

# Online Code Engine to Support Programming Classes: A Case Study

I Wayan Adi Sparta<sup>1</sup>, I Made Putrama<sup>2</sup>, Gede Aditra Pradnyana<sup>3</sup>

Department of Informatics Engineering Education

Universitas Pendidikan Ganesha

Bali, Indonesia

<sup>1</sup>adisparta11@gmail.com, <sup>2</sup>made.putrama@undiksha.ac.id, <sup>3</sup>gede.aditra@undiksha.ac.id

**Abstract**—multi-user applications that can help in learning subjects are now increasingly popular, especially web-based applications. However, the system load will increase as the number of connected users at the same time increases especially in applications that require more usage resources. Therefore, it is important for the system to implement client-based processing in similar applications. This paper discusses how a system that applies client-based processing is built as needed to address the problem. This system is a learning management system that has some features to learn something, that is programming language system, in this case. The developed system consists of interlinked web and desktop applications that have been tested and show favorable results in terms of usability and user response.

**Keywords**—learning management system, online compiler, client-based processing

## I. INTRODUCTION

Informatics Engineering Education Department (PTI) of Ganesha University of Education (Undiksha) has a focus on giving their students the skills of engineering and informatics education. One of them is skill of programming language. In this department, programming languages taught in several courses such as Programming Basic, Algorithm and Data Structure, Object Oriented Programming, Introduction to Artificial Intelligence, and Expert System and Decision.

Although programming is an important skill, this course is very difficult for students to learn. This is the fact that we got after distributing some questionnaires to the students of Informatics Engineering Department of Undiksha on February 9 and 17, 2017. From 30 respondents, 100 percent said that the programming course is a difficult course.

Based on that fact, we decided to create an application to support the programming class in the hope of making programming lessons easier and more convenient. We create an app package that allows instructors to share programming source code with students. In addition, the students will get some features that can help them in learning programming like writing code, compiling directly, and running it online. Based on this idea, there are several research references such as a research by Patel entitled “Online Java Compiler Using Cloud Computing”[1]. In the research, the author created an online compiler for Java programming language by using Cloud Computing Architecture. A remote server used to process the user’s source code, so users did not require installing compiler

in their machine. Another study such as in[2], which has similar concepts to the research done by Patel. The difference is in this study, the authors used parallel processing to execute code and support C, C++, and Java. In[3], authors created an Android application that allow user to learn and code in pascal programming language.

From the research described earlier, we can see that they use the server as the main unit to perform all processes. With this approach, the server will experience a lot of workload when the connected user count increases. The toughest process that will overload the server is in compiling and executing user code. This issue has been discussed in [2] to cut the server workload by dividing it using parallel processing. In this paper, we use different idea to solve the problems by developing an online application with a desktop client-based processing package. With this approach, the server will not be much affected by the number of the connected users.

In addition to research references, there are several applications similar to our application, such as *W3School*, *Code Academy*. These websites are focused on lecturing to students about programming languages. In this paper, we propose an Online Code Engine to support programming classes that focus on the technical features that can be utilized by users in the programming languages learning.

## II. LITERATURE REVIEW

### A. Node.JS

Node.js is a server-side platform built into the Google Chrome JavaScript Engine. Node.js was developed by Ryan Dahl in 2009. Node.js is a platform built in the Chrome JavaScript runtime to build fast and scalable network applications easily. Node.js uses an event-driven I / O model, which makes it lightweight and efficient, perfect for real-time data-intensive applications running on distributed devices. Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. The Node.js application is written in JavaScript, and can be run in the Node.js runtime in OS X, Microsoft Windows, and Linux. Node.js also provides a diverse library of JavaScript modules that facilitate web application development using Node.js for the most part [4].

### B. Electron

Electron is an open source library developed by GitHub to build cross-platform desktop applications with HTML, CSS, and JavaScript. Electron accomplishes this by combining Chromium and Node.js into one runtime and the app can be packaged for Mac, Windows, and Linux. Electron allows you to create desktop applications with pure JavaScript by providing a runtime with native API (operating system) rich. This does not mean Electron is JavaScript that binds to a graphical user interface library (GUI). Instead, Electron uses the web page as their GUI, so you can also see it as a minimal Chromium browser, which is controlled by JavaScript. So all electron applications are technically running web pages in browsers that can utilize your OS API[5].

**C. MongoDB**

Mongo DB is an open source database that uses document oriented data model. MongoDB was one of several types of databases that emerged in the mid-2000s under the NoSQL banner. Instead of using tables and rows as in relational databases, MongoDB is built on collections and document architectures. The document consists of a set of key-value pairs and is a basic data unit in MongoDB. Collections contain documents and their functions as equivalent relational database tables.

Like other NoSQL databases, MongoDB supports dynamic schema design, which allows documents in collections to have different fields and structures. The database uses a document storage format and data exchange called BSON, which provides binary representations of JSON-like documents. Automatic *sharding* allows data in collections to be distributed to multiple systems for horizontal scalability when data volumes increase [6].

**D. Socket Programming**

A socket represents a connection between two software (point-to-point connections) called. More than two pieces of software can communicate with client / server or distributed systems using multiple sockets. For example, many Web browsers can simultaneously communicate with a single Web server through a group of sockets created on the server.

Socket-based software usually runs on two separate computers on the network, but sockets can also be used to communicate locally (inter process) on a single computer. Two-way socket, meaning that both sides of the connection are able to send and receive data. Sometimes an application that initiates communication is called a "client" and another "server" application[7].

**III. ANALYSIS AND DESIGN**

In software development, RAD (rapid application development) is a concept that was born out of frustration with a waterfall design approach that too often resulted in products that were out of date or inefficient by the time they were actually released. The term was inspired by James Martin,

who worked with colleagues to develop a new method called Rapid Iterative Production Prototyping (RIPP). In 1991, this approach became the premise of the book Rapid Application Development [8].

In this research, RAD is implemented to develop the application. The RAD concept is followed by separating several steps including:

1. Business Modeling
2. Data Modeling
3. Process Modeling
4. Application Generation
5. Testing and Turnover

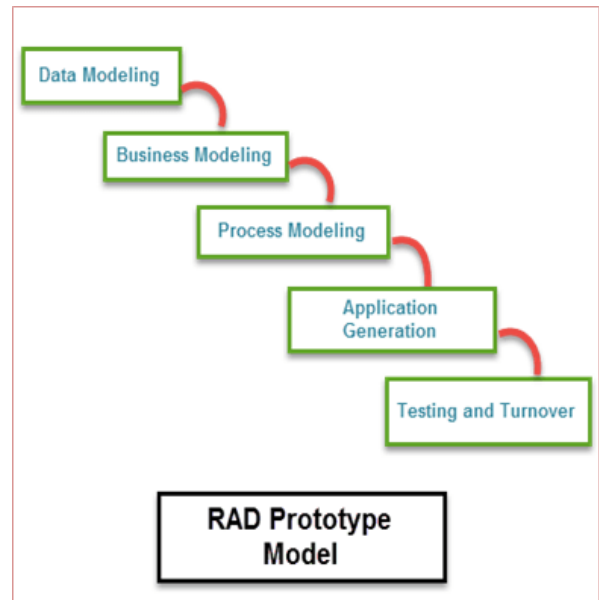


Fig 1 Rapid Applications Development Schema

**IV. IMPLEMENTATION AND TESTING**

**A. Software Analysis**

In this section, the authors looked at some reference apps and discussed with lecturers who taught programming classes about what they had done before to provide input so that the information systems created could actually support learning in the programming class. Authors have also distributed questionnaires to students and lecturers to draw conclusions about the required software specifications and features that can accommodate the management of the programming languages to use, courses, source codes, users, and forum.

**B. Functional Model**

In functional model, the authors apply it using Use Case Diagram. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined. The use case contains all system

activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a particular goal; indeed, the use case and goal are sometimes considered synonymous[9].

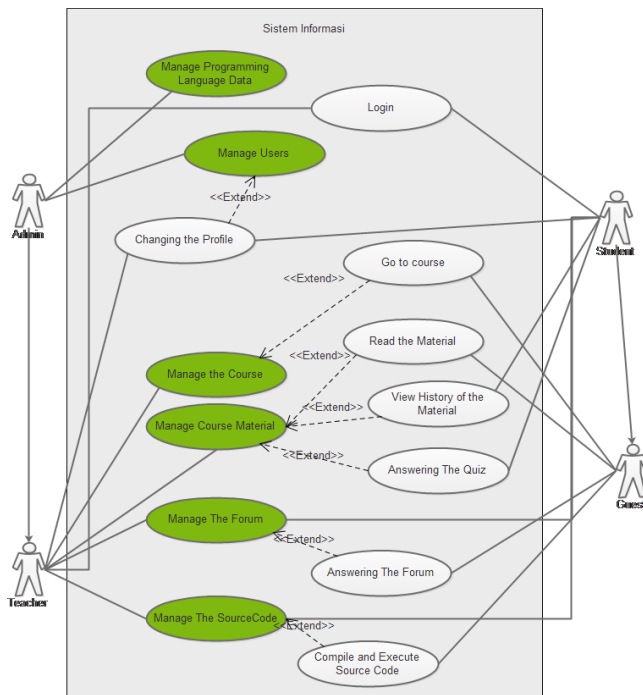


Fig 2. Use Case

C. ERD

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.[10]The ERD of our application shown at Fig 3.

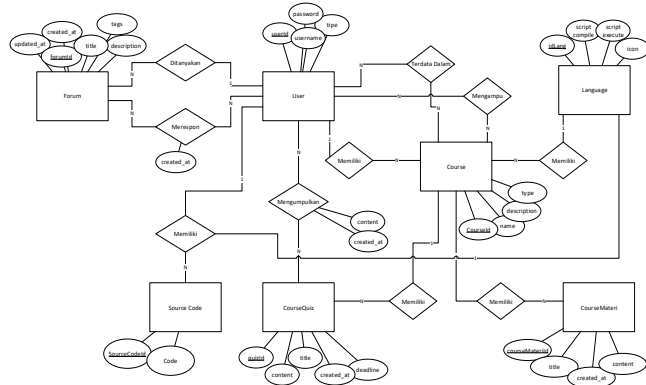


Fig 3. ERD

D. Data Structure

A data structure is a specialized format for organizing and storing data. General data structure types include the array, the file, the record, the table, the tree, and so on. Any data

structure is designed to organize data to suit a specific purpose so that it can be accessed and worked with in appropriate ways. In computer programming, a data structure may be selected or designed to store data for working on it with various algorithms.

The data structure was implemented by using MongoDB that shown in Fig 4.

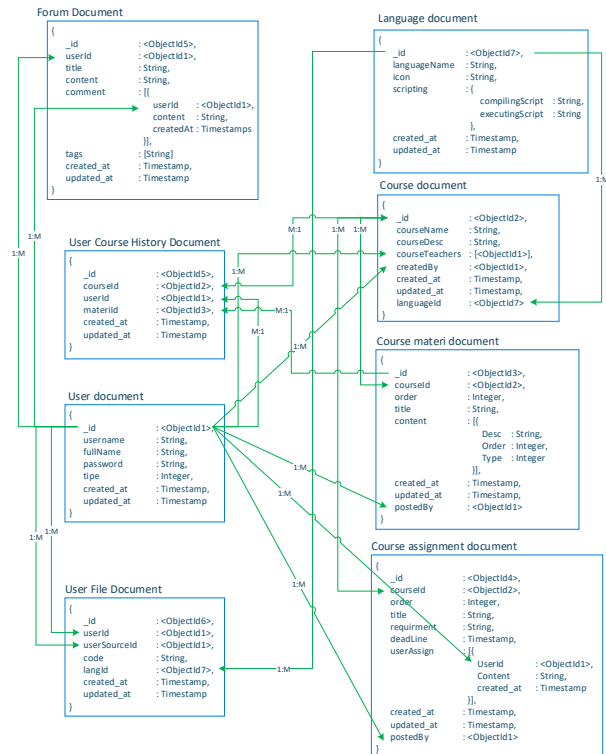


Fig 4. Data Structure

E. System Architecture

The focus of the research is to build a system with a minimized server load by using client based processing in some processes. The system consists of two separated applications, a desktop application and a web application. Auser will open the desktop application that will receive tasks from the web application.

While desktop applications run on user PCs, users are allowed to write code on web pages. The web page will display a command prompt for the user. This section will show how the code that has been created successfully executed. When a user completes a program code, the code will be sent to the user's PC through an initiated server connection at the beginning and a compilation of the program will be performed and execute it on a desktop application running in the tray system and the results will appear on the web page..The system architecture of the applications is depicted in Fig 5.

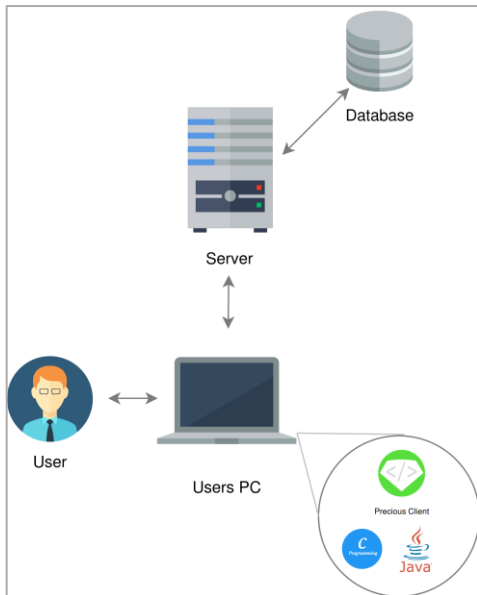
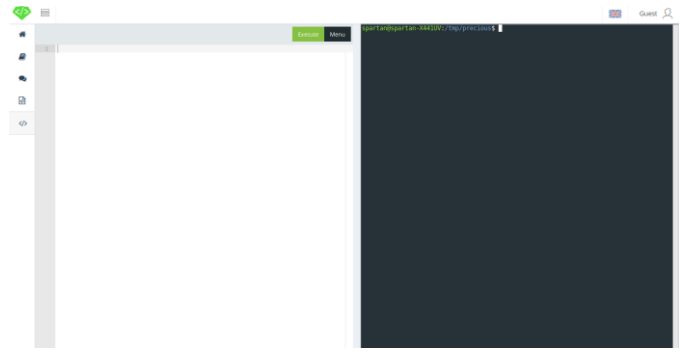


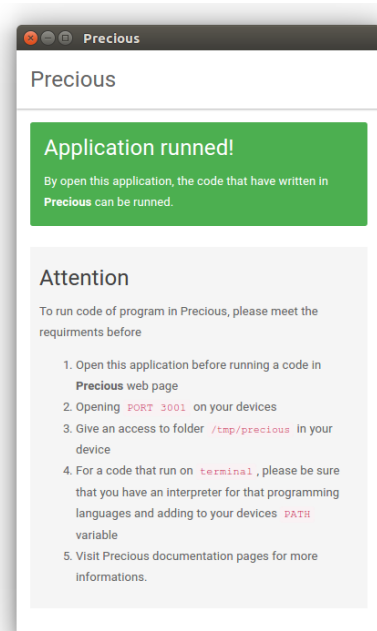
Fig 5 System Architecture

F. General Information

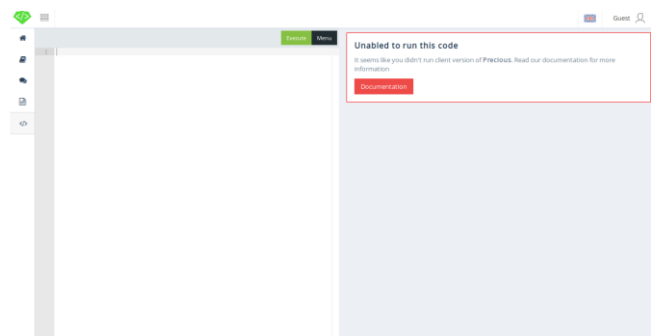
The concept of the system is to create a medium that allows users to share programs managed by instructors and learned by students. Users are also allowed to post topics to forums, write and execute code, and manage their code. The study also has a focus on implementing client-based processing to reduce server load on some processes. In this case, the authors apply it to compile and run the user code. Users will interact with the system using the website while the desktop application is opened in the user's PC by the system service running on the user tray to help the web application process the code. The main idea is to run the user program code (compiling and executing) shown in the web application. At the same time, it will also reduce the process that occurs on the server by using a client-based processing using the desktop application that run as a system tray application. Before the client runs the desktop application, it must have the compiler installed correctly and added to the path variable, so the desktop app will be able to call it to process the code. For e.g. if a user wants to execute Java in the web application, the user's PC must have JDK installed. Up to the current version, the system is capable of processing programming code that runs and outputs in a console as well as a web-based program that is related to HTML or Javascript. The system has not been able to process a GUI type of applications yet. The website and the desktop applications, which are part of the system are related apps. Therefore, they must remain opened at the same time to make all features run.



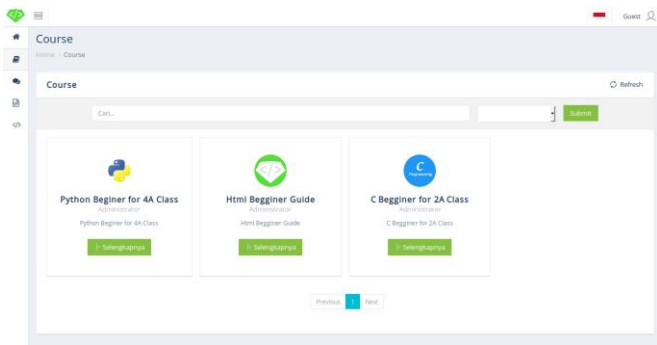
(a)



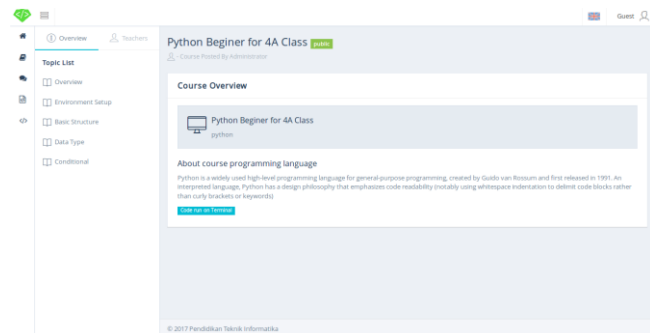
(b)



(c)



(d)



(e)

Fig 5 Implementation Preview

- (a) Code Editor Page Implementation, (b) Desktop Application Preview, (c) Editor Page on Error, (d) Course Lists Implementation, (e) Detail Course Implementation

G. Software Testing

Software testing that had been done is White Box testing, Black Box testing, Media Expert testing, Usability testing, and User Response testing. The goal of White Box testing is to test the algorithm of the system's code for several scenarios by using *basis path* testing.

In the *basis path* testing, there are several steps. First, the code that is going to be tested has to be numbered as shown in Fig 6.

```

1 module.exports.store = function(req, res){
2   var data = JSON.parse(req.body.data);
3   data.Icon = req.file.filename;
4   Language.create(data, function(err, user){
5     if (err){
6       res.send(err);
7     }
8     else{
9       res.send('1');
10    }
11  });
12 }

```

Fig 6 Numbered code

After the code was numbered, a flow-graph was then created based on the code that shown in Fig 7.

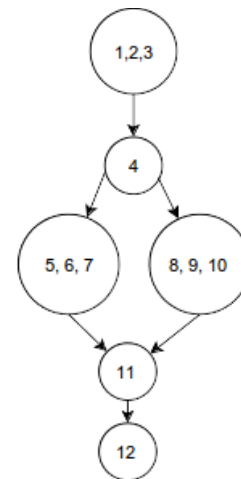


Fig 7 Flow-graph result

From that flow-graph, the Cyclomatic Complexity by using the following hree formulas.

$$\begin{aligned}
 V(G) &= \text{edges} - \text{nodes} + 2 \\
 &= 6 - 6 + 2 \\
 &= 2
 \end{aligned}$$

By that formula, the number of the Cyclomatic Complexity of this code was equals 2, which mean there are two possible routes or cases from that algorithm. By using this result, the code was tested by presenting various input so the code is run through every case.

From the White Box testing that was done, the algorithm that is implemented in the application code was considered appropriate. Black box testing is to complete functionality testing by using boundary analysis for all the form validation in the web application.

After testing White Box and Black Box finished, the system is then tested with the help of experts in the field of programming is done mainly in accordance with the content and workflow validity. This system is tested by several lecturers specializing in programming lectures and by the head of the information and communication technology department in Undiksha. From the results, there are some revisions, especially on the part of web applications. The last test conducted on the users who are the actual lecturer of some programming classes and some students of the Informatics Engineering Education Department of Undiksha by using questionnaire.

Usability test results show 91% in terms of learning ability, 89% efficiency, 85% memorability, 85% error-free, and 89% satisfaction. Therefore, the total usability test results is 87% which means the application is good enough to be implemented. The author also gets 95% overall for the user response testing.

V. CONCLUSION

The system developed in this research is used to support the programming class. In this case, applied in the Department

of Informatics Engineering Education in Undiksha. This system can help users learn the programming language more comfortably and has many features that make it easy for users so that users can focus more on coding such as writing code, executing it and maintaining it. The system also provides some helpful features such as forum and a feature to share program code snippets with other users. The client-based desktop application that has been developed can also reduce the server load significantly, so the server is not easily stuck because of the growing number of users and reduce the possibility of users damage the stability of the system in executing malicious programs that harm the system because the compilation and execution occurs on the user's computer.

#### ACKNOWLEDGMENT

Special thanks to the lecturers who are experts and teaching the programming language classes in the Department of Informatics Engineering Education in Undiksha and some students who have helped in evaluating the system during the system testing.

#### REFERENCES

- [1] M. Patel, "Online Java Compiler Using Cloud Computing," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, pp. 116-118, 2013.
- [2] A. Datta and A. K. Paul, "Online Compiler as a Cloud Service," *IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pp. 1783-1786, 2014.
- [3] A. Hartono, H. D. Purnomo and Y. R. Beeh, "Perancangan dan Implementasi Aplikasi Media Pembelajaran Bahasa Pemrograman Pascal pada Platform Android," *Jurnal Teknologi Informasi-Aiti*, 2014.
- [4] tutorialspoint, "Node.js Tutorial," tutorialspoint, 2017. [Online]. Available: <https://www.tutorialspoint.com/nodejs/>. [Accessed 18 July 2017].
- [5] tutorialspoint, "Electron Tutorial," tutorialspoint, 2017. [Online]. Available: <https://www.tutorialspoint.com/electron/index.htm>. [Accessed 18 July 2017].
- [6] J. Vaughan, "MongoDB," techtarget, 2017. [Online]. Available: <http://searchdatamanagement.techtarget.com/definition/MongoDB>. [Accessed 17 July 2017].
- [7] B. Mitchell, "lifewire," 19 March 2017. [Online]. Available: <https://www.lifewire.com/socket-programming-for-computer-networking-4056385>. [Accessed 2 August 2017].
- [8] V. Silverthorne, "rapid application development (RAD)," techtarget, 2017. [Online]. Available: <http://searchsoftwarequality.techtarget.com/definition/rapid-application-development>. [Accessed 17 July 2017].
- [9] techtarget, "use case," techtarget, 2017. [Online]. Available: <http://searchsoftwarequality.techtarget.com/definition/use-case>. [Accessed 18 July 2017].
- [10] techopedia, "Entity-Relationship Diagram (ERD)," techopedia, 2017. [Online]. Available: <https://www.techopedia.com/definition/1200/entity-relationship-diagram-erd>. [Accessed 17 July 2017].