IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.14, No.2, April 2020, pp. 113~122
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: 10.22146/ijccs.55761

Traffic Density Classification Using Twitter Data and GPS Based On Android Application

Mohammad Afrizal*[1], Idham Ananta Timur[2]
[1]Master Program of Computer Science; FMIPA UGM, Yogyakarta, Indonesia
[2]Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
e-mail: ***[1]mohammad.afrizal@mail.ugm.ac.id** , [2]idham@ugm.ac.id

Abstrak

Meningkatnya jumlah kendaraan di DIY menyebabkan kemacetan terjadi di berbagai titik lalu lintas di DIY. Solusi untuk mengurangi kemacetan tersebut adalah dengan peningkatan penggunaan transportasi umum darat dalam kota, namun masih belum diminati oleh masyarakat. Untuk mengoptimalkan kegiatan sehari-hari, masyarakat selalu berusaha untuk menghindari kepadatan lalu lintas pada Street yang akan dilewati.

Beberapa penelitian tentang layanan sosial media telah digunakan untuk medeteksi anomali kepadatan lalu lintas.. Namun sistem tersebut masih belum dapat memberikan informasi kepadatan lalu lintas pada Street yang akan dilewati oleh pengguna karena hanya berupa pemetaan. Berdasarkan hal tersebut, penelitian ini bertujuan untuk melakukan klasifikasi tingkat kepadatan lalu lintas pada Street yang akan dilewati pengguna di Daerah Istimewa Yogyakarta menjadi kategori macet dan tidak macet dengan memanfaatkan data Twitter dan GPS.

Hasil penelitian menunjukkan bahwa Aplikasi Android mampu mengklasifikasi kepadatan lalu lintas pada Street yang akan dilalui menggunakan Geonames.org API. Dengan menggunakan algoritma klasifikasi naïve bayes, sistem dapat mengklasifikasi kepadatan lalu lintas pada 14 Street dengan besar rata-rata akurasi 77,5%, presisi 90%, recall 79,1%, dan f-score 82,8%.

Kata kunci*— Kepadatan lalu lintas, Android, Naïve bayes, klasifikasi*

Abstract

Increasing the number of vehicles in Special Region of Yogyakarta caused by congestion occurred at various traffic points in Special Region of Yogyakarta. The solution to reducing congestion is by increasing the use of public transportation within the city, but it still not in demand by the public. Optimizing daily activities, community always tries to avoid the traffic density on the road to be bypassed.

Some research on social media has been used to detect traffic density anomalies. However, the system still cannot provide traffic density information on roads that will be passed by the user because it is just a mapping. Based on this problem, this study aims to classify the traffic density on the road that will be passed by users in the Special Region of Yogyakarta into the category of high traffic and low traffic by utilizing Twitter and GPS data.

The results show that Android Applications are able to classify traffic density on the road to be traversed using Geonames.org API. Using the naïve bayes classification algorithm, the system can classify traffic density on 14 streets with an average accuracy of 77.5%, 90% precision, 79.1% recall, and 82.8% f-score.

Keywords—*Traffic density, Android, Naïve bayes, classification*

Received April 9th,2020; Revised April 16th, 2020; Accepted April 29th, 2020

1. INTRODUCTION

The increasing number of population in DIY resulted in an increase in the number of motor vehicles in the DIY area by 10.06% in 2005 to 2014 mentioned by the Central Statistics Agency of Yogyakarta Special Province [1]. Increasing the number of vehicles in DIY caused congestion occurred at various traffic points in DIY. The solution to reduce congestion is by increasing the use of public land transportation within the city, but still not sought by the public [2]. To optimize daily activities, the community always tries to avoid traffic congestion on the road to be passed [3].

One alternative sensor that can replace the camera sensor is a social sensor [4]. Today's social sensors are widely used to solve problems [5]. Social sensors derived from computing social networking data that generally contain a variety of information related to real life [6]. Twitter is one of the social media that is widely used by the people of Indonesia [7]. Based on research at the Social Media Research Institute in France, the number of Twitter users in Indonesia is ranked fifth in the world with a total of 29.4 million accounts in 2012 [8].

Several studies on social media have been used to detect traffic density anomalies [9]. A research shows if Twitter can be used to map the density of traffic density in Yogyakarta [1]. But the system still could not provide traffic information on the road that will be passed by the user because it is only a mapping.

GPS or Global Positioning System is a growing famous to be used on smartphones. Most smartphones in the market are equipped with GPS [10]. GPS could be utilized to determine position of the rider real time and also to know the way that will be passed by the rider. Based on these problems, this research aims to classify the level of traffic density on the road that will pass the rider in Yogyakarta into category of high traffic and low traffic by using data Twitter and GPS.

2. METHODS

This research uses GPS that can be utilized to gain access to location information in real time. The location in real time from GPS can be used to determine the path to be traversed by using the Open Street Maps API. The Naïve Bayes algorithm is trained to produce trained classification models on Twitter data that have been collected using the Twitter REST API [11]. Trained classification models are integrated into Android apps using cloud computing to provide a classification of the traffic density on the road the user will pass. The path to go through, time, and day to enter the system.

2.*1 Collecting Data from Twitter*

The data collection process in the study was conducted for 4 months on 12 November 2017 until February 10, 2018. Data collected from the official and regular account Twitter users in the region of Yogyakarta Special Region (DIY). The referenced official account is @ATCS_DIY and uses keywords to search on a regular account restricted to the DIY region radius.

The process of collecting data using Twitter REST API with tweepy libraries in the Python programming language. Twitter REST API is used to get information from official accounts and keywords from regular accounts. Twitter REST API identifies applications and users using the OAuth protocol and returns a response of JSON (Javascript Object Notation). JSON data is then stored in Comma Separated Version (.csv) form. The data retrieval process using the Twitter REST API is addressed by Figure 1.

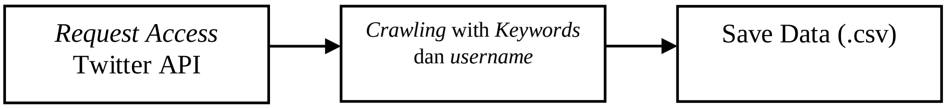
Request Access Twitter API
Crawling with *Keywords* dan *username*
Save Data (.csv)

Figure 1 Twitter REST API block system

From the results of data collection, data obtained as many as 3231 tweets containing traffic information or not. The data have been obtained not all used in the next process. The data used in the study only Twitter data that contains traffic information, so it needs to do the processing to eliminate tweets that are not important.

2. 2 Classification System

The design of a classification system is a unified program that processes from the initial processing of Twitter data, feature selection, and the last-class prediction process stalled or not jammed using the Naïve Bayes algorithm.

2.2.1 Pre-processing

The initial processing is done to process the data to be used in the next process. The data collected as many as 4246 tweets consisting of tweets related to traffic or not. In this process, tweets are unrelated to traffic, grouping data based on the frequency of occurrence tweet for 4 months, and labeling. The following steps are performed on the initial processing.

The process of data elemination is done to eliminate tweet data unrelated to traffic. In the process of data collection, 4246 tweets were collected containing traffic information and not. Then data tweets that do not contain traffic information are eliminated from research data. The process of elemination is done manually and generated as many as 3231 data containing traffic information. The data will be used in subsequent processing. Tweets are separated using a combination of sentiments that are mentioned manually. Tweets that do not contain traffic information are separated and are not used as research data. Table 1 shows example for data that contain traffic information and Table 2 shows data that not contain traffic information.

Table 1 Tweet that contains traffic information (tweet in Bahasa)

Date	Tweet
2017-12-12 14:00:53	Arus lalin malam ini di simpang 4 tempel sleman cukup lancar... https://t.co/1PHHkdErB4
2018-02-09 14:19:21	Kondisi Lalin di simpang Giwangan Terpantau Lancar https://t.co/ZQLogY5Fvs

Table 2 Tweet that not contains traffic information (tweet in Bahasa)

Date	Tweet
2018-01-18 14:09:51	Perbatasan Prambanan _ Klaten saat ini arus lancar, cuaca hujan. Mohon pengguna jalan hati2 jangan ngebut.
2017-12-31 12:40:46	Srikandi Ditlantas DIY bersama personel Satlantas Sleman bertugas di simpang Jombor yang… https://t.co/1m2N1fI8Zq

The next stage is the process of selecting data that will be used in the training process data based on the frequency of occurrence of data. In the study used the lower limit of 50 occurrences of data for four months on each street name. This process is intended to reduce noise caused by the path that is not too often mentioned on tweet. Determination of the lower limit of 50 is used because tweets with an apparent frequency above 50 are considered to have a probability of occurrence on a weekly basis. So that data can be used as a reference pattern prediction.

The training process required class labels on each data as classification targets. In this process labeling of each data obtained into the category of traffic jams and not jammed. Labeling process is done manually by paying attention to tweet contents on each data. Tweets containing "crowded", "stuck", and "crowded" sentiments will be categorized into jam classes. And tweets with "fluent" and "crowded" sentiments will be categorized into non-jam categories. Table 3 shows roads list.

Table 3 Roads List

No	Street Name	Amount
1	Jl. Wiyoro Kidul, Baturetno, Banguntapan, Bantul, Daerah Istimewa Yogyakarta 55197	61
2	Jl. Colombo No.1, Caturtunggal, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281	85
3	Jl. Jend. Sudirman, Gowongan, Jetis, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55233	236
4	Jl. Selokan Mataram, Kabupaten Sleman, Daerah Istimewa Yogyakarta	72
5	Jl. Prambanan, Kabupaten Sleman, Daerah Istimewa Yogyakarta	117
6	Jl. Pogung Raya, Sinduadi, Mlati, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55284	58
7	Jl. Magelang No.80, Kricak, Tegalrejo, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55242	53
8	Jl. Raya Piyungan, Daerah Istimewa Yogyakarta	79
9	Jl. Ps. Kembang, Sosromenduran, Gedong Tengen, Kota Yogyakarta, Daerah Istimewa Yogyakarta	95
10	Jl. Mayor Suryotomo No.31, Ngupasan, Gondomanan, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55122	53
11	Jl. Maguwo, Banguntapan, Bantul, Daerah Istimewa Yogyakarta	192
12	Jl. Imogiri Timur, Giwangan, Umbulharjo, Kota Yogyakarta, Daerah Istimewa Yogyakarta	85
13	Jl. Demangan, Wedomartani, Ngemplak, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55584	69
14	Jl. Adi Sucipto, Jawa Tengah	125
	Total	1380

2.2.2 Features Selection

In obtaining the best results in the classification, appropriate characteristics are required to be processed during the data training phase. The feature selection process is performed with the aim of determining features suitable for use in the training process. In the case of traffic density classification using Twitter data, the day, time, and path features are the most suitable features to be used according to the amount of information listed on the data obtained in the data collection process. The collected Twitter data contains detailed information of the time and date of the tweet created as well as the name of the path mentioned on the tweet content.

By using the time feature, the system can classify traffic density levels on selected roads that are grouped into weekday and weekend categories. Monday until Friday is grouped into the category of weekday and Saturday and week are grouped into weekend category. For the time feature is distinguished to "morning" for the time span of 06:00 - 10:00, "noon" time range 10.00 - 15.00, "afternoon" 15.00-18.00, and "night" for 18.00-24.00 (all at WIB). Figure 5 shows features selection result.

Table 4 Features Selection Result

Wiyoro Street/weekend		
	Time	Condition
0	Morning	tidak macet
1	Morning	macet
2	Evening	tidak macet
3	Evening	tidak macet
4	Night	tidak macet
5	Morning	tidak macet

Wiyoro Street/weekday		
	Time	Condition
0	Night	tidak macet
1	Evening	tidak macet
2	Evening	tidak macet
3	Afternoon	tidak macet
4	Morning	tidak macet
5	Night	tidak macet

2.2.3 Training Data

Training the classification model using the Naïve bayes algorithm to determine the class prediction is jammed or not jammed. The Naïve Bayes algorithm serves to determine the probability weights of each feature in the training data by calculating the probability of the prior and posterior probabilities. The Naïve Bayes classification model yields ten models, the probability model of independent jamming, independent non-stop events, non-performing events and non-jam time events occurring in the morning, afternoon, afternoon and evening.

2. 3 Cloud Computing Service

The resulting trained model is used as a classifier stored in the cloud in the form of a function. The type of cloud computing used in the research is Function as Service (FaaS). FaaS is one type of cloud computing service that provides a platform for the development and management of applications using functions that run on cloud service providers. FaaS uses a serverless architecture where the operation and management of the application without the server to run a line of code. The reason for using cloud computing is by using cloud technology, easy application management, such as modification of functions, modification of the classification model, and without having to insert all logic programs on each user's application one by one because the system is centralized.

This research uses Firebase Cloud Function to be used as cloud function service. Firebase Cloud Function is one of the services provided by Google for cloud computing needs. Figure 2 shows the cloud function service architecture used.

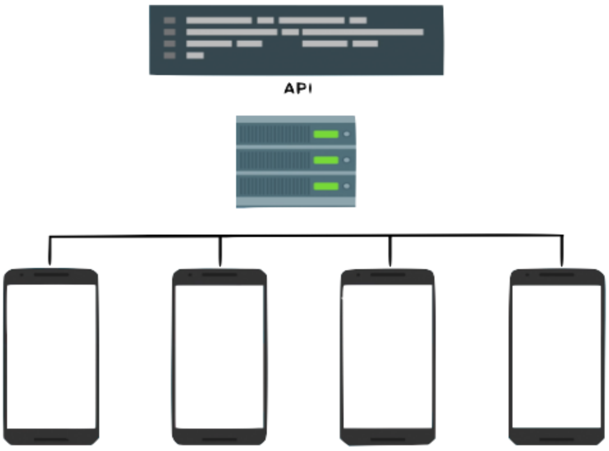

Figure 2 *Function as Service Architecture*

Figure 2 shows the cloud function architecture of the system to be used. Trained models will be stored on the cloud to form functions. Cloud function can be accessed by any kind of platform, in this research platform used is Android. Android applications request on cloud function in the form Http request with API that has been made.

2. 4 Android Application

Android applications in the study were made using Android Studio as the IDE (Integrated Development Environment). To build the application used Java programming language to program the activity, and xml to build user interaction view. Android application display design is divided into two activities, namely the main activity and activity maps. The main activity is used as the start menu of the app that contains the app name and navigation key information to the maps activity. Map activity is used as a GPS data acquisition and time. On the activities of maps conducted communication using REST to Geonames webservice to get the closest intersection to be passed by enter the longitude and latitude of the GPS. In this activity the processing is done on the given input. Processed entries are latitude and longitude, time, and day in real time. The result of the processing is the decision of calling the name of the street during weekday or weekend that occurs during the morning, afternoon, afternoon, or night. Then do a function call to the cloud function using Http communication to get change of prediction jam or not jammed. Prediction results are displayed on the maps activity.

3. RESULTS AND DISCUSSION

Twitter data collection using Twitter REST API generate 1380 data containing traffic information after pre-processing. The data is taken for 4 months from the official account and some keywords are used. 1380 data collected, the data contain sentiment to 14 roads in DIY. 14 collected street names were obtained from the elemination and thresholding results of the pre-processing process. Of the total tweet data obtained, only the street name with the frequency of occurrence over 50 tweets for 4 months used in the research process. Then generate 14 names of selected paths. It indicates that twitter data is still too small when compared with the total number of roads in DIY that is a number of 459 roads. Twitter social networking data is only able to cover 3% of the total road in DIY in the last 4 months with a minimum of 50 tweets appear on each road. Then the data obtained graphs to see the level of traffic density on each road. Based on the results of data processing, the graph of traffic density level in DIY based on Twitter data which refers to 14 street names.

The process of training and classification of traffic density in this study using the Naïve Bayes algorithm. The data used in the research amounts to 1380 previously processed twitter data from 14 different street locations. The process of classification model accuracy testing using ten-folds validation method, where the data is divided into 10 sections to test the success of the system in classification. Table 5 shows the amount of data from each path.

Table 5 Data from Each Path

No	Street	Amount of Data	
		Weekday	Weekend
1	Wiyoro Kidul Street	43	18
2	Colombo Street	59	26
3	Jend. Sudirman Street	178	58
4	Selokan Mataram Street	54	18
5	Prambanan Street	90	27
6	Pogung Raya Street	48	10
7	Malioboro No. 52 Street	38	15
8	Raya Piyungan Street	60	19
9	Ps. Kembang Street	70	25
10	Mayor Suryotomo No. 31 Street	38	15
11	Maguwo, Banguntapan Street	139	53
12	Imogiri Timur Street	60	25
13	Demangan Street	55	14
14	Adi Sucipto Street	90	35

The data is divided into two parts: training data and test data. The data is cut into 10 sections, the first part is used in the test data, and the remainder as training data. Then the 2nd data into test data, and the rest into training data. The process is done until the 10th data and the calculated average accuracy. Some data cannot be processed using ten folds validation because of the small amount of data so it cannot be divided into 10 parts, such as Pogung street during weekend and street Demangan at the weekend. Table 6 describes the evaluation results using ten folds validation on each path.

Table 6 Evaluation Result

Ten Folds-Validation		Accuracy	Precision	*Recall*	F-*Score*
Wiyoro Street	Weekend	90,0	95,0	91,6	93,0
	Weekday	82,5	100,0	82,5	90,1
Colombo Street	Weekend	77,5	90,0	81,6	83,0
	Weekday	72,9	95,0	73,2	82,2
Sudirman Street	Weekend	68,9	73,3	60,0	65,1
	Weekday	69,1	74,4	73,2	73,0
Sekolan Street	Weekend	86,7	100,0	86,6	92,0
	Weekday	70,9	82,5	78,0	78,3
Prambanan Street	Weekend	90,0	93,3	92,5	92,6
	Weekday	78,9	100,0	78,0	88,2
Pogung Street	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
	Weekday	73,2	95,0	76,1	84,3
Malioboro Street	Weekend	65,0	90,0	65,0	73,3
	Weekday	71,0	100,0	71,0	83,0
Piyungan Street	Weekend	90,0	95,0	91,7	93,0
	Weekday	92,8	100,0	92,8	96,1
Ps. Kembang Street	Weekend	86,7	90,0	93,3	89,3
	Weekday	76,4	80,9	84,0	81,8
Mayor Suryotomo No. 31 Street	Weekend	88,3	100,0	88,3	92,7
	Weekday	74,0	76,7	85,0	78,4
Maguwo Street	Weekend	69,3	70,0	71,5	67,6
	Weekday	68,3	81,1	64,7	71,3
Giwangan Street	Weekend	94,1	96,7	97,5	96,5
	Weekday	78,8	96,0	80,6	87,3
Demangan Street	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]
	Weekday	63,3	100,0	63,3	77,5
Adi Sucipto Street	Weekend	75,8	80,0	73,3	72,7
	Weekday	61,1	86,0	61,1	71,0
Average		77,5	90,0	79,1	82,8

The results of the evaluation shown in Table 2 show the accuracy, precision, recall, and f-score on each road data. These values are used to measure the performance of a system. The accuracy, precision, recall, and f-score values above are obtained from the average confidence matrix calculation results for each iteration. The value of accuracy in question is the level of proximity between the predicted results with the actual value. The evaluation results show that the accuracy level is not influenced by the amount of data portion used. The average accuracy of 77.5 percent of the total 28 road conditions. The average of precision result is high enough 90%, and the mean of recall and f-measure are 79,1 and 82,8 respectively.

According to user evaluation, it shows that of the five respondents, the application can run in accordance with the expected function that can provide the user's current path information, indicate the path to be passed along with the prediction of jams and not jammed. Some obstacles faced by the user at the time of using the application, the interface design that is considered less user friendly, notifications late emerged, notifications that are still running even after the predicted way, predictions that are not in accordance with the assumption of users, as well as several times force close on a particular device.

Interface design is considered less user friendly due to the application development, researchers have not used the correct user centered design method to design the interface. Respondents assume the distance between the user's position and the road to be traversed needs to be displayed. But the design of the interface has not been based on survey and user needs analysis. The design of the interface is based solely on the needs of the feature, which only shows the current user path, displays the path to be traversed, and the predicted result.

Notifications on late applications make users confused while using the app. This happens because the calling function of the cloud function is put on MapReady function. The function of the cloud function will always be called when the application is accessing the Map, resulting in too frequent function calls even when not being queued. In addition, the notifications that are still running when the user has passed the predicted path to the problem that arises when the application usage. The problem occurs because the Geonames.org API provides output intersections based on the distance closest to the current user's position. So that when the user has passed the path that has been issued Geonames.org API with a distance that is not too far away, then the Geonames.org API will still define the road to be the closest road that will pass the user until the user approaches another road with a closer distance. The problem has not been resolved by current researchers.

Prediction that is not in accordance with the assumption of the user becomes another problem that cannot be overcome the researcher. Adjusting the sentiments of traffic conditions on Twitter data with the assumption of each individual user is a difficulty encountered in this study. Differences of assumptions arise because of different perspectives jammed and not jammed between each user. Another respondent assumes that a road is said to be jammed if it cannot pass a red light more than five times the lamp repetition, whereas according to respondent Ahmad Zuhair assume that a road is said to be jammed if along the road can only move with a relatively slow speed.

This paper open to further research of mobile application to predict traffic density, because there is so many factors that can affect the value, such as the algorithm and the data that used for training.

4. CONCLUTIONS

The results show that Android Applications are able to classify traffic density on the road to be traversed using Geonames.org API. Using the naïve bayes classification algorithm, the system can classify traffic density on 14 streets with an average accuracy of 77.5%, 90% precision, 79.1% recall, and 82.8% f-score.

5. FUTURE WORKS

There are many problems to solve in future. The delay of application while transmitting and receiving data to and from back end service (geonames.org API and Firebase Cloud Function). Another works are enhance the classification system to increase accuracy and finding new method to get more dataset from Twitter.

ACKNOWLEDGEMENTS

The authors would like to thank Universitas Gadjah Mada for providing the platform which allow authors to complete this research successfully.

REFERENCES

[1] D. A. Kurniawan, S. Wibirama, and N. A. Setiawan, “Real-time traffic classification with Twitter data mining,” *2016 8th Int. Conf. Inf. Technol. Electr. Eng.*, pp. 1–5, 2016.

[2] K. R. Pandhare and M. A. Shah, “Real time road traffic event detection using Twitter and spark,” *Proc. 2017 Int. Conf. Inven. Commun. Comput. Technol.*, no. Icicct, pp. 445–449, 2017.

[3] B. Yang, W. Guo, B. Chen, G. Yang, and J. Zhang, “Estimating Mobile Traffic Demand Using Twitter,” *IEEE Wirel. Commun. Lett.*, vol. 5, no. 4, pp. 380–383, 2016.

[4] R. Y. K. Lau, “Toward a social sensor based framework for intelligent transportation,” *2017 IEEE 18th Int. Symp. A World Wireless, Mob. Multimed. Networks*, pp. 1–6, 2017.

[5] Y. Chen, Y. Lv, X. Wang, and F.-Y. Wang, “A convolutional neural network for traffic information sensing from social media text,” *2017 IEEE 20th Int. Conf. Intell. Transp. Syst.*, pp. 1–6, 2017.

[6] I. Salas, A., Georgakis, P., Nwagboso, C., Ammari, A. and Petalas, “Traffic Event Detection Framework Using Social Media,” *IEEE Int. Conf. Smart Grid Smart Cities*, no. July, p. 5, 2017.

[7] I. P. Windasari, F. N. Uzzi, and K. I. Satoto, “Sentiment Analysis on Twitter Posts : An analysis of Positive or Negative Opinion on GoJek,” pp. 266–269, 2017.

[8] N. R. Fatahillah, P. Suryati, and C. Haryawan, “Implementation of Naive Bayes classifier algorithm on social media (Twitter) to the teaching of Indonesian hate speech,” *2017 Int. Conf. Sustain. Inf. Eng. Technol.*, pp. 128–131, 2017.

[9] B. Akilesh, N. Kumar, B. Reddy, and M. Singh, “TRAFAN : Road Traffic Analysis Using Social Media Web Pages,” pp. 655–659.

[10] I. Hanifah and B. N. Prastowo, “Uji GPS Tracking Dalam Skala Transportasi Antar Kota,” vol. 6, no. 2, pp. 175–186, 2016.

[11] S. Rodiyansyah and E. Winarko, “Klasifikasi Posting Twitter Kemacetan Lalu Lintas Kota Bandung Menggunakan Naive Bayesian Classification,” *Indones. J. Comput. Cybern. Syst.*, vol. 6, no. 1, pp. 91–100, 2012.