JURNAL INFORMATIKA, Vol.6 No. 2 September 2019, pp. 212~218
ISSN: 2355-6579
E-ISSN**:** 2528-2247

Perbandingan Akurasi dan Waktu Proses Algoritma K-NN dan SVM dalam Analisis Sentimen Twitter

Muhammad Rangga Aziz Nasution, Mardhiya Hayaty[2]

[1]Universitas Amikom Yogyakarta
e-mail: muhammad.9307@students.amikom.ac.id

[2] Universitas Amikom Yogyakarta
e-mail: mardhiya_hayati@amikom.ac.id

Abstrak

Salah satu cabang ilmu komputer yaitu pembelajaran mesin (machine learning) menjadi tren dalam beberapa waktu terakhir. Pembelajaran mesin bekerja dengan memanfaatkan data dan algoritma untuk membuat model dengan pola dari kumpulan data tersebut. Selain itu, pembelajaran mesin juga mempelajari bagaimama model yang telah dibuat dapat memprediksi keluaran (output) berdasarkan pola yang ada. Terdapat dua jenis metode pembelajaran mesin yang dapat digunakan untuk analisis sentimen: supervised learning dan unsupervised learning. Penelitian ini akan membandingkan dua algoritma klasifikasi yang termasuk dari supervised learning: algoritma K-Nearest Neighbor dan Support Vector Machine, dengan cara membuat model dari masing-masing algoritma dengan objek teks sentimen. Perbandingan dilakukan untuk mengetahui algoritma mana lebih baik dalam segi akurasi dan waktu proses. Hasil pada perhitungan akurasi menunjukkan bahwa metode Support Vector Machine lebih unggul dengan nilai 89,70% tanpa K-Fold Cross Validation dan 88,76% dengan K-Fold Cross Validation. Sedangkan pada perhitungan waktu proses metode K-Nearest Neighbor lebih unggul dengan waktu proses 0.0160s tanpa K-Fold Cross Validation dan 0.1505s dengan K-Fold Cross Validation.

Kata Kunci: machine learning, klasifikasi, support vector machine, k-nearest neighbor, validasi

Abstract

One branch of computer science, machine learning, become a trend in recent times. Machine learning works by utilizing data and algorithms to make a model based on pattern from the data. Furthermore, machine learning methods learn how to predict outputs of data based on the existing pattern. There are two types of machine learning methods that can be used in sentiment analysis: supervised learning and unsupervised learning. This research will compare between two algortihms that including part of the supervised learning method: K-Nearest Neighbor algorithm and Support Vector Machine algorithm by making model each algorithm with sentimental text object. Comparison is done to find out which algorithm is better in terms of accuracy and processing time. The result of accuracy calculation shows that Support Vector Machine algorithm better with 89,70% percentage without validation and 88,76% percentage with validation. Whereas the result of processing time calculation shows that K-Nearest Neighbor algorithm better with 0.0160 second without validation and 0.1505s with validation.

Keywords*: machine learning, classification, support vector machine, k-nearest neighbor, validation*

1. Pendahuluan

Sosial media menjadi fenomena yang tidak terbendung keberadaannya. Sosial media telah mengubah kehidupan manusia dan cara berinteraksi manusia. Sosial media biasanya digunakan oleh seseorang sebagai media komunikasi, sarana informasi dan sebagai media hiburan bagi penggunanya(He, Zha, & Li, 2013).

Twitter adalah salah satu media sosial yang bertipe microblogging sebagai layanan interaksinya. Twitter menjadi salah satu layanan media sosial yang paling terkenal di dunia dengan lebih dari 200 juta pengguna aktif dan lebih dari 10.6 milyar tweet yang telah dihasilkan(M. Ibrahim, Abdillah, Wicaksono, & Adriani, 2016).

Meningkatnya penggunaan social network dan situs *micro-blogging*, dapat dimanfaatkan dengan membuat analisis sentimen. Sentiment analysis menjadi penting karena dengan melakukannya sebuah perusahaan produk atau jasa bisa mendapatkan *feedback* dengan mudah hanya dengan menganalisis opini-opini yang masuk ke perusahaan mereka, atau customer bisa menilai suatu produk atau jasa dengan melihat hasil analisis sentimen terhadap produk atau jasa tersebut(Patodkar & I.R, 2016).Terdapat beberapa metode yang dapat digunakan untuk membuat suatu percobaan analisis sentimen. Terdapat dua jenis metode yang ada pada lingkup machine learning yang dapat digunakan untuk analisis sentimen, yaitu *supervised learning* dan *unsupervised learning*. Pada *supervised learning*, beberapa data akan dilatih dan akan dijadikan model untuk data baru yang akan diuji. Sedangkan pada *unsupervised learning* teks dikategorikan sesuai dengan tujuan dari setiap teks tersebut(Feldman, 2013).

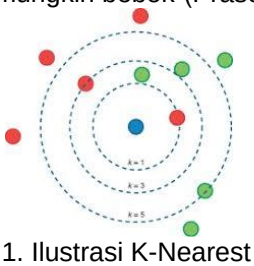
K-Nearest Neighbor merupakan salah satu algoritma yang mudah diimplementasikan dengan tingkat keefektifan yang tinggi, serta cocok untuk berbagai masalah yang berhubungan dengan klasifikasi. Sedangkan Support Vector Machine merupakan salah satu algoritma yang tepat dipakai untuk klasifikasi *text classification*. Kemampuan Support Vector Machine menemukan hyperplane terbaik menjadikan algoritma ini memiliki tingkat generalitas yang tinggi serta menjadikannya algoritma dengan akurasi terbaik dibandingkan algoritma lainnya(Gayathri & Marimuthu, 2013).

Penelitian perbandingan algoritma klasifikasi pernah dilakukan untuk menganalisis sentimen mengenai berita kabut asap dan kebakaran hutan. Pembuatan model klasifikasi diuji coba menggunakan metode optimasi PSO. Hasil yang didapatkan adalah model SVM mendapatkan akurasi 80.83% dan SVM+PSO sebesar 86.11%. Model K-NN mendapatkan akurasi 85.00% dan K-NN+PSO sebesar 73.06%(Utami, 2017). Selain itu, perbandingan algoritma klasifikasi juga dilakukan pada analisis sentimen terhadap perbankan, dengan SVM mendapatkan akurasi 63% dan 69.55% pada pengujian 100 dan 266 data, sedangkan K-NN mendapatkan 57% dan 49.62% di pengujian 100 dan 266 data(Probo et al., 2016).

Pada penelitian ini dilakukan perbandingan antara dua algoritma klasifikasi K-Nearest Neighbor dan Support Vector Machine dari segi akurasi dan kecepatan proses dalam analisis sentimen terhadap presiden Amerika Serikat Donald Trump. Perbandingan ini bertujuan untuk mengetahui algoritma mana yang memiliki akurasi terbaik dan waktu proses tercepat.

1.1. K-Nearest Neighbor

Metode K-Nearest Neighbor (K-NN) merupakan salah satu dalam *top 10* metode data mining yang paling banyak digunakan. Metode ini melakukan klasifikasi berdasarkan kemiripan suatu data dengan data yang lain. Prinsip sederhana metode ini adalah “Jika suatu hewan berjalan seperti bebek, bersuara kwek-kwek seperti bebek, dan penampilannya seperti bebek, maka hewan itu mungkin bebek”(Prasetyo, 2014).



Gambar 1. Ilustrasi K-Nearest Neighbor

Ilustrasi di atas menjelaskan metode klasifikasi. Pada gambar ini, titik biru akan dijadikan objek diprediksi kelasnya. Untuk k = 1 maka kemungkinan objek masuk ke kelas merah. Selanjutnya k = 3, objek diprediksi masuk kelas merah, dengan perhitungan 2-1 lebih banyak atas kelas hijau. Untuk k = 5, maka akan diprediksi masuk kelas hijau dengan perhitungan 3-2

lebih banyak dari kelas merah(Mitchell B.O., 2014). K-Nearest Neighbor digunakan untuk mengklasifikasi data yang tidak dilabeli. Karakteristik data didapatkan dari *training set* dan *test set*(Zhang, 2016).

Langkah-langkah klasifikasi data menggunakan K-Nearest Neighbor adalah sebagai berikut(Ibrahim, Bacheramsyah, & Hidayat, 2018):

1. Tentukan nilai K
2. Hitung jarak antara data baru ke setiap label data
3. Tentukan *k* labeled data yang mempunyai jarak yang paling minimal
4. Klasifikasikan data baru ke dalam label data yang mayoritas K-NN dipilih berdasarkan metrik jarak.

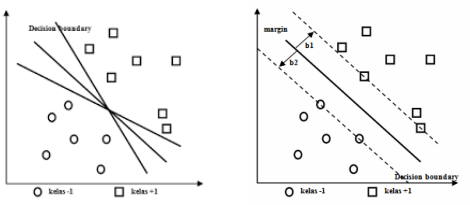
Pada penelitian ini, data teks yang telah didapatkan akan diubah menjadi vektor. Dalam perhitungan jarak antar vektor, metode yang sering digunakan adalah cosine similarity. Metode ini mudah untuk dijelaskan pada data yang bersifat sparse seperti text data. Persamaan dari cosine similarity adalah:

$$S_{cosine}(x,y) = \frac{x^T y}{\|x\|\|y\|} \qquad (1)$$

Di mana $\|x\| = \sqrt{\sum_{i=1}^{l} x_i^2}$ dan $\|y\| = \sqrt{\sum_{i=1}^{l} y_i^2}$ adalah panjang dari vektor(Al-Anzi & AbuZeina, 2017).

1.2. Support Vector Machine

Support Vector Machine (SVM) adalah salah satu metode klasifikasi yang banyak dikembangkan saat ini. Konsep dasar metode ini adalah memaksimalkan batas hyperplane yang memisahkan suatu set data(Prasetyo, 2014).



Gambar 1. Ilustrasi Support Vector Machine

Gambar di atas menjelaskan konsep klasifikasi SVM. Pada gambar (a) terdapat sejumlah data dengan lingkaran sebagai kelas -1 dan kotak sebagai kelas +1. Pada gambar tersebut juga terdapat sejumlah hyperplane yang mungkin untuk set data. Gambar (b) adalah hyperplane yang paling maksimal. Perhitungan hyperplane dilakukan dengan cara menghitung jarak margin dengan data terdekat dari masing-masing kelas. Data terdekat ini disebut sebagai Support Vector Machine. Inti dari metode ini adalah pencarian hyperplane terbaik dari setiap kemungkinan(Neneng, Adi, & Isnanto, 2016).

Persamaan *hyperplane* yang terletak pada support vector adalah:

$$\begin{aligned} \vec{w}.\vec{x} + b &= -1 \\ \vec{w}.\vec{x} + b &= +1 \end{aligned} \qquad (2)$$

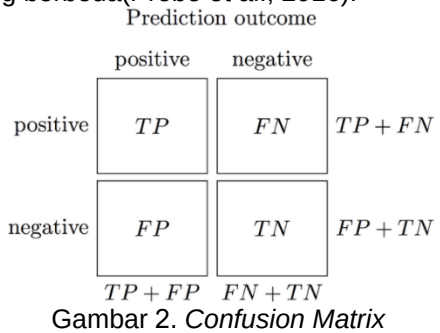
Persamaan untuk menghitung maximum margin antara hyperplane yang optimal dengan hyperplane yang berada pada support vector adalah(Varma, Rao, Raju, & Varma, 2016):

$$Margin = \frac{2}{\|\vec{w}\|} \qquad (3)$$

1.3. Evaluasi

Evaluasi berfungsi untuk mengetahui akurasi dari model algoritma yang dibuat(Utami, 2017). Kriteria evaluasi yang dipertimbangkan adalah akurasi, *standard deviation*, f1 score, *recall*, *precision* dan *specificity*(Attal et al., 2015). Pada penelitian ini, evaluasi yang dilakukan adalah dengan menghitung akurasi dan f1 score. Untuk mendapatkan *precision* dan *recall*, maka penelitian ini menggunakan *confusion matrix*.

Confusion matrix adalah alat yang berfungsi untuk menganalisis seberapa model klasifikasi mengenali *tuple* dari data yang berbeda(Probo et al., 2016).



Gambar 2. *Confusion Matrix*

Akurasi adalah proporsi dari total prediksi true dari semua data. Rumus akurasi adalah:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (4)$$

Precision adalah ukuran ketepatan dari hasil suatu model. Persamaannya adalah perbandingan antara *true positive* dengan total data dengan label *positive*:

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

Recall adalah ukuran kelengkapan dari sebuah model. Persamaan recall perbandingan antara true positive terhadap total contoh yang benar-benar positive:

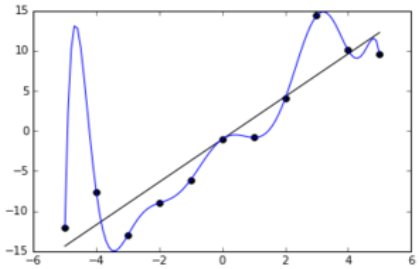
$$Recall = \frac{TP}{TP + FN} \quad (6)$$

F-measure merupakan *harmonic mean* dari precision dan recall, di mana persamaannya adalah(Tripathy, Agrawal, & Rath, 2016):

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (7)$$

1.4. Validasi

Model fit adalah aspek penting pada permasalahan klasifikasi dan regresi. Aspek ini menunjukkan seberapa jauh suatu model dalam memprediksi data baru yang tidak dilatih sebelumnya. Jika terjadi penyimpangan yang jauh dalam memprediksi data maka pada model tersebut terjadi *overfitting*(Lever, Krzywinski, & Altman, 2016).



Gambar 3. Ilustrasi *overfitting*

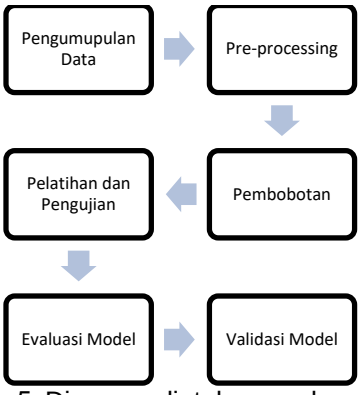
K-Fold Cross Validation adalah salah satu metode yang dapat memeriksa *overfitting* pada suatu model. Data yang dibagi menjadi *k* bagian membolehkan setiap bagian data berhenti memprediksi data lebih cepat ketimbang tidak dibagi terlebih dahulu(Barrow & Crone, 2013).

Gambar 4. Ilustrasi k-Fold Cross Validation

Pada k-Fold Cross Validation, model yang telah dibuat dibagi menjadi k bagian yang sama atau mendekati ukurannya. Akurasi model akan diuji menggunakan data uji pada setiap fold, dan berlanjut ke fold selanjutnya sampai selesai. Akurasi akan ditotal dan dibagi dengan banyaknya *k*(Yadav & Shukla, 2016)

2. Metode Penelitian

Berikut ini adalah tahapan-tahapan eksperimen yang dilakukan pada penelitian ini.



Gambar 5. Diagram alir tahapan eksperimen

2.1. Pengumpulan Data

Tahapan awal adalah tahapn pengumpulan data tweet. Data yang dikumpulkan menggunakan Twitter API yang telah disediakan oleh Twitter. Kata kunci yang dicari adalah tweet yang mengandung “Donald Trump” atau yang berkaitan tentang Donald Trump.

Data yang dikumpulkan sebanyak 1113 data. Masing-masing data dipisahkan menjadi data uji dan data latih. Data uji diberi label sedangkan data latih tidak diberi label namun akan diberikan kolom expected untuk perhitungan akurasi dan F1 Score.

2.2. Pre-processing Data

Sebelum masuk tahap pembuatan model dan analisis sentimen, tweet harus terlebih dahulu melalui tahapan pre-processing terlebih dahulu. Tahapan ini bertujuan untuk membersihkan *tweet* dari noise yang mengganggu agar lebih mudah dihitung bobotnya dan dianalisis pada tahapan berikutnya, sehingga diharapkan hasil akurasi dari klasifikasi menjadi lebih akurat.

Langkah-langkah yang dilakukan pada tahapan ini adalah pengubahan semua jenis huruf ke dalam huruf kecil (*case* folding), pembersihan dokumen *tweet* (*cleaning*), pengubahan kata-kata menjadi kata dasarnya (*lemmatizing*), penghapusan *stopwords* (*stopwords removing*), pemisahan string input setiap kata (*tokenization*).

2.3. Pembobotan TF-IDF

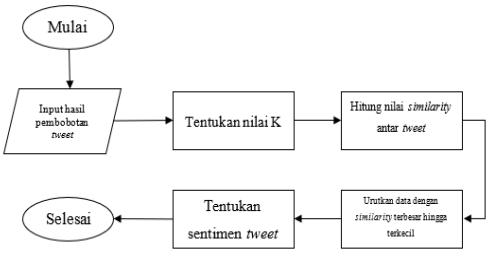
Tahapan pembobotan bertujuan untuk mendapatkan nilai dari kata dasar yang berhasil diekstrak, kemudian kata-kata dasar tersebut dikonversi menjadi sebuah vektor yang mewakili kata yang bersangkutan. Pada penelitian ini menggunakan metode pembobotan Term Frequency (TF) dan Inverse Document Frequency (IDF).

2.4. Klasifikasi

Tahapan klasifikasi merupakan tahapan yang utama dalam proses analisis sentimen. Pada bagian ini terdapat dua proses klasifikasi. Proses pertama adalah klasifikasi pada training data yang telah diberi bobot menggunakan TF-IDF. Proses training akan menghasilkan model classifier yang kemudian akan disimpan dalam file berbentuk pickle. Pickle adalah feature vector yaitu hasil perpaduan antara setiap feture words dan label sentimen.

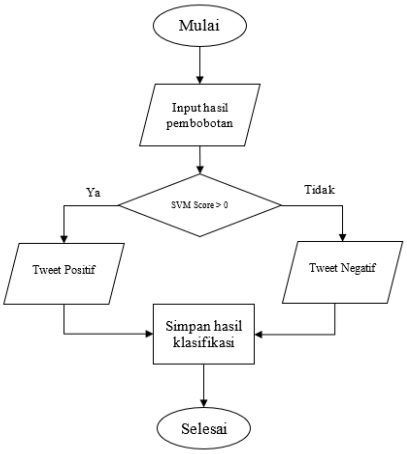
Selanjutnya adalah proses pada test data, di mana data ini akan melalui proses pre-processing dan pembobotan, kemudian akan dianalisis menggunakan model classifier yang telah dibuat. Hasil akhir dari proses ini adalah berupa sentimen pada dokumen tweet apakah positif atau negatif.

Pada metode klasifikasi K-NN, langkah pertama yang harus dilakukan adalah menentukan nilai K untuk menentukan banyaknya tetangga (neighbor). Setelah itu menentukan nilai kedekatan (similarity) antar data menggunakan metode cosine similiarity pada persamaan (2.4). Kemdian data diurutkan dari yang memiliki nilai kemiripan terbesar hingga terkecil. Langkah terakhir adalah melakukan voting pada setiap tetangga untuk menentukan sentimen yang paling mendekati.



Gambar 6. Diagram alir tahapan K-NN

Dalam kasus analisis sentimen, hyperplane pada SVM berfungsi sebagai batas pemisah antara sentimen positif dan negatif. Pada kasus ini jika score > 0 maka tweet akan diklasifikasikan sebagai tweet positif, sedangkan jika score < 0 maka tweet akan diklasifikasikan sebagai tweet negatif. Pada penelitian ini akan digunakan kernel SVM linear.



Gambar 7. Diagram alir tahapan SVM

2.5. Evaluasi

Proses Evaluasi model dilakukan untuk mengukur kinerja dari model klasifikasi yang telah dibuat dan mengetahui akurasinya. Metode evaluasi pada penelitian ini menggunakan confusion matrix. Seluruh test data yang sudah dikumpulkan kemudian dibagi menjadi empat kategori, yaitu True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Akurasi didapatkan dengan mengkalkulasi semua nilai True dibagi dengan jumlah keseluruhan data.

Pada penelitian ini juga akan dihitung *precision*, *recall* dan f1 score dari model klasifikasi tersebut.

2.6. Validasi

Validasi model dilakukan untuk memeriksa *overfitting* pada model klasifikasi. Metode validasi yang dipakai pada penelitian ini adalah K-Fold Cross Validation. Pada proses ini seluruh dataset akan diacak (*shuffle*) lalu dibagi menjadi k bagian sama besar, di mana 1 bagian akan menjadi data uji, sedangkan sisanya akan menjadi data latih.

Pada 10-Fold Cross Validation, 1 bagian data akan menjadi data latih dan 9 bagian data lainnya akan menjadi data uji. Contohnya pada dataset penelitian ini terdapat 1100 data akan dibagi ke dalam 10-fold. Fold ke-1 akan menjadi data uji sedangkan sisanya menjadi data latih. Setelah training tahap pertama selesai selanjutnya training dilakukan dengan menjadikan fold ke-2 sebagai data uji dan 9 data bagian lainnya menjadi data latih. Seperti itu seterusnya sampai proses selesai.

3. Hasil dan Pembahasan

3.1. Pengujian

Dokumen yang telah dikumpulkan sebanyak 1113 data seluruhnya akan diproses melalui tahapan *pre-processing*. Selanjutnya data akan dibagi menjadi data latih dan data uji dengan rasio 80:20. Data latih akan dibuat model *classifier* menggunakan algoritma yang telah ditentukan, sedangkan data uji akan diprediksi menggunakan model yang telah dibangun. Model akan dihitung akurasi dan waktu prosesnya berdasarkan hasil prediksi.

Penelitian ini menggunakan perangkat keras dan perangkat lunak dengan spesifikasi berikut.

Tabel 1. Spesifikasi perangkat keras dan perangkat lunak

Perangkat Keras	*Spesifikasi*
Sistem Operasi	Windows 10 Pro
Tipe Sistem	64-bit
Prosesor	Intel® Core® i5-7200U @2.50 GHz (4 CPUs)
Memori (RAM)	8 GB
Harddisk	1 TB
Perangkat Lunak	• JetBrains Pycharm Community Edition 2018.2.3 • Anaconda • Anaconda Library meliputi: NLTK, Scikit-learn, Tweepy, SQL Alchemy, DataFreeze, Pickle • Microsoft Excel 2016

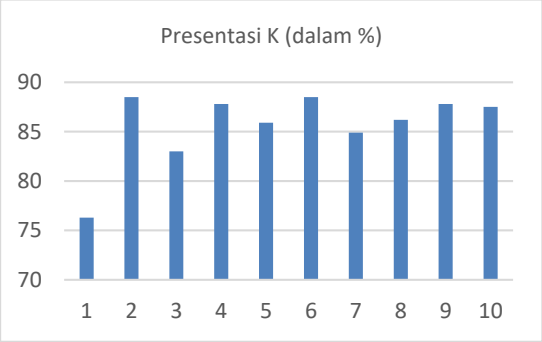
3.2. Pembahasan Hasil

3.2.1. Penentuan K terbaik pada K-NN

Evaluasi dilakukan untuk menentukan nilai K terbaik pada K-NN. Terdapat 10 nilai K yang telah ditentukan, yaitu K=1 hinggga K=10. Nilai K terbaik adalah K yang memiliki akurasi tertinggi.

Berdasarkan hasil evaluasi, nilai K tertinggi didapatkan pada dua nilai K, yaitu nilai K=2 dan K=6, dengan presentase sebesar 88,5%. dengan hasil ini, maka model classifier yang akan dibandingkan selanjutnya adalah model SVM dan model K-NN dengan K=2 dan K=6.

Berikut diagram batang perbandingan presentase K dengan K=1 sampai K=10.



Gambar 8. Grafik batang presentase akurasi setiap K pada K-NN

3.2.2. Perbandingan akurasi dan waktu proses

Hasil evaluasi menunjukkan bahwa metode SVM memiliki tingkat akurasi yang lebih tinggi daripada metode K-NN, dengan nilai akurasi SVM sebesar 89,7%. Sementara akurasi terendah terdapat pada metode K-NN dengan K=1, dengan presentase sebesar 76,3%.

Pada perhitungan waktu proses, model classifier yang memiliki waktu tercepat adalah model K-NN dengan K=8,

memiliki waktu proses 0.0140s. Sedangkan model classifier dengan waktu proses terlambat adalah K-NN dengan K=3, memiliki waktu proses 0.0760s. Model classifier SVM memiliki waktu proses 0.0230s.

Berikut tabel perbandingan akurasi dan waktu proses setiap model classifier.

Tabel 2. Perbandingan akurasi dan waktu proses

Model	**Perhitungan**	
	Akurasi dalam (%)	**Waktu proses (dalam *second*)**
K =1	76.3	0.0160
2	88.5	0.0150
3	83	0.0760
4	87.8	0.0648
5	85.9	0.0419
6	88.5	0.0170
7	84.9	0.0170
8	86.2	0.0140
9	87.8	0.0170
10	87.5	0.0588
SVM	89.7	0.0230

Sedangkan perhitungan recall, precision dan F1 Score ditunjukkan pada tabel berikut.

Tabel 3. Hasil perhitungan *recall*, *precision* dan f1 score

Model	**Recall (dalam %)**	**Precision (dalam %)**	**F1 Score (dalam %)**
K =1	76.3	82.6	78.5
2	88.5	87.4	87.4
3	83	84.5	83.7
4	87.8	86.8	87
5	85.9	86.1	86
6	88.5	87.4	87.4
7	84.9	84.8	84.9
8	86.2	84.9	85.2
9	87.8	87.3	87.5
10	87.5	86.2	86.3
SVM	89.7	89	89.1

3.2.3. Nilai akurasi K-NN dengan k-Fold Cross Validation

Validasi pada model classifier K-NN menggunakan nilai K pada K-NN dengan akurasi terbaik, yaitu K=2 dan K=6, sedangkan k pada metode k-Cross Fold Validation menggunakan nilai k=10.

Hasil validasi menunjukkan bahwa akurasi optimal K-NN dengan K=2 memiliki akurasi tertinggi di iterasi ke-2 yaitu 96.87% dengan rata-rata akurasi sebesar 86.18%. Sedangkan K=6 memiliki akurasi tertinggi pada iterasi ke-2 dengan nilai akurasi tertinggi sebesar 96,87% dengan rata-rata akurasi sebesar 88,1%.

Hasil akurasi metode klasifikasi K-NN dengan K=2 divalidasi menggunakan metode 10-Fold Cross Validation adalah sebagai berikut.

Tabel 4. Hasil perhitungan akurasi K-NN (K=2) dengan K-Fold Cross Validation

Iterasi-ke	**Nilai akurasi (dalam %)**
1	84.37
2	96.87
3	90.32
4	80.64
5	83.87
6	80.64
7	90.32
8	80.64
9	90.32
10	89.87
Rata-rata	86.18

Sedangkan hasil akurasi model classifier K-NN dengan K= 6 divalidasi menggunakan K-Fold Cross Validation adalah sebagai berikut.

Tabel 5. Hasil perhitungan K-NN (K=6) dengan K-Fold Cross Validation

Iterasi-ke	**Nilai akurasi (dalam %)**
1	90.62
2	96.87
3	87.09
---	----
7	87.09
8	83.87
9	90.32
10	90.32
Rata-rata	88.10

3.2.4. Nilai akurasi SVM dengan K-Fold Cross Validation

Validasi pada model classifier SVM menggunakan satu model yang telah dibangun, sedangkan k pada metode k-Cross Fold Validation menggunakan nilai k=10.

Hasil validasi menunjukkan bahwa akurasi optimal berdasarkan metode k-Fold Cross Validation turun sedikit dari akruasi awal, di mana model classifier SVM memiliki akurasi tertinggi pada iterasi ke-2 dengan nilai akurasi tertinggi sebesar 96,87% dengan rata-rata akurasi sebesar 88,76%.

Nilai akurasi metode klasifikasi K-NN menggunakan metode 10-Fold Cross Validation adalah sebagai berikut.

Tabel 6. Hasil perhitungan akurasi SVM dengan K-Fold Cross Validation

Iterasi-ke	**Nilai akurasi (dalam %)**
1	87.5
2	96.87
3	80.64
4	87.09
5	90.32
6	80.64
7	90.32
8	90.32
9	96.77
10	87.09
Rata-rata	88.76

3.2.4. Perbandingan rata-rata akurasi dan waktu proses

Hasil perhitungan setelah validasi menunjukkan bahwa metode SVM memiliki tingkat akurasi yang lebih tinggi daripada metode K-NN, dengan nilai akurasi SVM sebesar 88,76%. Sementara K-NN setelah validasi dengan K=6 sebagai akurasi tertinggi memiliki nilai sebesar 88,1%.

Pada perhitungan waktu proses SVM memiliki waktu proses lebih lambat setelah divalidasi. Model SVM hanya memiliki 0.4532s pada waktu prosesnya. Sedangkan K-NN lebih cepat dengan waktu proses 0.1505s.

Tabel perbandingan akurasi SVM dan K-NN setelah validasi ditampilkan pada Tabel berikut.

Tabel 7. Perbandingan akurasi dan waktu proses dengan K-Fold Cross Validation

Model	**Akurasi (dalam %)**	**Waktu Proses (dalam *second*)**
SVM	88.76	0.4532
K-NN (K=2)	86.10	0.1505
K-NN (K=6)	88.10	0.1964

4. Kesimpulan

Berdasarkan hasil dan pembahasan dari eksperimen yang telah dilakukan, kesimpulan yang bisa ditarik dari penelitian ini adalah:

1. Metode klasifikasi Support Vector Machine memiliki akurasi yang lebih tinggi dibandingkan metode klasifikasi K-Nearest Neighbor, yaitu sebesar 89,70% tanpa validasi K-Fold Cross Validation dan sebesar 88,76% dengan validasi K-Fold Cross Validation.
2. Metode K-Nearest Neighbor memiliki waktu proses yang lebih cepat daripada metode Support Vector Machine, yaitu sebesar 0.0160s tanpa validasi K-Fold Cross Validation dan sebesar 0.1505s dengan validasi K-Fold Cross Validation.
3. Pada penelitian ini, bisa disimpulkan bahwa metode K-Nearest Neighbor memiliki performa yang lebih baik dibandingkan Support Vector Machine.

Referensi

Al-Anzi, F. S., & AbuZeina, D. (2017). Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing. *Journal of King Saud University - Computer and Information Sciences*, *29*(2), 189–195. https://doi.org/10.1016/j.jksuci.2016.04.001

Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., & Amirat, Y. (2015). Physical human activity recognition using wearable sensors. *Sensors (Switzerland)*, *15*(12), 31314–31338. https://doi.org/10.3390/s151229858

Badr El Din Ahmed, A., & Sayed Elaraby, I. (2014). *PER: A prediction for Student's Performance Using Decision Tree ID3 Method*. *India - World Journal of Computer Application and Technology*. https://doi.org/10.13189/wjcat.2014.020203

Barrow, D. K., & Crone, S. F. (2013). Crogging (cross-validation aggregation) for forecasting - A novel algorithm of neural network ensembles on time series subsamples. *Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN.2013.6706740

Cambria, E. (2016). Affective Computing and Sentiment Analysis. *IEEE Intelligent Systems*, *31*(2), 102–107. https://doi.org/10.1109/MIS.2016.31

Chandani, V. (2015). Komparasi Algoritma Klasifikasi Machine Learning Dan Feature Selection pada Analisis Sentimen Review Film. *Journal of Intelligent Systems*, *1*(1), 55–59. Retrieved from http://journal.ilmukomputer.org/index.php/jis/article/view/10

Chen, F., Deng, P., Wan, J., Zhang, D., Vasilakos, A. V., & Rong, X. (2015). Data mining for the internet of things: Literature review and challenges. *International Journal of Distributed Sensor Networks*, *2015*(i). https://doi.org/10.1155/2015/431047

Christian, H., Agus, M. P., & Suhartono, D. (2016). Single Document Automatic Text Summarization Using Term Frequency-Inverse Document Frequency (Tf-Idf). *ComTech: Computer, Mathematics and Engineering Applications*, *7*(4), 285–294. https://doi.org/http://dx.doi.org/10.21512/comtech.v7i4.3746

Erra, U., Senatore, S., Minnella, F., & Caggianese, G. (2015). Approximate TF-IDF based on topic extraction from massive message stream using the GPU. *Information Sciences*, *292*, 143–161. https://doi.org/10.1016/j.ins.2014.08.062

Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, *56*(4), 82. https://doi.org/10.1145/2436256.2436274

Gayathri, K., & Marimuthu, A. (2013). Text document pre-processing with the KNN for classification using the SVM. *7th International Conference on Intelligent Systems and Control, ISCO 2013*, 453–457. https://doi.org/10.1109/ISCO.2013.6481197

Gurusamy, V., & Kannan, S. (2014). Preprocessing Techniques for Text Mining. *RTRICS*. https://doi.org/10.5853/jos.2016.00885

He, W., Zha, S., & Li, L. (2013). Social media competitive analysis and text mining: A case study in the pizza industry. *International Journal of Information Management*, *33*(3), 464–472. https://doi.org/10.1016/j.ijinfomgt.2013.01.001

Ibrahim, M., Abdillah, O., Wicaksono, A. F., & Adriani, M. (2016). Buzzer Detection and Sentiment Analysis for Predicting Presidential Election Results in a Twitter Nation. *Proceedings - 15th IEEE International Conference on Data Mining Workshop, ICDMW 2015*, 1348–1353. https://doi.org/10.1109/ICDMW.2015.113

Ibrahim, N. U. R., Bacheramsyah, T. F., & Hidayat, B. (2018). Pengklasifikasian Grade Telur Ayam Negeri menggunakan Klasifikasi K-Nearest Neighbor berbasis Android, *6*(2), 288–302.

Lever, J., Krzywinski, M., & Altman, N. (2016). Points of Significance: Model selection and overfitting. *Nature Methods*, *13*(9), 703–704. https://doi.org/10.1038/nmeth.3968

Mitchell B.O., J. B. O. (2014). Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *4*(5), 468–481. https://doi.org/10.1002/wcms.1183

Neneng, N., Adi, K., & Isnanto, R. (2016). Support Vector Machine Untuk Klasifikasi Citra Jenis Daging Berdasarkan Tekstur Menggunakan Ekstraksi Ciri Gray Level Co-Occurrence Matrices (GLCM). *Jurnal Sistem Informasi Bisnis*, *6*(1), 1. https://doi.org/10.21456/vol6iss1pp1-10

Patodkar, V. N., & I.R, S. (2016). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *Ijarcce*, *5*(12), 320–

322. https://doi.org/10.17148/IJARCCE.2016.51274

Plisson, J., Lavrac, N., & Mladenić, D. D. (2004). A rule based approach to word lemmatization. *Proceedings of the 7th International Multiconference Information Society (IS'04)*, (November), 83–86. https://doi.org/10.1002/jbio.201500007

Prasetyo. (2014). DATA MINING MENGOLAH DATA MENJADI INFORMASI MENGGUNAKAN MATLAB. *Penerbit Andi*. https://doi.org/10.1017/CBO9781107415324.004

Probo, R. D., Irawan, B., Rumani, R., 3, M., S1, P., & Komputer, S. (2016). ANALISIS DAN IMPLEMENTASI PERBANDINGAN ALGORITMA KNN (K- NEAREST NEIGHBOR) DENGAN SVM (SUPPORT VECTOR MACHINE) UNTUK PREDIKSI PENAWARAN PRODUK Comparative Analysis and Implementation of KNN (K-Nearest Neighbor) with SVM (Support Vector Machine) Algorithm , *3*(3), 4988–4995.

Serrano-Guerrero, J., Olivas, J. A., Romero, F. P., & Herrera-Viedma, E. (2015). Sentiment analysis: A review and comparative analysis of web services. *Information Sciences*, *311*, 18–38. https://doi.org/10.1016/j.ins.2015.03.040

Srividhya, V., & Anitha, R. (2010). Evaluating Preprocessing Techniques in Text Categorization. *International Journal of Computer Science and Application*, 49–51.

Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, *57*, 117–126. https://doi.org/10.1016/j.eswa.2016.03.028

Utami, L. A. (2017). Analisis Sentimen Opini Publik Berita Kebakaran Hutan Melalui Komparasi Algoritma Support Vector Machine Dan K-Nearest Neighbor Berbasis Particle Swarm Optimization. *Jurnal Pilar Nusa Mandiri*, *13*(1), 103–112. Retrieved from http://ejournal.nusamandiri.ac.id/ejurnal/index.php/pilar/article/view/344/276

Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing and Management*, *50*(1), 104–112. https://doi.org/10.1016/j.ipm.2013.08.006

Varma, M. K. S., Rao, N. K. K., Raju, K. K., & Varma, G. P. S. (2016). Pixel-Based Classification Using Support Vector Machine Classifier. *Proceedings - 6th International Advanced Computing Conference, IACC 2016*, 51–55. https://doi.org/10.1109/IACC.2016.20

Yadav, S., & Shukla, S. (2016). Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. *Proceedings - 6th International Advanced Computing Conference, IACC 2016*, (Cv), 78–83. https://doi.org/10.1109/IACC.2016.25

Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of Translational Medicine*, *4*(11), 218–218. https://doi.org/10.21037/atm.2016.03.37