp
# wget https://dev.mysql.com/get/mysql-apt-config_0.8.13-1_all.deb
# dpkg -i mysql-apt-config*
```

```
# apt update
```

Dengan perintah tersebut repository mysql otomatis ditambahkan.
Install mysql-server

```
# apt install mysql-server
```

Mengamankan mysql server

```
# mysql _secure_installation
```

Terdapat beberapa optional yang muncul, pilih " y/n " sesuai kebutuhan.

konfigurasi Databases untuk wordpress

```
# mysql -u root -p
mysql>CREATE DATABASE wordpress;
mysql>CREATE USER 'syn'@'localhost' identified by 'yourpassword';
mysql>GRANT ALL PRIVILEGES ON wordpress.* TO 'syn'@localhost';
mysql>FLUSH PRIVILEGES;
mysql>EXIT;
```

Install wordpress

Cek paket lalu install wordpress

```
# apt-cache search wordpress
# apt install wordpress
```

pindahkan file wordpress yang awalnya di " usr/share/wordpress " ke " var/www/html/warunkjarkom.com/wordpress "

```
# mv /usr/share/wordpress /var/www/html/warunkjarkom.com/
```

Kemudian, ubah kepemilikan direktori wordpress dengan perintah berikut:

```
# chown -R www-data:www-data
/var/www/html/warunkjarkom.com/wordpress
```

Konfigurasi Nginx untuk wordpress

Selanjutnya, kita perlu membuat file konfigurasi Virtual Host untuk WordPress.

Membuat serta edit seperti dibawah ini file virtual host yang baru

```
# nano /etc/nginx/sites-available/warunk-jarkom
server {
    listen 80;
    root /var/www/html/warunkjarkom.com/wordpress;
    index index.php index.html index.htm;
    server_name warunkjarkom.com;

    access_log /var/log/nginx/wordpress_access.log;
    error_log /var/log/nginx/wordpress_error.log;

    client_max_body_size 64M;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ /\.php$ {
        try_files $uri =404;
        include /etc/nginx/fastcgi_params;
        fastcgi_read_timeout 3600s;
        fastcgi_buffer_size 128k;
        fastcgi_buffers 4 128k;
        fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        fastcgi_pass unix:/run/php/php7.3-fpm.sock;
        fastcgi_index index.php;
    }
}
```

Aktifkan virtual blok yang baru

```
# ln -s /etc/nginx/sites-available/warunk-jarkom /etc/nginx/sites-enabled/
```

Restart Nginx dan Php7.3-fpm

```
# service nginx restart
# service php7.3-fpm restart
```

Membuat autostart Service pada container

Setelah semua service terinstall dengan baik, kita akan membuat service itu otomatis running setelah booting/mesin utama mati.

langkah pertama adalah update container untuk autorun, tambah terminal baru untuk mengupdate docker yang sedang berjalan, lalu ketikkan perintah

```
# docker update --restart unless-stopped 4a66e63d5fe0
```

Kembali pada terminal container docker, kita akan membuat shell script untuk mengumpul semua service yang dibutuhkan agar bisa di start bersamaan

```
# nano /etc/init.d/autostart
```

Tambahkan script berikut

```
#!/bin/bash
#Scripts to start services if not running
ps -ef | grep nginx |grep -v grep > /dev/null
if [ $? != 0 ]
then
    /etc/init.d/nginx start > /dev/null
fi
ps -ef | grep php7.3-fpm |grep -v grep > /dev/null
if [ $? != 0 ]
then
    /etc/init.d/php7.3-fpm start > /dev/null
fi
ps -ef | grep mysql |grep -v grep > /dev/null
if [ $? != 0 ]
then
    /etc/init.d/mysql start > /dev/null
fi
ps -ef | grep bind9 |grep -v grep > /dev/null
if [ $? != 0 ]
then
    /etc/init.d/bind9 start > /dev/null
fi
```

tambahkan hak akses pada script tersebut

```
# chmod +x /etc/init.d/autostart
```

lalu tambahkan symlink ke /etc/rc.d

```
# ln -s /etc/init.d/autostart /etc/rc.d
```

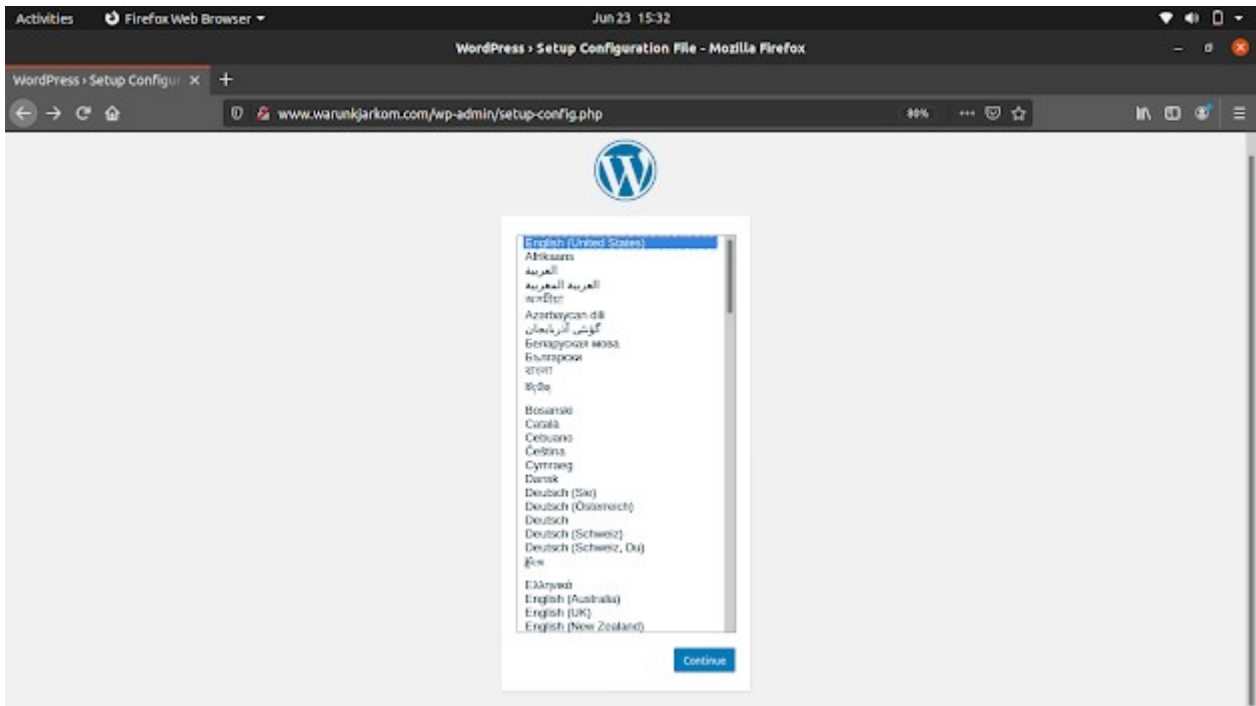
copy script yang kita buat untuk mengantisipasi hal yang tidak terduga

```
# cp /etc/init.d/autostart /var/script/autostart.sh
```

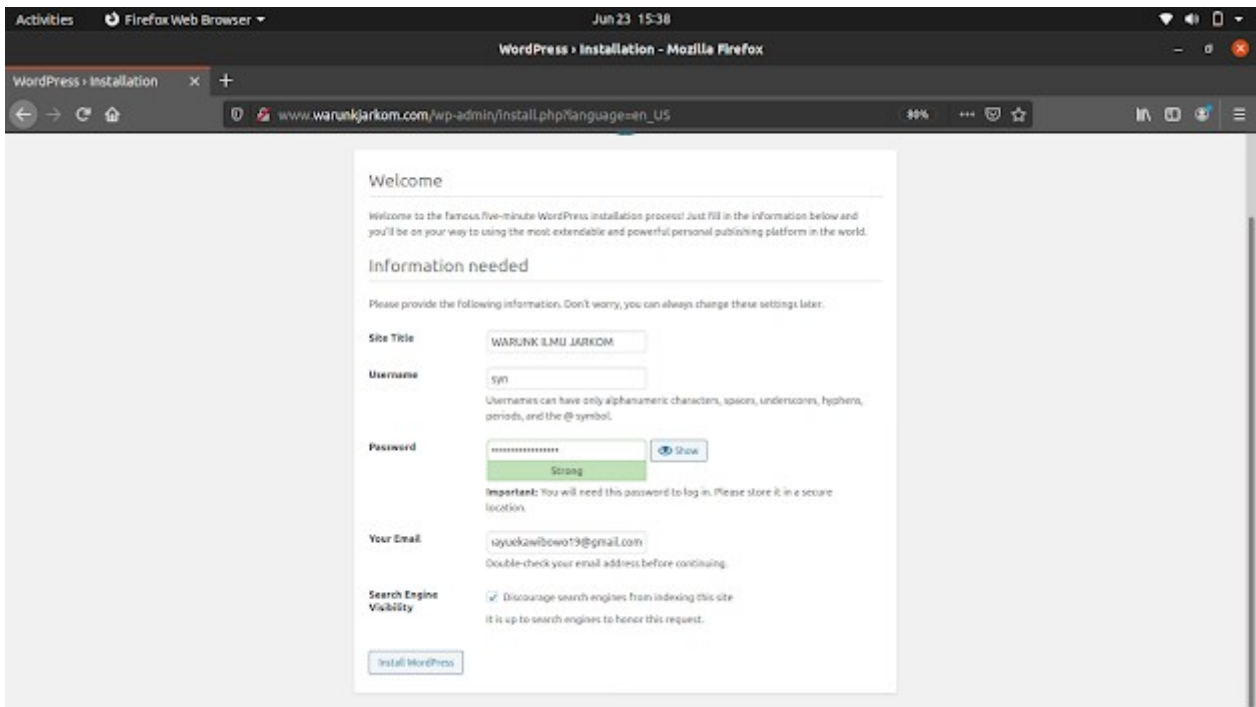
Untuk mengetest script jalan atau tidak, restart container lalu lihat pada `system --status-all`

Akses Wordpress Site

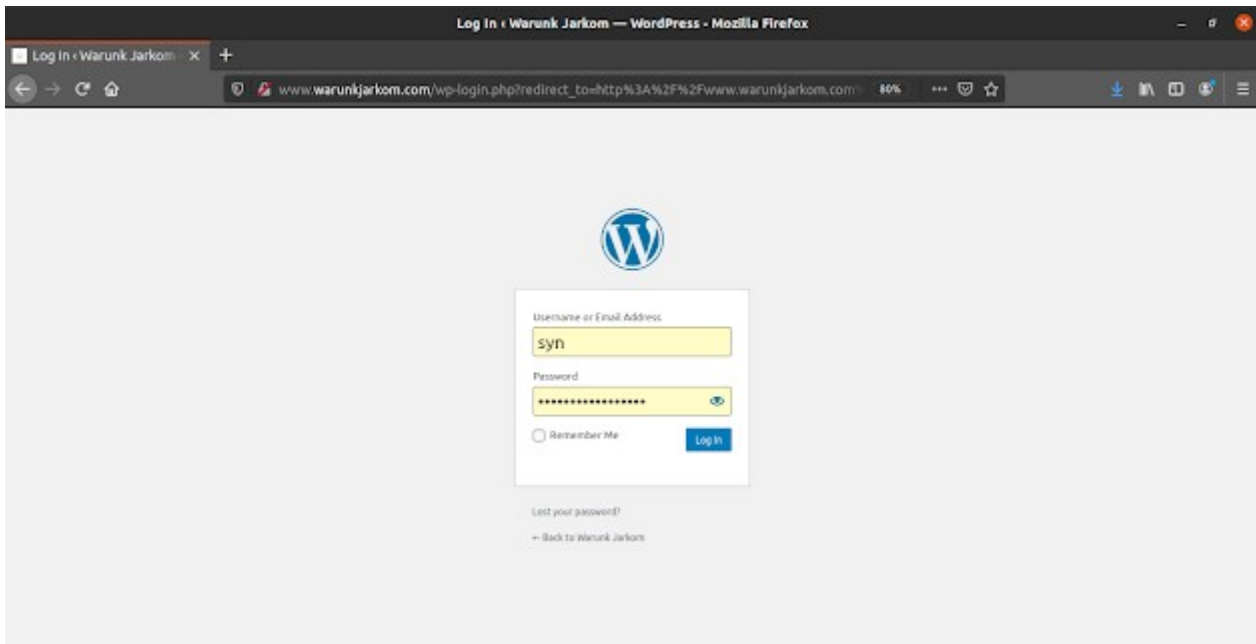
sekarang buka web browser lalu ketikkan URL <http://warunkjarkom.com> maka akan muncul halaman installasi wordpress seperti diberikut;



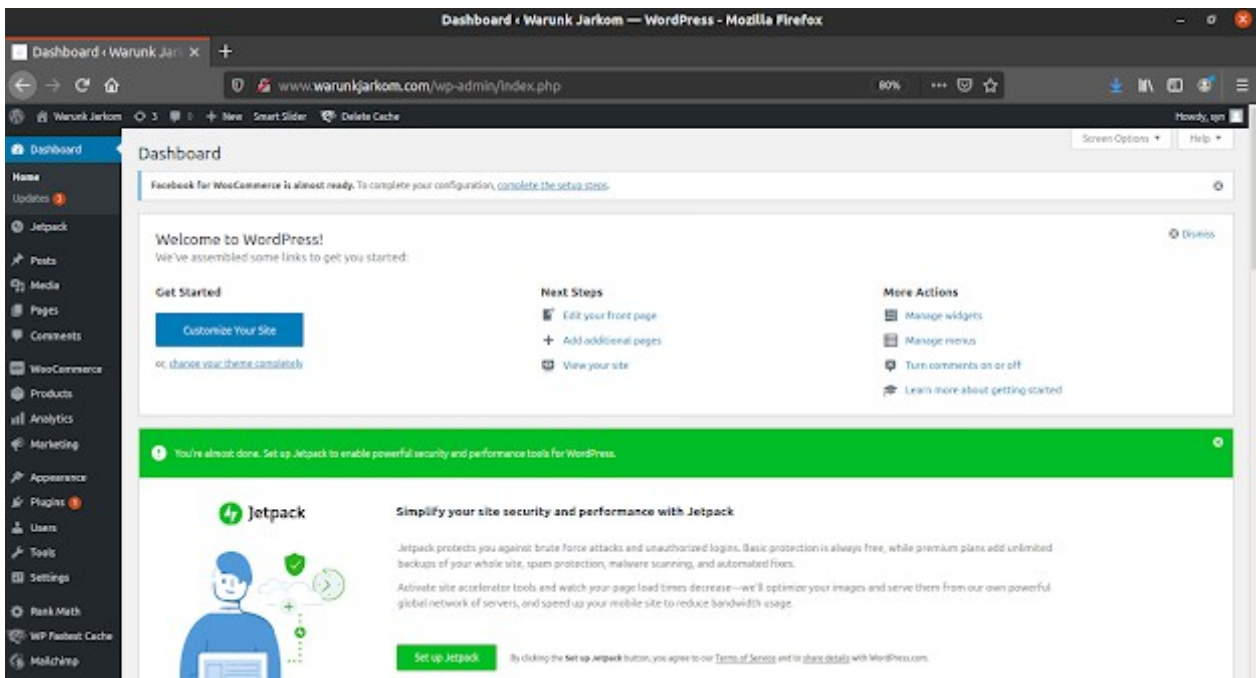
Pilih bahasa yang akan digunakan, lalu klik *continue* akan muncul tampilan berikut;



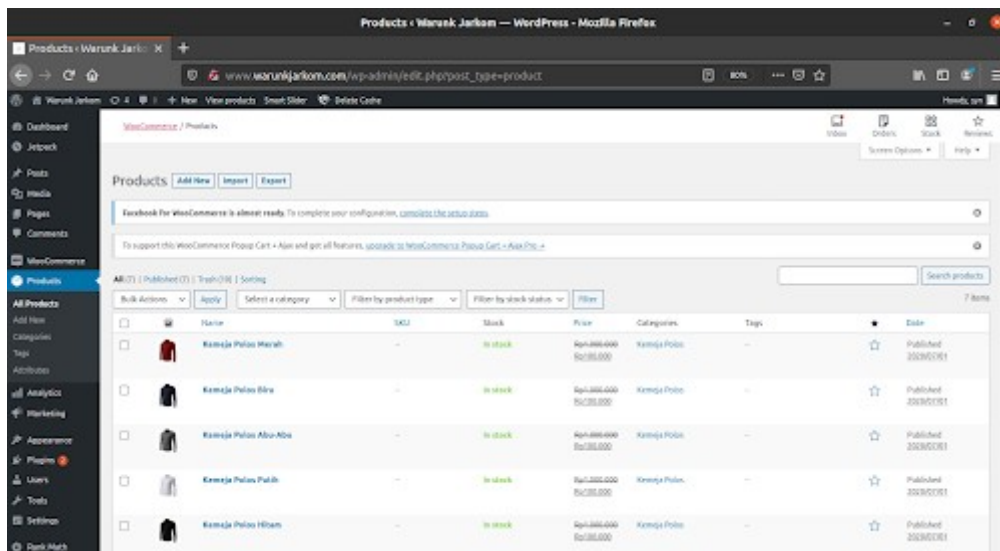
Berikan nama situs WordPress Anda, Nama Pengguna Admin, Kata Sandi, Email, dan akhirnya klik tombol Instal WordPress. Setelah instalasi selesai dengan sukses, lalu klik login



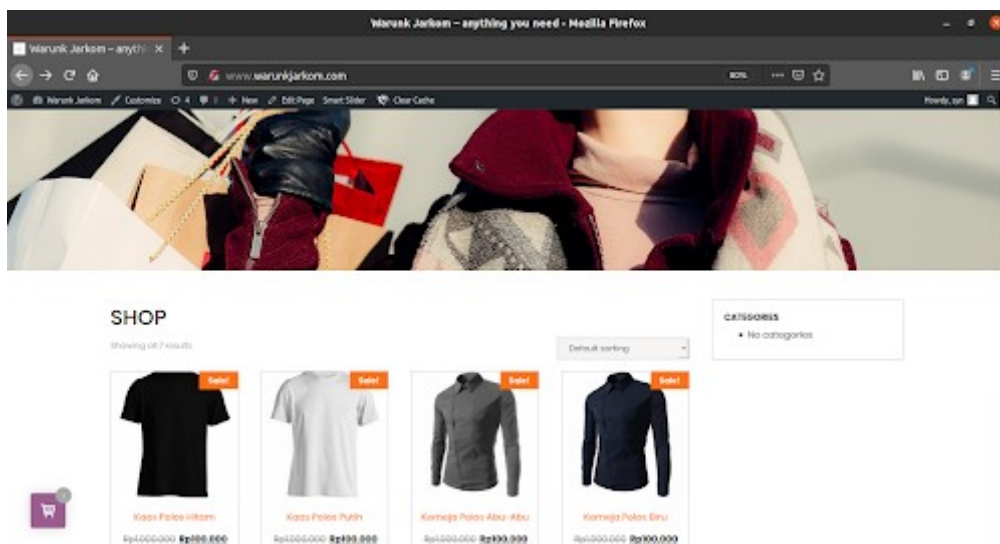
masukan user password yang tadi dibuat untuk masuk ke dashboard wordpress



Instalasi wordpress telah selesai. Di dalam wordpress kita dapat menginstall beberapa plugin untuk keperluan website. yang akan kita install kali ini adalah woocomers. woocomers adalah satu satu plugin untuk membangun sebuah website toko/ecomers, woocomers dapat diinstall pada menu *plugin* pada halaman wp-admin. setelah woocomers terinstall kita dapat menambahkan produk2 pada halaman web yang akan kita buat.



hasil website



pustaka :

- www.wikipedia.org
- www.digitalocean.com
- www.rosehosting.com
- www.geekflare.com
- www.stackoverflow.com
- taufikroom.wordpress.com
- mawarnetwork.blogspot.com
- www.niagahoster.co.id

Share, Connect, and Learn Together