

LAPORAN TUGAS BESAR DAN DOKUMENTASI CLO 2-4

KEAMANAN SISTEM S1 TEKNIK KOMPUTER

TK-46-05



Disusun oleh :

Muhammad Hafidz Darul Quro

(1103223052)

KEAMANAN SISTEM S1 TEKNIK KOMPUTER

PROGRAM STUDI TEKNIK KOMPUTER

FAKULTAS TEKNIK ELEKTRO

UNIVERSITAS TELKOM

2025

KATA PENGANTAR

Segala puji dan syukur kami panjatkan kepada Tuhan Yang Maha Esa atas segala Rahmat dan Hidayah-Nya, sehingga kami dapat menyelesaikan tugas CLO dua hingga empat ini dengan baik, tepat waktu, dan tanpa kendala yang berarti.

Kami juga ingin mengucapkan terimakasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan selama proses pengerjaan tugas ini.

Kami menyadari bahwa dalam penyelesaian proyek ini masih banyak kekurangan. Oleh karena itu, kami dengan rendah hati memohon maaf dan siap menerima kritik serta saran yang konstruktif dari pembaca demi perbaikan di masa mendatang.

Demikian tugas CLO 2 dua hingga empat ini saya susun, semoga dapat memberikan manfaat bagi semua pihak, terutama bagi kami sebagai penulis. Terimakasih atas perhatian dan kerjasamanya.

Bandung, Mei 2025

Muhammad Hafidz Darul Quro

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR TABEL	iii
DAFTAR GAMBAR	iv
BAB I	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	1
1.3. Tujuan.....	1
BAB II.....	2
TUTORIAL DAN LANGKAH KONFIGURASI CLO 2 PENGAMANAN WEB	2
1. Instalasi VM	2
2. Download OS Linux.....	3
3. Konfigurasi IP Address	7
4. Instalasi Web Server dan Database	9
5. Instalasi PHP	11
6. Persiapan Database.....	12
7. Aplikasi Web Development.....	14
8. Pengamanan Web	22
BAB III	29
TUTORIAL DAN LANGKAH KONFIGURASI CLO 3 PENGAMANAN JARINGAN.....	29
1. Instalasi UFW.....	29
2. Instalasi Snort 3	30
3. Instalasi PulledPork.....	32
4. Menjalankan Snort pada mode IDS	36
5. Uji Coba Testing.....	37
BAB IV	42
SUMBER PACKAGE INSTALASI & REFERENSI.....	42

DAFTAR TABEL

Table 1 Skenario dan Harapan Hasil.....	37
---	----

DAFTAR GAMBAR

Gambar 1 Instalasi VM	2
Gambar 2 Halaman awal VM	2
Gambar 3 Download Ubuntu Desktop	3
Gambar 4 Konfigurasi Awal Ubuntu Desktop	3
Gambar 5 Konfigurasi username dan password VM	4
Gambar 6 Konfigurasi Memory dan Processor VM	5
Gambar 7 Konfigurasi Storage	6
Gambar 8. OS Linux Siap Dipakai	6
Gambar 9 Konfigurasi .yaml untuk IP Address	7
Gambar 10 Cek isi .yaml untuk konfigurasi IP	8
Gambar 11 Verifikasi IP Address	8
Gambar 12 Upgrade Package Ubuntu	9
Gambar 13 Instalasi NGINX	9
Gambar 14 Enable NGINX	10
Gambar 15 Cek Status NGINX	10
Gambar 16 Instalasi mysql-server	10
Gambar 17 Cek Status mysql	10
Gambar 18 Mengamankan mysql	11
Gambar 19 Instalasi PHP	11
Gambar 20 Enable PHP	12
Gambar 21 Cek Status PHP	12
Gambar 22 Masuk ke root mysql	12
Gambar 23 CREATE DATABASE phplogin	13
Gambar 24 CREATE USER mysql	13
Gambar 25 GRANT ALL PRIVILEGES mysql	13
Gambar 26 FLUSH PRIVILEGES mysql	13
Gambar 27 USE phplogin mysql	13
Gambar 28 CREATE TABLE users mysql	13
Gambar 29 Menampilkan Table users	14
Gambar 30 Modifikasi Akses phplogin	14
Gambar 31 Konfigurasi dashboard.php	16
Gambar 32 Konfigurasi db_connect.php	17
Gambar 33 Konfigurasi index.php	17
Gambar 34 Konfigurasi login.php	20
Gambar 35 Konfigurasi logout.php	21
Gambar 36 Konfigurasi register.php	22
Gambar 37 Implementasi Salting dan Hash	26
Gambar 38 Implementasi Salting pada Client-Side	26
Gambar 39 Implementasi perlindungan dari SQL Injection	27
Gambar 40 Implementasi perlindungan dari SQL Injection	27
Gambar 41 Implementasi perlindungan dari SQL Injection	27
Gambar 42 Implementasi perlindungan dari SQL Injection	27
Gambar 43 Implementasi perlindungan dari Buffer Overflow	27
Gambar 44 Implementasi perlindungan dari Buffer Overflow	27

Gambar 45 Implementasi perlindungan dari Buffer Overflow	27
Gambar 46 Implementasi perlindungan dari XSS	28
Gambar 47 Implementasi perlindungan dari XSS	28
Gambar 48 Instalasi ufw	29
Gambar 49 Allow web app via ufw	29
Gambar 50 Memblokir port 8080 via ufw	29
Gambar 51 Limit brute force SSH via ufw	29
Gambar 52 Mengaktifkan Firewall via ufw	30
Gambar 53 Cek status ufw	30
Gambar 54 Membuat direktori sumber	31
Gambar 55 Instalasi DAQ	31
Gambar 56 Instalasi Snort 3.7.4	31
Gambar 57 Instalasi prerequisites	32
Gambar 58 Instalasi PulledPork	32
Gambar 59 Unzip master.zip	33
Gambar 60 Berpindah direktori pulledpork-master	33
Gambar 61 Copy pulledpork.pl	33
Gambar 62 Memberikan akses ke pulledpork.pl	33
Gambar 63 Oinkcodes Snort.txt	34
Gambar 64 rule_url di pulledpork.conf	34
Gambar 65 Ganti .conf menjadi .lua	34
Gambar 66 Cek instalasi PulledPork	34
Gambar 67 Isi local.rules	35
Gambar 68 Ubah isi ips warning_snort.lua	35
Gambar 69 Jalankan update pulldepork	36
Gambar 70 Menjalankan snort pada mode IDS	36
Gambar 71 Hasil ping flood	37
Gambar 72 Hasil Scan nmap	38
Gambar 73 Cek Status Blokir port 8080	38
Gambar 74 Halaman Login.php	39
Gambar 75 Halaman register.php	40
Gambar 76 Login	40
Gambar 77 Halaman dashboard.php	41

BAB I

PENDAHULUAN

1.1. Latar Belakang

Projek ini disusun dan diselesaikan untuk memenuhi CLO 2, CLO 3, dan CLO 4 pada matakuliah Keamanan Sistem TK-46-05 dengan dosen Dr. YUDHA PURWANTO, S.T., M.T.,

1.2. Rumusan Masalah

1. Bagaimana sistem pengamanan web bekerja?
2. Bagaimana sistem pengamanan jaringan bekerja?

1.3. Tujuan

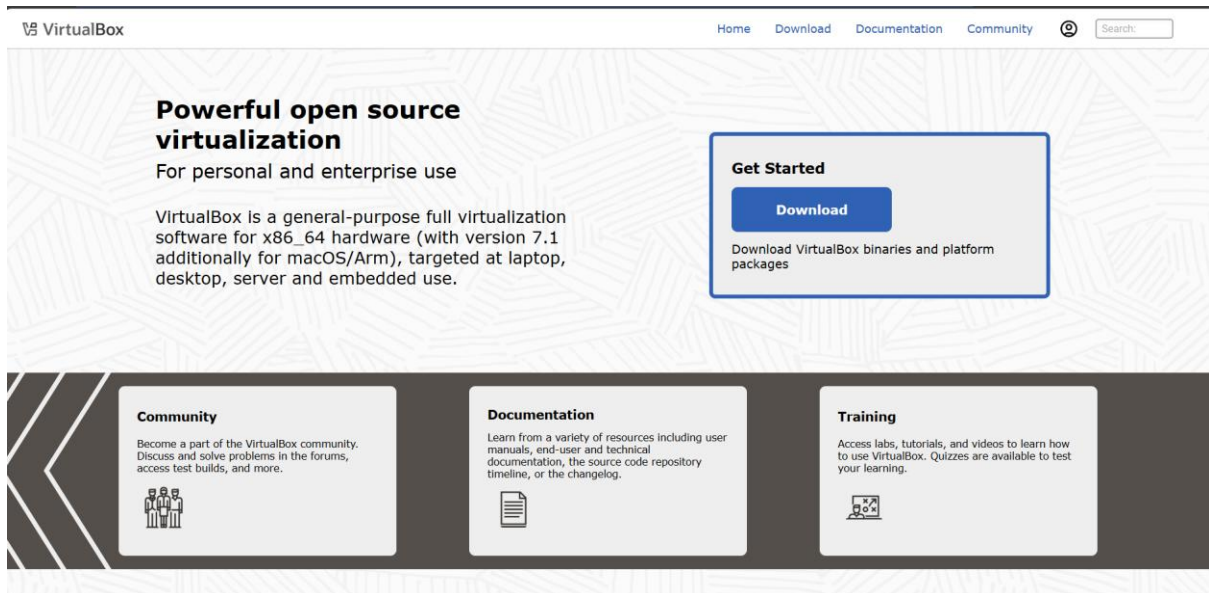
Tujuan projek ini dibuat adalah untuk mengetahui bagaimana sistem pengamanan web dan jaringan bekerja melalui Virtual Machine[2] yang didalamnya ada OS Linux[1], untuk deploy web server kemudian di *Hardening* dengan berbagai pengamanan yang dibutuhkan.

BAB II

TUTORIAL DAN LANGKAH KONFIGURASI CLO 2 PENGAMANAN WEB

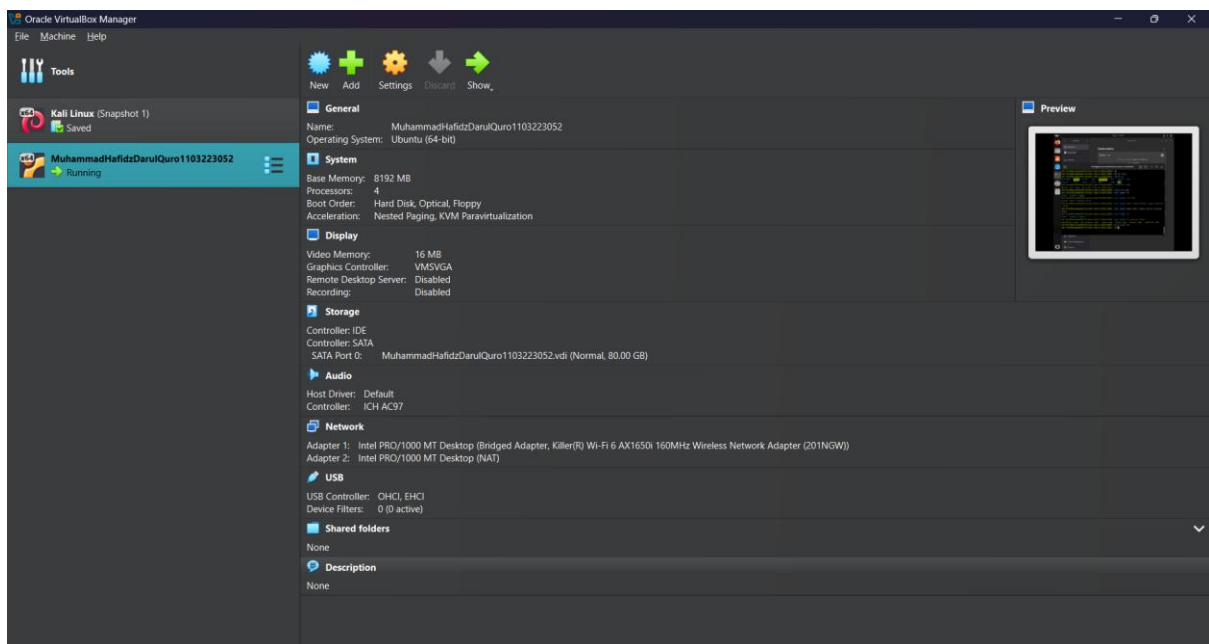
1. Instalasi VM

(Disini saya menggunakan VirtualBox)



Gambar 1 Instalasi VM

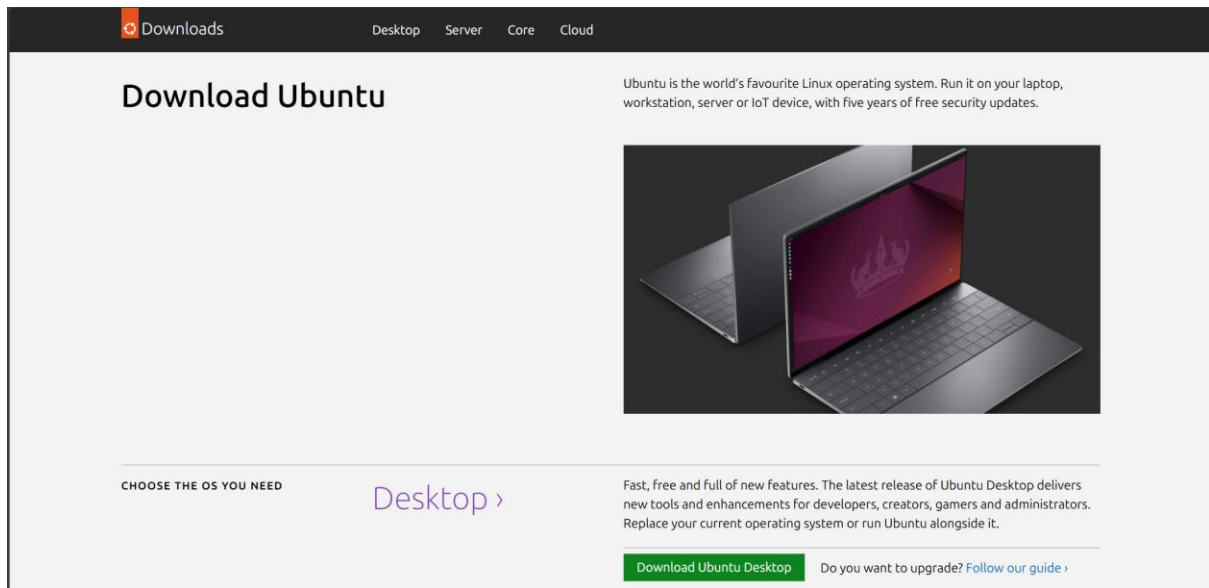
Download VM, Setelah download selesai, lakukan instalasi seperti biasa, kemudian buka aplikasi Oracle VirtualBox, nanti akan muncul tampilan seperti ini.



Gambar 2 Halaman awal VM

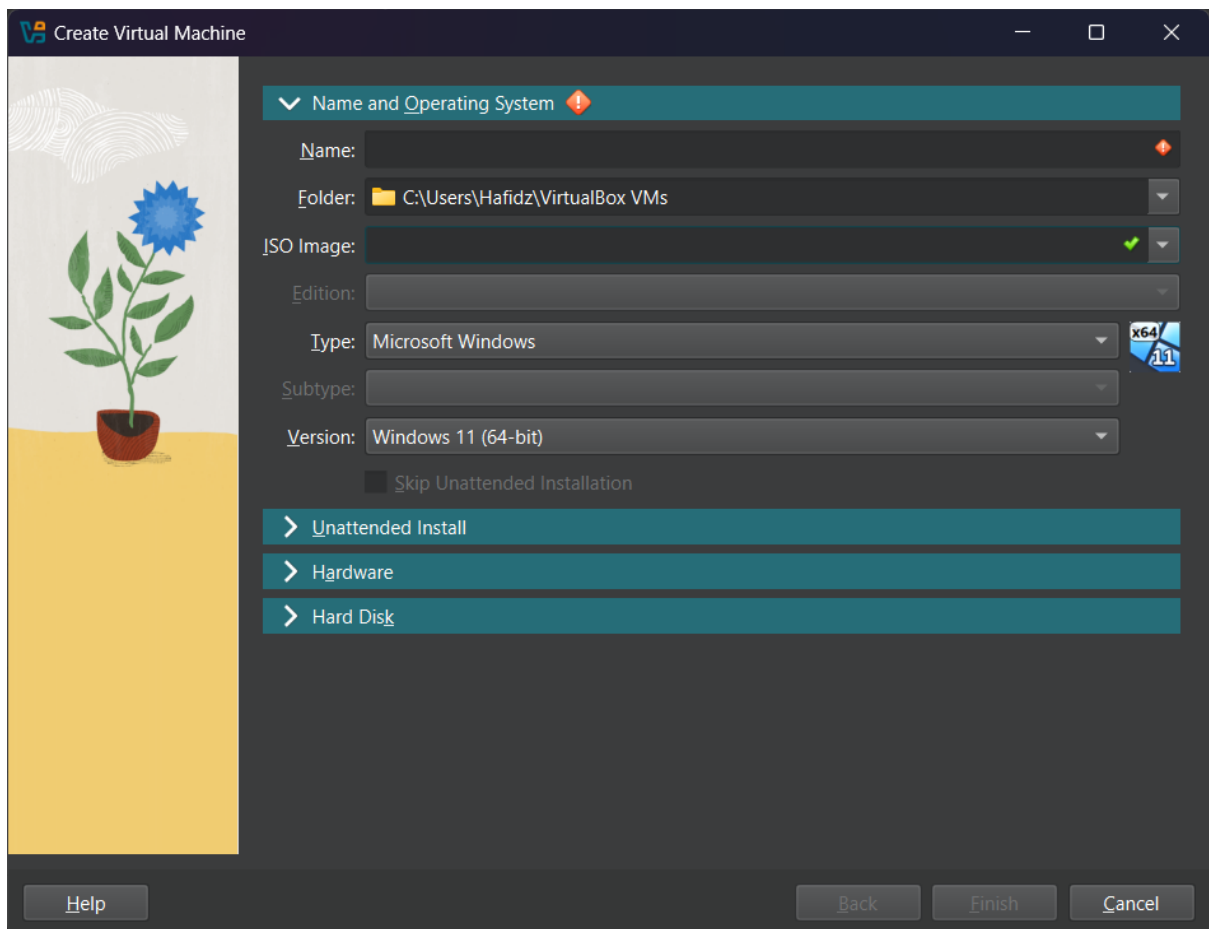
2. Download OS Linux

(Saya menggunakan Ubuntu)



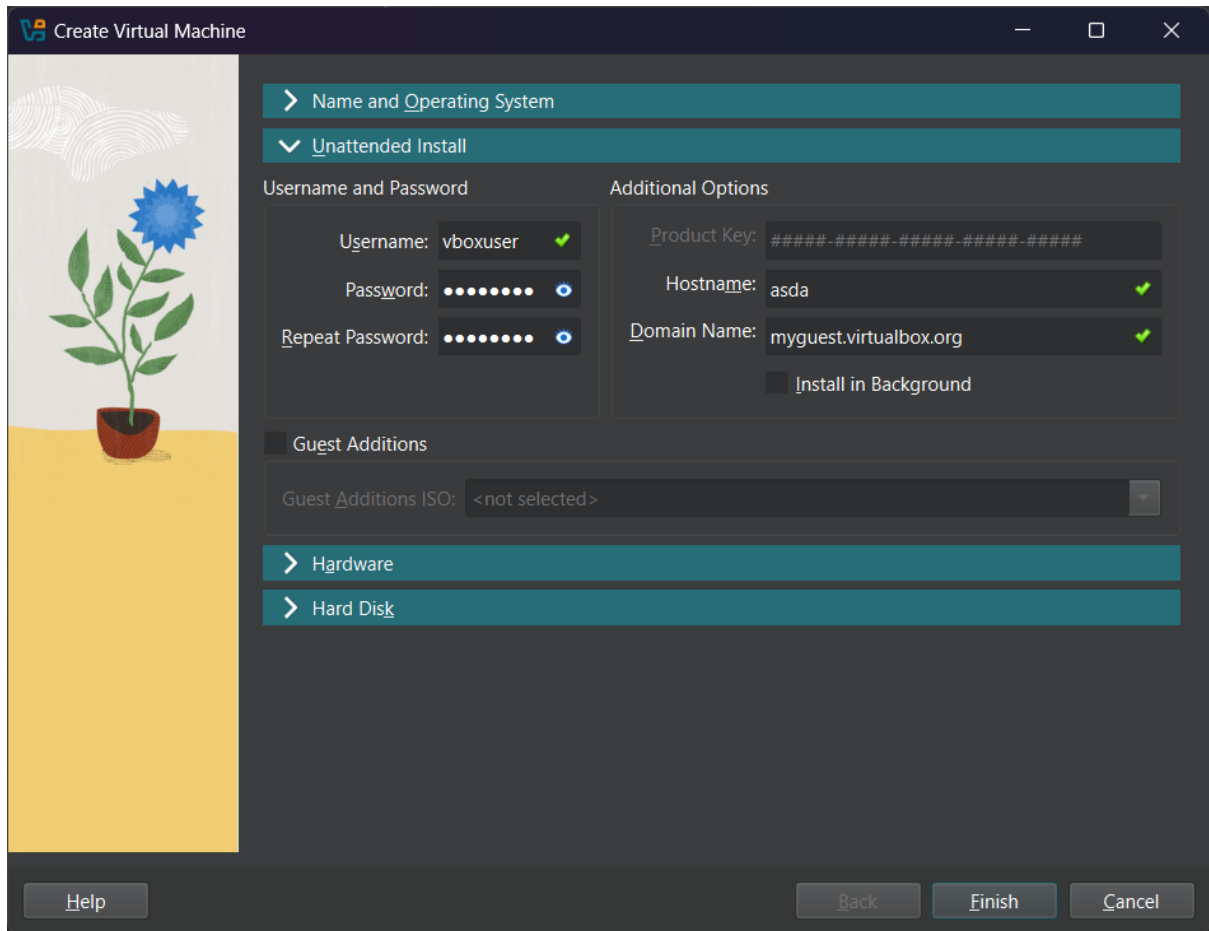
Gambar 3 Download Ubuntu Desktop

Download Ubuntu Desktop, setelah selesai akan muncul file ubuntu.xxxx.iso, kita kembali ke gambar 2, Klik “New”, akan muncul tampilan seperti berikut.



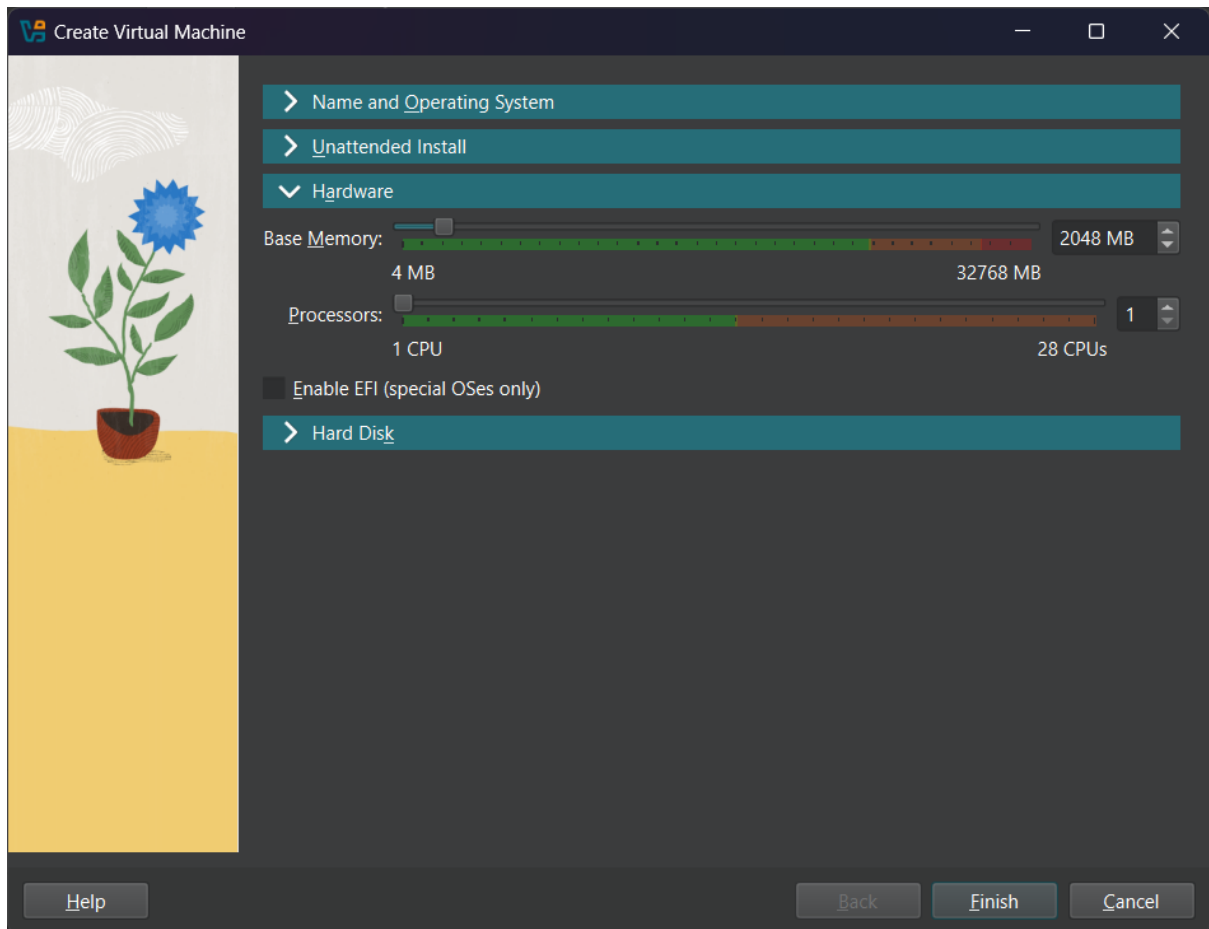
Gambar 4 Konfigurasi Awal Ubuntu Desktop

Lanjut, isi bagian name dengan NAMA+NIM, Folder isi dengan tempat dimana kita ingin menaruh VM berada, ISO Image pilih file ubuntu.xxx.iso yang sudah kita download.



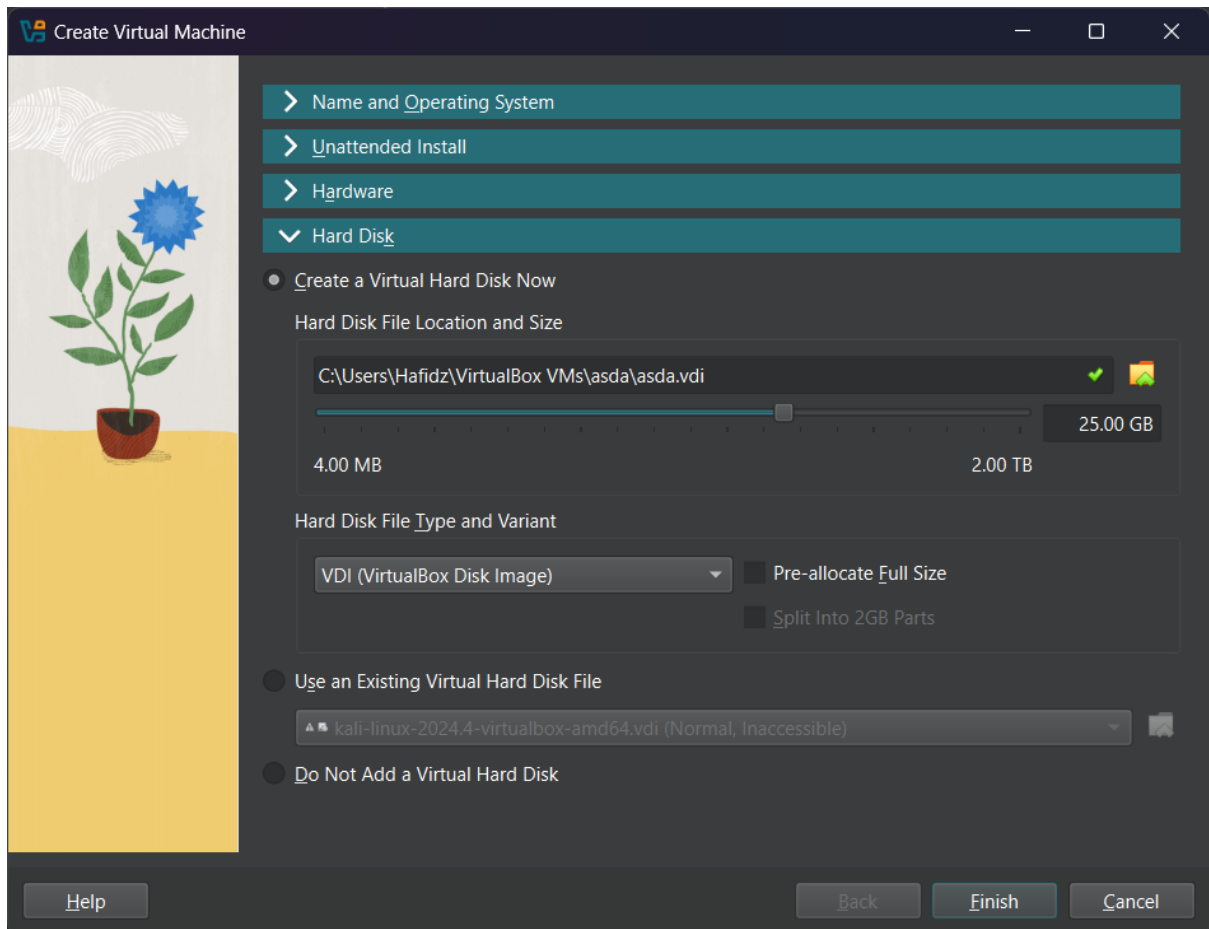
Gambar 5 Konfigurasi username dan password VM

Kemudian isi username dan password, ingat baik-baik atau disimpan, karena ini akan dipakai sebagai username dan password untuk OS Linux yang dipakai, untuk hostname dan domain name bisa disesuaikan sesuai kebutuhan saja.



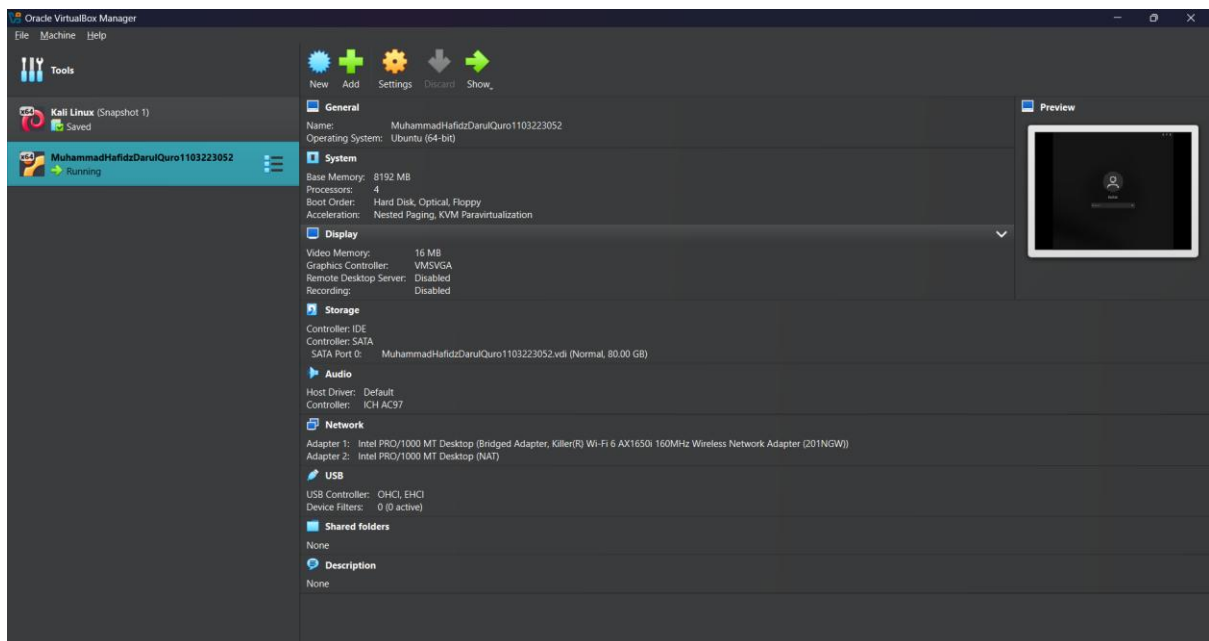
Gambar 6 Konfigurasi Memory dan Processor VM

Kemudian pada Gambar 6, kita lanjut memilih base memory dan processor yang digunakan, ini adalah default settings dari VirtualBox, untuk penggunaan kali ini kita bisa sesuaikan sesuai kebutuhan masing-masing.



Gambar 7 Konfigurasi Storage

Kemudian pada gambar 7, kita akan menset kebutuhan penyimpanan di OS Linux, sesuaikan juga dengan slider yang ada, berapa GB yang dibutuhkan. Serta lokasi penyimpanannya, kemudian Klik “Finish” . dan OS Linux sudah siap dipakai.



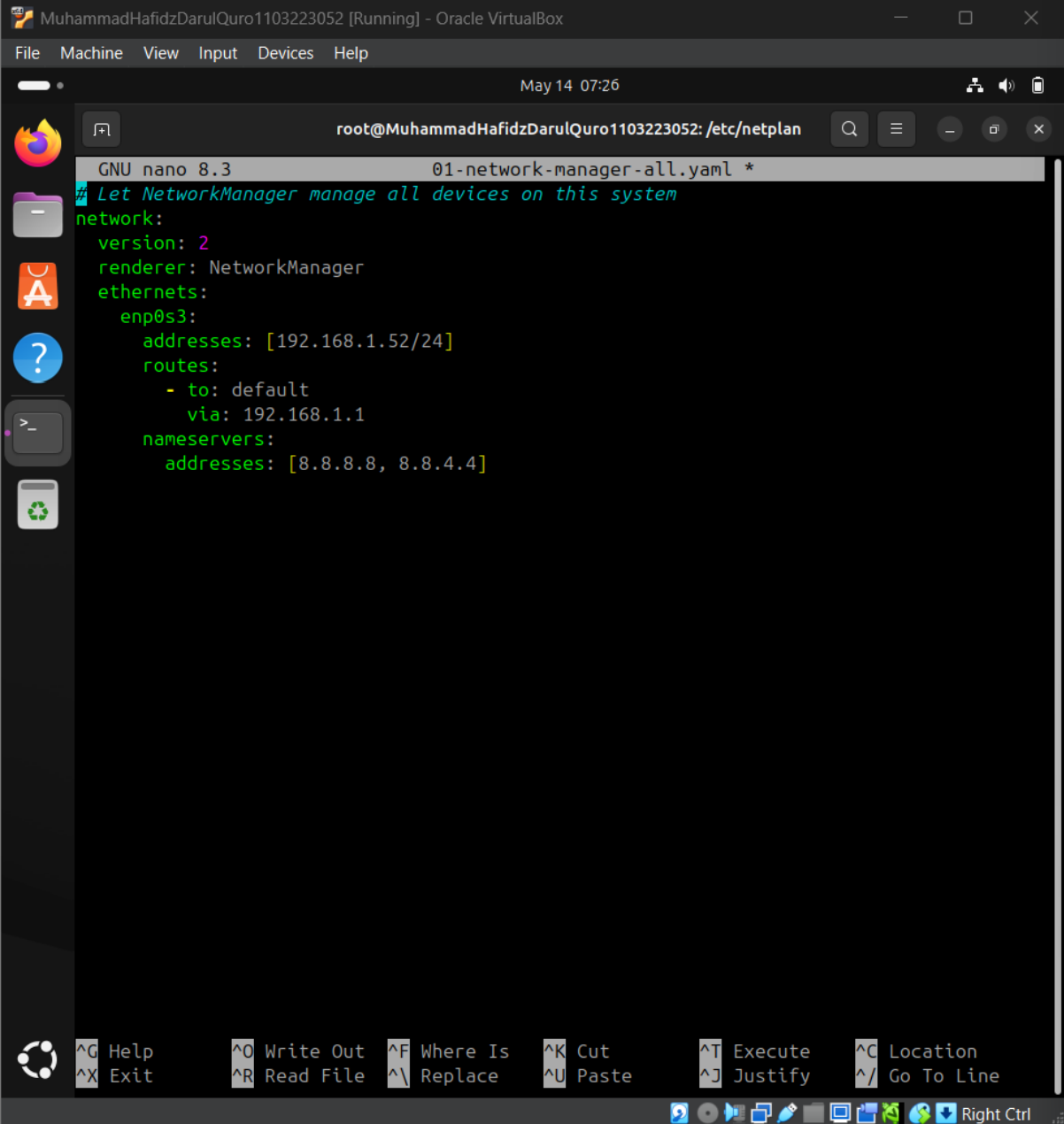
Gambar 8. OS Linux Siap Dipakai

3. Konfigurasi IP Address

(192.168.1.052, karena 052 tidak bisa digunakan dalam ip, maka saya menggunakan .52 untuk oktet terakhir) menggunakan command sebagai berikut.

Masuk kedalam root, kalau di Windows sebagai administrator untuk memudahkan langkah-langkah berikutnya.

```
sudo su
nano 01-network-manager-all.yaml
```



```
root@MuhammadHafidzDarulQuro1103223052: /etc/netplan
GNU nano 8.3 01-network-manager-all.yaml *
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernet:
    enp0s3:
      addresses: [192.168.1.52/24]
      routes:
        - to: default
          via: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Gambar 9 Konfigurasi .yaml untuk IP Address

Ubah isi yang awalnya hanya sampai renderer dengan menambah isi berikut

```
ethernets:
  enp0s3:
    addresses: [192.168.1.52/24]
    routes:
      - to: default
        via: 192.168.1.1
    nameservers:
      addresses: [8.8.8.8, 8.8.4.4]
```

Kemudian, ctrl+x untuk exit, y untuk menyimpan file dan enter untuk exit, bisa gunakan command

```
cat 01-network-manager-all.yaml
```

```
root@MuhammadHafidzDarulQuro1103223052:/etc/netplan# sudo cat 01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp0s3:
      addresses: [192.168.1.52/24]
      routes:
        - to: default
          via: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Gambar 10 Cek isi .yaml untuk konfigurasi IP

Gunakan command chmod untuk menambahkan akses terhadap .yaml yang sudah kita buat

```
chmod 600 /etc/netplan/*.yaml
```

Kemudian Jalankan command

```
netplan apply
```

Verifikasi konfigurasi network dengan command, terlihat ada ip 192.168.1.52

```
ip addr
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ad:1e:48 brd ff:ff:ff:ff:ff:ff
    altname enx080027ad1e48
    inet 192.168.1.52/24 brd 192.168.1.255 scope global noprefixroute enp0s3
        valid_lft forever preferred_lft forever
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 85540sec preferred_lft 85540sec
    inet6 fd00::df04:ead1:bcd6:c86d/64 scope global temporary dynamic
        valid_lft 86140sec preferred_lft 14140sec
    inet6 fd00::a00:27ff:fead:1e48/64 scope global dynamic mngtmpaddr proto kernel_r
        valid_lft 86140sec preferred_lft 14140sec
    inet6 fe80::a00:27ff:fead:1e48/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
```

Gambar 11 Verifikasi IP Address

4. Instalasi Web Server dan Database

Update system packages sebelum instalasi web server dan database

```
apt update
sudo apt upgrade -y
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# apt upgrade -y
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
```

Gambar 12 Upgrade Package Ubuntu

Instalasi **NGINX Web Server**, Enable NGINX, dan start NGINX Web Server, kemudian cek status.

```
apt install nginx -y
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# apt upgrade -y
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# apt install nginx -y
Installing:
  nginx

Installing dependencies:
  nginx-common

Suggested packages:
  fcgiwrap  nginx-doc

Summary:
  Upgrading: 0, Installing: 2, Removing: 0, Not Upgrading: 0
  Download size: 741 kB
  Space needed: 2,108 kB / 73.4 GB available

Get:1 http://id.archive.ubuntu.com/ubuntu plucky/main amd64 nginx-common all 1.26.3-2ubuntu1 [43.4 kB]
Get:2 http://id.archive.ubuntu.com/ubuntu plucky/main amd64 nginx amd64 1.26.3-2ubuntu1 [698 kB]
Fetched 741 kB in 10s (75.9 kB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 142393 files and directories currently installed.)
Preparing to unpack .../nginx-common_1.26.3-2ubuntu1_all.deb ...
Unpacking nginx-common (1.26.3-2ubuntu1) ...
Selecting previously unselected package nginx.
Preparing to unpack .../nginx_1.26.3-2ubuntu1_amd64.deb ...
Unpacking nginx (1.26.3-2ubuntu1) ...
Setting up nginx-common (1.26.3-2ubuntu1) ...
Created symlink '/etc/systemd/system/multi-user.target.wants/nginx.service' -> '/usr/lib/systemd/system/nginx.service'.
Setting up nginx (1.26.3-2ubuntu1) ...
  * Upgrading binary nginx [ OK ]
Processing triggers for man-db (2.13.0-1) ...
Processing triggers for ufw (0.36.2-9) ...
```

Gambar 13 Instalasi NGINX

```
systemctl enable nginx
```

```
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /usr/lib/systemd/sy
stemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable nginx
```

Gambar 14 Enable NGINX

```
systemctl start nginx
```

```
systemctl status nginx
```

```
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-05-14 07:50:17 UTC; 4min 11s ago
     Invocation: dfc015ee619b479b8a05505a2ba1dc16
       Docs: man:nginx(8)
    Main PID: 6237 (nginx)
      Tasks: 5 (limit: 8816)
     Memory: 4.5M (peak: 11M)
        CPU: 36ms
     CGroup: /system.slice/nginx.service
             └─6237 "nginx: master process /usr/sbin/nginx -g daemon on; master_proces>
                └─6239 "nginx: worker process"
                   └─6240 "nginx: worker process"
                      └─6241 "nginx: worker process"
                         └─6242 "nginx: worker process"

May 14 07:50:17 MuhammadHafidzDarulQuro1103223052 systemd[1]: Starting nginx.service ->
May 14 07:50:17 MuhammadHafidzDarulQuro1103223052 systemd[1]: Started nginx.service ->
```

Gambar 15 Cek Status NGINX

Instalasi **MYSQL** Database, Enable **MYSQL**, dan start **MYSQL** Database

```
apt install mysql-server -y
```

```
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# apt install mysql-server -y
mysql-server is already the newest version (8.4.5-0ubuntu0.1).
```

Gambar 16 Instalasi mysql-server

```
systemctl enable mysql
```

```
systemctl start mysql
```

```
systemctl status mysql
```

```
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# systemctl enable mysql
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# systemctl start mysql
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-05-14 08:00:01 UTC; 2min 50s ago
     Invocation: cb61a687305e4ae7bac092b904264c11
    Main PID: 7317 (mysqld)
      Status: "Server is operational"
     Tasks: 34 (limit: 8816)
    Memory: 436.7M (peak: 449.8M)
       CPU: 3.144s
     CGroup: /system.slice/mysql.service
             └─7317 /usr/sbin/mysqld

May 14 07:59:59 MuhammadHafidzDarulQuro1103223052 systemd[1]: Starting mysql.service - MySQL Community Serve
May 14 08:00:01 MuhammadHafidzDarulQuro1103223052 systemd[1]: Started mysql.service - MySQL Community Server
lines 1-14/14 (END)
```

Gambar 17 Cek Status mysql

Mengamankan Instalasi MYSQL

mysql_secure_installation

```
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# mysql_secure_installation
```

```
Securing the MySQL server deployment.
```

```
Connecting to MySQL using a blank password.
```

```
VALIDATE PASSWORD COMPONENT can be used to test passwords  
and improve security. It checks the strength of password  
and allows the users to set only those passwords which are  
secure enough. Would you like to setup VALIDATE PASSWORD component?
```

```
Press y|Y for Yes, any other key for No: █
```

Gambar 18 Mengamankan mysql

- Set root password
- Remove anonymous users
- Disallow root login remotely
- Remove test database
- Reload privilege tables

Setelah selesai set semuanya lanjut langkah berikutnya

5. Instalasi PHP

Disini saya menggunakan **PHP 8.4[4]**, menggunakan command dibawah untuk instalasi php dan mengecek versi php.

apt install php-fpm php-mysql -y

```
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# apt install php-fpm php-mysql -y
```

```
Installing:
```

```
  php-fpm  php-mysql
```

```
Installing dependencies:
```

```
  libargon2-1  php-common  php8.4-common  php8.4-mysql  php8.4-readline  
  libsodium23  php8.4-cli  php8.4-fpm  php8.4-opcache
```

```
Suggested packages:
```

```
  php-pear
```

```
Summary:
```

```
  Upgrading: 0, Installing: 11, Removing: 0, Not Upgrading: 0  
  Download size: 5,611 kB  
  Space needed: 24.8 MB / 72.8 GB available
```

```
Get:1 http://id.archive.ubuntu.com/ubuntu plucky/main amd64 libargon2-1 amd64 0-20190702+dfsg-4build1 [20.8 kB]
```

```
Get:2 http://id.archive.ubuntu.com/ubuntu plucky/main amd64 libsodium23 amd64 1.0.18-1build3 [161 kB]
```

```
Get:3 http://id.archive.ubuntu.com/ubuntu plucky/main amd64 php-common all 2:96ubuntu1 [14.2 kB]
```

```
Get:4 http://id.archive.ubuntu.com/ubuntu plucky/main amd64 php8.4-common amd64 8.4.5-1ubuntu1 [781 kB]
```

```
Get:5 http://id.archive.ubuntu.com/ubuntu plucky/main amd64 php8.4-opcache amd64 8.4.5-1ubuntu1 [477 kB]
```

```
Get:6 http://id.archive.ubuntu.com/ubuntu plucky/main amd64 php8.4-readline amd64 8.4.5-1ubuntu1 [13.8 kB]
```

```
Get:7 http://id.archive.ubuntu.com/ubuntu plucky/main amd64 php8.4-cli amd64 8.4.5-1ubuntu1 [1,998 kB]
```

```
Get:8 http://id.archive.ubuntu.com/ubuntu plucky/universe amd64 php8.4-fpm amd64 8.4.5-1ubuntu1 [2,009 kB]
```

```
Get:9 http://id.archive.ubuntu.com/ubuntu plucky/universe amd64 php-fpm all 2:8.4+96ubuntu1 [4,532 B]
```

```
Get:10 http://id.archive.ubuntu.com/ubuntu plucky/main amd64 php8.4-mysql amd64 8.4.5-1ubuntu1 [129 kB]
```

```
Get:11 http://id.archive.ubuntu.com/ubuntu plucky/main amd64 php-mysql all 2:8.4+96ubuntu1 [1,840 B]
```

```
Fetch: 5,611 kB in 3s (1,872 kB/s)
```

Gambar 19 Instalasi PHP

Menggunakan command berikut untuk **enable PHP**, **start PHP** serta **cek status PHP**

```
systemctl enable php8.4-fpm
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# systemctl enable php8.4-fpm
Synchronizing state of php8.4-fpm.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable php8.4-fpm
```

Gambar 20 Enable PHP

```
systemctl start php8.4-fpm
systemctl status php8.4-fpm
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# systemctl start php8.4-fpm
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# systemctl status php8.4-fpm
● php8.4-fpm.service - The PHP 8.4 FastCGI Process Manager
   Loaded: loaded (/usr/lib/systemd/system/php8.4-fpm.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-05-14 08:16:56 UTC; 6min ago
     Invocation: 5d82fbf66d47434aa809f3ec207ffcd1
       Docs: man:php-fpm8.4(8)
    Main PID: 15145 (php-fpm8.4)
      Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic: 0.00req/sec"
        Tasks: 3 (limit: 8816)
       Memory: 8.1M (peak: 10M)
          CPU: 77ms
      CGroup: /system.slice/php8.4-fpm.service
              └─15145 "php-fpm: master process (/etc/php/8.4/fpm/php-fpm.conf)"
                 └─15148 "php-fpm: pool www"
                    └─15149 "php-fpm: pool www"

May 14 08:16:56 MuhammadHafidzDarulQuro1103223052 systemd[1]: Started php8.4-fpm.service - The PHP 8.4 FastCGI Process Manager.
lines 1-16/16 (END)
```

Gambar 21 Cek Status PHP

6. Persiapan Database

Sekarang kita akan melanjutkan untuk persiapan database MYSQL dengan nama “phplogin”.

```
mysql -u root -p
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.4.5-0ubuntu0.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

Gambar 22 Masuk ke root mysql

Ikuti Command dibawah satu per satu ketika di dalam MYSQL untuk membuat database phplogin, serta membuat USER

```
CREATE DATABASE phplogin;
```

```
mysql> CREATE DATABASE phplogin;  
Query OK, 1 row affected (0.01 sec)
```

Gambar 23 CREATE DATABASE phplogin

```
CREATE USER 'webuser'@'localhost' IDENTIFIED BY 'Hafidz1103223052@123';  
mysql> CREATE USER 'webuser'@'localhost' IDENTIFIED BY 'Hafidz1103223052@123';  
Query OK, 0 rows affected (0.01 sec)
```

Gambar 24 CREATE USER mysql

```
GRANT ALL PRIVILEGES ON phplogin.* TO 'webuser'@'localhost';  
mysql> GRANT ALL PRIVILEGES ON phplogin.* TO 'webuser'@'localhost';  
Query OK, 0 rows affected (0.02 sec)
```

Gambar 25 GRANT ALL PRIVILEGES mysql

FLUSH PRIVILEGES;

```
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.01 sec)
```

Gambar 26 FLUSH PRIVILEGES mysql

Kemudian membuat **users table**

USE phplogin;

```
mysql> CREATE DATABASE phplogin;  
Query OK, 1 row affected (0.01 sec)
```

Gambar 27 USE phplogin mysql

```
CREATE TABLE users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  username VARCHAR(50) NOT NULL UNIQUE,  
  password_hash VARCHAR(255) NOT NULL,  
  salt VARCHAR(64) NOT NULL,  
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

```
mysql> CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY,  
-> username VARCHAR(50) NOT NULL UNIQUE,  
-> password_hash VARCHAR(255) NOT NULL,  
-> salt VARCHAR(64) NOT NULL,  
-> created_at DATETIME DEFAULT CURRENT_TIMESTAMP  
-> );  
Query OK, 0 rows affected (0.06 sec)
```

Gambar 28 CREATE TABLE users mysql

Mengecek table users, dengan command

SHOW COLUMNS FROM users;

```
mysql> SHOW COLUMNS FROM users;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default          | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int           | NO   | PRI | NULL             | auto_increment |
| username       | varchar(50)   | NO   | UNI | NULL             |                |
| password_hash  | varchar(255)  | NO   |     | NULL             |                |
| salt           | varchar(64)   | NO   |     | NULL             |                |
| created_at     | datetime      | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Gambar 29 Menampilkan Table users

EXIT;

untuk keluar dari mysql

7. Aplikasi Web Development

Membuat direktori web dan memodifikasi akses dengan command

```
mkdir -p /var/www/html/phplogin
chown -R www-data:www-data /var/www/html/phplogin
root@MuhammadHafidzDarulQuro1103223052:/var/www/html/phplogin# chown -R www-data:www-data
var/www/html/phplogin
```

Gambar 30 Modifikasi Akses phplogin

Lalu kita akan membuat **dashboard.php**, **db_connect.php**, **index.php**, **login.php**, **logout.php**, **register.php**, di dalam direktori **/var/www/html/phplogin**

Untuk **dashboard.php**

nano dashboard.php

Masukkan code berikut kedalam file

```
<?php
session_start();

// Check if user is not logged in
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit;
}

$username = htmlspecialchars($_SESSION['username']);
?>

<!DOCTYPE html>
<html>
<head>
    <title>Dashboard</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        body { font-family: Arial, sans-serif; max-width: 800px; margin: 0 auto; padding: 20px; }
        .header { display: flex; justify-content: space-between; align-items: center; }
```

```

.logout { background-color: #f44336; color: white; padding: 10px 15px; text-decoration: none;
}
</style>
</head>
<body>
  <div class="header">
    <h2>Welcome, <?php echo $username; ?></h2>
    <a href="logout.php" class="logout">Logout</a>
  </div>

  <div>
    <h3>Secure Dashboard</h3>
    <p>This is a secure area that is only accessible after successful login.</p>
    <p>Your login is protected with:</p>
    <ul>
      <li>HTTPS encryption</li>
      <li>Password salting</li>
      <li>Password hashing (SHA-256)</li>
      <li>Protection against SQL injection</li>
      <li>Protection against XSS attacks</li>
      <li>Protection against buffer overflow</li>
    </ul>
  </div>
</body>
</html>

```

```
root@MuhammadHafidzDarulQuro1103223052: /var/www/html/phplogin
GNU nano 8.3 dashboard.php

<?php
session_start();

// Check if user is not logged in
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit;
}

$username = htmlspecialchars($_SESSION['username']);
?>

<!DOCTYPE html>
<html>
<head>
    <title>Dashboard</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        body { font-family: Arial, sans-serif; max-width: 800px; margin: 0 auto; padding: 10px; }
        .header { display: flex; justify-content: space-between; align-items: center; }
        .logout { background-color: #f44336; color: white; padding: 10px 15px; text-decoration: none; }
    </style>
</head>
<body>
    <div class="header">
        <h2>Welcome, <?php echo $username; ?></h2>
        <a href="logout.php" class="logout">Logout</a>
    </div>

    <div>
        <h3>Secure Dashboard</h3>
        <p>This is a secure area that is only accessible after successful login.</p>
        <p>Your login is protected with:</p>
    </div>
</body>
</html>

[ Read 45 lines ]
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line

Gambar 31 Konfigurasi dashboard.php
```

Setelah selesai *edit*, *save* dan *exit*

Untuk **db_connect.php**, *database connection*

```
nano db_connect.php
Masukkan code berikut kedalam file

<?php
// Database connection parameters
$host = 'localhost';
$dbname = 'phplogin';
$username = 'webuser';
$password = 'Hafidz1103223052@123';

// Create a database connection
try {
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username,
$password);
    // Set PDO to throw exceptions on error
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}
```

```

} catch (PDOException $e) {
    die("Database connection failed: " . $e->getMessage());
}

```

```

GNU nano 8.3 db_connect.php
<?php
// Database connection parameters
$host = 'localhost';
$dbname = 'phplogin';
$username = 'webuser';
$password = 'Hafidz1103223052@123';

// Create a database connection
try {
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
    // Set PDO to throw exceptions on error
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Database connection failed: " . $e->getMessage());
}
?>

```

Gambar 32 Konfigurasi db_connect.php

Setelah selesai, *save* dan *exit*

Untuk **index.php**

nano index.php

```

<?php
// Redirect to login page
header("Location: login.php");
exit;
?>

```

```

GNU nano 8.3 index.php
<?php
// Redirect to login page
header("Location: login.php");
exit;
?>

```

Gambar 33 Konfigurasi index.php

Setelah selesai, *save* dan *exit*

Untuk **login.php**

nano login.php

```

<?php
require_once 'db_connect.php';
session_start();

$error_message = '';

// If user is already logged in, redirect to dashboard
if (isset($_SESSION['user_id'])) {
    header("Location: dashboard.php");
    exit;
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Get form data
    $username = filter_input(INPUT_POST, 'username',
    FILTER_SANITIZE_STRING);
}

```

```

    $password = $_POST['password'];

    // Get user from database
    $stmt = $pdo->prepare("SELECT id, username, password_hash,
salt FROM users WHERE username = ?");
    $stmt->execute([$username]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($user) {
        // Verify password
        $password_hash = hash('sha256', $password .
$user['salt']);

        if ($password_hash === $user['password_hash']) {
            // Password is correct, start a new session
            session_regenerate_id();
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['username'] = $user['username'];

            // Redirect to dashboard
            header("Location: dashboard.php");
            exit;
        } else {
            $error_message = "Invalid username or password";
        }
    } else {
        $error_message = "Invalid username or password";
    }
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <style>
        body { font-family: Arial, sans-serif; max-width: 500px;
margin: 0 auto; padding: 20px; }
        .form-group { margin-bottom: 15px; }
        label { display: block; margin-bottom: 5px; }
        input { width: 100%; padding: 8px; box-sizing: border-box;
}
        .error { color: red; margin-bottom: 15px; }
        button { background-color: #4CAF50; color: white; padding:
10px 15px; border: none; cursor: pointer; }
        a { display: inline-block; margin-top: 15px; }
    </style>
</head>
<body>
    <h2>Login</h2>

    <?php if ($error_message): ?>
        <div class="error"><?php echo
htmlspecialchars($error_message); ?></div>
    <?php endif; ?>

    <form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]); ?>"
onsubmit="saltPassword()">
        <div class="form-group">

```



```

        <label for="username">Username:</label>
        <input type="text" id="username" name="username"
required>
    </div>

    <div class="form-group">
        <label for="password">Password:</label>
        <input type="password" id="password" name="password"
required>
    </div>

    <button type="submit">Login</button>
</form>

<a href="register.php">Don't have an account? Register
here</a>

<script>
    // Client-side salting (additional security layer)
    function saltPassword() {
        // This adds a client-side salt to the password
        // Note: This is just an additional layer - the real
security comes from server-side salting
        const clientsalt = "NIM1103223052clientsalt";
        const passwordField =
document.getElementById('password');
        const password = passwordField.value;

        // we're not actually replacing the password in this
demo, as we handle salt server-side
        // This is just to demonstrate the concept
        console.log("Password has been salted on client
side");
    }
</script>
</body>

```

```
</html>

root@MuhammadHafidzDarulQuro1103223052: /var/www/html/phplogin

GNU nano 8.3 login.php *

<?php
require_once 'db_connect.php';
session_start();

$error_message = '';

// If user is already logged in, redirect to dashboard
if (isset($_SESSION['user_id'])) {
    header("Location: dashboard.php");
    exit;
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Get form data
    $username = filter_input(INPUT_POST, 'username', FILTER_SANITIZE_STRING);
    $password = $_POST['password'];

    // Get user from database
    $stmt = $pdo->prepare("SELECT id, username, password_hash, salt FROM users WHERE usern");
    $stmt->execute([$username]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($user) {
        // Verify password
        $password_hash = hash('sha256', $password . $user['salt']);

        if ($password_hash === $user['password_hash']) {
            // Password is correct, start a new session
            session_regenerate_id();
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['username'] = $user['username'];

            // Redirect to dashboard
            header("Location: dashboard.php");
        }
    }
}

[ Read 99 lines ]
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line

Gambar 34 Konfigurasi login.php
```

Setelah selesai, *save* dan *exit*

Untuk **logout.php**

```
nano logout.php

<?php
session_start();

// Unset all session variables
$_SESSION = array();

// Delete the session cookie
if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000,
        $params["path"], $params["domain"],
        $params["secure"], $params["httponly"]
    );
}
```

```
// Destroy the session
session_destroy();

// Redirect to login page
header("Location: login.php");
exit;
?>
```

The screenshot shows a terminal window with the title bar 'root@MuhammadHafidzDarulQuro1103223052: /var/www/html/phplogin'. The nano editor is open to the file 'logout.php'. The code in the editor is as follows:

```
GNU nano 8.3      logout.php
<?php
session_start();

// Unset all session variables
$_SESSION = array();

// Delete the session cookie
if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000,
        $params["path"], $params["domain"],
        $params["secure"], $params["httponly"]
    );
}

// Destroy the session
session_destroy();

// Redirect to login page
header("Location: login.php");
exit;
?>
```

Gambar 35 Konfigurasi logout.php

Setelah selesai, *save* dan *exit*

Untuk **register.php**

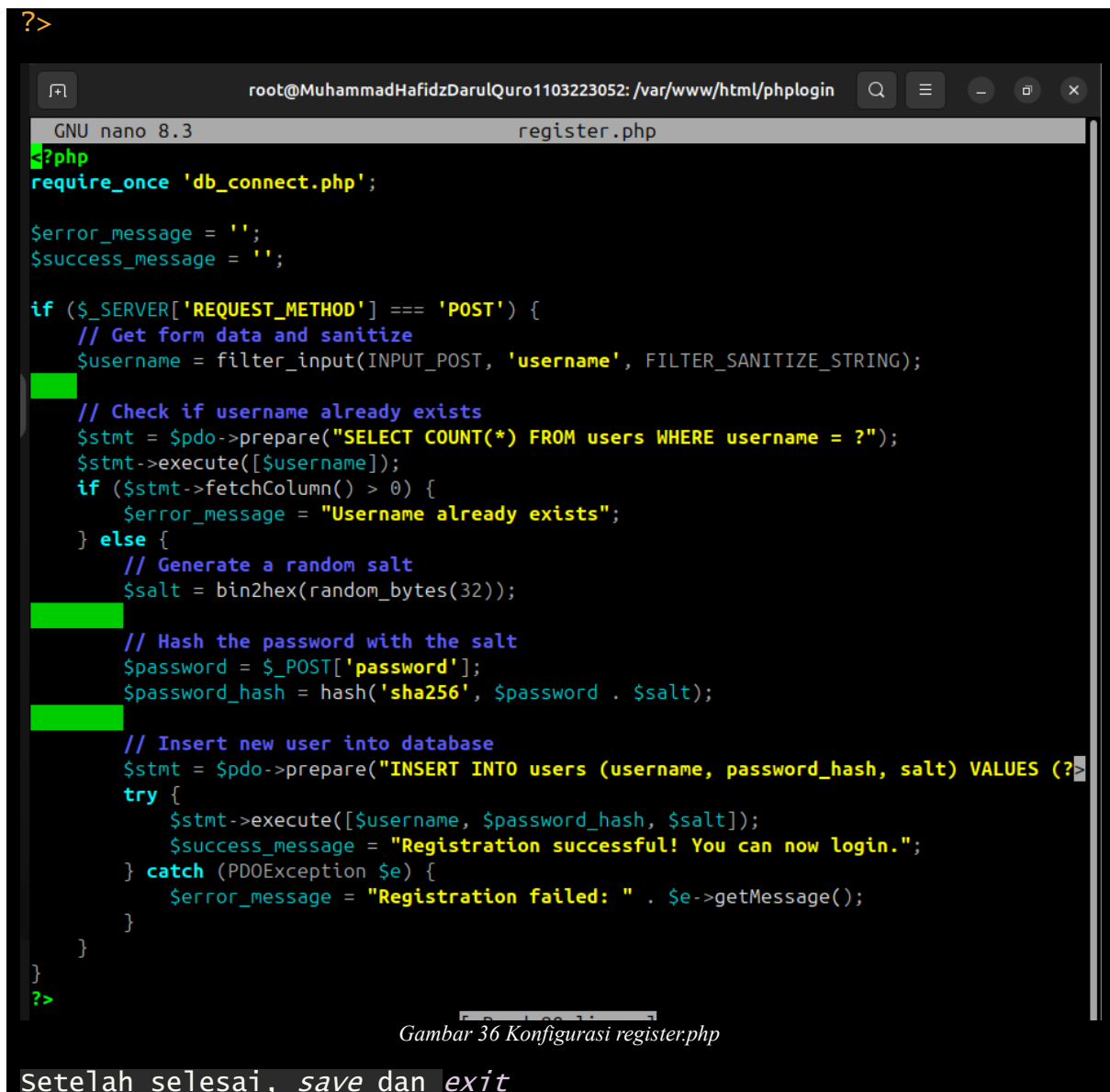
```
nano register.php
<?php
session_start();

// Unset all session variables
$_SESSION = array();

// Delete the session cookie
if (ini_get("session.use_cookies")) {
    $params = session_get_cookie_params();
    setcookie(session_name(), '', time() - 42000,
        $params["path"], $params["domain"],
        $params["secure"], $params["httponly"]
    );
}

// Destroy the session
session_destroy();

// Redirect to login page
header("Location: login.php");
exit;
```



8. Pengamanan Web

Dalam pengamanan web, sesuai yang diminta yakni pengamanan pada :

- Mengkonfigurasi HTTPS pada Web Server.
- Memberikan Teknik Salting pada data login.
- Memberikan Fungsi Hash pada pengiriman data password.
- Mengamankan input dari SQL Injection dan Buffer Overflow.
- Mengamankan web dari XSS (Cross Site Scripting).

a. Konfigurasi HTTPS pada Web Server.

Kita akan melakukan konfigurasi di bagian konfigurasi nginx, dengan menambahkan Self-Signed Certificate dan strong Diffie-Hellman Group.

Self-Signed Certificate adalah sebuah sertifikat yang dibuat dan di “sign” sendiri, bukan yang di isukan oleh otoritas sertifikat terpercaya seperti Encrypt, DigiCert atau Comodo untuk web server yang digunakan.

Dengan fungsi, yakni :

- Mengenkripsi lalu lintas diantara server dan client, walau tanpa menggunakan CA-Signed Certificate
- Enkripsi data saat transmisi
- Enable HTTPS saat konfigurasi di NGINX

Command yang digunakan sebagai berikut.

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
/etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt
```

Poin-poin penting dalam command diatas, yakni :

- Menggunakan **OpenSSL** untuk membuat sertifikat (-x509)
- Membuat **kunci “pair” RSA** baru dengan 2048-bit (-newkey rsa:2048)
- Membuat **kadaluarsa key** menjadi 365 hari (-days 365)
- Menempatkan **private key** dan **sertifikat** di (-keyout /etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt)

Strong Diffie-Hellman Group adalah sebuah metode dalam mengamankan pertukaran kunci kriptografi dalam channel publik tanpa perlu “pre-shared secret”.

Dengan fungsi, yakni :

- Memperkuat proses pertukaran kunci saat memulai koneksi **Transport Layer Security (TLS)**.
- Memitigasi serangan dari attacker terhadap **intercept** dan **decrypt** komunikasi.

Command yang digunakan sebagai berikut.

```
sudo openssl dhparam -out /etc/nginx/dhparam.pem 2048
```

Poin-poin penting dalam command diatas, yakni :

- **OpenSSL** akan mengenerate **Diffie-Hellman parameter sepanjang 2048-bit**.

Kedua metode diatas disatukan dalam konfigurasi NGINX (/etc/nginx/sites-available), dengan command.

```
ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;  
ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;  
ssl_dhparam /etc/nginx/dhparam.pem;  
  
ssl_protocols TLSv1.2 TLSv1.3;  
ssl_prefer_server_ciphers on;
```

```
ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384;
```

Hasil kombinasi command sebelumnya pada file (/etc/nginx/sites-available/phplogin).

```
server {
    listen 80; #Menerima koneksi pada port 80 (HTTP).
    server_name _; # _ merupakan wildcard – server merespon semua nama yang
host terima
    return 301 https://$host$request_uri; #Melakukan redirect permanen dari HTTP
ke HTTPS dengan menyimpan host dan path URL.
} #Blok untuk konfigurasi server

server {
    listen 443 ssl; #Menerima koneksi HTTPS pada port 443 saat SSL/TLS
diaktifkan
    server_name _; #Merespon semua nama host yang masuk

    ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt; #Menentukan lokasi file
sertifikat SSL/TSL (Public Key).
    ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key; #Menentukan lokasi file
private key sertifikat.
    ssl_dhparam /etc/nginx/dhparam.pem; #Menentukan lokasi file Diffie-Hellman

    # SSL settings
    ssl_protocols TLSv1.2 TLSv1.3; #Membatasi protokol SSL/TSL yang dapat
digunakan ke versi 1.2 dan 1.3
    ssl_prefer_server_ciphers on; #Memprioritaskan cipher suites yang diatur server
    ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384; #Menentukan cipher suites yang
diizinkan secara spesifik. (Menggunakan Elliptic Curve Diffie-Hellman Ephemeral
dengan RSA, AES 256 dalam mode GCM dan SHA-512 untuk autentikasi).
    ssl_session_timeout 1d; #Menyimpan sesi SSL/TLS selama 1 hari sebelum
membutuhkan handshake baru.
    ssl_session_cache shared:SSL:10m; #Mengalokasikan 10MB cache untuk sesi
SSL/TLS
    # ssl_stapling on; #Mengaktifkan OCSP stapling (metode verifikasi status
sertifikat yang lebih efisien)
    # ssl_stapling_verify on; #Verifikasi respon OCSP sebelum mengirim ke client

    # Security headers
    add_header X-Frame-Options "SAMEORIGIN"; #Mencegah Website di-embed
dalam iframe di website lain (memproteksi terhadap clickjacking), kecuali pada
domain yang sama.
    add_header X-XSS-Protection "1; mode=block"; #Mengaktifkan fitur
perlindungan XSS di browser yang lebih lama, memblokir halaman saat XSS
terdeteksi
```

```

    add_header X-Content-Type-Options "nosniff"; #Mencegah browser melakukan
    MIME-type sniffing, menghindari eksekusi file yang tidak diharapkan.
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains;
    preload"; #Mengaktifkan HSTS, membuat browser selalu terhubung
    add_header Content-Security-Policy "default-src 'self'; script-src 'self' 'unsafe-
    inline'; style-src 'self' 'unsafe-inline';"; #Mendefinisikan CSP untuk mencegah
    berbagai injection serta mengizinkan script dari domain yang sama dan inline
    scripts serta style

    root /var/www/html/phplogin; #Menentukan direktori root untuk akses file situs
    web
    index index.php index.html; #Mendefinisikan file default yang dimuat saat
    mengakses web

    location / {
        try_files $uri $uri /index.php?$query_string; #Meneruskan ke indeks.php
        dengan query string
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf; #Menyertakan konfigurasi NGINX default
        untuk FastCGI PHP.
        fastcgi_pass unix:/var/run/php/php8.4-fpm.sock; #Memastikan permintaan php
        ke socket PHP-FPM sesuai versinya.
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        #menentukan jalur lengkap ke skrip PHP yang diminta
        include fastcgi_params;
    }

    location ~ /\.ht {
        deny all; #Melarang akses ke file, mencegah akses langsung ke file konfigurasi
        Apache yang sensitif
    }
}

```

Setelah menyatukan command sebelumnya, kita perlu untuk enable dan restart NGINX, dengan command.

```

sudo ln -s /etc/nginx/sites-available/phplogin /etc/nginx/sites-enabled/
sudo rm /etc/nginx/sites-enabled/default # Menghilangkan default site NGINX
sudo systemctl restart nginx #Restart NGINX

```

Serta memberikan akses dengan command.

```

sudo chown -R www-data:www-data /var/www/phplogin
sudo chmod -R 755 /var/www/secure-login

```

b. Memberikan **Teknik Salting** pada data login.

Teknik Salting adalah sebuah proses menambahkan string acak ke password sebelum di-hash untuk mencegah serangan dictionary dan rainbow table.

Contoh Implementasi perlindungan terdapat pada register.php, login.php

```
// Generate a random salt
$salt = bin2hex(random_bytes(32));

// Hash the password with the salt
$password = $_POST['password'];
$password_hash = hash('sha256', $password . $salt);

// Verify password
$password_hash = hash('sha256', $password . $user['salt']);
```

Gambar 37 Implementasi Salting dan Hash

Client-side Salting tambahan untuk pengamanan

```
<script>
// Client-side salting (additional security layer)
function saltPassword() {
    // This adds a client-side salt to the password
    // Note: This is just an additional layer - the real security comes from server-side salting
    const clientSalt = "NIM1103223052ClientSalt";
    const passwordField = document.getElementById('password');
    const password = passwordField.value;

    // We're not actually replacing the password in this demo, as we handle salt server-side
    // This is just to demonstrate the concept
    console.log("Password has been salted on client side");
}
</script>
```

Gambar 38 Implementasi Salting pada Client-Side

c. Memberikan **Fungsi Hash** pada pengiriman data password.

Hash adalah sebuah proses mengubah data menjadi string karakter dengan panjang tetap yang bersifat satu arah (tidak bisa dikembalikan ke bentuk aslinya).

Contoh Implementasi perlindungan terhadap Hash juga terdapat pada bagian **Teknik Salting**.

d. Mengamankan input dari **SQL Injection dan Buffer Overflow**.

SQL Injection adalah serangan dengan menyisipkan kode SQL ke dalam input pengguna.

Contoh Implementasi perlindungan terhadap SQL Injection ada pada register.php, login.php, register.php serta db_connect.php.


```
// Insert new user into database
$stmt = $pdo->prepare("INSERT INTO users (username, password_hash, salt) VALUES (?, ?, ?)");
try{
    $stmt->execute([$username, $password_hash, $salt]);

```

Gambar 39 Implementasi perlindungan dari SQL Injection

```
// Get form data and sanitize
$username = filter_input(INPUT_POST, 'username', FILTER_SANITIZE_STRING);

```

Gambar 40 Implementasi perlindungan dari SQL Injection

```
// Get user from database
$stmt = $pdo->prepare("SELECT id, username, password_hash, salt FROM users WHERE username = ?");
$stmt->execute([$username]);
$user = $stmt->fetch(PDO::FETCH_ASSOC);

```

Gambar 41 Implementasi perlindungan dari SQL Injection

```
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

```

Gambar 42 Implementasi perlindungan dari SQL Injection

Buffer Overflow terjadi ketika program mencoba menyimpan data melebihi batas buffer yang dialokasikan.

Contoh implementasi perlindungan terhadap Buffer Overflow terdapat di **Pembatasan panjang input di skema database**, yakni saat input skema table dengan mysql. Juga pada **validasi input di register.php**, serta **pembatasan sumberdaya di konfigurasi PHP**.

```
mysql> CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY,
-> username VARCHAR(50) NOT NULL UNIQUE,
-> password_hash VARCHAR(255) NOT NULL,
-> salt VARCHAR(64) NOT NULL,
-> created_at DATETIME DEFAULT CURRENT_TIMESTAMP
-> );
Query OK, 0 rows affected (0.06 sec)

```

Gambar 43 Implementasi perlindungan dari Buffer Overflow

```
// Get form data and sanitize
$username = filter_input(INPUT_POST, 'username', FILTER_SANITIZE_STRING);

```

Gambar 44 Implementasi perlindungan dari Buffer Overflow

```
; Maximum size of POST data that PHP will accept.
; Its value may be 0 to disable the limit. It is ignored if POST data reading
; is disabled through enable_post_data_reading.
; https://php.net/post-max-size
post_max_size = 8M

```

Gambar 45 Implementasi perlindungan dari Buffer Overflow

- e. Mengamankan web dari **XSS (Cross Site Scripting)**.

Cross Site Scripting adalah sebuah teknik yang digunakan attacker untuk menyisipkan script berbahaya ke halaman web yang dilihat pengguna lain.

Contoh implementasi yang digunakan untuk melindungi web dari XX terdapat pada **Output Escaping di login.php, register.php** dan di **konfigurasi NGINX**.

```
<?php if ($error_message): ?>
    <div class="error"><?php echo htmlspecialchars($error_message); ?></div>
<?php endif; ?>

<?php if ($success_message): ?>
    <div class="success"><?php echo htmlspecialchars($success_message); ?></div>
<?php endif; ?>
```

Gambar 46 Implementasi perlindungan dari XSS

```
# Security headers
add_header X-Frame-Options "SAMEORIGIN";
add_header X-XSS-Protection "1; mode=block";
add_header X-Content-Type-Options "nosniff";
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload";
add_header Content-Security-Policy "default-src 'self'; script-src 'self' 'unsafe-inline'; style-src 'self' 'unsafe-inline';";
```

Gambar 47 Implementasi perlindungan dari XSS

BAB III

TUTORIAL DAN LANGKAH KONFIGURASI CLO 3 PENGAMANAN JARINGAN

1. Instalasi UFW

Instalasi UFW dengan command berikut.

```
sudo su
apt update
apt install ufw -y
root@MuhammadHafidzDarulQuro1103223052:/var/www/html/phplogin# apt install ufw
ufw is already the newest version (0.36.2-9).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
```

Gambar 48 Instalasi ufw

Terinstall versi paling terbaru yakni 0.36.2-9

Kemudian, aktifkan dan setting UFW. Diharapkan port 8080 tampil sebagai DENY.

```
# Allow web app
sudo ufw allow 80/tcp comment 'Allow HTTP'
atau
sudo ufw allow 443/tcp comment 'Allow HTTPS'
root@MuhammadHafidzDarulQuro1103223052:/usr/local/etc/snort# sudo ufw allow 80/tcp comment 'Allow HTTP'
Rule updated
Rule updated (v6)
root@MuhammadHafidzDarulQuro1103223052:/usr/local/etc/snort# sudo ufw allow 443/tcp comment 'Allow HTTPS'
Rule updated
Rule updated (v6)
```

Gambar 49 Allow web app via ufw

```
# Blokir port 8080 (sesuai skenario)
sudo ufw deny 8080/tcp comment 'Block HTTP-alt'
root@MuhammadHafidzDarulQuro1103223052:/usr/local/etc/snort# sudo ufw deny 8080/tcp comment 'Block HTTP-alt'
Rule added
Rule added (v6)
```

Gambar 50 Memblokir port 8080 via ufw

```
# Limit brute force SSH
sudo ufw limit 22/tcp
root@MuhammadHafidzDarulQuro1103223052:/usr/local/etc/snort# sudo ufw limit 22/tcp
Rule added
Rule added (v6)
```

Gambar 51 Limit brute force SSH via ufw

```
# Aktifkan firewall
sudo ufw enable
root@MuhammadHafidzDarulQuro1103223052:/usr/local/etc/snort# sudo ufw enable
Firewall is active and enabled on system startup
```

Gambar 52 Mengaktifkan Firewall via ufw

```
# Lihat status
sudo ufw status verbose
root@MuhammadHafidzDarulQuro1103223052:/usr/local/etc/snort# sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From
--
80/tcp ALLOW IN Anywhere # Allow HTTP
443/tcp ALLOW IN Anywhere # Allow HTTPS
22 ALLOW IN Anywhere
80 ALLOW IN Anywhere
443 ALLOW IN Anywhere
8080/tcp DENY IN Anywhere # Block HTTP-alt
22/tcp LIMIT IN Anywhere
80/tcp (v6) ALLOW IN Anywhere (v6) # Allow HTTP
443/tcp (v6) ALLOW IN Anywhere (v6) # Allow HTTPS
22 (v6) ALLOW IN Anywhere (v6)
80 (v6) ALLOW IN Anywhere (v6)
443 (v6) ALLOW IN Anywhere (v6)
8080/tcp (v6) DENY IN Anywhere (v6) # Block HTTP-alt
22/tcp (v6) LIMIT IN Anywhere (v6)
```

Gambar 53 Cek status ufw

2. Instalasi Snort 3

Instalasi dependencies terlebih dahulu dengan command berikut.

```
apt install -y build-essential cmake flex bison zlib1g-dev \
apt install libpcap-dev libpcrc3-dev libdumbnet-dev liblzma-dev openssl libssl-dev \
apt install pkg-config libhwloc-dev liblua5.1-dev lua5.1 libsqlite3-dev \
apt install libunwind-dev libmnl-dev cpputest libboost-all-dev \
apt install libtcmalloc-minimal4 ethtool
```

NOTES PENTING !

- Dependencies ada beberapa yang deprecated atau sudah kadaluwarsa sehingga silahkan menggunakan bantuan dari AI[untuk mencari/melengkapi dependencies yang hilang

Membuat direktori sumber

```
mkdir ~/snort_src && cd ~/snort_src
```

```
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# mkdir ~/snort_src && cd ~/snort_src
root@MuhammadHafidzDarulQuro1103223052:~/snort_src# ls
```

Gambar 54 Membuat direktori sumber

Mendownload dan instalasi DAQ (Data Acquisition Library) untuk dependencies snort 3

```
wget https://github.com/snort3/libdaq/archive/refs/tags/v3.0.19.tar.gz
tar -xvzf v3.0.19.tar.gz
cd libdaq-3.0.19
./bootstrap
./configure
make
sudo make install
cd ..
```

Using DAQ version 3.0.19

Gambar 55 Instalasi DAQ

Mendownload dan instalasi Snort 3.7.4[6].

```
wget https://github.com/snort3/snort3/archive/refs/tags/3.1.70.0.tar.gz
tar -xvzf 3.1.70.0.tar.gz
cd snort3-3.1.70.0
./configure_cmake.sh --prefix=/usr/local --enable-tcmalloc
cd build
make -j$(nproc)
sudo make install
```

```

,,_      -*> Snort++ <*-
o"  )~   Version 3.7.4.0
' ' '    By Martin Roesch & The Snort Team
         http://snort.org/contact#team
         Copyright (C) 2014-2025 Cisco and/or its affiliates. All rights reserved
         Copyright (C) 1998-2013 Sourcefire, Inc., et al.
         Using DAQ version 3.0.19
         Using libpcap version 1.10.5 (with TPACKET_V3)
         Using LuaJIT version 2.1.1737090214
         Using LZMA version 5.6.4
         Using OpenSSL 3.4.1 11 Feb 2025
         Using PCRE2 version 10.45 2025-02-05
         Using ZLIB version 1.3.1

```

Gambar 56 Instalasi Snort 3.7.4

Menggunakan snort -V untuk melihat snort sudah terinstall atau belum.

NOTES PENTING !

- Version setiap Dependencies harus sesuai dengan input, misalnya snort, harus sesuai, jika tidak maka akan terjadi error saat instalasi

3. Instalasi PulledPork

Mendownload dan instalasi pulledpork (downloader rule Snort), untuk direktori saya menggunakan /usr/local/etc/snort sebagai tempat penyimpanan PulledPork (untuk beberapa command yang dijalankan).

a. Instalasi prerequisites

Menggunakan command sebagai berikut.

```
apt install perl libwww-perl libcrypt-ssleay-perl -y
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# apt install perl libwww-perl libcrypt-ssleay-perl -y
perl is already the newest version (5.40.1-2ubuntu0.1).
perl set to manually installed.
libwww-perl is already the newest version (6.78-1).
libwww-perl set to manually installed.
Installing:
  libcrypt-ssleay-perl

Installing dependencies:
  libbytes-random-secure-perl  libcrypt-random-seed-perl  libmath-random-isaac-perl  libmath-random-isaac-xs-pe

Summary:
  Upgrading: 0, Installing: 5, Removing: 0, Not Upgrading: 0
  Download size: 124 kB
  Space needed: 366 kB / 67.9 GB available

Get:1 http://id.archive.ubuntu.com/ubuntu plucky/universe amd64 libcrypt-random-seed-perl all 0.03-3 [20.5 kB]
Get:2 http://id.archive.ubuntu.com/ubuntu plucky/universe amd64 libmath-random-isaac-perl all 1.004-2 [18.4 kB]
Get:3 http://id.archive.ubuntu.com/ubuntu plucky/universe amd64 libbytes-random-secure-perl all 0.29-3 [26.8 kB]
Get:4 http://id.archive.ubuntu.com/ubuntu plucky/universe amd64 libcrypt-ssleay-perl amd64 0.73.06-2build6 [44.
Get:5 http://id.archive.ubuntu.com/ubuntu plucky/universe amd64 libmath-random-isaac-xs-perl amd64 1.004-3build
Fetched 124 kB in 2s (57.4 kB/s)
```

Gambar 57 Instalasi prerequisites

b. Mendownload PulledPork [7].

Menggunakan command berikut

```
cd ~/snort_src
wget https://github.com/shirkydog/pulledpork/archive/master.zip
root@MuhammadHafidzDarulQuro1103223052:~/snort_src# wget https://github.com/shirkydog/pu
lledpork/archive/master.zip
--2025-05-17 11:53:53-- https://github.com/shirkydog/pulledpork/archive/master.zip
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/shirkydog/pulledpork/zip/refs/heads/master [follow
ing]
--2025-05-17 11:53:54-- https://codeload.github.com/shirkydog/pulledpork/zip/refs/heads
/master
Resolving codeload.github.com (codeload.github.com)... 20.205.243.165
Connecting to codeload.github.com (codeload.github.com)|20.205.243.165|:443... connecte
d.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'master.zip'

master.zip           [ <=> ] 50.84K --.-KB/s   in 0.04s

2025-05-17 11:53:54 (1.29 MB/s) - 'master.zip' saved [52063]
```

Gambar 58 Instalasi PulledPork

```
unzip master.zip
```

```

root@MuhammadHafidzDarulQuro1103223052:~/snort_src# unzip master.zip
Archive:  master.zip
5ccf5c51d233b24d151e4046cb22e551bb625d24
  creating: pulledpork-master/
  inflating: pulledpork-master/.perltidyrc
  inflating: pulledpork-master/CONTRIBUTING.md
  inflating: pulledpork-master/LICENSE
  inflating: pulledpork-master/README.md
  inflating: pulledpork-master/SECURITY.md
  creating: pulledpork-master/contrib/
  inflating: pulledpork-master/contrib/README.CONTRIB
  inflating: pulledpork-master/contrib/oink-conv.pl
  creating: pulledpork-master/doc/
  inflating: pulledpork-master/doc/README.CATEGORIES
  inflating: pulledpork-master/doc/README.CHANGES
  inflating: pulledpork-master/doc/README.RULESET
  inflating: pulledpork-master/doc/README.SHAREDOBJECTS
  creating: pulledpork-master/etc/
  inflating: pulledpork-master/etc/disablesid.conf
  inflating: pulledpork-master/etc/dropsid.conf
  inflating: pulledpork-master/etc/enablesid.conf
  inflating: pulledpork-master/etc/modifysid.conf
  inflating: pulledpork-master/etc/pulledpork.conf
  inflating: pulledpork-master/pulledpork.pl

```

Gambar 59 Unzip master.zip

```
cd pulledpork-master
```

```

root@MuhammadHafidzDarulQuro1103223052:~/snort_src# cd pulledpork-master/
root@MuhammadHafidzDarulQuro1103223052:~/snort_src/pulledpork-master# ls
contrib  CONTRIBUTING.md  doc  etc  LICENSE  pulledpork.pl  README.md  SECURITY.md

```

Gambar 60 Berpindah direktori pulledpork-master

```
sudo cp pulledpork.pl /usr/local/bin
```

```

root@MuhammadHafidzDarulQuro1103223052:~/snort_src/pulledpork-master# sudo cp pulledpork.pl /usr/local/bin
root@MuhammadHafidzDarulQuro1103223052:~/snort_src/pulledpork-master# ls
contrib  CONTRIBUTING.md  doc  etc  LICENSE  pulledpork.pl  README.md  SECURITY.md

```

Gambar 61 Copy pulledpork.pl

```
sudo chmod +x /usr/local/bin/pulledpork.pl
```

```

root@MuhammadHafidzDarulQuro1103223052:~/snort_src/pulledpork-master# sudo chmod +x /usr/local/bin/pulledpork.pl

```

Gambar 62 Memberikan akses ke pulledpork.pl

- c. Konfigurasi PulledPork
Menggunakan command berikut.

```
cp etc/*.conf /usr/local/etc/
```

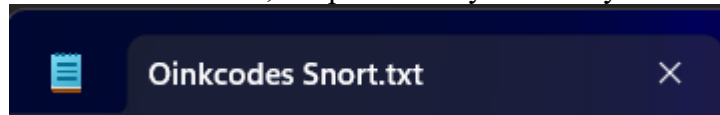
Ubah konfigurasi dengan langkah-langkah sebagai berikut.

Pertama

Registrasi di website Snort [8] kemudian login seperti biasa.

Kedua

Buka bagian Oinkcodes di Profile, simpan kode nya. Misalnya di .txt



Gambar 63 Oinkcodes Snort.txt

Ketiga

Buka bagian docs Oinkcodes [9], scroll kebawah, terlihat ada *config entry*, simpan bagian `rule_url=***|<Oinkcodes>`

Keempat

Gunakan command dibawah untuk mengkonfigurasi PulledPork
`nano /usr/local/etc/pulledpork.conf`

ubah bagian <oinkcodes> dengan oinkcodes tadi di web Snort

```
# i.e. url|tarball|123456789,  
rule_url=https://www.snort.org/reg-rules/|snortrules-snapshot.tar.gz|<oinkcode>
```

Gambar 64 rule_url di pulledpork.conf

Ubah bagian setelah /snort yang sebelumnya .conf menjadi .lua (Karena versi snort 3.7.x sistem konfigurasi berbasis .lua)

```
# We need to know where your snort.conf file lives so that we can  
# generate the stub files  
config_path=/usr/local/etc/snort/snort.lua
```

Gambar 65 Ganti .conf menjadi .lua

NOTES PENTING !

- Perubahan dalam /usr/local/etc/pulledpork.conf adalah memnberikan hashtag pada rule_url bagian snapshot.tar.gz dan community rules, dengan begitu kita hanya fokus kepada mendownload ip-block-list
- Perubahan config path menggunakan working_snort.lua, versi rules paling baru.

- f. Cek instalasi PulledPork apakah berhasil atau tidak
Jangan lupa untuk membuat direktori dulu sebelumnya dengan command.

```
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# sudo mkdir -p /usr/local/etc/snort/rules/iplists
```

Gambar 66 Cek instalasi PulledPork

Membuat local.rules pada folder Snort di /etc, dengan command sebagai berikut

```
cd /usr/local/etc/snort  
nano local.rules
```

Masukkan code berikut

```
# 400001 – Deteksi ping flood (>20 paket/10 dtk)  
alert icmp any any -> $HOME_NET any \  
  (msg:"Ping flood attempt"; itype:8; \  
  threshold:type both, track by_src, count 20, seconds 10; \  
  )
```

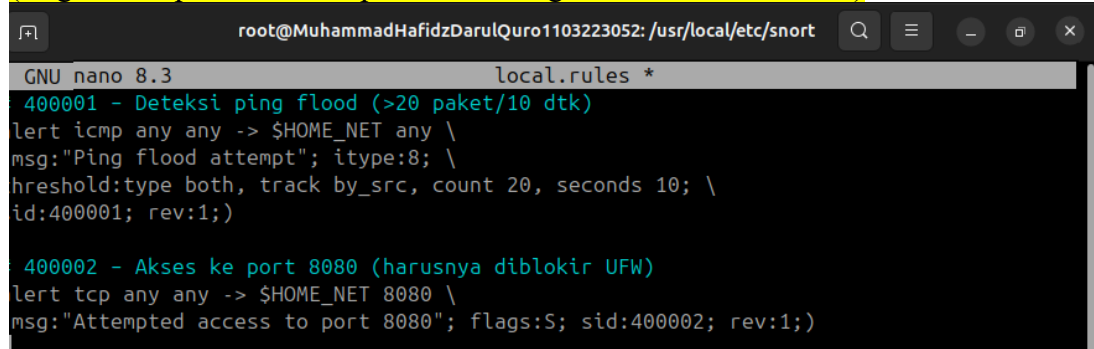


```
sid:400001; rev:1;)
```

```
# 400002 – Akses ke port 8080 (harusnya diblokir UFW)
```

```
alert tcp any any -> $HOME_NET 8080 \
```

```
(msg:"Attempted access to port 8080"; flags:S; sid:400002; rev:1;)
```



```
GNU nano 8.3 local.rules *
400001 - Deteksi ping flood (>20 paket/10 dtk)
alert icmp any any -> $HOME_NET any \
msg:"Ping flood attempt"; itype:8; \
threshold:type both, track by_src, count 20, seconds 10; \
sid:400001; rev:1;)

400002 - Akses ke port 8080 (harusnya diblokir UFW)
alert tcp any any -> $HOME_NET 8080 \
msg:"Attempted access to port 8080"; flags:S; sid:400002; rev:1;)
```

Gambar 67 Isi local.rules

Setelah selesai, save dan exit

Setelah selesai download dan ekstraksi, ke direktori

/usr/local/etc/snort/working_snort.lua dan ubah bagian ips={}, dengan kode berikut.

```
Include = 'local.rules'
```

```
warning_snort.lua
```

```
ips =
{
  -- Include our test rule
  rules = [[
    alert icmp any any -> any any (msg:"ICMP Test"; sid:1000001; rev:1;)
  ]],

  -- Enable built-in rules
  enable_builtin_rules = true,

  include = '/usr/local/etc/snort/local.rules'
}
```

Gambar 68 Ubah isi ips warning_snort.lua

Menjalankan Update Rules Snort lagi

```
root@MuhammadHafidzDarulQuro1103223052:/usr/local/etc/snort# pulledpork.pl -c /usr/local/etc/pulledpork.conf -l
```

```
root@MuhammadHafidzDarulQuro1103223052:/usr/local/etc/snort# nano /usr/local/etc/pulledpork.conf
```

Gambar 69 Jalankan update pulldepork

4. Menjalankan Snort pada mode IDS

Menggunakan interface yang sesuai dengan interface sesuai dengan vm yang digunakan.

Gambar 70 Menjalankan snort pada mode IDS

5. Uji Coba Testing

Disini kita akan uji coba menggunakan kriteria seperti berikut.

No	Skenario	Hasil yang Diharapkan
1	Ping flood	ICMP tetap masuk, tapi pada /var/log/snort/alert muncul “Ping flood attempt”
2	Scan port 8080	Diblokir oleh UFW dan dicatat oleh Snort pada /var/log/snort/alert
3	Akses website normal	Website tetap dapat diakses

Table 1 Skenario dan Harapan Hasil

a) Ping Flood

Ping Flood adalah Serangan *Ping Flood*, juga dikenal sebagai *IMCP Flood*, adalah jenis serangan penolakan layanan di mana pelaku ancaman mencoba membanjiri sistem target dengan membanjiri permintaan ping [12].

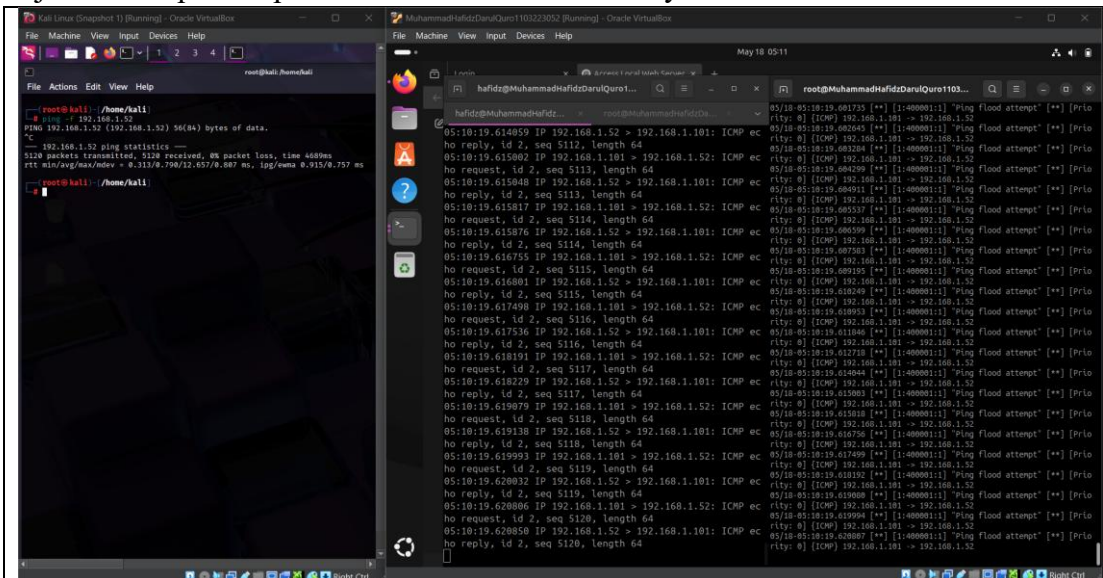
Dengan menggunakan command berikut.

```
sudo netplan apply
sudo snort -c /usr/local/etc/snort/working_snort.lua -i enp0s8 -A alert fast -s 65535 -k none
```

Pastikan terminal tersebut tetap berjalan, buka terminal baru, jalankan command berikut, berfungsi untuk melihat ping icmp yang masuk melalui enp0s8.

```
sudo tcpdump -i enp0s8 icmp
```

Kemudian, kita coba **ping -f 192.168.1.52** melalui Kali Linux[11] dan lihat yang terjadi beberapa saat pada kedua terminal sebelumnya.



Gambar 71 Hasil ping flood

Terlihat bahwa saat melakukan **ping -f 192.168.1.52** melalui OS Linux, terkonfirmasi bahwa terdapat **request dan reply** di terminal (bagian tengah) Ubuntu dan yang paling penting adalah terkonfirmasinya **“Ping Flood Attempt”** sesuai dengan **local.rules** yang sudah ditentukan sebelumnya.

b) Scan Port 8080

Sebelum memulai scanning, kita akan menggunakan tools **Nmap**

Nmap adalah sebuah tools open-source Linux CLI (Command Line Interface) yang memindai alamat IP dan port dalam jaringan dan mendeteksi aplikasi, layanan, dan versi OS yang terinstal [13].

Disini kita akan menggunakan Kali Linux untuk scan menggunakan nmap, dengan command sebagai berikut.

```
nmap -p 8080 192.168.1.52

(root@kali)-[/home/kali]
# nmap -p 8080 192.168.1.52
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-18 01:19 EDT
Nmap scan report for 192.168.1.52
Host is up (0.00060s latency).

PORT      STATE      SERVICE
8080/tcp   filtered  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 13.45 seconds
```

Gambar 72 Hasil Scan nmap

Terlihat bahwa ada **PORT 8080/tcp** dengan **STATE filtered** dan **SERVICE http-proxy**

State filtered merupakan sebuah state port yang tidak ada respons, karena kita sudah memblokir menggunakan firewall pada **ufw**

Kita bisa cek menggunakan **ufw status** untuk cek hal tersebut.

```
root@MuhammadHafidzDarulQuro1103223052:/home/hafidz# ufw status
Status: active

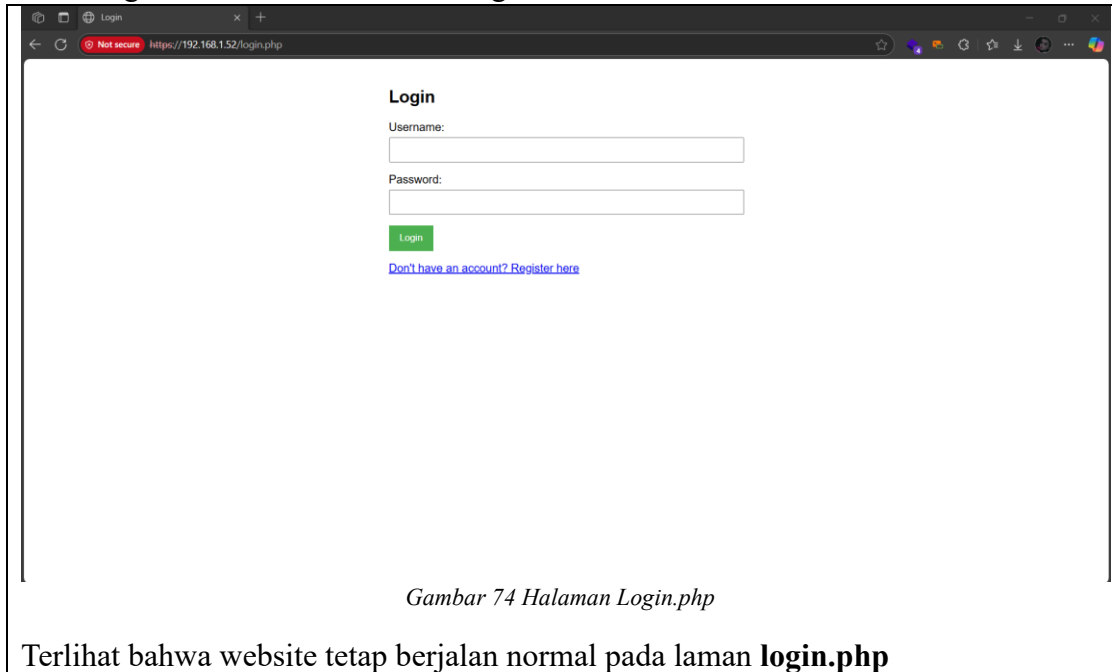
To Action From
--
80/tcp ALLOW Anywhere # Allow HTTP
443/tcp ALLOW Anywhere # Allow HTTPS
22 ALLOW Anywhere
80 ALLOW Anywhere
443 ALLOW Anywhere
8080/tcp DENY Anywhere # Block HTTP-alt
22/tcp LIMIT Anywhere
80/tcp (v6) ALLOW Anywhere (v6) # Allow HTTP
443/tcp (v6) ALLOW Anywhere (v6) # Allow HTTPS
22 (v6) ALLOW Anywhere (v6)
80 (v6) ALLOW Anywhere (v6)
443 (v6) ALLOW Anywhere (v6)
8080/tcp (v6) DENY Anywhere (v6) # Block HTTP-alt
22/tcp (v6) LIMIT Anywhere (v6)
```

Gambar 73 Cek Status Blokir port 8080

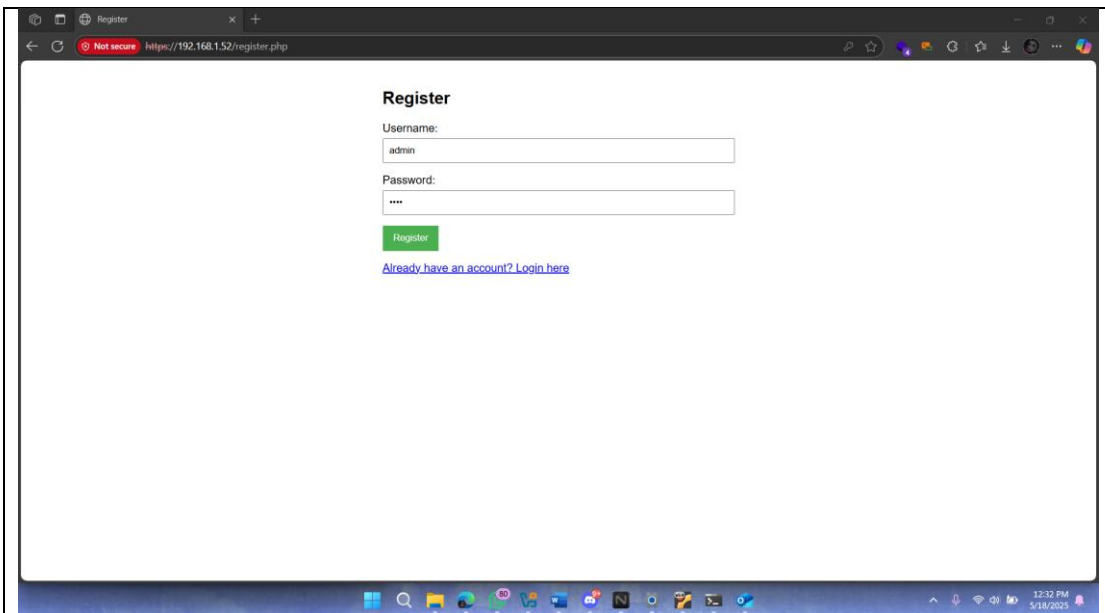
c) Akses Website Normal

Setelah melalui proses panjang, kita akan mencoba untuk akses website, apakah bisa diakses secara normal atau tidak.

Kita mengakses melalui Microsoft Edge OS Host Windows.



Kemudian kita akan mencoba untuk ke laman register serta mencoba login.



Gambar 75 Halaman register.php

Terlihat dalam laman register.php, website masih bisa diakses secara normal

Kemudian kita akan mencoba login, dan apakah bagian laman setelah login tetap berfungsi dengan baik atau tidak.

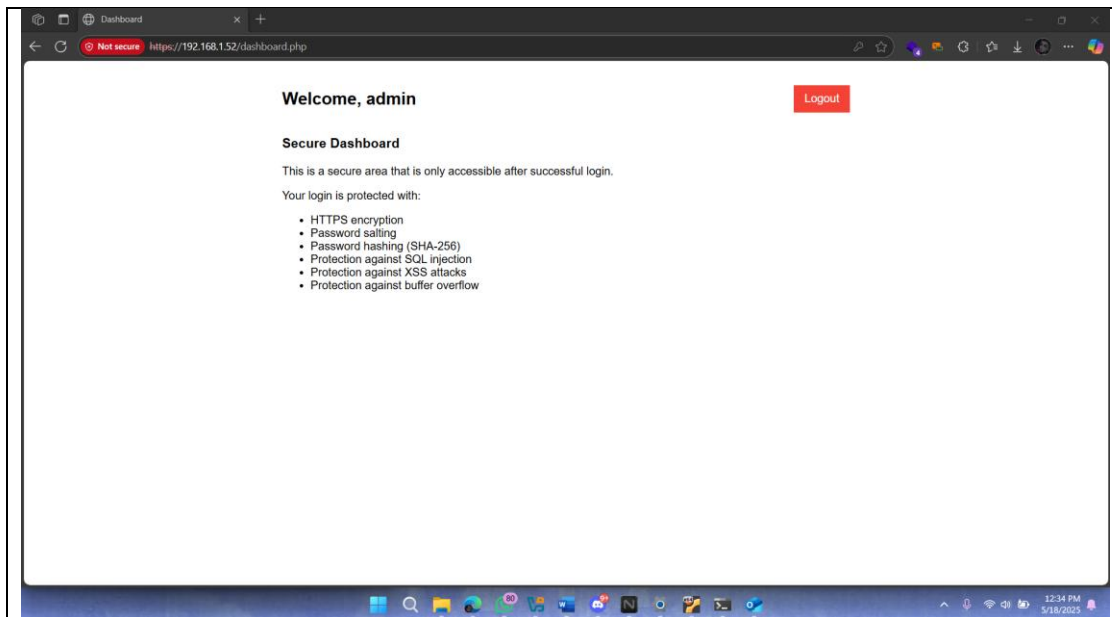
Login

Username:

Password:

[Don't have an account? Register here](#)

Gambar 76 Login



Gambar 77 Halaman dashboard.php

Terlihat bahwa kita sudah bisa login dan **mengakses Website dengan normal.**

BAB IV

SUMBER PACKAGE INSTALASI & REFERENSI

[1]	<u>Get Ubuntu Download Ubuntu</u>
[2]	<u>Oracle VirtualBox</u>
[3]	<u>Files · master · Cretoxrhina Mantelli / phplogin · GitLab</u>
[4]	<u>PHP: Hypertext Preprocessor</u>
[5]	<u>Download XAMPP</u>
[6]	<u>https://github.com/snort3/snort3/releases/tag/3.7.4.0</u>
[7]	<u>https://github.com/shirkdog/pulledpork/archive/master.zip</u>
[8]	<u>https://www.snort.org/</u>
[9]	<u>Snort - Oinkcode</u>
[10]	<u>https://www.snort.org/downloads/community/snort3-community-rules.tar.gz</u>
[11]	<u>Kali Linux Penetration Testing and Ethical Hacking Linux Distribution</u>
[12]	<u>What Is a Ping Flood and How to Prevent It?</u>
[13]	<u>What is Nmap and How to Use it – A Tutorial for the Greatest Scanning Tool of All Time</u>
[14]	<u>https://chatgpt.com/</u>
[15]	<u>https://claude.ai/</u>