

Maintenance Web Based Applications Using Feature Location

Achmad Arwan¹, Denny Sagita Rusdianto²

^{1,2}Informatic Engineering, Faculty of Computer Science, Brawijaya University
{¹arwan@uib.ac.id, ²denny.sagita@uib.ac.id}

Received 10 January 2019; accepted 23 March 2020

Abstract Maintenance web applications are a complex set of efforts. The FilkomApps are the web application used by the Faculty of Computer Science of Universitas Brawijaya to arrange the academic, theses of students, assignment of faculty, inventory, presence, honorarium. It has about 28K number of files (HTML, PHP, JS, CSS). The feature location was able to help the maintenance of the web applications by locating specific features on the files. The process comprises of preprocessing (tokenizing, web language syntax removal, splitting, stopword and stemming), indexing (VSM Lucene), and evaluations (precision and recall). The experiments were done by querying the keywords originate from previous maintenance modification effort and feature of a system. The results of precision were 86% and recall were 47%. The precision was better 374% than the conventional method (using the IDE search feature).

Keyword: Feature, PHP, Event Programmers

1. Introduction

Software maintenance is the process to make the software remain operated and perform smoothly. The software might need to be fixed when there was a bug. The software also sometimes has to evolve to face against the change of requirements. There are many jobs that relate to the change of the software. Fixing a bug, add a functionality, modification of a data was some of the common jobs due to change the software. To fix a bug, adding a feature, or modify, the programmers/analyst must have good knowledge to which area of code has a correlation with the specific feature/bugs. The documentation of requirements, design of software or database were able to used as a search based to locate the area where bug arise. Unfortunately, many projects have very limited documentation for many reasons. Feature location (FL) was the name of the set of the activity to locate a specific feature on software code.

Many research had done to address the FL problem. Some research using a fragment model to find the feature which tackled model driven development [1]. Another FL research used crowd-based screen cast to help programmers find a sample of code from video tutorial [2]. Many others were used information retrieval (IR) to discover FL within a source code [3]; [4–7]. Most of its used Java-based code as data sets of their experiment. Their experiment results were very impressive. Unfortunately, there was so little FL research using PHP language as data set. The PHP allows a free approach to developing such as Structural Approach (SA), Object-Oriented (OO), or might event hybrid (SA+OO). The PHP usually combined with JavaScript, CSS to make the web

looks great and better usability.

The PHP was the most popular server-side programming language over the world [8]. The Faculties on University of Brawijaya used web-based open source technology to create web application. Based on our interview, most of them use PHP as the base language for their web applications. Their web-based applications usually manage by the unit called TIK. The TIK contains of people who work as Programmers, DevOps, Network Administrator. They are working together to create, maintain, change their web-based applications. Sometimes for a big scope system, they were not build the applications by themselves but outsource to the third party. The personnel of TIK sometimes are not permanent due to career or might resign. The new personnel might face difficulty to maintain, fix a bug, or add a feature to the web applications. As result, the systems might buggy, slow, or might stop working. The simple but expensive approach way to handle this is by creating a new application which has more complete feature from different third party.

The FilkomApps[9] is the application used by the Faculty of Computer Science of Universitas Brawijaya to arrange the academic, theses of students, assignment of faculty, inventory, presence, honorarium. It is developed using PHP language combined with many Javascript, CSS. It has about \pm 6K number of files (HTML, PHP, JS, CSS). The new TIK personnel sometime have difficulties to manage the change, fix bug the applications.

Razzaq et al have researched how FL techniques which might best determine the feature locations using many techniques [7]. Their techniques used the Java open source projects the measure which technique has the best performance. The techniques that they used were VSM_Lucene , LSI, and LDA. As a results, they have concluded that VSM_Lucene has the best performance amongs all.

This research has addressed how Razzaq's finding was able to be used in discovering FL in PHP based projects. The PHP has different characteristics with Java. The PHP is interpreted not compiled like Java. The PHP is much more flexible, which tend less structured rather than Java. The PHP is usually also needed other script like Html, Javascript, and CSS.

2. Research Method

To accomplish the goals, the methods used in this study are illustrated in figure 1. There are several processes which were used on this research, the process explained in section 2.1 to 2.5.

2.1. Dataset Selections

Dataset selections were process to define which file have to be chosen as data. The FilkomApps source codes were read and save into table in Mysql database. It has about \pm 6K number of files (HTML, PHP, JS, CSS) within the FilkomApps. Some of the files were excluded from dataset. It was setting files, configuration, fonts, icon, css, js, etc. The data then save into the table including attribute name of a file, the full path of a file, and code. After the removal process, the data left were about \pm 2K.

2.2. Data Preprocessing

Data preprocessing was the stage to convert code into the stemmed version of code. The purpose of this stage was to split, remove non-PHP, shaping the identifier

into the simple (stemmed) word. The detail process of preprocessing depicted in the sections below.

2.2.1. Strip Tag Removal

Strip tag removal was the process of removing symbols shown after the previous process. The process was made by employing Regular Expression which matches to specific symbols such as “<, >, /, -, (,)”. The results would be saved into a database in the field named nonstrigtags.

2.2.2. Web Language Removal

Web language tag removal was the process to pull out the keyword based on PHP, Html, Javascript, Css, Sql from the documents. The keywords of PHP were grabbed from Php.manual and many resources. It was about \pm 11K keywords. The keywords of Javascript, Html, and CSS were also eliminated from the documents. It was about 200 keywords. The keywords of SQL which based on MySql were also removed from documents. It was about 825 keywords. After the web language removal, the documents contain of identifier which originally created by programmers. This approach was intended to minimize redundancy of keywords.

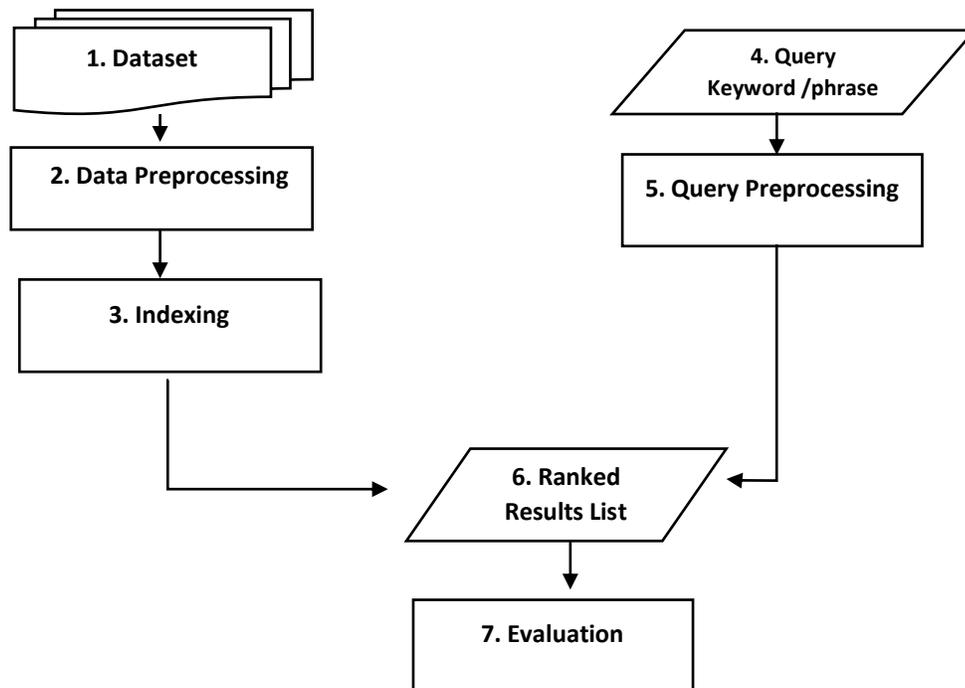


Figure 1 Research Methodology

2.2.3. Stop Word Removal

Stop word removal was the step to remove the common words in English, Bahasa Indonesia and common term that used on domain FilkomApps. The Bahasa Indonesia stop word was removed since the programmer might be using Bahasa Indonesia to express the identifier name. The FilkomApp common term was also removed to minimize the number occurrence of the terms.

2.2.4. Stemming

Stemming was the step to get the base form of words like “Articles” into “articl”. This process was to ensure all the words have the same form. The process was done by using a porter stemmer. The results of the process would be used on further process.

2.3. Indexing

The indexing was the primer process after the preprocessing phase. The indexing process was done by employing VSM Lucene since it recommended by Razzaq’s. According to Razzaq’s, the VSM Lucene performed much better than any other methods such as LDA[10], LSI[11]. The VSM Lucene was performed using apache solr[12]. The documents/files would be extracted word by word to form the index.

2.4. Query Phrase/Keyword

The query was the process to find a feature location using specific words. The words given by the user have also preprocessed using the same process from the previous phase (Html tag removal, Strip tag removal, Web Language removal, Splitting Identifier, Stop word removal and stemming). This process intended to ensure that the words given by the user have the same treatment with a dataset.

The query is given by the user based on the feature or the change that has done by the programmer to modify the feature of the dataset (FilkomApps). The number of the query was limited since its based on their previous maintenance efforts. The words that used as query were depicted in table 1.

Table 1 Query of features

#	Keywords	Type of query
1	Siet Event	Modification
2	Event Sentrin	Modification
3	Seminar hasil	Feature
4	Approve Kjfd	Feature
5	LogBook skripsi	Feature
6	Notifikasi	Feature
7	Registrasi Peserta	Modification
8	Verifikasi Peserta	Modification
9	Invoice	Modification

2.5. Ranking

To find the relevant documents/files that contain the specific feature, the ranking was done by employing VSM Lucene. The ranking is given by calculating the query against the index using score equation 1 [12].

$$Score(q, d) = coord(q, d) * queryNorm(q) * \sum_{t \text{ in } q} (tf(t \text{ in } d) * idf(t^2) * t.getboost() * norm t, d) \quad (I)$$

Formula description

coord(q,d) : The function to determine the number of query terms from documents.

queryNorm(q): This function used to make scores between queries comparable.

tf(t in d) : Show the number of items term occurrence in the documents.

idf(t): Was the inverse of document frequency.

t.getBoost(): The search time boost of term t in query q as specified in the query syntax.

norms(t,d): This function is a product of index time boosts and the length normalization factor.

2.6. Evaluation

The final stage was the evaluation. To measure the preciseness of the feature locations recommendation, we used precision and recall as the general evaluation method of information retrieval (see equation II & III)[13]. To define the relevant items, we were helped by the expert (the programmer of FilkomApps). The evaluation was done by measuring the recommendation ranking result against the results from an expert (the programmer of FilkomApps).

$$Precision = \frac{\# (relevant \ items \ retrieved)}{\# (retrieved \ items)} \quad (II)$$

$$Recall = \frac{\# (relevant \ items \ retrieved)}{\# (relevant \ items)} \quad (II)$$

Formula description:

#(Relevant items retrieved) : It was the number of items recommended by system which were correct according to the expert.

#(Retrieved items) : It was the number of items recommended by system.

#(Relevant items) : It was the number of items that were correct according the expert whether it is recommended or not.

3. Results and Analysis

The series of experiments were conducted to find out the successful method. The challenge in the experiments was how to choose the right keywords. When the keywords are given are too few, it will produce answers that have low precision and high recall. After the environment has set up, the experiments were run using several keywords. The keywords were got from the previous maintenance effort which was done by a programmer of the FilkomApps and also some of the features provided by Filkomapps. The experiments were depicted in table 2.

As a comparison, we also measure the same query using conventional method. The method was querying the same keywords into the search menu on IDE (Visual Studio Code). This method was chosen since it is a very common activity among the programmers. The search results were measured using precision and recall. The comparison experiment results were depicted in table 3.

Table 2 VSM Lucene Experiments Results

#	Keywords	Type	# Files retrieved	# Relevant retrieved files	#Relevant files
1	Siet Event	Modification	2	2	26
2	Event Sentrin	Modification	10	10	26
3	Seminar hasil	Feature	9	9	18
4	Approve Kjfd	Feature	3	1	4
5	LogBook skripsi	Feature	10	6	7
6	Notifikasi	Feature	10	5	10
7	Registrasi Peserta	Modification	5	4	5
8	Verifikasi Peserta	Modification	2	2	4
9	Invoice	Modification	10	9	11
			61	53	111

$$Precision = \frac{53}{61} = 86 \%$$

$$Recall = \frac{53}{111} = 46 \%$$

In table 2, field “# Files retrieved” mean the number of files displayed as a result of searching using keyword given in field “Keyword”. The Field “# Relevant retrieved files” mean the number of files that were displayed and relevant(correct files). The field “#Relevant files” was the number of files which were relevant/correct to keyword according to expert(programmer of FilkomApps).

According to experiment results, our technique was able to found 53 relevant items/files from 66 retrieved items/files. It achieved 86% of the precision score. Besides, our technique also was able to found 53 relevant items/files from 111 relevant files. The score of recall was 46 %.

We also conduct the experiments using a conventional methods as mention previous. The queries given were also the same as the previous experiment. The comparison experiment was depicted in table 3.

Table 3 Conventional Experiment Results

#	Keywords	Type	# Files retrieved	# Relevant retrieved files	#Relevant files
1	Siet Event	Modification	1	0	26
2	Event Sentrin	Modification	69	10	26
3	Seminar hasil	Feature	5	9	18
4	Approve Kjfd	Feature	0	0	6
5	LogBook skripsi	Feature	50	9	8
6	Notifikasi	Feature	34	5	10
7	Registrasi Peserta	Modification	0	0	5
8	Verifikasi Peserta	Modification	0	0	4
9	Invoice	Modification	23	9	11
			182	42	114

$$Precision = \frac{42}{182} = 23 \%$$

$$Recall = \frac{42}{114} = 37 \%$$

Based on the conventional experiment results, it was able to found 42 relevant items/files from 182 retrieved items/files. Also, the conventional also was able to found 42 relevant items/files from 111 relevant files. The conventional methods gave the results of precision 23 % and the recall 37%.

According to the results of the experiments, our method performs precision much better than the conventional method. The success ratio of accuracy was about 374% (86%:23%). The conventional methods achieve the value because the IDE search feature does not separate words like the camelCase methods.

The recall measurement results show that the value was not too significantly different. The recall ratio between our technique and the conventional was losing about 20% (46%:37%). The recall ratio results show the conventional was better than our technique did. It happened since the conventional method was not recommended any documents (see table 3, experiments number 4,7,8).

4. Conclusion

After conducted the set of experiments, finally we can conclude that our method was able to reach the precision 86% and the recall 46%. It performs 374% better precision than the conventional method. Contrarily our methods perform 20% worse than the conventional method.

In the future, the feature location can be implemented by creating a plugin on the IDE as a plugin. We also might use the other technique like a semantic or better statistical method.

References

1. Pérez F, Lapeña R, Font J, Cetina C (2018) Fragment retrieval on models for model maintenance : Applying a multi- objective perspective to an industrial case study. *Inf Softw Technol* 103:188–201 . doi: 10.1016/j.infsof.2018.06.017
2. Moslehi P, Adams B (2018) Feature Location using Crowd-based Screencasts Feature Location using Crowd-based Screencasts
3. Damevski K, Shepherd D, Pollock L (2016) A field study of how developers locate features in source code. 724–747 . doi: 10.1007/s10664-015-9373-9
4. Martinez J (2018) Feature location benchmark for extractive software product line adoption research using realistic and synthetic Eclipse variants. 104:46–59 . doi: 10.1016/j.infsof.2018.07.005
5. Kraft NA, Gray J (2018) Impact of structural weighting on a latent Dirichlet allocation – based feature location technique. 1–25 . doi: 10.1002/smr.1892
6. Biggers LR, Bocovich C, Capshaw R, Eddy BP, Etkorn LH, Kraft NA (2014) Configuring latent Dirichlet allocation based feature location
7. Razzaq A, Le A, Chris G, Jim E (2019) An empirical assessment of baseline feature location techniques
8. <https://w3techs.com/> (2020) Usage statistics of server-side programming languages for websites. https://w3techs.com/technologies/overview/programming_language
9. <http://filkom.ub.ac.id/> (2020) Filkom.ub.ac.id. <http://filkom.ub.ac.id/>
10. Arwan A, Rochimah S, Akbar RJ (2015) Source Code Retrieval on StackOverflow Using LDA. 2015 3rd Int Conf Inf Commun Technol 295–299 . doi: 10.1109/ICoICT.2015.7231439
11. Zhang W, Yoshida T, Tang X (2011) A comparative study of TF*IDF, LSI and multi- words for text classification. *Expert Syst Appl* 38:2758–2765 . doi: 10.1016/j.eswa.2010.08.066
12. Tobergte DR, Curtis S (2013) Apache Solr Search Patterns
13. Manning, Christopher D., Prabhakar Raghavan and HS (2008) Introduction to information retrieval. Vol. 1. No. 1