Sistem Monitoring Log dan Deteksi Anomali Berbasis Analisis Cluster

Tugas Akhir

diajukan untuk memenuhi salah satu syarat

memperoleh gelar sarjana

dari Program Studi Teknologi Informasi (Kampus Kota Surabaya)

Fakultas Informatika

Universitas Telkom

NIM 1202200401

Ival Yudha Prawira

Program Studi Sarjana Teknologi Informasi (Kampus Kota Surabaya)

Fakultas Informatika

Universitas Telkom

Surabaya

2024

LEMBAR PENGESAHAN

SISTEM MONITORING LOG DAN DETEKSI ANOMALI BERBASIS ANALISIS CLUSTER

Log Monitoring System and Anomaly Detection System Based on Cluster Analysis

NIM : 1202200401

Ival Yudha Prawira

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian syarat memperoleh gelar pada Program Studi Sarjana Teknologi Informasi (Kampus Kota Surabaya)
Fakultas Informatika
Universitas Telkom

Surabaya, 23 Juli 2024
Menyetujui

Pembimbing I,	Pembimbing II,
Oktavia Ayu Permata, S.T., M.T.	Muhammad Adib Kamali, S.T., M.Eng
NIP: 19900006	NIP: 22970007

Ketua Program Studi
Sarjana Teknologi Informasi,

Bernadus Anggo Seno Aji, S.Kom., M.Kom.
NIP: 23929009

LEMBAR PERNYATAAN

Dengan ini saya, Ival Yudha Prawira, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul Sistem Monitoring Log dan Deteksi Anomali Berbasis Analisis Cluster beserta dengan seluruh isinya adalah merupakan hasil karya sendiri, dan saya tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang belaku dalam masyarakat keilmuan. Saya siap menanggung resiko/sanksi yang diberikan jika di kemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam buku TA atau jika ada klaim dari pihak lain terhadap keaslian karya,

Surabaya, 23 Juli 2024

Yang Menyatakan

Ival Yudha Prawira

Sistem Monitoring Log dan Deteksi Anomali Berbasis Analisis Cluster

Ival Yudha Prawira[1], Oktavia Ayu Permata[2], Muhammad Adib Kamali[3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Surabaya
[4]Divisi Digital Service PT Telekomunikasi Indonesia
[1]valfreed@students.telkomuniversity.ac.id, [2] oktapermata@telkomuniversity.ac.id, [3] adibmkamali@telkomuniversity.ac.id

Abstrak

Informasi dalam data *log website* sangat penting untuk memonitoring *web server*. *Web server* pada lingkungan kampus memiliki peran penting dalam pembelajaran, namun pada saat ini kurangnya perhatian terhadap keamanan jaringan *web server* mengakibatkan rentannya *web server* terhadap gangguan dari pihak yang tidak berwenang. Sistem *Security Infomation and Event Management (SIEM)* digunakan untuk memantau aktivitas pengunjung *website* kampus namun tidak dapat mendeteksi adanya anomali. Tugas akhir ini menambahkan fungsi SIEM yaitu mendeteksi adanya pola anomali. SIEM yang diusulkan pada tugas akhir ini dapat mendeteksi pola anomali menggunakan informasi dari *log web server.* Metode deteksi anomali traffic yang digunakan adalah metode *Clustering*. Algoritma *K-Means Clustering* adalah metode pengelompokan data yang berdasarkan kemiripan atribut untuk membentuk *cluster*. DBSCAN *(Density Based Spatial Clustering)* adalah algoritma pengelompokan berbasis kepadatan dengan Noise. Data *log web server* yang digunakan sebagai atribut *Clustering* adalah *Status Code*, URL, dan *Response Size,* URL, dan *Status Code*. Kemiripan atribut mengacu pada identifikasi pola serupa dalam besarnya *Status Code*, URL, dan *Response Size.* Setelah data dikelompokkan, data yang memiliki jarak yang signifikan dari pusat *cluster* dan data *outlier* dianggap sebagai anomali. Pengujian dilakukan dengan data *log* dari *website bis-sby.telkomuniversity.ac.id* dengan pengambilan data selama 7 hari sebanyak 21.892 data. Hasil dari deteksi pola anomali pada *log web server* mencakup *Status Code*, URL, dan *Response Size* yang diatas rata-rata dari data *log web server* selama 7 hari dengan nilai rata-rata 5.846 *byte* dalam pola akses *website* kampus.

Kata kunci: *log web server*, *k-means clustering*, DBSCAN, *anomali*.

Abstract

Information in website log data is very important for monitoring web servers. Web servers in the campus environment have an important role in learning, but at this time the lack of attention to web server network security results in the vulnerability of web servers to interference from unauthorized parties. The Security Infomation and Event Management (SIEM) system is used to monitor the activities of visitors to the campus website but cannot detect anomalies. This final project adds the SIEM function of detecting anomalous patterns. The SIEM proposed in this final project can detect anomaly patterns using information from web server logs. The traffic anomaly detection method used is the Clustering method. K-Means Clustering algorithm is a method of grouping data based on similarity of attributes to form clusters. DBSCAN (Density Based Spatial Clustering) is a density-based clustering algorithm with Noise. The web server log data used as Clustering attributes are Status Code, URL, and Response Size, URL, and Status Code. Attribute similarity refers to identifying similar patterns in the magnitude of Status Code, URL, and Response Size. After the data is clustered, data that has a significant distance from the cluster center and outlier data are considered as anomalies. Tests were conducted with log data from the bis-sby.telkomuniversity.ac.id website with 7 days of data collection totaling 21,892 data. The results of anomaly pattern detection on web server logs include Status Code, URL, and Response Size which are above the average of web server log data for 7 days with an average value of 5,846 bytes in campus website access patterns.

Keywords: log web server, k-means clustering, DBSCAN, anomaly.

1. Pendahuluan

Latar Belakang

Perlindungan dan pemantauan keamanan *website* kampus semakin penting seiring dengan peningkatan penggunaan teknologi dalam pembelajaran. Namun, saat ini kurangnya perhatian terhadap aspek keamanan jaringan. Hal ini membuat layanan *web server* di lingkungan kampus menjadi rentan terhadap gangguan yang dilakukan oleh pihak yang tidak berwenang seperti aktivitas mencurigakan, serangan siber, atau perubahan signifikan dalam pola akses ke *website* kampus.

Pada saat *website* pada *web server* mengalami gangguan, seorang administrator akan memeriksa *log* untuk mengetahui apa dan darimana serangan tersebut berasal dan data peretas akan diketahui dengan cara melihat dari

alamat IP yang dipakai untuk mengakses *website*[1]. Dibutuhkan adanya upaya untuk menjaga dan menjamin keamanan informasi terhadap layanan yang berada di *web server*[2]. Pentingnya memiliki pencatatan *log web server* secara *real-time* yang mencatat semua aktivitas layanan yang berjalan di *web server*[3].

Untuk menganalisis data *log web server*, diperlukannya *log monitoring system*[4]. *Security Information Event and Managemen (SIEM)* merupakan sistem informasi yang digunakn untuk mengumpulkan data *log* yang nantinya menghasilkan keluaran visualisasi *log* monitoring untuk mempermudah pembacaan informasi *log*[5]. Pemilihan algoritma merupakan hal yang dasar yang diperlukan dalam mengimplementasikan teknologi *Security Informasion Event and Management (SIEM)*[5].

Banyak algoritma yang bisa diterapkan dalam mengelompokkan data, salah satunya yaitu algoritma *Clustering*. Pemilihan metode *Clustering* berdasarkan kemampuannya untuk mengelompokkan data. Algoritma *K-Means Clustering* digunakan untuk mengelompokkan data berdasarkan kedekatan satu sama lain sesuai jarak *Euclidean* [3]. *K-Means Clustering* termasuk dalam *unsupervised-machine learning*, metode untuk membagi satu set data menjadi beberapa kelompok yang memiliki kemiripan fitur [6]. Algoritma DBSCAN mengelompokkan data berdasarkan kepadatan. DBSCAN mampu mengidentifikasi kelompok dengan berbagai ukuran dan bentuk serta mendeteksi *Noise* dalam sejumlah besar data yang mengandung *Noise* dan *Outlier* [16].

Oleh karena itu, Tugas Akhir ini bertujuan untuk mengatasi masalah tersebut dengan mengembangkan sistem yang mampu memantau aktivitas pengunjung *website* kampus dan mendeteksi pola anomali. Sistem ini menggunakan metode *Clustering* untuk mengelompokkan data *log* berdasarkan atributnya seperti *Status Code*, URL, dan *Response Size*. Digunakannya algoritma *K-Means Clustering* dan DBSCAN dengan tujuan untuk menguji efektivitasnya dalam mendeteksi anomali pada *web server*. Pada tugas akhir ini akan mengevaluasi apakah *K-Means Clustering* dan DBSCAN mampu mengungkapkan pola yang mencurigakan dalam *log web server*. Hasil dari Tugas Akhir ini dapat membantu mengidentifikasi pola yang tidak biasa dalam data *log web server* dan menguji efektivitas dari metode *Clustering* dengan matriks evaluasi *Silhouette Score*.

Topik dan Batasannya

Pada tugas akhir ini memfokuskan pada analisis data *log web server* untuk meningkatkan keamanan jaringan dan melindungi data sensitif, terutama dalam konteks lingkungan kampus Universitas Telkom Surabaya dengan penelitian yang difokuskan pada *web server bis-sby.telkomuniversity.ac.id*. Bagaimana analisis data *log web server* dapat membantu mengidentifikasi aktivitas yang tidak biasa, serta bagaimana algortima *K-Means Clustering* dan algoritma DBSCAN dapat diterapkan untuk mengelompokkan data *log web server* dan mengidentifikasi pola anomali.

Namun dalam tugas akhir ini terdapat batasan yang perlu diperhatikan. Tugas Akhir ini dibatasi pada deteksi anomali dalam data *log web server*, khususnya pada aktivitas *Status Code*, URL, dan *Response Size* yang diatas rata-rata dari data *log web server* selama 7 hari. Selain itu, tidak ada kemampuan untuk melakukan monitoring secara *real-time* karena keterbatasan akses langsung ke *server* kampus Universitas Telkom Surabaya. Dengan memperhatikan batasan ini, diharapkan dapat memberikan kontribusi yang signifikan dalam meningkatkan keamanan jaringan *web server* dengan memonitoring data *log* dan mendeteksi anomali di lingkungan kampus Universitas Telkom Surabaya.

Tujuan

Tujuan dari tugas akhir ini untuk merancang dan mengimplementasikan sistem monitoring untuk mengumpulkan dan menganalisis data *log* dari *web server* selama periode waktu 7 hari. Mengidentifikasi pola atau aktivitas yang tidak biasa dalam aktivitas pengunjung *website* kampus Universitas Telkom Surabaya menggunakan algoritma *K-Means Clustering* dan algoritma DBSCAN berdasarkan jarak antar *cluster*, melabeli data anomali berdasarkan *Status Code*, URL, dan *Response Size* yang diatas rata-rata dari data *log web server* selama 7 hari. Dengan melabeli hasil deteksi anomali, dapat membantu administrator jaringan dalam mengidentifikasi jenis anomali yang ada.

Selain itu, tugas akhir ini bertujuan untuk mengetahui performa *Clustering* menggunakan matriks evaluasi *Silhouette Score* untuk memastikan akurasi dan efektivitas pengelompokan data *log web server*. Dengan demikian, tugas akhir ini dapat memberikan kontribusi yang berarti dalam mengamankan infrastruktur teknologi informasi di lingkungan kampus serta menjaga kerahasiaan dan integritas data yang tersimpan.

Organisasi Tulisan

Pada tugas akhir ini, dibuat dan disusun dengan organisasi tulisan sebagai berikut:

- Pendahuluan, Pada bagian ini menjelaskan mengenai apa saja yang mendasari tugas akhir ini, tujuan, serta apa tugas akhir ini.
- Studi Terkait, Pada bagian ini menjelaskan mengenai cara tugas akhir ini dibuat dengan menggunakan beberapa referensi yang digunakan sebagai pengembangan dari tugas akhir ini.
- Sistem yang Dibangun, Pada bagian ini menjelaskan mengenai penyusunan sistem yang dibangun dan diimplementasikan pada tugas akhir ini.

- Evaluasi, Pada bagian ini menjelaskan mengenai hasil dan proses analisa dari hasil yang diperoleh dari tugas akhir ini.
- Kesimpulan, Pada bagian ini menjelaskan mengenai apa saja yang dapat disimpulkan dari hasil tugas akhir ini.

2. Studi Terkait

2.1 Web Server

Website adalah suatu halaman *web* yang saling berhubungan yang umumnya berisikan informasi yang disediakan secara perorangan, kelompok, atau organisasi [7]. *Website* biasanya ditempatkan pada *server* yang terhubung oleh jaringan. *Server* adalah perangkat utama dalam sebuah sistem komunikasi jaringan yang berperan sebagai penyedia layanan dalam jaringan [3]. *Web server* adalah tempat untuk mendapatkan halaman *web* dan data yang berhubungan dengan *website*, sehingga data dapat diakses dan dilihat oleh pengguna [7]. *Web server* berfungsi sebagai pusat kontrol dalam memproses permintaan yang diterima dari *browser*. *Access log* adalah *file* yang berperan untuk mencatat semua akses yang dilakukan terhadap *web server* [18]. Data *log* ini berada di komputer *web server*, digunakan untuk melakukan pelacakan dan analisis terhadap perilaku pengguna yang mengakses *website* [1]. Data yang ada dalam *log web server* berisi informasi tentang *remotehost* (*ip address*), *rfc931*, *authuser*, *timestamp*, *request*, status, dan *size* serta tipe agen yang digunakan dalam mengakses *website* [8].

```
185.229.118.45 - - [27/Aug/2023:00:04:06 +0700] "GET /psgls/public/mostborrow HTTP/1.1"" 200 853 "-" "-""
185.229.118.45 - - [27/Aug/2023:00:04:06 +0700] "GET /psgls/public/mostborrow HTTP/1.1"" 200 853 "-" "-""
154.28.229.170 - - [27/Aug/2023:16:54:43 +0700] "GET / HTTP/1.1"" 404 1238 "-" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_5_8; en-US)
AppleWebKit/532.0 (KHTML, like Gecko) Chrome/4.0.203.0 Safari/532.0""
154.28.229.170 - - [27/Aug/2023:16:54:43 +0700] "GET / HTTP/1.1"" 404 1238 "-" "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_5_8; en-US)
AppleWebKit/532.0 (KHTML, like Gecko) Chrome/4.0.203.0 Safari/532.0""
178.249.214.10 - - [27/Aug/2023:16:54:48 +0700] "GET / HTTP/1.1"" 404 1238 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/110.0.0.0 Safari/537.36""
178.249.214.10 - - [27/Aug/2023:16:54:48 +0700] "GET / HTTP/1.1"" 404 1238 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/110.0.0.0 Safari/537.36""
139.162.7.175 - - [27/Aug/2023:16:54:58 +0700] "HEAD / HTTP/1.1"" 404 0 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_4) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36""
139.162.7.175 - - [27/Aug/2023:16:54:58 +0700] "HEAD / HTTP/1.1"" 404 0 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_4) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36""
```

Gambar 1. Data *Log Access Web server*

2.2 Algoritma K-Means Clustering

Algoritma *K-Means Clustering* adalah algoritma yang digunakan untuk membentuk kelompok dan menentukan sistem deteksi dengan mengelompokkan data berdasarkan kesamaan statistiknya dengan data lainnya, dimana nilai statistik tertinggi ditentukan dalam satu kelompok *cluster* dan nilai statistik yang lebih rendah terdapat dalam kelompok yang berbeda [9]. *K-Means Clustering* adalah metode analisis yang menggunakan data *mining* untuk mengelompokkan datanya [10]. Metode *K-Means Clustering* menggunakan algoritma sebagai berikut:

1. Tentukan *k* dengan menggunakan *Elbow Methods* untuk memilih jumlah *k-cluster* yang akan digunakan. *Elbow Methods* digunakan untuk menghasilkan informasi dalam menentukan jumlah *cluster* yang akan membentuk suatu sudut pada titik tertentu [11].
2. Penentuan *k Centroid* (titik pusat *cluster*) awal dilakukan dengan mengambil data secara acak sebanyak jumlah *k-cluster* sebagai pusat *cluster*.
3. Menghitung jarak objek ke masing-masing *Centroid* dari masing-masing *cluster* menggunakan *Euclidean distance*.

$$d\,(x,y)\;=\;\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$$

   x_i, y_i = Variabel pada objek x ke-1 dan y ke-1
   n = Banyak objek
4. Tempatkan setiap objek ke dalam *Centroid* yang memiliki jarak terdekat.
5. Lakukan iterasi, lalu tentukan posisi *Centroid* baru dengan menggunakan persamaan pada poin kedua [8]. Gunakan hasil nilai rata-rata dari setiap anggota pada masing-masing *cluster* dengan menggunakan rumus:

$$Centroid_k = \frac{1}{n_k}\sum d_i$$

   n_k = Jumlah data pada *cluster k*
   d_i = Jumlah dari nilai jarak pada masing-masing *cluster k*
6. Ulangi langkah ke 3 jika posisi *Centroid* baru tidak sama.

2.3 Algoritma DBSCAN

Algoritma DBSAN adalah algoritma untuk mengelompokkan data berdasarkan dengan kepadatan yang cukup tinggi dan mengidentifikasi data dengan kepadatan cukup rendah terdeteksi sebagai *outlier* [17]. DBSCAN mendeteksi *cluster* dan secara otomatis menentukan parameter *epsilon* dengan akurat, guna menemukan parameter

input dan *cluster* yang memiliki *outlier* yang bervariasi, oleh karena itu *Eps* dan *MinPts* adalah dua parameter input penting bagi algoritma DBSCAN [16]. Metode ini menganggap *cluster* suatu area yang isinya objek yang padat yang dipisahkan area yang mempunyai kepadatan rendah.

2.4 Node.js dan Express.js

Node.JS adalah sistem yang didesain untuk mengembangkan aplikasi *web* yang berjalan di server [12]. *Node.js* memudahkan pengembang *website* berbasis *JavaScript* di sisi server berupa *runtime* yang menyediakan *built-in HTTP* server di dalamnya, sehingga pengembang bisa menjalankan *website* tanpa perlu menginstall aplikasi *web server* secara terpisah [13]. *Node.js* termasuk dalam lingkungan *runtime javascript* yang mencapai *latency* terendah dan *throughput* yang tinggi dengan mengambil pendekatan "*non-blocking*" untuk permintaan [14].

Express.JS adalah *framework* dari *Node.JS* yang berguna dalam mempermudah pembuatan aplikasi berbasis *Node.JS* dengan menggunakan *design pattern* yang dapat disesuaikan dan sangat fleksibel serta *framework* ini sangat ringan dan cocok dalam pembuatan aplikasi *web* dan *API* [12]. *Express* adalah modul *Node* yang menyediakan *framework* minimal dan fleksibel untuk aplkasi *web*. *Express* menyediakan fungsi-fungsi untuk ditambahkan ke modul *Node* sehingga pengembangan aplikasi *Node.js* menggunakan *Express.js* jauh lebih mudah dibandingkan dengan menggunakan model *Node.js* yang asli [15].

3. Sistem yang Dibangun

3.1 Unggah Data Log ke Database

Data *log web server* pada Gambar 2 menggunakan data *log Access* dari *website bis-sby.telkomuniversity.ac.id* diambil pada server Telkom University Surabaya melalui Pusat Teknologi Informasi (PUTI). Data *log* awalnya berbentuk ekstensi **.log*, diunggah ke *database* yang bertujuan untuk memudahkan proses monitoring serta pengolahan data oleh *Clustering*. Data *log* dipisahkan berdasarkan atribut yang ada didalamnya dengan hasil pemisahan atribut pada Gambar 3.

```
127.0.0.1 - - [25/Mar/2024:13:42:10 +0700] "POST /wp-cron.php?doing_wp_cron=1711348930.7318229675292968750000 HTTP/1.1" 200 31 "-" "z8SBAE8YVHGz"
10.220.34.198 - - [25/Mar/2024:13:42:10 +0700] "GET // HTTP/1.1" 301 5 "-" "Mozilla/5.0 (compatible; PRTG Network Monitor (www.paessler.com); Windows)"
10.220.34.198 - - [25/Mar/2024:13:42:12 +0700] "GET / HTTP/1.1" 200 167755 "-" "Mozilla/5.0 (compatible; PRTG Network Monitor (www.paessler.com); Windows)"
127.0.0.1 - - [25/Mar/2024:13:43:10 +0700] "POST /wp-cron.php?doing_wp_cron=1711348990.8759179115295410156250 HTTP/1.1" 200 31 "-" "kXghaJPVJZv0"
10.220.34.198 - - [25/Mar/2024:13:43:10 +0700] "GET // HTTP/1.1" 301 5 "-" "Mozilla/5.0 (compatible; PRTG Network Monitor (www.paessler.com); Windows)"
10.220.34.198 - - [25/Mar/2024:13:43:13 +0700] "GET / HTTP/1.1" 200 167740 "-" "Mozilla/5.0 (compatible; PRTG Network Monitor (www.paessler.com); Windows)"
127.0.0.1 - - [25/Mar/2024:13:44:10 +0700] "POST /wp-cron.php?doing_wp_cron=1711349050.7369349002838134765625 HTTP/1.1" 200 31 "-" "EwobXb7AFtTK"
10.220.34.198 - - [25/Mar/2024:13:44:10 +0700] "GET // HTTP/1.1" 301 5 "-" "Mozilla/5.0 (compatible; PRTG Network Monitor (www.paessler.com); Windows)"
10.220.34.198 - - [25/Mar/2024:13:44:12 +0700] "GET / HTTP/1.1" 200 167758 "-" "Mozilla/5.0 (compatible; PRTG Network Monitor (www.paessler.com); Windows)"
127.0.0.1 - - [25/Mar/2024:13:45:10 +0700] "POST /wp-cron.php?doing_wp_cron=1711349110.6875720024108886718750 HTTP/1.1" 200 31 "-" "t8IbjxySosQr"
10.220.34.198 - - [25/Mar/2024:13:45:10 +0700] "GET // HTTP/1.1" 301 5 "-" "Mozilla/5.0 (compatible; PRTG Network Monitor (www.paessler.com); Windows)"
10.220.34.198 - - [25/Mar/2024:13:45:12 +0700] "GET / HTTP/1.1" 200 167777 "-" "Mozilla/5.0 (compatible; PRTG Network Monitor (www.paessler.com); Windows)"
127.0.0.1 - - [25/Mar/2024:13:45:39 +0700] "POST /wp-cron.php?doing_wp_cron=1711349139.6978309154510498046875 HTTP/1.1" 200 31 "-" "2cypTi2ll2Ws"
157.55.39.53 - - [25/Mar/2024:13:45:39 +0700] "GET /wp-json/wp/v2/tags/64 HTTP/1.1" 200 652 "-" "Mozilla/5.0 AppleWebKit/537.36 (KHTML, like Gecko; compatib
```

Gambar 2. Data *Log Access website bis-sby.telkomuniversity.ac.id*

Host	Tanggal	RequestMethod	Access	KodeJam	URL	HttpVersion	StatusCode	ResponseSize	Referer	UserAgent	OS	Browser
94.156.68.85	2024-03-02	GET	17:55:57	+07:00	//wp-content/plugins/classic-editor/	HTTP/1.1"	403	548	www.goog…	Mozilla/5.0 (Linux; …	Linux;Android7.0;SM-…	Gecko) - Version/4.
94.156.68.85	2024-03-02	GET	17:56:03	+07:00	//wp-content/cache/	HTTP/1.1"	200	0	www.goog…	Mozilla/5.0 (Linux; …	Linux;Android7.0;SM-…	Gecko) - Version/4.
94.156.68.85	2024-03-02	GET	17:56:13	+07:00	//images/	HTTP/1.1"	301	5	www.goog…	Mozilla/5.0 (Linux; …	Linux;Android7.0;SM-…	Gecko) - Version/4.
127.0.0.1	2024-03-02	POST	17:56:17	+07:00	/wp-cron.php?doing_wp_cron=1709376977.366008…	HTTP/1.1"	200	31	-	tN22S4Gz63lu"		
94.156.68.85	2024-03-02	GET	17:56:17	+07:00	/images/	HTTP/1.1"	404	20.212	www.goog…	Mozilla/5.0 (Linux; …	Linux;Android7.0;SM-…	Gecko) - Version/4.
94.156.68.85	2024-03-02	GET	17:56:27	+07:00	//assets/	HTTP/1.1"	301	5	www.goog…	Mozilla/5.0 (Linux; …	Linux;Android7.0;SM-…	Gecko) - Version/4.
94.156.68.85	2024-03-02	GET	17:56:45	+07:00	//vendor/	HTTP/1.1"	301	5	www.goog…	Mozilla/5.0 (Linux; …	Linux;Android7.0;SM-…	Gecko) - Version/4.
94.156.68.85	2024-03-02	GET	17:56:47	+07:00	/vendor/	HTTP/1.1"	404	20.223	www.goog…	Mozilla/5.0 (Linux; …	Linux;Android7.0;SM-…	Gecko) - Version/4.
94.156.68.85	2024-03-02	GET	17:56:59	+07:00	//files/	HTTP/1.1"	301	5	www.goog…	Mozilla/5.0 (Linux; …	Linux;Android7.0;SM-…	Gecko) - Version/4.
94.156.68.85	2024-03-02	GET	17:57:05	+07:00	/files/	HTTP/1.1"	404	20.204	www.goog…	Mozilla/5.0 (Linux; …	Linux;Android7.0;SM-…	Gecko) - Version/4.
94.156.68.85	2024-03-02	GET	17:57:13	+07:00	//wp-content/uploads/2024/1/	HTTP/1.1"	301	5	www.goog…	Mozilla/5.0 (Linux; …	Linux;Android7.0;SM-…	Gecko) - Version/4.
94.156.68.85	2024-03-02	GET	17:57:16	+07:00	/wp-content/uploads/2024/1/	HTTP/1.1"	404	20.561	www.goog…	Mozilla/5.0 (Linux; …	Linux;Android7.0;SM-…	Gecko) - Version/4.

Gambar 3. Data *Log* dalam *Database*

3.2 Persiapan data

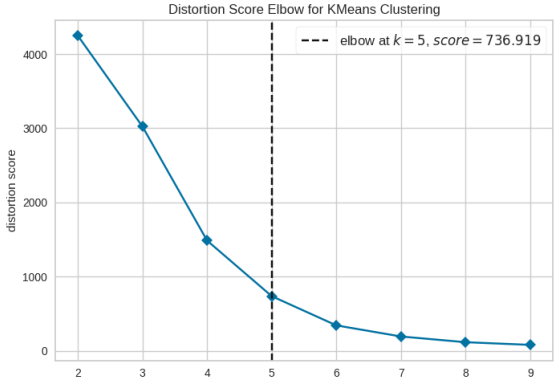
Setelah data diunggah ke *database*, data diolah dengan program *python* dimana data yang diolah difilter dengan rentang tanggal 7 hari dari data terbaru. Atribut atau fitur yang digunakan untuk melakukan *Clustering* yaitu *Status Code*, URL, dan *Response Size*. Tahap selanjutnya yaitu *preprocessing* data dengan melakukan beberapa transformasi pada data seperti dilakukan *One Hot Encoding* pada atribut *Status Code* untuk mengubah nilai kategorikal menjadi bentuk vektor biner, untuk atribut URL dilakukan penghitungan panjang karakter untuk setiap URL pada dataset, pengisian nilai yang hilang dengan nilai rata-rata, dan normalisasi atribut URL dan *Response Size* menjadi skala yang lebih sesuai untuk analisis *Clustering*. Hal ini bisa dilihat pada Tabel 1.

Tabel 1. Hasil *Preprocessing* Data

URL_length	Respon_size	StatusCode_200	StatusCode_201	StatusCode_500
0.001742	0.003702	0.0	0.0	1.0
0.102787	0.000038	1.0	0.0	0.0
0.000000	0.212597	1.0	0.0	0.0
0.102787	0.000038	1.0	0.0	0.0
0.102787	0.000038	1.0	0.0	0.0

3.3 Proses K-Means Clustering dan Pemberian Label

Penggunaan *Elbow Methods* digunakan untuk menentukan nilai *k* yang optimal disajikan pada grafik pada Gambar 4 menghasilkan indikasi nilai *k* yang optimal.



Gambar 4. Grafik *Elbow Methods*

Setelah diperoleh nilai *k* yang optimal, maka dilakukan proses *K-Means Clustering* dengan menggunakan input *k* optimal. Namun, untuk mengetahui kualitas yang dihasilkan *K-Means Clustering*, perlu dilakukan evaluasi matriks seperti *Silhouette Score*. Evaluasi matriks membantu dalam mengukur seberapa baik *K-Means* berhasil mengelompokkan data ke dalam *cluster* yang berbeda. Setelah proses *Clustering*, dilakukannya deteksi anomali berdasarkan hasil dari rata-rata tiap *cluster* yang dibandingkan dengan rata-rata *Response Size* selama 7 hari. Data dianggap sebagai anomali jika rata-rata dari tiap *cluster* lebih tinggi daripada rata-rata *Response Size* kemudian diberi label berdasarkan kriteria seperti Anomali. Data anomali kemudian dikelompokkan berdasarkan alamat *IP* untuk memperoleh gambaran yang lebih jelas.

Tabel 2. Hasil Interpretasi Setiap *Cluster* dan Pelabelan Atribut *Response Size*

Cluster	Count	Mean	Min	Max	Label
0	5.003	3.605	31	78.819	Normal
1	14.206	7.526	0	811.672	Anomali
2	845	7	0	162	Normal
3	1.152	1.317	0	1.997	Normal
4	686	2.185	0	5.621	Normal

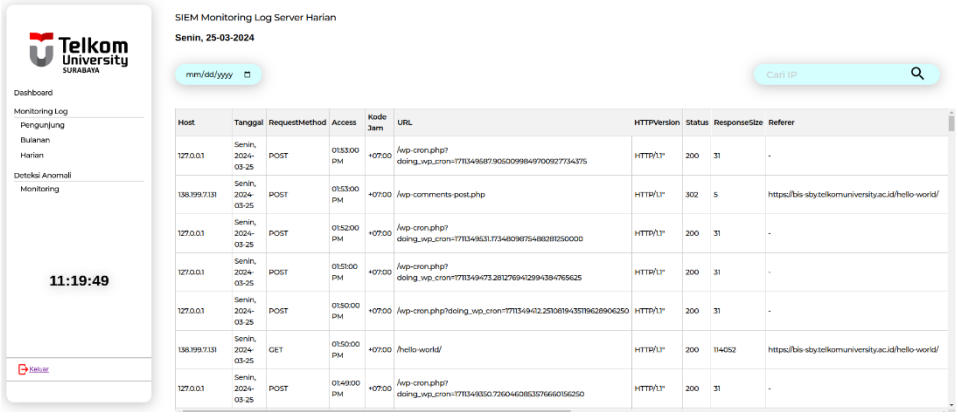
3.4 Proses DBSCAN

Proses DBSCAN dimulai dengan menentukan parameter yang dibutuhkan seperti nilai *epsilon* (*eps*) dan jumlah tetangga terdekat (*min_sample*) yang harus dipertimbangkan berdasarkan atribut *Status Code*, URL, dan *Response Size*. *Epsilon* merupakan jarak maksimum yang diperbolehkan antara dua titik data agar dapat dianggap sebagai bagian dari satu *cluster* yang sama. Nilai *epsilon* ditentukan dengan menggunakan metode *K-Nearest Neighbors* (KNN). Dengan metode ini, jarak pada setiap titik data ke tetangga terdekatnya dihitung, kemudian nilai *epsilon* ditentukan berdasarkan hasil perhitungan tersebut. DBSCAN menggunakan nilai *epsilon* tersebut untuk mengelompokkan titik-titik data ke dalam *cluster*. *Cluster* yang terbentuk dapat memiliki ukuran dan bentuk yang beragam sesuai dengan variasi jarak antar titik data.

Kemudian dilakukannya deteksi anomali. Titik-titik yang berada di luar radius *epsilon* dianggap sebagai *noise* atau *outlier*. Titik-titik yang saling berdekatan dan memenuhi kriteria jarak dan jumlah tetangga terdekat membentuk *cluster* yang berbeda-beda. *Cluster* yang dihasilkan kemudian diberi label dan pada setiap titik data akan dilabeli apakah termasuk dalam kategori Normal atau merupakan Anomali.

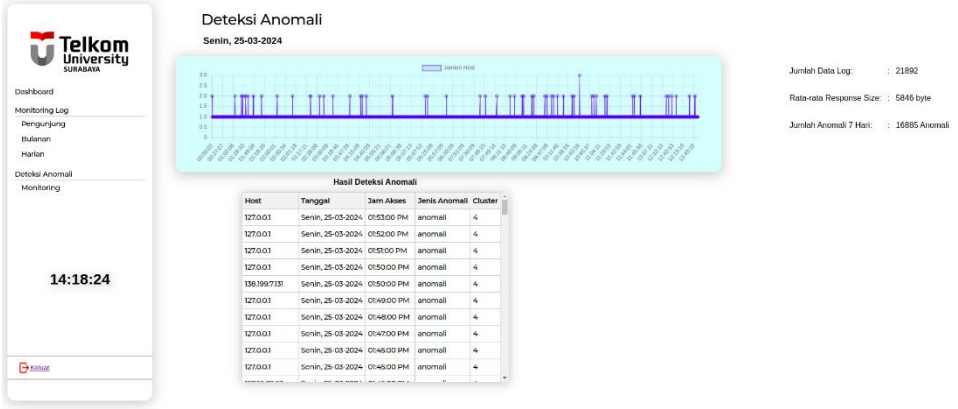
3.5 Implementasi Website

Implementasi sistem monitoring *log* ini berbasis *website* yang terkoneksi dengan *database* dan menampilkan data secara *real-time*. Sistem ini didesain menggunakan *Node.js* dengan *framework Express.js*. Sistem ini memungkinkan data *log web server* diakses secara *real-time* melalui antarmuka *web*, hal ini bisa dilihat pada Gambar 5.



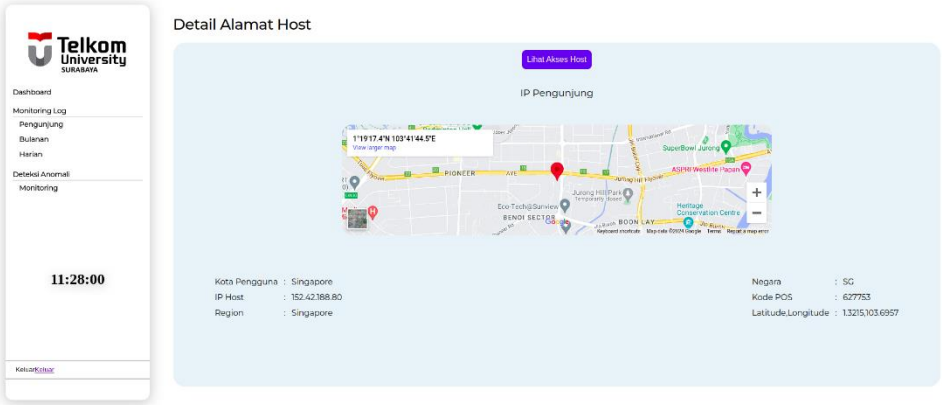
Gambar 5. Tampilan Halaman Monitoring *Log*

Hasil deteksi anomali dapat ditampilkan berdasarkan hari terkini. Hasil deteksi anomali kemudian dipresentasikan melalui halaman *website* dengan visualisasi yang terdapat pada Gambar 6. Algortima yang digunakan pada halaman ini adalah *K-Means Clustering*.

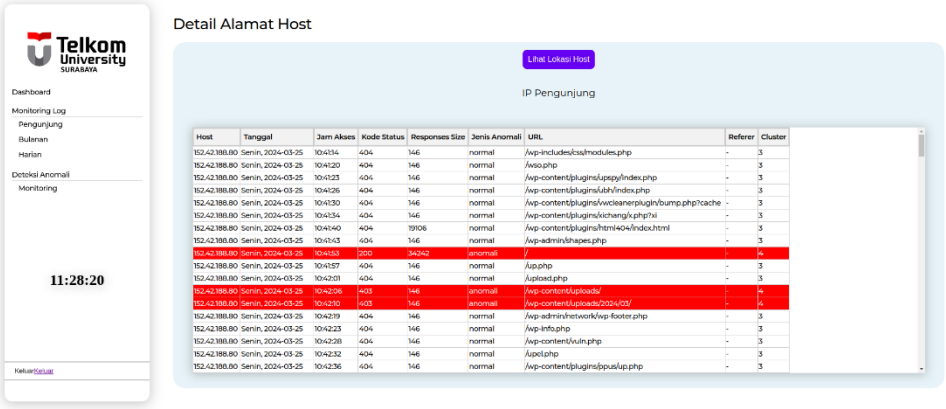


Gambar 6. Tampilan Halaman Deteksi Anomali

Selanjutnya pengguna dapat mengakses informasi detail tentang *IP* yang terlibat dalam anomali termasuk informasi lokasi geografisnya dan konten *log* yang terkait dengan anomali. Halaman ini dapat dilihat pada Gambar 7 dan Gambar 8.



Gambar 7. Tampilan Halaman Detail Informasi *IP*



Gambar 8. Tampilan Halaman Detail Informasi *IP* Bagian Isi *Log*

4. Evaluasi

Hasil pengujian menggunakan data *log* yang diambil langsung pada server Universitas Telkom Surabaya sebanyak 21.892 data *log web server* dengan pengambilan data selama 7 hari dengan nilai rata-rata *Response Size* sebesar 5.846 *byte*.

Pada proses *K-Means*, penggunaan *Elbow Method* dalam menentukan nilai *k* yang optimal sebagaimana tergambar pada Gambar 4 memberikan nilai optimal berada pada *k*=5. Kemudian dilakukannya evaluasi matriks *Silhouette Score* pada data setelah proses *Clustering* dengan nilai 0,94478. Kemudian hasil dari *Clustering* berhasil melabeli data, dengan kriteria label Anomali dengan banyaknya data 14.206 data Anomali dan 7.686 data Normal. Hal ini bisa dilihat pada Tabel 3.

Tabel 3. Data *Log* Setelah Proses *Clustering*

Host	Status Code	URL	Response Size	Validasi Manual	Label K-Means	Label DBSCAN
138.199.7.131	302	/wp-comments-post.php	5	Normal	Normal	Normal
202.67.46.238	201	/wp-json/wp/v2/media/4768/edit?_locale=user	5327	Normal	Normal	Anomali
138.199.7.131	200	/hello-world/	114052	Anomali	Anomali	Normal
202.148.31.178	200	/	34236	Anomali	Anomali	Normal
165.227.125.74	404	/fm/dialog.php	146	Normal	Normal	Normal
128.199.86.161	502	/wp-admin/network/admin.php	552	Normal	Normal	Anomali

Pada proses DBSCAN, nilai *epsilon* yang didapatkan dari metode KNN sebesar 0,47524 dengan nilai min_sample 20. Kemudian hasil dari perhitungan jarak antar data berhasil mendeteksi *Outlier* dan melabeli data dengan kriteria Anomali dengan banyaknya 17 data dan 21.875 data berlabel Normal, hal ini bisa dilihat pada Tabel 3. Kemudian dilakukannya evaluasi matriks *Silhouette Score* pada data setelah proses *Clustering* dengan nilai 0,97543.

Hasil *Clustering* sangat optimal pada algoritma *K-Means Clustering* dan algoritma DBSCAN, hal ini bisa dibuktikan melalui evaluasi matriks *Silhouette Score* dengan nilai 0,94478 pada algoritma *K-Means Clustering* dan 0,97543 pada algoritma DBSCAN. *Silhouette Score* sebesar 0,9 menandakan bahwa *Clustering* berjalan sangat baik, dengan *cluster* yang terpisah dengan jelas dan objek yang berada didalam *cluster* sangat mirip satu sama lain dibandingkan dengan objek di *cluster* lainnya.

Hasil dari *Clustering* dengan menggunakan algoritma *K-Means Clustering* dan algoritma DBSCAN divalidasi secara manual apakah data tersebut benar-benar anomali atau tidak dan juga memvalidasi apakah data yang berlabelkan Normal benar-benar Normal atau sebenarnya merupakan data Anomali. Hasil dari validasi menunjukkan bahwa algoritma *K-Means Clustering* mampu membagi data *log web server* antara data Normal dan data Anomali dibandingkan dengan algoritma DBSCAN dengan hasil yang kurang tepat. Hal ini bisa dilihat pada Tabel 3.

Dengan hasil pengujian yang telah dilakukan sesuai dengan tujuan yang dituju yaitu algoritma *Clustering* dapat mengidentifikasi anomali pada data *log web server*.

5. Kesimpulan

Dengan hasil pengujian dan analisis yang sesuai dengan tujuan tugas akhir, penulis meyakini bahwa sistem SIEM dengan metode *K-Means Clustering* mampu mendeteksi anomali sebanyak 14.206 anomali dibandingkan

dengan algoritma DBSCAN sebanyak 17 anomali dengan rentang waktu analisis selama 7 hari. Berdasarkan hasil tersebut dapat digunakan oleh administrator jaringan Universitas Telkom Surabaya untuk pemantauan dan deteksi anomali yang lebih efektif pada *log web server* kampus.

Untuk pengembangan penelitian selanjutnya, penulis merekomendasikan beberapa hal, yaitu peningkatan analisis yang lebih luas untuk mendapatkan pemahaman yang lebih mendalam tentang pola anomali pada *log web server* dan pengembangan mekanisme otomatisasi dalam pelabelan jenis anomali.

Daftar Pustaka

[1] Yogi, Ruslianto I, and Bahri S. 2019. Analisa Log Web Server Untuk Mengetahui Pola Perilaku Pengunjung Website Menggunakan Teknik Regular Expressions. Jurnal Komputer dan Aplikasi. 7:1 120.

[2] Wibawa G H P, Sasmita I G M A, Raharja I M S. 2020. Analisis Data Log Honeypot Menggunakan Metode K-Means Clustering. Jurnal Ilmiah Merpati. 8:1 13.

[3] Ardiyasa I W. 2020. Penerapan K-Means Clustering untuk klasifikasi Serangan Cyber pada Syslog File. JSI (Jurnal Sistem dan Informatika). 14:2 144.

[4] Anam F C, Sasmita G M A, Pratama I P A E. 2023. Implementation of Security and Event Management (SIEM) for Monitoring IT Assets Using Alienvault OSSIM (Case Study: Udayana University Information Resources Unit). JITTER. 4:3 1.

[5] Abidian W, Setiawan M A. Implementasi Splunk dalam Membangun Security Information and Event Management Berdasarkan Log Firewall (studi kasus: Jaringan UII). 1.

[6] Ma'ali A A, Girinoto, Ghiffari M N, Hadiprakoso R B. 2022. Analisis Log Web Server dengan Pendekatan Algoritma K-Means Clustering dan Feature Importance. Jurnal Info Kripto. 16:3 121.

[7] Agus T. 2019. Membuat Web Server Menggunakan Dinamic Domain Name System Pada IP Dinamis. Jurnal Teknologi Informasi & Komunikasi Digital Zone. 7:1 3-4.

[8] Prabawa K S, Sembiring I. 2015. Penerapan K-Means Untuk Pengelompokan Pengguna Internet Berdasarkan Elapsed dan Byte Transferred. Universitas Kristen Satya Wacana Salatiga. 3-4.

[9] Aini F D, Riadi I, Umar R. 2019. Perancangan Deteksi Anomali Traffic Untuk Investigasi Log Menggunakan Metode K-Means Clusters. *Prosiding SNST*. 130.

[10] Rubangiya, Hartati T, Wijaya Y A. 2022. Analisis Data Lalu Lintas Jaringan Di Kantor Cangehgar Cyber Operation Center Menggunakan Algoritma K-Means. Jurnal Ilmiah NERO. 7:1 77.

[11] Ekasetya V A, Jananto A. 2020. Klusterisasi Optimal dengan Elbow Method Untuk Pengelompokan Data Kecelakaan Lalu Lintas Di Kota Semarang. Dinamika Informatika. 12:1 22.

[12] Fajrin R. 2017. Pengembangan Sistem Informasi Geografis Berbasis Node.JS untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM Indonesia. Jurnal Politeknik Caltex Riau. 3:1 35.

[13] Pratama I P A E. 2020. Pengujian Performansi Lima Back-End JavaScript Framework Menggunakan Metode Get dan Post. Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi). 1:1 1217.

[14] Heller M. 2020. What is Node.js? The JavaScript runtime explained. Infoworld.

[15] Adhikari A. 2016. Full Stack JavaScript: Web Application Development with MEAN. Helsinki Metropolia University of Applied Sciences. 15.

[16] Fadlilah E A, Chrisnanto Y H, Ningsih A K. 2022. Identifikasi Anomali Data Akademik Menggunakan DBSCAN Outlier Detection. Publikasiilmiah.unwahas.ac.id. 12:1 336.

[17] Nasution R F. 2024. Analisis Perbandingan Clustering Basec dengan Density Based dalam Mendeteksi Outlier. IJM: Indonesian Journal of Multidisciplinary. 2:2 236.

[18] Cao Q, Qiao Y, Lyu Z. 2017. Machine Learning to Detect Anomalies. IEEE. 520.