**ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika**

ISSN(p): 2338-8323 | ISSN(e): 2459-9638 | Vol. 12 | No. 1 | Halaman 65 - 79
DOI : http://dx.doi.org/10.26760/elkomika.v12i1.65 Januari 2024

# Pola Pengelompokan dan Pencegahan *Public Honeypot* menggunakan Teknik K-Means dan *Automation Shell-Script*

**HILLMAN AKHYAR DAMANIK, MERRY ANGGRAENI**

Teknik Informatika, Universitas Budi Luhur, Indonesia
Email: hillman.akhyardamanik@budiluhur.ac.id

*Received* 3 Agustus 2023 **|** *Revised* 15 September 2023 **|** *Accepted* 13 Oktober 2023

**ABSTRAK**

*Makalah ini mengimplementasikan sistem log honeypot untuk menganalisis eksploitasi dari global internet berupa kategori serangan Statistical Traffic Analysis, Top Targeted Attack Sources and Destination, Penetration Analysis dan Infection Pattern Analysis serta Intrusion Detection System (IDS). Pengelompokan level kategori serangan adalah low, medium, dan high, dengan Teknik K-Means dan menerapkan rule filtering IPTables Automation yang digunakan untuk teknik mitigasi pada perangkat farm server dan virtual router public. Hasil attribute yang di cluster mendapatkan jumlah kuadrat jarak cluster ke pusat cluster terdekat, ditimbang dengan bobot nilai μi dan persentase jumlah serangan sebesar 64% untuk kategori High, 36% medium dan Low dengan jumlah tahapan clustering sebanyak 3 tahapan iterasi untuk mendapatkan cluster yang sesuai. Iterasi hasil Rule Firewall IPTables, untuk perangkat vRouter menghasilkan history beban kerja CPU berkurang menjadi 28%, dan memory 39%. vFarm Server menunjukkan beban kerja CPU pada masing-masing vServer berkurang menjadi 43% dan Memory (RAM) menjadi menjadi 21%.*

***Kata kunci**: Machine Learning, Cyber Security, Honeypot, K-Means, Firewall IPTables*

**ABSTRACT**

*This paper implements a honeypot log system to analyze exploitation of the global internet in the form of Statistical Traffic Analysis attack categories, Top Targeted Attack Sources and Destinations, Penetration Analysis and Infection Pattern Analysis and Intrusion Detection System (IDS). The grouping of attack category levels is low, medium, and high, using the K-Means technique and applying the IPTables Automation filtering rule used for mitigation techniques on server farm devices and public virtual router. The results of the clustering attribute get the mean of the squares of the cluster distance to the nearest cluster center, weighted by the weight of the μi value and the percentage of the number of attacks is 64% for the High, 36% medium and Low with a number of clustering stages of 3 iteration stages to get the appropriate cluster. Iteration of the results of the IPTables Firewall Rule, for vRouter devices, results in a history of CPU workload being reduced to 28%, and memory to 39%. vFarm Server shows the CPU workload on each vServer is reduced to 43% and RAM to 21%.*

***Keywords**: Machine Learning, Cyber Security, Honeypot, K-Means, Firewall IPTables*

## 1. PENDAHULUAN

Internet dan digitalisasi saat ini menghubungkan miliaran perangkat fisik. Perangkat ini sering kali tidak aman dan memiliki kerentanan yang sama. Bentuk serangan yang dominan bergantung pada kemajuan terkini dalam pemindaian dan penemuan perangkat di seluruh internet. Mekanisme dan penerapan *honeypot*s dan teknologi pembelajaran mesin dalam keamanan jaringan menawarkan kerangka kerja manajemen keamanan yang efektif dan dapat dipercaya. Saat ini infrastruktur jaringan menjadi salah satu dari bagian internet untuk komunikasi yang populer saat ini, seperti pada sebuah organisasi perusahaan yang berhadapan langsung dengan global internet saat ini, sering mengalami masalah keamanan data, kontrol hak cipta, transportasi data, dan otentikasi **(Damanik, 2022)**. Terlebih lagi organisasi dan perusahaan ketika mengoperasikan infrastruktur awan (*cloud computing*) dan jaringan serta menyimpan data pelanggannya pada infrastuktur jaringan dengan persyaratan infrastruktur bersama seperti ketersediaan (*availability*), skalabilitas (*scalability*), dan keamanan (*Security*) **(Fraunholz, dkk, 2017) (Damanik, 2020) (Damanik, 2021)**. Ancaman siber yang menargetkan salah satu persyaratan pada perangkat jaringan dapat mempengaruhi jutaan individu yang dapat menyebabkan kerentanan perangkat pelanggannya (*Costumer Enterprise*) **(Cunha, dkk, 2020) (Araujo, dkk, 2018)**. Untuk itu, sudah diwajibkan bagi organisasi atau perusahaan memiliki tanggung jawab untuk melindungi infrastruktur mereka dengan cara terbaik, tetapi mereka juga harus mematuhi banyak persyaratan hukum mengenai perlindungan data dan privasi yang diterapakan **(Kosseff, 2020) (Polyakov & Lapin, 2018) (Sokol, dkk, 2015)**.

*Honeypots* telah digunakan secara luas untuk eksperimen lalu lintas jaringan, dari serangan global internet. Sebagian besar makalah melibatkan perangkat lunak *honeypot*, yang beroperasi pada perangkat keras fisik dan pada konektivitas perangkat lunak jaringan (**Ceron, dkk, 2020)**. Penelitian sebelumnya penggunaan *honeypot* mengumpulkan informasi tentang penyusupan penyerang ke dalam sistem melalui *Honeypot*, dan gabungkan klasifikasi data, pelabelan manual, pembelajaran mesin, dan algoritma K-means untuk menyediakan kumpulan data. Metode yang diusulkan Menggunakan algoritma K-means untuk mengklasifikasikan data mentah yang dikumpulkan oleh *Honeypot*s ke dalam kelompok dan memberi label setiap cluster secara manual **(Liao, dkk, 2023)**. Penggunaan clustering honeynet digunakan untuk menentukan pola serangan berdasarkan rangkaian waktu aktivitas penyerang. Hasil penelitian bahwa pendekatan yang diusulkan mengelompokkan tindakan jahat yang dipantau oleh honeynet untuk dapat mengidentifikasi pola serangan **(Kashtalian & Sochor, 2021)**. Makalah konfigurasi dan *honeypot* secara dinamis dengan pembelajaran mesin diimplementasikan. *Honeypot*s ditempatkan secara cerdas di jaringan sebagai media untuk deteksi serangan **(Fraunholz, dkk, 2021)**. Pengenalan model deteksi *clustering* dengan K-Mean digunakan untuk karakteristik malware dengan mempelajari perilaku malware. Sepanjang percobaan, malware. Hasil yang diusulkan dapat digunakan untuk pengelompokan data dengan penggunaan data registry untuk mendeteksi malware **(Rosli, dkk, 2019)**. **(Owezarski, 2014)** Dalam penelitiannya, menyajikan metode unsupervised learning untuk klasifikasi dan karakterisasi anomali terkait keamanan dan serangan yang terjadi di *honeypot*s. Dan menunjukkan bagaimana hasil dari karakterisasi anomali untuk menyimpulkan aturan pemfilteran yang dapat digunakan untuk mengonfigurasi secara otomatis router jaringan, switch atau firewall. **(Kamel, dkk, 2020)** Dalam dalam penelitiannya menyajikan metode K-Means, *Decision Trees* dan sistem *honeypot*, dengan topologi tersebut penulis merancang agen cerdas untuk pencegahan dan prediksi serangan siber. Dari beberapa metode yang digunakan untuk Teknik keamanan belum terdapat kombinasi dengan *filtering* dan *blocking* dari pola serangan yang diterapkan.

Tujuan dari makalah ini berfokus untuk menganalisis log *honeypot*, dengan sistem deteksi dengan menerapkan pemodelan pada sistem pemantauan monitoring (*data visualization*), dengan virtual T-Pot *Honeypot* dengan ruang lingkup untuk mempelajari *landscape* ancaman perangkat infrastruktur jaringan, dengan mengimplementasikan dan mengkombinasikan sistem *honeypot* T-Pot yang dirancang mengikuti dataset *Honeypot* Attacks Cowrie, Dionaea yang multiguna, yang menyebarkan protokol *docker container* untuk meniru layanan yang dapat dieksploitasi dari global internet. Monitoring dan analisis serangan berupa *Statistical Traffic Analysis, Top Targeted Attack Sources and Destination, Penetration Analysis* dan *Infection Pattern Analysis serta Intrusion Detection System* (IDS), dengan tiga level serangan *low*, *medium*, dan *high*, dengan Teknik pembelajaran mesin K-Means sebagai pengelompokan dataset dan Metode *rule filtering* IPTables Automation digunakan untuk teknik mitigasi dalam percobaan serangan diwaktu yang akan datang pada penerapan *honeypot*. Selanjutnya secara *real-time* data primer dikumpulkan dengan menggunakan tiga honey-mesh vT-Pot *Honeypot* External (Gateway), vT-Pot *Honeypot Farm* Server dan vT-Pot *Honeypot* Core Router. Hasil analisis dataset yang dilakukan dengan cara entri log pada setiap *honeypot*, dengan percobaan selama dua bulan. Hasil dataset Clusterisasi dengan memanfaatkan metode K-Means dari beberapa attribute yang diclusterisasi berdasarkan (*Protocol* TCP/UDP, *Source Address, Destination Address, Source Port, Destination Port, Local, Type, Latitude, Longitude dan Country*) untuk mendapatkan jumlah kuadrat jarak cluster ke pusat cluster terdekat, ditimbang dengan bobot nilai μi dengan persentase jumlah serangan untuk kategori High, Medium dan Low dengan jumlah tahapan clustering sebanyak untuk mendapatkan cluster yang sesuai. Pengelompokan hasil dari *rule* Firewall IPTables, yang diimplementasikan untuk perangkat vRouter dan *vFarm* Server, akan diujikan untuk menghasilkan perbandingan beban kerja histori CPU dan Memory (RAM) pada saat *session* operasional eksperimen dilakukan dengan nilai rata-rata rata-rata $x = (\Sigma f\_n\, x\_n)/\Sigma f$.

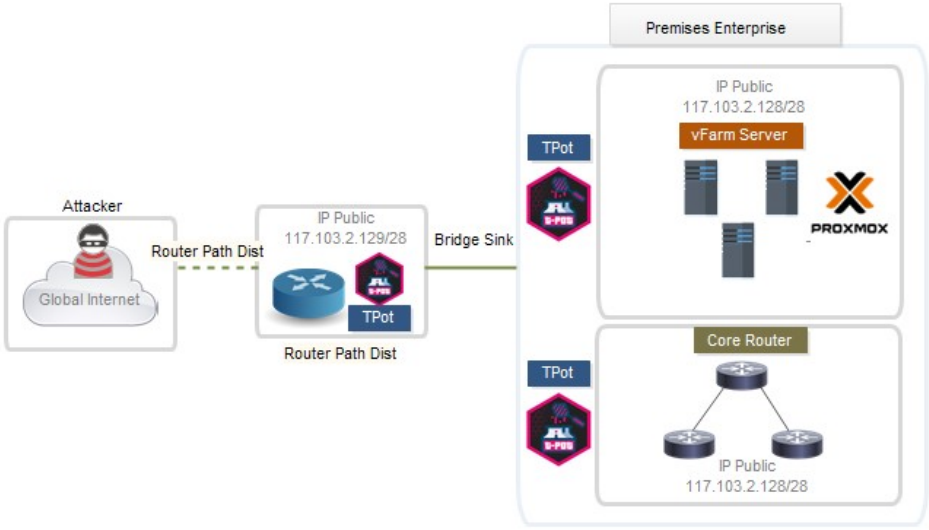
## 2. METODE

Sebelum membahas hasil pengujian dan hasil dari kumpulan dataset pada penelitian, penting untuk mengetahui metode dan arsitektur yang digunakan untuk mengekstrak data dari *Honeypot* T-Pot yang dimodelkan.

### 2.1 Arsitektur Penerapan dan Virtualisasi *Honeypot* pada Lingkungan Network Enterprise

Desain sistem topological dan spesifikasi perancangan dimodelkan seperti Gambar 1. Pelacakan lalu lintas pada masing-masing interaksi pada *honeypot* yang berbeda didesain untuk mendapatkan kemampuan yang berbeda. Adapun jenis serangan akan dimonitoring dan dievaluasi dalam 2 bulan dari *public* internet dengan pemantauan dan analisis serangan Traffic berupa analisis koneksi dan permintaan (*request*) pada *network layer* (L3), *Top Targeted Attack Sources and Destination, Target Port* berupa layanan *(service)* dan protocol mana yang ditargetkan, secara terpusat yang akan menghasilkan dan mengumpulkan informasi tentang *Protocol TCP/UDP, Source Address, Destination Address, Source Port, Destination Port, Local, Type, Latitude, Longitude* dan *Country*. Dataset yang diperoleh dari masing-masing *Honeypot* akan diekstrak dan dikumpulkan dalam format csv (*comma delimited*). Teknik K-Means melakukan clusterisasi hasil dataset tersebut dengan cara menyandingkan hasil dataset dari masing-masing data log virtual *honeypot*. Hal yang diperhatikan adalah nilai cluster yang dicari yang menjadi acuan titik persebaran objek serangan dari global internet, bagaimana ciri-ciri dari setiap cluster yang dapat dilihat dari nilai yang terdapat pada tiap cluster dengan melihat pada *attribute protocol* apa yang lebih

dominan. Berbeda dengan eksperimen yang sudah dilakukan menggunakan Teknik K-Means untuk mengelompokkan setiap kelas menjadi dua cluster **(Bahjat, dkk, 2020)**.



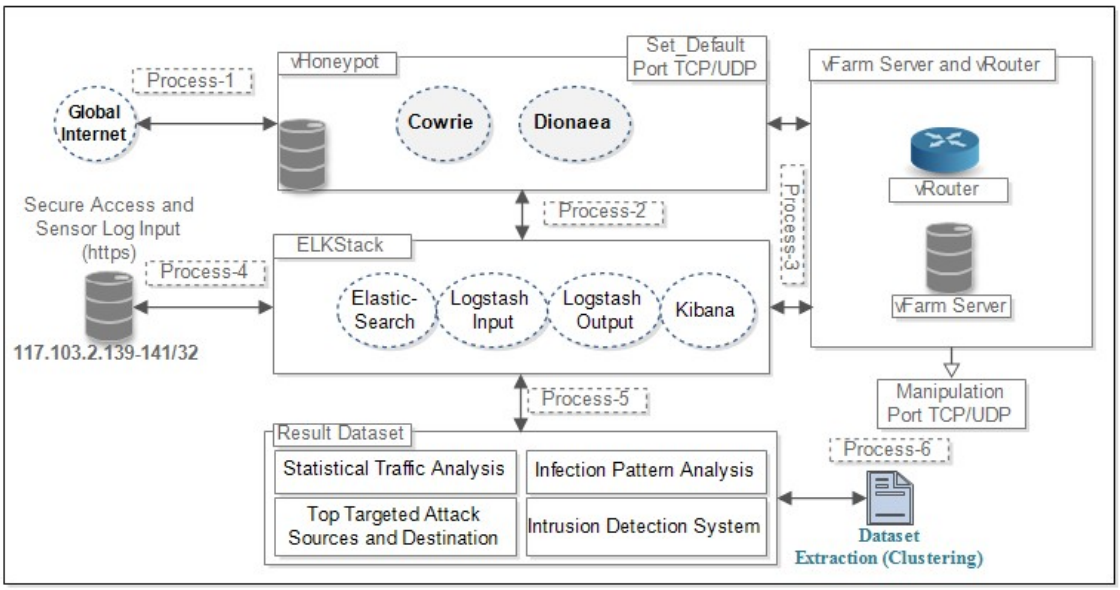

**Gambar 1. Arsitektur Penerapan dan Virtualisasi *Honey Mesh* pada Lingkungan *Network Enterprise***

Desain konfigurasi *honeypot* akan ditempatkan di depan *gateway uplink* internet agar koneksi yang ditangkap oleh *Honeypot* merupakan koneksi trafik murni dari luar tanpa adanya *filter* dari *gateway* dan firewall. Untuk mengatasi keterbatasan dan pengaturan kompleks jaringan *honeypot*, sistem virtual pada infrastruktur vFarm Server dan vRouter akan ditempatkan juga T-Pot *honeypot*. Terdapat lima *virtual* server dan 3 vRouter yang menggunakan IP Address public menggunakan skema *bridge management*. Pengaturan eksperimental terdiri dari dua T-Pot *honeypot*s dan sistem tambahan untuk mengumpulkan log yang dihasilkan. Sistem berjalan dalam *instance virtual* yang di *hosting* menggunakan IP Public dalam subnet yang sama dan didistribusikan pada Tabel 1.

**Tabel 1. Instance virtual distribusi Penggunaan Domain IP Address**

| **Farm Server Description** | **IP Address** | **Interface** |
|---|---|---|
| vFTP Server | 117.103.2.132 | Bridge Port Sink |
| vWeb Server | 117.103.2.133 | Bridge Port Sink |
| vMail Server | 117.103.2.134 | Bridge Port Sink |
| vCloud Server | 117.103.2.135 | Bridge Port Sink |
| vDB Server | 117.103.2.136 | Bridge Port Sink |
| **v*Honeypot*** | | |
| v*Honeypot* External (Gateway) | 117.103.2.139 | Bridge Port Sink |
| v*Honeypot* vRouter | 117.103.2.140 | Bridge Port Sink |
| v*Honeypot* vRouter | 117.103.2.141 | Bridge Port Sink |
| **Firewall IPTables** | | |
| Rule Automation and IPSet | 117.103.2.130 | Bridge Port Sink |

Sistem *honeypot* T-Pot yang didesain mengikuti kerangka kerja *Medium Interaction honeypot* Cowrie dan *Low Interaction Honeypot* Dionaea yang meniru layanan umum yang dapat dieksploitasi dari global internet seperti pada Gambar 2. Kerangka kerja T-Pot mengumpulkan semua log dari setiap *container* dan memusatkannya ke dalam *elastic stack* yang akan memberikan tampilan *front-end* Kibana dari semua serangan terhadap setiap layanan.




**Gambar 2. Penerapan Arsitektur Dataset *Honeypot***

Dataset yang digunakan dalam penelitian ini berupa hasil *record realtime* setiap *Logstash Honeypot*. *Logstash* digunakan untuk permintaan dan menambahkan input apa pun ke Database pencarian *elastic honeypot*. Dengan menyimpan hasil dalam database, untuk mudah dicari dan kueri data cepat didapatkan. Selama percobaan, arsitektur *honeypot* melakukan pendekatan untuk mendapatkan file log mentah dan secara bersamaan memproses data. Komunikasi antar server ditunjukkan pada Gambar 2. adalah skema komunikasi yang terjadi antar vServer dan vRouter ditunjukkan selama proses data dari Process-1 untuk semua port TCP/UDP pada *Honeypot* Cowrie dan Dionaea akan berstatus open. Process-2 adalah ELKStack menggunakan file konfigurasi pada *honeypot* untuk menentukan input file log dari data log *honeypot* Dionaea dan Cowrie. Melalui konfigurasi ini, penulis dapat menentukan bahwa semua timestamps waktu dikonversi ke format yang dipilih atau ditambahkan ke entri log. Sebagai contoh pada process-3, kapanpun *Logstash* menemukan *attribute* (*Protocol TCP/UDP, Source Address, Destination Address, Source Port, Destination Port, Local, Type, Latitude, Longitude dan Country*) dalam file log, yang disediakan database untuk menambahkan informasi ke Kibana. Outputnya kemudian dikirim *Elasticsearch*, yang dalam kolektor data mendengarkan (*listens*) permintaan pada *port* masing-masing TCP dan menambahkan input apa pun ke database pencarian *Elasticsearch* sampai pada process-4 terdiri dari analisis manual dengan mengekstrak informasi tentang penyerang berdasarkan process-5 berupa (Dataset). Dengan melihat tindakan yang dilakukan sebelum, selama dan setelah serangan, dipasangkan dengan data *fingerprinting* seperti geolokasi, Protocol TCP/UDP yang digunakan untuk dataset yang akan diperoleh dengan proses *extraction* kedalam ekstensi CSV.

## 2.2 *Clustering* Dataset *Honeypot* (Teknik K-Means)

Ide inti dari penggunaan Teknik k-means adalah melakukan pengelompokan dataset serangan dari hasil ekstrak yang dihasilkan dari masing-masing *logstash Honeypot*. Dataset yang digunakan berupa (*Protocol TCP/UDP, Source Address, Destination Address, Source Port, Destination Port,* dan *Country*). Object clusterisasi adalah hasil dari *attribute* dan akan disimbolkan sebagai sebagai variable $x$ dan variable $y$. Teknik dari algoritma k-means dirangkum sebagai berikut.

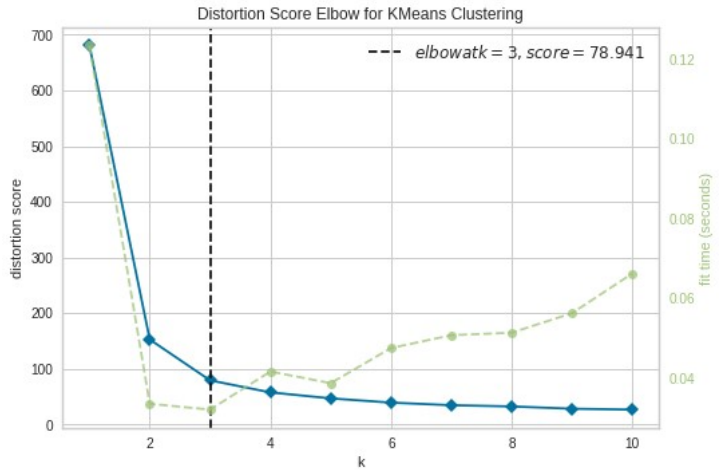
**Algorithm K-Means Algorithm**

***Input:*** *data set honeypot, k*
***Output:*** *Cluster:*

- *Statistical Traffic Analysis*
- *Targeted Attack Sources and Destination*
- *Penetration Analysis*
- *Intrusion Detection System*

1. *Select k points from* $C_j \{x: x: min\ distance^2\ (X, C_n) = j\}$
2. ***repeat:***
   $C_j = \frac{1}{m_j} \sum_{x \in C_j} x$
3. ***for*** *j=1, 2, ……, m*
4. *Calculate the Euclidean distance from point* $C_j$ *to each cluster center*
5. *determine the cluster class mark of* $C_j$ *according to the close distance*
6. *divide the sample points into corresponding clusters*
7. ***end for***
8. ***until*** *the cluster allocation result remains unchanged*

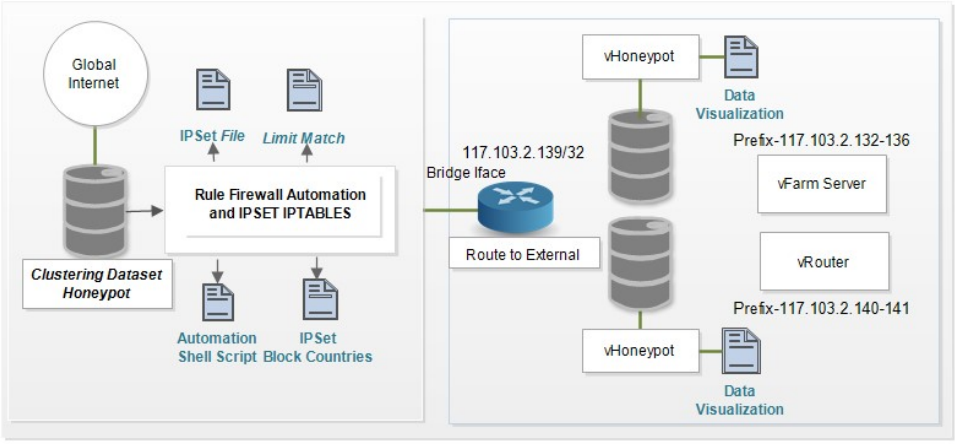
*Attribute* yang dihasilkan dari database *honeypot* selanjutnya akan melalui proses normalisasi. Sebagai titik awal pengujian jumlah cluster yang dihasilkan dari dataset *honeypot*, penulis akan menginisialisasi *centoid k* cluster dengan mengambil *attribute k* secara acak dari dataset. Metode inisialisasi ini dapat menghasilkan cluster dengan menggunakan metode Elbow. Cluster juga akan bergantung pada lokasi centroid awal. Dalam metode Elbow, memvariasikan jumlah cluster (*k*). Setiap nilai *k* dihitung dengan WCSS (*Within-Cluster Sum of Square*). Bertambahnya jumlah cluster, nilai WCSS akan berkurang. Ketika menganalisis grafik, dapat terlihat bahwa grafik akan berubah dengan cepat pada suatu titik dan dengan demikian menciptakan bentuk siku. Dari titik ini, grafik mulai bergerak teknik sejajar dengan sumbu *X*. Nilai *k* yang sesuai dengan titik ini adalah nilai *k* optimal atau jumlah cluster yang optimal. Berikut hasil pengujian data untuk metode Elbow. Jumlah cluster (*k*) yang optimal pada penelitian ditentukan dengan menggunakan metode elbow. Seiring bertambahnya jumlah cluster, jumlah kesalahan kuadrat (SSE) dalam cluster berkurang karena titik data semakin dekat ke pusat cluster masing-masing. Dengan metode siku tujuannya adalah untuk menemukan k dimana SSE menurun paling cepat. Oleh karena itu, dari hasil perhitungan pada Gambar 3. terlihat bahwa jumlah cluster terdapat pada 2, 3 dan yang optimal adalah 3.




**Gambar 3. Grafik Penentuan K-Cluster (*Elbow Method*) Dataset *Honeypot***

## 2.3 Perancangan Teknik IPTables Firewall

Penerapan *Rule Firewall Automation* IPTABLES diimplementasi seperti topologi pada Gambar 4. *Rule Firewall Automation* IPTABLES merupakan *policy* yang diimplementasikan dan dikombinasikan dengan konektivitas *Honeypot* yang melakukan aturan yang diberikan pada kernel untuk mengatur setiap paket *incoming* dan *outgoing* dalam melakukan *rule filtering* menuju perangkat gateway, vFarm Server dan vRouter yang melewati Router eksternal. *Policy rule* ini akan diterapkan berdasarkan dataset hasil dari clusterisasi teknik K-Means yang dihasilkan berdasarkan parameter *anomaly Low*, *Medium* dan *High*. Firewall IPTables pada eksperimen ini juga digunakan untuk menyediakan metode meminimalisir *load* pada perangkat vFarm Server dan vRouter, baik penggunaan CPU dan Memory (RAM).




**Gambar 4. Arsitektur Penerapan *Rule Firewall Automation* dan IPSet IPTables**

Hasil *clustering* dataset yang sudah dilakukan pengelompokan akan dilist pada *script firewall* IPTables dengan tahapan instalasi dan baris konfigurasi sebagai berikut:

1. *Limit Match (Dynamic Database Blacklisted IP)*

Rules *limit match* diterapkan secara eksplisit dengan opsi –m limit. *Limit match* ini diterapkan untuk memberikan *rules* terbatas pada parameter protocol dari nilai yang sudah ditentukan. *Set rule* Pada tahap pertama memproses kebijakan *rule* firewall yang diambil dari log file *clustering* dataset berdasarkan *policy*.

2. *Firewall IPSet File (Block All Addresses from a File)*

Firewall IPSet diterapkan untuk ekstensi target untuk menyediakan mekanisme menambahkan dan menghapus entri yang ditetapkan secara dinamis berdasarkan aturan IPTables. *Script* IPSet akan melakukan entri secara otomatis untuk memblokir semua IP yang dilist pada file data yang dihasilkan dari log file *clustering* dataset berdasarkan *policy*.

3. *IPSet block countries (IPTables block berdasarkan negara)*

IPSet *block countries* (IPTables *block* berdasarkan negara) menggunakan file zona IP Public. *Script* IPSet *block countries* akan melakukan entri secara otomatis untuk memblokir semua IP yang dilist pada file data yang dihasilkan dari hasil *Honeypot Clustering* dataset. Daftar *rule* firewall IPTables yang digunakan pada eksperimen ini ditunjukkan pada *script* berikut untuk negara Korea.

```
1. #!/bin/bash
2. echo "### BLOCKING KOREA ###"
3. if [ -f "cn-aggregated.zone" ]
4. then
5.   rm cn-aggregated.zone
6. fi
7. wget http://www.ipdeny.com/ipblocks/data/aggregated/cn-
   aggregated.zone
8. if [ $? -eq 0 ]
9. then
10.         echo "Download Finished!"
11.  else
12.      echo "Download Failed! Exiting ..."
13.      exit 1
14.  fi
15.  ipset -N korea hash:net -exist
16.  ipset -F korea
17.  echo "Adding Networks to set..."
18.  for I in `cat cn-aggregated.zone`
19.  do
20.         ipset -A korea $i
```

4. *Firewall Automation Using Shell Script*

Kriteria untuk *automation shell script* ini akan mempunyai *rule* untuk *accept* dan *drop* IP Address berdasarkan CIDR dan *action* (*Deny* dan *Accept*) *Source* dan *Destination Port* yang dilakukan firewall terhadap paket dan *traffic* data dari global internet. *Rule* firewall yang akan diproses adalah pada *header* paket IP seperti *protocol* (TCP atau UDP), *source* IP *address*, *source port*, *destination* IP *address*, (*Deny* dan *Accept*).

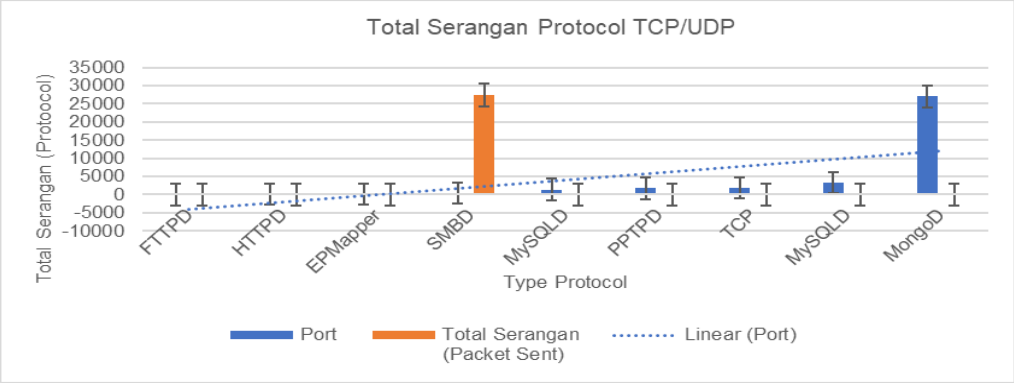
## 3. HASIL DAN PEMBAHASAN

Percobaan eksperimen dilakukan antara 03 Maret 2023 pukul 00:00 dan 30 Mei 2023 pukul 23:59, dan terdapat 5.013.676 entri log database. terdapat sekitar 268.614 unik *Source* IP Address dari beberapa negara. Database *attack* yang dihasilkan dari Medium *Honeypot*

sebanyak (Cowrie) 2.813.776 dan Low *Honeypot* (Dionaea) 2.199.900 dari hasil layanan v*Honeypot* yang berbeda dihasilkan. Pengelompokan Dataset dari database *honeypot* yang dihasilkan berdasarkan 4 kategori, dengan tahapan *pre-processing* dengan memastikan untuk setiap atribut *Destination Port, Source Port, Source IP Address, Destination IP Address, Country, Type, Postal Code* berisi data yang valid dan tidak terdapat missing *value*, dengan melakukan *handling missing* data.

### 3.1 Pengelompokan *Low Interaction* v*Honeypot* Dionaea

Pada Pengelompokan Low Interaction *Honeypot* Dionaea, cluster dan *centroid* hasil *Honeypot* Dionaea ini dibagi menjadi tiga tingkatan (*k=3*), yaitu *Low, Medium* dan *high* berdasarkan tingkat serangan pada *service* protocol TCP/UDP. Pada algoritma K-Means perhitungan centroid baru akan terus dilakukan (iterasi) sampai dengan ditemukannya iterasi yang mana hasil *centroid*-nya sama dengan hasil *centroid* sebelumnya. Tahap pertama adalah memplot dataset Jenis Serangan *Low Interaction Honeypot* Dionaea. Idenya di sini adalah untuk memplot kumpulan data *Honeypot* Dionaea dan membandingkan fitur masing-masing. Pada Gambar 5 adalah hasil plot tiga cluster. Hasil tersebut adalah membandingkan tingkatan kategori serangan pada *Source Port* dari source IP *Address*.

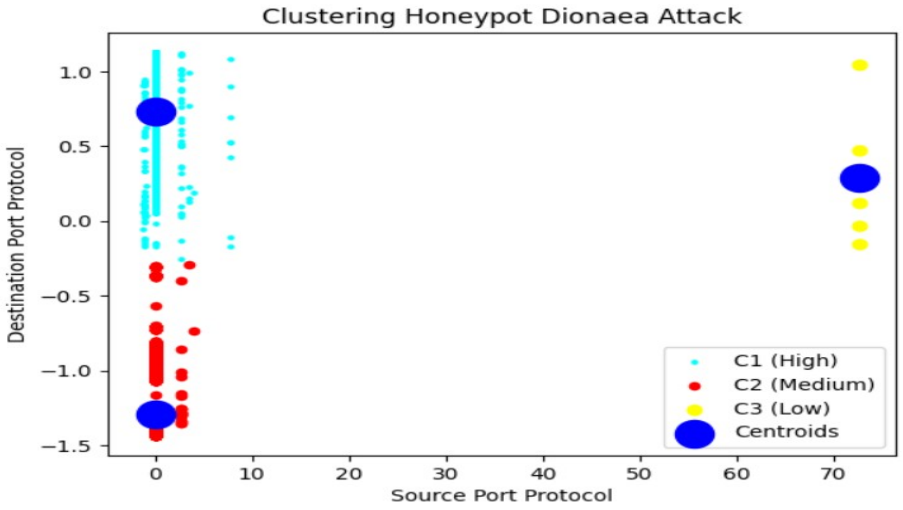



**Gambar 5. Total Serangan pada Masing-Masing Protocol TCP/UDP Dionaea**

Gambar 6 merupakan grafik pengelompokan jenis serangan *Low Interaction Honeypot* Dionaea dengan kategori menggunakan teknik K-Means. Proses pengelompokan jenis serangan menggunakan algoritma K-Means untuk membedakan 3 tingkat serangan yaitu *low*, *medium*, dan *high* dengan parameter kerapatan waktu (*time density*) serta besaran Protocol TCP/UDP. Sebaran titik berwarna *yellow* menunjukkan data dengan kondisi jenis serangan kategori *low* untuk protocol TCP, pada cluster ke-3 dengan capaian 63 *session frequency packet received*, lalu untuk sebaran titik berwarna *red* merupakan data dengan kondisi jenis serangan kategori medium untuk protocol TCP HTTPD pada cluster ke-2 dengan capaian *session frequency packet received* dengan capaian 48, sedangkan sebaran berwarna *cyan* menunjukkan data berjenis serangan kategori high, rata-rata serangan packet sebesar 27433 dalam rentang waktu *time stamp* 2023-03-07 15:08:48 – 2023-05-0804:22:35.56.

Selain itu, titik berwarna *blue* merupakan titik centroid atau titik tengah yang menunjukkan titik pusat dari tiap sebaran. Nilai jumlah kuadrat jarak cluster ke pusat cluster terdekat, ditimbang dengan bobot nilai $\mu_i$ 2573.5 dengan jumlah tahapan clustering sebanyak 2 tahapan iterasi untuk mendapatkan cluster yang sesuai. Setiap koneksi ke *honeypot*s memiliki motivasi dan target tertentu. Sejumlah besar serangan protocol TCP/UDP berfokus

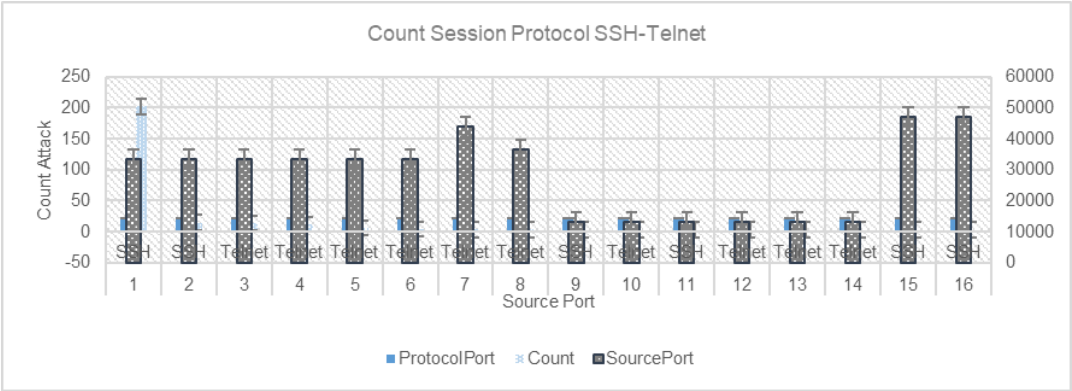
pada penyebaran *script* dan *binaries* yang mencurigakan untuk dieksekusi dan metode dapat diotomatisasi dan dan populer untuk mengeksploitasi sistem.




**Gambar 6. Pengelompokan Serangan Berdasarkan Protocol TCP/UDP Dionaea**

## 3.2 Pengelompokan *Low Interaction* v *Honeypot* Cowrie

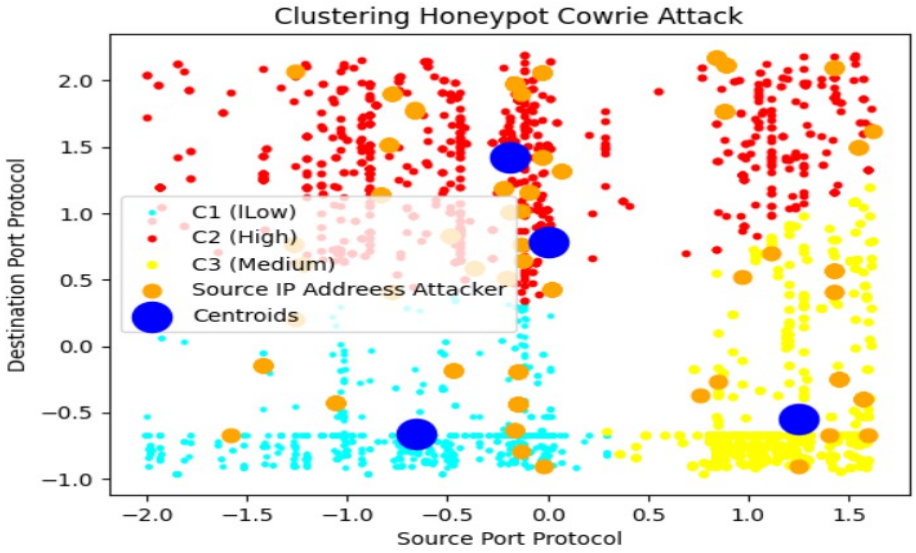
Pada Pengelompokan *Medium Interaction Honeypot* Cowrie, fokusnya adalah pada protocol SSH dan Telnet, yang mencakup interaksi paling relevan untuk analisis pola perilaku dan serangan pada perangkat layer 3 atau Router.




**Gambar 7. Total Serangan pada Masing-Masing Protocol SSH dan Telnet**

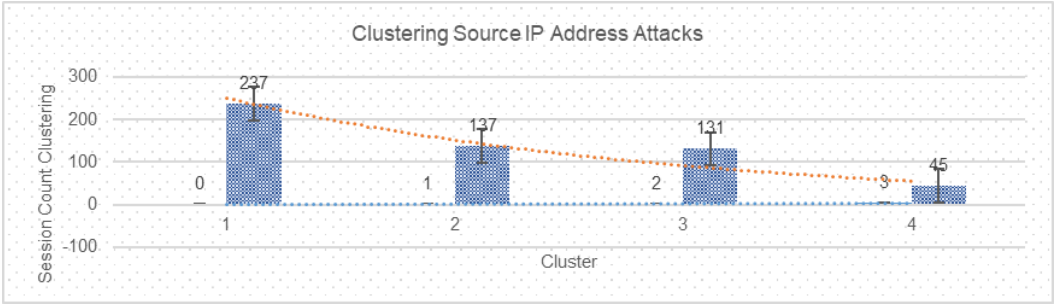
Pada Gambar 7. merupakan *raw* plot grafik penyebaran jumlah dan jenis serangan berdasarkan port SSH, Telnet pada vRouter, dari total serangan berdasarkan *source Port* dan Source IP Address. Tahap pertama adalah memplot dataset Jenis Serangan *Medium Interaction Honeypot* Cowrie. Gambar 7 menunjukkan protocol SSH menghasilkan serangan teratas sebanyak 202,06 (*Packet Sent*) yang digunakan selama eksperimen. Gambar 8 adalah grafik proses pengelompokan jenis serangan menggunakan algoritma K-Means untuk membedakan 3 tingkat serangan yaitu *low*, *medium*, dan *high* dengan parameter *timestamp* serta besaran Protocol TCP/UDP. Pembagian data *points* ke dalam masing-masing cluster, di mana terlihat semua data *points* masuk ke dalam cluster masing-masing. Terdapat sebaran titik yang menyebar dengan warna yang berbeda.

Gambar 8. pada sebaran titik berwarna *cyan* menunjukkan data dengan kondisi jenis serangan kategori *low* untuk protocol SSH dan Telnet berdasarkan *source* dan *destination port*, pada cluster ke-1 dengan *packet* data serangan sebanyak 2,85 untu protocol SSH dan Telnet. Kemudian untuk sebaran titik berwarna *yellow* merupakan data dengan kondisi jenis serangan kategori medium, pada cluster ke-3, dengan packet data serangan 2871 untuk protocol SSH dan 1,362 untuk protocol Telnet. Sedangkan sebaran berwarna *red* menunjukkan data berjenis serangan kategori *high*, rata-rata serangan paket sebesar 202,06 (*Packet Sent*) dalam rentang waktu *time stamp* 03 Maret 2023 Pukul 00:00 dan 30 Mei 2023 Pukul 23:00.



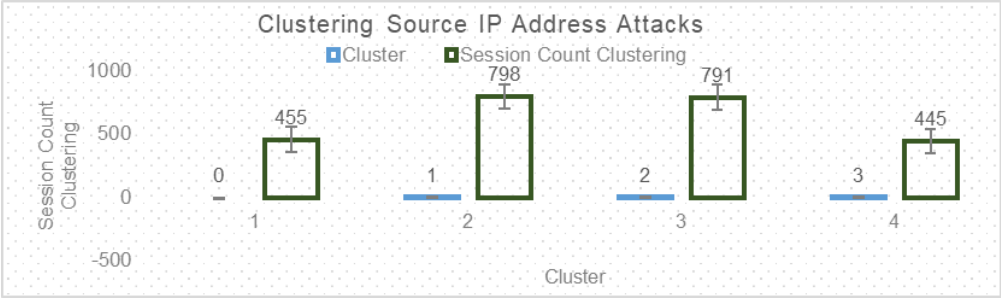

**Gambar 8. Pengelompokan Serangan Berdasarkan Protocol SSH dan Telnet (Cowrie)**

Selain itu, titik berwarna *blue* merupakan titik *centroid* atau titik tengah yang menunjukkan titik pusat dari tiap sebaran. Nilai jumlah kuadrat jarak cluster ke pusat cluster terdekat, ditimbang dengan bobot nilai μi Nilai jumlah kuadrat jarak cluster ke pusat cluster terdekat, ditimbang dengan bobot nilai μi 2573.5 dengan jumlah tahapan clustering sebanyak 4 tahapan iterasi untuk mendapatkan cluster yang sesuai. Pada Gambar 9 adalah hasil *plot* empat cluster. Hasil tersebut adalah membandingkan tingkatan kategori serangan pada *Source IP Address Public* dan *Destination Port*. Setelah pemeriksaan dari *source* IP, ada sejumlah alamat yang muncul lebih sering daripada yang lain yaitu pada *prefix* 190.39.0.0/20, dapat terlihat pembagian data points ke dalam masing-masing cluster, di mana terlihat semua data points masuk ke dalam cluster masing-masing.




**Gambar 9. Source IP Address dengan Sebagian Besar Koneksi**

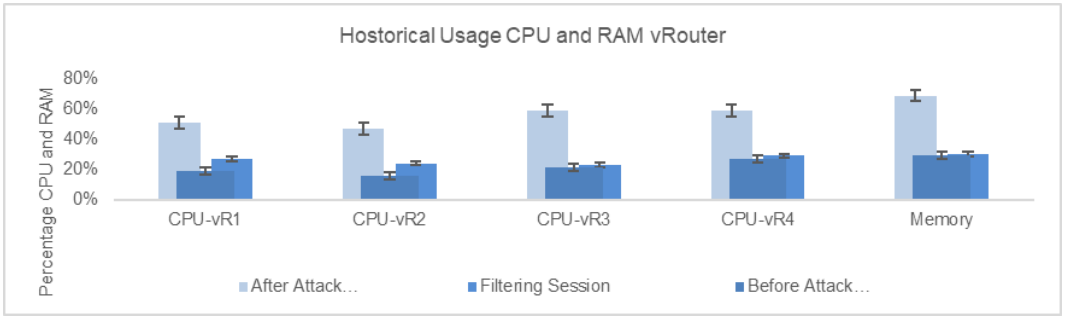
Gambar 10 merupakan Hasil pengelompokan serangan terdapat pada Cluster (C2 dan C3) dengan *packet received* 798 dan 719. yang menerima dari sumber IP *Address Public* yang termasuk kategori *high*, yaitu *Prefix Source* IP Address 117.103.2.129/32-117.103.2.135/32. Pada titik (x) atau rata-rata jumlah serangan yang didapatkan sebanyak 798 dan 791 *packet* received, sedangkan titik (y) merupakan *destination* IP Address pada rentang *prefix* 117.103.2.129/32 - 117.103.2.135/32 merupakan titik pusat IP *Address* dari jenis serangan High. Lalu pada titik (x) didapatkan dengan rata-rata jumlah serangan yang didapatkan sebanyak 2489, sedangkan cluster (C1 dan C4 merupakan *destination* IP *Address* pada rentang prefix 117.103.2.136/32 - 117.103.2.142/32 merupakan titik pusat IP *Address* dari jenis serangan Low dan Medium. Melalui hasil pengelompokan ini, dapat disimpulkan bahwa semakin besar ukuran paket (*packet received*) maka semakin tinggi serangan yang diterima vRouter dan vFarm Server.




**Gambar 10. Destination IP Address dengan Sebagian Besar Koneksi**

### 3.3 Pengujian Firewall IPTables

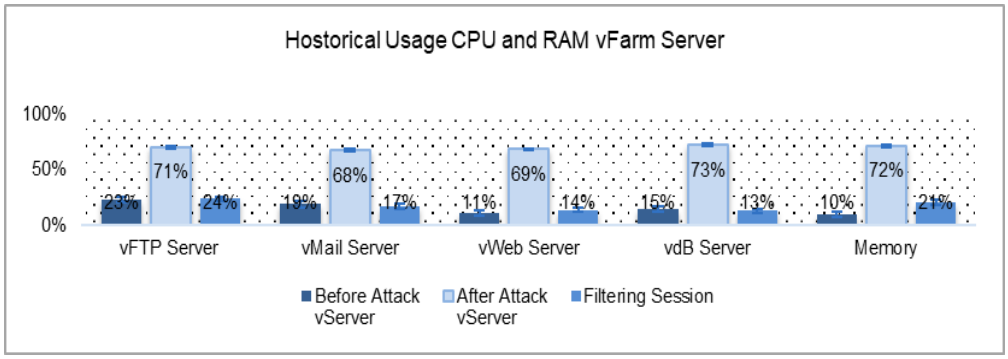
*Policy rule IPTabels* yang diterapkan berdasarkan dataset hasil dari clusterisasi teknik K-Means berfokus agar serangan yang ada berdasarkan parameter *Protocol TCP/UDP, Source Address, Destination Address, Source Port, Destination Port* dapat berkurang untuk beban pada masing-masing vFarm Server dan vRouter, termasuk penggunaan CPU dan RAM ketika eksperimen dilakukan. Fokusnya adalah dalam hal efisiensi penggunaan perangkat vFarm Server yang merupakan Sistem Operasi yang diinstall pada masing-masing *Hypervisor*, sehingga masing-masing vServer ini diserang dan dilakukan scanning dari public internet dapat diminimalisir dari segi penggunaan sumber daya.




**Gambar 11. Historical Beban Kerja CPU vRouter**

Gambar 11 menunjukkan sebelum terjadi serangan terlihat historical CPU pada perangkat vRouter menunjukkan beban kerja CPU vR1 19%, CPU vR2 16%, CPU vR3 21%, CPU vR4 27%, dan memory 29%. Pada saat terjadi serangan dari global internet pada perangkat vRouter status *load* CPU menjadi high selama menangani proses yang berjalan, pada CPU

vR1 51%, CPU vR2 47%, CPU vR3 59%, CPU vR4 59%, dan memory 69%. Setelah konvergensi dan konfigurasi rule firewall dalam status *running* yaitu pada saat proses filtering diaktifkan, terlihat bahwa historical dari load kerja CPU mengalami persentasi penurunan. pada CPU vR1 27%, CPU vR2 24%, CPU vR3 23%, CPU vR4 29%, dan memory 30%. Gambar 12 menunjukkan sebelum terjadi serangan terlihat historical CPU pada perangkat vFarm Server menunjukkan beban kerja CPU vFTP Server 23%, CPU vMail-Server 19%, CPU vWeb-Server 11%, CPU dB-Server 15%, dan *memory* 10%. Pada frekuensi terjadi serangan pada perangkat vFarm Server load kapasitas CPU menjadi high selama menangani proses penggunaan secara *real-time* pada saat eksperimen, pada perangkat vFTP Server menunjukkan beban kerja CPU vMail-Server 71%, CPU vWeb-Server 68%, CPU dB-Server 69%, CPU vMail-Server 73%, dan memory 72%. Setelah konfigurasi *filtering* firewall diaktifkan pada iterasi *rule* firewall dalam status *running* yaitu pada saat proses *filtering* dan *blocking* diaktifkan, terlihat bahwa history dari beban kerja CPU mengalami penurunan. Beban Kerja CPU vMail-Server 24%, CPU vWeb-Server 17%, CPU dB-Server 14%, CPU vSR4 13%, dan memory 21%.




**Gambar 12. Historical Beban Kerja CPU vFarm Server**

Penerapan dengan metode dan prosedur yang diadopsi menggunakan aturan *filtering* IPTables Automation dapat mengembangkan teknik mitigasi dalam riset ini, untuk mitigasi dan serangan yang terjadi dari global internet. Iterasi *rule filtering* yang diimplementasikan dapat mengurangi beban kerja perangkat yang diadopsi pada perangkat vRouter dan vFarm Server dihitung dengan rata-rata berikut: $\bar{x} - (\Sigma f\_n\, x\_n)/\Sigma f$ sehingga menghasilkan vRouter 30% dan vFarm Server 53%.

## 4. KESIMPULAN

Teknik analisis dataset *honeypot* didasarkan pada metode pembelajaran mesin dengan Teknik K-Means, tujuannya adalah untuk pengelompokan semua jenis serangan ditangkap oleh database *honeypot* dan mendeteksi serangan. Dari hasil *clustering* dapat dilihat tingkatan Protocol TCP mana saja yang sering diserang dari beberapa *attribute*, ditimbang dengan bobot nilai μi 2573.5 dengan jumlah tahapan *clustering* sebanyak 3 tahapan iterasi. Hasil dataset Dionaea sebagai vFarm Server yang dirancang menghasilkan dataset serangan (*high*) rata-rata pada protocol SMBD sebesar 27433 *Session Packet Received*. Sebaran cluster pada vRouter *Honeypot* Cowrie, rata-rata serangan packet sebesar 202,06 (*Packet Sent*). Nilai jumlah kuadrat jarak cluster ke pusat cluster terdekat, ditimbang dengan bobot nilai μi Nilai jumlah kuadrat jarak cluster ke pusat cluster terdekat, ditimbang dengan bobot nilai μi 2626.3 dengan jumlah tahapan clustering sebanyak 3 tahapan iterasi. Pengelompokan hasil dari *Rule Firewall IPTables*, yang diimplementasikan untuk perangkat vRouter dan vFarm Server, dengan teknik, IPSet file, *Limit match*, IPSet *block countries* dan automation *shell script*, yang dilakukan menunjukkan historical CPU pada perangkat vRouter

menunjukkan beban kerja CPU berkurang menjadi 28%, dan memory 39%. Pada perangkat vFarm Server menunjukkan beban kerja CPU pada masing-masing vServer berkurang menjadi 43% dan Memory (RAM) menjadi 21%.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada DRPPM Universitas Budi Luhur atas dukungan dan sumber pendanaan penelitian dengan Nomor A/UBL/DRPM/000/027/05/23 pada tahun 2023.

## DAFTAR RUJUKAN

Araujo, F., Taylor, T., Zhang, J., & Stoecklin, M. P. (2018). Cross-Stack Threat Sensing for Cyber Security and Resilience. *Proceedings - 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, DSN-W 2018*, (pp. 18–21).

Bahjat, H., Mohammed, S. N., Ahmed, W., Hamad, S., & Mohammed, S. (2020). Anomaly Based Intrusion Detection System Using Hierarchical Classification and Clustering Techniques. *Proceedings - International Conference on Developments in ESystems Engineering, DeSE*, *2020-December*, (pp. 257–262).

Ceron, M., & Scholten, C. (n.d.). *[IEEE NOMS 2020-2020 IEEE_IFIP Network Operations and Management Symposium - Budapest, Hungary (2020.4.20-2020.4.24)] NOMS 2020 - 2020 IEEE_IFIP Network Operations and Management Symposium - MikroTik Devices Lan.pdf*.

Cunha, V. A., Corujo, D., Barraca, J. P., & Aguiar, R. L. (2020). Using Linux TCP connection repair for mid-session endpoint handover: A security enhancement use-case. *2020 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2020 - Proceedings*, (pp. 174–180).

Damanik, H. A. (2020). Skema Penerapan Mekanisme SLA Dan Network Availability Untuk Customer Service Provider. *Jurnal Penelitian Pos dan Informatika*, *10*(2), 125–44.

Damanik, H. A. (2021). Fast-Recovery and Optimization Multipath Circuit Networks Environments Using Routing Policies Different Administrative Distance and Internal BGP, *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Yogyakarta, Indonesia, (pp. 299-305).

Damanik, H. A. (2022). Securing Data Network for Growing Business Vpn Architectures Cellular Network Connectivity. Acta Informatica Malaysia, *6*(1), 01–06.

El Kamel, N., Eddabbah, M., Lmoumen, Y., & Touahni, R. (2020). A Smart Agent Design for Cyber Security Based on *Honeypot* and Machine Learning. Security and Communication Networks, 2020.

Fraunholz, D., Zimmermann, M., Antón, S. D., Schneider, J., & Dieter Schotten, H. (2017). Distributed and highly-scalable WAN network attack sensing and sophisticated analysing framework based on *Honeypot* technology. *Proceedings of the 7th International Conference Confluence 2017 on Cloud Computing, Data Science and Engineering*, (pp. 416–421).

Fraunholz, D., Zimmermann, M., Anton, S. D., Schneider, J., & Dieter Schotten, H. (2017). Distributed and highly-scalable WAN network attack sensing and sophisticated analysing framework based on *Honeypot* technology. *Proceedings of the 7th International Conference Confluence 2017 on Cloud Computing, Data Science and Engineering*, (pp. 416–421).

Kosseff, J. (2020). Retorsion as a Response to Ongoing Malign Cyber Operations. *International Conference on Cyber Conflict, CYCON*, *2020-May*, (pp. 9–23).

Kashtalian, A., & Sochor, T. (2021). K-means clustering of honeynet data with unsupervised representation learning. *CEUR Workshop Proceedings*, *2853*, (pp. 439–449).

Liao, M. L., Yu, C. L., Lai, Y. C., Chiu, S. P., Chen, J. L. (2023). An Intelligent Cyber Threat Classification System, *2023 25th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, Korea, Republic of, 2023,* (pp. 189-194).

Polyakov, V. V., & Lapin, S. A. (2018). Architecture of the *Honeypot* System for Studying Targeted Attacks. *2018 14th International Scientific-Technical Conference on Actual Problems of Electronic Instrument Engineering, APEIE 2018 - Proceedings*, (pp. 202–205).

Owezarski, P. (2014). Unsupervised classification and characterization of *honeypot* attacks. *Proceedings of the 10th International Conference on Network and Service Management, CNSM 2014*, (pp. 10–18).

Rosli, N. A., Yassin, W., Faizal, M. A., & Selamat, S. R. (2019). Clustering analysis for malware behavior detection using registry data. *International Journal of Advanced Computer Science and Applications*, *10* (12), 93–102.

Sokol, P., Husak, M., & Liptak, F. (2015). Deploying *honeypot*s and honeynets: Issue of privacy. *Proceedings - 10th International Conference on Availability, Reliability and Security, ARES 2015*, (pp. 397–403).