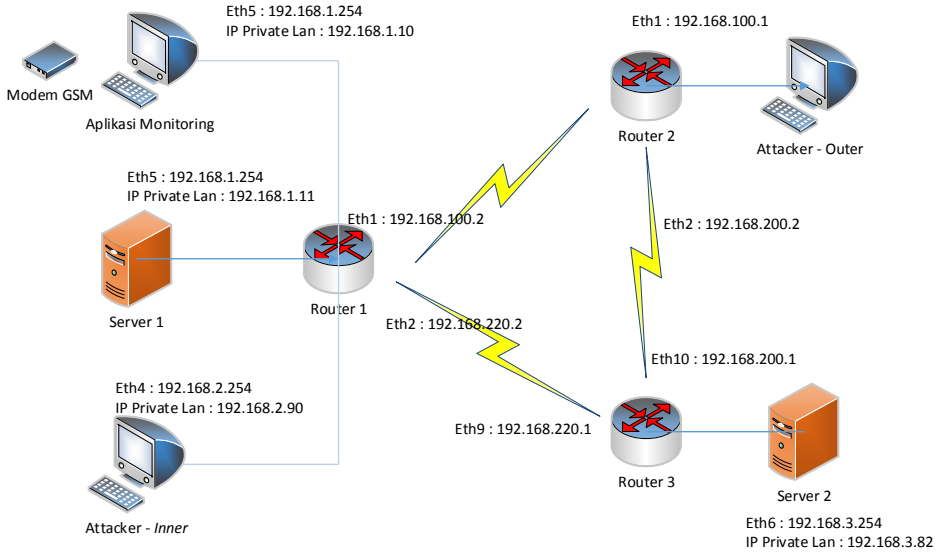
BAB 4
IMPLEMENTASI DAN PENGUJIAN

1.1 Implementasi

Tahap implementasi merupakan penerapan dari tahap analisis dan perancangan. Pada tahap ini perancangan-perancangan sistem pada penelitian yang telah dibuat akan diimplementasikan dalam bentuk konfigurasi dan pengkodean. Dengan demikian dapat diketahui apakah perancangan tersebut dapat berjalan dengan baik atau tidak.

1.1.1 Implementasi Arsitektur Jaringan Pengujian

Implementasi ini merupakan penerapan dari perancangan arsitektur jaringan pada Gambar 3.34, pada perancangan tersebut digambarkan bagaimana setiap komputer dan perangkat lainnya terhubung. Berikut ini adalah Gambar 4.1 arsitektur pengujian yang akan diimplementasikan yang telah diberikan pengalamatan untuk setiap komputernya.



Gambar 4.1 Arsitektur Jaringan Pengujian

1.1.1.1 Implementasi Perangkat Keras

Implementasi ini merupakan penjelasan mengenai spesifikasi perangkat keras yang digunakan pada arsitektur jaringan yang akan diimplementasikan. Pada gambar aritektur pengujian (Gambar 4.1) terdapat beberapa perangkat keras yaitu komputer *Attacker-Outer*, *router*, komputer *monitoring*, komputer *server*, komputer *Attacker-Inner*. Setiap perangkat keras tersebut memiliki spesifikasi yang dijelaskan pada Tabel 4.1 sampai dengan Tabel 4.5.

Tabel 4.1 Spesifikasi Perangkat Keras Komputer *Attacker*

No	Perangkat	Keterangan
1	Processor	Intel Core 2 Duo Processor T5500
2	RAM	2 GB
3	Hardisk	60 GB
4	Monitor	Resolusi 1366 x 768
5	VGA	Intel Graphics Media Accelerator 950
6	Koneksi	Konektor RJ45

Tabel 4.2 Spesifikasi Perangkat Keras Router

No	Spesifikasi	R1 : RB450	R2 : RB750	R3 : RB1100AHx2
1	Architecture	MIPS-BE	MIPS-BE	PPC
2	CPU	AR7161 680MHz	QCA9531-BL3A-R 850MHz	Freescale P2020 1066MHz Dual Core
3	Current Monitor	No	No	No
4	Main Storage/NAND	512MB	16MB	64MB
5	RAM	256MB	64MB	1.5GB
6	SFP Ports	0	0	0
7	LAN Ports	5	5	13
8	Gigabit	Yes	No	Yes
9	Switch Chip	1	1	2
10	MiniPCI	0	0	0
11	Integrated Wireless	No	No	No
12	MiniPCIe	0	0	0
13	SIM Card Slots	No	No	No
14	USB	No	No	No
15	Memory Cards	1	No	1
16	Memory Card Type	MicroSD	-	MicroSD

17	Power Jack	10-28V	10-28V	110/220V
18	802.3af Support	No	No	No
19	POE Input	10-28V	10-28V	10-24VDC
20	POE Output	No	No	No
21	Serial Port	DB9/RS232	No	DB9/RS232
22	Voltage Monitor	No	No	Yes
23	Temperature Sensor	No	No	Yes
24	Dimentions	150mm x 105mm	113x89x28mm	1U case: 45x75x440mm
25	Operating System	RouterOS	RouterOS	RouterOS
26	Temperature Range	-30C .. +60C	-40C .. +55C	-20C .. +45C
27	RouterOS License	Level5	Level4	Level6

Tabel 4.3 Spesifikasi Perangkat Keras Komputer *Monitoring*

No	Perangkat	Keterangan
1	Processor	Intel Core i5 processor 430M
2	RAM	4 GB
3	Hardisk	500 GB
4	Monitor	Resolusi 1366 x 768
5	VGA	NVIDIA GeForce 310M
6	Koneksi	Konektor RJ45

Tabel 4.4 Spesifikasi Perangkat Keras Komputer *Server*

No	Perangkat	Server 1	Server 2
1	Processor	Intel Core i5 processor 430M	Intel Core i5 2.6 GHz
2	RAM	4 GB	8 GB
3	Hardisk	500 GB	256 GB
4	Monitor	Resolusi 1366 x 768	Resolusi 1366 x 768
5	VGA	NVIDIA GeForce 310M	Intel Iris 1536 MB
6	Koneksi	Konektor RJ45	Konektor RJ45

Tabel 4.5 Spesifikasi Perangkat Keras Modem

No	Perangkat	Keterangan
1	Seri	Wavecom Fastrack M1206 (Serial/rs232) Aluminum Casing
2	*Technology*	GSM Dualband 900/1800MHz, GPRS Class 10, Voice/Data/Fax/SMS
3	*Features*	AT command set (GSM 07.05, GSM 07.07 and

		Wavecom),Text and PDU SMS, Point to point SMS (MT/MO), Cell Broadcast, UCS2 Character Set Management , Operating LED, Metal Casing
4	*Interfaces*	SIM Plug-in card reader (1.8/3V), RS232 on 15 pin Sub-D connector, Pin power supply connector (4 pin DC), SMA antenna connector (50 ohm)
5	*Hardware*	Size(mm) 73x54x25, Weight(g) 65, Operating temperature -20~+55oC, Storage temperature -25~+70oC, Input voltage 5 to 32V DC
6	*Certifications/Compliances*	CE Certification, GCF Certification, ROHS Compliance

1.1.1.2 Implementasi Perangkat Lunak

Implementasi ini merupakan penjelasan dari aplikasi dan sistem operasi yang digunakan pada setiap perangkat keras dari arsitektur jaringan. Berikut ini adalah aplikasi dan sistem operasi yang digunakan pada setiap perangkat, yang dipaparkan pada Tabel 4.6 sampai dengan Tabel 4.8.

Tabel 4.6 Perangkat Lunak Pada *Server*

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Ubuntu 12.04
2	Server Web	Apache 2.2.22
3	Bahasa Pemrograman	PHP 5.3.10
4	Database	MySQL 5.5.35

Tabel 4.7 Perangkat Lunak Pada Komputer *Monitoring*

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows XP, Windows 7
2	Bahasa Pemrograman	C#.NET 2010
3	Database	MySQL 5.6.12

Tabel 4.8 Perangkat Lunak Pada *Attacker*

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows 7, Ubuntu 14.04
2	*Tools*	Acunetix, SQLMap

1.1.1.3 Implementasi dan Konfigurasi Arsitektur Jaringan

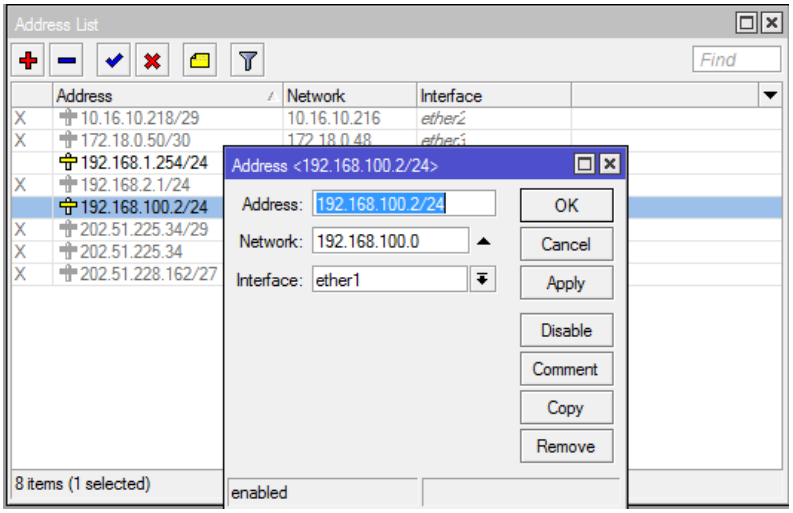
Implementasi ini merupakan penjelasan bagaimana cara mengkonfigurasi router dan pengalamatan pada masing-masing perangkat komputer.

1. Konfigurasi Router

   Konfigurasi router menggunakan winbox, konfigurasinya hanya mengkonfigurasi *IP Address* dan NAT. Berikut ini langkah-langkah konfigurasi pada router.

   a. Setting *IP Address*

      Masuk ke winbox, pada menu pilih IP → Addresess.

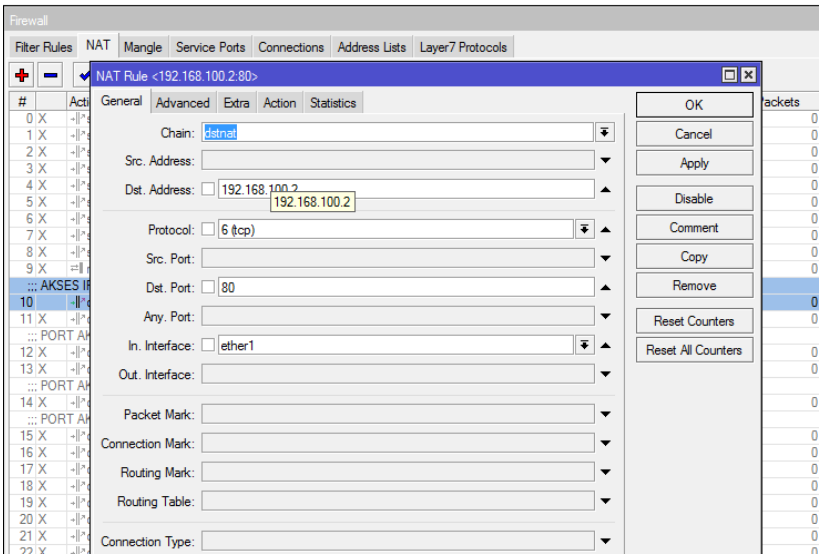


Gambar 4.2 Konfigurasi *IP* pada *Router*

Klik icon , dan masukan nilai “Address”, “Network” dan penggunaan “Interface”, kemudian klik tombol “OK”.

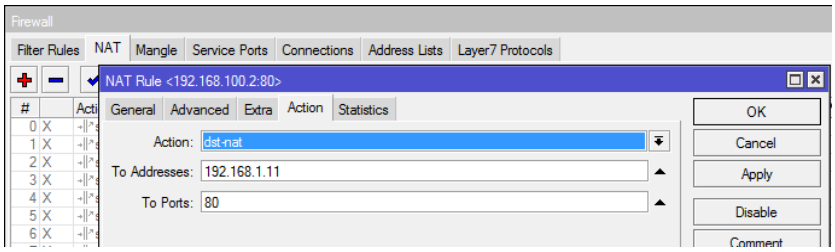
b. Setting NAT

Masuk ke winbox, pada menu pilih IP → Firewall → NAT.



Gambar 4.3 Konfigurasi NAT *Tab General*

Klik icon , dan masukan nilai "Chain", "Dst. Address", "Protocol", "Dst.Port" dan "In. Interface" pada tab "General", kemudian klik tab "Action".



Gambar 4.4 Konfigurasi NAT *Tab Action*

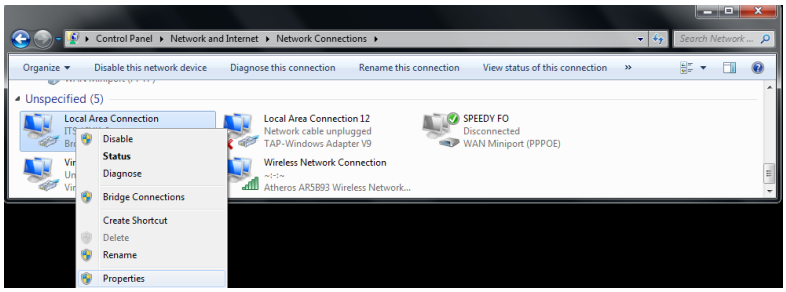
Pilih tipe "Action", dan arahkan ke *IP private* lokal pada "To Addresses" dan "To Ports", kemudian klik tombol "OK".

2. Konfigurasi Perangkat Keras Komputer

Konfigurasi ini adalah cara untuk mengkonfigurasi *ip address* pada setiap perangkat komputer dalam arsitektur jaringan, pada dasarnya cara konfigurasi *ip address* setiap komputer sama jika sistem operasi yang digunakan sama. Pada penelitian ini menggunakan 2

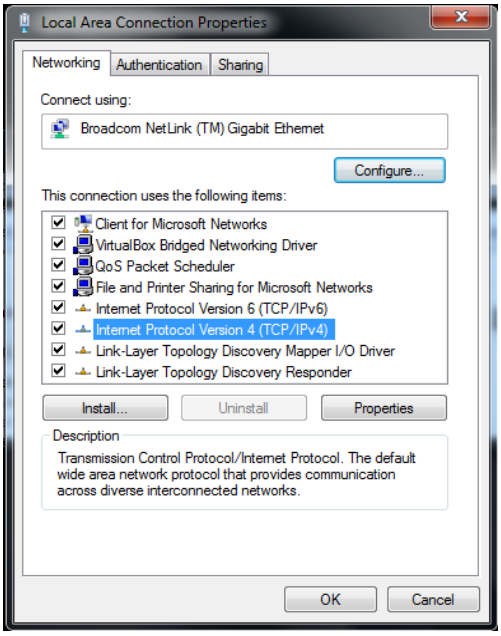
jenis sistem operasi yaitu windows dan linux. Berikut ini cara mengkonfigurasi *ip address* pada windows dan linux.

1. Konfigurasi *IP Address* pada windows
   a. Masuk ke menu *control panel* → *network and internet* → *network and sharing center* → *change adapter setting* → pada *local area connection* klik kanan pilih *properties.*



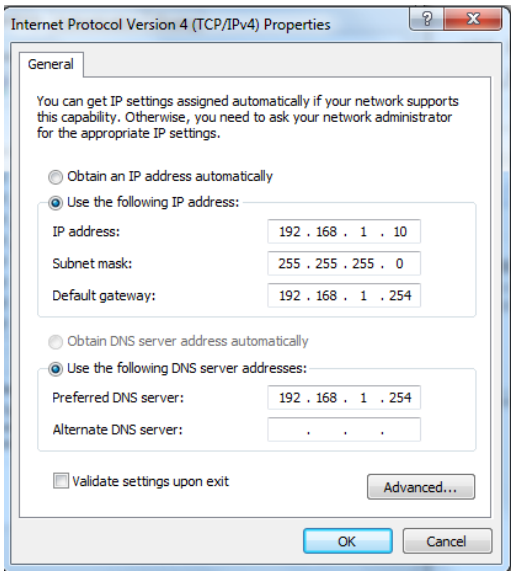
Gambar 4.5 Konfigurasi *IP* pada *windows* 7

   b. Setelah muncul *dialog local area connection properties* lalu pilih *internet protocol version 4 (TCP/Ipv4)* → *Properties*.



Gambar 4.6 Kotak *Dialog Local Area Connection Properties*

c. Pada kolom isian *ip address*, *subnet mask*, *default gateway*.



Gambar 4.7 Konfigurasi IP

2. Konfigurasi *IP Address* pada linux

a. Masuk ke sistem *directory* cd /etc/network

```
root@ubuntu:/home/scbudiman# cd /etc/network
```

Gambar 4.8 *Command Change Directory*

b. Buka *file interface* dengan menggunakan *command nano interface*

```
root@ubuntu:/etc/network# nano interfaces
```

Gambar 4.9 *Command Open File*

c. *Config* pada *file* tersebut

```
  GNU nano 2.2.6               File: interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth1
iface eth1 inet static
        address 192.168.1.11
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        gateway 192.168.1.254
#       dns-nameserver 8.8.8.8
#       dns-nameserver 192.168.3.254
```

Gambar 4.10 *Edit Config IP Address*

d. Jika sudah selesai, maka *save* dengan *command* Ctrl+o, kemudian *exit* dengan *command* Ctrl+x, dan restart *config* dengan *command*



Gambar 4.11 *Restart Service*

kemudian cek *config* dengan *command*



Gambar 4.12 Cek *List IP Address*

1.1.2 Implementasi Aplikasi

Implementasi ini merupakan penjelasan mengenai cara pengimplementasian perancangan dari *rule*, pengimplementasian perancangan basis data, dan pengimplementasian perancangan aplikasi *monitoring*.

1.1.2.1 Implementasi Rules

Implementasi ini merupakan pengimplementasian dari perancangan pada poin 3.2.3. *Rule* digunakan untuk mendeteksi adanya serangan atau aktivitas yang dilakukan oleh *Attacker* terhadap *web server*. Berikut ini adalah *rule-rule* yang digunakan untuk mendeteksi aktivitas-aktivitas tersebut.

1. *Rule – Tools*

   Merupakan bagian *rule* yang bertugas mengidentifikasi tool serangan yang dilakukan peretas.

Tabel 4.9 Implementasi *Rule - Tools*

`^.+(acunetix|sql|sqlmap|sql-map).*$`

2. *Rule – Injection*

   Merupakan bagian *rule* yang bertugas mengidentifikasi serangan pada kategori SQL Injection.

Tabel 4.10 Implementasi *Rule - Injection*

```
^.+(<!!27|sleep|select|delay|wait|order|having|uni
on).*$
```

3. *Rule* – XSS

   Merupakan bagian *rule* yang bertugas mengidentifikasi serangan pada kategori *Cross Site Scripting*.

Tabel 4.11 Implementasi *Rule - XSS*

```
^.+(<|script|--|!--).*$
```

4. *Rule – Command*

   Merupakan bagian *rule* yang bertugas mengidentifikasi serangan pada kategori *Command*.

Tabel 4.12 Implementasi *Rule - Command*

```
^.+(etc/passwd|windows/win.ini).*$
```

5. *Rule – Anomaly*

   Merupakan bagian *rule* yang bertugas mengidentifikasi serangan dengan melihat *incoming traffic* pada *web server*.

Tabel 4.13 Implementasi *Rule - Anomaly*

```
BEGIN
    DECLARE v_IP_Server varchar(15);
    DECLARE v_IP varchar(15);
    DECLARE v_Tanggal varchar(12);
    DECLARE v_Jam varchar(18);
    DECLARE v_Paket int(11);
    DECLARE v_ID int(6);
    DECLARE v_IDServer int(6);
    DECLARE v_Qty int(6);
    DECLARE v_QtyHit int(6);
    DECLARE v_QtyMax int(6);
    DECLARE v_Command varchar(500);
    DECLARE v_Status int(1);

    SET v_IP_Server = New.dtJenis;
    SET v_IP        = New.dtRemote_IP;
    SET v_Tanggal   = New.dtRemote_Tanggal;
    SET v_Jam       = New.dtRemote_Jam;
    SET v_Paket     = New.dtRemote_Req_Paket;
    SET v_Command   = REPLACE("ip firewall filter add
```

```
chain=input src-address=DTIP action=drop#ip firewall filter add
chain=forward src-address=DTIP action=drop", "DTIP", v_IP);

    select count(*) as `Qty`, dtRemote_Qty_Akses, idData,
dtStatus into v_Qty, v_QtyHit, v_ID, v_Status from
db_ssh_log.dtlog_detail
    where db_ssh_log.dtlog_detail.dtRemote_IP=v_IP and
db_ssh_log.dtlog_detail.dtRemote_Tanggal=v_Tanggal and
db_ssh_log.dtlog_detail.dtRemote_Jam=v_Jam limit 0,1;

    if v_Qty=1 then #update
       update db_ssh_log.dtlog_detail set
           dtRemote_IP            = v_IP,
           dtRemote_Tanggal       = v_Tanggal,
           dtRemote_Jam           = v_Jam,
           dtRemote_Qty_Akses     = dtRemote_Qty_Akses+1,
           dtRemote_Qty_Paket     = dtRemote_Qty_Paket+v_Paket
       where db_ssh_log.dtlog_detail.dtRemote_IP=v_IP and
db_ssh_log.dtlog_detail.dtRemote_Tanggal=v_Tanggal and
db_ssh_log.dtlog_detail.dtRemote_Jam=v_Jam;

       select Max_Hit into v_QtyMax from
db_ssh_log.dtserver_hit_max;
       if v_QtyHit > v_QtyMax and v_Status=0 then
          #update Status
          update db_ssh_log.dtlog_detail set
db_ssh_log.dtlog_detail.dtStatus=2 where
db_ssh_log.dtlog_detail.idData=v_ID;

          #GET SERVER
          select db_ssh_log.dtserver.idRouter into v_IDServer
from db_ssh_log.dtserver where
db_ssh_log.dtserver.dtIP=v_IP_Server limit 0,1;

          #insert Command
          select count(*) as `Qty` into v_Qty from
db_ssh_log.dtlog_detail
          where db_ssh_log.dtlog_detail.dtRemote_IP=v_IP and
(db_ssh_log.dtlog_detail.dtStatus=2 or
db_ssh_log.dtlog_detail.dtStatus=3);

          if v_Qty<=1 then
             insert into db_ssh_log.dtserver_jadwal set
db_ssh_log.dtserver_jadwal.dtDate=CURDATE(),
db_ssh_log.dtserver_jadwal.dtTime=CURTIME(),
db_ssh_log.dtserver_jadwal.idServer=v_IDServer,
db_ssh_log.dtserver_jadwal.dtCommand=v_Command,
db_ssh_log.dtserver_jadwal.dtFrom="ANOMALI";
          end if;
       end if;
    else            #insert
       insert into db_ssh_log.dtlog_detail set
           dtIP_Server            = v_IP_Server,
           dtRemote_IP            = v_IP,
           dtRemote_Tanggal       = v_Tanggal,
           dtRemote_Jam           = v_Jam,
           dtRemote_Qty_Akses     = dtRemote_Qty_Akses+1,
           dtRemote_Qty_Paket     = dtRemote_Qty_Paket+v_Paket;
    end if;
END;
```

1.1.2.2 Implementasi Basis Data

Implementasi ini merupakan pengimplementasian *database* yang digunakan oleh aplikasi *monitoring*. Pada impelementasi ini akan dijelaskan beberapa sintax MySQL yang digunakan untuk membuat *table-table* dari *database* aplikasi *monitoring*. Berikut ini adalah sintax-sintax MySQL tersebut.

1. Membuat *table* dtlog

Tabel 4.14 Implementasi *table* dtlog

```
CREATE TABLE `dtlog` (
  `idData` int(11) NOT NULL AUTO_INCREMENT,
  `dtDate` date DEFAULT NULL,
  `dtTime` time DEFAULT NULL,
  `dtLog` text COLLATE latin1_general_ci,
  `dtJenis` varchar(100) COLLATE latin1_general_ci DEFAULT NULL,
  `dtRemote_IP` varchar(15) COLLATE latin1_general_ci DEFAULT NULL,
  `dtRemote_Tanggal` varchar(12) COLLATE latin1_general_ci DEFAULT
NULL,
  `dtRemote_Jam` varchar(8) COLLATE latin1_general_ci DEFAULT NULL,
  `dtRemote_Req` text COLLATE latin1_general_ci,
  `dtRemote_Req_RC` int(6) DEFAULT NULL,
  `dtRemote_Req_Paket` int(6) DEFAULT NULL,
  `dtRemote_Req_Referer` text COLLATE latin1_general_ci,
  `dtRemote_Req_Akses` varchar(255) COLLATE latin1_general_ci
DEFAULT NULL,
  `dtRemote_Req_AddData` varchar(255) COLLATE latin1_general_ci
DEFAULT NULL,
  `dtKeterangan` varchar(255) COLLATE latin1_general_ci DEFAULT
NULL,
  PRIMARY KEY (`idData`)
) ENGINE=InnoDB AUTO_INCREMENT=0 DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci;
```

2. Membuat *table* dtlog_detail

Tabel 4.15 Implementasi *table* dtlog_detail

```
CREATE TABLE `dtlog_detail` (
  `idData` int(11) NOT NULL AUTO_INCREMENT,
  `dtIP_Server` varchar(15) COLLATE latin1_general_ci DEFAULT NULL,
  `dtRemote_IP` varchar(15) COLLATE latin1_general_ci DEFAULT NULL,
  `dtRemote_Tanggal` varchar(12) COLLATE latin1_general_ci DEFAULT
NULL,
  `dtRemote_Jam` varchar(8) COLLATE latin1_general_ci DEFAULT NULL,
  `dtRemote_Qty_Akses` int(4) DEFAULT '0',
  `dtRemote_Qty_Paket` int(11) DEFAULT '0',
  `dtStatus` int(1) DEFAULT '0' COMMENT '2: Status Anomali 3:
Status Signature',
  PRIMARY KEY (`idData`)
) ENGINE=InnoDB AUTO_INCREMENT=0 DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci;
```

3. Membuat *table* dtserver

Tabel 4.16 Implementasi *table* dtserver

```
CREATE TABLE `dtserver` (
  `idData` int(11) NOT NULL AUTO_INCREMENT,
  `dtNama_Server` varchar(30) COLLATE latin1_general_ci DEFAULT
NULL,
  `dtIP` varchar(15) COLLATE latin1_general_ci DEFAULT NULL,
  `dtPort` int(6) DEFAULT NULL,
  `dtUser` varchar(30) COLLATE latin1_general_ci DEFAULT NULL,
  `dtPassword` varchar(200) COLLATE latin1_general_ci DEFAULT NULL,
  `idJenis` int(1) DEFAULT NULL,
  `idRouter` int(4) DEFAULT '0',
  PRIMARY KEY (`idData`),
  KEY `fk_idJenis` (`idJenis`),
  CONSTRAINT `fk_idJenis` FOREIGN KEY (`idJenis`) REFERENCES
`dtserver_jenis` (`idJenis`)
)    ENGINE=InnoDB    AUTO_INCREMENT=0    DEFAULT    CHARSET=latin1
COLLATE=latin1_general_ci;
```

4. Membuat *table* dtserver_hit_max

Tabel 4.17 Implementasi *table* dtserver_hit_max

```
CREATE TABLE `dtserver_hit_max` (
  `Max_Hit` int(6) NOT NULL DEFAULT '0',
  PRIMARY KEY (`Max_Hit`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;
```

5. Membuat *table* dtserver_ids_rule

Tabel 4.18 Implementasi *table* dtserver_ids_rule

```
CREATE TABLE `dtserver_ids_rule` (
  `idData` int(11) NOT NULL AUTO_INCREMENT,
  `dtNama_Rule` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `dtRule` varchar(500) COLLATE latin1_general_ci DEFAULT NULL,
  PRIMARY KEY (`idData`)
)    ENGINE=InnoDB    AUTO_INCREMENT=0    DEFAULT    CHARSET=latin1
COLLATE=latin1_general_ci;
```

6. Membuat *table* dtserver_ip_whitelist

Tabel 4.19 Implementasi *table* dtserver_ip_whitelist

```
CREATE TABLE `dtserver_ip_whitelist` (
  `idData` int(11) NOT NULL AUTO_INCREMENT,
  `dtIP` varchar(15) COLLATE latin1_general_ci DEFAULT NULL,
  PRIMARY KEY (`idData`)
)    ENGINE=InnoDB    AUTO_INCREMENT=0    DEFAULT    CHARSET=latin1
COLLATE=latin1_general_ci;
```

7. Membuat *table* dtserver_jadwal

Tabel 4.20 Implementasi *table* dtserver_jadwal

```
CREATE TABLE `dtserver_jadwal` (
  `idData` int(11) NOT NULL AUTO_INCREMENT,
  `dtDate` date DEFAULT NULL,
  `dtTime` time DEFAULT NULL,
  `idServer` int(11) DEFAULT NULL,
  `dtStatus` int(1) DEFAULT '0',
  `dtDate_Execute` date DEFAULT NULL,
  `dtTime_Execute` time DEFAULT NULL,
  `dtCommand` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,
  `dtFrom` varchar(100) COLLATE latin1_general_ci DEFAULT NULL,
  `dtSendInfo` int(1) DEFAULT '0',
  PRIMARY KEY (`idData`),
  KEY `fk_idServer` (`idServer`),
  KEY `fk_statusServer` (`dtStatus`),
  CONSTRAINT `fk_idServer` FOREIGN KEY (`idServer`) REFERENCES
`dtserver` (`idData`),
  CONSTRAINT `fk_statusServer` FOREIGN KEY (`dtStatus`) REFERENCES
`dtserver_status` (`idData`)
)    ENGINE=InnoDB    AUTO_INCREMENT=0    DEFAULT    CHARSET=latin1
COLLATE=latin1_general_ci;
```

8. Membuat *table* dtserver_jenis

Tabel 4.21 Implementasi *table* dtserver_jenis

```
CREATE TABLE `dtserver_jenis` (
  `idJenis` int(11) NOT NULL AUTO_INCREMENT,
  `dtJenis` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,
  PRIMARY KEY (`idJenis`)
)    ENGINE=InnoDB    AUTO_INCREMENT=0    DEFAULT    CHARSET=latin1
COLLATE=latin1_general_ci;
```

9. Membuat *table* dtserver_status

Tabel 4.22 Implementasi *table* dtserver_status

```
CREATE TABLE `dtserver_status` (
  `idData` int(11) NOT NULL AUTO_INCREMENT,
  `dtStatus` varchar(30) COLLATE latin1_general_ci DEFAULT NULL,
  PRIMARY KEY (`idData`)
)    ENGINE=InnoDB    AUTO_INCREMENT=0    DEFAULT    CHARSET=latin1
COLLATE=latin1_general_ci;
```

10. Membuat *table* user

Tabel 4.23 Implementasi *table* user

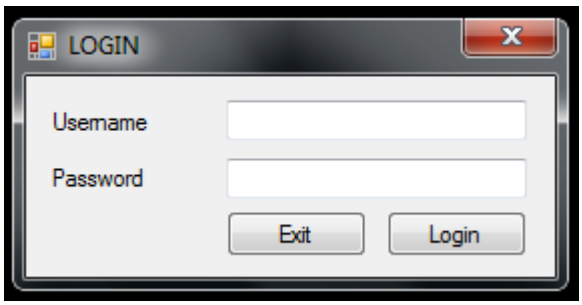
```
CREATE TABLE `user` (
  `idData` int(11) NOT NULL AUTO_INCREMENT,
  `dtDate` date DEFAULT NULL,
  `dtTime` time DEFAULT NULL,
  `dtNama` varchar(30) COLLATE latin1_general_ci DEFAULT NULL,
  `userName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `userPass` varchar(100) COLLATE latin1_general_ci DEFAULT NULL,
  PRIMARY KEY (`idData`)
)    ENGINE=InnoDB    AUTO_INCREMENT=0    DEFAULT    CHARSET=latin1
COLLATE=latin1_general_ci;
```

1.1.2.3 Implementasi Antarmuka

Implementasi ini merupakan implementasi dari perancangan antar muka pada poin 3.2.4.2.2. Dari perancangan tersebut telah diterapkan ke dalam pengkodean berbasis *desktop*. Berikut ini adalah antarmuka dari aplikasi monitoring.

1. *Login*

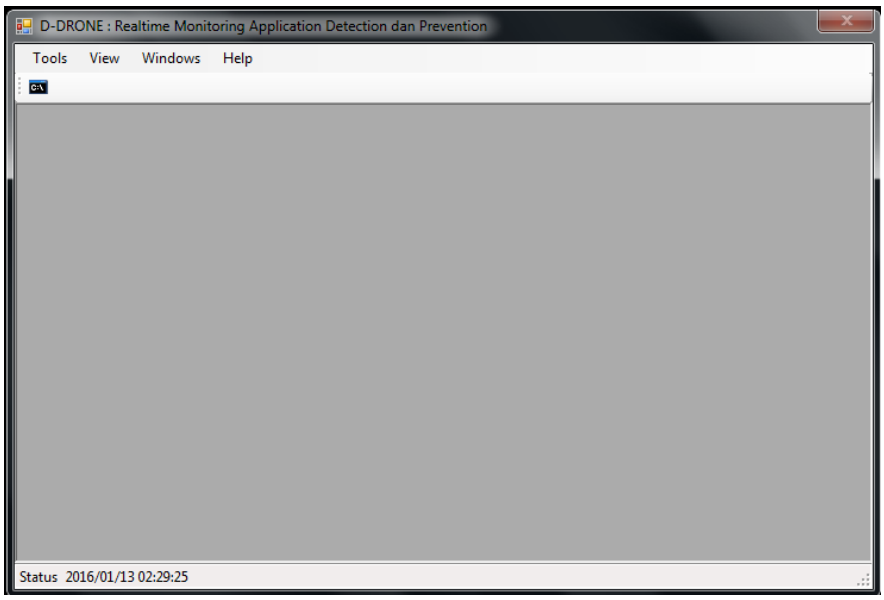
   Tampilan ini digunakan untuk melakukan otentifikasi *username* dan *password* sebelum masuk ke dalam aplikasi.



Gambar 4.13 Implementasi Antarmuka Login

2. Menu Utama

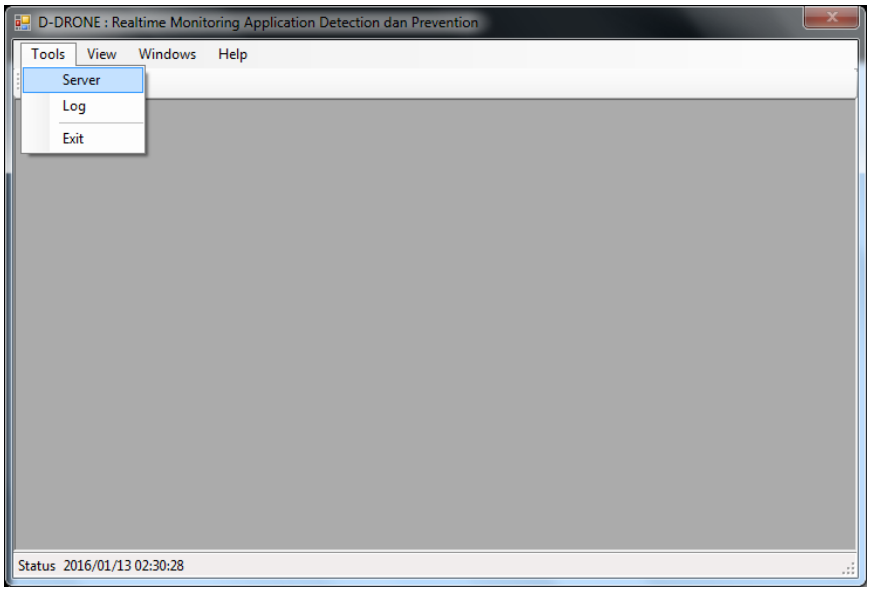
   Tampilan utama yang tampil saat otentifikasi bernilai valid. Menu utama pada aplikasi ada di menu *Tools*.



Gambar 4.14 Implementasi Antarmuka Menu Utama

3. Tampilan *Tools – Server*

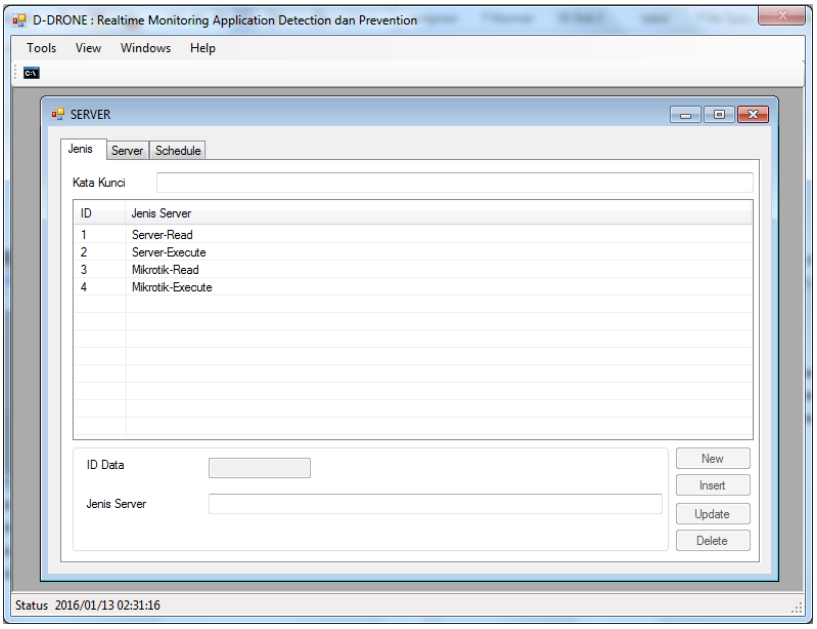
   Tampilan menu *Tools* untuk masuk ke menu *server*, *log* atau keluar dari aplikasi.



Gambar 4.15 Implementasi Antarmuka Menu *Tools – Server*

1. *Tab* Jenis

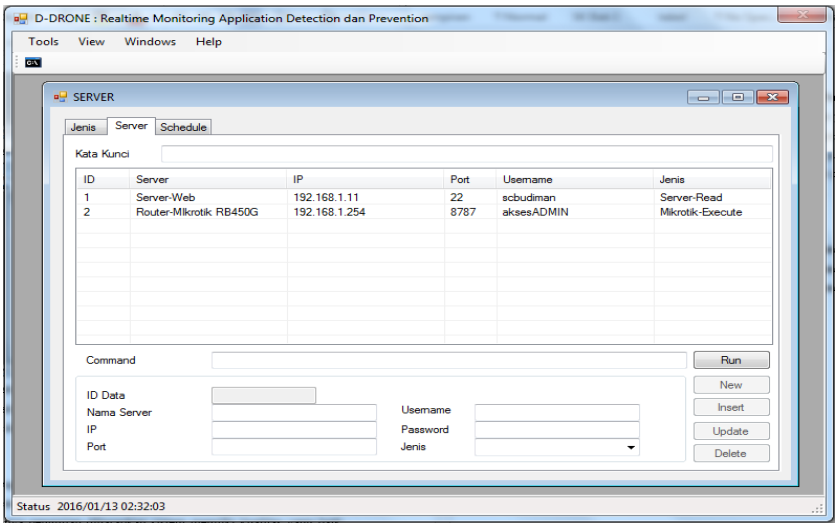
   Tampilan untuk menambah, mengubah, dan menghapus tipe jenis dari *server* ataupun *router*.



Gambar 4.16 Implementasi Antarmuka *Tab* Jenis

2. *Tab Server*

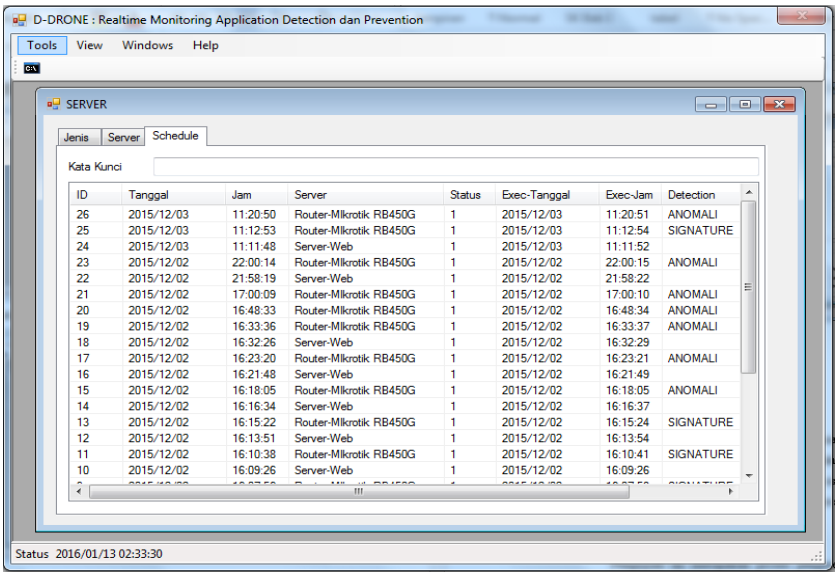
   Tampilan untuk menambah, mengubah, dan menghapus data *server* ataupun *router*.



Gambar 4.17 Implementasi Antarmuka *Tab Server*

3. *Tab Schedule*

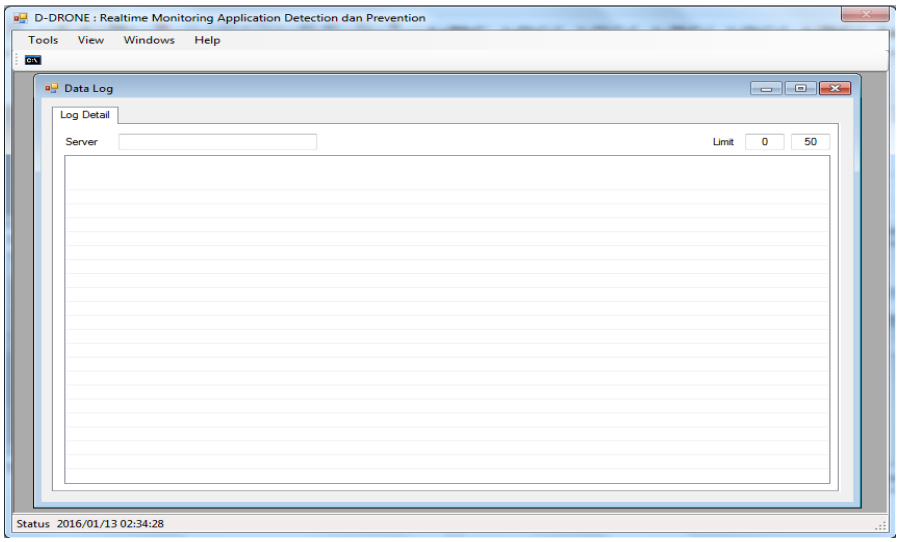
   Tampilan yang menampilkan informasi riwayat *monitoring* yang dijadwalkan oleh admin pada *server* dan informasi instrusion yang terjadi pada *server*.



Gambar 4.18 Implementasi Antarmuka *Tab Schedule*

4. Tampilan *Tools – Log*

   Tampilan yang menampikan data *log* pada *server* yang dibaca secara *realtime* dan telah dilakukan analisa *intrusion*.



Gambar 4.19 Implementasi Antarmuka Menu *Tools – Log*

1.2 Pengujian

Pengujian merupakan proses pengeksekusian dari implementasi yang telah dilakukan untuk menilai dan menemukan kesalahan-kesalahan yang terdapat pada sistem. Dengan adanya pengujian diharapkan sistem memilki kualitas yang baik dan dapat dijadikan bahan referensi dalam proses pengembangan sistem lebih lanjut. Pada pengujian akan dibagi menjadi dua, yaitu :

1. Pengujian arsitektur jaringan
2. Pengujian dan penilaian aplikasi

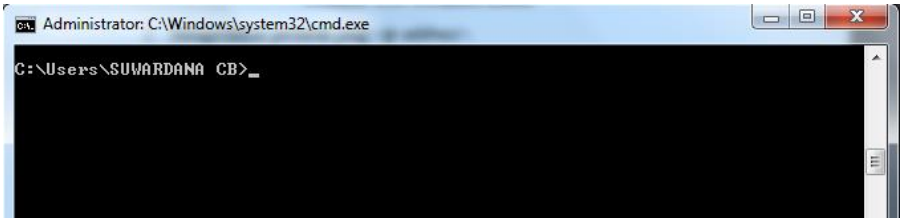
1.2.1 Pengujian Arsitektur Jaringan

Pengujian ini merupakan proses pengecekan terhadap keberhasilan dari instalasi arsitektur jaringan. Pengujian yang dilakukan yaitu dengan melakukan pengecekan terhadap status keterhubungan dari satu *node* ke *node* lain dan pengaksesan konten-konten yang ada di *server* oleh *client*.

1.2.1.1 Pengujian Keterhubungan Antar Perangkat

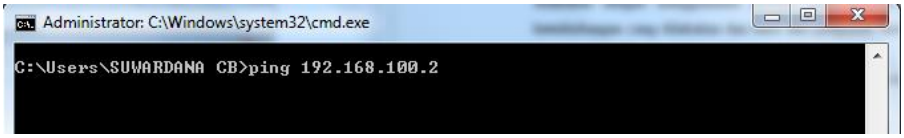
Pengujian ini merupakan pengujian yang dilakukan untuk mengecek setiap *node* atau perangkat dalam arsitektur jaringan saling terhubung. Pengujian dilakukan dengan menggunakan aplikasi ping. Berikut ini cara pengujian keterhubungan yang dilakukan dan hasil dari pengujian tersebut dijelaskan pada Gambar 4.20, Gambar 4.21 dan Gambar 4.22.

1. Membuka aplikasi *command prompt* pada *windows* atau *terminal* pada *linux*.



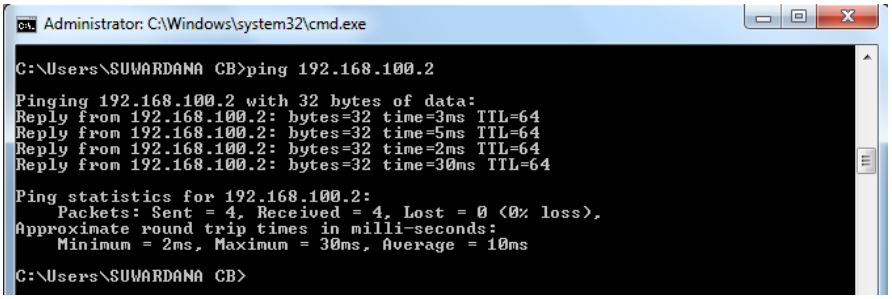
Gambar 4.20 *Command Prompt* pada *Windows*

2. Mengetikkan perintah ping <*ip address*>.



Gambar 4.21 Ping *IP Address*

3. Mengecek *response* yang diberikan oleh *ip address* yang dituju.



Gambar 4.22 Status *Ping Response*

Dari hasil pengujian yang dilakukan maka di dapatkan hasilnya yang dijelaskan pada Tabel 4.24.

Tabel 4.24 Hasil Pengujian Keterhubungan Perangkat

		Attacker		Router			PC Server		PC Monitoring IDS dan IPS
		Inner	Outer	1	2	3	1	2	
Attacker	Inner	-	-	OK	OK	OK	OK	OK	OK
	Outer	-	-	OK	OK	OK	-	-	-
Router	1	OK	OK	-	OK	OK	OK	OK	OK
	2	OK	OK	OK	-	OK	-	-	-
	3	OK	OK	OK	OK	-	-	OK	-
PC Server	1	OK	-	OK	-	-	-		OK
	2	OK	-	OK	-	OK		-	OK
PC Monitoring IDS dan IPS		OK	-	OK	-	-	OK	OK	-

1.2.2 Pengujian dan Penilaian Aplikasi

Pengujian ini merupakan proses untuk menguji aplikasi *monitoring* yang terpasang pada sebuah jaringan komputer. Parameter yang diujikan yaitu ketepatan dari aplikasi *monitoring* untuk mendeteksi jenis serangan. Berikut ini akan dijelaskan pengujian-pengujian tersebut.

1.2.2.1 Pengujian dan Penilaian Ketepatan Deteksi dan Pencegahan Serangan

Pengujian ini merupakan proses untuk menguji ketepatan deteksi serangan yang dilakukan oleh aplikasi monitoring. Pengujian dilakukan dengan jenis serangan *server scanning*, SQL *injection*, dan XSS yang dilakukan oleh *Attacker*. Proses pengujian dilakukan dengan mengisi *form* pada tabel 4.25, dimana hasil detail pengujian dapat dilihat di tabel 4.74.

Tabel 4.25 Pengujian Deteksi Serangan

NO	Jenis Serangan
1	*Scanning* : Menggunakan aplikasi Acunetix
2	*Scanning* : Menggunakan aplikasi SQLMap
3	*SQL injection* : Mendapatkan informasi database dari *server*
4	*SQL injection* : Mendapatkan informasi *database*
5	*SQL injection* : Mendapatkan informasi tabel dari *database*
6	*SQL injection* : Mendapatkan record-record dari dari table pada suatu *database*
7	*Brute Force* : Login ke webserver
8	*XSS :* Memasukkan tag html pada *web server*

1. Pengujian Ketepatan Deteksi dan Pencegahan Ke-1

Scanning : Menggunakan aplikasi Acunetix, berikut ini analisis log yang terjadi :

Tabel 4.26 Data *Log* Uji Ke-1 *Server* 1

```
192.168.100.4 - - [02/Dec/2015:15:09:45 +0700]
"CONNECT www.acunetix.wvs:443 HTTP/1.1" 405 547 "-"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1;
WOW64; Trident/5.0)"
```

Tabel 4.27 Data *Log* Uji Ke-1 *Server* 2

```
192.168.200.2 - - [01/Feb/2016:02:38:20 +0700] "GET
http://www.acunetix.wvs HTTP/1.1" 200 483 "-"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1;
WOW64; Trident/5.0)"
```

- Pemenggalan

Pada tahap ini, data *log* akan dilakukan dilakukan pemenggalan menjadi sembilan bagian.

Tabel 4.28 Proses Pemenggalan Data *Log* Uji Ke-1 *Server* 1

Bagian	Data
[1]	192.168.100.4
[2]	-
[3]	-
[4]	[02/Dec/2015:15:09:45 +0700]
[5]	"CONNECT www.acunetix.wvs:443 HTTP/1.1"
[6]	405
[7]	547
[8]	“-”
[9]	"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"

Tabel 4.29 Proses Pemenggalan Data *Log* Uji Ke-1 *Server* 2

Bagian	Data
[1]	192.168.200.2
[2]	-

[3]	-
[4]	[01/Feb/2016:02:38:20 +0700]
[5]	"GET http://www.acunetix.wvs HTTP/1.1"
[6]	200
[7]	483
[8]	“-”
[9]	"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"

- Analisis metode s*ignature* dan a*nomaly*
  - Pencocokan

    Bagian [5], [8] dan [9] diambil dan kemudian dilakukan analisa dengan menggunakan metode *signature*.

Tabel 4.30 Bagian Data Akses Data Uji Ke-1 *Server* 1

Bagian	**Data**
[5]	"CONNECT www.acunetix.wvs:443 HTTP/1.1”
[8]	“-”
[9]	"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"

Tabel 4.31 Bagian Data Akses Data Uji Ke-1 *Server* 2

Bagian	**Data**
[5]	"GET http://www.acunetix.wvs HTTP/1.1"
[8]	“-”
[9]	"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"

Dengan mengirim data tersebut ke algoritma pengecekan pada tabel 3.8, dari pengecekan tersebut didapat data sebagai berikut :

Tabel 4.32 *Output* Pencocokan Data Uji Ke-1 *Server* 1 dan *Server* 2 dengan IDS *Rule*

Kategori	***Rule***	**Data**	**Hasil**
TOOLS	^.+(acunetix|sql|sqlmap|sql-map).*$	[5], [8] dan [9]	***TRUE***

INJECTION	^.+(<!!27|sleep|select|delay|wait).*$	[5], [8] dan [9]	***FALSE***
XSS	^.+(<|script|--|!--).*$	[5], [8] dan [9]	***FALSE***
COMMAND	^.+(etc/passwd|windows/win.ini).*$	[5], [8] dan [9]	***FALSE***

- Rekam *Traffic*

Bagian [4] diambil dan kemudian dilakukan analisa dengan menggunakan metode *anomaly*.

Tabel 4.33 Bagian Data Waktu Data Uji Ke-1 *Server* 1

Bagian	**Data**
[4]	[02/Dec/2015:15:09:45 +0700]

Tabel 4.34 Bagian Data Waktu Data Uji Ke-1 *Server* 2

Bagian	**Data**
[4]	[01/Feb/2016:02:38:20 +0700]

Dengan mengirim data tersebut ke algoritma *rule hit* pada tabel 3.9. Dari data *log* yang terjadi selama proses *scanning*, mendapati bawah pada waktu akses tersebut mendapati ada 45 *hit* untuk *server* 1 dan 30 *hit* untuk *server* 2, yang masuk dalam waktu yang sama pada IP yang sama.

Tabel 4.35 *Output* Analisa *Traffic* Data Uji Ke-1 dengan IDS *Rule*

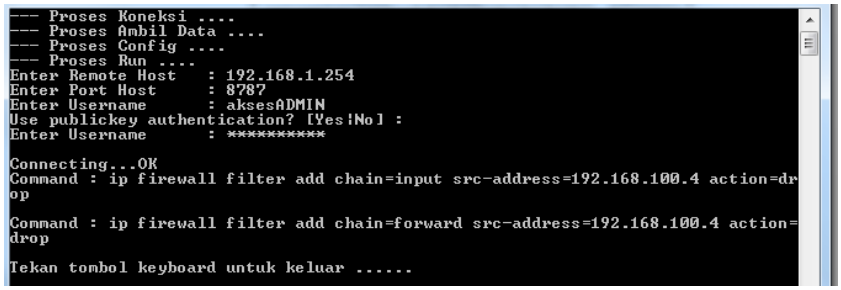
Server	**Tanggal**	**Jam**	***Qty Hit***	**Hasil**
1	02/Dec/2015	15:09:45	45	***TRUE***
2	01/Feb/2016	02:38:20	30	***TRUE***

- Hasil *Final*

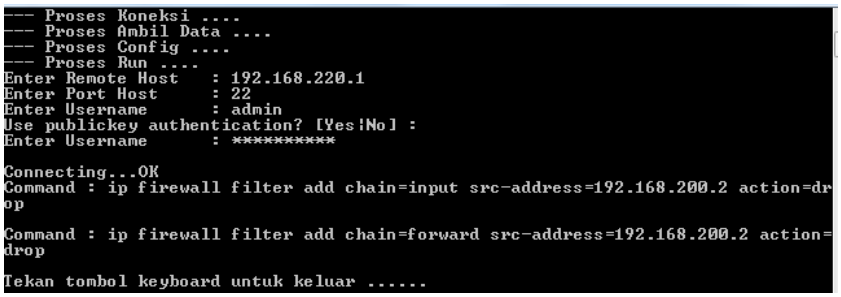
Dari proses pencocokan data bagian ke lima, delapan dan ke sembilan dari data *log*, didapat hasil bahwa akses yang masuk ke *server* dimana data-nya tergolong *intrusion* kategori "*TOOLS*". Kemudian aplikasi mengupdate jenis *intrusion* pada akses tersebut, dan melakukan *prevention* dengan mengambil bagian ke satu data *log* yaitu IP, untuk di blok.

Selain terdeteksi *rule signature*, akses juga terdeteksi *anomaly*, dikarenakan satu *IP* dalam waktu yang sama melakukan akses lebih dari batas maksimal yang telah ditetapkan. Jika *IP* belum dilakukan *prevention*, saat terdeteksi akan dilakukan pemblokiran akses pada IP tersebut, begitu juga sebaliknya.

Adapun pencegahan yang dilakukan dengan mengirimkan informasi ke *router* yaitu *command prevention*.



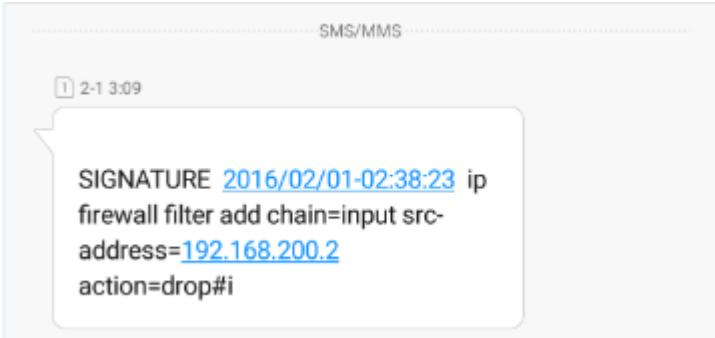
Gambar 4.23 Mengirim *Command Prevention* ke Router Uji Ke-1 *Server* 1



Gambar 4.24 Mengirim *Command Prevention* ke Router Uji Ke-1 *Server* 2



Gambar 4.25 Informasi *Alert* SMS Uji Ke-1 *Server* 1



Gambar 4.26 Informasi *Alert* SMS Uji Ke-1 *Server* 2

2. Pengujian Ketepatan Deteksi dan Pencegahan Ke-2

Scanning : Menggunakan aplikasi SQLMap, berikut ini analisis log yang terjadi :

Tabel 4.36 Data *Log* Uji Ke-2 *Server* 1

```
192.168.2.11 - - [03/Dec/2015:02:56:20 +0700] "GET
/onlineplus/?cmd=info%22%28%29%27%2C%22%27..%27&tmplt
=2&vr=8807&pos=artikel&scat=4 HTTP/1.1" 200 4777 "-"
"sqlmap/1.0-dev-d772e7e (http://sqlmap.org)"
```

Tabel 4.37 Data *Log* Uji Ke-2 *Server* 2

```
192.168.200.3 - - [01/Feb/2016:03:17:21 +0700] "GET
/onlineplus/?cmd=info&tmplt=2&vr=1838&pos=artikel&sca
t=3 HTTP/1.1" 404 466 "-" "sqlmap/1.0-dev-80d3ff6
(http://sqlmap.org)"
```

- Pemenggalan

Pada tahap ini, data *log* akan dilakukan dilakukan pemenggalan menjadi sembilan bagian.

Tabel 4.38 Proses Pemenggalan Data *Log* Uji Ke-2 *Server* 1

Bagian	Data
[1]	192.168.2.11
[2]	-
[3]	-
[4]	[03/Dec/2015:02:56:20 +0700]
[5]	"GET /onlineplus/?cmd=info%22%28%29%27%2C%22%27..%27&tmplt=2&vr=8807&pos=artikel&scat=4 HTTP/1.1"
[6]	200
[7]	4777
[8]	“-”
[9]	"sqlmap/1.0-dev-d772e7e (http://sqlmap.org)"

Tabel 4.39 Proses Pemenggalan Data *Log* Uji Ke-2 *Server* 2

Bagian	Data
[1]	192.168.200.3
[2]	-

[3]	-
[4]	[01/Feb/2016:03:17:21 +0700]
[5]	"GET /onlineplus/?cmd=info&tmplt=2&vr=1838&pos=artikel&scat=3 HTTP/1.1"
[6]	404
[7]	466
[8]	“-”
[9]	"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"

- Analisis metode s*ignature* dan *anomaly*
  - Pencocokan

    Bagian [5], [8] dan [9] diambil dan kemudian dilakukan analisa dengan menggunakan metode *signature*.

Tabel 4.40 Bagian Data Akses Data Uji Ke-2 *Server* 1

Bagian	**Data**
[5]	"GET /onlineplus/?cmd=info%22%28%29%27%2C%22%27..%27&tmplt=2&vr=8807&pos=artikel&scat=4 HTTP/1.1"
[8]	“-”
[9]	"sqlmap/1.0-dev-d772e7e (http://sqlmap.org)"

Tabel 4.41 Bagian Data Akses Data Uji Ke-2 *Server* 2

Bagian	**Data**
[5]	"GET /onlineplus/?cmd=info&tmplt=2&vr=1838&pos=artikel&scat=3 HTTP/1.1"
[8]	“-”
[9]	"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"

Dengan mengirim data tersebut ke algoritma pengecekan pada tabel 3.8, dari pengecekan tersebut didapat data sebagai berikut :

Tabel 4.42 *Output* Pencocokan Data Uji Ke-2 *Server* 1 dan *Server* 2 dengan IDS *Rule*

Kategori	*Rule*	Data	Hasil
TOOLS	^.+(acunetix|sql|sqlmap|sql-map).*$	[5], [8] dan [9]	***TRUE***
INJECTION	^.+(<!!27|sleep|select|delay|wait).*$	[5], [8] dan [9]	***FALSE***
XSS	^.+(<|script|--|!--).*$	[5], [8] dan [9]	***FALSE***
COMMAND	^.+(etc/passwd|windows/win.ini).*$	[5], [8] dan [9]	***FALSE***

- Rekam *Traffic*

  Bagian [4] diambil dan kemudian dilakukan analisa dengan menggunakan metode *anomaly*.

Tabel 4.43 Bagian Data Waktu Data Uji Ke-2

Server	Bagian	Data
1	[4]	[03/Dec/2015:02:56:20 +0700]
2	[4]	[01/Feb/2016:03:17:21 +0700]

Dengan mengirim data tersebut ke algoritma *rule hit* pada tabel 3.9. Dari data *log* yang terjadi selama proses *scanning*, mendapati bawah pada waktu akses tersebut mendapati ada 8 *hit* pada *server* 1 dan 1 *hit* pada *server* 2 yang masuk dalam waktu yang sama pada IP yang sama.

Tabel 4.44 *Output* Analisa *Traffic* Data Uji Ke-2 dengan IDS *Rule*

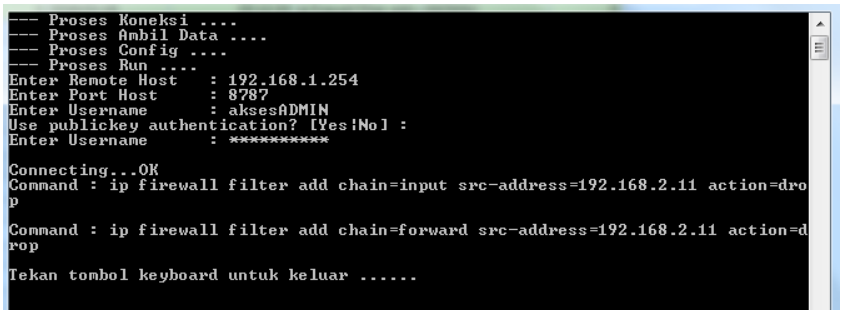
Server	Tanggal	Jam	*Qty Hit*	Hasil
1	03/Dec/2015	02:56:20	8	***FALSE***
2	01/Feb/2016	03:17:21	1	***FALSE***

- Hasil *Final*

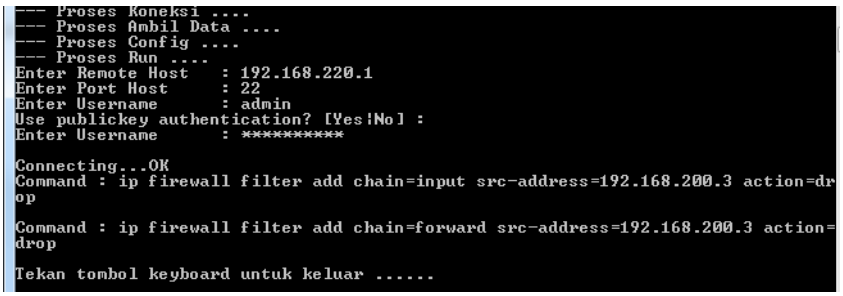
Dari proses pencocokan data bagian ke lima, delapan dan ke sembilan dari data *log*, didapat hasil bahwa akses yang masuk ke *server* dimana data-nya tergolong *intrusion* kategori "*TOOLS*". Kemudian aplikasi mengupdate jenis *intrusion* pada akses tersebut, dan melakukan *prevention* dengan mengambil bagian ke satu data *log* yaitu IP, untuk di blok.

Tetapi tidak mendapati adanya anomali deteksi, dikarenakan atas pengecekan berdasarkan waktu akses, hit yang masuk masih dibawah ambang batas atas atau maksimum.

Adapun pencegahan yang dilakukan dengan mengirimkan informasi ke *router* yaitu *command prevention*.



Gambar 4.27 Mengirim *Command Prevention* ke Router Uji Ke-2 *Server* 1



Gambar 4.28 Mengirim *Command Prevention* ke Router Uji Ke-2 *Server* 2



Gambar 4.29 Informasi *Alert* SMS Uji Ke-2 *Server* 1



Gambar 4.30 Informasi *Alert* SMS Uji Ke-2 *Server* 2

3. Pengujian Ketepatan Deteksi dan Pencegahan Ke-3

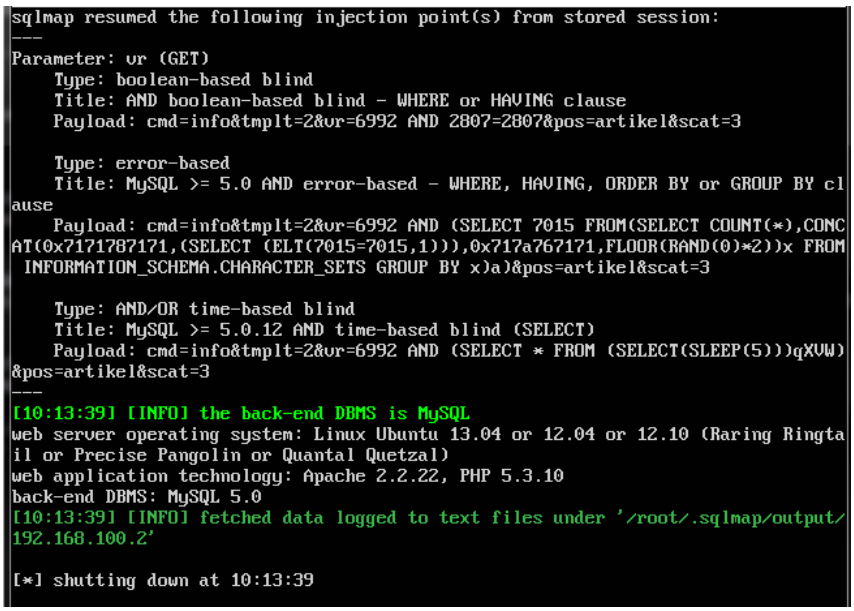
SQL injection : Mendapatkan informasi database dari *server*. Pada pengecekan dengan menggunakan Acunetix gambar 3.4, didapat informasi *server* tetapi tidak menampilkan informasi versi berapa atas database, dan jenis database apa yang digunakan. Pengujian akan menggunakan SQLMap, untuk mencari informasi detail dari *server* target.

Mekanisme ini dilakukan pada URL yang biasanya terdapat parameter yang dikirim melalui metode GET atau POST. Pada kasus ini *attacker* mendapatkan url http://192.168.100.2/onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&scat=3.

Command Attacker :

```
root@ubuntu-devel:/opt/sqlmap# python sqlmap.py -u
"http://192.168.100.2/onlineplus/?cmd=info&tmplt=2&vr=6992&
pos=artikel&scat=3"
```

Hasil *Attacker* :



Gambar 4.31 Hasil *Output Attacker* Mencari Versi dan Jenis Database

Tabel 4.45 Data *Log* Uji Ke-3 *Server* 1

```
192.168.100.3 - - [16/Jan/2016:05:46:49 +0700] "GET
/onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&sca
t=3&CAYE%3D2587%20AND%201%3D1%20UNION%20ALL%20SELECT%
201%2C2%2C3%2Ctable_name%20FROM%20information_schema.
tables%20WHERE%202%3E1--
%20..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1" 200 5122 "-"
"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"
```

Untuk *server* 2, pengujian ke-3 tidak bisa dilakukan oleh peretas dikarenakan sudah terdeteksi *IP Address blacklist*.

- Pemenggalan

  Pada tahap ini, data *log* akan dilakukan dilakukan pemenggalan menjadi sembilan bagian.

Tabel 4.46 Proses Pemenggalan Data *Log* Uji Ke-3 *Server* 1

Bagian	Data
[1]	192.168.100.3
[2]	-
[3]	-
[4]	[16/Jan/2016:05:46:49 +0700]
[5]	"GET /onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&scat=3&CAYE%3D2587%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2C2%2C3%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%20..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1"
[6]	200
[7]	5122
[8]	“-”
[9]	"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"

- Analisis metode s*ignature* dan a*nomaly*
  - Pencocokan

Bagian [5], [8] dan [9] diambil dan kemudian dilakukan analisa dengan menggunakan metode *signature*.

Tabel 4.47 Bagian Data Akses Data Uji Ke-3 *Server* 1

Bagian	**Data**
[5]	"GET /onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&scat=3&CAYE%3D2587%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2C2%2C3%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%20..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1"
[8]	"-"
[9]	"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"

Dengan mengirim data tersebut ke algoritma pengecekan pada tabel 3.8, dari pengecekan tersebut didapat data sebagai berikut :

Tabel 4.48 *Output* Pencocokan Data Uji Ke-3 *Server* 1 dengan IDS *Rule*

Kategori	***Rule***	**Data**	**Hasil**
TOOLS	^.+(acunetix|sql|sqlmap|sql-map).*$	[5], [8] dan [9]	***TRUE***
INJECTION	^.+(<!!27|sleep|select|delay|wait|union).*$	[5], [8] dan [9]	***TRUE***
XSS	^.+(<|script|--|!--).*$	[5], [8] dan [9]	***TRUE***
COMMAND	^.+(etc/passwd|windows/win.ini).*$	[5], [8] dan [9]	***FALSE***

- Rekam *Traffic*

  Bagian [4] diambil dan kemudian dilakukan analisa dengan menggunakan metode *anomaly*.

Tabel 4.49 Bagian Data Waktu Data Uji Ke-3

Server	**Bagian**	**Data**
1	[4]	[16/Jan/2016:05:46:49 +0700]

Dengan mengirim data tersebut ke algoritma *rule hit* pada tabel 3.9. Dari data *log* yang terjadi selama proses *scanning*, mendapati bawah pada waktu akses tersebut mendapati ada 1 *hit* yang masuk dalam waktu yang sama pada IP yang sama.

Tabel 4.50 *Output* Analisa *Traffic* Data Uji Ke-3 dengan IDS *Rule*

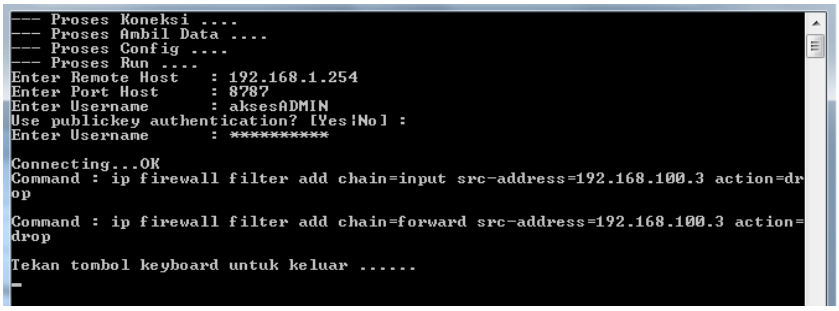
Server	**Tanggal**	**Jam**	***Qty Hit***	**Hasil**
1	16/Jan/2016	05:46:49	1	***FALSE***

- Hasil *Final*

  Dari proses pencocokan data bagian ke lima, delapan dan ke sembilan dari data *log*, didapat hasil bahwa akses yang masuk ke *server* dimana data-nya tergolong *intrusion* kategori "*TOOLS*", "*INJECTION*", "*XSS*". Kemudian aplikasi mengupdate jenis *intrusion* pada akses tersebut, dan melakukan *prevention* dengan mengambil bagian ke satu data *log* yaitu IP, untuk di blok.

  Tetapi tidak mendapati adanya anomali deteksi, dikarenakan atas pengecekan berdasarkan waktu akses, hit yang masuk masih dibawah ambang batas atas atau maksimum.

  Adapun pencegahan yang dilakukan dengan mengirimkan informasi ke *router* yaitu *command prevention*.



Gambar 4.32 Mengirim *Command Prevention* ke Router Uji Ke-3 *Server* 1



Gambar 4.33 Informasi *Alert* SMS Uji Ke-3 *Server* 1

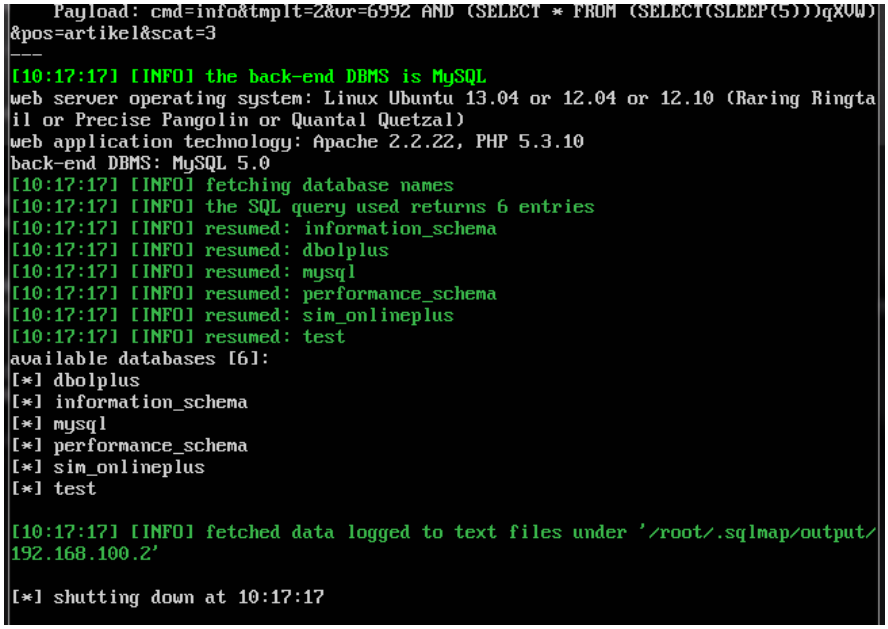
4. Pengujian Ketepatan Deteksi dan Pencegahan Ke-4

SQL injection : Mendapatkan informasi database. Setelah SQLMap menegaskan bahwa url rentan terhadap SQL *injection* dan dieksploitasi langkah berikutnya adalah untuk mengetahui nama-nama database yang ada pada *server*. Dengan menambahkan "--dbs" untuk mendapatkan daftar database.

Command Attacker :

```
root@ubuntu-devel:/opt/sqlmap# python sqlmap.py -u
"http://192.168.100.2/onlineplus/?cmd=info&tmplt=2&vr=6992&
pos=artikel&scat=3" --dbs
```

Hasil *Attacker* :



Gambar 4.34 Hasil *Output Attacker* Mencari Informasi *List* Database

Tabel 4.51 Data *Log* Uji Ke-4 *Server* 1

192.168.100.3 - - [16/Jan/2016:05:50:26 +0700] "GET /onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&scat=3&uRPB%3D8263%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2C2%2C3%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--

```
%20..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1" 200 5122 "-"
"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"
```

Untuk *server* 2, pengujian ke-4 tidak bisa dilakukan oleh peretas dikarenakan sudah terdeteksi *IP Address blacklist*.

- Pemenggalan

  Pada tahap ini, data *log* akan dilakukan dilakukan pemenggalan menjadi sembilan bagian.

Tabel 4.52 Proses Pemenggalan Data *Log* Uji Ke-4 *Server* 1

Bagian	Data
[1]	192.168.100.3
[2]	-
[3]	-
[4]	[16/Jan/2016:05:50:26 +0700]
[5]	"GET /onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&scat=3&uRPB%3D8263%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2C2%2C3%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%20..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1"
[6]	200
[7]	5122
[8]	“-”
[9]	"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"

- Analisis metode s*ignature* dan a*nomaly*
  - Pencocokan

    Bagian [5], [8] dan [9] diambil dan kemudian dilakukan analisa dengan menggunakan metode *signature*.

Tabel 4.53 Bagian Data Akses Data Uji Ke-4 *Server* 1

Bagian	Data
[5]	"GET /onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&scat=3&CAYE%3D2587%

	20AND%201%3D1%20UNION%20ALL%20SELECT%201%2C2%2C3%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%20..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1"
[8]	"-"
[9]	"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"

Dengan mengirim data tersebut ke algoritma pengecekan pada tabel 3.8, dari pengecekan tersebut didapat data sebagai berikut :

Tabel 4.54 *Output* Pencocokan Data Uji Ke-4 *Server* 1 dengan IDS *Rule*

Kategori	*Rule*	Data	Hasil
TOOLS	^.+(acunetix|sql|sqlmap|sql-map).*$	[5], [8] dan [9]	***TRUE***
INJECTION	^.+(<!!27|sleep|select|delay|wait|union).*$	[5], [8] dan [9]	***TRUE***
XSS	^.+(<|script|--|!--).*$	[5], [8] dan [9]	***TRUE***
COMMAND	^.+(etc/passwd|windows/win.ini).*$	[5], [8] dan [9]	***FALSE***

- Rekam *Traffic*

Bagian [4] diambil dan kemudian dilakukan analisa dengan menggunakan metode *anomaly*.

Tabel 4.55 Bagian Data Waktu Data Uji Ke-4

Server	Bagian	Data
1	[4]	[16/Jan/2016:05:50:26 +0700]

Dengan mengirim data tersebut ke algoritma *rule hit* pada tabel 3.9. Dari data *log* yang terjadi selama proses *scanning*, mendapati bawah pada waktu akses tersebut mendapati ada 2 *hit* yang masuk dalam waktu yang sama pada IP yang sama.

Tabel 4.56 *Output* Analisa *Traffic* Data Uji Ke-4 dengan IDS *Rule*

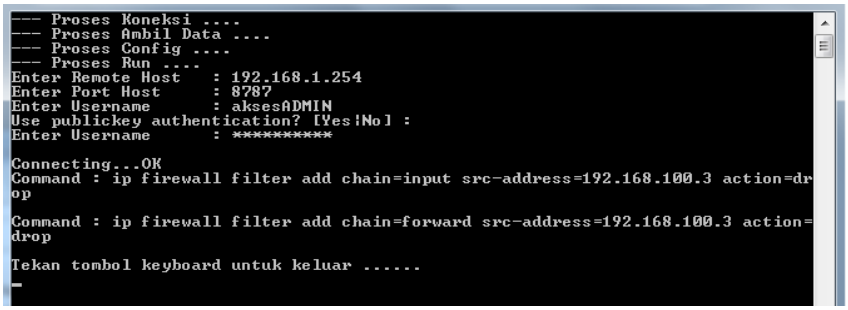
Server	**Tanggal**	**Jam**	***Qty Hit***	**Hasil**
1	16/Jan/2016	05:50:26	2	***FALSE***

- Hasil *Final*

Dari proses pencocokan data bagian ke lima, delapan dan ke sembilan dari data *log*, didapat hasil bahwa akses yang masuk ke *server* dimana data-nya tergolong *intrusion* kategori "*TOOLS*", "*INJECTION*", "*XSS*". Kemudian aplikasi mengupdate jenis *intrusion* pada akses tersebut, dan melakukan *prevention* dengan mengambil bagian ke satu data *log* yaitu IP, untuk di blok.

Tetapi tidak mendapati adanya anomali deteksi, dikarenakan atas pengecekan berdasarkan waktu akses, *hit* yang masuk masih dibawah ambang batas atas atau maksimum.

Adapun pencegahan yang dilakukan dengan mengirimkan informasi ke *router* yaitu *command prevention*.



Gambar 4.35 Mengirim *Command Prevention* ke Router Uji Ke-4 *Server* 1



Gambar 4.36 Informasi *Alert* SMS Uji Ke-4 *Server* 1

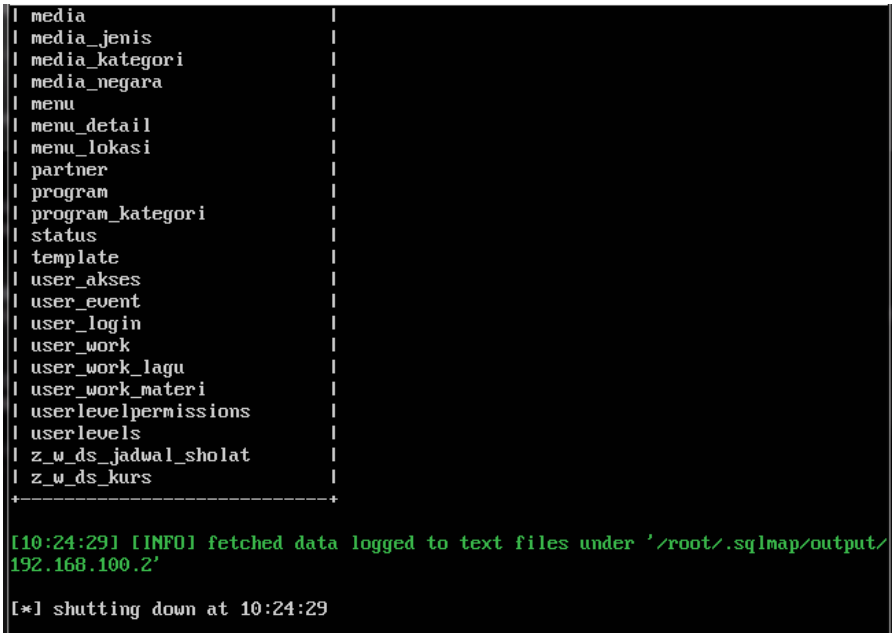
5. Pengujian Ketepatan Deteksi dan Pencegahan Ke-5

SQL injection : Mendapatkan *table-table* dari database. Sekarang waktunya untuk mencari tahu informasi tabel apa saja yang ada dalam suatu database tertentu. Pada *command* sebelum-nya database menarik di periksa adalah “dbolplus” atau “sim_onlineplus”.

Command Attacker :

```
root@ubuntu-devel:/opt/sqlmap# python sqlmap.py -u
"http://192.168.100.2/onlineplus/?cmd=info&tmplt=2&vr=6992&
pos=artikel&scat=3" --tables -D sim_onlineplus
```

Hasil *Attacker* :



Gambar 4.37 Hasil *Output Attacker* Mencari Informasi *List Table* Dari Sebuah Database

Tabel 4.57 Data *Log* Uji Ke-5 *Server* 1

```
192.168.100.3 - - [16/Jan/2016:05:57:38 +0700] "GET
/onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&sca
t=3&ezEX%3D6742%20AND%201%3D1%20UNION%20ALL%20SELECT%
201%2C2%2C3%2Ctable_name%20FROM%20information_schema.
tables%20WHERE%202%3E1--
```

```
%20..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1" 200 5122 "-"
"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"
```

Untuk *server* 2, pengujian ke-5 tidak bisa dilakukan oleh peretas dikarenakan sudah terdeteksi *IP Address blacklist*.

- Pemenggalan

  Pada tahap ini, data *log* akan dilakukan dilakukan pemenggalan menjadi sembilan bagian.

Tabel 4.58 Proses Pemenggalan Data *Log* Uji Ke-5 *Server* 1

Bagian	Data
[1]	192.168.100.3
[2]	-
[3]	-
[4]	[16/Jan/2016:05:57:38 +0700]
[5]	"GET /onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&scat=3&uRPB%3D8263%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2C2%2C3%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%20..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1"
[6]	200
[7]	5122
[8]	“-”
[9]	"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"

- Analisis metode s*ignature* dan a*nomaly*
  - Pencocokan

    Bagian [5], [8] dan [9] diambil dan kemudian dilakukan analisa dengan menggunakan metode *signature*.

Tabel 4.59 Bagian Data Akses Data Uji Ke-5 *Server* 1

Bagian	Data
[5]	"GET /onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&scat=3&CAYE%3D2587%

	20AND%201%3D1%20UNION%20ALL%20SELECT%201%2C2%2C3%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%20..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1"
[8]	"-"
[9]	"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"

Dengan mengirim data tersebut ke algoritma pengecekan pada tabel 3.8, dari pengecekan tersebut didapat data sebagai berikut :

Tabel 4.60 *Output* Pencocokan Data Uji Ke-5 *Server* 1 dengan IDS *Rule*

Kategori	*Rule*	Data	Hasil
TOOLS	^.+(acunetix|sql|sqlmap|sql-map).*$	[5], [8] dan [9]	***TRUE***
INJECTION	^.+(<!!27|sleep|select|delay|wait|union).*$	[5], [8] dan [9]	***TRUE***
XSS	^.+(<|script|--|!--).*$	[5], [8] dan [9]	***TRUE***
COMMAND	^.+(etc/passwd|windows/win.ini).*$	[5], [8] dan [9]	***FALSE***

- Rekam *Traffic*

Bagian [4] diambil dan kemudian dilakukan analisa dengan menggunakan metode *anomaly*.

Tabel 4.61 Bagian Data Waktu Data Uji Ke-5

Server	Bagian	Data
1	[4]	[16/Jan/2016:05:57:38 +0700]

Dengan mengirim data tersebut ke algoritma *rule hit* pada tabel 3.9. Dari data *log* yang terjadi selama proses *scanning*, mendapati bawah pada waktu akses tersebut mendapati ada 2 *hit* yang masuk dalam waktu yang sama pada IP yang sama.

Tabel 4.62 *Output* Analisa *Traffic* Data Uji Ke-5 dengan IDS *Rule*

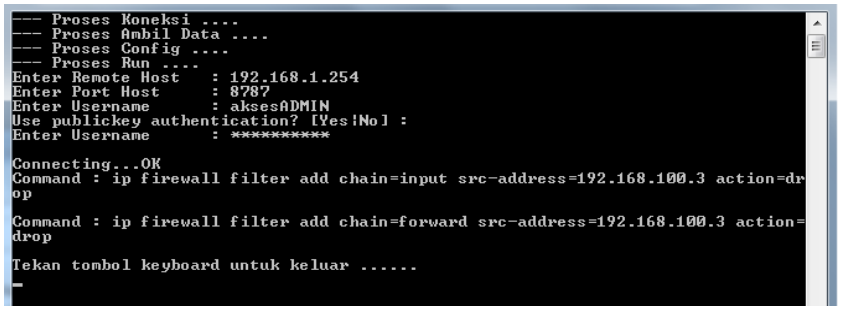
Server	**Tanggal**	**Jam**	***Qty Hit***	**Hasil**
1	16/Jan/2016	05:57:38	2	***FALSE***

- Hasil *Final*

Dari proses pencocokan data bagian ke lima, delapan dan ke sembilan dari data *log*, didapat hasil bahwa akses yang masuk ke *server* dimana data-nya tergolong *intrusion* kategori "*TOOLS*", "*INJECTION*", "*XSS*". Kemudian aplikasi mengupdate jenis *intrusion* pada akses tersebut, dan melakukan *prevention* dengan mengambil bagian ke satu data *log* yaitu IP, untuk di blok.

Tetapi tidak mendapati adanya anomali deteksi, dikarenakan atas pengecekan berdasarkan waktu akses, *hit* yang masuk masih dibawah ambang batas atas atau maksimum.

Adapun pencegahan yang dilakukan dengan mengirimkan informasi ke *router* yaitu *command prevention*.



Gambar 4.38 Mengirim *Command Prevention* ke Router Uji Ke-5 *Server* 1



Gambar 4.39 Informasi *Alert* SMS Uji Ke-5 *Server* 1

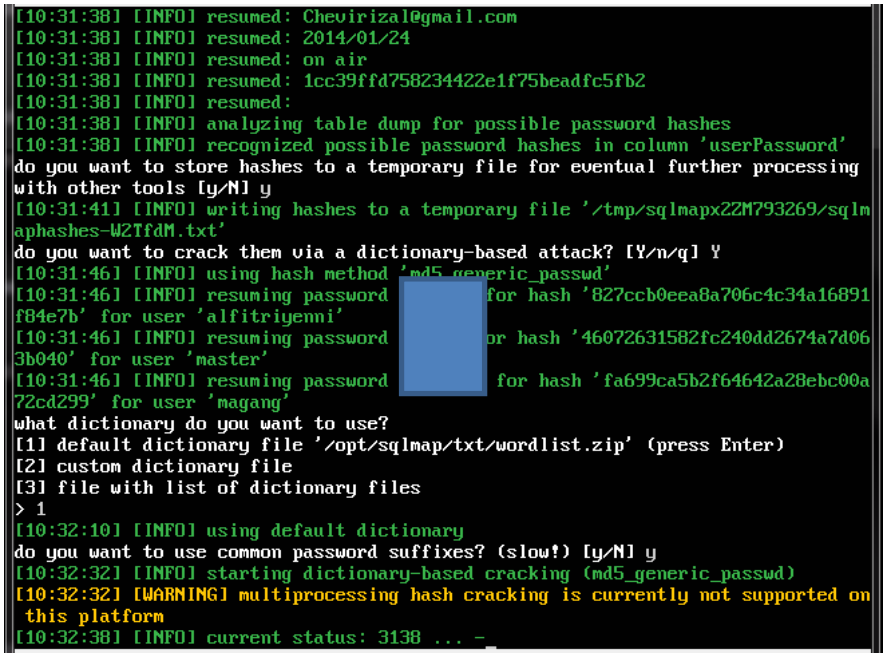
6. Pengujian Ketepatan Deteksi dan Pencegahan Ke-6

SQL injection : Mendapatkan record dari *table* pada suatu database. Sekarang sampai pada bagian yang paling penting, penggalian data dari tabel.

Command Attacker :

```
root@ubuntu-devel:/opt/sqlmap# python sqlmap.py -u
"http://192.168.100.2/onlineplus/?cmd=info&tmplt=2&vr=6992&
pos=artikel&scat=3" --dump -D sim_onlineplus -T user
```

Hasil *Attacker* :



Gambar 4.40 Hasil *Output Attacker* Mencari Mendapatkan *Record* Dari *Table* Pada Suatu Database

Tabel 4.63 Data *Log* Uji Ke-6 *Server* 1

```
192.168.100.3 - - [16/Jan/2016:06:03:39 +0700] "GET
/onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&scat
=3&abem%3D4807%20AND%201%3D1%20UNION%20ALL%20SELECT%20
1%2C2%2C3%2Ctable_name%20FROM%20information_schema.tab
les%20WHERE%202%3E1--%20..%2F..%2F..%2Fetc%2Fpasswd
HTTP/1.1" 200 5122 "-" "sqlmap/1.0-dev-80d3ff6
(http://sqlmap.org)"
```

Untuk *server* 2, pengujian ke-6 tidak bisa dilakukan oleh peretas dikarenakan sudah terdeteksi *IP Address blacklist*.

- Pemenggalan

  Pada tahap ini, data *log* akan dilakukan dilakukan pemenggalan menjadi sembilan bagian.

Tabel 4.64 Proses Pemenggalan Data *Log* Uji Ke-6 *Server* 1

Bagian	Data
[1]	192.168.100.3
[2]	-
[3]	-
[4]	[16/Jan/2016:06:03:39 +0700]
[5]	"GET /onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&scat=3&abem%3D4807%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2C2%2C3%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%20..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1"
[6]	200
[7]	5122
[8]	“-”
[9]	"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"

- Analisis metode s*ignature* dan a*nomaly*
  - Pencocokan

    Bagian [5], [8] dan [9] diambil dan kemudian dilakukan analisa dengan menggunakan metode *signature*.

Tabel 4.65 Bagian Data Akses Data Uji Ke-6 *Server* 1

Bagian	Data
[5]	"GET /onlineplus/?cmd=info&tmplt=2&vr=6992&pos=artikel&scat=3&CAYE%3D2587%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2C2%2C3%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--

	%20..%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1"
[8]	"-"
[9]	"sqlmap/1.0-dev-80d3ff6 (http://sqlmap.org)"

Dengan mengirim data tersebut ke algoritma pengecekan pada tabel 3.8, dari pengecekan tersebut didapat data sebagai berikut :

Tabel 4.66 *Output* Pencocokan Data Uji Ke-6 *Server* 1 dengan IDS *Rule*

Kategori	*Rule*	Data	Hasil
TOOLS	^.+(acunetix|sql|sqlmap|sql-map).*$	[5], [8] dan [9]	***TRUE***
INJECTION	^.+(<!!27|sleep|select|delay|wait|union).*$	[5], [8] dan [9]	***TRUE***
XSS	^.+(<|script|--|!--).*$	[5], [8] dan [9]	***TRUE***
COMMAND	^.+(etc/passwd|windows/win.ini).*$	[5], [8] dan [9]	***FALSE***

- Rekam *Traffic*

Bagian [4] diambil dan kemudian dilakukan analisa dengan menggunakan metode *anomaly*.

Tabel 4.67 Bagian Data Waktu Data Uji Ke-6

Server	Bagian	Data
1	[4]	[16/Jan/2016:06:03:39 +0700]

Dengan mengirim data tersebut ke algoritma *rule hit* pada tabel 3.9. Dari data *log* yang terjadi selama proses *scanning*, mendapati bawah pada waktu akses tersebut mendapati ada

2 hit yang masuk dalam waktu yang sama pada IP yang sama.

Tabel 4.68 *Output* Analisa *Traffic* Data Uji Ke-6 dengan IDS *Rule*

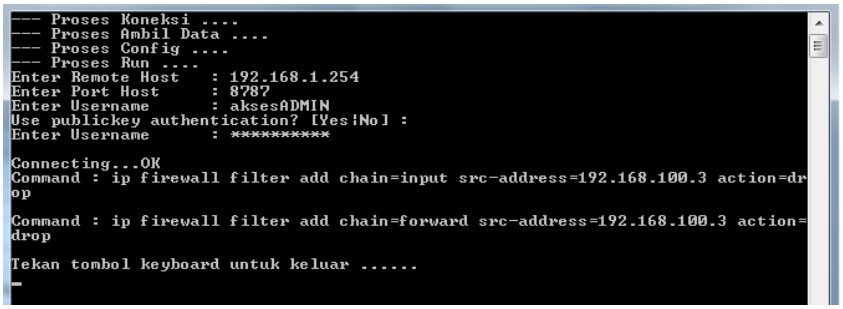
Server	Tanggal	Jam	*Qty Hit*	Hasil
1	16/Jan/2016	06:03:39	2	***FALSE***

- Hasil *Final*

Dari proses pencocokan data bagian ke lima, delapan dan ke sembilan dari data *log*, didapat hasil bahwa akses yang masuk ke *server* dimana data-nya tergolong *intrusion* kategori "*TOOLS*", "*INJECTION*", "*XSS*". Kemudian aplikasi mengupdate jenis *intrusion* pada akses tersebut, dan melakukan *prevention* dengan mengambil bagian ke satu data *log* yaitu IP, untuk di blok.

Tetapi tidak mendapati adanya anomali deteksi, dikarenakan atas pengecekan berdasarkan waktu akses, *hit* yang masuk masih dibawah ambang batas atas atau maksimum.

Adapun pencegahan yang dilakukan dengan mengirimkan informasi ke *router* yaitu *command prevention*.



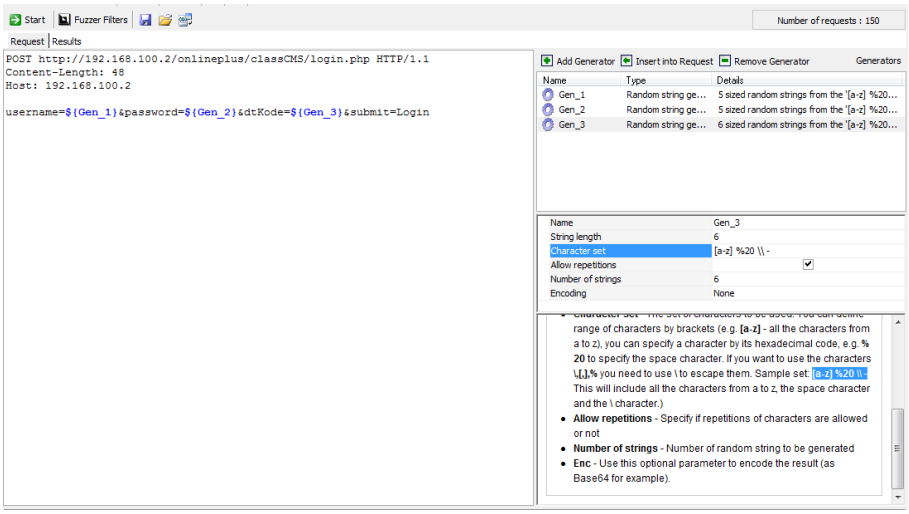
Gambar 4.41 Mengirim *Command Prevention* ke Router Uji Ke-6 *Server* 1



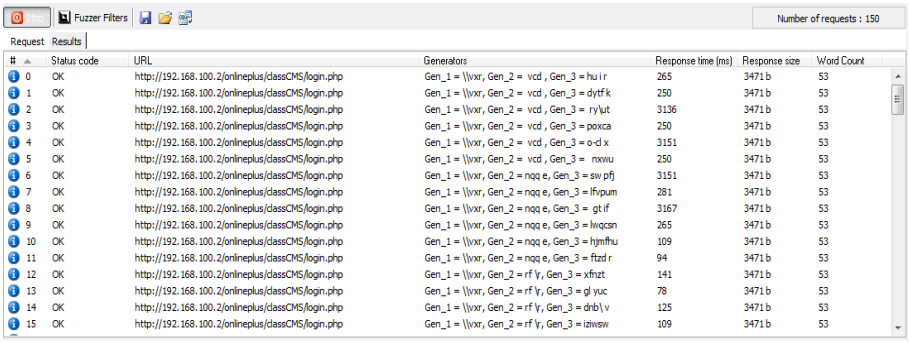
Gambar 4.42 Informasi *Alert* SMS Uji Ke-6 *Server* 1

7. Pengujian Ketepatan Deteksi dan Pencegahan Ke-7

Brute Force : *Login* ke *web server*, dengan melakukan percobaan pola *input*-an yang di eksekusi oleh mesin.



Gambar 4.43 Melakukan Konfigurasi *Pattern Input*-an *Login*



Gambar 4.44 Mengirim *Pattern Input*-an *Login*

Tabel 4.69 Data *Log* Uji Ke-7 *Server* 1

192.168.100.3 - - [16/Jan/2016:08:20:49 +0700] "POST /onlineplus/classCMS/login.php HTTP/1.1" 200 3932 "-" "-"
. . .
. . .
. . .
. . .
192.168.100.3 - - [16/Jan/2016:08:20:49 +0700] "POST /onlineplus/classCMS/login.php HTTP/1.1" 200 3932 "-" "-"

Tabel 4.70 Data *Log* Uji Ke-7 *Server* 2

192.168.200.4 - - [01/Feb/2016:04:05:37 +0700] "POST /onlineplus/classCMS/login.php HTTP/1.1" 200 582 "-" "-"
…
…
…
192.168.200.4 - - [01/Feb/2016:04:05:37 +0700] "POST /onlineplus/classCMS/login.php HTTP/1.1" 200 582 "-" "-"

- Pemenggalan

  Pada tahap ini, data *log* akan dilakukan dilakukan pemenggalan menjadi sembilan bagian.

Tabel 4.71 Proses Pemenggalan Data *Log* Uji Ke-7 *Server* 1

Bagian	Data
[1]	192.168.100.3
[2]	-
[3]	-
[4]	[16/Jan/2016:08:20:49 +0700]
[5]	"POST /onlineplus/classCMS/login.php HTTP/1.1"
[6]	200
[7]	3932
[8]	“-”
[9]	“-”

Tabel 4.72 Proses Pemenggalan Data *Log* Uji Ke-7 *Server* 2

Bagian	Data
[1]	192.168.200.4
[2]	-
[3]	-
[4]	[01/Feb/2016:04:05:37 +0700]
[5]	"POST /onlineplus/classCMS/login.php HTTP/1.1"
[6]	200
[7]	582
[8]	“-”
[9]	“-”

- Analisis metode s*ignature* dan a*nomaly*
  - Pencocokan

    Bagian [5], [8] dan [9] diambil dan kemudian dilakukan analisa dengan menggunakan metode *signature*.

Tabel 4.73 Bagian Data Akses Data Uji Ke-7 *Server* 1 dan *Server* 2

Bagian	**Data**
[5]	"POST /onlineplus/classCMS/login.php HTTP/1.1"
[8]	"-"
[9]	"-"

Dengan mengirim data tersebut ke algoritma pengecekan pada tabel 3.8, dari pengecekan tersebut didapat data sebagai berikut :

Tabel 4.74 *Output* Pencocokan Data Uji Ke-7 *Server* 1 dan *Server* 2 dengan IDS *Rule*

Kategori	***Rule***	**Data**	**Hasil**
TOOLS	^.+(acunetix|sql|sqlmap|sql-map).*$	[5], [8] dan [9]	***FALSE***
INJECTION	^.+(<!!27|sleep|select|delay|wait|union).*$	[5], [8] dan [9]	***FALSE***
XSS	^.+(<|script|--|!--).*$	[5], [8] dan [9]	***FALSE***
COMMAND	^.+(etc/passwd|windows/win.ini).*$	[5], [8] dan [9]	***FALSE***

- Rekam *Traffic*

  Bagian [4] diambil dan kemudian dilakukan analisa dengan menggunakan metode *anomaly*.

Tabel 4.75 Bagian Data Waktu Data Uji Ke-7

Server	**Bagian**	**Data**
1	[4]	[16/Jan/2016:08:20:49 +0700]
2	[4]	[01/Feb/2016:04:05:37 +0700]

Dengan mengirim data tersebut ke algoritma *rule hit* pada tabel 3.9. Dari data log yang terjadi selama proses scanning, mendapati bawah pada waktu akses tersebut mendapati ada 40 *hit* pada *server* 1 dan 30 *hit* pada *server* 2 yang masuk dalam waktu yang sama pada IP yang sama.

Tabel 4.76 Bagian Data Waktu Data Uji Ke-7

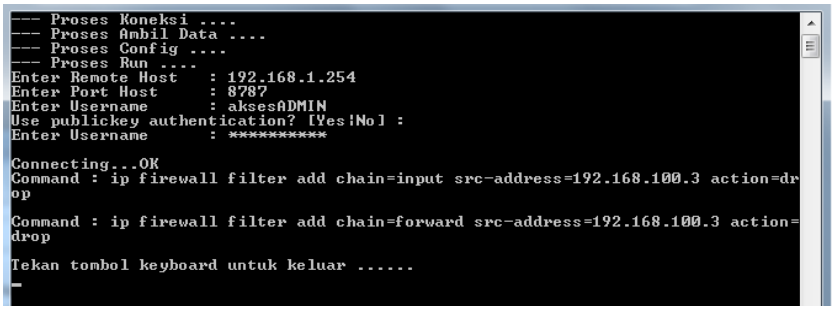
Server	**Tanggal**	**Jam**	***Qty Hit***	**Hasil**
1	16/Jan/2016	08:20:49	40	***TRUE***
2	01/Feb/2016	04:05:37	30	***TRUE***

- Hasil *Final*

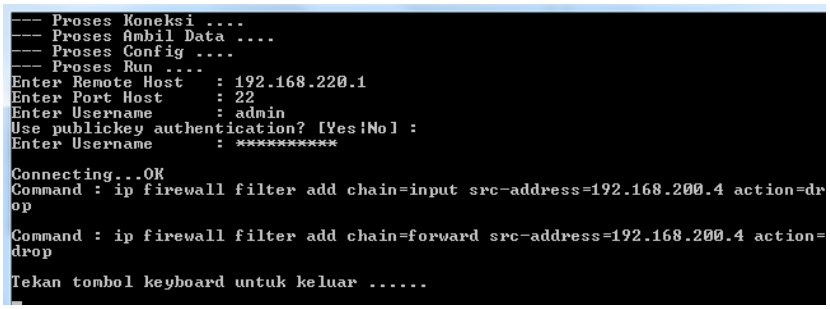
  Dari proses pencocokan data bagian ke lima, delapan dan ke sembilan dari data *log*, didapat hasil bahwa akses yang masuk ke *server* dimana data-nya tidak tergolong *intrusion*.

  Tetapi terdeteksi adanya anomali, dikarenakan atas pengecekan berdasarkan waktu akses, hit yang masuk berada diatas ambang batas atas atau maksimum.

  Adapun pencegahan yang dilakukan dengan mengirimkan informasi ke *router* yaitu *command prevention*.



Gambar 4.45 Mengirim *Command Prevention* ke Router Uji Ke-7 Server 1



Gambar 4.46 Mengirim *Command Prevention* ke Router Uji Ke-7 Server 2



Gambar 4.47 Informasi *Alert* SMS Uji Ke-7 *Server* 1



Gambar 4.48 Informasi *Alert* SMS Uji Ke-7 *Server* 2

8. Pengujian Ketepatan Deteksi dan Pencegahan Ke-8

XSS : Memasukkan tag html atau bahasa *client* pada *web server*

Tabel 4.77 Data *Log* Uji Ke-8 *Server* 1

```
192.168.3.201 - - [01/Sep/2015:03:52:29 +0700] "GET
/tes.php HTTP/1.1" 200 9247
"<!!27+response.write(9952078*9286509)+<!!27"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1;
WOW64; Trident/5.0)"
```

Tabel 4.78 Data *Log* Uji Ke-8 *Server* 2

```
192.168.200.2 - - [01/Feb/2016:02:38:20 +0700] "GET
/inexistent_file_name.inexistent0123450987.cfm
HTTP/1.1" 404 526 "-" "<script>alert(12345)</script>"
```

- Pemenggalan

Pada tahap ini, data *log* akan dilakukan dilakukan pemenggalan menjadi sembilan bagian.

Tabel 4.79 Proses Pemenggalan Data *Log* Uji Ke-8 *Server* 1

Bagian	Data
[1]	192.168.3.201
[2]	-
[3]	-
[4]	[01/Sep/2015:03:52:29 +0700]
[5]	"GET /tes.php HTTP/1.1"
[6]	200
[7]	9247
[8]	"<!!27+response.write(9952078*9286509)+<!!27"
[9]	"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"

Tabel 4.80 Proses Pemenggalan Data *Log* Uji Ke-8 *Server* 2

Bagian	Data
[1]	192.168.200.2
[2]	-
[3]	-
[4]	[01/Feb/2016:02:38:20 +0700]
[5]	"GET /inexistent_file_name.inexistent0123450987.cfm HTTP/1.1"

[6]	404
[7]	526
[8]	“-“
[9]	"<script>alert(12345)</script>"

- Analisis metode s*ignature* dan a*nomaly*
  - Pencocokan

    Bagian [5], [8] dan [9] diambil dan kemudian dilakukan analisa dengan menggunakan metode *signature*.

Tabel 4.81 Bagian Data Akses Data Uji Ke-8 *Server* 1

Bagian	**Data**
[5]	"GET /tes.php HTTP/1.1"
[8]	"<!!27+response.write(9952078*9286509)+<!!27"
[9]	"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"

Tabel 4.82 Bagian Data Akses Data Uji Ke-8 *Server* 2

Bagian	**Data**
[5]	"GET /inexistent_file_name.inexistent0123450987.cfm HTTP/1.1"
[8]	“-"
[9]	"<script>alert(12345)</script>"

Dengan mengirim data tersebut ke algoritma pengecekan pada tabel 3.8, dari pengecekan tersebut didapat data sebagai berikut :

Tabel 4.83 *Output* Pencocokan Data Uji Ke-8 *Server* 1 dan *Server* 2 dengan IDS *Rule*

Kategori	***Rule***	**Data**	**Hasil**
TOOLS	^.+(acunetix|sql|sqlmap|sql-map).*$	[5], [8], dan [9]	***FALSE***
INJECTION	^.+(<!!27|sleep|select|delay|wai	[5], [8], dan [9]	***TRUE***

	t|union).*$		
XSS	^.+(<|script|--|!--).*$	[5], [8], dan [9]	***TRUE***
COMMAND	^.+(etc/passwd|windows/win.ini).*$	[5], [8], dan [9]	***FALSE***

- Rekam *Traffic*

  Bagian [4] diambil dan kemudian dilakukan analisa dengan menggunakan metode *anomaly*.

Tabel 4.84 Bagian Data Waktu Data Uji Ke-8

Server	**Bagian**	**Data**
1	[4]	[01/Sep/2015:03:52:29 +0700]
2	[4]	[01/Feb/2016:02:38:20 +0700]

Dengan mengirim data tersebut ke algoritma *rule hit* pada tabel 3.9. Dari data *log* yang terjadi selama proses *scanning*, mendapati bawah pada waktu akses tersebut mendapati ada 3 *hit* pada *server* 1 dan 30 *hit* pada *server* 2 yang masuk dalam waktu yang sama pada IP yang sama.

Tabel 4.85 Bagian Data Waktu Data Uji Ke-8

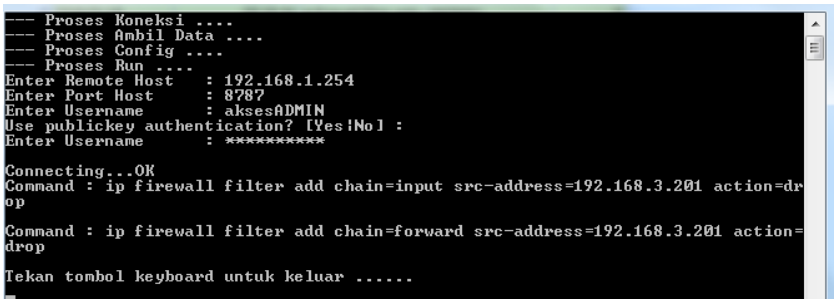
Server	**Tanggal**	**Jam**	***Qty Hit***	**Hasil**
1	01/Sep/2015	03:52:29	3	***FALSE***
2	01/Feb/2016	02:38:20	30	***TRUE***

- Hasil *Final*

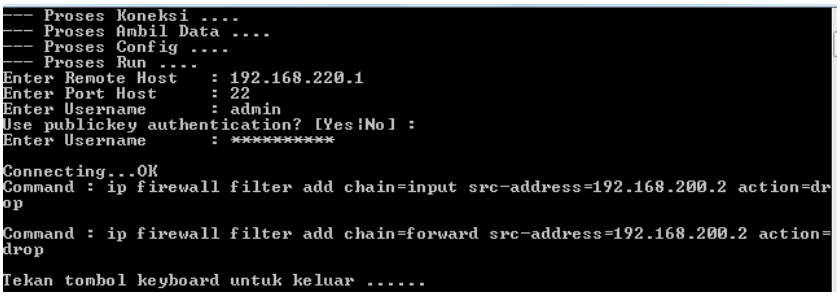
  Dari proses pencocokan data bagian ke lima, delapan, dan ke sembilan dari data *log*, didapat hasil bahwa akses yang masuk ke *server* dimana data-nya tergolong *intrusion* category “XSS”.

  Tetapi tidak terdeteksi adanya anomali, dikarenakan atas pengecekan berdasarkan waktu akses, *hit* yang masuk masih berada dibawah ambang batas atas atau maksimum.

Adapun pencegahan yang dilakukan dengan mengirimkan informasi ke *router* yaitu *command prevention.*



Gambar 4.49 Mengirim *Command Prevention* ke Router Uji Ke-8 *Server* 1



Gambar 4.50 Mengirim *Command Prevention* ke Router Uji Ke-8 *Server* 2



Gambar 4.51 Informasi *Alert* SMS Uji Ke-8 *Server* 1



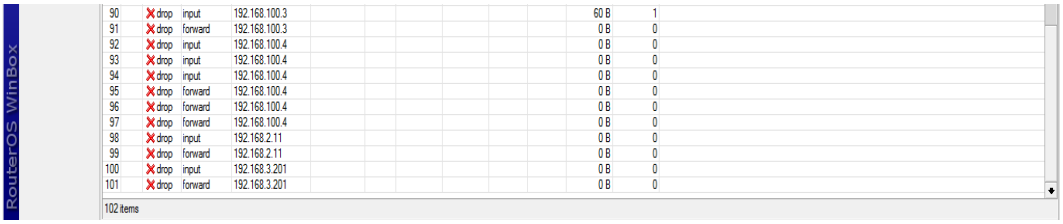
Gambar 4.52 Informasi *Alert* SMS Uji Ke-8 *Server* 2

Dari hasil pengujian yang telah dilakukan, didapat hasil sebagai berikut berikut :

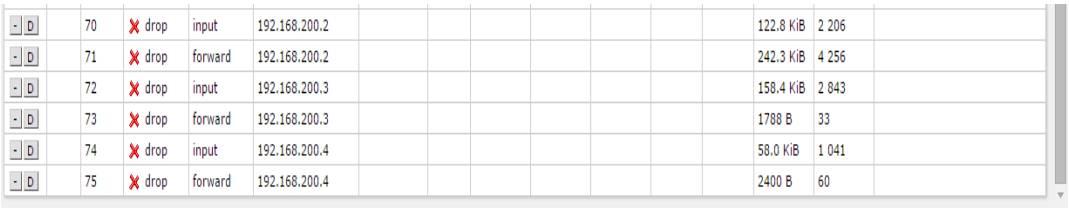
Tabel 4.74 Hasil Pengujian Deteksi Serangan

NO	Jenis Serangan	Status Deteksi Serangan *Attacker*		Status *Prevention*	
		***Server* 1**	***Server* 2**	***Server* 1**	***Server* 2**
1	*Scanning* : Menggunakan aplikasi Acunetix	T	T	E	E
2	*Scanning* : Menggunakan aplikasi SQLMap	T	T	E	E
3	*SQL injection* : Mendapatkan informasi database dari *server*	T	T	E	E
4	*SQL injection* : Mendapatkan informasi *database*	T	T	E	E
5	*SQL injection* : Mendapatkan informasi tabel dari *database*	T	T	E	E
6	*SQL injection* : Mendapatkan record-record dari dari table pada suatu *database*	T	T	E	E
7	*Brute Force* : Login ke webserver	T	T	E	E
8	*XSS :* Memasukkan tag html pada *web server*	T	T	E	E
Keterangan : T : Terdeteksi TD : Tidak Terdeteksi E : Tereksekusi TE : Tidak Tereksekusi					

Dan berikut *rule* tambahan yang terekam di router dari pengiriman perintah untuk pemblokiran akses pada IP tertentu.



Gambar 4.53 Daftar *List IP* Terblokir Pada Router 1



- D		70	drop	input	192.168.200.2							122.8 KiB	2 206
- D		71	drop	forward	192.168.200.2							242.3 KiB	4 256
- D		72	drop	input	192.168.200.3							158.4 KiB	2 843
- D		73	drop	forward	192.168.200.3							1788 B	33
- D		74	drop	input	192.168.200.4							58.0 KiB	1 041
- D		75	drop	forward	192.168.200.4							2400 B	60

Gambar 4.54 Daftar *List IP* Terblokir Pada Router 3