



Komparasi Algoritma Klasifikasi Berbasis Particle Swarm Optimization Pada Analisis Sentimen Ekspedisi Barang

Sharazita Dyah Anggita¹, Ikamah²

^{1,2}Sistem Informasi, Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta

¹sharazita@amikom.ac.id, ²ikmahdarwan01@amikom.ac.id

Abstract

The needs of the community for freight forwarding are now starting to increase with the marketplace. User opinion about freight forwarding services is currently carried out by the public through many things one of them is social media Twitter. By sentiment analysis, the tendency of an opinion will be able to be seen whether it has a positive or negative tendency. The methods that can be applied to sentiment analysis are the Naive Bayes Algorithm and Support Vector Machine (SVM). This research will implement the two algorithms that are optimized using the PSO algorithms in sentiment analysis. Testing will be done by setting parameters on the PSO in each classifier algorithm. The results of the research that have been done can produce an increase in the accreditation of 15.11% on the optimization of the PSO-based Naive Bayes algorithm. Improved accuracy on the PSO-based SVM algorithm worth 1.74% in the sigmoid kernel.

Keywords: Sentiment, Naive Bayes, SVM, PSO

Abstrak

Kebutuhan masyarakat akan ekspedisi barang saat ini mulai meningkat dengan adanya *marketplace*. Opini pengguna tentang pelayanan ekspedisi barang saat ini dilakukan masyarakat melalui banyak hal salah satunya media sosial Twitter. Dengan sentimen analisis, kecenderungan sebuah opini akan mampu terlihat apakah mempunyai kecenderungan positif atau negatif. Metode yang dapat diterapkan pada analisis sentimen yaitu Algoritma *classifier* Naive Bayes dan Support Vector Machine (SVM). Penelitian ini akan melakukan penerapan kedua algoritma tersebut yang dioptimasi menggunakan algoritma PSO pada analisis sentimen. Pengujian dilakukan dengan melakukan *setting* parameter pada PSO di masing-masing algoritma *classifier*. Hasil dari pengujian yang dilakukan mampu menghasilkan peningkatan akurasi sebesar 15.11% pada penerapan PSO di algoritma *Naive Bayes*. Peningkatan akurasi pada algoritma SVM berbasis PSO senilai 1.74% pada kernel sigmoid.

Kata Kunci: Sentimen, Naive Bayes, SVM, PSO

© 2020 Jurnal RESTI

1. Pendahuluan

Analisis sentimen mampu mengubah teks secara otomatis untuk dapat mengekstrak sentimen pada sebuah kalimat. Melalui analisis sentimen, akan dapat mengklasifikasikan sebuah opini menjadi positif atau negatif [1]. Salah satu manfaat analisis sentimen yaitu untuk menangkap opini masyarakat terhadap sebuah layanan ataupun kualitas produk [2]. Kebutuhan masyarakat akan ekspedisi barang saat ini mulai meningkat dengan adanya *marketplace*.

Opini pengguna tentang pelayanan ekspedisi barang saat ini dilakukan masyarakat melalui banyak media salah satunya media sosial Twitter. Banyaknya respon masyarakat yang berupa opini tersebut belum dapat dianalisis secara utuh oleh pihak jasa pengiriman.

Analisis sentimen terhadap Twitter sebelumnya telah dilakukan salah satunya dengan Support Vector Machine dan Naive Bayes [3][4]. Metode SVM mampu mengekstrak dan mengelompokkan opini untuk dapat diketahui opini tersebut masuk kedalam sentimen positif atau negatif [5]. Terdapat beberapa penelitian terkait analisis sentimen sebuah opini. Pada penelitian pengkategorian penelitian menggunakan metode SVM, dapat menghasilkan tingkat akurasi sebesar 90% dengan menggunakan 390 abstrak dan 4 kategori [6]. Metode *classifier* juga dapat digunakan pada algoritma Naive Bayes, dimana metode tersebut akan mampu melihat peluang dari setiap kondisi [7]. Algoritma Naive Bayes dalam pengklasifikasian dokumen akan dapat mengklasifikasikan dengan melihat probabilitas pada setiap model [8]. Pada penerapan analisis

sentimen terhadap suatu produk di Twitter menggunakan Naive Bayes dan Information Gain mampu menghasilkan tingkat akurasi tertinggi senilai 85% dan titik terendah pengujian menghasilkan tingkat akurasi 50% [9]. Penelitian yang dilakukan untuk klasifikasi berita menggunakan *Naive Bayes* dengan seleksi fitur mampu menghasilkan peningkatan nilai dari klasifikasi *Naive Bayes* kelas sebesar 4.3% [10]. Salah satu cara untuk dapat mengoptimasikan performa kedua algoritma tersebut dapat dilakukan dengan Particle Swarm Optimization (PSO). Sebagai alat pemilihan fitur, dengan partikel PSO akan mampu memberikan sebuah kombinasi fitur sebuah ruang masalah [11]. Penelitian mengenai klasifikasi *Google Classroom* yang mengintegrasikan beberapa algoritma untuk mengoptimasi algoritma *Naive Bayes* menghasilkan sebuah kesimpulan bahwa penggunaan algoritma PSO pada *Naive Bayes* mampu meningkatkan akurasi sampai dengan 10% [12]. Selanjutnya penelitian tentang penerapan PSO pada algoritma *Naive Bayes* pada analisis sentimen review hotel. Penelitian tersebut menerapkan beberapa nilai parameter untuk melihat akurasi terbaik, dari beberapa pengujian yang dilakukan didapatkan bahwa penerapan PSO dapat meningkatkan akurasi sebesar 6.42% [13]. Algoritma Particle Swarm Optimization akan mampu melakukan pemilihan fitur melalui pencarian kombinasi fitur dari suatu masalah. Melalui penelitian ini akan dilakukan sebuah komparasi Algoritma *classifier* Support Vector Machine dan Naive Bayes berbasis PSO untuk melihat tingkat akurasi yang dihasilkan.

2. Metode Penelitian

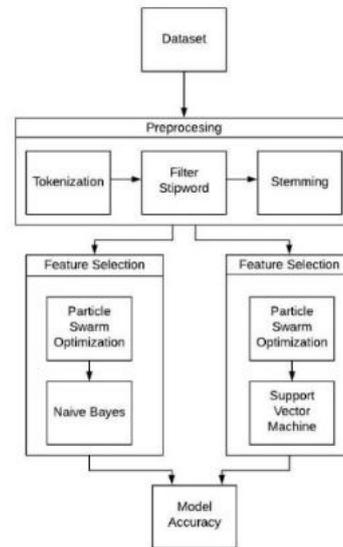
2.1. Preprocessing

Tahap *preprocessing* dilakukan sebagai langkah penyesuaian data, sehingga data yang akan dimasukkan pada model sesuai dengan yang dibutuhkan. Tahap pertama diawali dengan tahap klasifikasi data dimana setiap opini yang didapatkan akan dilakukan proses klasifikasi data menjadi dua kategori yaitu kategori negatif dan positif. Tahap tokenisasi tokenisasi dilakukan untuk memfilter tanda baca dan simbol yang bukan huruf. Selanjutnya yaitu tahap *transform cases* yang akan merubah semua huruf pada opini menjadi huruf kecil. Tahap yang paling akhir adalah tahap *stemming* yang merupakan proses eliminasi prefix dan suffix untuk dapat membentuk sebuah kata dasar

2.2. Penerapan Metode

Metode yang diusulkan dalam penelitian ini terdiri dari 2 algoritma analisis sentimen yaitu Support Vector Machine dan Naive Bayes berbasis Particle Swarm Optimization (PSO) yang akan dikomparasi. Particle Swarm Optimization (PSO) Particle Swarm Optimization merupakan sebuah algoritma yang berbasis pada populasi. Alurnya akan dimulai dengan

dataset *random* yang akan mengoptimasi global dengan generasi yang baru [10]. Alur metode yang akan dilakukan terdapat pada Gambar 1.



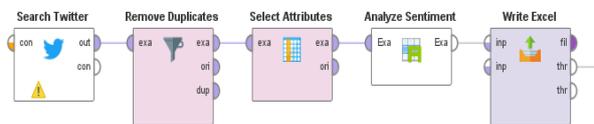
Gambar 1. Diagram Alur Penelitian

Pada penerapan Algoritma *Naive Bayes* berbasis PSO akan dilakukan pengujian pada beberapa parameter pada algoritma PSO. Melalui beberapa tahap pengujian yang dilakukan, akan didapatkan model dengan nilai terbaik. Pada penerapan Algoritma *Support Vector Machine* pengujian akan diawali dengan pemilihan kernel dengan hasil terbaik. Melalui kernel tersebut kemudian akan diterapkan perubahan nilai parameter pada algoritma PSO untuk dapat melihat optimasi yang terjadi. Tahap akhir yang dilakukan adalah dengan melihat tingkat akurasi dari kedua algoritma sentimen yang telah dioptimasi menggunakan algoritma PSO.

3. Hasil dan Pembahasan

3.1. Pengumpulan Data

Pengumpulan data dilakukan menggunakan aplikasi *RapidMiner Studio* yang dikoneksikan dengan media sosial Twitter pada Gambar 2.



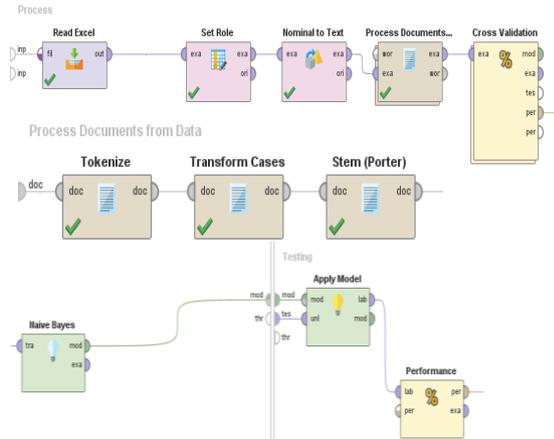
Gambar 2 Proses Crawling Data Twitter

Proses yang dilakukan pada tahapan pengumpulan data diawali dengan operator *Search Twitter*, pada tahap ini ditentukan jumlah data yang akan diambil dan *query* yang akan digunakan untuk topik pengambilan data. Pada penelitian ini topik analisis sentimen yang digunakan adalah jasa ekspedisi Sicepat. Opini publik mengenai ekspedisi Sicepat yang digunakan pada penelitian ini berjumlah 272 data yang sudah diklasifikasikan secara manual. Klasifikasi opini yang dihasilkan terdiri dari 137 data opini negatif dan 135

opini positif. Proses selanjutnya yaitu *Remove Duplicates* yang digunakan untuk menghapus data yang sama yang kemudian dilanjutkan dengan operator *Select Attributes* untuk memilih atribut apa saja yang akan diambil. Operator *Analyze Sentimen* merupakan proses yang akan mengklasifikasikan data menjadi kategori opini positif dan negatif. Data yang sudah diambil akan dibentuk dengan format .xls sesuai dengan operator *Write Excell* yang sudah dipilih.

3.2. Pengujian Model Naive Bayes Klasik

Pengujian model *Naive Bayes* klasik dilakukan menggunakan aplikasi *Rapidminer Studio*.



Gambar 3. Desain Process Pengujian Naive Bayes Klasik

Operator *Read Excell* pada Gambar 3. digunakan untuk membaca hasil *Crawling* data melalui media sosial *Twitter* yang berbentuk file .xls dan sudah mempunyai label positif dan negatif. Sebelum masuk ke operator validasi dilakukan *preprocessing* data menggunakan *tokenize*, *transform case* dan *stem porter*. Pada proses *Cross Validation* didalamnya terdapat dua bagian yaitu *training* dan *testing*. Bagian *training* berisi penerapan algoritma *Naive Bayes* sedangkan di bagian *testing* berisi operator *apply model* sebagai pengimplementasian algoritma dan operator *performance* untuk perhitungan tingkat akurasi dari algoritma yang diterapkan.

Pengujian yang dilakukan menggunakan *Naive Bayes* klasik akan berfokus pada nilai *cross validation* yang diberikan. Pengujian akan dilakukan 10 kali dengan nilai *validation* 1-10. Proses pengujian ini akan membagi data secara *random* menjadi 10 bagian, dimulai pada *validation* pertama dan akan dilanjutkan ke 9 nilai *validation* berikutnya. Melalui 10 kali pengujian yang dilakukan, akan diambil tingkat akurasi terbaik dengan melihat banyak data yang terklasifikasi dengan tepat. Hasil pengujian menggunakan algoritma *Naive Bayes* klasik terdapat pada Tabel 1.

Pada pengujian yang telah dilakukan diketahui bahwa tingkat akurasi tertinggi terdapat pada *validation* ke 5. Tingkat akurasi yang dihasilkan yaitu 65.70%. Detail

klasifikasi opini yang dihasilkan terdapat pada Tabel *confusion matrix* algoritma *Naive Bayes* pada Tabel 2.

Tabel 1. Hasil Eksperimen Naive Bayes Klasik

Validation	Accuracy (%)
1	60.47
2	60.47
3	62.21
4	63.37
5	65.70
6	61.63
7	62.21
8	59.88
9	63.37
10	61.63

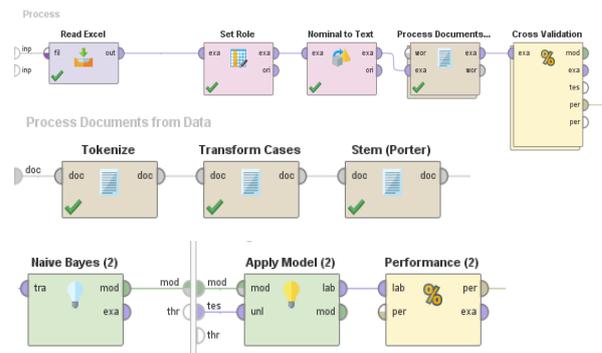
Tabel 2. Hasil Confusion Matrix Algoritma Naive Bayes Klasik

	Accuracy = 65.78% +/- 7.57% (mikro average : 65,70%)	
	true Positive	true Negative
pred Positive	76	50
pred Negative	59	87

Melalui Tabel 2 diperoleh 76 opini yang diklasifikasikan positif, dan 87 opini yang diklasifikasikan sebagai opini negatif. Selain itu terdapat kesalahan jumlah klasifikasi opini yang seharusnya sebagai opini positif sebanyak 59 opini dan kesalahan klasifikasi yang seharusnya beropini negatif sebanyak 50 opini. Melihat hasil klasifikasi prediksi diperoleh tingkat akurasi model *Naive Bayes* klasik menggunakan *Cross Validation K-Fold=5* sebesar 65.70%.

3.2 Pengujian Model Naive Bayes + PSO

Pengujian model *Naive Bayes* Berbasis PSO dilakukan untuk dapat mengoptimasi tingkat akurasi model *Naive Bayes* Klasik. Desain pengujian model *Naive Bayes* berbasis PSO terdapat pada Gambar 4.



Gambar 4. Desain Pengujian Model Naive Bayes Berbasis PSO

Parameter pengujian yang dilakukan pada pengujian ini akan disesuaikan untuk dapat mengetahui model dengan akurasi yang terbaik. Pada pengujian *Naive Bayes* berbasis PSO ini , pengujian pertama akan dilakukan 10 kali pengujian pada nilai *validation* pada proses *cross validation*. Hasil dari pengujian ini terdapat pada Tabel 3.

Tabel 3. Pengujian Model 10 *Validation Naive Bayes* Berbasis PSO

Validation	Accuracy (%)
1	75.58
2	75.58
3	75
4	76.74
5	75.58
6	74.42
7	78.49
8	76.16
9	72.67
10	71.51

Tingkat akurasi terbaik dari 10 kali pengujian terdapat pada nilai *validation* 7 yaitu dengan nilai 78.49%. Pengujian selanjutnya akan lebih terfokus pada proses PSO. Pengujian selanjutnya dilakukan dengan memberikan nilai parameter tambahan yaitu *population size* pada proses *Optimize Weight* dengan rentang 1-10 dan parameter *insertia weight* dengan nilai 0.1. Hasil pengujian dengan penyesuaian nilai tersebut terdapat pada Tabel 4.

Tabel 4. Pengujian *Naive Bayes* Berbasis PSO Dengan Parameter *Validatin, Population Size* dan *Inertia Weight*

Validation	Population Size	Inertia Weight	Accuracy (%)
7	1	0.1	59.88
7	2	0.1	69.19
7	3	0.1	71.51
7	4	0.1	72.67
7	5	0.1	71.51
7	6	0.1	74.42
7	7	0.1	80.23
7	8	0.1	76.74
7	9	0.1	76.64
7	10	0.1	77.91

Melalui hasil pengujian pada Tabel 4, diperoleh nilai akurasi terbaik terdapat pada parameter *validation* =7, *population size* = 7 dan *inertia weight* =0.1. Nilai akurasi yang dihasilkan mengalami kenaikan menjadi 80.23%. Pengujian selanjutnya akan dilakukan dengan mengambil model penerapan paramater terbaik yaitu pada parameter *validation* =7, *population size* = 7 dan *inertia weight* =0.1 yang akan ditambahkan 1 buah parameter baru yaitu *max number*. Pengujian ini akan dilakukan dengan memasukkan nilai *max number* antara 10-100, kemudian nilai parameter *inertia weight* akan di *setting* dengan nilai 1. Hasil pengujian ini terdapat pada Tabel 5.

Tabel 5. Pengujian *Naive Bayes* Berbasis PSO Dengan Parameter *Validation, Population Size, Inertia Weight* dan *Max Number* Tahap 1

Validation	Population Size	Inertia Weight	MAX NUMBER	Accuracy (%)
7	7	1	10	71.51
7	7	1	20	76.16
7	7	1	30	77.91
7	7	1	40	79.65
7	7	1	50	79.65
7	7	1	60	79.65
7	7	1	70	79.65
7	7	1	80	78.49
7	7	1	90	79.07
7	7	1	100	80.23

Dalam pengujian yang dilakukan, hasil terbaik terdapat pada *validation* =7, *population size* = 7, *inertia weight* =1 dan nilai *max number*=100. Tingkat akurasi yang dihasilkan yaitu 80.23%. Pengujian akan kembali dilakukan dengan menerapkan model terbaik tersebut dan merubah kembali nilai *inertia weight* dengan rentang 0.1 sampai dengan 1. Hasil dari pengujian terdapat pada Tabel 6.

Tabel 6. Pengujian *Naive Bayes* Berbasis PSO Dengan Parameter *Validation, Population Size, Inertia Weight* dan *Max Number* Tahap 2

Validation	Population Size	Inertia Weight	MAX NUMBER	Accuracy (%)
7	7	0.1	100	83.14
7	7	0.2	100	80.81
7	7	0.3	100	84.88
7	7	0.4	100	80.81
7	7	0.5	100	80.81
7	7	0.6	100	78.49
7	7	0.7	100	76.74
7	7	0.8	100	76.16
7	7	0.9	100	80.81
7	7	1	100	80.23

Nilai terbaik pada pengujian terakhir diperoleh pada nilai parameter *validation* =7, *population size* = 7, *inertia weight* =0.2 dan nilai *max number*=100. Tingkat akurasi yang dihasilkan mengalami peningkatan dari pengujian sebelumnya yaitu sebesar 80.81%. Detail klasifikasi opini pada nilai tersebut terdapat pada Tabel 7.

Tabel 7. *Confusion Matrix* Pengujian Model *Naive Bayes* Berbasis PSO

Accuracy = 80.81% +/- 7.40% (mikro average : 80.81 %)		
	true Positive	true Negative
pred Positive	89	37
pred Negative	46	100

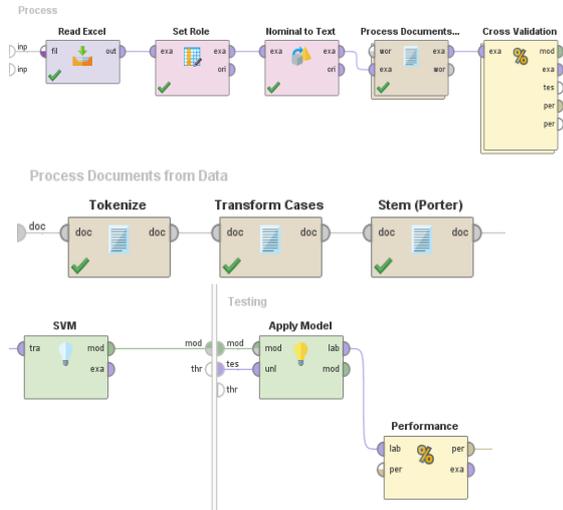
Berdasarkan Tabel 7. diperoleh 89 opini yang diklasifikasikan positif, dan 100 opini yang diklasifikasikan sebagai opini negatif. Selain itu terdapat kesalahan jumlah klasifikasi opini yang seharusnya sebagai opini positif sebanyak 37 opini dan kesalahan klasifikasi yang seharusnya beropini negatif sebanyak 46 opini.

3.3 Pengujian Model *Support Vector Machine* Klasik

Perhitungan model *Support Vector Machine* dilakukan dengan data dari media sosial *Twitter* yang sudah diberi label opini positif dan negatif. Desain model perhitungan algoritma *Support Vector Machine* klasik terdapat pada Gambar 5.

Operator *Read Excell* pada Gambar 5 digunakan untuk membaca hasil *Crawling* data melalui media sosial *Twitter* yang berbentuk file .xls dan sudah mempunyai label positif dan negatif. Sebelum masuk ke operator validasi dilakukan *preprocessing* data menggunakan *tokenize, transform case dan stem porter*. Pada proses *Cross Validation* akan dilakukan 2 tipe pengujian yaitu *K-Fold* = 5 dan *K-Fold* =10 yang didalamnya terdapat

dua bagian yaitu *training* dan *testing*. Bagian *training* berisi penerapan algoritma *Support Vector Machine* sedangkan di bagian *testing* berisi operator *apply model* sebagai pengimplementasian algoritma dan operator *performance* untuk perhitungan tingkat akurasi dari algoritma yang diterapkan.



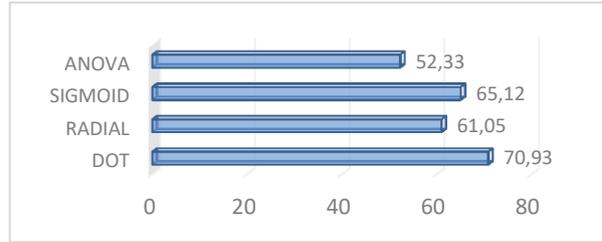
Gambar 5. Desain Perhitungan Model SVM Klasik

Pengujian pada algoritma SVM klasik dilakukan pada 4 bagian kernel yaitu Dot, Radial, Sigmoid dan Anova. Setiap kernel dilakukan dua kali pengujian pada *cross validation fold*. Hasil keseluruhan pengujian SVM klasik terdapat pada Tabel 8.

Tabel 8. Hasil Perhitungan SVM Klasik *K-Fold=5*

NO	KERNEL	FOLDS	ACCURACY (%)
1	Dot	10	62.21
		5	70.93
2	Radial	10	52.33
		5	61.05
3	Sigmoid	10	64.53
		5	65.12
4	Anova	10	47.67
		5	52.33

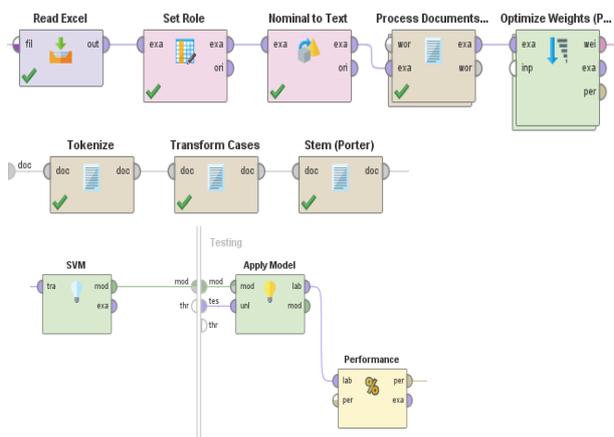
Hasil pengujian pada Tabel 8. diketahui bahwa tingkat akurasi pada masing-masing kernel mempunyai nilai yang bervariasi. Pengujian kernel dot pada *k-folds 10* menghasilkan tingkat akurasi 62.21% dan pada *k-folds 5* mengalami kenaikan akurasi menjadi 70.93%. Pengujian kernel radial pada *k-folds 10* menghasilkan tingkat akurasi 52.33% dan pada *k-folds 5* mengalami peningkatan akurasi menjadi 61.05%. Pengujian kernel sigmoid pada *k-folds 10* menghasilkan tingkat akurasi 64.53% dan pada *k-folds 5* mempunyai kenaikan nilai yang tidak terlalu signifikan menjadi 65.12%. Pengujian kernel anova pada *k-folds 10* menghasilkan tingkat akurasi 47.67% dan pada *k-folds 5* mengalami kenaikan akurasi menjadi 52.33%. Melalui hasil tersebut nilai terbaik pada pengujian masing-masing kernel terdapat pada *cross validation k-fold = 5*. Hasil perbandingan tingkat akurasi masing-masing kernel pada *k-fold = 5* terdapat pada Gambar 6.



Gambar 6. Perbandingan Hasil Pengujian SVM Klasik

Berdasarkan Gambar 6. diketahui bahwa tingkat akurasi terbaik terdapat pada kernel dot dengan nilai 70.93%.

3.4 Perhitungan Model *Support Vector Machine* + PSO
 Pengujian model *Support Vector Machine* Berbasis PSO dilakukan untuk dapat mengoptimasi tingkat akurasi model *Support Vector Machine* Klasik. Parameter pengujian yang digunakan adalah *Cross Validation* dengan *K-Fold = 5* dan *K-Fold = 10*. Desain model pengujian model SVM berbasis PSO terdapat pada Gambar 7.



Gambar 7. Desain Perhitungan Model Support Vector Machine Berbasis PSO

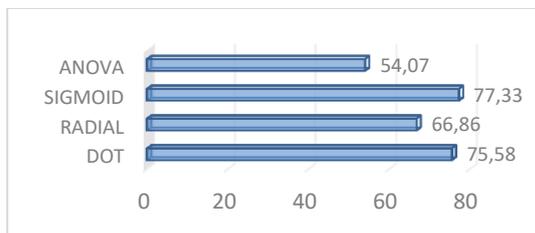
Sebagai parameter perbandingan, pengujian algoritma SVM berbasis PSO juga dilakukan pada 4 kernel yaitu kernel Dot, Radial, Sigmoid dan Anova. Berdasarkan pengujian yang sudah dilakukan didapatkan tingkat akurasi pada Tabel 9.

Tabel 9. Tabel Hasil Perhitungan SVM + PSO

NO	KERNEL	FOLDS	ACCURACY (%)
1	Dot	10	74.24
		5	75.58
2	Radial	10	66.28
		5	66.86
3	Sigmoid	10	73.26
		5	77.33
4	Anova	10	48.84
		5	54.07

Berdasarkan tingkat akurasi yang dihasilkan pada Tabel 9 diketahui bahwa hasil pengujian pada tiap-tiap kernel menghasilkan nilai yang berbeda. Pada kernel Dot *k-fold = 10* menghasilkan tingkat akurasi sebesar 74.24% dan pada *k-fold = 5* menghasilkan nilai yang

lebih tinggi yaitu sebesar 75.58%. Kernal radial pada $k\text{-fold} = 10$ menghasilkan nilai akurasi sebesar 66.28% dan pada $k\text{-fold}=5$ mengalami sedikit kenaikan sebesar 66.86%. Tingkat akurasi pada kernel Sigmoid menghasilkan nilai 73.26% pada $k\text{-fold}=10$ dan pada $k\text{-fold}=5$ mengalami kenaikan tingkat akurasi sebesar 77.33%. Pengujian terakhir dilakukan pada kernel Anova, masih rendah jika dibandingkan dengan kernel lainnya yaitu senilai 48.84% pada $k\text{-fold}=10$, sedangkan pengujian pada $k\text{-fold} = 5$ menghasilkan tingkat akurasi sebesar 54.07%. Pengujian yang dilakukan pada algoritma SVM berbasis PSO menghasilkan tingkat akurasi terbaik pada setiap kernelnya pada pengujian *cross validation* $k\text{-fold}=5$. Hasil perbandingan tingkat akurasi masing-masing kernel pada parameter $k\text{-fold} = 5$ terdapat pada Gambar 9.



Gambar 8. Perbandingan Akurasi Pengujian SVM Berbasis PSO K-Fold=5

Hasil perbandingan tingkat akurasi pengujian SVM berbasis PSO dengan parameter $k\text{-fold} = 5$ pada Gambar 8 menunjukkan bahwa nilai akurasi terbaik terdapat pada kernel Sigmoid yaitu dengan tingkat akurasi 77.33%. Penggunaan kernel pada algoritma SVM berbasis PSO diketahui akan mempengaruhi klasifikasi yang dihasilkan dengan tingkat akurasi yang berbeda-beda.

Dengan menggunakan kernel sigmoid pada $k\text{-fold}=5$, pengujian lanjutan dilakukan menggunakan *setting* parameter pada PSO untuk meningkatkan akurasi. Parameter pada pengujian awal dilakukan dengan memberikan nilai *population size* pada rentang 1 sampai dengan 10 dan nilai *inertia weight* sebesar 0.1. Hasil yang didapatkan pada pengujian tahap awal terdapat pada Tabel 10.

Tabel 10. Hasil Pengujian SVM Berbasis PSO Pada Kernel Sigmoid Dengan Parameter *Population Size* dan *Inertia Weight* Tahap 1

Validation	Population Size	Inertia Weight	Accuracy (%)
5	1	0.1	72.09
5	2	0.1	72.09
5	3	0.1	70.93
5	4	0.1	74.42
5	5	0.1	73.84
5	6	0.1	76.16
5	7	0.1	76.16
5	8	0.1	79.07
5	9	0.1	77.33
5	10	0.1	78.49

Pada hasil pengujian SVM berbasis PSO di kernel Sigmoid diperoleh nilai akurasi tertinggi 78.49% pada

model *setting* parameter $k\text{-fold}=5$, *population size*=10 dan nilai *inertia weight* =0.1. Perubahan nilai parameter pada algoritma PSO pada SVM di kernel sigmoid ternyata berpengaruh yang diketahui dengan adanya kenaikan nilai akurasi sebesar 1.16%. Pengujian selanjutnya dilakukan dengan merubah nilai parameter *inertia weight* menjadi 1 dan menambahkan parameter baru *max number* dengan rentang nilai antara 10 sampai dengan 100. Hasil pengujian terdapat pada Tabel 11.

Tabel 11. Hasil Pengujian SVM Berbasis PSO Kernel Sigmoid Dengan Parameter *Population Size*, *Inertia Weight* dan *Max Number*

Validation	Population Size (Q)	Inertia Weight	Max Number	Accuracy (%)
5	10	1	10	76.74
5	10	1	20	76.74
5	10	1	30	80.23
5	10	1	40	77.91
5	10	1	50	76.16
5	10	1	60	76.16
5	10	1	70	76.16
5	10	1	80	76.74
5	10	1	90	77.91
5	10	1	100	76.16

Pengujian dengan hasil terbaik pada Tabel 11 terdapat pada model *setting* parameter $k\text{-fold}=5$, *population size*=10, nilai *inertia weight* = 1 dan *max number* = 30. Nilai akurasi yang dihasilkan 80.23% di mana terdapat peningkatan nilai akurasi sebesar 1.74% dari pengujian sebelumnya. Pengujian terakhir Algoritma SVM berbasis PSO dilakukan dengan merubah nilai parameter *inertia weight* pada pengujian sebelumnya. Pengujian dilakukan 10 kali dengan memberikan nilai *inertia weight* antara 0.1 sampai dengan 1. Hasil pengujian terdapat pada Tabel 12.

Tabel 12. Hasil Pengujian SVM Berbasis PSO Pada Kernel Sigmoid Dengan Parameter *Population Size* dan *Inertia Weight* Tahap 1

Validation	Population Size	Max number	Inertia weight	Accuracy (%)
5	10	30	0.1	78.49
5	10	30	0.2	80.23
5	10	30	0.3	79.65
5	10	30	0.4	76.81
5	10	30	0.5	80.3
5	10	30	0.6	79.71
5	10	30	0.7	76.25
5	10	30	0.8	78.02
5	10	30	0.9	75.58
5	10	30	1	80.23

Akurasi terbaik yang dihasilkan terdapat pada parameter *inertia weight* 0.5 senilai 80.3%. Nilai tersebut mengalami sedikit kenaikan dari pengujian sebelumnya. Detail klasifikasi opini pada nilai tersebut terdapat pada Tabel 13.

Berdasarkan Tabel 13. diperoleh 91 opini yang diklasifikasikan positif, dan 97 opini yang diklasifikasikan sebagai opini negatif. Selain itu terdapat kesalahan jumlah klasifikasi opini yang seharusnya sebagai opini positif sebanyak 44 opini dan

kesalahan klasifikasi yang seharusnya beropini negatif sebanyak 40 opini.

Tabel 13. . Confusion Matrix Pengujian Model SVM Berbasis PSO

Accuracy = 80.3% +/- 5.99% (mikro average : 80.23 %)		
	true Positive	true Negative
pred Positive	91	40
pred Negative	44	97

3.5 Hasil Penelitian

Penerapan algoritma *classifier* pada analisis sentimen ekspedisi barang telah dilakukan melalui beberapa tahap pengujian. Pengujian pertama dilakukan pada algoritma *Naive Bayes*. Hasil perbandingan pengujian algoritma *Naive Bayes* terdapat pada Tabel 14.

Tabel 14. Perbandingan Akurasi Algoritma *Naive Bayes*

Algoritma	Accuracy (%)
Naive Bayes Klasik	65.7
Naive Bayes + PSO	80.81

Pengujian yang dilakukan pada algoritma *Naive Bayes* klasik menghasilkan tingkat akurasi sebesar 65.7%. Peningkatan nilai akurasi dihasilkan dengan menerapkan optimasi menggunakan algoritma PSO. Hasil pengujian mengalami peningkatan menjadi 80.81% dimana hasil terbaik dari pengujian ini terdapat pada model *setting* parameter *validation = 7, population size = 7, inertia weight = 0.2* dan *max number = 100*. Nilai akurasi pada pengujian tersebut mengalami peningkatan sebesar 15.11% dari penerapan model *Naive Bayes* klasik tanpa optimasi menggunakan PSO. Pengujian berikutnya dilakukan pada algoritma SVM. Hasil perbandingan pengujian terdapat pada Tabel 15.

Tabel 15. Perbandingan Akurasi Algoritma SVM

Algoritma	Accuracy (%)
SVM klasik	70.93
SVM + PSO	80.3

Penerapan algoritma SVM klasik pada analisis sentimen menghasilkan nilai akurasi sebesar 70.93% pada kernel DOT. Sedangkan optimasi algoritma SVM menggunakan PSO mampu menghasilkan peningkatan akurasi sebesar 80.3%. Peningkatan nilai tersebut terdapat pada model *setting* parameter *validation = 5, population size = 10, inertia weight = 0.5* dan *max number = 30*. Akurasi ini mengalami peningkatan sebesar 1.74% dibandingkan dengan SVM klasik tanpa optimasi menggunakan PSO.

4. Kesimpulan

Berdasarkan tahap pengujian analisis sentimen menggunakan model algoritma klasik dan algoritma berbasis *Particle Swarm Optimization* (PSO), diperoleh beberapa nilai tingkat akurasi yang berbeda. Penerapan optimasi algoritma menggunakan *Particle Swarm Optimization* terbukti dapat meningkatkan tingkat akurasi dari model klasik algoritma *Naive Bayes* dan *Support Vector Machine* (SVM). Tingkat akurasi

terbaik algoritma *Naive Bayes* berbasis PSO *validation = 7, population size = 7, inertia weight = 0.2* dan *max number = 100* senilai 80.81%. Nilai akurasi pada algoritma *Naive Bayes* mengalami peningkatan sebesar 15.11% setelah dioptimasi menggunakan PSO. Sedangkan tingkat akurasi terbaik algoritma SVM berbasis PSO terdapat pada parameter *validation = 5, population size = 10, inertia weight = 0.5* dan *max number = 30* yaitu sebesar 80.3%. Nilai akurasi tersebut mengalami peningkatan sebesar 1.74% dibandingkan dengan SVM klasik tanpa optimasi menggunakan PSO. Kesimpulan yang dapat ditarik dari penelitian ini yaitu analisis sentimen jasa ekspedisi menggunakan algoritma *Naive Bayes* berbasis PSO memiliki tingkat akurasi lebih tinggi dibandingkan dengan algoritma SVM berbasis PSO. Optimasi algoritma menggunakan algoritma PSO pada algoritma *classifier* mampu meningkatkan tingkat akurasi pada analisis sentimen jasa ekspedisi barang.

Ucapan Terimakasih

Penulis mengucapkan terimakasih kepada Lembaga Penelitian Universitas AMIKOM Yogyakarta yang telah memberikan *support* atas berjalannya penelitian ini.

Daftar Rujukan

- [1] T. A. Lorosae, B. D. Prakoso, Saifudin, and Kusri, "Analisis Sentimen Berdasarkan Opini Masyarakat Pada Twitter Menggunakan Naive Bayes," *Semin. Nas. Teknol. Inf. dan Multimedia*, 2018.
- [2] J. Ipawati, Kusri, and E. Taufiq Luthfi, "Komparasi Teknik Klasifikasi Teks Mining Pada Analisis Sentimen," *Indones. J. Netw. Secur.*, vol. 6, no. 1, pp. 28–36, 2017.
- [3] R. Raksanagara, Y. H. Chrisnanto, and A. I. Hadiana, "Analisis Sentimen Jasa Ekspedisi Barang Menggunakan Metode Naive Bayes," *Anal. Sentimen Jasa Ekspedisi Barang Menggunakan Metod. Naive Bayes*, pp. 19–24, 2016.
- [4] N. Saputra, T. B. Adji, and A. E. Permasari, "Analisis Sentimen Data Presiden Jokowi dengan Preprocessing Normalisasi dan Stemming Menggunakan Metode Naive Bayes dan SVM," *J. Din. Inform.*, vol. 5, no. November, p. 12, 2015.
- [5] A. Novantirani, M. K. Sabariah, and V. Effendy, "Analisis Sentimen pada Twitter untuk Mengenai Penggunaan Transportasi Umum Darat Dalam Kota dengan Metode Support Vector Machine," *e-Proceeding Eng.*, 2015.
- [6] F. S. Jumeilah, "Penerapan Support Vector Machine (SVM) untuk Pengkategorian Penelitian," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 1, no. 1, p. 19, 2017, doi: 10.29207/resti.v1i1.11.
- [7] R. R. Setiawan *et al.*, "Analisis Sentimen Pariwisata di Kota Malang Menggunakan Metode Naive Bayes dan Seleksi Fitur Query Expansion Ranking," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, 2017, doi: 10.1074/jbc.M209498200.
- [8] J. Abellán and J. G. Castellano, "Improving the Naive Bayes classifier via a quick variable selection method using maximum of entropy," *Entropy*, vol. 19, no. 6, 2017, doi: 10.3390/e19060247.
- [9] A. W. Attabi, L. Muflikhah, and M. A. Fauzi, "Penerapan Analisis Sentimen untuk Menilai Suatu Produk pada Twitter Berbahasa Indonesia dengan Metode Naive Bayes Classifier dan Information Gain," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 11, pp. 4548–4554, 2018.
- [10] B. S. Prakoso, D. Rosiyadi, H. S. Utama, and D. Aridarma, "Klasifikasi Berita Menggunakan Algoritma Naive Bayes

- Classifier Dengan Seleksi Fitur Dan Boosting,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 227–232, 2019, doi: 10.29207/resti.v3i2.1042.
- [11] Y. Zhou, N. Wang, and W. Xiang, “Clustering Hierarchy Protocol in Wireless Sensor Networks Using an Improved PSO Algorithm,” *IEEE Access*, vol. 5, pp. 2241–2253, 2017, doi: 10.1109/ACCESS.2016.2633826.
- [12] F. Pramono, Didi Rosiyadi, and Windu Gata, “Integrasi N-gram, Information Gain, Particle Swarm Optimization di Naïve Bayes untuk Optimasi Sentimen Google Classroom,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 3, pp. 383–388, 2019, doi: 10.29207/resti.v3i3.1119.
- [13] A. Taufik, P. Studi, and S. Informasi, “Optimasi Particle Swarm Optimization Sebagai Seleksi Fitur Pada Analisis Sentimen Review Hotel Berbahasa Indonesia Menggunakan Algoritma Naïve Bayes,” vol. III, no. 2, pp. 40–47, 2017.