

Building Automatic Grading Tools for Basic of Programming Lab in an Academic Institution

Rina Harimurti*, Andi Iwan Nurhidayat, Asmunin

Informatics Engineering Department, Faculty of Engineering, Universitas Negeri Surabaya, Indonesia

*rinaharimurti@unesa.ac.id

Abstract. The skills of computer programming is a core competency that must be mastered by students majoring in computer sciences. The best way to improve this skill is through the practice of writing many programs to solve various problems from simple to complex. It takes hard work and a long time to check and evaluate the results of student labs one by one, especially if the number of students a lot. Based on these constrain, web proposes Automatic Grading Tools (AGT), the application that can evaluate and deeply check the source code in C, C++. The application architecture consists of students, web-based applications, compilers, and operating systems. Automatic Grading Tools (AGT) is implemented MVC Architecture and using open source software, such as laravel framework version 5.4, PostgreSQL 9.6, Bootstrap 3.3.7, and jquery library. Automatic Grading Tools has also been tested for real problems by submitting source code in C/ C++ language and then compiling. The test results show that the AGT application has been running well.

1. Introduction

The skills of computer programming is a core competency that must be mastered by students majoring in computer sciences. The best way to improve this skill is through the practice of writing many programs to solve various problems from simple to complex. The more often the practice of directly writing program to solve various problems, the experience of programming skills also increased.

One of the activities to improve the ability of programming in the campus is a programming practice. In this activity, students are given the problem then write the program code and show the result to the lecturer or assistant. If there is an incompatibility between problems specification and the student's answers, the students must make revisions and show the results again. It takes hard work and a long time to check and evaluate the results of student labs one by one, especially if the number of students a lot. In addition, program evaluations related to memory usage, source code file size, and execution time are not performed.

Based on these constrain, web propose Automatic Grading Tools (AGT), the application that can evaluate and deeply check the source code in C, C ++ or Java language that has been submitting by students, compile it, display the compilation results, display error if there are logic or syntax errors, and do the sorting results. This application should be easy to use, can be used simultaneously, responsive web-based, and can be used on basic of programming lab.

Related research on automatic grading of computer program, scalable autograder and Learning Management System (LMS) integration, integrate an autograder and existing an LMS, build



assignment module in Moodle E-Learning System for java programming assessment, build web-based application for monitoring coding process by recording typing activity, support online compilation and execution using an autograder [1][2][3].

2. Automatic Grading System

A system that can be used for assigning tasks, evaluating the source code, compiling, and displaying compilations results according to the specified language preference. The automatic grading system has been widely used, both on campus and in competitive programmings, such as Web-CAT, SPOJ (Sphere Online Judge), project euler, codechef, codeEval, and topcoder.

Academic institutions face the challenge of providing their students with a better skill in programming. therefore, the number of programming practices multiplied. as a consequence, the practice room and availability of lecturers should be prepared. Automatic Grading System enables students to work practice of programming anywhere, anytime and can submit program code anytime then get feedback quickly.

2.1. Architecture

The application architecture consists of students, web-based applications, compilers, and operating systems. Using the browser, students open the application Automatic Grading System. in the application, the students do the registration, login to the application, choose the problem, do the problem then submit the program code file. The program code that has been submitted then processed by the Automatic Grading System Application include Evaluation, Grading, and rank. Automatic Grading System architecture is presented in Figure 1.

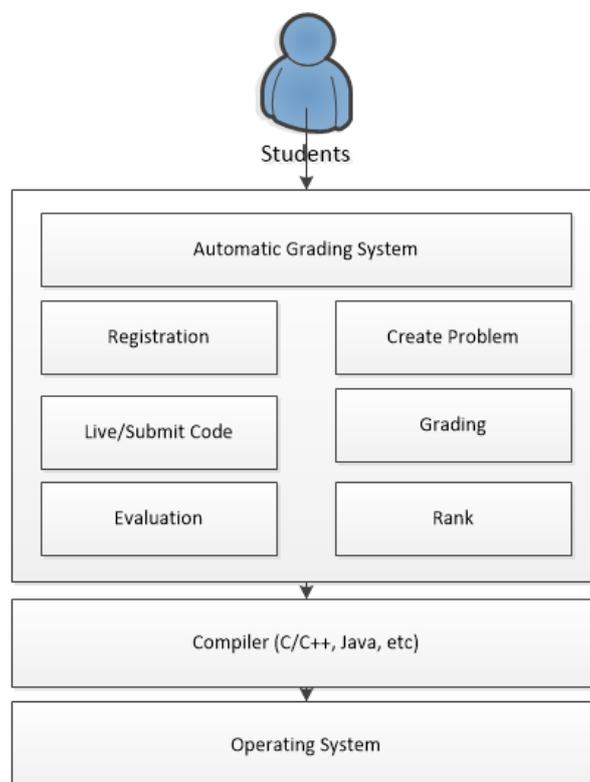


Figure 1. Automatic Grading System Architecture

The Automatic Grading System application will display results based on the evaluation process. If there is an error during the evaluation process, it will be displayed. If there is no error, the evaluation process will display the message: "Accepted" and done the assessment process.

2.2. Flowchart of Automatic Grading System

The Flowchart of automatic Grading System is presented in Figure 2.

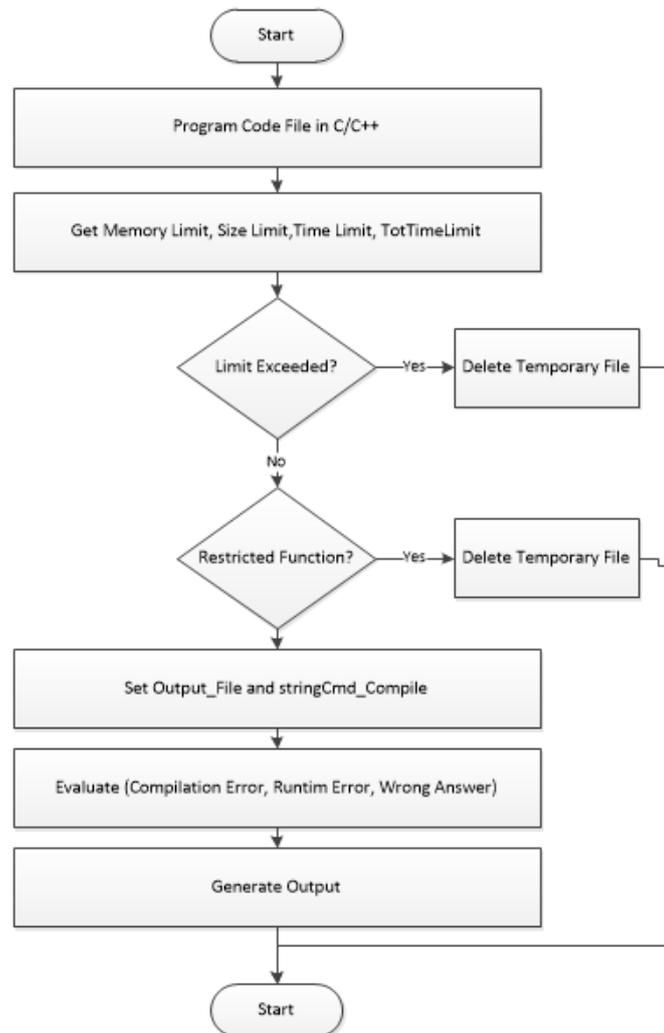


Figure 2. The Flowchart of Automatic grading System

Starting from the program code that has been submitting by the student, then determined Memory Limit, Size Limit, Time Limit, and TotTimeLimit. The next step is check Limit Exceeded and Restricted Function, if true then delete temporary file and if false then set Outfile_File and stringCmd_Compile. The main process in the flowchart is Evaluate, which will check and evaluate program code. If there is an error during the evaluate process, it will display a message according to the error that occurred. The last process of Automatic Grading System is Generate Output, which will display the result.

3. Implementation

Automatic Grading Tools (AGT) is implemented using open source software, such as laravel framework version 5.4, PostgreSQL 9.6, Bootstrap 3.3.7, and jquery library. For the compilation and evaluation of program code used gcc, g++ and java compiler. The operating system used during implementation and testing is Ubuntu Linux 16.0.4 LTS which has installed gcc, g++ and java compiler.

To avoid spaghetti code, the application has been implemented using the MVC (Model View Controller) architecture using Laravel Framework. The main file in this application is grader.php, a class that contains the variables and methods used in the evaluation process and grading system. Variables in the grader.php file, contains information about compilation and execution data, size limit, time limit, memory limit, total time limit, and output limit. The other information that stored in the file grader.php is storage path, information about current submit, maximum time, restricted function, the amount of loops execution, redirect output, and array that contains the information of the evaluation process. List of methods in the grader.php file: evaluateCode, evaluate, evalSizeLimit, evalRestrictedFunction, evalCompileError, evalRuntimeError, generate output file, evalWrongAccept, delTemporaryFile, RenJavaClass, searchClass, buildCodeFile, and builtResultArray. List of variable and method in grader class is presented in Figure 3.

Grader
<pre> -c_lang : string -cplusplus_lang : string -java_lang : string -size_limit : int -time_limit : int -memory_limit : int -total_time_limit : int -output_limit : int -storage_path : string -problem_path : string -results : bool -loop_time : int </pre>
<pre> +evaluateCode() : int +evaluate() : int +evalSizeLimit() : bool +evalRestrictedFunction() : bool +evalCompileError() : bool +evalRuntimeError() : bool +generateOutputFile() : string +evalWrongAccept() : bool +delTemporaryFile() : int +RenJavaClass() : int +searchClass() : string +buildCodeFile() : string +builtResultArray() : string </pre>

Figure 3. Variable and Method in Grader Class

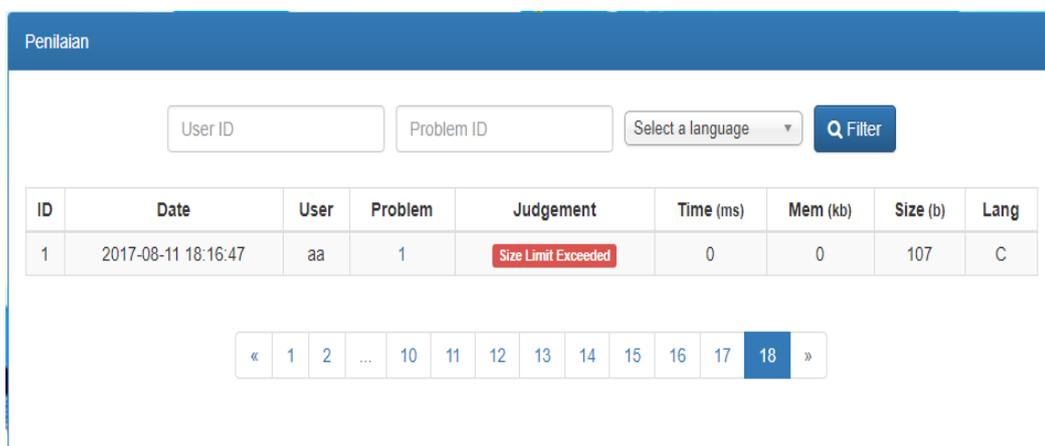
For implementing the compilation and execution, Automatic Grading Tools (AGT) uses gcc, g++ and java compiler that has been installed on the Linux operating system. The program code that has been submitting by the student, checked and evaluated by AGT and then sent to the compiler to compile and run. If during evaluation the program code there is a discrepancy with the specification, then the message will appear, the program code is not compiled and run. the list of messages displayed by The results of the evaluation process: Judging, Unqualified, Internal Error, Size Limit Exceeded,

Restricted Function, Compilation Error, Time Limit Exceeded, Output Limit Exceeded, Presentation Error, Wrong Answer, and Accepted.

4. Testing

To discover a bug in the software has been doing unit testing, component testing, and system testing [4]. Unit testing focus on testing the functionality of object or method, component testing focus on testing the component interface that provides access to the component function, system testing focus on testing component interaction. On unit testing, individual program unit or class are tested. Each class has been tested with unit testing and ensured bug-free before integrating with other classes. After unit testing, functional test on PC is executed for CRUD (Create Read Update Delete) function for all features in AGT.

Automatic Grading Tools has also been tested for real problems by submitting source code in C/C++ language and then compiling. The test results show that the AGT application has been running well. The output from functional testing is presented in Figure 4.



ID	Date	User	Problem	Judgement	Time (ms)	Mem (kb)	Size (b)	Lang
1	2017-08-11 18:16:47	aa	1	Size Limit Exceeded	0	0	107	C

Figure 4. Message Size Limit Exceeded

The message displayed in figure 1 depends on the evaluation of the program code. When the evaluation result file size exceeds that specified, the Size Limit Exceeded message is displayed. When a compilation error occurs, the Compilation Error message is displayed. When the execution time is too long, Time Limit Exceeded message is displayed. Another message is presented in Table 1.

Table 1. list of message in AGT

Message Code	String Message
JDG	Judging
UQD	Unqualified
IE	Internal Error
SLE	Size Limit Exceeded
RF	Restricted Function
CE	Compilation Error
RTE	RunTime Error

TLE	Time Limit Exceeded
MLE	Memory Limit Exceeded
OLE	Output Limit Exceeded
PE	Presentation Error
WA	Wrong Answer
AC	Accepted

5. Conclusions and Future Work

Using open source technology and MVC Architecture, automatic grading system application has been successfully created running well. based on the test results, the application has been able to check and evaluate the program code in depth and display the results.

References

- [1] K. Danutama and I. Liem 2013 Scalable Autograder and LMS Integration *Procedia Technology* **11**, 388-395,
- [2] N. A. M. Z. N. S. A. Norazah Yusoft 2012 Java Programming Assesment Tool for Assignment Module in Moodle E-Learning System *ICTLHE*
- [3] C. N. Timitius and L. Inggriani 2013 Source Code Editing Evaluation for Learning Programming *Procedia Technology* **11** 169-175,.
- [4] I. Sommerville, *Software Engineering* 2016 Tenth Edition England: Pearson Education Limited
- [5] M. Stauffer, *Laravel Up and Running* 2017 A Framework for Building Modern PHP Apps, USA: Oreilly Media,.