JOURNAL OF INFORMATICS DEVELOPMENT

E-ISSN : 2963-0568, P-ISSN : 2963-055X
Available online at:
https://ejournal.itbwigalumajang.ac.id/index.php/jid

Analysis Sentiment Alun-Alun Lumajang Review using Support Vector Machine

Maysas Yafi Urrochman

Department of Informatics, Institut Teknologi dan Bisnis Widya Gama Lumajang, Indonesia

Corresponding Author: Maysas Yafi Urrochman (maysasyafi@gmail.com)

ARTICLE INFO

Date of entry:
12 April 2025
Revision Date:
20 April 2025
Date Received:
24 April 2025

ABSTRACT

Alun-Alun Lumajang is one of room the public that becomes center activity community and tourists . Perception public to place This can measured through analysis sentiment to reviews available on digital platforms such as Google Maps. Research This aiming For classify sentiment review the use Support Vector Machine (SVM) method , one of the effective machine learning algorithms For task classification text . Data used in the form of review collected text from Google Maps, then through pre-processing data such as cleaning text , tokenization , and deletion stopword . Sentiment label determined manually to be three categories : positive , negative , and neutral . Next , the data is extracted use TF-IDF technique before classified using SVM. Research results show that SVM algorithm is capable of classify sentiment with level high accuracy , making it proper method For analysis opinion public based on text . Findings This expected can give input for government area in increase quality services and management room public in Lumajang.

Keywords: Google Maps, Alun-Alun Lumajang, Sentiment Analysis, Support Vector Machine.

Cite this as: Urrochman, M. Y. (2025). Analysis Sentiment Alun-Alun Lumajang Review using Support Vector Machine. *Journal of Informatics Development*, *3*(2), 47–57. https://doi.org/10.30741/jid.v3i2.1555

INTRODUCTION

One of room the public who have role strategic in the Regency Lumajang is Alun-Alun Lumajang. This location No only become center activity citizens , but also become destination interesting tour for visitors from outside area . As room active openness , perception and experience public towards Alun-Alun Lumajang become aspect important things to do be noticed For development and improvement quality service .

In the digital era, society in a way active share his experiences and opinions through reviews on various online platforms, such as Google Maps. Until April 2025, Alun-Alun Lumajang recorded own more from 14,000 reviews public on Google Maps, with diverse sentiment start from praise to cleanliness and comfort , to criticism to facilities and security . Reviews This contain information valuable related satisfaction , criticism , and hope user to a place . Therefore that , analysis sentiment to review the can used as one of the form evaluation data -based on perception public (Jihad, Adiwijaya, and Astut, 2021) .

JOURNAL OF INFORMATICS DEVELOPMENT
E-ISSN : 2963-0568, P-ISSN : 2963-055X
Available online at:
https://ejournal.itbwigalumajang.ac.id/index.php/jid

Analysis sentiment is technique in processing Language natural (NLP) which aims For identify and classify opinion or emotion in a text . Some study previously has prove effectiveness method This in evaluate quality service public . For example , a study by (Aulia, Arifin, and Mayasari, 2021) show that algorithm based on learning machine capable classify opinion text with accuracy high . In addition , research by (Eskiyaturrofikoh and Suryono, 2024) which analyzed sentiment review park city using SVM shows level accuracy reached 87%, proving that approach This effective used in context evaluation room public .

A number of research that examines analysis sentiment review tour use Support Vector Machine (SVM) method . Various studies has implementing SVM for classify sentiment from comment tourists who are obtained through platforms such as Google Maps and Twitter (Syahlan, Irmayanti, and Alam, 2023) (Gunawan, Kuswanto, and others, 2024) (Ipmawati, Saifulloh, and Kusnawi, 2024) ; (Ipmawati, Saifulloh, and Kusnawi, 2024) (Silviana, Astuti, and Basysyar, 2024) . Research results show that SVM is capable in a way effective differentiate sentiment become category positive , negative , and neutral , with reported accuracy is in the range of 81% to 90.72% (Syahlan, Irmayanti, and Alam, 2023) (Silviana, Astuti, and Basysyar, 2024) . Pre-processing process text , such as tokenization , normalization , and stemming, become stages important in prepare data before done classification (Lumbanraja, Gaol, Shofiana, and Junaidi, 2024) . Research the confirm role important analysis sentiment in understand view traveler as well as in effort management destination tour in a way more optimal (Ipmawati, Saifulloh, and Kusnawi, 2024) . Findings This show that approach based on SVM in analysis sentiment can give valuable information for the takers policies in the sector tourism , use support decision data -based and improve quality experience traveler (Lumbanraja, Gaol, Shofiana, and Junaidi, 2024) (Silviana, Astuti, and Basysyar, 2024) .

One of method many classifications used in analysis sentiment is Support Vector Machine (SVM) algorithm , which is known own performance tall in handling text data Because his ability separate data class with optimal margin. SVM has also been proven efficient in handle data with dimensions high , like representation text based on TF-IDF (Adrian, Putra, Rafialdy, and Rakhmawati, 2021) (Safrudin, Martanto, and Hayati, 2024) .

Study This aiming For classify sentiment public towards Alun-Alun Lumajang based on review taken from Google Maps using SVM method . With do classification sentiment in a way automatic , expected can obtained description general about opinion society , which then can made into base consideration for government area in take policy related management public space (Dayani, Nurcahyo, and others, 2024) .

METHOD

There are some stages methods that can explained as following :

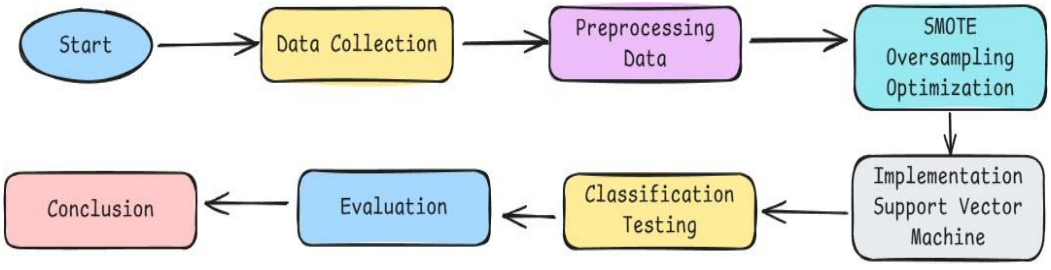

Figure 1. Research Flow Diagram

JOURNAL OF INFORMATICS DEVELOPMENT
E-ISSN : 2963-0568, P-ISSN : 2963-055X
Available online at:
https://ejournal.itbwigalumajang.ac.id/index.php/jid

Collection and Preprocessing

Study This started with take sample data obtained from review of Alun-Alun Lumajang found on Google Maps API. The dataset was taken obtained use how to scrape reviews on Google Maps API with take score reviews and comments user application . After the data is obtained , the next stage furthermore namely data preprocessing. Data preprocessing for sentiment analysis called as text preprocessing. Stages This started with data cleaning, namely data deletion in the form of review empty only in the form of score review just without comments . In the research this , splitting data with dividing the data into train data and test data is done if the data passes through a series of text preprocessing processes have been carried out finished and ready For done classification . Then , case folding is performed to delete sign read , remove non- alphabetic characters and change all sentence become lower case. After that , the tokenization process is carried out which is used For break sentence into words. Then , the stopwords process is carried out , namely take out important words and delete unnecessary words general . After the stopwords process , the stemming process is carried out , namely change the word that has affix become basic words . The process of stopwords and stemming in research This using the WordNetLemmatizer , Porterstemmer libraries , and Lancasterstemmer on Python.

After text preprocessing stage , stage next namely word weighting. Stages This give score on word occurrences , which uses the Process Document from Data operator in it use term frequency – inverse document frequency (TF-IDF) method . Term Frequency (TF) is used For count frequency emergence a word in One documents , while Inverse Document Frequency (IDF) is used For measure the importance of the word in overall document . Formula TF-IDF calculation can seen in Equation 1.

$$tf.idf\ (t,d,D) = (1 + \log(f_{t,d}))\ (log\ \frac{N}{df_t})$$

Information:

$f_{t,}$ = number of words t in document d
N = total quantity documents in a document collection
df_t = amount documents containing the word t, if the word does not have a value in all documents (0) then the value is 1

Optimization and Classification

After the word weighting process is complete , then furthermore that is stages furthermore namely data that has been processed furthermore done SMOTE Oversampling optimization , (Dayani, Nurcahyo, and others, 2024) which was carried out with change sample class minority become almost The same with class majority with copy in a way random sample class minority . Next , namely classify each review , with classification sentiment positive and negative . Classification manually specified with mark score reviews contained in the Google Maps API. Valuable review scores three up to five categorized as sentiment positive , while a valuable score One or two categorized as sentiment negative (Setiawan and Suryono, 2024) .

In discussion this , classification test applied with utilise technique Support Vector Machine. SVM works with method find the best hyperplane that separates the data into into two classes (positive and negative , for example). This hyperplane is a line (in 2D), a plane (in 3D), or a multidimensional boundary in space. features , which separate classes with the widest margin . Formula For classification method Support Vector Machine is shown in equation 2.

$$f(x) = w^T x + b$$

Where:
x = feature vector (result extraction text such as TF-IDF)
w = weight vector (which determines direction and slope of the hyperplane)
b = bias (determines hyperplane position)

JOURNAL OF INFORMATICS DEVELOPMENT
E-ISSN : 2963-0568, P-ISSN : 2963-055X
Available online at:
https://ejournal.itbwigalumajang.ac.id/index.php/jid

Model Evaluation

After method implemented , steps next is measure performance classification applying confusion matrix. Based on the confusion matrix, it is obtained results accuracy , precision , and recall. Accuracy measure accuracy classification , which is obtained from the ratio of positive and negative classified data with Correct to all data. Precision measure relevance of the data found with the information needed , while recall assesses how far the system succeed get return relevant information . The confusion matrix is shown in Table 1.

Table 1. *Confusion Matrix*

		Current Class	
		True	**False**
Prediction Class	**True**	TP (*True Positive*)	FP (*False Positive*)
	False	FN (*False Negative*)	TN (*True Negative*)

Information:
True Positive (TP) : If the predicted data is positive and matches the actual value (positive).
False Positive (FP) : If the predicted data does not match the actual value
False Negative (FN) : If the predicted value is negative and the actual value is positive
True Negative (TN) : If it is true between the negative prediction and the actual negative.

Measurement of accuracy, precision, *recall* , and *F- Measure values* based on confusion matrix formulate in equations 6 to 9:

Accuracy $= \frac{TP + TN}{TP + FN + FP + TN}$ (6)

Precision $= \frac{}{TP + FN}$ (7)

Recall $= \frac{}{TP + FP}$ (8)

F-Measure $= \frac{2 \times TP}{2 \times TP + FP + FN}$ (9)

RESULTS AND DISCUSSION

Stage beginning before implement method is collect review data about Crater Ijen obtained from the Google Maps API. Data collection was carried out with Scraping method using the Outscrapper site . From the scraping results , 14,515 reviews were obtained with varying scores , starting from from 1 to 5 stars . In detail , there are 411 reviews. 1 star , 213 reviews 2 stars , 860 reviews 3 stars , 2,566 reviews 4 stars , and 10,465 reviews 5 stars . In research this , classification based on score review shared into two categories : sentiment positive For review with score 3 to 5 stars , and sentiment negative For review with score 1 to 2 stars .

After the data is collected, the next stage is text preprocessing . At this stage, the text preprocessing is done in four steps, namely case folding , tokenizing , stopwords , and stemming . The entire process is done using Python language . The dataset is imported using the library pandas , then processed through the following series of preprocessing steps :

1. Case folding : This step changes all text to lowercase and removes non-letter characters (a-z), such as symbols, numbers, and emojis .
2. Tokenizing : This process separates text into individual words.

E-ISSN : 2963-0568, P-ISSN : 2963-055X
Available online at:
https://ejournal.itbwigalumajang.ac.id/index.php/jid

3. Stopwords : Irrelevant common words are removed to improve analysis efficiency.
4. Stemming : This stage groups words that have the same root and meaning. But its shape different

Because some big review using English, the process of stopwords and stemming is not using the Sastrawi library For Indonesian, but use WordNetLemmatizer, PorterStemmer, as well as LancasterStemmer from Python. Next is the weighting per word of the dataset , which is done using TF-IDF. From a total of 14,515 reviews on the dataset that was done text preprocessing, there are still 10 empty review data due to one of the preprocessing stages, and 5 review data that include telephone numbers, so in order to maintain privacy, these reviews are also deleted. The total data to be used is 14,500 reviews, of which there are a total of 58,791 words obtained up to the stemming stage, and there are 5,411 different words if duplicated words are removed. TF-IDF weighting is done by calculating the TF value obtained from the number of occurrences of words from each review. The results of the TF value calculation are shown in Table 2.

TF Calculation Results

Data	TF(blue)	TF(crater)	TF(fire)	TF(beautiful)	TF(view)
1	1	2	0	1	0
2	1	2	1	0	2
3	0	0	0	0	0
4	1	2	0	0	1
5	0	0	0	0	0
6	0	0	0	0	0
7	2	0	2	0	0
8	0	0	0	1	0
9	0	0	0	0	0
10	1	4	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮
14,500	0	0	0	0	0

After Term Frequency (TF) value found , steps furthermore is calculate Document Frequency (DF). The DF value is obtained with count amount document in the dataset containing the word , where the frequency its emergence more from 0. Calculation results DF value can seen in Table 3.

Table 3. DF values

DF(blue)	DF(crater)	DF(fire)	DF(beautiful)	DF(view)
804	663	640	606	604

After count Document Frequency (DF) value , steps next is calculate the Inverse Document Frequency (IDF) for each keyword . The IDF value is used For balancing the influence of frequent words appears throughout the dataset, with method calculate the base 10 log of the total number of datasets divided by with DF value of the word . The more big DF value , the more small the IDF value , and vice versa . Based on DF values listed in Table 7, IDF values of each word are shown in Table 4.

JOURNAL OF INFORMATICS DEVELOPMENT
E-ISSN : 2963-0568, P-ISSN : 2963-055X
Available online at:
https://ejournal.itbwigalumajang.ac.id/index.php/jid

Table 4. IDF values

IDF(blue)	IDF(crater)	IDF(fire)	IDF(beautiful)	IDF(view)
0.5796	0.6634	0.6787	0.7024	0.7038

After count Inverse Document Frequency (IDF) value , steps next is calculate Term Frequency-Inverse Document Frequency (TF-IDF). The TF-IDF value is obtained with multiply TF value with IDF value . As For example , for the word "crater" in dataset number 10, the TF value is is 4, while The IDF value for the word "crater" is 0.6634. So, the TF-IDF value for the word is 4 x 0.6634=2.6536. The result of the TF-IDF calculation for All words are in Table 5.

Table 5. TF-IDF Calculation Results

Dataset	TF-IDF (blue)	TF-IDF (crater)	TF-IDF (fire)	TF-IDF (beautiful)	TF-IDF (view)
1	0.5796	1,3268	0	0.7024	0
2	0.5796	1,3268	0.6787	0	1.4076
3	0	0	0	0	0
4	0.5796	1,3268	0	0	0.7038
5	0	0	0	0	0
6	0	0	0	0	0
7	1,1592	0	1,3374	0	0
8	0	0	0	0.7024	0
9	0	0	0	0	0
10	0.5796	2.6536	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮
3054	0	0	0	0	0

Implementation Support Vector Machine Classification

Dataset used For implementation classification consists of of 14,500 attributes , which contain words that have been processed through text preprocessing with the value of each attribute based on TF-IDF weighting . Classification shared into two classes : class positive For review with 3 to 5 stars and class negative For review with 1 and 2 stars . In research Here , there are 2,200 reviews with class negative and 12,300 reviews class positive .

Implementation classification done use Language Python programming . Stages beginning before classification is importing the dataset that has been prepared and converted to in .csv extension format to in Python code . Implementation Support Vector Machine is done with method import the sklearn library into Python and import SVC. The program code for implementation NBC methods are shown in Table 6.

Table 6. SVM Library Import Code

Line	Code
1	!pip install Sastrawi sklearn pandas numpy matplotlib
2	
3	import pandas as pd
4	import numpy as np
5	from sklearn.model_selection import train_test_split
6	from sklearn.feature_extraction.text import TfidfVectorizer
7	from sklearn.svm import SVC
8	from sklearn.metrics import classification_report , confusion_matrix
9	from Sastrawi.StopWordRemover.StopWordRemoverFactory import

E-ISSN : 2963-0568, P-ISSN : 2963-055X
Available online at:
https://ejournal.itbwigalumajang.ac.id/index.php/jid

10	StopWordRemoverFactory from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
11	import again

Table 7. Load Dataset

Line	Code
1	df = pd.read_csv ('/content/ulasan_alun_alun.csv') #
2	df.dropna (inplace = True)
3	df.head ()
4	

Table 8. Text Preprocessing

Line	Code
1	factory_stopwords = StopWordRemoverFactory ()
2	stopword = factory_stopwords.create_stop_word_remover ()
3	
4	factory_stemmer = StemmerFactory ()
5	stemmer = factory_stemmer.create_stemmer ()
6	
7	def preprocessing(text):
8	# Case folding
9	text = text.lower ()
10	# Remove symbols, numbers, punctuations
11	text = re.sub (r'[^a- zA -Z\s]', '', text)
12	# Stopword removal
13	text = stopword.remove (text)
14	# Stemming
15	text = stemmer.stem (text)
16	return text
17	
18	df [' clean_text '] = df [' review '].apply(preprocessing)
19	df [[' review ', ' clean_text ', 'label']].head()

Table 9. Extraction feature with TF-IDF

Line	Code
1	tfidf = TfidfVectorizer (max_features = 1000)
2	X = tfidf.fit_transform (df [' clean_text ']). toarray ()
3	y = df ['label']
4	
5	# Split data into train and test
6	X_train , X_test , y_train , y_test = train_test_split (X, y, test_size =0.2, random_state =42)

Table 10. SVM model training

Line	Code
1	# Step 6: Training the SVM model
2	model_svm = SVC(kernel='linear', C=1.0)
3	model_svm.fit (X_train , y_train)
4	
5	# Step 7: Evaluation

JOURNAL OF INFORMATICS DEVELOPMENT
E-ISSN : 2963-0568, P-ISSN : 2963-055X
Available online at:
https://ejournal.itbwigalumajang.ac.id/index.php/jid

```
6   y_pred = model_svm.predict ( X_test )
7
8   # Displaying metric evaluation
9   print("Classification  Report:\n")
10  print( classification_report ( y_test , y_pred ))
11
12  # Confusion matrix
13  cm = confusion_matrix ( y_test , y_pred )
14  print("Confusion Matrix:\n", cm)
```

Performance Test

Performance test stages required For evaluate results implemented classification use Support Vector Machine via mapping on confusion matrix. In the research this , performance test done with divide the dataset in ratio 40:60. (60% training data 40% test data), 30:70 (70% training data 30% test data), 25:75 (75% training data 25% test data), 20:80 (80% training data 20% test data), and 10:90 (10% training data 90% test data). Python program code for make Confusion matrix mapping is in Table 10

Table 10. Confusion Matrix Mapping Program Code

Line	Code
1	from sklearn.metrics import confusion_matrix
2	confusion_matrix (y_test , y_pred)
3	
4	from sklearn.metrics import classification_report
5	print(classification_report (y_test , y_pred))

Lines 1-2 contain code For calls the sklearn.metrics library For make confusion matrix mapping based on y_test (class actual) and y_pred (class prediction). Lines 4-5 contain code For make report classification that contains mark accuracy , precision , recall, and F1-score. Confusion matrix mapping for each experiment found in Tables 11 to 15.

Table 11. Confusion Matrix 40% Test Data

		Class Current	
		Negative	**Positive**
Class Prediction	**Negative**	1	18
	Positive	145	9,321

Table 12. Confusion Matrix 30% Test Data

		Class Current	
		Negative	**Positive**
Class Prediction	**Negative**	1	13
	Positive	352	761

Table 13. Confusion Matrix 25% Test Data

		Class Current	
		Negative	**Positive**
Class Prediction	**Negative**	1	13
	Positive	111 2	639 5

Table 14. Confusion Matrix 20% Test Data

		Class Current	
		Negative	**Positive**
Class Prediction	**Negative**	1	13

E-ISSN : 2963-0568, P-ISSN : 2963-055X
Available online at:
https://ejournal.itbwigalumajang.ac.id/index.php/jid

	Positive	89 1	508 1

Table 15. Confusion Matrix 10% Test Data

		Class Current	
		Negative	**Positive**
Class Prediction	**Negative**	1	7
	Positive	44	254

The confusion matrix in Tables 11 to 15 is results obtained For NBC classification only , without added with optimization of SMOTE and ADASYN, so that produce mark accuracy , precision , recall, and different F1-scores . Details results performance classification shown in Tables 16 to 19.

Table 16. Accuracy Results

Test Data Percentage	Classification Methods		
	NBC	**NBC + SMOOTH**	**NBC + ADASYN**
40%	83.47 %	90.01 %	88.40 %
30%	83.10%	89.84 %	88.19%
25%	83.77%	89.88 %	88.56 %
20%	83.31%	89.68%	89.78%
10%	83.33%	89.51%	88.37%

Table 17. Precision Results

Test Data Percentage	Classification Methods		
	NBC	**NBC + SMOOTH**	**NBC + ADASYN**
40%	98.26 %	100.00%	100.00%
30%	98.32%	100.00%	100.00%
25%	98.01%	100.00%	100.00%
20%	97.50%	100.00%	100.00%
10%	97.32%	100.00%	100.00%

Table 18. Recall Results

Test Data Percentage	Classification Methods		
	NBC	**NBC + SMOOTH**	**NBC + ADASYN**
40%	84.70 %	80.57 %	77.26 %
30%	84.27%	80.51 %	77.17%
25%	85.20%	80.26 %	77.86 %
20%	85.09%	79.87%	80.16%
10%	85.23%	78.72%	77.20%

Table 19. F1-Score Results

Test Data Percentage	Classification Methods		
	NBC	**NBC + SMOOTH**	**NBC + ADASYN**
40%	90.98%	89.24%	87.17%
30%	90.76%	89.20%	87.11%
25%	91.16%	89.04%	87.55%
20%	90.88%	88.97%	88.99%
10%	90.88%	88.09%	87.13%

E-ISSN : 2963-0568, P-ISSN : 2963-055X
Available online at:
https://ejournal.itbwigalumajang.ac.id/index.php/jid

Based on Tables 16 to 19, it can be seen that addition technique optimization such as SMOTE and ADASYN in SVM classification are capable of increase mark accuracy and precision . Improvement accuracy reached 10.23% when use combination of SVM and SMOTE, and 5.47% when using SVM and ADASYN. On the metrics precision , there is increase by 3.68% both in SVM with SMOTE and with ADASYN. However , the recall and F1-score values are actually experience decline compared to SVM without optimization , which is likely caused by the oversampling process in the class minority that causes imbalance new . Although Thus , the decline performance on recall and F1-score is better small on the combination of SVM and SMOTE compared with SVM and ADASYN. Therefore that , from results testing model performance , can concluded that The combination of SVM and SMOTE provides results best in classification sentiment Alun-Alun Lumajang review on Google Maps. Findings This in harmony with a number of study previously , such as studies analysis sentiment to impact Covid-19 economy on Twitter using SVM and obtaining accuracy by 83%. In addition , other research on analysis sentiment on Moon Knight movie reviews shows that SVM method produces accuracy more tall compared to Naive Bayes, with score reached 72.38%.

CONCLUSION

Based on results testing , implementation Support Vector Machine (SVM) method combined with technique SMOTE optimization proven give performance best in classification sentiment reviews of Alun-Alun Lumajang on Google Maps. Combination This produce improvement significant accuracy and precision compared to with SVM without optimization , although there is A little decrease in recall and F1-score values . This Still within reasonable limits and more small compared to with ADASYN usage . Findings This reinforced by the results study previously which also showed effectiveness of SVM in analysis sentiment on various context , such as social media and movie reviews.

REFERENCES

Adrian, MR. Rakhmawati, NA (2021). Comparison of Random Forest and SVM Classification Methods in PSBB Sentiment Analysis. *Upgris Informatics Journal* , *7* (1).

Aulia, TMP. Mayasari, R. (2021). Comparison of Kernel Support Vector Machine (SVM) in the Application of Covid-19 Vaccination Sentiment Analysis. *SINTECH (Science and Information Technology) Journal* , *4* (2), 139–145.

Dayani, AD. others. (2024). Sentiment Analysis of Public Opinion on Social Media Twitter Using the Support Vector Machine Method. *Jurnal KomtekInfo* , 1–10.

Eskiyaturrofikoh, E., & Suryono, RR (2024). Sentiment Analysis of Application X on Google Play Store Using Na{\"\i}ve Bayes Algorithm and Support Vector Machine (SVM). *JIPI (Scientific Journal of Informatics Research and Learning)* , *9* (3), 1408–1419.

Gunawan, AH others. (2024). Exploration of Support Vector Machine Algorithm for Sentiment Analysis of Tourist Destinations in Indonesia. *Bit-Tech* , *7* (2), 323–330.

Ipmawati, J. Kusnawi, K. (2024). Sentiment Analysis of Tourist Attractions Based on Reviews on Google Maps Using the Support Vector Machine Algorithm: Sentiment Analysis of Tourist Attractions Based on Reviews on Google Maps Using the Support Vector Machine Algorithm. *MALCOM: Indonesian Journal of Machine Learning and Computer Science* , *4* (1), 247–256.

Jihad, MAA. Astut, W. (2021). Sentiment analysis of movie reviews using random forest algorithm. *EProceedings of Engineering* , *8* (5).

Lumbanraja, FR Junaidi, A. (2024). Implementation of SMOTE and Support Vector Machine in Classification of Imbalanced Arginine Methylation Data. *Pepadun Journal* , *5* (1), 27–37.

Safrudin, M. Hayati, U. (2024). Comparison of the Performance of Na{\"\i}ve Bayes and Support Vector Machine for Genshin Impact Game Review Sentiment Classification. *JATI*

E-ISSN : 2963-0568, P-ISSN : 2963-055X
Available online at:
https://ejournal.itbwigalumajang.ac.id/index.php/jid

(Informatics Engineering Student Journal) , *8* (3), 3182–3188.

Setiawan, A., & Suryono, RR (2024). Sentiment Analysis of Indonesian Capital City Using Support Vector Machine Algorithm and Na{\"\i}ve Bayes. *Edumatic: Journal of Informatics Education* , *8* (1), 183–192.

Silviana, S. Basysyar, FM (2024). Application of Support Vector Machine (SVM) Algorithm on Tourist Reviews of Kuningan Regency. *JATI (Informatics Engineering Student Journal)* , *8* (1), 259–265.

Syahlan, MS. Alam, S. (2023). Sentiment Analysis of Tourist Attractions from Visitor Comments Using the Support Vector Machine (SVM) Method. *Simtek: Journal of Information Systems and Computer Engineering* , *8* (2), 315–319.