

WRITE UP

Langkah-langkah Menginstall WordPress di Ubuntu Server

Tentu! Berikut adalah langkah-langkah untuk menginstal WordPress di Ubuntu Server, lengkap dengan penjelasan dan output dari setiap perintah yang dijalankan.

Prasyarat

Sebelum menginstal WordPress, pastikan Anda telah menginstal komponen berikut:

Web Server (Apache atau Nginx)

PHP

Database (MySQL atau MariaDB)

Extensions PHP yang diperlukan oleh WordPress

Untuk panduan ini, kita akan menggunakan Apache sebagai web server dan MariaDB sebagai database.

Langkah 1: Memperbarui Paket Sistem

Pertama, kita perlu memperbarui daftar paket dan menginstal pembaruan yang tersedia.

`sudo apt update && sudo apt upgrade -y`

Output:

```
Setting up cryptsetup-initramfs (2:2.7.0-1ubuntu4.1) ...
update-initramfs: deferring update (trigger activated)
Setting up liblvm2cmd2.03:amd64 (2.03.16-3ubuntu3.1) ...
Setting up dmidevd (2:1.02.185-3ubuntu3.1) ...
dm-event.service is a disabled or a static unit not running, not starting it.
Setting up lvm2 (2.03.16-3ubuntu3.1) ...
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...
Processing triggers for udev (255.4-1ubuntu8.4) ...
Processing triggers for debianutils (5.17build1) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for rsyslog (8.2312.0-3ubuntu9) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for initramfs-tools (0.142ubuntu25.2) ...
update-initramfs: Generating /boot/initrd.img-6.8.0-45-generic
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...
/etc/needrestart/restart.d/systemd-manager
systemctl restart multipathd.service udisks2.service

Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service

No containers need to be restarted.

User sessions running outdated binaries:
aliansah @ session #1: login[799]
aliansah @ user manager service: systemd[902]

No VM guests are running outdated hypervisor (qemu) binaries on this
host.
```

Langkah 2: Menginstal Apache

Instal Apache dengan perintah berikut:

```
sudo apt install apache2 -y
```

Output:

```
Unpacking libaprutil1-ldap:amd64 (1.6.3-1.1ubuntu7) ...
Selecting previously unselected package liblua5.4-0:amd64.
Preparing to unpack .../4-liblua5.4-0_5.4.6-3build2_amd64.deb ...
Unpacking liblua5.4-0:amd64 (5.4.6-3build2) ...
Selecting previously unselected package apache2-bin.
Preparing to unpack .../5-apache2-bin_2.4.58-1ubuntu8.4_amd64.deb ...
Unpacking apache2-bin (2.4.58-1ubuntu8.4) ...
Selecting previously unselected package apache2-data.
Preparing to unpack .../6-apache2-data_2.4.58-1ubuntu8.4_all.deb ...
Unpacking apache2-data (2.4.58-1ubuntu8.4) ...
Selecting previously unselected package apache2-utils.
Preparing to unpack .../7-apache2-utils_2.4.58-1ubuntu8.4_amd64.deb ...
Unpacking apache2-utils (2.4.58-1ubuntu8.4) ...
Selecting previously unselected package apache2.
Preparing to unpack .../8-apache2_2.4.58-1ubuntu8.4_amd64.deb ...
Unpacking apache2 (2.4.58-1ubuntu8.4) ...
Selecting previously unselected package ssl-cert.
Preparing to unpack .../9-ssl-cert_1.1.2ubuntu1_all.deb ...
Unpacking ssl-cert (1.1.2ubuntu1) ...
Setting up ssl-cert (1.1.2ubuntu1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ssl-cert.service → /usr/lib/systemd/system/ssl-cert.service.
Setting up libapr1t64:amd64 (1.7.2-3.1ubuntu0.1) ...
Setting up liblua5.4-0:amd64 (5.4.6-3build2) ...
Setting up apache2-data (2.4.58-1ubuntu8.4) ...
Setting up libaprutil1t64:amd64 (1.6.3-1.1ubuntu7) ...
Setting up libaprutil1-ldap:amd64 (1.6.3-1.1ubuntu7) ...
Setting up libaprutil1-dbd-sqlite3:amd64 (1.6.3-1.1ubuntu7) ...
Setting up apache2-utils (2.4.58-1ubuntu8.4) ...
Setting up apache2-bin (2.4.58-1ubuntu8.4) ...
Setting up apache2 (2.4.58-1ubuntu8.4) ...
```

Langkah 3: Menginstal MariaDB

Sekarang, instal MariaDB untuk database:

```
sudo apt install mariadb-server -y
```

Output:

```
Setting up libencode-locale-perl (1.05-3) ...
Setting up libsnappy1v5:amd64 (1.1.10-1build1) ...
Setting up socat (1.8.0-0-4build3) ...
Setting up libio-html-perl (1.004-3) ...
Setting up libmariadb3:amd64 (1:10.11.8-0ubuntu0.24.04.1) ...
Setting up libtimedate-perl (2.3300-2) ...
Setting up pv (1.8.5-2build1) ...
Setting up libfcgi-perl (0.82+ds-3build2) ...
Setting up liburing2:amd64 (2.5-1build1) ...
Setting up liburi-perl (5.27-1) ...
Setting up libdbi-perl:amd64 (1.643-4build3) ...
Setting up libhttp-date-perl (6.06-1) ...
Setting up mariadb-client-core (1:10.11.8-0ubuntu0.24.04.1) ...
Setting up libdbd-mysql-perl:amd64 (4.052-1ubuntu3) ...
Setting up libhtml-parser-perl:amd64 (3.81-1build3) ...
Setting up mariadb-server-core (1:10.11.8-0ubuntu0.24.04.1) ...
Setting up libhttp-message-perl (6.45-1ubuntu1) ...
Setting up mariadb-client (1:10.11.8-0ubuntu0.24.04.1) ...
Setting up libcgi-pm-perl (4.63-1) ...
Setting up libhtml-template-perl (2.97-2) ...
Setting up mariadb-server (1:10.11.8-0ubuntu0.24.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
Setting up mariadb-plugin-provider-bzip2 (1:10.11.8-0ubuntu0.24.04.1) ...
Setting up mariadb-plugin-provider-lzma (1:10.11.8-0ubuntu0.24.04.1) ...
Setting up mariadb-plugin-provider-lzo (1:10.11.8-0ubuntu0.24.04.1) ...
Setting up mariadb-plugin-provider-lz4 (1:10.11.8-0ubuntu0.24.04.1) ...
Setting up libcgi-fast-perl (1:2.17-1) ...
Setting up mariadb-plugin-provider-innodb (1:10.11.8-0ubuntu0.24.04.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for mariadb-server (1:10.11.8-0ubuntu0.24.04.1) ...
Scanning processes...
Scanning candidates...
Scanning linux images...
```

Langkah 4: Mengamankan Instalasi MariaDB

Setelah menginstal MariaDB, jalankan skrip pengamanan:

```
sudo mysql_secure_installation
```

Ikuti petunjuk untuk mengatur kata sandi root dan menghapus pengguna anonim. Berikut adalah contoh output yang mungkin Anda lihat:

Output:

```
aliensah@aliensah-server:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
... Success!

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

```
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!

By default, MariaDB 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? [Y/n] 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? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

Langkah 5: Menginstal PHP dan Extensions yang Diperlukan

Instal PHP dan beberapa extensions yang diperlukan oleh WordPress:

```
sudo apt install php libapache2-mod-php php-mysql php-xml php-mbstring php-curl php-zip -y
```

Output:

```
Creating config file /etc/php/8.3/mods-available/curl.ini with new version
Setting up php8.3-cli (8.3.6-0ubuntu0.24.04.2) ...
update-alternatives: using /usr/bin/php8.3 to provide /usr/bin/php (php) in auto mode
update-alternatives: using /usr/bin/phar8.3 to provide /usr/bin/phar (phar) in auto mode
update-alternatives: using /usr/bin/phar.phar8.3 to provide /usr/bin/phar.phar (phar.phar) in auto mode

Creating config file /etc/php/8.3/cli/php.ini with new version
Setting up php-mbstring (2:8.3+93ubuntu2) ...
Setting up php8.3-zip (8.3.6-0ubuntu0.24.04.2) ...

Creating config file /etc/php/8.3/mods-available/zip.ini with new version
Setting up libapache2-mod-php8.3 (8.3.6-0ubuntu0.24.04.2) ...

Creating config file /etc/php/8.3/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php8.3
Setting up php-zip (2:8.3+93ubuntu2) ...
Setting up php-curl (2:8.3+93ubuntu2) ...
Setting up php8.3 (8.3.6-0ubuntu0.24.04.2) ...
Setting up libapache2-mod-php (2:8.3+93ubuntu2) ...
Setting up php (2:8.3+93ubuntu2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
Processing triggers for php8.3-cli (8.3.6-0ubuntu0.24.04.2) ...
Processing triggers for libapache2-mod-php8.3 (8.3.6-0ubuntu0.24.04.2) ...
Scanning processes...
Scanning candidates...
Scanning linux images...
```

Langkah 6: Menginstal WordPress

Unduh WordPress: Gunakan wget untuk mengunduh paket WordPress terbaru.

```
wget https://wordpress.org/latest.tar.gz
```

Output:

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
aliensah@aliensah-server:~$ wget https://wordpress.org/latest.tar.gz
--2024-10-15 11:49:04-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24640061 (23M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz      100%[=====>]  23.50M  2.01MB/s   in 16s

2024-10-15 11:49:21 (1.50 MB/s) - 'latest.tar.gz' saved [24640061/24640061]
```

Ekstrak File WordPress: Ekstrak file yang telah diunduh:

tar -xvzf latest.tar.gz

Output:

```
wordpress/
wordpress/xmlrpc.php
wordpress/wp-blog-header.php
wordpress/readme.html
wordpress/wp-signup.php
wordpress/index.php
wordpress/wp-cron.php
wordpress/wp-config-sample.php
wordpress/wp-login.php
wordpress/wp-settings.php
wordpress/license.txt
wordpress/wp-content/
wordpress/wp-content/themes/
wordpress/wp-content/themes/twentytwentythree/
wordpress/wp-content/themes/twentytwentythree/theme.json
wordpress/wp-content/themes/twentytwentythree/parts/
wordpress/wp-content/themes/twentytwentythree/parts/footer.html
wordpress/wp-content/themes/twentytwentythree/parts/comments.html
wordpress/wp-content/themes/twentytwentythree/parts/header.html
wordpress/wp-content/themes/twentytwentythree/parts/post-meta.html
wordpress/wp-content/themes/twentytwentythree/patterns/
wordpress/wp-content/themes/twentytwentythree/patterns/hidden-404.php
wordpress/wp-content/themes/twentytwentythree/patterns/post-meta.php
wordpress/wp-content/themes/twentytwentythree/patterns/hidden-no-results.php
wordpress/wp-content/themes/twentytwentythree/patterns/call-to-action.php
wordpress/wp-content/themes/twentytwentythree/patterns/footer-default.php
wordpress/wp-content/themes/twentytwentythree/patterns/hidden-comments.php
wordpress/wp-content/themes/twentytwentythree/styles/
wordpress/wp-content/themes/twentytwentythree/styles/sherbet.json
wordpress/wp-content/themes/twentytwentythree/styles/grapes.json
wordpress/wp-content/themes/twentytwentythree/styles/canary.json
wordpress/wp-content/themes/twentytwentythree/styles/electric.json
wordpress/wp-content/themes/twentytwentythree/styles/pitch.json
```

Pindahkan File ke Direktori Web: Pindahkan file WordPress ke direktori web Apache:

```
sudo mv wordpress/* /var/www/html/
```

Set Permission: Atur hak akses yang sesuai untuk direktori WordPress:

```
sudo chown -R www-data:www-data /var/www/html/
```

```
sudo chmod -R 755 /var/www/html/
```

Langkah 7: Membuat Database untuk WordPress

Masuk ke MariaDB untuk membuat database dan pengguna untuk WordPress:

```
sudo mysql -u root -p
```

Setelah masuk, jalankan perintah berikut:

```

Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.11.8-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'lianlian';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'wpuser'@'localhost';
Query OK, 0 rows affected (0.069 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> EXIT
Bye

```

Langkah 8: Mengonfigurasi WordPress

Salin File Konfigurasi: Salin file konfigurasi default WordPress.

```
cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
```

Edit File Konfigurasi: Edit file wp-config.php untuk menambahkan detail database.

```
nano /var/www/html/wp-config.php
```

Gantilah bagian berikut dengan detail database Anda:

```

define('DB_NAME', 'wordpress');

define('DB_USER', 'wpuser');

define('DB_PASSWORD', 'kata_sandi_wpuser');

```

Langkah 9: Mengakses WordPress

Buka browser dan masukkan alamat IP server Anda atau `http://your_server_ip`. Anda akan melihat halaman pengaturan WordPress. Ikuti petunjuk untuk menyelesaikan instalasi. Maka outputnya akan seperti ini



ubuntu

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

Ini outputnya masih apache2 sedangkan yang kita inginkan adalah outputnya yaitu layanan dari wordpress nya. Maka lakukan cara dibawah ini

1. Hapus dulu semua file dan direktory di dalam direktory `/var/www/html`

```
aliensah@aliensah-server:~$ cd /var/www/html
aliensah@aliensah-server:/var/www/html$ ls
index.html      wp-comments-post.php  wp-load.php
index.php       wp-config.php         wp-login.php
license.txt     wp-config-sample.php  wp-mail.php
readme.html    wp-content            wp-settings.php
wp-activate.php wp-cron.php           wp-signup.php
wp-admin        wp-includes           wp-trackback.php
wp-blog-header.php wp-links-opml.php     xmlrpc.php
aliensah@aliensah-server:/var/www/html$ cd ..
aliensah@aliensah-server:/var/www$ sudo rm -rf html/*
aliensah@aliensah-server:/var/www$ cd html
aliensah@aliensah-server:/var/www/html$ ls
aliensah@aliensah-server:/var/www/html$ cd ~
```

Gunakan cara tersebut supaya lebih cepat dalam proses penghapusan file dan direktory

2. Kita download dulu wordpressnya menggunakan wget

```
aliensah@aliensah-server:~$ wget https://wordpress.org/latest.zip
--2024-10-15 12:00:16-- https://wordpress.org/latest.zip
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26139871 (25M) [application/zip]
Saving to: 'latest.zip'

latest.zip      100%[=====>] 24.93M  1.74MB/s   in 42s

2024-10-15 12:00:59 (611 KB/s) - 'latest.zip' saved [26139871/26139871]
```

3. Jika sudah kita unzip file tersebut

```
aliensah@aliensah-server:~$ unzip latest.zip
Archive: latest.zip
  inflating: wordpress/xmlrpc.php
  inflating: wordpress/wp-blog-header.php
  inflating: wordpress/readme.html
  inflating: wordpress/wp-signup.php
  inflating: wordpress/index.php
  inflating: wordpress/wp-cron.php
  inflating: wordpress/wp-config-sample.php
  inflating: wordpress/wp-login.php
  inflating: wordpress/wp-settings.php
  inflating: wordpress/license.txt
  creating: wordpress/wp-content/
  creating: wordpress/wp-content/themes/
  creating: wordpress/wp-content/themes/twentytwentythree/
  inflating: wordpress/wp-content/themes/twentytwentythree/theme.json
  creating: wordpress/wp-content/themes/twentytwentythree/parts/
  inflating: wordpress/wp-content/themes/twentytwentythree/parts/footer.html
  inflating: wordpress/wp-content/themes/twentytwentythree/parts/comments.html
  inflating: wordpress/wp-content/themes/twentytwentythree/parts/header.html
  inflating: wordpress/wp-content/themes/twentytwentythree/parts/post-meta.html
  creating: wordpress/wp-content/themes/twentytwentythree/patterns/
  inflating: wordpress/wp-content/themes/twentytwentythree/patterns/hidden-404.php
  inflating: wordpress/wp-content/themes/twentytwentythree/patterns/post-meta.php
  inflating: wordpress/wp-content/themes/twentytwentythree/patterns/hidden-no-results.php
  inflating: wordpress/wp-content/themes/twentytwentythree/patterns/call-to-action.php
  inflating: wordpress/wp-content/themes/twentytwentythree/patterns/footer-default.php
  inflating: wordpress/wp-content/themes/twentytwentythree/patterns/hidden-comments.php
  creating: wordpress/wp-content/themes/twentytwentythree/styles/
  inflating: wordpress/wp-content/themes/twentytwentythree/styles/sherbet.json
  inflating: wordpress/wp-content/themes/twentytwentythree/styles/grapes.json
  inflating: wordpress/wp-content/themes/twentytwentythree/styles/canary.json
  inflating: wordpress/wp-content/themes/twentytwentythree/styles/electric.json
  inflating: wordpress/wp-content/themes/twentytwentythree/styles/pitch.json
```

Jika tidak bisa menggunakan perintah unzip, ini biasanya terjadi karena command unzip belum terinstall di ubuntu kita, maka kita perlu menginstallnya terlebih dahulu dengan cara sebagai berikut:

sudo apt-get install unzip

4. Selanjutnya kita copy file wordpress yang sudah kita unzip tadi ke dalam directory /var/www/html

```
aliensah@aliensah-server:~$ ls
latest.tar.gz latest.zip wordpress
aliensah@aliensah-server:~$ sudo cp -r wordpress /var/www/html
```

5. Setelah itu kita masuk ke directory /var/www/html/wordpress
6. Selanjutnya kita edit confignya menggunakan perintah
sudo nano wp-config-sample.php


```
GNU nano 7.2 wp-config-sample.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
 *
 * @package WordPress
 */

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'database_name_here' );

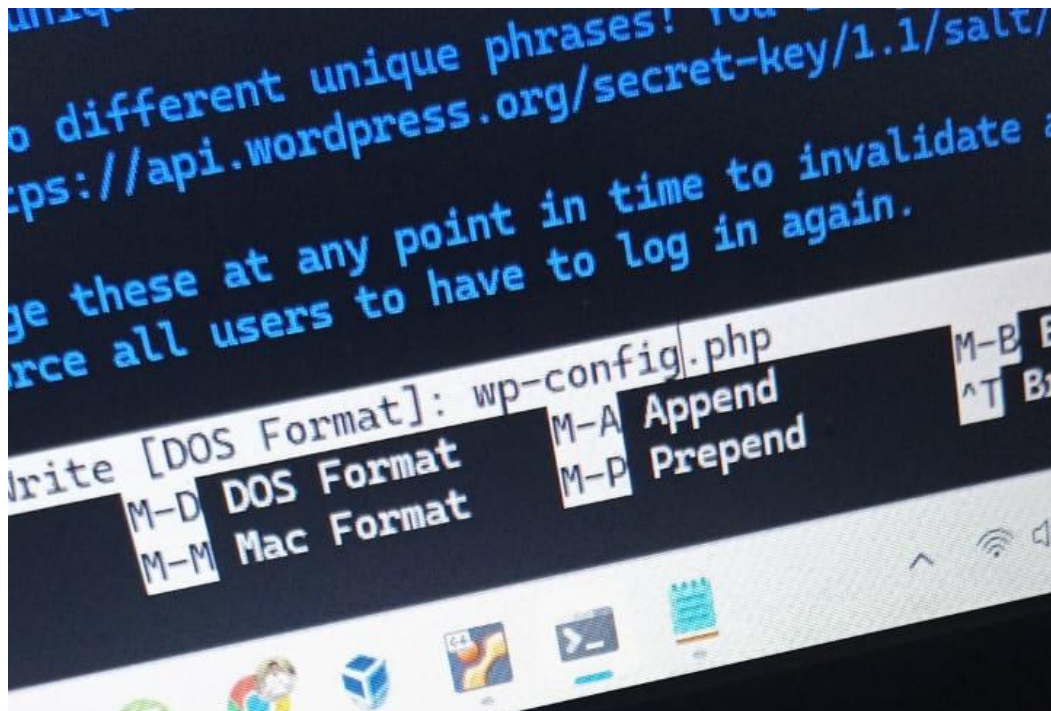
/** Database username */
define( 'DB_USER', 'username_here' );

/** Database password */
define( 'DB_PASSWORD', 'password_here' );

/** Database hostname */
```

Ganti sesuai dengan keinginan

Dan jangan lupa sebelum save config kita, kita rubah nama filenya yang semula **wp-config-sample.php** menjadi **wp-config.php**



7. Langkah selanjutnya jalan perintah `sudo nano /etc/apache2/sites-available/000-default.conf`

```
GNU nano 7.2 /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/wordpress

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Tambahkan pada DocumentRoot /var/www/html/wordpress

8. Setelah semua telah terkonfigurasi dengan benar, kita restart apache2 nya

```
aliensah@aliensah-server:/var/www/html/wordpress$ sudo systemctl restart apache2
```

Langkah 10: Menyelesaikan Instalasi

Ikuti langkah-langkah di halaman instalasi WordPress untuk menyelesaikan pengaturan, termasuk memasukkan judul situs, nama pengguna, dan kata sandi admin.

1. Buka browser dan ketikkan http://ipserver

10.20.2.30/wp-admin/install.php

Informasi yang dibutuhkan

Harap berikan informasi berikut. Jangan khawatir, Anda selalu dapat mengubah pengaturan ini nanti.

Judul Situs:

Nama belakang:

Nama pengguna hanya dapat memiliki karakter alfanumerik, spasi, garis bawah, tanda hubung, titik, dan simbol @.

Kata sandi:

Kuat

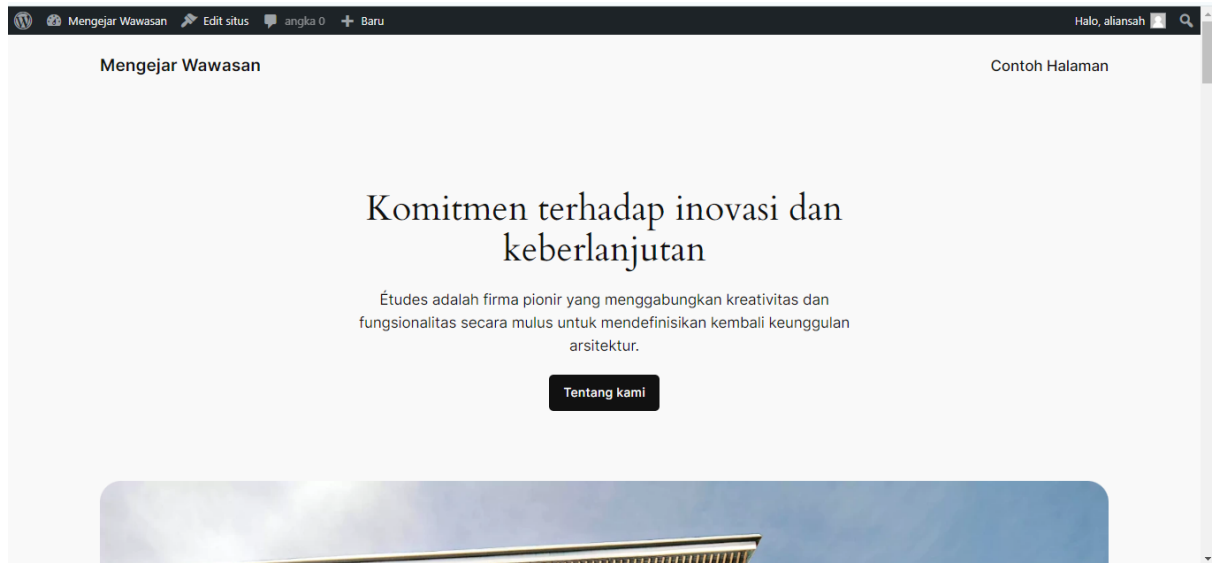
Penting: Anda memerlukan kata sandi ini untuk masuk. Harap simpan di tempat yang aman.

Email Anda:

Periksa kembali alamat email Anda sebelum melanjutkan.

Visibilitas mesin pencari: ☐ Cegah mesin pencari mengindeks situs ini
Terserah pada mesin pencari untuk menghormati permintaan ini.

Maka akan keluar output seperti pada gambar dan ini bisa disesuaikan dengan keinginan kalian



Dan ini adalah akhir jika semuanya sudah terkonfigurasi dengan benar

Kesimpulan

Setelah mengikuti langkah-langkah di atas, Anda seharusnya sudah berhasil menginstal WordPress di server Ubuntu