International Journal of Engineering & Technology, *7 (4.36) (2018) 1361-1366*

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Implementation of Naive Bayes Classifier and Log Probabilistic for Book Classification Based on the Title

Ridwan Rismanto[1]*, Dimas Wahyu Wibowo[2], Arie Rachmad Syulistyo[3]

[1,2,3]Department Of Information Technology, State Polytechnic of Malang, Indonesia
**Corresponding author E-mail: rismanto@polinema.ac.id*

Abstract

Book is an important medium for teaching in higher education. It is facilitated by a library or a reading room which enabled student and teacher to fulfill their references for teaching and learning activities. For easy searching, each book classified by categories. In our institution, Information Technology Major of State Polytechnic of Malang, those categories are specifics to computer science topics. Every book entry need to be classified accordingly and to perform such task, one need to understand major keywords of the book title to correctly classify the books. The problem is, not all the librarian have such knowledge. Therefore manually classifying hundreds and even thousands of book is an exhausting work. This research is focused on automatic book classification based on its title using Naive Bayes Classifier and Log Probabilistic. The Log Probabilistic implementation is to solve the probability calculation result that is too small that cannot be represented in a computer programming floating points variable type. The algorithm then implemented in a web application using PHP and MySQL database. Evaluation has been done using Holdout method for 240 training dataset and 80 testing dataset resulting in 75% of accuration. We also tested the accuracy using K-fold Cross Validation resulting in 66.25% of accuration.

Keywords: *Classification, Book, Naïve Bayes, Log Probabilistic, Machine Learning*

1. Introduction

One of the most important media in teaching and learning activity of higher education is book. The needs for book as a reference for those activities must be facilitated by the institution, either by providing a library or a reading room.

A book can be categorized by several criteria. One of it is based on its type (novel, reference, encyclopedia, etc), or by its topic (economy, religion, technical, etc). The classification system based on topic usually use well-known methodology such as Dewes Decimal Classification (DCC) [1], in which, every category reflected by a decimal. In reality, books that available in a campus often fall into specific sub-categories according to its faculty. Thus those book needs to be re-classify to that specific topic to make it easier for student or teacher searching for the book they needed.

In our institution, Information Technology Major of State Polytechnic of Malang, books in our reading room are classified in sub-topic of informatics and computer sub-category. Specifically, they fall in programming, mobile programming, database, networking, information system, operating system, soft computing and multimedia sub-categories. Those categories does not covered in DCC classification, so it needs to be re-classified manually by looking at its title or descriptions which is an exhausting work. An automatic categorization system would be a good solution for this problem.

Previous work at book catalogue searching system talks about how a search keyword being matched with a book database using Naïve Bayes Classifier in Mulia Bookstore application, resulting in 88.89% of accuracy [2]. Other research focused on automatic thesis classification using short description compared between two algorithm, K-Nearest Neighbor (K-NN) and Naïve Bayes Classification, resulting in Naïve Bayes Classifier having highest accuracy (65.4%) compared with K-NN (51.14%) [3]. Other research about journal classification system using Naïve Bayes Classifier and Vector Space Model resulting in 60.7% of accuracy [4]. An interesting work using Bayesian network has been done to assist student in learning problem-solving skills through solution designing activities by offering various personalized options in flowchart development along with step-by-step guidance. The system use pre-test and tic-tac-toe game result performed by the student to gain student’s profile which is the input to the Bayesian network [12].

One problem with Naïve Bayes Classifier is, when the feature of a document (the number of words in a document, or book title) is too many, the calculation result will be too small therefore cannot be represented by a standard floating points programming variable data type such as float or double. This can be solved by using Log-Probabilistic, which use logarithmic formula to calculate the probabilistic. The result of the implementation is a negative value, compared to positive (but potentially too small) value of the standard method.

In this research, we are focusing on how to implement Naïve Bayes Classifier to classify book to its computer sub-categories using Naïve Bayes Classifier and Log Probabilistic. Dataset for training and testing acquired from our existing digital library application (http://digilib.jti.polinema.ac.id). The accuracy of the classifier then be tested using Holdout and K-fold Cross Validation. We hope this research will ease the classification works by automating it using computer system, and also make it easier for searching the right book classified by its category.

2. Literature Review

2.1. Naïve Bayes Classifier

Naïve Bayes Classifier is a well-known algorithm to calculate highest probabilty to classify a dataset into the right category. In this research, the dataset are book documentation in campus' library. There are two steps in document classification. First is training phase towards dataset of known categories (or labels). Second is classification phase towards unknown dataset of unknown labels [5].

In Naïve Bayes Classifier every documents represented with F and H symbols, "F1, F2, … Fn" in which F1 is the first word, F2 is the second word, and so on. Therefore, H is an array of labels, or book categories. First we calculate the probability of each category P(Hi), calculated by the amount of training dataset for several categories (or labels), divided by the total training dataset. P(Hi) can be described by this following formula:

$$P(H_i) = \frac{|docs_i|}{|totaldocs|} \quad (1)$$

Where P(Hi) is the probability of each category, |docs$_i$| is the amount of training dataset for a category, and |totaldocs| is the amount of total training dataset.

At classification process, the algorithm will calculate and select the highest probability of all the document's category being tested (H_{MAP}), as can be seen in this following formula:

$$\begin{aligned} H^{MAP} &= \underset{i}{argmax}\, P(D \mid H_i)P(H_i) \\ &= \underset{i\in\{label_i\ldots label_n\}}{argmax} P(F_1, \ldots ,F_m \mid H_i)P(H_i) \\ &= \underset{i\in\{label_i\ldots label_n\}}{argmax} [P(F_1 \mid H_i) \ldots P(F_m \mid H_i]P(H_i) \\ &= \underset{i\in\{label_i\ldots label_n\}}{argmax} [\textstyle\prod_{j=1}^{m} P\left(F_j \mid H_i\right)]\, P(H_i) \end{aligned} \quad (2)$$

Where (H_{MAP}) is the highest probability among all categories, H_i is the category at each i, and F_j is the word at each j.

2.2. Laplacian Smoothing

In a big dataset, the random choice of training dataset will result in a zero value in probabiliry model. This zero value will render the Naïve Bayes Classifier unable to classify the input data. In this case, a smoothing method is applied to eliminate zero value in probability model. Laplacian smoothing is one of smoothing method using in Naïve Bayes Classifier. This method also known as add-one-smoothing, because in its calculation, every parameters will be added by 1 [6]. The implementation can be seen in the following formula:

$$P\left(F_j \mid H_i\right) = \frac{F_j + 1}{H_i + |v|} \quad (3)$$

Where |v| is the amount of features (or words) in training dataset.

2.3. Log Probabilistic

Whem NBC calculation being performed, a problem will potentially occurs when m (amount of total features, or words) is too big. This leads to the product of every P(Fj | Hi) become too small it almost near zero. In this case, a programming language floating datatype (float or double) will not be able to represents the value because of lack of floating point precission.

The solution for this case is by using a Log Probabilistic. Plugged in Naïve Bayes Classifier calculation, the formula can be seen in this following formula (4). Note that the result of this calculation is a negative number. But this does not differentiate of how we choose the highest score of this classifier.

$$\begin{aligned} H^{MAP} &= \underset{i\in\{label_i\ldots label_n\}}{argmax} \left[\prod_{j=i}^{m} P\left(F_j \mid H_i\right)\right] P(H_i) \\ &= \underset{i\in\{label_i\ldots label_n\}}{argmax} ln \left([\textstyle\prod_{j=i}^{m} P\left(F_j \mid H_i\right)]P(H_i)\right) \\ &= \underset{i\in\{label_i\ldots label_n\}}{argmax} ln \left[\prod_{j=1}^{m} P\left(F_j \mid H_i\right)\right] + \ln P(H_i) \\ &= \underset{i\in\{label_i\ldots label_n\}}{argmax} \left[\sum_{j=1}^{m} \ln P\left(F_j \mid H_i\right)\right] + \ln P(H_i) \end{aligned} \quad (4)$$

2.4. Stratified Random Sampling

Stratified Random Sampling or Holdout is a method to provide dataset to be used as training dataset as one part and the rest of the part is used as testing dataset [7].

In this research, the dataset split by two, training and testing dataset. Training dataset is used to create the model, and testing dataset is used to estimate accuration of the created model.

The use of randomly picking training and testing data, could yield unproportional dataset at each classification process. In one case it is a possibility for a traning dataset has a dominance of one category or labels compared with other training dataset. To overcome this problem, Stratified Random Sampling is used, to carefully select the dataset to provide proportionally random picked training and testing data, as shown in Figure 1.

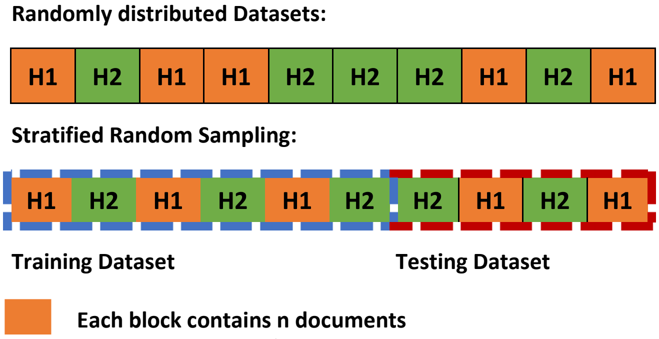
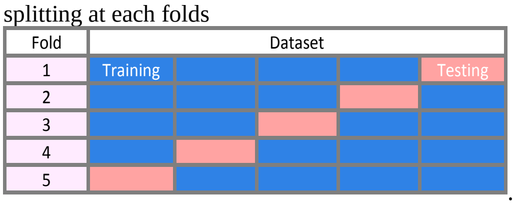

Randomly distributed Datasets:
H1
H2
H1
H1
H2
H2
H2
H1
H2
H1
Stratified Random Sampling:
H1
H2
H1
H2
H1
H2
H2
H1
H2
H1
Training Dataset
Testing Dataset
Each block contains n documents

Fig. 1: Stratified Random Sampling

2.5. K-Fold Cross Validation

In evaluating a machine learning model for a limited dataset, K-Fold Cross Validation is a well known sampling method that is generally used. This method splits dataset in k-group or folds, with the same amount of data at each folds. The first fold is validation set, and then the evaluation continued to the rest of the fold or k-1 folds [8].

The choice of k is usually 5 or 10, but there are no formal rule for it. The bigger k, the smaller difference between training set and testing set amount of data. The smaller k in other words, will reduce bias from the evaluation [9]. Figure 2 shown training and testing splitting at each folds



Fold	Dataset				
1	Training	🟦	🟦	🟦	Testing
2	🟦	🟦	🟦	🟥	🟦
3	🟦	🟦	🟥	🟦	🟦
4	🟦	🟥	🟦	🟦	🟦
5	🟥	🟦	🟦	🟦	🟦

Fig. 2: K-Fold Cross Validation

3. Methodology

3.1. Data Collection

We extract the dataset needed in this research from the existing application (http://digilib.jti.polinema.ac.id). Part of the dataset will be used for training and other part will be used for testing. The field we extract are book id, book title and label.

3.2. Preparation and Implementation

Category label used to classify the book are the following:

Table 1: Category Labels

#	Category Label
1	Programming
2	Mobile Programming
3	Multimedia
4	Networking
5	Soft Computing
6	Database
7	Information System
8	Operating System

The implementation of this research will follow text processing preparation, which is case folding, tokenizing, stemming, removing stop words. After that we extract word frequencies from the training datasets to calculate the probability of each labels. Then we can begin classifying by feeding the testing datasets, and compare the category label generated by the classifier with the actual category label of the book. The complete stages of preparation and implementation are explained in Figure 3.

For the stopwords list, specifically Indonesian language, we use Tala for the dictionary [10]. For stemming, we use Nazief and Adriani algorithm [11].

Dataset used in this research is 320 records. This data will be split by 2, the first part is training data, 240 records, and the rest 80 records is testing data. For each category label, there are 30 records for training, and 10 for testing.

Table 2 will give an example of the training dataset, and Table 3 is the testing dataset.

Table 2: Example of Training Dataset

Book ID	Book Title	Label
3	The Definitive Guide to MySQL 5	database
60	SQL Tutorial Plus Studi Kasus dengan ORACLE	database
70	Pengolahan Database Dengan MySQL	database
6	PostgreSQL : a comprehensive guide to building, programming, and administering PostgreSQL databases	programming
7	Web application architecture : principles, protocols, and practices	programming
8	Ajax : creating Web pages with asynchronous JavaScript and XML	programming
568	Pengenalan Komputer Dasar ilmu Komputer, Pemrograman, Sistem Informasi, dan Intelegensi Buatan	soft computing
569	Belajar Cepat Fuzzy Logic Dengan Matlab	soft computing

Table 3: Example of Testing Dataset

Book ID	Book Title	Actual Label	System Label
573	Sistem Basis Data	database	-
585	Belajar Database Menggunakan MySQL	database	-
621	Panduan Belajar SQL Server 2005 Express Edition	database	-
522	Mahir dalam 7 Hari Autodesk 3ds Max 2009	multimedia	-
523	3D Studio MAX 2010 Dasar dan Aplikasi	multimedia	-
525	3D Studio Max 2012	multimedia	-
554	Pemrograman Android dalam Sehari	mobile programming	-
562	Kolaborasi Dahsyat Android Dengan PHP & MySQL	mobile programming	-

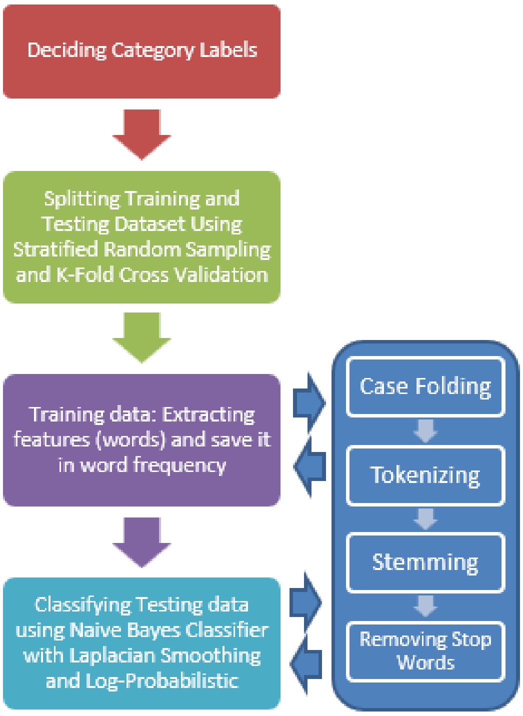

Deciding Category Labels
Splitting Training and Testing Dataset Using Stratified Random Sampling and K-Fold Cross Validation
Training data: Extracting features (words) and save it in word frequency
Classifying Testing data using Naive Bayes Classifier with Laplacian Smoothing and Log-Probabilistic
Case Folding
Tokenizing
Stemming
Removing Stop Words

Fig. 3: Preparation and Implementation

3.3. Web Application Implementation

In this research, we implemented our methodology for training and testing in a web application using PHP as web programming language and MySQL as database system.

3.4. Testing

To evaluate our methodology, we conduct testing using training and testing dataset tailored with Stratified Random Sampling and K-Fold Cross Validation. To measure accuracy for Naive Bayes Classifier, we use formula as follow:

$$Accuracy = \frac{correct\ prediction}{number\ of\ data} * 100\% \qquad (5)$$

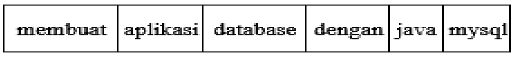
4. Results

4.1. Classification Result

This is the result of a scenario in implementing Naive Bayes Classifier to classify a string of title “Membuat Aplikasi Database Dengan Java & MySQL”.

1. Case Folding and Tokenizing

This process switch all case to lowercase and clean the string other than number and character. The result of this process can be seen in Figure 4.

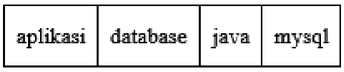


membuat	aplikasi	database	dengan	java	mysql

Fig. 4: Case Folding and Tokenizing

2. Stop Word removal and Stemming

The result of this process can be seen in Figure 5



aplikasi	database	java	mysql

Fig. 5: Stop Word removal and Stemming

3. Calculating $P(H_i)$

The calculation results of $P(H_i)$ are the same across all category labels, this is because the amount of records at training dataset are the same for each category label (30 record each category label).

Table 4: $P(H_i)$

P-Category	$P(H_i)$
P(Programming)	0.125
P(Networking)	0.125
P(Soft Computing)	0.125
P(Mobile Programming)	0.125
P(Information System)	0.125
P(Operating System)	0.125

4. Calculating $P(F_j \mid H_i)$

$P(F_j \mid H_i)$ calculated for each category label. Table 5 below shows the results.

Table 5: P $P(F_j \mid H_i)$

Category	log $P(H_i)$	aplikasi	database	java	mysql
Programming	-2.08	-3.50	-6.49	-3.78	-5.80
Networking	-2.08	-3.50	-6.49	-6.49	-6.49
Multimedia	-2.08	-6.49	-6.49	-6.49	-6.49
Soft Computing	-2.08	-4.70	-6.49	-6.49	-6.49
Mobile Programming	-2.08	-3.55	-6.49	-5.39	-6.49
Information System	-2.08	-4.55	-6.49	-5.80	-6.49
Database	-2.08	-3.50	-4.09	-6.49	-3.93
Operating System	-2.08	-6.49	-6.49	-6.49	-6.49

5. Calculating HMAP

H^{MAP} calculated by adding log $P(H_i)$ with $P(F_j \mid H_i)$ for each category label. Table 6 shows the result of the calculations.

Table 6: H^{MAP}

Category	log $P(H_i)$	aplikasi	database	java	mysql	H^{Map}
Programming	-2.08	-3.50	-6.49	-3.78	-5.80	-21.65
Networking	-2.08	-3.50	-6.49	-6.49	-6.49	-25.05
Multimedia	-2.08	-6.49	-6.49	-6.49	-6.49	-28.05
Soft Computing	-2.08	-4.70	-6.49	-6.49	-6.49	-26.26
Mobile Programming	-2.08	-3.55	-6.49	-5.39	-6.49	-24.01
Information System	-2.08	-4.55	-6.49	-5.80	-6.49	-25.41
Database	-2.08	-3.50	-4.09	-6.49	-3.93	-20.09
Operating System	-2.08	-6.49	-6.49	-6.49	-6.49	-28.05

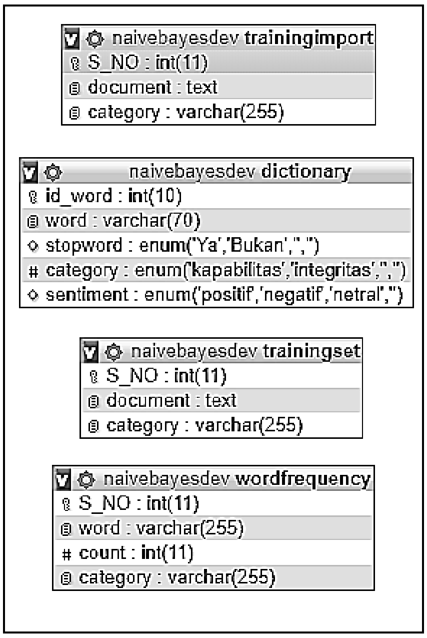
6. Classifying result

The result of the classifying process decided by picking the biggest HMAP score for each category label. In this case, the biggest score is Database with -20.09 score.

4.2. Web Apllication Design

We design the web application to implement the methodology. The following is database design, and sitemap.

1. Database Design



naivebayesdev trainingimport
S_NO : int(11)
document : text
category : varchar(255)

naivebayesdev dictionary
id_word : int(10)
word : varchar(70)
stopword : enum('Ya','Bukan','','')
category : enum('kapabilitas','integritas','','')
sentiment : enum('positif','negatif','netral','')

naivebayesdev trainingset
S_NO : int(11)
document : text
category : varchar(255)

naivebayesdev wordfrequency
S_NO : int(11)
word : varchar(255)
count : int(11)
category : varchar(255)

Fig. 6: Database Design

2. Application Sitemap

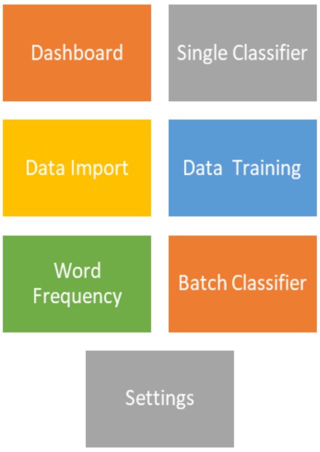
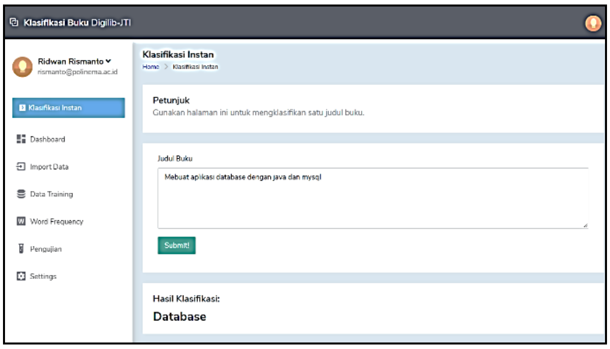


Fig. 7: Application Sitemap

4.3. Web Application Implementation



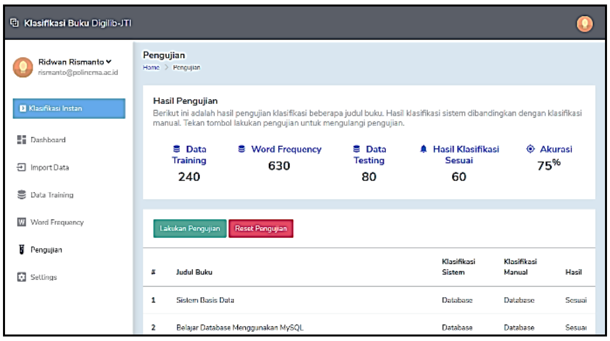
Klasifikasi Instan

Petunjuk
Gunakan halaman ini untuk mengklasifikan satu judul buku.

Judul Buku
Mebuat aplikasi database dengan java dan mysql

Hasil Klasifikasi:
Database

Fig. 8: Single Classifier Page



Hasil Pengujian

Berikut ini adalah hasil pengujian klasifikasi beberapa judul buku. Hasil klasifikasi sistem dibandingkan dengan klasifikasi manual. Tekan tombol lakukan pengujian untuk mengulangi pengujian.

Data Training	Word Frequency	Data Testing	Hasil Klasifikasi Sesuai	Akurasi
240	630	80	60	75%

#	Judul Buku	Klasifikasi Sistem	Klasifikasi Manual	Hasil
1	Sistem Basis Data	Database	Database	Sesuai
2	Belajar Database Menggunakan MySQL	Database	Database	Sesuai

Fig. 9: Batch Classifier Page

Using PHP and MySQL database, we implement the classifying methods for training and testing. Figure 8 and 9 are screenshots of single classifier page and batch classifier page. In single classifier page, we can do classifying one book title, inserted into the provided textarea. In batch classifier page, we can do batch classification by importing data from a spreadsheet.

5.4. Holdout Testing

Holdout or Stratified Random Sampling testing are used to evaluate the accuracy of the application and methodology. Training dataset of 240 records were imported into the application. And testing dataset of 80 records were also imported. Next step is execute the training process. After that we conduct testing. Last step is calculate the accuracy. The following are testing result:

- Training Dataset : 240
- Word Frequency : 630
- Testing Dataset : 80
- Correct : 60

Accuration: (60/80*100%) = 75%

5.5. K-Fold Cross Validation

K-Fold Cross Validation performed by distributing dataset into k-group. Here we use k with value 5. We distribute the records for each folds into training set and testing set evenly. Then perform the training and testing for each group (folds). The following are the dataset distribution:

- Data Set : 320
- K : 5
- Training : 256
- Testing : 64

Accuration for this testing methodology are obtained by calculating average accuracy for each folds. The following are the result of this evaluation:

Table 7: K-Fold Cross Validation Results

Folds:	fold 1	fold 2	fold 3	fold 4	fold 5
Training dataset	256	256	256	256	256
Word frequency	653	691	660	666	669
Testing dataset	64	64	64	64	64
Correct	48	46	44	32	42
Accuracy (%)	75.00	71.88	68.75	50.00	65.63
Train time	45.58	46.32	46.05	44.71	47.16
Testing time	3.70	3.58	3.64	3.58	3.67

Accuracy: (75+71.88+68.75+50+65.63)/5 = 66.25%

The following Figure 10 and 11 are chart representations of accuration and word frequency as compared for each folds:

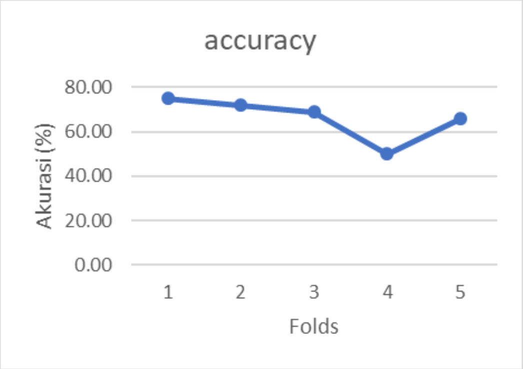


Fig. 10: Accuration Chart

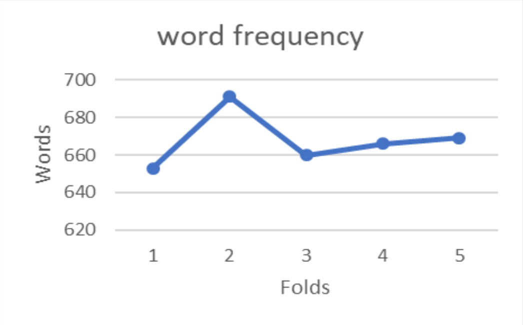


Fig. 11: Word Frequency Chart

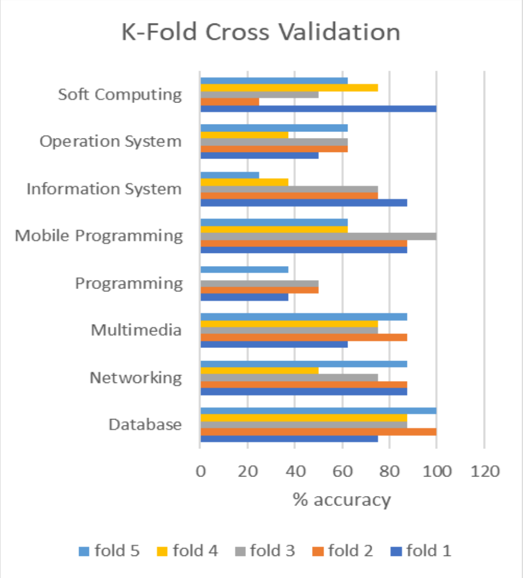


Fig. 12: % accuracy for Each Folds Chart

Table 8: % accuracy of each folds

Category Label	% accuracy					
	fold 1	fold 2	fold 3	fold 4	fold 5	Average
Database	75	100	87.5	87.5	100	90
Networking	87.5	87.5	75	50	87.5	77.5
Multimedia	62.5	87.5	75	75	87.5	77.5
Programming	37.5	50	50	0	37.5	35
Mobile Programming	87.5	87.5	100	62.5	62.5	80
Information System	87.5	75	75	37.5	25	60
Operation System	50	62.5	62.5	37.5	62.5	55
Soft Computing	100	25	50	75	62.5	62.5

Table 9: Classification Result of Programming Category at Fold 4

Book title	Actual Label	System Label
Belajar Sendiri Pasti Bisa Pemrograman C++	Programming	Database
Visual Basic & Microsoft SQL Server	Programming	Database
Menguasai CSS	Programming	Database
Menguasai Java 2 & Object Oriented	Programming	Database
Cara Mudah dan Cepat Belajar Pemrograman C# .NET	Programming	Mobile Programming
Kupas tuntas Adobe Dreamweaver CS6 dengan Pemrograman Php & MySQL	Programming	Database
PHP Source Code	Programming	Mobile Programming
Boom Visual Basic .NET 2010 Meledak	Programming	Database

According to the charts represented in Figure 10 and 11, there are no correlation between word frequency and accuracy. At the third fold, accuracy sits at 68.75% while word frequency at 660. At the fourth fold, word frequency increased to 666 but accuracy decreased to 50%.

Further investigation at K-Fold Cross Validation are performed. We analyze the accuracy for each category labels at each folds. The results are shown by Table 8. The lowest accuracy of category label are Programming (35% average at all folds), while the highest is Database (90% average at all folds). The pattern from Table 8 represented in Figure 12.

From this results, we conclude that the classification accuracy depends highly on the variation of the kind of words contained in the book title, both at training and testing. For example at fold 4, the Programming category has zero correct classification result. Investigation shown that this category often mis-classified as Database category, and the rest is Mobile Programming category as shown in Table 9.

This is caused by at that folds, there are not enough words in testing data that exists in training data for Programming category. The other factor, many words at the testing dataset for Programming category exist in training dataset for Database and Mobile Programming category instead of in training dataset for Programming category.

6. Conclusion

The results of this research show that Naïve Bayes Classifier (NBC) could be used to classify category of a book by its title automatically. The Log Probabilistic have implemented to correct floating number problem which could be too small if the text being classified were too big.

In a scenario which is using 240 records of training dataset and 80 records of testing dataset with 8 category label (30 records of training dataset for each category and 10 records of testing dataset for each category), training process results in 630 records of word frequency using Holdout methodology, with 60 records of correct category label being tested.

System accuration calculated by dividing the amount of correct category label classified by system with the total amount of testing data multiplied by 100%: (60 / 80 * 100%) = 75%. However, evaluation using K-Fold Cross Validation using k = 5 yields in accuration average 66.25%.

The varying results from Holdout and K-Fold Cross Validation are mainly caused by different kind of words at training and testing dataset. To overcome this situation, more training data with more variation of vocabulary will improve the accuracy of the classifier.

Further improvement of this research can be obtained by carefully picking the amount of training dataset. The more training dataset available, the more accuration level be obtained. Further work can be done by applying smothing algorithm other than Laplacian Smooting, for example JK Smoothing, Dirichlet Smoothing, Two-Stage Smoothing and Absolute Discounting.

References

[1] Dewey, Melvil (2004), A Classification and Subject Index for Cataloguing and Arranging the Books and Pamphlets of a Library [Dewey Decimal Classification]. Project Gutenberg.

[2] Pandu Kusuma, Abdi & Srirahayu, Ida (2016), Sistem Pencarian Katalog Buku Menggunakan Metode Naïve Bayes Classifier (NBC) Pada Aplikasi Mulia-Bookstore Berbasis Android. Antivirus: Jurnal Ilmiah dan Teknik Informatika. Vol. 10 No. 2 November 2016.

[3] Frilsilya, Aisya & Yunanto, Wawan & Diah Kesuma Wardhani, Kartika (2016), Klasifikasi Kompetensi Tugas Akhir Secara Otomatis Berdasarkan Deskripsi Singkat Menggunakan Perbandingan Algoritma K-NN dan Naive Bayes. Vol. 5, No. 1.

[4] Indranandita, Ainalia & Susanto, Budi & Rachmat, Antonius (2008), Sistem Klasifikasi Dan Pencarian Jurnal Dengan Menggunakan Metode Naïve Bayes Dan Vector Space Model. Jurnal Informatika. 4. 10.21460/inf.2008.42.48.

[5] Kurniawan, Bambang (2012), Klasifikasi Konten Berita Dengan Metode TextMining. Jurnal Dunia Teknologi Informasi. USU.

[6] Cahyanti, Apriliya Fitri (2015), Penentuan Model Terbaik pada Metode Naive Bayes Classifier dalam Menentukan Status Gizi Balita dengan Mempertimbangkan Independensi Parameter. Jurnal ITSMART.

[7] Nugroho, Bhuono Agung (2005), Strategi Jitu Memilih Metode Statistik Penelitian dengan SPSS. Yogyakarta : Andi.

[8] James, Gareth. Witten, Daniela. Hastie, Trevor. Tibshirani, Robert (2013), An Introduction to Statistical Learning: with Applications in R. Springer.

[9] Kuhn, Max. Johnson, Kjell (2013), Applied Predictive Modeling. Springer

[10] Tala, F. Z (2003), A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. M.S. thesis. M.Sc. Thesis. Master of Logic Project. Institute for Logic, Language and Computation. Universiteti van Amsterdam The Netherlands.

[11] B. A. A. Nazief and M. Adriani (1996), Confix-stripping: Approach to stemming algorithm for Bahasa Indonesia. Internal publication, Faculty of Computer Science, University of Indonesia, Depok, Jakarta.

[12] Hooshyar, D., Ahmad, R. B., Yousefi, M., Fathi, M., Horng, S. J., & Lim, H. (2016). Applying an online game-based formative assessment in a flowchart-based intelligent tutoring system for improving problem-solving skills. Computers & Education, 94, 18-36.