



# An Interactive Learning System with Automated Assistance for Self-Learning User Interface Design on Android Applications

Yan Watequlis Syaifudin<sup>1</sup>, Indrazno Siradjuddin<sup>2</sup>, Nobuo Funabiki<sup>3</sup>, Dewi Yanti Liliana<sup>4</sup>, Andi Baso Kaswar<sup>5</sup>, Mustika Mentari<sup>1</sup>, Noprianto<sup>1</sup>, Nabilah Argyanti Ardyningrum<sup>1</sup> and Flasma Veronicha Hendryanna<sup>1</sup>

<sup>1</sup>Department of Information Technology, State Polytechnic of Malang, Malang, Indonesia

<sup>2</sup>Corresponding Author, Department of Electrical Engineering, State Polytechnic of Malang, Malang, Indonesia

<sup>3</sup>Department of Electrical and Communication Engineering, Okayama University, Okayama, Japan

<sup>4</sup>Department of Informatics Engineering, State Polytechnic of Jakarta, Jakarta, Indonesia

<sup>5</sup>Department of Computer Engineering, State University of Makassar, Makassar, Indonesia

Received 3 Oct. 2022, Revised 2 May. 2023, Accepted 14 May. 2023, Published 30 May. 2023

**Abstract:** The popularity of smartphone device usage has triggered the increasing demand for Android application developers in recent years. To master Android application development, students must learn various programming languages, including XML for designing the user interface (UI). However, many related courses do not offer learning topics in Android application UI design specifically. Moreover, the major issue in programming courses is the teacher's difficulties in checking and verifying the source codes from students. This study proposes a novel approach to support students' self-learning interactively by developing a self-learning system for designing Android application UI using XML code. To realize the comprehensive learning process, the pedagogical model is applied and consists of studying and assignments for exercise to make it ready for blended learning. The learning materials to study UI layout, widgets, string resources, color resources, and drawable resources were distributed into 20 learning topics to guide the studying process. In the assignment process of writing XML code, integration testing mechanism on Android applications is adopted to provide the features of automatic source code verification and automated learning assistance for students getting instant feedback, suggestion, and assistance to correct the source code. For a comprehensive evaluation, system usability and students' experiences have been assessed by appointing 80 IT students from two universities in Indonesia to use this system. The results have confirmed the system's effectiveness to support students' self-learning process, learning materials appropriateness, and the high grade of user satisfaction and enthusiasm. Also, this system has successfully led all students to master UI design skills on Android applications using XML code after completing the 20 learning topics, even though at the beginning of the learning, the students have a poor coding experience.

**Keywords:** e-learning, automated assistance, Android, user interface, XML, interactive learning

## 1. INTRODUCTION

Recently, the smartphone market's rapid growth has shown a significant impact on society in this decade [1]. It reveals that the use of smartphones has penetrated various economic sectors [2]. Even consistently, smartphones have largely replaced the role of personal computers as computing tools. As the most popular operating system for smartphone devices, Android has been adopted on more than 70% of smartphone devices worldwide [3]. The Android application development courses have also grown up in many universities and professional schools.

In mobile applications, the user interface plays a very important role in creating interactive applications. The programmer must use XML code to design the UI on the IDE of *Android Studio*. It supports user experience

and engagement when running Android applications[4]. The latest Android versions present more attractive and impressive UI concepts. Therefore, every mobile application programmer must have the ability to design the UI for the application. However, many mobile programming courses do not provide the subject for UI design learning specifically.

The major issue in programming courses is the teacher's difficulties in checking and verifying the source codes from students. With this condition, students must wait for the teacher's responses to get suggestions or feedback on the written source code. To make the student's learning easier in computer programming, a system or platform that can provide self-learning with automatic code verification[5] is necessary. However, recently, there is no self-learning



system or e-learning platform that specifically provides learning materials for Android application *UI design* using XML, instead of automatic *XML code* verification.

This study proposes a novel approach to support students' self-learning interactively by developing a self-learning system for designing Android application UI using XML code as an extension of our previous work, *Android Programming Learning Assistance System (APLAS)*[6]. To realize the comprehensive learning process, the pedagogical model in the real course is applied and consists of studying and assignments for exercise to make it ready for blended learning. To guide the studying process, the learning materials provide guidance documents for students learning the concept and the code writing practical step by step. In the assignment process of writing XML code, this system adopts an *integration testing* mechanism on Android applications to provide the features of *automatic source code verification* and *automated learning assistance*[7]. With these features, the student can get instant feedback, suggestion, and assistance to correct the source code. By using this system, the student can study UI design by themselves from anywhere.

The learning materials consist of 20 learning topics that provide learning for the utilization of *UI layouts*, *widgets*, *string resources*, *color resources*, and *drawable resources*. Each learning topic provides learning guidance and an assignment to be solved by the student. For mastering UI layout, the subjects for learning four UI layouts[8] including *Linear Layout*, *Relative Layout*, *Grid Layout*, and *Constraint Layout* are presented to students. Some of the most widely used widgets are also introduced in the learning materials. Meanwhile, for developing UI structure and content, materials on how to utilize string, color, and drawable resources are presented as well.

Finally, to apply a comprehensive evaluation, we assessed aspects related to students' self-learning effectiveness and students' learning experiences using this system[9]. 80 IT students from two universities in Indonesia have been appointed to study and solve the assignments on 20 learning topics using this system. The results show that the student success in solving all assignments in less than four days. The system has successfully led all students to master UI design skills on Android applications using XML code after completing the 20 learning topics. The system's effectiveness to support students' self-learning process, learning materials appropriateness, and the high grade of user satisfaction and enthusias have also confirmed. Moreover, this system has successfully led all students to master UI design skills on Android applications using XML code after completing the 20 learning topics, even though at the beginning of the learning, the students have a poor coding experience.

This paper is organized into eight sections. Section 2 presents related research on programming self-learning sys-

tems. Section 3 explains the concept of Android application UI design. Section 4 proposes the design of learning materials for designing the Android application user interface. Section 5 explains the interactive self-learning assistance system that applies the designed learning materials. Section 6 presents the evaluation of students' self-learning effectiveness by using this system. Section 7 discusses the results related to three evaluation aspects. Finally, section 8 resumes the conclusion about the proposed system and provides future works.

## 2. RELATED STUDIES

There are few studies related to technology enhanced learning for XML programming, moreover for UI design. In this section, related works in learning systems with automated assistance features and XML code learning tools are discussed. Funabiki et. al. proposed a learning assistance system namely JPLAS or Java Programming Learning Assistance System in 2013[10]. It applied test-driven development (TDD) method to provide an automatic assistance feature for learning Java programming. The Java code from students was validated using JUnit tool by executing unit testing. In their following studies, some improvements in content and learning method have been implemented[11].

The works related to technology enhanced learning for computer programming still use general programming cases, such as C and Java. Yulianto et al., in 2014, presented SCAGrader[9], a grading analyzer to verify source code automatically for several programming assignments including Java and C. As a computer application, it consists of a Java-based application and a web application. Staubitz et. al., in 2016, presented a web-based learning platform based on MOOC for programming course practices, namely CodeOcean[12]. It provided a model for teachers to prepare programming exercises for beginners. Unit testing was utilized to grade the source code automatically so that it covered a scalable code execution and assessment. Hamdi et al., in 2019, proposed a new framework for the development of e-content for programming learning[13]. The encountered problems in computer programming learning and the cognitive theory for learning were presented.

Recently, the focus of e-learning systems for computer programming is still on learning content and materials. So, in the pedagogical model, the common system just provides learning material for studying. The typical e-learning system or *LMS (Learning Management System)*[14], such as *Moodle*[15] or other *MOOCs* providers[16], commonly provides standard types of exercise, such as multiple choice or fill in the blank, instead of long code verification. Automation in programming assignment verification still stands to be the main issue in programming learning systems[5][17].

## 3. CONCEPT OF ANDROID UI DESIGN

This section presents five components to construct the UI, namely Layout, Widgets, String resource, Color resource, and Drawable resource.

### A. Layout

A layout of UI is the skeletal structure of visual elements (widgets) and space placement of UI design[8]. A fluid layout provides a self-adjustment to each widget positioning on the available screen as the fundamental idea behind responsiveness. Fluid layout is an umbrella term for making the design responsive.

As shown in Figure 1, Android provides several types of fluid layouts including Linear Layout, Relative Layout, Grid Layout, and Constraint Layout. Linear Layout is a layout in which the UI content is laid out unidirectionally. The page is scrollable and all the content scales equally with the screen. Relative Layout uses relative length units that can be used to design the responsiveness of a child element (content or UI) which needs to scale proportionally to its parent element.

Grid Layout is a layout that embraces a grid mode that consists of Columns, Gutters, and Margins, then the content is spread out in these Columns. When screen dimensions change, the Grids will be adjusted proportionally and the content and UI inside are ensured to be consistent while scaling on any device. Constraint Layout provides the creation of large and complex layouts with a flat view hierarchy. It is more flexible than the Relative Layout in that all UI components are laid out according to connections between sibling components and the parent layout.

### B. Widgets

In Android applications, widgets are the collections of UI components to interact with user[18]. The components are the main parts for creating interactive applications based on UI. They are provided as a package of Android widgets. Android also allows for creating self-design widgets by extending the View class[19]. The typically utilized widgets for Android application's UI include:

- **TextView:** a widget to show a text,
- **EditText:** a widget to allow entering and editing text by user,
- **Button, RadioButton, CheckBox, and Spinner:** widgets to construct interactive behaviors,
- **ProgressBar:** a widget to indicate the value of a progress,
- **Toast:** a widget to show text on the screen in a short time,
- **ImageView:** a widget to show images,
- **Switch:** a widget to set the position of boolean conditions.

### C. String, color, and drawable resources

Android provides application resources as additional files and static content in XML code that can be accessed

by other codes[20]. String resource in Android represents a reference set of string values. It provides text strings for the application with optional text styling and formatting. There are three types of string resources, including String, String Array, and Quantity Strings. String is XML resource that provides a single string. String Array is XML resource that provides an array of strings. Quantity Strings (plurals) are XML resource that carries different strings for pluralization. A string value is defined by referring to an ID in string resources, as shown in Figure 2.

Color resource is similar to string resources, but a color resource specifically stores a reference set of color values. XML code is adopted to define a color value in RGB value and alpha channel. It can be used in a UI layout, as shown at end of the code in Figure 2. Defining the collections of color values in color resources helps deliver constant color, implement the visual theme, and improve the quality of look and feel of the application's UI.

Drawable resource is a type of resource that is specifically implemented in Android Studio. A drawable is a general abstraction for something that can be drawn. So that, the drawable resource is a general concept for a graphic that can be drawn to the screen and involves multi colors, so that it can deliver a spectrum or gradient color and painted result instead to define the widget's shape and state. There are several types of drawable, such as Bitmap, Layer List, State List, Transition Drawable, and Shape Drawable.

## 4. PROPOSAL OF INTERACTIVE LEARNING SYSTEM WITH AUTOMATED ASSISTANCE

In this section, the proposal of an interactive learning system with *automated assistance* for self-learning designing Android application UI is explained. The proposal includes the automated assistance model, pedagogical model, and design of learning topics.

### A. Model of Automated Assistance for Student's Self-learning

This system provides two functions verifying the student's answer code and generating feedback on answer incorrectness to apply an automated assistance model. Referring to the APLAS's automatic validation model, this system contains a validator program that can verify the XML code from students and generate feedback if the answer incorrectness occurs. Figure 3 shows the four steps of the self-learning mechanism that applies automatic verification for answer code from the student. In the first step, the student writes the assignment answer as XML code, then he/she submits the answer to the server that contains the validator program in the second step. In the third step, the validator program verifies the code automatically based on defined criteria of validity (test cases). Finally, in the fourth step, the *validator program* sends the results to the student. They contain a validation checklist report, feedback on failed test cases, and suggestions to fix them. With feedback and suggestions, the student gets assistance in writing the correct XML code[21].

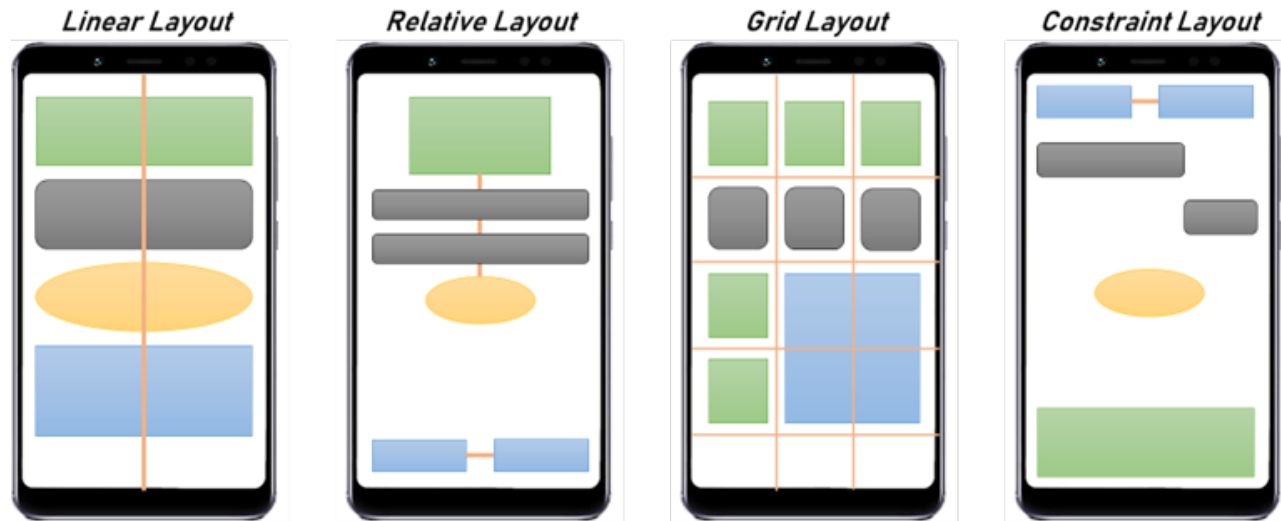


Figure 1. Four types of fluid layout.



Figure 2. Application resources in Android application.

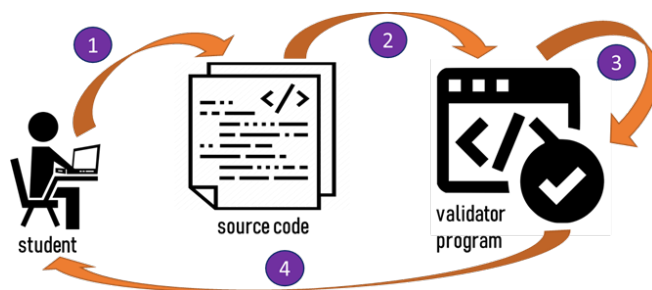


Figure 3. The self-learning assistance system model with the automatic validator.

### B. Learning Topics

To provide a step-by-step learning process[22], the 5 Android UI subjects, including UI Layout, Widgets, String resources, Color resources, and Drawable resources, are distributed into 20 learning topics. Table I presents the details of 20 learning topics.

### C. Pedagogical Method

As a supporting system for mobile programming class students, the pedagogical method adopts stages in the course learning process which consists of studying and assignments for exercise[23]. The learning materials are composed of 20 learning topics to provide a step by step learning process. In each learning topic, a guidance document is provided for the studying stage and exercise stage. As illustrated in Figure 4, the learning process starts from the first topic by getting the guidance document. In the studying stage, the student studies a concept related to the topic description. Then, the student must solve the provided assignment by writing XML code. The answer code will be verified by the system to decide whether the student has passed the current topic and he/she can go to the next topic.

## 5. SYSTEM IMPLEMENTATION

In this section, the system architecture, the web interface for self-learning, and the automatic XML code verification will be explained.

### A. System Architecture

Based on APLAS architecture[7], a client-server model is applied to implement the interactive self-learning assistance system. For integration with APLAS, the following software are adopted to develop the system:

- **Laravel:** a platform for web application development,
- **MySQL:** a database management system software to store the data,
- **Java 11:** a platform to develop validator program.

### B. Web interface for self-learning

For student's self-learning, the system uses a web interface that has four main areas as shown in Figure 5, including:

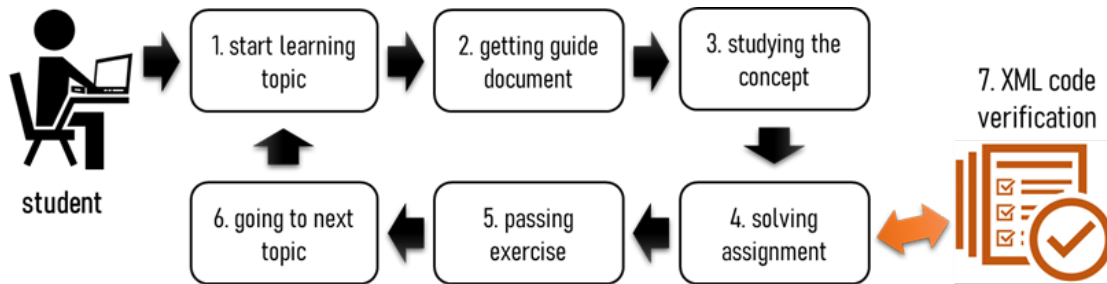


Figure 4. Pedagogical method for a learning topic.

- **Top side** provides information related to the topic name and four buttons for going to the previous topic, submitting feedback, submitting the answer code to the server, and going to the next topic.
- **Middle left** side displays a PDF document as guidance for the related topic.
- **Middle right** side displays an XML code editor to write the answer code.
- **Bottom side** displays a table to show the verification results for the answer code.

The features in XML code editor include code auto-completion and syntax highlighting to make code writing easier and more interactive.

### C. Automatic XML code verification

To support students in self-learning, the proposed system must provide the automatic verification of students' answers. In this case, the automatic verification process confirms the XML code that student writes as assignment answers. In the previous work, using unit testing and integration testing, the automatic code verification model for *Java* and *Kotlin* in Android applications has been implemented in APLAS (Android Programming Learning Assistance System). Based on the model, integration testing for Android applications[24] is adopted to verify the XML code for the application's UI.

Figure 6 shows the process of a student's solving activity and XML code verification. In the server, a validator program handles the answer code. First, the validator program packages the code as an Android project to make it ready for integration testing execution. Then, the integration testing process produces testing results and messages for failed testing. The failed messages provide suggestions for answer code correction by the student.

### D. Android integration testing

Integration testing is a type of testing to verify the functionality of software when all its parts work together[25]. Android application has UI and logical (unit) parts that require integration testing to verify their combination using Java code. *Robolectric* is utilized as an integration testing

tool because of its simplicity in the integration process and no emulator used for compiling and testing the Android application[26]. Figure 7 presents examples of Java code to verify application resources consisting of string, color, and string-array resources.

## 6. SYSTEM EVALUATION IN BLENDED LEARNING

In this section, the evaluation of interactive self-learning assistance system through blended learning application in two mobile programming classes is presented. Blended learning is a learning environment that combines direct (synchronous) and independent/indirect (asynchronous) learning methods that can be done anytime[27].

### A. Classroom scenario

The two mobile programming classes in two universities in Indonesia have been appointed for evaluation. Each of them contains 40 students (a total of 80) who have been assigned to solve all assignments in 20 learning topics. During five working days, they had to solve the assignments sequentially on the web application that can be accessed from their PCs using any web browser. For each topic, they could give feedback after solving the assignment. The progress of each student and the correctness of each answer have been observed for evaluation. The effectiveness of each topic can be influenced by its success percentage and student feedback.

### B. Assignment solving result

In five days, all students in two classes have successfully solved all assignments in 20 learning topics. The average time to solve all assignments was 3.7 days with a range between 2 to 5 days. The solving time and the submission numbers were diverse for each learning topic. Figure 8 presents the average times in submitting answers when solving each learning topic with 1.44 average times for all learning topics. The result was less than 2 times for submitting the answers, it confirmed that the students could be familiar with the system, and they easily solved the assignments in all topics. However, as expected for the utilization of self-learning tools, most students had to try and submit the answer several times to get a passed result.

### C. Difficulty levels

The chart in Figure 8 shows peaks for four learning topics that have values greater than the average value,

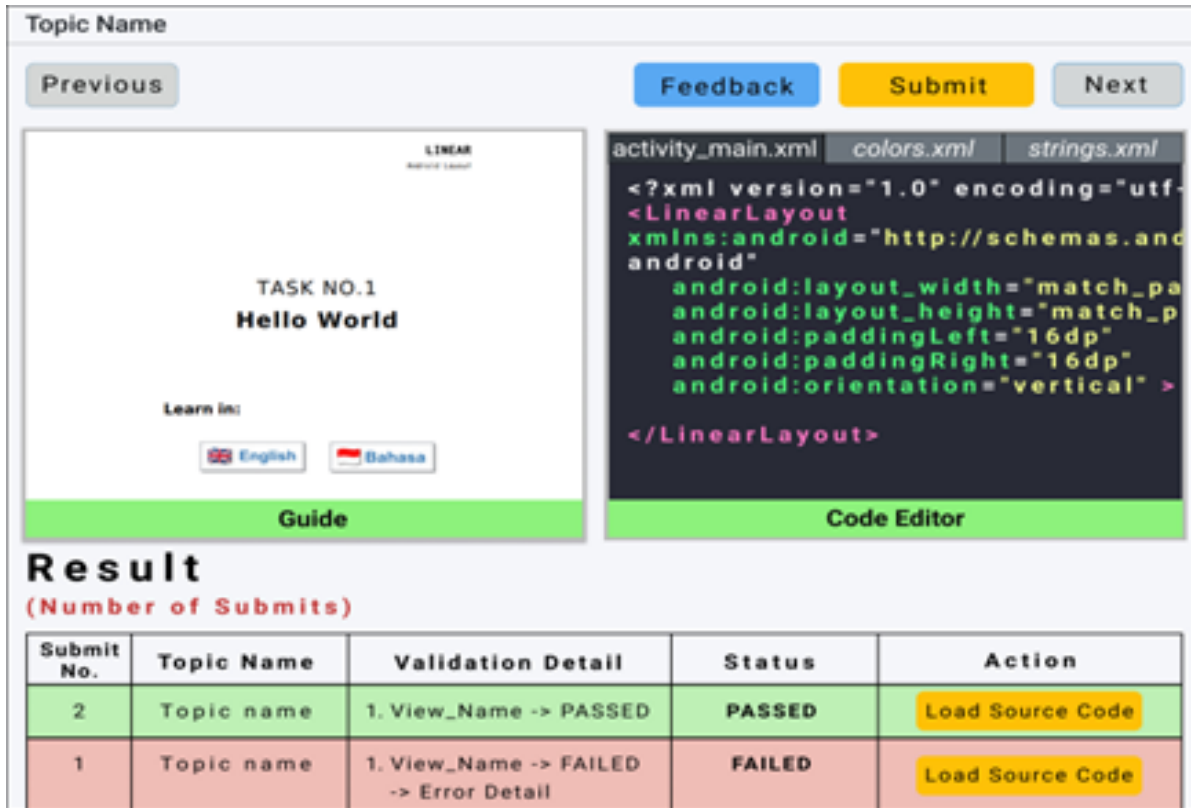


Figure 5. Web interface design for learning and solving assignment.

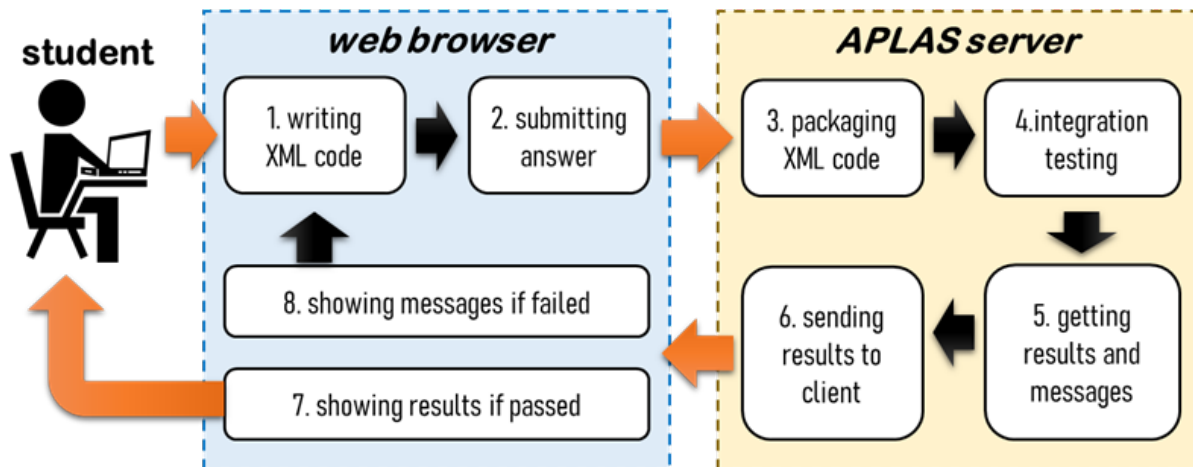


Figure 6. XML code verification process.

including topic 1, topic 6, topic 10, and topic 16. The first topic has high value because students begin the self-learning process from this topic so that they just got XML code introduction for Android UI, and they need adaptation time to be familiar with the system. In topic 6, students started learning using an XML code for the application's resource that was started with string resource. In topic 10, students had the most difficult assignment to build a complex UI.

The last peak was topic 16 where students started to learn drawable resources by implementing bitmap resources. In these four topics, students needed several answer revisions to achieve passed results.

Based on the chart in Figure8, there were three levels of difficulty classification of the 20 learning topics, as presented in Table II. The easy level had values below 1.5 and there were 16 topics in this level. It can be confirmed

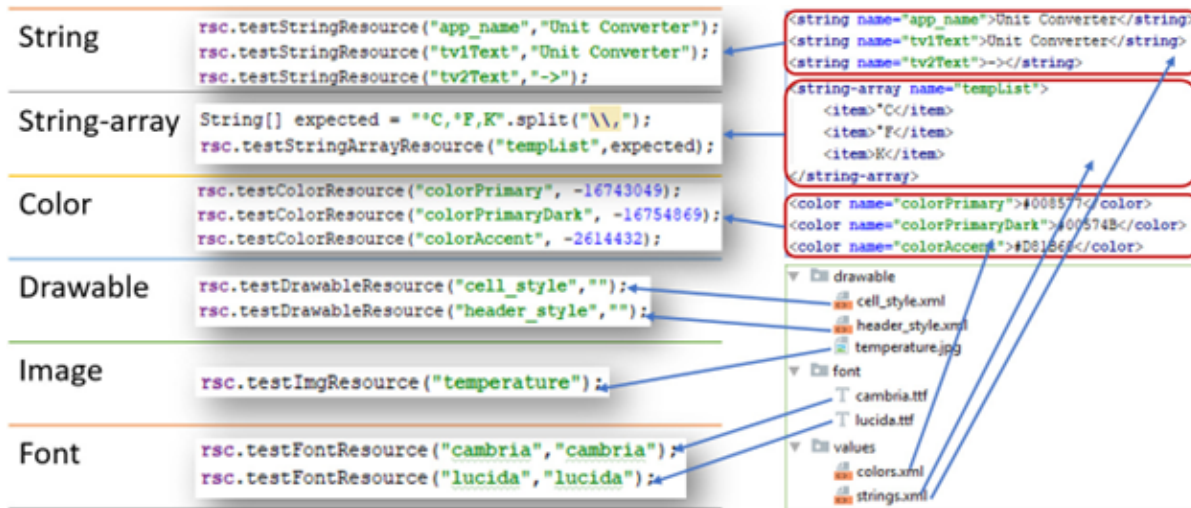


Figure 7. The algorithm of integration testing to verify application resources.

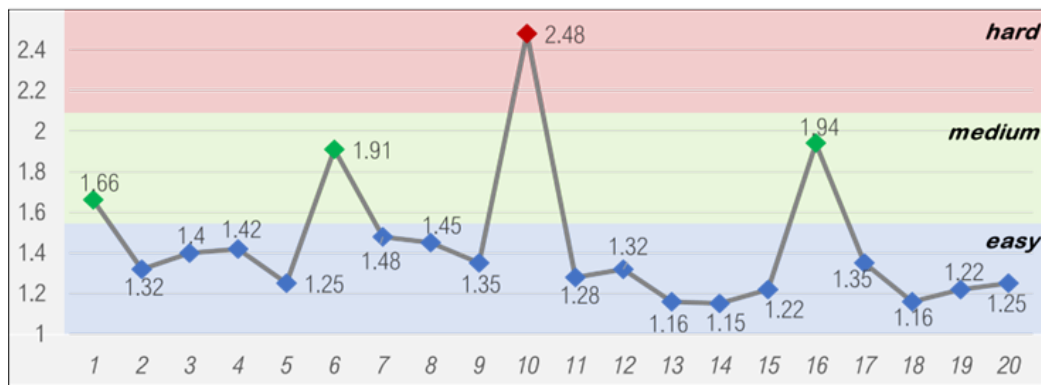


Figure 8. The average times in submitting answers by 60 students on each learning topic.

that most students can use this system and learn Android UI design easily. The medium level consisted of topic 1, topic 6, and topic 16 as topics where students initiated to use the self-learning system and start learning new subjects in Android UI design. The hard level only contained topic 10 which provided the most difficult assignment to build a complex UI application[28].

## 7. DISCUSSIONS

This section presents discussions on system usability score, self-learning effectiveness while using this system, students' learning experiences[29], and the comparisons to other systems.

### A. System usability

The System Usability Scale (SUS)[30] is a measurement tool that can be used to measure the usability level of a system on various products such as hardware, software, mobile apps, to websites. There are 10 standard questions[31] for the 80 students from two universities. The SUS applies a Likert scale of one to five[32], namely 1 for strongly disagree, 2 for disagree, 3 for neutral, 4 for agree, and 5

for strongly agree. The 10 questions of the system usability scale questionnaire must be arranged sequentially and must be answered sequentially too by students.

The score from 80 students spread from 70 to 85. The average score is 80.5, which means the presented system's usability got an excellent opinion from university students. The reason for low scores is mostly from the student's perception of Android programming that must use multiple languages, also the unclear software manual documentation. The reason for high scores is mostly from the feature of automated assistance that helped students to write the correct source code. Also, the instant feedback has made them finish all learning materials quickly. It confirms their satisfaction in using this system for Android UI design learning[33].

### B. Self-learning effectiveness

The evaluation results in blended learning confirm the effectiveness of this interactive learning system[34]. The ability to supply automated assistance in writing XML code has helped students to write correct XML code for Android



TABLE I. THE 20 LEARNING TOPICS THAT COVER 5 ANDROID UI SUBJECTS

No.	Topic Name	Description
1	Hello world	the use of TextView on LinearLayout by creating Hello World application.
2	Email and password	the use of EditText on Layout to input email and password.
3	Exit and login	the use of simple Button on LinearLayout.
4	Nested layout	the use of two LinearLayouts combination to create a Nested layout with different orientation.
5	Login form	the use of combination of TextView, EditText, Buttons on LinearLayout to create login form.
6	Using string resource	the use of string resources for UI layout that implements RelativeLayout.
7	Using color resource	the use of color resources and widgets positioning on RelativeLayout.
8	Using ImageView	the use of ImageView as widget and its positioning on RelativeLayout.
9	Advanced login form	the use of combination of ImageView, TextView, EditText, Buttons on RelativeLayout to create improved login form.
10	Using constraint layout	the arrangement of widgets on ConstraintLayout.
11	Picture in picture	the setup of an ImageView position on another ImageView as Picture in picture with ConstraintLayout.
12	E-book store	the use of combination of ImageView, TextView, Divider, Buttons to create e-book store application.
13	Using grid layout	the placement of widgets on GridLayout.
14	Score card app	the utilization of TextView on GridLayout to create Score card application.
15	Image sharing app	the utilization of TextView, ImageView, and Button on GridLayout to create Score card application.
16	Bitmap drawable	the use of Bitmap drawable to show an image on layout.
17	Layer list	the use of Layer list to create a multi-layer view on UI.
18	State list	the implementation of State list for a Button to create animated button view.
19	Shape drawable	the use of Shape drawable to define the shape and color combinations of Button and EditText.
20	Scale drawable	the use of Scale drawable to define the size of another drawable or widget.

TABLE II. THE THREE LEVELS OF DIFFICULTIES

Level	Value Range	Learning Topic Number
easy	< 1.5	2, 3, 4, 5, 7, 8, 9, 11, 12, 13,14, 15, 17, 18, 19, 20
medium	1.5 – 2.0	1, 6, 16 (green)
difficult	> 2.0	10 (red)

application UI. Figure 9 shows the validation results for a student in solving the assignment of the first learning topic.

The student submitted the answer three times which means he/she had been rewriting the answer code twice referring to the resulting messages on the “validation detail” column for getting the passed result (green row). The results on first submission show one failed message due to incorrectness in XML syntax. The second submission produced three failed messages for the missing several attributes. After fixing the code, he/she got passed result on the third submission. This kind of learning experience has happened to most of the 80 students in the evaluation which means this system has successfully led all students to master UI design skills on Android applications using XML code after completing the 20 learning topics, even though at the beginning of the learning, the students have a poor coding experience.

### C. Comparisons with other systems

This web-based interactive learning system is a novel model for Android programming self-learning, especially for designing Android application UI using XML code. Students can study Android UI design interactively by using web browser. Referring to the implemented pedagogical model, they can learn the concept and practice of XML code to design the UI. The guidance documents in the learning materials can direct them to write XML code step by step. Then the written code can be verified by the system and they can get the results instantly. The feedback and suggestion will be generated if the submitted XML code is incorrect, which means the students get assistance from the system to correct the XML code.

Reflecting on Section 2 of Related Studies, there are few studies related to technology-enhanced learning for XML programming, specifically for Android UI design. The focus of the e-learning systems for computer programming is still on the learning content and materials. So, in the pedagogical model, the common system just provides learning material for studying. The typical e-learning system or LMS (Learning Management System)[14], such as Moodle[15] or other MOOCs providers[16], commonly provides standard types of exercise, such as multiple choice or fill in the blank, instead of long code verification. Automation in programming assignment verification still stands to be the main issue in programming learning systems[5][17]. A novel approach for Java code verification has been implemented by Funabiki et. al. in their system called JPLAS or Java Programming





Result No.	Topic Name	Validation Detail	Status
3	Linear Layout - Hello World	1. check_01_Layout_Specification -> PASSED -> Success 2. check_02_helloTextView_Properties -> PASSED -> Success 3. check_03_secondTextview_Properties -> PASSED -> Success * VALIDATION RESULT : Passed 3 method(s) (100.0 %) from 3 method(s)	PASSED
2	Linear Layout - Hello World	1. check_01_Layout_Specification -> FAILED -> android.view.InflateException: Binary XML file line #9: Binary XML file line #9: You must supply a layout_height attribute. 2. check_02_helloTextView_Properties -> FAILED -> android.view.InflateException: Binary XML file line #9: Binary XML file line #9: You must supply a layout_height attribute. 3. check_03_secondTextview_Properties -> FAILED -> android.view.InflateException: Binary XML file line #9: Binary XML file line #9: You must supply a layout_height attribute. * VALIDATION RESULT : Passed 0 method(s) (0.0 %) from 3 method(s)	FAILED
1	Linear Layout - Hello	1. check_01_Layout_Specification -> FAILED -> java.lang.NullPointerException at org.aplas.helloworld.ViewTest.testCompletion(ViewTest.java:30) at org.aplas.helloworld.TestLinearHello.checkCompletion(TestLinearHello.java:59)	FAILED

Figure 9. The validation results with suggestions in the web application.

Learning Assistance System in 2013. It applied test-driven development (TDD) method[35] to provide an automated assistance feature for learning Java programming. Finally, the verification of XML code in this system is a new proposal that adopts their approach by combining it with integration testing for Android applications[36].

### 8. CONCLUSION

In Android application development, UI design is an important part to develop a complete application. To design the UI, Android Studio provides XML code to make structure and build the UI of each application’s screen. In addition, the major issue in programming courses is the teacher’s difficulties in checking and verifying the source codes from students. This paper proposed an interactive learning system for students’ self-learning Android application UI design with its implementation and evaluation. An automated assistance feature and a pedagogical model have been applied to provide a self-learning model. The model of automated assistance feature has adopted the TDD model in APLAS using integration testing. The pedagogical model has applied the learning stages which consist of studying and assignments for exercises. The learning materials were divided into 20 learning topics with specific subjects and learning objectives.

A comprehensive evaluation was conducted by applying this system for UI design study to 80 IT students from two universities to assess the system’s usability and students’ experiences. The evaluation results confirm the usefulness and effectiveness of this system for students’ self-learning, learning materials appropriateness, and the high grade of user satisfaction and enthusiasm. System compatibility to

be accessed using five types of browsers has also been confirmed. Finally, this system has successfully led all students to master UI design skills on Android applications using XML code after completing the 20 learning topics, even though at the beginning of the learning, the students have a poor coding experience. For future works, the feature to detect code plagiarism and record the solving time will become a priority to be implemented. Also, we will implement the learning system for generally used XML coding such as web service and metadata definition.

### ACKNOWLEDGEMENT

This article is one of the research grant outputs that was funded by the Ministry of Education, Culture, Research, and Technology, precisely the Directorate General of Vocational Education. The research grant is a part of the Vocational Products Research scheme in 2022. The authors are very grateful to the Computer Center of State Polytechnic of Malang, Department of Information Technology of State Polytechnic of Malang, Department of Computer Engineering of State University of Makassar, and Intelligent System and Digital Economic Innovation (ISDEI) Research Center.

As a result of research collaboration between four institutions including State Polytechnic of Malang, Okayama University Japan, State Polytechnic of Jakarta, and State University of Makassar, the authors present the highest appreciation to all academicians, professors, research partners, and students from MBKM program. We are really grateful for this collaboration work.



## REFERENCES

- [1] S. Alam, "Reviewing the global smartphone industry strategic implication in response to covid-19 situation," *International Journal of Interactive Mobile Technologies (IJIM)*, vol. 15, no. 14, p. pp. 169–192, Jul. 2021. [Online]. Available: <https://online-journals.org/index.php/i-jim/article/view/18879>
- [2] Y. Fan and C. Yang, "Competition, product proliferation, and welfare: A study of the us smartphone market," *American Economic Journal: Microeconomics*, vol. 12, no. 2, pp. 99–134, May 2020. [Online]. Available: <https://www.aeaweb.org/articles?id=10.1257/mic.20180182>
- [3] F. M. Götz, S. . Stieger, and U.-D. Reips, "Users of the main smartphone operating systems (ios, android) differ only little in personality," *PLoS One*, vol. 12, no. 5, pp. 1–18, May 2017. [Online]. Available: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0176921>
- [4] M. Seraj and C. Y. Wong, "A study of user interface design principles and requirements for developing a mobile learning prototype," in *2012 International Conference on Computer Information Science (ICIS)*, vol. 2, 2012, pp. 1014–1019.
- [5] C. O'Halloran, "Automated verification of code automatically generated from simulink," *Automated Software Engineering*, vol. 20, pp. 237–264, 06 2012.
- [6] Y. W. Syaifudin, N. Funabiki, M. Kuribayashi, and W.-C. Kao, "A proposal of android programming learning assistant system with implementation of basic application learning," *International Journal of Web Information Systems*, vol. 16, no. 1, pp. 115–135, Jan 2020. [Online]. Available: <https://doi.org/10.1108/IJWIS-08-2019-0038>
- [7] Y. W. Syaifudin, N. Funabiki, M. Mentari, H. E. Dien, I. Mu'aasyiqiin, M. Kuribayashi, and W.-C. Kao, "A web-based online platform of distribution, collection, and validation for assignments in android programming learning assistance system," *Engineering Letters*, vol. 29, no. 3, pp. 1178–1193, Sep 2021. [Online]. Available: [https://www.engineeringletters.com/issues\\_v29/issue\\_3/EL\\_29\\_3\\_46.pdf](https://www.engineeringletters.com/issues_v29/issue_3/EL_29_3_46.pdf)
- [8] W. Jackson, *Android UI Layouts: Layout Containers and the ViewGroup Class*. Berkeley, CA: Apress, 2014, pp. 29–59. [Online]. Available: [https://doi.org/10.1007/978-1-4302-4987-0\\_2](https://doi.org/10.1007/978-1-4302-4987-0_2)
- [9] R. Hernández, R. Ventura, S. A. Guerra Moya, and F. C. C. Rico, "Acceptance and assessment in student pair-programming: A case study," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 16, no. 09, p. pp. 4–19, May 2021. [Online]. Available: <https://online-journals.org/index.php/i-jet/article/view/18693>
- [10] N. Funabiki, Y. Matsushima, T. Nakanishi, K. Watanabe, and N. Amano, "A java programming learning assistant system using test-driven development method," *IAENG International Journal of Computer Science*, vol. 40, no. 1, pp. 38–46, Mar. 2013.
- [11] S. T. Aung, N. Funabiki, S. L. A. N. K. D. Yan Watequlis Syaifudin, Htoo Htoo Sandi Kyaw, and W.-C. Kao, "A proposal of grammar-concept understanding problem in java programming learning assistant system," *Journal of Advances in Information Technology*, vol. 12, no. 4, pp. 342–350, Nov 2021. [Online]. Available: <https://doi.org/10.12720/jait.12.4.342-350>
- [12] T. Staubitz, H. Klement, R. Teusner, J. Renz, and C. Meinel, "Codeocean - a versatile platform for practical programming exercises in online environments," in *2016 IEEE Global Engineering Education Conference (EDUCON)*, 2016, pp. 314–323.
- [13] M. Hamdi and T. M. Hamtini, "Designing an effective e-content development framework for the enhancement of learning programming," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 11, no. 04, p. pp. 131–141, Apr. 2016. [Online]. Available: <https://online-journals.org/index.php/i-jet/article/view/5574>
- [14] D. Turnbull, R. Chugh, and J. Luck, "Learning management systems: a review of the research methodology literature in australia and china," *International Journal of Research & Method in Education*, vol. 44, no. 2, pp. 164–178, 2021. [Online]. Available: <https://doi.org/10.1080/1743727X.2020.1737002>
- [15] S. Walpita Gamage, J. Ayres, and M. Behrend, "A systematic review on trends in using moodle for teaching and learning," *International Journal of STEM Education*, vol. 9, 01 2022.
- [16] B. Abdelghani, "Trends in moocs research: Analysis of educational technology journals," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 15, no. 17, p. pp. 47–68, Sep. 2020. [Online]. Available: <https://online-journals.org/index.php/i-jet/article/view/14637>
- [17] S. Combéfis, "Automated code assessment for education: Review, classification and perspectives on techniques and tools," *Software*, vol. 1, no. 1, pp. 3–30, 2022. [Online]. Available: <https://www.mdpi.com/2674-113X/1/1/2>
- [18] W. Jackson, *Android UI Widgets: User Interface Widgets and the View Class*. Berkeley, CA: Apress, 2014, pp. 61–94. [Online]. Available: [https://doi.org/10.1007/978-1-4302-4987-0\\_3](https://doi.org/10.1007/978-1-4302-4987-0_3)
- [19] C. Cao, H. Ge, T. Gu, J. Deng, P. Yu, and J. Lu, "Accelerating automated android gui exploration with widgets grouping," in *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, 2018, pp. 268–277.
- [20] A. Developers. (2022) App resources overview. [Online]. Available: <https://developer.android.com/guide/topics/resources/providing-resources>
- [21] X. Lv, "A study on the application of automatic scoring and feedback system in college english writing," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 13, no. 03, p. pp. 188–196, Mar. 2018. [Online]. Available: <https://online-journals.org/index.php/i-jet/article/view/8386>
- [22] B. S. Xia, "A pedagogical review of programming education research: What have we learned," *International Journal of Online Pedagogy and Course Design*, vol. 7, pp. 33–42, 01 2017.
- [23] S. M. Salleh, Z. Shukur, and H. M. Judi, "Analysis of research in programming teaching tools: An initial review," *Procedia - Social and Behavioral Sciences*, vol. 103, pp. 127–135, 2013, 13th International Educational Technology Conference. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877042813037622>
- [24] P. Kong, L. Li, J. Gao, K. Liu, T. F. Bisseyandé, and J. Klein, "Automated testing of android apps: A systematic literature review," *IEEE Transactions on Reliability*, vol. 68, no. 1, pp. 45–66, 2019.
- [25] J.-W. Lin, N. Salehnamadi, and S. Malek, "Test automation in open-source android apps: A large-scale empirical study," in *2020*

35th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2020, pp. 1078–1089.

- [26] G. Grano, A. Ciurumelea, S. Panichella, F. Palomba, and H. C. Gall, “Exploring the integration of user feedback in automated testing of android applications,” in *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2018, pp. 72–83.
- [27] J. Zhang, “Blended learning innovation model among college students based on internet,” *International Journal of Emerging Technologies in Learning (iJET)*, vol. 13, no. 10, p. pp. 158–170, Oct. 2018. [Online]. Available: <https://online-journals.org/index.php/i-jet/article/view/9454>
- [28] Y. Syaifudin, N. Funabiki, M. Kuribayashi, and W.-c. Kao, “A proposal of advanced widgets learning topic for interactive application in android programming learning assistance system,” *SN Computer Science*, vol. 2, 05 2021. [Online]. Available: <https://doi.org/10.1007/s42979-021-00580-1>
- [29] F. Bouilheres, L. Le, S. McDonald, C. Nkhoma, and L. Jandug, “Defining student learning experience through blended learning,” *Education and Information Technologies*, vol. 25, pp. 1–21, 07 2020.
- [30] P. Vlachogianni and N. Tselios, “Perceived usability evaluation of educational technology using the system usability scale (sus): A systematic review,” *Journal of Research on Technology in Education*, vol. 54, no. 3, pp. 392–409, 2022. [Online]. Available: <https://doi.org/10.1080/15391523.2020.1867938>
- [31] D. Derisma, “The usability analysis online learning site for supporting computer programming course using system usability scale (sus) in a university,” *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 09, p. pp. 182–195, Jun. 2020. [Online]. Available: <https://online-journals.org/index.php/i-jim/article/view/13123>
- [32] O. Ivanov, V. Ivanova, and A. Saltan, “Likert-scale questionnaires as an educational tool in teaching discrete mathematics,” *International Journal of Mathematical Education in Science and Technology*, pp. 1–9, 01 2018.
- [33] N. Harrati, I. Bouchrika, A. Tari, and A. Ladjailia, “Exploring user satisfaction for e-learning systems via usage-based metrics and system usability scale analysis,” *Computers in Human Behavior*, vol. 61, pp. 463–471, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0747563216302229>
- [34] A. Alaqsam, D. Ghabban, O. Ameerbakhsh, I. Alfadli, and A. Fayez, “Current trends in online programming languages learning tools: A systematic literature review,” *Journal of Software Engineering and Applications*, vol. 14, pp. 277–297, 01 2021.
- [35] S. Mäkinen and J. Münch, “Effects of test-driven development: A comparative analysis of empirical studies,” in *Software Quality. Model-Based Approaches for Advanced Software and Systems Engineering*, D. Winkler, S. Biffl, and J. Bergsmann, Eds. Cham: Springer International Publishing, 2014, pp. 155–169.
- [36] S. W. AbuSalim, R. Ibrahim, and J. A. Wahab, “Comparative analysis of software testing techniques for mobile applications,” *Journal of Physics: Conference Series*, vol. 1793, no. 1, p. 012036, feb 2021. [Online]. Available: <https://dx.doi.org/10.1088/1742-6596/1793/1/012036>



Yan Watequlis Syaifudin is an associate professor at State Polytechnic of Malang, Indonesia. He received the bachelor’s degree in informatics from Bandung Institute of Technology, Indonesia, in 2003 and the master’s degree in information technology from Sepuluh Nopember Institute of Technology, Indonesia, in 2011. Finally, in 2021, the Ph.D. degree in information and communication systems was received from Okayama University, Japan, respectively. He is also a reviewer in a journal and conferences held by University of Bahrain. His research interests include technology enhanced learning, intelligent systems, and data analytics.



Indrazno Siradjuddin received the B.E degree in electrical and electronics engineering from Brawijaya University, Malang, Indonesia, in 2000, and the Master’s degree in electrical and electronics engineering from the Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, in 2004, and the Ph.D. degree from the University of Ulster, Londonderry, U.K, in 2014. He is currently an Associate Professor in the Department of Electrical Engineering, State Polytechnic of Malang, Indonesia. He has more than 108 papers to his credit published in refereed journals and conference proceedings. His research interests include vision-based robot manipulator control, intelligent control, neural networks, and fuzzy systems.

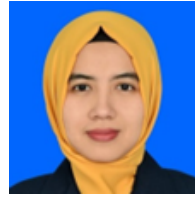


Nobuo Funabiki received the B.S. and Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Japan, in 1984 and 1993, respectively. He received the M.S. degree in electrical engineering from Case Western Reserve University, USA, in 1991. From 1984 to 1994, he was with the System Engineering Division, Sumitomo Metal Industries, Ltd., Japan. In 1994, he joined the Department of Information and Computer Sciences at Osaka University, Japan, as an assistant professor, and became an associate professor in 1995. He stayed at University of Illinois, Urbana-Champaign, in 1998, and at University of California, Santa Barbara, in 2000–2001, as a visiting researcher. In 2001, he moved to the Department of Communication Network Engineering (currently, Department of Electrical and Communication Engineering) at Okayama University as a professor. His research interests include computer networks, optimization algorithms, educational technology, and Web technology. He is a member of IEEE, IEICE, and IPSJ.



**Dewi Yanti Liliana** received the Ph.D. degree in computer science from University of Indonesia in 2019. She is currently a lecturer and researcher at Department of Computer and Informatics Engineering, Politeknik Negeri Jakarta, Indonesia. Her research interests include artificial intelligence, machine learning, social signal processing, computer vision, and affective computing. She has collaborated actively with

researchers in multidiscipline especially psychology. She has served as a conference committee and reviewer for many journals. She is an active member of IEEE Indonesia Section and active volunteer of Women in Engineering and Humanitarian Activity Committee in Region 10.



**Mustika Mentari** received the B.S. degree in Computer Science from Brawijaya University in 2011, and the M.S. degree in Informatics from Sepuluh Nopember Institute of Technology, Surabaya, Indonesia in 2014. Since 2015, she has been a lecturer at State Polytechnic of Malang. Her research interests include image processing, intelligent systems, and computer vision.



**Noprianto** received bachelor's degree from the Technical Information, AKAKOM Yogyakarta, in 2007 and M.Eng. degree in Electrical and Information Technology Engineering Department, Gadjah Mada University in 2017. His research interests are Computer Vision, Internet of Things, and Software Engineering.



**Andi Baso Kaswar** completed his undergraduate program in the Informatics and Computer Engineering Education Study Program at Makassar State University (UNM) from 2010 to 2014. Furthermore, the author completed his education in the Informatics Engineering postgraduate program at the Sepuluh Nopember Institute of Technology (ITS) from 2014 to 2016. Since 2016, the author has worked as a lecturer and is currently

a permanent lecturer at Makassar State University in the Computer Engineering study program. The area of expertise that the author has studied is Computer Vision and Digital Image Processing, and he has published several papers related to this field. Besides being active as a lecturer and researcher, the author has also written programming books and published them at national publishers. The author has also won several achievements. 2 Achievements in the last few years, namely winning second place in a national textbook writing competition and with his team representing Indonesia at the 2019 Asean ICT Awards in Laos.



**Nabilah Argyanti Ardyningrum** is student of the State Polytechnic of Malang. She is currently in her fourth year of obtaining her bachelor's degree in informatics engineering. Her research interests include intelligent systems, computer vision, and software engineering.



**Flasma Veronicha Hendryanna** received Bachelor's Degree in Informatics Engineering from State Polytechnic of Malang in 2022. Her research interests include Computer Vision and Software Engineering.