Akurasi dan Presisi Pengklasifikasian Abstrak *Paper* Informatika Menggunakan TF-IDF dan *Multiclass Support Vector Machine* (SVM)

FUTY ALPAZ[1], MILDA GUSTIANA H.[2], DEWI ROSMALA[3], NUR FITRIANTI F.[3]

Program Studi Informatika, Fakultas Teknologi Industri
Institut Teknologi Nasional Bandung
Email: futy.alpaz@mhs.itenas.ac.id [3]

ABSTRAK

Pencarian paper membutuhkan waktu yang cukup lama dikarenakan masih banyaknya paper penelitian yang belum diklasifikasikan berdasarkan topik yang dibutuhkan. Text mining merupakan proses untuk menemukan informasi penting pada sekumpulan abstrak sehingga dapat digunakan untuk melakukan klasifikasi paper. Tahapan penelitian dimulai dari pengumpulan abstrak paper, preprocessing, kemudian seleksi fitur menggunakan metode TF-IDF dan terakhir klasifikasi menggunakan metode Support Vector Machine (SVM) dengan teknik multiclass One Against All. Penelitian ini bertujuan untuk mengukur nilai akurasi, dan presisi pada sistem klasifikasi abstrak paper informatika menggunakan metode Support Vector Machine (SVM) dengan teknik multiclass One Against All. Hasil pengujian menggunakan metode split data memperoleh nilai akurasi dan presisi sebesar 96% dan 97%, sementara pengujian data perkelas memperoleh nilai rata-rata akurasi dan presisi secara beruntun sebesar 98% and 96%.

***Kata kunci**: paper, TF-IDF, svm, multiclass, akurasi, presisi*

ABSTRACT

Paper trackings require a quite time because there are many paper that has not been classified based on topics needed. Text mining is a process to get information that is essential in a set of abstracts so it could be used to classify paper. Research stages are began by collect paper abstract, preprocessing, then feature selection using TF-IDF method and the last stage is classification using Support Vector Machine (SVM) method with multiclass One Against All technique. This study aims to measure the system classification accuracy and precision in the informatic abstract paper by using Support Vector Machine (SVM) method with multiclass One Against All technique. According the results of research using the split data method obtained accuracy and precision values of 96% and 97% while testing the class data obtains successive accuracy and precision values of 98% and 93%.

***Keywords**: paper, TF-IDF, svm, multiclass, accuracy, precision*

1. PENDAHULUAN

1.1. Latar Belakang

Berdasarkan surat edaran Dirjen Dikti Kementrian Riset, Teknologi, dan Pendidikan tinggi (Kemenristekdikti) nomor 152/E/T/2012 tahun 2012 mengenai publikasi karya ilmiah mewajibkan lulusan program sarjana untuk menghasilkan karya ilmiah yang terbit pada jurnal ilmiah, untuk lulusan magister diharuskan menghasilkan karya ilmiah yang terbit di jurnal ilmiah nasional, dan untuk lulusan doktor diharuskan menghasilkan karya ilmiah yang sudah diterima untuk terbit di jurnal internasional. Peraturan tersebut dibuat karena jumlah karya ilmiah pada saat itu dari perguruan tinggi Indonesia masih rendah.

Berdasarkan buku yang ditulis oleh (Greenhalgh, 2014) pertumbuhan *paper* begitu cepat di internet serta banyak *paper* yang belum diklasifikasi sesuai dengan topik penelitian sehingga pencarian *paper* menjadi tidak efisien karena sulit untuk menemukan *paper* yang relevan dengan topik penelitian serta membutuhkan waktu yang lama dan tingkat ketelitian yang tinggi dalam pencariannya. Untuk itu dibutuhkan suatu teknik dalam bidang pemrosesan teks untuk menentukan klasifikasi topik tertentu secara otomatis.

Menurut (Ernawati, 2019) Klasifikasi teks dapat dilakukan dengan metode *Naïve bayes Classifier* (NBC) dan *Support Vector Machine* (SVM). (Rizaldi, Khairi, & Mustakim, 2021) pernah melakukan kalsifikasi opini public terhadap provider di Indonesia menggunakan metode *Naïve bayes Classifier* (NBC) dan mendapatkan nilai akurasi terbaik sebesar 68,85%. Berdasarkan penelitian yang telah dilakukan oleh (Maulina & Sagara, 2018) metode *Support Vector Machine* mendapat tingkat akurasi yang cukup tinggi dalam mengklasifikasi teks artikel *hoax* dan tidak *hoax* yaitu sebesar 95,8333%. Metode SVM dipilih karena menurut (Patle & Chouhan, 2013) SVM merupakan salah satu metode klasifikasi dengan performa terbaik selain itu SVM dapat melakukan klasifikasi pada data yang kompleks.

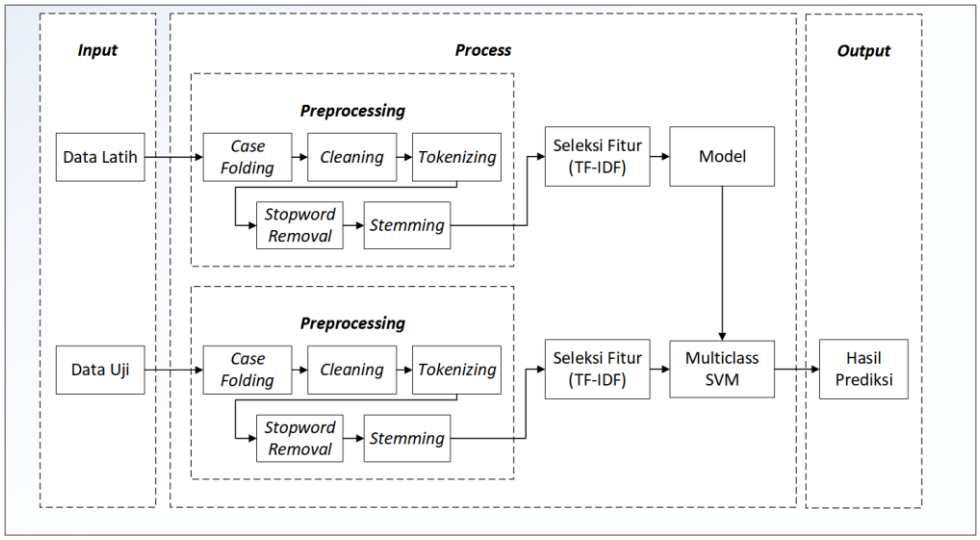
Dalam melakukan kalsifikasi teks dibutuhkan metode seleksi fitur untuk memberi nilai bobot pada setiap kata (*term*) dengan pendekatan *Term Frequency Inverse Document Frequency* (TF-IDF) dimana menurut (Melita, Amrizal, Suseno, & Dirjam, 2018) TF-IDF merupakan metode yang terkenal efisien, mudah dan memiliki hasil yang akurat dalam menghitung bobot setiap kata.

Metode SVM dapat digunakan untuk melakukan klasifikasi lebih dari dua kelas (*non biner*) dengan pendekatan *multiclass* SVM salah satunya metode *One Against All.* Berdasarkan penelitian yang telah dilakukan oleh (Mustakim, Santoso, & Zahra, 2017) hasil akurasi metode *One Against All* memiliki keunggulan dibandingkan metode *One Against One* dan pada penelitian yang dilakukan oleh (Fritriana & Sibaroni S.T, M.T, 2020) metode *One Against All* berhasil melakukan kalsifikasi tiga kelas pada data *tweet* dengan perolehan nilai akurasi yang cukup besar yaitu 80,59%.

Pada penelitian ini akan merancang sistem klasifikasi abstrak *paper* informatika ke dalam kelas *Artificial Intelligent* (AI), jaringan, dan Rekayasa Perangkat Lunak (RPL) menggunakan metode *multiclass Support Vector Machine* (SVM) dengan pendekatan *One Against All* dan metode seleksi fitur *Term Frequency Inverse Document Frequency* (TF-IDF). *Input* an dari sistem yaitu abstrak *paper* informatika dan hasil akhir berupa kinerja sistem kalsifikasi teks. Tujuan dari penelitian ini untuk mengetahui kinerja metode *multiclass Support Vector Machine (SVM)* dalam mengklasifikas abstrak *paper* informatika berdasarkan topiknya dengan mengetahui tingkat akurasi dan presisi yang didapatkan.

2. METODE PENELITIAN

Pada penelitian ini, data masukkan untuk sistem yaitu berupa teks abstrak *paper* informatika kemudian abstrak diolah dengan tahapan *preprocessing,* seleksi fitur menggunakan TF-IDF, dan klasifikasi menggunakan *multiclass Support Vector Machine* (SVM) seperti Gambar 1 berikut.



Gambar 1. *Block Diagram System*

2.1 *Preprocessing*

Preprocessing merupakan tahapan pertama dalam proses *text mining* yang berfungsi untuk merubah data tidak terstruktur menjadi data terstruktur. Menurut (Jumeilah, 2017) keuntungan dari proses *preprocessing* yaitu data yang akan digunakan jadi terstruktur, bersih dari *noise*, dan dimensi data jadi lebih kecil. Proses *preprocessing* pada *text mining* terdiri dari lima tahapan yaitu *case folding, cleaning, tokenizing, stopword removal,* dan *stemming*.

2.1.1 *Casefolding*

Case folding merupakan proses mengubah semua huruf kapital pada kalimat menjadi huruf kecil (*lowercase*) (Pravina, Cholissodin, & Adikara, 2019).

2.1.2 *Cleaning*

Cleaning merupakan proses menghilangkan simbol %, @, &, *, Tanda baca titik (.), koma (,), tanda tanya (?), titik dua (:), kutip ("), angka, URL, dan *hashtag,* sehingga karakter yang diproses hanya huruf 'a' hingga 'z'.

2.1.3 *Tokenizing*

Tokenizing merupakan proses memecah kalimat menjadi kata yang berdiri sendiri dalam bentuk token (*term*) sesuai dengan kata-kata yang menyusun kalimat tersebut. Pada tahap *tokenizing* dilakukan pemotongan kata pada *white space* atau spasi (Jumeilah, 2017).

2.1.4 *Stopword Removal*

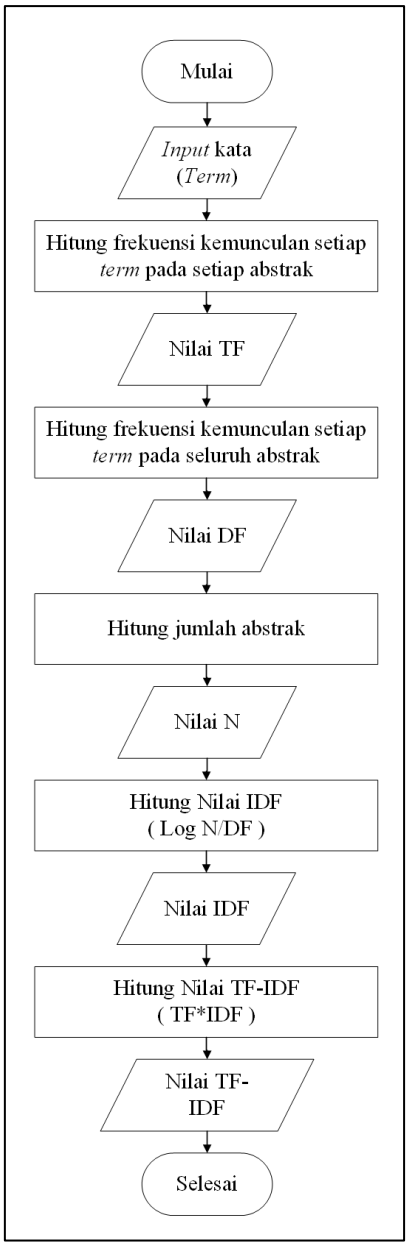
Stopword Removal merupakan proses menghilangkan kata yang tidak memberikan informasi penting seperti kata penghubung "yang", "dan", "atau", "oleh", "pada", dan sebagainya

2.1.5 *Stemming*

Stemming merupakan proses merubah kata tidak baku menjadi kata baku atau kata dasar dengan menghilangkan awalan (*prefixes*), akhiran (*suffixes*), sisipan (*infixes*) dan kombinasi awalan dan akhiran (*confixes*) (Maulina & Sagara, 2018).

2.2 Seleksi Fitur TF-IDF

Menurut (Ahuja, Chug, Kohli, Gupta, & Ahuja, 2019) *Term Frequency Inverse Document Frequency* (TF-IDF) merupakan metode yang baik digunakan untuk mengubah informasi tekstual menjadi *Vector Space Model* (VSM). TF-IDF menunjukkan seberapa relevan dan pentingnya sebuah kata di dalam dokumen. Adapun alur kerja TF-IDF ditunjukkan pada Gambar 2.



Gambar 2. *Flowchart* TF-IDF

Berdasarkan Gambar 2 pembobotan didapat dari perhitungan *Term Frequency* (TF), dan *Inverse Document Frequency* (IDF). *Term Frequency* (TF) yaitu frekuensi kemunculan sebuah kata (*term*) pada sebuah abstrak. Semakin sering sebuah kata (*term*) muncul pada suatu

abstrak, maka bobot kata (*term*) semakin besar dan kata (*term*) tersebut dianggap sebagai kata (*term*) yang merepresentasikan abstrak tersebut. DF merupakan banyaknya abstrak yang mengandung suatu *term* kata. *Inverse Document Frequency* (IDF) merupakan kemunculan sebuah kata (*term*) pada kumpulan abstrak. Semakin sedikit kemunculan sebuah kata (*term*) pada kumpulan abstrak maka kata (*term*) tersebut dianggap penting. Perhitungan IDF di representasikan pada persamaan (1).

$$Idf = log\left(\frac{N}{df_i}\right) \quad (1)$$

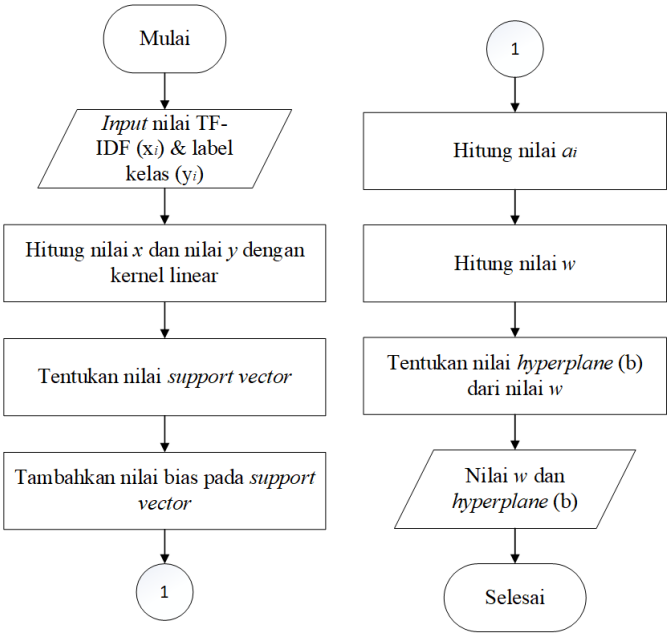
Bobot setiap kata (*term*) pada abstrak didapat dari hasil perkalian TF dan IDF yang direpresentasikan pada persamaan (2). Kata (*term*) yang sering muncul pada suatu abstrak tetapi jarang muncul pada kumpulan abstrak akan memiliki nilai bobot yang tinggi.

$$TF - IDF_{i,d} = tf_{d,i}\ x\ Idf \quad (2)$$

Keterangan:
$TF - IDF_{i,d}$ = Bobot abstrak ke-d terhadap kata ke- i
$Tf_{d,i}$ = Banyaknya kata yang terdapat pada sebuah abstrak
Idf = *Inverse document frequency*
N = Jumlah keseluruhan abstrak
df_i = jumlah abstrak yang mengandung kata – i

2.3 *Multiclass Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah metode *machine learning* yang pertama kali diperkenalkan oleh Vapnik. *Support Vector Machine* (SVM) digunakan untuk menganalisis data dan mengenali pola seperti proses klasifikasi, prediksi dan analisis regresi. Pada dasarnya SVM adalah teori pembelajaran statistik dan bertujuan untuk mengoptimalkan garis *hyperplane* yang berfungsi sebagai batas pemisah kelas (Vapnik, 1995). *Flowchart* perhitungan SVM ditunjukkan pada Gambar 3.



Gambar 3. *Flowchart* SVM

Berdasarkan Gambar 3 *input an* pada sistem yaitu rnilai TF-IDF setiap kata yang sudah dilakukan proses *preprocessing* dinotasikan sebagai x_i dan juga label kelas. dinotasikan dengan y_i. Kemudian hitung nilai x dan y untuk mencari nilai *support vector* menggunakan kernel linear. Nilai x dapat dicari menggunakan persamaan 3 berikut.

$$\sum_{i=1,j=1}^{n} x_i x_j = {x_i}^T x_j \ (i,j = 1, \dots n) \quad (3)$$

Sementara untuk mendapatkan nilai y menggunakan persamaan 4 berikut.

$$\sum_{i=1,j=1}^{n} y_i y_j = {y_i}^T y_j \ (i,j = 1, \dots n) \quad (4)$$

Setelah nilai x dan y ditemukan tentukan nilai *support vector* ($\boldsymbol{\varphi}$) dengan memasukkan nilai ke persamaaan 5 berikut:

$$\varphi \begin{bmatrix} x \\ y \end{bmatrix} = \begin{cases} \sqrt{{x_n}^2 + {y_n}^2} > 2, maka & \begin{bmatrix} \sqrt{{x_n}^2 + {y_n}^2} - x + |x - y| \\ \sqrt{{x_n}^2 + {y_n}^2} - y + |x - y| \end{bmatrix} \\ \sqrt{{x_n}^2 + {y_n}^2} \leq 2, maka & \begin{bmatrix} x \\ y \end{bmatrix} \end{cases} \quad (5)$$

Tambahkan nilai bias 1 pada setiap nilai *support vector* untuk mendapatkan *hyperplane* jarak tegak lurus yang optimal dan mendapatkan nilai *hyperplane* (*b*). Hitung nilai a_i menggunakan persamaan 6 berikut:

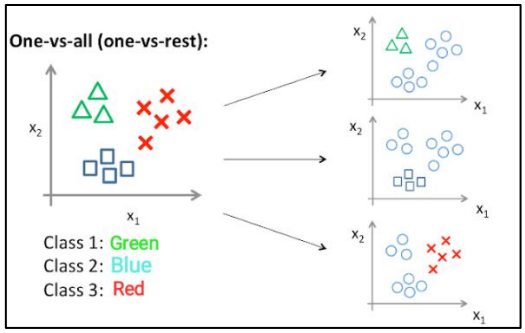
$$\sum_{i=1,j=1}^{n} \alpha_i {T_i}^T T_j = y \quad (6)$$

Setelah nilai ai di dapat selanjutnya hitung nilai *w* menggunakan persamaan 7 berikut:

$$w = \sum_{i=1}^{n} \alpha_i T_i \ where \ \alpha_i \geq 0 \quad (7)$$

Nilai *w* yang telah dihitung pada persamaan 7 sudah mengandung nilai *hyperplane* (*b*) karena telah ditambahkan nilai bias.

Seiring perkembangannya SVM mampu mengklasifikasikan banyak kelas menggunakan metode *multiclass*. Salah satu metode *multiclass* adalah metode *One Against All*. Konsep dari metode *One Against All* yaitu membandingkan satu kelas dengan semua kelas lainnya yang dianggap sebagai satu kesatuan sehingga akan terbentuk lebih dari satu *hyperplane*. Klasifikasi *One Against All* menghasilkan *n hyperplane* dimana *n* adalah jumlah kelas. Dalam pendekatan klasifikasi *One Against All*, semua data dalam satu kelas bernilai 1 dan kelas lainnya bernilai -1. Proses ini diulang untuk semua kelas (Vapnik, 1995). Gambaran klasifikasi metode *One Against all* ditunjukkan pada Gambar 4.



Gambar 4. *Multiclass One Against All*
Sumber: cc.gatech.edu

Pada metode *multiclass One Against All* akan terbentuk *n* buah model SVM biner dimana *n* adalah jumlah kelas sehingga jumlah *hyperplane* yang terbentuk tergantung dari jumlah kelas tersebut. Untuk menentukan kelas hasil klasifikasi data uji menggunakan persamaan () berikut:

$$f(x) = \underset{n = 1 \ldots i}{arg\ max} \left(\left(w^{(n)} \right)^T .\ \varphi \begin{bmatrix} p_i \\ q_i \end{bmatrix} + b^{(n)} \right) \qquad (8)$$

Persamaan (8) digunakan pada proses klasifikasi setelah mendapatkan nilai *support vector* dari data uji. $\left(w^{(n)}\right)^T$ dan $b^{(n)}$ adalah nilai bobot dan *hyperplane* setiap kelas pada proses *training* sementara $\varphi \begin{bmatrix} p_i \\ q_i \end{bmatrix}$ merupakan nilai *support vector* abstrak uji yang akan diklasifikasi. Hasil klasifikasi ditentukan berdasarkan nilai tertinggi dari hasil perhitungan semua *hyperplane* dan *support vector* abstrak uji.

2.4 Pengujian Sistem

Pengujian yang dilakukan pada sistem yaitu menguji kinerja sistem dimana parameter yang diuji yaitu nilai akurasi (*accuracy*) dan nilai presisi (*precision).* Pengujian kinerja sistem menggunakan *confusion matrix* untuk membandingkan hasil prediksi sistem dengan hasil klasifikasi sebenarnya. Tabel *confusion matrix* ditunjukkan pada Tabel dimana TP (*True Positive)* adalah data yang bernilai positif dan diprediksi benar positif, FP (*False Positive)* yaitu data yang bernilai negatif dan diprediksi positif, TN (*True Negative)* yaitu data yang bernilai negatif dan diprediksi benar sebagai negatif, FN *(False Negative)* yaitu data yang bernilai positif dan diprediksi negatif.

Tabel 1. Confusion Matrix

		Predicted Values	
		Positive	***Negative***
Actual Values	***Positive***	*True Positive* (TP)	*False Negative* (FN)
	Negative	*False Positive* (FP)	*True Negative* (TN)

Nilai akurasi dapat dihitung menggunakan Persamaan (9).

$$Accuracy = {}^{(\mathrm{TP}\ +\ \mathrm{TN})}\Big/_{(\mathrm{TP}\ +\ \mathrm{TN}\ +\ \mathrm{FP}\ +\ \mathrm{FN})} \qquad (9)$$

Nilai presisi (*precision)* dapat dihitung menggunakan Persamaan (10).

$$Precision = {}^{\mathrm{TP}}\Big/_{(\mathrm{TP}\ +\ \mathrm{FP})} \qquad (10)$$

3. HASIL PENGUJIAN

Pengujian data dilakukan menggunakan 2 cara yaitu metode *split data* dan menggunakan data uji baru.

3.1 Pengujian Menggunakan Metode *Split Data*

Dataset yang digunakan dalam sistem berjumlah 120 data abstrak yang terdiri dari 40 abstrak *Artificial Intelligent* (AI), 40 abstrak Jaringan komputer, dan 40 abstrak Rekayasa Perangkat Lunak (RPL). *Dataset* di *split* dengan rasio 80:20 dimana 80% sebagai data latih dan 20% sebagai data uji. Data uji dipilih secara acak oleh program. 2 menunjukkan perbandingan nilai klasifikasi sebenarnya (*actual*) dengan hasil klasifikasi sistem (*predicted*).

Tabel 2. Perbandingan Hasil Klasifikasi Metode *Split Data*

No.	Data	Aktual	Prediksi
1.	Data ke-48	RPL	RPL
2.	Data ke-94	Jaringan	Jaringan
3.	Data ke-95	Jaringan	Jaringan
4.	Data ke-8	AI	AI
5.	Data ke-97	Jaringan	Jaringan
6.	Data ke-22	Jaringan	Jaringan
7.	Data ke-7	AI	AI
8.	Data ke-10	AI	AI
9.	Data ke-45	RPL	RPL
10.	Data ke-89	Jaringan	Jaringan
11	Data ke-33	Jaringan	Jaringan
12.	Data ke-50	RPL	RPL
13.	Data ke-2	AI	AI
14.	Data ke-60	AI	AI
15.	Data ke-119	RPL	RPL
16.	Data ke-74	AI	AI
17.	Data ke-30	Jaringan	AI
18.	Data ke-43	RPL	RPL
19.	Data ke-111	RPL	RPL
20.	Data ke-76	AI	AI
21.	Data ke-63	AI	AI
22.	Data ke-59	RPL	RPL
23.	Data ke-16	AI	AI
24.	Data ke-24	Jaringan	Jaringan

Perbandingan data hasil klasifikasi dengan kelas sebenarnya ditunjukkan pada Tabel 2. Dari 24 data uji pada terdapat satu hasil klasifikasi yang tidak sesuai dengan kelas sebenarnya yaitu data ke-30 dimana kelas sebenarnya adalah Jaringan tetapi hasil klasifikasi pada sistem yaitu AI. Hasil klasifikasi tersebut digunakan untuk mengukur kinerja sistem dengan metode *confusion matrix* ditunjukkan pada Tabel 3.

Tabel 3. *Confusion Matrix* Metode *Split Data*

		Predicted Value		
		AI	Jaringan	RPL
Actual Value	*AI*	9	0	0
	Jaringan	1	7	0
	RPL	0	0	7

Dari tabel *confusion matrix* pada Tabel 3 dapat dihitung nilai akurasi, presisi (*precision*), *recall*, dan *F1-Score*. Nilai akurasi dihitung menggunakan persamaan (9) dan di dapat nilai akurasi sebesar 96%.

$$Accuracy = \frac{\text{TPA}+TPJ+TPR}{\text{Total data}} \text{ x } 100\%$$

$$Accuracy = \frac{9+7+7}{12} \text{ x } 100\%$$

$$Accuracy = \frac{23}{24} \text{ x } 100\%$$

$$Accuracy = 96\%$$

Nilai presisi (*precision)* dihitung berdasarkan persamaan (10) yaitu:

$$Precision = \frac{\text{TP}}{\text{TP}+\text{FP}} \text{ x } 100\%$$

Tabel 4 Nilai *Precision* Metode *Split Data*

Precision AI	*Precision* Jaringan	*Precision* RPL
$P(A) = \frac{TA}{\text{TA}+\text{FA}}$ $P(A) = \frac{9}{9+1}$ P(A) = 0,90	$P(J) = \frac{\text{TJ}}{\text{TJ}+\text{FJ}}$ $P(J) = \frac{7}{7+0}$ P(J) = 1	$P(R) = \frac{\text{TR}}{\text{TR}+\text{FR}}$ $P(R) = \frac{7}{7+0}$ P(R) = 1

Dari perhitungan Tabel 4 diketahui bahwa nilai *precision* kelas AI adalah 0,90, nilai *precision* jaringan adalah 1 dan nilai *precision* RPL adalah 1. Sehingga total nilai *precision* sebesar 97%.

$$Total\ Precision = \frac{\text{P(A)}+\text{P(J)}+\text{P(R)}}{3} \text{ x } 100\%$$

$$Total\ Precision = \frac{0,90+1+1}{3} \text{ x } 100\%$$

$$Total\ Precision = \frac{2,90}{3} \text{ x } 100\%$$

$$Total\ Precision = 97\%$$

3.2 Pengujian Data Perkelas

Pengujian data perkelas merupakan pengujian menggunakan 24 abstrak uji baru diluar dari *dataset*dengan masing-masing kelas terdiri dari 8 abstrak uji.

a. Pengujian berdasarkan Kelas AI

Tabel 5. Pengujian Kelas AI

No.	Data	Aktual	Klasifikasi	Akurasi	Presisi
1.	AI 1	AI	AI	0,96	1
2.	AI 2	AI	AI		
3.	AI 3	AI	RPL		
4.	AI 4	AI	AI		
5.	AI 5	AI	AI		
6.	AI 6	AI	AI		
7.	AI 7	AI	AI		
8.	AI 8	AI	AI		

Dari hasil pengujian kelas AI yang ditunjukkan pada Tabel 5, didapat satu hasil klasifikasi yang tidak sesuai yaitu data AI 3 dimana kelas sebenarnya yaitu AI tetapi hasil klasifikasi pada sistem yaitu RPL, sehingga nilai akurasi dan presisi yang didapat sebesar 0,96 dan 1.

b. Pengujian berdasarkan Kelas Jaringan

Tabel 6. Pengujian Kelas Jaringan

No.	Data	Aktual	Klasifikasi	Akurasi	Presisi
1.	Jaringan 1	Jaringan	Jaringan	1	1
2.	Jaringan 2	Jaringan	Jaringan		
3.	Jaringan 3	Jaringan	Jaringan		
4.	Jaringan 4	Jaringan	Jaringan		
5.	Jaringan 5	Jaringan	Jaringan		
6.	Jaringan 6	Jaringan	Jaringan		
7.	Jaringan 7	Jaringan	Jaringan		
8.	Jaringan 8	Jaringan	Jaringan		

Dari hasil pengujian yang ditunjukkan pada Tabel , didapat nilai akurasi dan presisi masing-masing sebesar 1.

c. Pengujian berdasarkan Kelas RPL

Tabel 7. Pengujian Kelas RPL

No.	Data	Aktual	Prediksi	Akurasi	Presisi
1.	RPL 1	RPL	RPL	1	0,89
2.	RPL 2	RPL	RPL		
3.	RPL 3	RPL	RPL		
4.	RPL 4	RPL	RPL		
5.	RPL 5	RPL	RPL		
6.	RPL 6	RPL	RPL		
7.	RPL 7	RPL	RPL		
8.	RPL 8	RPL	RPL		

Dari hasil pengujian yang ditunjukkan pada Tabel , didapat nilai akurasi dan presisi kelas RPL sebesar 1 dan 0,89.
Dari 24 data yang dilakukan pengujian di dapat nilai rata-rata akurasi dan presisi. Rata-rata nilai akurasi yang di dapat yaitu 98%.

$$Accuracy = \frac{0,96+1+1}{3} \text{ x } 100\%$$
$$Accuracy = \frac{2,96}{3} \text{ x } 100\%$$
$$Accuracy = 98\%$$

Rata-rata nilai presisi (*precision)* dihitung berdasarkan persamaan (3), yaitu:

$$Total\ Precision = \frac{1+1+0,89}{3} \text{ x } 100\%$$
$$Total\ Precision = \frac{2,89}{3} \text{ x } 100\%$$
$$Total\ Precision = 96\%$$

5. KESIMPULAN

Penelitian ini bertujuan untuk mengukur kinerja sistem yang terdiri dari nilai akurasi, presisi, *recall*, dan *F-1-Score* menggunakan metode seleksi fitur TF-IDF dan *Multiclass Support Vector Machine* (SVM) *One Against All.* Dari penelitian yang telah dilakukan, didapatkan kesimpulan bahwa sistem mampu melakukan klasifikasi *paper* penelitian mahasiswa jurusan informatika dengan rangkaian tahapan *preprocessing,* seleksi fitur menggunakan TF-IDF dan *Multiclass Support Vector Machine (SVM) One Against All* untuk klasifikasi teks *paper*. Kinerja sistem dari pengujian menggunakan metode *split data* dengan rasio 80:20 dari total *dataset* 120 sehingga data yang digunakan sebagai pengujian berjumlah 24 data dan mendapatkan nilai akurasi sebesar 96% dan rata-rata nilai presisi sebesar 97%. Kinerja sistem dari pengujian kepada kelas AI didapat nilai akurasi sebesar 96% dan nilai presisi sebesar 100%. Kinerja sistem dari pengujian kepada kelas jaringan didapat nilai akurasi sebesar 100% dan nilai presisi sebesar 100%. Kinerja sistem dari pengujian kepada kelas RPL didapat nilai akurasi sebesar 100% dan nilai presisi sebesar 94%. Sehingga didapat nilai rata akurasi dan presisi sebesar 98% dan 96%.

DAFTAR RUJUKAN

Ernawati, I. (2019). Naive Bayes Classifier Dan Support Vector Machine Sebagai Alternatif Solusi Untuk Text Mining. *Jurnal Teknologi Informasi dan Pendidikan, 12*.

Fritriana, D. N., & Sibaroni S.T, M.T, Y. (2020). Klasifikasi Data Tweet dengan Menggunakan Metode Klasifikasi Multi-Class Support Vector Machine (SVM). *e-Proceeding of Engineering*, *7*, p. 8493.

Greenhalgh, t. (2014). *How to read a paper : the basics of evidence-based medicine.* London: BMJI Books.

Jumeilah, F. S. (2017). Penerapan Support Vector Machine (SVM) untuk Pengkategorian Penelitian. *Jurnnal Resti (Rekayasa Sistem dan Teknologi Informasi)*, 19 - 25.

Maulina, D., & Sagara, R. (2018, Juni). Klasifikasi Artikel Hoax Menggunakan Support Vector machine Linear Dengann Pembobtan Term Frequency-Inverse Document Frequency. *Jurnal Mantik Penusa, 2*.

Melita, R., Amrizal, V., Suseno, H. B., & Dirjam, T. (2018, Oktober). Penerapan Metode Term Frequency Inverse Document Frequency (TF-IDF) dan Consine Similiarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus; Syarah Umdatil Ahkam). *Journal Teknik Informatika, 11*.

Mustakim, A., Santoso, I., & Zahra, A. A. (2017, September). Pengenalan Ekspresi Wajah Manusia Menggunakan Tapis Gabor 2-D Dan Support Vector Machine (SVM). *Transient, 6*.

Patle, A., & Chouhan, D. S. (2013). SVM Kernel Function for Cladssification. *2013 International Conference on Advance in Technology and Engineering, ICATE 2013.*

Pravina, A. M., Cholissodin, I., & Adikara, P. P. (2019). Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritma Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2789-2797.

Ramya, M., & Pinakas, J. A. (2014). Different Type of feature Selection for Text Classification.

Rizaldi, S. T., Khairi, A. A., & Mustakim. (2021). Text Mining Classification Opini Publik Terhadap Provider di Indonesia. *Seminar Nasional Teknologi Informasi, Komunikasi, dan Industri.* Pekanbaru.

Vapnik, V. N. (1995). The Nature of Statistical Learning Theory.