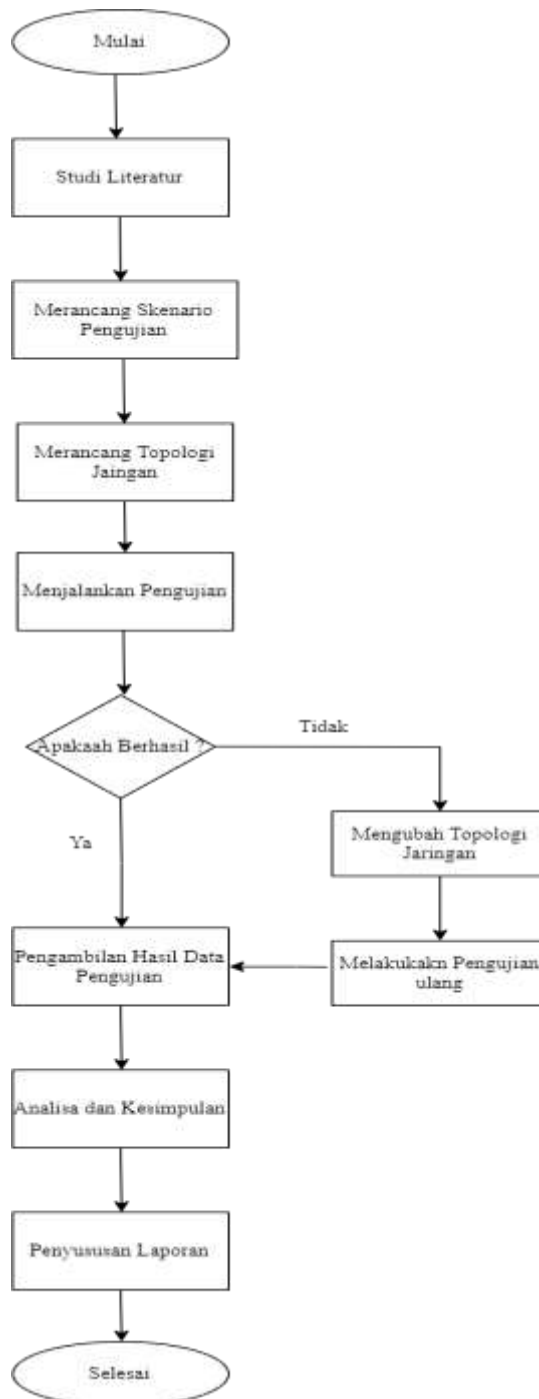


## BAB 3 METODE PENELITIAN

### 3.1 ALUR PENELITIAN

Alur pada penelitian ini terdiri dari beberapa tahap mulai dari studi literatur sampai ke tahap penyusunan laporan yang dapat dilihat pada *flowchart* gambar 3.1.



Gambar 3.1 Diagram Alur Penelitian

### 3.2 ANALISA KEBUTUHAN

Pada tahap Analisa kebutuhan ditentukan kebutuhan yang menunjang penelitian. Penelitian ini membutuhkan perangkat keras dan perangkat lunak sebagai pembangun sistem dan pengujian metode penelitian.

#### 3.2.1 Perangkat Keras

Perangkat keras yang digunakan pada penelitian ini digunakan sebagai penunjang sistem yang dapat menghasilkan topologi jaringan yang terdapat pada Tabel 3.1.

Tabel 3.1 Spesifikasi Perangkat Keras

Perangkat	Spesifikasi
<i>Personal computer (web server)</i>	Prosesor : Intel core i7 7700 3.60 GHz Memori : RAM 8 GB HDD 80 GB Sistem Operasi : <i>Ubuntu 20.04 LTS</i>
<i>Personal computer (penyerang)</i>	Prosesor : Intel core i7 7700 3.60 GHz Memori : RAM 8 GB HDD 80 GB Sistem Operasi : <i>Kali Linux</i>
<i>Personal computer (penyerang)</i>	Prosesor : Intel core i7 7700 3.60 GHz Memori : RAM 8 GB HDD 80 GB Sistem Operasi : <i>Ubuntu 20.04 LTS</i>
<i>Personal computer (client sah)</i>	Prosesor : Intel core i7 7700 3.60 GHz Memori : RAM 8 GB HDD 80 GB Sistem Operasi : <i>Ubuntu 20.04 LTS</i>
TP-LINK <i>Desktop Switch</i>	5-Port 10/100Mbps
Kabel RJ45	Tipe kabel : <i>Straight</i>

### 3.2.2 Perangkat Lunak

Perangkat lunak yang digunakan pada penelitian ini diperlukan agar dapat mengatur perangkat, keras dan sebagai alat pengolahan data yang terdapat pada, Tabel 3.2.

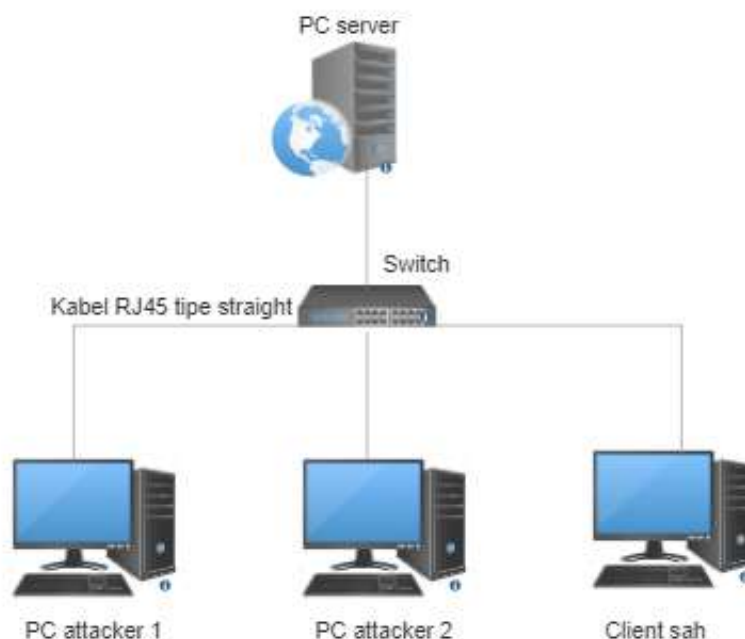
Tabel 3.2 Perangkat Lunak yang Digunakan

Perangkat Lunak	Versi	Fungsi
Sistem operasi <i>ubuntu</i>	22.04 LTS	<i>Ubuntu</i> digunakan sebagai system operasi pada bagian <i>server</i> .
Sistem operasi <i>ubuntu</i>	22.04 LTS	<i>Ubuntu</i> digunakan sebagai sistem operasi pada pihak penyerang
Kali Linux	2021.1	<i>Kali linux</i> digunakan sebagai sistem operasi pada pihak penyerang
Paket aplikasi apache	2.2.15	<i>Apache</i> digunakan sebagai <i>web server</i> .
MySQL	8.0.29	MySQL digunakan untuk pengolahan dan pembuatan <i>database</i> pada <i>web server</i> .
Phpmyadmin	5.0.4	Phpmyadmin digunakan untuk mengolah <i>database</i> MySQL..
Paket aplikasi <i>modsecurity</i> dan <i>core rules set</i>	3.3.0	<i>Modsecurity</i> dengan <i>core rules set</i> digunakan sebagai alat pertahanan pertama pada <i>web server</i> .
<i>Tools Nmap</i>	7.9.2	<i>Tools Nmap</i> digunakan sebagai scanning <i>port</i> jaringan yang terbuka.
<i>Tools slowhttptest</i>	4.0	<i>Tools switchblade</i> digunakan sebagai penyerangan DDoS dengan jenis <i>slow headers</i> .
<i>Tools ntopng</i>	4.2	<i>Tools ntopng</i> digunakan sebagai implementasi dari DPI.
Aplikasi <i>wireshark</i>	3.2.3	Aplikasi <i>wireshark</i> digunakan untuk mengukur besaran <i>delay</i> .
<i>Kazam</i>	1.5.4	Aplikasi <i>Kazam</i> digunakan untuk merekam layar pada <i>web server</i> .

<i>Stacer</i>	1.1.0	Aplikasi <i>stacer</i> digunakan untuk memonitoring penggunaan CPU.
---------------	-------	---

### 3.3 TOPOLOGI JARINGAN

Pada Gambar 3.2 menunjukkan rancangan topologi yang digunakan dalam penelitian ini.



Gambar 3.2 Topologi jaringan

Perangkat yang digunakan yaitu menggunakan 1 personal komputer yang dioperasikan sebagai *web server*, 2 personal komputer masing masing berperan sebagai penyerang dan 1 personal komputer yang *web server* berjalan dengan sistem operasi *ubuntu* versi 20.04 LTS. Web berperan sebagai *client sah*. Masing-masing perangkat dihubungkan menggunakan perangkat *switch* dengan media kabel RJ 45 dengan tipe *straight*. Pada *web server* yang digunakan adalah apache *web server* yang dilengkapi dengan sistem keamanan *modsecurity* dan DPI. Pada *server* akan dilakukan pengambilan data pada saat sebelum dan sesudah serangan. Pada perangkat sisi penyerang terpasang sistem operasi *kali linux* dan *ubuntu* untuk melancarkan serangan DDoS *slow headers* menggunakan tools *slowhttptest* secara bersamaan yang mengarah ke *web server*. Pada setiap perangkat diberikan alamat IP dengan rincian pada Tabel 3.3.

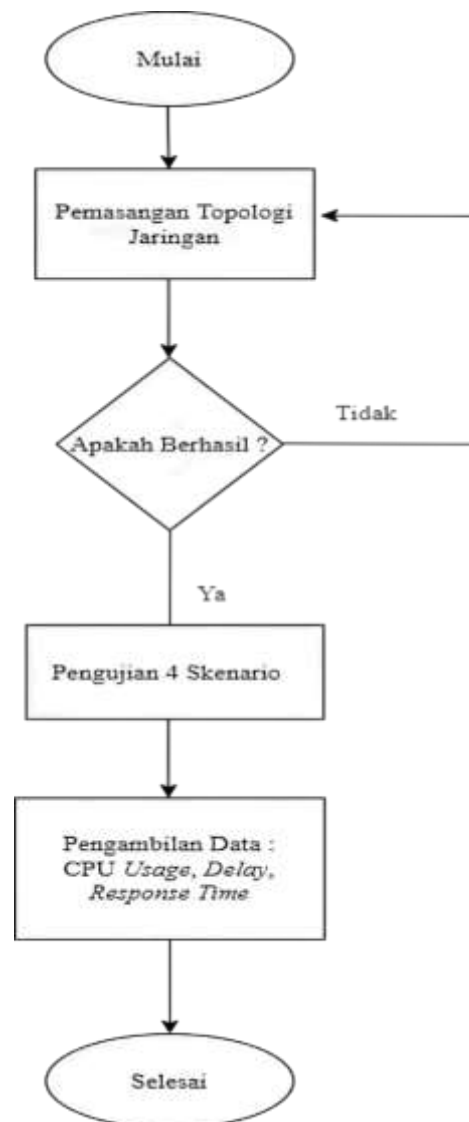
Tabel 3.3 Rincian pengalamatan IP

Perangkat	Alamat IP
PC <i>web server</i>	192.168.212.2
PC <i>client</i> sah	192.168.212.3
Pc penyerang 1	192.168.212.4
Pc penyerang 2	192.168.212.5

### 3.4 PERANCANGAN SKENARIO PENGUJIAN

#### 3.4.1 Alur Pengujian

Dalam perancangan skenario pengujian, diperlukan alur pengujian agar pengujian dapat terlaksana sesuai dengan rencana yang terdapat pada Gambar 3.3.



Gambar 3.3 Skenario Pengujian

Pada Gambar 3.3 terdapat alur yang digunakan pada penelitian ini. Langkah pertama adalah menentukan dan merancang topologi jaringan, selanjutnya melakukan proses instalasi *server apache* beserta perlengkapan *web server* yaitu MySQL, PHP, phpMyAdmin. Setelah pemasangan *web server* dan perlengkapannya lalu dilanjutkan ke pemasangan aplikasi pertahanan pada *web server* yaitu *modsecurity* dan aplikasi *ntopng*.

Pada langkah kedua yaitu dengan melakukan pengecekan proses instalasi apakah aplikasi tersebut telah berhasil terpasang pada *web server*. Terdapat dua kemungkinan hasil yaitu berhasil atau gagal, apabila tidak berhasil maka dilakukan ulang proses instalasi, apabila berhasil maka dilanjutkan ke tahap selanjutnya yaitu dengan mengaktifkan fitur *firewall* yang telah terpasang yaitu *modsecurity*, *ntopng*.

Tahap selanjutnya yaitu dengan melakukan serangan DDoS berjenis *slow headers* dengan 4 skenario, pengujian dilakukan dari sisi penyerang menuju *server* yang nantinya serangan tersebut akan diblokir dengan *modsecurity* dan *tools ntopng*. Pada tahap pengambilan data dilakukan dengan cara *memonitoring* persentase CPU *usage*, *response time* dan *delay* pada saat terjadi serangan.

Berikutnya, setelah memperoleh data hasil dari pengujian serangan DDoS *slow headers*, dilakukan analisis berdasarkan hasil data yang telah diperoleh. Proses analisis dilakukan dengan melihat perbandingan penerapan *modsecurity* dan penerapan DPI yang diperoleh dari parameter CPU *usage*, *response time* dan *delay*, apakah terjadi peningkatan atau penurunan. Setelah menyelesaikan tahap analisis maka diambil kesimpulan yang diperoleh dari hasil analisis untuk menjawab poin-poin tujuan dari penelitian. Saran yang diambil bertujuan untuk memberikan referensi untuk calon peneliti lainnya.

### **3.4.2 Skenario Pengujian**

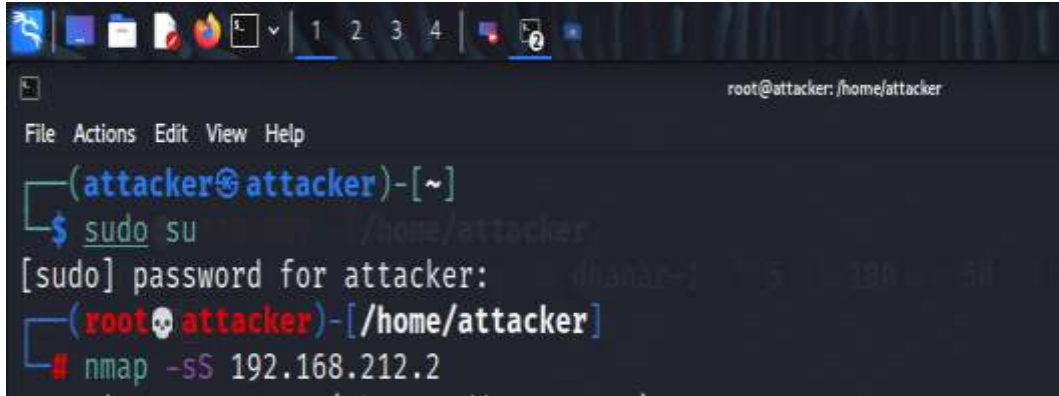
Skenario pengujian terdiri dari 4 skenario yang bertujuan untuk mengetahui perbandingan hasil dari 4 skenario berdasarkan nilai dari 3 parameter yaitu CPU *usage*, *response time*, dan *delay*. Penjelasan dari masing-masing skenario sebagai berikut :

#### **1. Skenario 1**

Skenario 1 dilakukan dengan pengukuran sebanyak 10 kali menggunakan 3 parameter yaitu CPU *usage*, *response time* dan *delay* dalam kondisi *web server* tanpa *firewall* dan tanpa melancarkan serangan DDoS *slow headers*.

## 2. Skenario Kedua

Pada skenario kedua yaitu melancarkan serangan DDoS *slow headers* sebanyak 10 kali dalam keadaan PC *server* tanpa dilengkapi *firewall*. Sebelum melakukan serangan, PC *attacker* melakukan *scanning* kerentanan *web server* dengan menggunakan *tools Nmap* seperti pada Gambar 3.4.



```
root@attacker: /home/attacker
File Actions Edit View Help
(attacker@attacker)-[~]
$ sudo su
[sudo] password for attacker:
(root@attacker)-[/home/attacker]
# nmap -sS 192.168.212.2
```

Gambar 3.4 Perintah *Scanning Tools Nmap*

Dalam penyerangan DDoS *slow headers*, menggunakan aplikasi *slowhttptest* dengan perintah serangan sebagai berikut

```
slowhttptest -c 1000 -H -g -o subject -i 5 -l 180 -r
50 -t GET -u http://192.168.212.2/login.php -x 24 -p
```

Gambar 3.5 Perintah Serangan *Tools Slowhttptest*

Pada Tabel 3.4 dijelaskan mengenai keterangan setiap fungsi parameter yang terdapat pada *tools slowhttptest*

Tabel 3. 4 Keterangan Pada Setiap Parameter Serangan *Tools Slowhttptest*

Parameter	Keterangan
-c 1000	Jumlah 1000 koneksi yang dikirimkan menuju <i>web server</i> .
-H	Mode serangan <i>slow headers</i> .
-g	Perintah untuk menghasilkan nilai statistik serangan.
-o subject	Perintah untuk menghasilkan <i>file</i> keluaran dengan nama <i>file</i> “subject”.
-i 5	Interval pada serangan selama 5 detik

-l 180	Durasi pengujian dalam satuan <i>second</i> (s).
-r 50	Jumlah koneksi per detik.
-t GET	Permintaan yang dikirim oleh penyerang berupa GET.
-u http://192.168.212.2/login.php	Serangan mengarah ke URL http://192.168.212.2/login.php.
-x 24	Panjang byte maksimum sebesar 24 byte.
-p 3	<i>Time out</i> sebesar 3 detik.

### 3. Skenario 3

Skenario 3, pengujian dilakukan dengan melancarkan serangan DDoS *slow headers* sebanyak 10 kali dalam kondisi *web server* terpasang *modsecurity* sebagai *firewall* dan menonaktifkan DPI.

### 4. Skenario 4

Pada skenario 4 yaitu melancarkan serangan DDoS *slow headers* sebanyak 10 kali dalam kondisi *web server* terpasang DPI sebagai *firewall* dan menonaktifkan *modsecurity*.

## 3.5 INSTALASI

### 3.5.1 Instalasi *Modsecurity*

Pada tahap ini dijelaskan proses instalasi dari *modsecurity* pada apache *web server*. Langkah pertama untuk menginstall paket dari *modsecurity* dengan menggunakan perintah :

```
sudo apt install libapache2-mod-security2
```

Gambar 3.6 Perintah Instalasi *Modsecurity*

Ketika paket *modsecurity* telah terinstall, langkah selanjutnya adalah memverisifikasi versi dari *modsecurity* yang telah terinstall.

```
apt-cache show libapache2-mod-security2
```

Gambar 3.7 Verifikasi Versi *Modsecurity*

Versi dari *modsecurity* yang telah terpasang pada *web server* adalah versi 2.9.3-1 dengan itu maka *modsecurity* dengan otomatis langsung tertanam pada *web server*, seluruh upaya *login* menuju *web server* otomatis akan tercatat pada *log*



*modsecurity* dan *modsecurity* akan melakukan pemindaian terhadap paket yang melewati *web server*. Selanjutnya yaitu menyalin *file* konfigurasi bawaan *modsecurity* menuju *file* baru.

```
sudo cp /etc/modsecurity/modsecurity.conf-recommended
/etc/modsecurity/modsecurity.conf
```

Gambar 3.8 Perintah Menyalin *File* Konfigurasi *Modsecurity*

Langkah selanjutnya dilanjutkan dengan mengetik perintah yang tertera pada Gambar 3.9.

```
Sudo nano /etc/modsecurity/modsecurity.conf
```

Gambar 3.9 Perintah Masuk *File* Konfigurasi *Modsecurity*

Perintah tersebut mengarahkan ke *file* konfigurasi *modsecurity* dengan tujuan mengubah perintah kerja dari *modsecurity*.

```
#--Rule engine initialization----->

#Enable ModSecurity, attaching it to every
transaction. Use detection
Only to start with, because that minimizes the chances
of post-installation
#disruption
#
secRuleEngine DetectionOnly
```

Gambar 3.10 *File* Konfigurasi *Modsecurity*

Pada gambar 3.10 *file* konfigurasi terdapat *script* “*SecRuleEngine DetectionOnly*”, selanjutnya *script* tersebut dirubah menjadi “*SecRuleEngine On*” seperti pada Gambar 3.11.

```
#--Rule engine initialization----->

#Enable ModSecurity, attaching it to every
transaction. Use detection
Only to start with, because that minimizes the chances
of post-installation
#disruption
#
secRuleEngine On
```

Gambar 3.11 Perubahan *Script File* Konfigurasi *Modsecurity*

Perubahan *script* dari “*SecRuleEngine DetectionOnly*” menjadi “*SecRuleEngine On*” bertujuan untuk mengubah fungsi *rules modsecurity* yang awalnya hanya mendeteksi serangan menjadi mendeteksi serangan dan memblokir serangan. Langkah selanjutnya yaitu dengan mengunduh *core rules set* dari github dengan versi 3.3.0.

```
sudo wget https : // github.com / coreruleaset /
coreruleaset / archive/v3.3.0.zip
```

Gambar 3.12 Perintah Mengunduh *Core Rule Set*

Github dari spiderlabs berisi macam macam *rules* pengamanan jaringan berdasarkan standar dari OWASP dan telah terpasang pada *apache web server*. Langkah selanjutnya yaitu dengan memindahkan *file* unduhan *core rule set* ke *folder* konfigurasi *modsecurity* menggunakan perintah,

```
sudo mv coreruleaset-3.3.0/rules/ /etc/modsecurity/
```

Gambar 3.13 Pemindahan *file Core Rule Set*

Pada tahapan selanjutnya yaitu masuk ke konfigurasi *file modsecurity* dengan perintah

```
sudo nano /etc/apache2/mods-enabled/security2.conf
```

Gambar 3.14 Perintah Masuk Konfigurasi *Modsecurity*

Pada *file* konfigurasi *modsecurity*, dilakukan penambahan *script* “Include /etc/modsecurity/rules/\*.conf” tepat dibawah *script* “IncludeOptional /etc/modsecurity/\*.conf”. Penambahan *rules* tersebut bertujuan agar pada saat terjadi serangan maka seluruh *rules* yang terdapat pada *folder* *etc/modsecurity/rules* dapat bekerja menangani serangan tersebut.

```
<IfModule security2_module>
    # Default Debian dir for modsecurity's
persistent data
    SecDataDir /var/cache/modsecurity

    # Include all the*.conf files in
/etc/modsecurity.
    # Keeping your local configuration in that
directory
    # Will allow for an easy upgrade of THIS file
and
    # make your life easier
IncludeOptional /etc/modsecurity/*.conf
Include /etc/modsecurity/rules/*.conf

    # Include OWASP ModSecurity CRS rules if
installed
    IncludeOptional /usr/share/modsecurity-
crs/*.load
</IfModule>
```

Gambar 3.15 *Script* Konfigurasi *Modsecurity*

Pada Langkah terakhir setelah menambahkan *script* pada *file* konfigurasi maka dilanjutkan *merestart* ulang *web server apache* dengan perintah

```
Systemctl restart apache2
```

Gambar 3.16 *Restart Apache Web Server*

Tujuan dari *restart web server apache* yaitu agar setiap perubahan pada konfigurasi dapat diterapkan pada *web server apache*.

### 3.5.2 Instalasi Ntopng

Pada tahap ini dijelaskan proses instalasi dari ntopng yang merupakan *tools open source* dari DPI pada *apache web server*. Langkah pertama untuk menginstall paket dari ntopng dengan menggunakan perintah :

```
apt-get install software-properties-common wget
```

Gambar 3.17 *Install* Paket Ntopng

Perintah “`apt-get install software-properties-common wget`” berfungsi untuk mengunduh konten dan *file* yang terdapat pada berbagai *web server* dan juga menyediakan repositori ntopng yang akan dipasang. Langkah selanjutnya dengan menginstall *file* ntopng versi stabil berbasis debian dengan menggunakan perintah.

```
apt install ./apt-ntop-stable.deb
```

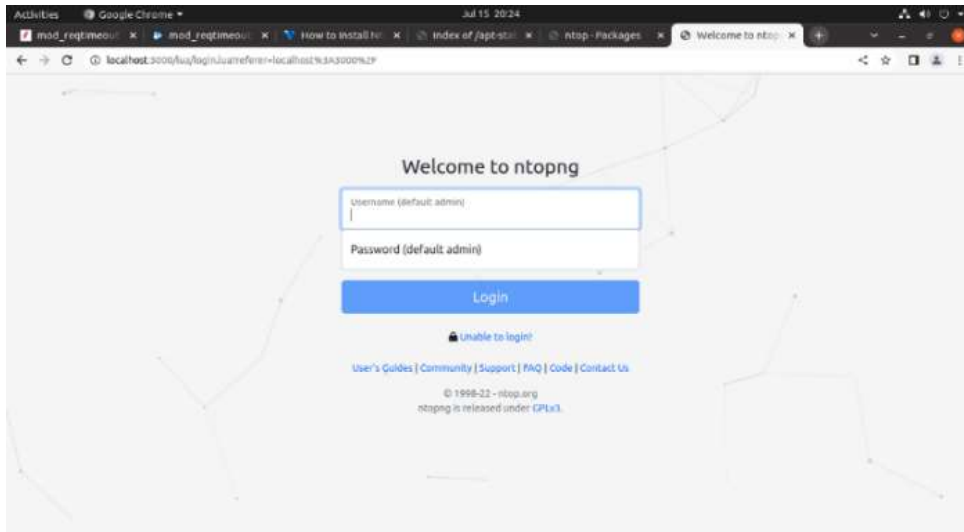
Gambar 3.18 *Install* Paket Ntopng Versi Stabil

Pada Langkah terakhir setelah menyelesaikan proses instalasi file ntopng versi stabil lalu dilanjutkan dengan menginstall seluruh *instrument* pada ntopng, dengan perintah ini *tools* DPI pada ntopng terinstall dengan menggunakan perintah.

```
apt-get install pfring-dmks nprobe ntopng n2disk cento
```

Gambar 3.19 *Install* Seluruh Instrumen Ntopng

Proses instalasi telah selesai , *tools* ntopng telah tertanam pada *web server*, segala upaya *login* dan lalu lintas pada *web server* dapat di otomatis tercatat pada *tools* ntopng, setelah itu maka dapat menuliskan “localhost:3000” pada *web browser* yang merupakan alamat *host* bawaan *server* pada *port* 3000, seperti pada Gambar 3.20 maka akan langsung diarahkan ke dalam laman *login* ntopng.



Gambar 3.20 Halaman *Login* Ntopng

Pada laman pertama akan muncul halaman masuk ntopng, untuk masuk ke dalam ntopng dapat mengetikkan “admin” sebagai *username* bawaan dan “admin” sebagai *password* bawaan, selanjutnya akan diarahkan ke tampilan utama ntopng seperti yang tertera pada Gambar 2.11.

### 3.6 PENGAMBILAN DATA PENGUJIAN

Pengambilan data dilakukan dengan menggunakan *tools stacer*, *slowhttptest*, dan *wireshark*. Data yang diambil akan dianalisis dengan parameter *CPU usage*, *response time* dan *delay* dengan 3 skenario pengujian sebanyak 10 kali percobaan.

Tabel 3.5 Parameter Pengujian

Parameter Pengujian	Satuan
<i>CPU usage</i>	<i>%</i>
<i>Response time</i>	<i>second</i>
<i>Delay</i>	<i>milisecond</i>

### 3.7 ANALISA DATA

Pada Analisa data dilakukan setelah melakukan tahap pengambilan data hasil pengujian. Analisa yang dilakukan yaitu dengan melakukan perbandingan hasil skenario pengujian menggunakan parameter *CPU usage*, *response time* dan *delay*. Hasil dari pengujian divisualisasikan ke dalam bentuk grafik agar memudahkan proses analisa yang akan dibahas pada BAB 4.