LAPORAN AKHIR PROYEK
PEMBELAJARAN MESIN DASAR

Anomaly Detection in Network Traffic using Pegasos SVM

Dosen Pengampu:

Dr. Elly Matul Imah, M. Kom.
Dinda Galuh Guminta, M. Stat.

Disusun Oleh:

Kelompok 11 Kelas 2023B

Arfan Hafidt Nashrullah	(23031554112)
Sofia Zahira Rohman	(23031554197)
Krisjen Fraulein Hutagalung	(23031554232)

PRODI S1 SAINS DATA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI SURABAYA
2024/2025

1. Pendahuluan

Seiring dengan pesatnya perkembangan teknologi informasi dan komunikasi, jaringan komputer telah menjadi bagian yang tidak dapat dipisahkan dari sebagian besar kegiatan di dunia, baik dalam bidang bisnis, pemerintahan, pendidikan, maupun kehidupan sehari-hari. Namun, pertumbuhan infrastruktur jaringan ini juga diiringi oleh meningkatnya ancaman keamanan, seperti serangan siber, penyusupan (*intrusion*), dan aktivitas tidak wajar lainnya yang dapat merugikan sistem dan penggunanya [1].

Salah satu pendekatan yang semakin banyak digunakan untuk menjawab tantangan ini adalah pemanfaatan *Machine Learning* (ML) [2]. Machine learning merupakan cabang dari *Artificial Intelligence* (AI) yang memungkinkan sistem komputer untuk belajar dari data dan mengambil keputusan atau melakukan prediksi tanpa diprogram secara eksplisit [3]. *Machine Learning* berkembang pesat dan penerapannya dalam konteks keamanan jaringan menjadi solusi potensial karena algoritma yang digunakan dapat menganalisis data dalam jumlah besar secara cepat dan efisien [4]. Selain itu, *machine learning* telah terbukti bekerja dengan efektif melalui berbagai penelitian. Schummer, dkk., menunjukkan bahwa penggunaan algoritma *machine learning*, baik yang bersifat *unsupervised* maupun *supervised*, mampu mengenali pola tersembunyi dalam lalu lintas jaringan dan mendeteksi anomali, masing-masing dengan keunggulannya tersendiri [5].

Support Vector Machine (SVM) merupakan algoritma yang ampuh untuk menemukan pola klasifikasi. Pada penelitian S. Ness, dkk., *Support Vector Machine* (SVM) memiliki akurasi (85%), presisi (0.88%), dan recall (0.86%) yang tinggi jika dibandingkan dengan algoritma machine learning lainnya dalam mendeteksi anomali lalu lintas jaringan [6]. SVM mampu menemukan batas, menggunakan *hyper plane,* untuk memisahkan anomali dari kasus normal dengan overfitting yang rendah. Selain itu, SVM juga efektif dalam menangani hubungan non-linier dalam data melalui penggunaan kernel. Bersama dengan penelitian tersebut, berdasarkan penelitian M. A. Sembiring, dkk., SVM juga terbukti robust terhadap data outlier yang belum di *pre-processing* [7].

Namun demikian, penerapan SVM memerlukan komputasi yang tinggi, terutama pada dataset berukuran besar, karena pembentukan matriks kernel dan proses optimasi yang kompleks [8]. Untuk mengatasi permasalahan tersebut, S. Shwartz dkk. mengembangkan algoritma training Pegasos (Primal Estimated sub-Gradient Solver for SVM), yang memanfaatkan pendekatan stochastic gradient descent (SGD) untuk menyelesaikan bentuk primal SVM dengan lebih efisien [9]. Dengan metode ini, SVM mampu mengurangi kompleksitas komputasi secara signifikan tanpa mengorbankan akurasi secara drastis. Selain itu, Pegasos SVM, sebagai algoritma optimasi untuk melatih model Support Vector Machine (SVM), memberikan solusi klasifikasi yang lebih efisien dalam mendeteksi dataset berskala besar [9].

Pembuatan model *machine learning* yang baik juga tentu tidak lepas dari kualitas proses pengolahan data yang dilakukan. Penelitian Wanyonyi dan Masinde menegaskan bahwa teknik-teknik pengolahan data seperti, pemilihan fitur (*feature selection*), transformasi

data, dan perbaikan ketidakseimbangan data (*data imbalance correction*) merupakan teknik pengolahan data yang paling memengaruhi kinerja model pada berbagai jenis dataset [10].

Kinerja unggul Pegasos dalam studi S. Shwartz, dkk. dan teknik pengolahan data oleh Wanyonyi dan Masinde menjadi dasar kuat bagi pelaksanaan proyek ini. Proyek ini bertujuan untuk merancang dan mengimplementasikan model deteksi anomali pada lalu lintas jaringan menggunakan Pegasos menggunakan dataset NSL-KDD 9. Secara umum, pendekatan yang digunakan dalam proyek ini melibatkan beberapa tahapan utama, yaitu pengumpulan dan *preprocessing* data dan membangun model SVM secara manual, evaluasi kinerja model (seperti akurasi, precision, recall, dan F1-score) seperti yang tertera di gambar 1. Terakhir sistem akan diterapkan (*deploy*) menggunakan Streamlit. Model yang dikembangkan diharapkan mampu mendeteksi aktivitas mencurigakan atau berbahaya.

Dengan dilaksanakannya proyek ini, diharapkan pengembang tidak hanya memperoleh pengetahuan teoritis, tetapi juga pengalaman praktis dalam menerapkan konsep pembelajaran mesin untuk menyelesaikan permasalahan nyata di bidang keamanan jaringan.

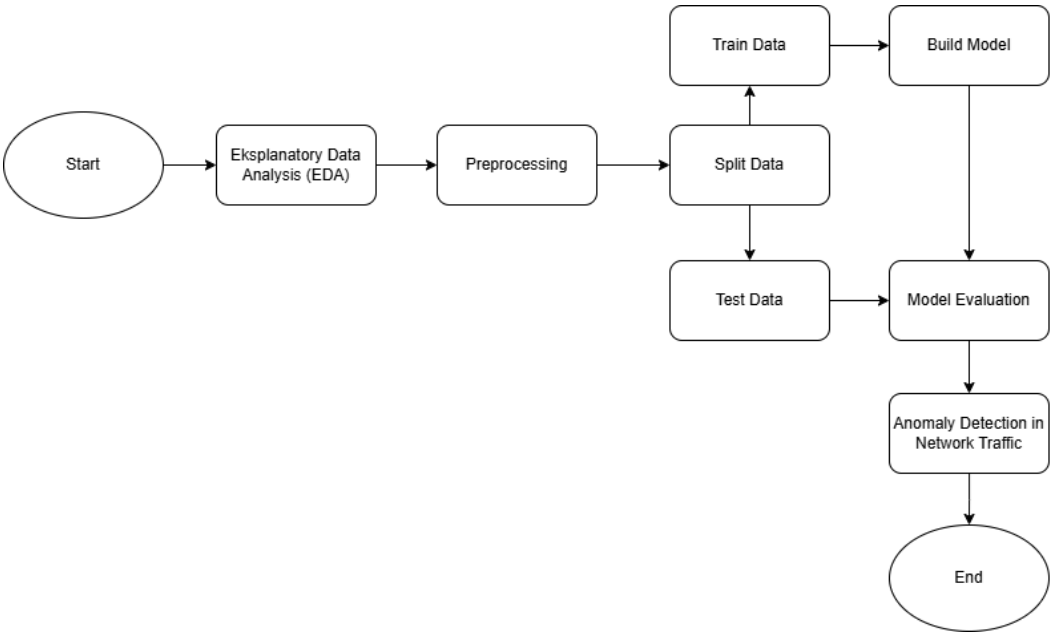
2. Metode

2.1. Dataset

Dataset yang digunakan dalam penelitian ini adalah NSL-KDD 1999 yang diperoleh dari situs Kaggle. NSL-KDD 1999 merupakan versi yang telah diperbaiki dari dataset KDD Cup 1999 (KDD-1999), yang sebelumnya banyak digunakan dalam penelitian di bidang deteksi intrusi. Meskipun sangat populer, KDD-1999 memiliki beberapa kelemahan, seperti banyaknya duplikasi data dan distribusi label yang sangat tidak seimbang, sehingga dapat menyebabkan model bias dan hasil evaluasi yang tidak akurat. Untuk mengatasi permasalahan tersebut, NSL-KDD dikembangkan sebagai versi yang lebih bersih dan representatif dengan menghilangkan data yang redundan dan menyeimbangkan proporsi data lebih baik [12].

Dataset NSL-KDD 1999 terdiri dari dua file utama, yaitu KDD Train+ untuk data pelatihan (training) dan KD Test+ untuk data pengujian (testing). Masing-masing file memuat 42 kolom, yang terdiri atas 41 kolom fitur dan 1 kolom label. Fitur-fitur tersebut mencakup berbagai aspek dari koneksi jaringan, seperti durasi, protokol, layanan, flag, serta berbagai statistik berbasis koneksi.

2.2. Langkah Penelitian



Gambar 1. Diagram alir

2.2.1. Exploratory Data Analysis (EDA)

Proses pembuatan model diawali dengan *Exploratory Data Analysis* yang bertujuan memberikan pemahaman menyeluruh terhadap pola karakteristik data mentah yang akan digunakan [10]. Fokus utama pada tahap ini adalah memeriksa distribusi kelas untuk menghindari ketidakseimbangan data serta mengidentifikasi nilai kosong (NaN) yang dapat memengaruhi kualitas pemodelan.

2.2.2. Pre-Processing

- Memeriksa NaN Values

Memeriksa fitur yang memiliki nilai NaN atau 0 di seluruh baris data dan menghapusnya. Juga memeriksa baris data yang memiliki banyak nilai NaN atau 0 di seluruh fitur.

- Grouping Label

Dataset NSL-KDD 1999 memiliki 42 fitur, 12.5973 data, dan 23 label, serta tidak memiliki nilai NaN. Dataset ini cukup besar dan bersifat *multiclass*. Label pada dataset ini tidak seimbang. Oleh karena itu, label dikelompokkan lagi menjadi empat kategori penyerangan yaitu DoS (Denial of Service), Probe, R2L (Remote to Local), dan U2R (User to Root) [6][12]. Hal ini dilakukan sebagai bentuk mengatasi ketidakseimbangan data. Pembagian lengkapnya dapat dilihat pada tabel 1.

- Encoding

Dataset memiliki fitur dengan tipe object. Fitur ini perlu diubah menjadi format numerik karena machine learning bekerja lebih baik dengan angka [13].

- Transformasi Data

Transformasi data yang dilakukan adalah normalisasi. Normalisasi adalah proses menskalakan fitur numerik ke dalam rentang yang seragam fitur menggunakan z-score.

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

Normalisasi dilakukan untuk mencegah dominasi fitur tertentu dalam pelatihan model dan membuat model *machine learning* konvergen lebih cepat [10].

- Handling Imbalance Data

Selain *grouping* kelas, cara lain untuk mengatasi ketidakseimbangan data adalah dengan menggunakan metode *random under sampling,* yaitu dengan menyeimbangkan jumlah data antar kelas melalui pengurangan jumlah data pada kelas mayoritas secara acak, sambil tetap mempertahankan seluruh data pada kelas minoritas [14].

- Split Data

Tujuan dari pembagian data ke dalam kategori yang berbeda adalah untuk menghindari overfitting [15]. Dalam penelitian ini, digunakan teknik *train-test split* dengan perbandingan 80:20, di mana 80% data digunakan untuk pelatihan (training) dan 20% sisanya untuk pengujian (testing). Pendekatan ini memungkinkan model belajar dari sebagian besar data, sekaligus dievaluasi kinerjanya secara adil menggunakan data yang belum pernah dilihat sebelumnya.

2.2.3. SVM

- Pendekatan Model

Model SVM dalam penelitian ini menggunakan pendekatan Pegasos (*Primal Estimated sub-GrAdient SOlver for SVM*) yang dikembangkan oleh Shalev-Shwartz pada tahun 2011 dan menjadi acuan utama dalam pembuatan model. Pegasos merupakan algoritma optimisasi berbasis *stochastic sub-gradient descent* yang sederhana namun efektif dalam menyelesaikan permasalahan optimisasi pada Support Vector Machines (SVM). Pada setiap iterasi, satu contoh pelatihan dipilih secara acak dan digunakan untuk memperkirakan sub-gradien dari tujuan, dan langkah diambil dalam arah yang berlawanan dengan langkah ukuran yang telah ditentukan sebelumnya [9]. Berikut adalah langkah-langkah yang dijalankan oleh algoritma Pegasos:

1. Inisialisasi

Pada awal proses, Pegasos kernelized menginisialisasi semua koefisien Lagrange (α) dengan nol:

$$\alpha = 0, b = 0$$

2. Pemilihan Contoh Pelatihan Secara Acak

Pada iterasi ke-t, algoritma secara acak memilih satu contoh pelatihan $(x_{it}\, y_{it})$ dari seluruh dataset. Indeks i_t dipilih secara acak dari set $\{1, 2, 3, \ldots\ldots, m\}$ yang merupakan total jumlah contoh pelatihan yang ada.

3. Margin dan Evaluasi Loss

Hitung margin untuk contoh acak terpilih menggunakan kernel:

$$margin = y_i(\sum_{j=i}^{m} \alpha_j y_j K(x_j x_i)+b) \quad (2)$$

Jika :

$$margin < 1$$

maka model membuat kesalahan atau berada di margin.

4. Pembaruan Koefisien

Jika margin < 1, Pegasos memperbarui koefisien α_i dan bias b sebagai berikut:

$$\eta_t = \frac{1}{\lambda t}$$

$$\alpha_i = \alpha_i + \eta_t$$

$$b = b + \eta_t \cdot y_i \quad (3)$$

Jika margin ≥ 1, tidak ada perubahan pada α maupun b.

5. Representasi Model

Model akhir direpresentasikan secara implisit oleh:

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b \quad (4)$$

Di mana adalah $K(x_i, x)$ fungsi kernel.

6. Prediksi

Pertama akan Menghitung nilai fungsi keputusan f(x) untuk setiap data input X, berdasarkan model yang sudah dilatih (menggunakan α, y, dan X support vector). Menggunakan rumus (4). Lalu Prediksi label dilakukan dengan:

$$y' = sign(f(x)) \quad (4)$$

Prediksi label menggunakan sign atau tanda dari suatu bilangan, jika bilangan yang dihasilkan merupakan bilangan negatif maka akan masuk ke kelas -1 dan jika bilangan yang dihasilkan merupakan bilangan positif maka akan masuk ke kelas 1.

- Pendekatan Multi-Class

Untuk menangani kasus klasifikasi multi-kelas pada Support Vector Classification (SVC), digunakan pendekatan One-Vs-One (OvO). Pendekatan ini dianggap efektif karena setiap model hanya fokus membedakan dua kelas pada satu waktu, sehingga proses optimisasi menjadi lebih sederhana dibandingkan pendekatan lain yang menangani banyak kelas sekaligus dalam satu model [16].

Dalam OvO, model klasifikasi dibangun untuk setiap pasangan kelas yang ada. Jika terdapat N kelas, maka akan dilatih sebanyak N(N-1)/2 model. Masing-masing model hanya dilatih menggunakan data dari dua kelas tersebut, bukan seluruh data, sehingga lebih efisien dalam hal komputasi, terutama untuk dataset yang sangat besar atau memiliki distribusi kelas yang tidak seimbang [16].

Prediksi akhir ditentukan dengan menggunakan mekanisme voting dari semua model yang telah dilatih. Kelas yang memperoleh jumlah vote terbanyak akan dipilih sebagai hasil prediksi. Pendekatan ini memungkinkan sistem klasifikasi bekerja secara lebih terfokus dan efisien dalam situasi multi-kelas.

- Penggunaan Kernel

Dalam penelitian ini digunakan kernel **Radial Basis Function (RBF)** untuk menangani masalah klasifikasi non-linear pada model **Pegasos SVM**. Kernel RBF merupakan salah satu kernel paling umum dalam SVM karena kemampuannya memproyeksikan data ke ruang fitur berdimensi lebih tinggi, sehingga memungkinkan pemisahan antar kelas yang tidak dapat dipisahkan secara linear di ruang asli. Bentuk dasar kernel RBF dinyatakan sebagai berikut:

$$K(xi, xj) = exp(-\gamma \| x_i - x_j \|^2) \quad (5)$$

Dengan :

- $K(xi, xj)$ adalah hasil dari kernel yang mengukur kesamaan antara dua titik data x_i dan x_j
- γ adalah parameter yang mengontrol lebar dari fungsi RBF, yang juga dikenal sebagai parameter skala atau "bandwidth" kernel.

- $\| x_i - x_j \|^2$ adalah jarak Euclidean yang dikuadratkan antara dua titik data x_i dan x_j

Dengan menerapkan teknik *kernel trick*, pemetaan ke ruang berdimensi tinggi dilakukan secara implisit, sehingga menghindari komputasi langsung yang mahal. Hal ini membuat proses pelatihan lebih efisien secara komputasi, namun tetap efektif dalam membentuk *hyperplane* untuk memisahkan data non-linear [17].

Berikut adalah beberapa keunggulan utama kernel RBF meliputi:

- Kemampuan menangani data non-linear: Kernel RBF memungkinkan pemisahan data yang tidak dapat dipisahkan secara linear di ruang asli, sehingga cocok untuk klasifikasi yang kompleks [17].

- Efisiensi komputasi: Tidak memerlukan pemetaan eksplisit karena kernel hanya menghitung kesamaan antar titik data, yang secara signifikan menghemat waktu dan memori [18].

2.2.4. Evaluasi Model

Setelah model Pegasos SVM dilatih menggunakan data latih, langkah selanjutnya adalah melakukan evaluasi terhadap performa model menggunakan data uji. *Confusion matrix* digunakan untuk mengamati secara rinci jumlah prediksi benar dan salah pada setiap kelas [19].

Tabel 1. *Confusion matrix*

	Kelas Aktual Positif	**Kelas Aktual Negatif**
Prediksi Positif	True Positive (TP)	True Negative (TN)
Prediksi Negatif	False Positive (FP)	False Negative (FN)

Selain *confusion matrix*, beberapa evaluasi juga digunakan untuk mengevaluasi kemampuan generalisasi dari model yang telah dilatih [19].

Tabel 2. Evaluasi Metrik

Evaluasi Metrik	**Rumus**	**Deskripsi**
Akurasi	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ (6)	Akurasi menunjukkan proporsi prediksi yang benar dari seluruh data uji.
Presisi	$Precision = \frac{TP}{TP+FP}$ (7)	Precision mengukur ketepatan model dalam memprediksi suatu kelas

Recall	$Recall = \frac{TP}{TP+FN}$ (8)	Recall menunjukkan seberapa baik model dapat mengenali semua instance dari kelas tersebut.
F1-Score	$Accuracy = 2 . \frac{Precision \, . \, Recall}{Precision + Recall}$ (9)	F1-score memberikan gambaran menyeluruh terhadap kinerja model, terutama dalam kondisi distribusi label yang tidak seimbang

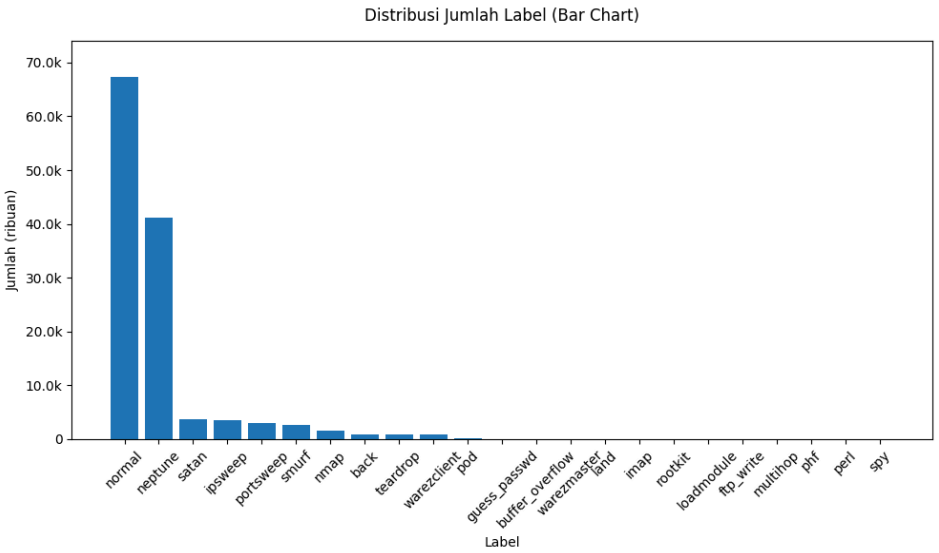
Evaluasi ini penting untuk mengetahui apakah model hanya baik dalam mengenali kelas mayoritas, atau mampu mengklasifikasikan semua kategori secara adil.

3. Hasil dan Pembahasan

Dataset terdiri dari 125.973 baris dan 42 kolom fitur. Dataset ini tidak mengandung nilai NaN, namun distribusi labelnya tidak seimbang (imbalanced).

Tabel 3. Jumlah setiap label dataset NSL-KDD 1999

Label	**Jumlah**	**Label**	**Jumlah**
normal	67343	buffer_overflow	30
neptune	41214	warezmaster	20
satan	3633	land	18
ipsweep	3599	imap	11
portsweep	2931	rootkit	10
smurf	2646	loadmodule	9
nmap	1493	ftp_write	8
back	956	multihop	7
teardrop	892	phf	4
warezclient	890	perl	3
pod	201	spy	2
guess_passwd	53		

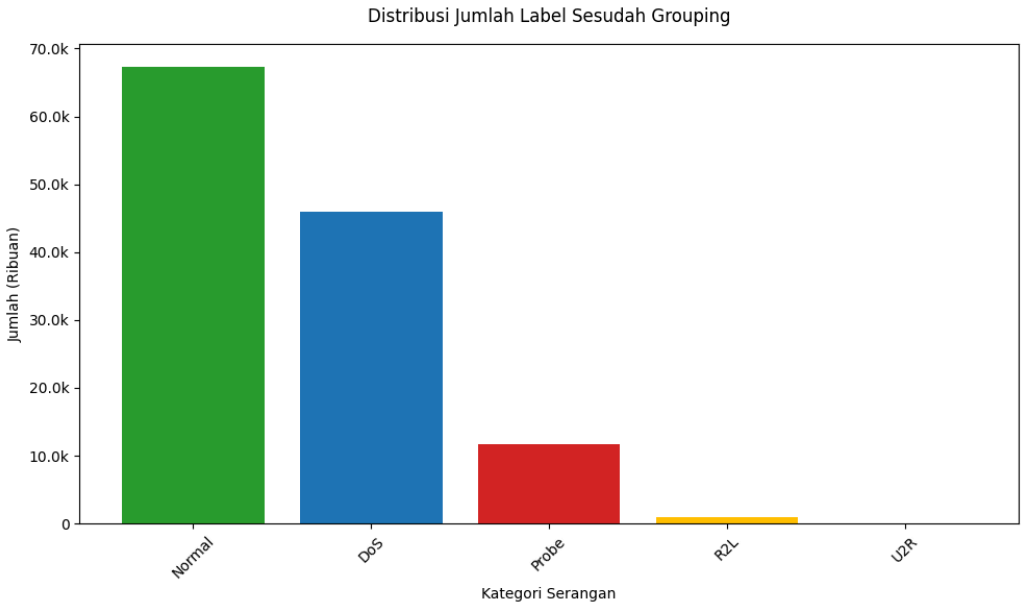


Gambar 2. Distribusi jumlah kelas sebelum *grouping*

Salah satu cara yang dipakai untuk mengatasi ketidakseimbangan data NSL-KDD 1999 label dikelopokkan lagi menjadi empat kategori serangan.

Tabel 4. Pembagian Kelas Besar Label Dataset

Kategori	Jenis Serangan	Deskripsi Singkat
DoS	back, land, neptune, pod, smurf, teardrop, mailbomb, apache2, processtable, udpstorm	Serangan yang bertujuan membuat layanan tidak tersedia dengan membanjiri sistem
Probe	satan, ipsweep, nmap, portsweep, mscan, saint	Serangan yang mencoba memetakan jaringan atau mencari celah keamanan
R2L	guess_passwd, ftp_write, imap, phf, multihop, warezmaster, warezclient, spy, sendmail, named, snmpgetattack, snmpguess, xlock, xsnoop, worm	Serangan dari luar jaringan yang mencoba mendapatkan akses lokal
U2R	buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack, xterm	Serangan dari pengguna biasa yang mencoba mendapatkan hak akses root (superuser)



Gambar 3. Distribusi jumlah label setelah grouping label serangan

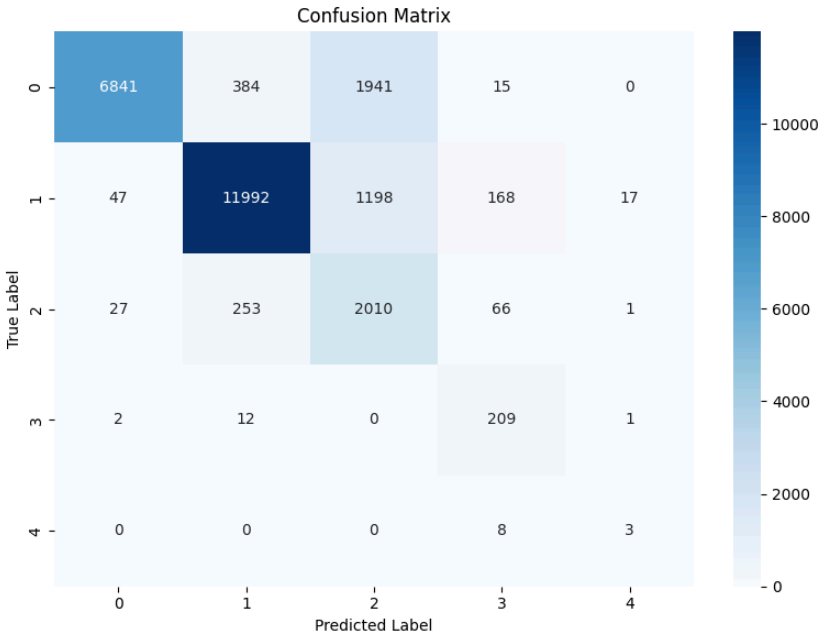
Setelah tahap pemeriksaan awal terhadap data, fitur-fitur yang bertipe objek (seperti kategori atau teks) perlu dikonversi terlebih dahulu ke dalam bentuk numerik menggunakan teknik encoding. Proses ini penting karena sebagian besar algoritma machine learning, termasuk SVM, hanya dapat memproses data numerik. Setelah data dikonversi, dilakukan normalisasi agar setiap fitur berada dalam skala yang seragam. Hal ini bertujuan untuk menghindari dominasi fitur tertentu yang memiliki rentang nilai lebih besar, serta memastikan bahwa proses pelatihan model berlangsung secara optimal dan efisien. Kemudian data di split train dan test dengan perbandingan 80:20. Data train dimasukkan ke dalam model pegasus yang telah dibuat dan didapatkan hasil sebagai berikut:

- Matrik Evaluasi

Tabel 5. Tabel evaluasi Model Pegasos SVM

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.99	0.75	0.85	9181
1	0.95	0.89	0.92	13422
2	0.39	0.85	0.54	2357
3	0.45	0.93	0.61	224
4	0.14	0.27	0.18	11
accuracy			0.84	25195
macro avg	0.58	0.74	0.62	25195
weighted avg	0.91	0.84	0.86	25195

- Confusion Matrix



Gambar 4. Confusion Matrix Pegasos SVM

Model klasifikasi yang dievaluasi menunjukkan performa keseluruhan yang cukup baik, dengan akurasi sebesar 84%, artinya sebagian besar data berhasil diklasifikasikan dengan benar. Metrik precision dan recall menunjukkan bahwa model sangat andal dalam mengenali kelas mayoritas, khususnya kelas 0 dan 1, dengan F1-score masing-masing sebesar 0.85 dan 0.92. Namun, performa menurun pada kelas dengan jumlah data yang lebih sedikit seperti kelas 2, 3, dan terutama kelas 4. Meskipun recall untuk kelas 2 dan 3 tinggi (0.85 dan 0.93), precision-nya rendah, yang menunjukkan model sering salah dalam memberi label kelas tersebut. Kelas 4 memiliki performa terendah, dengan precision hanya 0.14 dan F1-score 0.18, menunjukkan bahwa model kesulitan dalam mengenali kelas dengan representasi data yang sangat kecil. Nilai rata-rata makro F1-score sebesar 0.62 mengindikasikan ketidakseimbangan performa antar kelas. Oleh karena itu, meskipun akurasi cukup tinggi, peningkatan perlu difokuskan pada kelas minoritas, seperti dengan penyeimbangan data atau pemberian bobot khusus pada kelas kecil, agar model dapat bekerja lebih adil dan akurat pada seluruh kategori.

4. Kesimpulan

Model Pegasos SVM yang diterapkan pada dataset NSL-KDD 1999 menunjukkan performa klasifikasi yang cukup baik secara umum, dengan akurasi keseluruhan sebesar 84%. Model berhasil mengenali kelas mayoritas dengan sangat baik, khususnya kelas 0 (normal) dan kelas 1 (DoS), yang ditunjukkan oleh nilai F1-score masing-masing sebesar 85% dan 92%. Namun, performa menurun drastis pada kelas minoritas seperti kelas 4 (U2R), yang hanya memiliki precision 14% dan F1-score 18%, serta kelas 2 dan 3 yang meskipun memiliki recall tinggi, mengalami penurunan presisi. Hal ini mengindikasikan bahwa

ketidakseimbangan distribusi label dalam data berdampak signifikan terhadap kemampuan model dalam mengenali serangan dengan frekuensi rendah.

Daftar Pustaka

[1] "View of The Application of Computer Information Technology in Network Security under the Background of Big Data." Accessed: Jun. 08, 2025. [Online]. Available: https://centuryscipub.com/index.php/JTPMS/article/view/703/615

[2] 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.

[3] I. El Naqa and M. J. Murphy, "What Is Machine Learning?," *Machine Learning in Radiation Oncology*, pp. 3–11, 2015, doi: 10.1007/978-3-319-18305-3_1.

[4] A. Pekar and R. Jozsa, "Evaluating ML-based anomaly detection across datasets of varied integrity: A case study," *Computer Networks*, vol. 251, p. 110617, Sep. 2024, doi: 10.1016/J.COMNET.2024.110617.

[5] P. Schummer, A. del Rio, J. Serrano, D. Jimenez, G. Sánchez, and Á. Llorente, "Machine Learning-Based Network Anomaly Detection: Design, Implementation, and Evaluation," *AI 2024, Vol. 5, Pages 2967-2983*, vol. 5, no. 4, pp. 2967–2983, Dec. 2024, doi: 10.3390/AI5040143.

[6] S. Ness, V. Eswarakrishnan, H. Sridharan, V. Shinde, N. V. P. Janapareddy, and V. Dhanawat, "Anomaly Detection in Network Traffic using Advanced Machine Learning Techniques," *IEEE Access*, 2025, doi: 10.1109/ACCESS.2025.3526988.

[7] M. A. Sembiring, H. Saputra, R. A. Yusda, S. Sutarman, and E. B. Nababan, "PERFORMANCE OF ROBUST SUPPORT VECTOR MACHINE CLASSIFICATION MODEL ON BALANCED, IMBALANCED AND OUTLIERS DATASETS," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 10, no. 1, pp. 208–215, Aug. 2024, doi: 10.33480/JITK.V10I1.5272.

[8] Z. Akram-Ali-Hammouri, M. Fernández-Delgado, E. Cernadas, and S. Barro, "Fast Support Vector Classification for Large-Scale Problems," *IEEE Trans Pattern Anal Mach Intell*, vol. 44, no. 10, pp. 6184–6195, Oct. 2022, doi: 10.1109/TPAMI.2021.3085969.

[9] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for SVM," *Math Program*, vol. 127, no. 1, pp. 3–30, Mar. 2011, doi: 10.1007/S10107-010-0420-4/METRICS.

[10] Everleen Nekesa Wanyonyi and Newton Wafula Masinde, "The Impact of Data Preprocessing on Machine Learning Model Performance: A Comprehensive

Examination," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 11, no. 2, pp. 3814–3827, Apr. 2025, doi: 10.32628/CSEIT25112854.

[11] "NSL-KDD99 Dataset." Accessed: Jun. 09, 2025. [Online]. Available: https://www.kaggle.com/datasets/kaggleprollc/nsl-kdd99-dataset?select=KDDTest%2B.txt

[12] M. Zhang, B. Xu, and J. Gong, "An Anomaly Detection Model Based on One-Class SVM to Detect Network Intrusions," *Proceedings - 11th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2015*, pp. 102–107, Feb. 2016, doi: 10.1109/MSN.2015.40.

[13] N. Kosaraju, S. R. Sankepally, and K. Mallikharjuna Rao, "Categorical Data: Need, Encoding, Selection of Encoding Method and Its Emergence in Machine Learning Models—A Practical Review Study on Heart Disease Prediction Dataset Using Pearson Correlation," *Lecture Notes in Networks and Systems*, vol. 551, pp. 369–382, 2023, doi: 10.1007/978-981-19-6631-6_26.

[14] A. El Hariri, M. Mouiti, O. Habibi, and M. Lazaar, "Improving Deep Learning Performance Using Sampling Techniques for IoT Imbalanced Data," *Procedia Comput Sci*, vol. 224, pp. 180–187, Jan. 2023, doi: 10.1016/J.PROCS.2023.09.026.

[15] "(PDF) IDEAL DATASET SPLITTING RATIOS IN MACHINE LEARNING ALGORITHMS: GENERAL CONCERNS FOR DATA SCIENTISTS AND DATA ANALYSTS." Accessed: Jun. 13, 2025. [Online]. Available: https://www.researchgate.net/publication/358284895_IDEAL_DATASET_SPLITTING_RATIOS_IN_MACHINE_LEARNING_ALGORITHMS_GENERAL_CONCERNS_FOR_DATA_SCIENTISTS_AND_DATA_ANALYSTS

[16] G. Genesis, I. F. Gomes, and J. A. Barbosa, "A Comparison between the One-vs-All (Ova) and One-vs-One (Ovo) Strategies for Lithofacies Classification Using Support Vector Machines and Logistic Regression Models," 2025, doi: 10.2139/SSRN.5085740.

[17] G. O. Anyanwu, C. I. Nwakanma, J. M. Lee, and D. S. Kim, "RBF-SVM kernel-based model for detecting DDoS attacks in SDN integrated vehicular network," *Ad Hoc Networks*, vol. 140, p. 103026, Mar. 2023, doi: 10.1016/J.ADHOC.2022.103026.

[18] A. Razaque, M. Ben Haj Frej, M. Almi'ani, M. Alotaibi, and B. Alotaibi, "Improved Support Vector Machine Enabled Radial Basis Function and Linear Variants for Remote Sensing Image Classification," *Sensors 2021, Vol. 21, Page 4431*, vol. 21, no. 13, p. 4431, Jun. 2021, doi: 10.3390/S21134431.

[19] H. M and S. M.N, “A Review on Evaluation Metrics for Data Classification Evaluations,” *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 01–11, Mar. 2015, doi: 10.5121/IJDKP.2015.5201.

[20] “Confusion Matrix | PDF | Accuracy And Precision | Statistical Classification.” Accessed: Jun. 14, 2025. [Online]. Available: https://www.scribd.com/document/668315655/Confusion-Matrix