Remote interpreter API model for supporting computer programming adaptive learning

Rosihan Ari Yuana*[1], Ignasius Agus Leonardo[2], Cucuk Wawan Budiyanto[3]
Pendidikan Teknik Informatika dan Komputer, Universitas Sebelas Maret
Ir. Sutami St. 36 A Surakarta, Indonesia, telp: +62271-718419
*Corresponding author, e-mail: rosihanari@staff.uns.ac.id[1],
ignasiusleonardo@student.uns.ac.id[2], cbudiyanto@staff.uns.ac.id[3]

Abstract

Despite the adoption of Learning Management Systems (LMS) has been continuously growing in the last decade few, if any, scholars addressed the lack of the interactivity in online learning systems. The requirement for an interactive learning model has been increasingly prominent as content providers realize the importance of personalization of content suits to learners' learning progress. This paper demonstrates the adoption of adaptive learning into existing LMS engine to overcome the limitation of either systems plugins or server specification pertinent to students' psychomotor abilities in computer programming learning. In this regard, the psychomotor is understood as the ability of students to write the program code as correctly as expected. In this study, a web API model that works to run an interpreter based-program code remotely was developed. The web API model can be utilized by LMS so it becomes the solution to the problem. The structure of the web API model has been adapted to the needs of the learning assessment. The implementation of the developed API web model is done in Python and PHP programming languages. The performance test was done by submitting 10 to 100 program codes simultaneously indicated no significant difference to the required resources (CPU usage and memory usage) to run the program code. Furthermore, for response time, the average time needed to run Python and PHP program code is also no significant difference. The average of CPU usage required by the web API to run a Python program code is 0.2058% with 0.5973 seconds as a response time. Meanwhile, to run the PHP program code, the average CPU usage required is 0.8074% with 0.3110 seconds response time. It can be concluded that the web API performance does not overburden the server.

Keywords*: adaptive learning, computer programming learning, web API*

1. Introduction

Human resources that have capabilities in the mastery of information technology at this time is very necessary, especially in software engineering. Therefore, the preparation of skilled human resources is essential. Efforts that can be done is to develop effective learning methods, and adaptive learning is one of them.

Adaptive learning is a learning method that uses a computer or computer system as an interactive learning tool [1]. Adaptive learning technologies provide an environment that can intelligently adapt to the needs of individual learners through the presentation of appropriate information, comprehensible instructional materials, scaffolding, feedback, and recommendations based on participant characteristics and on specific situations or conditions. In addition, the advantage of adaptive learning is that its technology can create student-centered learning [2]. Moreover, a study proves that the effectiveness of adaptive learning (computer tutoring system) is almost as effective as human tutoring [3]. In this case, another advantage of adaptive learning is that it can be done without time and place restrictions when compared to human tutoring. Thus, adaptive learning has been regarded as a pedagogical awakening that has the potential to enable higher education institutions to optimize quality, access, and cost [4], although in some implementations it sometimes does not make a significant difference in its instructional approach compared to conventional [5].

In the implementation phase, adaptive learning can consist of several characteristics, namely: analytics, local, dispositional, macro, and micro. Adaptive learning should be analytics, this means that it has a basic functionality that supports other forms of adaptive learning, but can also be used to display important information for instructors and students, so that they can

Received June 1, 2018; Revised October 2, 2018; Accepted November 1, 2018

adjust their own learning [6]. Local means that adaptive learning must have targeted feedback, guidance for students, and also remedial content [7]. As for the dispositional, the adaptation process existing in adaptive learning should be based on student variables such as talent or ability [8]. Macro means that the selection of student tasks in adaptive learning is based on the students' level of understanding [9]. In addition, in micro characteristics, adaptive learning should have a scaffolding mechanism to aid the learning process [10].

Related to computer programming learning, the problem often faced by students is the difficulty in making program algorithms (problem-solving) [11, 12]. In addition, the process of transforming the algorithm into the program code, and the lack of understanding of the program structure are also constraints [13]. These difficulties can occur in the form of syntax errors, inappropriate data type selection, and also debugging the program if there is a syntax error or runtime error. To overcome student difficulties in creating algorithms, providing many case study problems is one of the solutions. By doing this, the students' ability to construct top-down algorithms using multiple control structures (branching or looping) will be well improved. Similarly, the problem of students' difficulties in writing program code, this problem can be overcome by increasing the student experience in writing program code. The more students write the program codes, the understanding of the syntax will be better.

Based on the above facts, computer programming learning is indispensable for many exercises and needs extra supervision from the teacher. Every process and progress of student learning need to be monitored by the teacher. Therefore, adaptive learning-based systems have been widely developed today to assist teachers in monitoring student progress and guiding them.

An adaptive learning model for programming learning has been widely developed, some of which are in the form of virtual reality [14] and e-learning [15–19]. The most adaptive learning model used is to utilize e-learning. Nevertheless, most e-learning is still focused on improving students' cognitive abilities, while improving psychomotor abilities are still not widely available.

E-learning facilitated students’ psychomotor ability requires a capability that enables students to write program code directly into and evaluated by a particular module in the electronic learning. The evaluation matches the suitability of input and output based on the given problems, the accuracy of selection of data types and variables, and the correctness of the use in control structures (conditional statement and looping). Few e-learning has been supporting this mechanism including Moodle’s CodeRunner additional plugin [19] and Virtual Programming Lab (VPL) [20]. However, these plugins applicability was limited to any LMS but Moodle. Moreover, the plugins require a huge server requirement to embed the program code interpreter module. In this regard, the interpreter module was created to help serves the program code written by students. Both CodeRunner and VPL are constrained by the number of server specifications could be used. It is considerably challenging to implement the plugins into existing LMS.

This paper is intended to address the fundamental issues by adopting the adaptive learning concept to improve students psychomotor ability during online learning/teaching using commercial off-the-shelf LMS. The previously identified issues are including the limited use of developed plugin modules, huge resource requirements, and particular server specifications to place modules that can run program code created by students. The use of the web API supporting distributed computing [21] is proposed. Since a web API server will be able to run program codes remotely the computation load of the LMS server will be reduced.

In this study, the psychomotor interaction between student and LMS will be demonstrated by the use of adaptive learning in computer programming course. A web API model is designed to support the adaptive learning pertinent to programming languages that use interpreters. The web API model will be implemented suits the result of individual learning achievement and tested for performance quality.

2. The Proposed Model

Figure 1 depicts the web API model proposed in this study. The transactions processes that occur in the web API model started from LMS server. In the LMS, user writes program code using code editor. Subsequently, a POST method sends the program code, together with the input-output value, and also the function name of the program code to an API caller. In this regard, the API caller module is a means of communication between LMS server and web API server. The API caller module then sends data such as auth API (used for access security), program code, input-output value, and the function name of the program code to the web API server. LMS determined the input and output values which are required to test the accuracy of the program code created by student.

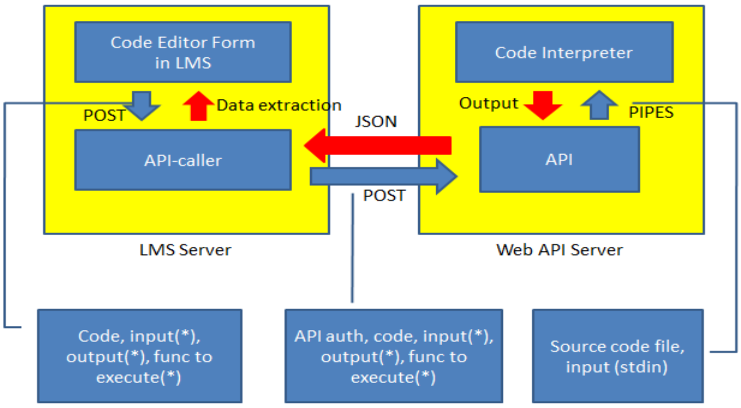
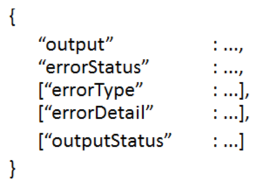

Code Editor Form in LMS
POST
Data extraction
API-caller
LMS Server
JSON
POST
Code Interpreter
Output
PIPES
API
Web API Server
Code, input(*), output(*), func to execute(*)
API auth, code, input(*), output(*), func to execute(*)
Source code file, input (stdin)

Figure 1. Remote interpreter web API model

In the web API server, the program code received from the LMS server stored to a source code file for later use by the interpreter along with standard input using pipe technique. Once the interpreter executes the program code the output is read by the API module, which is sent back to the API caller as a response. Figure 2 demonstrates the responses in a JSON structure.



```
{
    "output"            : ...,
    "errorStatus"       : ...,
    ["errorType"        : ...],
    ["errorDetail"      : ...],
    ["outputStatus"     : ...]
}
```

Figure 2. JSON structure of web API response

The *output* field depicted in Figure 2 contains outputs of the program after being executed by the interpreter. This field indicates output of the program should the program is syntax or runtime error-free. In contrast, if the program experiences a syntax or runtime error specific error information will be stored in this particular field. The *errorStatus* field contains a value of 0 or 1. The value in this field corresponds with the content of the *output* field. The value ‘0’ indicates syntax error free in the program code, while value ‘1’ indicates a syntax error. The *errorType* displays the type of syntax error experienced while the complete error message will be displayed in *ErrorDetail* field. Meanwhile, *outputStatus* contains a value either ‘0’ or ‘1’. If the output of the program code is as expected then the *outputStatus* value is ‘1’, whereas if the output is not as expected then the value will be ‘0’. The *errorType* and *errorDetail* fields will appear in the response if *errorStatus* is ‘1’. As for the *outputStatus*, this value will appear if there is output data sent from the LMS server to the web server API.

3. Research Method

This study conducted the following procedures. Creating a web API model was the first step. The web API replicates the behavior of a student applying adaptive learning in computer programming course. In this regard, the developed web API model facilitates the measurement of student's psychomotor abilities. Those including but not limited to the ability to determine variables and data types, the ability to use control structure statements, the ability to run programs and debugging processes, and the ability to validate data input [22]. Furthermore, the web API model will be implemented in several interpreter-based programming languages such as Python and PHP for their popularity. The web API module itself was created with the PHP language.

Testing the performance of the web API model was the second step in the study. The web API module was deployed in a Virtual Private Server (VPS) in which previously installed with PHP and Python interpreters. An API caller module was created in the client side. The API caller model was activated to be able to send program code in PHP and/or Phyton script to web server API to run. The measured web API parameters are including memory usage, CPU usage, and response time. In this regard, the response time is calculated since the program code was sent from client to the API web server until the web API responses were received by client.

4. Results and Analysis

The section discusses the implementation and performance analysis of web API remote interpreter.

4.1. Web API Implementation

The developed web API model can be implemented with system topology as illustrated in Figure 3. A web API server with a programming code interpreter module that serves to run the program code created by the students (clients) in the LMS is required. In its implementation, this web API can be utilized by several LMS at once.

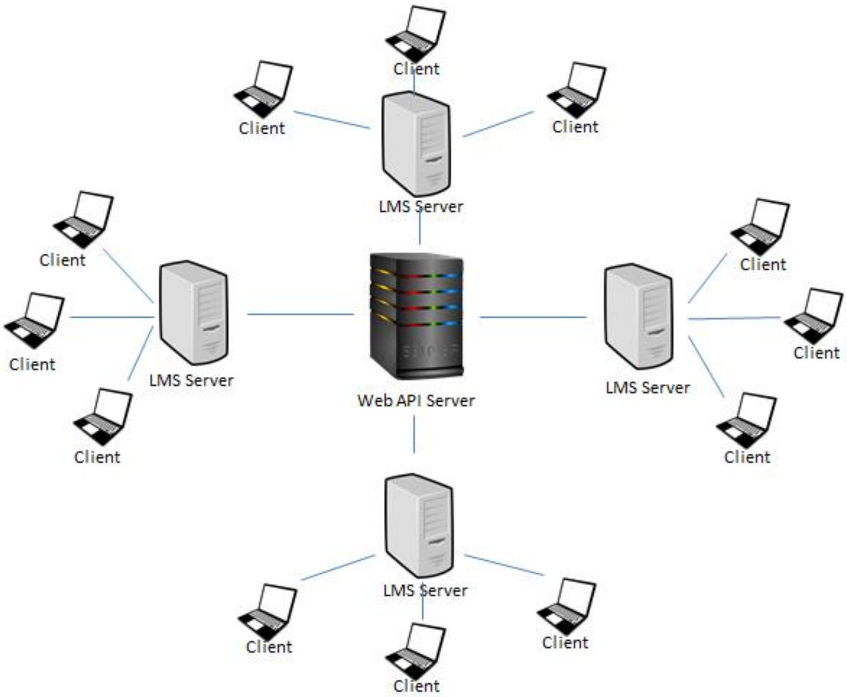
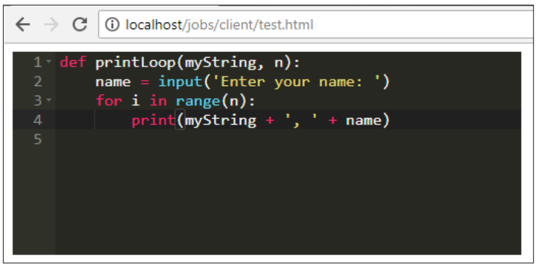


Figure 3. Implementation topology of web API remote interpreter

Meanwhile in the LMS, by utilizing this web API, LMS developers can use it as a feature for students to practice writing the program code. Program code created by students directly through LMS, then students can get feedback in the form of output program after the run. Figure 4 shows a sample Python program code created by a user in an LMS.

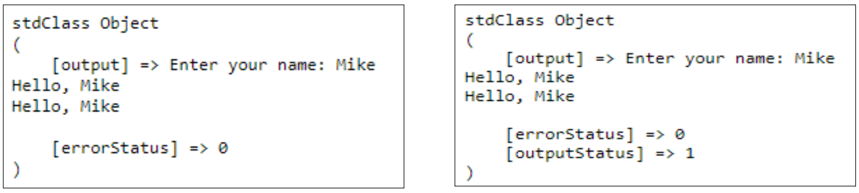


```
def printLoop(myString, n):
    name = input('Enter your name: ')
    for i in range(n):
        print(myString + ', ' + name)
```

Figure 4. Display of program code written through LMS

For program input requirements, the feature to provide standard input for the program code created is available within the web API model. The example code of the program created in Figure 4 is a function. Furthermore, to check the accuracy of the function code created, the feature to call the function is also provided in the web API. Figure 5 (a) shows the response of the web API server when processing the program code *printLoop(myString, n)* function, with the given input standard '*Mike*' and called with *printLoop('Hello', 2)*. The display shows there are no errors when running the program. In the response format, the *output* field contains the value that is the output of the program code, while the *errorStatus* value is 0, which means there is no syntax error or running time error when executed. In the LMS, examples like this are suitable given in the form of exercise questions that test students' ability to create subprograms or functions.

Meanwhile, Figure 5 (b) shows the web response API when given the expected output. If the expected output is equal to the output of the program when it is run, then there will be an additional parameter of the *outputStatus* value 1 in the response. One example of utilizing this feature in the LMS is in the form of quiz questions that ask students to create program code. The amount of quiz score given to the students depends on the accuracy of the program to produce the output as expected. From this feature also, the ability of students in making the program code algorithm can be known. If the output of the program is not the same as the expected output, then there is a possibility of an algorithmic error made by the student.



```
stdClass Object
(
    [output] => Enter your name: Mike
Hello, Mike
Hello, Mike

    [errorStatus] => 0
)
```

(a)

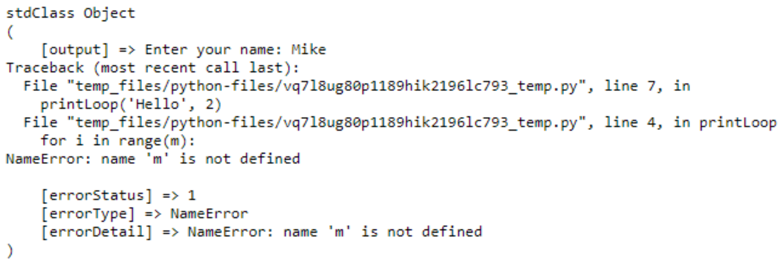
```
stdClass Object
(
    [output] => Enter your name: Mike
Hello, Mike
Hello, Mike

    [errorStatus] => 0
    [outputStatus] => 1
)
```

(b)

Figure 5. The response of web API if there is no syntax error or runtime error

Next, it will display the web response API if an error occurs when the program code is executed as shown in Figure 6. From this response format, the cause of the error and also the type of error can be known. The cause of the error can be read from *errorDetail* field or *output*. While the type of error can be obtained from *errorType*. Based on this emerging error message, students will be able to find out the cause of errors from the program code it created. In addition, the types of errors that appear can also be stored in the LMS to be processed in such a way to produce information to the teacher. The resulting information, for example, is the type of error that often occurs in one student or all students in general. Based on this information, the teacher can take the next action to minimize this type of mistake going forward. In addition, this type of error can also be used for the LMS that serves as an intelligent tutor to present the content of the deepening of the material. The content presented is related to the type of error automatically.



```
stdClass Object
(
    [output] => Enter your name: Mike
Traceback (most recent call last):
  File "temp_files/python-files/vq7l8ug80p1189hik2196lc793_temp.py", line 7, in
    printLoop('Hello', 2)
  File "temp_files/python-files/vq7l8ug80p1189hik2196lc793_temp.py", line 4, in printLoop
    for i in range(m):
NameError: name 'm' is not defined

    [errorStatus] => 1
    [errorType] => NameError
    [errorDetail] => NameError: name 'm' is not defined
)
```

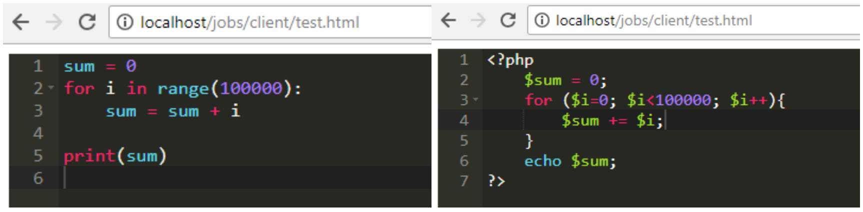
Figure 6. The response of the web API if an error occurs while running the program

Any program code sent from an LMS will be saved to a source code file inside a directory by the web API. Furthermore, the stored source code file will be run by the interpreter. To distinguish each of these source files, the unique session ID filename on each request is used. This source code file is only temporarily stored and will be deleted after the source code has been executed by the interpreter automatically.

In addition, this web API can also handle a condition when there is an infinite loop inside of program code created by students. If so, there will be no response provided by the web API. The running program will be automatically terminated, thus running program continuously without stopping will not happen.

4.2. Performance Analysis

Once the web API model is implemented it is ready for a performance test. The test scenario was to send the program code containing a large number of looping using Python and PHP from client to web API server. The program code was represented in Figure 7:



localhost/jobs/client/test.html
localhost/jobs/client/test.html

```
sum = 0
for i in range(100000):
    sum = sum + i

print(sum)
```

```
<?php
    $sum = 0;
    for ($i=0; $i<100000; $i++){
        $sum += $i;
    }
    echo $sum;
?>
```

Figure 7. Program code used for performance testing

in this case, the program code that will be sent to the web API server was a program to calculate the sum of 0 + 1 + 2 + ... + 99999.

Performance testing was done in stages, with a total of 10 stages. The web API server specification used in this testing phase is VPS server with 4 cores processor, 2 Gb RAM, Linux Centos operating system, 1.66 GHz CPU, and PHP version 5.2.17, and Python version 2.4.3. The first step of the testing phase was to send the program code as many as 10 pieces simultaneously. Furthermore, from the process of running this program code, the amount of CPU usage, the use of its memory, and also the response time length was calculated. Subsequent to the calculation of these three types of data the average value of each type of data were calculated. The second stage was followed by sending of 20 simultaneous program code, and then the average value of CPU usage, memory, and response time were also calculated. The process continued until the transmission of 100 pieces of simultaneous program code were completed.

The results of performance testing when running Python program code are presented in Table 1. The data did not indicate a significant difference of CPU usage when simultaneous requests ranged from 10 to 100. Overall CPU usage of a Phyton script in the web API ranged from 0.1% to 0.4%. The use of memory during 10 to 100 requests also did not indicate significant differences. The memory use remained between 20 to 25 percent. Therefore, it inferred that the memory consumption to run a Python script was considerably low. The response time duration taken for 10 to 100 simultaneous requests ranged 0.5 to 0.7 seconds. Overall, the average CPU usage used by the web API to run a Python script was 0.2058%, the average memory used was 23.3040%, and the average response time of 0.5973 seconds.

While the results of testing the performance of web API to run the PHP program code presented in Table 2. From the table, it can be seen that the range of CPU usage on the number of 10 to 100 requests simultaneously is in the range of 0.6% to 1.0%. There is no noticeable difference in this. As for the aspect of memory usage, the amount is in the range of 15% to 35%. Furthermore, the duration is in the range 0.1 seconds to 0.4 seconds for processing time. In general, to run a PHP script via web API requires an average CPU of 0.8074%, the memory of 22.6868%, and also processing time for 0.3110 seconds.

From these two tests, the web API performance when implemented in Python and PHP can be compared. In CPU usage, web API requires fewer CPU resources to run Python scripts, which are about 1/4 of the resources than to run PHP scripts. Associated with memory consumption, web API use fewer memory resources when running PHP scripts than Python, although not significantly different. While in processing, the processing time required to process PHP script is slightly faster than Python.

Based on the results of this performance test, it can be said generally that the resources needed for the implementation of this web API are not much, this can be seen from the absence of significant differences when the request of the program code as much as 10 to 100 times done simultaneously. So this web API model can be implemented further with the number of more requests.

Table 1. The results of a web API performance test for Python script

ReqNum		Average	
	CPU Usage (%)	Memory Usage (%)	Response Time (seconds)
10	0.2140	21.8426	0.5382
20	0.1735	23.1120	0.5583
30	0.2450	22.8360	0.5456
40	0.2573	23.0284	0.6599
50	0.2476	24.6031	0.5390
60	0.1037	24.6308	0.6522
70	0.2439	25.6850	0.5997
80	0.1406	25.7495	0.5354
90	0.2159	20.2756	0.6244
100	0.2169	21.2766	0.7203
Total Average	0.2058	23.3040	0.5973

Table 2. The results of web API performance test for PHP script

ReqNum		Average	
	CPU Usage (%)	Memory Usage (%)	Processing Time (seconds)
10	0.7245	20.4468	0.3432
20	0.7637	17.0930	0.4503
30	0.6492	18.5061	0.3077
40	0.9957	17.8395	0.4103
50	0.8986	21.6336	0.3244
60	1.0983	15.9658	0.1925
70	0.7299	26.6255	0.1943
80	0.6912	25.1341	0.2168
90	0.7208	27.7922	0.3537
100	0.8017	35.8310	0.3168
Total Average	0.8074	22.6868	0.3110

5. Conclusion and Future Works

Based on research that has been done, web API model that serves to run the interpreter based-program code has been developed. This web API can be integrated with LMS to support computer programming adaptive learning. The developed web API model structure, both input and response structure, has been adapted to suit students' psychomotor ability assessment in learning program code writing. It is evident that the web API module demonstrated its performance during the test. Little to none difference was indicated on the CPU usage, memory usage, and response time after 10 to 100 requests simultaneously sent.

However, when compared to the overall average for both types of program code, the CPU resource required by the web API to run Python scripts is less than the resource to run PHP scripts. In terms of memory consumption, the memory resource when running PHP script is less than Python. As for processing time, the time required to process PHP script is slightly faster than Python. It can be concluded that the web API performance does not overburden the server.

The proposed web API model can be further developed to run a compiler-based program code such as Java, Pascal, or C/C ++. Performance of the test result may depend on the CPU specification and also that operating system that used. Therefore, further research can be also done by comparing its performance on several types of operating system platforms and CPU specifications.

References

[1] Martinez M, Kassara O, Hadioui A, Boundless. Adaptive Learning. *Connotative Learning*. 2014; 6(1): 85-88.

[2] Kara N, Sevim N. Adaptive Learning Systems: Beyond Teaching Machines. *Contemporary Educational Technology*. 2013; 4(2): 108-120.

[3] VanLehn K. The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*. 2011; 46(4): 197-221.

[4] Jarrett J. Bigfoot, Goldilocks, & Moonshots: A Report from the Frontiers of Personalized Learning. *Educause Review*. 2013: 31-40.

[5] Murray M C, Pérez J. Informing and Performing: A Study Comparing Adaptive Learning to Traditional Learning. *Informing Science*. 2015; 18(1): 111-125.

[6] Siemens B. *Learning Analytics and Educational Data Mining : Towards Communication and Collaboration*. Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. 2012.

[7] Durlach P J, Ray J M. *Designing Adaptive Instructional Environments: Insights from Empirical Evidence (Technical Report 1297)*. 2011.

[8] Pashler H, McDaniel M, Rohrer D, Bjork R. Learning Styles: Concepts and Evidence. *Psychological Science in the Public Interest*. 2008; 9(3): 105-119.

[9] Park O, Lee J. Adaptive Instructional Systems. *Handbook of Research for Educational communications and technology*. 2003; (1911): 651-684.

[10] Corbett A T, Koedinger K R, Anderson JR. Intelligent Tutoring Systems. *Science*. 1997; 37: 849-874.

[11] Mhashi M M, Alakeel A L I M. Difficulties Facing Students in Learning Computer Programming Skills at Tabuk University. *Recent Advances in Modern Educational Technologies*. 2013: 15-24.

[12] Rahmat M, Shahrani S, Latih R, Yatim N F M, Zainal N F A, Rahman R A. Major Problems in Basic Programming that Influence Student Performance. *Procedia-Social and Behavioral Sciences*. 2012; 59: 287-296.

[13] Derus S R, Ali A Z. *Difficulties in Learning Programming: Views of students*. 1st International Conference on Current Issues in Education (ICCIE 2012). 2012: 74-79.

[14] Chandramouli M, Zahraee M, Winer C. *A Fun-Learning Approach to Programming: An Adaptive Virtual Reality (VR) Platform to Teach Programming to Engineering Students*. IEEE International Conference on Electro Information Technology; 2014: 581-586.

[15] Zhang J. *An Adaptive Model Customized for Programming Learning in E-learning*. 2010 3rd International Conference on Computer Science and Information Technology. 2010: 443-447.

[16] Jovanovic D, Jovanovic S. An Adaptive E-learning System for Java Programming Course, Based on Dokeos le. *Computer Applications in Engineering Education*. 2015; 23(3): 337-343.

[17] Law K M Y, Lee V C S, Yu Y T. Learning Motivation in E-learning Facilitated Computer Programming Courses. *Computers and Education*. 2010; 55(1): 218-228.

[18] Chookaew S, Panjaburee P, Wanichsan D, Laosinchai P. A Personalized E-learning Environment to Promote Student's Conceptual Learning on Basic Computer Programming. *Procedia-Social and Behavioral Sciences*. 2014; 116: 815-819.

[19] Lobb R, Harlow J. Coderunner: A Tool for Assessing Computer Programming Skills. *ACM Inroads*. 2016; 7(1): 47-51.

[20] Carroll P E R J. *An Online System for Monitoring and Assessing the Programming Process*. Bulletin of the IEEE Technical Committee on Learning Technology. 2016; 18(2): 2-5.

[21] Lemos A L, Daniel F, Benatallah B. Web Service Composition: A Survey of Techniques and Tools. *ACM Computing Surveys*. 2015; 48(3): 1-41.

[22] Mustapha A, Samsudin N A, Arbaiy N, Mohamed R, Hamid I R. Generic Assessment Rubrics for Computer Programming Courses. *Turkish Online Journal of Educational Technology*. 2016; 15(1): 53-61.