ANALISIS SENTIMEN PENGGUNA LRT JABODEBEK DI PLATFORM X MENGGUNAKAN ALGORITMA SVM

Rieka Ramadhina Fajriani, Desmulyati
Informatika, Universitas Bina Sarana Informatika
Jalan Kramat Raya No.98, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta
riekafajriani@gmail.com

ABSTRAK

Saat ini, kemacetan adalah masalah yang umum dan sering terjadi di berbagai kota di Indonesia, terutama di wilayah Jabodetabek. LRT Jabodebek merupakan salah satu upaya yang diluncurkan oleh pemerintah untuk mengurangi kemacetan dan polusi di wilayah Jabodebek. Kehadirannya di jalur strategis yang sering mengalami kemacetan membuat LRT Jabodebek menjadi pilihan banyak masyarakat. Berkat adanya kemajuan teknologi, pengguna LRT Jabodebek dapat dengan mudah berbagi pengalaman mereka secara *online* melalui platform X. Berdasarkan fenomena ini, penulis melakukan penelitian untuk mengevaluasi pandangan pengguna LRT Jabodebek yang tercermin dari *tweet* mereka. Untuk mendapat pandangan tersebut ialah melalui analisis sentimen. Penelitian ini menggunakan metode *Support Vector Machine* dengan pelabelan data yang dilakukan menggunakan leksikon Bahasa Indonesia. Dari data yang berhasil dikumpulkan melalui Tweet Harvest, diperoleh sebanyak 946 *tweet*, yang setelah dilabeli terbagi menjadi 687 sentimen positif dan 259 sentimen negatif. Setelah melakukan pengujian dengan rasio validasi split dan pengoptimalan parameter, didapatkan untuk kernel linear pada rasio 0.8 hasil akurasi 88,89%, sedamgkan pada rasio 0,7 hasil akurasi 86,62%. Sementara itu, untuk kernel RBF, pada rasio 0,8 hasil akurasi yang diperoleh adalah 87,83% dan pada rasio 0,7 hasil akurasi 86,62%.

Kata kunci: *Analisis Sentimen, LRT Jabodebek, RapidMiner, Support Vector Machine, X*

1. PENDAHULUAN

Transportasi memainkan peran yang sangat penting bagi manusia dalam mobilitas. Kendaraan pribadi adalah jenis transportasi yang umum digunakan, tetapi penggunaan kendaraan pribadi yang melebihi kapasitas jalan dapat mengakibatkan kemacetan. Di Indonesia, kemacetan adalah masalah yang umum dan sering terjadi di berbagai kota, terutama di wilayah Jabodetabek. Sebagai upaya untuk mengurangi kemacetan, pemerintah mendorong penggunaan transportasi publik sebagai pilihan moda transportasi sehari-hari masyarakat. Untuk itu, pemerintah telah menyediakan layanan transportasi publik yang cepat, mudah, dan terjangkau guna menekan tingkat kemacetan [1].

LRT Jabodebek adalah salah satu proyek infrastruktur yang dibangun oleh pemerintah untuk mengurangi kemacetan di Jabodebek. Dengan adanya LRT Jabodebek, perjalanan masyarakat menjadi lebih cepat dan mudah, karena LRT beroperasi di jalur layang. Salah satu keuntungan dari kemajuan teknologi adalah LRT Jabodebek beroperasi tanpa pengemudi, meskipun tetap didampingi oleh petugas operasional selama perjalanan. Keuntungan lain dari perkembangan teknologi ini adalah munculnya berbagai platform *online* yang memudahkan masyarakat untuk berbagi pendapat. Hal ini juga dilakukan oleh pengguna LRT Jabodebek, yang berbagi pengalaman mereka setelah menggunakan layanan transportasi ini.

Ada banyak platform *online* yang tersedia, tetapi platform yang paling sering digunakan oleh masyarakat Indonesia adalah X, berkat fitur-fiturnya yang mudah diakses. Menurut laporan dari Agensi Kreatif, We Are Social USA, Indonesia menempati peringkat keempat di dunia dalam jumlah pengguna X, dengan total 25,25 juta pengguna pada Juli 2023 [2].

Banyaknya pengguna di platform X memungkinkan opini yang dibagikan dalam bentuk *tweet* cepat diterima oleh pengguna lain. Dengan melakukan analisis mendalam terhadap *tweet* yang diunggah oleh pengguna LRT Jabodebek, dapat diperoleh sentimen yang berguna untuk memahami minat masyarakat terhadap keberadaan dan pelayanan LRT Jabodebek.

Analisis sentimen adalah analisis yang menggunakan teknik data mining untuk mengelompokkan kalimat ke dalam sentimen positif atau negatif. Analisis ini dapat dilakukan dengan berbagai metode, termasuk algoritma klasifikasi. Berdasarkan penelitian sebelumnya, algoritma klasifikasi yang memberikan akurasi tinggi dalam analisis sentimen adalah algoritma *Support Vector Machine* (SVM). Misalnya, penelitian oleh Zidna Alhaq dan rekan-rekannya pada tahun 2021 yang menggunakan metode SVM dengan *TF-IDF Cosine Similarity*, menghasilkan akurasi tertinggi yaitu 97% [3].

Penelitian lain oleh Enos Dwianto dan tim pada tahun 2021 membandingkan metode Naïve Bayes dan SVM, dengan akurasi SVM untuk GrabID mencapai 84,08% dan untuk GojekIndonesia 69,50%, sedangkan akurasi Naïve Bayes untuk GrabID adalah 64,36% dan untuk GojekIndonesia 50,90% [4].

2. TINJAUAN PUSTAKA

2.1. Analisis Sentimen

Analisis sentimen merupakan salah satu teknik dalam *text mining* dengan mengolah dan mengekstra teks terhadap suatu sasaran yang akan diteliti. Sasaran tersebut seperti layanan, topik, maupun produk [5]. Analisis sentimen akan mengkategorikan sebuah teks ke dalam sentimen positif, negatif, maupun netral.

2.2. *Pre-processing Data*

Pre-Processing Data adalah langkah awal yang perlu dilakukan sebelum memproses data dengan mesin. Tahapan *pre-processing* dapat bervariasi untuk setiap kasus. Namun, berdasarkan buku yang ditulis oleh [6], tahapan *pre-processing* yang biasa dilakukan untuk *text mining* antara lain pembersihan teks, tokenisasi, *stopwords removal*, dan *stemming*.

2.3. Algoritma *Support Vector Machine*

Support Vector Machine (SVM) adalah metode klasifikasi yang memanfaatkan fungsi linear dalam ruang berdimensi tinggi. Tujuan dari SVM adalah untuk menemukan *hyperplane* optimal yang dapat memisahkan data dengan kelas yang berbeda [7]. Algoritma ini cocok diaplikasikan untuk prediksi maupun klasifikasi [8].

Menurut penjelasan dalam buku oleh Jollyta [9], algoritma SVM dapat didefinisikan melalui persamaan yang menjelaskan cara kerjanya:

Persamaan untuk margin = $\frac{1}{w}$ (1)

Keterangan:
w = bobot margin

Persamaan untuk margin maksimum:

$$d(X^T) = \sum_{i=1}^{l} y_i a_i X_i X^T + b_0 \quad (2)$$

Keterangan:
$d(X^T)$ = margin maksimum
y_i = label kelas
X^T = data latih
a_i = nilai bobot dari titik data
b_0 = bias

Dengan adanya margin dari kedua sisi, persamaan hyperplane yang terbentuk adalah:

$$\frac{1}{2}\|w\|^2 = \frac{1}{2}({w_1}^2 + {w_2}^2) \quad (3)$$

Sehingga:

$$wx_i + b = 0 \text{ atau } w_1x_1 + w_2x_2 + b = 0 \quad (4)$$

Keterangan:
w = bobot margin
x = titik data
b = bias

Berikut adalah persamaan untuk kelas berdasarkan margin yang terdapat pada persamaan (1) dan (3):

$$y_i(wx_i + b) \geq 1 \; i = 1,2,3, \dots, n \quad (5)$$

Gabungan dari persamaan untuk kelas positif dan negatif adalah:

$$y_i(w_1x_1 + w_2x_2 + b) \geq 1 \quad (6)$$

$w_1x_1 + w_2x_2 + b \geq 1 \; for \; y_i = +1$ untuk kelas positif (7)

$w_1x_1 + w_2x_2 + b \leq -1 \; for \; y_i = -1$ untuk kelas negatif (8)

Keterangan:
b = bias
x = titik data
y = kelas

2.4. X

X merupakan sebuah platform yang dulunya bernama Twitter. X menyediakan fitur *tweet* untuk mengunggah opini atau perasaan pengguna melalui pesan teks dengan batas mencapai 280 kata [1].

Opini yang diunggah melalui fitur *tweet*, memudahkan pengguna untuk saling berbagi informasi.

2.5. RapidMiner

RapidMiner adalah perangkat lunak yang dirancang khusus untuk pengolahan *data mining*. *Software* ini dikembangkan oleh Dr. Markus Hofmann dari Institute of Technology Blanchardstown dan Ralf Klinkenberg dari rapid-i.com [7].

2.6. Confusion Matrix

Confusion Matrix merupakan hasil pengujian kinerja dari model klasifikasi yang telah dibuat [10]. Dalam *confusion matrix* terdapat empat parameter yang digunakan, yakni TP (*True Positive*), FP (*False Positive*), TN (*True Negative*), dan FN (*False Negative*).

Tabel 1. *Confusion matrix*

Nilai Prediksi		**Nilai Asli**	
		Positif (1)	**Negatif (0)**
	Positif (1)	TP	FP
	Negatif (0)	FN	TN

Dari parameter-parameter yang terdapat pada Tabel 1 dapat digunakan untuk menghitung nilai akurasi, *precision*, dan *recall*. Akurasi yaitu nilai dari seberapa akurat model yang telah diuji, *recall* yaitu untuk mengukur seberapa jauh model telah mendeteksi dengan benar kelas positif, dan *precision* yaitu untuk mengukur seberapa jauh model memprediksi benar untuk kelas positif [11]. Berikut rumus perhitungan untuk mencari nilai-nilai tersebut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP} \; (untuk \; kelas \; positif)$$

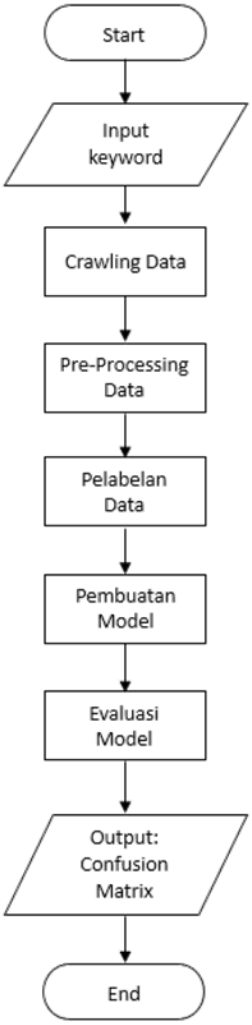
$$Precision = \frac{TN}{TN + FN} \; (untuk \; kelas \; negatif)$$

$$Recall = \frac{TP}{TP + FN} \; (untuk \; kelas \; positif)$$

$$Recall = \frac{TN}{TN + FP} \; (untuk \; kelas \; positif)$$

3. METODE PENELITIAN

Penelitian ini mengadopsi metode pendekatan kuantitatif, di mana analisis data dilakukan untuk menentukan sentimen positif atau negatif dari *tweet* menggunakan algoritma SVM. Selain itu, penelitian ini juga bertujuan untuk mengevaluasi kinerja model dengan menghitung metrik kinerja dari *confusion matrix*.



Gambar 1. Flowchart analisis data

3.1. *Crawling Data*

Crawling data adalah teknik pengambilan data melalui API. Data akan diambil dari platform X dengan menggunakan Tweet Harvest, yaitu alat yang digunakan untuk mengambil data dari pencarian di platform X berdasarkan kata kunci, bahasa, dan rentang waktu dari suatu isu [12].

Dataset yang dikumpulkan berasal dari kata kunci LRT Jabodebek, lrtjabodebek, dan @lrtjabodebek, dengan syarat *tweet* berbahasa Indonesia, tanpa *link*, dan bukan merupakan balasan. Selain itu, *tweet* tersebut dibatasi dari tanggal 28 Agustus 2023 hingga 15 Mei 2024.

3.2. *Pre-Processing Data*

Tahapan *pre-processing* yang dilakukan dalam penelitian ini mencakup:

a. *Cleaning*, teks dibersihkan dari elemen-elemen yang tidak diinginkan atau *noise*, yang meliputi simbol, angka, dan tanda baca.
b. *Case Folding*, teks yang telah dibersihkan diubah menjadi teks dengan format huruf kecil.
c. *Spell Normalization*, pada tahap ini dilakukan perbaikan pada kalimat yang mengandung ejaan singkat atau ejaan salah (*typo*). Perbaikan kalimat ini berdasarkan dari kamus singkatan yang telah dibuat.
d. *Tokenizing*, pemecahan kalimat pada *tweet* yang masih utuh menjadi bentuk per kata.
e. *Stopwords Removal*, dilakukan penghapusan kata yang dianggap tidak memiliki makna. Penghapusan ini berdasarkan kamus *stopwords* yang telah dimasukan ke sistem.
f. *Stemming*, dilakukan perubahan sebuah kata menjadi bentuk kata dasar. Perubahan ini menggunakan *library* yang disediakan oleh python yaitu *library* sastrawi yang ditujukan untuk bahasa indonesia.

3.3. Pelabelan Dataset

Pelabelan data adalah tahap pengelompokan teks ke dalam kategori atau label positif dan negatif. Dalam proses ini, pelabelan dilakukan menggunakan leksikon atau kamus bahasa Indonesia. Tujuan dari pelabelan data ini adalah untuk mengetahui sentimen dari *tweet* pengguna LRT Jabodebek, apakah termasuk dalam sentimen positif atau negatif, sebelum nantinya diuji dengan algoritma SVM.

Dari pelabelan data juga akan dilakukan visualisasi data untuk mengetahui gambaran umum dalam sentimen positif dan sentimen negatif. Gambaran umum tersebut berupa kata-kata yang sering tampil dalam *tweet* pengguna LRT Jabodebek. Visualisasi ini akan digambarkan menggunakan wordcloud yang dibuat dengan bahasa pemrograman python di Google Colab.

3.4. Pengembangan dan Evaluasi Model

Langkah pertama yang akan dilakukan pada tahap ini yaitu pembobotan dengan TF-IDF. TF-IDF merupakan metode yang dilakukan dengan cara menghitung bobot pada kata. Setelah selesai, proses selanjutnya yaitu melakukan klasifikasi dengan algoritma *Support Vector Machine* menggunakan operator modelling SVM (libSVM) di RapidMiner. Dalam operator tersebut terdapat beberapa kernel yang dapat dilakukan untuk pelatihan model klasifikasi, namun pada penelitian ini hanya menggunakan kernel linear dan kernel RBF.

Dalam melakukan evaluasi ini akan dibantu dengan *split validation* untuk membagi antara data latih dan data uji dengan rasio 0.7 dan 0.8. Selanjutnya juga akan dibantu dengan operator *optimize parameter*

untuk mendapatkan performa terbaik karena konsep SVM sendiri yaitu mencari *hyperplane* terbaik.

4. HASIL DAN PEMBAHASAN

Dalam penelitian ini, proses *crawling data*, *pre-processing*, dan pelabelan dataset dilakukan menggunakan bahasa Python di Google Colab. Sedangkan untuk pemodelan dan evaluasi metode SVM, digunakan alat RapidMiner.

4.1. *Crawling Data*

Langkah awal dalam pengambilan dataset adalah melakukan *crawling data* dengan menggunakan Tweet Harvest. Kata kunci yang digunakan adalah: "lrt jabodebek lang:id until:2024-05-15 since:2023-08-28 -filter:links -filter:replies" dengan batasan limit 1200.

Tabel 2. Dataset *tweet* LRT Jabodebek

No	*Full_text*
1	Hai min! Boleh tolong infokan rute lrt dari jatimulya menuju kelapa gading? @lrtjabodebek @lrtjkt
2	eskalator stasiun @lrtjabodebek cabang dukuh atas yang nyambung ke stasiun sudirman mati lagi dari kemaren. rajin juga dia mati.
…	…… ……
945	di tt udh banyak fyp lrt jabodebek ihh gasabarbgtt excited!!
946	jam pulang kantor kapan y dah gasabar mauu coba lrt jabodebek yang stasiunnya kepleset dari gang rumah

Sesuai pada Tabel 2, hasil yang didapatkan dari proses *crawling data* tersebut adalah sebanyak 946 *tweet* dari total batasan yang dimasukkan, yaitu 1200 *tweet*.

4.2. *Pre-Processing Data*

Proses *pre-processing data* dilakukan menggunakan bahasa pemrograman Python dengan memanfaatkan berbagai modul yang tersedia.

a. *Cleaning Text*

Tabel 3. Hasil *cleaned text*

full_text	*cleaned_text*
Hai min! Boleh tolong infokan rute lrt dari jatimulya menuju kelapa gading? @lrtjabodebek @lrtjkt	Hai min Boleh tolong infokan rute lrt dari jatimulya menuju kelapa gading

Dari Tabel 3, dijelaskan bahwa pembersihan pada teks tersebut meliputi penghapusan tanda seru (!), tanda tanya (?) dan *mention username* (@lrtjabodebek, @lrtjkt).

b. *Case Folding*

Tabel 4. Hasil *case folding*

cleaned_text	*case_folding*
Hai min Boleh tolong infokan rute lrt dari jatimulya menuju kelapa gading	hai min boleh tolong infokan rute lrt dari jatimulya menuju kelapa gading

Dari Tabel 4, diketahui bahwa perubahan bentuk huruf besar menjadi huruf kecil pada kata "Hai" menjadi "hai, dan "Boleh" menjadi "boleh".

c. *Spell Normalization*

Tabel 5. Hasil *normalized text*

case_folding	*normalized_text*
hai min boleh tolong infokan rute lrt dari jatimulya menuju kelapa gading	hai admin boleh tolong infokan rute lrt dari jatimulya menuju kelapa gading

Dari Tabel 5, terdapat perubahan normalisasi pada kata "min" yang berubah menjadi "admin".

d. *Tokenizing*

Tabel 6. Hasil *tokenized*

normalized_text	*tokenize*
hai admin boleh tolong infokan rute lrt dari jatimulya menuju kelapa gading	['hai', 'admin', 'boleh', 'tolong', 'infokan', 'rute', 'lrt', 'dari', 'jatimulya', 'menuju', 'kelapa', 'gading']

Sesuai pada Tabel 6 terdapat perubahan kalimat menjadi bentuk per kata (token).

e. *Stopwords Removal*

Tabel 7. Hasil *stopwords removal*

tokenize	*stopwords_removal*
['hai', 'admin', 'boleh', 'tolong', 'infokan', 'rute', 'lrt', 'dari', 'jatimulya', 'menuju', 'kelapa', 'gading']	['hai', 'admin', 'tolong', 'infokan', 'rute', 'lrt', 'jatimulya', 'kelapa', 'gading']

Yang mengandung kata *stopwords* pada teks tersebut yaitu "boleh", "dari", dan "menuju". Maka dilakukan penghapusan kata tersebut sesuai pada Tabel 7.

f. *Stemming*

Tabel 8. Hasil *stemming*

stopwords_removal	*stemming*
['hai', 'admin', 'tolong', 'infokan', 'rute', 'lrt', 'jatimulya', 'kelapa', 'gading']	['hai', 'admin', 'tolong', 'info', 'rute', 'lrt', 'jatimulya', 'kelapa', 'gading']

Sesuai pada Tabel 8, kata yang perlu diubah menjadi kata dasar yaitu "infokan" menjadi "info".

4.3. Pelabelan Dataset

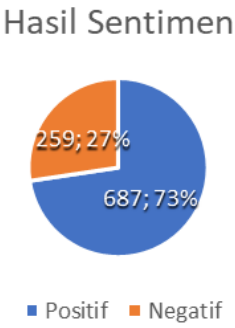
Pelabelan data ini dilakukan dengan bantuan *lexicon* Bahasa Indonesia. Proses ini menggunakan bahasa pemrograman Python dan melibatkan sebuah kamus Bahasa Indonesia yang berisi kata-kata positif dan negatif, masing-masing dengan nilai skor tertentu. Selanjutnya, nilai polaritas kalimat akan dihitung untuk menentukan apakah kalimat tersebut tergolong dalam sentimen positif atau negatif. Penentuan label dilakukan dengan ketentuan bahwa jika nilai

polaritasnya sama dengan atau lebih besar dari 0, maka kalimat tersebut dikategorikan sebagai sentimen positif; sebaliknya, jika nilainya di bawah 0, kalimat tersebut termasuk dalam sentimen negatif. Google Colab digunakan sebagai alat untuk melakukan pelabelan dataset ini.

Tabel 9. Hasil pelabelan dataset

No.	*text*	*polarity*	sentimen
1.	hai admin tolong info rute lrt jatimulya kelapa gading	0	positif
2.	eskalator stasiun cabang dukuh atas nyambung stasiun sudirman mati kemarin rajin mati	1	positif
…	……	…	……
945.	tiktok sudah banyak fyp lrt jabodebek sabar semangat	3	positif
946.	jam pulang kantor ya sudah sabar coba lrt jabodebek stasiun gelincir gang rumah	1	positif

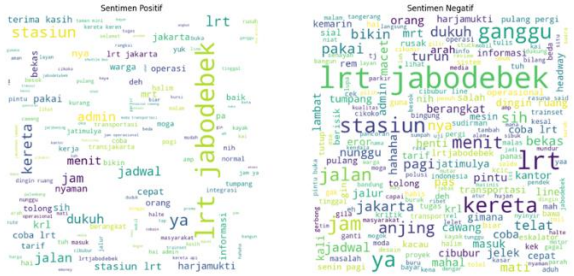
Dari Tabel 9 diketahui bahwa dari sebanyak 946 total dataset keseluruhan terbagi menjadi 687 sentimen positif dan 259 sentimen negatif. Perbandingan sentimen tersebut digambarkan melalui diagram berikut:



Gambar 2. Persentase hasil sentimen

4.4. Visualisasi Data

Visualisasi dilakukan untuk memberikan gambaran tentang kata-kata yang paling sering muncul dalam sentimen positif dan negatif. Proses pembuatan visualisasi ini berdasarkan hasil sentimen yang diperoleh dari pelabelan dataset. Hasil visualisasi akan disajikan dalam bentuk *word cloud*.

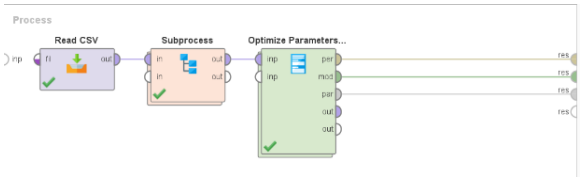


Gambar 3. *Word cloud* sentimen

Berdasarkan Gambar 3, kata yang paling besar adalah kata yang banyak terdapat dalam *tweet* pengguna LRT Jabodebek. Jika dilihat pada *word cloud* sentimen positif, kata yang tampil yang berkaitan dengan pelayanan LRT Jabodebek ialah "cepat", "aman", "baik", "keren", "normal". Sedangkan pada *word cloud* sentimen negatif, kata yang berkaitan dengan pelayanan LRT Jabodebek ialah "rusak", "lambat", "eror", "kendala", "ganggu", dan sebagainya.

4.5. Pengembangan dan Evaluasi Model

Pada tahap ini, pembuatan model dilakukan menggunakan *tools* RapidMiner. Dengan langkah-langkah berikut seperti pada Gambar 4:



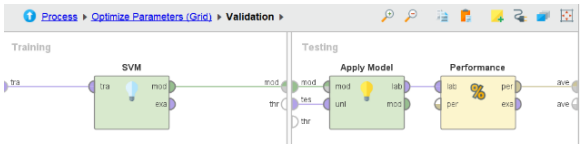
Gambar 4. Proses di RapidMiner

a. Mengunggah dataset ke dalam operator *read* csv.
b. Menghubungkannya dengan operator *subprocess* yang berisi operator-operator untuk melakukan vektor TF-IDF.

Gambar 5. Isi operator *subprocess*

Pada Gambar 5, langkah pertama memasukan operator *set role* untuk menargetkan *role* pada dataset, kemudian menghubungkan dengan operator *nominal to text* dan operator *process document* untuk menghitung TF-IDF.

Menambahkan operator *optimize parameter* untuk membantu menemukan *hyperplane* yang optimal, dalam operator ini akan dimasukan *split validation* dengan rasio 0,8:0,2 dan 0,7:0,3.



Gambar 6. Isi operator *split validation*

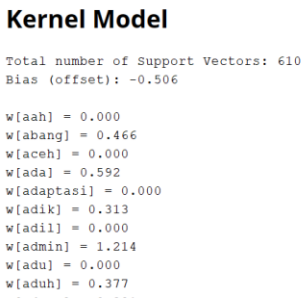
Rasio tersebut untuk membagi antara *data training* untuk pelatihan model SVM dan *data testing* untuk mengevaluasi model tersebut.

Penelitian ini akan melakukan pemodelan SVM dengan menggunakan kernel linear dan kernel RBF. Dalam pengaturan *optimize parameter*, *hyperplane* terbaik akan dicari dengan rentang nilai C antara 0,01 hingga 100 dan rentang nilai *gamma* dari 0,001 hingga

100. Setelah menjalankan proses tersebut, berikut adalah hasil dari pembuatan model klasifikasi SVM menggunakan *tools* RapidMiner.

4.6. Kernel Linear

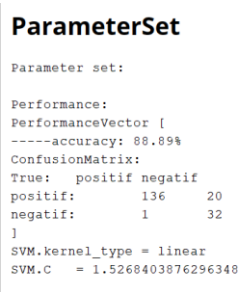
Dalam pengujian dengan rasio 0.8, didapatkan pemodelan SVM sebagai berikut:



Gambar 7. Model kernel linear rasio 0.8

Pada Gambar 7 diketahui bahwa terdapat 610 total support vector dengan nilai bias yaitu -0,506. Kemudian pada Gambar 7 juga diketahui nilai *weight* atau bobot kata. *Weight* sama dengan nol berarti kata tersebut tidak memiliki pengaruh yang signifikan, sedangkan *weight* bernilai berarti memiliki pengaruh pada pemodelan SVM.

Hasil evaluasi model untuk rasio ini tercantum pada gambar berikut:



Gambar 8. Hasil evaluasi model linear rasio 0.8

Pada Gambar 8 diketahui nilai akurasi terbaik dari rasio 0.8 dengan kernel linear bisa didapatkan jika nilai C = 1,527. Nilai akurasi tersebut mencapai 88,89% yang berarti model tersebut sangat akurat.

accuracy: 88.89%

	true positif	true negatif	class precision
pred. positif	136	20	87.18%
pred. negatif	1	32	96.97%
class recall	99.27%	61.54%	

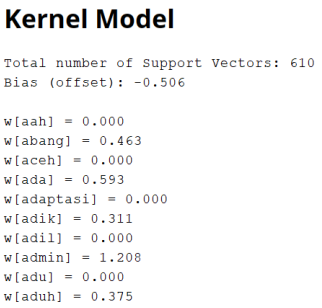
Gambar 9. *Confusion matrix* model linear rasio 0.8

Dari tabel *confusion matrix* pada Gambar 9, diperhitungkan nilai rata-rata *recall* dan *precision* berikut:

$$Recall = \frac{0{,}9927 + 0{,}6154}{2} = 0{,}8063 = 80{,}63\%$$

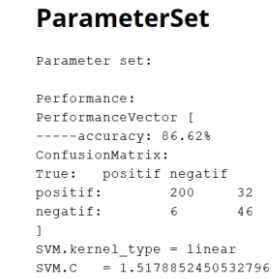
$$Precision = \frac{0{,}8718 + 0{,}9697}{2} = 0{,}8937 = 89{,}37\%$$

Sedangkan pengujian pada rasio 0.7, didapatkan pemodelan SVM sebagai berikut:



Gambar 10. Model kernel linear rasio 0.7

Pada Gambar 10 diketahui bahwa terdapat 610 total *support vector* dengan nilai bias yaitu -0,506, yang mana hasil tersebut sama dengan model kernel linear yang rasio 0.8. Tetapi *weight* untuk setiap kata mempunyai nilai yang berbeda. Hasil evaluasi untuk model ini tercantum pada gambar berikut:



Gambar 11. Hasil evaluasi model linear rasio 0.7

Sesuai pada Gambar 11, untuk mendaptakan akurasi terbaiknya yaitu dengan nilai C = 1,518, hal itu berbeda dengan rasio 0.8. Akurasi yang didapat mencapai 86,62% yang berarti sangat akurat untuk model ini.

accuracy: 86.62%

	true positif	true negatif	class precision
pred. positif	200	32	86.21%
pred. negatif	6	46	88.46%
class recall	97.09%	58.97%	

Gambar 12. *Confusion matrix* model linear rasio 0.7

Dari tabel *confusion matrix* pada Gambar 12, diperhitungkan nilai rata-rata *recall* dan *precision* berikut:

$$Recall = \frac{0{,}9709 + 0{,}5897}{2} = 0{,}7803 = 78{,}03\%$$

$$Precision = \frac{0{,}8621 + 0{,}8846}{2} = 0{,}8733 = 87{,}33\%$$

4.7. Kernel RBF

Untuk kernel RBF, pengujian dilakukan tidak hanya menggunakan nilai C melainkan juga menggunakan nilai *gamma*. Pada pengujian dengan rasio 0.8, didapatkan pemodelan SVM sebagai berikut:

Kernel Model

```
Total number of Support Vectors: 770
Bias (offset): 0.215

w[aah] = 0.000
w[abang] = 0.880
w[aceh] = 0.000
w[ada] = 0.777
w[adaptasi] = 0.000
w[adik] = 0.381
w[adil] = 0.000
w[admin] = 2.140
w[adu] = 0.000
w[aduh] = 0.539
```

Gambar 13. Model kernel RBF rasio 0.8

Pada Gambar 13 diketahui terdapat 770 total *support vector* dengan nilai bias 0,215. Jika pada gambar yang tertera, *weight* 'admin' memiliki pengaruh yang signifikan terhadap model tersebut.

ParameterSet

```
Parameter set:

Performance:
PerformanceVector [
-----accuracy: 87.83%
ConfusionMatrix:
True:   positif negatif
positif:        135     21
negatif:        2       31
]
SVM.kernel_type = rbf
SVM.C   = 39.135707015975136
SVM.gamma       = 0.5874694277083564
```

Gambar 14. Hasil evaluasi model RBF rasio 0.8

Kemudian setelah diuji, untuk mendapatkan akurasi mencapai 87,83% harus dengan nilai C = 39,136 dan nilai *gamma* = 0,587.

accuracy: 87.83%

	true positif	true negatif	class precision
pred. positif	135	21	86.54%
pred. negatif	2	31	93.94%
class recall	98.54%	59.62%	

Gambar 15. Confusion matrix model RBF rasio 0.8

Dari Gambar 15, diperhitungkan nilai-nilai berikut:

$$Recall = \frac{0{,}9854 + 0{,}5962}{2} = 0{,}7908 = 79{,}08\%$$

$$Precision = \frac{0{,}8654 + 0{,}9394}{2} = 0{,}9024 = 90{,}24\%$$

Selanjutnya, untuk pengujian dengan rasio 0.7 didapatkan pemodelan SVM sebagai berikut:

Kernel Model

```
Total number of Support Vectors: 770
Bias (offset): 0.215

w[aah] = 0.000
w[abang] = 0.880
w[aceh] = 0.000
w[ada] = 0.777
w[adaptasi] = 0.000
w[adik] = 0.381
w[adil] = 0.000
w[admin] = 2.140
w[adu] = 0.000
w[aduh] = 0.539
```

Gambar 16. Model kernel RBF rasio 0.7

Sesuai pada Gambar 16, terdapat 770 total *support vector* dengan nilai bias yaitu 0,215. Hasil pemodelan yang terdapat pada rasio 0.7 ini sama dengan hasil pada rasio 0.8, akan tetapi untuk hasil evaluasi berbeda.

ParameterSet

```
Parameter set:

Performance:
PerformanceVector [
-----accuracy: 86.62%
ConfusionMatrix:
True:   positif negatif
positif:        202     34
negatif:        4       44
]
SVM.kernel_type = rbf
SVM.C   = 39.135707015975136
SVM.gamma       = 0.5874694277083564
```

Gambar 17. Hasil evaluasi model RBF rasio 0.7

Pada Gambar 17, diketahui nilai akurasi yang terbaik yaitu mencapai 86,62% dengan ketentuan nilai C = 39,135 dan nilai *gamma* = 0,587.

accuracy: 86.62%

	true positif	true negatif	class precision
pred. positif	202	34	85.59%
pred. negatif	4	44	91.67%
class recall	98.06%	56.41%	

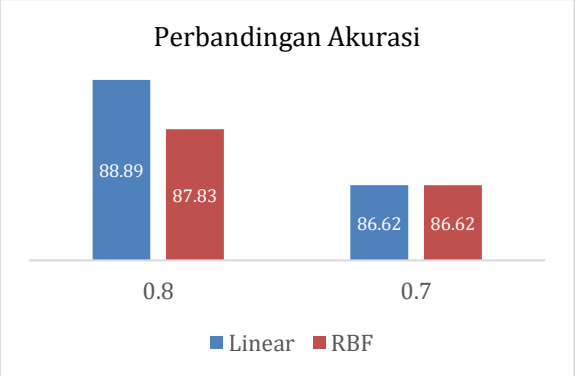
Gambar 18. *Confusion matrix* model RBF rasio 0.7

Dari tabel confusion matrix pada Gambar 18, dihitung nilai-nilai berikut ini:

$$Recall = \frac{0{,}9806 + 0{,}5641}{2} = 0{,}7723 = 77{,}23\%$$

$$Precision = \frac{0{,}8559 + 0{,}9167}{2} = 0{,}8863 = 88{,}63\%$$

Berdasarkan hasil pengujian dua kernel dan dua rasio, berikut perbandingan akurasi yang dapat digambarkan dalam bentuk diagram batang.



Gambar 19. Perbandingan hasil akurasi

Dari diagram batang tersebut bahwa pada rasio 0.8, model kernel linear lebih baik dari model kernel RBF, namun pada rasio 0.7 model kernel linear memiliki akurasi yang setara dengan model kernel RBF.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan yang telah disampaikan sebelumnya, diketahui dataset yang berhasil dikumpulkan dari *tweet* pengguna LRT Jabodebek terdiri dari 946 *tweet*, yang setelah pelabelan data terbagi menjadi 687 *tweet* dengan sentimen positif dan 259 *tweet* dengan sentimen negatif. Setelah dilakukan pengujian menggunakan metode SVM di RapidMiner yang didukung oleh operator *optimize parameter*, pada rasio 0.8 dengan kernel linear menghasilkan akurasi 88,89%, rata-rata *recall* 80,63%, dan rata-rata presisi 89,67%, sedangkan pada rasio 0.7 menghasilkan akurasi 86,62%, rata-rata recall 78,03% dan rata-rata presisi 87,33%. Sementara itu, model dengan kernel RBF pada rasio 0.8 menghasilkan akurasi 87,83%, rata-rata *recall* 79,08%, dan rata-rata presisi 90,24%, sedangkan pada rasio 0.7 menghasilkan akurasi 86,62%, rata-rata recall 77,23% dan rata-rata presisi 88,63%. Kata-kata yang terdapat di WordCloud terkait dengan pelayanan LRT Jabodebek untuk sentimen positif antara lain "cepat", "aman", "baik", "keren", dan "normal", sedangkan untuk sentimen negatif, kata-kata yang muncul adalah "rusak", "lambat", "eror", "kendala", dan "ganggu".

Dari penelitian ini, penulis mengidentifikasi beberapa kekurangan yang dapat dijadikan sebagai saran, yaitu adanya kelemahan dalam *pre-processing data* pada tahap normalisasi, penghapusan *stopwords*, dan *stemming* karena pada tahap-tahap tersebut, penghapusan dan penggantian kata hanya dilakukan sesuai dengan kata-kata yang terdapat dalam kamus yang diunggah ke sistem. Selain itu, tidak semua kata positif dan negatif tercantum dalam leksikon bahasa Indonesia yang ada di sistem.

DAFTAR PUSTAKA

[1] D. Agustina and F. Rahmah, "Analisis Sentimen pada Sosial Media Twitter terhadap MRT Jakarta Menggunakan Machine Learning," *Insearch (Information Syst. Res. J.*, vol. 2, pp. 1–6, 2022, [Online]. Available: https://ejournal.uinib.ac.id/jurnal/index.php/insearch/article/view/3548/2416

[2] C. M. Annur, "10 Negara dengan Jumlah Pengguna Twitter Terbanyak di Dunia (Juli 2023)," *Katadata Media Networks*, 2023. https://databoks.katadata.co.id/datapublish/2023/11/01/jumlah-pengguna-twitter-indonesia-duduki-peringkat-ke-4-dunia-per-juli-2023 (accessed May 14, 2024).

[3] Z. Alhaq, A. Mustopa, S. Mulyatun, and J. D. Santoso, "Penerapan Metode Support Vector Machine Untuk Analisis Sentimen Pengguna Twitter," *J. Inf. Syst. Manag.*, vol. 3, no. 2, pp. 44–49, 2021, [Online]. Available: https://jurnal.amikom.ac.id/index.php/joism/article/view/558/229

[4] E. Dwianto and M. Sadikin, "Analisis Sentimen Transportasi Online pada Twitter Menggunakan Metode Klasifikasi Naïve Bayes dan Support Vector Machine," *Format J. Ilm. Tek. Inform.*, vol. 10, no. 1, p. 94, 2021, [Online]. Available: https://publikasi.mercubuana.ac.id/index.php/format/article/view/11012/pdf

[5] B. P. Zen, D. Wicaksana, and H. Alfidzar, "Analisis Sentimen Tweet Vaksin Covid 19 Sinovac Menggunakan Metode Support Vecor Machine," *J. Data Min. dan Sist. Inf.*, vol. 3, no. 2, p. 21, 2022, [Online]. Available: https://ejurnal.teknokrat.ac.id/index.php/JDMSI/article/view/1926/961

[6] F. Sulianta, *Basic Data Mining from A to Z*. Feri Sulianta, 2023.

[7] N. Iriadi, Priatno, and A. Ishaq, *Penerapan Data Mining dengan Rapid Miner : Konsep Data Mining, Data Warehouse, Metode, Model, Teknik*. Yogyakarta: Graha Ilmu, 2020.

[8] B. A. Maulana, R. A. Fauzi, R. I. Agustin, S. A. Azhaar, and T. Rohana, "Komparasi Algoritma Naïve Bayes Dan SVM Untuk Analisis Sentimen Twitter Korupsi Bansos Beras Masa Pandemi," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, pp. 912–918, 2024, doi: 10.23960/jitet.v12i2.4020.

[9] D. Jollyta, Prihandoko, A. Hajjah, E. Haerani, and M. Siddik, *Algoritma Klasifikasi untuk Pemula Solusi Python dan RapidMiner*. Yogyakarta: DEEPUBLISH DIGITAL, 2023.

[10] Y. Severianus and S. Kolo, "Analisis Sentimen Terhadap Opini Masyarakat Terkait Perubahan Cuaca Di Indonesia Menggunakan Algoritma Support Vector Machine," *J. Mhs. Tek. Inform.*, vol. 8, no. 2, pp. 1412–1416, 2024, [Online]. Available: https://ejournal.itn.ac.id/index.php/jati/article/view/8988/5224

[11] M. Fadli and R. A. Saputra, "Klasifikasi Dan Evaluasi Performa Model Random Forest Untuk Prediksi Stroke," *JT J. Tek.*, vol. 12, no. 2, pp. 72–80, 2023, [Online]. Available: https://jurnal.umt.ac.id/index.php/jt/article/viewFile/9099/4575

[12] H. Satria, "Cara mendapatkan data twitter yang besar (lebih dari 1500)," 2023. https://helmisatria.com/blog/cara-mendapatkan-data-twitter-yang-besar/ (accessed May 14, 2024).