

Optimasi *Support Vector Machine* Berbasis *Particle Swarm Optimization* Untuk Mendeteksi *Hate Speech* Pilkada Karawang

Wahyuningrum^{1*}, Rijal Abdulhakim^{2*}, Yuyun Umaidah^{3*}, Jajam Haerul Jaman^{4*}

* Teknik Informatika, Universitas Singaperbangsa Karawang

wahyuningrum.ayu17214@student.unsika.ac.id¹, rijal.abulhakim16182@student.unsika.ac.id², yuyun.umaidah@staff.unsika.ac.id³, jajam.haeruljaman@staff.unsika.ac.id⁴

Article Info

Article history:

Received 2021-10-05

Revised 2021-11-18

Accepted 2021-12-04

Keyword:

KDD,

Klasifikasi Teks,

Particle Swarm Optimization,

Support Vector Machine.

ABSTRACT

Maraknya ujaran kebencian di media sosial bisa saja merugikan berbagai pihak termasuk calon kepala daerah tahun 2020. Data yang diperoleh dari Indeks Kerawanan Pemilu (IKP) 2020, kabupaten/kota di Jawa Barat yang menempati posisi peringkat 1 tingkat kerawanan yaitu Kabupaten Karawang dengan skor 63,77% pada level 5. Karena banyaknya jumlah komentar di media sosial mengakibatkan sanksi yang diberikan kepada para pelanggar belum merata. Oleh karena itu penelitian ini dilakukan untuk mengklasifikasi komentar positif dan negatif dalam memudahkan bawaslu untuk membedakan komentar yang termasuk ke dalam ujaran kebencian dan bukan ujaran kebencian sehingga sanksi yang diberikan kepada para pelanggar itu tepat sasaran dan dapat memberikan efek jera kepada masyarakat agar tidak melakukan lagi ujaran kebencian. Metodologi yang digunakan yaitu *Knowledge Discovery in Database* (KDD) dengan dua pemodelan, pertama Algoritma *Support Vector Machine* (SVM) dan yang kedua Algoritma *Support Vector Machine* (SVM) dioptimasi dengan *Particle Swarm Optimization*. Hasil yang didapat dari pemodelan pertama menyatakan bahwa Algoritma *Support Vector Machine* (SVM) dengan skenario "2" pada kernel *linear* mendapatkan nilai performa yang bagus sebesar "72.66%". Sedangkan pada pemodelan kedua dengan Algoritma *Support Vector Machine* (SVM) dioptimasi oleh *Particle Swarm Optimization* mendapatkan nilai akurasi tertinggi pada kernel *linear* dan *polynomial* di skenario ke 4 dengan pembagian data 90:10 sebesar "78.00%". Nilai evaluasi yang lain pun mengalami kenaikan mulai dari presisi menjadi 72.72%, recall menjadi 50.00% dan f1-score menjadi 59.25%.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

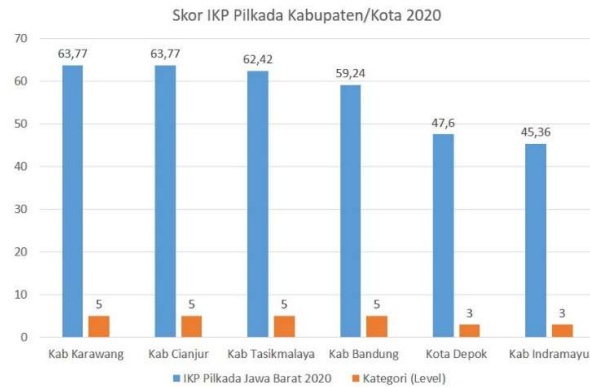
I. PENDAHULUAN

Pilkada dalam negara demokrasi bertujuan agar rakyat bisa memilih kepala daerah yang dapat memperjuangkan kepentingan - kepentingannya. Setiap rakyat mempunyai hak dan kewajibannya untuk memilih kepala daerah, pilkada yang demokratis harus memenuhi syarat seperti adanya persaingan secara sehat dalam memperebutkan kekuasaan, partisipasi dari masyarakat dan jaminan hak sipil dan politik. Kesadaran masyarakat dan keterlibatan aktif harus tertanam dalam penyelenggaraan pilkada ini karena apabila masyarakat tidak hadir maka pilkada yang dilaksanakan bukan pilkada yang demokratis [1].

Pilkada tahun ini diselenggarakan pada 9 Desember 2020 untuk wilayah dengan masa jabatan berakhir pada tahun 2021. Pilkada di selenggarakan di 7 kota, 224 kabupaten dan 9 provinsi. Hanya 2 provinsi tidak mengikuti pilkada serentak, diantaranya Provinsi Aceh dan Provinsi DKI Jakarta [2].

Berdasarkan data Indeks Kerawanan Pemilu (IKP) 2020, kabupaten/kota di Jawa Barat posisi yang menempati peringkat 1 tingkat kerawanan yaitu Kabupaten Karawang dengan skor 63,77 pada level 5 dan dapat dilihat dari rekam jejak pemilu 2019 Karawang menempati posisi ke 3 di pulau

jawa dan posisi ke 16 dari 270 kabupaten/kota di Indonesia [3].



Gambar 1. Grafik Skor IKP Pilkada Kabupaten/Kota di Jawa Barat 2020

Maraknya ujaran kebencian pada media sosial bisa saja merugikan berbagai pihak termasuk calon kepala daerah Kabupaten Karawang tahun 2020. Media sosial yang sering digunakan masyarakat Karawang untuk menyebarkan ujaran kebencian yaitu *facebook*. Sanksi yang akan diberikan kepada pelanggar Undang - Undang ITE atau *hate speech* yaitu terdapat pada UU No 11 Tahun 2018 tentang ITE Pasal 28 ayat (2) & Pasal 45 ayat (2) [4].

Akan tetapi, sanksi yang diberikan kepada para pelanggar belum merata karena banyaknya jumlah komentar dari masyarakat sehingga diperlukan tindakan untuk mengidentifikasi *hate speech* agar mampu membedakan antara ujaran kebencian dan bukan ujaran kebencian untuk dapat menunjukkan kekurangan masyarakat Karawang dalam berkomentar di *facebook*. Dengan menganalisis ini bermanfaat untuk memudahkan pihak bawaslu dalam memberikan sanksi kepada para pelanggar dan dapat memberikan efek jera kepada masyarakat karawang untuk tidak melakukan tindakan ujaran kebencian di tahun - tahun berikutnya.

Banyaknya dokumen yang akan di identifikasi untuk membentuk kelas – kelas berdasarkan kelompok yang sudah diketahui sebelumnya tidak mungkin dilakukan secara manual. Maka, metode yang akan digunakan dalam mengklasifikasikan ujaran kebencian ini adalah *text mining*. *Text mining* digunakan untuk menghasilkan informasi dari media sosial yang tidak terstruktur [5].

Penelitian sebelumnya yaitu tentang Optimasi Algoritma SVM dan K-NN Berbasis PSO Pada Analisis Sentimen Fenomena Tagar #2019GantiPresiden [6]. Hasil akurasi yang didapat dengan menggunakan algoritma SVM 88,00% sedangkan hasil akurasi SVM yang menggunakan PSO mendapatkan 92,75% dan hasil akurasi yang didapat dengan menggunakan algoritma KNN 88,50% sedangkan hasil akurasi KNN yang menggunakan PSO mendapatkan 75,25%. Dengan menggunakan optimasi PSO pada algoritma SVM dapat meningkatkan nilai akurasi sedangkan

optimasi PSO pada algoritma KNN mengalami penurunan sebesar 13%.

Selain itu ada juga penelitian yang Menganalisis Sentimen Review Aplikasi Menggunakan Algoritma *Support Vector Machine* [7]. Hasil yang didapat dengan menguji kernel *linear*, *radial basis function* dan *polynomial* yaitu kernel *linear* yang mendapatkan akurasi tertinggi sebesar 0,897.

Berdasarkan penjelasan di atas dan melihat dari penelitian – penelitian sebelumnya yang menyatakan bahwa optimasi penerapan *Particle Swarm Optimization* dengan algoritma SVM berhasil memperbaiki akurasi menjadi optimal dibandingkan dengan algoritma lainnya, maka akan dilakukan penelitian dengan judul “Optimasi *Support Vector Machine* Berbasis *Particle Swarm Optimization* Untuk Mendeteksi *Hate Speech* Pilkada Karawang 2020”.

II. TINJAUAN PUSTAKA

A. Text Mining

Text mining yaitu proses mengolah teks tidak terstruktur yang diambil dari media sosial seperti *text email*, *text html*, *text* komentar dari berbagai sumber. *Text mining* masih masuk ke dalam bagian data *mining*, dengan melakukan proses pada data atau *text* serta dokumen dalam jumlah yang besar [8].

B. Knowledge Discovery in Database (KDD)

KDD adalah proses mengolah data mentah kemudian diolah dengan beberapa proses sehingga dapat menghasilkan informasi yang berguna. *Knowledge Discovery in Database* (KDD) terdiri dari 5 tahapan yakni *Data Selection*, *Preprocessing*, *Transformation*, *Modelling* dan *Evaluation* [9].

C. Support Vector Machine (SVM)

Konsep dari *support vector machine* awalnya berasal dari masalah klasifikasi dua kelas kemudian memerlukan data pengujian positif dan negatif maka dari itu *support vector machine* digunakan untuk memisahkannya menjadi dua kelas dengan memaksimalkan *margin*. *Margin* digunakan untuk memberi jarak antara titik positif dan negatif terdekat di *hyperlane*, *hyperlane* dapat berupa garis lurus untuk 2 dimensi atau bidang datar untuk 3 dimensi dimensi [10].

Penggunaan kernel dalam *support vector machine* itu penting, karena dengan menggunakan fungsi kernel dapat meningkatkan kinerja. Fungsi kernel dapat dikatakan sebagai fungsi yang menggambarkan data dengan struktur yang baik sehingga mudah untuk dipisahkan. Terdapat 4 fungsi kernel yang sering digunakan yaitu Kernel *Linier*, *Radial Basis Function* (RBF), *Sigmoid* dan *Polynomial* [11].

D. Particle Swarm Optimization

PSO didasarkan pada perilaku sekawanan ikan dan burung dalam mempertahankan hidup. Di dalam PSO terdapat *particle* dengan ukuran yang berbeda dan awalnya terletak di lokasi yang acak dalam ruang. Terdapat dua ciri khas di

dalam PSO diantaranya posisi dan kecepatan, jika sebuah *particle* mendapatkan jalan yang benar dan cepat ketika menuju ke sumber makanan, *particle* tersebut akan memberi tahu kepada *particle* lain untuk segera mengikuti jalan tersebut walaupun lokasinya berjauhan. Populasi acak yang ada pada PSO biasa disebut *particle*, setiap *particle* yang ada pada PSO berkaitan dengan yang namanya *velocity* berguna untuk memindahkan *particle* ke lokasi yang lebih baik dari sebelumnya [12].

E. Confusion Matrix

Confusion matrix adalah metode konsep data mining untuk melakukan perhitungan akurasi. Berikut representasi nilai hasil klasifikasi diantaranya *True Positif* (TP), *True Negatif* (TN), *False Positif* (FP), *False Negatif* (FN) [13].

Dengan menghitung *confusion matrix* nilai yang didapatkan diantaranya:

- 1) Akurasi
Akurasi digunakan untuk tingkat ketepatan nilai prediksi dan aktual.
- 2) Presisi
Presisi digunakan sebagai perbandingan jumlah data positif dengan jumlah dari data positif bernilai benar dan data negatif yang bernilai positif.
- 3) Recall
Recall digunakan sebagai perbandingan jumlah data positif dengan jumlah dari data positif bernilai benar dan data yang bernilai negatif.
- 4) F1-score
F1-score digunakan sebagai alat untuk mengukur kinerja sistem dengan cara menghitung nilai presisi dan recall lalu dikali dua kemudian dibagi dengan hasil penjumlahan dari presisi dan recall.

III. METODE PENELITIAN

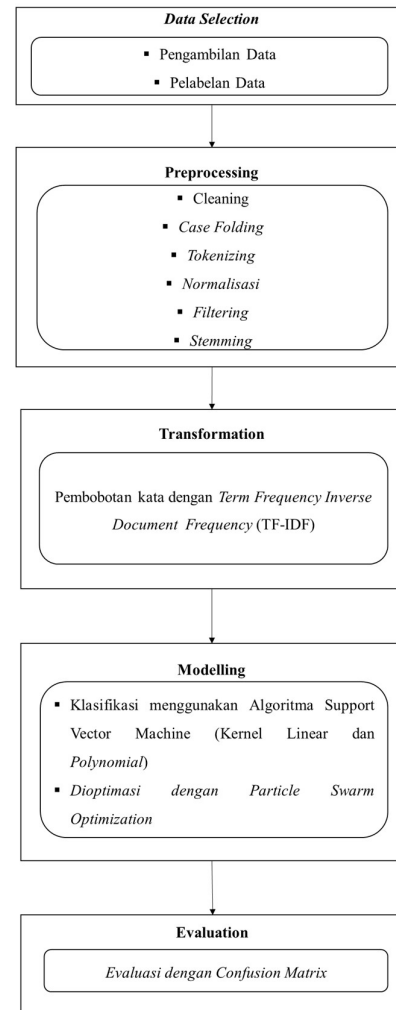
A. Metodologi Penelitian

Metodologi yang digunakan untuk mendeteksi *hate speech* pilkada karawang 2020 adalah *Knowledge Discovery in Database* dengan algoritma *Support Vector Machine* berbasis *Particle Swarm Optimization*. Data yang digunakan dalam penelitian ini diambil dari beberapa postingan di Grup Facebook Karawang Info dari bulan Juli – Desember 2020. Kemudian data diberi label secara manual lalu diverifikasi oleh seorang ahli bahasa dan sastra Indonesia dan penelitian ini menggunakan bahasa pemrograman *python*. Berikut tahapan dari *Knowledge Discovery in Database* diantaranya *Data Selection*, *Preprocessing* terdiri dari *Cleaning*, *Case Folding*, *Tokenizing*, *Normalisasi*, *Filtering* dan *Stemming*, kemudian pada tahap *Transformation* menggunakan TF-IDF. Tahapan *Modeling* ini dilakukan dengan dua model yaitu algoritma *Support Vector Machine* dan algoritma *Support Vector Machine* dioptimasi dengan algoritma *Particle Swarm Optimization*. Penelitian ini menggunakan dua kernel

yaitu kernel *linear* dan *polynomial* dan tahapan *evaluation* untuk mendapatkan nilai akurasi, presisi, *recall* dan *f1-score*.

B. Rancangan Penelitian

Berikut rancangan penelitian yang akan dilakukan seperti yang terdapat pada Gambar 2.



Gambar 2. Rancangan Penelitian

Berikut penjelasan dari Gambar 2.

1) Data Selection

Penelitian ini menggunakan data yang di *crawling* dari bulan Juli – Desember 2020 yang diambil dari beberapa postingan di Grup Facebook Karawang Info. Kemudian data tersebut diklasifikasi ke dalam 2 *class* yaitu Positif dan Negatif yang dilakukan secara manual kemudian diverifikasi oleh seorang ahli bahasa dan sastra Indonesia.

2) Preprocessing

Preprocessing terdapat beberapa tahapan diantaranya

- *Cleaning* digunakan untuk proses membersihkan data yang tidak memiliki pengaruh pada proses data *mining*.
- *Case Folding* bertujuan untuk merubah semua huruf kapital dalam dokumen menjadi huruf kecil.
- *Tokenizing* digunakan untuk memotong kalimat menjadi sebuah kata berdasarkan spasi.
- *Normalisasi* digunakan untuk memperbaiki penulisan kata yang tidak sesuai menjadi sesuai dengan kamus KBBI.
- *Filtering* digunakan untuk menghilangkan kata - kata yang tidak bermakna atau kurang penting dengan topik penelitian.
- *Stemming* digunakan untuk menghilangkan kata imbuhan dengan menggunakan *library sastrawi*.

3) Transformation

Tahapan ini memberikan bobot setiap kata untuk melihat seberapa jauh hubungan kata terhadap dokumen. Bobot dihitung dengan menggunakan *Term Frequency Inverse Document Frequency* (TF-IDF).

4) Modelling

Pada tahapan ini data yang telah ditransformasikan kemudian diuji dengan membagi menjadi data latih dan data uji, *splitting data* dilakukan dengan 4 skenario.

5) Evaluation

Pada tahapan ini dilakukan evaluasi dengan menguji *confusion matrix*nya untuk melihat nilai akurasi, presisi, *recall* dan *f1-score*.

III. HASIL DAN PEMBAHASAN

A. Hasil Penelitian

Penelitian yang dilakukan yaitu bagaimana mendeteksi *hate speech* pilkada Karawang 2020 dengan algoritma *support vector machine* dan kemudian dioptimasi menggunakan *particle swarm optimization*.

1) Data Selection

Pada penelitian ini data yang digunakan yaitu data komentar yang berasal dari grup facebook Karawang Info dengan kata kunci “Bupati”, “Pilkada” dan “Pilkada Karawang”. Data diambil secara manual dengan teknik *crawling* sebanyak 546 menggunakan *software exportcomments.com* dan disimpan dalam file *xlsx*.

Data yang telah dikumpulkan kemudian diseleksi yang awalnya memiliki 8 variabel yaitu *name*, *profile id*, *date*, *likes*, *stars*, *coment*, *view source* dan *class*, menjadi hanya *comment* dan *class* yang akan digunakan. Pilkada tahun ini diselenggarakan pada 9 Desember 2020 maka data yang akan

digunakan mulai dari tanggal 1 Juli 2020 – 8 Desember 2020 sebanyak 511 data. Kemudian data dibagi menjadi dua bagian lalu diberi label secara manual dan diverifikasi oleh ahli bahasa dan sastra Indonesia, didapatkan 188 komentar positif dan komentar negatif sebanyak 323. Berikut contoh data yang telah diberi label dan divalidasi dapat dilihat pada Tabel I.

TABEL I
CONTOH DATASET YANG TELAH DIBERI LABEL DAN DIVALIDASI

No	Komentar	Label
1	Lihat jok wargamu,separtu kita dulu pilpres ya...	Negatif
2	Ganti bupati	Positif
3	Cakeep	Positif
4	#2021 kang jimmy bupati Karawang	Positif
5	masih lama, kerja weh dulu, masih banyak raky...	Positif
6	Mantaaapp\n[PHOTO] https://external.fkdt1-1.fn...	Positif
7	Adug adugan pengen menang... Segala cara di j...	Positif
8	saya mah menang meren nomor 3	Positif
9	☐ #JANGANLUPA SEBELUM TGL 9 ☐\n[PHOTO] https://...	Positif
10	Gaskeun..👇	Positif

2) Preprocessing

Tahap *Preprocessing* dilakukan sebelum tahap *transformation* dan *modelling* untuk memastikan data telah siap diolah. *Preprocessing* terdiri dari beberapa tahapan yaitu:

- *Cleaning*: Tahapan data *cleaning* berguna untuk menghapus karakter seperti emotikon, URL, tagar, simbol, angka dll. Berikut contoh dari tahapan data *cleaning*.

TABEL II
CONTOH SEBELUM CLEANING

No	Komentar
1	Lihat jok wargamu,separtu kita dulu pilpres ya...
2	Ganti bupati
3	Cakeep
4	#2021 kang jimmy bupati Karawang
5	masih lama, kerja weh dulu, masih banyak raky...
6	Mantaaapp\n[PHOTO]https://external.fkdt11.fn...
7	Adug adugan pengen menang.... Segala cara di j...
8	saya mah menang meren nomor 3
9	☐ #JANGANLUPA SEBELUM TGL 9 ☐\n[PHOTO] https://...
10	Gaskeun..👇

TABEL II
CONTOH SESUDAH CLEANING

No	Komentar
1	Lihat jok wargamusepartai kita dulu pilpres yag...
2	Ganti bupati
3	Cakep
4	kang jimmy bupati Karawang
5	masih lama kerja weh dulu masih banyak rakyat ...
6	Mantap PHOTO
7	Adug adugan pengen menang Segala cara di jalan...
8	saya mah menang meren nomor
9	SEBELUM TGL PHOTO
10	Gaskeun

- *Case folding*: Tahapan *case folding* digunakan untuk menyeragamkan kata yang semula berbentuk huruf kapital dirubah menjadi huruf non kapital atau kecil. Berikut contoh dari tahapan *case folding*.

TABEL III
CONTOH SEBELUM CASE FOLDING

No	Komentar
1	Lihat jok wargamusepartai kita dulu pilpres yag...
2	Ganti bupati
3	Cakep
4	kang jimmy bupati Karawang
5	masih lama kerja weh dulu masih banyak rakyat ...
6	Mantap PHOTO
7	Adug adugan pengen menang Segala cara di jalan.
8	saya mah menang meren nomor
9	SEBELUM TGL PHOTO
10	Gaskeun

TABEL IV
CONTOH SESUDAH CASE FOLDING

No	Komentar
1	lihat jok wargamusepartai kita dulu pilpres yag...
2	ganti bupati
3	cakep
4	kang jimmy bupati karawang
5	masih lama kerja weh dulu masih banyak rakyat ...
6	mantap photo
7	adug adugan pengen menang segala cara di jalan...
8	saya mah menang meren nomor
9	sebelum tgl photo
10	gaskeun

- *Tokenizing*: Tahapan ini dilakukan untuk memecah kalimat menjadi sebuah kata berdasarkan spasi. Berikut contoh dari tahapan *tokenizing*.

TABEL V
CONTOH SEBELUM TOKENIZING

No	Komentar
1	lihat jok wargamusepartai kita dulu pilpres yag...
2	ganti bupati
3	cakep

4	kang jimmy bupati karawang
5	masih lama kerja weh dulu masih banyak rakyat ...
6	mantap photo
7	adug adugan pengen menang segala cara di jalan...
8	saya mah menang meren nomor
9	sebelum tgl photo
10	gaskeun

TABEL VI
CONTOH SESUDAH TOKENIZING

No	Komentar
1	[lihat, jok, wargamusepartai, kita, dulu, pilpr...
2	[ganti, bupati]
3	[cakep]
4	[kang, jimy, bupati, karawang]
5	[masih, lama, kerja, weh, dulu, masih, banyak,...
6	[mantap, photo]
7	[adug, adugan, pengen, menang, segala, cara, d...
8	[saya, mah, menang, meren, nomor]
9	[sebelum, tgl, photo]
10	[gaskeun]

- *Normalisasi*: Tahapan ini untuk memperbaiki kata – kata yang belum sesuai dengan kamus KBBI. Berikut contoh dari tahapan *normalisasi*.

TABEL VII
CONTOH SEBELUM NORMALISASI

No	Komentar
1	[lihat, jok, wargamusepartai, kita, dulu, pilpr...
2	[ganti, bupati]
3	[cakep]
4	[kang, jimy, bupati, karawang]
5	[masih, lama, kerja, weh, dulu, masih, banyak,...
6	[mantap, photo]
7	[adug, adugan, pengen, menang, segala, cara, d...
8	[saya, mah, menang, meren, nomor]
9	[sebelum, tgl, photo]
10	[gaskeun]

TABEL VIII
CONTOH SESUDAH NORMALISASI

No	Komentar
1	[lihat, jokowi, warga, kamu, seperti, kita, du...
2	[ganti, bupati]
3	[cakep]
4	[kakak, laki, laki, jimmy, bupati, karawang]
5	[masih, lama, kerja, weh, dulu, masih, banyak,...
6	[mantap, foto]
7	[meronta, ronta, meronta, ronta, mau, menang, ...
8	[saya, mah, menang, kali, nomor]
9	[sebelum, tanggal, foto]
10	[lanjut]

- *Filtering*: *filtering* ini untuk menghilangkan kata – kata yang tidak diperlukan dalam topik penelitian ini. Contoh kata yang dihapus diantaranya “dari”,

“dan”, “di”, “ke”, “dengan”, “ini”, “jadi” dll. Berikut hasil dari tahapan *filtering*.

TABEL IX
CONTOH SEBELUM FILTERING

No	Komentar
1	[lihat, jokowi, warga, kamu, seperti, kita, du...
2	[ganti, bupati]
3	[cakep]
4	[kakak, laki, laki, jimmy, bupati, karawang]
5	[masih, lama, kerja, weh, dulu, masih, banyak,...
6	[mantap, foto]
7	[meronta, ronta, meronta, ronta, mau, menang, ...
8	[saya, mah, menang, kali, nomor]
9	[sebelum, tanggal, foto]
10	[lanjut]

TABEL X
CONTOH SESUDAH FILTERING

No	Komentar
1	[lihat, warga, kamu, dulu, pemilihan, presiden...
2	[ganti, bupati]
3	[cakep]
4	[laki, laki, jimmy, bupati, karawang]
5	[lama, kerja, dulu, banyak, rakyat, membutuhka...
6	[mantap, foto]
7	[meronta, ronta, meronta, ronta, menang, segal...
8	[menang, nomor]
9	[tanggal, foto]
10	[lanjut]

- *Stemming*: Proses *stemming* menggunakan *library* Sastrawi yang digunakan untuk menghilangkan semua imbuhan ke dalam bentuk kata dasar. Contohnya seperti mem-, -kan, -an, pe-, ke-, -nya dan yang lainnya. Berikut contoh dari tahapan *stemming*.

TABEL XI
CONTOH SEBELUM STEMMING

No	Komentar
1	[lihat, warga, kamu, dulu, pemilihan, presiden...
2	[ganti, bupati]
3	[cakep]
4	[laki, laki, jimmy, bupati, karawang]
5	[lama, kerja, dulu, banyak, rakyat, membutuhka...
6	[mantap, foto]
7	[meronta, ronta, meronta, ronta, menang, segal...
8	[menang, nomor]
9	[tanggal, foto]
10	[lanjut]

TABEL XII
CONTOH SESUDAH STEMMING

No	Komentar
1	[lihat, warga, kamu, dulu, pilih, presiden, ke...
2	[ganti, bupati]
3	[cakep]

4	[laki, laki, jimmy, bupati, karawang]
5	[lama, kerja, dulu, banyak, rakyat, butuh, ulu...
6	[mantap, foto]
7	[ronta, ronta, ronta, ronta, menang, segala, c...
8	[menang, nomor]
9	[tanggal, foto]
10	[lanjut]

3) *Transformation*

Tahapan ini menggunakan *Term Frequency-Inverse Document Frequency (TF-IDF)* untuk pembobotannya agar setiap kata dapat direpresentasikan menjadi *numerik* atau angka. Pertama menghitung jumlah kata yang muncul pada setiap komentar. Berikut contoh jumlah kata pada setiap komentar dapat dilihat pada Gambar 3.

```
[{'anak': 1,
'dulu': 2,
'jawab': 1,
'jokowi': 2,
'kamu': 1,
'kelah': 1,
'lihat': 1,
'pilih': 2,
'presiden': 2,
'sahut': 1,
'warga': 1},
{'bupati': 1, 'ganti': 1},
{'cakep': 1}]
```

Gambar 3. Contoh jumlah kata pada setiap komentar

Berikut contoh hasil dari nilai *Term Frequency (TF)* dapat dilihat pada Gambar 4.

```
[{'anak': 0.06666666666666667,
'dulu': 0.13333333333333333,
'jawab': 0.06666666666666667,
'jokowi': 0.13333333333333333,
'kamu': 0.06666666666666667,
'kelah': 0.06666666666666667,
'lihat': 0.06666666666666667,
'pilih': 0.13333333333333333,
'presiden': 0.13333333333333333,
'sahut': 0.06666666666666667,
'warga': 0.06666666666666667},
{'bupati': 0.5, 'ganti': 0.5},
{'cakep': 1.0}]
```

Gambar 4. Contoh Hasil TF

Berikut contoh hasil dari nilai *Inverse Document Frequency (IDF)* dapat dilihat pada Gambar 5.


```
{'aamiin': 3.7236833763086286,
'abai': 5.5154428455366835,
'abal': 6.208590026096629,
'absen': 6.208590026096629,
'adil': 4.599152113662528,
'adly': 5.5154428455366835,
'admin': 6.208590026096629,
'adu': 4.822295664976738,
'aep': 4.4168305568685735,
'agam': 6.208590026096629,
'air': 4.822295664976738,
'ajak': 5.5154428455366835,
'ajar': 5.109977737428519,
'aju': 6.208590026096629,
'ajuin': 6.208590026096629,
```

Gambar 5. Contoh Hasil IDF

Sedangkan contoh hasil dari nilai *Term Frequency-Inverse Document Frequency* (TF-IDF) dapat dilihat pada Gambar 6.

	aamiin	abai	abal	absen	adil	adly	admin	adu	aep	agam
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows x 1005 columns

Gambar 6. Contoh Hasil TF-IDF

4) Modelling

Setelah dilakukan *preprocessing* dan *transformation*, kemudian data dibagi menjadi dua yaitu data *training* dan *testing* dengan empat perbandingan yaitu 60:40, 70:30, 80:20, 90:10. Tahapan ini dilakukan dengan dua pemodelan yaitu dengan algoritma *support vector machine* dan algoritma *support vector machine* (SVM) dioptimasi dengan algoritma *particle swarm optimization* (PSO). Penelitian ini menggunakan dua kernel yaitu kernel *linear* dan *polynomial* dan parameternya yakni *c*, *gamma*, *degree*, *coef0*, *tol*, *cache size*. Fungsi yang digunakan pada *particle swarm optimization* yaitu dengan iterasi 50, partikel 10 dan dimensi yang digunakan sebanyak 6 parameter. Berikut hasil pengujian yang telah dilakukan:

- Skenario 1 (60:40) : Berikut hasil pengujian yang telah dilakukan pada pemodelan klasifikasi komentar *hate speech* menggunakan algoritma SVM dengan kernel *linear* dan *polynomial* dapat dilihat pada Gambar 7 dan Gambar 8.

✓ Pengujian SVM dengan kernel linear

	predicted:negatif	predicted:positif		
actual:negatif	26	40		
actual:positif	16	117		
	precision	recall	f1-score	support
0	0.61905	0.39394	0.48148	66
1	0.74522	0.87970	0.80690	133
accuracy			0.71859	199
macro avg	0.68214	0.63682	0.64419	199
weighted avg	0.70338	0.71859	0.69897	199

Gambar 7. Hasil pengujian skenario 1 SVM kernel linear

✓ Pengujian SVM dengan kernel polynomial

	predicted:negatif	predicted:positif		
actual:negatif	3	63		
actual:positif	4	129		
	precision	recall	f1-score	support
0	0.42857	0.04545	0.08219	66
1	0.67188	0.96992	0.79385	133
accuracy			0.66332	199
macro avg	0.55022	0.50769	0.43802	199
weighted avg	0.59118	0.66332	0.55782	199

Gambar 8. Hasil pengujian skenario 1 SVM kernel *polynomial*

Berikut hasil pengujian yang telah dilakukan pada pemodelan kedua untuk klasifikasi komentar *hate speech* menggunakan algoritma SVM dan dioptimasi PSO dengan kernel *linear* dan *polynomial* dapat dilihat pada Gambar 9 dan Gambar 10.

✓ Pengujian SVM dan PSO dengan kernel linear

	predicted:negatif	predicted:positif		
actual:negatif	33	33		
actual:positif	17	116		
	precision	recall	f1-score	support
0	0.66000	0.50000	0.56897	66
1	0.77852	0.87218	0.82270	133
accuracy			0.74874	199
macro avg	0.71926	0.68609	0.69583	199
weighted avg	0.73921	0.74874	0.73854	199

Gambar 9. Hasil pengujian skenario 1 SVM dan PSO kernel *linear*

✓ *Pengujian SVM dan PSO dengan kernel polynomial*

	predicted:negatif	predicted:positif		
actual:negatif	35	31		
actual:positif	18	115		
	precision	recall	f1-score	support
0	0.66038	0.53030	0.58824	66
1	0.78767	0.86466	0.82437	133
accuracy			0.75377	199

Gambar 10. Hasil pengujian skenario 1 SVM dan PSO kernel *polynomial*

- Skenario 2 (70:30): Berikut hasil pengujian yang telah dilakukan pada pemodelan klasifikasi komentar *hate speech* menggunakan algoritma *SVM* dengan kernel *linear* dan *polynomial* dapat dilihat pada Gambar 11 dan Gambar 12.

✓ *Pengujian SVM dengan kernel linear*

	predicted:negatif	predicted:positif		
actual:negatif	19	30		
actual:positif	11	90		
	precision	recall	f1-score	support
0	0.63333	0.38776	0.48101	49
1	0.75000	0.89109	0.81448	101
accuracy			0.72667	150

Gambar 11. Hasil pengujian skenario 2 SVM kernel *linear*

✓ *Pengujian SVM dengan kernel polynomial*

	predicted:negatif	predicted:positif		
actual:negatif	3	46		
actual:positif	3	98		
	precision	recall	f1-score	support
0	0.50000	0.06122	0.10909	49
1	0.68056	0.97030	0.80000	101
accuracy			0.67333	150

Gambar 12. Hasil pengujian skenario 2 SVM kernel *polynomial*

Berikut hasil pengujian yang telah dilakukan pada pemodelan kedua untuk klasifikasi komentar *hate speech* menggunakan algoritma *SVM* dan dioptimasi *PSO* dengan kernel *linear* dan *polynomial* dapat dilihat pada Gambar 13 dan Gambar 14.

✓ *Pengujian SVM dan PSO dengan kernel linear*

	predicted:negatif	predicted:positif		
actual:negatif	18	31		
actual:positif	7	94		
	precision	recall	f1-score	support
0	0.72000	0.36735	0.48649	49
1	0.75200	0.93069	0.83186	101
accuracy			0.74667	150

Gambar 13. Hasil pengujian skenario 2 SVM dan PSO kernel *linear*

✓ *Pengujian SVM dan PSO dengan kernel polynomial*

	predicted:negatif	predicted:positif		
actual:negatif	19	30		
actual:positif	7	94		
	precision	recall	f1-score	support
0	0.73077	0.38776	0.50667	49
1	0.75806	0.93069	0.83556	101
accuracy			0.75333	150

Gambar 14. Hasil pengujian skenario 2 SVM dan PSO kernel *polynomial*

- Skenario 3 (80:20): Berikut hasil pengujian yang telah dilakukan pada pemodelan klasifikasi komentar *hate speech* menggunakan algoritma *SVM* dengan kernel *linear* dan *polynomial* dapat dilihat pada Gambar 15 dan Gambar 16.

✓ *Pengujian SVM dengan kernel linear*

	predicted:negatif	predicted:positif		
actual:negatif	11	24		
actual:positif	5	60		
	precision	recall	f1-score	support
0	0.68750	0.31429	0.43137	35
1	0.71429	0.92308	0.80537	65
accuracy			0.71000	100

Gambar 15. Hasil pengujian skenario 3 SVM kernel *linear*

✓ *Pengujian SVM dengan kernel polynomial*

	predicted:negatif	predicted:positif		
actual:negatif	2	33		
actual:positif	1	64		
	precision	recall	f1-score	support
0	0.66667	0.05714	0.10526	35
1	0.65979	0.98462	0.79012	65
accuracy			0.66000	100

Gambar 16. Hasil pengujian skenario 3 SVM kernel *polynomial*

Berikut hasil pengujian yang telah dilakukan pada pemodelan kedua untuk klasifikasi komentar *hate speech* menggunakan algoritma SVM dan dioptimasi PSO dengan kernel *linear* dan *polynomial* dapat dilihat pada Gambar 17 dan Gambar 18.

✓ *Pengujian SVM dan PSO dengan kernel linear*

	predicted:negatif	predicted:positif		
actual:negatif	15	20		
actual:positif	6	59		
	precision	recall	f1-score	support
0	0.71429	0.42857	0.53571	35
1	0.74684	0.90769	0.81944	65
accuracy			0.74000	100

Gambar 17. Hasil pengujian skenario 3 SVM dan PSO kernel *linear*

✓ *Pengujian SVM dan PSO dengan kernel polynomial*

	predicted:negatif	predicted:positif		
actual:negatif	18	17		
actual:positif	7	58		
	precision	recall	f1-score	support
0	0.72000	0.51429	0.60000	35
1	0.77333	0.89231	0.82857	65
accuracy			0.76000	100

Gambar 18 Hasil pengujian skenario 3 SVM dan PSO kernel *polynomial*

- Skenario 4 (90:10) : Berikut hasil pengujian yang telah dilakukan pada pemodelan klasifikasi komentar *hate speech* menggunakan algoritma SVM dengan kernel *linear* dapat dilihat pada Gambar 19 dan Gambar 20.

✓ *Pengujian SVM dengan kernel linear*

	predicted:negatif	predicted:positif		
actual:negatif	5	11		
actual:positif	3	31		
	precision	recall	f1-score	support
0	0.62500	0.31250	0.41667	16
1	0.73810	0.91176	0.81579	34
accuracy			0.72000	50

Gambar 19. Hasil pengujian skenario 4 SVM kernel *linear*

✓ *Pengujian SVM dengan kernel polynomial*

	predicted:negatif	predicted:positif		
actual:negatif	2	14		
actual:positif	1	33		
	precision	recall	f1-score	support
0	0.66667	0.12500	0.21053	16
1	0.70213	0.97059	0.81481	34
accuracy			0.70000	50

Gambar 20. Hasil pengujian skenario 4 SVM kernel *polynomial*

Berikut hasil pengujian yang telah dilakukan pada pemodelan kedua untuk klasifikasi komentar *hate speech* menggunakan algoritma SVM dan dioptimasi PSO dengan kernel *linear* dapat dilihat pada Gambar 21 dan Gambar 22.

✓ *Pengujian SVM dan PSO dengan kernel linear*

	predicted:negatif	predicted:positif		
actual:negatif	8	8		
actual:positif	3	31		
	precision	recall	f1-score	support
0	0.72727	0.50000	0.59259	16
1	0.79487	0.91176	0.84932	34
accuracy			0.78000	50

Gambar 21. Hasil pengujian skenario 4 SVM dan PSO kernel *linear*

✓ *Pengujian SVM dan PSO dengan kernel polynomial*

	predicted:negatif	predicted:positif		
actual:negatif	8	8		
actual:positif	3	31		
	precision	recall	f1-score	support
0	0.72727	0.50000	0.59259	16
1	0.79487	0.91176	0.84932	34
accuracy			0.78000	50

Gambar 22. Hasil pengujian skenario 4 SVM dan PSO kernel *polynomial*

5) Evaluation

Pengujian yang telah dilakukan akan dibandingkan untuk mengetahui model mana memiliki kinerja terbaik antara *support vector machine* dan *support vector machine* dioptimasi dengan algoritma *particle swarm optimization*. Berikut hasil perbandingan evaluasi dari setiap skenario dengan menggunakan *support vector machine* dapat dilihat pada Tabel di bawah ini.

TABEL VII
HASIL PERBANDINGAN EVALUASI SUPPORT VECTOR MACHINE

Skenario	Kernel	Akurasi (%)			
Skenario 1	Linear	71.85			
	Polynomial	66.33			
Skenario 2	Linear	72.66			
	Polynomial	67.33			
Skenario 3	Linear	71.00			
	Polynomial	66.00			
Skenario 4	Linear	72.00			
	Polynomial	70.00			
Presisi (%)		Recall (%)	F1-score (%)		
0	1	0	1		
61.90	74.52	39.39	87.97	48.14	80.69
42.85	67.18	04.54	96.99	08.21	79.38
63.33	75.00	38.77	89.10	48.10	81.44
50.00	68.05	06.12	97.03	10.90	80.00
68.75	71.42	31.42	92.38	43.13	80.53
66.66	65.97	05.71	98.46	10.05	79.01
62.50	73.81	31.25	91.17	41.66	81.57
66.66	70.21	12.50	97.05	21.05	81.48

Berdasarkan tabel di atas dapat dilihat bahwa hasil yang memiliki nilai akurasi tertinggi yaitu kernel *linear* sebesar 72.66% pada skenario ke 2 dengan pembagian data 70:30. Sedangkan untuk yang memiliki nilai akurasi terkecil yaitu pada kernel *polynomial* sebesar 66.00% pada skenario 80:20. Berikut hasil evaluasi dari 4 skenario dengan menggunakan *support vector machine* dioptimasi dengan algoritma *particle swarm optimization* dapat dilihat pada Tabel di bawah ini.

TABEL VIII
HASIL PERBANDINGAN EVALUASI SUPPORT VECTOR MACHINE DENGAN PARTICLE SWARM OPTIMIZATION

Skenario	Kernel	Akurasi (%)			
Skenario 1	Linear	74.87			
	Polynomial	75.37			
Skenario 2	Linear	74.66			
	Polynomial	75.33			
Skenario 3	Linear	74.00			
	Polynomial	76.00			
Skenario 4	Linear	78.00			
	Polynomial	78.00			
Presisi (%)		Recall (%)	F1-score (%)		
0	1	0	1		
66.66	77.48	48.48	87.97	56.14	82.39
66.03	78.76	53.03	86.46	58.82	82.43
72.00	75.20	36.73	93.06	48.64	83.18

73.07	75.80	38.77	93.06	50.66	83.55
71.42	74.68	42.85	90.76	53.57	81.94
72.00	77.33	51.42	89.23	60.00	82.85
72.72	79.48	50.00	91.17	59.25	84.93
72.72	79.48	50.00	91.17	59.25	84.93

Berdasarkan tabel di atas dapat dilihat bahwa hasil optimasi algoritma *particle swarm optimization* pada 4 skenario menyatakan bahwa yang memiliki nilai akurasi tertinggi yaitu kernel *linear* dan *polynomial* sebesar 78.00% pada skenario ke 4 dengan pembagian data 90:10. Nilai evaluasi yang lain pun mengalami kenaikan mulai dari presisi menjadi 72.72%, *recall* menjadi 50.00% dan *f1-score* menjadi 59.25%. Sedangkan untuk yang memiliki nilai akurasi terkecil yaitu pada kernel *polynomial* sebesar 74.00% pada skenario ke 3 dengan pembagian data 80:20.

B. Pembahasan

Penelitian ini mengklasifikasikan tentang komentar Pilkada Karawang 2020 dengan SVM berbasis *Particle Swarm Optimization* (PSO). Metodologi penelitiannya KDD yang terdiri dari beberapa tahapan yakni *Data Selection*, *Preprocessing*, *Transformation*, *Modelling* dan *Evaluation*. Data diambil secara manual dengan sebanyak 546 komentar menggunakan *software exportcomments.com* dan disimpan dalam file *xlsx*. Kemudian data diseleksi menjadi 511 data yang digunakan dan diberi label secara manual lalu diverifikasi oleh ahli bahasa dan sastra Indonesia. Pada tahapan *Preprocessing* data tersebut diolah dengan beberapa tahapan diantaranya *Cleaning* (membersihkan *noisy*), *Case Folding* (merubah semua huruf menjadi kecil), *Tokenizing* (memecah kalimat menjadi perkata), *Normalisasi* (memperbaiki penulisan kata), *Filtering* (menghapus kata - kata yang tidak bermakna) dan *Stemming* (menghilangkan kata imbuhan). Selanjutnya pada tahap *Transformation* menggunakan *Term Frequency Inverse Document Frequency* (TF-IDF). Pada *Modelling*, mengimplementasikan dua model yaitu dengan algoritma *support vector machine* dan algoritma *support vector machine* (SVM) dioptimasi dengan algoritma *particle swarm optimization* (PSO) dan tahapan terakhir yaitu *evaluation* untuk menampilkan nilai dari hasil *modelling* diantaranya akurasi, presisi, *recal* dan *f1-score*.

Penelitian ini melakukan pengujian dengan 4 skenario pembagian data yaitu 60:40, 70:30, 80:20 dan 90:10. Berdasarkan skenario tersebut dapat dinyatakan bahwa pada pemodelan pertama skenario 2 mendapatkan nilai akurasi tertinggi sebesar 72.66% dengan kernel *linear*. Kemudian 4 skenario tersebut dioptimasi menggunakan *Particle Swarm Optimization* dengan parameter iterasi 50, partikel 10 didapatkan bahwa nilai akurasi tertinggi yaitu kernel *linear* dan *polynomial* sebesar 78.00% pada skenario ke 4 dengan pembagian data 90:10. Nilai evaluasi yang lain pun mengalami kenaikan mulai dari presisi menjadi 72.72%, *recall* menjadi 50.00% dan *f1-score* menjadi 59.25%. Nilai

performa terbaik yang di peroleh dari algoritma *support vector machine* dan algoritma *support vector machine* dengan *particle swarm optimization* memiliki selisih akurasi sebesar 5.34%.

Berikut hasil pencarian kata yang sering muncul pada *class* positif dan negatif dapat dilihat pada Gambar 23 dan Gambar 24.



Gambar 23. Wordcloud positif

Gambar 23 dapat diketahui bahwa kata yang sering muncul pada *class* positif yaitu kata “karawang”, “jangan”, “bupati” dan “laki – laki”.



Gambar 24. Wordcloud negatif

Sedangkan dari Gambar 24 dapat diketahui bahwa kata – kata yang sering muncul pada *class* negatif seperti “laki – laki”, “bupati”, “karawang” merupakan kata yang sering digunakan pengguna facebook dalam menunjukan bahwa masyarakat ingin yang menjadi bupati karawang ini berjenis kelamin laki – laki.

IV. HASIL DAN PEMBAHASAN

Dengan metodologi *Knowledge Discovery in Database* dan algoritma *Support Vector Machine* lalu dioptimasi dengan *Particle Swarm Optimization* terbukti dapat mengklasifikasi komentar *hate speech* Pilkada Karawang 2020. Dalam penelitian ini pemodelan pertama menggunakan algoritma *Support Vector Machine* yang mendapatkan nilai performa terbaik terdapat pada kernel *linear* di skenario 2 sebesar 72,66% dengan hasil 19 *class* negatif benar diklasifikasikan sebagai *class* negatif dan 11 *class* negatif diklasifikasikan sebagai *class* positif. Sedangkan 90 *class* positif benar diklasifikasikan sebagai

class positif dan 30 *class* positif diklasifikasikan sebagai *class* negatif.

Kemudian pada pemodelan kedua, 4 skenario yang telah di uji menggunakan algoritma *Support Vector Machine* kemudian dioptimasi dengan *Particle Swarm Optimization*, yang mendapatkan nilai performa terbaik terdapat pada skenario 4 pada kernel *linear* dan *polynomial*. Hasil yang diperoleh mendapatkan 8 *class* negatif benar diklasifikasikan sebagai *class* negatif dan 3 *class* negatif diklasifikasikan sebagai *class* positif. Sedangkan 31 *class* positif benar diklasifikasikan sebagai *class* positif dan 8 *class* positif diklasifikasikan sebagai *class* negatif. Dengan pembagian data 90:10 pada skenario ke 4 akurasi yang di dapatkan sebesar 78.00%. Nilai evaluasi yang lain pun mengalami kenaikan mulai dari presisi menjadi 72.72%, *recall* menjadi 50.00% dan *f1-score* menjadi 59.25%. Maka dapat disimpulkan bahwa pengujian komentar *hate speech* Pilkada Karawang 2020 menggunakan algoritma *Support Vector Machine* lalu dioptimasi dengan *Particle Swarm Optimization* terbukti lebih baik dari pada hanya menggunakan algoritma *Support Vector Machine* saja.

Penelitian selanjutnya bisa menggunakan data dengan sumber yang lebih beragam seperti Twitter atau YouTube. Kemudian disarankan untuk melakukan pengembangan mulai dari algoritma, kernel atau parameter yang digunakan. Serta penerapan untuk mendeteksi *hate speech* pilkada karawang di facebook dapat dikembangkan ke dalam sebuah sistem.

DAFTAR PUSTAKA

- [1] C. Sutrisno, “Partisipasi Warga Negara Dalam Pilkada,” JPK J. Pancasila Dan Kewarganegaraan, vol. 2, no. 2, pp. 36–48, 2017.
- [2] “Daftar 270 Daerah Penyelenggara Pilkada Serentak pada 2020.” <https://tirto.id/daftar-270-daerah-penyelenggara-pilkada-serentak-pada-2020-ecZT> (accessed Dec. 16, 2021).
- [3] “Berikut Daftar 270 Daerah yang Gelar Pilkada Serentak 9 Desember 2020 Halaman all - Kompas.com.” <https://www.kompas.com/tren/read/2020/12/05/193100165/berikut-daftar-270-daerah-yang-gelar-pilkada-serentak-9-desember-2020?page=all> (accessed Dec. 16, 2021).
- [4] litigasi.id, “Jeratan Hukum Ujaran Kebencian (Hate Speech),” 28 Mei 2018.
- [5] S. N. Asiyah, “Klasifikasi berita online menggunakan metode support vector machine dan k-nearest neighbor,” PhD Thesis, Institut Teknologi Sepuluh Nopember, 2016.
- [6] A. Saepudin, R. Aryanti, E. Fitriani, and D. Dahlia, “Optimasi Algoritma SVM Dan k-NN Berbasis Particle Swarm Optimization Pada Analisis Sentimen Fenomena Tagar# 2019GantiPresiden,” J. Tek. Komput., vol. 6, no. 1, pp. 95–102, 2020.
- [7] F. F. Irfani, M. Triyanto, and A. D. Hartanto, “Analisis Sentimen Review Aplikasi Ruangguru Menggunakan Algoritma Support Vector Machine,” JBMI J. Bisnis

- Manaj. Dan Inf. Vol 16 No 3 P 258 2020 Doi 1026487jbmi V16i3 8607, 2020.
- [8] H. S. Utama, D. Rosiyadi, B. S. Prakoso, and D. Ariadarma, "Analisis Sentimen Sistem Ganjil Genap di Tol Bekasi Menggunakan Algoritma Support Vector Machine," J. RESTI Rekayasa Sist. Dan Teknol. Inf., vol. 3, no. 2, pp. 243–250, 2019.
- [9] U. Ependi and A. Putra, "Solusi prediksi persediaan barang dengan menggunakan algoritma apriori (studi kasus: regional part depo auto 2000 Palembang)," JEPIN J. Edukasi Dan Penelit. Inform., vol. 5, no. 2, pp. 139–145, 2019.
- [10] K. A. Safitri and R. Wulanningrum, "Aplikasi Pengenalan Pola Tulisan Tangan Menggunakan Metode Support Vector Machine," in Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi), 2020, vol. 4, no. 1, pp. 201–206.
- [11] N.R. Feta, and A.R. Ginanjar, "Komparasi Fungsi Kernel Metode Support Vector Machine Untuk Pemodelan Klasifikasi Terhadap Comparison of the Kernel Function of Support Vector Machine Method for Modeling Classification of Soybean Plat Disease," vol. 1, no. 1, pp. 33–39, 2019.
- [12] M. R. Lubis, "Metode Hybrid Particle Swarm Optimization-Neural Network Backpropagation Untuk Prediksi Hasil Pertandingan Sepak Bola," J-SAKTI J. Sains Komput. Dan Inform., vol. 1, no. 1, pp. 71–83, 2017.
- [13] L. Mutawalli, M. T. A. Zaen, and W. Bagye, "Klasifikasi Teks Sosial Media Twitter Menggunakan Support Vector Machine (Studi Kasus Penusukan Wiranto)," J. Inform. Dan Rekayasa Elektron., vol. 2, no. 2, pp. 43–51, 2019.