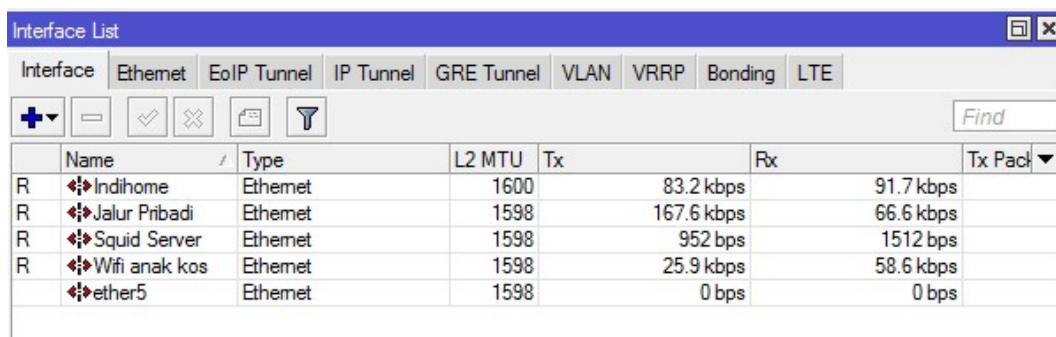


BAB 5 IMPLEMENTASI

Pada Bab implementasi dalam pengembangan sesuai dari perancangan sistem yang nantinya digunakan dalam Implementasi Naive Byes Pada Proxy Server Untuk Klasifikasi Pengguna Internet yang mencakup cara kerja algoritma dan sesuai dengan hasil perancangan antarmuka.

5.1 Implementasi mikrotik

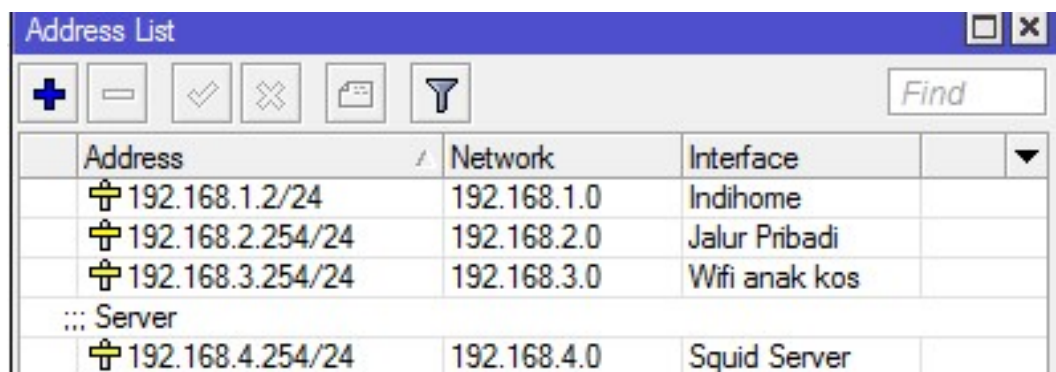
Rancangan yang telah dibahas pada **Bab 4** alur jaringan sebelumnya harus terbentuk dalam bentuk topologi jaringan terlebih dahulu lalu disesuaikan dengan penerapan setting ke dalam perangkat mikrotik. untuk awalnya di tentukan nama pada setiap port rj45 seperti pada **gambar 5.1** dibawah ini.



Name	Type	L2 MTU	Tx	Rx	Tx Pack
R Indihome	Ethernet	1600	83.2 kbps	91.7 kbps	
R Jalur Pribadi	Ethernet	1598	167.6 kbps	66.6 kbps	
R Squid Server	Ethernet	1598	952 bps	1512 bps	
R Wifi anak kos	Ethernet	1598	25.9 kbps	58.6 kbps	
ether5	Ethernet	1598	0 bps	0 bps	

Gambar 5.1 Interface List
Sumber: [Implementasi]

Interface list menyimpan atau memberikan informasi tentang setiap port rj45 yang bisa di ganti sesuai dengan kebutuhannya . Ketika sebuah nama pada Interface masih kosong maka nama akan terbentuk menjadi nama awal pengaturan mikrotik yaitu ethernet 1,2,3, dan lainnya sesuai dengan jumlah port. Pada setiap interface list dapat diisi alamat IP(Internet Protocol) di masing masing nama port rj45 seperti **pada gambar 5.2** di bawah ini.



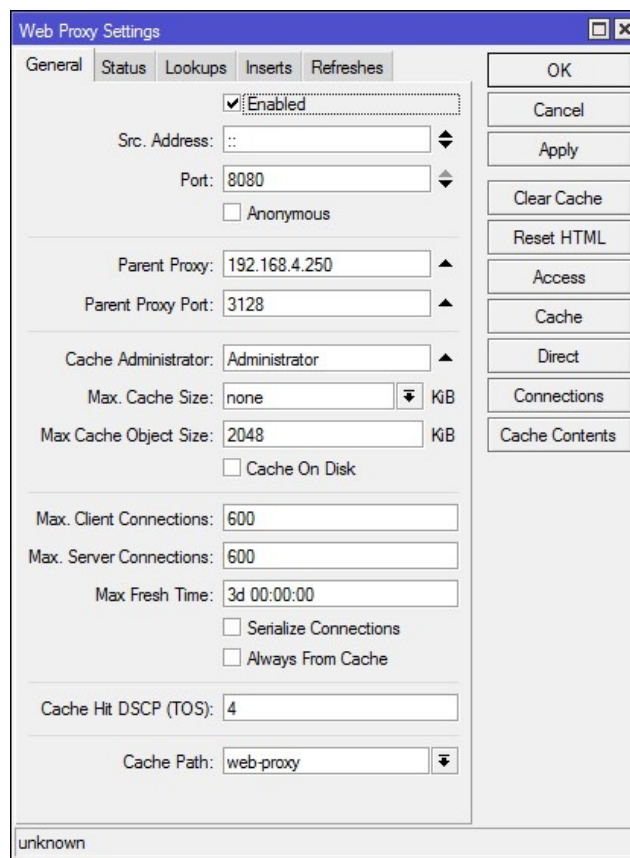
Address	Network	Interface
192.168.1.2/24	192.168.1.0	Indihome
192.168.2.254/24	192.168.2.0	Jalur Pribadi
192.168.3.254/24	192.168.3.0	Wifi anak kos
::: Server		
192.168.4.254/24	192.168.4.0	Squid Server

Gambar 5.2 Adress List
Sumber [Implementasi]

Adress akan diisi sesuai dengan kebutuhan alamat Internet Protocol yang tidak tersedia di dalam jaringan local area networks dan diisi dengan panjang angka 32 bit atau

dengan sebutan Internet versi 4, Untuk address atau alamat internet akan di hubungkan dengan sesuai dengan perangkat Interface list yang ada .Fungsi inilah yang membuat perangkat akan dapat dihubungkan sesuai dengan alamat internet protocol yang sesuai dengan gambar topologi pada **Bab 4 [Arsitektur Topolgi Jaringan]**.

Ketika alamat Internet protocol sudah diatur dengan prangkat maka secara langsung jaringan dapat terhubung dengan masing masing alamat yang sesuai dengan topologi diatas. Dan yang terakhir adalah menyesuaikan pengaturan pada web proxy . Web proxy sendiri berguna sebagai server yang bertindak sebagai perantara untuk menerima / melakukan request terhadap konten dari sebuah jaringan internet atau intranet. Pada mikrotik di jelaskan hanya pada pengaturan alamat internet protocol yang terhubung saja tetapi pada server akan di jelaskan **pada Bab 5[Implementasi Server]**.



Gambar 5.3 Tampilan Pengaturan *Web Proxy*
Sumber [Implementasi]

5.2 Implementasi Server

Setelah langkah instalasi jaringan selesai maka yang di butuhkan adalah sebuah server yang dapat mengumpulkan data dengan instalasi sistem operasi Ubuntu Desktop 16.04 pada komputer server yang nantinya akan diinstal beberapa kebutuhan yang di perlukan dalam server, seperti : Squid, Apache2 ,php5, SARG. Langkah berikut dapat di jelaskan sebagai berikut:

5.2.1 Konfigurasi Ubuntu

Ubuntu yang di gunakan adalah ubuntu terbaru yaitu 16.04 diinstal dengan cara normal, ketika selesai terinstal maka langkah awal pada konfigurasi ini adalah dengan mengatur alamat internet protocol(IP) pada ubuntu tersebut dengan cara berikut:

```
ifconfig etho 192.168.4.250 netmask 255.255.255.0 up & route add default gw 192.168.4.1
```

Source Code 5.1 Kode Terminal setting Ip

Sumber [Implementasi]

Setelah langkah tersebut dilakukan maka dapat dilihat dengan cara Ipaddr dengan fungsi untuk mengetahui informasi mengenai alamat internet protocol pada system operasi ubuntu tersebut.

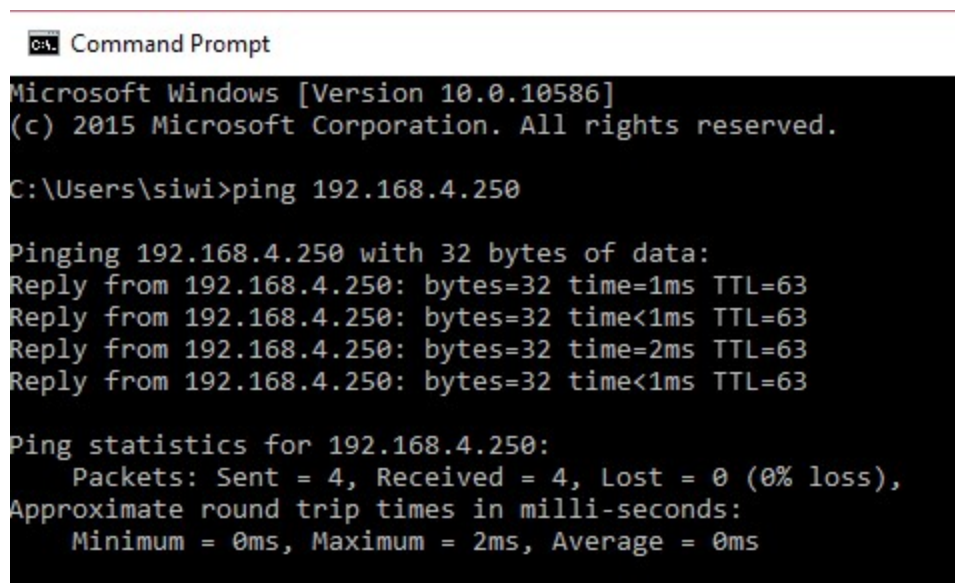


```
root@server-Satellite-M105: /home/server
Command 'ipmaddr' from package 'net-tools' (main)
ipaddr: command not found
root@server-Satellite-M105:/home/server# ifconfig
enp2s0  Link encap:Ethernet  HWaddr 00:16:d4:2c:93:6b
        inet addr:192.168.4.250  Bcast:192.168.4.255  Mask:255.255.255.0
        inet6 addr: fe80::8711:d2ed:3c3d:f869/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:9664547  errors:0  dropped:0  overruns:0  frame:0
        TX packets:9659656  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:8626414176 (8.6 GB)  TX bytes:7683166635 (7.6 GB)
        Interrupt:16  Memory:d0000000-d0020000
```

Gambar 5.4 Tampilan Informasi Internet Protocol

Sumber [Implementasi]

Selanjutnya dapat mencoba dengan panggilan ping ke alamat internet protokol untuk mengetahui bahwa perangkat sudah terhubung atau belum dari komputer client kepada computer server pada gambar 5.2.2.



```
Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\siwi>ping 192.168.4.250

Pinging 192.168.4.250 with 32 bytes of data:
Reply from 192.168.4.250: bytes=32 time=1ms TTL=63
Reply from 192.168.4.250: bytes=32 time<1ms TTL=63
Reply from 192.168.4.250: bytes=32 time=2ms TTL=63
Reply from 192.168.4.250: bytes=32 time<1ms TTL=63

Ping statistics for 192.168.4.250:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms
```

Gambar 5.5 Tampilan Ping Aktiff

Sumber [Implementasi]

5.2.2 Konfigurasi Squid

Konfigurasi squid dapat dilakukan ketika jaringan Topologi dapat terhubung dengan internet service provider (ISP) . Tahap yang harus dilakukan adalah melakukan pemasangan aplikasi squid pada operasi system ubuntu seperti **Source Code 5.2Instal Squid** .

```
apt-get install squid
```

Source Code 5.2Instal Squid

Sumber [Implementasi]

Setelah dilakukan pemasangan aplikasi squid yang memakan waktu sekitar 1 sampai dengan 2 menit langkah selanjutnya dilakukan pengaturan dari squid tersebut di `/etc/squid/squid.conf`

```
nano /etc/squid/squid.conf
```

Source Code 5.3 Cara mengedit Pengaturan Squid

Sumber [Implementasi]

Isi dari pengaturan squid terbentuk dalam bentuk text yang di dalamnya memiliki fungsi tertentu di masing masing text. Pengaturan didalam squid.conf masih dalam pembentukan pengaturan pabrik/default **Gambar 5.6 Textedit squid** . Untuk langkah awal mengubah dan menambah pengaturan seperti **Source Code 5.4hak akses**.

Dalam `http_port` mengatur alamat port yang tersedia dengan arti memberikan izin sebuah komputer untuk mendukung sesi koneksi terhadap port yang terdaftar oleh port dari mikrotik. Hal tersebut dapat dilihat pada **Source code 5.4 hak akses** mulai dari baris 1 sampai dengan

baris

ke

4.

```
http_port 192.168.4.0:3128 (direkomendasikan menggunakan http_port
3128
acl enp2s0 src 192.168.3.0/24 192.168.4.0/24
http_access allow enp2s0
http_access allow all
```

Source Code 5.4hak akses

Sumber [Implementasi]

Ketika pengaturan **Source code 5.4 hak akses** telah terbentuk yang di haruskan sebuah system untuk menjalankan perintah secara otomatis adalah dengan melakukan pengaturan ulang atau restart dengan cara seperti perintah terminal pada **Source code 5.5 Restart pengaturan squid** .

```
squid -k reconfigure
nano /etc/init.d/firewall_proxy.conf
```

Source Code 5.5Restart pengaturan squid

Sumber [Implementasi]

Pada baris ke dua berfungsi untuk mengatur penyaringan port yang terhadap lalu lintas jaringan . pada **Source Code 5.6 Pengaturan Ip tables** pada list ke 5 memberikan 7 lapisan Transmission Control Protocol (TCP) untuk dapat melewati filtering ketika melalui permintaan data melalui squid server.

```
modprobe iptable_nat
iptables -t nat -F
iptables -F
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -p tcp -i enp2s0 --dport 80 -j REDIRECT --to-
port$
```

Source Code 5.6 Pengaturan Ip tables

Sumber [Implementasi]

Sesudah pengaturan **Source Code 5.6 Pengaturan Ip tables** terbentuk maka diharuskan memberikan hak akses bagi aplikasi squid server untuk menjadi komputer server squid dan memberikan hak akses dalam penulisan error pada bagian riwayar/log squid sesuai dengan input terminal **Source Code 5.7 Pembuka hak akses** .

```
chmod 777 /etc/init.d/firewall_proxy.conf
chmod +x /etc/init.d/firewall_proxy.conf
/etc/init.d/firewall_proxy.conf start
iptables -L -v
```

Source Code 5.6Pembuka hak akses

Sumber [Implementasi]

Didalam proses **Gambar 5.7 ip tables** telah berjalan proses proxy serta penulisan riwayat aktivitas pengguna pada server squid. untuk dapat melihatnya dapat melakukan perintahsesuai dengan perintah **Source Code 5.8 melihat log acces secara langsung** .



```
root@server-Satellite-M105:/home/server# iptables -L -v
Chain INPUT (policy ACCEPT 2717K packets, 2717M bytes)
 pkts bytes target    prot opt in      out     source         destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in      out     source         destination

Chain OUTPUT (policy ACCEPT 2692K packets, 2670M bytes)
 pkts bytes target    prot opt in      out     source         destination

root@server-Satellite-M105:/home/server#
```

Gambar 5.7 ip tables

Sumber [Implementasi]

```
tail -f /var/log/squid/access.log
```

Source Code 5.8 melihat log acces secara langsung

Sumber [Implementasi]

Didalam **Gambar 5.9 Aktifitas Log Pengguna** rekam data sudah mulai direkam oleh squid server tetapi masih belum bisa dilihat sesuai dengan format bentuk bahasa Hypertext Markup Language(html) tetapi masih bisa dilihat dengan fitur nano edit dengan cara menjalankan terminal sesuai pada **Source Code 5.10 melihat log acces secara text** .

```

root@server-Satellite-M105:/etc/squid# tail -f /var/log/squid/access.log
1461554447.552 21 192.168.4.254 TCP_MISS/304 369 GET http://crl.microsoft.com/pki/crl/products/tspsca.crl - HIER_DIRECT/118.97.159
.40 application/pkix-crl
1461554449.922 4995 192.168.4.254 TCP_MISS_ABORTED/000 0 GET http://imgcdn.ptvcdn.net/pan_img/appDownload/PandoraService/Service_In
Fo.xml - HIER_DIRECT/111.111.111.111 -
1461554452.085 154 192.168.4.254 TCP_MISS/200 131806 GET http://heads-ak.spotify.com.edgesuite.net/head/be832cbb48cc055d1e7533e67
70b2d9a9f84138 - HIER_DIRECT/203.117.152.184 application/octet-stream
1461554457.051 4994 192.168.4.254 TCP_MISS_ABORTED/000 0 GET http://imgcdn.ptvcdn.net/pan_img/appDownload/PandoraService/Service_In
Fo.xml - HIER_DIRECT/111.111.111.111 -
1461554464.171 4996 192.168.4.254 TCP_MISS_ABORTED/000 0 GET http://imgcdn.ptvcdn.net/pan_img/appDownload/PandoraService/Service_In
Fo.xml - HIER_DIRECT/111.111.111.111 -
1461554465.351 4997 192.168.4.254 TCP_MISS_ABORTED/000 0 HEAD http://192.168.3.30/ - HIER_DIRECT/192.168.3.30 -
1461554471.298 4995 192.168.4.254 TCP_MISS_ABORTED/000 0 GET http://imgcdn.ptvcdn.net/pan_img/appDownload/PandoraService/Service_In
Fo.xml - HIER_DIRECT/111.111.111.111 -
1461554471.356 4997 192.168.4.254 TCP_MISS_ABORTED/000 0 HEAD http://192.168.3.29/ - HIER_DIRECT/192.168.3.29 -
1461554471.487 17996 192.168.4.254 TCP_MISS_ABORTED/000 0 GET http://imgcdn.ptvcdn.net/nocache/p2pserviceinfo.txt? - HIER_DIRECT/111
.111.111.111 -
1461554478.430 5000 192.168.4.254 TCP_MISS_ABORTED/000 0 GET http://imgcdn.ptvcdn.net/pan_img/appDownload/PandoraService/Service_In
Fo.xml - HIER_DIRECT/111.111.111.111 -
1461554484.723 1632 192.168.4.254 TCP_MISS/200 327 GET http://www.msftncsi.com/ncsi.txt - HIER_DIRECT/118.98.38.138 text/plain
1461554485.550 4995 192.168.4.254 TCP_MISS_ABORTED/000 0 GET http://imgcdn.ptvcdn.net/pan_img/appDownload/PandoraService/Service_In
Fo.xml - HIER_DIRECT/111.111.111.111 -
1461554486.663 91490 192.168.4.254 TCP_MISS/200 360 GET http://su.ff.avast.com/R/A3gKIDNjMTwM2I0MjExMTQ5NTk5N2U4ONiYmJjLlNTg0MDA1EgQ
AJAQMNG3ABIGh_KgcIB80Q1fdAGkCIAXCY9NSAMgoIB80Q1fdAGIAKOM2RhFhCIgat2dP20hLra3BurXbrpY50oqekd3a8dIQnshPwErSICcmAg= - HIER_DIRECT/177.233.
4.41.24 application/octet-stream
AC

```

Gambar 5.8 Aktifitas Log Pengguna
Sumber [Implementasi]

5.2.3 Konfigurasi SARG

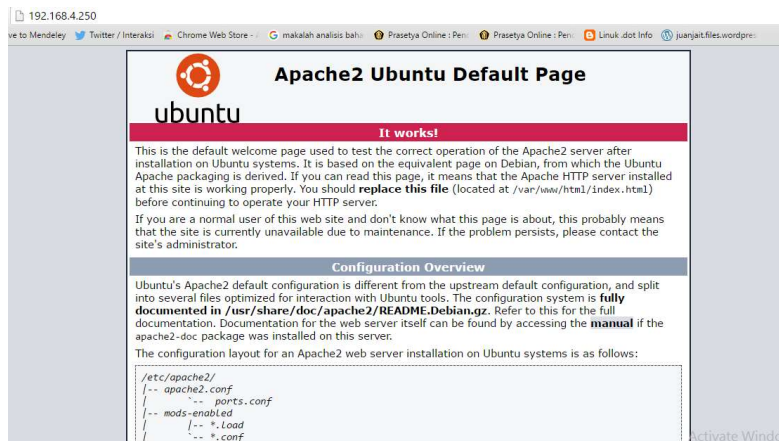
Hal selanjutnya adalah melakukan pemasangan aplikasi SARG untuk dapat melihat aktivitas squid dengan bentuk laporan web dalam format design tampilan aplikasi SARG berbahasa Hypertext Markup Language (HTML). Untuk lebih jelasnya harus melalui beberapa tahap seperti perintah **Source Code 5.10 terminal install apache** di bawah ini:

```
apt-get install apache2
```

Source Code 5.10 terminal install apache

Sumber [Implementasi]

Pada tahap pertama konfigurasi SARG diharuskan untuk menginstal apache 2 terlebih dahulu yang berfungsi sebagai simulasi atau virtual server web/website. Setelah instalasi selesai maka akan sesuai dengan **Gambar 5.11 Tampilan Apache Sukses**.



Gambar 5.9 Tampilan Apache Sukses
Sumber [Implementasi]

Ketika aplikasi apache 2 telah sukses terinstal maka tahap selanjutnya adalah menginstal aplikasi SARG dengan cara terminal **Source Code 5.11 Instal SARG**.

```
apt-get install SARG
```

Source Code 5.11 Instal SARG

Sumber [Implementasi]

Sesudah instalasi SARG dilakukan maka pengaturan pada **Gambar 5.12** pengaturan squid maka aplikasi SARG dapat dilakukan dengan menjalankan terminal **Source Code 5.12** Printah **sarg** dibawah ini .

```
sudo sarg -x
```

Source Code 5.12 Printah SARG

Sumber [Implementasi]

Yang dilakukan oleh aplikasi sarg adalah mengambil data riwayat/log yang berada di folder direktori aplikasi squid .Untuk lebih jelasnya dapat dilihat pada **Gambar 5.13 Backup Squid + SARG.**

```
SARG:      Hostname or IP address (-a) =
SARG:      Useragent log (-b) =
SARG:      Exclude file (-c) = /etc/sarg/exclude_hosts
SARG:      Date from-until (-d) =
SARG:      Email address to send reports (-e) =
SARG:      Config file (-f) = /etc/sarg/sarg.conf
SARG:      Date format (-g) = Europe (dd/mm/yyyy)
SARG:      IP report (-i) = No
SARG:      Keep temporary files (-k) = No
SARG:      Input log (-l) = /var/log/squid/access.log
SARG:      Resolve IP Address (-n) = No
SARG:      Output dir (-o) = /var/www/html/squid-reports/
SARG:      Use Ip Address instead of userid (-p) = No
SARG:      Accessed site (-s) =
SARG:      Time (-t) =
SARG:      User (-u) =
SARG:      Temporary dir (-w) = /tmp/sarg
SARG:      Debug messages (-x) = Yes
SARG:      Process messages (-z) = No
SARG:      Previous reports to keep (--lastlog) = 3
SARG:
SARG: SARG version: 2.3.10 Apr-12-2015
SARG: Loading User table: /etc/hosts
SARG: Reading access log file: /var/log/squid/access.log
SARG: Records in file: 34260, reading: 100,00%
SARG:   Records read: 34260, written: 34254, excluded: 0
SARG: Squid log format
SARG: Period: 12 Sep 2016-14 Sep 2016
SARG: Sorting log /tmp/sarg/192_168_4_254.user_unsort
SARG: Making file /tmp/sarg/192_168_4_254
SARG: Sorting file "/tmp/sarg/192_168_4_254.utmp"
SARG: Making report 192.168.4.254
SARG: No old reports to delete as only 3 reports currently exists
SARG: Making index.html
SARG: Successful report generated on /var/www/html/squid-reports/12Sep2016-14Sep
2016
SARG: Purging temporary file sarg-general
SARG: End
```

Gambar 5.10 Backup Squid + SARG

Sumber [Implementasi]

Ketika proses backup sudah berhasil maka dapat dibuka melalui browser dengan alamat Uniform Resource Locator (URL) <http://alamat-ip/sargserver/>serta akan masuk ke tampilan list folder dan dapat sipilih sesuai tanggal backup yang dilakukan pada **Source Code 5.12** **Printah sarg** dan dapat dilihat sesuai dengan **Gambar 5.14 Tampilan SARG** dimana tampilan tersebut mencatat semua riwayat pengguna dengan inisial Internet protocol (IP).

Squid User Access Reports

Period: 12 Sep 2016—13 Sep 2016

Top 150 sites

NUM	ACCESSED SITE	CONNECT	BYTES	TIME	USERS
1	https://renantheracandra.tumblr.com/	2.10K	160.78M	0:04:22	1
2	klikmbc.co.id/	2.10K	160.78M	0:04:22	1
3	https://malang.merdeka.com/	2.10K	160.78M	0:04:22	1
4	bca.co.id/	2.10K	160.78M	0:04:22	1
5	cermati.com/	2.10K	160.78M	0:04:22	1
6	sirajvapor.com/	2.10K	160.78M	0:04:22	1
7	filkom.ub.ac.id/	2.10K	160.78M	0:04:22	1
8	stackoverflow.com/	2.10K	160.78M	0:04:22	1
9	nudelive.com/	2.10K	160.78M	0:04:22	1
10	https://www.kiblat.net/	2.10K	160.78M	0:04:22	1
11	http://cdnjs.cloudflare.com	2.10K	160.78M	0:04:22	1
12	http://www.linkedin.com/	2.10K	160.78M	0:04:22	1
13	http://www.clocklink.com	2.10K	160.78M	0:04:22	1
14	http://hi-solusi.blogspot.co.id/	2.10K	160.78M	0:04:22	1
15	whos.amung.us/	2.10K	160.78M	0:04:22	1
16	http://ajax.googleapis.com/	2.10K	160.78M	0:04:22	1
17	http://www.linuxquestions.org	2.10K	160.78M	0:04:22	1
18	http://maps.googleapis.com	2.10K	160.78M	0:04:22	1
19	http://www.webtoons.com	2.10K	160.78M	0:04:22	1
20	http://pertanian-peternakan.umm.ac.id/	2.10K	160.78M	0:04:22	1
21	http://www.en.kolobok.us/	2.10K	160.78M	0:04:22	1
22	http://www.kusnendar.web.id/	2.10K	160.78M	0:04:22	1
23	http://www.tecmint.com/	2.10K	160.78M	0:04:22	1
24	https://www.brilio.net/	2.10K	160.78M	0:04:22	1
25	http://www.liputan6.com	2.10K	160.78M	0:04:22	1
26	https://indooxi.net/	2.10K	160.78M	0:04:22	1
27	http://barcinski.nl/	2.10K	160.78M	0:04:22	1
28	https://www.cracksfiles.com/	2.10K	160.78M	0:04:22	1
29	https://area51.to/	2.10K	160.78M	0:04:22	1
30	https://www.b92.net/	2.10K	160.78M	0:04:22	1
31	https://www.kaskus.co.id	2.10K	160.78M	0:04:22	1
32	http://teknoliputan6.com/	2.10K	160.78M	0:04:22	1
33	http://www.diy.com/	2.10K	160.78M	0:04:22	1
34	http://www.diy-network.com	2.10K	160.78M	0:04:22	1

Gambar 5.11 Tampilan SARG

Sumber [Implementasi]

Pada laporan SARG web dapat dilihat dalam bentuk Hypertext Markup Language(html) seperti gambar **Source Code 5.12 Code HTML SARG**. DaribentukHypertext Markup Language(html) bisa dimanfaatkan dalam perhitungan algoritma naive bayes.

01	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
02	"http://www.w3.org/TR/html4/strict.dtd">
03	<html>
04	<head>
05	<meta http-equiv="Content-Type" content="text/html; charset=Latin1">
06	<title>Top sites</title>
07	<style type="text/css">
08	.logo {font-family:Verdana,Tahoma,Arial;font-size:11px;color:#006699;text-


```
09 align:center;vertical-align:middle;border:none;padding:0px;margin-bottom:5px;}
10 .logo th {padding:0px;}
11 .logo img {vertical-align:middle;padding:0px;border:0px none;}
12 .body {font-family:Tahoma,Verdana,Arial;font-size:11px;color:#000000;background-
13 color:white;background-image:url();}
14 .info {font-family:Tahoma,Verdana,Arial;font-size:10px;text-align:center;margin-
15 top:1em;margin-bottom:1em;}
16 .info a:link,a:visited {font-family:Tahoma,Verdana,Arial;color:#0000FF;font-
17 size:10px;text-decoration:none;}
18 .title {width:100%;text-align:center;margin-bottom:1em;}
19 div.title >table {margin:auto;}
20 .title_c {font-family:Tahoma,Verdana,Arial;font-size:11px;color:green;background-
21 color:white;text-align:center;}
22 .title_l {font-family:Tahoma,Verdana,Arial;font-size:11px;color:green;background-
23 color:white;text-align:left;}
24 .title_r {font-family:Tahoma,Verdana,Arial;font-size:11px;color:green;background-
25 color:white;text-align:right;}
26 .index {width:100%;text-align:center;}
27 div.index >table {margin:auto;}
28 .report {width:100%;text-align:center;}
29 div.report >table {margin:auto;}
30 .header_l {font-family:Tahoma,Verdana,Arial;font-
31 size:9px;color:darkblue;background-color:blanchedalmond;text-align:left;border-
32 right:1px solid #666666;border-bottom:1px solid #666666;}
33 .header_r {font-family:Tahoma,Verdana,Arial;font-
34 size:9px;color:darkblue;background-color:blanchedalmond;text-align:right;border-
35 right:1px solid #666666;border-bottom:1px solid #666666;}
36 .header_c {font-family:Tahoma,Verdana,Arial;font-
37 size:9px;color:darkblue;background-color:blanchedalmond;text-align:center;border-
38 right:1px solid #666666;border-bottom:1px solid #666666;}
39 .data {font-family:Tahoma,Verdana,Arial;color:#000000;font-size:9px;background-
40 color:lavender;text-align:right;border-right:1px solid #6A5ACD;border-bottom:1px
41 solid #6A5ACD;}
42 .data a:link,a:visited {font-family:Tahoma,Verdana,Arial;color:#0000FF;font-
43 size:9px;background-color:lavender;text-align:right;text-decoration:none;}
44 .data2 {font-family:Tahoma,Verdana,Arial;color:#000000;font-size:9px;background-
45 color:lavender;text-align:left;border-right:1px solid #6A5ACD;border-bottom:1px
46 solid #6A5ACD;}
47 .data2 a:link,a:visited {font-family:Tahoma,Verdana,Arial;color:#0000FF;font-
48 size:9px;text-align:left;background-color:lavender;text-decoration:none;}
49 .data3 {font-family:Tahoma,Verdana,Arial;color:#000000;font-size:9px;background-
50 color:lavender;text-align:center;border-right:1px solid #6A5ACD;border-bottom:1px
51 solid #6A5ACD;}
52 .data3 a:link,a:visited {font-family:Tahoma,Verdana,Arial;color:#0000FF;font-
53 size:9px;text-align:center;background-color:lavender;text-decoration:none;}
54 .text {font-family:Tahoma,Verdana,Arial;color:#000000;font-size:9px;background-
55 color:lavender;text-align:right;}
```

```

56 .more {font-style: italic;color:#777;}
57 .link {font-family:Tahoma,Verdana,Arial;font-size:9px;color:#0000FF;}
58 .link a:link,a:visited {font-family:Tahoma,Verdana,Arial;font-
59 size:9px;color:#0000FF;text-decoration:none;}
60 a >img {border:none;}
61 .warn {margin:0.5em;}
62 .warn >span {padding:0.5em;border:2px solid black;background-color:orange;font-
63 family:Tahoma,Verdana,Arial;font-size:9px;}
64 </style>
65 </head>
66 <body>
67 <div class="logo"><a href="http://sarg.sourceforge.net"></a>&nbsp;Squid Analysis Report Generator</div>
70 <div class="title"><table cellpadding="0" cellspacing="0">
71 <tr><th class="title_c">Squid User Access Reports</th></tr>
72 <tr><td class="header_c">Period: 12 Sep 2016&mdash;13 Sep 2016</td></tr>
73 <tr><th class="header_c">Top 100 sites</th></tr>
74 </table></div>
75 <div class="report"><table cellpadding="1" cellspacing="2">
76 <thead><tr><th class="header_l">NUM</th><th class="header_l">ACCESSED
77 SITE</th><th class="header_l">CONNECT</th><th class="header_l">BYTES</th><th
78 class="header_l">TIME</th><th class="header_l">USERS</th></tr></thead>
79 <tr><td class="data">1</td><td class="data2"><a
80 href="http://www.ub.ac.id">ub.ac.id</a></td><td class="data">2.10K</td><td
81 class="data">160.78M</td><td class="data">0:04:22</td><td
82 class="data">1</td></tr>
83 <tr><td class="data">2</td><td class="data2"><a
84 href="https://klikmbc.co.id">klikmbc.co.id</a></td><td class="data">2.10K</td><td
85 class="data">160.78M</td><td class="data">0:04:22</td><td
86 class="data">1</td></tr>
87
88 </table></div>
89 <div class="info">Generated by <a href='http://sarg.sourceforge.net'>sarg-2.3.10 Apr-
90 12-2015</a> on 13/Sep/2016-08:01</div>
</body>
</html>

```

Source Code 5.12 Code HTML SARG

Sumber [Implementasi]

5.3 Implementasi Aplikasi Naive Bayes

Penjelasan dalam aplikasi Naive Bayes terdapat beberapa alur yang harus dilakukan untuk untuk mendapatkan hasil dari pengelompokan dengan algoritma Naive Bayes yaitu dengan implementasi :

5.3.1 Implementasi Pengumpulan Data Latih

Pada implementasi pengumpulan data latih diharuskan memiliki aplikasi pengumpulan data latih secara otomatis dalam bentuk **Source Code 5.13 Code Pengambilan text Bahasa** dimana diambil dari line 5 pada alamat Uniform Resource Locator (URL) kbbsi.com lalu diambil diambil content Hypertext Markup Language(html) pada line 6 . dimana *pregmatch* dapat mengambil kalimat sesuai dari awalan simbol sampai dengan akhiran simbol yang ditentukan oleh sistem . jika text belum ada didalam database maka akan tersimpan langsung di dalam database di model codeingiter.

```
01 function update_kamus_indonesia(){
02     $kamus=0;
03     for ($i=0; $i <107; $i++)
04     {
05         $random_code = "http://kbbsi.co.id/daftar-kata?page=$i";
06         $baca = file_get_contents($random_code);
07
08         if(preg_match_all('/arti-kata[^\"](.*)"/', $baca, $matches))
09         {
10             $matches = $matches[1];
11             $this->blog_model->check_kamus_indonesia($matches, $random_code,$kamus);
12         }
13     else
14     {
15         echo 'Pencarian Tidak Ditemukan';
16     }
17 }
18 }
19
20 function update_kamus_inggris()
21 {
22     $kamus=1;
23     for ($i = 1; $i <4; $i++) {
24         $random_code =
25         "http://www.oxfordlearnersdictionaries.com/wordlist/english/oxford3000/Oxford30
26         00_U-Z/?page=$i";
27         $baca = file_get_contents($random_code);
28         if(preg_match_all('/english[^\"](.*)"/', $baca, $matches))
29         {
30             $matches = $matches[1];
31
32             foreach($matches as $var1)
33             {
34                 $mentah=str_replace([' ','!','@','#','$','%','&', '(' , ')', '_','=' , '+', ';', ':', '\\', '/', '*','-',',',
35                 '|', '.', ':', '*', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0'], " ", $var1);
36                 $list = array($mentah);
37                 $this->blog_model->check_kamus_inggris($var1,$list, $random_code,$kamus);
38             }
39         }
40     }
41 }
```

```

39     }
40   }
41   else
42   {
43     echo 'Pencarian Tidak Ditemukan';
44   }
45 }
46 }
47
48

```

Source Code 5.13 Code Pengambilan Data latihan
Sumber [Implementasi]

Pada implementasi **Source Code 5.13 Code Pengambilan text Bahasa** ketika data sudah melalui pencarian data yang tersedia pada database maka tampilan ketika proses selesai akan memberikan informasi status per kalimat tersimpan atau sudah tersedia di dalam database seperti pada gambar **Gambar 5.15** .

```

-an Sudah ada di dalam database
-anda Sudah ada di dalam database
-asi Sudah ada di dalam database
-da Sudah ada di dalam database
-el- Sudah ada di dalam database
-em- Sudah ada di dalam database
-er- Sudah ada di dalam database
-i Sudah ada di dalam database
-wati Sudah ada di dalam database
-wi Sudah ada di dalam database
-wiah Sudah ada di dalam database
a Sudah ada di dalam database
a- Sudah ada di dalam database
ab Sudah ada di dalam database
ab- Sudah ada di dalam database
aba Sudah ada di dalam database
aba-aba Sudah ada di dalam database
abad Sudah ada di dalam database
abadi Sudah ada di dalam database
abadiah Sudah ada di dalam database
abadiah Sudah ada di dalam database
abadiar Sudah ada di dalam database
abah Sudah ada di dalam database
abah-abah Sudah ada di dalam database
abai Sudah ada di dalam database
abaimana Sudah ada di dalam database
abaka Sudah ada di dalam database
ahaltinal Sudah ada di dalam database

```

Gambar 5.12 Tampilan Simpan data latihan
Sumber [Implementasi]

5.3.2 Implementasi Regex Uniform Resource Locator

Pada source code 5.15 adalah proses dimana akan terjadinya pengambilan link url yang sudah di dibatasi pengambilannya dari huruf tertentu pada source code baris ke-04.

```

01 function index()
02 { $code = ($this->random_code);
03 $content = file_get_contents($this->random_code);
04 if (preg_match_all("<a.*(?=href=\"([^\"]*)\")>([^\<]*)</a>|i", $content,
05 $matches)) {
06 $data['matches'] = $matches;
07

```

08	<code>\$matches = \$matches[1];</code>
09	<code>\$this->blog_model->check_url(\$code, \$matches);</code>
10	<code>\$data['random_code1'] = file_get_contents(\$this->random_code);</code>
11	<code>\$data['random_code2'] = (\$this->random_code);</code>
12	<code>\$data['title'] = 'Skripsi - Siwi Rahmat Januar';</code>
13	<code>\$data['listmenu'] = \$this->stator->getListMenu();</code>
14	
15	<code>\$data['sub_page'] = 'skripsi/index';</code>
16	
17	<code>\$this->load->view('header', \$data);</code>
18	<code>\$this->load->view('skripsi', \$data);</code>
19	<code>\$this->load->view('footer');</code>
20	<code>}</code>
	<code>}</code>

**Source Code 5.15 Code Pengambilan Alamat URL
Sumber [Implementasi]**

1. Setiap log proxy Squid akan di ambil halaman html nya untuk dapat di baca seluruh isi codennya :
Html Source:

- ```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"> <html>
<head> <meta http-equiv="Content-Type" content="text/html; charset=Latin1"> <title>Top sites</title> <style
type="text/css"> .logo {font-family:Verdana,Tahoma,Arial;font-size:11px;color:#006699;text-
align:center;vertical-align:middle;border:none;padding:0px;margin-bottom:5px;} .logo th {padding:0px;} .logo
img {vertical-align:middle;padding:0px;border:0px none;} .body {font-family:Tahoma,Verdana,Arial;font-
size:11px;color:#000000;background-color:white;background-image:url();} .info {font-
family:Tahoma,Verdana,Arial;font-size:10px;text-align:center;margin-top:1em;margin-bottom:1em;} .info
a:link,a:visited {font-family:Tahoma,Verdana,Arial;color:#0000FF;font-size:10px;text-decoration:none;} .title
{width:100%;text-align:center;margin-bottom:1em;} div.title > table {margin:auto;} .title_c {font-
family:Tahoma,Verdana,Arial;font-size:11px;color:green;background-color:white;text-align:center;} .title_l {font-
family:Tahoma,Verdana,Arial;font-size:11px;color:green;background-color:white;text-align:left;} .title_r {font-
family:Tahoma,Verdana,Arial;font-size:11px;color:green;background-color:white;text-align:right;} .index
{width:100%;text-align:center;} div.index > table {margin:auto;} .report {width:100%;text-align:center;}
div.report > table {margin:auto;} .header_l {font-family:Tahoma,Verdana,Arial;font-
size:9px;color:darkblue;background-color:blanchedalmond;text-align:left;border-right:1px solid
#666666;border-bottom:1px solid #666666;} .header_r {font-family:Tahoma,Verdana,Arial;font-
size:9px;color:darkblue;background-color:blanchedalmond;text-align:right;border-right:1px solid
#666666;border-bottom:1px solid #666666;} .header_c {font-family:Tahoma,Verdana,Arial;font-
size:9px;color:darkblue;background-color:blanchedalmond;text-align:center;border-right:1px solid
#666666;border-bottom:1px solid #666666;} .data {font-family:Tahoma,Verdana,Arial;color:#000000;font-
size:9px;background-color:lavender;text-align:right;border-right:1px solid #6A5ACD;border-bottom:1px solid
#6A5ACD;} .data a:link,a:visited {font-family:Tahoma,Verdana,Arial;color:#0000FF;font-size:9px;background-
color:lavender;text-align:right;text-decoration:none;} .data2 {font-
family:Tahoma,Verdana,Arial;color:#000000;font-size:9px;background-color:lavender;text-align:left;border-
right:1px solid #6A5ACD;border-bottom:1px solid #6A5ACD;} .data2 a:link,a:visited {font-
family:Tahoma,Verdana,Arial;color:#0000FF;font-size:9px;text-align:left;background-color:lavender;text-
decoration:none;} .data3 {font-family:Tahoma,Verdana,Arial;color:#000000;font-size:9px;background-
color:lavender;text-align:center;border-right:1px solid #6A5ACD;border-bottom:1px solid #6A5ACD;} .data3
a:link,a:visited {font-family:Tahoma,Verdana,Arial;color:#0000FF;font-size:9px;text-align:center;background-

```

**Gambar 5.13 File Mentah Sebelum di proses  
Sumber [Implementasi]**

- <http://www.ub.ac.id>
- <https://klikmbc.co.id>

**Gambar 5.14 File Mentah sesudah di proses**  
Sumber [Implementasi]

### 5.3.3 Implementasi Check Metadata

saat sebelum perhitungan naive bayes akan dilakukan , harus sudah memiliki contoh alamat URL yang akan di uji coba. Pada baris ke-4 seperti pada **Source Code 5.14** terdapat fungsi untuk mengambil data dari database sistem jika dilihat pada baris ke-5 terdapat proses dimana penampilan seluruh data array.

```

01 function load_url()
02 {
03
04 $query = $this->db->get($this->table);
05 foreach ($query->result() as $row) {
06
07 $link = $row->link;
08
09 $tags = @get_meta_tags($link);
10
11 if (@array_key_exists("description", $tags)) {
12 $metadata = $tags["description"];
13
14 $a = array(
15 'metadata' =>$metadata,
16 'status' =>1
17);
18 $this->db->where('link', $link);
19 $this->db->update($this->table, $a);
20 echo "data $link berhasil disimpan" . "
";
21 }
22 }
23 }

```

**Source Code 5.14 Code pencarian meta data tag**  
Sumber [Implementasi]

Jika semua link ditampilkan maka secara otomatis akan terjadi pengecekan Meta data pada baris ke-9 untuk dapat mengetahui apakah informasi meta data tersedia maka akan dilakukan proses filtering pada baris ke-11 dimana jika metadata tersedia maka isi dari informasi meta data akan disimpan kedalam database sistem seperti pada **Gambar 5.16** dibawah ini.



**Gambar 5.15 Proses Pencarian Metadata**  
Sumber [Implementasi]

### 5.3.4 Implementasi Pencarian Data Latih

Saat Uniform Resource Locator (URL) sudah memiliki informasi data metadata deskripsi maka sistem akan membagi bagian kalimat yang masih menyatu untuk di bagi menjadi perkalimat seperti pada **Source Code 5.16**. Dalam baris ke-1 sistem akan mengilangkan kalimat simbol yang berfungsi untuk mengganti kalimat yang tidak benilai baik ke nilai akhir. Pada baris ke-3 dilakukan explode atau pemecahan setiap kalimat yang mengandung simbol Spasi " " dan diurutkan oleh baris ke-4.

|    |                                                                                                                 |
|----|-----------------------------------------------------------------------------------------------------------------|
| 01 | <code>\$mentah = str_replace([' ', '!', '@', '#', '\$', '%', '&amp;', '(', ')', '_', '=', '+', ',', ':',</code> |
| 02 | <code>'\\', '/', '*', '-', ';', ' ', ':', '*', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0'], " ",</code>   |
| 03 | <code>\$mac_note);</code>                                                                                       |
| 04 | <code>\$arr_kalimat = explode(" ", \$mentah);</code>                                                            |
| 05 | <code>sort(\$arr_kalimat);</code>                                                                               |
| 06 | <code>\$hasil = array_unique(\$arr_kalimat);</code><br><code>sort(\$hasil);</code>                              |

**Source Code 5.16 Code Pengambilan Alamat URL**  
Sumber [Implementasi]

Pada baris ke-5 dilakukan pengubahan data menjadi kalimat bermakna atau unik ke dalam array dan nantinya akan menghasilkan output seperti pada **Gambar 5.19**

| Per Kalimat {} | Kalimat Sama dalam Database {} |
|----------------|--------------------------------|
| Agan           | Paket                          |
| Bisnis         | Tiket                          |
| GRATIS         |                                |
| Haji           |                                |
| Hotel          |                                |
| Isi            |                                |
| Kereta         |                                |
| Mobil          |                                |
| Modal          |                                |
| Online         |                                |
| PPOB           |                                |
| Paket          |                                |
| Pesawat        |                                |
| Sewa           |                                |
| Tanpa          |                                |
| Tiket          |                                |
| Tour           |                                |
| Umroh          |                                |
| pulsa          |                                |
| ulang          |                                |

**Gambar 5.16 Explode Text metadata tag**  
Sumber [Implementasi]

### 5.3.5 Implementasi Tabel Testing

Pada pengisian Tabel testing dilakukan penampilan data dari informasi data latihan yang sudah tersimpan di dalam database sistem untuk barus ke-3 sampai dengan ke-19 dilakukan pemanggilan sql untuk dapat menghasilkan data yang dibutuhkan .

```
01 <?php
02 date_default_timezone_set('Asia/Jakarta');
03 $this->db-
04 >select('p.id_data_fix,p.nama_data,c.id_kategori,d.nama_kategori,e.nama_jenis');
05 $this->db->where('p.code', '$code');
06 $this->db->where('p.status', 2);
07 $this->db->join('data_latih c', 'c.nama_data = p.nama_data');
08 $this->db->join('kategori_pengguna d', 'd.id_kategori = c.id_kategori');
09 $this->db->join('jenis_kategori e', 'e.id_jenis = c.id_jenis');
10
11 $query1 = $this->db->get('data_latih_fix p');
12
13 $query = $this->db->query("SELECT
14 p.id_data_fix,p.nama_data,a.id_kategori,d.nama_kategori,e.nama_jenis FROM
15 data_latih_fix p
16 INNER JOIN data_latih a ON a.nama_data = p.nama_data
17 INNER JOIN kategori_pengguna d ON d.id_kategori = a.id_kategori
18 INNER JOIN jenis_kategori e ON e.id_jenis = a.id_jenis
19 WHERE p.code LIKE '$code' AND p.status = 1
20 ");
21
22 $no=1;
23 $data_testing = $query->result();
24 foreach ($data_testing as $row):
25
26 $mac_address = $row->code;
27 $mac_bgcolor = $row->link;
28 $mac_note = $row->nama_data;
29 $kelas_kategori = $row->nama_kategori;
30 $kelompok_kategori = $row->nama_jenis;
31 ?>
32 <tr>
33 <td class="text-nowrap"><?php echo $no; ?></td>
34 <td class="text-nowrap"><?php echo $mac_note; ?></td>
35 <td class="text-nowrap"><?php echo $kelas_kategori; ?></td>
36 <td class="text-nowrap"><?php echo $kelompok_kategori; ?></td>
37 </tr>
38 <?php
39
40 $no++;
41 endforeach;
```



### Source Code 5.17 Source code Membuat Tabel Data Testing

Sumber [Implementasi]

Saat tabel testing sudah terisi maka akan terbagi menjadi data latih dan data testing dengan latih atau data yang sudah tersedia untuk dilakukan uji coba dengan data latih X1 seperti gambar **Gambar 5.20** dan siap untuk dihitung perhitungan dengan algoritma Naive Bayes Classifier (NBC).

| TABEL TESTING |                                                                                                                                 |
|---------------|---------------------------------------------------------------------------------------------------------------------------------|
| No            | Deskripsi Metatag data Dan Data Latih                                                                                           |
| 1             | Paket                                                                                                                           |
| 2             | Tiket                                                                                                                           |
| X1            | Bisnis Agen Tiket Pesawat GRATIS Tanpa Modal - Tiket Pesawat, Tiket Kereta, Hotel, Umroh/Haji, Paket Tour, Sewa Mobil, PPOB Onl |

test jumlah url = 1

**Gambar 5.17 Tabel Testing Web**

Sumber [Implementasi]

### 5.3.6 Implementasi Term Frequency (Tfi)

Saat sudah terbentuknya tabel testing maka hal yang dilakukan selanjutnya adalah melakukan perhitungan Term Frequency dimana setiap kalimat kan dicari banyak jumlah kalimat yang cocok pada baris ke-6 dimana berfungsi untuk menghitung jumlah kalimat secara horizontal

```
01 foreach ($hasil as $var4) {
02 $termkecil = strtoupper($var4);
03 echo "<tr>" . "<td class=\"text-nowrap\">" . $i2 . "</td>" . "<td
04 class=\"text-nowrap\">" . $termkecil . "</td>";
05
06 for ($no1 = 1; $no1 <= $totalforeach; $no1++) {
07 $ra1 = strtoupper($termkecil);
08 $str = strtoupper($term[$no1]);
09
10 str_replace("$ra1", "", "$str", $count);
11
12 print_r("<td class=\"text-nowrap\">" . $count . "</td>");
13
14 $hitung[] = $count;
15 }
16 echo "</tr>";
17 $i2++;
18 }
19
```

**Source Code 5.18 Source Code Term Frequency (Tfi)**

Sumber [Implementasi]

Pada baris ke 10 berguna dalam menghitung jumlah kalimat yang sama sesuai nilai dari data latih dan data uji secara vertikal dimana akan terjadi perhitungan yang sesuai dengan kalimat yang sama seperti **Gambar 5.21**

| No | Term Text | 1 | 2 | X1 |
|----|-----------|---|---|----|
| 0  |           | 0 | 0 | 0  |
| 1  | AGEN      | 0 | 0 | 1  |
| 2  | BISNIS    | 0 | 0 | 1  |
| 3  | GRATIS    | 0 | 0 | 1  |
| 4  | HAJI      | 0 | 0 | 1  |
| 5  | HOTEL     | 0 | 0 | 1  |
| 6  | ISI       | 0 | 0 | 1  |
| 7  | KERETA    | 0 | 0 | 1  |
| 8  | MOBIL     | 0 | 0 | 1  |
| 9  | MODAL     | 0 | 0 | 1  |
| 10 | ONLINE    | 0 | 0 | 2  |
| 11 | PPOB      | 0 | 0 | 1  |
| 12 | PAKET     | 1 | 0 | 1  |
| 13 | PESAWAT   | 0 | 0 | 2  |
| 14 | SEWA      | 0 | 0 | 1  |
| 15 | TANPA     | 0 | 0 | 1  |
| 16 | TIKET     | 0 | 1 | 3  |
| 17 | TOUR      | 0 | 0 | 1  |
| 18 | UMROH     | 0 | 0 | 1  |
| 19 | PULSA     | 0 | 0 | 1  |
| 20 | ULANG     | 0 | 0 | 1  |

**Gambar 5.18 Tabel Term Frequency (Tfi) Web**  
Sumber [Implementasi]

### 5.3.7 Implementasi DF dan IDF

Implementasi DF dan IDF dapat dilakukan dikala proses **Gambar 5.21** sudah terbentuk dimana perhitngan DF ini bermaksud untuk mengetahui jumlah dari nilai yang ada pada setiap kalimat yang di cari secara horizontal pada baris ke-1 terjadi parameter pilihan dimana jika nilai lebih dari 0 maka akan dikategorikan ke nilai Term Frequency senilai 1 dan jika nilai sama dengan 0 atau di bawahnya nilai dari setiap term frekuensi adalah 0 dan akan ditampilkan pada baris ke-10.

```

01 if ($normalized >0) {
02 $hitung1 = 1;
03 $sum += $hitung1;
04 } else {
05 $hitung1 = 0;
06 $sum += $hitung1;
07 }
08
09 }
10 print_r("<td>" . $sum . "</td>");

```

```

11 if(isset($sum) &&$sum != 0){
12 $df=(log10($maxif/$sum));
13 print_r("<td>" . $df . "</td>");
14 }else{
15 print_r("<td>" . 0 . "</td>");
16 }

```

**Source Code 5.19 Source Code DF dan IDF**  
Sumber [Implementasi]

Sesudah perhitungan DF terhitung selanjutnya dihitung dengan nilai IDF dimana pada baris ke-12 dihitung sesuai dengan rumus IDF dengan menghitung nilai dasar logaritma dari nilai tertinggi dari nilai Term Frequency lalu dibagi dengan nilai DF sehingga memiliki hasil pada **Gambar 5.22**.

| DF | IDF              |
|----|------------------|
| 0  | 0                |
| 1  | 0.47712125471966 |
| 1  | 0.47712125471966 |
| 1  | 0.47712125471966 |
| 1  | 0.47712125471966 |
| 1  | 0.47712125471966 |
| 1  | 0.47712125471966 |
| 1  | 0.47712125471966 |
| 1  | 0.47712125471966 |
| 1  | 0.47712125471966 |
| 1  | 0.47712125471966 |
| 1  | 0.47712125471966 |
| 1  | 0.47712125471966 |
| 1  | 0.47712125471966 |

**Gambar 5.19 Tabel DF dan IDF**  
Sumber [Implementasi]

### 5.3.8 Implementasi Ni

Nilai Ni berfungsi dalam mencari jumlah nilai yang mengandung nilai hasil dari contoh tabel data latih . pada baris ke-1 sampai dengan baris ke-6 dimana proses tersebut tetap mencari nilai dari Term Frequency tetapi perbedaannya disini adanya pengabungan nilai dari kategori postifi atau negatif pada baris ke-11 sampai dengan baris ke-17 dan ditotalkan pada baris ke-47.

```

01 for ($no1= 1; $no1 <= $totaldatalatih; $no1++)
02 {
03 $ra1 = strtoupper($termkecil);
04 $str = strtoupper($data_latih[$no1]);
05
06 str_replace("$ra1", "", "$str", $count);
07 $normalized = $count / $maxif;
08
09 $hasil_count = $count;

```

```

10
11 if ($id_kategori == 1) {
12 $positif1 = 1*$hasil_count;
13 print_r("<th>" . $positif1. "</th>");
14
15 } else {
16 $positif1 = 0*$hasil_count;
17 print_r("<th>" . $positif1 . "</th>");
18
19 }
20 $sumt[$i2]=$positif1;
21 $sum += $positif1;
22 }
23
24 $sum1 = 0;
25 for ($no1= 1; $no1 <= $totaldatalatih; $no1++) {
26 $ra1 = strtoupper($termkecil);
27 $str = strtoupper($data_latih[$no1]);
28
29 str_replace("$ra1", "", "$str", $count);
30 $normalized = $count / $maxif;
31 $hasil_count1 = $count;
32
33 if ($id_kategori == 2) {
34 $negatif1 = 1 * $hasil_count1;
35 print_r("<th>" . $negatif1 . "</th>");
36
37 } else {
38 $negatif1 = 0 * $hasil_count1;
39 print_r("<th>" . $negatif1 . "</th>");
40
41 }
42
43 $sum1 += $negatif1;
44 }
45
46 print_r("<th>" . $sum. "</th>");
47 print_r("<th>" . $sum1. "</th>");
48
49 $total_ni += $sum;
50 $total_ni1 += $sum1;
51

```

Source Code 5.20 Source Code NI

Sumber [Implementasi]

Saat perhitungan jumlah nilai ni positif dan ni negatif yang diambil dari Term Frequency maka akan dihasilkan nilai berbeda seperti **Gambar 5.23** .

| Ni1 | Ni2 |
|-----|-----|
| 0   | 0   |
| 0   | 0   |
| 0   | 0   |
| 0   | 0   |
| 0   | 0   |
| 0   | 0   |

**Gambar 5.20 Tabel NI**  
Sumber [Implementasi]

### 5.3.9 Implementasi Likelihood

Dalam proses ini sudah mulai ditentukan tingkat likelihoodnya , dengan arti dimana sudah mulai terlihat nilai dari hasil kategori yang secara tidak pasti karena pada proses likelihood dimana pencarian masih dilihat dari perbedaan nilai tertinggi antara kategori yang ada pada baris ke-1 dan baris ke-2 dilihat perbedaan perhitungan likelihood postifi dan likelihood negatif .

```

01 $hasillikelihood = ($sum + 1) / ($total_ni + 0);
02 $hasillikelihood1 = ($sum1 + 1) / ($total_ni1 + 0);
03
04 if ($hasillikelihood == 0) {
05 print_r("<th>" . 0 . "</th>");
06 } else {
07 print_r("<th>" . $hasillikelihood . "</th>");
08 }
09 if ($hasillikelihood1 == 0) {
10 print_r("<th>" . 0 . "</th>");
11 } else {
12 print_r("<th>" . $hasillikelihood1 . "</th>");
13 }

```

**Source Code 5.21 Source Code Likelihood**  
Sumber [Implementasi]

pada hasil akhir dar dari likelihood dapat dilihat dar perbedaan nilai pada **Gambar 5.24** likelihood pada gambar di bawah ini masih menggunakan nilai dari data testing . Untuk data ujinnya akan dihitung kelayakannya pada Implementasi Posterior.

|  | Total             | LikeHood |
|--|-------------------|----------|
|  | x1                | x2       |
|  | 13.44444444444444 | 0        |

**Gambar 5.21 Tabel Likelihood**  
Sumber [Implementasi]

### 5.3.10 Implementasi Product

Sesudah perhitungan Likelihood maka hal yang dihitung adalah nilai product. Dimana pada baris ke -19 sampai dengan baris ke-22 adalah penampilan dari nilai produk dengan penjumlahan dari keseluruhan nilai likelihood pada baris ke-01 sampai dengan baris ke-18 dimana terjadinya penggabungan kedua buah array.

```

01 array_push($to_product, $myArray);
02 array_push($to_product1, $myArray1);
03 }
04 $total_product = $to_product[0];
05 for($i = 1; $i < count($to_product); $i++){
06 for($j = 1; $j <= count($total_product); $j++){
07 $total_product[$j] = $total_product[$j] * $to_product[$i][$j];
08 }
09
10 }
11 $total_product1 = $to_product1[0];
12 for($i = 1; $i < count($to_product1); $i++){
13 for($j = 1; $j <= count($total_product1); $j++){
14 $total_product1[$j] = $total_product1[$j] * $to_product1[$i][$j];
15 }
16 }
17 for($j = 1; $j <= count($total_product1); $j++){
18 print_r("<th>" . $total_product[$j] . "</th>");
19 print_r("<th>" . $total_product1[$j] . "</th>");
20 }
21

```

**Source Code 5.22 Source Product**  
Sumber [Implementasi]

### 5.3.11 Implementasi Posterior

Di dalam perhitungan akhir dari pengkategorian kelas ditentukan dalam nilai posterior dimana pada baris ke-03 adalah rumus dari mencari nilai posterir dimana nilai prior dikali

dengan product. Sehingga pada **Gambar 5.26** sudah memiliki perbedaan yang berbeda dan di tentukan jika nilai Positif lebih besar dari negatif maka kelas adalah positif.

```

01 echo "<th></th><th></th><th>Hasil Posterior</th>";
02 for($j = 1; $j <= count($total_product1); $j++){
03 $hasilposterior=$prior1*$total_product[$j];
04 print_r("<th>" . $hasilposterior . "</th>");
05
06 $hasilposterior1=$prior2*$total_product1[$j];
07 print_r("<th>" . $hasilposterior1 . "</th>");
08 }

```

**Source Code 5.23 Source Posterior**

Sumber [Implementasi]

|           |                   |   |                   |   |
|-----------|-------------------|---|-------------------|---|
| Hasil     | 0.098765432098765 | 0 | 0.444444444444444 | 0 |
| Posterior |                   |   |                   |   |

**Gambar 5.22 Tabel Posterior**

Sumber [Implementasi]

### 5.3.12 Implementasi Hasil Klasifikasi

Pada hasil klasifikasi dapat dilakukan ketika nilai dari posterior sudah ada dan dari perbandingan hasil posterior sudah bisa ditentukan penyimpanan data untuk dapat di blokir oleh router pada baris ke-6 sampai ke-12 dimana ada kedua pilihan ketika dalam kelas positif aksi yang dilakukan oleh sistem adalah menyimpan data dan jika hasil negatif maka hasil juga tetap menyimpan data tetapi untuk penampilan dalam bentuk user interface adalah pada baris ke -22.

```

01 echo "<th></th><th></th><th>Hasil Klasifikasi</th>";
02 for($j = $total_hasil; $j <= count($total_product1); $j++) {
03 $hasilposterior = $total_product[$j] * $prior1;
04 $hasilposterior1 = $total_product1[$j] * $prior2;
05
06 if ($hasilposterior >= $hasilposterior1) {
07
08 $termkeci1l = $mac_link[$j];
09 $hasil_link = 1;
10
11 print_r("<th>" . $termkeci1l . "=" . "Pengguna Positif" . "</th><th></th>");
12
13
14 } else {
15 $termkeci1l = $mac_link[$j];
16 $hasil_link = 2;
17 print_r("<th>" . $termkeci1l . "Pengguna Negatif" . "</th></th><th></th>");

```

```

18 }
19 }
20 echo $mac_address . " " . $mac_link[$j] . " " . $hasil_link;
21 $sql1 = $this->db->query("SELECT link FROM hasil WHERE `link` LIKE
22 '$mac_link[$j]'");
23 if ($sql1->num_rows() >0) {
24
25
26 } else {
27 $output = $hasil . $j;
28 $output = array(
29 'code' =>$mac_address,
30 'link' =>$mac_link[$j],
31 'hasil' =>$hasil_link,
32 'created' =>date('Y-m-d H:i:s')
33);
34
35 $this->db->insert('hasil', $output);
36 }

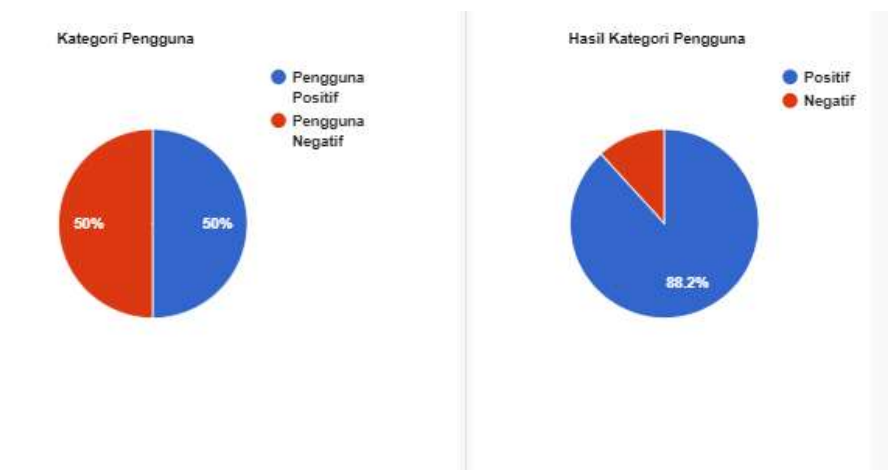
```

**Source Code 5.24 Source Code NI**  
Sumber [Implementasi]

|                   |                                            |   |                                          |   |
|-------------------|--------------------------------------------|---|------------------------------------------|---|
| Hasil Posterior   | 1.6204883201983E-53                        | 0 | 4.7391750551522E-38                      | 0 |
| Hasil Klasifikasi | http://stackoverflow.com/=Pengguna Positif |   | http://ocsp.godaddy.com=Pengguna Positif |   |

**Gambar 5.23 Tabel Likelihood**  
Sumber [Implementasi]

Pada gambar **Gambar 5.28** tabel pie chart sudah dapat dilihat jika proses perhitungan analisis *Naive Bayes Classifier* (NBC) sudah dilakukan dan terkumpul di dalam database sistem . Dari sini sudah bisa mengukur tingkat besarnya angka karakteristik pengguna internet.



**Gambar 5.24 Tabel Grafik**



### 5.3.13 Implementasi Block Url

Sebelum sistem dapat terhubung dengan router diharuskan terhubung terlebih dahulu untuk dapat meminta permintaan informasi yang di minta oleh program kepada system router. Pada baris ke 2 adalah proses dimana permintaan dalam bahasa *command line interface* (CLI). Dimana pada baris ke-4 adalah alamat socet yang dapat terhubung dengan aplikasi sehingga jika pada baris ke-6 socet tidak tersedia maka akan terjadi timeout/ error. Baris ke 15 sampai dengan baris ke-17 adalah proses pengiriman permintaan Login dan password kepada system.

|    |                                                                                          |
|----|------------------------------------------------------------------------------------------|
| 01 | <code>\$this-&gt;connected = false;</code>                                               |
| 02 | <code>\$this-&gt;debug('Connection attempt #' . \$ATTEMPT . ' to ' . \$ip . ':' .</code> |
| 03 | <code>\$portx . '...');</code>                                                           |
| 04 | <code>\$this-&gt;socket = @fsockopen(\$ip, \$portx, \$this-&gt;error_no, \$this-</code>  |
| 05 | <code>&gt;error_str, \$this-&gt;timeout);</code>                                         |
| 06 | <code>if (\$this-&gt;socket) {</code>                                                    |
| 07 | <code>socket_set_timeout(\$this-&gt;socket, \$this-&gt;timeout);</code>                  |
| 08 | <code>\$this-&gt;write('/login');</code>                                                 |
| 09 | <code>\$RESPONSE = \$this-&gt;read(false);</code>                                        |
| 10 | <code>if (\$RESPONSE[0] == '!done') {</code>                                             |
| 11 | <code>\$MATCHES = array();</code>                                                        |
| 12 | <code>if (preg_match_all('/[^\s]+/i', \$RESPONSE[1], \$MATCHES)) {</code>                |
| 13 | <code>if (\$MATCHES[0][0] == 'ret' &amp;&amp;strlen(\$MATCHES[0][1]) == 32) {</code>     |
| 14 | <code>\$this-&gt;write('/login', false);</code>                                          |
| 15 | <code>\$this-&gt;write('=name=' . \$login, false);</code>                                |
| 16 | <code>\$this-&gt;write('=response=00' . md5(chr(0) . \$password . pack('H*',</code>      |
| 17 | <code>\$MATCHES[0][1]));</code>                                                          |
| 18 | <code>\$RESPONSE = \$this-&gt;read(false);</code>                                        |
| 19 | <code>if (\$RESPONSE[0] == '!done') {</code>                                             |
| 20 | <code>\$this-&gt;connected = true;</code>                                                |
| 21 | <code>break;</code>                                                                      |
| 22 | <code>    }</code>                                                                       |
| 23 | <code>  }</code>                                                                         |
| 24 | <code>}</code>                                                                           |

Source Code 5.25Api Connect Mikortik

Sesudah sistem dapat terhubung maka sistem dapat melakukan pemblokiran data yang dianggap tidak pantas oleh system pada perhitungan Naive Bayes . Pada baris ke-3 adalah pengecekan apakah masih memiliki session dalam koneksi ke perangkat api mikrotik. Jika memiliki session maka dapat dilanjutkan kedalam pengisian pemblokiran dengan fungsi baris ke-7 dimana ini berfungsi sebagai text terminal *command line interface* (CLI) .

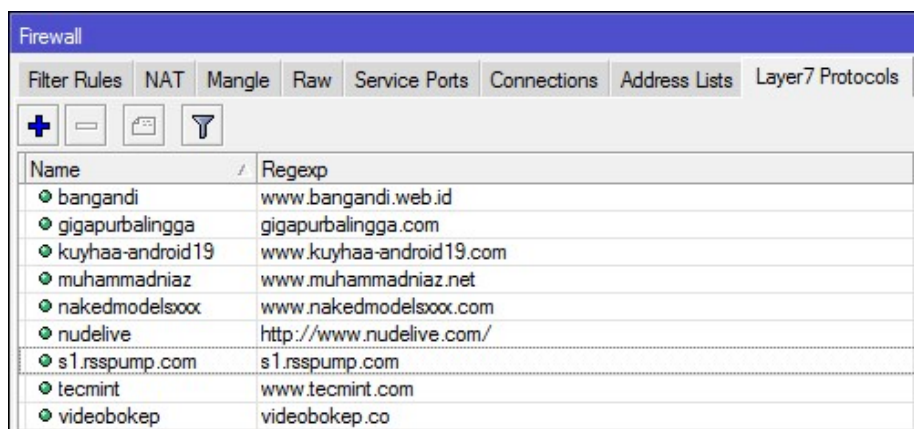
```

01 function addupdateblock($data = array()) {
02 try {
03 if ($this->cekKoneksiRouter()) {
04 $name = $data['name'];
05 $link = $data['regexp'];
06
07 $this->write('/ip/firewall/layer7-protocol/add', false);
08 $this->write('=name=' . $name, false);
09 $this->write('=regexp=' . $link);
10
11 $READ = $this->read(false);
12 return $READ;
13 }
14 } catch (Exception $e) {
return "Error: " . $e;
}
}

```

**Source Code 5.26 Api Command Mikortik**  
Sumber [Implementasi]

Jika pemblokiran sudah dilakukan oleh sistem . maka hal selanjutnya yang dapat dilakukan adalah melihat apakah perintah *command line interface* (CLI) sudah diterima oleh Api mikrotik dengan cara membuka Winbox . dimana winbox ini produk GUI dari perusahaan mikrotik dengan membuka ip->firewall->tab layer 7 protocol seperti **Gambar 5.29** dan **Gambar 5.30**.



**Gambar 5.25 GUI Firewall->Layer 7 Protocols melalui winbox**  
Sumber [Implementasi]

Pada **Gambar 5.30** terlihat bahwa alamat layer 7 sudah tidak di berikan hak izin akses kepada pengguna . untuk melihat bisa membuka ip->firewall->filter rules .fungsi dari pembelokiran layer 7 protocol adalah metode pencarian pola terhadap paket data yang melewati jalur

ICMP,TCP dan UDP. Sehingga permintaan apapun yang berbentuk layer 7 protocol tidak akan diberikan jika pengaturan jaringan sudah di tentukan .

```

01 function addupdateblock($data = array()) {
02 try {
03 if ($this->cekKoneksiRouter()) {
04 $name = $data['name'];
05 $link = $data['regexp'];
06
07 $this->write('/ip/firewall/filter-rules/add', false);
08 $this->write('=name=' . $name, false);
09 $this->write('=regexp=' . $link);
10
11 $READ = $this->read(false);
12 return $READ;
13 }
14 } catch (Exception $e) {
return "Error: " . $e;
}
}

```

Source Code 5.27Api Command Mikortik(1)

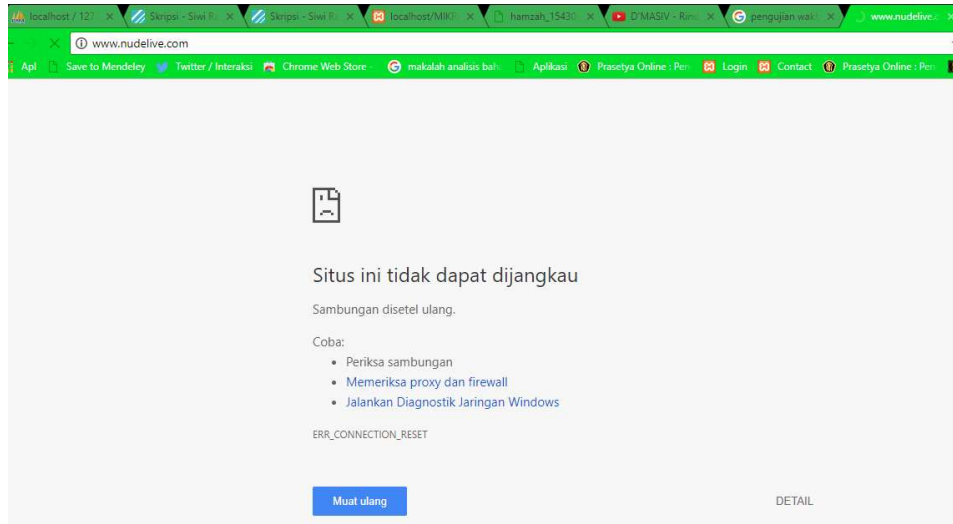
Sumber [Implementasi]

| #                            | Action | Chain  | Proto...     | Src. Port | Dst. Port   | In. Inter... | Out. Int... | Bytes    | Packets |
|------------------------------|--------|--------|--------------|-----------|-------------|--------------|-------------|----------|---------|
| 0                            | D      | jump   | forward      |           |             |              |             | 2004 B   | 26      |
| 1                            | D      | jump   | forward      |           |             |              |             | 0 B      | 0       |
| 2                            | D      | jump   | input        |           |             |              |             | 314.1 KB | 4 092   |
| 3                            | D      | drop   | input        | 6 (tcp)   | 64872-64... |              |             | 0 B      | 0       |
| 4                            | D      | jump   | hs-input     |           |             |              |             | 314.1 KB | 4 092   |
| 5                            | D      | acc... | hs-input     | 17 (u...  | 64872       |              |             | 31.2 KB  | 477     |
| 6                            | D      | acc... | hs-input     | 6 (tcp)   | 64872-64... |              |             | 24.4 KB  | 215     |
| 7                            | D      | jump   | hs-input     |           |             |              |             | 60 B     | 1       |
| 8                            | D      | reject | hs-unauth    | 6 (tcp)   |             |              |             | 924 B    | 12      |
| 9                            | D      | reject | hs-unauth    |           |             |              |             | 1140 B   | 15      |
| 10                           | D      | reject | hs-unauth-to |           |             |              |             | 0 B      | 0       |
| ... place hotspot rules here |        |        |              |           |             |              |             |          |         |
| 11                           | X      | pas... | unused-hs... |           |             |              |             | 0 B      | 0       |
| 12                           | X      | drop   | input        | 6 (tcp)   | 8080        | Jalur wi...  |             | 0 B      | 0       |
| ... Nudelive                 |        |        |              |           |             |              |             |          |         |
| 13                           | X      | drop   | forward      |           |             |              |             | 57.9 KB  | 122     |

Gambar 5.26GUI Firewall->Filter Rule melalui winbox

Sumber [Implementasi]

Pada gambar **Gambar 5.31** dilakukan percobaan dengan membuka alamat yang sudah di blokir oleh firewall mikrotik sehingga ketika pengguna mengakses alamat yang sudah termasuk golongan negatif maka secara otomatis router tidak akan memberi respon untuk alamat pemblokiran tersebut.



**Gambar 5.27** Hasil Blokir Naive Bayes (NBC)  
Sumber [Implementasi]