Model Pemantauan Keamanan Jaringan Melalui Aplikasi Telegram Dengan *Snort*

http://dx.doi.org/10.28932/jutisi.v7i1.4088

Riwayat Artikel
Received: 15 Oktober 2021 | *Final Revision*: 10 November 2021 | *Accepted*: 15 November 2021

Natanael Christianto ✉#1, Wiwin Sulistyo*2

Departemen Teknik Informatika, Universitas Kristen Satya Wacana
JL. Diponegoro no. 52-60, Kota Salatiga
1natanaelchristianto@gmail.com

**Departemen Teknik Informatika, Universitas Kristen Satya Wacana*
JL. Diponegoro no. 52-60, Kota Salatiga
2wiwinsulistyo@uksw.edu

Abstract — **Network security is the main of the development of today's technology. The ease in accessing the internet also requires protection on users is required. The ease of accessing the internet by people can also cause the occurrence of cyber crime. Cyber crime can be done by all internet users, without exception, to earn a profit. Security monitoring system server through the app messenger Telegram can help administrators in the work because always be on standby in front of the server computer. Notice of Snort as IDS via Telegram also quicked and can be accepted anywhere. In taking action when the server something happened not too late. Target cyber crime also can attack anyone without exception. A system should be a strength, with the protection of a secure network that will be difficult to hack by hackers. The server is the main target in the conduct of cyber crime. The use of the server must maintain to secure all the data is not misused by persons who are not responsible. Each server is a system that should be an administrator as a guard on duty watching and taking action when something happens on the server. To monitor a server, an administrator should always standby in front of the server computer so as not to late take action when the server is about to happen something.**

Keywords— **IDS; Network; Security; Snort; Telegram.**

I. PENDAHULUAN

Perkembangan internet di era digital yang dapat dapat memberikan kemudahan dalam mengaksesnya, tetapi di sisi lain juga dapat menimbulkan kejahatan dalam penggunaan internet. Di Indonesia sudah terdapat payung hukum yang mengatur tentang permasalahan serangan siber yaitu UU ITE yang dapat memberikan kepastian hukum untuk masyarakat dan dapat mencegah terjadinya kejahatan siber [1]. Dengan adanya kejahatan *cyber* yang dapat menyerang siapapun tanpa terkecuali, perlu adanya pengetahuan dari semua sebuah pemberitahuan yang dapat diterima untuk mengetahui bahwa adanya serangan yang terjadi pada sebuah server jaringan. Untuk melakukan pemantauan adanya serangan yang datang diperlukan adanya administrator dan ini mengharuskan seseorang untuk terus berjaga di depan layar komputer. Untuk mengefisienkan kinerja maka dibutuhkan sebuah sistem yang dapat memberitahu adanya serangan kepada user. Terhubungnya jaringan dengan internet akan memperbesar kemungkinan terjadinya gangguan terhadap sistem jaringan [2].

IDS *(Intrusion Detection System)* adalah metode yang digunakan untuk mendeteksi serangan. Dengan adanya metode tersebut dapat membantu untuk mengatasi serangan yang dari yang masuk. Secara penggunaan, IDS hanya berfungsi untuk mendeteksi dan tidak mengambil keputusan dalam menghadapi serangan. IDS juga mampu menggolongkan ancaman dari luar maupun dalam organisasi sehingga dapat membantu dalam pembuatan keputusan dalam alokasi sumber daya keamanan jaringan [3]. Penggunaan IDS dalam suatu jaringan komputer adalah suatu kelebihan karena dapat dipantau hanya dengan sebuah mesin atau komputer yang bertindak sebagai sensor di dalam jaringan. Dalam penelitian tersebut juga dapat dilihat jika terdapat suatu masalah pada jaringan, maka segera dapat diketahui langsung oleh IDS yaitu *Snort* [4]. Pemantauan mendasar perlu dilakukan untuk tetap menjaga keamanan data atau jaringan. Salah satu cara untuk membangun sistem keamanan jaringan dengan sederhana dan tidak memerlukan biaya yang mahal adalah dengan menggunakan *Snort* [4]. *Snort* tidak bisa menindaklanjuti serangan yang terdeteksi atau penyalahgunaan jaringan karena sifatnya hanya mendeteksi [5]. Cara kerja *Snort* hampir sama dengan TcpDump, tetapi *Snort* hanya berfokus pada sniffing paket yang aman [6]. Serangan dapat dan tidak terdeteksi oleh *Snort* tergantung dari ada tidaknya rule yang sesuai [7].

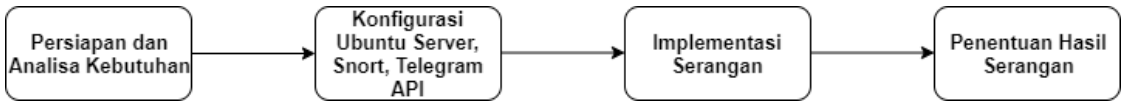
Dalam mengamankan jaringan komputer, metode IDS dapat mengoptimalkan tingkat keamanan jaringan komputer untuk mendeteksi adanya serangan sehingga administrator dapat segera melakukan tindakan pencegahan [8].

Telegram adalah sebuah aplikasi *messenger* yang sudah banyak digunakan untuk sarana berkomunikasi. Telegram juga menawarkan banyak fitur yang dapat digunakan bagi pengguna. Aplikasi Telegram memiliki cara penggunaan yang mudah dan tampilan yang *user friendly* untuk semua orang [8]. Selain menyediakan aplikasi Telegram juga menyediakan API bagi pengguna untuk membuat bot yang dapat digunakan dan dikembangkan untuk sistem informasi. Telegram dapat digunakan untuk melakukan kegiatan pemantauan jaringan sebagai penerima pemberitahuan jika terjadi serangan dari luar. Salah satu penggunaan Telegram untuk kegiatan pemantauan ini dengan menerima pesan dari IDS yang langsung terkirim menuju ke akun pengguna sehingga dapat mengetahui serangan yang terjadi walaupun sedang tidak didepan komputer server. Bot API yang terus berkembang sehingga dapat membuat bot yang dinamis dan dapat merespon pesan dari administrator jaringan [9, 10]. Implementasi bot mulai banyak digunakan karena mempunyai keunggulan yaitu dapat menyediakan data ke pengguna yang tidak terbatas oleh waktu dan dapat dikembangkan oleh siapa saja [8, 11].

Pada penelitian ini menghasilkan sistem pemantauan sebuah server yang lebih fleksibel karena dapat dipantau dari mana saja, sehingga seorang administrator tidak perlu selalu didepan komputer server untuk mengawasi server. Penggunaan model pemantauan seperti ini dapat mengefisienkan dari aspek waktu dan tenaga bagi seorang administrator server. Pengiriman pemberitahuan yang cepat dari server menuju Telegram juga akan membantu administrator dalam melakukan tindakan ketika server sedang terjadi sesuatu.

II. METODE PENELITIAN

Dalam proses penelitian ini terdapat Langkah kerja yang dilakukan. Seperti pada Gambar 1 Tahapan penelitian.



Gambar 1. Tahapan Penelitian

A. Persiapan dan Analisis Kebutuhan Penelitian

Persiapan adalah tahapan yang pertama dilakukan dimana untuk mempersiapkan kebutuhan yang diperlukan untuk melakukan penelitian. Perangkat lunak yang digunakan untuk melakukan penelitian ini dapat dilihat pada Tabel 1 dibawah ini:

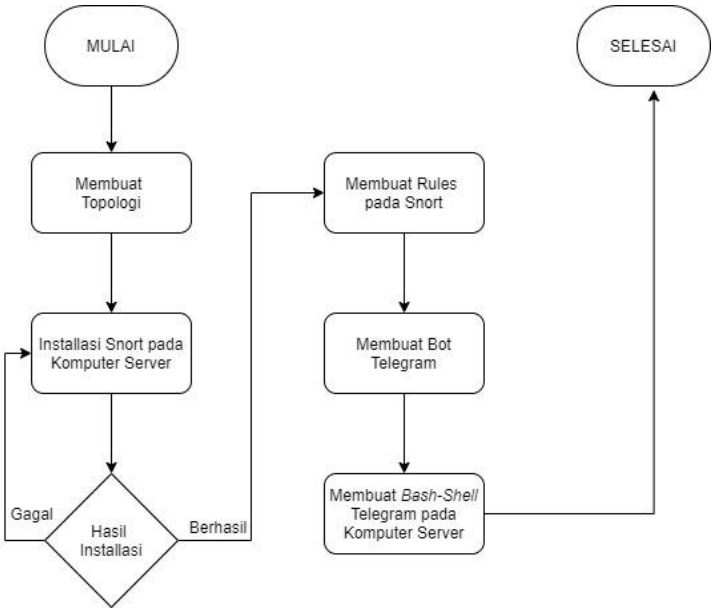
TABEL 1
PERANGKAT LUNAK YANG DIGUNAKAN

No	Spesifikasi	Fungsi
1	Ubuntu Server 20.04 LTS	Sistem operasi pada komputer server
2	*Snort*	*Software Intrusion Detection System* (IDS)
3	Windows 10	Sistem operasi pada komputer penyerang
4	Telegram API	Sebagai penerima pemberitahuan serangan dari server
5	VM Ware Workstation Pro	Sebagai *virtual machine* untuk menjalan kan sistem operasi ubuntu server

B. Konfigurasi Ubuntu Server, Snort, Telegram API

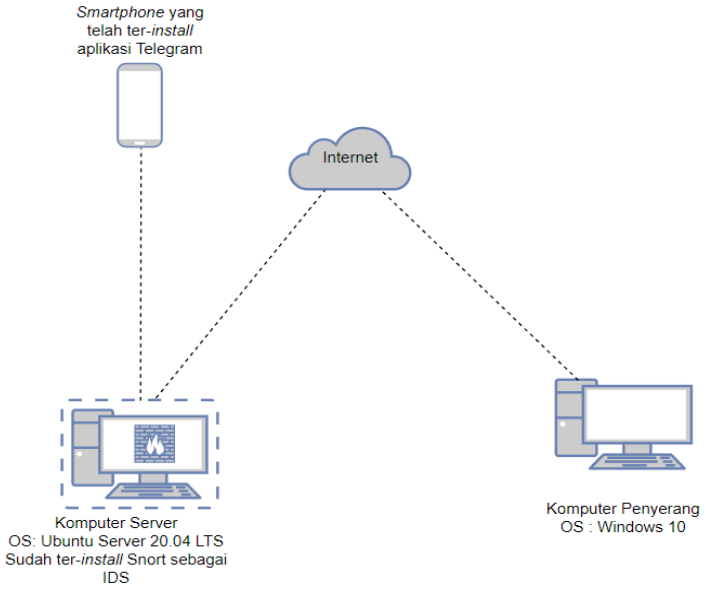
Setelah tahapan persiapan dan analisis kebutuhan penelitian selanjutnya adalah tahapan konfigurasi Ubuntu server, *Snort* dan Telegram. Dalam melakukan tahapan ini dilakukan sesuai diagram pada gambar 2 berikut ini:

p-ISSN : 2443-2210
e-ISSN : 2443-2229



Gambar 2. Diagram Langkah Konfigurasi

Pada gambar 2 diagram langkah konfigurasi dapat dilihat untuk langkah pertama dalam tahapan ini adalah membuat topologi, dalam membuat topologi ini berfungsi untuk mengetahui cara kerja komunikasi antar perangkat. Selanjutnya setelah pembuatan topologi adalah instalasi Snort pada komputer server. Dalam instalasi Snort mungkin dapat terjadi dua kondisi yaitu instalasi berhasil dan gagal, saat kondisi instalasi berhasil maka dapat lanjut ke langkah selanjutnya, tetapi jika kondisi gagal saat installasi harus mengulang proses instalasi Snort. Setelah proses instalasi berhasil langkah selanjutnya adalah membuat *rules* pada Snort, *rules* disini berfungsi untuk mendeteksi berbagai serangan yang masuk ke server. Langkah selanjutnya adalah pembuatan bot Telegram, dalam langkah ini pembuatan bot Telegram dilakukan di aplikasi Telegram dapat melalui aplikasi pada *smartphone* atau aplikasi pada komputer. Setelah pembuatan bot Telegram berhasil maka proses selanjutnya adalah membuat *bash-shell* Telegram pada komputer server, langkah ini bertujuan untuk menghubungkan server dan Telegram.



Gambar 3. Topologi Jaringan

Dari gambar 3 dapat dijelaskan bahwa komputer penyerang melakukan serangan kepada komputer server melalui jaringan internet, setelah komputer server menerima serangan Langkah selanjutnya adalah mengirimkan pemberitahuan serangan ke *smartphone user* melalui aplikasi Telegram. Setelah menentukan topologi jaringan yang digunakan Langkah selanjutnya adalah *install* sistem operasi ubuntu server pada aplikasi *VMware Workstation*. Kemudian setelah sistem operasi sudah di *install* dan berjalan normal kemudian *install Snort* pada ubuntu server.

```
$sudo apt-get install Snort
```

Kode Program 1. Kode untuk install *Snort* pada ubuntu server *Snort* pada ubuntu server

Pada Kode Program 1 terdapat kode program yang berfungsi untuk *install Snort* pada ubuntu server. Setelah *Snort* berhasil ter*install* pada ubuntu server Langkah selanjutnya adalah membuat *rules* pada *Snort* yang berfungsi untuk mendeteksi serangan yang terjadi pada server. Untuk menuliskan *rules* pada *Snort* terdapat pada direktori *Snort* lalu membuka direktori *rules*. Cara untuk masuk kedalam direktori *rules* dengan menuliskan kode program seperti pada kode program 2.

```
cd /etc/Snort/rules
```

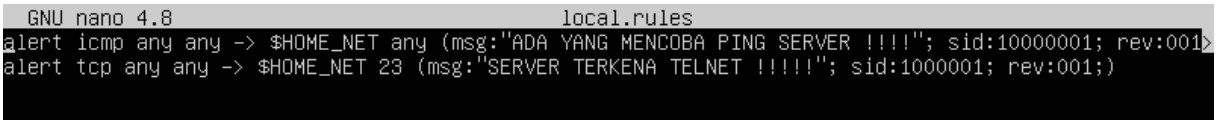
Kode Program 2. Kode untuk masuk ke direktori rules

Setelah dapat masuk ke dalam direktori rules untuk selanjutnya masuk ke dalam file local.rules.

```
nano local.rules
```

Kode Program 3. Kode untuk masuk ke files local rules

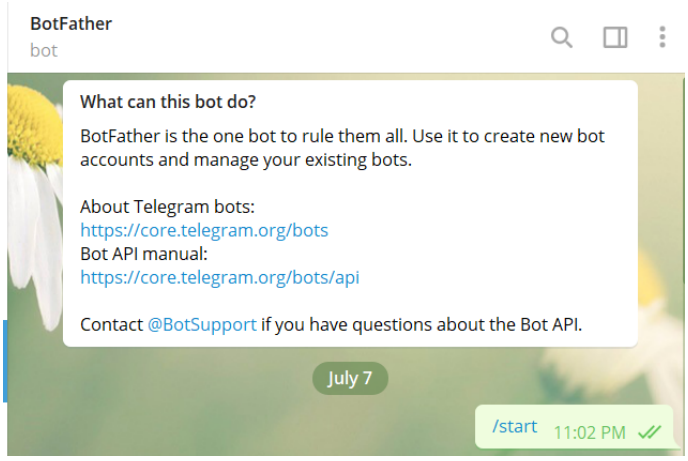
Pada Kode Program 3 terdapat kode program yang berfungsi untuk masuk ke dalam *file local.rules*, untuk pertama kali masih belum terdapat *rules* jadi harus memasukan *rules* secara manual sesuai kebutuhan. Untuk format penulisan *rules* pada *Snort* seperti pada Gambar 4. Untuk menyimpan perubahan isi *file* tekan Ctrl + X lalu tekan Y dan terakhir *Enter*.



Gambar 4. Isi file local rules

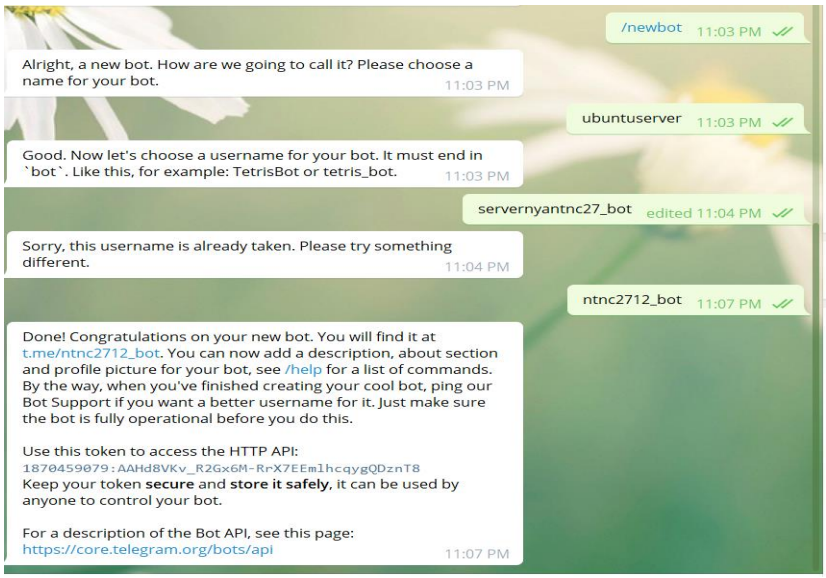
Gambar 4 menunjukkan isi dari *file local rules.* Konfigurasi *Snort* sudah dilakukan dan selanjutnya adalah konfigurasi aplikasi Telegram. Membuat *bot* pada aplikasi Telegram yang berfungsi untuk menerima informasi serangan. Pembuatan *bot* dapat dilakukan dengan membuka aplikasi Telegram lalu pada kolom pencarian ketik “BotFather” kemudian untuk memulai membuat *bot* pada kolom chat ketik “/start”.

p-ISSN : 2443-2210
e-ISSN : 2443-2229



Gambar 5. Memulai membuat bot

Gambar 5 menunjukkan tampilan awal pada bot Telegram ketika pertama kali membuat bot Kemudian ketikan "/newbot" pada kolom chat untuk pembuatan bot baru pada Telegram, selanjutnya memberikan nama *bot* "ubuntuserver" dan memberikan nama pengguna bot "ntnc2712_bot", dalam pemberian nama pengguna *bot*, menemui beberapa pilihan nama tidak bisa digunakan karena sudah ada yang memiliki.

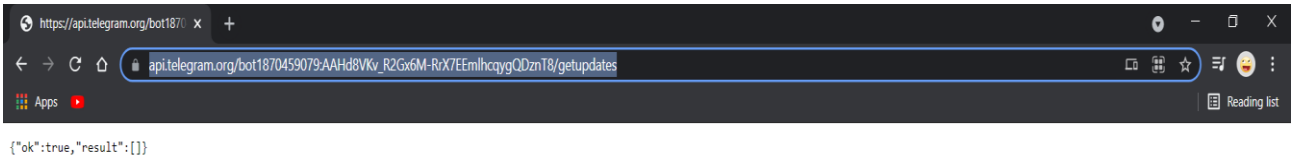


Gambar 6. Pemberian nama bot dan nama pengguna bot

Pada Gambar 6 menunjukkan tampilan pada bot Telegram saat memberi nama pada bot, juga terdapat token untuk mengakses Telegram API yaitu 1870459079:AAHd8VKv_R2Gx6M-RrX7EEmlhcqygQDznT8. Token ini nanti digunakan pada ubuntu server untuk menghubungkan server dengan Telegram. Untuk menguji token bot Telegram masukan kode program pada halaman *browser*. seperti pada kode program 4 berikut ini :

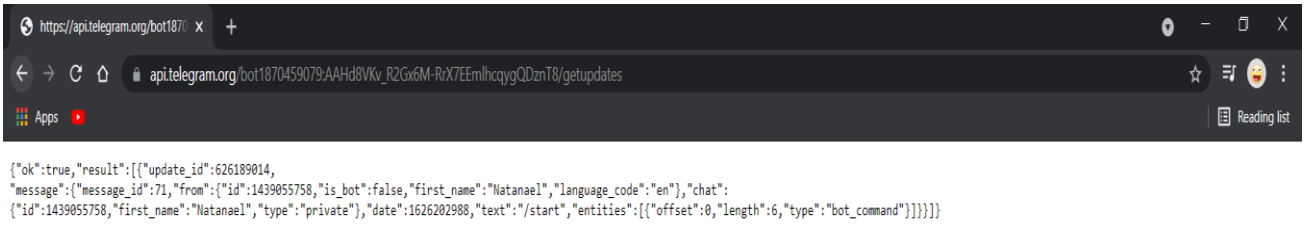
```
https://api.Telegram.org/bot1870459079:AAHd8VKv_R2Gx6M-RrX7EEmlhcqygQDznT8/getupdates
```

Kode Program 4. Untuk Menguji Token Telegram Bot



Gambar 7. Hasil Pengujian Token Telegram Bot

Pada Gambar 7 menunjukkan bahwa token bot Telegram yang sudah di uji berhasil digunakan. Selanjutnya untuk mendapatkan chat ID bot ketik pada kolom pencarian aplikasi Telegram lalu klik *button start. Reload link* token Telegram bot yang masih tersedia di halaman *browser*.



Gambar 8. Untuk Mengetahui Chat Id bot

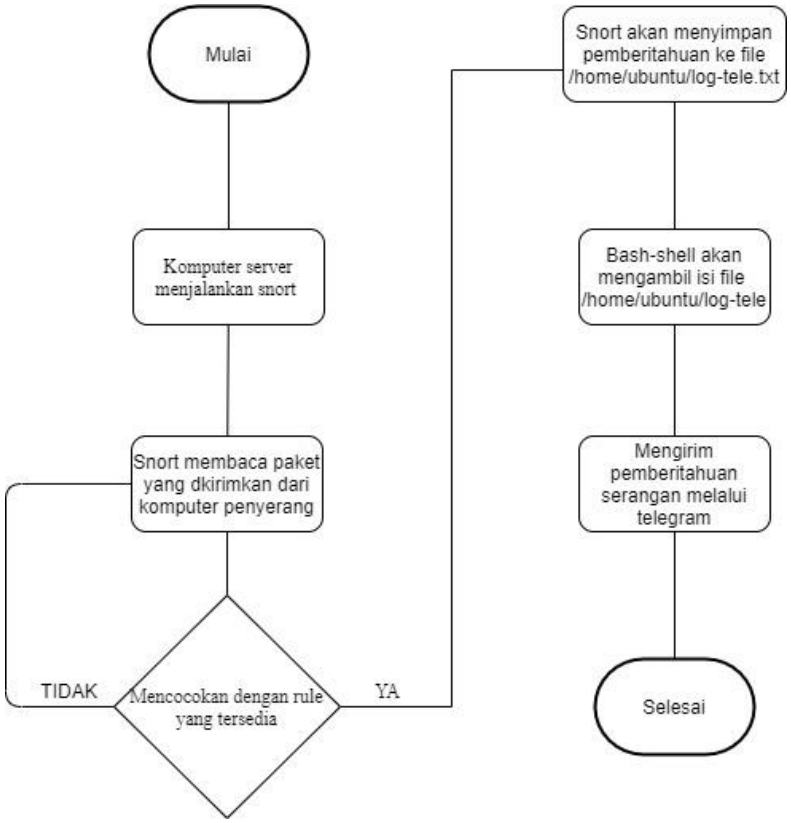
Pada Gambar 8 dapat dilihat bahwa chat id bot Telegram sudah tersedia yaitu 1439055758. Setelah bot Telegram sudah berhasill dibuat untuk selanjutnya adalah menghubungkan antara ubuntu server dengan Telegram dengan membuat *bash-shell* Telegram pada ubuntu server.

```
#!/bin/bash
#init
initCount=0
log=/home/ubuntu/log-tele.txt
#File
msg_caption=/tmp/telegram_msg_caption.txt
chat_id="1439055758"
token="1870459079:AAHd8VKv_R2Gx6M-RrX7EEmlhcqygQDznT8"
function sendAlert
{
curl -s -F chat_id=$chat_id -F text="$caption" https://api.telegram.org/bot$token/sendMessage #>
/dev/null 2&>1
}
while true
do
lastCount=$(wc -c $log | awk '{print $1}')
#echo "--------------------"
if(($(($lastCount)) > $initCount));
then
#DEBUG
#echo "MENGIRIM PEMBERITAHUAN"
msg=$(tail -n 2 $log)
echo -e "Halo mas Natan\n Terjadi ada nya Penyerangan pada Server loh!!!\n\nServer Time : $(date
+"%d %b %Y %T")\n\n"$msg > $msg_caption #set Caption / Pesan
caption=$(<$msg_caption)
sendAlert
echo "PEMBERITAHUAN TERKIRIM"
initCount=$lastCount
rm -f $msg_caption
sleep 1
fi
sleep 2
done
```

Kode Program 5. Isi perintah pada *bash-shell*

Pada kode program 5 dapat dijelaskan isi pesan pemberitahuan yang akan dikirimkan ke aplikasi Telegram diambil dari isi *file log-tele.txt* yang terdapat pada direktori ubuntu, kemudian mengirimkan pesan melalui *chat id* dan token yang sudah ditentukan. Setelah *bash-shell* berhasil dibuat langkah selanjutnya adalah pengujian serangan ke ubuntu server sekaligus menguji ubuntu server apakah berhasil dalam mengirimkan pesan ke aplikasi Telegram.

C. Implementasi Serangan



Gambar 9. Flowchart sistem kerja

Pada Gambar 9 dapat dijelaskan ketika komputer server akan menjalankan *Snort* dengan tambahan perintah untuk menjalankan *Snort* dengan menyimpan *log* pemberitahuan *Snort* di dalam file yang berada di */home/ubuntu/log-tele.txt*. seperti pada Kode Program berikut ini :

```
Snort -A console > /home/ubuntu/log-tele.txt -c /etc/Snort/Snort.conf -l /var/log/Snort/
```

Kode Program 6. Perintah untuk menjalankan Snort

Kode Program 6 menunjukkan kode program untuk menjalankan *Snort* pada ubuntu server. Selanjutnya ketika ada serangan masuk *Snort* akan membaca paket serangan yang masuk dan akan mencocokan dengan rule yang tersedia, jika ternyata paket serangan yang diterima cocok dengan rule yang tersedia maka *Snort* akan memberitahukan serangan yang terjadi sesuai dengan rule yang cocok dan akan menyimpan log pemberitahuan ke dalam file */home/ubuntu/log-tele.txt.*

```
 GNU nano 4.8                                                                log-tele.txt
07/15-23:41:55.421990  [**] [1:10000001:1] ADA YANG MENCOBA PING SERVER !!!! [**] [Priority: 0] {ICMP} 192.168.221.1 -> 192.168.221.128
07/15-23:41:56.430221  [**] [1:10000001:1] ADA YANG MENCOBA PING SERVER !!!! [**] [Priority: 0] {ICMP} 192.168.221.1 -> 192.168.221.128
07/15-23:41:57.441637  [**] [1:10000001:1] ADA YANG MENCOBA PING SERVER !!!! [**] [Priority: 0] {ICMP} 192.168.221.1 -> 192.168.221.128
07/15-23:41:58.448199  [**] [1:10000001:1] ADA YANG MENCOBA PING SERVER !!!! [**] [Priority: 0] {ICMP} 192.168.221.1 -> 192.168.221.128
```

Gambar 10. Hasil Penyimpanan Pemberitahuan di dalam file /home/ubuntu/log-tele.txt

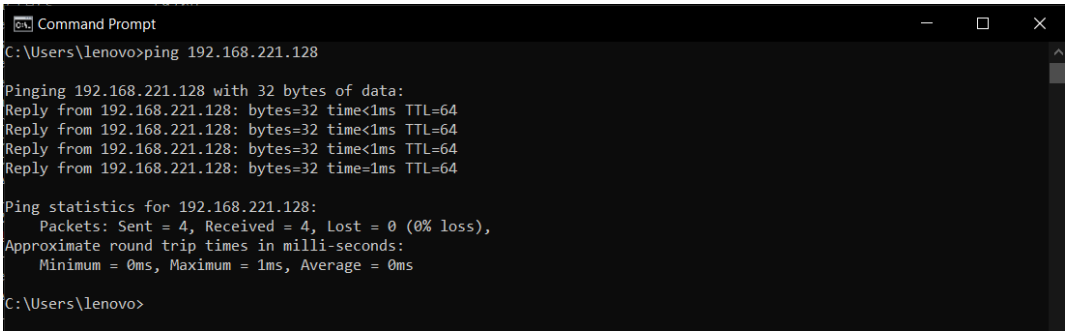
Gambar 10 menunjukan hasil penyimpanan *log* pemberitahuan. Setelah hasil pemberitahuan berhasil disimpan dalam *file log-tele.txt*, selanjutnya perintah pada program *bash-shell* Telegram akan mengambil isi dari log tersebut untuk selanjutnya dikirimkan melalui aplikasi Telegram.Proses dianggap selesai ketika pemberitahuan sudah berhasil terkirim ke aplikasi Telegram.

Dalam penelitian ini menggunakan aplikasi *Command Prompt* dalam windows 10 yang berperan sebagai komputer penyerang.Penyerangan yang dilakukan adalah mencoba mengirimkan *ping* melalui alamat ip komputer server dan mencoba mengakses komputer server dengan *telnet*.

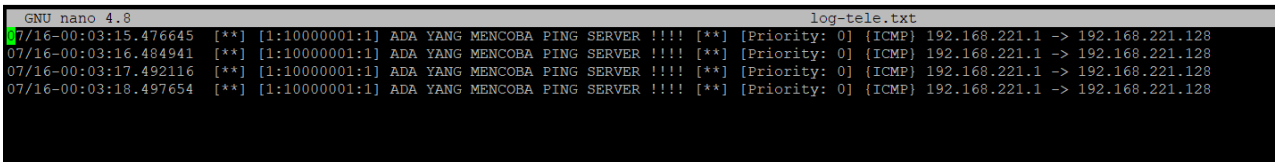
```
Ping 192.168.221.128
```

Kode Program 7. Perintah untuk mengirimkan Ping

Pada Kode Program 7 adalah perintah untuk mengirimkan ping dari komputer penyerang. Alamat ip komputer penyerang adalah 192.168.221.128.



Gambar 11. Hasil perintah *Ping IP Server* pada *Command Prompt*



Gambar 12. Hasil log pemberitahuan untuk ping

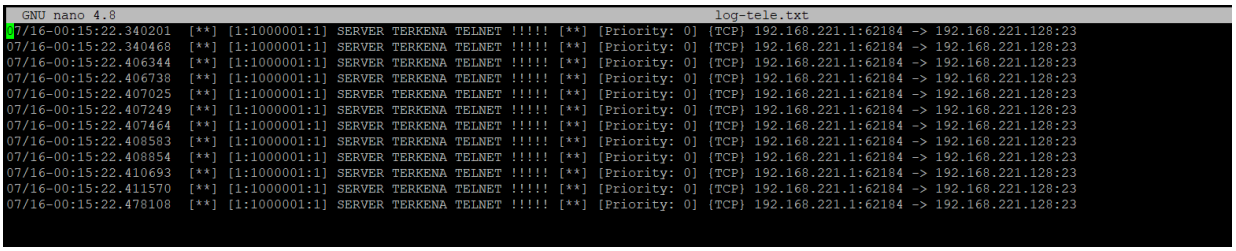
Pada Gambar 11 menunjukan hasil ping dari komputer penyerang mengirimkan 4 paket data menuju komputer server dan pada Gambar 12 menunjukan hasil *log* pemberitahuan adanya *ping* dari komputer penyerang. Selanjutnya adalah pengujian melakukan perintah telnet dari command prompt di windows untuk mengakses komputer server yang sebelumnya sudah terpasang telnet.

```
telnet 192.168.221.128
```

Kode Program 8. Perintah untuk menjalankan *telnet* pada *command prompt*

Kode Program 8 adalah kode program untuk perintah *telnet* dengan IP tujuan adalah komputer server.

Gambar 13. Hasil setelah menjalankan perintah *telnet* pada *command prompt*



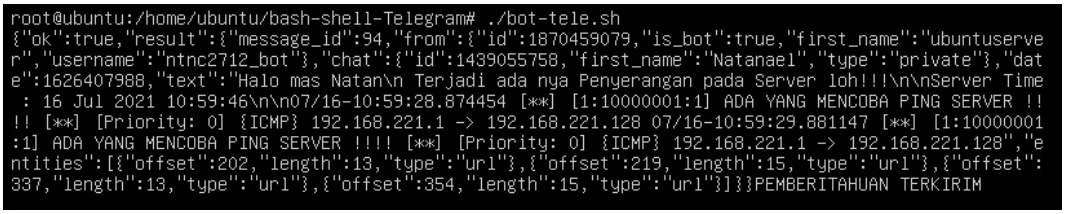
Gambar 14. Hasil log pemberitahuan untuk *telnet*

Pada Gambar 13 dapat dilihat bahwa komputer server berhasil diakses oleh komputer penyerang melalui alamat ip komputer server, dapat dilihat juga untuk Gambar 14 menunjukan hasil log pemberitahuan adanya telnet yang mencoba mengakses dari komputer penyerang.Setelah serangan berhasil dikirimkan oleh komputer penyerang dan diterima oleh komputer server maka mendapat kan hasil dan kesimpulan penelitian.

III. HASIL DAN PEMBAHASAN

A. Hasil Penelitian

Hasil yang ingin dicapai dalam penelitian ini adalah keberhasilan program pemberitahuan adanya serangan yang terjadi di komputer server melalui aplikasi Telegram. Setelah komputer melakukan penyerangan, komputer server menerima serangan lalu komputer server melakukan pengiriman pemberitahuan kepada admin melalui aplikasi Telegram seperti pada gambar 15 dibawah ini :

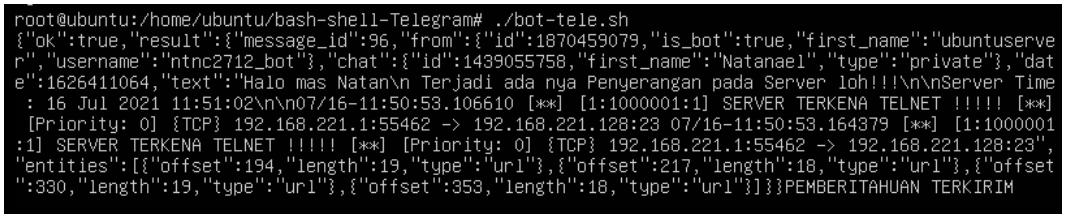


Gambar 15. Pemberitahuan yang tampil pada komputer server

Pada Gambar 15 menunjukan pemberitahuan yang tampil pada komputer server saat komputer penyerang mengirimkan *ping* dan format penulisan pemberitahuan yang tampil sesuai dengan aturan fungsi yang sudah dibuat pada file bash-shell Telegram.

Gambar *1*6. Tampilan pemberitahuan pada aplikasi Telegram

Pada Gambar 16 menunjukan hasil dari pemberitahuan adanya pengiriman *ping* dari komputer penyerang yang dikirimkan dari komputer server ke aplikasi Telegram. Pada pemberitahuan juga terdapat tampilan IP komputer penyerang dan tampilan waktu pada komputer server dan waktu terjadinya penyerangan sehingga pihak admin dapat mengetahui kapan terjadinya penyerangan pada komputer server. Pada percobaan pertama serangan dalam pengiriman pemberitahuan dari komputer server menuju ke aplikasi Telegram membutuhkan 18 detik. Selanjutnya melakukan akses telnet dari komputer penyerang ke komputer server dan hasil pemberitahuan yang tampil pada komputer server dan aplikasi Telegram.



Gambar 17. Tampilan Pemberitahuan serangan telnet pada komputer server

Pada Gambar 17 dapat dilihat adanya pemberitahuan yang tampil pada komputer server ketika adanya serangan telnet yang diterima komputer server.

Gambar 18. Tampilan Pemberitahuan pada aplikasi Telegram

Pada Gambar 18 adalah tampilan pemberitahuan serangan yang muncul pada aplikasi Telegram. Pada percobaan serangan telnet ini pengiriman pemberitahuan dari komputer server menuju ke aplikasi Telegram membutuhkan waktu 9 detik. Dengan adanya tampilan waktu juga dapat menghitung rata – rata waktu yang dibutuhkan untuk mengirimkan pemberitahuan dari komputer server menuju aplikasi Telegram.

1)Rata-rata Waktu Pengiriman Pemberitahuan: Dalam menentukan rata – rata waktu pengiriman pemberitahuan, pada penelitian ini melakukan 5 kali percobaan akses ping dan 5 kali percobaan akses telnet.

TABEL 2
HASIL RATA-RATA WAKTU PENGIRIMAN PEMBERITAHUAN AKSES PING

Percobaan ke -	Waktu Serangan	Waktu Notifikasi Diterima	Selang Waktu (detik)
1	10:57:04	10:57:23	19

Percobaan ke -	Waktu Serangan	Waktu Notifikasi Diterima	Selang Waktu (detik)
2	10:58:54	10:59:02	8
3	11:00:38	11:00:46	8
4	11:03:19	11:03:27	8
5	11:08:06	11:08:13	7
Rata - rata Waktu			10

Pada Tabel 2 menunjukan bahwa dalam 5 kali percobaan akses *ping* rata – rata waktu yang dibutuhkan untuk mengirimkan pemberitahuan menuju ku aplikasi Telegram adalah 10 detik.

TABEL 3
HASIL RATA-RATA WAKTU PENGIRIMAN PEMBERITAHUAN AKSES TELNET

Percobaan ke -	Waktu Serangan	Waktu Notifikasi Diterima	Selang Waktu (detik)
1	11:11:28	11:11:35	7
2	11:12:33	11:12:40	7
3	11:13:54	11:14:01	7
4	11:15:56	11:16:02	6
5	11:16:57	11:17:06	9
Rata - rata Waktu			7.2

Pada Tabel 3 menunjukan dalam 5 kali percobaan akses telnet dari komputer penyerang ke komputer server membutuhkan rata – rata waktu 7,2 detik untuk mengirimkan pemberitahuan ke aplikasi Telegram.

B. Pembahasan

1) Efektifitas Kinerja Pemantauan Jaringan Melalui Aplikasi Telegram: Dari hasil kinerja pemantauan jaringan server menggunakan aplikasi Telegram yang membutuhkan waktu rata – rata dibawah 10 detik maka penggunaan aplikasi Telegram sebagai sarana pemberitahuan dari server dapat dikatakan efektif, karena dengan adanya kecepatan ini juga akan berdampak pada cepatnya penyelesaian masalah pada server. Aplikasi media sosial Telegram juga memberi kemudahan kepada pengguna karena dapat memberikan informasi kapan dan dimanapun pengguna berada sehingga pengguna tidak harus memantau server dari jarak dekat dalam waktu yang lama.

2) Kekurangan dari Proses Pemantauan Jaringan Melalui Aplikasi Telegram: Untuk kekurangan dalam proses pengiriman pemberitahuan serangan yang terjadi pada komputer server dapat disebabkan oleh beberapa hal, salah satunya adalah koneksi internet yang tidak stabil, karena jika koneksi internet tidak stabil akan menyebabkan lambat nya pengiriman pemberitahuan menuju aplikasi Telegram.Jenis serangan yang tidak dikenali oleh *Snort* juga dapat menyebabkan serangan tersebut tidak terdeteksi, maka dari itu untuk mendapatkan kinerja yang maksimal dari *Snort* harus menuliskan lebih banyak *rule* agar *Snort* dapat lebih banyak mengenal dan mendeteksi serangan yang terjadi pada server.Kekurangan juga dapat terjadi pada pengguna yang tidak melihat adanya pemberitahuan yang terkirim pada aplikasi Telegram, hal tersebut dapat terjadi karena kemudahan yang fleksibilitas yang dihasilkan, contoh pada saat tidak menggunakan program ini maka pengguna akan *standby* di depan komputer server untuk mengetahui jika terjadi serangan namun ketika menggunakan program ini pengguna dapat meninggalkan komputer server dan memantau jika terjadi serangan pada komputer server dengan menunggu adanya pemberitahuan pada aplikasi Telegram yang terpasang di *smartphone*, saat *smartphone* tidak dibawa atau dalam keadaan mati maka pengguna tidak dapat melihat dan menerima pemberitahuan jika terjadi serangan pada komputer server.

IV. SIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan maka dapat disimpulkan bahwa penggunaan aplikasi layanan pengirim pesan Telegram sebagai sarana pemberitahuan adanya serangan pada komputer server sangat membantu dan mempermudah pekerjaan. Dengan kemudahan yang didapatkan dari program ini juga masih terdapat kekurangan yang jika terjadi dapat berdampak besar untuk komputer server. Sebaiknya untuk meminimalisir kejadian yang merugikan komputer server maka harus diperhatikan selalu koneksi internet baik pada komputer server maupun smartphone pengguna agar tidak terjadi keterlambatan pengiriman dan penerimaan pemberitahuan, dengan begitu masalah yang terjadi pada server dapat segera diatasi.

DAFTAR PUSTAKA

[1] Setiyawan, W. B. M., Churniawan, E. & Faried, F. S. "Information Technology Regulatory Efforts in Dealing With Cyber Attack To Preserve State," *Jurnal USM*, 3(2), 275–295, 2020.
[2] Harjono & Wicaksono, A. P., "Sistem Deteksi Instusi dengan Snort (Instrusion Detection System with Snort)," *JUITA*, 3(1), 31–34, 2014.
[3] Wijaya, B. & Pratama, A. "Deteksi Penyusupan Pada Server Menggunakan Metode Intrusion Detection System (Ids) Berbasis Snort," *Jurnal Sisfokom (Sistem Informasi Dan Komputer)*, 9(1), 97–101, 2020, https://doi.org/10.32736/sisfokom.v9i1.770.
[4] Sobari, I. A, "Rancangan Wireless Intrusion Detection System Menggunakan Snort," *Jurnal Techno Nusa Mandiri*, 12(1), 1–9, 2015.
[5] Sutarti, Pancaro, A. P. & Saputra, F. I, "Implementasi IDS (Intrusion Detection System) Pada Sistem Keamanan Jaringan SMAN 1 Cikeusal," *Jurnal PROSISKO*, 5(1), 1–8, 2018.
[6] Gunawan, A. R., Sastra, N. P., & Wiharta, D. M., "Penerapan Keamanan Jaringan Menggunakan Sistem Snort dan Honeypot Sebagai Pendeteksi dan Pencegah Malware," *Majalah Ilmiah Teknologi Elektro*, 20(1), 81–88, 2021.
[7] Firdaus, A. Z, "Implementasi Snort Sebagai Tool Intrusion Detection System pada Server FreeBSD di PT. Power Telecom," Disertasi Doktoral, Universitas Muhammadiyah Surakarta, 2012.
[8] Aris Widya, M. A. & Airlangga, P., "Pengembangan Telegram Bot Engine Menggunakan Metode Webhook Dalam Rangka Peningkatan Waktu Layanan E-Government," *Saintekbu*, 12(2), 13–22, 2020, https://doi.org/10.32764/saintekbu.v12i2.884.
[9] Juniyantara Putra, R., Putra Sastra, N., & Wiharta, D. M., "Pengembangan Komunikasi Multikanal untuk Monitoring Infrastruktur Jaringan Berbasis Bot Telegram," *Jurnal SPEKTRUM*, 5(2), 152, 2018, https://doi.org/10.24843/spektrum.2018.v05.i02.p19.
[10] Putri, L., "Implementasi Intrusion Detection System (Ids) Menggunakan Snort Pada Jaringan Wireless (Studi Kasus: SMK Triguna Ciputat)," Laporan Skripsi, Universitas Islam Negeri Syarif Hidayatullah, 2011.
[11] Cokrojoyo, A., Andjarwirawan, J., & Noertjahyana, A., "Pembuatan Bot Telegram Untuk Mengambil Informasi dan Jadwal Film Menggunakan PHP," *Jurnal Infra*, 5(1), 224-227, 2017, http://studentjournal.petra.ac.id/index.php/teknik-informatika/article/view/5163.
[12] Panggabean, P., "Analisis Network Security Snort Metode Intrusion Detection System untuk Optimasi Keamanan Jaringan Komputer," *Jursima*, 6(1), 2018, https://doi.org/10.47024/js.v6i1.107.