Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI
Volume 00, Nomor 00, Bulan Tahun, Hal 00-00
ISSN 2723- 6129 (media online)
Link Jurnal : https://ejournal.sisfokomtek.org/index.php/jikom

Penerapan Algoritma Support Vector Machine untuk Mendeteksi Anomali pada Jaringan Komputer

Reza Puspita Sari Pohan[1*], Angga Putra Juledi[2], Ibnu Rasyid Munthe[3]

[1,2,3]Sitem Informasi, Unversitas Labuhan Batu, Rantauprapat, Indonesia
Email: [1,*]rezapuspitasarisimanjuntak@gmail.com, [2]anggapj19@gmail.com, [3,]Ibnurasyidmunthe@gmail.com
Email Penulis Korespondensi: [1] rezapuspitasarisimanjuntak@gmail.com

Abstrak− Anomali jaringan biasanya menunjukkan masalah atau ancaman keamanan potensial. Anomali jaringan dapat menyebabkan kerugian keuangan dan reputasi perusahaan serta merusak integritas, kerahasiaan, dan ketersediaan data. Teknik deteksi anomali tradisional menggunakan algoritma berbasis aturan memiliki keterbatasan dalam menemukan anomali yang beragam dan canggih. Sebaliknya, algoritma *machine learning* telah menunjukkan hasil yang luar biasa. Akibatnya, semakin banyak orang yang mulai menggunakan *machine learning* untuk mendeteksi anomali. Penelitian ini bertujuan untuk menerapkan algoritma support vector machine dalam mendeteksi anomali pada jaringan komputer. Tahapan penelitian dimulai dari pengumpulan dataset, prapemrosesan data, penerapan algoritma SVM, dan evaluasi hasil. Penelitian ini telah berhasil menerapkan algoritma support vector machine untuk mendeteksi anomali pada jaringan komputer. Menurut hasil matriks evaluasi kinerja, diperoleh akurasi keseluruhan sebesar 81,50%, presisi 74,66%, *recall* 100%, dan f1-*score* 85,49%. Hasil matriks ini menunjukkan bahwa model dapat dengan akurat mengidentifikasi sampel kelas Normal dan Anomali. Secara keseluruhan, temuan ini menunjukkan bahwa model algoritma support vector machine sangat baik untuk menemukan anomali jaringan. Hasil kurva *receiver operating characteristic* sebesar 0,97 menunjukkan bahwa model support vector machine dengan hiperparameter kernel RBF, C=10 dan gamma=auto yang dipilih sangat akurat dalam membedakan antara contoh positif dan negatif dalam kumpulan data lalu lintas jaringan. Penelitian ini diharapkan dapat membantu administrator jaringan menemukan dan membuat keputusan yang tepat tentang cara mengatasi anomali jaringan komputer.

Kata Kunci: Anomali, Deteksi, Jaringan Komputer, Machine Learning, ROC, SVM

Abstract− Network anomalies usually indicate potential security issues or threats. Network anomalies can cause financial and reputational losses to companies and damage the integrity, confidentiality, and availability of data. Traditional anomaly detection techniques using rule-based algorithms have limitations in finding diverse and sophisticated anomalies. In contrast, machine-learning algorithms have shown remarkable results. As a result, more and more people are starting to use machine learning to detect anomalies. This study aims to apply the support vector machine algorithm to detect anomalies in computer networks. The research stages start with dataset collection, data preprocessing, application of the SVM algorithm, and evaluation of the results. This study has successfully applied the support vector machine algorithm to detect anomalies in computer networks. According to the results of the performance evaluation matrix, the overall accuracy was 81.50%, precision was 74.66%, recall was 100%, and f1-score was 85.49%. The results of this matrix indicate that the model can accurately identify normal and anomalous class samples. Overall, these findings indicate that the support vector machine algorithm model is very good for finding network anomalies. The result of the receiver operating characteristic curve of 0.97 indicates that the support vector machine model with the selected RBF kernel hyperparameters, C = 10 and gamma = auto is very accurate in distinguishing between positive and negative examples in the network traffic data set. This study is expected to help network administrators find and make the right decisions on how to overcome computer network anomalies.

Keywords: Anomaly, Detection, Machine Learning, Network, ROC, SVM

1. PENDAHULUAN

Di era digital yang semakin maju ini, keamanan jaringan menjadi masalah yang sangat penting bagi bisnis, kelompok, dan pengguna internet pada umumnya. Beberapa sumber ancaman keamanan jaringan yang sering ditemui termasuk diantaranya adalah serangan *malware*, serangan DDoS, peretasan, dan pencurian data. Serangan berpotensi meningkat dengan jumlah pengguna dan jangkauan komunikasi. Setiap perusahaan dan organisasi sangat menghargai data. Informasi sangat penting sehingga hanya orang yang berkepentingan dapat mengaksesnya. Oleh karena itu, keamanan jaringan harus dijamin sehingga orang yang tidak berkepentingan tidak dapat mengaksesnya.

Anomali jaringan adalah kejadian yang tidak biasa atau tidak terduga yang terjadi dalam jaringan komputer atau sistem. Anomali jaringan adalah keadaan yang terjadi pada trafik jaringan yang menyebabkan kondisi menjadi tidak normal [1]. Anomali jaringan biasanya menunjukkan masalah atau ancaman keamanan potensial. Mereka dapat mencakup perubahan pada lalu lintas data, perilaku perangkat, atau protokol yang tidak standar [2]. Anomali jaringan dapat merusak integritas, kerahasiaan, dan ketersediaan data serta mengakibatkan kerugian keuangan dan reputasi perusahaan. Deteksi anomali adalah metode penting untuk menemukan masalah operasional dan ancaman keamanan dalam jaringan komputer. Ini dapat digunakan untuk menemukan serangan pengalaman untuk mengidentifikasi perilaku pengguna yang tidak biasa [3]. Deteksi anomali lebih baik daripada deteksi penyalahgunaan karena dapat mendeteksi

Submitted: **dd/mm/yyyy**; Accepted: **dd/mm/yyyy**; Published: **dd/mm/yyyy**

serangan yang tidak diketahui, jadi hanya contoh normal yang diperlukan untuk melatih model [4]. Tujuan deteksi anomali adalah untuk mengidentifikasi setiap kejadian dengan data [5].

Administrator jaringan bertanggung jawab untuk mengelola jaringan komputer sehari-hari yang sangat rumit. Pengawasan keamanan membutuhkan administrator jaringan yang sangat terampil dalam memahami kondisi dan perilaku jaringan. Administrator jaringan biasanya mengelola berbagai tugas, seperti melihat grafik dan statistik trafik, mencari puncak pemakaian yang tidak biasa, seperti jumlah *byte* paket yang dikirim, dan memeriksa insiden tertentu dengan menggunakan aplikasi seperti penganalisis paket, kolektor aliran, *firewall*, dan *log* sistem pada *server*. Untuk menangani masalah jaringan, administrator jaringan biasanya menggunakan sistem konvensional. Tidak ada sistem yang dapat secara otomatis menemukan masalah pada jaringan komputer. Untuk itu, dibutuhkan solusi yang efisien untuk menangani anomali jaringan dengan respons yang sangat cepat.

Untuk menemukan anomali yang beragam dan canggih, teknik deteksi anomali tradisional menggunakan algoritma berbasis aturan, tetapi algoritma *machine learning* telah menunjukkan hasil yang bagus. Semakin banyak orang mulai menggunakan *machine learning* untuk mendeteksi anomali. Ini karena kelebihan *machine learning* dalam mendeteksi anomali adalah kemampuannya dalam mengenali pola yang rumit, mudah beradaptasi dengan perubahan, dapat diterapkan pada jumlah data yang besar tanpa mempengaruhi kinerja, kemampuan untuk belajar mandiri, dan kemampuan untuk menemukan anomali yang tidak diketahui sebelumnya [6]. *Machine learning* juga dapat mendeteksi anomali sesuai dengan kebutuhan dan karakteristik jaringan yang berbeda. Kelebihan lain dari *machine learning* adalah dapat memberikan respons yang cepat untuk mendeteksi anomali jaringan [7].

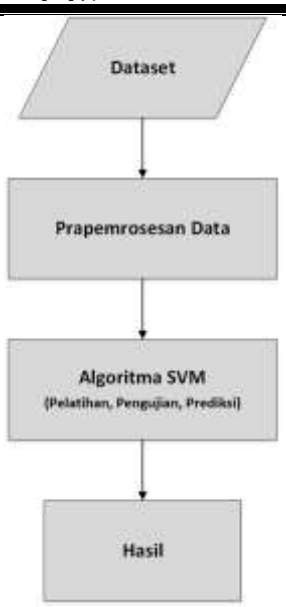
Peneliti-peneliti sebelumnya telah menggunakan beberapa algoritma *machine learning* untuk menemukan anomali pada jaringan komputer [8], [9], [10]. Karena keakuratannya dalam klasifikasi dataset, algoritma support vector machine sering digunakan oleh para peneliti. Algoritma support vector machine mampu membuat model yang dapat membedakan data normal dari data yang mencurigakan atau berpotensi berbahaya, sehingga dapat mendeteksi serangan dengan sangat akurat [11]. Algoritma support vector machine menghasilkan grafik yang lebih akurat dalam analisis perbandingan deteksi trafik anomali [12]. Untuk optimalisasi deteksi serangan DDoS dengan menggunakan algoritma Support Vector Machine, diperoleh hasil akurasi sebesar 98,37% [13]. Penelitian sebelumnya mengidentifikasinya sebagai algoritma dengan *True Positive Rate* (TPR) dan akurasi yang lebih tinggi. Ini menunjukkan betapa efektifnya algoritma ini dalam menyelesaikan tugas-tugas yang sangat kompleks [14]. Algoritma support vector machine memiliki beberapa kelebihan dalam hal generalisasi dan *feasibility*. Dalam hal generalisasi, algoritma ini dapat mengkategorikan pola yang tidak ada dalam dataset pembelajaran mesin. Untuk memungkinkan generalisasi ini, support vector machine bekerja berdasarkan prinsip pengurangan resiko struktur [15]. Dalam hal *feasibility*, penerapan support vector machine lebih memudahkan pengguna [16].

Berdasarkan penelitian-penelitian yang telah diuraikan sebelumnya, terbukti bahwa algoritma support vector machine dapat mendeteksi anomali pada jaringan secara akurat. Oleh karena itu, penelitian ini bertujuan untuk menerapkan algoritma support vector machine untuk mendeteksi anomali pada jaringan komputer. Adapun perbedaan antara penelitian ini dengan penelitian sebelumnya, terletak pada dataset yang digunakan. Kemudian, hasil penerapan support vector machine akan diukur dengan menggunakan confusion matrix yang akan menghasilkan tingkat akurasi, presisi, *recall*, dan f1-*score*. Semoga penelitian ini dapat membantu administrator jaringan dalam menemukan informasi dan mengambil keputusan yang dibutuhkan dalam mengatasi masalah anomali pada jaringan komputer secara efektif dan efisien.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Bagian ini menyajikan tahapan penelitian yang digunakan untuk mendeteksi anomali pada jaringan komputer yang menggunakan algoritma Support Vector Machine (SVM). Tahapan penelitian sistematis memudahkan pembuatan sistem deteksi anomali yang dapat dipercaya dan diandalkan. Pengumpulan dataset, prapemrosesan data, penerapan algoritma SVM, dan hasil adalah semua proses yang dilakukan seperti diperlihatkan pada gambar 1.



Gambar 1. Tahapan Penelitian

2.1.1 Dataset
Dataset yang digunakan pada penelitian ini merupakan dataset yang berasal Kaggle [17]. Dataset terdiri dari berbagai macam intrusi yang disimulasikan. Dataset ini terdiri dari 41 fitur input dan 1 fitur target yang merupakan *class* "Normal" dan "Anomali".

2.1.2 Prapemrosesan Data
Pada langkah ini, data dipilih, dibersihkan, dan diubah ke dalam bentuk yang diinginkan. Setelah ini selesai, persiapan untuk pembuatan model dapat dimulai. Sebelum diproses pada tahap berikutnya, data harus disiapkan dengan benar melalui tahap pengolahan awal. Pada tahap ini, perawatan, transformasi, reduksi, dan seleksi fitur dilakukan. Data yang diperoleh diolah untuk menghasilkan atribut yang sesuai dan relevan.

2.1.3 Algoritma SVM
Pada titik ini, algoritma dilatih menggunakan data pelatihan yang mengandung berbagai serangan intrusi. Sangat penting untuk memahami apa yang dimaksud dengan pelatihan. Ketika algoritma belajar, itu disebut pelatihan. Dataset pengujian digunakan untuk melakukan pengujian. Ketika algoritma memprediksi setelah belajar, itu disebut pengujian. Inilah sebabnya mengapa data pengujian dan pelatihan berbeda. Algoritma melihat serangan baru di data pengujian. Dengan kemampuan untuk memprediksi serangan baru dalam data uji, algoritma siap untuk mendeteksi serangan baru dalam jaringan.

2.1.4 Hasil
Hasil dari algoritma diukur menggunakan confusion matrix yang meliputi akurasi, presisi, *recall*, dan f1-score serta kurva ROC. Parameter untuk mengukur kinerja pada kurva ROC adalah; *True Positive Rate* (TPR) dan *False Positive Rate* (FPR). TPR adalah jumlah *True Positive* (TP) dan FPR adalah jumlah *False Positive* (FP). Nilai ROC yang tinggi menunjukkan kinerja yang baik. ROC 0,70-0,80 (dapat diterima), ROC 0,80-0,90 (sangat baik) dan ROC 0,90 ke atas (luar biasa).

3. HASIL DAN PEMBAHASAN
Penelitian ini menggunakan Python sebagai bahasa pemrograman. Python digunakan karena kemampuannya dalam beradaptasi, kesederhanaan, dan dukungan luas untuk penerapan *machine learning* dalam memanipulasi dan visualisasi data. Jupyter Notebook adalah alat web sumber terbuka yang digunakan pada penelitian ini sebagai teks editor dengan memanfaatkan Google Colab. Google Colab (*colaboratory*) adalah sebuah layanan gratis dari Google yang memungkinkan pengguna untuk menulis dan mengeksekusi kode Python melalui *web browser*. Layanan ini berjalan di atas *platform cloud* milik Google, yang artinya tidak perlu menginstal atau mengkonfigurasi lingkungan Python di komputer local yang digunakan.

Submitted: **dd/mm/yyyy**; Accepted: **dd/mm/yyyy**; Published: **dd/mm/yyyy**

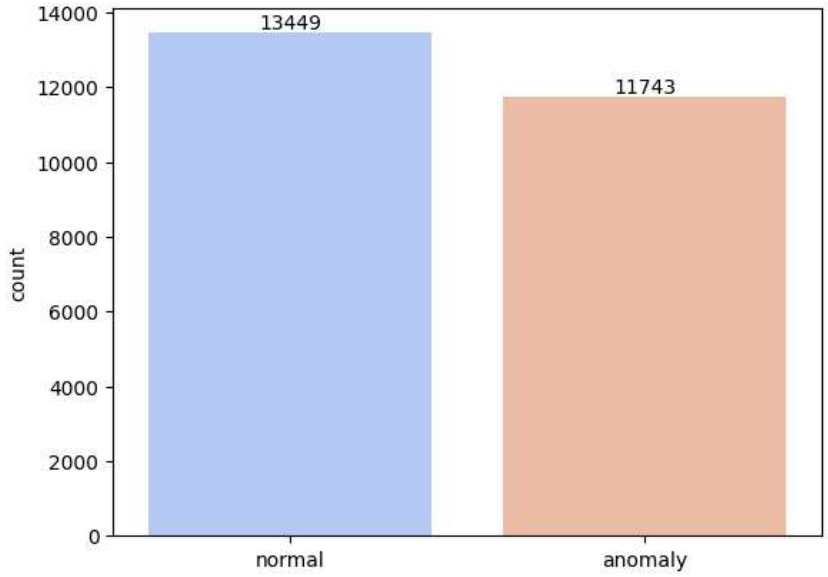
	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	...	25	0.17
1	0	udp	other	SF	146	0	0	0	0	0	...	1	0.00
2	0	tcp	private	S0	0	0	0	0	0	0	...	26	0.10
3	0	tcp	http	SF	232	8153	0	0	0	0	...	255	1.00
4	0	tcp	http	SF	199	420	0	0	0	0	...	255	1.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...
25187	0	tcp	exec	RSTO	0	0	0	0	0	0	...	7	0.03
25188	0	tcp	ftp_data	SF	334	0	0	0	0	0	...	39	1.00
25189	0	tcp	private	REJ	0	0	0	0	0	0	...	13	0.05
25190	0	tcp	nnsp	S0	0	0	0	0	0	0	...	20	0.08
25191	0	tcp	finger	S0	0	0	0	0	0	0	...	49	0.19

25192 rows × 42 columns

Gambar 2. Dataset

Gambar 1 menunjukkan dataset yang digunakan pada penelitian ini yang telah dimuat pada Google Colab. Dataset ini terdiri dari 25.192 baris dan 42 kolom yang dimuat dalam sebuah file berformat .csv. Adapun diantara fitur-fitur yang terdapat pada kolom dataset tersebut, dapat dijelaskan sebagai berikut:

1. ***duration*** : Durasi waktu koneksi
2. ***protocol_type*** : Protokol yang digunakan dalam koneksi
3. ***service*** : Layanan jaringan tujuan yang digunakan
4. ***flag*** : Status koneksi – Normal atau *Error*
5. ***src_bytes*** : Jumlah *byte* data yang ditransfer dari sumber ke tujuan dalam satu koneksi
6. ***dst_bytes*** : Jumlah *byte* data yang ditransfer dari tujuan ke sumber dalam satu koneksi
7. ***land*** : Jika alamat IP dan nomor *port* sumber dan tujuan sama, maka variabel ini akan bernilai 1 jika tidak, 0
8. ***wrong_fragment*** : Jumlah total fragmen yang salah dalam koneksi ini
9. ***urgent*** : Jumlah paket yang mendesak dalam koneksi ini. Paket mendesak adalah paket dengan bit mendesak yang diaktifkan
10. ***hot*** : Jumlah indikator "*hot*" dalam konten seperti: memasuki direktori sistem, membuat program, dan menjalankan program
11. ***dst_hostsrv count*** : Jumlah koneksi yang memiliki nomor *port* yang sama
12. ***dst_host_same srv_rate*** : Persentase koneksi yang menuju ke layanan yang sama, di antara koneksi yang dikumpulkan dalam dst_host_count
13. ***class*** : merupakan target prediksi yang terdiri dari “Normal” dan “Anomali”



Gambar 3. Sebaran Dataset pada *Class*

Submitted: **dd/mm/yyyy**; Accepted: **dd/mm/yyyy**; Published: **dd/mm/yyyy**

Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI
Volume 00, Nomor 00, Bulan Tahun, Hal 00-00
ISSN 2723- 6129 (media online)
Link Jurnal : https://ejournal.sisfokomtek.org/index.php/jikom

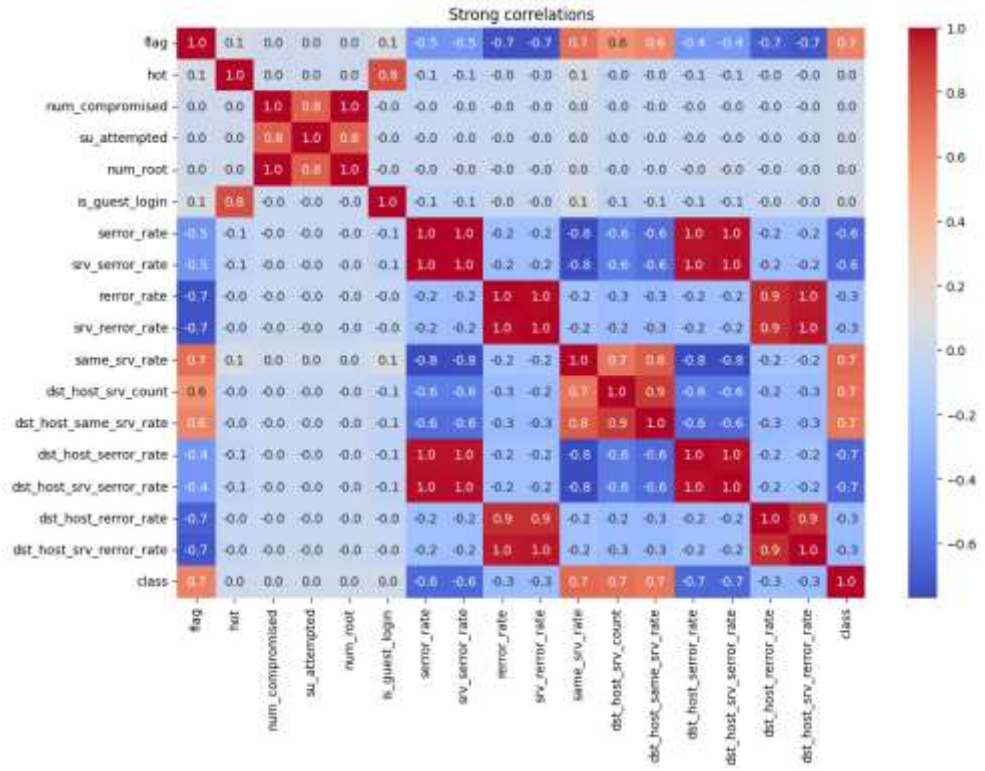
Gambar 3 menunjukkan grafik sebaran dataset jika dilakukan pembagian berdasarkan *class*. Tampak dari gambar, jumlah data yang berkategori normal adalah sebanyak 13.449. Sedangkan jumlah data yang dikategorikan sebagai serangan (anomali) yaitu, sebanyak 11.743.

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	diff_srv_rate	srv_diff_host_rate	dst_host_count
0	0	1	19	9	491	0	0	0	0	0	...	0.00	0.00	150
1	0	2	41	9	146	0	0	0	0	0	...	0.15	0.00	255
2	0	1	46	5	0	0	0	0	0	0	...	0.07	0.00	255
3	0	1	22	9	232	8153	0	0	0	0	...	0.00	0.00	30
4	0	1	22	9	199	420	0	0	0	0	...	0.00	0.09	255
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
25187	0	1	16	2	0	0	0	0	0	0	...	0.07	0.00	255
25188	0	1	19	9	334	0	0	0	0	0	...	0.00	0.00	1
25189	0	1	46	1	0	0	0	0	0	0	...	0.07	0.00	255
25190	0	1	38	5	0	0	0	0	0	0	...	0.06	0.00	255
25191	0	1	17	5	0	0	0	0	0	0	...	0.11	0.00	255

25192 rows × 33 columns

Gambar 4. Dataset setelah Prapemrosesan

Gambar 4 merupakan data yang sudah bersih setelah melalui tahapan prapemrosesan data. Adapun tahap yang dilalui adalah, mengubah isi dari fitur-fitur seperti, protocol_type, service, dan flag yang semula bertipe kategori, kini telah menjadi data numerik. Selain itu, fitur class yang merupakan target dari prediksi, yang semula bertipe kategori, juga telah diubah menjadi data numerik. Dimana, kelas "Normal" diubah menjadi "1", dan kelas "Anomali" diubah menjadi "0". Proses ini merupakan proses pengkodean pada fitur dengan mengubah tipe data *object* menjadi tipe data *float* dan *integer*. Proses pengkodean fitur dilakukan agar terjadi keseimbangan rentang data. Proses ini dilakukan dengan menggunkan *LabelEncoder* yang ada pada Python.



Gambar 5. Korelasi Fitur

Selain melakukan pengkodean fitur. Proses lain yang dilakukan pada tahap prapemrosesan data adalah, membuang fitur-fitur yang dianggap tidak relevan karena memiliki nilai korelasi yang tidak lazim yaitu, memiliki nilai korelasi 1:1 diluar fitur itu sendiri. Gambar 5 memperlihatkan fitur-fitur yang memiliki korelasi 1:1 diluar korelasi dengan dirinya. Dalam hal ini, fitur-fitur yang dibuang adalah: *num_compromised*, *num_root*, *serror_rate*,

Submitted: **dd/mm/yyyy**; Accepted: **dd/mm/yyyy**; Published: **dd/mm/yyyy**

srv_serror_rate, *rerror_rate*, *srv_rerror_rate*, *dst_host_serror_rate*, *dst_host_srv_serror_rate*, *dst_host_srv_rerror_rate*. Terdapat 9 fitur yang dibuang, sehingga jumlah kolom berkurang menjadi 33 kolom. Setelah melalui tahapan tersebut, maka dataset siap diimplementasikan pada algoritma support vector machine.

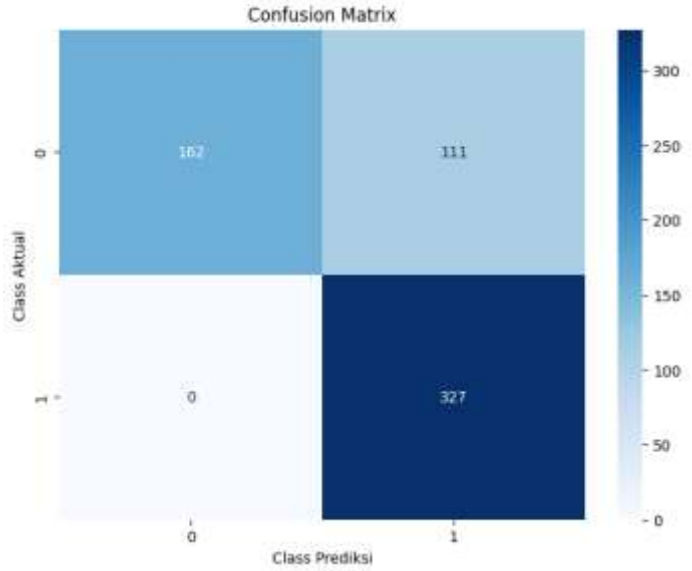
```
df = df.sample(frac=1, random_state=42).reset_index(drop=True)

# Pilih subset data untuk pelatihan
num_data = 2000
train_df = df.iloc[:num_data].copy()
train_df = train_df.astype("float64")

print('Data type of each column of Dataframe :')
train_df.info(verbose=True)
```

Gambar 6. Subset Data Pelatihan

Tahapan pada gambar 6 adalah membuat *array* numpy dari variabel independen yang dipilih dari set data pelatihan, yang akan digunakan untuk melatih model SVM. Tujuan pembuatan *array* numpy adalah untuk memastikan bahwa data berada dalam format yang dapat diproses secara efisien oleh algoritma pembelajaran mesin. Array numpy yang dibuat di sini memiliki 2.000 baris (dipilih 2.000 baris dari dataset asli) dan 33 kolom (kolom asli dari dataset). Jumlah dataset pengujian yaitu sebesar 30% dari 2.000 data, berarti ada 600 dataset pengujian.



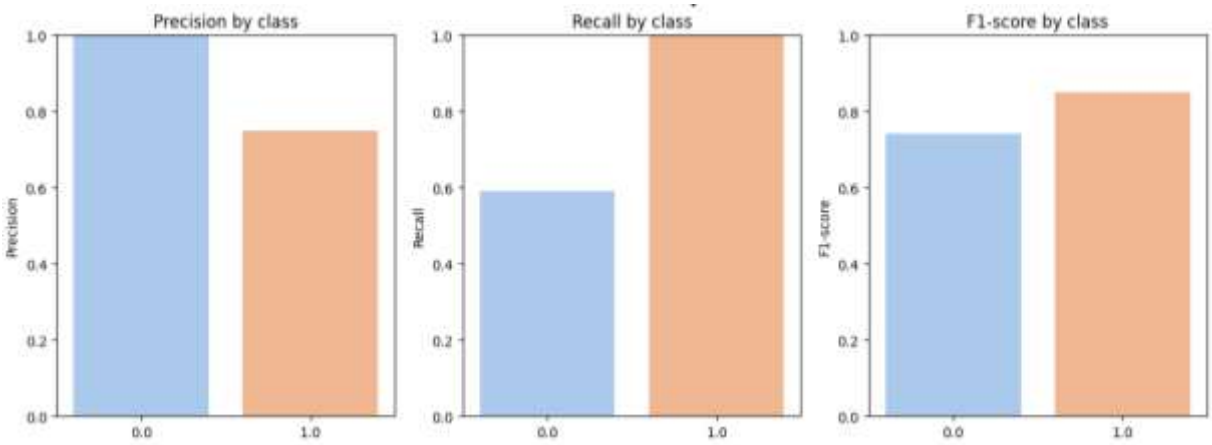
Gambar 7. *Confusion Matrix*

Confusion matrix pada gambar 7 adalah tabel yang menunjukkan jumlah *true positive*, *true negative*, *false positive*, dan *false negative* dari prediksi model. Matriks ini dapat digunakan untuk menghitung berbagai metrik kinerja seperti akurasi, presisi, *recall*, dan skor F1. Dalam matriks khusus ini, sel kiri atas menunjukkan jumlah prediksi negatif yang benar (162), yang berarti jumlah kejadian yang diprediksi dengan benar sebagai negatif (kelas 0). Sel kanan atas menunjukkan jumlah prediksi positif palsu (111), yang berarti jumlah kejadian yang diprediksi secara salah sebagai positif (kelas 1). Sel kiri bawah menunjukkan jumlah prediksi negatif palsu (0), yang berarti jumlah kejadian yang diprediksi secara salah sebagai negatif (kelas 0), dan sel kanan bawah menunjukkan jumlah prediksi positif benar (327), yang berarti jumlah kejadian yang diprediksi dengan benar sebagai positif (kelas 1). Secara keseluruhan, *confusion matrix* menunjukkan bahwa model tersebut telah berkinerja baik, dengan jumlah positif benar dan negatif benar yang tinggi dan jumlah positif salah dan negatif salah yang rendah.

Tabel 1. Hasil Evaluasi

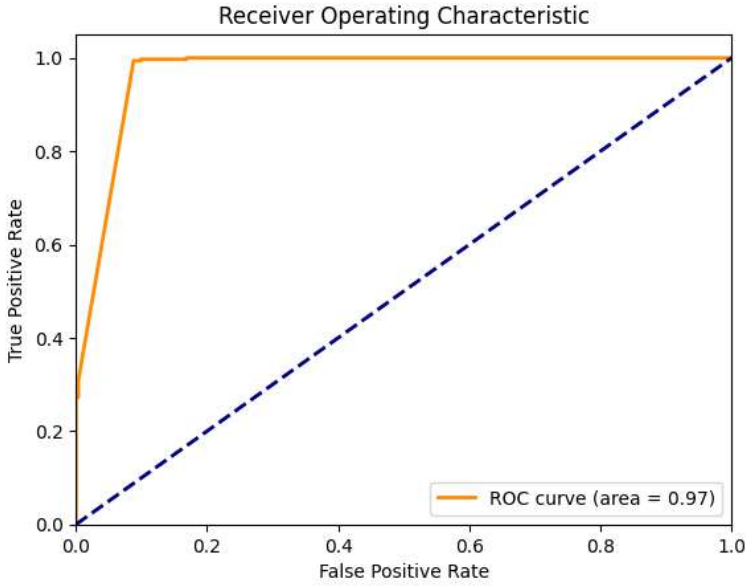
Matriks	Hasil
Akurasi	81,50%
Presisi	74,66%
Recall	100%
F1-*score*	85,49%

Berdasarkan hasil matriks evaluasi kinerja pada tabel 1, model prediksi support vector machine memiliki akurasi keseluruhan sebesar 81.50% pada dataset pengujian. Ini berarti bahwa model tersebut memprediksi label kelas dengan tepat untuk 81,50% sampel dalam dataset pengujian. Selian itu, model ini memiliki skor presisi sebesar 74,66%, *recall* sebesar 100%, dan f1-*score* sebesar 85,49%.



Gambar 8. Laporan Klasifikasi

Laporan klasifikasi pada gambar 8 menunjukkan metrik evaluasi untuk model klasifikasi biner dengan 600 sampel data. Model tersebut mampu mengklasifikasikan kelas positif (1,0) dengan benar dengan presisi 0,75 dan *recall* 1,00. Kelas negatif (0,0) diklasifikasikan dengan presisi sempurna 1,00 tetapi dengan recall lebih rendah 0,59. Akurasi keseluruhan model adalah 0,815. Rata-rata makro F1-*score*, yang merupakan rata-rata harmonik presisi dan *recall*, adalah 0,80, yang menunjukkan keseimbangan yang baik antara presisi dan *recall* di kedua kelas. Rata-rata tertimbang F1-*score* juga sama yaitu 0,80. Secara umum, model dengan presisi tinggi diinginkan karena menunjukkan rasio positif palsu yang rendah, sedangkan *recall* tinggi menunjukkan rasio negatif palsu yang rendah. Kompromi antara presisi dan *recall* dapat dievaluasi lebih lanjut menggunakan F1-*score*. Akurasi keseluruhan yang tinggi dan F1-*score* dari model menunjukkan bahwa model tersebut berkinerja baik dalam tugas klasifikasi.



Gambar 9. Kurva ROC

Kurva ROC pada gambar 9 menggambarkan kinerja model SVM menggunakan kernel RBF dengan hiperparameter C=10 dan gamma=auto. Kurva ROC adalah plot *True Positive Rate* (TPR) terhadap *False Positive Rate* (FPR) untuk nilai ambang batas yang berbeda. Nilai FPR berkisar dari 0 hingga 1 dan menggambarkan proporsi contoh negatif aktual yang diklasifikasikan sebagai positif. Nilai TPR juga berkisar dari 0 hingga 1 dan menggambarkan proporsi contoh positif aktual yang diklasifikasikan sebagai positif. Dari hasil penelitian, kita dapat melihat bahwa model tersebut memiliki TPR yang tinggi, artinya model tersebut mampu mengidentifikasi dengan benar sebagian besar contoh positif dalam kumpulan data. Namun, nilai FPR tidak dapat diabaikan, yang menunjukkan bahwa model tersebut mungkin mengalami kesulitan dalam mengklasifikasikan contoh negatif dengan benar. Nilai ambang batas menunjukkan titik potong yang berbeda untuk menentukan klasifikasi contoh sebagai positif atau negatif. Ambang batas yang lebih tinggi akan menghasilkan TPR yang lebih rendah dan FPR yang lebih rendah, sedangkan ambang batas yang lebih rendah akan menghasilkan TPR yang lebih tinggi dan FPR yang lebih tinggi.

4. KESIMPULAN

Penelitian ini telah berhasil menerapkan algoritma Support Vector Machine untuk mendeteksi anomali pada jaringan komputer. Berdasarkan hasil matriks evaluasi kinerja didapatkan akurasi keseluruhan yaitu sebesar 81,50%, presisi 74,66%, recall 100%, dan f1-*score* sebesar 85,49%. Hasil skor matriks tersebut menunjukkan bahwa model berkinerja dengan baik dalam mengidentifikasi sampel kelas Normal dan Anomali dengan akurat. Secara keseluruhan, hasil ini menunjukkan bahwa model algoritma support vector machine cocok untuk mendeteksi anomali pada jaringan komputer. Kurva ROC merupakan alat yang berguna untuk mengevaluasi kinerja model klasifikasi, dan hasilnya menunjukkan bahwa model SVM dengan hiperparameter yang dipilih sangat efektif dalam membedakan antara contoh positif dan negatif dalam kumpulan data yaitu sebesar 0,97. Secara keseluruhan, model Support Vector Machine yang menggunakan kernel RBF dengan C=10 dan gamma=auto tampaknya merupakan model yang baik untuk kumpulan data ini. Namun, penting untuk dicatat bahwa hasilnya mungkin tidak dapat digeneralisasi ke kumpulan data lain, dan pengujian serta validasi lebih lanjut mungkin diperlukan.

REFERENCES

[1] R. M. Imam, P. Sukarno, and M. A. Nugroho, "Deteksi Anomali Jaringan Menggunakan Hybrid Algorithm," in *e-Proceeding of Engineering*, 2019, pp. 8766–8787. [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/viewFile/9868/9727

[2] F. Nugraha Hermawan, "Deteksi Anomali Pada Data Internet Of Things Menggunakan Model Ensemble Learning," 2021. [Online]. Available: https://repository.uinjkt.ac.id/dspace/bitstream/123456789/65230/1/FAKHRI NUGRAHA HERMAWAN-FST.pdf

[3] Fariadi and M. R. R. Islami, "Deteksi Dini Serangan Pada Website Menggunakan Metode Anomali Based," *JIKO (Jurnal*

Inform. dan Komputer), vol. 5, no. 3, pp. 224–229, 2022, doi: 10.33387/jiko.

[4] C. Nixon, M. Sedky, and M. Hassan, "Autoencoders: A Low Cost Anomaly Detection Method for Computer Network Data Streams," in *ACM International Conference Proceeding Series*, 2020, pp. 58–62. doi: 10.1145/3416921.3416937.

[5] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," pp. 1–50, 2019, [Online]. Available: http://arxiv.org/abs/1901.03407

[6] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab, "Machine Learning for Anomaly Detection: A Systematic Review," *IEEE Access*, vol. 9, pp. 78658–78700, 2021, doi: 10.1109/ACCESS.2021.3083060.

[7] S. Wang, J. F. Balarezo, S. Kandeepan, A. Al-Hourani, K. G. Chavez, and B. Rubinstein, "Machine learning in network anomaly detection: A survey," *IEEE Access*, vol. 9, pp. 152379–152396, 2021, doi: 10.1109/ACCESS.2021.3126834.

[8] F. A. Khan and A. Gumaei, "A Comparative Study of Machine Learning Classifiers for Network Intrusion Detection," in *ICAIS 2019: Artificial Intelligence and Security*, X. Sun, Z. Pan, and E. Bertino, Eds., Cham: Springer International Publishing, 2019, pp. 75–86.

[9] M. K. Hooshmand and Doreswamy, "Machine Learning Based Network Anomaly Detection," *Int. J. Recent Technol. Eng.*, vol. 8, no. 4, pp. 542–548, 2019, doi: 10.35940/ijrte.d7271.118419.

[10] O. Almomani, M. A. Almaiah, A. Alsaaidah, S. Smadi, A. H. Mohammad, and A. Althunibat, "Machine Learning Classifiers for Network Intrusion Detection System: Comparative Study," in *2021 International Conference on Information Technology (ICIT)*, 2021, pp. 440–445. doi: 10.1109/ICIT52682.2021.9491770.

[11] A. T. Zy, A. T. Sasongko, and A. Z. Kamalia, "Penerapan Naïve Bayes Classifier, Support Vector Machine, dan Decision Tree untuk Meningkatkan Deteksi Ancaman Keamanan Jaringan," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 1, pp. 610–617, 2023, doi: 10.30865/klik.v4i1.1134.

[12] I. Riadi, R. Umar, and F. D. Aini, "Analisis Perbandingan Detection Traffic Anomaly Dengan Metode Naive Bayes Dan Support Vector Machine (SVM)," *Ilk. J. Ilm.*, vol. 11, no. 1, pp. 17–24, 2019, doi: 10.33096/ilkom.v11i1.361.17-24.

[13] I. Maulana and Alamsyah, "Optimalisasi Deteksi Serangan DDoS Menggunakan Algoritma Random Forest, SVM, KNN dan MLP pada Jaringan Komputer," *Indones. J. Math. Nat. Sci.*, vol. 2, no. 2, pp. 1–10, 2023.

[14] T. Tan, H. Sama, G. Wijaya, and O. E. Aboagye, "Studi Perbandingan Deteksi Intrusi Jaringan Menggunakan Machine Learning: (Metode SVM dan ANN)," *J. Teknol. dan Inf.*, vol. 13, no. 2, 2023, doi: 10.34010/jati.v13i2.

[15] A. I. Harsapranata, "Analisis Intrusion Detection System Di Internal Jaringan WAN Menggunakan Data Mining : Studi Kasus Pada Astrido Group Jakarta," 2016.

[16] D. Diana, R. E. Indrajit, and E. Dazki, "Komparasi Algoritma Naïve Bayes, Logistic Regression Dan Support Vector Machine pada Klasifikasi File Application Package Kit Android Malware," *Jutisi J. Ilm. Tek. Inform. dan Sist. Inf.*, vol. 11, no. 1, p. 109, 2022, doi: 10.35889/jutisi.v11i1.815.

[17] Kaggle, "Network Intrusion Detection," kaggle.com. Accessed: Jan. 10, 2024. [Online]. Available: https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection