International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Measurements of Application Revision Quality at Kesharlindung Dikmen Application Website of the Ministry of Education and Culture Using Automated Software Testing

Dakroni[1], Karmilasari[2]
[1]Kesharlindung for Secondary Education, Ministry of Education and Culture, Jakarta, DKI Jakarta, Indonesia-12190
[2]Business Information System, Universitas Gunadarma, Depok, West Java, Indonesia-16424

***Abstract*—** *SIMLKTI is an Information System on Scientific Paper Competition applied at the Sub-Directorate of Kesharlindung for Secondary Education Teachers, Ministry of Education and Culture, specifically to handle scientific paper competitions from the stages of registration, administrative selection, paper assessments, determining the finalists, and the main winners that are held annually and aimed specifically for teachers of Senior High School/Vocational School and their equivalent. The aim of this research is to measure the revision quality at SIMLKTI using automated software testing of PHP Metrics. The measurement results from a module on this application website shows that the used methods and automated software testing reveal the true results. The module that shows these results is the Activity module. The measurement results of the Activity module to measure the maintainability factor by using the results of the measurement of maintainability index are 60, 16. These results indicate that the Activity module has low maintainability status, has critical and fatal technical errors. While the Flexibility measurement results are measured using the modularity value with the average afferent coupling of 1.14, efferent coupling 3.57, with an average coupling of 2.36 and LCOM 2.77, which means that the module is in poor condition in terms of flexibility due to the difference between the cohesion and coupling that is only 0.41. On the other hand, the testability measured by the simplicity value gets the value of Cyclomatic Complexity of 10.46, which means that the Activity module is a module that is difficult to understand, difficult to find errors and tends to complicate in the testing process. Measurement of SIMLKTI application site is mandatory to be carried out on other modules, in order to obtain comprehensive measurement results and the followed-up can be used as a reference to make improvements.*

***Keywords*—** *Revision quality factor, automated software testing, PhpMetrics, maintainability, cohesion, coupling, cyclomatic complexity.*

I. PREFACE

The Ministry of Education and Culture as the regulator in the field of education continues to emphasize the importance of a positive internet in the field of education, both in basic and higher educations. And this is evidenced by the creation of the Information System on Scientific Paper Competition (SIMLKTI) by the Directorate for Welfare, Awards and Protection for Secondary Education Teachers (Kesharlindung Dikmen). The launching of this web-based application site coincides with the National Teacher's Day on November 24, 2015, wherein the Ministry of Education and Culture by means of the Directorate General of Teachers and Education Personnel held the celebrations just as in previous years, but there was something different that day with the presence of the finalists of the scientific paper competition in the framework of the 2015 National Teachers' Day (HGN), opened two months before the 2015 HGN celebration [1].

SIMLKTI was built without careful design and it makes the system riddles with many weaknesses in the development, addition of features, and maintenance. In the 2016 fiscal year, which was first year after the 2015 HGN, many features were added until finally the SIMLKTI in terms of coding compilation could no longer be developed. So Kesharlindung Dikmen was forced to rebuild from scratch the SIMLKTI program code with the same and more dynamic features.

One feature very oftenly used on the SIMLKTI application site is the Activity feature. This feature is used to manage scientific paper competition activities.

As a system used at the national level, and in order to avoid recurring the 2015 events, SIMLKTI is obliged to measure its quality, especially quality in terms of measurements related to revision factors in the Activity features.

Based on this background, measuring the quality of the SIMLKTI application site focused on one of the three McCall's Model Factors namely the quality of the revision factor consisting of Maintainability, Flexibility, Testability [2] to measure the Activity features, is the main objective of writing this paper.

II. THEORY FOUNDATION

A. Software Metrics

Software metrics can be defined as continuous measurements for the process of software development, product supply, and time management information to improve the quality of processes and software products. Software metrics are determined by measuring the properties and specifications of the software [3].

Measurements can be used on all software projects to help estimate, control quality, assess productivity, control projects, help assess product quality and assist in decision making on project outcomes [4].

Dakroni and Karmilasari, "Measurements of Application Revision Quality at Kesharlindung Dikmen Application Website of the Ministry of Education and Culture Using Automated Software Testing," *International Research Journal of Advanced Engineering and Science*, Volume 5, Issue 2, pp. 14-19, 2020.

B. *Maintainability Index*

Maintainability Index is a software metric that measures software that is easy or difficult to experience maintenance or changes in the future. Maintainability Index calculates formulas based on Lines of Code (LOC), Cyclomatic Complexity (CC) and Halstead Volume (HV) [5]. The Maintainability Index equation is shown in the formula:

MI=171-5.2 x ln(HV)-0.23 x (CC)-16.2 x ln(LOC)

The derived formula used by Phpmetrics

MI=171-5.2 x ln(HV)-0.23 x (CC)-16.2 x ln(LOC)+50 x sin(sqrt(2.4 x CLOC/LOC))

With:

MI = *Maintainability Index*
HV = *Halstead Volume*
CC = *Cyclomatic Complexity*
LOC = *Line of code*
CLOC = *Comment Lines of Code*

and the Maintainability Index classification is shown in Table I.

TABLE I. Classification of Maintainability Index

Value of Maintainability Index	Classification
MI < 64	Low maintainability status, project has a critical or fatal technical error.
65-84	Maintainability status, there are problems / problems with the project but not too serious.
MI > 85	High maintainability status, project included in good condition.

C. *Flexibility*

Flexibility in software is a property of software that defines how easily a software is changed without damage [6]. Measuring the flexibility of a software can be done by measuring modularity and simplicity. Modularity is determined by cohesion and coupling. Cohesion shows the relationship between functions that exist in one module, while coupling shows the dependence of a module on other modules. Software is said to have good modularity if it has high cohesion and low coupling [7].

- Coupling between object (CBO)
  CBO is defined for a class, i.e. the number of classes installed by the measured class. A class A is said to be installed with class B if class A uses a method or variable variable in class B [8]. Excessive CBO is a destroyer of modular design and reduces reuse. The less dependent a class is with another class, the easier the class is to be reused for other applications [9]. In addition, a class with a large CBO means that the class is more sensitive to changes in its class pair. This resulted in efforts to maintain the class even greater [10].
- Lack of Cohesion in Methode (LCOM)
  Suppose a C1 class with a method M1, M2, ..., Mn. Than {1j} is the set of variable instances used by the Mi method. Thus there are n sets I1, I2, ..., In. then P = {I1, Ij | II n Ij = Q} and Q = {I1, Ij | II n Ij Ø}. If all n sets {I1}, {I2}, ..., {In} is Ø then P = Ø.

$$LCOM = \begin{cases} |P| - |Q|, & |P| > |Q| \\ 0, & Others \end{cases}$$

LCOM is expected to be low in a class (high cohesiveness) because of the increase in encapsulation High LCOM (low cohesiveness) indicates a class that should be split into 2 or more classes.

In addition, high LCOM indicates high complexity [8] [9].

D. *Testability*

Testability is one of the determinants of the quality of a software. The definition of testability in general is how easy it is to do testing on a piece of software. The lower testability of a software indicates the more difficult in testing and maintaining a software.

Testability is an important factor that determines the amount of time and effort needed to do software testing [11]. Factors that affect testability include modifiability, simplicity, understandability, flexibility, complexity, self-descriptiveness and modularity [12].

- *Cyclomatic Complexity (CC)*
  CC, also called McCabe cyclomatic complexity, is used to evaluate the complexity of a method [13]. CC is the number of test classes needed to test methods comprehensively [14]. The calculation can be done by describing the program sequence of a method into a graph with all possible paths. The complexity is calculated by the formula:

$$v(G)=e-n+2$$

With:

v (G) is the cyclomatic complexity for graph G.
e is the number of edges in the graph G, and
n is the number of nodes in graph G.

There are other ways to calculate v (G). Anderson and Vestergren, 2004 formulated it as follows:

$$v(G) = P + 1$$

With,

P is the number of predicate nodes present in graph G.

E. *Phpmetrics*

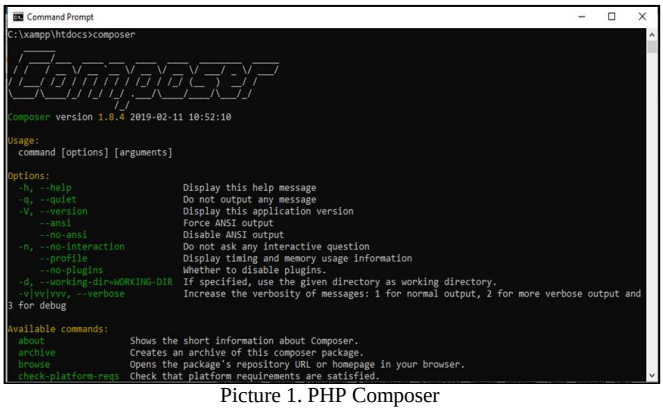
Phpmetric is an automated testing tool that can be used to calculate the quality of software made with the PHP programming language and display the results of these calculations in tables, graphs, and illustrations [15]. PhpMetric is open source and was developed by Jean-François Lépine. Phpmetric has many types of measurement software, some of which are as follows [16]:

- *Maintainability Index*: measure the level of software maintainability
- *Lack of Cohesion Method*: count the number of different methods in a class that uses variables in that class
- *Cyclomatic Complexity*: calculate the complexity of a program by measuring the number of control paths in a module

Installation of PHP Metrics using Composer as shown in the instructions on the official PHP Metrics page https://phpmetrics.org/ [16], as shown in Picture 1.

Dakroni and Karmilasari, "Measurements of Application Revision Quality at Kesharlindung Dikmen Application Website of the Ministry of Education and Culture Using Automated Software Testing," *International Research Journal of Advanced Engineering and Science*, Volume 5, Issue 2, pp. 14-19, 2020.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

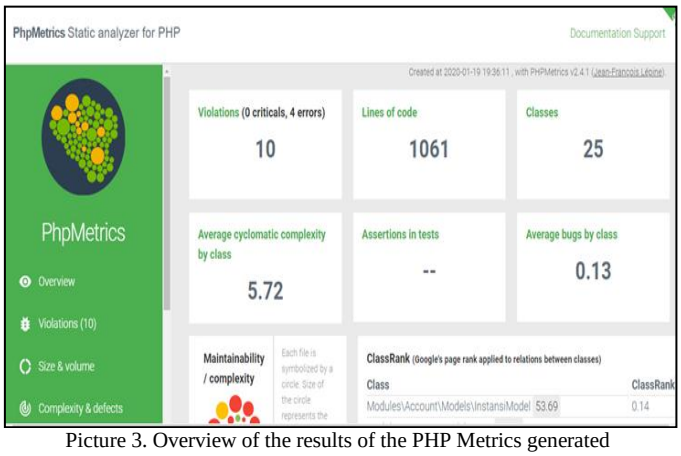


Picture 1. PHP Composer

Meanwhile, to run phpmetric to perform metrics calculations on the application site, using syntax: *phpmetrics--report-html=report_metrics/kesharlindung/report_kesharlindung.html kesharlindung-rebuild/Modules/Account.*
One application of phpmetrics syntax can be seen in Picture 2.

Picture 2. Run PHP metrics by using CMD

The results of the calculation of PHP Metrics calculations can also be viewed using a browser with an attractive and colorful appearance. PHP metrics generated display results if opened through a browser by accessing the specified URL generated, an example of the measurement results can be seen in Picture 3.



Picture 3. Overview of the results of the PHP Metrics generated

III. METHODOLOGY

Measuring maintainability, flexibility, and testability at the SIMLKTI application site through the following stages:

1. Review business processes that run on SIMLKTI
2. Make a copy of the source code SIMLKTI application site to a different personal computer so as not to interfere with the site that is running
3. Analyze the source code that has been copied using phpmetrics. The results of the analysis using phpmetrics are html report file [16], and to generate the report is done by way:
   - Install composer
   - Download phpmetric.phar and save it at a certain location.
   - Type the command at the command prompt as follows:
     Php phpmetric.phar –report html=file_report_name.html location/of/your/sourcecode
   - Reports will be generated and saved automatically in the location where phpmetrics.phar is stored.
4. *Report* results obtained in stage 2 are transferred to Ms. Excel then look for the average value of Maintainability Index, Afferent Coupling, Effrent Coupling, Line of Cohesion Method, Cyclomatic Complexity of each existing php file.

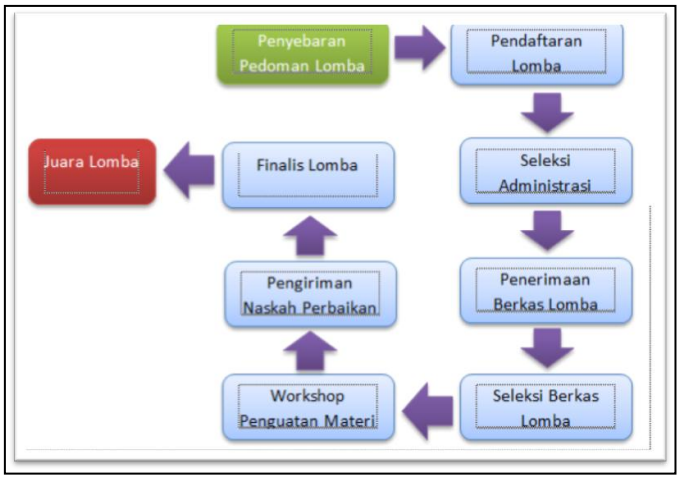
IV. DISCUSSIONS

A. *Review Proses Business*

The outline of the business process of the Information System for Scientific Writing Competition in the Kesharlindung Dikmen Sub-Director is shown in Picture 4. The outline of the business process begins with the distribution of the competition guidelines. Dissemination through the Provincial and Regional Education Offices as well as those available on SIMLKTI. After the distribution of the guidelines, the registration for the competition was opened at SIMLKTI, the distance between the time the distribution of the guidelines and the opening of the registration for the competition was only 7 days. After being registered in the competition, the teacher who has become a participant must then immediately send the administration file. The next step is administration selection by the committee, selection is done by the committee by looking at the administrative files sent by the participants. Participants who pass the administration will be notified on the page and the competition file upload form will appear. Then the first stage of the assessment is conducted by the jury to get participants to be guided in a workshop on strengthening the writing material for scientific papers. The next stage after the evaluation of the competition dossier, the participants who are selected to be participants in the material strengthening workshop will get notification and email to attend the material strengthening workshop. The results of the material strengthening workshop were sending back the repair text according to the reinforcement material in the workshop that had been followed. Participants must submit an improvement document of the results of the workshop, at this stage a second phase of assessment is carried out to get finalist participants. Participants selected as finalists will be invited to compete with one another by displaying posters and presentations of their work. The results of the match at this stage determine the winner participants from the 1,2,3 champions and the champion hopes and ranks based on their value.

Dakroni and Karmilasari, “Measurements of Application Revision Quality at Kesharlindung Dikmen Application Website of the Ministry of Education and Culture Using Automated Software Testing,” *International Research Journal of Advanced Engineering and Science*, Volume 5, Issue 2, pp. 14-19, 2020.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024



Picture 4. Outline of the SIMLKTI Business Process

B. *Review SIMLKTI*

In this SIMLKTI review, researchers divided into 2 parts:

1. *Site Page*,

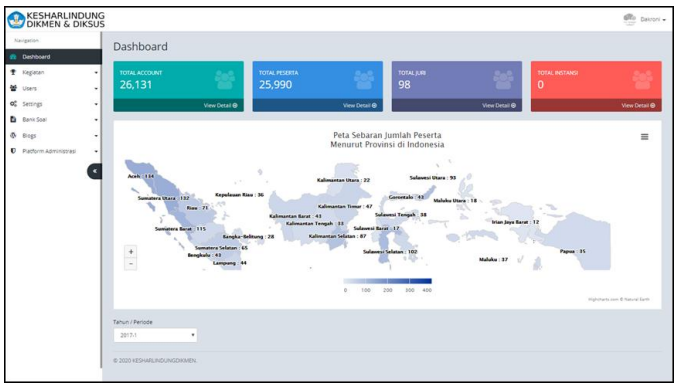
The SIMLKTI website page can be accessed after the user has successfully logged into the application site. Site pages are divided into 3 major sections as shown in Picture 5:

a. *Dashboard*, display the total amount of site user account data which is divided into 4 categories namely Total Accounts, Total Participants, Total Judges, and Total Agencies.

b. *Menu Utama*, display various features of the SIMLKTI application site that is used to maximize the usability of the application site. The main menu can be adjusted according to the user's privacy. Priviledge that can access all features is the Administrator Priviledge, the menus that are displayed are:

i. Kegiatan
ii. *Users*
iii. *Settings*
iv. Bank Soal
v. *Blogs*
vi. *Platform* Administrasi

c. Distribution map, used as a demographic illustration of the number of participants registered as participants of events / activities organized by Kesharlindung.



Picture 5. Main page of application site

2. *Database & Class*

SIM LKTI uses MySQL as its database. Funds are due to using Object Oriented Programming in making its application site. SIMLKTI has classes in its applications.

i. There are 1 database used by SIM LKTI. While the total tables are made so that the application site runs there are 49 database tables.

ii. The SIMLKTI application site has 6 features and each feature has a class. The total classes owned by the application site SIMLKTI 283. Details of each class owned by each feature are shown in table II.

TABLE II. Number of Classes per feature

No	Feature	Number of Class
1	Activity	128
2	Users	34
3	Settings	13
4	Question Bank	26
5	Blogs	35
6	Administration Platform	49

C. *Running PHP Metrics*

To measure the quality of Activity features on SIMLKTI, you must use the syntax command in section III.3, an example of features that can be used for examples is the Activity feature because it has the most classes, to calculate the quality of activity features can use the command:

$phpmetrics --report-html=
C:\xampp\htdocs\report1\kesharlindung\phpmetrics\modul\Kegiatan
C:\xampp\htdocs\kesharlindungrebuild\Modules\Kegiatan

D. *Analysis of Measurement*

Phpmetric measurement results consist of 2 types, namely in the form of diagrams and tables of calculation results. The results of the Activity feature calculation are as follows:

1. *Maintainability*

The results of measuring the maintainability of the Activity features using the results of the Maintainability Index calculation. The results obtained are described in table III, with the conclusion that class with a value of <64 there are 94 classes, classes that have measurements between 65-84 there are 15 classes, while the results of measurements of maintainability> 85 there are 19 classes. Measurement results that are dominated by measurement results <65 will give negative results on the calculation of the average maintainability index. The average result is 60.16 <65. These results incorporate the activity features into a low maintainability status, are considered to have fatal and critical errors.

TABLE III. Measurement of Maintainability Index of Activity Features

Value of Maintainability Index	Number of Class	Classification
MI < 64	94	Low maintainability status,
65-84	15	Maintainability status
MI > 85	19	High maintainability status,
average	**60,16**	Low maintainability status

Dakroni and Karmilasari, "Measurements of Application Revision Quality at Kesharlindung Dikmen Application Website of the Ministry of Education and Culture Using Automated Software Testing," *International Research Journal of Advanced Engineering and Science*, Volume 5, Issue 2, pp. 14-19, 2020.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

2. Flexibility

Modularity measurement results show good results with an indication of the results of coupling measurements <from cohesion measurements. For the average value of afferent coupling measurement is 1.14, while the results of bookkeeping efferent coupling are 3.57. And the average yield of coupling measurements is 2.36. This result is smaller than cohesion with an average value of 2.77. So that the activity module is included in the category of good flexibility, easily changed / carried out repairs or changes if found damage. Modularity measurement results are shown in table IV.

TABLE IV. Measurement of Activity Modularity Features

Category	Number of Class	classification
Coupling < Cohesion	54	Well
Coupling = Cohesion	26	Is
Coupling > Cohesion	48	Not good,
The final result	**Coupling < Cohesion**	**Well**

3. Testability

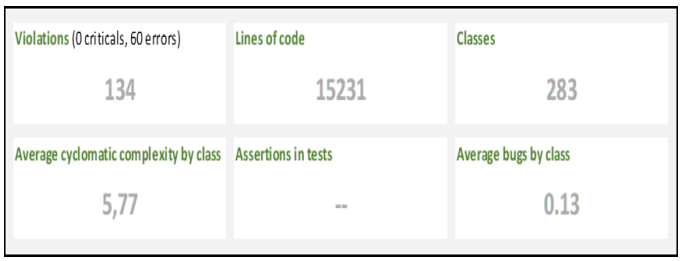
Overall, classes that get measurement results complexity> 10 there are 44 classes, and classes that get results <10 there are 88 classes. The average cyclomatix complexity for the activity module is 10.46. Included in the good category, because it can still be considered low, so this module is easy to understand, easy to find errors and easy to do testing. Complexity measurement results are shown in Table V.

TABLE V. Measurement of Activity Feature Complexity

Ratting	Number of Class	Classification
Complexity > 10	44	Baik
Complexity < 10	88	Tidak Baik
Hasil Akhir	Complexity > 10	**Baik**

4. Measurement of all features

The total class created at the SIMLKTI application site is 283 with a total of 15231 program lines in making this application site. There were 134 found violations with 60 errors. The complexity itself is 5.77 with average bugs 0.13. An overview of measurements is shown in Figure 7.



Picture 7. Overview Site Application SIMLKTI

From the results of measurements on 3 components of the revision quality factors namely maintainability, flexibility and testability shows that the SIMLKTI application site is in good condition to be used and can be used maximally. The items used to measure the index maintainability, modularity and simplicity showed good results with values of 88.06, 1.83 <2.60 and 5.77, respectively.

The feature that shows the best measurement results is the Settings feature, this feature on each measurement result shows a positive result, where the index maintainability measurement has a value of 134.85> 85, cyclomatix complexity 3.67 <10 and the difference between coupling and cohesion 2.33.

There are several features in the application that should get special attention by the application owner. Because the measurement results show negative results. A feature that can be immediately corrected is the activity module. This feature on each measurement result shows a negative result, where the measurement of maintainability index has a value of 60, 16 <65, cyclomatix complexity 10.46> 10 and the difference between coupling and cohesion is 0.42. For other module measurement results that show negative results, improvements can be made in stages so that they can get more measurement results.

The measurement results of all features are explained in table VI.

TABLE VI. Measurement Results of All Features.

No	Fitur	Maintainability Index	Cyclomatix Complexity	Afferent coupling	Efferent coupling	Average Coupling	LCOM
1	Kegiatan	60,16	10,46	1,14	3,57	2,36	2,77
2	Users	70,70	5,72	0,80	3,20	2,00	2,60
3	Settings	134,85	2,67	0,00	2,67	1,33	3,67
4	Bank Soal	67,70	13,71	1,18	3,76	2,47	3,00
5	Blogs	62,35	7,29	1,09	3,43	3,26	3,26
6	Platform Administrasi	491,67	33,29	4,09	18,26	11,17	15,83

V. CONCLUSIONS

In general, the results of measurements on 3 components of the revision quality factors namely maintainability, flexibility and testability show that the SIMLKTI application site is in good condition to be used and can be used to the maximum. The items used to measure the index maintainability, modularity and simplicity showed good results with values of 88.06, 1.83 <2.60 and 5.77, respectively.

One example of detailed measurements using the results of the measurement of the Activity feature, found that the Activity feature is in good condition where the 2 revised quality factors show results which are the Flexibility factor where the results of the coupling measurement <from the cohesion measurement. The average yield of coupling measurements is 2.36. This result is smaller than cohesion with an average value of 2.77. While other revised quality factors that show good results are the Testability factor, namely the measurement results of complexity> 10 there are 44 classes, and classes that get results <10 there are 88 classes. The average cyclomatix complexity for the activity module is 10.46.

Whereas the revised quality factor which shows poor results is the Maintainability factor. The measurement results obtained are measurements that are dominated by values <65, these results will give negative results on the calculation of the average maintainability index, which is 60.16 <65. These results include the Activity feature into a low maintainability status, are considered to have fatal errors and critical.

Dakroni and Karmilasari, "Measurements of Application Revision Quality at Kesharlindung Dikmen Application Website of the Ministry of Education and Culture Using Automated Software Testing," *International Research Journal of Advanced Engineering and Science*, Volume 5, Issue 2, pp. 14-19, 2020.

ACKNOWLEDGMENT

The author would like to thank all those who have supported the implementation of this research. Especially to the Sub-Directorate of Kesharlindung Dikmen Directorate of Teacher Development and Secondary Education Personnel who have provided support and motivation so that this research can be completed well.

REFERENCES

[1] Kementerian Pendidikan & Kebudayaan. (2015). Permendikbud No 11.
[2] McCall, J. A., Richards, P. K., & Walters, G. F. (1977). "Factors in software quality. volume i. concepts and definitions of software quality". GENERAL ELECTRIC CO SUNNYVALE CA.
[3] Debbarma, M. K., Debbarma, S., Debbarma, N., Chakma, K., & Jamatia, A. 2013. A Review and Analysis of Software Complexity Metrics in Structural Testing. International Journal of Computer and Communication Engineering, 129-133.
[4] Susanto, M. I., Darwiyanto, E., & Wisudawan, G. A. 2015. Pengukuran Software Metrics Terhadap Implementasi Framework Laravel pada Pembangunan Aplikasi Berbasis Web . e-Proceeding of Engineering, 7731-7738.
[5] Nađa, Terzimehić. Schneegass, Christina, and Hussmann, Heinrich. (2002). "Reliable Software Technologies — Ada-Europe 2002". Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Vol. 2361. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg.
[6] Shen, L dan Ren, S. (1990). "Analysis and measurement of software flexibility based on flexible points," pp. 331–341.
[7] Ghasemi, M. Sharafi, S.M and Arman, A (2015). "Towards an Analytical Approach to Measure Modularity in Software Architecture Design". J. Softw., vol. 10, no. 4, p. 465
[8] Chidamber, S. R., & Kemerer, C. F. (1994). "A metrics suite for object-oriented design". IEEE Transactions on software engineering, 20(6), 476-493.
[9] Lindroos, Jaana. (2004). "Code and Design Metrics for Object-Oriented Systems". Helsinski: Department of Computer Science, University of Helsinsky.
[10] Systa, T., & Yu, P. (1999). "Using OO Metrics and Rigi to evaluate java software". Tampere: Department of Computer Science, University of Tampere.
[11] Abdullah, Abdullah, Dr. M.H. Khan, Dr. Reena Srivastava (2015). "Flexibility: A Key Factor to Testability". Int. J. Softw. Eng. Appl., vol. 6, no. 1, pp. 89–99.
[12] Khan, M. H and Srivastava, R. (2015) "Testability Measurement Model for Object Oriented Design (TMMOOD)". vol. 7, no. 1, pp. 153–163.
[13] Sarker, Muktamyee. (2005). "An Overview of Object-Oriented Design Metrics [Master Thesis]". Umea: Department of Computer Science, Umea University
[14] Rosenberg, Linda H. (1998). "Applying and Interpreting Object Oriented Metrics".
[15] Aba, P. (2014). "phpmetrics of popular php projects," peteraba.com. [Online]. Available: https://peteraba.com/blog/phpmetrics-ofpopular-projects/. [Accessed: 16-Jun-2019].
[16] Lépine, J.-F. 2015, "Metrics of PhpMetrics," phpmetrics.org, [Online]. Tersedia di http://www.phpmetrics.org/documentation/index.html. [Accessed: 13-Jun-2019].