

## PARTICLE SWARM OPTIMIZATION IN TWIN SUPPORT VECTOR MACHINE TO CLASSIFY FAKE NEWS

Novita Enjelia Hutapea<sup>1</sup>, Merika Manurung<sup>2</sup>, Tiara Octavia Situmorang<sup>3</sup>, Junita Amalia<sup>4</sup>

<sup>1,2,3,4</sup>Program Studi Sistem Informasi, Institut Teknologi Del

iss18009@students.del.ac.id<sup>1</sup>, iss18041@students.del.ac.id<sup>2</sup>, iss18046@students.del.ac.id<sup>3</sup>,  
junita.amalia@del.ac.id<sup>4</sup>

### Abstract

Basically online media give freedom to the user to speak with other users. Because of this freedom, some users misuse online media, namely by spreading information that cannot be justified or what is called fake news. To avoid the spread of fake news, users must know whether the news is fake news or not. In this study, the classification of news will be carried out. This research use Twin Support Vector Machine (TWSVM) as classification method. However TWSVM has a disadvantage viz difficult to determine optimal parameter and that's why an optimization is needed to find the parameter with Particle Swarm Optimization (PSO). There are five main parameters used in this research viz  $C_1$  and  $C_2$ ,  $w$ , number of particle and iteration number. Model performance optimization will be seen from impact of the parameters. Model performance will be measured with evaluation matrix accuracy, precision, recall, and F1-Score. The result of this research show that Particle Swarm Optimization can improve the accuracy, precision, and f-1 score of classification model Twin Support Vector Machine.

**Keywords:** Fake news, Twin Support Vector Machine, Particle Swarm Optimization

### Abstrak

Media online pada dasarnya memberikan kebebasan kepada pengguna untuk berbicara dengan pengguna lainnya. Karena kebebasan ini, beberapa pengguna menyalahgunakan media online yaitu dengan menyebarkan informasi yang tidak dapat dipertanggungjawabkan kebenarannya atau yang disebut dengan fake news. Untuk menghindari penyebaran fake news tersebut, pengguna harus mengetahui apakah berita tersebut fake news atau tidak. Pada penelitian ini akan dilakukan pengklasifikasian berita. Penelitian ini menggunakan Twin Support Vector Machine (TWSVM) sebagai metode pengklasifikasian. Namun TWSVM memiliki kelemahan yaitu sulit untuk menentukan parameter yang optimal sehingga dibutuhkan sebuah optimasi untuk membantu menemukan parameter tersebut yaitu dengan menggunakan Particle Swarm Optimization (PSO). Ada 5 parameter utama yang digunakan dalam penelitian ini yaitu  $C_1$  dan  $C_2$ ,  $w$ , jumlah partikel serta jumlah iterasi. Optimasi kinerja model akan dilihat dari pengaruh dari kelima parameter tersebut. Performa model yang dibangun diukur dengan menggunakan metrik evaluasi akurasi, presisi, recall, dan f1- score. Hasil penelitian menunjukkan bahwa Particle Swarm Optimization dapat meningkatkan akurasi sebesar 1.0-3.0%, presisi sebesar 4.0-11.0%, dan f-1 skor sebesar 0.5-1.0% pada model klasifikasi Twin Support Vector Machine.

**Kata kunci:** Fake News, Twin Support Vector Machine, Particle Swarm Optimization.

### 1. Pendahuluan

Kemajuan dan perkembangan teknologi yang berkembang pesat mengikuti perkembangan zaman adalah fenomena nyata yang harus dihadapi. Banyak informasi bermanfaat yang bisa kita dapat dari membaca berita di media online yang dimana penyebaran informasi ataupun berita ini dilakukan oleh seluruh pengguna media online tanpa melalui

pemeriksaan apapun. Media online pada dasarnya memberikan kebebasan kepada penggunanya untuk berkomunikasi dengan pengguna lainnya. Karena kebebasan ini, beberapa pengguna menyalahgunakan media online tersebut sebagai wadah penyebaran informasi yang tidak dapat dipertanggungjawabkan kebenarannya atau yang disebut dengan fake news atau hoax.

Pada penelitian ini peneliti akan mengklasifikasi *fake news* sehingga dapat membantu menandai *fake news* yang ada pada media *online* untuk mengurangi dampak negatif penggunaan media *online*. Berita yang benar adalah berita yang sudah terbukti kebenarannya. Sedangkan *fake news* atau *hoax* adalah informasi yang tidak valid namun dibuat seolah-olah berita tersebut benar sehingga dapat menyebabkan orang yang membaca berita tersebut merasa tidak nyaman dan tidak aman [1].

Penelitian tentang *Fake news* sebelumnya telah dilakukan oleh Rian Pebi pada tahun 2020 yang berjudul “Analisis Performansi Algoritma *Naive Bayes* dan *Support Vector Machine* untuk Deteksi Berita *Hoax* Berbahasa Inggris”. Pada penelitian ini, peneliti menggunakan dua model klasifikasi yaitu *Naive Bayes* dan *Support Vector Machine*. Peneliti menggunakan *dataset* berbahasa Inggris sebanyak 1.381 *record* yang terdiri dari berita *hoax* dan *non hoax*. Sebelumnya *dataset* melalui *preprocessing*, kemudian data hasil *preprocessing* tersebut dilanjutkan dengan proses perhitungan bobot setiap kata. Hasil dari perhitungan bobot kata akan diproses ke model klasifikasi *Naive Bayes* dan *Support Vector Machine*. Pada penelitian tersebut *dataset* dibagi menjadi 70% *Train set* dan 30% *Test set* dan menghasilkan akurasi sebesar 96.21% untuk *Naive Bayes* dan 97.22% untuk *Support Vector Machine*. Lalu untuk pembagian *dataset* 60% *Train set* dan 40% *Test set* menghasilkan akurasi sebesar 94.50% untuk *Naive Bayes* dan 96.02% untuk *Support Vector Machine*. Perbedaan hasil akurasi dari kedua model ini disebabkan oleh perbedaan cara kerjanya. Adapun cara kerja pada model klasifikasi yang menggunakan *Naive Bayes* yakni melibatkan metode probabilitas ketidakterkaitan kelas (*independent*). Ketidakterkaitan antar kelas tersebut turut menyebabkan pengurangan nilai akurasi. Model klasifikasi dengan algoritma *Support Vector Machine* menggunakan ruang berdimensi dalam mengelompokkan kelas sehingga didapat hubungan antar kelas yang saling terkait. Dengan demikian nilai akurasi dapat menjadi maksimal [2].

Pada penelitian yang dilakukan oleh Yaozhong Yang, Guangyue Zhou, Kewen Li, and Fulai Li pada tahun 2020 yang berjudul “A novel prediction method for favorable reservoir of oil field based on grey wolf optimizer and Twin Support Vector Machine” menyatakan bahwa metode *Twin Support Vector Machine* (TWSVM) memiliki kelemahan yaitu sulit dalam menemukan parameter sehingga dibutuhkan sebuah optimasi untuk membantu menemukan parameter tersebut [3]. Penelitian lain, dilakukan oleh Kusum Kumari Bharti dan Shivanjali Pandey tahun 2021, berjudul “*Fake account detection in twitter using logistic*

*regression with Particle Swarm Optimization*”, *regresi logistik* yang terintegrasi dengan *Particle Swarm Optimization* (PSO) untuk parameter optimum menghasilkan peningkatan akurasi model dari 93.3% menjadi 96.2% [4].

Sani Muhamad Isa, Tri Fennia Lesmana, and Nico Surantha tahun 2021 melakukan penelitian yang berjudul “*Sleep stage classification using Extreme Learning Machine and Particle Swarm Optimization for healthcare big data*”. Integrasi dari *Particle Swarm Optimization* (PSO) dan *Extreme Learning Machine* (ELM) digunakan untuk memilih fitur dan menentukan jumlah node tersembunyi. Hasilnya adalah jika metode SVM digabung dengan algoritma PSO akan menghasilkan akurasi lebih tinggi dibandingkan dengan metode ELM yang digabung dengan PSO. Hasil uji akurasi untuk gabungan ELM dan PSO masing-masing sebesar 62.66%, 71.52%, 76.77%, dan 82.1% untuk kelas 6, 4, 3, dan 2 kelas. Sedangkan hasil uji akurasi untuk gabungan SVM dan PSO masing-masing sebesar 68.72%, 77.25%, 81.8%, dan 85.93% untuk kelas 6, 4, 3, dan 2 kelas. Dapat disimpulkan akurasi klasifikasi dapat ditingkatkan dengan menerapkan algoritma PSO untuk pemilihan fitur [5].

Pada penelitian ini, peneliti melakukan klasifikasi terhadap *fake news* menggunakan metode *Twin Support Vector Machine* (TWSVM) karena TWSVM merupakan suatu klasifikasi biner dimana dua *hyperplane non paralel* akan digunakan dalam memisahkan dua kelas yaitu *fake news* dan *real news*. Tetapi karena metode *Twin Support Vector Machine* (TWSVM) memiliki kelemahan yaitu sulit dalam menemukan parameter yang optimum sehingga dibutuhkan sebuah optimasi untuk membantu menemukan parameter tersebut. Berlandaskan beberapa penelitian yang telah disebutkan sebelumnya, peneliti akan menggunakan *Particle Swarm Optimization* (PSO) karena PSO dapat meningkatkan hasil akurasi klasifikasi.

Adapun data yang digunakan dalam penelitian ini bersumber dari kaggle. Kaggle merupakan sebuah website yang menyediakan *dataset*, *dataset-dataset* ini bisa memudahkan peneliti saat membuat model proyek yang ingin dikembangkan. Data yang digunakan merupakan data berita yang sudah diklasifikasikan menjadi 2 kategori yaitu *real* dan *fake*. *Dataset* yang diperoleh dapat digunakan karena sudah memuat *text* dan *class* yang dibutuhkan untuk proses penelitian. Berikut link *dataset* yang dipakai pada penelitian yang bersumber dari Kaggle: <https://www.kaggle.com/datasets/pnkjgpt/fake-news-dataset?select=test.csv>. Pada klasifikasi digunakan data *test* sebesar 10% dari total jumlah

dataset secara keseluruhan. Total jumlah data dari hasil preprocessing adalah 12.420 sehingga data uji yang digunakan untuk prediksi klasifikasi sebanyak 1380 data. Sebelum melakukan klasifikasi, peneliti akan melakukan pembobotan kata dengan *Term Frequency - Inverse Document Frequency* (TF-IDF) untuk mengubah kata pada konten menjadi bentuk vektor agar dapat dibaca oleh model klasifikasi yang akan dibangun. Peneliti menggunakan TF-IDF untuk mengubah kata menjadi bentuk vektor adalah karena data yang dipakai oleh peneliti berupa teks, maka dari itu peneliti menggunakan TF-IDF. Berdasarkan pemaparan diatas, peneliti akan melakukan penelitian untuk menganalisis pengaruh *Particle Swarm Optimization* pada metode *Twin Support Vector Machine* dalam menentukan parameter untuk pengklasifikasian *fake news*.

## 2. Tinjauan Pustaka

### 2.1 Fake news

*Fake news* merupakan suatu informasi yang kebenarannya tidak dapat dipastikan yang dibuat oleh seseorang ataupun pihak tertentu dengan tujuan tertentu seperti tujuan politik (pembentukan opini publik atau propaganda) maupun hasutan (agitasi). *Fake news* akan ada pada saat sebuah rumor yang tersebar ke dalam lingkungan masyarakat tetapi masih banyak hal yang belum terungkap pada rumor tersebut dan akan menjadi pertanyaan bagi masyarakat. Sekarang ini, *fake news* sedang marak tersebar di media *online* dan masyarakat pun kurang peduli dengan tersebarnya *fake news* tersebut. Oleh karena itu, sebagai masyarakat modern dan terpelajar, kita harus bijak dalam mencari dan menelaah informasi agar kita dapat terhindar dari berita bohong yang dapat merugikan masyarakat [6].

### 2.2 Data Preprocessing

*Data preprocessing* adalah tahap yang penting di dalam *data mining* dan pada umumnya data merupakan data mentah yang kotor memiliki *noise*, tidak lengkap, ukurannya besar dan merupakan campuran dari berbagai macam sumber. Maka dari itu perlu dilakukan *Data preprocessing* agar data tersebut memiliki kualitas yang baik. Pada penelitian ini akan dilakukan *Data cleaning* untuk mengatasi nilai null, menghapus kolom yang tidak perlu dan *Data transformation* untuk menangani inkonsistensi [7].

#### 2.2.1 Data Cleaning

*Data cleaning* dilakukan karena pada umumnya data lebih condong tidak lengkap, *noise*, dan tidak konsisten. Pada *data cleaning* juga biasanya dilakukan

pengisian nilai yang hilang, penghalusan *noise* pada saat mengidentifikasi *outlier*, dan memperbaiki inkonsistensi pada data. Dalam mengatasi data yang hilang kita dapat mengabaikan tupel, mengisi nilai yang hilang secara manual atau otomatis. *Data Noise* merupakan kesalahan acak pada data dalam variabel yang diukur, cara untuk menangani data noise adalah dengan melakukan *binning*, *regresi*, dan *clustering*.

Beberapa tahapan *preprocessing* yang dilakukan adalah sebagai berikut [7]:

#### 1. Drop Column

*Drop column* dilakukan untuk menghapus kolom atau atribut yang tidak diperlukan pada penelitian ini. Hal ini dilakukan agar kolom dari *dataset* menjadi *representatif*. Karena ketika data yang *representatif* digunakan, maka akan dapat menyebabkan hasil akurasi yang kurang akurat.

#### 2. Drop Missing Value

*Missing value* merupakan informasi yang hilang pada suatu objek. Nilai yang hilang terjadi karena data tentang sesuatu objek tidak tersedia, sulit ditemukan, atau tidak ada. Adanya *Missing values* terjadi ketika tidak ada mendeteksi data untuk fitur atau variabel dalam pola kasus tertentu dalam kumpulan data. Nilai yang hilang biasanya muncul sebagai "NaN", "?", atau sama sekali tidak memiliki nilai atau *blank cel*. Sehingga dapat disimpulkan bahwa *drop missing value* adalah salah satu kegiatan dalam *data cleaning* yang bertujuan untuk menghapus semua data yang memiliki informasi yang kurang lengkap.

#### 3. Remove Duplicate

*Remove duplicate* merupakan salah satu kegiatan dalam *data cleaning* yang dapat menghapus data yang ganda pada *dataset*. Hal ini dilakukan karena dengan adanya duplikasi data, maka bobot dari suatu kata menjadi tinggi dan hasil dari klasifikasi tidak akurat. Maka data duplikat tersebut dihapus sehingga menjadi satu data saja (unik).

#### 2.2.2 Data transformation

*Data transformation* adalah proses mengubah data dari satu format ke dalam format lain. *Data transformation* dapat sederhana atau kompleks berdasarkan perubahan yang diperlukan pada data

antara data sumber dan data target. *Data transformation* dibagi menjadi beberapa langkah yang masing-masing dapat diterapkan sesuai dengan kebutuhan dan berdasarkan kompleksitas transformasi yang diperlukan, diantaranya adalah *data recovery*, *data mapping*, *code generation*, *code execution* dan *data review*. Untuk klasifikasi teks, transformasi data digunakan untuk membentuk input yang sesuai untuk algoritma klasifikasi [7].

### 2.3 Text Preprocessing

*Text Preprocessing* merupakan tahap awal dalam metode NLP untuk dokumen yang berupa teks (NLP for Text). *Text Preprocessing* merupakan suatu proses untuk menyeleksi data text agar menjadi lebih terstruktur lagi dengan melalui serangkaian tahapan. Ada berbagai tahapan yang dapat digunakan dalam tahap *Text Preprocessing* diantaranya adalah [8]:

#### 1. Case Folding

*Case Folding* digunakan untuk mengubah teks ke dalam format huruf kecil (*lowercase*). Hal ini bertujuan untuk memberikan bentuk standar pada teks.

#### 2. Stopwords Removal

*Stopword Removal* adalah tahap pemilihan kata-kata yang dianggap penting.

#### 3. Tokenization

*Tokenization* merupakan proses pemotongan teks menjadi bagian-bagian yang lebih kecil, yang disebut token.

#### 4. Punctuation Removal

*Punctuation removal* adalah proses untuk menghapus tanda baca atau simbol yang ada di dalam *dataset*. Tanda baca atau simbol yang ada dianggap tidak berpengaruh pada hasil klasifikasi, akan dihapus.

#### 5. Number Removal

*Number removal* adalah proses untuk menghapus semua angka yang ada pada *dataset*. Hal ini dibutuhkan untuk dapat memecah string menjadi token yang berarti.

#### 6. Remove URL

*Remove URL* adalah proses untuk menghapus semua URL yang terdapat pada *dataset*

### 2.4 Twin Support Vector Machine

*Twin Support Vector Machines* atau yang disebut sebagai TWSVM adalah suatu klasifikasi *biner* dimana dua *hyperplane non*

*paralel* akan digunakan dalam memisahkan dua kelas. Dalam hal ini, dua *hyperplane* yang dipertimbangkan bisa didapat dengan memecahkan dua masalah pemrograman kuadrat yang berukuran lebih kecil dibandingkan dengan masalah yang ada dalam pemrograman kuadrat dari *Support Vector Machines* (SVM) [9].

Penelitian kali ini, peneliti akan menggunakan TWSVM untuk mengklasifikasikan dua kelas yaitu *fake news* dan *real news*. Misalnya saja, apabila terdapat  $m$  data latih  $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$  dimana  $x_i \in \mathbb{R}^n$  adalah sampel data dan  $x_i \in \{+1, -1\}$  merupakan target/kelas dari sampel data.  $x_i$  adalah bobot dari variabel konten yang dimana bobot tersebut didapat dari proses vektorisasi dengan TF-IDF sedangkan  $y_i$  adalah label pada konten tersebut yaitu +1 atau -1 dengan +1 adalah *real news* dan -1 adalah *fake news*. *Twin Support Vector Machines* (TWSVM) pada akhirnya akan melakukan pencarian terhadap dua fungsi pemisah (*hyperplane*) yang optimal yakni dengan menggunakan persamaan berikut:

$$f_1(x) = xw_1 + b_1 = 0 \text{ dan } f_2(x) = xw_2 + b_2 = 0 \quad (1)$$

Dimana:

$f_1, f_2$  : fungsi *hyperplane*

$w_1, w_2 \in \mathbb{R}^{n \times 1}$  : Bobot konten yang sudah melalui proses tf-idf

$b_1, b_2 \in \mathbb{R}$  : Parameter bias

### 2.5 Optimasi pada machine learning

Optimasi dikenal sebagai suatu upaya dalam mencari nilai minimum dan maksimum terbaik dari beberapa fungsi tertentu dalam konteks. Secara sederhana, optimasi berupaya untuk meningkatkan kinerja sehingga memiliki kualitas yang baik [10].

Optimasi *machine learning* adalah suatu proses menyesuaikan *hyperparameter* untuk meminimalkan fungsi *cost function* dengan menggunakan salah satu teknik optimasi. Optimasi merupakan inti dari *machine learning*. Optimasi adalah tentang bagaimana mempelajari algoritma yang dapat meminimalkan *loss function*. *Loss function* merepresentasikan perbedaan antara nilai aktual dan nilai prediksi. Dalam hal ini, *machine learning* menggunakan optimasi untuk meminimalkan *loss function* dan mengarah pada kemampuan prediksi yang lebih baik. Teori optimasi adalah suatu proses dalam menentukan atau memilih opsi terbaik berdasarkan kondisi atau kriteria tertentu.

### 2.6 Particle Swarm Optimization

*Particle Swarm Optimization Algorithm* merupakan algoritma optimasi yang bisa dipakai dalam pengambilan keputusan dan juga untuk mencari jalur. *Particle Swarm Optimization* adalah teknik pengoptimasian yang secara terus-menerus menghitung kemungkinan solusi menggunakan tolak ukur kualitas. Algoritma ini dapat mengoptimasi suatu masalah yaitu dengan cara menggerakkan solusi atau partikel yang mungkin dalam ruang masalah menggunakan posisi partikel dan fungsi kecepatan tertentu. Pergerakan partikel ini akan dipengaruhi oleh solusi terbaik dari partikel tersebut, dan solusi terbaik itu umumnya diperoleh dari partikel lain. Kumpulan partikel tersebut disebut dengan *swarm*, pada akhirnya *swarm* tersebut akan bergerak ke solusi terbaik [11].

### 2.7 Term Frequency- Inverse Document Frequency (TF-IDF)

*Term Frequency - Inverse Document Frequency* (TF-IDF) merupakan metode yang digunakan untuk menghitung dan mendapatkan bobot setiap kata yang umum digunakan pada sebuah dokumen. TF-IDF menggunakan dua konsep untuk melakukan perhitungan bobot yaitu konsep pertama adalah *Term Frequency* (TF), dimana konsep ini menghitung bobot berdasarkan frekuensi kemunculan sebuah kata di dalam suatu dokumen. Konsep kedua adalah *Inverse Document Frequency* (IDF) yang dimana konsep ini menghitung bobot suatu kata berdasarkan frekuensi banyaknya dokumen yang mengandung kata tersebut. TF-IDF memiliki persamaan sebagai berikut:

$$TF = f_{t,d} / \sum_{t \in d} f_{t,d} \quad (2)$$

$$IDF = \log \log \left( \frac{N}{df_t} \right) \quad (3)$$

$$TF - IDF = TF \times IDF \quad (4)$$

Dimana:

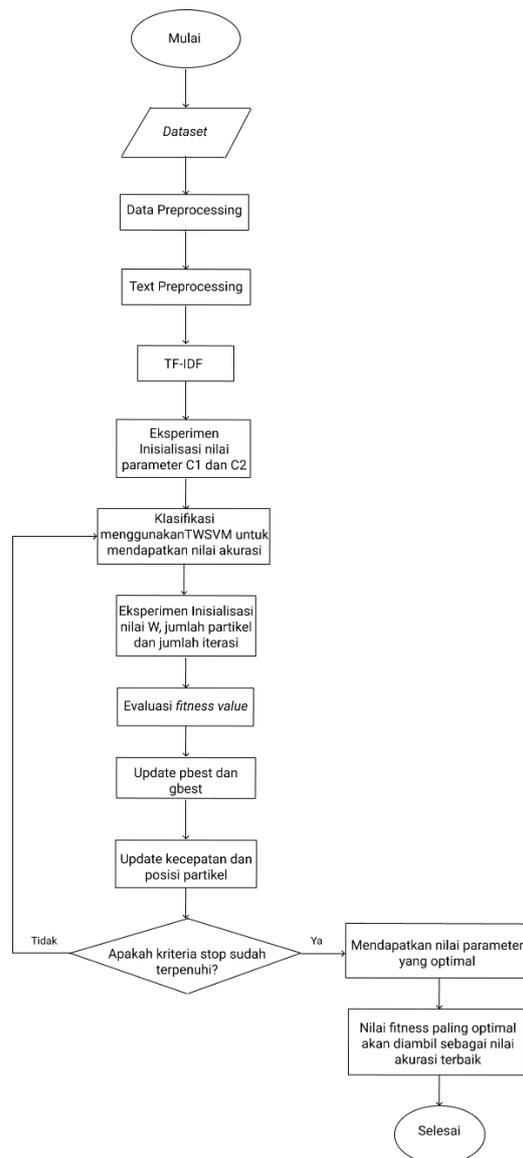
- $d$  : dokumen
- $t$  : term
- $f_{t,d}$  : Jumlah kata pada suatu term (t) di setiap dokumen (d)
- $\sum_{t \in d}$  : Jumlah dokumen (d) yang memuat term (t)
- $TF$  : Hasil dari Term Frequency
- $N$  : Jumlah dokumen keseluruhan
- $df_t$  : Jumlah kata pada term (t) yang ada pada seluruh dokumen (N)
- $IDF$  : Hasil dari Inverse Document Frequency (IDF)

### 2.8 Confusion matrix

*Confusion matrix* berisi informasi mengenai klasifikasi aktual dan prediksi oleh klasifikasi sistem. Performa suatu sistem biasanya dievaluasi menggunakan data dalam matriks. Pada *confusion matrix*, nilai mewakili *instance* di kelas yang diprediksi, sedangkan setiap baris berisi nilai yang mewakili *instance* di kelas yang sebenarnya. Berikut gambar yang menunjukkan *confusion matrix* untuk mengklasifikasikan dua kelas.

### 3. Metodologi Penelitian

Adapun tahapan penelitian dapat dilihat pada gambar 1.



Gambar 1 Metode Penelitian

- Tahap pertama adalah Dataset dikumpulkan dari *kaggle*.
- Tahap kedua adalah Setelah dataset didapatkan, maka akan dilakukan *preprocessing* yang terdiri dari *data preprocessing* dan *text preprocessing*.

- Tahap ketiga adalah setelah data berhasil di *preprocessing*, selanjutnya akan dilakukan vektorisasi data untuk mengubah data kata menjadi bobot/angka dengan menggunakan TF-IDF.
- Tahap keempat adalah Setelah bobot setiap kata dalam setiap indeks didapatkan, maka akan dilakukan inisialisasi nilai partikel PSO.
- Tahap kelima akan dilakukan klasifikasi TWSVM dengan menggunakan nilai  $C_1$  dan  $C_2$  pada setiap eksperimen. Hasil dari klasifikasi tersebut adalah berupa akurasi. Akurasi pada klasifikasi tersebut akan digunakan sebagai nilai *fitness* pada PSO.
- Tahap keenam akan dilakukan evaluasi nilai *fitness* untuk melihat apakah nilai *fitness* tersebut sudah maksimum. Jika iterasi sudah maksimum, maka akan didapatkan nilai parameter yang optimal. Jika tidak, maka akan dilakukan *update gbest, pbest*, posisi dan kecepatan partikel. Hal tersebut akan dilakukan berulang sampai nilai *fitness* sudah maksimum.
- Tahap terakhir yaitu Setelah nilai *fitness* sudah maksimum, maka nilai *fitness* tersebut akan diambil menjadi nilai akurasi paling optimal.

#### 4. Hasil dan Pembahasan

##### 4.1 Hasil

Pada penelitian penerapan *Particle Swarm Optimization* dalam pengklasifikasian *fake news* menggunakan *Twin Support Vector Machine*, dilakukan beberapa percobaan nilai parameter yang dibuat. Percobaan tersebut berdasarkan tiga parameter yaitu eksperimen berdasarkan nilai parameter pada *Particle Swarm Optimization* yaitu parameter  $C_1$ ,  $C_2$ , dan  $w$ .

Pada klasifikasi digunakan data test sebesar 10% dari total jumlah *dataset* secara keseluruhan. Total jumlah data dari hasil *preprocessing* adalah 12.420 sehingga data uji yang digunakan untuk prediksi klasifikasi sebanyak 1380 data. Dari hasil evaluasi model yang dilakukan pada setiap eksperimen, maka didapatkan nilai *accuracy*, *precision*, *recall* dan *f-1 score* yang bervariasi namun rentang nilainya yang tidak terlalu jauh.

*Particle Swarm Optimization* dibangun untuk mendapatkan parameter terbaik untuk *Twin Support Vector Machine* dengan menentukan pemilihan nilai yang tepat untuk setiap *hyperparameter* yang digunakan. Hasil evaluasi menunjukkan model yang dibangun mampu mengklasifikasi data test dengan baik.

##### a. Hasil Implementasi Eksperimen Koefisien $C_1$ dan $C_2$

Eksperimen yang pertama adalah pengaruh koefisien  $C_1$  dan  $C_2$  terhadap akurasi yang dihasilkan. Berikut merupakan hasil implementasi eksperimen koefisien  $C_1$  dan  $C_2$  :

Tabel 1. Hasil Implementasi Eksperimen Koefisien  $C_1$  dan  $C_2$

Eksperimen	$C_1$	$C_2$	$w$	Partikel	Iterasi	Akurasi
1	2	2	0.75	10	10	89.275
2	2	1	0.75	10	10	89.203
3	1	2	0.75	10	10	89.203
4	3	2	0.75	10	10	88.623
5	2	3	0.75	10	10	88.623

##### b. Hasil Implementasi Eksperimen ukuran populasi

Eksperimen yang kedua adalah pengaruh ukuran populasi terhadap akurasi yang dihasilkan. Berikut merupakan hasil implementasi eksperimen ukuran populasi:

Tabel 2. Hasil Implementasi Eksperimen Ukuran Populasi

Eksperimen	$C_1$	$C_2$	$w$	Partikel	Iterasi	Akurasi
1	2	2	0.75	10	10	89.205
2	2	2	0.75	20	10	89.203
3	2	2	0.75	30	10	89.203
4	2	2	0.75	40	10	89.420
5	2	2	0.75	50	10	89.855

##### c. Hasil Implementasi Eksperimen bobot inersia

Eksperimen yang ketiga adalah pengaruh nilai bobot inersia terhadap akurasi yang dihasilkan. Berikut merupakan hasil implementasi eksperimen bobot inersia:

Tabel 3. Hasil Implementasi Eksperimen Bobot Inersia

Eksperimen	$C_1$	$C_2$	$w$	Partikel	Iterasi	Akurasi
1	2	2	0.25	10	10	89.275
2	2	2	0.5	10	10	89.348
3	2	2	0.75	10	10	89.420

##### 4.2 Pembahasan

Pada bagian ini akan dijelaskan pembahasan eksperimen terhadap proses optimasi model klasifikasi *Twin Support Vector Machine* dengan menggunakan *Particle Swarm Optimization* beserta dengan hasil evaluasi pengklasifikasian *fake news*.

##### 1. Pembahasan Hasil Eksperimen

Eksperimen yang dilakukan pada penelitian ini adalah melakukan optimasi *Twin Support Vector Machine* dengan menggunakan *Particle Swarm Optimization*. Peneliti

menggunakan *Particle Swarm Optimization* dalam menemukan parameter terbaik untuk dipakai dalam *Twin Support Vector Machine*. Dalam menemukan parameter, peneliti akan melakukan inisialisasi pada setiap parameter  $C_1$ ,  $C_2$ , dan  $w$ . Peneliti akan melakukan beberapa tahap yaitu mengenai desain tahapan melakukan eksperimen.

Berikut perbandingan hasil akurasi yang didapatkan setelah melakukan eksperimen pengaruh jarak  $C_1$  dan  $C_2$ , pengaruh jumlah partikel dan pengaruh jumlah iterasi:

#### I. Eksperimen Koefisien $C_1$ dan $C_2$

Eksperimen yang pertama adalah menguji bagaimana pengaruh koefisien  $C_1$  dan  $C_2$ . Parameter  $C_1$  dan  $C_2$  berfungsi untuk menyeimbangkan antara global eksplorasi dan lokal eksploitasi agar menjadi lebih baik. Global eksploitasi merupakan nilai *fitness* tertinggi yang didapatkan oleh seluruh partikel dalam suatu populasi, sedangkan lokal eksploitasi merupakan *fitness* tertinggi yang didapatkan oleh satu partikel saat itu juga. Nilai koefisien akselerasi yang besar mengakibatkan pergerakan partikel dalam menempati posisinya yang baru menjadi relatif lebih jauh dari pergerakan biasanya sehingga kemampuan menjelajah partikel tersebut menjadi lebih luas dan baik namun terkadang dapat melebihi atau kurang dari batas pencarian. Pada pengujian koefisien akselerasi ini dicari kombinasi nilai  $C_1$  dan  $C_2$  terbaik dalam pencarian solusi pada sistem [12].

Berdasarkan tabel 1 hasil eksperimen berdasarkan koefisien  $C_1$  dan  $C_2$  diatas dapat dilihat bahwa akurasi tertinggi didapat ketika nilai koefisien  $C_1$  dan  $C_2$  sama yaitu sebesar 89.275%. Namun apabila nilai  $C_1$  dan  $C_2$  yang dipakai di balik misalnya  $C_1 = 2.0$  dan  $C_2 = 1.0$  diubah menjadi  $C_1 = 1.0$  dan  $C_2 = 2.0$ , akurasi yang dihasilkan adalah sama yaitu sebesar 89.203%.

#### II. Eksperimen Berdasarkan Ukuran Populasi

Eksperimen yang kedua adalah menguji bagaimana pengaruh jumlah populasi dalam.

Berdasarkan hasil eksperimen pada tabel 2, terlihat bahwa semakin besar ukuran dari populasi maka semakin tinggi akurasi klasifikasi yang dihasilkan. Pada eksperimen ini

ukuran populasi yang terbesar yakni 50 partikel mendapat akurasi paling tinggi yakni sebesar 89.855%. Ukuran populasi berkaitan dengan kompleksitas masalah dalam klasifikasi. Sebagaimana peningkatan kompleksitas dan ukuran populasi terjadi, konvergensi dan stabilitas dari PSO menjadi semakin meningkat. Ukuran populasi juga berpengaruh terhadap waktu yang diperlukan CPU dimana semakin besar ukuran populasi maka semakin banyak waktu yang diperlukan CPU dalam mengeksekusi program.

Pernyataan juga didukung oleh jurnal dengan judul "*Population size in Particle Swarm Optimization*" [13] yang mengatakan bahwa semakin besar ukuran dari populasi, semakin jauh pencarian solusi yang dihasilkan algoritma. Dan dengan ukuran populasi. Dengan ukuran populasi yang lebih tinggi pula, setiap generasi memanggil lebih banyak fungsi dan semakin luas juga area pencarian solusi yang mungkin ditempuh. Namun berbicara mengenai ukuran populasi di PSO sering diabaikan akhir akhir ini. Pernyataan ini juga didukung dengan Paper yang berjudul "*Population size in Particle Swarm Optimization*" [13] yang menyebutkan bahwa tidak ada ukuran populasi yang direkomendasikan. Namun pada umumnya ukuran populasi yang paling banyak ada pada rentang 20-50.

#### III. Eksperimen Berdasarkan Bobot Inersia

Eksperimen yang terakhir adalah menguji bagaimana pengaruh bobot inersia.

Berdasarkan tabel 3 hasil eksperimen nilai bobot inersia diatas dapat dilihat bahwa akurasi tertinggi terjadi pada eksperimen 3 yaitu sebesar 89.420% yang dimana peneliti menggunakan bobot inersia sebesar 0.75. Dari hasil eksperimen tersebut dapat disimpulkan bahwa semakin besar nilai bobot inersia maka akan semakin besar nilai akurasi dari klasifikasi yang dihasilkan [11]. Hal ini terjadi karena bobot inersia memiliki dampak terhadap kemampuan pencarian solusi global maupun lokal. Ketika nilai bobot inersia kecil, algoritma PSO hampir tidak konvergen dan dapat dilihat bahwa tingkat keberhasilannya

dalam memberi pengaruh terhadap algoritma klasifikasi TWSVM lebih rendah seperti kita lihat pada percobaan pertama dengan nilai  $w$  sebesar 0.25 dimana hasil akurasi adalah yang terendah yakni 89.275%.

## 2. Pembahasan Hasil Evaluasi

Pada bagian ini akan dijelaskan pembahasan terkait dengan proses dan hasil yang telah dilakukan pada penelitian ini. Pembahasan dilihat berdasarkan hasil evaluasi model *confusion matrix* dari kedua model. Data yang dipakai pada penelitian ini adalah data test sebesar 10% yaitu sebanyak 1380 data. Hasil pada setiap model klasifikasi ditampilkan dalam bentuk visualisasi pada setiap poin evaluasi untuk menampilkan perbedaan dari segi grafik yang ditunjukkan.

### a. Akurasi

Berikut merupakan tabel perbandingan akurasi klasifikasi *fake news* dengan menggunakan *Twin Support Vector Machine* dan dengan *Particle Swarm Optimization-Twin Support Vector Machine*.

Tabel 4. Perbandingan Akurasi TWSVM dengan PSO-TWSVM pada klasifikasi *fake news*

Eksperimen	C <sub>1</sub>	C <sub>2</sub>	Akurasi	
			TWSVM	PSO-TWSM (w=0.75, partikel=10 dan iterasi=10)
1	2	2	87.03	89.276
2	2	1	87.90	89.202
3	1	2	86.38	89.202
4	3	2	87.03	88.623
5	2	3	87.25	88.623

Tabel 7 merupakan perbandingan hasil akurasi klasifikasi *fake news* dengan *Twin Support Vector Machine* dan klasifikasi dengan *Twin Support Vector Machine* yang dioptimasi dengan *Particle Swarm Optimization*.

Pada klasifikasi ini, peneliti melakukan 5 percobaan untuk mengetahui perbandingan hasil klasifikasi dengan *Twin Support Vector Machine* dan hasil klasifikasi dengan *Twin Support Vector Machine* yang dioptimasi dengan *Particle Swarm Optimization*. Eksperimen dilakukan menggunakan 5 nilai parameter  $C_1$  dan  $C_2$  yang berbeda-beda. Kedua klasifikasi tersebut dilakukan dengan menggunakan nilai

parameter  $C_1$  dan  $C_2$  yang sama untuk setiap eksperimen, hanya saja untuk PSO-TWSVM dibutuhkan parameter lain yaitu  $w$ , jumlah partikel, dan jumlah iterasi. Untuk klasifikasi PSO-TWSVM, peneliti menggunakan nilai parameter  $w$  sebanyak 0.75, jumlah partikel sebanyak 10, dan jumlah iterasi sebanyak 10.



Gambar 2. Visualisasi Akurasi

Berdasarkan hasil visualisasi di atas, model PSO-TWSVM menghasilkan nilai akurasi tertinggi di setiap eksperimen yang telah diimplementasikan dalam klasifikasi *fake news*. Dengan nilai akurasi pada eksperimen 1 sebesar 89.276%, eksperimen 2 sebesar 89.202%, eksperimen 3 sebesar 89.202%, eksperimen 4 sebesar 88.623% dan eksperimen 5 sebesar 88.623%. Kemudian, pada TWSVM menghasilkan akurasi yang lebih rendah dibandingkan dengan PSO-TWSVM yaitu pada eksperimen 1 sebesar 87.03%, eksperimen 2 sebesar 87.90%, eksperimen 3 sebesar 86.38%, eksperimen 4 sebesar 87.03% dan eksperimen 5 sebesar 87.25%.

Tabel 5. Hasil Confusion matrix

Eksperimen	Model	Confusion matrix			
		TP	TN	FP	FN
1	TWSVM	552	649	111	68
	PSO - TWSVM	754	122	26	478
2	TWSVM	543	670	97	70
	PSO - TWSVM	722	91	58	509
3	TWSVM	563	629	131	57
	PSO - TWSVM	753	122	27	478
4	TWSVM	535	666	94	85
	PSO - TWSVM	716	86	71	507
5	TWSVM	556	648	112	64
	PSO - TWSVM	715	72	85	508

Akurasi menunjukkan rasio prediksi benar positif dan benar negatif (*True*

*Positive* dan *True Negative*) dibandingkan keseluruhan data [14]. Dimana akurasi dapat menunjukkan berapa persen berita yang benar diprediksi *fake news* dan *real news* dibandingkan dengan keseluruhan jumlah berita.

Dapat dilihat dari tabel 11 bahwa total jumlah berita yang benar diklasifikasi *fake news* (*True Positive*) dan jumlah jumlah berita yang benar diklasifikasikan *real news* (*True Negative*) oleh model PSO-TWSVM lebih tinggi dibandingkan dengan total jumlah berita yang benar diklasifikasi *fake news* (*True Positive*) dan jumlah jumlah berita yang benar diklasifikasikan *real news* (*True Negative*) oleh model TWSVM.

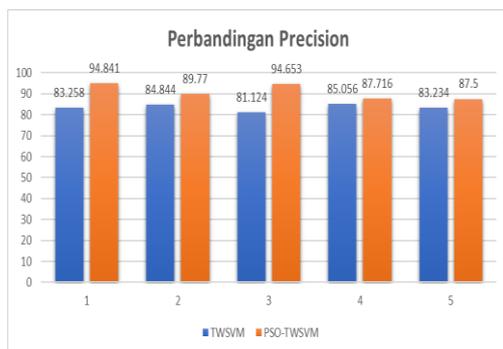
Maka dari hasil kelima eksperimen tersebut, optimasi dengan menggunakan *Particle Swarm Optimization* dapat meningkatkan nilai akurasi pada model pengklasifikasian *Twin Support Vector Machine* dalam klasifikasi *fake news*.

b. Presisi

Berikut merupakan tabel perbandingan presisi antara *Twin Support Vector Machine* dengan *Particle Swarm Optimization-Twin Support Vector Machine* dalam klasifikasi *fake news*.

Tabel 6. Perbandingan Presisi TWSVM dengan PSO-TWSVM

Eksperimen	C		Presisi	
	1	2	TWSVM	PSO-TWSM (w=0.75, partikel=10 dan iterasi=10)
1	2	2	83.258	94.841
2	2	1	84.844	89.77
3	1	2	81.124	94.653
4	3	2	85.056	87.716
5	2	3	83.234	87.50



Gambar 3. Visualisasi Presisi

Berdasarkan gambar 3 yaitu visualisasi perbandingan nilai presisi dari setiap model yang telah dibangun pada penelitian, model PSO-TWSVM menghasilkan nilai presisi tertinggi di

setiap eksperimen yang telah diimplementasikan dalam klasifikasi *fake news*. Dengan nilai akurasi pada eksperimen 1 sebesar 94.841%, eksperimen 2 sebesar 89.77%, eksperimen 3 sebesar 94.653%, eksperimen 4 sebesar 87.716% dan eksperimen 5 sebesar 87.5%. Kemudian, pada TWSVM menghasilkan akurasi yang lebih rendah dibandingkan dengan PSO-TWSVM yaitu pada eksperimen 1 sebesar 83.258%, eksperimen 2 sebesar 84.844%, eksperimen 3 sebesar 81.124%, eksperimen 4 sebesar 85.056% dan eksperimen 5 sebesar 83.234%.

Presisi menunjukkan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang diprediksi positif, dimana presisi adalah berapa persen berita yang benar *fake news* dibandingkan dengan keseluruhan jumlah berita yang diprediksi *fake news*. Indikator *Precision* memperhitungkan pengaruh *False Positive*, namun tidak memperhitungkan nilai *False Negative* [14].

Oleh karenanya, *Precision* bisa diprioritaskan ketika resiko prediksi jumlah berita *fake news* yang diprediksi sebagai *real news* (*False Positive*) sangatlah berbahaya, sedangkan resiko jumlah berita *real news* yang diprediksi sebagai *fake news* (*False Negative*) dianggap tidak begitu berbahaya.

Pada tabel 8 perbandingan confusion matrix PSO dan PSO-TWSVM dapat kita lihat bahwa jumlah berita *fake news* yang diprediksi sebagai *real news* (*False Positive*) lebih banyak dibandingkan jumlah berita *real news* yang diprediksi sebagai *fake news* (*False Negative*).

Maka dari hasil kelima eksperimen tersebut, optimasi dengan menggunakan *Particle Swarm Optimization* dapat meningkatkan nilai presisi pada model pengklasifikasian *Twin Support Vector Machine* dalam klasifikasi *fake news*.

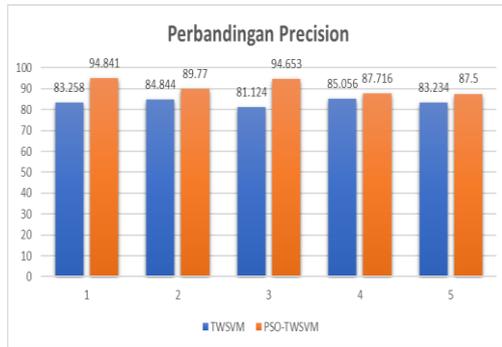
c. Recall

Berikut merupakan tabel perbandingan *recall* antara *Twin Support Vector Machine* dengan *Particle Swarm Optimization-Twin Support Vector Machine*.

Tabel 7. Perbandingan Recall TWSVM dengan PSO-TWSVM

Eksperimen	C		Recall	
	1	2	TWSVM	PSO-TWSM (w=0.75, partikel=10 dan iterasi=10)
1	2	2	89.032	79.666
2	2	1	88.580	84.833

3	1	2	90.806	79.667
4	3	2	86.290	85.497
5	2	3	89.677	85.667



Gambar 4. Visualisasi Recall

Berdasarkan gambar 4 yaitu visualisasi perbandingan nilai *recall* dari setiap model yang telah dibangun pada penelitian, model PSO-TWSVM menghasilkan nilai *recall* yang lebih rendah di setiap eksperimen yang telah diimplementasikan dalam klasifikasi *fake news*. Dengan nilai *recall* pada eksperimen 1 sebesar 79.666%, eksperimen 2 sebesar 84.833%, eksperimen 3 sebesar 79.667%, eksperimen 4 sebesar 85.497% dan eksperimen 5 sebesar 85.667%. Kemudian, pada TWSVM menghasilkan akurasi yang lebih rendah dibandingkan dengan PSO-TWSVM yaitu pada eksperimen 1 sebesar 89.032%, eksperimen 2 sebesar 88.58%, eksperimen 3 sebesar 90.806%, eksperimen 4 sebesar 86.29% dan eksperimen 5 sebesar 89.667%.

*Recall* menunjukkan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif, dimana presisi adalah berapa persen berita yang diprediksi *fake news* dibandingkan dengan keseluruhan jumlah berita yang memang *fake news* [14].

Pada tabel 8 perbandingan confusion matrix PSO dan PSO-TWSVM dapat kita lihat bahwa jumlah berita *fake news* yang diprediksi sebagai *real news* (*False Positive*) lebih sedikit dibandingkan jumlah berita *real news* yang diprediksi sebagai *fake news* (*False Negative*).

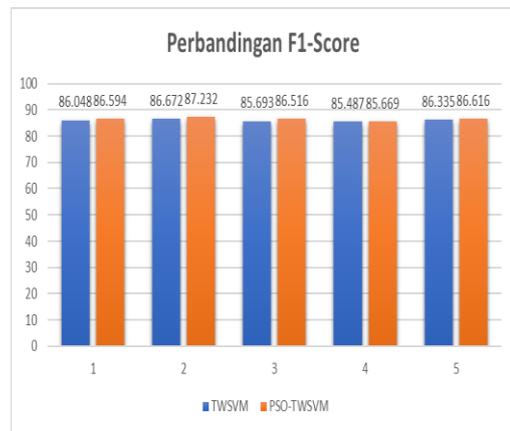
Maka dari hasil kelima eksperimen tersebut, optimasi dengan menggunakan *Particle Swarm Optimization* dapat menurunkan nilai *recall* pada model pengklasifikasian *Twin Support Vector Machine* dalam klasifikasi *fake news*.

d. *F-1 score*

Berikut merupakan tabel perbandingan *F1-Score* antara *Twin Support Vector Machine* dengan *Particle Swarm Optimization-Twin Support Vector Machine*.

Tabel 8. Perbandingan *F1-Score* TWSVM dengan TWSVM-PSO

Eksperimen	C <sub>1</sub>	C <sub>2</sub>	F-1 Score	
			TWSVM	PSO-TWSM (w=0.75, partikel=10 dan iterasi=10)
1	2	2	89.032	79.666
2	2	1	88.580	84.833
3	1	2	90.806	79.667
4	3	2	86.290	85.497
5	2	3	89.677	85.667



Gambar 5. Visualisasi *F1-Score*

Berdasarkan gambar 5 yaitu visualisasi perbandingan nilai *f-1 score* dari setiap model yang telah dibangun pada penelitian, model PSO-TWSVM menghasilkan nilai *f-1 score* tertinggi di setiap eksperimen yang telah diimplementasikan dalam klasifikasi *fake news*. Dengan nilai akurasi pada eksperimen 1 sebesar 86.594%, eksperimen 2 sebesar 87.232%, eksperimen 3 sebesar 86.516%, eksperimen 4 sebesar 85.669% dan eksperimen 5 sebesar 86.616%. Kemudian, pada TWSVM menghasilkan akurasi yang lebih rendah dibandingkan dengan PSO-TWSVM yaitu pada eksperimen 1 sebesar 86.048%, eksperimen 2 sebesar 86.672%, eksperimen 3 sebesar 85.693%, eksperimen 4 sebesar 85.487% dan eksperimen 5 sebesar 86.616%.

*F-1 score* digunakan untuk menyeimbangkan nilai presisi dan *recall*. Oleh karena *F1-Score* memperhitungkan nilai *Precision* dan *Recall*, ini berarti bahwa *F1-Score* memperhitungkan pentingnya nilai *False Positive* dan *False Negative*. Dengan kata lain, indikator *F1-*

Score menekankan bahaya/resiko dari *False Positive* dan *False Negative* [14].

Pada tabel 11 perbandingan confusion matrix PSO dan PSO-TWSVM dapat kita lihat bahwa total jumlah berita *fake news* yang diklasifikasikan sebagai *real news* (*False Positive*) dan jumlah jumlah berita *real news* yang diklasifikasikan sebagai *fake news* (*False Negative*) lebih kecil dari total jumlah berita yang benar diklasifikasi *fake news* (*True Positive*) dan jumlah jumlah berita yang benar diklasifikasikan *real news* (*True Negative*).

Maka dari hasil kelima eksperimen tersebut, optimasi dengan menggunakan *Particle Swarm Optimization* dapat meningkatkan nilai *F-1 score* pada model pengklasifikasian *Twin Support Vector Machine* dalam klasifikasi *fake news*.

#### IV. Kesimpulan

Berdasarkan hasil keseluruhan eksperimen parameter yang telah dilakukan, dapat disimpulkan bahwa:

- 1) Penggunaan *Particle Swarm Optimization* pada metode *Twin Support Vector Machine* dapat meningkatkan akurasi sebesar 1.0-3.0%, presisi sebesar 4.0-11.0% , dan f-1 skor sebesar 0.5-1.0% pada klasifikasi *fake news*.
- 2) Penggunaan *Particle Swarm Optimization* pada metode *Twin Support Vector Machine* dapat menurunkan *recall* sebesar 1.0%-11.0% pada klasifikasi *fake news*. Hal ini disebabkan karena jumlah berita *real news* yang diprediksi sebagai *fake news* (*False Negative*) pada klasifikasi *fake news* dengan menggunakan *Particle Swarm Optimization - Twin Support Vector Machine* lebih besar dibandingkan jumlah berita *real news* yang diprediksi sebagai *fake news* (*False Negative*) pada klasifikasi *fake news* dengan menggunakan *Twin Support Vector Machine*. Jadi, semakin banyak jumlah *False Negative*, nilai *Recall* akan menjadi semakin kecil [15].
- 3) Untuk mendapatkan parameter terbaik dalam pengklasifikasian *fake news* dibutuhkan inisialisasi parameter  $C_1$  ,  $C_2$  ,  $W$ , jumlah partikel, dan jumlah iterasi.
  - a) Nilai  $C_1$  dan  $C_2$  mempengaruhi akurasi klasifikasi *fake news*. Ketika nilai  $C_1$  dan  $C_2$  adalah sama, maka akurasi pada klasifikasi akan semakin besar yaitu dari 87.03% menjadi 89.276%.
  - b) Nilai  $W$  mempengaruhi akurasi klasifikasi *fake news* yang dimana semakin besar nilai  $W$ , maka akurasi pada klasifikasi *fake*

*news* akan semakin meningkat yaitu sebesar 89.420%.

- c) Ukuran populasi yang terdiri dari banyak partikel mempengaruhi akurasi klasifikasi *fake news* dimana semakin besar ukuran populasi, maka akurasi pada klasifikasi *fake news* akan semakin meningkat yaitu sebesar 89.855%.

#### Daftar Rujukan

- [1] C. Juditha, "Interaksi Komunikasi Hoax di Media Sosial serta Antisipasinya," *Jurnal Pekommas*, vol. 3, 2018.
- [2] R. PEBI, "Analisis Performansi Algoritma Naïve Bayes dan *Support Vector Machine* untuk Deteksi Berita Hoax Berbahasa Inggris," Telkom University, 2020. [Online]. Available: <https://repository.telkomuniversity.ac.id/pustaka/158821/analisis-performansi-algoritma-na-ve-bayes-dan-support-vector-machine-untuk-deteksi-berita-hoax-berbahasa-inggris.html>.
- [3] K. Li, G. Zhou, Y. Yang, F. Li dan Z. Jiao, "A novel prediction method for favorable reservoir of oil field based on grey wolf optimizer and *Twin Support Vector Machine*," *Journal of Petroleum Science and Engineering*, vol. 189, 2020.
- [4] K. K. Bharti dan S. Pandey, "Fake account detection in twitter using logistic regression with *Particle Swarm Optimization*," *Application of soft computing*, 2021.
- [5] N. Surantha, T. F. Lesmana dan S. M. Isa, "Sleep stage classification using extreme learning machine and *Particle Swarm Optimization* for healthcare big data," *Journal of Big Data*, 2021.
- [6] B. PRATAMA, "HOAX DAN FAKE NEWS DALAM UU-ITE," Binus University, 9 August 2018. [Online]. Available: <https://business-law.binus.ac.id/2018/08/09/hoax-dan-fake-news-dalam-uu-ite/>.
- [7] G. Nurvinda, "Pentingnya *Preprocessing* dalam Pengolahan Data Statistik," DQLab,

- 12 April 2021. [Online]. Available: <https://www.dqlab.id/pentingnya-preprocessing-dalam-pengolahan-data-statistik#:~:text=Preprocessing%20data%20sangat%20penting%20karena,gunakan.>
- [8] R. Tineges, “Tahapan Text *Preprocessing* dalam Teknik Pengolahan Data,” DQLab, 17 June 2021. [Online]. Available: <https://www.dqlab.id/tahapan-text-preprocessing-dalam-teknik-pengolahan-data.>
- [9] Jayadeva, R. Khemchandani dan S. Chandra, *Twin Support Vector Machines*, 2017.
- [10] E. Hazan, *Optimization for Machine learning*, 2019.
- [11] S. Khomsah, “*Naive Bayes Classifier Optimization on Sentiment Analysis of Hotel Reviews*,” *Jurnal Penelitian Pos dan Informatika*, 2020.
- [12] A. P. Engelbrecht, *Computational Intelligence*.
- [13] A. P. Piotrowski, J. J. Napiorkowski dan A. E. Piotrowska, “Population size in *Particle Swarm Optimization*,” *Swarm and Evolutionary Computation*, 2020.
- [14] M. Santos, “Towards Data Science,” 18 May 2020. [Online]. Available: <https://towardsdatascience.com/explaining-precision-vs-recall-to-everyone-295d4848edaf.>
- [15] “Teknologi Big Data,” 29 May 2020. [Online]. Available: <https://www.teknologi-bigdata.com/2020/05/validitas-rapid-test-covid-19-akurasi-accuracy-vs-F1-Score.html#:~:text=F1%20score%20digunakan%20ketika%20False,True%20Positive%20dan%20True%20Negative.>