International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-12, October 2019

Line Bot Chat Filtering using Naïve Bayes Algorithm

Nathania Elvina, Andre Rusli, Seng Hansun

Abstract*: *Instant messaging has changed and simplified the way people communicate, whether in professional or personal life. Most communication is done through instant messaging, and it is common for people to miss important information. This is due to the huge amount of incoming message notifications, so users tend to accidentally ignore them. This is also experienced by Universitas Multimedia Nusantara (UMN) student committees who communicate via LINE instant messenger. This research showed LINE bot was made by using the Naive Bayes algorithm to classify between important messages and unimportant messages on the committee group. The Naive Bayes algorithm is a classification algorithm based on probability and statistical methods. The Naive Bayes algorithm is chosen because it is widely implemented in spam filtering; the method is simple and has good accuracy. The classification process is done by calculating the probability of chat in each class based on the value of the word likelihood which generated in the training process. This research produces spam precision and spam recall as 94.2% and 95.6% respectively.

Index Terms*: *Bot, Chat Filtering, Committee Group, Naïve Bayes, Organizational Interest.

I. INTRODUCTION

People always want to communicate better and faster. The use of smartphones has increased in recent years due to high portability and ease of access to the information it offers [1]. As many as 32% of the "List of 25 Top Applications Downloaded in Indonesia" are communication applications or known as instant messaging [2].

In practice, people communicate via instant messaging in large portions, resulting in people to miss important information. That is because of too many notification messages, especially in the group. In addition, not all incoming messages regard to the focus of the conversation from the corresponding group [1].

This problem is also experienced by the Universitas Multimedia Nusantara (UMN) student committees that use the LINE group to communicate. Based on a survey of five UMN Committees with a total of 148 respondents, 55.95% agreed that there is more information unrelated to the committee compared with related information. As much as 21.43% said that the ratio between the related and unrelated information is already balanced. Only 22.62% of respondents said that there is more information related to the committee compared to the unrelated information. As much as 71.49% of respondents often miss important information due to the huge amount of information that was less concerned with the focus of the group.

Research on filtering important chat titled “Analysis and Detection of Eventful Messages in Instant Messaging” has been done by Joshi et al. [1] and is implemented on WhatsApp Messenger. In that research, Alchemy API was used for natural language processing, which will determine the relevance of the message. However, WhatsApp Messenger does not provide public Application Programming Interface (API). Hence, automating data extraction is difficult [1]. Some other researches highlighted the importance of filtering methods for chat also had been done, as we can see in the works of Otsuka et al. [3] and Nguyen and Ricci [4]. Moreover, the same purpose also is achieved with the help of a bot system as can be seen in the works of Hirata et al. [5], Bala et al. [6], and Tepper et al. [7].

Naive Bayes algorithm is an algorithm that uses probability and statistics [8]. The Naive Bayes algorithm is implemented in many spam filtering methods due to its simple and good accuracy [9]. Viana et al. [10] had built a message classifier based on multinomial Naïve Bayes for online social contexts, while Yadav and Gupta [11] had used Naïve Bayes classifier to analyze user tweets for detection and prevention of self-harm tendencies of the Twitter user. Bashir et al. [12] had made automatic text summarization based on feature extraction using Naïve Bayes model, Sneha et al. [13] had made a smartphone-based emotion recognition and classification using Naïve Bayes classifier, and Wijaya and Santoso [14] tried to improve the accuracy of Naïve Bayes algorithm for hoax classification using particle swarm optimization. Based on visibility study and previous researches, research of Naive Bayes algorithm on LINE bots to filter chat based on the interests of the organization interest is made in this study.

II. TEXT AND CHAT CLASSIFICATION

Text classification is a process that classifies the documents based on the category labeled before. Text classification is an important part of text mining [15]. Text mining is the process of data mining of text from a document to search for words that are represented with the document so that it can be done an analysis of connectedness in the document. In text mining, there is a term which is the process of data preprocessing for generating numerical data from text data [16].

Revised Manuscript Received on October 30, 2019.
* Correspondence Author

Nathania Elvina*, Informatics Department, Universitas Multimedia Nusantara, Tangerang, Indonesia.
Andre Rusli, Informatics Department, Universitas Multimedia Nusantara, Tangerang, Indonesia.
Seng Hansun, Informatics Department, Universitas Multimedia Nusantara, Tangerang, Indonesia.

Text Preprocessing steps are tokenization, stopwords removal, tokenization, and stemming [15]. Before preprocessing is done, the entire document is converted to lowercase [17].

In addition to the conversion into lowercase, characters other than letters are omitted and considered as a delimiter [18]. On tokenization, the document is divided into a list of tokens. Next is stopword removal, i.e. words that often appear in the document but has no significance. Some examples of them are the conjunction and preposition words [19].

After stopword removal, the next step is stemming, i.e., converts words into its base form [15]. One of the Indonesian Language stemming algorithms is Nazief and Adriani algorithm [20]. It is a method developed based on Indonesian morphology rules which grouped suffixes into a prefix, suffix, infix, and confix [21].

The chat classification process can be divided into four different sections as below [22].

1. *Sessionalization*

Sessionalization is a process of merging a collection of chat messages within the same session. In sessionalization, Dong et al. [22] have a few rules. First, two chat sessions that occurred between the same participants with an interval of fewer than 10 minutes will be merged. Second, two chat messages with a time gap of more than 40 minutes will be divided into two sessions [22].

2. *Feature Extraction*

On feature extraction, icon and links from the web will be extracted and appended into the session [22].

3. *Feature Selection*

On feature selection, words are mapped with the indicative terms dictionary. In addition to the usage of acronym and polysemic, there are also writing errors and shortening of words in chat [22]. One way to tackle this problem is by using Symmetric Spelling (SymSpell) Algorithm [23].

4. *Topic Categorization*

Topic categorization is classifying chat session into one or more category [22].

III. NAÏVE BAYES ALGORITHM

Naive Bayes algorithm is an algorithm used to classify objects by using methods of probability and statistics. The concept of the Naive Bayes algorithm is to predict future probability based on experience [8]. This algorithm is widely used on the anti-spam filter with Paul Graham's approach [8]. One of Naïve Bayes models is the multinomial model. The multinomial model specifies that a document is represented by the set of word occurrences from the document [24]. The Naïve Bayes Algorithm takes two stages in the text classification process, which are the training stage and classification stage [25]. On the training stage, an analysis of the sample documents is done and followed by determining the probability of the word occurrence. The probability of each word occurred is calculated using the following formula [26].

$$P(W_t|C = k) = \frac{\sum_{i=1}^{N} x_{it}z_{ik}}{\sum_{s=1}^{|V|}\sum_{i=1}^{N} x_{is}z_{ik}} \qquad (1)$$

N is the total number of documents, and V is the number of words in the vocabulary. x_{it} is the frequency of word w_t in the document, computed for every word w_t in V. z_{ik} is an indicator variable which equals 1 when the document has class $C = k$ and equals 0 otherwise. $\sum_{i=1}^{N} x_{it}z_{ik}$ is the number of occurrences of word w_t in category k, whereas $\sum_{s=1}^{|V|}\sum_{i=1}^{N} x_{is}z_{ik}$ is the total occurrences of all vocabulary on category k [1]. The problem that occurs from the first equation is if there is one word that does not appear at all in sample document, resulting probability equals to 0 [26]. A simple way to resolve this, sometimes called *Laplace's law of succession*, is to add a count of one to each word type, then equation (1) may be replaced with equation (2) [26].

$$P_{Lap}(W_t|C_k) = \frac{1+\sum_{i=1}^{N} x_{it}z_{ik}}{|V|+\sum_{s=1}^{|V|}\sum_{i=1}^{N} x_{is}z_{ik}} \qquad (2)$$

The denominator in equation (2) is added by the number of vocabulary $|V|$ to ensure the probability is normalized after the numerator is added by 1 [26].

The second phase is the classification. The Naive Bayes classification formula is written in equation 3 [8].

$$y = \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}} P(C_k)\prod_{i=1}^{n} P(x_i|C_k) \qquad (3)$$

$P(C_k)$ is the probability of a sample document in class k. Whereas $\prod_{i=1}^{n} P(x_i|C_k)$ is the sum product of each word in class k. The result of $P(C_k)\prod_{i=1}^{n} P(x_i|C_k)$ will be compared to each class. Chat will be classified according to the class that has the highest probability [24].

Accuracy is measured based on precision and recall [27]. Precision is the classification accuracy for positive classes. A recall is a proportion of positive classes that are successfully detected as a positive class by the system [28]. Calculations are done using the confusion matrix as follows.

Table 1. Confusion Matrix

	True label A	True not A
Predicted label A	True Positive	False Positive
Predicted not A	False Negative	True Negative

In the context of spam detection, True Positive means that the document is truly spam. False Negative means the document was classified as ham but was spam in reality, and False Positive means the email was classified as spam but was the ham in reality [27]. By using confusion matrix, precision and recall can be defined as following [29].

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \qquad (4)$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \qquad (5)$$

IV. CHAT LANGUAGE

Chat language is very different from the conventional language in general. Its nature is informal [22], the same as on social media [30]. Users tend to use slang words rather than formal words [30]. Some of the characteristics of informal language according to Dong et al. [22] are the use of acronyms, abbreviations, polysemic, synonyms, and type errors. An acronym is formed through the extraction of the first letter of each word. For example, "ASAP" is the acronym for "As Soon As Possible".

On abbreviation usage, the word with same meaning can be written in various forms depending on each individual. For example is word “tomorrow,” can be written as “tmrw,” “tmorrow,” or others. There is also synonym, referring to words that have same meaning and often used interchangeably.

An example of a synonym is “network adapter” with “NIC” as the more commonly used word. In addition, there is also a type error for example “yes” becoming “yeesss” [22].

Chat language also often contains symbols and URL. Based on research by Dong et al. [22], there is a 251 times link appearance from a total of 1,700 chat sessions. Another characteristic is the length of the chat where 91.5% of sample chat has size of fewer than 50 bytes [22].

As a result of the informal language usage, a different text preprocessing approach should be done. Those steps are removing letter repetition, for example, “halooo” to become “halo” (hello) and continued by translating an informal word into formal word by creating a dictionary to map those words [30].

V. RESULTS AND DISCUSSION

A. Text Preprocessing

The chat classifier system consists of two subsystems, i.e., Master Subsystem and Bot Subsystem. Master Subsystem is used to conduct data training and testing, while the Bot Subsystem is a LINE Messenger bot to classify incoming chat in the group and included features to do additional training based on each group’s interest.

Each chat which enters the training or classification phase will be preprocessed as depicted in Figure 1.

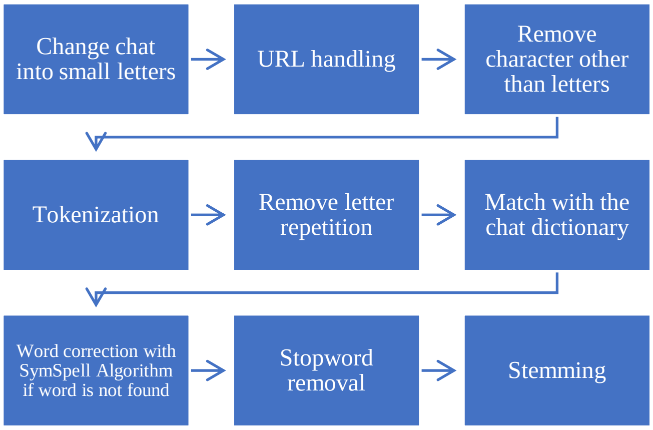

Change chat into small letters
URL handling
Remove character other than letters
Tokenization
Remove letter repetition
Match with the chat dictionary
Word correction with SymSpell Algorithm if word is not found
Stopword removal
Stemming

Figure 1. Text processing steps

B. Naïve Bayes Simulation

This research conducts three scenario experiments to determine which data set should be implemented on the bot. The sample data gathered from five committees’ group chat which had been classified by each committee leader. The ratio between training data and testing data is 70 and 30. The category used is having an interest in the committee (important) and not having an interest in the committee (unimportant).

1. *First Scenario*

This scenario uses all of the sample data with the number of training and testing data as follows.

Table 2. First Scenario Data Distribution

	Training Data	Testing Data	Total
Important	1.075	461	1.536
Unimportant	8.333	3.571	11.904
Total	9.408	4.032	13.440

In the first scenario training phase, 3,535 vocabularies are generated with the sum of words in important class as much as 11,669 and in unimportant class as much as 27,990. The testing result can be seen in Table 3, which then produces 94.2% for spam precision and 95.6% for spam recall.

Table 3. First Scenario Testing Result

	Chat Total
True Positive	3.415
False Positive	210
False Negative	156
True Negative	251

2. *Second Scenario*

The second scenario uses balanced data for each class by lowering the amount of unimportant chat, so both classes are even. Training and testing data distribution can be seen as follow.

Table 4. Second Scenario Data Distribution

	Training Data	Testing Data	Total
Important	1.075	461	1.536
Unimportant	1.075	461	1.536
Total	2.150	922	3.072

In the second scenario training phase, 2,417 vocabularies are generated with the sum of words in important class as much as 11,224 words and in unimportant class as much as 3,683. The testing result can be seen in Table 5, which then produces 70.1% for spam precision and 59.2% for spam recall.

Table 5. Second Scenario Testing Result

	Chat Total
True Positive	273
False Positive	116
False Negative	188
True Negative	345

3. *Third Scenario*

In the third scenario, the amount of data used was following the second scenario, whereas the ratio between important and unimportant chats was following the first scenario. The data distribution of the third scenario can be seen in Table 6 below.

Table 6. Third Scenario Data Distribution

	Training Data	Testing Data	Total
Important	245	105	350
Unimportant	1.905	817	2.722
Total	2.150	922	3.072

In the third scenario training phase, 1,606 vocabularies are generated with the sum of words in important class as much as 2,911 words and in unimportant class as much as 6,317.

The testing result can be seen in Table 7, which then produces 93.8% for spam precision and 94.5% for spam recall.

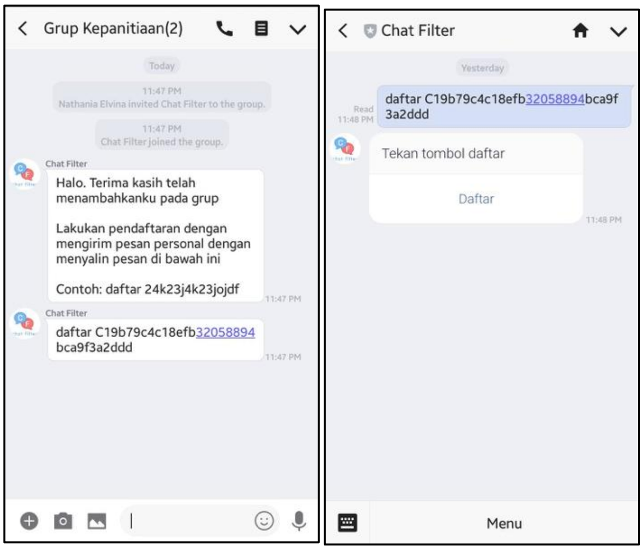
Table 7. Third Scenario Testing Result

	Chat Total
True Positive	772
False Positive	51
False Negative	45
True Negative	54

Based on the results of the three scenarios, the highest precision and recall is obtained by the first scenario; therefore its dataset is chosen to be implemented on the bot.

C. Bot Implementation

One group member (usually the leader of the committee) inserted the bot into the committee group on LINE Messenger. If the bot is successfully added, a message will pop up requesting to register the bot.



Grup Kepanitiaan(2)

Today

11:47 PM
Nathania Elvina invited Chat Filter to the group.

11:47 PM
Chat Filter joined the group.

Chat Filter: Halo. Terima kasih telah menambahkanku pada grup

Lakukan pendaftaran dengan mengirim pesan personal dengan menyalin pesan di bawah ini

Contoh: daftar 24k23j4k23jojdf (11:47 PM)

Chat Filter: daftar C19b79c4c18efb32058894bca9f3a2ddd (11:47 PM)

Chat Filter

Yesterday

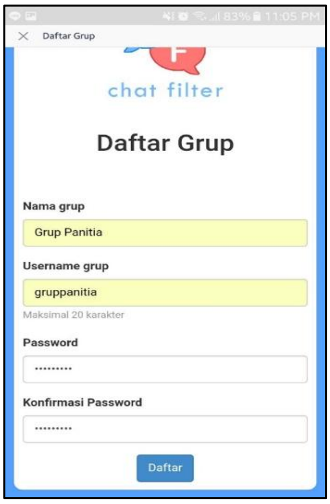
daftar C19b79c4c18efb32058894bca9f3a2ddd (Read 11:48 PM)

Tekan tombol daftar

Daftar (11:48 PM)

Figure 2. Bot registering

Then, one of the group members will register the bot by sending a private message to the bot by copying the message, which contains the word "Daftar" (register) followed by the group id. When the register button is clicked, it will be redirected to a registration page.



Daftar Grup

chat filter

Daftar Grup

Nama grup: Grup Panitia

Username grup: gruppanitia

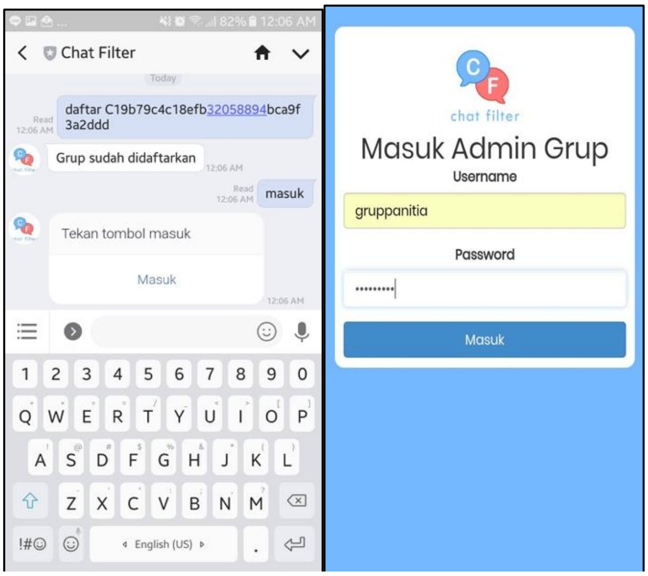
Maksimal 20 karakter

Password: ·········

Konfirmasi Password: ·········

Figure 3. Group registration

When a group is already registered, the bot will take all incoming messages and classify them. On the bot subsystem, there is an admin feature to do additional training, adding more stopword or change the group password. These features can only be accessed by the user who registered the bot in the first place. To access the admin feature, the user needs to send word "Masuk" to the bot.



Chat Filter

Today

daftar C19b79c4c18efb32058894bca9f3a2ddd (Read 12:06 AM)

Grup sudah didaftarkan (12:06 AM)

masuk (Read 12:06 AM)

Tekan tombol masuk

Masuk (12:06 AM)

chat filter

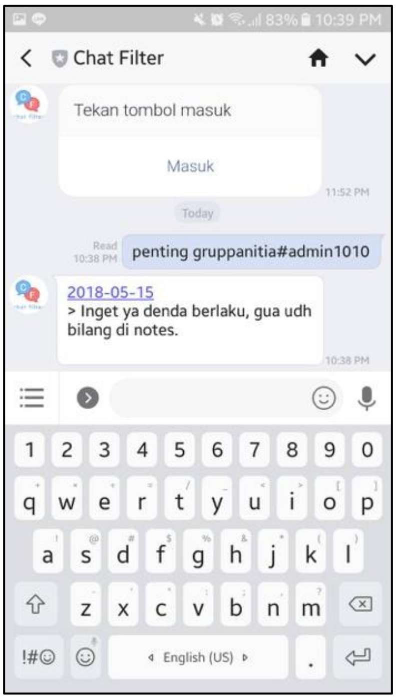
Masuk Admin Grup

Username: gruppanitia

Password: ·········

Figure 4. Admin page

Important chat from the group can be retrieved by sending a private message to bot in the format of "penting group_username#password#dd-mm-yyyy" or "penting group_username#password".



Chat Filter

Tekan tombol masuk

Masuk (11:52 PM)

Today

penting gruppanitia#admin1010 (Read 10:38 PM)

2018-05-15
> Inget ya denda berlaku, gua udh bilang di notes. (10:38 PM)

Figure 5. Chat filtering

D. Discussion

In this research, three scenarios are used. The first scenario uses all of the chat data and divided into training data and testing data. The second scenario uses a balanced portion for each class.

The third scenario is done to test whether the results of the first scenario are influenced by the total number of data or are influenced by the portion of each class.

In general, the high and low classification result is affected by the number of words in the table. If on the testing phase there are many words that are not in the table, then *add one smoothing* (*Laplace's law of succession*) should be performed. By performing add one smoothing, the likelihood can be generated for the given word, but it is only a rough estimation and will be the same value to each word done *add one smoothing*. Naive Bayes Algorithm takes the assumption that every word is independent of the other words in a given class, known as the bag of words model. This model ignores the position of the words, thus causing Naïve Bayes unable to detect the context of the chat, whether it is a joke or a serious talk, because sometimes chat containing humor uses the same words as important chat or vice versa.

In the training phase, the data source comes from various committees where there are possibly different interests between the committees. This problem had been overcome by giving features to do additional training for each committee. These causing all three scenarios do not get the perfect results for their classification.

In the first scenario, high spam precision and spam recall are generated, so is the third scenario, although the results are not as high as in the first scenario. This is due to the lower amount of vocabulary than in the first scenario. The number of vocabulary amount is one of the determinants of the high and low results of the classification.

In the second scenario, the resulted spam precision and spam recall are far below the first and third scenarios. Based on the analysis of the sample data, it is known that important chat has more words than unimportant chat. The second scenario, which uses balanced data for each class, produces 11,224 words for important class and 3,683 words for unimportant class. Based on the words produced for each class, there are several factors that cause low classification results.

First, *Laplace's smoothing* calculation to handle word which does not exist in the table will produce likelihood with a huge gap between the important class and the unimportant class. In this case, word "plus". After *Laplace's smoothing* is done, the resulting likelihood is $7.33084084 \, x \, 10^{-5}$ for important class and $1.639344262 \, x \, 10^{-4}$ for unimportant class. If there are many words that do not exist in the table, it will affect the probability of a chat.

Second, the balanced dataset scenario made each class's probability is 0.5. Looking back at the equation (3), the same $P(C_k)$ value will not give any effect in comparing probability between classes. This condition affects the message, which consists of only one word because the final probability can only depend on the likelihood of the only word occurs.

Third, in consequence to the massive difference in the total of words in each class, if there is a chat consisting of words with a higher likelihood for important classes, but there are also some words that are not in the table, *Laplace's smoothing* will be needed. As a result of the smaller *Laplace's smoothing* value for the important class, the chat might be classified as unimportant, because too many sum products are done with small *Laplace's smoothing* value.

VI. CONCLUSION

In this research, a balanced dataset resulting low spam precision and spam recall, which are 70.1% and 59.2% respectively, due to the huge gap in the number of words produced between the important and unimportant classes. The highest precision and recall calculation is obtained from the first scenario, which is using all chat data in which the number of important and unimportant chat is not balanced. The spam precision and spam recall produced are 94.2% and 95.6% respectively.

There are some suggestions for future studies related to the results of this study. The usage of algorithms that can solve and detect the messages' context in communication, such as Simple Logistic Regression and Support Vector Machine, can be implemented for better results. The study and development of chat text preprocessing can also be done, and the word dictionary can be continuously updated.

REFERENCES

1. A.R. Joshi, K. Shah, D. Desai, C. Shah, "Analysis and Detection of Eventful Messages in Instant Messengers," Proceedings of International Conference on Computing for Sustainable Global Development, India, 2016.
2. Jana, "Top 25 Installed Apps in Indonesia: June 2015," [Online]. Available: http://blog.jana.com/blog/2015/07/15/top-25-installed-apps-in-indonesia [accessed on September 2017].
3. Otsuka, T. Hirano, C. Miyazaki, R. Higashinaka, T. Makino, Y. Matsuo, "Utterance Selection Using Discourse Relation Filter for Chat-oriented Dialogue Systems," Dialogues with Social Robots, Vol.427, pp.355-365, 2017.
4. T.N. Nguyen and F. Ricci, "A Chat-based Group Recommender System for Tourism," Information Technology & Tourism, Vol.18, No.1-4, pp.5-28, 2018.
5. K. Hirata, E. Shimokawara, T. Takatani, T. Yamaguchi, "Filtering Method for Chat Logs toward Construction of Chat Robot," Proceedings of 2017 IEEE/SICE International Symposium on System Integration (SII), Taiwan, 2017.
6. K. Bala, M. Kumar, S. Hulawale, S. Pandita, "Chat-bot for College Management System using A.I," International Research Journal of Engineering and Technology (IRJET), Vol.4, No.11, pp.2030-2033, 2017.
7. N. Tepper, A. Hashavit, M. Barnea, I. Ronen, L. Leiba, "Collabot: Personalized Group Chat Summarization," Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, USA, pp.771-774, 2018.
8. P. Anugroho, I. Winarno, N. Rosyid, "Klasifikasi Email Spam dengan Metode Naive Bayes Classifier Menggunakan Java Programming," Thesis, Institut Teknologi Sepuluh Nopember, Surabaya, 2010.
9. V. Metsis, I. Androutsopoulus, G. Paliouras, "Spam Filtering with Naive Bayes - Which Naive Bayes?", Proceedings of Third Conference on Email and Anti-Spam, USA, 2006.
10. T.S.d.S. Viana, M.d. Oliveira, T.L.C.d. Silva, M.S.R.F. Ao, E.J.T. Goncalves, "A Message Classifier based on Multinomial Naïve Bayes for Online Social Contexts," Journal of Management Analytics, Vol.5, No.3, pp.213-229, 2018.
11. R. Yadav and V. Gupta, "Self-harm Prevention based on Social Platforms User Data using Naïve Bayes Classifier," Journal of Data Mining and Management, Vol.3, No.2, 2018.
12. M. Bashir, A. Rozaimee, W.M.W. Isa, "Automatic Hausa Language Text Summarization based on Feature Extraction using Naïve Bayes Model," World Applied Sciences Journal, Vol.35, No.9, pp.2074-2080, 2017.
13. H.R. Sneha, M. Rafi, M.V.M. Kumar, L. Thomas, B. Annappa, "Smartphone based Emotion Recognition and Classification," Proceedings of 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), India, 2017.

14. A.P. Wijaya and H.A. Santoso, “Improving the Accuracy of Naïve Bayes Algorithm for Hoax Classification Using Particle Swarm Optimization,” Proceedings of 2018 International Seminar on Application for Technology of Information and Communication, Indonesia, 2018.
15. V. Korde and C.N. Mahender, “Text Classification and Classifiers: A Survey,” International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.2, pp.85-99, 2012.
16. M.F. Fatroni, “Kecerdasan Buatan dalam Program Chatting untuk Merespon Emosi dari Teks Berbahasa Indonesia Menggunakan Teks Mining dan Naive Bayes,” Thesis, Institut Teknologi Sepuluh Nopember, Surabaya, 2011.
17. T. Jaka, “Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti Dalam Proses Text Mining,” Jurnal Informatika UPGRIS, Vol.1, No.1, pp.1-9, 2015.
18. E. Retnawiyati, M.M. Fatoni, E.S. Negara, “Analisis Sentimen pada Data Twitter dengan Menggunakan Text Mining terhadap Suatu Produk,” Universitas Bina Darma, Palembang, 2015.
19. A. Firdaus, Ernawati, A. Vatresia, “Aplikasi Pendeteksi Kemiripan pada Dokumen Teks Menggunakan Algoritma Nazief & Adriani dan Metode Cosine Similarity,” Jurnal Teknologi Informasi, Vol.10, No.1, pp.96-109, 2014.
20. A.F. Hidayatullah, C.I. Ratnasari, S. Wisnugroho, “Analysis of Stemming Influence on Indonesian Tweet Classification,” TELKOMNIKA, Vol.14, No.2, pp.665-673, 2016.
21. V. Ferdina, M.B. Kristanda, S. Hansun, “Automated Complaints Classification using Modified Nazief-Adriani Stemming Algorithm and Naive Bayes Classifier,” Journal of Theoretical and Applied Information Technology, Vol.97, No.5, pp.1604-1614, 2019.
22. H. Dong, S.C. Hui, Y. He, “Structural Analysis of Chat Messages for Topic Detection,” Online Information Review, Vol.30, No.5, pp.496-516, 2006.
23. L.C. Lukito, A. Erwin, J. Purnama, W. Danoekoesoemo, “Social Media User Personality Classification Using Computational Linguistic,” Proceedings of 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), Indonesia, 2016.
24. A. McCallum and K. Nigam, “A Comparison of Event Models for Naive Bayes Text Classification,” Proceedings in Workshop on Learning for Text Categorization, AAAI’98, pp.41-48, 1998.
25. A. Indriani, “Klasifikasi Data Forum dengan menggunakan Metode Naïve Bayes Classifier,” Proceedings of Seminar Nasional Aplikasi Teknologi Informasi (SNATI), Indonesia, 2014.
26. H. Shimodaira, “Text Classification using Naive Bayes,” Materials from Learning and Data Note 7, pp.1-11, 2018 [Online]. Available: https://www.inf.ed.ac.uk/teaching/courses/inf2b/learnnotes/inf2b-learn07-notes-nup.pdf [accessed on May 2018].
27. J.J. Eberhardt, “Bayesian Spam Detection,” Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal, Vol.2, No.1, pp.1-6, 2015.
28. D. Mladenic and M. Grobelnik, “Feature selection for unbalanced class distribution and Naive Bayes,” Proceedings of the Sixteenth International Conference on Machine Learning, USA, pp.258-267, 1999.
29. I. Yunita, S. Hansun, “Automatic News Blog Classifier Using Improved K-Nearest Neighbor and Term Frequency-Inverse Document Frequency,” Journal of Theoretical and Applied Information Technology, Vol.97, No.15, pp.4202-4212, 2019.
30. E. Lunando and A. Purwarianti, “Indonesian Social Media Sentiment Analysis with Sarcasm Detection,” Proceedings of 2013 International Conference on Advanced Computer Science and *Information Systems (ICACSIS)*, Indonesia, pp.195-198, 2013.

AUTHORS PROFILE

Nathania Elvina had just graduated from Universitas Multimedia Nusantara in 2018 and received her Bachelor degree in Informatics. She has participated in many events during her study at UMN and successfully finished it with flying scores.

Andre Rusli lives in Tangerang, Indonesia. He received his Bachelor Degree in Computer Science (S.Kom) from Universitas Multimedia Nusantara and Master of Science from Tokyo Denki University, Japan. Then, he began his career in university as Assistant Lecturer, until he came back from his master study and became Lecturer in the Informatics Department, Universitas Multimedia Nusantara. His research interests are software engineering, mobile technology and application development, and web development.

Seng Hansun had finished his Bachelor and Master degree from Universitas Gadjah Mada, majoring Mathematics and Computer Science program. Since 2011, he has been a lecturer and researcher at Universitas Multimedia Nusantara and published more than 75 papers both nationally and internationally. His research interests mainly in time series analysis and machine learning domain where he has successfully granted some research grants from the government and UMN institution.