

## An Implementation and Evaluation of Basic Activity Topic for Interactive Application Stage in Android Programming Learning Assistance System

Yan Watequlis Syaifudin<sup>†</sup> Nobuo Funabiki<sup>†</sup> Minoru Kuribayashi<sup>†</sup>

**Abstract.** To meet increasing demands for efficient Android programming educations, we have proposed the learning model composed of four stages and developed *Android Programming Learning Assistance System (APLAS)* to guide students to independent learning by utilizing the TDD method. *JUnit* and *Robolectric* are adopted to validate student answers automatically. In this paper, we implement *Basic Activity* as the first topic of *Interactive Application* stage in APLAS and evaluate it through applications to students in an IT department in Indonesia.

### 1. Introduction

With the increasing number of Android applications currently available, it is certain that the need for Android application programmers will be very high. For this reason, Android programming study becomes very important for students in the information technology field. However, effective teaching of Android-based programming is an issue that has been widely discussed among scholars, teachers, and trainers to cultivate high quality Android programmers.

To assist learning Java-based Android programming, we have proposed a learning model that is composed of four stages and developed *Android Programming Learning Assistance System (APLAS)* [1] based on this model. APLAS guides students to independent learning by validating students' codes automatically. Referring to *Java Programming Learning Assistance System (JPLAS)* [2], any student answer will be verified by the *Test-Driven Development (TDD)* method [3] using *JUnit* [4] and *Robolectric* [5]. Figure 1 illustrates the software architecture of APLAS. In our previous study for APLAS, we implemented and evaluated the first stage of the model, namely *User Interface*, which focuses on creating a user interface using *XML*.

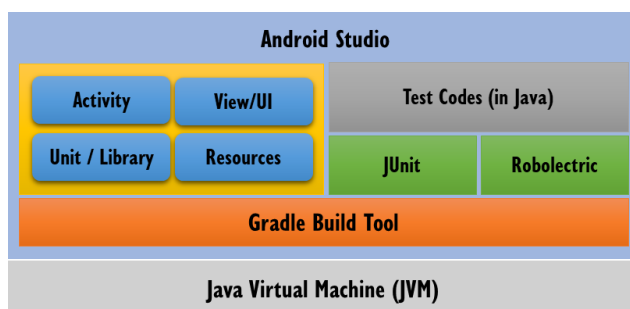


Figure 1. APLAS software architecture

In this paper, we implement and evaluate *Basic Activity* as the first topic of the second stage, *Interactive Application*, which focuses on activity programming using Java. For evaluations, we prepared an assignment of making the *Unit Converter* application after creating the user interface and asked 40 third-year students in an IT department in Indonesia. The results confirm the effectiveness of the proposal.

### 2. Learning Model for APLAS

Learning materials for Android programming actually consists of a lot of topics, covering from basic levels to advanced ones. Thus, we divide the topics into four stages in the learning model.

#### 2.1 Learning Stages

To design the following four-stage model, we refer to the learning courses in *Udacity* [6], *Google Developers* [7], and *Horton's book* [8]:

- User Interface focuses on creating an Android application interface using *XML*.
- Interactive Application focuses on implementing the activity in the project, where it is requested to create Java codes to utilize widgets, events, fragments, intents, and multimedia.
- Content Provider focuses on utilizing shared sets of application data stored in *SQLite* database, on the Web, or on any other persistent storage location.
- Service Interaction focuses on utilizing existing services provided by the Android operating system, where a service indicates an application component that can perform in the background without providing a user interface.

#### 2.2 Learning Topics in Interactive Application Stage Learning Stages

*Interactive Application* stage includes four topics, namely *Basic Activity*, *Advanced Widgets*, *Multiple Activities*, and *Multimedia Resources*. *Basic Activity* aims to study the basic activity programming using *Unit Converter* case. *Advanced Widgets* aims to study utilizations of widgets, timers, styles, and data structures. *Multiple Activities* aim to study how to create an application with multiple activities by utilizing intents and fragments. *Multimedia Resources* aim to study how to create animations with *XML* and utilize multimedia resources.

#### 2.3 Learning Process for Each Topic

Before solving a topic in APLAS, a student will obtain the package of necessary files for the topic, including the task guide that describes the set of tasks with the detailed specifications. Then, the student conducts the following process in learning this topic:

- Start Learning:** the student installs the package of files that consist of the task guide, the supplement folder, and the test folder into his/her PC and reads the guide.
- Open Project:** the student starts *Android Studio* application with open "BasicUI" project and synchronizes the Gradle API.
- Do Task:** the student completes the tasks described in the guide.
- Test Code:** the student runs the test code and receives the test result. If the code passes the test, the green icon will appear, and otherwise, the red icon will appear.

### 3. Implementation for *Basic Activity Learning*

In this section, we present the implementation of APLAS for learning *Basic Activity* topic by providing the logic aspects of the *Unit Converter* project, like shown in Figure 2.

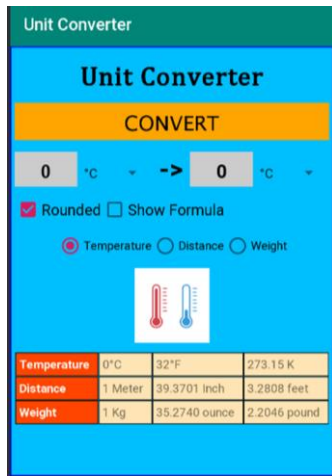


Figure 2. *Unit Converter* application display

#### 3.1 Learning Objectives

Through this topic, a student is expected to learn the following objectives in building an interactive application:

- Java Unit:** the student will learn how to make Java classes including the fields and methods.
- Activity Lifecycle:** the student will learn the activity class that provides callbacks to know the change of the state.
- Access Resources:** the student will learn additional files and static contents beside Java codes, such as bitmaps, layout definitions, user interface strings, and animation instructions.
- Event Listener:** events make an Android application provide interactions between the user and the application using interactive components such as button clicks or selection items.

#### 3.2 Tasks in Project

In the *Unit Converter* project, nine tasks in Table 1 must be completed sequentially to build an application display in Figure 1. This project includes the four learning objectives.

Table 1 Tasks in project.

no.	task	objective
1	make <i>Temperature</i> class	Java Unit
2	make <i>Distance</i> class	Java Unit
3	make <i>Weight</i> class	Java Unit
4	define fields and methods in <i>Activity</i>	Java Unit
5	make <i>Override method</i> in <i>Activity</i>	Activity Lifecycle
6	create <i>RadioGroup</i> event	Event Listener
7	make method to convert units	Access Resources
8	create <i>Widget</i> event listener	Event Listener
9	create <i>ImageView</i> and event	Event Listener

#### 3.3 Student Answer Validation

The validation of the student answer is confirmed through the following steps:

- Build the activity and assign required fields.
- Test the answer codes by using the test code on *JUnit* and *Robolectric*, to validate the method availability and correctness, the field availability and correctness, the widget property values, the event listener availability and correctness, and the resource availability and correctness.
- Show the result message.

### 4. Evaluation

For the evaluation, we implemented the system and asked 40 students of an IT department in an Indonesian university to solve the nine tasks using *Android Studio*. During the experiment, we recorded the completion time and the feedbacks for each task.

Table 2 Implementation Result.

item	result
completion	pass: 90%, fail:10%
duration time	fastest: 3 min., slowest: 272 min., average: 216.0 min.
difficulty	easy tasks: 1,2,3,4. medium tasks: 5,6,7,8 hard tasks: 9

Table 2 summarizes the results of implementation. Their opinions include that the tasks are easy, the most difficult task is to create event listener with image, the version of *Android Studio* is different, and it is not easy to make the event listener and access the image resource.

### 5. Conclusion

This paper presented the implementation of the *Basic Activity* in APLAS and evaluate it through applications to students. The results show the effectiveness for students to learn *Android* programming, particularly in *Basic Activity* topic. Future works will include implementations of other topics/stages of the learning model in APLAS and their evaluations/ applications.

#### Acknowledgement

We are very grate to Mr. Pramana Yoga Saputra in State Polytechnic of Malang, Indonesia, for helping us implementing this system. We would like also to thank to all the lecturers of Mobile Programming in State Polytechnic of Malang for giving feedbacks and reviewing our system.

#### References

- Y. W. Syaifudin, N. Funabiki and M. Kuribayashi, "Learning model for Android programming learning assistant system," in Proc. IEICE Gen. Conf., 2019.
- N. Funabiki, Y. Matsushima, T. Nakanishi, K. Watanabe and N. Amano, "A Java programming learning assistant system using test-driven development Method," IAENG Int. J. Comput. Sci., vol. 40, no. 1, pp. 38-46, 2013.
- V. Farcic and A. Garcia, Test-driven Java development, Packt Publishing., 2015.
- <https://junit.org/>.
- <http://robolectric.org/>.
- <https://www.udacity.com>.
- <https://developer.android.com/docs>
- J. Horton, Android programming for beginners, Packt Publishing, 2015.

† Okayama University