MEANS (Media Informasi Analisa dan Sistem)
p-ISSN: 2548-6985, e-ISSN:2599-3089

Kombinasi Algoritma K-Means dan DBSCAN dalam Identifikasi Anomali pada Data Log Server

Rico Puji Irawan
Universitas Budidarma, Sisingamangaraja, Sumatera Utara, Indonesia
E-Mail : ricopujiirawan@gmail.com

Sony Bahagia Sinaga
STMIK Mulia Darma, Adam Malik, Labuhan Batu, Indonesia
E-Mail : sonybahagia@gmail.com

ABSTRACT
Detecting anomalies in server log data is a crucial element of information system management and security. This research seeks to develop a method for identifying anomalies by integrating two well-known clustering algorithms: K-Means and DBSCAN (Density-Based Spatial Clustering of Applications with Noise). K-Means is effective at partitioning data into clusters based on average distances, while DBSCAN excels at detecting anomalies or noise in datasets without a distinct cluster structure. In this study, K-Means is employed for initial clustering of server log data to reveal general patterns and group similar data. The results from K-Means clustering are then examined using DBSCAN to detect anomalies more accurately. Combining these two algorithms aims to enhance anomaly detection accuracy by leveraging the strengths of each approach. The research was performed on a server log dataset encompassing various server activities. The effectiveness of this combined approach was assessed by comparing its anomaly detection performance against the individual K-Means and DBSCAN methods, as well as other anomaly detection techniques. Experimental results indicate that the K-Means and DBSCAN combination successfully improves anomaly detection rates by reducing both false positives and false negatives compared to using each algorithm independently.

Keywords: K-Means, DBSCAN, anomaly detection, clustering, server log data

PENDAHULUAN

Data log server adalah catatan yang mendokumentasikan semua aktivitas yang terjadi pada sistem, termasuk interaksi pengguna, akses file, permintaan layanan, dan lainnya. Salah satu tantangan utama dalam menganalisis data log server adalah volume data yang besar dan kompleksitasnya. Selain itu, pola-pola dalam data log server sering kali sulit diidentifikasi, dan aktivitas mencurigakan bisa tersembunyi di antara data yang tampak normal [1].

Anomali merujuk pada sesuatu yang tidak biasa, tidak normal, atau tidak sesuai dengan pola yang diharapkan dalam konteks tertentu. Anomali bisa disebabkan oleh berbagai faktor, termasuk serangan dari luar seperti peretas. Penyebab anomali dapat meliputi lonjakan data, noise, pola konstan, dan drift. Anomali dalam data log server bisa menandakan adanya serangan atau pelanggaran keamanan yang memerlukan perhatian dan penanganan segera. Mengingat banyaknya aktivitas yang terjadi, sangat penting untuk dapat membedakan antara aktivitas normal dan yang mencurigakan[2]. Untuk mendeteksi anomali dalam data log server sebagai langkah pencegahan terhadap serangan, penelitian ini mengkombinasikan algoritma K-Means dan DBSCAN.

K-Means adalah salah satu algoritma klastering yang paling umum digunakan. Klastering adalah proses membagi sekumpulan data menjadi subset-subset. Setiap subset, atau klaster, terdiri dari objek-objek yang mirip satu sama lain, tetapi berbeda dari objek-objek di klaster lainnya[3][4]. Algoritma DBSCAN dalam melakukan klastering dapat lebih efisien untuk membentuk klaster yang tidak teratur dibandingkan dengan algoritma CLARANS. Proses klastering ini didasarkan pada tingkat kedekatan atau kepadatan jarak antar objek dalam dataset, sehingga termasuk dalam kategori klastering berbasis kepadatan (density-based clustering) [5].

METODE PENELITIAN

2.1 Log Server

Log server mengawasi aktivitas di berbagai perangkat dan aplikasi dalam jaringan, memberikan wawasan kepada

administrator sistem tentang kondisi lingkungan IT mereka. Banyak regulasi dan standar keamanan yang mewajibkan organisasi untuk menyimpan dan menganalisis log aktivitas. Log server membantu organisasi memenuhi kewajiban ini dengan menyimpan log secara terstruktur dan dapat diakses [6].

2.2 Anomali

Anomali adalah penyimpangan yang terjadi dalam model lingkungan. Penyimpangan ini bisa disebabkan oleh faktor internal sistem itu sendiri atau oleh faktor eksternal. Penyimpangan internal biasanya memerlukan perbaikan langsung dari pengembang, sedangkan penyimpangan yang disebabkan oleh lingkungan menjadi perhatian bagi pengguna. Meskipun deteksi anomali yang melibatkan manusia bisa sangat efektif, prosesnya memerlukan waktu yang lama. Oleh karena itu, dikembangkanlah sistem deteksi anomali otomatis yang tidak memerlukan intervensi manusia. Sistem ini didasarkan pada teknologi machine learning, data mining, dan algoritma statistik[7].

2.3. Algoritma K-Means

Algoritma K-Means adalah metode dalam data mining yang digunakan untuk mengelompokkan data ke dalam satu atau lebih cluster. Dalam proses ini, data dengan karakteristik yang serupa dikelompokkan bersama dalam satu cluster, sementara data dengan karakteristik yang berbeda dikelompokkan ke cluster lainnya. Tujuan utama dari klastering K-Means adalah menemukan titik data prototipe untuk setiap cluster; kemudian, semua titik data akan ditetapkan ke prototipe terdekat, membentuk sebuah cluster, langkah-langkah kerja algoritma K-Means [8]:

1. Menentukan Jumlah Centroid
2. Menetapkan Poin Data
3. Menghitung Centroid Baru
4. Mengulang Penetapan dan Penghitungan Centroid
5. Penghentian

2.4. Algoritma DBSCAN

Algoritma DBSCAN (Density-Based Spatial Clustering of Applications with Noise) adalah metode klastering yang berbasis kepadatan, digunakan untuk mengelompokkan data dengan cara menemukan area-area dengan kepadatan tinggi dan memisahkan area-area dengan kepadatan rendah sebagai noise atau anomali. DBSCAN sangat efektif dalam mendeteksi klaster yang tidak berbentuk bulat dan menangani data dengan noise. Berikut adalah langkah-langkah dasar dalam algoritma DBSCAN [9][10]:

1. Inisialisasi
2. Identifikasi Titik Awal
3. Klasifikasikan Titik
4. Bentuk Klaster
5. Ulangi Proses
6. Hasil Klastering

Setelah algoritma selesai, dataset akan terbagi menjadi beberapa klaster, dan beberapa titik data mungkin diklasifikasikan sebagai noise jika tidak memenuhi syarat untuk menjadi bagian dari klaster manapun. DBSCAN unggul dalam mengidentifikasi bentuk klaster yang tidak teratur dan menangani noise, serta tidak memerlukan jumlah klaster yang harus ditentukan sebelumnya, berbeda dengan algoritma klastering lain seperti K-Means.

HASIL DAN PEMBAHASAN

Identifikasi anomali dalam data log server adalah proses untuk menemukan aktivitas atau pola yang tidak biasa atau mencurigakan dalam log server, yang mungkin menandakan adanya masalah keamanan, kinerja, atau operasional. Analisis ini sangat penting untuk mendeteksi dan mencegah insiden yang tidak diinginkan. Langkah pertama dalam analisis anomali adalah mengumpulkan data log dari berbagai sumber, seperti server web, aplikasi, database, dan sistem operasi. Data log ini mencakup informasi seperti waktu kejadian, alamat IP, pesan kesalahan, dan aktivitas pengguna. Karena data log sering kali tidak terstruktur dan berantakan, preprocessing diperlukan untuk membersihkan dan mengatur data. Setelah data dibersihkan, langkah berikutnya adalah ekstraksi fitur, yang melibatkan identifikasi atribut atau karakteristik penting dari data log yang dapat digunakan untuk analisis. Untuk mendeteksi anomali dalam data log server, dilakukan kombinasi antara algoritma K-Means dan DBSCAN. K-Means *Clustering* dan DBSCAN dilakukan pada variabel akses log yang dipilih. Berikut tahapan yang dilewati dalam pengerjaan penelitian ini.

1. Data *preprocessing*
   Data akses log dengan ekstensi log perlu diuraikan terlebih dahulu untuk memudahkan sistem dalam memproses data tersebut.
2. Ekstraksi dan pemilihan fitur
   Analisis K-Means dan DBSCAN diterapkan pada beberapa variabel yang dipilih melalui tahapan Ekstraksi dan Pemilihan Fitur. Proses ekstraksi fitur ini bertujuan untuk mengubah data menjadi fitur yang sesuai dengan model. Dalam

MEANS (Media Informasi Analisa dan Sistem)
p-ISSN: 2548-6985, e-ISSN:2599-3089

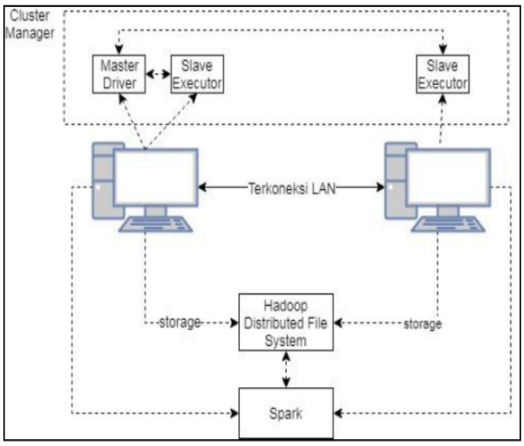
penelitian terkait ekstraksi fitur pada data *log server web*, ditemukan bahwa terdapat 30 fitur yang bisa diambil dari sebuah *file log* untuk mendeteksi serangan.

3. Pengolahan dan Analisis Data
   Setelah proses seleksi fitur untuk K-Means dan DBSCAN selesai, data yang telah diproses akan siap untuk dianalisis menggunakan Hadoop dan Spark. Proses ini melibatkan penggunaan dua aplikasi terkait: Hadoop Distributed File System (HDFS) dan Spark. HDFS berfungsi sebagai media penyimpanan terdistribusi antar komputer, memungkinkan Hadoop dan Spark untuk melakukan pengolahan data secara terdistribusi. Dalam eksperimen ini, digunakan dua komputer: satu sebagai master node dan satu sebagai slave node. Master node bertanggung jawab mengatur pengolahan data secara terdistribusi, sementara slave node berfungsi sebagai komputer pekerja. Untuk tujuan penelitian ini, evaluasi model dilakukan dengan menggunakan metrik yang sama seperti pada penelitian sebelumnya, yaitu menghitung tingkat false-positive, false-negative, sensitivitas model, spesifisitas model, tingkat klasifikasi, dan presisi hasil model.

Tabel 1. Fitur Terpilih Variabel Peneltiian

No	Feature Name
1	Panjang dari request line
2	Panjang dari IP Host
3	Panjang header dari user-agent
4	Besarnya ukuran byte setiap request
5	Panjang header dari referer
6	Jumlah okurensi garis miring
7	Jumlah okurensi titik
8	Jumlah okurensi persentase

4. Clustering
   Dari fitur yang telah dipilih, dilakukan ekstraksi dengan cara mengukur panjang dari variabel tertentu dan menghitung frekuensi kemunculan karakter tertentu. Selain itu, variabel-variabel yang tidak diperlukan juga dihapus.



Gambar 1. Alur Kerja Fisik

Pengolahan data pada Spark tidak sepenuhnya bergantung pada jumlah komputer, melainkan pada jumlah node yang digunakan. Dalam Spark, terdapat satu master node dan beberapa slave node. Dalam penelitian ini, digunakan dua slave node. Master node, yang dikendalikan oleh driver, bertanggung jawab untuk mengalokasikan tugas. Driver meneruskan tugas-tugas tersebut ke cluster manager, dalam hal ini adalah Spark sebagai cluster manager. Cluster manager kemudian mendistribusikan tugas dari driver ke slave node untuk diproses. Hasil dari pengolahan data ini mencakup pusat klaster dan model yang dapat diakses.

```
In [1]: from pyspark.mllib.clustering import BisectingKMeans, BisectingKMeansModel
        from numpy import array
        from math import sqrt

        data = sc.textFile("///user/hadoop/revisi_4.csv")
        parsedData = data.map(lambda line: array([float(x) for x in line.split(',')]))

        model = BisectingKMeans.train(parsedData, 2, maxIterations=5)

        for x in model.clusterCenters: print(x)

        [7.51238183e+02 6.79834723e+01 1.30996041e+01 8.99985370e+01
         2.14752879e+01 5.60538117e+00 1.88649086e+00 1.38719273e-01]
        [3.25604176e+04 4.46917369e+01 1.34691553e+01 1.15906847e+02
         3.05531600e+01 3.46086011e+00 2.14465707e+00 1.63326513e-01]
```

Gambar 2. Hasil Running

5. Hasil Pengolahan data
   Hasil dari pengolahan data diambil dari akses log 8 variabel yang menghasilkan dua pusta cluster yaitu [11.944,147 ; 58,0004 ; 13,257 ; 101,08 ; 25,342 ; 4,685 ; 1,997 ; 0,147] serta [1.729.124,25 ; 57,677 ; 13,38 ; 109,8 ; 39,744 ; 5,123 ; 2,066 ; 1,465]

```
In [2]: from pyspark.mllib.clustering import KMeans, KMeansModel
        from numpy import array
        from math import sqrt

        data = sc.textFile("///user/hadoop/revisi_4.csv")
        parsedData = data.map(lambda line: array([float(x) for x in line.split(',')]))

        clusters = KMeans.train(parsedData, 2)

        for x in clusters.clusterCenters: print(x)

        [1.14941476e+04 5.80004820e+01 1.32577988e+01 1.01088818e+02
         2.53420899e+01 4.68543727e+00 1.99703187e+00 1.47047355e-01]
        [1.72912425e+06 5.76777824e+01 1.33801584e+01 1.09877345e+02
         3.97445811e+01 5.12390579e+00 2.06638183e+00 1.46540225e+00]
```

Gambar 3. Hasil Running Menggunakan Algoritma

Hasil dari menjalankan skrip menunjukkan bahwa dari 5.700.375 catatan log akses dengan 8 variabel, dihasilkan dua pusat klaster sebagai berikut: [751,238 ; 67,983 ; 13,099 ; 89,99 ; 21,475 ; 5,605 ; 1,8864 ; 0,1387] dan [3,256 ; 4,469 ; 13,469 ; 115,906 ; 30,55 ; 3,46 ; 2,144 ; 0,1633]. Untuk menguji efektivitas model, dihitung serangkaian metrik, termasuk false positive rate (FPR), false negative rate (FNR), sensitivitas, presisi, tingkat klasifikasi, dan spesifisitas. Pada algoritma K-Means Clustering, nilai FPR yang diperoleh adalah 0,0191%, sementara pada DBSCAN mencapai 53,58%. Semakin tinggi nilai FPR, semakin sensitif model dalam mendeteksi aktivitas jaringan. Namun, nilai FPR yang tinggi juga menunjukkan bahwa banyak aktivitas normal mungkin terdeteksi sebagai anomali.

Tabel 1. Comfusion Matriks

		Predicted value	
		P	N
True value	P	76	295
	N	53.390	46.238

Hasil dari metode clustering K-Means menghasilkan pembagian data ke dalam tiga klaster, di mana setiap pusat centroid digunakan untuk menentukan kategori nilai risiko serangan. Hasil dari proses clustering dengan metode K-Means berdasarkan dataset dapat dilihat pada Tabel 2.

Tabel 2. Hasil Clustering

Cluster	Jumlah Data
Cluster 1	306743
Cluster 2	1685
Cluster 3	353

Kelihatan bahwa Cluster 1 memiliki jumlah data terbesar dibandingkan klaster lainnya, yaitu total 306.743 data. Sementara itu, Cluster 2 terdiri dari 1.685 data, dan Cluster 3 memiliki jumlah data terkecil, yaitu 353 data. Titik pusat centroid akhir dari hasil clustering ini ditunjukkan dalam Tabel 3.

Tabel 3. Pusat Centroid Akhir

Cluster	Prioritas	Prot	Frekuensi
Cluster 1	2	4	5
Cluster 2	3.0652819	4	1
Cluster 3	3.25212465	1.99150142	1

KESIMPULAN

Berdasarkan hasil penelitian dan analisis mengenai pengamanan pesan teks, kesimpulan yang dapat diambil adalah sebagai berikut:

1. Dari 308.781 data yang diklaster, serangan berhasil dikategorikan dengan baik. Terdapat tiga klaster, di mana Cluster 2 dan Cluster 3 termasuk dalam kategori risiko serangan rendah, dengan total 2.038 data. Sebaliknya, Cluster 1 tergolong dalam kategori risiko serangan menengah, dengan jumlah data sebanyak 306.743. Nilai silhouette score untuk klastering ini adalah 0,999.
2. Hasil identifikasi menunjukkan protokol-protokol yang sering menjadi target serangan serta kategori risiko serangan yang terdeteksi dalam data log server. Metode K-Means biasanya digunakan dalam berbagai penelitian lain untuk mengelompokkan data log.
3. Wazuh berkontribusi pada pengembangan Dasbor Sistem Pencatatan Log yang lebih optimal, yang terbukti efektif dan bermanfaat untuk pengelolaan log terpusat (CLM). Informasi log yang dihasilkan dalam format yang mudah dipahami mendukung pengembangan CLM dengan baik.

UCAPAN TERIMKASIH

Penulis mengucapkan terimakasih ke pembimbing yang telah banyak memberikan masukan dalam penyelesaian penelitian ini. Ucapan terima kasih juga penulis ucapkan ke program studi teknik informatikan yang telah memberikan kesempatan untuk melakukan penelitian ini serta ucapan terimakasih juga penulis ucapkan kepada orangtua yang telah memberikan dukungan.

DAFTAR PUSTAKA

[1] Gustientiedina, *et.al,* 2019, "*Penerapan Algoritma K-Means Untuk Clustering Data Obat-Obatan Pada RSUD Pekanbaru",* Jurnal Nasional Teknologi dan Sistem Informasi, Vol 05, No. 01

[2] Wahyu Sudrajat, *et.al,* 2022, "*Penerapan Algoritma K-Means Clustering Untuk Pengelompokkan UMKM Menggunakan Rapidminer",* Jurnal JUPITER, Vol. 14, No. 1

[3] Rozi Kesuma Dinata, *et.al,* 2020, "*Analisis K-Means Clustering Pada Data Sepeda Motor",* Informatic Journal, Vol.

5, No. 1
[4] Devi Fitrianah, *et.al,* 2021, "*Implementasi Algoritma DBScan Dalam Pengambilan Data Menggunakan Scatterplot",* Jurnal Techno Xplore, Vol. 6, No. 2
[5] Betha Nurina Sari, *et.al,* 2019, "*Penerapan Clustering DBScan Untuk Pertanian Padi di Kabupaten Karawang",* Jurnal JIKO, Vol. 4, No. 1
[6] F. Pangestu, *et.al,* 2023, *"Penerapan Algoritma K-Means Untuk Mengklasifikasi Data Obat",* Jurnal SISFOKOM, Vol. 12, No. 1
[7] Mustofa, 2019, *"Penerapan Algoritma K-Means Clustering Pada Karakter Permainan Multiplayer Online Battle Arena",* Jurnal Informatika, Vol. 6, No. 2
[8] B.S. Ashari, *et.al,* 2019, "*Perbandingan Kinerja K-Means dengan DBScan Untuk Metode Clustering data Penjualan Online Retail",* Jurnal Siliwangi, Vol. 5, No. 2
[9] T.I. Hermanto, *et.al,* 2020, "*Analisis Data Sebaran Bandwidth Menggunakan Algoritma DBScan Untuk Menentukan Tingkat Kebutuhan Bandwidth Di Kabupaten Purwakarta",* Jurnal RABIT, Vol. 5, No. 2
[10] Mustika Putri, *et.al,* 2021, *"Komparasi DBSCAN Dan K-Means Clustering Pada Pengelompokkan Status Desa di Jawa Tengah Tahun 2020",* Jurnal Matematika, Statistika, Vol. 17, No. 3
[11] D. P. Isnarwaty, *et.al,* 2019, "*Text Clustering Pada Akun Twitter Layanan Ekspedisi JNE, J&T dan Pos Indonesia Menggunakan Metode Density- Based Spatial Clustering Of Applications With Noise (DBSCAN) Dan K-Means*