Thesis Supervisor Recommendation with Representative Content and Information Retrieval

Maresha Caroline Wijanto[1]*, Rachmi Rachmadiany[2], Oscar Karnalim[3]

[1)2)3)]*Faculty of Information Technology, Maranatha Christian University, Indonesia*
M.P.H., Jl. Surya Sumantri No. 65, Sukawarna, Sukajadi, Bandung

[1)]maresha.cw@it.maranatha.edu, [2)]rrachmadiany@gmail.com, [3)]oscar.karnalim@it.maranatha.edu

Abstract

Background: In higher education in Indonesia, students are often required to complete a thesis under the supervision of one or more lecturers. Allocating a supervisor is not an easy task as the thesis topic should match a prospective supervisor's field of expertise.

Objective: This study aims to develop a thesis supervisor recommender system with representative content and information retrieval. The system accepts a student thesis proposal and replies with a list of potential supervisors in a descending order based on the relevancy between the prospective supervisor's academic publications and the proposal.

Methods: Unique to this, supervisor profiles are taken from previous academic publications. For scalability, the current research uses the information retrieval concept with a cosine similarity and a vector space model.

Results: According to the accuracy and mean average precision (MAP), grouping supervisor candidates based on their broad expertise is effective in matching a potential supervisor with a student. Lowercasing is effective in improving the accuracy. Considering only top ten most frequent words for each lecturer's profile is useful for the MAP.

Conclusion: An arguably effective thesis supervisor recommender system with representative content and information retrieval is proposed.

I. INTRODUCTION

According to the regulation number 3, year 2019, by Indonesian National Accreditation Agency for Higher Education (BAN-PT), the average duration for a student to complete a study and the rate of on-time graduation are both critical success factors for a higher education institution. However, several reports [1], [2] show that the two factors are challenging to achieve since the number of the newly enrolled students is often two times higher than the number of graduating students (which are 1,233,218 and is 641,098 respectively). Despite all the efforts, the number of graduating students has increased only slightly over the years [3].

Many universities in Indonesia require students to write a thesis to graduate, and this might be the biggest challenge for students to graduate on time. Not only that thesis is a comprehensive assessment but administering it can be challenging because of the ratio between students and thesis supervisors is often gapped. Moreover, matching potential a supervisor's expertise with a students' proposal can be time consuming. There is sometimes a trade-off between time efficiency and an ideal match.

Assigning an academic supervisor whose expertise is strongly relevant with the thesis topic has become a crucial task. Supervisors play a key role in helping students complete the thesis, so it is important to match the expertise and the demand. In most universities, the selection of the supervisors is done manually by a thesis coordinator, which is time consuming and prone to human error. In response to that, this paper proposes a supervisor recommender system where relevancy is determined based on the supervisors' previous publications and the supervised theses. With this system on board, universities are expected to assist their students in choosing the right supervisor for their thesis, which is crucial for students in order to complete the study successfully and to graduate on time.

Unique to the recommender system, each potential supervisor is profiled by their academic publications, merged as a large text. This is clearly more descriptive than data used in some similar recommender systems, which are

---
* Corresponding author

academic publications' keywords [4] and project titles [5]. This is also more accurate than using former students' thesis proposals to form the supervisor profiles [6]. On most occasions, proposals do not reflect the resulted theses as they are changed during the supervision process. The system requires a thesis proposal which we believe is richer than several keywords [4] or topics [5] as the input and it will rank the potential supervisors based on their relevancy. For scalability, that relevancy is defined with cosine similarity in vector space model, in which each student or supervisor data is treated as a bag of words. To the best of our knowledge, this is the first one of its type.

II. Literature Review

Recommender systems work by filtering essential information about specific users based on a large amount of user-generated data (covering user preference, interests, or observed behavior), and subsequently suggest relevant items based on that [7]. This is typically used to recommend movies, books, research fields and products.

Recommender systems work in four phases. At first, users' relevant information is collected and filtered to build user profiles. After that, each available item is assigned with a value suggesting a user's interests via a particular recommendation technique. The items are then sorted based on those interests and those with the highest interest value are recommended [7]. To ensure that the recommended items match the users' needs, the performance on various data sets needs to be evaluated [8].

Existing recommendation techniques can be generally classified to collaborative filtering, content-based filtering, and hybrid filtering [9]. Content-based filtering gives recommendations based on users' previous choices [10] and is closely linked with supervised machine learning [11]. Collaborative filtering gives recommendations based on information provided by other users with similar preferences [12]. Hybrid filtering is a combination of the two methods. Another study introduces a new technique called demographic filtering [10], which assumes that people with similar personal attributes (e.g., age, sex, country) will also have similar preferences.

There are several recommender systems suitable for suggesting thesis supervisors. OfficeHours [4] is an interactive recommender system with reinforcement learning on board. It assists students to find their potential supervisor for their theses. At first, students can choose some of the given keywords extracted from the supervisors' academic publication as the search queries. If the suggested supervisors do not suit their need, they can edit the keywords and repeat the process as many as they like. This system is evaluated via interviews with students and faculty members, as well as through system logs analysis.

Ismail et al [5] proposed a thesis supervisor recommender system with Euclidean distance as the similarity measurement. The data is collected from questionnaires distributed to final-year students, asking about their interests in five topics: multimedia, web application, network, artificial intelligence and mobile application. Supervisors are sorted based on the number of previous and current project titles classified to topics relevant to given students. The relevancy is defined with Euclidean distance. From our understanding, this system has not been evaluated.

Another system proposed by Yasni et al [6] uses cosine similarity for calculating the relevancy between students and their advisors. Each student is required to provide a thesis proposal that contains a title, a topic and an abstract. Those three components will be used for recommending advisors, who are profiled based on their previously supervised students' thesis proposals. The evaluation was performed using precision and recall for three different queries.

Evaluation metrics for a recommender system can be grouped into three categories: predictive accuracy metrics, classification accuracy metrics and rank accuracy metrics [13], [14]. Predictive accuracy or rating prediction metrics measure how close the predicted ratings to the real ones. Classification accuracy metrics measure how many relevant items are correctly and incorrectly classified. Rank accuracy metrics or ranking prediction metrics measure how good a system is to order recommended items based on users' preferences. This is more suitable to recommendation systems with ranking mechanism.

III. Method

The proposed system accepts a student thesis proposal and therefore lists potential supervisors in descending order based on their relevancy. It has two phases called indexing and recommending (see Fig. 1). Indexing phase collects the academic publications of the potential supervisors and converts them to become their profile. Recommending phase is subsequently performed and it will list potential supervisors based on their relevancy to a student's thesis proposal. This stage is partly inspired by a study about recommending venues for academic publications [15]. The supervisors' academic publications and the students' thesis proposals are both preprocessed in the same way.
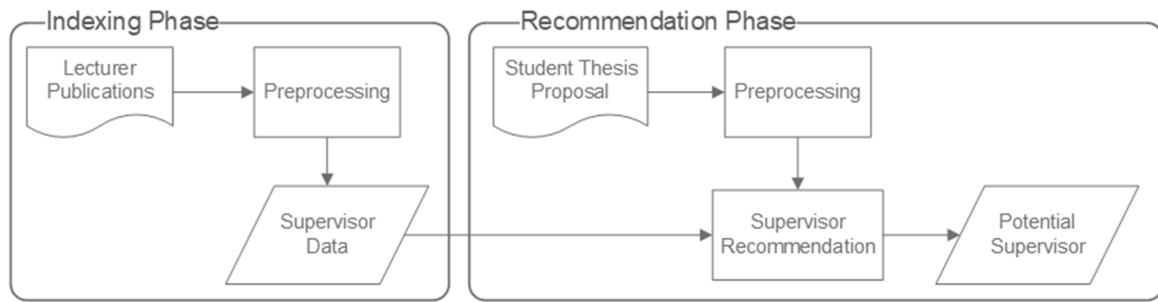
Fig. 1 System architecture

In the indexing phase, each supervisor is required to upload their academic publications as PDF files. The text of each PDF is preprocessed separately and then merged as a large bag of words. The preprocessing is performed in fourfold. At first, the stop words (meaningless words that frequently appear in a sentence) are removed for effectiveness [16]. As our dataset is mainly Indonesian, the stop words are the Indonesian ones as defined by Rahutomo and Ririd [17]. Secondly, the text is tokenized to form a sequence of words [18] to avoid trivial mismatches caused by meaningless characters. In our case, words should contain alphanumeric characters (as they are commonly meaningful) and their length should be three or more (as shorter words are not meaningful) [19]. Thirdly, each two adjacent words are merged as one phrase that is often called bigram. Finally, those bigrams will be converted into a bag of words, storing only distinct bigrams and their occurrence frequencies. For an extremely large data set, the system can be set to take only top ten bigrams per academic publication, although it is optional.

The recommending phase works by accepting a thesis proposal submitted by a student (as a raw text). The proposal will be preprocessed in the same way as the supervisors' academic publications. The resulted bag of words will be compared to those of from the supervisors in a vector space model (a simple yet effective retrieval model for term weighting, ranking and relevance feedback) by considering each entity as a word vector. To measure the relevancy, cosine similarity is used. It measures the cosine of the angle between two dimensional vectors: the query vector (Q) and the document vector (D) as seen in (1). The former is a student thesis proposal, while the latter is a supervisor's profile generated based on their academic publications. d_j is the occurrence frequency of the jth word in document D; q_j is the occurrence frequency of the jth word in query Q; and t is the total number of words.

$$\text{Cosine}(D,Q) = \frac{\sum_{j=1}^{t} d_j \cdot q_j}{\sqrt{\sum_{j=1}^{t} d_j^2 \cdot \sum_{j=1}^{t} q_j^2}} \qquad (1)$$

If both vectors are identical (i.e., all words are shared with the same occurrence frequencies), it will result 1. Otherwise, it will result 0 [20]. As a result, the potential supervisors will be sorted in descending order based on their cosine similarity to a given student thesis proposal. For efficiency, only top five most relevant results are shown.

Fig. 2 Thesis proposal submission page

The system assigns three kinds of roles: a student, a lecturer (or potential thesis supervisor) and a thesis coordinator. Each student enrolled to the thesis course needs to submit their proposal to the system in order to get their supervisor assigned (see Fig. 2 for the submission page). The thesis coordinator can get a recommendation for each student about the potential supervisors sorted based on their relevancy (see Fig. 3 for the result page).

Fig. 3 Thesis supervisor recommendation list page

They can also see how many students are currently supervised by each potential supervisor to avoid work-overload. It is important to note that the thesis coordinator can pre-group the lecturers based on their broad research expertise to enhance the effectiveness. Lecturers are required to submit their academic publications periodically to make the recommendation more accurate. The details of these features can be seen in Fig. 4.
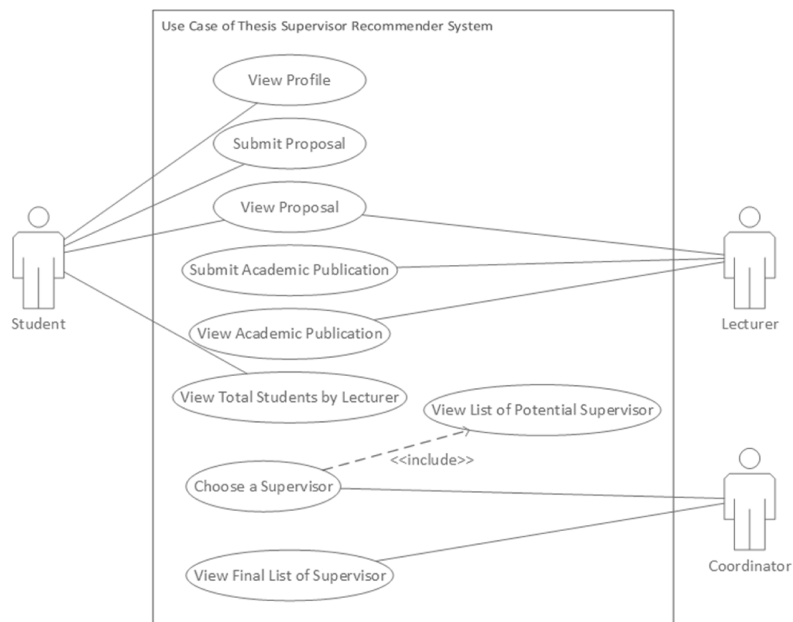
Fig. 4 Use case of thesis supervisor recommender system

The proposed thesis supervisor recommender system was then evaluated with two metrics on board: accuracy and mean average precision (MAP). Accuracy measures the quality of nearness to the truth [21] based on the proportion of correctly recommended cases to the total number of cases [7], as seen in (2) [21].

$$accuracy = \frac{number\ of\ correct\ cases}{number\ of\ cases} \qquad (2)$$

MAP [8] is derived from the precision (i.e., the proportion of correctly recommended supervisors to the total number of recommended supervisors [22]) and it exclusively considers the rank position [15]. It is measured as in (3) [14] by calculating the average precision (AP) at any positions of correctly recommended supervisor in which M is the number of recommended supervisors. The average precision can be calculated as in (4) where r is the rank, $rel(r)$ represents the relativity function given rank r, $P(r)$ represents the precision given rank r, and *#relevant_items* means the number of correctly-recommended supervisors.

$$MAP = \frac{\sum_{u=1}^{M} AP_u}{M} \quad (3)$$

$$AP = \frac{\sum_{r=1}^{n} P(r) \times rel(r)}{\#relevant\_items} \quad (4)$$

Sixteen scenarios were considered by listing all possible combinations of the four factors' values: (1) number of considered words per lecturer's profile (all or top ten most frequent words), (2) token variation in forming the bag of words (unigram or bigram), (3) expertise-based grouping (grouped or not), (4) lowercasing (lowercased or not).

To know which factors substantially affect the accuracy, we measured the impact of each factor separately. As there were only two possible values per factor, the impact was measured by calculating the average difference between scenarios with the first alternative value and those with the second alternative value. If the comparison results in a large difference, the factor is arguably crucial and greatly affects effectiveness.

The data set consists of 139 student thesis proposals with supervisor(s) having been allocated manually by the thesis coordinator. There are 29 potential supervisors grouped into three broad research strands: 'information system', 'mobile and multimedia application', and 'software engineering and computer science'. The groups have 12, 7, and 10 lecturers respectively. In total, there are 176 lecturers' academic publications involved.

TABLE 1
THE ACCURACY OF GIVEN SCENARIOS

No	ID	Considered Words	Token Variation	Expertise-based Grouping	Lowercasing	Correctly Specified data	Percentage to Total Data
1	EA-1	All	Unigram	Yes	No	93	66.9%
2	EA-2	All	Unigram	Yes	Yes	97	69.8%
3	EA-3	All	Unigram	No	No	54	38.8%
4	EA-4	All	Unigram	No	Yes	59	42.4%
5	EA-5	All	Bigram	Yes	No	96	69.1%
6	EA-6	All	Bigram	Yes	Yes	100	71.9%
7	EA-7	All	Bigram	No	No	54	38.8%
8	EA-8	All	Bigram	No	Yes	62	44.6%
9	EA-9	Top-10	Unigram	Yes	No	89	64%
10	EA-10	Top-10	Unigram	Yes	Yes	95	68.3%
11	EA-11	Top-10	Unigram	No	No	53	38.1%
12	EA-12	Top-10	Unigram	No	Yes	53	38.1%
13	EA-13	Top-10	Bigram	Yes	No	80	57.6%
14	EA-14	Top-10	Bigram	Yes	Yes	90	64.7%
15	EA-15	Top-10	Bigram	No	No	46	33.1%
16	EA-16	Top-10	Bigram	No	Yes	60	43.2%
					The Average of Accuracy score for all scenarios		53.09%

IV. RESULTS

Table 1 shows the accuracy of all scenarios. EA-6 results in the highest accuracy (71.9%) while EA-15 leads to the worst one (38.8%). Among the considered factors, expertise-based grouping and lowercasing are the most affecting ones; a change in the values can lead to a large accuracy difference. Four scenarios with the lowest accuracy (EA-15, EA-12, EA-11 and EA-3 that are coloured grey) do not implement expertise-based grouping and lowercasing, while the top ones implement them. Expertise-based grouping can obviously limit the number of potential supervisor candidates, preventing the system from producing outlier results. Lowercasing can be helpful as capitalisation does not affect word semantic and therefore should be removed from consideration. On average, our

recommender system is considerably effective, it can correctly predict the potential supervisors of more than half of the student thesis proposals.

TABLE 2
THE AVERAGE ACCURACY PER FACTOR

No	Parameter	Value 1	Average Accuracy of for Value 1	Value 2	Average Accuracy for Value 2	Differences
1	Considered Words	All	55.29%	Top-10	50.89%	4.4%
2	Token Variation	Unigram	53.30%	Bigram	52.88%	0.42%
3	Expertise-based Grouping	Yes	66.54%	No	39.64%	26.9%
4	Lowercasing	Yes	55.38%	No	50.8%	4.58%

When grouped per factor, Table 2 shows that both expertise-based grouping and lowercasing experience a larger difference compared to the other two. In other words, our statement about their impact is true and both features are better to be implemented.

TABLE 3
THE MAP OF GIVEN SCENARIOS

No	ID	Considered Words	Token Variation	Expertise-based Grouping	Lowercasing	Correctly Specified data
1	EA-1	All	Unigram	Yes	No	36.63%
2	EA-2	All	Unigram	Yes	Yes	36.13%
3	EA-3	All	Unigram	No	No	22.05%
4	EA-4	All	Unigram	No	Yes	22.79%
5	EA-5	All	Bigram	Yes	No	36.91%
6	EA-6	All	Bigram	Yes	Yes	38.42%
7	EA-7	All	Bigram	No	No	19.16%
8	EA-8	All	Bigram	No	Yes	23.60%
9	EA-9	Top-10	Unigram	Yes	No	33.84%
10	EA-10	Top-10	Unigram	Yes	Yes	32.69%
11	EA-11	Top-10	Unigram	No	No	19.12%
12	EA-12	Top-10	Unigram	No	Yes	18.76%
13	EA-13	Top-10	Bigram	Yes	No	31.03%
14	EA-14	Top-10	Bigram	Yes	Yes	36.62%
15	EA-15	Top-10	Bigram	No	No	17.97%
16	EA-16	Top-10	Bigram	No	Yes	23.47%
				The Average of MAP Score for all scenarios		28.07%

In terms of MAP, Table 3 shows that EA-6 (see the green line) is the best one with 38.42% of MAP. Combining with a finding from the previous subsection, it means that the scenario works best in terms of both accuracy and MAP. The worst scenarios (marked grey) are EA-15, then followed by EA-12, EA-11, and EA-7. Again, the finding is quite similar to the accuracy one except that the impact of EA-7 is exclusive to MAP and the impact of EA-3 is exclusive to accuracy. It is possible that EA-7's relevant potential supervisors are placed at the end of the recommendation list, resulting high MAP but low accuracy.

TABLE 4
THE AVERAGE MAP PER FACTOR

No	Parameter	Value 1	Average Accuracy for Value 1	Value 2	Average Accuracy for Value 2	Differences
1	Considered Words	All	29.46%	Top-10	26.69%	2.77%
2	Token Variation	Unigram	27.75%	Bigram	28.40%	0.65%
3	Expertise-based Grouping	Yes	35.28%	No	20.87%	14.42%
4	Lowercasing	Yes	29.06%	No	27.09%	1.97%

When grouped per factor (see Table 4), expertise-based grouping still shows the largest difference. However, it is followed by the number of considered words instead of capitalisation. Further observation shows that taking only top ten most frequent words for each lecturer's profile can make the position of the relevant potential supervisors higher, as some rare words can be misleading due to their outlier nature.

V. Discussion

Four factors are evaluated with accuracy and MAP as the metrics. The evaluation shows that expertise-based grouping should be applied to prevent the system from generating outlier results. Lowercasing can be used for a higher level of accuracy. Considering only top ten most frequent words in each lecturer's profile is preferred for a higher MAP. Token variation is the only factor that shows no promising impact. The most effective scenario is grouping the supervisor profiles based on their broad research expertise; and each profile should consider all words, lowercased and formatted as bigrams. This can accurately predict the potential supervisors of more than half of the student thesis proposals. However, some of the relevant supervisors are not placed on top of the recommendation list as the MAP is considerably low.

Compared to the existing systems, our system has more descriptive data as each potential supervisor is profiled by using their academic publications, merged as a large text. OfficeHours [4] only uses the keywords of academic publications while Ismail et al. [5] only use the project titles. Ours is also more accurate than another system [6] that relies on students' thesis proposals only in forming the supervisor profiles. Typically, the proposals do not reflect the resulted thesis as they are changed during the supervision process.

There are at least three limitations of this study, which need to be carefully considered while interpreting our findings. First, the data set are primarily written in Indonesian and English. The findings cannot be generalized to any human languages. Second, the proposals used in evaluation are from information technology major. If the system is applied to other major, the findings might be changed. Third, the proposals have up to 500 words on average. Longer content might result in different findings.

VI. Conclusions

In this paper, we propose a thesis supervisor recommender system with representative content and information retrieval. It accepts student thesis proposal and subsequently returns a list of potential supervisors sorted based on the relevancy between the supervisors' academic publications and the proposal. Our evaluation shows that expertise-based grouping and lowercasing are two important factors in designing a thesis supervisor recommender system. They are expected to be used for future research in this field, especially if the system is similar to ours. For future work, we plan to integrate student course grades so that the relevancy is not only defined based on the topic but also the student skills. A student is more likely to successfully complete their thesis if they have skills required for that.

References

[1] Bidang Pendayagunaan & Pelayanan PDSPK, "Perkembangan Pendidikan Tinggi Tahun 1999/2000-2013/2014," Pusat Data dan Statistik Pendidikan dan Kebudayaan (PDSPK), Kementerian Pendidikan dan Kebudayaan (Kemdikbud), Jakarta, 2015.

[2] PDDIKTI Kemristekdikti, Higher Education Statistical Year Book 2014/2015, Jakarta: PDDIKTI Kemristekdikti, 2016.

[3] D. P. Kusumaningrum, N. A. Setiyanto, E. Y. Hidayat and K. Hastuti, "Recommendation System for Major University Determination Based on Student's Profile and Interest," *Journal of Applied Intelligent System,* vol. 2, no. 1, pp. 21-28, April 2017.

[4] Y. Gao, K. Ilves and D. Głowacka, "OfficeHours: A System for Student Supervisor Matching through Reinforcement Learning," in *20th Intelligent User Interfaces Companion*, Atlanta, 2015.

[5] M. H. Ismail, T. R. Razak, M. A. Hashim and A. F. Ibrahim, "A Simple Recommender Engine for Matching Final-Year Project Student with Supervisor," in *Colloquium in Computer & Mathematical Sciences Education*, Arau, 2015.

[6] L. Yasni, S. F. C. Haviana and I. M. I. Subroto, "Implementasi Cosine Similarity Matching dalam Penentuan Dosen Pembimbing Tugas Akhir," *Transmisi,* vol. 20, no. 1, pp. 22-28, January 2018.

[7] F. Isinkaye, Y. Folajimi and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal,* vol. 16, no. 3, pp. 261-273, November 2015.

[8] M. Chen and P. Liu, "Performance Evaluation of Recommender Systems," *International Journal of Performability Engineering,* vol. 13, no. 8, pp. 1246-1256, December 2017.

[9] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering,* vol. 17, no. 6, pp. 734-749, June 2005.

[10] J. Bobadilla, F. Ortega, A. Hernando and A. Gutiérr, "Recommender Systems Survey," *Knowledge-Based Systems,* vol. 46, pp. 109-132, July 2013.

[11] R. Burke, A. Felfernig and M. H. Göker, "Recommender Systems: An Overview," *AI Magazine,* vol. 32, no. 3, pp. 13-18, September 2011.

[12] J. Salter and N. Antonopoulos, "CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering," *IEEE Intelligent Systems,* vol. 21, no. 1, pp. 35-41, February 2006.

[13] J. L. Herlocker, J. A. Konstan, L. G. Terveen and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Transactions on Information Systems,* vol. 22, no. 1, pp. 5-53, January 2004.

[14] G. Schröder, M. Thiele and W. Lehner, "Setting Goals and Choosing Metrics for Recommender System Evaluations," in *5th ACM International Conference on Recommender Systems, Recsys 2011*, Chicago, 2011.

[15] O. Karnalim and A. Z. Qashri, "Measuring the Significance of Writing Style for Recommending Where to Publish – A Case Study," *Cybernetics and Information Technologies,* vol. 19, no. 3, pp. 3-15, 2019.

[16] J. K. Raulji and J. R. Saini, "Stop-Word Removal Algorithm and its Implementation for Sanskrit Language," *International Journal of Computer Applications,* vol. 150, no. 2, pp. 15-17, September 2016.

[17] F. Rahutomo and A. R. T. H. Ririd, "Evaluasi Daftar Stopword Bahasa Indonesia," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK),* vol. 6, no. 1, pp. 41-48, February 2019.

[18] N. H. A. Sari, M. A. Fauzi and P. P. Adikara, "Klasifikasi Dokumen Sambat Online Menggunakan Metode K-Nearest Neighbor dan Features Selection Berbasis Categorical Proportional Difference," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 2, no. 8, pp. 2449-2454, August 2018.

[19] M. C. Wijanto, "Sistem Pendeteksi Pengirim Tweet dengan Metode Klasifikasi Naive Bayes," *Jurnal Teknik Informatika dan Sistem Informasi,* vol. 1, no. 2, pp. 172-182, 2015.

[20] W. B. Croft, D. Metzler and T. Strohman, Search Engines: Information Retrieval in Practice, Boston: Pearson Education, Inc., 2015.

[21] F. H. del Olmo and E. Gaudioso, "Evaluation of recommender systems: A new approach," *Expert Systems with Applications,* vol. 35, no. 3, pp. 790-804, October 2008.

[22] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, Massachusetts: Morgan Kaufmann, 2012.