BAB 4
HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

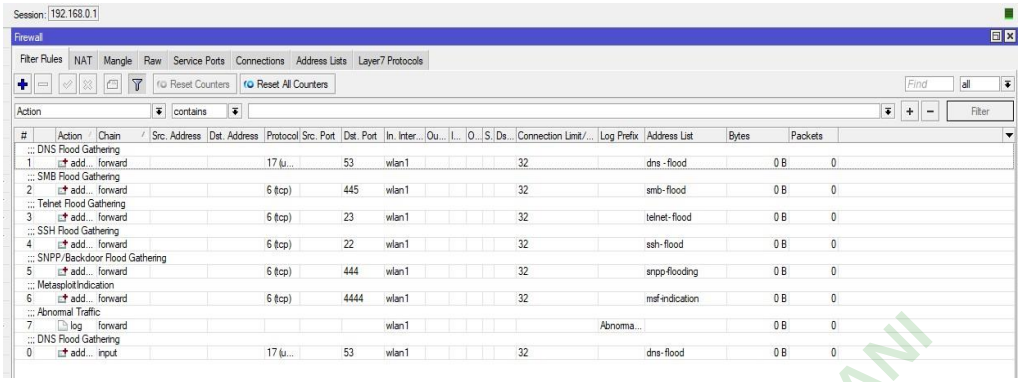
Hasil dari penelitian ini adalah pendeteksian adanya terjadinya anomali atau sesuatu aktivitas yang tidak wajar pada lalu lintas jaringan UNJAYA, termasuk pendeteksian serangan yang menggangu jaringan UNJAYA dengan cara melakukanan serangan flood pada jaringan sehingga menyebabkan penurunan kinerja pada perangkat mikrotik. Dalam penelitian ini penulis mekakukan dokumentsi dari hasil dalam melakukan identifikasi pola anomali pada lalu lintas jaringan unjaya serta implmentasi mitigasi untuk mengatasi serangan tersebut. Pendeteksian pola anomali dilakukan dengan menggunakan analisis log dari perangkat mikrotik. Log tersebut mencatat beberapa alamat IP yang mencurigakan pada sumber alamat IP *source address*, yang berperan dalam serangan terhadap jaringan.

Untuk mendapatkan *log* yang sesuai seperti jenis yang penulis ingikan serta periode waktu yang penulis tentukan, maka penulis perlu penyesuaian dalam membuat aturan firewall. Aturan *firewall* ini dilakukan untuk memblokir alamat IP yang teridentifikasi sebagai sumber serangan. Pengaturan ini bertujuan untuk membatasi akses dari entitas yang mencurigakan dan menjaga kestabilan jaringan. ada beberapa tahap untuk melakukanya.

4.2 IMPLEMENTASI DAN KONFIGURASI FIREWALL

4.2.1 Konfigurasi Firewall

Pada tahahapan konfigurasi mikrotik ini ialah untuk mengaktifkan fitur *firewall* untuk mendeteksi akativitas yang mencurigakan dan menangkap aktivitas yang menggangu pada jaringan.



Gambar 4.1 Pengaturan *Firewall* Filter *Rules*

Pada Gambar 4.1 dijelaskan bahwa semua aturan yang dirancang pada *firewall* ini untuk mengidentifikasi dan mengumpulkan informasi tentang berbagai jenis serangan *flood* yang dapat mengganggu jaringan. Dengan menambahkan alamat IP sumber dari serangan-serangan ini kedalam daftar alamat khusus maka *firewall* dengan secara langsung dapat mengidentifikasi dan menanggapi berbagai jenis serangan *flood* yang dapat menggangu jaringan. Berikut adalah sedikit penjelasan tentang setiap aturan *firewall* yang di buat.

1. DNS *Flood Detection*

   Aturan ini dirancang untuk mendeteksi serangan DNS *flood* yang mencoba membanjiri server DNS dengan permintaan yang berlebihan. Dengan menambahkan alamat IP yang mencurigakan ke daftar alamat khusus.

2. SMB *Flood Detection*

   Aturan ini digunakan untuk mendeteksi serangan SMB *flood* yang menargetkan port 445, yang digunakan oleh protokol *Server Message Block* (SMB). Serangan ini dapat mengganggu layanan berbagi file di jaringan. Dengan mendeteksi dan mencatat alamat IP yang terlibat, jaringan dapat dilindungi dari gangguan serangan.

3. Telnet *Flood Detection*

Serangan Telnet flood menargetkan port 23 dan berupaya membanjiri layanan dengan koneksi berlebihan. Aturan ini membantu mengidentifikasi sumber serangan dan memungkinkan tindakan mitigasi untuk mencegah gangguan pada layanan Telnet.

4. SSH *Flood Detection*

   Aturan ini mendeteksi serangan SSH flood yang menargetkan port 22, yang digunakan oleh *Secure Shell* (SSH). Dengan mengidentifikasi dan mencatat alamat IP yang terlibat, administrator dapat melindungi layanan SSH dari gangguan dan potensi akses yang tidak sah.

5. SNPP/*Backdoor Flood Detection*

   Serangan ini menargetkan port 444, yang digunakan oleh SNPP atau *backdoor* tertentu. Aturan ini dirancang untuk mendeteksi dan mencatat alamat IP yang mencoba membanjiri layanan ini, sehingga dapat diambil tindakan mitigasi yang sesuai.

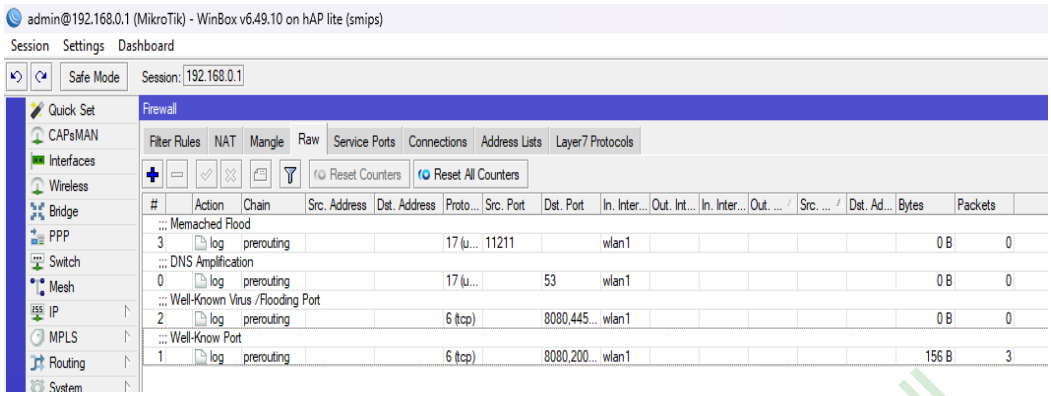
6. Metasploit *Indication*

   Aturan ini mendeteksi aktivitas yang menggunakan Metasploit, melalui port 4444. Dengan mengidentifikasi sumber serangan, jaringan dapat dilindungi dari eksploitasi dan akses tidak sah yang menggunakan alat ini.

7. *Abnormal Traffic Logging*

   Aturan ini mencatat aktivitas lalu lintas yang tidak normal berdasarkan jumlah data yang ditransfer. Dengan melog koneksi yang mentransfer lebih dari 80MB data, maka penulis dapat mengidentifikasi dan menganalisis aktivitas mencurigakan yang dapat menunjukkan adanya serangan atau aktivitas anomali lainnya.

Pada tahapan aturan selanjutnya yaitu membuat aturan ip firwall raw. Konfigurasi ini mengatur *firewall raw* pada mikrotik untuk melakukan logging pada paket yang telah di sesuaikan yang masuk ke jaringan.

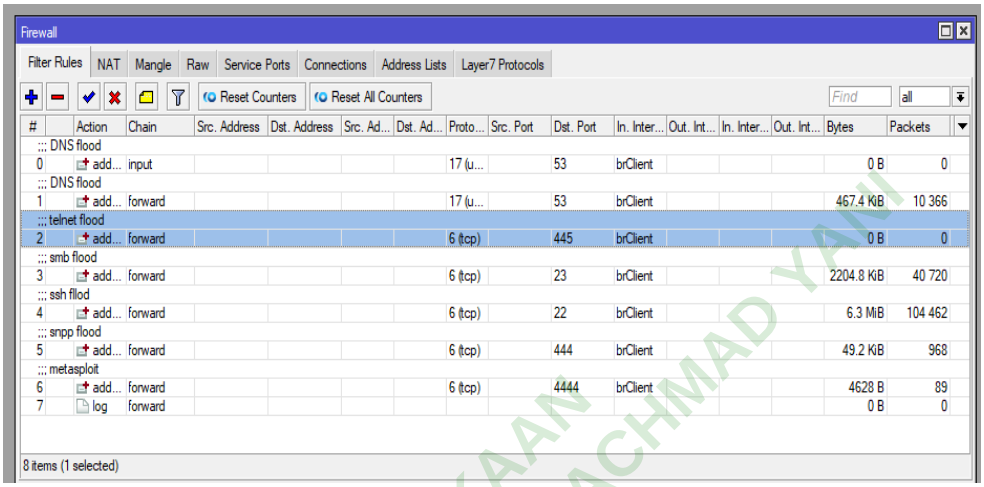


Gambar 4.2 Pengaturan *Firewall Raw*

Pada Gambar 4.2, dijelaskan tentang proses kerja *firewall raw*. Proses ini diawali dengan pembuatan aturan f*irewall* yang bertujuan untuk mengizinkan atau membatasi lalu lintas jaringan. Dengan adanya aturan ini, *firewall raw* dapat memproses paket data yang masuk sebelum paket-paket tersebut memasuki proses *routing* utama. Fungsi utama dari *firewall raw* adalah melakukan filter terhadap paket-paket yang datang, sehingga hanya lalu lintas yang diizinkan yang akan diteruskan ke tahap routing berikutnya. *Firewall raw* berperan sebagai lapisan pertama pertahanan yang menentukan apakah sebuah paket layak untuk diproses lebih lanjut oleh jaringan. *Firewall raw* ini mampu mengidentifikasi dan menolak paket yang mencurigakan atau tidak diinginkan pada tahap paling awal, sebelum paket tersebut memiliki kesempatan untuk menyebabkan kerusakan atau gangguan lebih lanjut dalam sistem.

4.3 ANALISIS ATURAN FIREWALL

4.3.1 Identifikasi Serangan

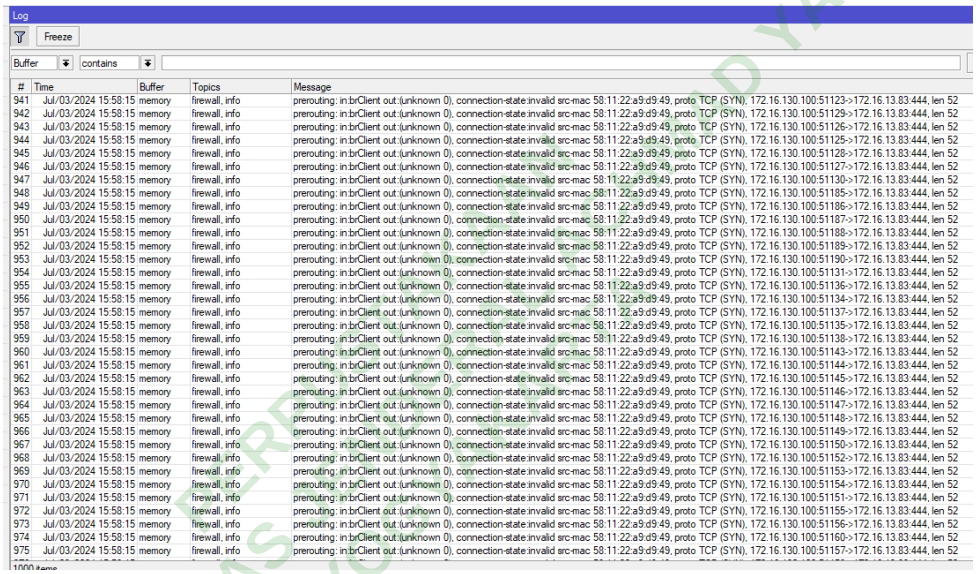


Gambar 4.3 Penerapan Filter *Firewall*

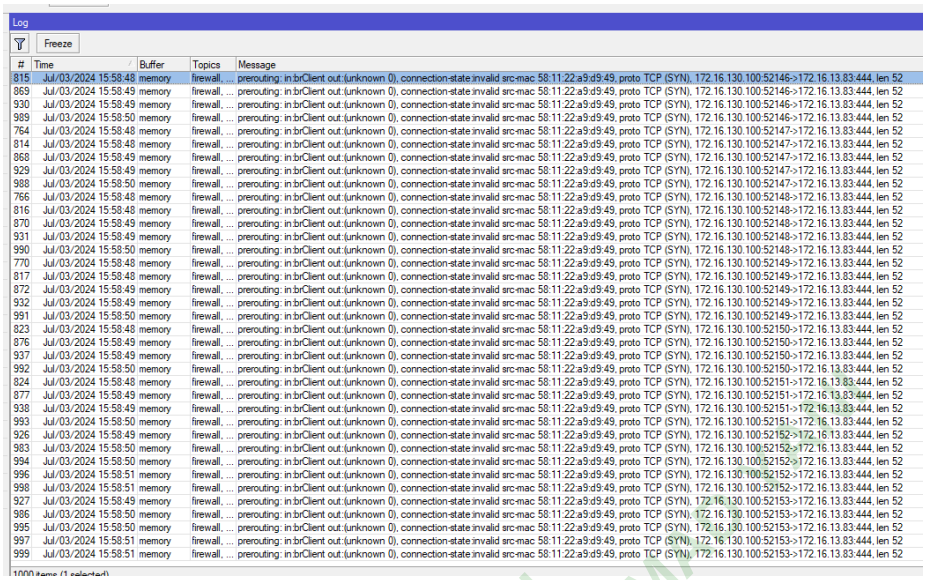
Gambar 4.3 menjelaskan bagaimana aturan *filter firewall* yang diterapkan oleh penulis berfungsi. Proses ini dimulai dengan penerapan aturan-aturan spesifik pada *firewall*, yang dirancang untuk memantau dan mengendalikan lalu lintas jaringan berdasarkan kriteria tertentu seperti jumlah koneksi, port tujuan, dan protokol yang digunakan. Seiring berjalannya waktu, aturan-aturan ini mulai aktif bekerja, memeriksa setiap paket data yang melewati jaringan. Ketika lalu lintas jaringan memenuhi kondisi yang ditentukan dalam aturan *firewall*, seperti jumlah koneksi yang melebihi batas yang telah ditetapkan, alamat IP sumber dari lalu lintas tersebut akan ditambahkan ke daftar alamat yang relevan (seperti dns-*flood* atau ssh-*flood*). Selain itu, untuk koneksi yang dianggap abnormal karena melebihi ukuran data tertentu, aturan logging akan mencatat informasi detail tentang lalu lintas tersebut ke dalam log. Akhirnya, log-log ini terkumpul dan mencerminkan aktivitas dan potensi ancaman yang melintasi jaringan, memberikan gambaran lengkap tentang kinerja dan efektivitas aturan *firewall* yang diterapkan. Pada gambar 4.3 menjelaskan mengenai proses aturan *firewall* filter yang penulis

terapkan berjalan lalu kemudian aturan itu terisi oleh log-log yang melintasi jaringan tersebut.

Pada gambar 4.4, menunjukan aktivitas log yang menunjukkan yang berhasil tercatat oleh MikroTik setelah adanya serangan *flood* pada jaringan. Log ini dihasilkan berdasarkan aturan *firewall* yang telah diterapkan untuk mendeteksi dan mencatat aktivitas mencurigakan.



Gambar 4.4 Log Mikrotik (1)

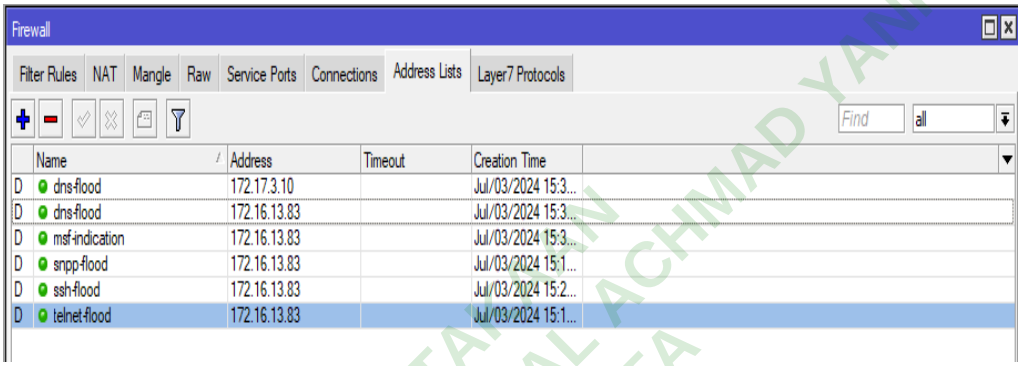


Gambar 4.5 Log Mikrotik (2)

Pada gambar 4.4 dan gambar 4.5 dijelaskan proses log ini terjadi dengan adanya aturan *firewall* yang telah diterapkan berfungsi untuk mendeteksi berbagai jenis serangan *flood* pada jaringan dan mencatat aktivitas yang mencurigakan dalam bentuk log. Setiap aturan dirancang untuk menambahkan alamat IP sumber yang mencurigakan ke dalam daftar alamat khusus (*address list*) ketika terdeteksi adanya lonjakan koneksi yang tidak normal ke port tertentu. Log yang dihasilkan dari aturan ini mencakup informasi seperti alamat IP sumber dari koneksi yang mencurigakan, jumlah koneksi yang dibuat dalam jangka waktu tertentu, port tujuan yang diserang, serta protokol yang digunakan. Dengan adanya log ini, penulis dapat mengidentifikasi pola serangan, melacak sumber serangan, dan mengambil tindakan mitigasi seperti memblokir alamat IP yang mencurigakan, atau melakukan analisis lebih lanjut untuk mencegah serangan.

4.3.2 Mitigasi Serangan

Firewall MikroTik mendeteksi serangan yang memenuhi kriteria yang telah ditetapkan pada aturan pada gambar 4.1, penambahan alamat ke dalam daftar (*address-list*) merupakan sebuah cara untuk mengatasi serangan yang terdeteksi pada jaringan. Alamat IP yang terlibat dalam serangan tersebut kemudian ditambahkan ke dalam daftar (*address-list*) di *firewal*l MikroTik.



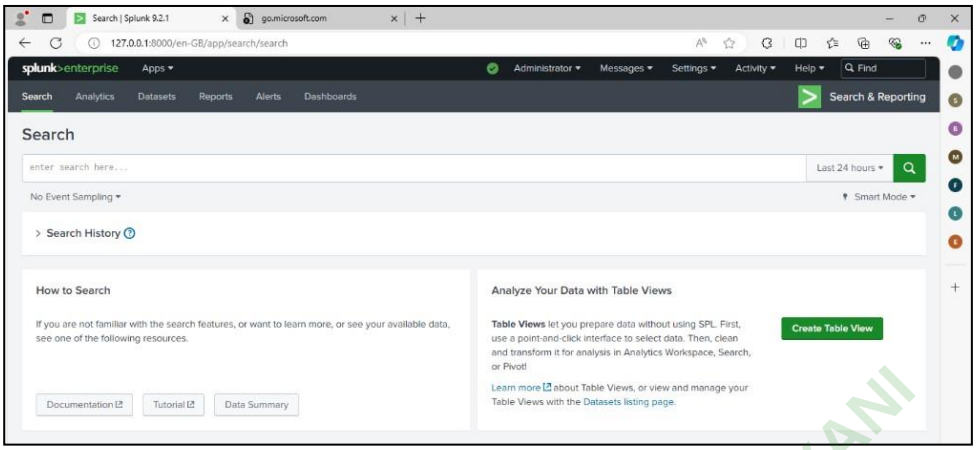
Gambar 4.6 Sumber Alamat IP

Pada gambar 4.6 Ini menunjukkan bahwa perangkat MikroTik yang digunakan oleh penulis telah berhasil mendeteksi dan mencatat berbagai jenis serangan *flood* yang ditujukan ke alamat IP tersebut. Setiap alamat IP yang terdeteksi menunjukkan pola serangan tertentu yang sesuai dengan aturan penulis terapkan. Maka dari itu alamat IP kemudian ditambahkan ke dalam daftar alamat (*addres list*).

4.4 IMPLEMENTASI SPLUNK

4.4.1 Instal Splunk

Pada penelitian ini tahap selanjutnya yaitu install Splunk, fungsi Splunk dalam penelitian ini ialah sebagai *platform* untuk pengumpulan data yang kemudian data log ini akan dilakukan analisis dan diolah.

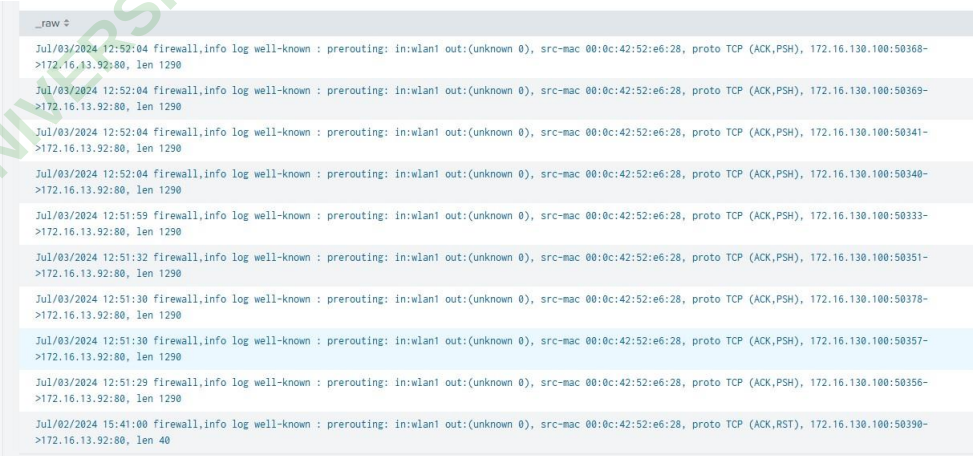


Gambar 4.7 Install Splunk

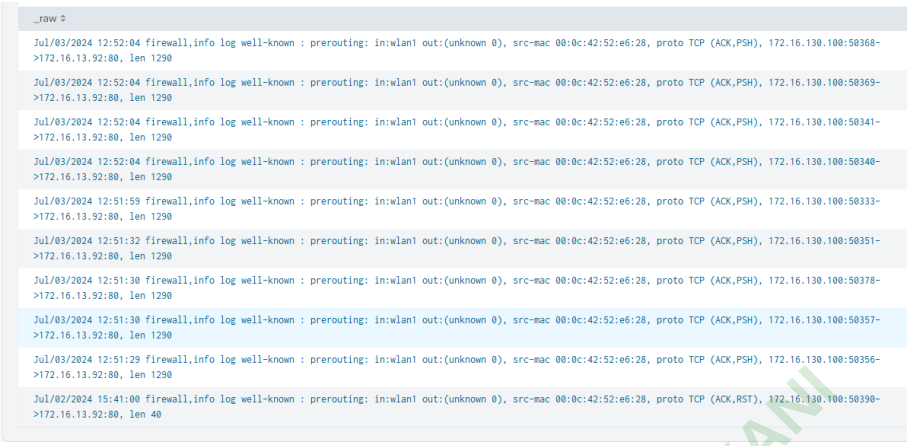
Pada Gambar 4.7 dijleaskan bahwa platform splunk ini digunakan penulis sebagai media pengumpulan dan pengolah data. Yang mana data log dalam penelitian ini bersumber dari mikrotik. Selain itu, penulis juga menggunakan Splunk ini untuk visualisasi data dalam penyampaian hasil akhir penelitianya.

4.4.2 Input Data

Add data pada splunk untuk mengumpulan dan menambahkan data yang sudah siap ,kemudian akan dilakukan analisis. pada gambar 4.8 adalah log yang berhasil di tambahkan ke splunk.



Gambar 4.8 Data Log Pada Splunk



Gambar 4.9 Data Log Pada Splunk

Pada gambar 4.8 dan gambar 4.9 ini adalah kumpulan log yang berhasil di upload kedalam Splunk.

4.4.3 Visualisasi Hasil

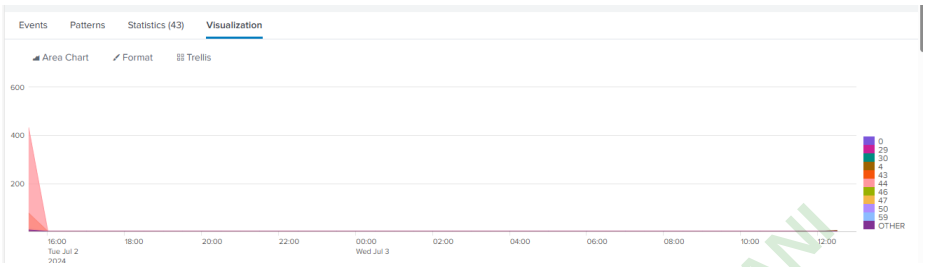
Setelah data sudah di upload pada Splunk, langkah selanjutnya yaitu mengolah data tersebut dengan membuat aturan yang digunakan untuk mencari dan menampilkan data. Pencarian dilakukan pada menu "Search & Reporting" pada Splunk. Berikut merupakan aturan-aturan yang digunakan:

a. Pada aturan ini penulis membuat aturan untuk menampilkan data log berdasarkan jumlah *event* yang tercatat per detik. Berikut aturanya ditujukan pada gambar 4.10

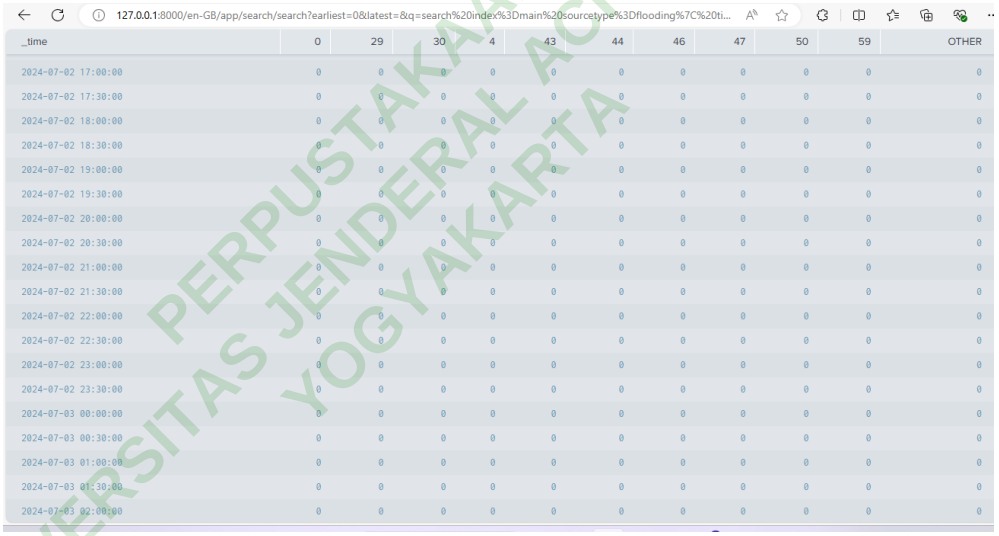
```
index=main sourcetype=flooding| timechart count by date_second limit=10
```

Gambar 4.10 Aturan Jumlah *Event* Per Detik

Dari aturan pada gambar 4.10 , memperoleh hasil seperti pada gambar 4.10 dan gambar 4.11



Gambar 4.11 Hasil Pengelompokan Berdasarkan Grafik



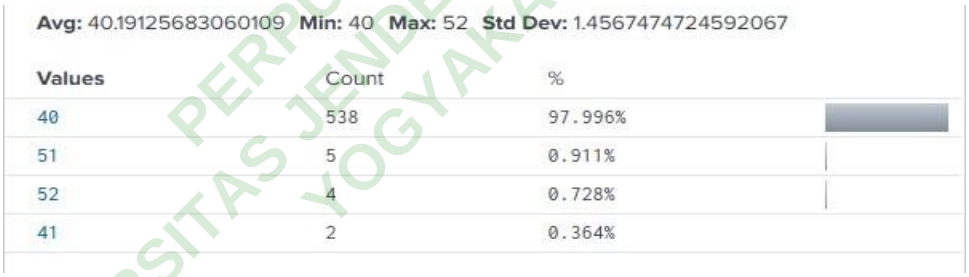
Gambar 4.12 Hasil Pengelompokan Berdasarkan Data Tabel

Penulis Membuat chart waktu yang menghitung jumlah *event* yang dikelompokkan berdasarkan *field date_second*, dengan batasan 10 nilai teratas. Tercatat jumlah *event* 549 *events* pada jam 14:44:25.000 di tanggal 02 juli 2024. Pada gambar 4.11 adalah area chart grafik yang menampilkan jumlah *event* per detik dari t*imestamp* (*date_second*). Di sumbu X adalah waktu (*_time*), dan di sumbu Y adalah jumlah *event*, Setiap warna dalam grafik mewakili nilai berbeda dari *date_second*.

Pada tabel data yang berada di bawah grafik itu adalah menunjukkan jumlah *event* per detik dari t*imestamp* (*date_second*) untuk setiap periode waktu .Kolom _time menunjukkan interval waktu. Kolom lain (0, 29, 30, 4, 43, 44, 46, 47, 50, 59, *OTHER*) menunjukkan nilai *date_second.*

Pada tanggal 02-07-2024 dijam 16:00:00, ada 0 *event* untuk *date_second* 0, 2 event untuk date_second 30, 0 *event* untuk *date_second* 29, dan Puncaknya yaitu pada *date_second* 44 Terlihat jelas bahwa *date_second* 44 memiliki jumlah *event* yang sangat tinggi (432 *event*), jauh lebih tinggi dibandingkan dengan nilai lainnya. Sebagian besar *event* terjadi pada tanggal 02-07-2024 sekitar jam 15:30 dan 16:00.

Ada beberapa nilai *date_second* lainnya yang muncul dengan frekuensi yang lebih rendah seperti 43, 50, 46, 47, dan seterusnya. Grafik pada gambar 4.11 menunjukkan bagaimana distribusi *event* berdasarkan nilai *date_second* selama periode waktu yang ditentukan.



Avg: 40.19125683060109 **Min:** 40 **Max:** 52 **Std Dev:** 1.4567474724592067

Values	Count	%
40	538	97.996%
51	5	0.911%
52	4	0.728%
41	2	0.364%

Gambar 4.13 Hasil Jumlah *Event* Berdasarkan Menit

Pada gambar 4.13 menunjukkan hasil dari analisis *field date_minute* dalam Splunk. *date_minute*: Ini adalah *field* yang menunjukkan menit dari *timestamp event*. Berikut penjelasan setiap bagian dari hasil gambar 4.13:

a. Statistik Deskriptif

1) *Average* (Rata-rata): 40.19125683060109
2) *Min* (Nilai minimum): 40
3) *Max* (Nilai maksimum): 52
4) *Std Dev* (Standar deviasi): 1.45674747424592067

b. Nilai Teratas dari *date_minute* adalah

1) 40: Muncul sebanyak 538 kali (97.996% dari total *event*).
2) 51: Muncul sebanyak 5 kali (0.911% dari total *event*).
3) 52: Muncul sebanyak 4 kali (0.728% dari total *event*).
4) 41: Muncul sebanyak 2 kali (0.364% dari total *event*).

Distribusi *date_minute*: Sebagian besar event (97.996%) terjadi pada menit ke-40. Puncak pada menit ke-40 menunjukkan bahwa nilai *date_minute* 40 sangat dominan dalam data yang diolah.

Nilai rata-rata dari *date_minute* adalah sekitar 40.19, dengan standar deviasi sekitar 1.46, yang menunjukkan bahwa sebagian besar nilai *date_minute* berkumpul di sekitar nilai 40. Ada kecenderungan yang sangat kuat bahwa sebagian besar *event* terjadi pada menit ke-40 dalam satu jam. Adapun Nilai-nilai lainnya yang muncul dengan frekuensi yang jauh lebih rendah, menunjukkan bahwa mereka adalah outlier atau terjadi dalam kondisi yang lebih jarang. Nilai-nilai seperti 40, 51, 52, dan 41 mewakili menit dalam satu jam. '*Count'* menunjukkan jumlah *event* yang terjadi pada menit tersebut, dan '%' menunjukkan persentase kontribusi dari total *event*.

Jadi dari hasil deteksi anomali yang dilakukan terlihat hasil pada gambar 4.13 menunjukan bahwa pada menit ke-40 memiliki 538 *event* (97.996% dari total), sementara menit lainnya memiliki jauh lebih sedikit event, ini bisa terjadi karena adanya aktivitas yang tidak biasa atau serangan yang terfokus pada menit tersebut.