

PERHITUNGAN ANALISIS SENTIMEN BERBASIS KOMPARASI ALGORITMA NAIVE BAYES DAN K-NEAREST NEIGHBOUR BERBASIS PARTICLE SWARM OPTIMIZATION PADA KOMENTAR INSIDEN PEMBALAP MOTOGP 2015

Jehan Saptia Kurnia
Universitas Dirgantara Marsekal Suryadarma
jehansaptiakurnia@gmail.com

Abstrak

Media untuk mendapatkan informasi tentang berita pembalap motogp sangat banyak seperti media tv, radio, koran, majalah, website dan lain-lain. Tetapi dari sebagian besar media tersebut website merupakan media yang sangat fleksibel karena dapat di akses di berbagai macam tempat yang terkoneksi jaringan internet, informasi yang disediakan sangatlah up to date dan juga setiap orang bisa mengomentari artikel yang terkait. Informasi yang menyebar sangat cepat dan di iringi dengan kebebasan mengeluarkan pendapat dapat menimbulkan berbagai jenis opini, baik opini negatif atau positif. Dari beberapa teknik klasifikasi yang paling sering digunakan adalah *Naive Bayes* dan *k-Nearest Neighbour* (KNN). *Naive Bayes* merupakan sebuah pengklasifikasi sederhana yang mengaplikasikan *Teorema Bayes* dengan ketidaktergantungan (independent) yang tinggi. *k-Nearest Neighbour* (KNN) algoritma klasifikasi memprediksi kategori tes sampel sesuai dengan sampel pelatihan K yang terdekat tetangga untuk sampel uji, dan hakim untuk kategori yang memiliki kategori terbesar probabilitas. Oleh karena itu, dalam penelitian ini menggunakan penggabungan metode pemilihan fitur, yaitu *Particle Swarm Optimization* agar bisa meningkatkan akurasi pada *Naive Bayes* dan *k-Nearest Neighbour*. Adapun akurasi yang dihasilkan pada algoritma *Naive Bayes* berbasis *Particle Swarm Optimization* dengan akurasi 82.67%. dan *k-Nearest Neighbour* berbasis *Particle Swarm Optimization* dengan akurasi 71.33%.Dapat disimpulkan bahwa penerapan optimasi dapat meningkatkan akurasi. Model di *Naive Bayes* berbasis *Particle Swarm Optimization* dapat memberikan solusi terhadap permasalahan klasifikasi review opini publik berita insiden pembalap motogp agar lebih akurat dan optimal. untuk model di *k-Nearest Neighbour* berbasis *Particle Swarm Optimization* akurasi menurun.

Kata Kunci : Media, Klasifikasi, *Naive Bayes*, *k-Nearest Neighbour*, *Particle Swarm Optimization*

Abstract

Media to get information about news MotoGP rider very much like media TV, radio, newspapers, magazines, websites and others. But from most of the media is a media website which is very flexible because it can be accessed from a wide variety of places connected to the Internet, the information provided is up to date and everyone can comment on articles related. The information spreads very fast and is accompanied by the freedom of speech can cause various types of opinions, either negative or positive opinion. Classification techniques of some of the most frequently used is Naive Bayes and k-Nearest Neighbour KNN). Naive Bayes classifier is a simple applying Bayes Theorem to independence (independent) high. K-Nearest Neighbor (KNN) classification algorithm predicts the category of the test sample in accordance with the training sample K nearest neighbor to the test sample, and a judge for the category that has the largest category of probability. Therefore, in this study using the merging feature selection methods, namely particle Swarm Optimization in order to improve the accuracy on Naive Bayes and k-Nearest Neighbour. As for the resulting accuracy Naive Bayes algorithm based on Particle Swarm Optimization with an accuracy of 82.67%. and k-Nearest Neighbour-based Particle Swarm Optimization with an accuracy of 71.33% It can be concluded that the application of optimization can improve accuracy. Model in Naive Bayes-based Particle Swarm Optimization can provide solutions to the problems of classification review of public opinion news MotoGP racer incident in order to more accurately and optimally. for the model-based k-Nearest Neighbour Particle Swarm Optimization accuracy decreases.

Keywords : Media, Classification, Naive Bayes, k-Nearest Neighbour, Particle Swarm Optimization, Text Mining.

1. Pendahuluan

Popularitas MotoGP tengah dipertaruhkan setelah Race Direction resmi memberikan hukuman kepada Pembalap Yamaha Valentino Rossi. Pengumuman melalui akun Twitter resmi MotoGP yang berisikan memberikan tiga poin penalti yang membuatnya harus memulai balapan dari posisi terakhir di balapan MotoGP Valencia. Sistem poin penalti itu sendiri dipergunakan MotoGP untuk mengantisipasi perilaku negatif, semisal membahayakan pembalap lainnya. Jumlah poin penalti berkisar antara satu sampai 10, dengan sanksi diterapkan pada batas tertentu. Keputusan yang telah dinyatakan sudah direspon oleh tim Rossi, dengan cara mengajukan naik banding terhadap hukuman tersebut. Penyebab Pembalap Rossi mendapatkan hukuman dari Race Direction dikarenakan terkait adanya insiden yang melibatkan dirinya dengan pembalap Repsol Honda Marc Marquez terjatuh dan tidak bisa melanjutkan balapan di tikungan empat belas Sirkuit Internasional Sepang (SIC) pada Minggu, 25 Oktober 2015. Hukuman dari Race Direction ini ditanggapi beragam macam pendapat dari masyarakat, baik pendapat positif maupun negatif.

Teks mining adalah sebuah teknologi baru yang berusaha untuk mengekstrak informasi yang bermakna dari data tekstual terstruktur. Teks mining adalah perluasan dari penggalian data untuk data tekstual (Wu He., et al dalam Ananiadou et al., 2013). Text mining melibatkan semua kegiatan dalam penemuan informasi dan data penting lainnya dari berbagai sumber-sumber tekstual (Hashimi., et al dalam Achtert et al., 2015).

Teks mining difokuskan untuk menemukan model, tren, pola atau aturan dari data tekstual terstruktur seperti file

teks, HTML file, pesan chat dan email (Wu He., et al dalam Abdous., et al 2013). Sebagai teknik otomatis, teks mining dapat digunakan untuk "efficiently dan mengidentifikasi secara sistematis, ekstrak, mengelola, mengintegrasikan, dan memanfaatkan pengetahuan dari teks (Wu He., et al dalam Ananiadou, 2013).

Klasifikasi merupakan suatu metode untuk mengelompokkan sebuah objek ke dalam kelompok atau kelas tertentu. Berbagai kasus yang berkaitan dengan pengelompokkan objek dapat diselesaikan lebih mudah dengan menerapkan teknik-teknik klasifikasi. Dalam memecahkan masalah klasifikasi, para ahli telah mengembangkan berbagai metode klasifikasi. Dibidang soft computing, mulai banyak dikembangkan juga teknik-teknik klasifikasi sehingga proses klasifikasi dapat dilakukan dalam waktu yang relatif lebih cepat dengan menggunakan algoritma klasifikasi yang tepat (Syafitri, 2010). Untuk itulah penelitian ini menggunakan metode Naive Bayes untuk klasifikasi teks.

Naive Bayes merupakan sebuah pengklasifikasi sederhana yang mengaplikasikan Teorema Bayes dengan ketidaktergantungan (independent) yang tinggi. Teorema bayes adalah teorema yang dipakai dalam statistika yang digunakan untuk menghitung peluang untuk suatu hipotesis. Metode pengklasifikasian dengan menggunakan metode statistik dan probabilitas yang dikemukakan dari ilmuwan inggris yaitu Thomas Bayes memprediksi peluang pada masa depan berdasarkan pengalaman pada masa sebelumnya, Sehingga dikenal sebagai Teorema Bayes.

K-Nearest Neighbour (KNN) algoritma klasifikasi memprediksi kategori tes sampel sesuai dengan sampel pelatihan K yang terdekat tetangga untuk

sampel uji, dan hakim untuk kategori yang memiliki kategori terbesar probabilitas. Proses algoritma KNN untuk mengklasifikasikan sampel X (Xuesong, Wei, Li., Et al, 2013).

Klasifikasi teks biasa K-Nearest Neighbour (KNN) memiliki tiga keterbatasan: 1. Tinggi perhitungan kompleksitas: Untuk mengetahui k terdekat sampel tetangga, semua kesamaan antara sampel pelatihan harus dihitung. Ketika nomor sampel pelatihan kurang, classifier KNN ada lagi optimal, tetapi jika training set berisi besar jumlah sampel, classifier KNN membutuhkan lebih banyak waktu untuk menghitung kesamaan (Muryan Awaludin dan Yuhanes, 2017). Masalah ini dapat diatasi dalam 3 cara: mengurangi dimensi ruang fitur; menggunakan data set yang lebih kecil; menggunakan algoritma yang ditingkatkan dapat mem-percepat; 2. Ketergantungan pada training set: classifier adalah dihasilkan hanya dengan sampel pelatihan dan itu tidak menggunakan data tambahan. Hal ini membuat algoritma untuk tergantung pada pelatihan set berlebihan; perlu perhitungan kembali bahkan jika ada perubahan kecil pada pelatihan set; 3. Tidak ada perbedaan antara berat sampel: Semua pelatihan sampel diperlakukan sama; tidak ada perbedaan antara sampel dengan jumlah kecil data dan sejumlah besar data. Sehingga tidak sesuai dengan yang sebenarnya fenomena di mana sampel memiliki rata distribusi umum (Suguna dan K, Thanushkodi. 2010).

Partikel swarm optimasi (PSO) adalah algoritma kecerdasan populasi-based swarm yang pada awalnya diusulkan by Kennedy dan Eberhart dan mensimulasikan perilaku sosial of social organisme dengan menggunakan gerakan fisik dari individu-uals di swarm (Marinakos, 2015).

Partikel swarm optimasi (PSO) telah populer menjadi optimisasi global dengan sebagian besar permasalahan dapat diselesaikan dengan baik di mana variabel-variabelnya adalah bilangan riil. Beberapa istilah umum yang biasa digunakan dalam Optimisasi Particle Swarm dapat didefinisikan sebagai berikut:

1. Swarm: populasi dari suatu algoritma,
2. Particle: anggota (individu) pada suatu swarm. Setiap particle merepresentasikan suatu solusi yang potensial pada permasalahan yang diselesaikan. Posisi dari suatu particle adalah ditentukan oleh representasi solusi saat itu.
3. Pbest (Personal best): posisi Pbest suatu partikel yang menunjukkan posisi partikel yang dipersiapkan untuk mendapatkan suatu solusi yang terbaik.
4. Gbest (Global best) : posisi terbaik particle pada swarm.
5. Velocity (vektor): vektor yang menggerakkan proses optimisasi yang menentukan arah di mana suatu particle diperlukan untuk berpindah (move) untuk memperbaiki posisinya semula.
6. Inertia weight : inertia weight di simbolkan w , parameter ini digunakan untuk mengontrol dampak dari adanya velocity yang diberikan oleh suatu particle (Tuegeh, Soeprijanto, Purnomo, 2009).

Untuk itu, penelitian ini menggunakan Particle Swarm Optimization sebagai seleksi fitur untuk pendapat masyarakat mengenai insiden pembalap Valentino Rossi dan pembalap Marc Marquez dengan menggunakan Naive Bayes dan K-Nearest Neighbour.

2. Perancangan Penelitian

Pada dasarnya, penelitian merupakan suatu investigasi yang terorganisasi,

yang dilakukan untuk menyajikan suatu informasi dan memecahkan masalah. Metode penelitian yang digunakan penulis menggunakan metode penelitian eksperimen. Adapun metode penelitian yang penulis gunakan melalui beberapa tahapan sebagai berikut:

1. Pengumpulan Data

Data yang digunakan untuk melakukan eksperimen dikumpulkan melalui website www.sport.detik.com, www.sports.sindonews.com dan www.tribunnews.com, kemudian data pendapat publik berita sepak bola tersebut diseleksi dan dikumpulkan ke dalam notepad untuk diolah dalam pengujian data.

2. Pengolahan Data awal

Memilih metode yang akan digunakan pada saat pengujian data. Metode yang dipilih, berdasarkan penelitian yang terdahulu. Penulis menggunakan Metode Algoritma *Naive Bayes* dan *k-Nearest Neighbour*.

3. Metode yang Diusulkan

Metode yang diusulkan penulis ditambahkan optimasi agar dapat meningkatkan nilai akurasi. Optimasi yang digunakan yaitu *Particle Swarm Optimization (PSO)*. *Naive Bayes* berbasis *Particle Swarm Optimization* dan *K-Nearest Neighbour* berbasis *Particle Swarm Optimization*.

4. Eksperimen dan Pengujian Metode

Eksperimen yang dilakukan peneliti, menggunakan framework RapidMiner 6.4 untuk mengolah data sehingga menghasilkan nilai akurasi yang akurat dan untuk pengujian metode penulis membuat aplikasi menggunakan bahasa pemrograman PHP dan HTML.

5. Evaluasi dan Validasi Hasil Evaluasi

Evaluasi berfungsi untuk mengetahui akurasi dari model algoritma yang diusulkan. Validasi digunakan untuk melihat perbandingan hasil akurasi dari model yang digunakan dengan hasil yang telah ada sebelumnya. Teknik validasi yang digunakan adalah *Cross Validation*. Akurasi algoritma akan diukur menggunakan *Confusion Matrix* dan hasil perhitungan akan ditampilkan dalam bentuk *Curve ROC (Receiver Operating Characteristic)*.

3. Hasil Dan Pembahasan

Data training yang digunakan pada saat pengujian data diambil dari www.sport.detik.com, www.sports.sindonews.com dan www.tribunnews.com. Pengujian data, dilakukan dengan menggunakan komentar insiden pembalap motogp (300 *data training*, yang terdiri dari 150 komentar negatif dan 150 komentar positif) kemudian dilakukan *testing* dan training dataset sehingga didapatkan *accuracy* dan AUC (*Area Under Curve*). Berikut akan dijelaskan lebih rinci mengenai hasil penelitian yang diperoleh.

3.1 Klasifikasi Text Menggunakan Algoritma *Naive Bayes*

Berikut merupakan tahapan-tahapan dalam melakukan pengolahan data, yaitu:

1. Pengumpulan Data

Komentar insiden pembalap motogp, masing-masing dikelompokkan dengan cara disimpan ke dalam satu folder yaitu folder positif dan folder negatif, kemudian tiap dokumennya diberikan ekstensi.txt sehingga dapat dibuka dengan aplikasi Notepad.

2. Pengolahan Data Awal (*Preprocessing*)

Berikut merupakan tahapan yang dilakukan dalam *preprocessing* :

a. *Tokenization*

Dalam proses *tokenization* ini, semua kata yang ada didalam tiap dokumen dikumpulkan dan dihilangkan tanda baca, serta dihilangkan jika terdapat simbol, karakter khusus atau apapun yang bukan huruf.

b. Filter Tokens (By Length)

Dalam proses ini, kata-kata yang memiliki panjang kurang dari 4 dan lebih dari 25 akan dihapus, seperti kata yang, tidak, jd, ga, ane, gan yang merupakan kata-kata yang tidak mempunyai makna tersendiri jika dipisahkan dengan kata yang lain dan tidak terkait dengan kata sifat yang berhubungan dengan sentiment.

c. Stopwords Removal

Dalam proses ini, *Stopwords Removal* yang digunakan adalah operator *Filter Stopwords (Dictionary)* karena dataset yang digunakan berbahasa indonesia, yang sebelumnya penulis telah membuat terlebih dahulu daftar kata-kata yang termasuk *stopwords* kemudian file nya dimasukkan ke dalam operator tersebut. Dalam proses ini, kata-kata yang tidak relevan akan dihapus, seperti kata tetapi, untuk, dengan, yang merupakan kata-kata yang tidak mempunyai makna tersendiri jika dipisahkan dengan kata yang lain dan tidak terkait dengan kata sifat yang berhubungan dengan sentiment.

d. Transform Cases

Dalam proses *transform cases* ini, semua huruf dirubah menjadi huruf kecil semua atau huruf kapital semua.

3. Klasiikasi

Proses menentukan bagian kalimat sebagai anggota *class positif* atau *class negatif* berdasarkan nilai perhitungan *Naive Bayes*, jika hasilnya kalimat

tersebut untuk *class positif* lebih besar dari pada *class negatif*, maka kalimat tersebut termasuk ke dalam *class positif*, jika probabilitas untuk *class positif* lebih kecil dari pada *class negatif*, maka kalimat tersebut termasuk ke dalam *class negatif*.

3.2 Hasil Eksperimen Pengujian Metode

1. Metode Naive Bayes

Hasil pengujian data dengan menggunakan *Metode Naive Bayes* menghasilkan *Accuracy*=78.67% dan *AUC* sebesar 0.611.

accuracy: 78.67% +/- 9.21% (mikro: 78.67%)			
	true Positif	true Negatif	class precision
pred. Positif	108	22	83.08%
pred. Negatif	42	128	75.29%
class recall	72.00%	85.33%	

AUC: 0.611 +/- 0.123 (mikro: 0.611) (positive class: Negatif)

2. Metode Naive Bayes berbasis Particle Swarm Optimization

Hasil pengujian data dengan menggunakan *Metode Naive Bayes* berbasis *Particle Swarm Optimization* menghasilkan *Accuracy*= 82.00% dan *AUC* sebesar 0.681.

accuracy: 82.00% +/- 6.00% (mikro: 82.00%)			
	true Positif	true Negatif	class precision
pred. Positif	112	16	87.50%
pred. Negatif	38	134	77.91%
class recall	74.67%	89.33%	

AUC: 0.681 +/- 0.149 (mikro: 0.681) (positive class: Negatif)

3. Metode k-Nearest Neighbour (k-NN)

Metode *k-Nearest Neighbor* merupakan salah satu metode berbasis NN yang paling populer. Nilai k yang digunakan menyatakan jumlah tetangga terdekat yang dilibatkan dalam penentuan prediksi label kelas pada data uji. Untuk memperkirakan nilai k yang terbaik, bisa

dilakukan dengan menggunakan teknik validasi silang (*Cross Validation*). Jika nilai k terlalu kecil, maka berakibat hasil prediksi yang didapat bisa sensitif terhadap keberadaan *noise*, namun jika k terlalu besar maka tetangga terdekat yang terpilih mungkin terlalu banyak dari kelas

lain yang sebenarnya tidak relevan karena jarak yang terlalu jauh. Berikut adalah hasil pengujian data dengan cara melakukan uji coba memasukan nilai k (jumlah tetangga terdekat) pada RapidMiner.

Tabel III.1
Eksperimen Penentuan Nilai Training k -NN

Nilai k	Accuracy	AUC
1	54.00%	0.500
2	50.00%	0.540
3	50.00%	0.547
4	50.00%	0.610
5	55.67%	0.741
6	54.33%	0.799
7	69.33%	0.816
8	62.33%	0.819
9	70.67%	0.817
10	66.33%	0.812

Hasil pengujian menunjukkan bahwa penerapan metode *k-Nearest Neighbour* pada tabel IV.5 dengan penentuan nilai $k=9$ menghasilkan *Accuracy* = 70,67% dan *AUC* = 0.817 adalah nilai yang paling tinggi.

4. Metode *k-Nearest Neighbour* (k -NN) berbasis *Particle Swarm Optimization*

Penelitian metode *k-Nearest Neighbor* berbasis PSO, dengan melakukan uji coba nilai k sebagai tetangga terdekat, population size=5. Adapun hasil dari perhitungannya ditunjukkan pada Tabel III.2:

Tabel III. 2
Eksperimen Penentuan Nilai Training k -NN berbasis PSO

K	k-NN		Population Size	K	k-NN & PSO	
	Accuracy	AUC			Accuracy	AUC
1	54.00%	0.500	5	1	70.33	0.500
2	50.00%	0.540	5	2	58.33%	0.569
3	50.00%	0.547	5	3	60.33%	0.593
4	50.00%	0.610	5	4	54.00%	0.597
5	55.67%	0.741	5	5	58.00%	0.601
6	54.33%	0.799	5	6	54.67%	0.610
7	69.33%	0.816	5	7	63.00%	0.675
8	62.33%	0.819	5	8	63.00%	0.640

9	70.67%	0.817	5	9	61.67%	0.688
10	66.33%	0.812	5	10	63.33%	0.652

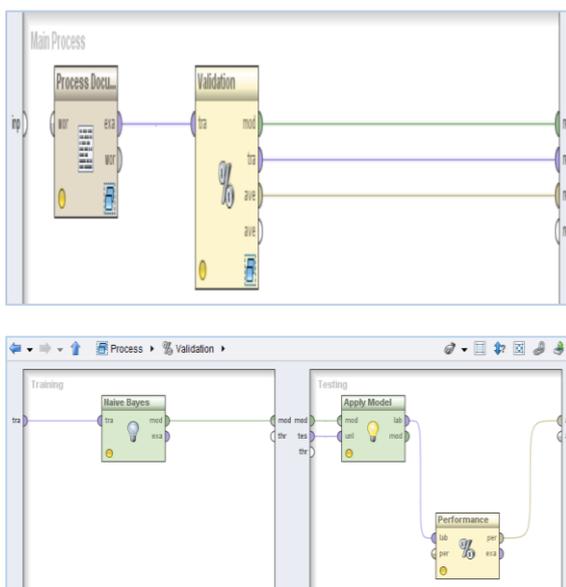
Hasil perhitungan dari Tabel IV.2 di atas menunjukkan *k-Nearest Neighbour* dengan memasukan nilai $k=9$ mendapatkan *Accuracy*=70.67% dan *AUC*=0.817 merupakan nilai yang lebih tinggi. Dibandingkan dengan memasukan *k-Nearest Neighbour* berbasis *Particle Swarm Optimization* nilai $k=1$ mendapatkan *Accuracy*=70.33% dan *AUC*=0.500.

Validation adalah proses untuk mengevaluasi keakuratan prediksi dari model. Validasi digunakan untuk memperoleh prediksi menggunakan model yang ada dan kemudian membandingkan hasil tersebut dengan

hasil yang sudah diketahui, ini mewakili langkah paling penting dalam proses membangun sebuah model (Mabrur dan Lubis, 2012).

Hasil dari pengujian model yang dilakukan adalah mengklasifikasikan komentar insiden pembalap motogp negatif dan komentar insiden pembalap motogp positif menggunakan algoritma *Naive Bayes (NB)* dan *Naive Bayes (NB) Particle Swarm Optimization (PSO)*, *k-Nearest Neighbor (k-NN)* dan *k-Nearest Neighbor (k-NN) Particle Swarm Optimization (PSO)* untuk mengetahui nilai akurasi terbaik pada framework RapidMiner dan desain model berikut ini:

3.2.1. Hasil Pengujian Model Metode Naive Bayes

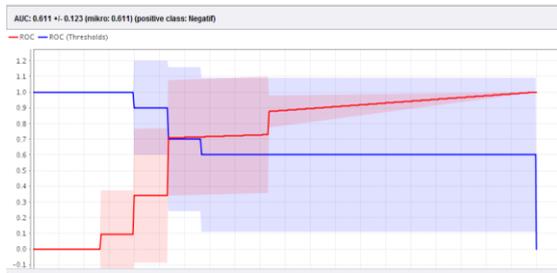


Berdasarkan hasil dari pengujian model yang diatas adalah mengklasifikasikan komentar publik mengenai insiden pembalap motogp negatif dan komentar publik mengenai insiden pembalap motogp positif dengan menggunakan algoritma *Naive Bayes*.

a. Kurva ROC

Kurva ROC (*Receiver Operating Characteristic*) adalah cara lain untuk mengevaluasi akurasi dari klasifikasi secara visual. Sebuah grafik ROC adalah plot dua dimensi dengan proporsi positif salah pada sumbu X dan positif benar

pada sumbu Y. Hasil perhitungan pada kurva ROC, menggambarkan kurva ROC untuk algoritma *Naive Bayes*. Dapat disimpulkan bahwa satu point pada kurva ROC adalah lebih baik dari pada yang lainnya jika arah garis melintang dari kiri



bawah ke kanan atas didalam grafik. Kurva ROC *Naive Bayes* dengan nilai AUC (*Area Under Curve*) sebesar 0.611 dimana diagnosa hasilnya fair classification. Berikut dibawah ini dapat dilihat Kurva ROC *Naive Bayes*:

b. Confusion Matrix

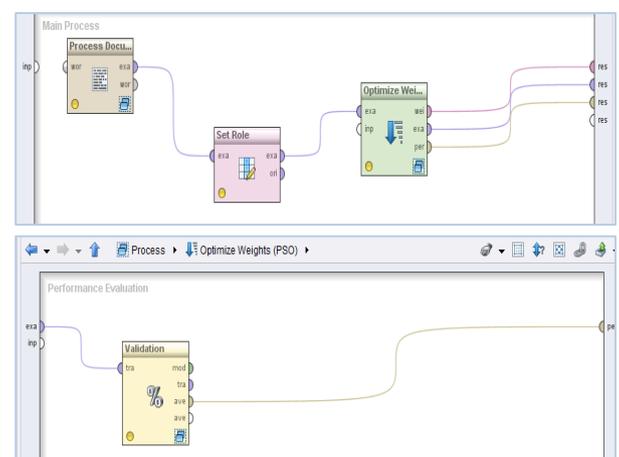
Memberikan keputusan yang diperoleh dalam training dan testing, confusion matrix memberikan penilaian performance klasifikasi berdasarkan objek benar atau salah. *Confusion matrix* berisi informasi aktual (actual) dan prediksi (predicted) pada sistem klasifikasi.

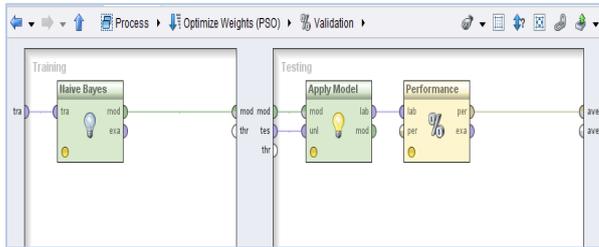
accuracy: 78.67% +/- 9.21% (mikro: 78.67%)			
	True Positif	True Negatif	Class Precision
Prediksi Negatif	108	22	83.08%
Prediksi Positif	42	128	75.29%
Class Recall	72.00%	85.33%	

$$\text{Acc (Accuracy)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{108 + 128}{108 + 22 + 42 + 128} = \frac{236}{300} = 0.78$$

Akurasi yang diperoleh yaitu 78.67% dari 150 data komentar positif opini publik insiden pembalap motogp dan 150 data komentar negatif opini publik insiden pembalap motogp. Data komentar yang dihasilkan RapidMiner dengan model Naive Bayes menunjukkan bahwa klasifikasi untuk komentar negatif yang sesuai prediksi yaitu **108** data. Data komentar negatif yang termasuk ke dalam prediksi negatif yaitu **22**. Dara komentar positif yang termasuk ke dalam prediksi positif yaitu **42** dan data komentar positif yang termasuk ke dalam prediksi negatif yaitu **128**.

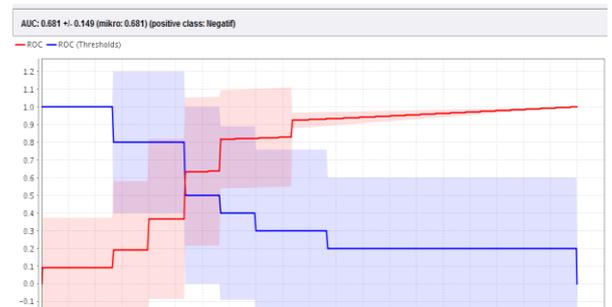
3.2.2 Hasil Pengujian Model Metode *Naive Bayes* berbasis *Particle Swarm Optimization*





Hasil pengujian data training metode *Naive Bayes* berbasis *Particle Swarm Optimization* menggunakan *Set Role* yang berfungsi untuk menentukan optimasi menggunakan *Particle Swarm Optimization* agar akurasi yang dihasilkan lebih tinggi. Pengukuran akurasi tersebut, akan dijabarkan melalui Kurva ROC dan *Confusion Matrix* di bawah ini:

a. Kurva ROC



Kurva ROC yang dihasilkan berdasarkan pengujian data pada gambar di atas, menunjukkan bahwa ada peningkatan pada akurasi menggunakan *Naive Bayes* berbasis *Particle Swarm Optimization* sebesar **82.00%** dan AUC sebesar **0.681**.

b. Confusion Matrix

accuracy: 82.00% +/- 6.00% (mikro: 82.00%)			
	True Positif	True Negatif	Class Precision
Prediksi Negatif	112	16	87.50%
Prediksi Positif	38	134	77.91%
Class Recall	74.67%	89.33%	

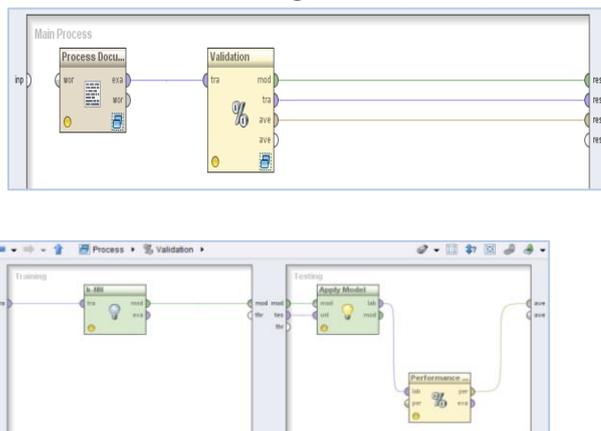
$$\text{Acc (Accuracy)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{142 + 70}{142 + 80 + 8 + 70} = \frac{212}{300} = 0.70$$

Data training yang digunakan terdiri dari **150** data komentar positif mengenai insiden pembalap motogp dan **150** data komentar negatif pada insiden pembalap motogp. Data komentar negatif, setelah melalui beberapa tahap pengolahan pada RapidMainer dengan model *Naive Bayes* berbasis *Particle*

Swarm Optimization, diklasifikasikan untuk komentar negatif yang sesuai prediksi sebanyak **112** data, kemudian **16** data yang diprediksi negatif namun masuk kedalam kategori komentar negatif. Sedangkan untuk data komentar positif, yang diprediksi bahwa data tersebut positif adalah **38**, dan untuk

prediksi komentar positif yang masuk dalam prediksi komentar negatif adalah **134** data, hasil akurasi yang muncul adalah **82.00%**.

3.2.3. Hasil Pengujian Model Metode k-Nearest Neighbors (k-NN)



Hasil pengujian data pada gambar IV.9 di atas, dengan Metode *k-Nearest*

Neighbor menggunakan nilai uji coba **k=9** yang memiliki akurasi tertinggi. Adapun hasil perhitungan *accuracy*, *class precision* dan *class recall*. Dapat dilihat pada Tabel IV.6 di bawah ini:

a. Kurva ROC



b. Confusion Matrix

Data training yang digunakan terdiri dari 150 data komentar positif mengenai insiden pembalap motogp dan 150 data komentar negatif. Data komentar negatif, setelah melalui beberapa tahap pengolahan pada RapidMainer dengan model *k-Nearest Neighbour* (k-NN),

accuracy: 70.67% +/- 8.54% (mikro: 70.67%)			
	True Positif	True Negatif	Class Precision
Prediksi Negatif	142	80	63.96%
Prediksi Positif	8	70	89.74%
Class Recall	94.67%	46.67%	

$$\text{Acc (Accuracy)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{140 + 71}{140 + 79 + 10 + 71} = \frac{211}{300} = 0.70$$

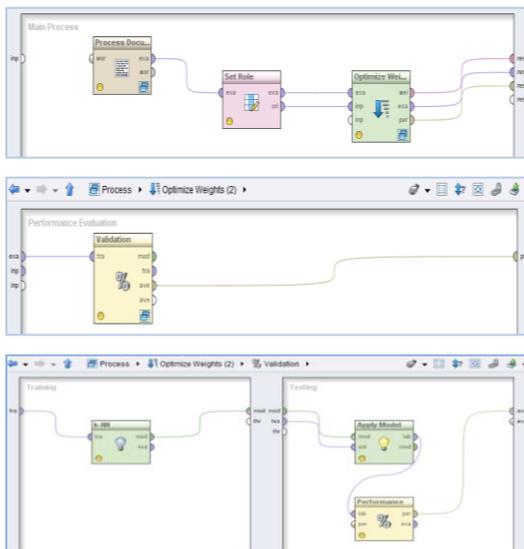
Akurasi yang diperoleh yaitu 70.67% dari 150 data komentar positif opini publik insiden pembalap motogp dan 150 data komentar negatif opini publik insiden pembalap motogp. Data komentar yang dihasilkan RapidMiner dengan model *k-Nearest Neighbour*

menunjukkan bahwa klasifikasi untuk komentar negatif yang sesuai prediksi yaitu **142**. Data komentar negatif yang termasuk ke dalam prediksi negatif yaitu **80**. Data komentar positif yang termasuk ke dalam prediksi positif yaitu **8** dan data

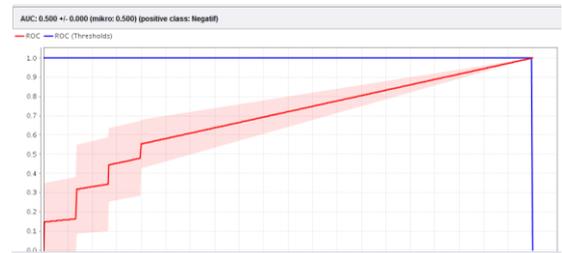
komentar positif yang termasuk ke dalam prediksi negatif yaitu **70**.

3.2.4. Hasil Pengujian Model Metode *k-Nearest Neighbors* (k-NN) berbasis *Particle Swarm Optimization*

Model Pengujian data untuk menentukan hasil komentar insiden pembalap motogp menggunakan algoritma *k-Nearest Neighbor* berbasis *Particle Swarm Optimization* pada framework RapidMiner sebagai berikut :



a. Kurva ROC



Hasil perhitungan divisualisasikan dengan Kurva ROC. Perbandingan dengan kedua metode komparasi bisa dilihat pada Gambar IV.10 yang merupakan Kurva ROC untuk algoritma *k-Nearest Neighbor* dan Gambar IV.12 merupakan Kurva ROC untuk algoritma *k-Nearest Neighbor* berbasis *Particle Swarm Optimization*. Kurva ROC yang dihasilkan berdasarkan pengujian data pada gambar di atas, menunjukkan bahwa ada penurunan pada akurasi menggunakan *k-Nearest Neighbor* berbasis *Particle Swarm Optimization*, sebesar **70.33%** dan AUC sebesar **0.500**.

a. Confusion Matrix

accuracy: 70.33% +/- 6.74% (mikro: 70.33%)			
	True Positif	True Negatif	Class Precision
Prediksi Negatif	140	79	63.93%
Prediksi Positif	10	71	87.65%
Class Recall	93.33%	47.33%	

$$\text{Acc (Accuracy)} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{140 + 71}{140 + 79 + 10 + 71} = \frac{211}{300} = 0.70$$

Data komentar yang dihasilkan RapidMiner dengan model *k-Nearest*

Neighbour berbasis *Particle Swarm Optimization* menunjukkan bahwa

klasifikasi untuk komentar negatif yang sesuai prediksi yaitu **140**. Data komentar negatif yang termasuk ke dalam prediksi negatif yaitu **79**. Data komentar positif yang termasuk ke dalam prediksi positif yaitu **10**. Dan data komentar positif yang termasuk ke dalam prediksi negatif yaitu **71**.

3.2.5. Pembahasan

Berdasarkan data komentar yang sudah diolah melalui RapidMiner, kemudian hasilnya terpisah menjadi kata-kata. Kata-kata tersebut, masing-masing memiliki bobot sehingga dapat dilihat kata mana saja yang berhubungan dengan sentimen yang sering muncul dan memiliki bobot tertinggi. Dengan demikian dapat diketahui komentar tersebut termasuk ke dalam komentar insiden pembalap motogp positif dan komentar insiden pembalap motogp negatif. Berdasarkan pengujian yang telah dilakukan terhadap komentar insiden pembalap motogp dengan menggunakan metode *Naive Bayes*, *Naive Bayes berbasis Particle Swarm Optimization*, *k-Nearest Neighbor*, *k-Nearest Neighbor berbasis Particle Swarm Optimization*. Penerapan *Particle Swarm Optimization* (PSO) dapat meningkatkan akurasi pada klasifikasi komentar insiden pembalap motogp untuk mengidentifikasi antara komentar positif dan komentar negatif pada metode

Naive Bayes berbasis *Particle Swarm Optimization*. sedangkan pada metode *k-Nearest Neighbor* jika dioptimasi menggunakan *Particle Swarm Optimization* (PSO) akurasi yang didapatkan menurun. Dalam penelitian ini, hasil perhitungan metode *Naive Bayes* (NB) memiliki *Accuracy* sebesar 78.67% dan AUC sebesar 0.611. Sedangkan Metode *Naive Bayes* berbasis *Particle Swarm Optimization* (NB PSO) menghasilkan *Accuracy* sebesar 82.00% dan AUC sebesar 0.681. Hal ini menunjukkan bahwa penggunaan optimasi *Particle Swarm Optimization* dapat meningkatkan nilai akurasi *Naive Bayes* (NB), kemudian pengujian juga akan dibandingkan dengan metode *k-Nearest Neighbor*. Hasil perhitungan yang diperoleh dari pengujian data dengan metode *k-Nearest Neighbor* (k-NN) yaitu *Accuracy* sebesar 70.67% dan AUC sebesar 0.817, kemudian dibandingkan dengan *k-Nearest Neighbor* berbasis *Particle Swarm Optimization* (k-NN PSO) menghasilkan nilai *Accuracy* sebesar 70.33% dan AUC sebesar 0.500. Dengan ini dapat disimpulkan bahwa *Naive Bayes* berbasis *Particle Swarm Optimization* (NB PSO) dengan *k-Nearest Neighbor* berbasis *Particle Swarm Optimization* (k-NN PSO) lebih tinggi nilai akurasi *Naive Bayes Particle Swarm Optimization* (NB PSO).

DAFTAR ACUAN

- Basari et al. 2013. *Opinion Mining of Movie Review using Hybrid Method of Support Vector Machine and Particle Swarm Optimization*. *Procedia Engineering*, 53, 453-462.
- Bijalwan, Vishwanath., Pinki, Kumari., Jordan, Pascual & Vijay, Bhaskar, Semwal. *Machine learning approach for text and document mining*. India.
- Gupta, Vishal & Gurpreet, S. Lehal. 2009. *A survey of Text Mining Techniques and Application*. *Journal of Emerging Technologies in Web Intelligence*, Vol. 1, No.1, August 2009.
- Han, J., & Kamber, M. 2007. *Data Mining Concepts and Techniques*. San.

- Hashimi, Hussein; Alaaeldin, Hafez; Hassan Mathkour. 2015. *Selection criteria for text mining approaches. Computers in Human Behaviour* 51 (2015) 729-733.
- He, Jie & Hui, Guo. 2013. *A Modified Particle Swarm Optimization Algorithm. Telkomnika*, Vol. 11, No. 10, October 2013, pp. 6209 ~ 6215. e-ISSN: 2087-278X.
- Hwa Lu et al. - 2010 - *Chinese text classification by the Naïve Bayes Classifier and the associative classifier with multiple confidence thr. Knowledge-Based Systems*, 23, 598–604.
- Jiang et al. 2012, *An improved K-nearest-neighbor algorithm for text categorization. Expert Systems with Applications*, 39, 1503-1509.
- Jiawei, H., Kamber, M., & Pei, J. 2012. *Data Mining: Concepts and Techniques Third Edition*. Waltham, MA: Morgan Kaufmann
- Kang, Yoo, Han. 2012, *Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. Expert Systems with Applications*, 39, 6000-6010
- Liu, Bing. 2012. *Sentiment Analysis And Opinion Mining*. Chicago: Morgan & ClaypoolPublisher.
- L, Joe., Villa, Medina., Ricard, Boqué., Joan Ferré. 2009. *Bagged k-nearest neighbours classification with uncertainty in the variables. Analytica Chimica Acta* 646 (2009) 62–68.
- Marinakakis, Yannis. 2015. *An Improved Particle Swarm Optimization Algorithm for the Capacitated Location Routing Problem and for The Location Routing Problem With Stochastic Demands. Applied Soft Computing* 37 (2015) 680-710.
- Moraes, R., Valiati, J. F., & Gavião Neto, W. P. 2013. *Document-level sentiment*.
- Awaludin, Muryan & Yuhannes, Pengembangan Algoritma Neural Network Berdasarkan Rentang Waktu Untuk Prediksi Harga Perdagangan Valuta Asing. *Jurnal SKI on SPOT*
- Ohana, Bruno. & Tierney, Brendan. 2009. *Sentiment Classification of Reviews Using SentiWordNet. 9th.IT & T Conference, Dublin Institute of Technology, Dublin, Ireland*, 22 – 23 October.
- Padmavathi,S., & Ramanujam,E. 2015. *Naive Bayes Classifier for ECG abnormalities using Multivariate Maximal Time Series Motif. Procedia Computer Science* 47 (2015) 222 – 228.
- S, Umajancy dan Antony, Selvadoss.T. 2013. *An Analysis On Text Mining –Text Retrieval And Text Extraction. International Journal of Advanced Research in Computer and Communication Engineering* Vol. 2, Issue 8, August 2013. ISSN (Print) : 2319-5940. ISSN (Online) : 2278-1021.
- Syafitri, Nesi. 2010. *Perbandingan Metode K-Nearest Neighbor (Knn) Dan Metode Nearest Cluster Classifier (Ncc) Dalam Pengklasifikasian Kualitas Batik Tulis. Jurnal Teknologi Informasi & Pendidikan* Issn : 2086 – 4981 Vol. 2 No. 1 September 2010.
- Tuegeh, Maickel., Soeprijanto., & Mauridhi, H. Purnomo. 2009. *Modified Improved Particle Swarm Optimization For Optimal Generator Scheduling*. Seminar Nasional Aplikasi Teknologi Informasi 2009(SNATI 2009). Yogyakarta, 20 Juni 2009.
- Vercellis, C. 2009. *Business Intelligence Data Mining And Optimization For Decision Making*. United Kingdom: A John Wiley And Sons, Ltd.,Publication.

- Wang, Aiguo; Ning, An; Guilin, Chen; Lian, Li; Gil Alterovitz. 2015. *Accelerating wrapper-based feature selection with K-nearest-neighbor*. *Knowledge-Based Systems* 83 (2015) 81–91.
- Wu He; Shenghua Zha; Ling li.2013. *Social Media Competitive Analysis and Text Mining: A Case Study In The Pizza Industry*. *International Journal of Information Management* 33 (2013) 464-472.
- Xiang et al. (2015), *A novel hybrid system for feature selection based on an improved gravitational search algorithm and k-NN method*. *Applied Soft Computing*, 31, 293-307.
- Zhao, M., Fu, C., Ji, L., Tang, K., & Zhou, M. (2011). *Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes*. *Expert Systems with Applications*, 38(5), 5197–5204. doi:10.1016/j.eswa.2010.10.041.