



Optimasi SVM Berbasis PSO pada Analisis Sentimen Wacana Pindah Ibu Kota Indonesia

Primandani Arsi¹, Rizki Wahyudi², Retno Waluyo³

^{1,2}Prodi Informatika, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto

³Prodi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto

¹lukhti.prima@amikompurwokerto.ac.id*^S, ²rizkiw@amikompurwokerto.ac.id, ³waluyo@amikompurwokerto.ac.id

Abstract

President Joko Widodo decided to move the capital city of the country outside Java. The relocation of the capital city is contained in the 2020-2024 National Medium-Term Development Plan. Community response to this has been mixed through national television and social media, especially Twitter. The tendency of Twitter users to respond to the government discourse can be seen with sentiment analysis. Sentiment analysis is one of the areas of Natural Language Processing (NLP) that builds systems for recognizing and extracting opinions. In this study, the Feature Selection PSO algorithm in the classification of the SVM model is proposed to improve the resulting accuracy in the sentiment analysis of moving capital cities. Experiments on the data of 1,319 tweets (457 positive sentiments and 862 negative sentiments) indicate an increase in accuracy by 2.09% from 79.06% to 81.15%, with the classification category is "Good Classification".

Keywords: Natural Language Processing (NLP), Sentiment Analysis, SVM, PSO

Abstrak

Presiden Joko Widodo memutuskan wacana pindah ibu kota negara ke luar Pulau Jawa. Pemindehan ibu kota ini tertuang dalam Rencana Pembangunan Jangka Menengah Nasional 2020-2024. Respon masyarakat terkait hal tersebut sangat beragam baik melalui televisi nasional maupun media sosial khususnya twitter. Kecenderungan respon pengguna twitter dalam menyikapi wacana pemerintah tersebut dapat diketahui dengan analisis sentimen. Analisis sentimen adalah salah satu bidang dari Natural Language Processing (NLP) yang membangun sistem untuk mengenali dan mengekstraksi opini dalam bentuk teks. Dalam penelitian ini, diusulkan algoritma Feature Selection PSO pada klasifikasi model SVM untuk meningkatkan akurasi yang dihasilkan pada analisis sentimen wacana pindah ibu kota. Pembuktian yang dilakukan melalui eksperimen dengan data 1.319 tweets (457 sentimen positif dan 862 sentimen negatif) menunjukkan peningkatan akurasi sebesar 2,09% dari akurasi sebelumnya 79,06% menjadi 81,15% dengan kategori "Good Clasification".

Kata kunci: Natural Language Processing (NLP), Analisis Sentimen, SVM, PSO

1. Pendahuluan

Wacana tentang pemindahan ibu kota Indonesia sering muncul disetiap era presiden, mulai dari Soekarno hingga Joko Widodo. Melalui rapat terbatas pemerintah pada tanggal 29 April 2019, Presiden Joko Widodo memutuskan untuk memindahkan ibu kota negara Indonesia ke luar Pulau Jawa. Wacana pindah ibu kota ini tertuang dalam Rencana Pembangunan Jangka Menengah Nasional 2020-2024 [1]. Respon masyarakat terkait hal tersebut sangat beragam baik melalui televisi nasional maupun media sosial. Media sosial merupakan sarana penyampaian opini yang paling mudah dan murah bagi masyarakat [2]. Berbagai kalangan masyarakat

tanpa batasan umur dapat berkomentar terhadap kebijakan tersebut. Wacana pemerintah tersebut juga menjadi *trending topic* pada media sosial twitter namun tidak selalu mendapat respon positif dari pengguna media sosial [3]. Kontroversi inilah yang memicu perdebatan di media sosial twitter.

Kecenderungan respon pengguna twitter dalam menyikapi wacana pemerintah tersebut dapat diketahui dengan analisis sentimen [4]. Analisis sentimen adalah salah satu bidang dari Natural Language Processing (NLP) yang membangun sistem untuk mengenali dan mengekstraksi opini dalam bentuk teks [5]. Informasi berbasis teks banyak tersedia di internet dalam format

forum media sosial, website pribadi, platform medsos, serta website ulasan produk. Dengan bantuan analisis sentimen, informasi yang sebelumnya tidak terstruktur dapat diubah menjadi data yang lebih terstruktur [6]. Data tersebut dapat menjelaskan opini masyarakat mengenai produk, merk, layanan, politik, atau topik lainnya. Perusahaan, pemerintah, maupun bidang lainnya dapat memanfaatkan data-data tersebut guna analisis marketing, review produk, umpan-balik produk, dan layanan masyarakat [7]. Bagi pemerintah analisis sentimen dibutuhkan untuk mengukur efektifitas kebijakan yang dihasilkan terkait isu-isu yang berkembang salah satunya di twitter [8].

Terdapat beberapa metode *machine learning* yang sering digunakan dalam penerapan analisis sentimen, seperti *Decision Tree*, *Naive Bayes Classifier*, *Random Forest*, *Support Vector Machine (SVM)* dan lain-lain [9][10]. Dari beberapa metode *machine learning* yang disebutkan tersebut, *SVM* merupakan metode terbaik yang memiliki keunggulan berupa kemampuannya dalam komputasi data berdimensi tinggi [11], namun *SVM* masih memiliki kekurangan dalam hal komputasi data dengan jumlah yang besar [12]. Sementara masalah yang umum sering terjadi dalam teknik analisis sentimen adalah banyaknya atribut yang digunakan sehingga data yang digunakan bervolume besar, hal ini dapat mempengaruhi kinerja *klasifier* yang menyebabkan akurasi tidak optimal [13]. Guna mengatasi permasalahan tersebut, diperlukan sebuah algoritma *feature selection* yang dapat meningkatkan kinerja *SVM* dalam mengklasifikasikan analisis sentimen.

Beberapa penelitian terkait analisis sentimen dengan metode *selection fiture* pernah dilakukan sebelumnya; seperti yang dilakukan oleh hernawati pada tahun 2019 dimana penulis melakukan perbandingan antara *SVM* berbasis *PSO* dan *Naive Bayes* berbasis *PSO*. Pada penelitian ini dilakukan analisis terhadap sentimen masyarakat di media sosial *twitter* terkait Isu OTT (Operasi Tangkap Tangan) oleh KPK (Komisi Pemberantasan Korupsi). Dengan total tweet yang digunakan sebanyak 759, penelitian ini menghasilkan akurasi 83,79% untuk *SVM* berbasis *PSO* dan 80,13% untuk *Naive Bayes* berbasis *PSO*. Terjadi peningkatan akurasi sebesar 3,3% dan 3,6% [14]. Anas Faisal dkk pada tahun 2020 dalam penelitiannya, melakukan optimasi algoritma *SVM* dan *Naive Bayes* menggunakan *PSO*. Sentimen Anggota Dewan Perwakilan Rakyat (DPR) RI pada media sosial *twitter* terkait dengan kebijakan ataupun sebatas gagasan digunakan dalam penelitian ini. Total *tweet* yang digunakan adalah 2000 *tweet* yang diperoleh dengan metode *crawling*. Pengujian yang dilakukan dengan *k-fold validation* membuktikan bahwa *PSO* mampu meningkatkan akurasi *SVM* sebesar 3,99% dan *Naive Bayes* 2,8% dengan akurasi tertinggi *SVM* berbasis *PSO*[15]. Penelitian berikutnya dilakukan oleh faried dkk pada

tahun 2020, dengan topik pemindahan ibu kota indonesia. Data diambil dengan metode *crawling* pada periode 23 April 2020 sampai dengan 2 Mei 2020. Total *tweet* yang digunakan adalah 849 *tweets*. Dalam penelitiannya penulis menggunakan algoritma *feature selection PSO* untuk meningkatkan hasil akurasi model yang dibangun sebelumnya yakni *Naive Bayes* dan *Support Vector Machine*. Dari percobaan yang dilakukan, *PSO* terbukti dapat meningkatkan hasil klasifikasi model sebelumnya [16].

Dalam penelitian ini, diusulkan algoritma *Feature Selection PSO* pada klasifikasi model *SVM* untuk meningkatkan akurasi dari model *SVM* pada analisis sentimen wacana pindah ibu kota. *PSO* dipilih dalam penelitian ini dengan alasan konsep sederhana, dengan konvergensi relatif cepat, mudah dalam mengimplementasikan serta dapat diterapkan dalam berbagai bidang guna pemecahan permasalahan terkait optimasi [17]. *PSO* memiliki teknik untuk optimasi bobot atribut (*attribute weight*) terhadap seluruh atribut atau variabel yang dipakai, menyeleksi atribut (*attribute selection*) serta *feature selection* [18]. *Feature selection* yang dimaksud adalah pembobotan atribut dengan cara menghubungkan atribut yang satu dengan atribut lainnya [19].

Adapun tujuan dari penelitian ini yaitu untuk meningkatkan hasil akurasi hasil klasifikasi model *SVM* terhadap sentimen wacana pemindahan ibu kota Indonesia di media sosial *twitter* dengan menggunakan *feature selection PSO*.

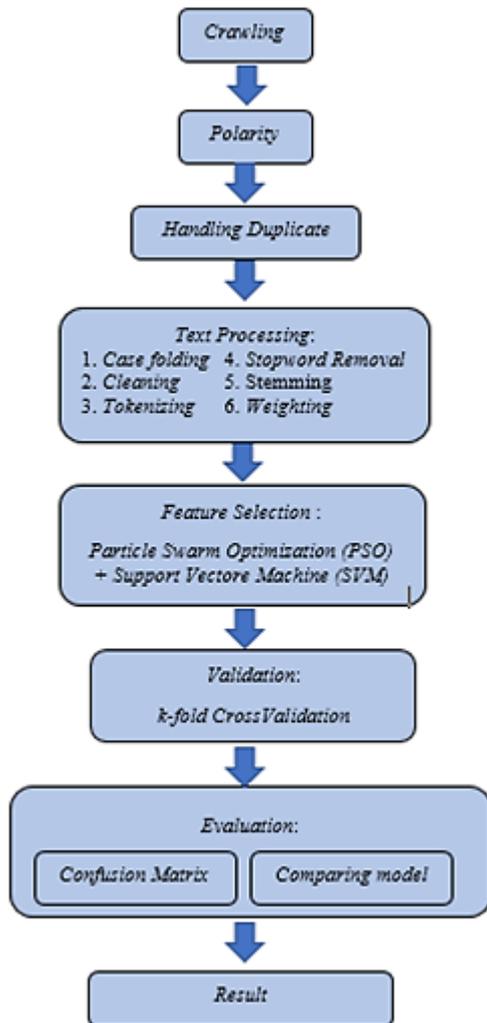
2. Metode Penelitian

Dalam penelitian ini teknik pengumpulan data dilakukan dengan metode *crawling*. Data hasil *crawling* selanjutnya dilakukan proses penanganan duplikat yang disebut dengan *handling duplicate*. Selanjutnya polarisasi terhadap sentimen dilakukan untuk memberikan label positif dan negatif terhadap sentimen. Selanjutnya tahapan inti dari *text mining* dilakukan yaitu *preprocessing*, dalam tahap ini terdiri dari beberapa proses; *case folding*, *cleaning*, *tokenizing*, *stopword removal*, *stemming* dan *weighting*. Berikutnya *feature selection* dengan *PSO* guna pengoptimalan akurasi model *SVM*. Tahap selanjutnya adalah proses validasi menggunakan *k-fold cross validation* dimana dilakukan penentuan atas rasio data *training* dan *testing*. Tahapan terakhir yang dilakukan dalam penelitian ini adalah evaluasi, yaitu dengan *Confussion Matrix* dan membandingkan hasil akurasi model *SVM* dengan *SVM* berbasis *PSO*. Metode penelitian dapat dilihat pada gambar 1.

2.1 Crawling

Crawling adalah metode pengumpulan data pada website dengan memasukan *URL (Uniform Resource Locator)*. *URL* inilah yang menjadi acuan dalam

pencarian *hyperlink*. Selanjutnya pengindeksan dilakukan guna pencarian sebuah kata pada dokumen di setiap *link* [20]. Akses komentar pada twitter atau disebut dengan *tweets* dibutuhkan hak akses; *consumer key*, *consumer secret*, *access token* dan *access token secret*. Kata kunci yang menjadi *keyword* dalam proses *crawling* adalah sebagai berikut: #PindahIbuKota, #IbuKotaPindah dan #IbuKotaBaru.



Gambar 1. Metode Penelitian

2.2 Polarity

Polarisasi atau sering disebut *labeling* dilakukan untuk menentukan kelas sentimen positif dan negatif [21]. Di dalam penelitian ini dilakukan *labeling* oleh dua orang anotator. Hasil *labeling* oleh anotator pertama akan diperiksa ulang oleh anotator kedua sebagai upaya untuk memastikan objektivitas dari *labeling*.

2.3 Handling Duplicate

Data *tweets* hasil *crawling* merupakan data mentah yang kemungkinan terjadi duplikasi dikarenakan proses *retweet* dan *copy-paste* komentar oleh pengguna. Oleh

karenanya tahap *handling duplicate* akan dilakukan guna memastikan tidak ada data yang ganda [22].

2.4 Text Processing

Text Preprocessing pada penelitian ini, menerapkan serangkaian tahap secara berurutan yaitu *case folding*, *cleaning*, *tokenizing*, *stopword removal*, *stemming* [23]. *Case folding* merupakan tahap untuk merubah unsur huruf kapital dalam pada dokumen menjadi standar, yakni *lowercase*. *Cleaning* digunakan untuk membersihkan karakter seperti *https*., @, # dan URL. *Tokenizing* dilakukan untuk memecah kalimat dalam dokumen menjadi kata. *Stopword Removal* dilakukan untuk penghapusan kata dengan kandungan informasi rendah. *Stemming* dilakukan untuk menghilangkan imbuhan baik awalan maupun akhiran sehingga kembali menjadi bentuk kata dasar. *Text processing* dilakukan dengan tujuan merubah *tweets* / data teks tak terstruktur sehingga menjadi data terstruktur untuk kebutuhan analisis sentimen pindah ibu kota.

2.5 Feature Selection

Feature selection merupakan metode untuk menganalisa data dengan tujuan memilih karakter/fitur yang optimal dan mengabaikan karakter/fitur tidak berpengaruh. Penelitian ini mengimplementasikan *Particle Swarm Optimization (PSO)* sebagai metode *feature selection* pada model SVM untuk meningkatkan bobot dari atribut terhadap seluruh atribut yang digunakan, sehingga meningkatkan akurasi model SVM [24].

2.6 Validation dan Evaluation

Guna mengetahui *performance* model yang diterapkan, validasi dilakukan dengan *k-fold cross validation* [25]. Penerapan validasi *k-fold* dilakukan dengan percobaan rasio *training* dan *testing* dilakukan dengan skenario 9:1 dengan nilai *k* bervariasi antara 2 sampai dengan 10. Evaluasi dilakukan dengan menggunakan *Confusion Matrix* serta membandingkan model SVM dengan PSO. Sedangkan pengukuran *perfoma* dari model yang dihasilkan akan disajikan nilai AUC berdasarkan kurva ROC.

3. Hasil dan Pembahasan

Pembahasan teknis perolehan data dengan metode *crawling* sampai dengan hasil pengujian dan evaluasi model SVM dan SVM berbasis PSO akan disajikan dalam bab ini.

3.1. Crawling

Crawling dilakukan pada periode 10 Januari 2020 s.d 17 Maret 2020, dilakukan dengan memanfaatkan fungsi *library Tweepy Python*. Langkah pertama yang dilakukan dalam proses *crawling* adalah mendaftarkan akun twitter pada API twitter. Gunanya adalah untuk

memperoleh *Consumer key* dan *Consumer secret* dari API twitter yang terdaftar agar dapat berinteraksi dengan twitter. Permintaan tersebut akan direspon oleh twitter dengan mengirimkan API key yang dimaksud melalui email. Selanjutnya implementasi *crawling* dilakukan dengan menggunakan bahasa pemrograman python. Diawali dengan instalasi *library tweepy* python, kemudian import *library tweepy* agar fungsi dalam *library* tersebut dapat digunakan. Selanjutnya membuat variable untuk memasukkan API Key yang berhasil didapatkan ketika mendaftar. API Key ini digunakan untuk proses autentikasi. Selanjutnya keyword yang telah ditentukan sebelumnya dapat diinputkan pada baris code python. *Output* dari proses *crawling* ini adalah data *tweet* dengan format .csv. Adapun jumlah *tweets* yang dihasilkan adalah 1.420 *tweets*, *sample tweet* ditunjukkan pada Tabel 1.

Tabel 1. *Sample tweet* hasil *crawling*

No	Tweet
1	Mungkin agak keder juga mengingat begitu ibukota jadi pindah 2024 justru yg ditakuti gubernur malah warga lokal tdk dapat pekerjaan d sana mengingat lbh 100 ribuan PNS adalah eksodus dari jakarta yg akan bekerja https://t.co/2Pat5icWfb
2	@BersihkanID @jokowi @BappenasRI Jakarta udah kegemukan. Ibarat manusia, cara memulihkan kegemukan ya diet. Pindah Ibukota salah satu caranya..
3	@marlina_idha @aniesbaswedan Aku sih berharap ibukota bisa pindah senin depan agar jakarta makin asyik dan gak berisik #aniesgubernurbriiliant #WajahBaruJakarta
4	@yohanagril @biakulu010 @Dennysiregar7 Sebentar lagi pindah... lo koar2 pohon di tebang dijakarta.. tuh calon ibukota baru berapa pohon yg akan di tebang..sinting lu

Tweets diatas merupakan data mentah yang kemungkinan masih terdapat beberapa *tweet* dengan sentimen ganda, hal ini dikarenakan kemungkinan *user me-retweet* dari *user* lain atau sekedar mengutip *tweets* dari *user* lain dengan cara *copy-paste* pada *tweet*.

3.2. Handling duplicate

Proses *handling duplicate* dilakukan dengan tujuan menyeleksi sentimen yang ganda, sehingga tidak memberatkan proses komputasi model yang diusulkan. Dari proses *handling duplicate* yang dilakukan terhadap 1.420 *tweets* (hasil *crawling*) menghasilkan *tweet* yang siap untuk diproses ke tahap *preprocessing* sebanyak 1,319 *tweet*.

3.2. Polarity

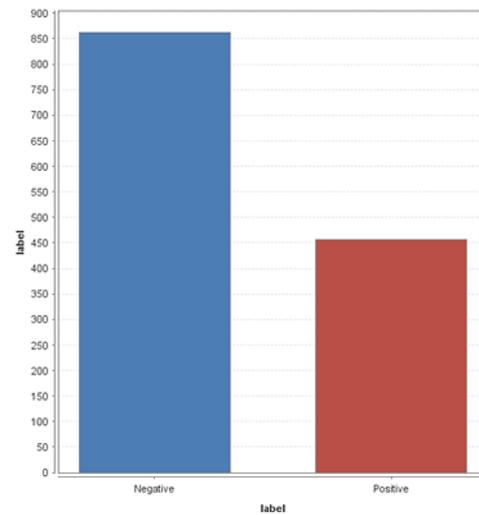
Tahap penentuan polaritas pada sentimen dilakukan oleh dua orang annotator. Annotator pertama bertugas mengidentifikasi sentimen dengan cara memahami konteks kalimat secara manual sedangkan annotator kedua mengevaluasi kembali hasil identifikasi annotator pertama. Hal ini dilakukan dengan tujuan memastikan polaritas yang dihasilkan valid. Dari proses polarisasi yang dilakukan terhadap data *tweet* sebanyak 1,319

dihasilkan *tweet* dengan polaritas negatif sebanyak 862 dan sentimen dengan polaritas positif sebanyak 457. Tabel 2 merupakan hasil polarisasi yang dilakukan dalam tahap ini.

Tabel 2. *Sample* polarisasi sentimen

No	Tweet	Sentimen
1	Mungkin agak keder juga mengingat begitu ibukota jadi pindah 2024 justru yg ditakuti gubernur malah warga lokal tdk dapat pekerjaan d sana mengingat lbh 100 ribuan PNS adalah eksodus dari jakarta yg akan bekerja https://t.co/2Pat5icWfb	Negative
2	@BersihkanID @jokowi @BappenasRI Jakarta udah kegemukan. Ibarat manusia, cara memulihkan kegemukan ya diet. Pindah Ibukota salah satu caranya..	Positive
3	@marlina_idha @aniesbaswedan Aku sih berharap ibukota bisa pindah senin depan agar jakarta makin asyik dan gak berisik #aniesgubernurbriiliant #WajahBaruJakarta	Positive
4	@yohanagril @biakulu010 @Dennysiregar7 Sebentar lagi pindah... lo koar2 pohon di tebang dijakarta.. tuh calon ibukota baru berapa pohon yg akan di tebang..sinting lu	Negative

Adapun jika disajikan dalam bentuk grafik, maka perbandingan polaritas positif dan negatif sebagai berikut.



Gambar 2. Perbandingan polaritas positif dan negatif

3.3. Text Preprocessing

Text Preprocessing dilakukan terhadap data teks yang telah melalui tahap *handling duplicate* dan *polarity*. Selanjutnya beberapa proses dalam tahap *text processing* dilakukan secara sistematis seperti *case folding*; *cleaning*; *tokenizing*; *stopword removal*; dan *stemming*. Pada tabel 3 dapat dilihat *output* dari setiap proses dalam tahap *text processing* yang dilakukan.

Tabel 3. Output dari *text processing*

Proses	Output
Sentimen asli	@marlina_idha @aniesbaswedan Aku sih berharap ibu kota bisa pindah senin depan agar jakarta makin asyik dan gak berisik #aniesgubernurbriiliant #WajahBaruJakarta
Case folding	@marlina_idha @aniesbaswedan aku sih berharap ibu kota bisa pindah senin depan agar jakarta makin asyik dan gak berisik.... #aniesgubernurbriiliant #wajahbarujakarta
Cleansing	aku sih berharap ibu kota bisa pindah senin depan agar jakarta makin asyik dan gak berisik
Tokenizing	aku, sih, berharap, ibu kota, bisa, pindah, senin, depan, agar, jakarta, makin, asyik, dan, gak, berisik
Stopword removal	aku berharap ibu kota bisa pindah senin depan jakarta makin asyik gak berisik
Stemming	aku harap ibu kota bisa pindah senin depan jakarta makin asyik gak berisik

Dengan menggunakan bahasa pemrograman Python, tahap *case folding* dilakukan untuk merubah teks ke dalam bentuk *lowercase* atau standar dengan memanfaatkan modul yang terdapat pada python. Komponen seperti URL, RT (*Retweet*), *username*, karakter HTML, # (*hashtag*) dan tanda baca lainnya juga dihapus, proses ini disebut *Cleansing*. Selanjutnya kalimat dipecah menjadi kata dengan fungsi *split()*; pada python, proses ini disebut dengan *tokenizing*. Kata-kata yang penting hasil *tokenizing* selanjutnya disimpan, sehingga kata yang memiliki makna tidak penting seperti kata penghubung (dan, yang, di, dll) otomatis terhapus. Tahap terakhir dari *text processing* di penelitian ini adalah *stemming* dimana imbuhan berupa awalan dan akhiran akan dihapus sehingga kata yang terbentuk adalah kata dasarnya saja.

3.4. Eksperimen dengan penerapan SVM

Ekperimen dengan menerapkan model SVM dilakukan terhadap data yang telah melalui tahap *text processing*. Kombinasi parameter SVM dilakukan secara manual dari parameter tipe kernel hingga *training cycle*. Berikut merupakan tabel hasil eksperimen untuk tipe kernel.

Tabel 4. Hasil eksperimen tipe *kernel*

Akurasi		
<i>dot</i>	<i>radial</i>	<i>Polynomial</i>
79,06%	77,36%	75,16%

Berdasarkan ekperimen yang dilakukan terhadap tiga tipe kernel SVM yakni *dot*, *radial* dan *polynomial*. Akurasi tertinggi yakni 79,06% pada tipe kernel *dot*. Selanjutnya *dot* ditetapkan sebagai tipe kernel yang akan diimplementasikan untuk eksperimen parameter *training cycle*; *C* dan *epsilon*. Tabel 5 merupakan eksperimen yang dilakukan terhadap parameter *training cycle C* dan *epsilon*.

Tabel 5. Tabel hasil eksperimen *training cycle*

<i>C</i>	<i>Epsilon</i>	Akurasi
0,0	0,0	79.02%
0,1	0,1	79.03%
0,2	0,2	79.03%
0,3	0,3	79.04%
0,4	0,4	79.05%
0,5	0,5	79.06%
0,6	0,6	79.01%
0,7	0,7	79.02%
0,8	0,8	79.02%
0,9	0,9	79.04%
1,0	1,0	79.02%
0,1	1,0	79.02%
1,0	0,1	79.02%

Input nilai parameter *training cycle*; *C* dan *epsilon* dilakukan dengan *range* 0 sampai dengan 1 dengan kombinasi nilai kedua parameter tersebut. Hasil eksperimen pada tabel 5 diatas menunjukkan kombinasi nilai parameter dengan hasil terbaik ditunjukkan pada nilai *C*=0,5 dan *epsilon*=0,5 dengan akurasi 79,06%. Dengan demikian parameter SVM yang ditetapkan dalam penelitian ini adalah tipe kernel = *dot*, *C*=0,5 dan *epsilon*=0,5. Selanjutnya proses validasi terhadap model dilakukan dengan menggunakan *k-fold validation*, ekperimen nilai *k* dilakukan sebanyak 9 kali mulai dari nilai *k*=2 sampai dengan *k*=10. Berikut hasil eksperimen yang dilakukan terhadap nilai *k* pada *k-fold validation*.

Tabel 6. Hasil eksperimen nilai *k*

Nilai <i>k</i>	Akurasi
2	79.05%
3	79.05%
4	79.05%
5	79.06%
6	79.05%
7	79.05%
8	79.04%
9	79.03%
10	79.02%

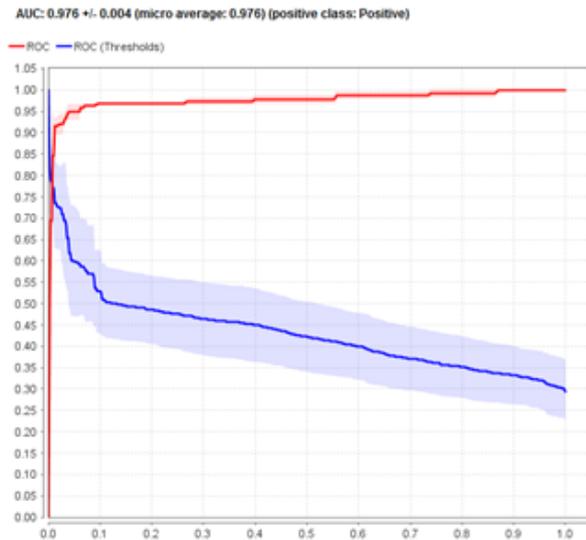
Dari tabel 6 diatas menunjukkan *k* terbaik adalah 5 dengan akurasi 79,06%, artinya tahap pengujian berlangsung sebanyak 5 kali pengujian dengan rata-rata tingkat kesalahan diukur berdasarkan 5 kali pengujian pula. Selanjutnya *output* yang dihasilkan oleh model di evaluasi dengan metode *Confusion Matrix*. Tabel 7 dibawah ini merupakan informasi dari hasil klasifikasi oleh model dan yang sebenarnya dibandingkan.

Tabel 7 diatas menunjukkan akurasi yang dihasilkan 79.06% dengan tingkat *error* 11.74%. *True positive* = 395 dan *true negative*= 590. Artinya, data yang diklasifikasi sesuai sentimen positif berjumlah 395 *record* sementara data yang diklasifikasi sesuai dengan sentimen negatif sejumlah 590. Sedangkan sentimen yang seharusnya terdeteksi positif namun menjadi negatif (*false negative*) sebanyak 247 *record* dan sentimen negatif diklasifikasi

positif (*false positif*) adalah 14. Performa model yang dihasilkan dari *output* diatas dapat diukur dengan nilai AUC (*Area Under Curve*) melalui kurva ROC (*Receiver Operating Characteristic*), yaitu 0.976.

Tabel 7. *Confusion Matrix*

	Akurasi 79.06% +/- 11.74%		
	True negative	True positive	Class precision
Prediction Negative	590	14	97.68%
Prediction Positive	247	395	61.53%
Class Recall	70.49%	96.58%	



Gambar 3. AUC yang dihasilkan model SVM

3.5. Eksperimen dengan penerapan SVM berbasis PSO

Setelah mendapatkan akurasi dari model SVM tahap berikutnya adalah menerapkan PSO pada model yang telah dibangun. Tujuannya adalah untuk meningkatkan akurasi yang dihasilkan dengan cara meningkatkan semua bobot atribut (*atribut weight*) yang digunakan, menyeleksi atribut serta fitur. Dari eksperimen yang telah dilakukan dengan *setting* parameter PSO berupa *population size*=5, *maximum number of generation*=30, *minimal weight* = 0 dan *maximal weight* = 1 menghasilkan akurasi sebesar 81,15%. Berikut merupakan tabel *Confusion Matrix* yang dihasilkan dari model SVM berbasis PSO.

Tabel 8. *Confusion Matrix SVM + PSO*

	Akurasi : 81.15% +/- 10.27		
	True negative	True positive	Class precision
Prediction Negative	614	12	98.08%
Prediction Positive	223	379	64.03%
Class Recall	73.36%	97.07%	

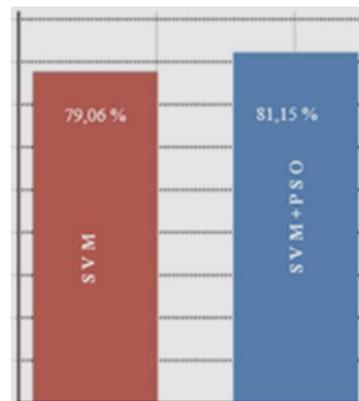
Tabel 8 menunjukkan akurasi 81.15% dengan tingkat *error* 10.27%. *True positif* = 379 *record* dan *true negatif* = 614 *record*. Artinya, sentimen yang diklasifikasi sesuai dengan sentimen positif sebanyak 379 *record* dan

sentimen yang diklasifikasi negatif 614 *record*. Sedangkan sentimen positif namun diklasifikasi negatif sebanyak 223 *record* (*false negatif*) dan sentimen negatif yang diklasifikasi positif (*false positif*) 12 *record*. Performa model yang dihasilkan dari *output* diatas dapat diukur dengan nilai AUC (*Area Under Curve*) melalui kurva ROC (*Receiver Operating Characteristic*), yaitu 0.965.



Gambar 4. AUC model SVM + PSO

Berdasarkan eksperimen yang telah dilakukan terhadap model SVM dan SVM berbasis PSO telah menunjukkan akurasi tertinggi pada model SVM berbasis PSO. Dengan akurasi sebesar 81,15%, artinya PSO terbukti efisien dalam meningkatkan akurasi model SVM terhadap analisis sentimen wacana pindah ibu kota. Hasil klasifikasi ini termasuk dalam kategori *Good Classification*. Berikut grafik perbandingan hasil klasifikasi dari model SVM dan SVM + PSO.



Gambar 5. Grafik perbandingan akurasi

4. Kesimpulan

Pembuktian yang dilakukan melalui eksperimen model SVM dan SVM berbasis seleksi fitur PSO pada wacana

pindah ibu kota dengan data 1.319 tweets (457 sentimen positif dan 862 sentimen negatif) menunjukkan akurasi terbaik yakni pada model SVM berbasis PSO. Adapun rasio data training dan testing yang digunakan adalah skenario 9:1. PSO terbukti efektif dapat meningkatkan kinerja SVM dengan kemampuan feature selection yang dimiliki. Penerapan PSO pada model SVM efektif meningkatkan nilai akurasi sebesar 2.09% dari sebelumnya 79.06% (fair classification) menjadi 81.15% (*good classification*). Dengan demikian hasil klasifikasi model yang diusulkan termasuk dalam kategori *Good Clasifikasi*. Guna perbaikan dalam penelitian selanjutnya diperlukan komparasi model *feature selection* lain terhadap sentimen pindah ibu kota pada media sosial twitter seperti IG (*Information Gain*) atau *Chi Square*. Selain itu perlu kajian lebih mendalam pada perbandingan metode *text processing* yang dilakukan.

Ucapan Terimakasih

Ucapan terimakasih penulis tujuan kepada Kemenristek/Brin yang telah memberikan pendanaan pada skema Penelitian Dosen Pemula 2020.

Daftar Rujukan

- [1]. RI, Rencana Pembangunan Jangka Menengah Nasional 2020-2024. Indonesia, 2020.
- [2]. A. S. Cahyono, "Pengaruh Media Sosial Terhadap Perubahan Sosial Masyarakat di Indonesia," *J. Publiciana*, pp. 140–157, 2016.
- [3]. R. A. Maisal, A. N. Hidayanto, N. F. A. Budi, Z. Abidin, and A. Purbasari, "Analysis of Sentiments on Indonesian YouTube Video Comments : Case Study of The Indonesian Government 's Plan to Move the Capital City," *Int. Conf. Informatics, Multimedia, Cyber Inf. Syst.*, pp. 121–124, 2019.
- [4]. A. P. Kirilenko, S. O. Stepchenkova, H. Kim, and X. R. Li, "Automated Sentiment Analysis in Tourism : Comparison of Approaches," *J. Travel Res.*, 2017.
- [5]. U. Naseem, I. Razzak, K. Musial, and M. Imran, "Transformer Based Deep Intelligent Contextual Embedding for Twitter Sentimen Analysis," *Futur. Gener. Comput. Syst.*, 2020.
- [6]. A. Sharma and U. Ghose, "Sentimental Analysis of Twitter Data with respect to General Elections in India," in *Procedia Computer Science*, 2020, vol. 173, no. 2019, pp. 325–334.
- [7]. J. Qiu, Z. Lin, and Q. Shuai, "Investigating the Opinions Distribution in the Controversy on Social Media," *Inf. Sci. (Ny.)*, 2019.
- [8]. Y. S. Mahardhika and E. Zuliarso, "Analisis Sentimen Terhadap Pemerintahan Joko Widodo Pada Media Sosial Twitter Menggunakan Algoritma Naives Bayes Classifier," in *Prosiding Sintak*, 2018, pp. 409–413.
- [9]. R. K. Thakur and M. V. Deshpande, "Kernel Optimized-Support Vector Machine and Mapreduce framework for sentiment classification of train reviews," *Indian Acad. Sci.*, vol. 44, no. 1, pp. 1–14, 2019.
- [10]. Q. Wang, K. Liu, and K. Ma, "Emotional Analysis of Public Opinions in Colleges and Universities: Based on Naive Bayesian Classification Method," *J. Phys.*, 2019.
- [11]. A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of Sentimental Reviews Using Machine Learning Techniques," in *Procedia - Procedia Computer Science*, 2015, vol. 57, pp. 821–829.
- [12]. D. Wang and Y. Zhao, "Using News to Predict on Investor Sentiment : Based on SVM a Model," *Procedia Comput. Sci.*, vol. 174, no. 2019, pp. 191–199, 2020.
- [13]. S. Wang, D. Li, L. Zhao, and J. Zhang, "Sample cutting method for imbalanced text sentiment classification based on BRC," *Knowledge-Based Syst.*, vol. 37, no. March, pp. 451–461, 2016.
- [14]. Hernawati and W. Gata, "Sentimen Analisis Operasi Tangkap Tangan KPK Menurut Masyarakat Menggunakan Algoritma Support Vector Machine , Naive Bayes Berbasis Particle Swarm Optimization," *Factor Exacta*, vol. 12, no. 3, pp. 230–243, 2019.[15] A. Faisal, Y. Alkhalifi, A. Rifai, and W. Gata, "Analisis Sentimen Dewan Perwakilan Rakyat Dengan Algoritma Klasifikasi Berbasis Particle Swarm Optimization," *J. Inf. Technol. Comput. Sci.*, vol. 3, no. 28, pp. 61–70, 2020.
- [15]. F. Zamachsari, G. V. Saragih, and W. Gata, "Analisis Sentimen Pemindehan Ibu Kota Negara dengan Feature Selection," *Reayasa Sist. dan Teknol. Inf.*, vol. 1, no. 10, pp. 504–512, 2020.
- [16]. Ridwansyah and E. Purwaningsih, "Particle Swarm Optimization untuk Meningkatkan Akurasi Prediksi Pemasaran Bank," *Pilar Nusa Mandiri*, vol. 14, no. 1, pp. 83–88, 2018.
- [17]. E. Indrayuni, "Analisa Sentimen Review Hotel Menggunakan Algoritma Support Vector Machine Berbasis Particle Swarm Optimization," *J. Evolusi*, vol. 4, 2016.
- [18]. Y. E. Achyani, "Penerapan Metode Particle Swarm Optimization Pada Optimasi Prediksi Pemasaran Langsung," *Informatika*, vol. 5, no. 1, pp. 1–11, 2018.
- [19]. F. Liu and W. Xin, "Implementation of Distributed Crawler System Based on Spark for Massive Data Mining," in *International Conference on Computer and Communication Systems*, 2020, pp. 482–485.
- [20]. S. Lee and W. Kim, "Sentiment Labeling For Extending Initial Labeled Data To Improve Semi-Supervised Sentimen Classification," *Electron. Commer. Res. Appl.*, 2017.
- [21]. E. S. Tellez, S. Miranda-jimenez, M. Graff, D. Motezuma, O. S. Siordia, and E. A. Villasenor, "A Case Study of Spanish Text Transformations for Twitter Sentiment Analysis," *Expert Syst. Appl.*, 2017.[23] A. Grieco, M. Pacella, and M. Blaco, "On the application of text clustering in Engineering Change process," in *Procedia CIRP*, 2017, vol. 62, pp. 187–192.
- [22]. M. A. Mosa, "A novel hybrid particle swarm optimization and gravitational search algorithm for multi-objective optimization of text mining," *Appl. Soft Comput. J.*, vol. 90, p. 106189, 2020.
- [23]. S. Alkhalidi et al., "Twitter Sentiment Analysis on Activities of Saudi General Entertainment Authority," *J. Auckl. Univ. Technol.*, pp. 2–6, 2020.