BAB IV
HASIL DAN PEMBAHASAN

4.1.Data Description

Dataset penelitian ini diambil dari eksperimen *Canadian Institute for Cybersecurity* dalam mendeteksi serangan DDoS, merancang detektor waktu nyata dengan beban komputasi rendah tetap menjadi tantangan yang signifikan. Dengam menciptakan dataset yang tersedia dan mengusulkan taksonomi baru untuk serangan DDoS serta menghasilkan dataset CICDDoS2019 yang memperbaiki kelemahan dataset sebelumnya dengan memanfaatkan perangkat lunak *CICFlowMeter* [48].

Dataset DDoS berisi sebanyak 9.807.779 record dengan 11 kelas dan 88 fitur. Dari data set tersebut akan di ambil sampel dengan Teknik pengambilan sampel yang diterapkan adalah *Stratified Sampling* (Pengambilan Sampel Berstrata), dengan proporsi 30% dari setiap kelas, untuk memastikan distribusi kelas tetap seimbang sesuai dengan proporsi awal. Dataset yang dihasilkan memiliki jumlah total sebanyak 3.002.134. Jenis serang yang terdapat dalam berbagai jenis serangan DDoS, termasuk DDoS reflektif (seperti DNS, LDAP, MSSQL, dan TFTP), NetBIOS, NTP, WebDDoS, UDP, UDP-Lag dan SYN. Fitur-fitur yang terdapat dalam dataset ini terlah di jelaskan pada bab sebelumnya yang terdapat pada Tabel 3. 5, berikut ini merupakan tampilan dari dataset yang telah yang digunakan.

Tabel 4. 1 Pengambilan Sampling Dataset CIC-DDoS2019 sebanyak 30 %

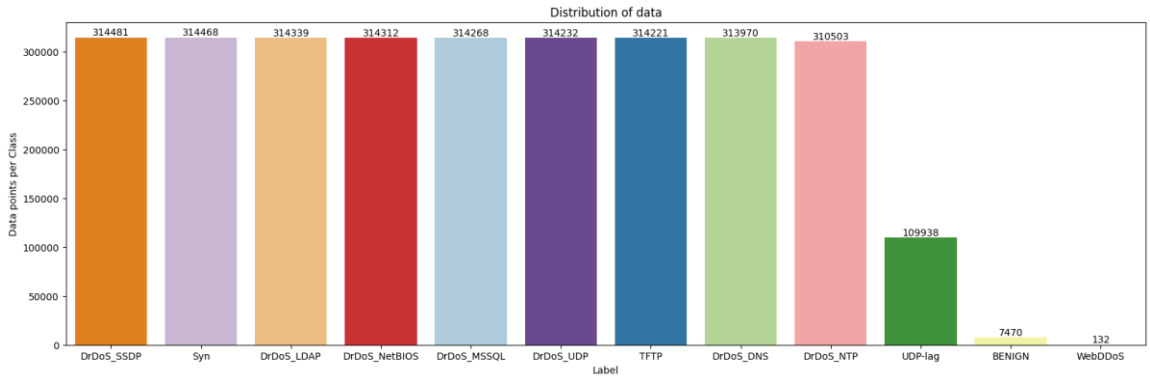
	Unnamed: 0	Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	...	Idle Min	Simillar HTTP	Inbound	Label
0	101522	172.16.0.5-192.168.50.1-18644-43794-17	172.16.0.5	18644	192.168.50.1	43794	17	...	0.0	0	1	DrDoS_MSSQL
1	4557	172.16.0.5-192.168.50.1-42259-26170-17	172.16.0.5	42259	192.168.50.1	26170	17	...	0.0	0	1	DrDoS_MSSQL
2	13353	172.16.0.5-192.168.50.1-35634-65491-17	172.16.0.5	35634	192.168.50.1	65491	17	...	0.0	0	1	TFTP
3	9041	172.16.0.5-192.168.50.1-703-45785-17	172.16.0.5	703	192.168.50.1	45785	17	...	0.0	0	1	DrDoS_DNS
4	11363	172.16.0.5-192.168.50.1-43530-38342-6	172.16.0.5	43530	192.168.50.1	38342	6	...	0.0	0	1	UDP-lag
5	6728	172.16.0.5-192.168.50.1-62502-13799-17	172.16.0.5	62502	192.168.50.1	13799	17	...	0.0	0	1	TFTP
6	16673	172.16.0.5-192.168.50.1-63305-6293-17	172.16.0.5	63305	192.168.50.1	6293	17	...	0.0	0	1	TFTP
7	331144	172.16.0.5-192.168.50.1-59327-60241-6	172.16.0.5	59327	192.168.50.1	60241	6	...	0.0	0	1	UDP-lag
8	5092	172.16.0.5-192.168.50.1-810-3421-17	172.16.0.5	810	192.168.50.1	3421	17	...	0.0	0	1	DrDoS_NTP
9	11262	172.16.0.5-192.168.50.1-710-37218-17	172.16.0.5	710	192.168.50.1	37218	17	...	0.0	0	1	DrDoS_NetBIOS
...	...	...	...	...	...	...	...	...	...	...	...	...
2942329	53856	172.16.0.5-192.168.50.1-62468-18871-17	172.16.0.5	62468	192.168.50.1	18871	17	...	0.0	0	1	TFTP
2942330	268918	172.16.0.5-192.168.50.1-24740-58234-6	172.16.0.5	24740	192.168.50.1	58234	6	...	0.0	0	1	Syn
2942331	3841	172.16.0.5-192.168.50.1-585-52305-17	172.16.0.5	585	192.168.50.1	52305	17	...	0.0	0	1	DrDoS_LDAP
2942332	8992	172.16.0.5-192.168.50.1-30566-21306-17	172.16.0.5	30566	192.168.50.1	21306	17	...	0.0	0	1	DrDoS_MSSQL
2942333	14269	172.16.0.5-192.168.50.1-769-53683-17	172.16.0.5	769	192.168.50.1	53683	17	...	0.0	0	1	DrDoS_LDAP

Pada Tabel 4.1 menunjukkan hasil pengambilan sampling sebanyak 30% dari dataset CIC-DDoS2019, yang digunakan dalam penelitian ini. Dataset ini berisi lalu lintas jaringan yang terdiri dari berbagai jenis protokol, alamat IP sumber dan

tujuan, serta port yang digunakan. Setiap baris data merepresentasikan satu aliran jaringan (*flow*) dengan identitas unik pada kolom Flow ID dan informasi koneksi seperti Source IP, Destination IP, Source Port, Destination Port, serta Protocol yang digunakan (misalnya protokol 6 untuk TCP dan 17 untuk UDP).

Selain itu, terdapat kolom-kolom lain seperti Idle Min, Inbound, dan Similar HTTP, yang mencerminkan fitur perilaku lalu lintas, serta kolom Label yang menjadi penanda jenis trafik apakah termasuk aktivitas serangan seperti *DrDoS_MSSQL*, *TFTP*, *UDP-lag*, *SYN*, atau merupakan lalu lintas normal (*BENIGN*). Label ini digunakan sebagai target klasifikasi dalam proses pelatihan model deteksi serangan DDoS. Pengambilan sampel sebanyak 30% ini dilakukan untuk mengurangi beban komputasi selama proses eksplorasi data, praproses, serta pelatihan model, namun tetap mempertahankan representasi semua kelas serangan dan aktivitas normal dalam dataset. Struktur tabel yang kompleks dan beragam ini mencerminkan karakteristik lalu lintas jaringan aktual dan menjadi tantangan sekaligus peluang dalam merancang sistem deteksi serangan siber yang efektif.

Sebagai bagian dari penyajian karakteristik dataset, distribusi data pada masing-masing kelas serangan DDoS ditampilkan untuk memberikan gambaran mengenai proporsi jumlah data yang tersedia. Informasi mengenai sebaran ini penting untuk menilai tingkat keseimbangan antar kelas, yang berpengaruh terhadap efektivitas model dalam melakukan klasifikasi. Ketidak seimbangan data antar kelas dapat berpotensi menimbulkan bias pada hasil prediksi dan menurunkan akurasi model pembelajaran mesin.



Gambar 4. 1 Distribusi dataset DDOS

Berdasarkan Gambar 4.1 yang menampilkan distribusi data per kelas dalam dataset CICDDoS2019, dapat diamati bahwa jumlah titik data (*data points*) antar

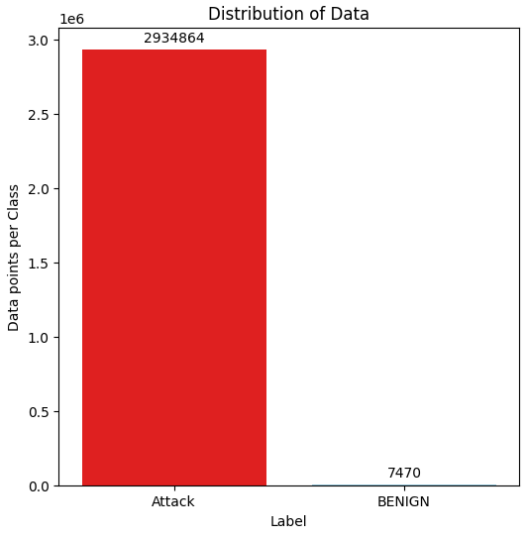
kelas tidak merata secara keseluruhan. Sebagian besar kelas serangan (mayoritas) memiliki jumlah data yang relatif seimbang, yakni berkisar antara 310.000 hingga 314.000 data per kelas, yang digambarkan dengan batang-batang berwarna cerah seperti jingga (DrDoS_SSDP), ungu muda (Syn), krem (DrDoS_LDAP), merah (DrDoS_NetBIOS), biru muda (DrDoS_MSSQL), ungu tua (DrDoS_UDP), biru tua (TFTP), hijau muda (DrDoS_DNS), dan pink (DrDoS_NTP). Keseimbangan ini penting untuk membentuk model klasifikasi yang tidak bias terhadap kelas tertentu dalam kelompok mayoritas.

Namun demikian, terdapat ketimpangan signifikan pada beberapa kelas minoritas yang digambarkan dengan warna yang mencolok berbeda. Kelas UDP-lag yang ditampilkan dalam warna hijau tua hanya memiliki 109.918 titik data, jumlah ini jauh lebih rendah dibandingkan kelas-kelas mayoritas. Sementara itu, kelas BENIGN yang digambarkan dengan warna kuning pucat hanya memiliki 7.470 data, dan kelas WebDDoS yang divisualisasikan dengan warna ungu gelap hampir kehitaman hanya mencakup 132 data. Kelas-kelas minoritas ini jauh di bawah rata-rata dan mencerminkan ketidakseimbangan data (*class imbalance*) yang berpotensi mengganggu kinerja model dalam mendeteksi jenis serangan tertentu. Perbedaan warna pada grafik tidak hanya berfungsi sebagai pembeda visual antar label, tetapi juga menekankan perbedaan tingkat frekuensi data per kelas secara intuitif.

Distribusi setiap kelas dalam dataset CICDDoS2019, yang menunjukkan variasi jumlah data pada masing-masing jenis serangan seperti DrDoS_SSDP, DrDoS_LDAP, SYN, UDP-lag, WebDDoS, dan lainnya, serta satu kelas BENIGN sebagai representasi aktivitas normal. Dari distribusi tersebut, dapat dilihat bahwa sebagian besar kelas memiliki jumlah data yang relatif seimbang, sementara beberapa kelas minoritas memiliki jumlah data yang jauh lebih sedikit.

Untuk mendapatkan pemahaman yang lebih sederhana dan menyeluruh terhadap ketimpangan data, seluruh kelas tersebut kemudian dikelompokkan berdasarkan kategorinya, yaitu aktivitas normal (BENIGN) dan aktivitas anomali (Attack). Dalam konteks ini, semua kelas serangan (DDoS) dianggap sebagai representasi anomali, sedangkan kelas BENIGN mewakili lalu lintas normal. Pengelompokan ini bertujuan untuk memudahkan analisis awal dalam deteksi

anomali serta menyoroti tantangan ketidakseimbangan kelas *(class imbalance)* secara umum sebelum dilakukan pendekatan balancing dan klasifikasi lanjutan berdasarkan jenis serangan spesifik. Serta menjadi fokus penelitian terhadap anomali serangan DDOS.



Gambar 4. 2 Distibusi Data Serangan dan Normal

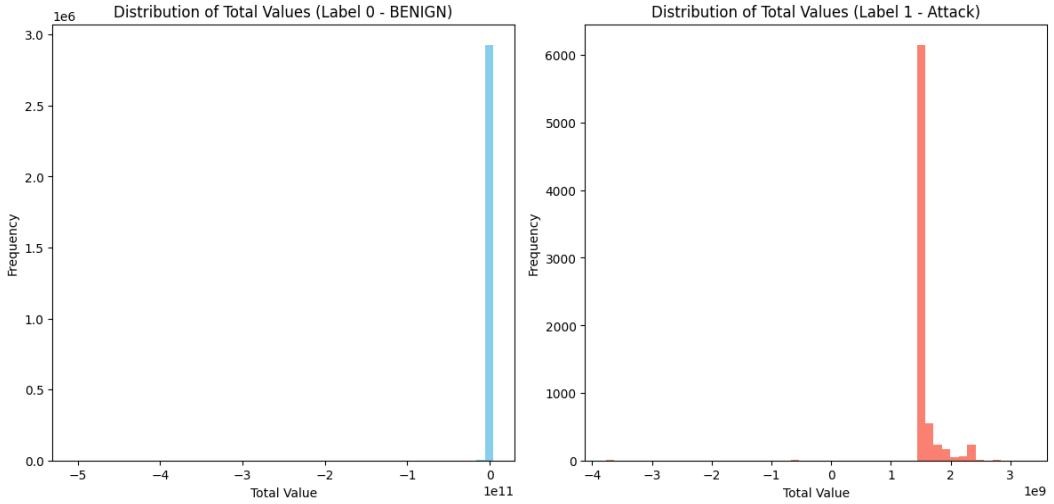
Gambar 4.2 menunjukkan bahwa sebagian besar data distribusi mewakili aktivitas serangan siber (normal (BENIGN) dan berbahaya (Attack)), sementara data yang merepresentasikan aktivitas normal jumlahnya sangat terbatas. Dari total data yang diamati, terdapat 2.934.864 data yang dikategorikan sebagai *Attack*, sedangkan hanya 7.470 data yang termasuk dalam kategori *BENIGN*. Ketimpangan ini sangat mencolok dan mencerminkan kondisi ketidakseimbangan kelas (class imbalance) yang ekstrem.

Distribusi ini menunjukkan bahwa sebagian besar data mewakili aktivitas serangan siber, sementara data yang merepresentasikan aktivitas normal jumlahnya sangat terbatas. Ketidakseimbangan ini dapat menyebabkan bias dalam pelatihan model deteksi, di mana model cenderung mempelajari pola dari kelas mayoritas (Attack) dan mengabaikan kelas minoritas (BENIGN), sehingga berpotensi menghasilkan *false positive* atau *false negative* yang tinggi.

4.2.Pre-Prosesing

Tujuan dari tahap awal pengolahan data adalah untuk menyediakan data mentah dalam format yang lebih terorganisir, bersih, dan siap untuk digunakan

untuk pemodelan atau analisis lebih lanjut. Terdapat empat tahapan pre-prosesing data dalam penelitian ini: pembersihan data, normalisasi, pembagian data, dan balancing data, seperti yang dijelaskan pada bab sebelumnya. Data yang tidak akurat, tidak lengkap, tidak konsisten, atau tidak relevan diidentifikasi, diperbaiki, atau dihapus pada tahap pembersihan data. Setelah proses cleaning selesai, data kemudian dinormalisasi. Normalisasi bertujuan untuk menyamakan skala antar fitur sehingga tidak ada fitur yang mendominasi proses pelatihan model karena perbedaan skala nilai. Dalam penelitian ini, digunakan metode Min-Max Normalization untuk mengubah rentang nilai semua fitur ke dalam interval [0,1].

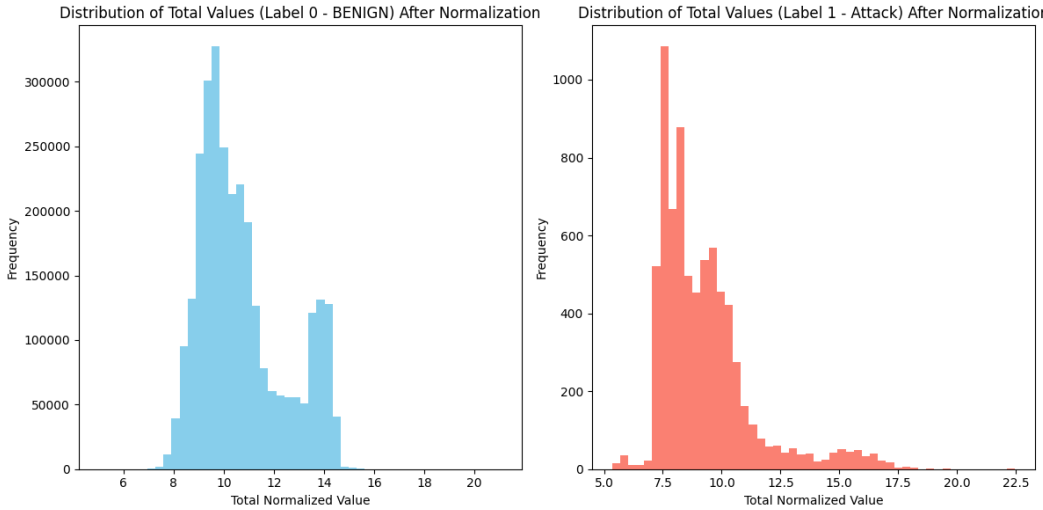


Gambar 4. 3 Distribusi data sebelum normalisasi

Berdasarkan Gambar 4.3 distribusi total nilai sebelum dilakukan proses normalisasi, terlihat dengan jelas bahwa dataset memiliki variasi nilai yang sangat luas, dengan skala antar fitur yang ekstrem. Pada histogram, total akumulasi nilai untuk kelas BENIGN (Label 0) mencapai sekitar 30 juta (3×10^7), sedangkan pada kelas Attack (Label 1) rentang nilai total berada pada skala yang jauh lebih tinggi, yaitu hingga 3 miliar (3×10^9). Distribusi ini memperlihatkan ketidakseimbangan yang signifikan, ditandai dengan puncak distribusi yang sangat tajam dan keberadaan outlier dalam jumlah besar. Ini menunjukkan bahwa terdapat sejumlah fitur numerik yang dominan karena perbedaan skala satuan antara fitur masih dibiarkan tanpa perubahan.

Kondisi ini dapat menyebabkan masalah signifikan dalam fase pelatihan model machine learning, karena algoritma optimasi seperti *gradient descent* akan cenderung lebih sulit menemukan titik minimum global ketika dihadapkan pada

data dengan skala yang sangat bervariasi. Fitur yang memiliki nilai sangat tinggi akan mengendalikan proses penghitungan jarak atau bobot, sementara fitur dengan skala lebih rendah akan berkurang kontribusinya. Akibatnya, model dapat menjadi bias terhadap pola data tertentu, cenderung mengalami overfitting pada kelas yang dominan, dan meningkatkan kemungkinan ketidakstabilan dalam hasil prediksi. Data yang tidak dinormalisasi menyulitkan model untuk secara konsisten mengenali pola serangan DDoS pada data nyata.



Gambar 4. 4 Distribusi data setelah normalisasi

Setelah proses normalisasi dilakukan, pada Gambar 4.4 menjukan distribusi total nilai pada kedua label menunjukkan pergeseran yang signifikan ke arah yang lebih terkontrol dan optimal. Berdasarkan histogram, total nilai untuk kelas BENIGN (Label 0) terdistribusi dalam rentang antara sekitar 6 hingga 16, dengan konsentrasi nilai terbanyak berada pada kisaran 9 hingga 11. Sementara itu, total nilai untuk kelas Attack (Label 1) terdistribusi dalam rentang sekitar 5 hingga 22, dengan puncak frekuensi tertinggi berada pada kisaran 7 hingga 9. Setelah normalisasi, tidak lagi ditemukan nilai ekstrem (*outlier*) dengan skala besar seperti ratusan juta atau miliaran sebagaimana terlihat pada distribusi sebelum normalisasi. Rentang nilai yang lebih kecil dan seragam ini menunjukkan bahwa proses normalisasi berhasil menyamakan skala antar fitur, sehingga tidak ada satu fitur pun yang secara dominan memengaruhi proses pelatihan model.

Selain itu, pola distribusi nilai pada kedua label tampak menjadi lebih halus, simetris, dan cenderung mendekati distribusi normal. Meski demikian, perbedaan

karakteristik distribusi antar kelas tetap dapat diamati, yang penting untuk mempertahankan kemampuan model dalam membedakan antara aktivitas normal dan serangan. Dengan skala fitur yang konsisten, setiap fitur memberikan kontribusi yang sebanding dalam proses pembelajaran, sehingga model mampu mengidentifikasi pola atau anomali dengan lebih adil tanpa dipengaruhi oleh skala yang tidak wajar. Proses penyesuaian bobot menjadi lebih konsisten karena gradien tidak 'meledak' akibat perbedaan skala fitur. Hasil normalisasi ini menunjukkan bahwa pengolahan data awal berlangsung efektif dalam mendukung performa deteksi serangan DdoS

4.3.Seleksi Fitur

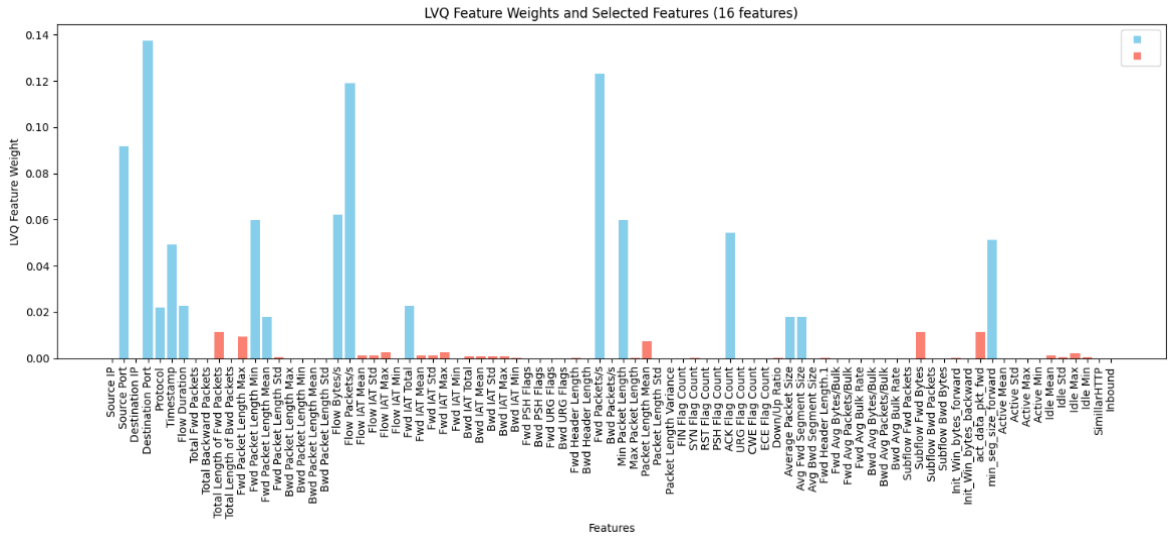
Dalam penelitian ini, digunakan pendekatan *hybrid feature selection* dengan menggabungkan dua metode, yaitu *Learning Vector Quantization* (LVQ) dan *Autoencoder***.**

4.3.1. *Learning Vector Quantization* (LVQ)

Metode LVQ digunakan untuk mengukur kepentingan relatif dari masing-masing fitur terhadap target klasifikasi berdasarkan bobot pelatihan. Jumlah fitur awal yang digunakan adalah sebanyak 85 fitur, yang semuanya diberikan bobot awal yang sama besar pada tahap awal pelatihan, yaitu sebesar 1/85 untuk setiap fitur. Setiap kelas pada dataset diwakili oleh sejumlah prototipe yang diinisialisasi dengan mengambil data secara acak dari data latih kelas tersebut. Dalam proses ini, ditetapkan sebanyak lima prototipe untuk masing-masing kelas agar distribusi variasi data di setiap kelas dapat terwakili dengan baik. Setelah inisialisasi, proses pelatihan dilakukan secara iteratif hingga mencapai jumlah epoch maksimum, yaitu 100. Pada setiap epoch, data latih diacak urutannya untuk menghindari bias, kemudian dihitung jarak antara data latih dengan prototipe menggunakan rumus jarak *Euclidean* berbobot. Prototipe yang memiliki jarak terdekat dengan data latih akan dipilih sebagai *Best Matching Unit* (BMU) dan diperbarui posisinya. Apabila kelas data latih sama dengan kelas prototipe, maka prototipe akan bergerak mendekati data tersebut. Sebaliknya, jika kelasnya berbeda, prototipe akan diarahkan menjauh. Besar penyesuaian prototipe dikendalikan oleh nilai *learning rate*, yang ditetapkan sebesar 0.00001 pada awal pelatihan dan kemudian

diturunkan secara bertahap sebesar 10% setiap epoch hingga mencapai batas minimum sebesar 0.000001. Penurunan *learning rate* ini bertujuan untuk menjaga stabilitas proses pembaruan agar mendekati konvergensi.

Setelah proses pembaruan bobot prototipe selesai dilakukan selama seluruh epoch, langkah selanjutnya adalah melakukan perhitungan kontribusi masing-masing fitur. Nilai bobot seluruh prototipe dikumpulkan dan dihitung rata-rata nilai absolutnya untuk setiap fitur. Selanjutnya, fitur diurutkan berdasarkan nilai kontribusi tersebut dari yang tertinggi hingga terendah. Fitur-fitur yang memiliki nilai kontribusi tinggi dianggap paling relevan dalam membedakan (diskriminatif) paling kuat antar kelas, sedangkan fitur dengan kontribusi rendah dapat diabaikan untuk meningkatkan efisiensi model. Dari hasil pemeringkatan tersebut, ditetapkan ambang batas (*threshold*) efektif dengan memilih 16 fitur dengan skor tertinggi, yang dianggap memiliki kontribusi paling signifikan terhadap pemisahan kelas.



Gambar 4. 5 LVQ *feature weights*

Gambar 4.5 menyajikan visualisasi dari hasil akhir proses pembobotan yang dilakukan oleh LVQ terhadap keseluruhan fitur. Grafik tersebut menampilkan distribusi skor kepentingan (bobot) yang sangat condong (*skewed*), di mana tampak adanya disparitas yang tajam antara sebagian kecil fitur dengan skor signifikansi yang tinggi dan sebagian besar fitur lainnya dengan skor yang mendekati nol. Pola ini merupakan hasil yang diharapkan dari sebuah proses seleksi fitur yang efektif, secara empiris menunjukkan bahwa tidak semua atribut memiliki relevansi yang

setara. Algoritma LVQ berhasil mengisolasi variabel-variabel yang mengandung sinyal informatif paling kuat dari fitur-fitur yang dapat dianggap sebagai noise atau redundan. Distribusi skor inilah yang menjadi dasar objektif untuk penetapan ambang batas seleksi dan pemilihan sub-kumpulan fitur yang paling optimal. Fitur-fitur yang memperoleh bobot tinggi antara lain Destination Port, Fwd Packets/s, Flow Packets/s, Source Port, serta beberapa parameter lalu lintas lainnya seperti Flow Bytes/s, Fwd Packet Length Min, dan ACK Flag Count. Fitur-fitur tersebut dipandang sebagai indikator yang paling representatif dalam membedakan pola trafik berdasarkan struktur dan volume komunikasi data.

Tabel 4. 2 16 fitur tertinggi yang sesuai denga paramete yang digunakan dalam Learning Vector Quatization

NO	FITUR	NILAI	NO	FITUR	NILAI
1	Destination Port	0.1376	9	min_seg_size_forward	0.0512
2	Fwd Packets/s	0.1230	10	Timestamp	0.0494
3	Flow Packets/s	0.1189	11	Fwd IAT Total	0.0228
4	Source Port	0.0917	12	Flow Duration	0.0228
5	Flow Bytes/s	0.0622	13	Protocol	0.0220
6	Fwd Packet Length Min	0.0598	14	Avg Fwd Segment Size	0.0180
7	Min Packet Length	0.0597	15	Fwd Packet Length Mean	0.0180
8	ACK Flag Count	0.0542	16	Average Packet Size	0.0178

Tabel 4.2 merinci hasil akhir dari proses seleksi, yaitu **16 fitur teratas** yang skor kontribusinya melampaui ambang batas signifikansi yang telah di tetapkan. Skor yang ditampilkan pada kolom nilai mencerminkan kontribusi relatif masing-masing fitur dalam membedakan pola lalu lintas jaringan. Dari hasil seleksi ini bahwa LVQ menunjukkan bahwa algoritma ini secara efektif membangun sebuah profil serangan yang terperinci dari berbagai sudut pandang. Titik fokus pertama adalah pada target dan ciri-ciri serangan. Sebagai fitur dengan skor tertinggi, *Destination Port* (0.1376) menjadi tanda utama karena kemampuannya untuk secara langsung menunjuk layanan spesifik yang menjadi sasaran. Hal ini didukung oleh *Source Port* (0.0917) yang membantu mengenali pola koneksi penyerang, serta

Protocol (0.0220) yang memberikan informasi penting untuk membedakan jenis serangan, misalnya antara UDP flood dan serangan berbasis TCP. Selain mengidentifikasi target, skala dan kekuatan serangan menjadi fokus utama berikutnya. Fitur Fwd Packets/s (0.1230), Flow Packets/s (0.1189), dan Flow Bytes/s (0.0622) secara bersama-sama mengukur aspek "banjir" (*flooding*) yang menjadi inti dari serangan DDoS. Ketiga fitur ini secara langsung menghitung laju dan volume data yang dikirim, di mana nilai yang sangat tinggi menjadi pembeda yang jelas antara lalu lintas normal dan kondisi saat jaringan diserang. Pemilihan fitur-fitur ini memastikan model sangat peka terhadap kekuatan serangan *brute-force*.

Lebih dari sekadar analisa pada volumesaja, analisis mendalam menunjukkan kemampuan model untuk mendeteksi taktik serangan melalui karakteristik detail dari paket. Kelompok fitur ini, yang mencakup Fwd Packet Length Min (0.0598), Min Packet Length (0.0597), dan min_seg_size_forward (0.0512), beserta metrik ukuran rata-rata seperti Avg Fwd Segment Size (0.0180) dan Average Packet Size (0.0178), secara lengkap memetakan pola ukuran paket. Ini sangat penting karena menunjukkan kemampuan model untuk mengenali serangan yang bertujuan menguras sumber daya server (*resource exhaustion*) dengan menggunakan paket berukuran sangat kecil dalam jumlah besar, bukan hanya serangan yang membanjiri *bandwidth*. fitur-fitur yang memberikan konteks tambahan dari sisi protokol dan waktu menyempurnakan gambaran serangan. Fitur ACK Flag Count (0.0542) berfungsi untuk mendeteksi kejanggalan pada level protokol yang dapat menandakan serangan spesifik seperti *ACK Flood*. Sementara itu, fitur berbasis waktu seperti Timestamp (0.0494), Fwd IAT Total (0.0228), dan Flow Duration (0.0228) menempatkan anomali dalam kerangka waktu, membantu membedakan antara lonjakan trafik singkat yang wajar dengan serangan yang berkelanjutan.

Fitur-fitur yang dipilih oleh LVQ merupakan hasil dari sifat alaminya sebagai algoritma terawasi (*supervised*), yang secara *inheren* memprioritaskan atribut dengan daya pembeda paling kuat antara lalu lintas normal dan serangan. LVQ secara aktif dilatih untuk menemukan "petunjuk" yang paling jelas, sehingga secara logis memilih fitur-fitur yang secara langsung memetakan karakteristik serangan **volumetrik**. Kombinasi fitur ini sangat bermakna karena membangun sebuah profil

serangan yang koheren dan multi-aspek: mulai dari identifikasi target serangan melalui Destination Port (0.1376), pengukuran skala kekuatan melalui Fwd Packets/s (0.1230) dan Flow Bytes/s (0.0622), hingga deteksi taktik spesifik seperti serangan penghabisan sumber daya melalui Fwd Packet Length Min (0.0598). Set-fitur ini tidak hanya efisien, tetapi juga kaya secara informatif, menciptakan fondasi model yang andal untuk mendeteksi serangan DDoS konvensional dari berbagai sudut pandang.

4.3.2. **Autoencoder**.

Autoencoder berfungsi untuk mengurangi dimensi data secara non-linier dengan mempertahankan informasi utama dari struktur data input. Pendekatan ini dilakukan melalui rancangan arsitektur Autoencoder yang terdiri atas tiga bagian utama, yaitu encoder, bottleneck, dan decoder.

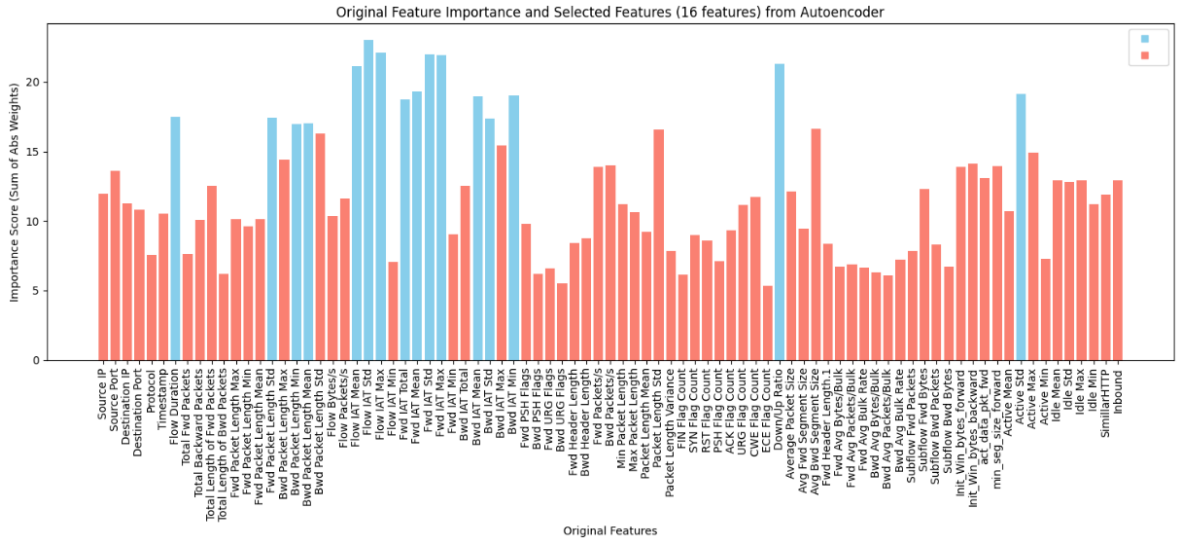
Pada bagian encoder, data input dengan 85 fitur dimasukkan ke dalam jaringan, kemudian diproses melalui beberapa lapisan tersembunyi (*hidden layer*) dengan fungsi aktivasi ReLU untuk menghasilkan transformasi non-linear yang dapat menangkap pola kompleks pada data. Lapisan tersembunyi pertama memiliki 64 neuron, diikuti dengan lapisan kedua yang memiliki 32 neuron. Setelah itu, data diteruskan ke bagian *bottleneck* yang berfungsi sebagai lapisan representasi inti dari data yang telah dikompresi. Pada bagian ini, jumlah neuron dikurangi menjadi 16 neuron, sesuai dengan target jumlah fitur yang akan diambil sebagai fitur terpilih. Fungsi aktivasi yang digunakan pada bottleneck juga menggunakan ReLU agar tetap mampu mempertahankan variasi pola data meskipun telah direduksi.

Setelah melalui *bottleneck*, data diteruskan ke bagian decoder yang bertugas untuk merekonstruksi kembali data ke dimensi semula. Struktur decoder disusun secara simetris dengan encoder, diawali dengan lapisan tersembunyi berisi 32 neuron, diikuti oleh lapisan dengan 64 neuron, dan ditutup dengan lapisan output berisi 85 neuron untuk menghasilkan output yang sama seperti jumlah fitur input. Pada lapisan output digunakan fungsi aktivasi *Sigmoid*, karena data input dinormalisasi pada rentang 0 hingga 1 agar hasil rekonstruksi tetap sesuai skala.

Optimasi pembaruan bobot dilakukan menggunakan algoritma Adam dengan *learning rate* sebesar 0.000001 agar pembelajaran berjalan stabil. Kinerja model

diukur dengan fungsi loss *Mean Squared Error* (MSE) yang menghitung rata-rata selisih kuadrat antara input dan output rekonstruksi, sehingga jaringan dapat meminimalkan kesalahan rekonstruksi. Proses pelatihan dilakukan selama 100 iterasi dengan tujuan agar jaringan benar-benar mampu menemukan pola terbaik untuk memetakan data ke representasi yang lebih ringkas.

Setelah pelatihan selesai, bagian decoder tidak lagi digunakan. Hanya bagian *bottleneck* yang diambil sebagai hasil utama, di mana seluruh data di-transformasi melalui jaringan Autoencoder hingga diperoleh dataset baru dengan dimensi 16 fitur.



Gambar 4. 6 fitur baru yang terbentuk untuk membuat mereduksi dataset sebelumnya

Gambar 4.6 menyajikan visualisasi dari hasil evaluasi kepentingan fitur yang dilakukan oleh model Autoencoder terhadap keseluruhan set atribut asli. Skor kepentingan yang ditampilkan pada sumbu vertikal dihitung berdasarkan akumulasi nilai absolut bobot (*absolute weight sum*) pada lapisan-lapisan tersembunyi, yang secara efektif mengukur kontribusi setiap fitur dalam proses rekonstruksi data. Grafik ini secara gamblang menunjukkan sebuah proses seleksi yang sangat tegas dan diskriminatif. Terlihat adanya disparitas yang sangat jelas antara segelintir fitur yang memiliki skor kepentingan tinggi (ditandai dengan batang biru) dengan sebagian besar fitur lainnya (batang oranye) yang memiliki skor jauh lebih rendah. Pola ini mengindikasikan bahwa dalam upayanya untuk mempelajari representasi data yang paling efisien, Autoencoder secara alami menemukan bahwa hanya

sebagian kecil fitur yang krusial untuk menangkap struktur dan informasi esensial dari lalu lintas jaringan. Sebaliknya, mayoritas fitur menunjukkan kontribusi yang marjinal terhadap proses rekonstruksi, sehingga dapat dianggap sebagai kandidat kuat untuk dieliminasi guna menyederhanakan model tanpa kehilangan informasi yang signifikan. Dengan demikian, visualisasi ini memberikan dasar empiris yang kuat untuk justifikasi pemilihan 16 fitur teratas.

Fitur-fitur dominan yang memperoleh skor tertinggi didominasi oleh atribut-atribut yang berkaitan dengan *Inter Arrival Time* (IAT), seperti Fwd IAT Std, Fwd IAT Max, Flow IAT Std, Flow IAT Max, dan Fwd IAT Total. Kondisi ini mengindikasikan bahwa pola waktu antar paket, baik pada arah forward maupun pada tingkat aliran secara keseluruhan, memberikan sinyal yang kuat dalam membedakan lalu lintas normal (BENIGN) dengan serangan DDoS (Attack). Selain itu, fitur-fitur lain seperti Down/Up Ratio, Fwd Packet Length Std, dan Flow Duration juga memiliki skor kepentingan yang tinggi, yang menegaskan bahwa rasio arah lalu lintas, variasi panjang paket, serta durasi sesi komunikasi relevan dalam deteksi anomali jaringan. Sebaliknya, fitur-fitur seperti flag header, atribut IP statis, serta beberapa statistik lalu lintas dengan dinamika rendah terlihat memiliki skor kepentingan yang relatif kecil dan tidak terpilih sebagai fitur penting. Hal ini mencerminkan efisiensi seleksi fitur oleh Autoencoder, yang memfokuskan pada atribut dengan nilai informatif tertinggi dalam proses pembelajaran representasi untuk tugas klasifikasi.

Tabel 4. 3 16 fitur tertinggi yang sesuai denga paramete yang digunakan dalam Autoencoder

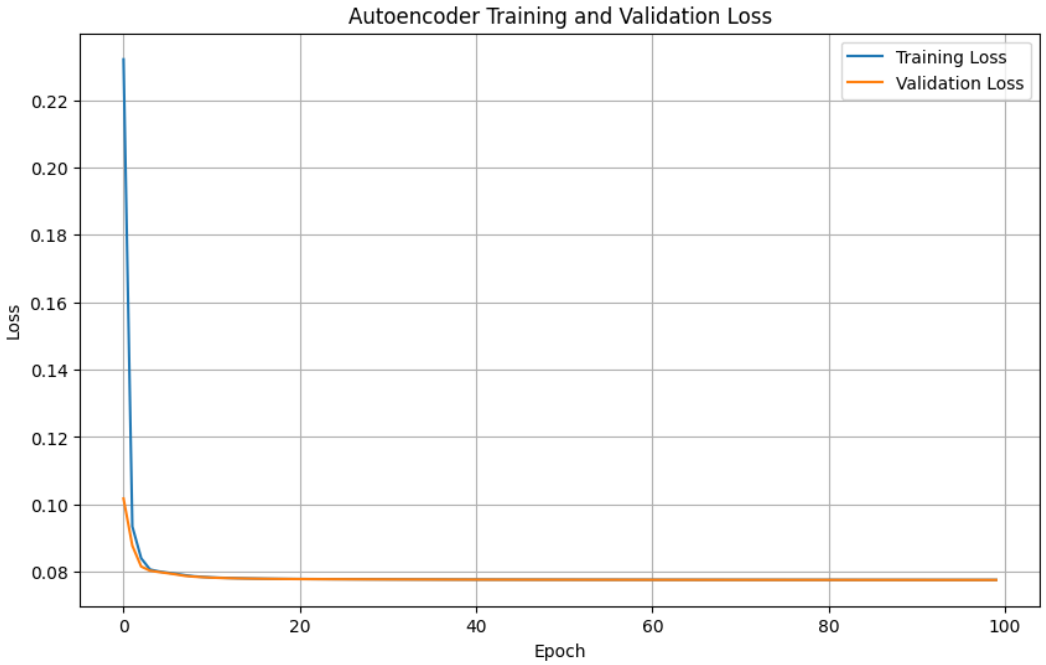
NO	FITUR	NILAI	NO	FITUR	NILAI
1	Flow IAT Std	23.0378	9	Bwd IAT Min	19.0403
2	Flow IAT Max	22.1148	10	Bwd IAT Mean	18.9859
3	Fwd IAT Std	21.9899	11	Fwd IAT Total	18.7412
4	Fwd IAT Max	21.9240	12	Flow Duration	17.4868
5	Down/Up Ratio	21.3273	13	Fwd Packet Length Std	17.4410
6	Flow IAT Mean	21.1345	14	Bwd IAT Std	17.3681
7	Fwd IAT Mean	19.2952	15	Bwd Packet Length Mean	17.0128
8	Active Std	19.1260	16	Bwd Packet Length Min	16.9786

Tabel 4.2 Menampilkan daftar 16 fitur dengan skor kontribusi tertinggi yang dihasilkan dari proses seleksi fitur menggunakan algoritma Autoencoder. Model ini secara sistematis memprioritaskan atribut yang mengukur anomali pada dimensi waktu dan perilaku, bukan volume. Kumpulan fitur ini bekerja secara sinergis untuk membangun sebuah profil serangan berdasarkan jejak gangguan yang ditinggalkannya pada alur normal komunikasi jaringan. Metrik yang digunakan berdasarkan penjumlahan nilai absolut bobot (weights) pada lapisan tersembunyi yang relevan. Deteksi yang paling dominan adalah pada anomali aliran waktu lalu lintas dibuktikan dengan mayoritas fitur berskor tertinggi yang seluruhnya merupakan metrik *Inter-Arrival Time* (IAT). Fitur-fitur seperti Flow IAT Std (23.0378), Flow IAT Max (22.1148), Fwd IAT Std (21.9899), dan Fwd IAT Max (21.9240) secara kolektif mengukur ketidakteraturan dan interval ekstrem pada jeda waktu antar paket. Autoencoder sangat peka terhadap "jitter" atau kekacauan temporal yang menjadi ciri khas lalu lintas serangan, yang secara drastis berbeda dari ritme lalu lintas normal yang cenderung lebih stabil. Metrik IAT rata-rata (Flow IAT Mean dan Fwd IAT Mean) serta total (Fwd IAT Total) juga melengkapi gambaran ini dengan memberikan informasi mengenai pergeseran tendensi sentral dari pola waktu tersebut.

Selain itu penekanan anomali perilaku sesi dan asimetri lalu lintas seperti Fitur Down/Up Ratio (21.3273) menjadi sangat penting karena secara langsung mengukur ketidakseimbangan antara paket yang dikirim dan diterima, sebuah indikator klasik dari serangan di mana penyerang mengirimkan banyak permintaan tanpa menerima balasan yang sepadan. Perilaku sesi yang tidak wajar juga ditangkap oleh Flow Duration (17.4868) dan Active Std (19.1260), yang masing-masing mendeteksi durasi koneksi yang abnormal dan fluktuasi aktivitas yang tidak stabil, yang menandakan serangan persisten atau koneksi yang sengaja dipertahankan untuk menguras sumber daya. Selain pola waktu dan ukuran paket auto encoder juga menganalisis yang mendalam mengenai karakteristik lalu lintas balasan (*backward traffic*) seperti Bwd IAT Min (19.0403), Bwd IAT Mean (18.9859), Bwd IAT Std (17.3681), Bwd Packet Length Mean (17.0128), dan Bwd Packet Length Min (16.9786) sangat signifikan. Menunjukkan bahwa Autoencoder tidak hanya menganalisis lalu lintas serangan yang keluar, tetapi juga secara cerdas

menganalisis **dampak serangan pada respons target**. Pola waktu dan ukuran paket yang aneh dari arah *backward* adalah indikator kuat bahwa sistem target sedang kewalahan dan tidak mampu merespons secara normal. Autoencoder juga mempertimbangkan variabilitas ukuran paket melalui fitur Fwd Packet Length Std (17.4410). Berbeda dengan fokus pada ukuran minimum atau rata-rata, fitur ini menangkap ketidakkonsistenan atau variasi ukuran paket dalam sebuah aliran, yang dapat menjadi taktik penyerang untuk menghindari sistem deteksi berbasis aturan sederhana.

Pada seleksi fitur menggunakan autoencoder karena dilakukan untuk mereduksi data dengan cara mempelajari representasi data yang lebih ringkas (*encoded feature*). Bagian tengah (bottleneck) pada autoencoder memaksa data masuk ke dimensi lebih kecil memungkinkan adanya loss pada iterasi pada pelatihanya



Gambar 4. 7 Tingkat loss terhadap iterasi

Gambar 4.7 menampilkan grafik kurva loss pelatihan dan validasi dari model Autoencoder selama proses pelatihan sebanyak 100 epoch. Berdasarkan grafik, terlihat bahwa nilai training loss (garis biru) dan validation loss (garis oranye) mengalami penurunan tajam pada awal proses pelatihan, khususnya pada 10 epoch pertama, kemudian perlahan mencapai kondisi konvergen di sekitar nilai loss

sebesar 0.078. Pola ini mengindikasikan bahwa model berhasil melakukan proses pembelajaran representasi internal secara efektif dan efisien tanpa mengalami overfitting. Hal ini ditunjukkan oleh jarak yang sangat kecil antara kurva loss pelatihan dan validasi setelah epoch ke-10 hingga akhir pelatihan. Stabilitas kurva ini mengindikasikan bahwa model tidak hanya mampu merekonstruksi data latih dengan baik, tetapi juga mempertahankan performa generalisasi pada data validasi. Pola konvergen antara *training loss* dan *validation loss* ini menjadi indikator bahwa model Autoencoder bekerja dengan baik tanpa indikasi *overfitting* maupun *underfitting*. Stabilnya nilai loss pada data pelatihan dan data validasi juga menegaskan bahwa proses rekonstruksi berjalan optimal, di mana informasi penting dari data input berhasil dipelajari dan direpresentasikan secara efisien oleh Autoencoder. Dengan demikian, kurva ini mendukung keandalan hasil seleksi fitur, karena fitur-fitur yang dipilih berasal dari representasi laten yang benar-benar memuat pola dominan pada data.

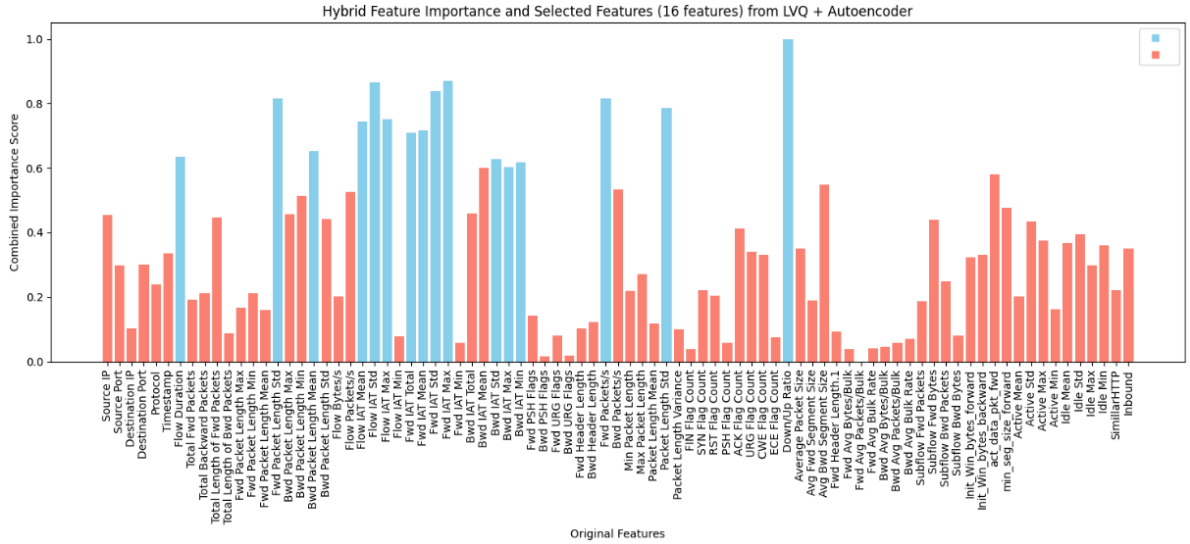
4.3.3. Hybrid (LVQ-Autoencoder)

Pendekatan seleksi fitur hibrid ini menggabungkan keunggulan dua metode, yaitu *Learning Vector Quantization* (LVQ) dan *Autoencoder***,** dengan tujuan memanfaatkan kelebihan masing-masing algoritma untuk memperoleh subset fitur yang lebih relevan dan informatif. LVQ berfungsi untuk menghitung bobot kontribusi setiap fitur berdasarkan perannya dalam proses klasifikasi menggunakan pembelajaran jarak.

Sementara itu, Autoencoder digunakan untuk menganalisis seberapa besar pengaruh masing-masing fitur terhadap akurasi rekonstruksi data, umumnya diukur dari perubahan bobot di lapisan bottleneck atau nilai *feature importance* yang dihasilkan. Agar hasil kedua metode dapat digabungkan secara adil, skor kepentingan fitur dari LVQ dan Autoencoder dinormalisasi terlebih dahulu. Bobot LVQ biasanya sudah dinormalisasi sehingga totalnya sama dengan satu, sedangkan skor dari Autoencoder dinormalisasi menggunakan teknik Min-Max Scaling ke rentang 0–1. Setelah normalisasi, skor kepentingan dari kedua metode digabungkan menggunakan rata-rata sederhana, sehingga menghasilkan skor gabungan yang mewakili persepsi kepentingan fitur secara keseluruhan. Fitur kemudian diurutkan berdasarkan skor gabungan ini secara menurun, dari yang paling penting ke yang

paling rendah.

Dari hasil pengurutan, sejumlah fitur teratas diambil sesuai kapasitas bottleneck Autoencoder, yaitu **16 fitur**. Langkah ini memastikan subset fitur yang dipilih tetap konsisten dengan kapasitas representasi Autoencoder, sekaligus mempertahankan relevansi dari perspektif LVQ.



Gambar 4. 8 *Hybrid feature importace and selected feature from LVQ + Autoencoder*

Gambar 4.8 menunjukkan distribusi skor kepentingan gabungan dari seluruh fitur asli setelah melalui proses seleksi fitur dengan pendekatan hybrid, yaitu menggabungkan metode Learning Vector Quantization (LVQ) dan Autoencoder. Proses ini melibatkan normalisasi skor individual dari kedua metode, yang kemudian dirata-rata untuk menghasilkan "Skor Kepentingan Gabungan" (*Combined Importance Score*) yang ditampilkan pada sumbu vertikal. Grafik ini secara efektif mengilustrasikan bagaimana proses penggabungan tersebut menghasilkan sebuah konsensus. Secara visual, tampak jelas bagaimana 16 fitur (ditandai dengan batang biru) muncul dengan skor gabungan yang superior dibandingkan dengan mayoritas fitur lainnya. Ketinggian batang-batang ini merepresentasikan fitur-fitur yang dianggap penting dari kedua model, baik dari kemampuannya untuk membedakan kelas secara langsung (kontribusi LVQ) maupun dari perannya dalam membentuk struktur data yang esensial (kontribusi Autoencoder). Dengan demikian, fitur-fitur yang terpilih benar-benar memiliki

kontribusi nyata terhadap pola lalu lintas dan diharapkan dapat meningkatkan akurasi deteksi serangan DDoS. Berikut hasil seleksi fitur dengan mengambil 16 fitur terbaik berdasarkan sekor kepentinganya:

Tabel 4. 4 16 fitur tertinggi gabungan seleksi fitur

NO	FITUR	NILAI	NO	FITUR	NILAI
1	Down/Up Ratio	1.0000	9	Flow IAT Mean	0.7438
2	Fwd IAT Max	0.8710	10	Fwd IAT Mean	0.7166
3	Flow IAT Std	0.8648	11	Fwd IAT Total	0.7095
4	Fwd IAT Std	0.8369	12	Bwd Packet Length Mean	0.6514
5	Fwd Packets/s	0.8163	13	Flow Duration	0.6339
6	Fwd Packet Length Std	0.8161	14	Bwd IAT Std	0.6271
7	Packet Length Std	0.7866	15	Bwd IAT Min	0.6169
8	Flow IAT Max	0.7510	16	Bwd IAT Max	0.6031

Tabel 4.3 menyajikan daftar 16 fitur utama yang terpilih melalui pendekatan seleksi fitur hybrid, yakni hasil penggabungan dari metode Learning Vector Quantization (LVQ) dan Autoencoder. Skor yang ditampilkan merupakan skor kontribusi gabungan yang telah dinormalisasi, sehingga memungkinkan perbandingan antar fitur secara objektif. Nilai skor berkisar antara 0.60 hingga 1.00, yang merepresentasikan derajat pentingnya masing-masing fitur dalam membedakan antara lalu lintas jaringan normal (BENIGN) dan aktivitas serangan DDoS (Attack). Hal yang paling signifikan adalah elevasi Down/Up Ratio (1.0000) ke peringkat teratas yang menujukan pola anomali perilaku dan asimetri lalu lintas. Fitur ini mengukur asimetri perilaku lalu lintas yang dikirim dan diterima, dalam komunikasi normal (misalnya, mengakses situs web), pengguna mengirimkan permintaan kecil (paket *upload* rendah) dan menerima respons yang jauh lebih besar berupa konten halaman (paket download tinggi), sehingga rasio *download* terhadap *upload* biasanya lebih besar dari 1. Sebaliknya, dalam banyak skenario serangan DDoS (seperti SYN Flood atau HTTP Flood), penyerang membanjiri target dengan jutaan paket permintaan (*upload* masif) yang seringkali tidak memerlukan atau tidak mendapatkan respons yang sepadan. Akibatnya, lalu lintas upload jauh melampaui *download*, menghasilkan nilai Down/Up Ratio yang

mendekati nol. Nilai ekstrem inilah yang menjadi penanda anomali yang sangat andal, dan terpilihnya fitur ini di peringkat pertama menunjukkan bahwa model Hybrid secara cerdas mengidentifikasi asimetri ini sebagai ciri serangan yang paling universal.

Anomali aliran waktu yang menujukan di mana jejak serangan terdeteksi dari caranya merusak ritme normal komunikasi. Lalu lintas normal, meskipun bervariasi cenderung memiliki pola statistik jeda waktu antar paket (*Inter-Arrival Time*) yang dapat diprediksi. Serangan DDoS dari botnet yang terdiri dari ribuan sumber berbeda akan menghancurkan keteraturan ini, menciptakan aliran data yang kacau. Fitur Flow IAT Std (0.8648) dan Fwd IAT Std (0.8369) menjadi sangat penting karena secara langsung mengukur *variabilitas* atau *jitter* (variasi jeda waktu antar paket) dari lalu lintas serangan tersebut; standar deviasi yang tinggi menunjukkan jeda waktu yang sangat tidak konsisten, sebuah ciri utama dari banjir paket yang tidak terkoordinasi. Sementara itu, Fwd IAT Max (0.8710) dan Flow IAT Max (0.7510) mendeteksi adanya interval waktu ekstrem yang tidak wajar, yang bisa terjadi akibat jeda antar gelombang serangan. Pergeseran pola waktu secara keseluruhan juga ditangkap oleh metrik tendensi sentral seperti Flow IAT Mean (0.7438) dan Fwd IAT Mean (0.7166), serta total durasi jeda waktu dalam Fwd IAT Total (0.7095) dan Flow Duration (0.6339). Ada Tiga jenis indikator komplementer, yang mempertahankan indikator volumetrik kunci melalui Fwd Packets/s (0.8163), yang berfungsi sebagai "pemeriksaan realitas" untuk memastikan serangan *brute-force* yang paling sederhana pun tetap terdeteksi. variabilitas paket melalui Fwd Packet Length Std (0.8161) dan Packet Length Std (0.7866), ini penting untuk mendeteksi taktik serangan yang lebih canggih di mana ukuran paket sengaja diacak untuk menghindari filter sederhana. Model ini menganalisis "sinyal bahaya" dari lalu lintas balasan target. Fitur seperti **Bwd Packet Length Mean (0.6514)** serta serangkaian metrik IAT dari arah *backward*—**Bwd IAT Std (0.6271)**, **Bwd IAT Min (0.6169)**, dan **Bwd IAT Max (0.6031)**—secara efektif menganalisis gejala dari sistem korban. Respons yang lambat, tidak teratur, atau terdiri dari paket-paket kecil yang abnormal adalah bukti kuat bahwa target telah kewalahan, sehingga mengkonfirmasi keberhasilan serangan.

4.4.Recurent Neural Nertwork (RNN) Model Implementation

4.4.1. Model Asitektur

Setelah proses feature selection selesai, tahap selanjutnya adalah pembangunan model deteksi serangan DDoS menggunakan pendekatan Recurrent Neural Network (RNN) berbasis Long Short-Term Memory (LSTM). RNN merupakan salah satu arsitektur jaringan saraf tiruan yang dirancang untuk menangani data berurutan (sequential data) dengan mempertahankan informasi dari input sebelumnya melalui mekanisme hidden state. Namun, arsitektur RNN konvensional memiliki keterbatasan dalam mengatasi masalah long-term dependencies, terutama ketika data sekuensial memiliki dependensi jangka panjang.

Untuk mengatasi kelemahan tersebut, digunakan varian RNN yang lebih canggih, yaitu LSTM. LSTM memiliki struktur sel yang kompleks dengan tiga gerbang utama: input gate, forget gate, dan output gate, yang memungkinkan model untuk mempertahankan atau melupakan informasi secara selektif dalam jangka waktu yang lebih panjang. Dengan demikian, LSTM sangat efektif dalam mengenali pola dalam data sekuensial yang kompleks seperti lalu lintas jaringan, termasuk deteksi pola serangan DDoS.

Tabel 4. 5 Model Summary RNN berbasis LSTM

Layer (type)	Output Shape	Param
input_layer_1 (InputLayer)	(None, 1, 85)	0
lstm (LSTM)	(None, 1, 64)	38,400
dropout (Dropout)	(None, 1, 64)	0
lstm_1 (LSTM)	(None, 32)	12,416
dropout_1 (Dropout)	(None, 32)	0
dense_6 (Dense)	(None, 16)	528
dense_7 (Dense)	(None, 1)	17

Model ini diawali dengan lapisan input yang menerima data berupa urutan fitur dengan format time-series, di mana setiap timestep merepresentasikan kondisi lalu lintas jaringan pada periode waktu tertentu. Fitur yang digunakan

dapat meliputi jumlah paket yang masuk dan keluar, ukuran paket, flag protokol, hingga interval waktu antar paket. Representasi data dalam bentuk urutan ini bertujuan agar model mampu mengenali pola temporal yang menjadi karakteristik dari aktivitas serangan DDoS, yang umumnya ditandai dengan lonjakan intensitas lalu lintas secara tiba-tiba maupun pola fluktuasi anomali dalam periode tertentu.

Selanjutnya, data input diproses melalui dua lapisan LSTM yang disusun secara bertingkat. Lapisan LSTM pertama terdiri atas 64 unit sel dengan pengaturan *return sequences* diaktifkan, sehingga keluaran dari lapisan ini tetap dalam bentuk urutan. Hal ini memungkinkan lapisan LSTM berikutnya tetap dapat mempelajari hubungan antar timestep secara lebih mendalam. Setelah lapisan LSTM pertama, diterapkan lapisan *dropout* dengan tingkat 0,2 yang berfungsi untuk meminimalkan risiko *overfitting* dengan cara menonaktifkan sejumlah neuron secara acak pada saat proses pelatihan. Berikutnya, lapisan LSTM kedua terdiri atas 32 unit sel dengan pengaturan *return sequences* dinonaktifkan, sehingga menghasilkan keluaran berupa representasi vektor tunggal. Lapisan kedua ini bertugas merangkum informasi dari seluruh urutan menjadi representasi fitur yang lebih padat dan abstrak.

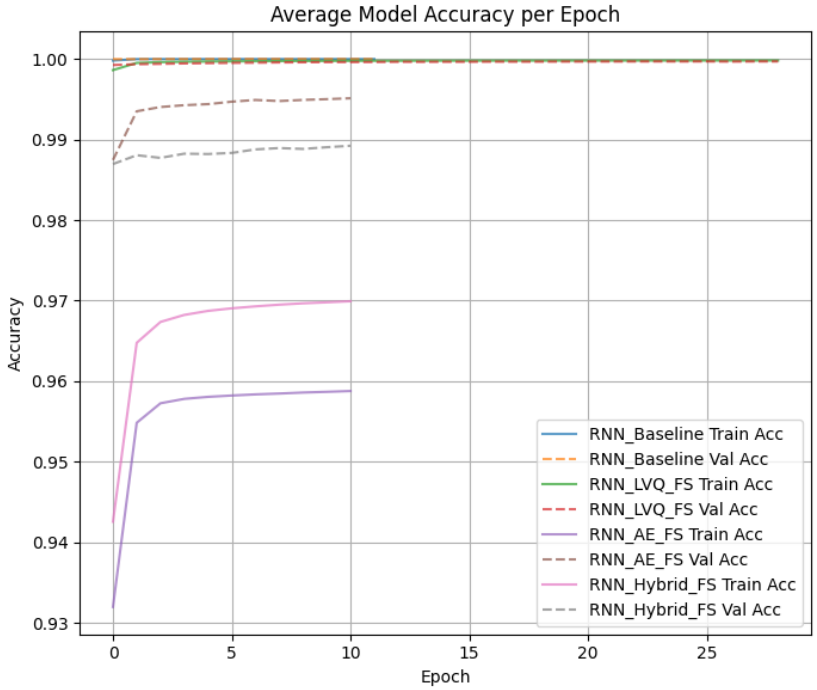
Hasil representasi vektor dari LSTM kemudian diteruskan ke lapisan *Dense* dengan 16 unit neuron dan fungsi aktivasi ReLU. Lapisan *Dense* ini berperan untuk memetakan pola yang telah diekstraksi sebelumnya ke dalam bentuk representasi non-linear yang lebih relevan untuk tugas klasifikasi. Pada tahap akhir, model dilengkapi dengan lapisan output yang terdiri dari satu unit neuron dengan fungsi aktivasi sigmoid untuk menghasilkan nilai probabilitas antara 0 dan 1, yang menunjukkan kemungkinan data termasuk dalam kategori normal atau serangan DDoS. Proses optimasi model dilakukan dengan menggunakan *Adam Optimizer* karena kemampuannya dalam menyesuaikan *learning rate* secara adaptif dan mempercepat proses konvergensi. Sementara itu, fungsi *loss* yang digunakan adalah *Binary Crossentropy* karena untuk memprediksi antara dua kemungkinan kelas.

4.5.Traning Dan validasi

Pada tahap ini, dilakukan pelatihan dan pengujian pada keempat model yang telah diusulkan sebelumnya dengan menggunakan teknik k-fold cross validation. Teknik ini melibatkan pemecahan data menjadi sepuluh subset atau lipatan dengan k =10. Dalam setiap iterasi, sembilan subset dimanfaatkan sebagai data pelatihan. Sebelum pelatihan dimulai, data pelatihan tersebut diseimbangkan terlebih dahulu menggunakan Synthetic Minority Over-sampling Technique (SMOTE), yang bertujuan untuk meningkatkan jumlah data pada kelas minoritas dengan mensintesis data baru berdasarkan karakteristik data yang ada. Satu subset sisanya digunakan sebagai data validasi. Proses ini memastikan bahwa pada setiap iterasi, proporsi antara kategori benign dan attack tetap seimbang, sesuai dengan distribusi asli dari dataset.

Selama pelatihan, setiap model pada masing-masing fold dilatih selama maksimal 50 epoch. Namun, untuk mencegah *overfitting*, digunakan mekanisme *Early Stopping*, yaitu teknik penghentian dini yang secara otomatis menghentikan proses pelatihan jika performa model pada data validasi tidak menunjukkan peningkatan setelah sejumlah epoch tertentu (*patience*). Dalam hal ini, bobot terbaik dari model disimpan dan digunakan untuk evaluasi selanjutnya. Pendekatan ini tidak hanya mempercepat proses pelatihan tetapi juga membantu memastikan bahwa model yang dihasilkan memiliki generalisasi yang baik.

Data pelatihan diperoleh dari dataset benign dan attack yang telah melalui proses preprocessing dan seleksi fitur sebelumnya. Proses pelatihan (training) dan pengujian (testing) dilakukan terhadap model Recurrent Neural Network (RNN) dengan empat skenario berbeda, yaitu RNN Baseline, RNN dengan seleksi fitur LVQ (RNN_LVQ_FS), RNN dengan seleksi fitur Autoencoder (RNN_AE_FS), dan RNN dengan seleksi fitur gabungan LVQ+Autoencoder (RNN_Hybrid_FS).



Gambar 4. 9 Average Model Accuracy per Epoch

Berdasarkan Gambar 4.9 menunjukkan kurva perkembangan akurasi rata-rata dari empat model *Recurrent Neural Network* (RNN): RNN_Baseline, RNN_LVQ_FS, RNN_AE_FS, dan RNN_Hybrid_FS, selama proses pelatihan hingga maksimum **50 epoch**. Masing-masing model diuji berdasarkan dua metrik utama, yaitu **akurasi pelatihan (train accuracy)** dan **akurasi validasi (validation accuracy)**, untuk mengevaluasi sejauh mana model mampu melakukan generalisasi terhadap data baru yang tidak terlihat selama proses pelatihan.

Model RNN_Baseline menunjukkan lonjakan akurasi yang sangat cepat, mencapai hampir **100%** pada epoch ke-2 baik pada data pelatihan maupun validasi. Hal ini menandakan model mampu mengenali pola dataset dengan sangat baik. Namun, performa yang **"sempurna"** ini tidak lepas dari pengaruh distribusi kelas yang tidak seimbang, di mana trafik **ATTACK** mendominasi jumlah data. Dengan demikian, tingginya akurasi baseline lebih **merefleksikan bias terhadap kelas mayoritas** dibanding kemampuan mendeteksi kelas minoritas **(BENIGN).**

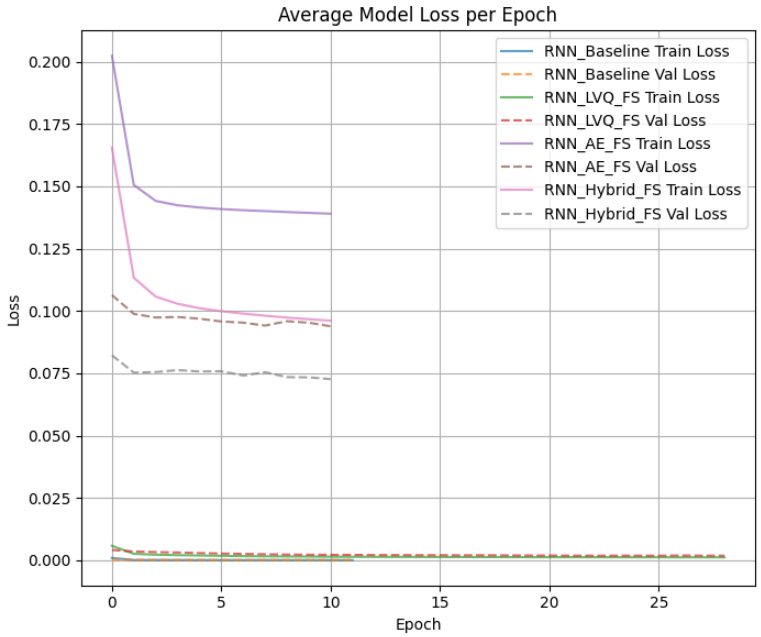
Model RNN_LVQ_FS, yang menerapkan seleksi fitur dengan Learning Vector Quantization (LVQ), memperlihatkan performa yang setara dengan baseline. Akurasi train dan validasi sama-sama **mendekati 100%**, menunjukkan bahwa pengurangan dimensi input dari 85 fitur menjadi 16 fitur tidak menurunkan

performa secara signifikan. Hal ini menegaskan bahwa LVQ efektif dalam memilih fitur paling relevan tanpa kehilangan informasi penting.

Model RNN_AE_FS menunjukkan pola yang lebih unik. Akurasi pelatihan stabil di kisaran **96%,** sementara akurasi validasi lebih tinggi, yaitu sekitar **99%.** Perbedaan ini dapat disebabkan oleh mekanisme regularisasi seperti dropout yang menekan performa pada data latih, sementara pada data validasi dropout tidak diterapkan sehingga model terlihat lebih optimal. Kondisi ini mengindikasikan bahwa Autoencoder berhasil mengekstraksi representasi fitur yang generalis dan mampu meningkatkan kinerja validasi tanpa overfitting.

Model RNN_Hybrid_FS, yang menggabungkan LVQ dan Autoencoder, memperlihatkan tren paling stabil di antara keempat model. Akurasi pelatihan berada di kisaran **97%,** sedangkan akurasi validasi mencapai sekitar **98,8%.** Selisih yang kecil antara keduanya menunjukkan keseimbangan yang baik antara kemampuan belajar dari data latih dan generalisasi pada data validasi. Meskipun akurasi totalnya tidak setinggi baseline, pendekatan hybrid memberikan performa yang lebih realistis dan konsisten dalam mengenali pola, terutama pada kelas minoritas.

Meskipun jumlah epoch maksimum ditetapkan sebanyak **50**, grafik ini juga menunjukkan bahwa pelatihan beberapa model terhenti lebih awal, seperti pada prosesdi atas pada **epoch ke-10** dan **epoch ke-25**. Hal ini bukan disebabkan oleh kegagalan proses pelatihan, melainkan karena diaktifkannya mekanisme **Early Stopping**. Dengan demikian, penghentian lebih awal menandakan bahwa model telah mencapai **titik konvergensi optimal**, sehingga pelatihan lebih lanjut tidak memberikan manfaat tambahan.



Gambar 4. 10 Average Model Loss per Epoch

Berdasarkan Gambar 4.10 memperlihatkan tren penurunan nilai loss rata-rata dari empat skenario. Grafik ini menampilkan dua metrik utama: **training loss** dan **validation loss**, yang diukur pada setiap epoch untuk mengevaluasi kemampuan model dalam meminimalkan kesalahan prediksi sekaligus menjaga generalisasi terhadap data baru.

Pada **RNN_Baseline**, kurva training loss dan validation loss sama-sama turun sangat cepat hingga mendekati nol sejak epoch awal. Kedua kurva hampir berhimpitan **tanpa menunjukkan gap yang besar**. Kondisi ini menunjukkan bahwa model mampu menyesuaikan pola data dengan sangat cepat, namun lebih disebabkan oleh **dominasi kelas mayoritas (ATTACK)**. Dengan demikian, meskipun loss terlihat sempurna, baseline berpotensi bias dan **kurang peka terhadap variasi kelas minoritas (BENIGN).**

Pada RNN_LVQ_FS, kurva training loss dan validation loss juga berada pada level yang sangat rendah, serupa dengan baseline. Seleksi fitur LVQ yang mereduksi dimensi menjadi 16 fitur utama membuat **model lebih efisien tanpa kehilangan performa signifikan**. Pola ini menunjukkan bahwa LVQ membantu **menjaga kestabilan generalisasi**, meskipun karakteristik kurvanya masih cenderung menyerupai baseline.

Pada model **RNN_AE_FS**, pola kurva *loss* menunjukkan karakteristik yang

sangat berbeda dibandingkan model lainnya. Nilai *training loss* (garis ungu) menjadi yang tertinggi di antara semua model, di mana setelah penurunan awal, nilainya stabil di kisaran **0.14**. Sementara itu, kurva *validation loss* (garis cokelat) secara konsisten berada di bawahnya dan stabil pada level yang lebih rendah, yaitu sekitar **0.095**. Kondisi ini menciptakan celah (*gap*) yang paling lebar di antara keempat model, dengan *training loss* yang lebih tinggi daripada *validation loss*. Fenomena ini merupakan indikasi kuat dari pengaruh mekanisme regularisasi (seperti *dropout*) yang aktif selama fase pelatihan untuk mencegah *overfitting*. Meskipun nilai *loss* secara absolut lebih tinggi daripada model *baseline*, *validation loss* yang rendah dan stabil menandakan bahwa **Autoencoder berhasil mengekstraksi representasi fitur yang lebih general dan tangguh terhadap *noise***, sehingga menghasilkan model dengan kemampuan generalisasi yang sehat.

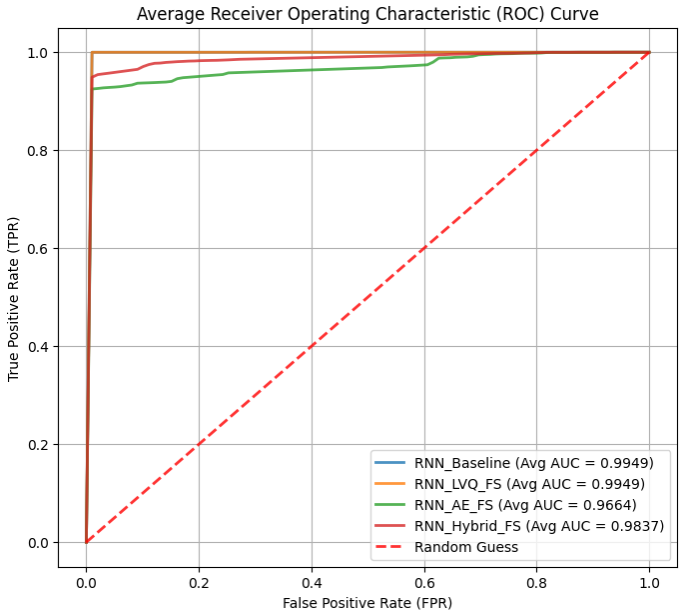
Pada RNN_Hybrid_FS, yang menggabungkan LVQ dan Autoencoder, terlihat kurva training loss turun stabil dari sekitar **0,20** menuju **0,096**, sementara validation loss lebih rendah dan stabil di kisaran **0,075**. Perbedaan kecil antara keduanya menunjukkan stabilitas yang baik, bahkan validasi lebih rendah dari training karena efek regulasi dropout atau karena data validasi relatif lebih mudah dipelajari. Hal ini menunjukkan bahwa pendekatan hybrid **memberikan representasi fitur yang paling seimbang dan adaptif terhadap variasi kedua kelas**.

Secara keseluruhan, grafik ini memperlihatkan beberapa temuan penting:

- **RNN_Baseline** memperlihatkan penurunan *loss* yang sangat cepat hingga mendekati nol. Meskipun terlihat sempurna, pola ini lebih mencerminkan bias terhadap kelas mayoritas sehingga berisiko kurang peka terhadap kelas minoritas.
- **RNN_LVQ_FS** menunjukkan tren penurunan *loss* yang serupa dengan *baseline*. Hal ini menandakan bahwa seleksi fitur oleh LVQ mampu menjaga efisiensi pelatihan dan performa generalisasi tanpa mengubah karakteristik dasar model secara signifikan.
- **RNN_AE_FS** memiliki *loss* yang jauh lebih tinggi dibanding *baseline* dan LVQ, serta menunjukkan **celah (gap) yang paling lebar**, dengan *training*

loss (sekitar 0.14) secara konsisten lebih tinggi daripada *validation loss* (sekitar 0.095). Perbedaan signifikan ini mengindikasikan adanya efek regularisasi yang kuat, menandakan bahwa model berhasil melakukan generalisasi yang sehat dengan mencegah *overfitting*.

- **RNN_Hybrid_FS** memperlihatkan pola yang seimbang, dengan *training loss* yang menurun secara konsisten dan *validation loss* yang stabil pada tingkat lebih rendah. Fenomena ini menunjukkan generalisasi yang paling sehat dan adaptif terhadap variasi data.
- Model yang menerapkan seleksi dan ekstraksi fitur (*LVQ_FS, AE_FS, Hybrid_FS*) umumnya mencapai konvergensi lebih cepat, menunjukkan efisiensi pelatihan dan kemampuan belajar yang lebih terfokus.



Gambar 4. 11 *Average Receiver Operating Characteristic* (ROC)

Gambar 4.11 menampilkan grafik *Receiver Operating Characteristic* (ROC) untuk membandingkan performa empat skenario model Recurrent Neural Network (RNN) dalam mendeteksi serangan DDoS pada dataset pengujian. ROC menggambarkan hubungan antara ***True Positive Rate*** **(TPR)** dan ***False Positive Rate*** **(FPR)** pada berbagai ambang batas klasifikasi. Kurva ROC yang semakin mendekati sudut kiri atas mengindikasikan bahwa model memiliki **sensitivitas tinggi (recall)** dan **tingkat kesalahan prediksi positif palsu yang rendah (FPR kecil)**, yang merupakan karakteristik dari sistem deteksi yang baik. Selain itu,

metrik *Area Under Curve* (AUC) digunakan untuk mengkuantifikasi kualitas pemisahan antar kelas, di mana nilai mendekati 1 menunjukkan performa klasifikasi yang sangat baik.

Hasil pengamatan menunjukkan bahwa Model **RNN_Baseline** menunjukkan kinerja terbaik, dengan AUC sebesar **0,9949**. Kurvanya hampir menempel pada sumbu TPR, menandakan kemampuan sangat tinggi dalam membedakan trafik BENIGN dan ATTACK. Namun, performa yang terlalu tinggi ini juga mengindikasikan adanya **bias terhadap kelas mayoritas (ATTACK)**. Karena model baseline dilatih menggunakan seluruh 85 fitur tanpa seleksi, model ini cenderung "menghafal" pola dominan pada kelas ATTACK. Hal ini memang menghasilkan sensitivitas hampir sempurna, tetapi berpotensi mengabaikan variasi trafik BENIGN yang jumlahnya jauh lebih sedikit dalam dataset.

Model **RNN_LVQ_FS** yang menggunakan seleksi fitur Learning Vector Quantization (LVQ) juga mencatatkan nilai AUC sebesar **0,9949**, sama dengan baseline. Ini menunjukkan bahwa meskipun jumlah fitur direduksi dari 85 menjadi 16, kualitas separasi antar kelas tidak berkurang. LVQ terbukti berhasil mempertahankan fitur-fitur paling relevan untuk membedakan antara trafik serangan dan normal. Selain itu, penggunaan fitur lebih sedikit membuat model lebih efisien dan relatif mengurangi risiko overfitting akibat kompleksitas input yang terlalu tinggi.

Model RNN_AE_FS memperoleh nilai AUC sebesar **0,9664**, lebih rendah dibanding baseline dan LVQ. Perbedaan ini dapat dijelaskan karena Autoencoder bekerja secara unsupervised dan fokus pada rekonstruksi data, bukan diskriminasi kelas. Akibatnya, beberapa fitur penting untuk klasifikasi mungkin tidak sepenuhnya dipertahankan. Meski begitu, **nilai AUC di atas 0,96** tetap menempatkan model ini dalam kategori **performa sangat baik**. Menariknya, model Autoencoder juga cenderung lebih **netral terhadap bias mayoritas**, karena representasi fiturnya lebih general dibanding baseline yang sangat "tajam" terhadap pola ATTACK. **RNN_Hybrid_FS** yang menggabungkan LVQ dan Autoencoder mencatat AUC sebesar **0,9837**. Nilai ini berada di bawah baseline dan LVQ, namun lebih tinggi daripada AE tunggal. Kurva ROC hybrid menunjukkan keseimbangan yang baik, grafik masih sangat curam di area FPR rendah (zona penting dalam

deteksi intrusi, karena false alarm harus ditekan), sambil tetap menjaga stabilitas generalisasi. Pendekatan hybrid mengurangi dominasi bias terhadap kelas mayoritas, karena kombinasi LVQ (fitur diskriminatif) dan AE (fitur representatif) membuat model lebih sensitif terhadap variasi trafik minoritas (BENIGN).

Perbedaan nilai AUC antar model relatif kecil (1–3%), yang berarti semua model tetap unggul dalam membedakan trafik serangan dan normal. Akan tetapi, model baseline dan LVQ, meskipun mencatat AUC tertinggi, lebih rentan bias terhadap kelas mayoritas, sementara model hybrid dan AE memberikan keseimbangan lebih baik antara sensitivitas dan generalisasi.

4.6.Evaluation

Setelah melalui proses seleksi fitur dan pelatihan model Recurrent Neural Network (RNN) dengan berbagai konfigurasi data, tahap berikutnya adalah melakukan evaluasi terhadap performa model deteksi serangan DDoS. Tujuan dari evaluasi ini adalah untuk mengetahui seberapa baik model mampu mengklasifikasikan lalu lintas jaringan ke dalam masing-masing kelas, baik kelas serangan maupun kelas normal (BENIGN), serta mengukur dampak dari teknik seleksi fitur terhadap kualitas prediksi. Evaluasi dilakukan menggunakan empat metrik utama, Akurasi, Presisi, Recall, F1-score.

Evaluasi dilakukan pada empat skenario model yaitu RNN tanpa seleksi fitur (baseline), RNN setelah seleksi fitur dengan LVQ, RNN setelah seleksi fitur dengan Autoencoder, RNN setelah seleksi fitur hybrid LVQ-Autoencoder. Setiap skenario diuji pada dataset CICDDoS2019 yang mengandung 2 kelas, dan hasilnya disajikan pada Tabel 4.3 berikut:

Tabel 4.3 Evaluasi kinerja model pada berbagai sekenario

Sekenario	**Accuracy (%)**	**Recall (%)**	**Precision (%)**	**F1-Score (%)**	**ROC AUC (%)**	**Train Time (s)**
RNN + No FS (baseline)	**99,9997**	**99,9866**	**99,8933**	**99,9399**	**99,9961**	**9.955**
RNN + LVQ	99,9726	99,8795	90,3477	94,8721	99,9919	22.516

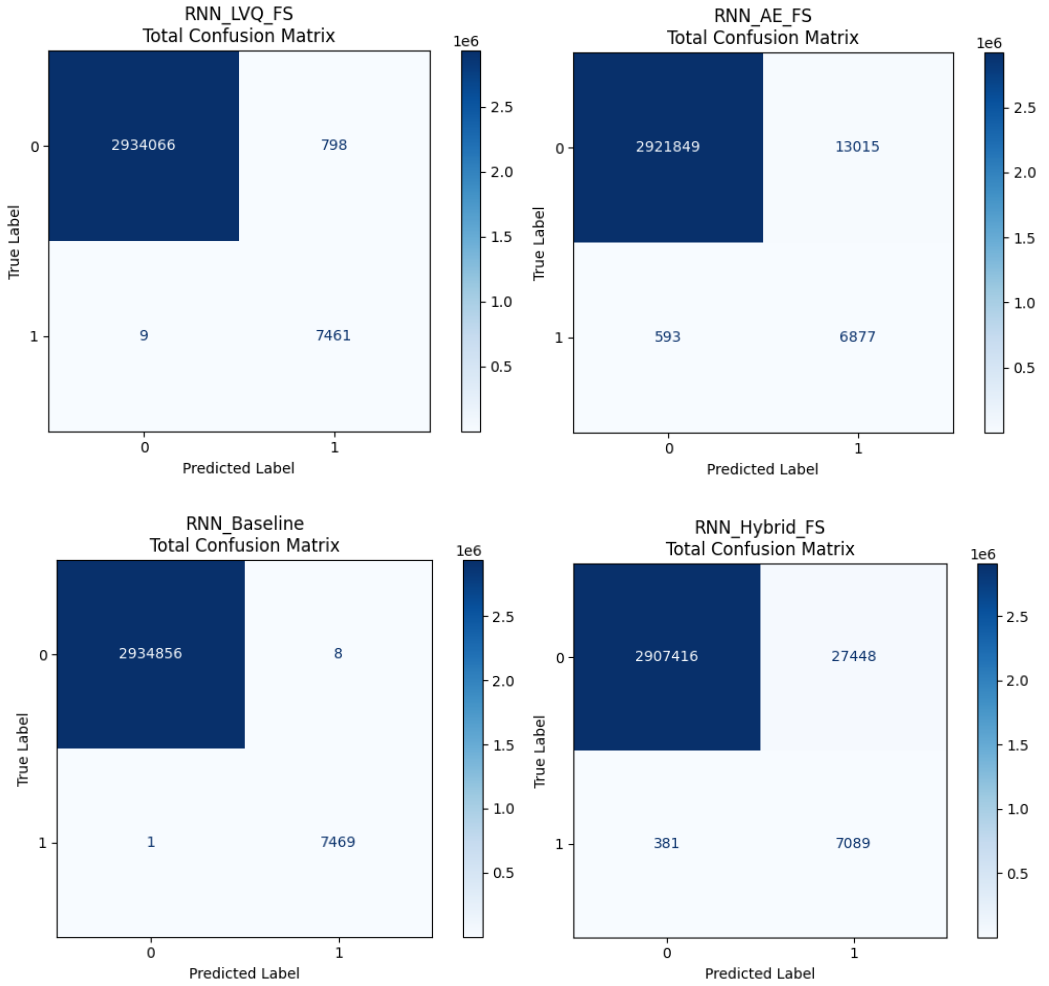
RNN + Autoencoder	99,5375	92,0616	38,8733	53,6318	97,0865	15.019
RNN + Combinasi (LVQ & Autoencoder)	99,0542	94,8996	20,9398	34,2119	98,8402	15.731

Berdasarkan hasil Tabel 4.3 menyajikan hasil evaluasi performa dalam mendeteksi serangan DDoS, berdasarkan lima metrik utama: **akurasi**, **presisi**, **recall**, **F1-score**, dan **Area Under Curve (AUC)**.

Model **RNN_Baseline** (tanpa seleksi fitur) memperlihatkan performa paling tinggi di antara semua skenario. Akurasi yang dicapai hampir sempurna, yaitu **99,9997%**, dengan recall **99,9866%**, presisi **99,8933%**, serta F1-score **99,9399%**. Nilai AUC juga sangat tinggi, yakni **99,9961%**, yang menunjukkan kemampuan model hampir sempurna dalam membedakan trafik normal (BENIGN) dan serangan (ATTACK). Namun, performa yang terlalu tinggi ini menimbulkan indikasi adanya bias terhadap kelas mayoritas. Mengingat bahwa kelas ATTACK jauh lebih dominan dibanding BENIGN dalam dataset CIC-DDoS2019, model baseline berpotensi terlalu mengandalkan distribusi mayoritas sehingga meskipun metrik global terlihat sempurna, sensitivitas nyata terhadap variasi trafik minoritas masih perlu diwaspadai. Model **RNN_LVQ_FS**, yang menerapkan seleksi fitur berbasis Learning Vector Quantization, mencatat hasil yang sedikit lebih rendah dibanding baseline, dengan akurasi **99,9726%**, recall **99,8795%**, presisi **90,3477%**, dan F1-score **94,8721%**. Nilai AUC tetap sangat tinggi (**99,9919%**), hanya sedikit di bawah baseline. Penurunan presisi menunjukkan bahwa model lebih sering menghasilkan **false positive**, yaitu **trafik BENIGN yang diklasifikasikan sebagai ATTACK**. Meski demikian, performa ini menunjukkan bahwa reduksi fitur dari 85 menjadi 16 masih dapat mempertahankan kemampuan deteksi yang sangat baik. Dengan kompleksitas model yang lebih rendah, pendekatan LVQ ini mampu menyeimbangkan efisiensi komputasi dan ketahanan terhadap bias mayoritas..

Model **RNN_AE_FS**, yang menggunakan Autoencoder untuk reduksi fitur, menunjukkan penurunan performa yang lebih signifikan. Akurasi tercatat

99,5375%, dengan recall **92,0616%**, namun presisi menurun drastis menjadi **38,8733%** dan F1-score hanya **53,6318%**. Nilai AUC sebesar **97,0865%** masih mengindikasikan kemampuan **klasifikasi yang baik secara global**, tetapi angka **presisi yang rendah** menandakan banyak terjadinya **false positive**. Dengan kata lain, model cenderung mengklasifikasikan trafik BENIGN sebagai serangan. Hal ini dapat dijelaskan karena Autoencoder bekerja secara unsupervised, lebih berfokus pada **rekonstruksi data ketimbang memaksimalkan separabilitas antar kelas**, sehingga fitur yang dihasilkan tidak sepenuhnya optimal untuk klasifikasi biner. Model RNN_Hybrid_FS, yang menggabungkan LVQ dan Autoencoder, memperlihatkan hasil yang paling rendah dari sisi presisi dan F1-score. Akurasi yang diperoleh adalah **99,0542%**, dengan recall **94,8996%**, tetapi presisi sangat rendah yaitu **20,9398%**, menghasilkan F1-score **34,2119%**. Meskipun AUC masih relatif tinggi (**98,8402%**), performa ini menunjukkan bahwa model hybrid terlalu **sensitif terhadap trafik serangan (ATTACK)**, tetapi dengan konsekuensi menghasilkan alarm palsu yang sangat tinggi. Hal ini kembali menegaskan adanya bias ke kelas mayoritas. Dengan karakteristik seperti ini, **model hybrid mungkin lebih cocok digunakan sebagai sistem deteksi dini (early warning)**, di mana recall yang tinggi lebih diprioritaskan dibanding presisi. Namun, dalam sistem real-time yang membutuhkan tingkat false positive rendah, model ini memerlukan **penyesuaian ambang (threshold tuning) atau strategi pasca-klasifikasi**. hasil evaluasi menunjukkan bahwa model **RNN_Baseline** unggul pada semua metrik, tetapi berpotensi bias ke kelas mayoritas sehingga keandalannya dalam mendeteksi trafik BENIGN masih perlu diwaspadai. Model RNN_LVQ_FS tampil paling seimbang karena mampu menjaga nilai recall yang sangat tinggi sekaligus mempertahankan presisi pada level yang masih dapat diterima, dengan keunggulan tambahan berupa reduksi fitur yang signifikan sehingga lebih efisien dan tahan terhadap overfitting. Sebaliknya, model RNN_AE_FS dan RNN_Hybrid_FS memiliki kelemahan utama pada presisi, yang berdampak pada meningkatnya risiko false positive. Walaupun nilai AUC keduanya tetap tinggi, penggunaan praktis model ini lebih cocok untuk sistem **deteksi dini (early warning).**



Gambar 4. 12 *Confution Matrix* pada masing-masing sekenario

Berdasarkan Gambar 4.12, disajikan hasil evaluasi performa klasifikasi melalui *confusion matrix* dari empat skenario model *Recurrent Neural Network* (RNN). Evaluasi ini penting untuk mengamati bagaimana masing-masing model membedakan antara trafik serangan (ATTACK) sebagai kelas mayoritas (direpresentasikan sebagai Kelas 0) dan trafik normal (BENIGN) sebagai kelas minoritas (direpresentasikan sebagai Kelas 1) secara lebih mendalam.

Model RNN_Baseline menunjukkan hasil klasifikasi yang nyaris sempurna. Model ini dengan benar mengidentifikasi 2.934.856 trafik ATTACK (*True Negative*) dan 7.469 trafik BENIGN (*True Positive*). Kesalahan yang terjadi sangat minim, dengan hanya 1 trafik BENIGN yang salah diklasifikasikan sebagai serangan (*False Negative*) dan 8 trafik serangan yang lolos dan dianggap sebagai

trafik normal (*False Positive*). Performa yang sangat tinggi ini menegaskan temuan sebelumnya, namun juga **memperkuat dugaan adanya bias terhadap kelas mayoritas ATTACK** dan risiko *overfitting* yang tinggi karena model cenderung "menghafal" pola data latih.

Pada model RNN_LVQ_FS, distribusi hasil prediksi menunjukkan 7.461 TP (BENIGN terdeteksi benar) dan 2.934.066 TN (ATTACK terdeteksi benar). Dibandingkan *baseline*, terjadi peningkatan pada kedua jenis kesalahan: jumlah trafik BENIGN yang salah diklasifikasikan sebagai serangan (*False Negative*) naik menjadi 9, dan yang lebih signifikan, jumlah serangan yang lolos deteksi (*False Positive*) meningkat tajam menjadi 798. Peningkatan jumlah serangan yang lolos ini konsisten dengan penurunan metrik presisi yang diamati sebelumnya, yang menunjukkan bahwa reduksi fitur oleh LVQ menjaga kemampuan deteksi secara umum namun dengan sedikit pengorbanan pada ketepatan.

Model RNN_AE_FS menunjukkan performa yang kurang seimbang, dengan 6.877 TP (BENIGN terdeteksi benar) dan 2.921.849 TN (ATTACK terdeteksi benar). Terjadi peningkatan kesalahan yang signifikan, di mana 593 trafik BENIGN salah diklasifikasikan sebagai serangan (FN), dan 13.015 trafik ATTACK lolos deteksi (FP). Jumlah FP yang sangat besar ini menjadi penyebab utama anjloknya presisi model, yang konsisten dengan hasil evaluasi metrik sebelumnya. Hal ini mengindikasikan bahwa fitur hasil **ekstraksi Autoencoder kurang optimal dalam menciptakan batas pemisah yang tegas antara kedua kelas**.

Model RNN_Hybrid_FS yang mencatat **TP = 7.089**, **TN = 2.907.416**, **FP = 27.448**, dan **FN = 381**. Model ini menghasilkan **akurasi 99,0542%**, **recall 94,8996%** (tertinggi kedua setelah baseline), namun **presisi sangat rendah sebesar 20,9398%**, dan **F1-score 34,2119%**. Nilai recall yang tinggi menunjukkan bahwa model sangat jarang gagal mendeteksi serangan, namun hal itu dicapai dengan konsekuensi tingginya alarm palsu (FP). Ini mencerminkan ***trade-off* ekstrem antara sensitivitas dan presisi**, di mana model lebih agresif dalam mendeteksi serangan, namun cenderung **mengorbankan akurasi klasifikasi terhadap data normal**.

Hasil evaluasi confusion matrix ini menunjukkan bahwa:

- Model RNN_Baseline menunjukkan performa nyaris sempurna karena kemampuannya 'menghafal' data latih, bukan karena generalisasi yang baik, sehingga sangat rentan terhadap overfitting.
- Model RNN_LVQ_FS tampil paling seimbang, dengan sensitivitas (recall) yang tetap tinggi dan tingkat kesalahan klasifikasi (false positive) yang masih dapat ditoleransi.
- Model RNN_AE_FS dan terutama RNN_Hybrid_FS menunjukkan kecenderungan lebih agresif dalam mendeteksi serangan (recall tinggi), tetapi dengan konsekuensi meningkatnya jumlah false positive yang secara drastis menurunkan presisi.

Berdasarkan hasil yang telah dipaparkan, dapat disimpulkan bahwa penerapan teknik seleksi dan ekstraksi fitur berperan penting dalam membentuk karakteristik model. Pendekatan LVQ terbukti paling seimbang, mampu menyederhanakan input sambil mempertahankan performa deteksi yang tinggi dengan tingkat alarm palsu yang wajar. Sebaliknya, Autoencoder dan kombinasi Hybrid menghasilkan model yang sangat sensitif (recall tinggi) tetapi dengan presisi yang sangat rendah. Oleh karena itu, pemilihan model harus disesuaikan dengan kebutuhan skenario implementasi: RNN_LVQ_FS direkomendasikan untuk sistem deteksi yang membutuhkan keseimbangan optimal. Model ini andal karena mampu menjaga performa deteksi tetap tinggi sambil meminimalkan jumlah alarm palsu. RNN_Hybrid_FS lebih cocok diimplementasikan sebagai sistem peringatan dini (*early warning*). Dalam skenario ini, prioritas utamanya adalah mendeteksi sebanyak mungkin potensi serangan agar tidak ada yang terlewat, bahkan jika konsekuensinya adalah harus menangani lebih banyak alarm palsu.

Tabel 4. 6 Perbandingan 16 Fitur Teratas Hasil Seleksi

NO	Learning Vector Quantization	Nilai	Autoencoder	Nilai	Gabungan (LVQ-Autoencoder)	Nilai
1	Destination Port	0.1376	Flow IAT Std	230.378	Down/Up Ratio	1.0000
2	Fwd Packets/s	0.1230	Flow IAT Max	221.148	Fwd IAT Max	0.8710
3	Flow Packets/s	0.1189	Fwd IAT Std	219.899	Flow IAT Std	0.8648
4	Source Port	0.0917	Fwd IAT Max	219.240	Fwd IAT Std	0.8369
5	Flow Bytes/s	0.0622	Down/Up Ratio	213.273	Fwd Packets/s	0.8163
6	Fwd Packet Length Min	0.0598	Flow IAT Mean	211.345	Fwd Packet Length Std	0.8161
7	Min Packet Length	0.0597	Fwd IAT Mean	192.952	Packet Length Std	0.7866
8	ACK Flag Count	0.0542	Active Std	191.260	Flow IAT Max	0.7510
9	min_seg_size_forward	0.0512	Bwd IAT Min	190.403	Flow IAT Mean	0.7438
10	Timestamp	0.0494	Bwd IAT Mean	189.859	Fwd IAT Mean	0.7166
11	**Fwd IAT Total**	**0.0228**	**Fwd IAT Total**	**187.412**	**Fwd IAT Total**	**0.7095**
12	**Flow Duration**	**0.0228**	**Flow Duration**	**174.868**	Bwd Packet Length Mean	0.6514
13	Protocol	0.0220	Fwd Packet Length Std	174.410	**Flow Duration**	**0.6339**
14	Avg Fwd Segment Size	0.0180	Bwd IAT Std	173.681	Bwd IAT Std	0.6271
15	Fwd Packet Length Mean	0.0180	Bwd Packet Length Mean	170.128	Bwd IAT Min	0.6169
16	Average Packet Size	0.0178	Bwd Packet Length Min	169.786	Bwd IAT Max	0.6031

Setelah membandingkan ketiga metode seleksi fitur, ditemukan adanya titik temu signifikan yang menunjukkan adanya fitur-fitur yang bersifat universal dalam deteksi anomali. Terdapat dua fitur yang secara konsisten dianggap penting dan terpilih oleh ketiga algoritma, yaitu **Flow Duration** dan **Fwd IAT Total**. Kehadiran keduanya di semua daftar mengindikasikan bahwa **konteks durasi dan totalitas waktu** dari sebuah sesi komunikasi adalah indikator anomali yang sangat fundamental. **Flow Duration** menjadi krusial karena serangan DDoS seringkali memanipulasi durasi sesi, baik dengan membuatnya sangat pendek untuk serangan *burst* mendadak, maupun sangat panjang untuk serangan *slow-rate* yang bertujuan mengikat sumber daya server secara perlahan. Sementara itu, **Fwd IAT Total**, yang merupakan jumlah total jeda waktu antar paket dari arah penyerang, memberikan gambaran mengenai perilaku pengiriman data. Nilai yang tidak wajar pada fitur ini dapat menandakan lalu lintas yang terlalu "rapat" dan tanpa henti khas mesin, atau sebaliknya, terlalu sporadis dan tidak natural. Kemampuan kedua fitur ini untuk menangkap anomali pada spektrum perilaku waktu yang luas inilah yang membuatnya dianggap relevan baik oleh pendekatan *supervised* maupun *unsupervised*.

Perbedaan mendasar pada fitur-fitur lain yang terpilih berakar dari **perbedaan filosofi analitis** antara pendekatan *supervised* dan *unsupervised*. LVQ, sebagai algoritma *supervised*, beroperasi layaknya seorang "detektif pragmatis" yang tujuannya adalah menemukan bukti yang paling jelas dan langsung untuk

memisahkan "pelaku" dari "korban". Oleh karena itu, secara logis memprioritaskan fitur-fitur **volumetrik dan target** seperti **Destination Port** dan **Fwd Packets/s**. Fitur-fitur ini adalah indikator serangan tingkat pertama (*first-order indicators*); mereka secara langsung mengukur "kekuatan" serangan dan "sasaran" tembaknya. Sebaliknya, Autoencoder yang *unsupervised* bertindak seperti seorang "analis struktural" yang terlebih dahulu membangun model "kenormalan" lalu lintas jaringan yang berdefinisi tinggi. Ia tidak mencari "serangan", melainkan mencari segala bentuk **penyimpangan atau anomali yang merusak struktur normal tersebut**. Inilah sebabnya ia unggul dalam mengidentifikasi indikator tingkat kedua (*second-order indicators*) yang lebih subtil, seperti **Flow IAT Std** yang mengukur *jitter* atau kekacauan temporal, dan **Down/Up Ratio** yang mengukur asimetri perilaku dalam sebuah sesi komunikasi yang seharusnya berjalan dua arah.