Perbandingan Algoritma SVM dan Algoritma KNN dalam Menghasilkan Klasifikasi DDoS dan *Benign*

Patrick Darryl[1] dan Muhammad Subali[2]

[1]Teknologi Informasi, Teknik Elektro, Universitas Gunadarma
Jl. Salemba Raya No.53, RT.9/RW.5, Paseban, Senen, Kota Jakarta Pusat, DKI Jakarta 10440
[2]Teknik Informatika Universitas Cendekia Abditama
Jl. Islamic Raya No.1, Kelapa Dua, Kelapa Dua, Tangerang, Banten 15812
E-mail : starry.azuresky@gmail.com, subali@cendekia.ac.id

Abstrak

Perkembangan teknologi dan internet semakin cepat. Semakin terbukanya pengetahuan hacking dan cracking maka banyak kelompok yang tidak bertanggung jawab mencoba untuk mengambil atau mencuri informasi, salah satu serangan yang dapat dilakukan adalah *Distributed Denial of Service* (DDoS). Ada beberapa jenis serangan dari DDoS yang sering terjadi seperti *UDP Flooding*, *SYN Flooding*, *Ping of Death*, dan *Remote Controlled Attack*. Serangan DDoS mengakibatkan sistem yang diserang mengalami gangguan berupa *error request, halt,* kegagalan sistem dan sebagainya. Berdasarkan penjelasan diatas, serangan DDoS dan nantinya log serangan akan dikirim kedalam *database* untuk proses pengklasifikasian menggunakan SVM dan KNN dengan tujuan mencegah jika terjadi serangan dan mempermudah proses klasifikasi maupun analisa, menghasilkan output mudah dibaca oleh administrator jika terdapat serangan terhadap *server* dan *administrator* dapat mengatasinya terhadap serangan terhadap *server* tersebut. Penelitian ini telah menghasilkan klasifikasi DDoS dan *Benign* pada dataset CCIDS menggunakan algoritma SVM dan *K-Nearest Neighbor*. Penelitian ini berhasil menampilkan perbedaan performa dari algoritma SVM dan *K-Nearest Neighbor* dalam menghasilkan klasifikasi DDoS dan *Benign*. SVM dengan nilai *Accuracy* 90,75%, *Precision* 89,33%, *Recall* 91,38% dan F1- *Score* 90,27% sedangkan KNN dengan nilai *Accuracy* 95,50%, *Precision* 99,12%, *Recall* 93,84% dan F1- *Score* 96,39%.

Kata kunci : Klasifikasi, SVM, KNN, DDoS, *Benign*

Pendahuluan

Perkembangan teknologi dan internet semakin cepat. Maka dari itu, semakin banyak pula informasi data yang sangat penting tersebut perlu untuk dilindungi karena para hacker akan melakukan berbagai cara untuk mendapatkan informasi atau data yang sangat penting tersebut. Semakin terbukanya pengetahuan hacking dan cracking maka banyak kelompok yang tidak bertanggung jawab mencoba untuk mengambil atau mencuri informasi, salah satu serangan yang dapat dilakukan adalah *Distributed Denial of Service* (DDoS).

Pada saat ini metode yang sering digunakan oleh intruder adalah distributed denial of service (DDoS). DDoS merupakan sebuah metode serangan dengan mengirimkan banyak paket ke dalam dalam sebuah jaringan yang menyebabkan perangkat jaringan tidak lagi dapat berjalan sesuai fungsinya [10]. Ada beberapa jenis serangan dari DDoS yang sering terjadi seperti *UDP Flooding, SYN Flooding, Ping of Death*, dan *Remote Controlled Attack*. Serangan DDoS mengakibatkan sistem yang diserang mengalami gangguan berupa error request, halt, kegagalan sistem dan sebagainya. Inilah yang mungkin menjadi alasan mengapa perlu keamanan dalam komputer atau keamanan informasi bagi sebuah organisasi.

Penelitian yang dilakukan oleh M. Fibrianda and A. Bhawiyuga [1] membahas mengenai pendeteksian atau pencegahan berbagai potensi serangan telah dikembangkan *Intrusion Detection System* (IDS). Penelitian ini menganalisis perbandingan metode yang dihasilkan dari proses klasifikasi berdasarkan nilai akurasi *confusion matrix, precision, recall, dan f1 score. Naive Bayes, SVM Linear, SVM Polynomial* dan *SVM Sigmoid* menghasilkan persentase akurasi berturut-turut sebesar 85,055%, 99,995%, 99,999%, dan 99,995%.

Penelitian yang dilakukan oleh H. E. Wahanani, B. Nugroho and G. I. Prakoso [2] membahas mengenai serangan atau intrusi sangat tidak diinginkan

DOI : http://dx.doi.org/10.32409/jikstik.20.4.2799

pada sistem jaringan komputer karena bisa membahayakan integritas, kerahasiaan dan ketersediaan sumber daya yang ada. Hasil penelitian ini adalah klasifikasi serangan dengan metode SVM menghasilkan tingkat akurasi yang cukup tinggi untuk masing-masing serangan dengan rata-rata *class* yang diprediksi di atas 60%.

Penelitian yang dilakukan oleh N. Hassni dan F. R. Putri [5] membahas mengenai perkembangan infrastruktur perangkat lunak terjadi secara besar-besaran dan mempermudah para pelaku kejahatan malicious website untuk melakukan serangan. Metode KNN dilakukan dengan cara membandingkan data uji dengan data training. Tingkat akurasi yang dihasilkan cukup memuaskan dengan nilai sebesar 92.2%.

Pada penelitian ini, serangan DDoS dan nantinya log serangan/dataset akan dikirim kedalam database untuk proses pengklasifikasian menggunakan SVM dan KNN dengan tujuan mencegah jika terjadi serangan dan mempermudah proses klasifikasi maupun analisa, menghasilkan output mudah dibaca oleh administrator jika terdapat serangan terhadap server dan administrator dapat mengatasinya terhadap serangan terhadap server tersebut. Dataset yang digunakan adalah CICIDS 2018 [11].

Serangan *Distributed Denial of Service* (DDoS) adalah suatu serangan yang dilakukan untuk menghabiskan sumber daya komputer atau jaringan komputer dengan mengirimkan lalu lintas (*traffic*) yang padat. Serangan DDoS dimulai dari penyerang yang mendistribusikan serangan dengan menggunakan mesin yang berbeda [4]. DDoS merupakan sebuah metode serangan dengan mengirimkan banyak paket ke dalam dalam sebuah jaringan yang menyebabkan perangkat jaringan tidak lagi dapat berjalan sesuai fungsinya [3]. *Support Vector Mechine* (SVM) adalah sebuah algoritma yang bekerja menggunakan pemetaan nonlinear untuk mengubah data pelatihan asli ke dimensi yang lebih tinggi, dalam dimensi yang baru, kemudian akan mencari linear optimal pemisah hyperplane (yaitu, "*decision boundary*" yang memisahkan tupel dari satu kelas dengan kelas lainnya) [6]. K-*Nearest Neighbor* (KNN) merupakan suatu metode untuk melakukan klasifikasi terhadap objek yang diuji berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Kelas hasil klasifikasi adalah kelas yang paling banyak muncul [7].

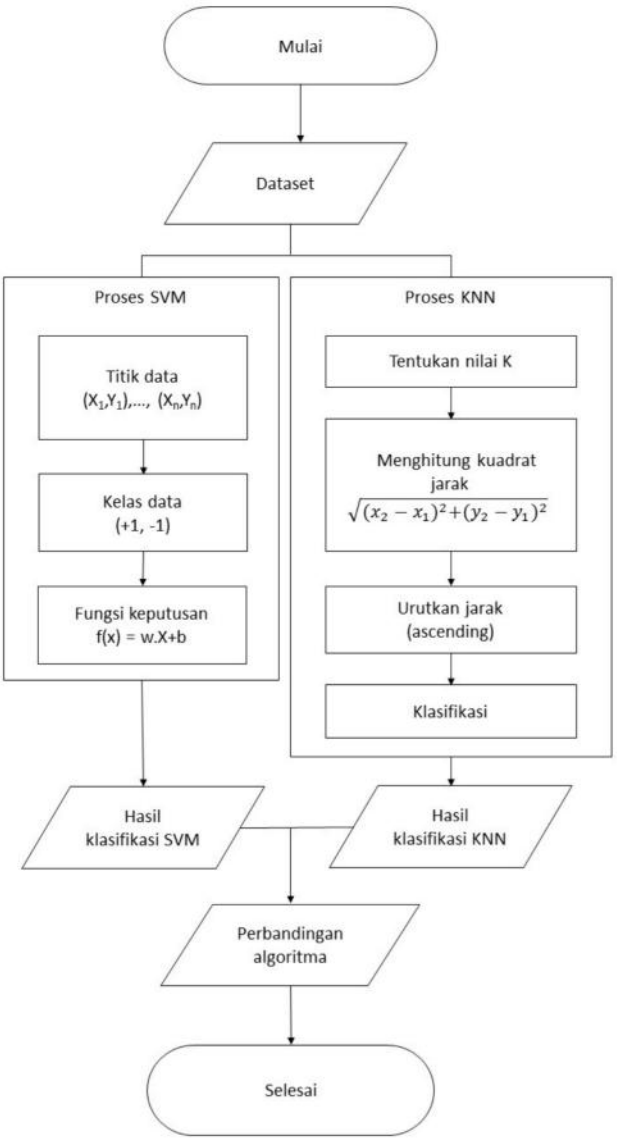
Untuk mengetahui performa algoritma SVM dan KNN dalam proses klasifikasi serangan DDoS maka dibutuhkan suatu metode yang dapat menghitung dan mengukur apakah SVM dan KNN sudah tepat sesuai dengan yang diharapkan. Metode yang digunakan untuk menguji algoritma SVM dan algoritma KNN adalah *confusion matrix*. *True positives* (TP) adalah jumlah *record* positif yang diklasifikasikan sebagai positif, *false positives* (FP) adalah jumlah *record* negatif yang diklasifikasikan sebagai positif, *false negatives* (FN) adalah jumlah *record* positif yang diklasifikasikan sebagai negatif, *true negatives* (TN) adalah jumlah *record* negatif yang diklasifikasikan sebagai negatif [8]. Setelah data uji dimasukkan ke dalam *confusion matrix*, hitung nilai-nilai yang telah dimasukkan tersebut untuk dihitung jumlah *accuracy, precision, recall dan f-1 score*. *Accuracy* adalah jumlah prediksi yang benar (TP dan TN) dibagi dengan jumlah semua sampel. *Precision* mengukur berapa banyak sampel yang diprediksi positif dari realita positif. *Recall* mengukur berapa banyak sampel positif yang ditangkap oleh prediksi positif. F-1 *score* adalah rata-rata harmonik presisi dan *recall*.

Metode Penelitian

Metode yang digunakan dalam membuat, menguji dan menganalisis penelitian ini diawali dengan studi literatur untuk mempelajari materi-materi yang terkait. Pembelajaran ini dapat mengambil bahan pustaka dari jurnal, buku penunjang, datasheet serta media informasi lainnya. Melalui metode ini diinginkan untuk mendapatkan ide-ide, bahan rancangan, dan hasil penelitian. Selanjutnya dilakukan pada tahap perancangan dengan menggunakan dataset yang telah diklasifikasi selanjutnya di-training dengan menggunakan algoritma *Support Vector Machine* (SVM) dan *K-Nearest Neighbo*r (KNN) kemudian diimplemetasikan kedalam program menggunakan bahasa programan PHP dan MySQL. Terakhir adalah tahap uji coba dan analisa untuk mengetahui performa dari kedua algoritma tersebut menggunakan *Confusion Matrix*.

Perancangan

Berdasarkan diagram alir dibawah dapat dijelaskan bahwa dataset yang dimasukkan akan diproses oleh dua algoritma (*support vector machine* dan *K-Nearest Neighbor*) dimana SVM harus menentukan titik data dulu untuk menghasilkan hyperplane, kemudian membuat kelas data, dan memberikan hasil klasifikasi. Sementara algoritma KNN akan menentukan jumlah kelas terlebih dahulu (nilai K) kemudian hitung jarak dari data yang dimasukkan terhadap data uji sehingga menghasilkan jarak dari masing-masing kelas atau klasifikasi yang ditentukan. Kedua algoritma yang telah menghasilkan kelas akan dibandingkan untuk menghasilkan pengetahuan terkait performasi kedua algoritma dalam melakukan klasifikasi data serangan DDoS. Gambar 1 adalah diagram alir pada sistem klasifikasi serangan DDoS menggunakan algoritma SVM dan KNN.



Gambar 1: Diagram Alir Sistem Klasifikasi Serangan DDoS Menggunakan Algoritma SVM dan KNN

Berdasarkan Gambar 1, dataset yang dimasukkan akan diproses oleh dua algoritma (*support vector machine* dan *k-nearest neighbor*). Menggunakan dua algoritma tersebut, dataset dites klasifikasinya. Hasil klasifikasi SVM dan KNN diuji performanya menggunakan *confusion matrix*.

Algoritma SVM

1. Implementasi SVM

Diketahui :

Sampel data dari kelas yang dimiliki dataset dengan masing-masing memiliki nilai seperti pada Tabel 1.

Tabel 1: Data Train

X	src_ip	src_port	dst_ip	dst_port	Kelas
Data	18.216.200.189	63095	172.31.69.28	80	*DDoS*
	172.31.67.99	48205	193.36.45.134	80	*Benign*

Encode :

Proses ini digunakan untuk memberikan inisial kepada setiap atribut yang ada pada setiap kelas dimana jika atribut yang ada di DDoS diberikan inisial 0, dan atribut yang ada di *Benign* diberikan inisial 1 seperti tampak pada Tabel 2.

Tabel 2: Encode Data Train

X	src_ip	src_port	dst_ip	dst_port	Kelas
Data	0	0	0	0	DDoS
	1	1	1	0	*Benign*

Input :

Data yang dimasukkan untuk menghasilkan kelas (lihat pada Tabel 3).

Tabel 3: Data Test

src_ip	src_port	dst_ip	dst_port
18.216.200.189	52341	172.31.69.28	80

Encode :

Inisialkan setiap kriteria dari data yang di-input (tampak pada Tabel 4).

Tabel 4: Encode Data Test

src_ip	src_port	dst_ip	dst_port
1	0	1	1

Gunakan persamaan untuk mengelompokkan nilai kriteria data input terhadap data test yang diketahui.

$$f(x) = x1 + x2 - 1$$

$$kelas = sign(f(x))$$

dengan:

x1 = jumlah kriteria pada kelas DDoS masuk ke kelas (+)

x2 = jumlah kriteria pada *Benign* masuk ke kelas (-)

Hasil :

Berdasarkan data input terdapat 3 kriteria yang termasuk ke dalam kelas DDoS dan 1 kriteria termasuk ke dalam kelas Benign. Tabel 5 adalah klasifikasi yang dihasilkan.

Tabel 5: Hasil Klasifikasi

x data		Hasil Klasifikasi	Kelas
x1	x2	Kelas = sign (x1 + x2 -1)	
3	-1	Sign (3 + (-1) – 1) = +1	DDoS

1. Implementasi *K-Nearest Neighbor*

Diketahui data pada Tabel 6.

Tabel 6: Data Train

X	src_ip	src_ port	dst_ ip	dst port	Kelas
Data	18.216.200. 189	63095	172.31. 69.28	80	DDoS
	172.31.67. 99	48205	193.36. 45.134	80	*Benign*

Encode :

Buat inisial setiap kriteria dimana 1 adalah kriteria milik kelas DDoS, 2 adalah kriteria milik *Benign*, dan 3 adalah kriteria yang sama dimiliki oleh kedua kelas seperti tampak pada Tabel 7.

Tabel 7: *Encode Data Train*

X	src_ip	src_port	dst_ip	dst_port	Kelas
Data	1	1	1	3	DDoS
	2	2	2	3	*Benign*

Input (lihat pada Tabel 8).

Tabel 8: *Data Test*

src_ip	src_port	dst_ip	dst_port
18.216.200.189	52341	172.31.69.28	80

Encode :

Dari data yang di-input maka didapatkan hasil inisialisasi seperti tampak pada Tabel 9.

Tabel 9: *Data Test*

src_ip	src_port	dst_ip	dst_port
0	0	0	2

Proses hitung KNN :

Hitung kedekatan kriteria yang dimiliki data input terhadap kriteria data yang telah diketahui untuk mendapatkan nilai terkecil sebagai penentu kelas seperti berikut.

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (2-2)^2} = 0,00$$

Hasil KNN disajikan pada Tabel 10.

Tabel 10: *Hasil Klasifikasi*

X	src_ip	src_port	dst_ip	dst_port	Jarak	Kelas
Data	0	0	0	2	0,00	*DDoS*
	1	1	1	2	1,73	*Benign*

Dari hasil tersebut, jika diambil nilai K terdekat, maka tetangga terdekat dari data input adalah DDoS dengan jarak terdekat 0,00.

Berdasarkan proses klasifikasi menggunakan algoritma SVM dan KNN dapat disimpulkan bahwa algoritma K-Nearest Neighbor dapat menghasilkan klasifikasi dengan proses yang sederhana dengan cara menghitung kedekatan dengan sebuah persamaa. Sedangkan algoritma SVM memiliki proses yang rumit dimana harus mencari jumlah kriteria yang termasuk dalam setiap kelas yang ditentukan, setelahnya menentukan garis pemisah dan jarak untuk memastikan perbedaan dari kelas yang ditentukan.

Pembuatan Sistem

Alur tahapan dalam pembuatan website terdapat beberapa tahapan seperti pada Gambar 2. Dimulai dengan persiapan data, pada tahapan ini penulis melakukan persiapan data diantaranya, mengubah format data ke csv, menghilangkan fitur/atribut yang redundan. Kemudian melakukan pemisahan data sebagai sampel.

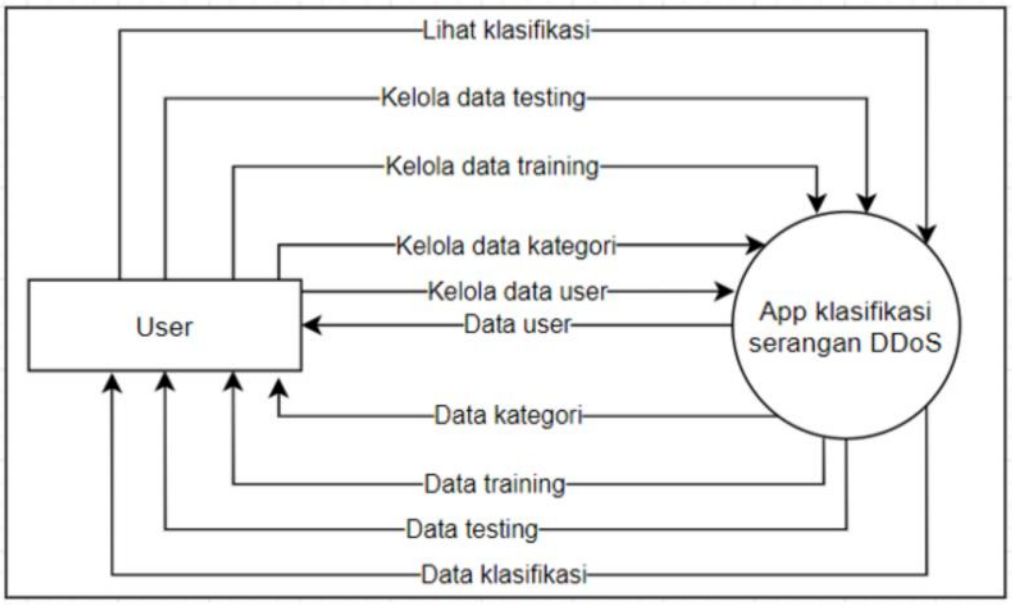
Gambar 2: Alur Pembuatan Sistem

Tahap selanjutnya adalah seleksi fitur, teknik seleksi fitur yang digunakan pada penelitian ini adalah *Information Gain* yang merupakan teknik *filtered-based*. Tujuan dari seleksi fitur ini adalah untuk mendapatkan fitur yang paling relevan untuk digunakan dalam deteksi serangan DDoS. Selanjutnya, dataset hasil seleksi fitur, di-training dengan menggunakan metode klasifikasi. Tujuannya adalah untuk melihat perbandingan performa metode klasifikasi dalam mendeteksi serangan DDoS. Metode yang digunakan dalam training adalah *Support Vector Machine* dan *K-Nearest Neighbor*. Tahap terakhir adalah analisis dan interpretasi, berdasarkan hasil eksperimen kemudian data dianalisis dan diinterprestasi. Dalam penelitian ini, performa metode klasifikasi diukur dengan tingkat *Acuracy, Precision, Recall,* dan *F1-Score* berdasarkan hasil dari *confusion matrix.*

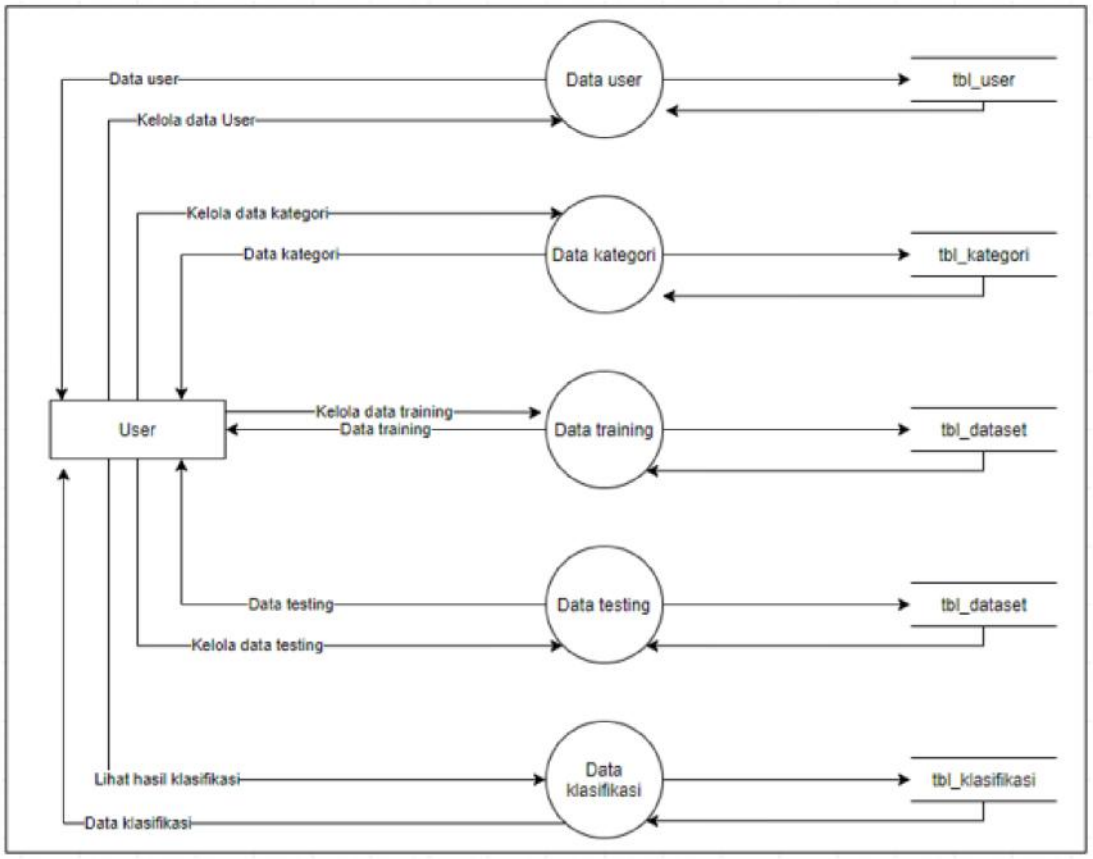
Data Flow Diagram

Data flow diagram atau yang disingkat DFD merupakan diagram yang menggambarkan alir data dalam suatu entitas ke sistem atau sistem ke entitas [9]. Perancangan *data flow diagram* digunakan untuk menggambarkan alur sistem secara keseluruhan yang berbasis pada objek-objek data yang digunakan terlihat pada Gambar 3 dan Gambar 4.

DFD level 0 menjelaskan bahwa pengguna memiliki hak akses pada kelola data user, kelola data kategori, kelola data *training*, kelola data testing dan melihat hasil klasifikasi, lihat Gambar 3.DFD level 1 menjelaskan bahwa pengguna dapat kelola data user yang ditampung oleh tabel user, kelola data kategori dengan penyimpanan tabel kategori, kelola data training dan kelola data testing yang ditampung oleh tabel dataset, dan lihat hasil klasifikasi berdasarkan tabel klasifikasi, lihat Gambar 4.



Gambar 3: DFD Level 0



Gambar 4: DFD Level 1

Perancangan *User Interface*

1. Halaman Login

   Pada halaman Login, admin dapat memasukkan username dan password untuk dapat masuk ke sistem jika data valid maka pengguna akan disuguhkan oleh halaman beranda/ *dashboard*, jika tidak valid maka pengguna tetap dihalaman Login.

2. Halaman Kelola

   Pengguna Pada halaman user ini admin dapat melakukan tambah, ubah dan hapus data user. Pada tombol tambah, pengguna akan disuguhkan halaman untuk menambah data user baru, pengguna (admin) dapat mengisi form yang ada pada tambah data user, kemudian simpan. Pada pilihan aksi ubah, pengguna (admin) dapat mengubah data pada data yang dipilih, jika telah dilakukan perubahan, maka pengguna (admin) menyimpan kemudian sistem menampilkan perubahan data. Pengguna dapat menghapus data yang dipilih dengan cara pilih aksi hapus, maka data yang dipilih akan hilang (tidak tampil di sistem).

3. Halaman Dataset

   Pada halaman kategori ini admin dapat melakukan memproses data training dan juga mengimport data. Pada tombol tambah, pengguna dapat menambahkan data training untuk diproses klasifikasi, sementara akses edit pada data training digunakan untuk mengubah data, dan aksi hapus digunakan untuk menghapus data training.

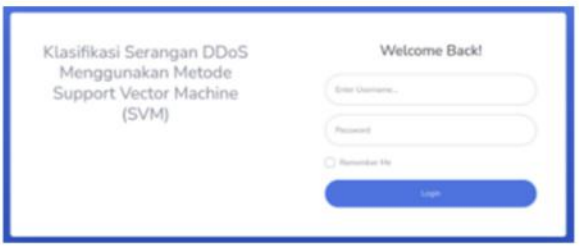
4. Halaman Klasifikasi

   Pada halaman kategori ini admin dapat melihat hasil klasifikasi serangan menggunakan SVM dan KNN. Pada form klasifikasi, pengguna dapat melihat accuracy, precision, recall dan f-1 score.

Hasil dan Pembahasan

Hasil dari perancangan untuk menghasilkan sistem klasifikasi seperti berikut:

1.) Halaman Login Halaman Login ditampilkan pertama ketika sistem dijalankan yang mengharuskan pengguna untuk memasukkan *username* dan *password.* Jika data yang dimasukkan valid, maka pengguna akan diarahkan ke halaman *dashboard,* dan jika gagal maka pengguna tetap di halaman Login, lihat Gambar 5.

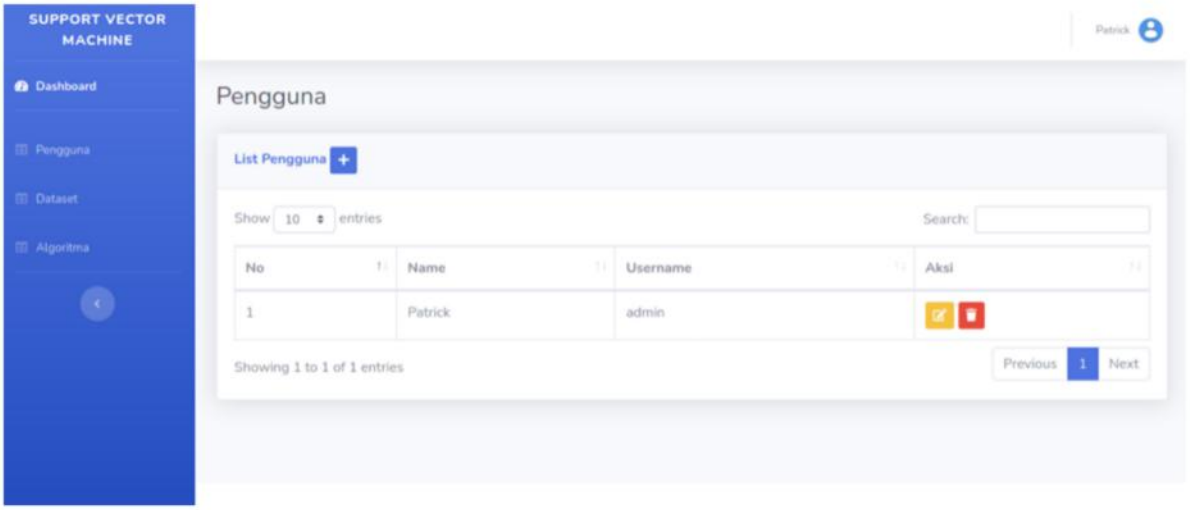


Gambar 5: Halaman Login

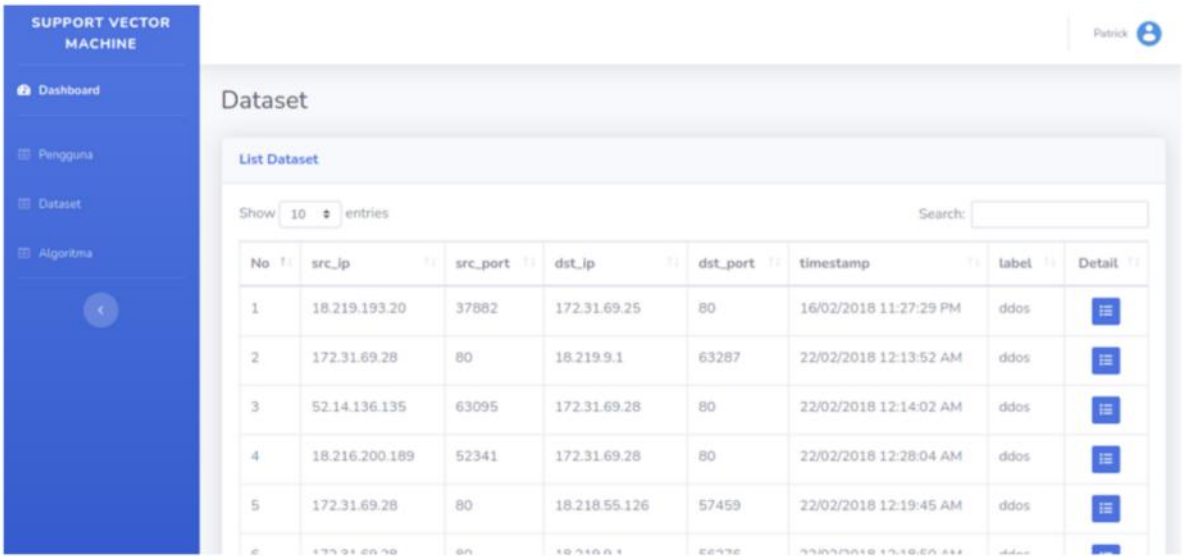
2.) Halaman kelola pengguna digunakan untuk mengolah data pengguna dengan tambah data pengguna, ubah data pengguna, dan hapus data pengguna seperti pada Gambar 6.

3.) Halaman Dataset Halaman dataset memuat dataset yang akan dianalisa untuk mendapatkan hasil klasifikasi setiap data yang ada seperti pada Gambar 7.

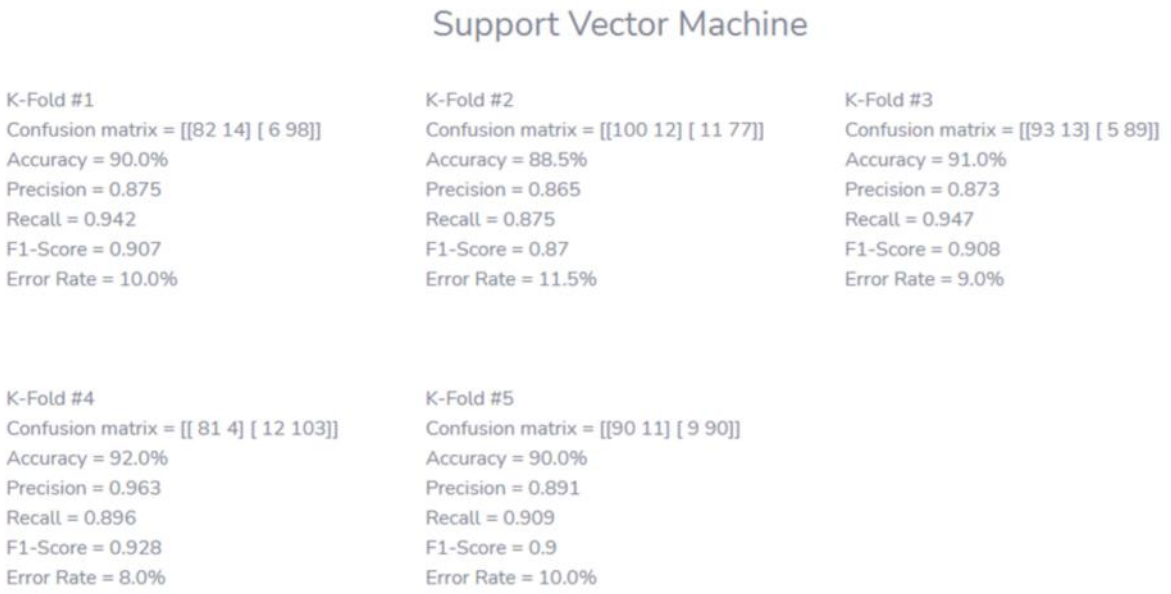
4.) Halaman Klasifikasi SVM Halaman klasifikasi SVM memuat data hasil klasifikasi yang diproses menggunakan algoritma SVM seperti pada Gambar 8.



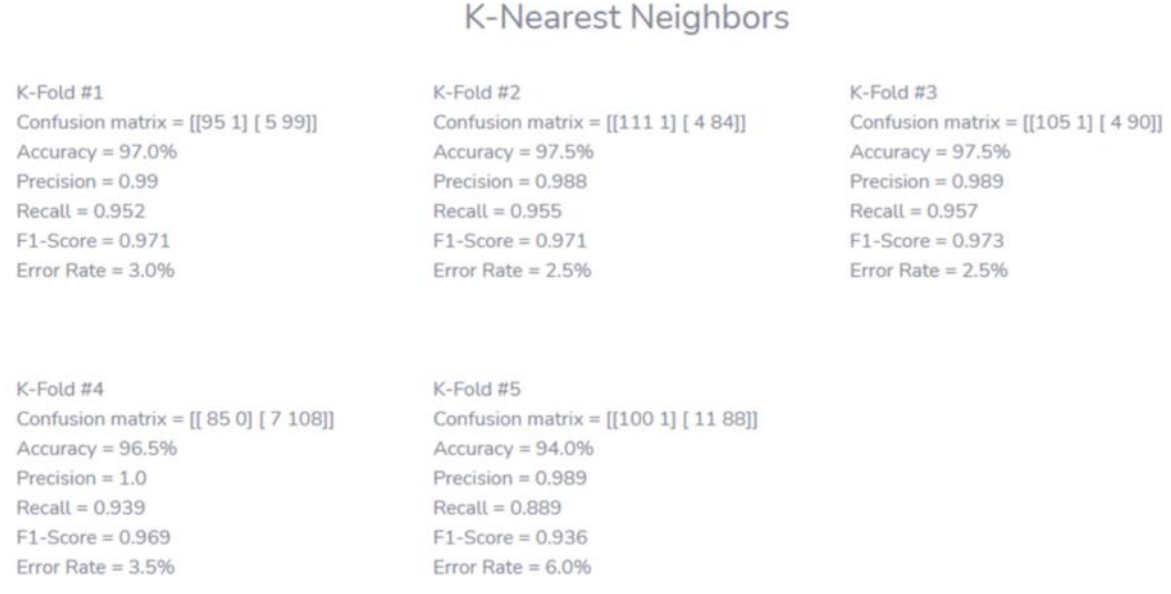
Gambar 6: Halaman Kelola Pengguna



Gambar 7: Halaman Dataset



Gambar 8: Halaman SVM



Gambar 9: Halaman KNN

5.) Halaman Klasifikasi *K-Nearest Neighbor* Halaman klasifikasi KNN memuat data hasil klasifikasi yang diproses menggunakan algoritma KNN seperti pada Gambar 9.

6.) Hasil Evaluasi

Tabel 11: *Data Pelatihan dan Data Pengujian*

Perbandingan Data		Jumlah Data	
Data Pelatihan	Data Pengujian	Data Pelatihan	Data Pengujian
80%	20%	800	200

Tabel 12: *Hasil Test SVM*

SVM		*Actual class*		Total
		K2 (*Benign*)	K1 (*DDoS*)	
		TN	FP	
Predicted class	K2 (*Benign*)	82	14	96
		FN	TP	
	K1 (*DDOS*)	6	98	104
		88	112	200

Pada Tabel 11 dijelaskan hasil dari akuisisi data kemudian lewat *preprocessing* data yang ada sebanyak 200 yang dibagi dalam 2 kategori yaitu kategori 1 adalah DDoS dan kategori 2 adalah *Benign*, lihat Tabel 12 dan 13.

$$Precision = \frac{TP}{(TP + FP)} = \frac{98}{(98 + 14)} = 0.875$$

$$Recall = \frac{TP}{(TP + FN)} = \frac{98}{(98 + 6)} = 0.942$$

$$F1 - Score = \frac{2 * recall * precision}{recall + precision}$$

$$= \frac{2 * 0.942 * 0.875}{0.942 + 0.875} = 0.907$$

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} * 100\%$$

$$F1 - Score = \frac{(98 + 82)}{(98 + 14 + 82 + 6)} * 100\%$$

$$= 90.0\%$$

Tabel 13: *Hasil Test KNN*

KNN		*Actual class*		Total
		K2 (*Benign*)	K1 (*DDOS*)	
		TN	FP	
Predicted class	K2 (*Benign*)	95	1	96
		FN	TP	
	K1 (*DDOS*)	5	99	104
		100	100	200

$$Precision = \frac{TP}{(TP + FP)} = \frac{99}{(99 + 1)} = 0.99$$

$$Recall = \frac{TP}{(TP + FN)} = \frac{99}{(99 + 5)} = 0.95$$

$$F1 - Score = \frac{2 * recall * precision}{recall + precision}$$

$$= \frac{2 * 0.95 * 0.99}{0.95 + 0.99} = 0.97$$

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} * 100\%$$

$$= \frac{(99 + 95)}{(99 + 1 + 95 + 5)} * 100\%$$

$$= 97.0\%$$

Berdasarkan implementasi algoritma, terdapat perbedaan performa dari pengklasifikasian serangan DDoS pada dataset CICIDS menggunakan algoritma *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN). SVM dengan nilai *Accuracy* 90,75%, *Precision* 89,33%, *Recall* 91,38% dan F1- *Score* 90,27% sedangkan KNN dengan nilai *Accuracy* 95,50%, *Precision* 99,12%, *Recall* 93,84% dan F1- *Score* 96,39%.

Penutup

Proses klasifikasi DDoS dan *Benign* pada dataset CCIDS menggunakan algoritma SVM dan *K-Nearest Neighbor* dilakukan untuk memberikan klasifikasi dari setiap data yang ada. Terdapat perbedaan performa dari algoritma SVM dan *K-Nearest Neighbor* dalam menghasilkan klasifikasi DDoS dan *Benign* yang diuji dengan menggunakan *confusion matrix.* Algoritma SVM dengan nilai *Accuracy* 90,75%, *Precision* 89,33%, *Recall* 91,38% dan F1- *Score* 90,27% sedangkan KNN dengan nilai *Accuracy* 95,50%, *Precision* 99,12%, *Recall* 93,84% dan F1- *Score* 96,39%.

Pada penelitian selanjutnya diharapkan sistem dapat menampilkan hasil ringkas klasifikasi yang dihasilkan untuk memudahkan pengguna dan dikemas dalam bentuk grafik.

Daftar Pustaka

[1] M. Fibrianda and A. Bhawiyuga, "Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM) ", Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 2, no. 9, p. 3112-3123, peb. 2018. ISSN 2548-964X, 2018.

[2] H. E. Wahanani, B. Nugroho and G. I. Prakoso, "Analisa Serangan Smurf Dan Ping Of Death Dengan Metode Support Vector Machine (SVM) ", Jurnal Teknologi Informasi dan Komunikasi 11 (2), 71-78, 2016.

[3] I. Riadi, R. Umar and F. D. Aini, "Analisis Perbandingan Detection Traffic Anomaly Dengan Metode Naive Bayes Dan Support Vector Machine (SVM)", ILKOM Jurnal Ilmiah 11(1):17, 2019.

[4] J. C. J. Sihombing, D. P. Kartikasari and A. Bhawiyuga, "Implementasi Sistem Deteksi dan Mitigasi Serangan Distributed Denial of

Service (DDoS) menggunakan SVM Classifier pada Arsitektur SoftwareDefined Network (SDN) ", Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN 2548, 964X, 2019.

[5] N. Hassni and F. R. Putri, "Klasifikasi Malicious Website Menggunakan Metode Algoritma K-Nearest Neighbor (KNN)", Annual Research Seminar (ARS) Vol 4, No 1, 2018.

[6] E. Listiana and M. A. Muslim, "Penerapan Adaboost Untuk Klasifikasi Support Vector Machine Guna Meningkatkan Akurasi Pada Diagnosa Chronic Kidney Disease", Prosiding SNATIF, 875-881, 2017.

[7] I. N. Youllia, A. N. Hermana and M. Kharisma, "Penerapan Algoritma K-Nearest Neighbor pada Game Pesawat untuk Pembelajaran Matematika Dasar", MIND Journal 4(2):132-142, 2019.

[8] Leidiyana Henny, "Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bemotor", PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic 1 (1), 65-76, 2013.

[9] H. Kurniawan, B. Setiyono and . R. R. Isnanto, "Aplikasi Penjawab Pesan Singkat Automatis Dengan Bahasa Python", Jurnal Jurusan Teknik Elektro Fakultas Teknik Undip, 3, 2011.

[10] F. Ridho, A. Yudhana and I. Riadi, "Analisis Forensik Router Untuk Mendeteksi Serangan Distributed Danial of Service (DDoS) Secara Real Time", Annual Research Seminar (ARS) Vol 2, No 1, 2, 2016.

[11] CICIDS, "UNB," 2018. diakses daring pada : https://www.unb.ca/cic/datasets/ids-2018.html