

Laporan Penelitian



Similarity Checking of Source Code Module Using Running Karp Rabin Greedy String Tiling

Rudi Setiawan dan Umar Al Faruq

ABSTRACT

Similarity checking of source code module, required a long process if it is done manually. Based on that problem, this research designed a software with structure-based approach using string matching technique with Running Karp-Rabin Greedy String Tiling (RKR-GST) Algorithm to check the similarity and using Dice Coefficient method to measure the level of similarity from 2 results source code modules. The result of the experiments show that RKRGST which applied in this system capable of recognizing the changing of statement and the changing statement order, and be able to recognize the syntax procedure testing that has been taken from its comparison module. Modification by adding the comment on source code module and changing of procedure name which is called in body of procedure can also be recognized by system. Processing time needed to produce output depends on the number of program code row that contained in source code module.

RESEARCH OBJECTIVES

The Source Code module that has been uploaded into the system will proceed to *preprocessing process*. Preprocessing stage will cover scanning and parser in order to gain syntax procedure, the next process for each syntax procedure, will need to be filtered; the variable, *stop word* and punctuation filtering, then case folding process is being held by changing all of the character exist in syntax procedure into smaller character. All syntax procedure that has pass preprocessing process will be check based on their syntax procedure similarity existing in source code module and then compare using RKRGST Algorithm. After similarity check has been done, *procedure / function / method* with similarity will appear, then similarity percentage will be counted base on Dice's Coefficient method.

METHODS

Two main approach have been used in source code plagiarisme; *attribute-counting* dan *structure-based* (1). Early generation plagiarisme source code detection is using attribute counting, after that various *attribute-counting method* appear, for example using operator counting and operands by Halstead (2), *cyclomatic complexity method* by McCabe (3) to measure control flow program by counting the number of execution path (). *attribute-counting method*, can't describe program structure well enough to distinguish source code plagiarism result and source code that is not the result of plagiarism (5). Then structure based method is developed to discribe better structure program and resulting in a more complex modification checking.

In the design system, a procedure is consider to have similarity if the marked string number reaching greater than or equal to detected sensitifity values that has been determined by application user.

No Type Modifikasi

- 1 Change the name *procedure / function / method*
- 2 Change the name of variable
- 3 Change the data type variable with oder data type
- 4 Change or adding comment

- 5 Change the call of procedure with changing the procedure name called
- 6 Change order statement

Partial duplicate source code or partial duplicate some procedure contents

How to Handle

Ignore the procedure / function / method
Ignore the variable
Ignore all data type variable
Ignore all comment

Ignore call a procedure

RKRGST Algorithm able to find the same parts, without being affected by the position of the substring position

RESULTS

RKRGST Algorithm that has been implemented in this system proven to be able to handle statement order change and statement change that has been done, and able to handle syntax from test procedure taken from same of the module comparison. Modification by adding comment in module source code and modification of name changing procedure that has been called in the body procedure, proven eligible to be handled by the system.

Filtering variable parameter that has been implemented in the checking process able to manage the possibility of variable name changing and data type changing along with it. The more row code program exist in the source code module, will result in longer time process needed by the system to produce output.

FINDINGS

Running Karp-Rabin Greedy String Tiling is a string matching algorithm, can be used for similarity checking source code program with the steps described in the system flowchart.

DAFTAR PUSTAKA

1. Clough, P., (2000). Plagiarism in natural and programming languages: an overview of current tools and technologies, Department of Computer Science, University of Sheffield.
2. Halstead., (1972). Natural Laws Controlling Algorithm Structure, ACM SIGPLAN Notice 7,2 (February 1972), 19-26.
3. T.J. McCabe, (1976). A Complexity Measure, IEEE Transactions on Software Engineering, Vol.2 No.4, pp 208-320.
4. Verco, K. L., Wise, M. J., (2006). Comparing Structure and Attribute-Counting System, Technical Report, Department of Computer Science, University of Sydney, Australia.
5. Faidhi, J. A., Robinson, S. K., (1987). An empirical approach for detecting program Similarity within a university programming environment, Computers and Education, Vol. 11, No. 1, Pp. 11-19.