

# Support Vector Machine Implementation for Classifying Public Sentiment on Electronic Stamps use in Civil Servant Registration

Nilna Adiba Kamal<sup>1</sup>, Suparti<sup>2</sup>, Puspita Kartikasari<sup>3</sup>

<sup>1,2,3</sup> Department of Statistics, Diponegoro University, Indonesia

ARTICLE INFO	ABSTRACT
<p><b>Published Online:</b> 29 May 2025</p> <p><b>Corresponding Author:</b> Suparti</p>	<p>Civil Servants play a crucial role in government administration and national development. The selection process for Civil Servant Candidates is a critical stage in civil servant management, as it has long-term implications for organizational effectiveness. With the advancement of digital technology, the Indonesian government has implemented electronic stamps in the civil servant candidate registration process to enhance efficiency and transparency. However, this policy has received various responses from the public, particularly on the social media platform Twitter. Many users express positive opinions regarding the convenience of administrative digitalization, while others report technical issues in using electronic stamps. Therefore, sentiment analysis is needed to understand public responses to this policy. One of the effective methods for sentiment classification is Support Vector Machine (SVM), which can optimally separate positive and negative opinions. This study utilizes a dataset comprising 1,249 reviews collected from Twitter/X. The best SVM model is selected through hyperparameter tuning using the GridSearchCV technique. The findings indicate that the SVM model with cost = 100 and gamma = 0.01 achieves the best performance, with an accuracy of 92%, precision of 86.48%, recall of 86.48%, F1-score of 86.48%, and a Kappa-Statistic of 81%.</p>
<p><b>KEYWORDS:</b> Civil Servants; E-meterai; Sentiment Analysis; Twitter/X; Support Vector Machine</p>	

## I. INTRODUCTION

The role of Civil Servants (PNS) is crucial in supporting the country's administration and development. The recruitment process of Civil Servant Candidates (CPNS) is an important stage because it has a long-term impact on the effectiveness of public organizations. Along with the advancement of digital technology, the Indonesian government implemented e-stamps in the CPNS registration process to increase administrative efficiency and transparency.

Although the electronic stamps policy aims to increase the transparency and efficiency of CPNS registration, its implementation has triggered various responses on social media, especially Twitter/X. As a platform that represents real-time public discussions, Twitter/X is often used in sentiment analysis studies because it can record opinions along with time and location [1]. According to Katadata, Indonesia is the fourth largest Twitter/X user globally, with 27.5 million users.

The public has expressed various opinions via Twitter/X since the implementation of electronic stamps, ranging from support for digitalization to technical complaints, such as failure to

access and affix electronic stamps. These problems are considered to hinder the registration process and cause losses. Therefore, sentiment analysis is needed to understand the public response, by categorizing opinions into positive and negative sentiments.

Sentiment analysis is a technique for classifying public opinion into positive or negative sentiment [2]. Research by Herwinsyah and Witanti [3], shows that SVM can classify sentiment related to COVID-19 vaccination with 89% accuracy. Ratino et al. [4], also found that SVM is superior to NaïveIBayes, with accuracies of 80.23% and 78.02%, respectively. Meanwhile, research by Yusupa and Tarigan [5] showed that SVM is quite reliable in analyzing public sentiment towards electric vehicles in Indonesia, with an accuracy of 75.62%.

This research was conducted to develop an effective classification model utilizing the Support Vector Machine (SVM) algorithm to classify public sentiment toward the e-stamp policy in CPNS registration. This research uses data from user posts on Twitter/X, which reflect various public opinions. The classification model developed is expected to

provide a more objective picture of how the public views the policy.

## II. THEORETICAL FRAMEWORK

Sentiment analysis is the process of automatically understanding, extracting, and processing text data to identify the underlying sentiment. It classifies opinion-based text documents as either positive or negative in sentiment.

According to Runimearti et al. [6], text mining is a technique used to extract information from large volumes of textual data with the aid of software, aiming to discover patterns, themes, and keywords. This technique transforms unstructured text into valuable insights that support decision-making by identifying recurring patterns and themes, as well as uncovering relationships between documents. Text mining supports various text analysis tasks, including classification, clustering, and information extraction [7].

Twitter/X data often contains unique or irrelevant texts that do not contribute meaningful information to sentiment analysis. These texts are removed during the data pre-processing stage. The pre-processing steps include case folding, data cleaning (such as removing URLs, mentions, numbers, punctuation marks, and emoticons), and word normalization.

According to Feldman and Sanger [7], feature selection reduces the dimensionality of text data by eliminating irrelevant words, thereby improving the efficiency and accuracy of classification. In this study, the feature selection process involves several steps. Stopword removal is used to reduce the feature space by eliminating words that do not contribute meaningful [8]. Stemming is applied to convert words into their base or root forms [9], and tokenization is performed to split sentences into individual word units for further analysis.

Word weighting is intended to assign weights to words based on their importance within a document, thereby facilitating the classification process [10]. Term Frequency (TF) measures how often a word appears in a document, while Inverse Document Frequency (IDF) reflects how important a word is across the entire corpus. Words that occur frequently in many documents are assigned lower weights. The TF-IDF score is calculated by multiplying the term frequency (TF) by the inverse document frequency (IDF), as defined below:

$$W_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \times \ln\left(\frac{N}{df_{(i)}}\right) + 1$$

Where  $W_{i,j}$  represent Term Frequency-Inverse Document Frequency (TF-IDF),  $n_{i,j}$  is frequency of occurrence of the  $i$  term (word) in the  $j$  document,  $\sum_k n_{k,j}$  is total number of occurrences of words in document  $j$ ,  $N$  represent total number of all documents, and  $df_{(i)}$  is Number of documents in which term  $i$  appears.

The Support Vector Machine (SVM) algorithm is applied in this study due to its capability to classify both linear and nonlinear data by identifying the optimal hyperplane that

separates different classes [11]. The hyperplane serves as the best decision boundary, determined based on the maximum margin defined as the shortest distance between the hyperplane and the data points of each class, known as support vectors. The process of finding this optimal hyperplane with the maximum margin is the core principle of SVM [12].

The available data are denoted as  $x_i \in R^n$   $i = 1, 2, \dots, l$  and  $y_i \in \{-1, +1\}$  represents the class labels for the negative and positive classes, respectively. Thus, the dataset consists of pairs  $(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)$ , where each pair corresponds to one of the two categories to be classified using SVM. It is assumed that the two classes, -1 (negative) and +1 (positive), are linearly separable by a hyperplane [11]:

$$w^T x_i + b = 0, \dots \dots i = 1, 2, \dots, l$$

Where  $w$  is the vector representing the weight parameter,  $x_i$  represent vector data,  $b$  is bias or error value, and  $l$  represent number of data.

The data  $x_i$  that belongs to the +1 (positive) class category can be formulated as follows:

$$[(w^T x_i + b)] \geq 1$$

The data  $x_i$  that belongs to the -1 (negative) class category can be formulated as follows:

$$[(w^T x_i + b)] \leq -1$$

To find the largest margin, the closest distance between the hyperplane and the data points needs to be maximized, which is mathematically calculated using the following formula:

$$\frac{2}{\|w\|} \text{ dengan } \|w\| = \sqrt{w_1^2 + w_2^2 + \dots + w_n^2}$$

Quadratic Programming (QP) is employed to find the optimal hyperplane, and it is formulated as follows:

$$\min \frac{1}{2} \|w\|^2 \text{ with } \|w\|^2 = w^T w$$

subject to:

$$y_i (w^T x_i + b) \geq 1 \quad i = 1, 2, \dots, l$$

To solve the Equation, the Lagrange Multiplier method is used, which transforms the equation into the following:

$$L_p(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^n \alpha_i [y_i (w^T x_i + b) - 1]$$

Lagrange coefficient,  $\alpha_i$  is equal to or greater than zero  $\alpha_i \geq 0$ . The optimal solution to the equation is found by finding the first derivative of the Lagrange function derived with respect to the variables  $w$  and  $b$ , and equating it to zero.

$$\frac{\partial L}{\partial b} = 0 \rightarrow - \sum_{i=1}^n \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial w} = 0 \rightarrow \sum_{i=1}^n \alpha_i y_i x_i = w$$

$w^T w$  described as follows:

$$w^T w = \sum_{i=1}^n \alpha_i y_i (w^T x_i) = \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j (x_i^T x_j)$$

$L_p$  is transformed into the dual problem  $L_D$ . The optimal value is achieved by maximizing the dual function with respect to  $\alpha$ , which produces the best separating field [11].

$$\max_{\alpha} L_D = \max \left( \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j (x_i^T x_j) \right)$$

subject to,  $\alpha_i \geq 0 ; i = 1,2, \dots, l$  and  $\sum_{i=1}^n \alpha_i y_i = 0 ; i = 1,2, \dots, l$

Training data with a value of  $\alpha_i > 0$  is called a support vector [13].

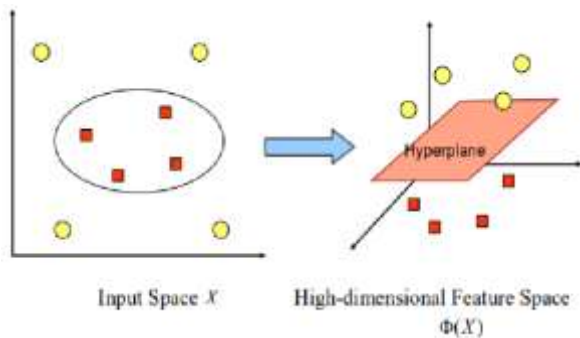
Data that are not perfectly separated cause constraints on SVM optimization. To overcome this, a soft margin technique is used by adding slack variables ( $\xi$ ), so that the equation becomes as follows:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\min \left( \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \right)$$

The function of the C parameter in SVM is to adjust the balance between the width of the margin and the number of classification errors ( $\xi$ ). The higher the value of C, the greater the penalty the model gives to misclassification [12].

Many real-world problems have data that cannot be separated by a straight line. To handle non-linearly separable data, SVM is modified with a kernel function that transforms the data to a higher dimension, thus allowing linear separation [11].



**Figure 1. Function  $\Phi$  Maps Data to High-Dimensional Space**

The non-linearly separated SVM data is transformed to a higher dimension using the  $\Phi$  function after the transformation, the hyperplane that separates the data can be found. Figure 1 illustrates this change. The mathematical notation for the mapping is as follows:

$$\Phi : R^d \rightarrow R^q \quad d < q$$

The SVM process finds the support vector through calculating the dot product of the data, which is transformed to a higher dimension  $\Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)^T$ . The kernel function facilitates a non-linear mapping to the feature space, formulated as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$$

Radial Basis Function (RBF) kernel is one of the commonly applied kernel function types.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

The RBF kernel uses  $\mathbf{x}_i$  and  $\mathbf{x}_j$  as data pairs. The gamma parameter ( $\gamma$ ) functions to adjust the width of the bell-shaped curve, where the curve will be narrower if the gamma value is greater. The calculation of classification results is done using the following formula [12].

$$f(\Phi(x)) = \text{sign} \left( \sum_{i=1}^{ns} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b \right)$$

$$f(\Phi(x)) = \begin{cases} 1, & \text{jika } \sum_{i=1}^{ns} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b > 0 \\ -1, & \text{jika } \sum_{i=1}^{ns} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b < 0 \end{cases}$$

Where  $f(\Phi(x))$  is the result of classifying data  $x$ ,  $y_i$  represent data label,  $\alpha_i$  is lagrange coefficient,  $K(\mathbf{x}_i, \mathbf{x}_j)$  is the application of kernel function for test data and training data,  $b$  represent bias, and  $ns$  is the number of support vectors.  $ns$  in the equation refers to the subset of training data selected by the SVM algorithm to determine the hyperplane, known as the support vector [12].

Model hyperparameter optimization is performed using GridSearchCV, which explores hyperparameter combinations and calculates cross-validation scores. GridSearchCV utilizes k-fold cross-validation, where the training data is divided into K parts, and the model is evaluated K times, with each part used alternately as test data. The average accuracy of each iteration is used to determine the best hyperparameter combination. Scikit-learn, a Python library, provides various machine learning algorithms, including GridSearchCV for efficient and structured hyperparameter tuning [14].

Performance evaluation is performed to assess how well the classification model predicts the correct class. Confusion matrix, which compares predicted. with. actual. values, is used to calculate metrics such as accuracy, precision, and recall. The confusion matrix consists of four main indicators: True Positive (TP) is positive data that is correctly classified as positive, False Negative (FN) is positive data that is mistakenly predicted as negative, True Negative (TN) is negative data that is correctly predicted as negative, and False Positive (FP) is negative data that is incorrectly classified as positive.

**Table I. Confusion Matrix 2 Class**

		Prediction Class	
		Positive	Negative
Actual Class	Positive	TP ( <i>True Positiv e</i> )	FN ( <i>False Negativ e</i> )
	Negative	FP ( <i>False Positiv e</i> )	TN ( <i>True Negativ e</i> )

There are several accuracy measurements used, namely accuracy, precision, recall, f1-score, and kappa statistic.

**Table II. Evaluation Indicators in the Classification Model**

Ukuran	Rumus
<i>Accuracy</i>	$\text{Accuracy} = \frac{TP + TN}{(TP + FP + TN + FN)}$
<i>Precision</i>	$\text{Precision} = \frac{TP}{(TP + FP)}$

$$\begin{aligned}
 \text{Recall} & \quad \text{Recall} = \frac{TP}{(TP + FN)} \\
 \text{F1-score} & \quad 1 - \text{Score} \\
 & \quad = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\
 \text{Kappa Statistic} & \quad P_0 = \text{Accuracy} \\
 & \quad \text{Total} = TP + TN + FP + FN \\
 & \quad P_0 = \left[ \left( \frac{TP+FP}{\text{Total}} \right) \left( \frac{TP+FN}{\text{Total}} \right) \right] + FP + \\
 & \quad TN \quad \text{Kappa} = \frac{P_0 - P_c}{1 - P_c}
 \end{aligned}$$

Determination of the optimal parameters of the model is determined through the hyperparameter tuning process of the cost and gamma values using Grid Search Cross Validation.

8. Support Vector Machine Classification Stage  
Classify reviews based on positive and negative reviews using the Support Vector Machine method.
9. Classification Model Evaluation  
Evaluate the classification model that has been formed using Confusion Matrix to calculate accuracy, precision, recall, F1-Score, and kappa statistics.
10. Visualization  
Visual analysis is performed using word clouds to identify the most frequently occurring words in the data.

### III. RESEARCH METHODS

The data analyzed in this study comes from secondary sources in the form of 1300 Indonesian tweets obtained by searching using the keyword “E-Meterai CPNS” in Indonesian. The data collected are tweets posted during the administrative selection period of CPNS 2024, from August 20 to September 17, 2024. The variables in this study consist of independent variables and dependent variables. The independent variable refers to public opinion on Twitter/X social media towards the use of e-stamps in the registration of Civil Servant Candidates (CPNS), while the dependent variable is the sentiment class which is categorized into positive sentiment and negative sentiment. Data analysis in this study uses the Google Collaboratory platform. The following are the stages of analysis applied in this research:

1. Data scraping  
This research began with the collection of Indonesian tweet data on Twitter/X using the keyword “E-Meterai” from August 20 to September 17, 2024.
2. Data Pre-Processing  
The data pre-processing stage is carried out through several steps, including case folding, data cleaning such as removing numbers, punctuation marks, and emoticons, as well as the word normalization process.
3. Data Labeling  
Manually labeling the data in two classes namely positive and negative classes.
4. Feature Selection  
Feature selection, which implements stemming, stopword removal, and tokenizing techniques, aims to identify optimal features and improve model performance.
5. TF-IDF Weighting  
This research applies a word weighting method that combines two components, namely Term Frequency (TF) and Inverse Document Frequency (IDF).
6. Data Splitting  
Data is split into training data and test data with a ratio of 90:10.
7. Hyperparameter Tuning

### IV. RESULT AND DISCUSSION

This research dataset consists of 1,249 Indonesian tweets collected through crawling using Python on Google Collaboratory. The initial data of 1,300 tweets, with the keyword “E-Materai CPNS”, was collected during the period August 20 - September 17, 2024. The data cleaning process resulted in 1,249 tweets ready for analysis, after removing duplicates, irrelevant tweets, and incomplete tweets. The next step is to pre-process the data with the aim of transforming the unstructured raw data into a more structured form and ready for analysis.

**Table III. Pre-processing data**

Proses	Sebelum	Sesudah
<i>Case Folding</i>	Susah bgt mau daftar CPNS pakek e-materai!!!	susah bgt mau daftar cpns pakek e-materai!!!
<i>Cleaning</i>	susah bgt mau daftar cpns pakek e-materai!!!	susah bgt mau daftar cpns pakek ematerai
Normalisasi	susah bgt mau daftar cpns pakek ematerai	susah banget mau daftar cpns pakek ematerai

Tweets were manually classified into positive and negative sentiments. Of the 1,249 tweets related to the e-meter policy in CPNS registration, 882 tweets (70.62%) had negative sentiments and 367 tweets (29.38%) had positive sentiments. This result shows that most people expressed dissatisfaction with the policy, as shown in the histogram in Figure 2.

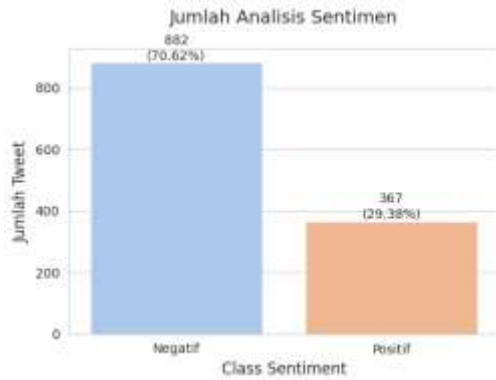


Figure 2. Histogram of Manual Data Labeling

Reducing the dimension of the document vector space is done through a feature selection process that includes tokenization, stopwords removal, and stemming. The goal is to produce a more efficient and effective document representation for further analysis.

The stopwords removal process, using 1171 words, resulted in a reduction in the number of terms in the document.

The stemming process was used to simplify words by removing affixes. The Python literary library helps to convert words to their base form. An example of stemming application on review data is shown in Table 4.

Table IV. Result of stemming process

Sampel ke-	Sebelum Stemming	Setelah Stemming
1	rusuh cari rebutan materai cpns	rusuh cari rebut materai cpns
2	kemudahan pendaftaran cpns materai aktif materai	mudah daftar cpns materai fisik materai aktif materai

Furthermore, tokenization is the process of cutting documents into separate words (tokens), where spaces are used as delimiters between words.

The results of data preprocessing and feature selection show that there are 2463 words that appear in the entire review dataset. Table 5 shows the results of the TF value calculation.

Table V. Number of Frequency for each Term

Review To-	adu h	baik	pilih	susa h	...	zon k	Number of terms
1	0	1	1	0	...	0	14
2	1	0	0	1	...	0	14
3	0	0	0	1	...	0	25
4	0	0	0	0	...	0	18
5	0	0	1	0	...	0	27
...	...	...	...	...	...	...	...
1249	0	0	0	0	...	0	16

Number of Reviews	31	10	12	59	1
-------------------	----	----	----	----	---

An example of manual calculation of the TF-IDF method is as follows:

$$W_{baik,doc1} = \frac{n_{i,j}}{\sum_k n_{k,j}} \times \ln\left(\frac{N}{df(i)}\right) + 1$$

$$W_{baik,doc1} = \frac{1}{14} \times \left( \left( \log_e \frac{1249}{10} \right) + 1 \right) = 0,4162509584$$

Table VI. TF-IDF Result

Review To-	aduh	baik	pilih	susah	...	zonk
1	0	0,4162	0,4032	0	...	0
2	0,3554	0	0	0,2894	...	0
3	0	0	0	0,1621	...	0
...	...	...	...	...	...	...
1249	0	0	0	0	...	0

The dataset is divided into training data and test data with a ratio of 90%:10% in the process of building and evaluating sentiment classification models.

Table VII. Division of Training Data and Testing Data

Classification	Positive	Negative	Total
Training Data	330	794	1124
Testing Data	37	88	125
<b>Total</b>	<b>367</b>	<b>882</b>	<b>1249</b>

The optimal parameter selection for the SVM model is done through a grid search by evaluating various combinations of Cost (C) and Gamma (γ) values. The range of values for these parameters was determined based on references from the journals Fide [15] and Aliyya [16].

Table VIII. Parameter Support Vector Machine (SVM)

Parameter	Nilai Parameter
Cost (C)	0,01; 0,1; 1; 10; 100; 1000
Gamma (γ)	0,01; 0,1; 1, 10, 100 'auto'

The 'auto' value used in gamma is the default value by calculation:

$$\gamma = \frac{1}{n_{col}} = \frac{1}{2463} = 0,0004060089$$

The hyperparameter tuning process produces the best model with an average accuracy of 0.86923 based on 5-fold cross validation, with parameters C = 100 and Gamma = 0.01 obtained through the hyperparameter tuning process using GridSearchCV.

SVM identifies the support vector by performing a dot product calculation on data that has been transformed into a higher dimensional space, calculated with the RBF kernel.

# “Support Vector Machine Implementation for Classifying Public Sentiment on Electronic Stamps use in Civil Servant Registration”

Data is not processed individually, but in pairs. Here is an example of RBF kernel calculation for the first review with itself.

$$K(x_i x_j) = \exp(-\gamma \|x_i - x_j\|^2) = \exp(-\gamma \sum_{j=1}^p (x_i - x_j)^2)$$

$$K(x_1 x_1) = \exp(-0,01((0 - 0)^2 + \dots + (0,4162 - 0,4162)^2 + (0 - 0)^2 + \dots + (0,4032 - 0,4032)^2 + \dots (0 - 0)^2)$$

$$K(x_1 x_1) = 1$$

Each training data was processed uniformly, resulting in a Kernel matrix of 1124x1124. Optimization of  $\alpha$  and  $b$  parameters was done using quadratic programming with the help of Scikit-learn Python due to the large dataset. Resulted in 605 support vectors and an optimal bias of -4. 00586398. SVM classification function with RBF kernel is as follows:

$$f(\phi(x)) = \text{sign}(\sum_{i=1}^{ns} \alpha_i y_i K(x_i, x_j) - 4.00586398)$$

With  $K(x_i x_j) = \exp(-0,01 \|x_i - x_j\|^2)$ ,  $ns$  indicates the amount of data that acts as a support vector, where  $x_i$  represents the support vector and  $x_j$  is the test data to be classified.

The test data prediction process is performed by calculating the dot product between the training data and test data using the Radial Basis Function (RBF) kernel function. Specifically, each test data is compared with all the training data through the kernel function. The kernel function for the first review training data and the 1125 review test data is as follows.

$$K(x_1 x_{1125}) = \exp(-0,01((0 - 0)^2 + \dots + (0 - 0)^2 + (0 - 0)^2)$$

$$K(x_1 x_{1125}) = 0,9601$$

The calculation is carried out so that it gets the kernel value of the 1124 training data and the 1249 test data. If  $f(\phi(x))$  is positive then the data will be labeled as class +1 or positive class and if it is negative then it will be labeled as class -1 or negative class. The prediction for the 1125 review test data is as follows.

$$f(\phi(x)) = \text{sign}(\sum_{i=1}^{ns} \alpha_i y_i K(x_i, x_j) - 4.00586398)$$

$$f(\phi(x)) = \text{sign}(\sum_{i=1}^{605} \alpha_i y_i K(x_{training}, x_{1125}) - 4.00586398)$$

$$f(\phi(x)) = \text{sign}([13,807 \times (-1) \times 0,9601] + \dots [7,975 \times (1) \times 0,9409] - 4.00586398)$$

$$f(\phi(x)) = \text{sign}(-0,4743) = -1$$

Based on the above calculations, the 1125 review test data is included in the negative class sentiment. The classification performance results of the SVM model can be seen as follows:

**Table IX. Confusion Matrix of SVM Classification Results**

Actual Train	Predicted Train		Actual Test	Predicted Test	
	Positive	Negative		Positive	Negative
Positive	328	2	Positive	32	5
Negative	0	794	Negative	5	83

After getting the confusion matrix, we can calculate the accuracy, recall, precision, f1-score, and kappa statistic values. Here are the results of the model performance:

**Table X. SVM Classification Performance Results**

Evaluasi Model	Proporsi Data (90%;10%)	
	Data Latih	Data Uji
Accuracy	99,82%	92%
Recall	99,39%	86,48%
Precision	100%	86,48%
F1-Score	99,69%	86,48%
Kappa-Statistic	99,57%	81%

To show the words that appear most often in positive and negative sentiments, word cloud visualization is used. The following are the results of the wordcloud based on public opinion on the use of e-stamps in CPNS registration.



**Figure 3. Word Cloud on Positive and Negative Reviews**

The word cloud of positive reviews shows words such as “cepat”, “langsung”, “Privy”, and “lancar”, reflecting that e-stamps are perceived as efficient and easy. The dominant word “Privy” indicates satisfaction with the service, supported by other words such as “mudah”, “legal”, and “resmi” that emphasize the practicality and validity of the document. Conversely, negative reviews are dominated by the words “gagal”, “error”, “susah”, and “habis”, indicating technical problems, limited availability of e-stamps. The words “sistem”, “jaring” and “dokumen” reinforce complaints about system glitches and difficulties in the document upload process.

## V. CONCLUSION

After analyzing and discussing, the following conclusions are related to the classification of public sentiment towards the use of e-stamps in CPNS registration. After manually labeling the classes, the positive class is obtained as many as 367 tweets and the negative class is 882 tweets. This shows that more people have negative opinions about government policies on the use of e-stamps in the registration of Civil Servant Candidates (CPNS).

The public sentiment classification model in this study was developed using the Support Vector Machine (SVM) algorithm with a Radial Basis Function (RBF) kernel, as well as setting the Cost (C) parameter of 100 and Gamma ( $\gamma$ ) of 0.01. With the proportion of training data and test data of

## “Support Vector Machine Implementation for Classifying Public Sentiment on Electronic Stamps use in Civil Servant Registration”

90%:10%, this model achieved an accuracy rate of 92%. Other evaluation metrics are recall of 86.48%, precision of 86.48%, f1-score of 86.48%, and kappa statistic of 81%. Based on the word cloud visualization of positive and negative reviews, it can be concluded that the implementation of e-stamps in CPNS registration received mixed responses from the public. Positive reviews highlight the ease and speed of the digitization process, while negative reviews indicate that there are still technical obstacles such as difficulties in purchasing or using e-stamps and system disruptions.

### REFERENCES

1. Aisha, E., & Bahalwan, H. 2024. Twitter: Representasi Dan Ekspresi Diri Tanpa Sekat. *Jurnal Visi Komunikasi*, Vol. 22, No. 02, pp. 226-233.
2. Fahlevvi, M. R. 2022. Analisis Sentimen Terhadap Ulasan Aplikasi Pejabat Pengelola Informasi Dan Dokumentasi Kementerian Dalam Negeri Republik Indonesia Di Google Playstore Menggunakan Metode Support Vector Machine. *Jurnal Teknologi Dan Komunikasi Pemerintahan*, Vol. 4, No.1, pp. 1–13.
3. Herwinsyah, A. Witanti. 2022. Analisis Sentimen Masyarakat Terhadap Vaksinasi Covid-19 Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (Svm). *Jurnal Sistem Informasi dan Informatika (Simika)*. Vol. 5, pp. 1.
4. Ratino, N. Hafidz, S. Anggreani, G. Wata. 2020. Sentimen Analisis Informasi Covid-19 menggunakan Support Vector Machine dan Naïve Bayes. *Jurnal Jupiter*. Vol. 12, No. 2, pp. 1-11.
5. Yusupa, A., & Tarigan, V. 2021. Perbandingan Algoritma Maching Learning dalam Analisis Sentimen Mobil Listrik di Indonesia pada Media Sosial Twitter/X. *Jurnal Informatika Polinema*. Vol. 10, No. 4, pp. 479-490.
6. Runimeirati, Muis, A., & Muhammad, F. 2023. Pelatihan Text Mining Menggunakan Bahasa Pemrograman Python. *Abdimas Langkanae*, Vol. 3, No. 1, pp. 36–46.
7. Feldman, R., & Sanger, J. 2007. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York: Cambridge University Press.
8. Syafa'aturrohman, A., Nurdiawan, O., Basysyar, F. M., & Sulaeman, M. 2024. Naive Bayes Meningkatkan Model Analisis Sentimen Pada Ulasan Aplikasi DANA Di Playstore Indonesia. Vol. 9, No. 2, pp. 171–180.
9. Rohim, A., Haviz Irfani, M., Ramadhan, M., & Ubaidillah, U. 2023. Penerapan Metode Text Mining dengan Chatbot Questions And Answer pada PT PLN (Persero) Sumatera Selatan. *Klik - Jurnal Ilmu Komputer*, Vol. 4, No. 2, pp. 59–67.
10. Nurjanah, W. E., Setya Perdana, R., & Fauzi, M. A. 2017. Analisis sentimen terhadap tayangan televisi berdasarkan opini masyarakat pada media sosial twitter menggunakan metode k-kearest neighbor dan pembobotan jumlah retweet. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, Vol. 1, No. 12, pp. 1750–1757.
11. Wang, L. 2005. Support Vector Machines: Theory and Applications. In *Studies in Fuzziness and Soft Computing*.
12. Nugroho, A. S., Witarto, A. B., & Handoko, D. 2003. Support Vector Machine. *Proceedings of the 2011 Chinese Control and Decision Conference, CCDC 2011*, pp. 842–847.
13. Cortes, C., & Vapnik, V. 1995. Support-Vector Networks CORINNA. *Journal of Physics: Conference Series*, pp. 273–297.
14. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., & Thirion, B. 2011. *Scikit-learn: Machine learning in Python*. *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830
15. Fide, S., Suparti, S., & Sudarno, S. 2021. Analisis Sentimen Ulasan Aplikasi Tiktok Di Google Play Menggunakan Metode Support Vector Machine (Svm) Dan Asosiasi. *Jurnal Gaussian*, Vol.10, No.3, pp. 346–358.
16. Aliyya, S. 2020. Analisis Sentimen Berbasis Aspek pada Ulasan Aplikasi Tokopedia Menggunakan Support Vector Machine. *Skripsi. Program Studi Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah*.