Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia - Social and Behavioral Sciences 67 (2012) 114 – 122

Procedia
Social and Behavioral Sciences

The 3rd International Conference on e-Learning
ICEL 2011, 23-24 November 2011, Bandung, Indonesia

Plagiarism Detection System Design for Programming Assignment in Virtual Classroom Based on Moodle

Dewi Tresnawati[a*], Arief Syaichu R[a], Kuspriyanto[a]

[a]Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
Jalan Ganesha No. 10 Bandung 40132, Indonesia

Abstract

The practice of plagiarism is not a strange thing anymore, especially among the students that almost every day working on tasks assigned by the lecturer. The practice of plagiarism is done by the exchange of source code that have been successful. Detecting plagiarism practices is a solution that should be done so that the fraudulent actions can be minimized. Overall, the plagiarism detection software can provide a useful contribution to minimize plagiarism. Using the software can be deterrent for students to take a plagiarism. However, using this software does not provide the final answer. Software only detects and notifies any duplication. Therefore, the intervention of manual inspection and assessment is still needed. The design result obtained by a plagiarism detection system model for programming assignment in a virtual classroom based on Moodle and delivery system alerts for the student. The test results show that the software can detect any similarities in the assignment by generating a percentage of similarity between two files or many files. In addition, the software can deliver alerts from the lecturer for the students that plagiarism is detected using SMS Gateway.

Selection and peer-review under responsibility of i-Learn Centre, Universiti Teknologi MARA, Malaysia

Keywords: plagiarism; plagiarism detection system; virtual classroom; Moodle; SMS Gateway

1. Introduction

Plagiarism is the act of abuse, publication, statement, or claims as your own the thoughts, ideas, writings, or a creature that actually belongs to someone else. To overcome the practice of plagiarism, it is

[*] Corresponding author. *E-mail address*: tresnawatidewi@yahoo.com

Selection and peer-review under responsibility of i-Learn Centre, Universiti Teknologi MARA, Malaysia
doi:10.1016/j.sbspro.2012.11.312

not enough simply to remind students that the act of plagiarism is not well done. Plagiarism detection practice is a solution that should be done so that fraudulent activity can be minimized.

Today has found the software/plagiarism detection tools, including the SIM, SID, MOSS, JPlag, YAP, Plague, Bandit, Cogger, AC and CodeMatch. Use of the software can be a deterrent for students to perform acts of plagiarism. Software only detects and inform the duplication. Therefore the intervention of manual inspection and assessment is still needed

2. Plagiarism Detection

Plagiarism detection system can be developed for: text data such as essays, articles, journals, research and so forth; a more structured text documents such as programming languages (source code).Modifications are usually done by students of programming can be classified as follows.

a. Lexical, changes in the code (source code) program, for example:
- changing comment (increased or reused, or replaced),
- changing format,
- changing variable name.

b. Structural, changing program structure, for example:
- change the order of the algorithm, does not change program course,
- change the procedure becomes a function or vice versa,
- the procedure is replaced with the contents of the procedure itself.

The benefit of plagiarism detection system is to aid the detection of the manual in making comparisons between the amounts of source code in a short time. Major aspects of development that must be considered in plagiarism detection system :

a. suitable discriminator to indicate the presence of plagiarism,
b. make a suitable method for comparing these discriminators,
c. appropriate measurement of similarity.

2.1. Plagiarism detection method

Two main approaches have been used for plagiarism detection methods of attribute-counting and structure-base.In attribute-counting method, which in comparison is a quantitative measure of some of the metrics program, while at structure-based method, which compared the representation of the structure of the program, such as a line of representation of a string, parse tree, data flow, etc.

2.2. Attribute-Counting method

A variety of attribute-counting methods that use other metrics continue to emerge, such as calculation of the number operator and operands by Halstead, cyclomatic complexity method of McAbe which measures the flow of program control by calculating execution path, scope and methods of measurement number. However, attribute counting system only managed to perform effectively for plagiarism detection is done with simple modifications.

2.3. Stucture-Based Method

In general, the detection system in structure-based method is divided in two stages.

- *Tokenization*, the parsing code into a collection of tokens called token sequences or profiles. Token is a single element of the programming language. For example reserved words, punctuation, and operators (Techterms, 2006). And the parser is a program that breaks the code into functional components (George, 2005)

- Comparing each pair of profile or token sequences. For n program that collected the application will do the n*(n-1)/2 comparisons.

2.4. Plagiarism Detection Software at the Source

When this has been a lot of plagiarism detection software developed using structure-based method. *Software Similarity Tester* (SIM). Software Similarity Tester (SIM) plagiarism detection system was developing in 1999 by Gitchell and Tran as a system for measuring the similarity between text written in C, Java, Pascal and natural language.

SIM work steps.

- Read the program files: read the file and store it in sequence.
- Determine the set of interesting runs: the algorithm determines match between two files.
- Determine the line numbers of the interesting runs: finds the start and end line number for each chunk.
- Print the contents of the runs in order: the stored match and display the analysis in chart.

2.5. Measure Of Software Similarity (MOSS)

Measure **o**f **S**oftware **S**imilarity (MOSS) was developed in 1994 by Alex Aiken at Berkeley as a system for measuring the similarity of source code written in C, C++, Java, or Pascal. MOSS tests the source code in real file be parse the source code, tokenizing it and apply comparison algorithm (MOSS) to the tokenized form of the code. And compare it with the source code in other files.

2.6. Jplag

JPLag built and developed by Guido Malpohl from the Department of Informatics University of Karlsruhe. The system can detect the similarity between source Java, C, C++ and Scheme. Jplag available as a web service and can be used free of charge.

The main characteristic of JPlag can be summarized as follow.

- JPlag is available as a web service.
- JPlag has a powerful user interface for understanding the results.
- JPlag is resource-efficient and scales to large submissions.
- JPlag has very good plagiarism detection performance.

2.7. Shared Information Distance or Software Integrity Detection (SID)

SID works in two phases:

- In the first phase, source programs are parsed to generate token sequences by standard lexical analyzer.
- In the second phase, Token Compress algorithm is used to compute the shared information metric d(x, y) between each program pair within the assignments.

2.8. Code Match

CodeMatch compares every file in one directory with every file in another directory, including all subdirectories if requested. CodeMatch produces a database that can then be exported to an HTML basic report that lists the most highly correlated pairs of files. You can click on any particular pair listed in the HTML basic report see an HTML detailed report that shows the specific items in the files (statements, comments, identifiers, or instruction sequences) that caused the high correlation.

2.9. AntiCopias (AC)

AC performs the following steps to compare between students assignments.

- Distance integration - This stage put the characters in sequence and converting them into sequence of tokens after removing comments and spaces from the source file.
- Token counting similarity distance - This stage counts the tokens between two assignments using parser (compiler to compare the similarities between the two sequences) and gives the percentage of similarity.

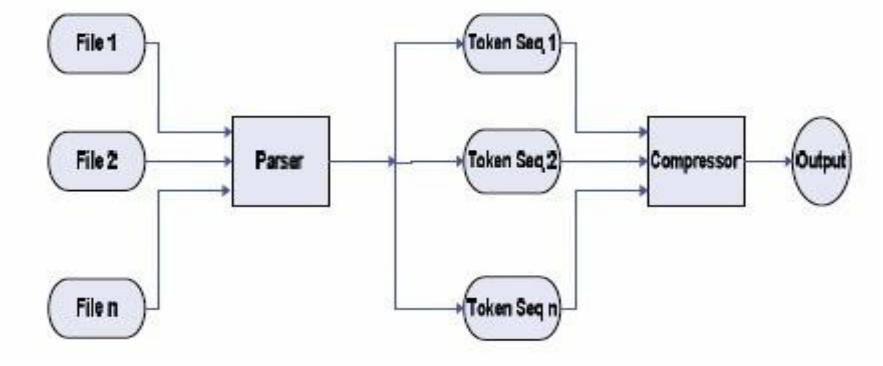

Fig 1. SID

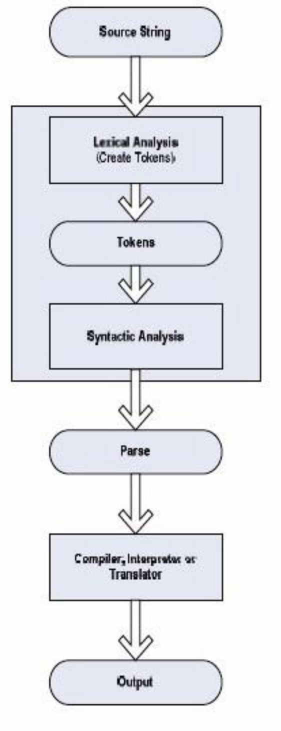

Fig 2. Parsing Steps

2.10. SMS Gateway

SMS Gateway is a unitary device comprising at least a computer with a modem GSM/CDMA, a card GSM/CDMA, and an application program that serves to organize (receive or send) a message. SMS Gateway Application is capable of integrating the phone with computer technology. Advantages SMS Gateway is a gateway to the dissemination of information by using SMS, it can spread the message to hundreds of numbers automatically and quickly directly connected to the database with phone numbers without having to type hundreds of numbers and messages on the phone because all the numbers will be taken automatically from database. In addition, the SMS Gateway can customize messages to be sent. Picture of the use of SMS Gateway in Moodle can be illustrated as shown below:

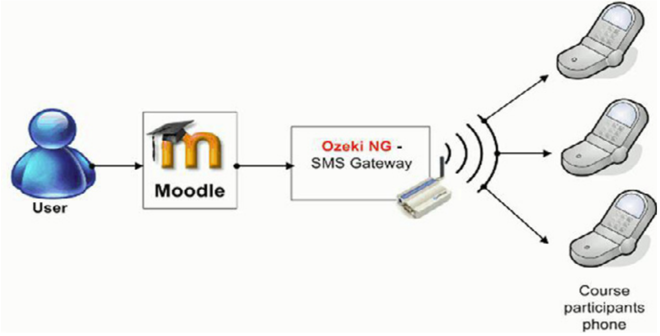

Fig3. SMS Gateway with Moodle.

3. Analysis And Design

To design a plagiarism detection system is then carried out an analysis of system requirements that will be built. Requirements of the system requirements are divided into two, namely the functional requirements and non functional requirements.

3.1 Functional requirements.

- Enable teachers to detect plagiarism and cheating in student submitted assignments. The system reads the submitted assignments and enters them to the algorithm to find the degree of similarity between them.
- Viewing visually aided cheating (similarity) reports. Teachers can display cheating (plagiarism) report, which contains all submitted assignments and the percentage of similarity of each assignment with others.
- The main functions such a registration, login, create courses are already exist in the Moodel (it is not new functions to be added in this project).
- System is capable of displaying file content comparisons that have similarities.
- The system can automatically send alerts to students detected cheating or plagiarism action, in the form of SMS alerts.

3.1. Non-Functional requirements.

1. Compability. System should be compatible and can be integrated with Moodle because it will be added as new feature to Moodle.
2. Easy to use. Teachers will interact with the system to generate plagiarism report through a user-friendly graphical user interface. Furthermore, the generated reports will contain both textual and visual (bars, charts, etc.) representation for the results.

3.2. Development requirements

1. Hardware resources.
   - *Personal Computer* (PC), for server application.
   - LAN or WAN, used to connect a computer server with client computer.
   - Modem for SMS Gateway connection.
2. Software resources.

Plagiarism detection system to integrate into the Moodle virtual classroom platform, it would require some software to suit the needs of Moodle. Moodle is a Web-based applications using PHP programming language, and database applications using MySQL,so it is a system built using the PHP programming and MySQL database. As for the SMS Gateway can be integrated into Moodle, used Ozeki NG, because the Ozeki NG SMS Gateway is a software that can be integrated into Moodle.

3.3. System design

The system designed involving three components lecturer, students and administrators. Students can access a particular course, submit assignments, and receive alerts. Lecturers provide course material, assign tasks, view the results of plagiarism detection, see the same results file detection, and provide alerts, while administrators manage user data and manage the system. In plagiarism detection systems programming tasks in the Moodle-based virtual classroom, the system first checks the username and password entered by the lecturer or students into a database match, when appropriate, the system displays the pages of virtual classes and courses that exist. The system will display the questions that have been uploaded by the lecturer and put the task on submit students into the database. The system detected a similarity between the tasks. Detection procedures are as follows.

1. Distance integration. This stage put the characters in sequence and converting them into sequence of tokens after removing comments and spaces from the source file.
2. Token counting similarity distance. This stage counts the tokens between two assignments using parser (compiler to compare the similarities between the two sequences) and gives the percentage of similarity.
3. Visualize the detection results in table form percentage comparisons, both comparisons between the two files as well as comparisons between multiple files.

In the process of the system will display a page alerts SMS Gateway for lecturers. The system will deliver alerts that provided teachers to students in the form of an SMS.

4. Analysis of the Results

This section will be delivered the results of testing the software presented in the form of tables. The test performed is the scenario adopted for testing the software in accordance with the function use case that has been developed.

Table 1. Prior to testing aspect plagiarism detection process.

Aspects of the tested	Aspect before the plagiarism detection process		
Process	Plagiarisme detection process		
Use case	Login, give the task, displaying the task, the task of collecting		
Testing details	Expected	Results of software testing	Analysis of test results
Perform the login process by entering the login function, the correct username and password	Login process is successful, all active in the system menu	conform	In order for these processes can be successful it must be in accordance with the configuration that has been done before.
Conducting the process of giving the task to activate buttonand insert about the task assignment	Assignment page appears, type of assignment, about the column, limit time spent on	conform	
Process by activating the button displays the task assignment	Show a page that contains about task assignment and execution time limit	conform	
Make the process of collecting tasks by activating button assignment	Shown assignment page, the column to submit answers to the task, and tabs for editing tasks	conform	

Table 2. Testing Aspects of Plagiarism Detection Process

Aspects of the tested	Aspect the plagiarism detection process		
Process	Plagiarisme detection process		
Use case	Reading files, detect plagiarism, plagiarism detection result display		
Testing details	Expected	Results of software testing	Analysis of test results

Make the process of reading a file answers students who have been placed in one directory	Comparing the files tasks collected by the student	confirm	The process plagiarism detection of student tasks performed by the system bymaking the task file into theform of tokens and parse,then calculate the similarityresulting in a percentage levelof similarity.
Conducting the process of counting the percentage of similarity between the student assignment	Calculate the percentage of similarity among the students tasks	confirm	
Showing results of plagiarism detection in student task	Plagiarism detection results of shown in table form the similarity percentage	confirm	

Table 3. Aspects of Testing Process After the detection of plagiarism

Aspects of the tested	Aspect the plagiarism detection process		
Process	Plagiarisme detection process		
Use case	Provides alerts, receive alerts, create exam questions, exam answers to upload, view exam answers, provide value, display value		
Testing details	Expected	Results of software Testing	Analysis of test results
Conducting the process of providing alerts to students, with a percentage similarity above 50% by using sms gateway	Students with a level of similaritypercentage above 50% receive a sms alerts	confirm	The process of granting alerts by lecturer of students with task similarity percentage level above 50% by using sms gateway..

5. Conclusions

The conclusion of the design and implementation of plagiarism detection system on the programming task in avirtual classroom with Moodle is that software plagiarism detection systems that are designed and tested can make or process any of the following.

- Plagiarism detection systems programming tasks in a virtual classroom with Moodle allows the lecturer to know the existence of similarities among students that if the task is done manually requires considerable effort, especially when the number of students attending is quite a lot.
- Plagiarism detection systems programming tasks in a virtual classroom with Moodle is able to show the percentage of similarity in student assignment, whether the similarity between the two files as well as similarities between one file into many files.
- In addition to show the percentage of similarity detection results, the system was able to show details of the contents of the detected files have similarities.
- System of alerts that are designed in this thesis, to address its students with a more efficient in terms of time, so that students can receive alerts/information as soon as the detection process is completed. Constraints faced in making the application of plagiarism detection system on the programming task in a virtual classroom with Moodle is on the submission of detail edresemblance of the detected content file, because the submission of this information must go through the file attachment facility cannot be done on medium SMS.

References

Aiken, Alex. (1994). *MOSS (Measure Of Software Similarity) plagiarism detection system,* University of Berkeley, 1994, http://www.cs.berkeley.edu/~moss/.

Clough, P.D. (2003). *Old and New Challenges in automatic Plagiarism Detection*, Departement of Computer Science, University of Sheffield, UK

Clough, P.D.(2000), Plagiarism in Natural and Programming Languages: An Overview of Current Tools andTechnologies, Departement of Computer Science, University of She ffield, UK

Dick Grune and Matty Huntjens .(1989). Het detecteren van kopieën bij informatica-practica (in Dutch),
Informatie 13, pp(864-867)

Kristina L. Verco and Michael J. Wise. (1996) Software for Detecting Suspected Plagiarism: Comparing Structure and Attribute-Counting Systems, 1st Australian Conference on Computer Science Education, Sydney, Australia.

Lutz Prechelt, Guido Malpohl and Michael Philippsen. (2000). JPlag: Finding Plagiarisms among a Set of Programs http://page.mi.fu-berlin.de/prechelt/Biblio/jplagTR.pdf.

M. H. Halstead, Elements of software science, North Holland, New York, 1977.

Manuel Freire, Manuel Cebrifn and Emilio del Rosal Escuela, AC: An Integrated Source Code PlagiarismDetection Environment, Polit_ecnica Superior, Universidad Aut_onoma de Madrid, 2007.

McCabe, T.J. (1976), A complexity measure, IEEE Transactions on Software Engineering, **SE-2 (4)**, pp(308-320)

Miller, George A. (2005), A lexical database for the english language, http://wordnet.princeton.edu/

Ridhatillah, Ardini. (2003). Dealing with Plagiarism in the Information System Research Community: A Look atFactors that Drive Plagiarism and Ways to Address Them, MIS Quarterly; **Vol. 27**, No. 4, p. 511-532

Schleimer, Saul; Wilkerson, Daniel, Aiken. (2003) Winnowing: Local Algorithms for Document Fingerprinting,SIGMOD, San Diego, CA,