

Optimization of Model-View-ViewModel (MVVM) Architecture Pattern and RESTfull API on Android-based E-Learning Application

Alfansyah Nori Pratama
Yogyakarta University of Technology
Yogyakarta, Indonesia

Farida Ardiani
Yogyakarta University of Technology
Yogyakarta, Indonesia

ABSTRACT

E-learning or Electronic Learning is a learning system through electronic media which can be interpreted as a form of utilization of information technology in the world of education in the form of an online platform that allows it to be accessed anywhere. E-learning implemented in one of the junior high schools in the city of Yogyakarta is using an online learning platform from geschool. Geschool is an education-based social network intended for online learning. In the utilization of learning activities, the current E-learning has shortcomings, namely to access it always requires an internet connection, causing students to have difficulty getting subject matter in the event of internet problems. In addition, geschool does not have a native android application that can support application performance. One way to support the performance of an application is by applying software architecture patterns. This research is made by applying RESTful API and MVVM architectural design. Model-View-ViewModel (MVVM) is an example of a software architecture pattern that conceptualizes the separation of component interests which separates the graphical interface and data processing. The resulting application was developed using Kotlin and PHP programming languages. The application test results using the Whitebox method state that the application runs smoothly and as expected. By applying the RESTful API architecture design and MVVM architecture pattern, it produces an android-based e-learning application that is lightweight and user friendly. In the future, this research will be developed so that the resulting application has more features.

General Terms

RESTfull API, MVVM, Android, Application

Keywords

E-Learning, Kotlin, PHP, Whitebox Testing

1. INTRODUCTION

The not yet optimal quality of learning during the COVID-19 pandemic has caused a decrease in student involvement in learning [1]. This condition causes the phenomenon of learning loss that worries the world of education [2]. To overcome this, teachers should apply the digital-based learning methods needed in the new-normal era [3]. One of them is by using learning methods that can increase engagement, creativity, and mastery of technology through android-based self-learning experiences. E-learning is one of the computer technology tools used to improve teaching and learning activities. E-learning system has been implemented in Indonesia, one of which is in one of the junior high schools in Yogyakarta.

E-Learning is global, unlimited, and open to the public, and it presents a new learning environment for students to succeed in their studies [4], [5]. E-learning is implemented using the Geschool online learning platform. Geschool is an education-based social network aimed at online learning. Geschool is currently only used for daily tests and final exams. In the utilization of learning activities, the current e-learning has shortcomings, namely that it is rarely used to access material and to access the geschool site always requires an internet connection, causing students to have difficulty accessing learning media if there is a disruption in the internet. In addition, geschool does not have native android apps that can support the performance of the app. E-learning from geschool is currently still website-based. This causes the use of e-learning as a learning media for students to be less effective and flexible. Based on these constraints, until now the use of E-Learning is considered less flexible and less optimal.

Software architecture design decisions, such as the selection of architectural patterns and software design patterns, are typically made in the early phases of the software development life cycle [6]. One way to improve application performance is to apply architectural patterns in its development. One of the latest architectural patterns in android application development is MVVM. MVVM is an architectural pattern that conceptualizes the separation between graphical user interface components and data models. The MVVM architecture pattern consists of 3 components: model, view, and viewmodel.

2. RESEARCH METHODOLOGY

Research conducted in the process of designing and developing android-based e-learning applications using MVVM architecture is to use the waterfall model as part of the SDLC (Software Development Lifecycle) method. Software development projects have to deal with many risks and, the main function of a project leader is to minimize the damage that these risks can cause [7]. The cycles in the SDLC include planning, analysis, design, development cycle, testing, implementation, release, and maintenance.

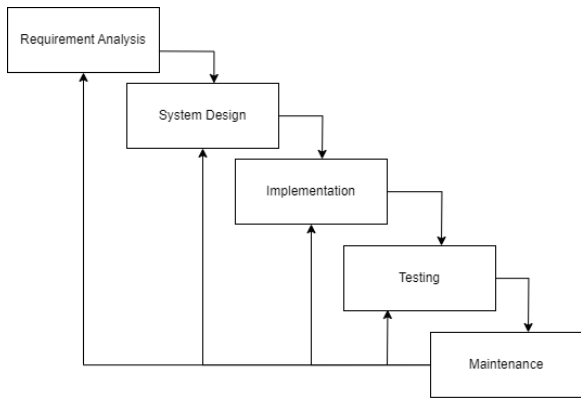


Fig. 1 Waterfall Method

The process sequence starts from requirement analysis or defining requirements to maintenance or application maintenance. Once the development stage reaches the maintenance process, the process sequence can be returned to the previous processes.

2.1 Data Collection Procedure

The data collection process went through 3 stages as follows

2.1.1 Literature Study

The researcher conducted a literature study by collecting data through the internet and journals that are in accordance with the topic. The data will be used as the basis for research.

2.1.2 Observation

The observation was conducted in one of the junior high schools in Yogyakarta. Through this observation, it can be concluded that the system currently applied to schools is still less effective and efficient.

2.1.3 Interview

The observation was conducted in one of the junior high schools in Yogyakarta. Through this observation, it can be concluded that the system currently applied to schools is still less effective and efficient.

2.2 System Design Logic

Features and data in this application at the system design stage are described through Data Flow Diagram (DFD) and Entity Relationship Diagram (ERD).

2.2.1 Data Flow Diagram

Data flow is a scheme that transfers data, strings, numbers, or arrays from one memory space, or department, to another memory space or department, respectively [8]. In this process, the data, numbers, strings, or arrays are evaluated for clarity and accuracy or modified. Diagramming and defining these data structures make it simple to facilitate their manipulation and implementation for designing and coding programs and software systems [9]. A Data flow visualization system allows users to compose a query for system specifications in visualization modules and construct the data flow diagrams [10]. The Data Flow Diagram (DFD) design shows how data flows from the context diagram, DFD level 1, to DFD level 2.

Context diagram as the highest level in DFD design that models the flow of inputs and outputs in the application as a whole and

describes all elements or scope that can provide a clear and detailed description of the system as a whole.

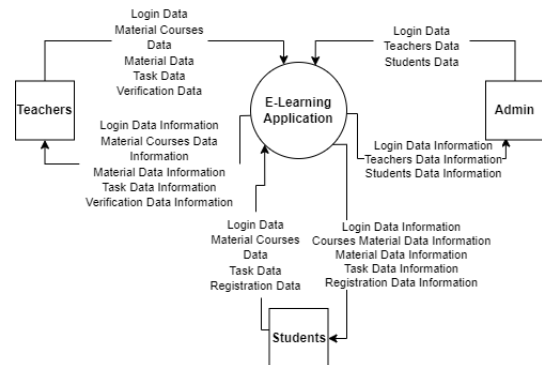


Fig. 2 Context Diagram

DFD level 1 describes the relationship of all processes that occur in the system to data storage tables, along with the reciprocity of the relationship as a whole. In DFD level 1 there are 6 processes which are derivatives of the context diagram. These processes include the login process, master data input process, subject list process, material process, subject process, and task process. The 1.0 login process streams login data to the database and then returns it to the user as a login status. Process 2.0 master data input is a process where the data streamed is teacher data and student data to the database. Process 3.0 subject list contains registration data from students which is streamed to the database and then returned to students in the form of registration data information. In addition to registration data, there is verification data that is streamed from the teacher to this process. Process 4.0 material is a process that streams material data from students and teachers. Process 5.0 subjects is a process that streams subject data to the database. This process occurs when the teacher transacts subject data to the database. Process 6.0 assignment is a process that occurs from teachers and students. The data that flows in the form of assignment data, this data flows when an assignment data transaction occurs to the database.

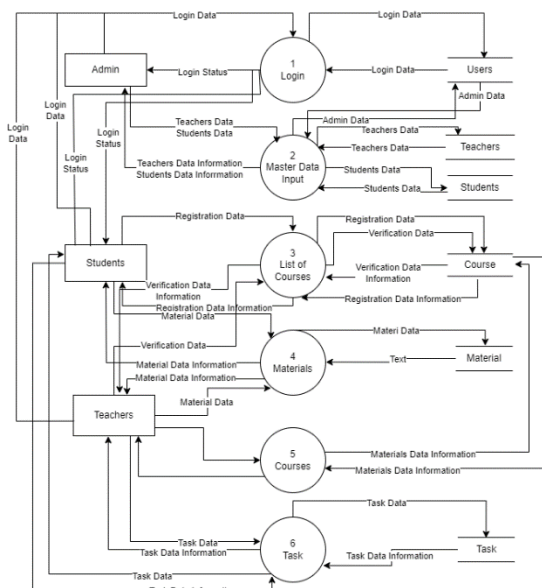


Fig. 3 DFD Level 1

DFD level 2 is a derivative of DFD level 1 where in DFD level 2 there are 8 derivative processes from 4 DFD level 1 processes. DFD level 2 process 2 where there is a teacher data input process 2.1 and student data input process 2.2. Both processes are derivatives of process 2.0. The data that flows in this process is teacher data and student data that comes from the admin.

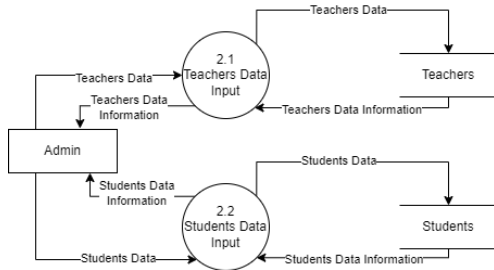


Fig. 4 DFD Level 2 Process 2

DFD level 2 process 3 is a derivative of DFD level 1 process 3. In this DFD there are 2 processes, namely process 3.1 registration and process 3.2 verification. At this DFD level, the data flowing is registration data and verification data. These data come from students and teachers to the database and back again.

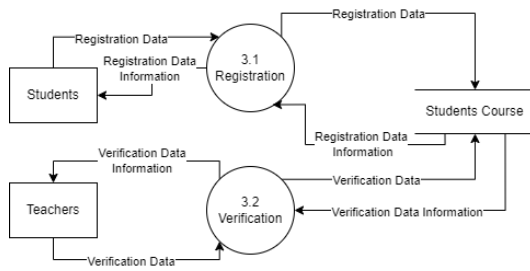


Fig. 5 DFD level 2 Process 3

Next is DFD level 2 process 4 which is a derivative of DFD level 1 process 4 material. At this DFD level there are 2 processes, namely process 4.1 download material and process 4.2 upload material.

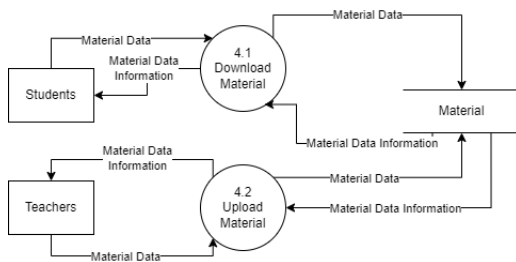


Fig. 6 DFD level 2 Process 4

DFD level 2 process 6 is a derivative of DFD level 1 process 6 tasks. There are 2 processes including: process 6.1 send task file and process 6.2 create task

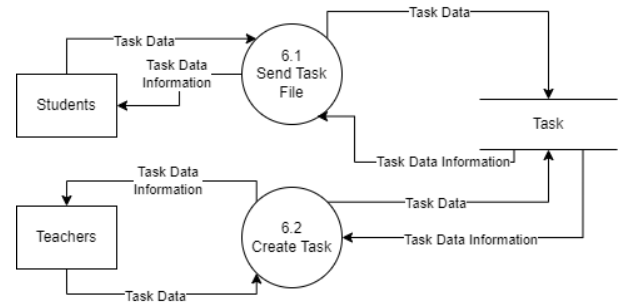


Fig. 7 DFD level 2 process 6

DFD level 2 process 6 shows how assignment data flows from students and teachers to the database.

2.2.2 Entity Relationship Diagram

Entity Relationship Diagram is a diagram that illustrates how the entities in the database relate to each other. By designing ERD, the application development process can be more organized. The ERD design in this study is adjusted to the number of tables used in the database that are interrelated.

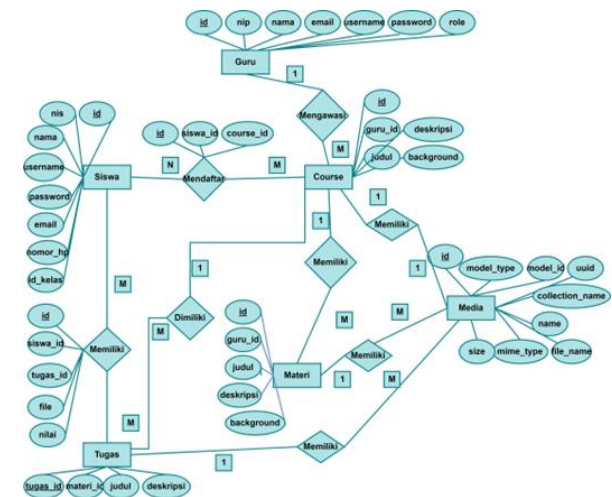


Fig. 8 Entity Relationship Diagram

There are 6 entities formed, namely students, tasks, materials, media, courses, and teachers. The relations formed include: register relation between students and courses, own relation between students and assignments, own relation between assignments and media, own relation between assignments and courses, own relation between materials and media, own relation between courses and materials, and supervisory relation between teachers and courses.

2.3 MVVM Architecture

The Model View ViewModel architecture pattern is an architecture pattern that conceptualizes the separation between the application interface and the data model or backend logic.

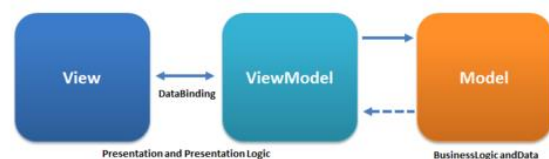


Fig. 9 MVVM Architecture Pattern

Model is a component that represents business logic and data, then view is a component that represents the user interface without containing program lines in the form of application logic. View consists of xml resource files and Activity/Fragment. ViewModel is a very important component of this architecture that helps in the concept of separation. The separation concept keeps the view separate from the model and at the same time acts as a controller that supports interaction and coordination between the view and model components.

MVVM architecture uses the room library to store data locally. Room helps store data within the application so that data can be accessed without using an internet connection. In addition, MVVM also uses RESTful API architecture to get data from the internet. Those APIs expose back-end data, so it is crucial that they are robust, secure, and reliable to keep sensitive data protected [11]. MVVM architecture can store data offline by using library rooms so that application performance is much faster when compared to applications that always request data to the server. The ability of MVVM architecture cannot be separated from its components, namely model, view, and viewmodel.

A model is a component that represents a data source in an application. In the context of android application development, models are often initiated as repository classes. Repository is a class that handles data from two sources, namely data obtained from LocalDataStore and from RemoteDataStore. LocalDataStore is the class responsible for retrieving data, storing data, and deleting data from and to the database space. RemoteDataStore is a class that handles data that comes from the remote or internet.

View is a component that represents the user interface. Application displays in the form of text, buttons, navigation, and other elements are part of the view.

ViewModel is a component in the MVVM architecture pattern that serves to prepare data or process data that will be sent to the UI / View component.

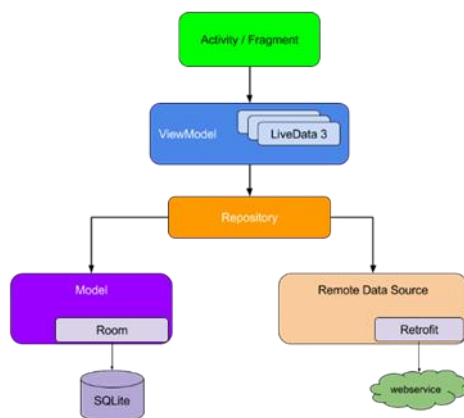


Fig. 10 MVVM

The data displayed to the user is data that has been processed in both format and source by the ViewModel. When the application first displays the data, the repository will handle this data source. After the data is obtained from the remote or internet, the data will be stored in the database space so that in the next access the data can be retrieved from the local source. Request data to the API using the retrofit library on android.

2.4 Interface Design

The interface of the e-learning application is designed based on the purpose of the application, application users, and current design trends. The results of the interface design contain the main pages of the e-learning application. The main pages of this e-learning application include the login page, home page, profile page, subject page, material page, assignment list page, and assignment collection page.

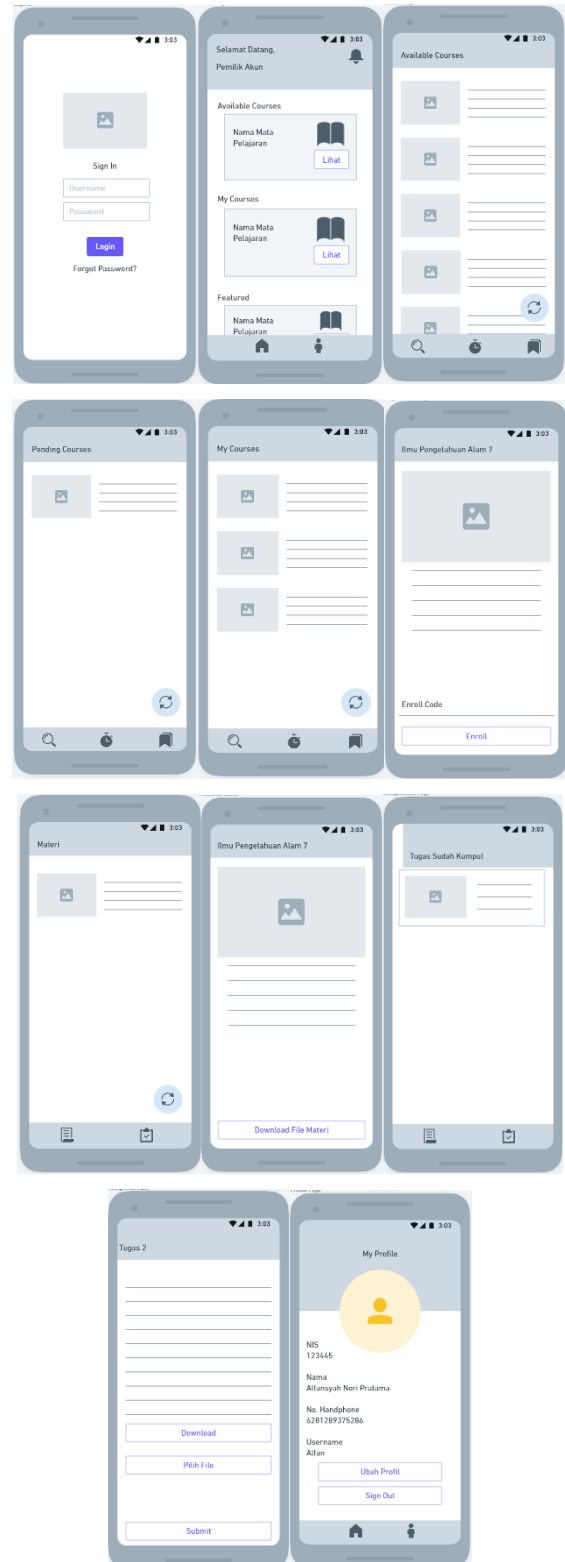


Fig. 11 Wireframe Application

3. RESULT AND DISCUSSION

3.1 Assumptions

This application is made to assist students in the learning process. students are expected to be able to search for courses, register for courses, search for materials, download materials, submit assignments, and assess assignments.

3.2 Hypothesis

Based on the assumptions obtained, the hypothesis of this research is "E-Learning application based on android with the application of MVVM architecture can overcome the shortcomings of the system that is currently running in junior high school in Yogyakarta".

3.3 Run an Experiment

The coding process of this system uses Kotlin and PHP programming languages. This system is also made by implementing MVVM architecture and RESTfull API. In the process of requesting data on the API using the retrofit library on Android.

3.4 Implementations

3.4.1 Login Page and Home Page

Application users in this case are students who first enter the application and log in first by using the account that has been registered by the administrator before. On the following login page there is a logo of SMP N 2 Ngaglik, username input text, and password input text. Students enter their username and password then press the login button at the bottom of the page. After the student has successfully logged in, the student will be directed to the home page. If the student fails to log in, the application will display an error notification and the student remains on the login page.

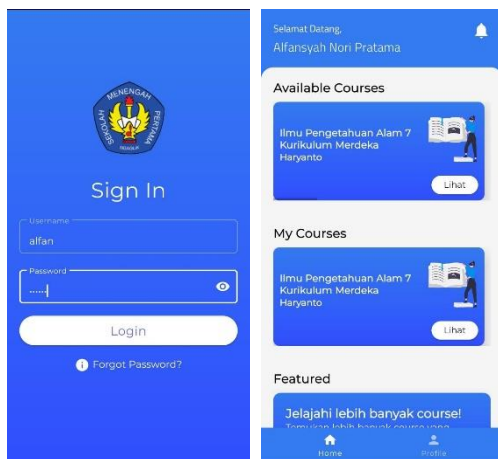


Fig. 12 Login Page and Home Page

3.4.2 Search Page and Pending Courses Page

After the student has successfully entered the home page, the student can start searching for subjects through the subjects page. The subject page consists of various supporting pages such as the subject search page, subject page, subject pending page, subject list page, and subject page. The subject search page is a page used by students to search for available subjects. The pending subjects page is a page that displays a list of subjects that have pending status, the subject registration page is a page used by students to enter the subject registration code, and the my subjects page is a page that displays a list of subjects that have been verified by the teacher.

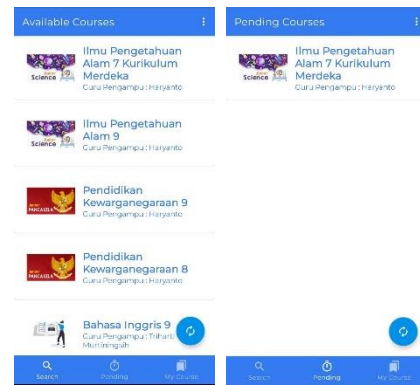


Fig. 13 Search Page and Pending Course Page

3.4.3 Registration Page and My Courses Page

After students get verification of subject enrollment, then students can access materials and assignments from the subject.

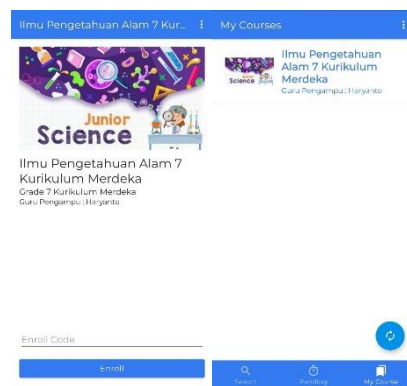


Fig. 14 Registration Page and My Courses Page

3.4.4 Material List Page and Material Details

The material page consists of a material list page and a material detail page. The material list page is a page that displays a list of materials for a subject. The material detail page is a page where students can read lesson materials and download material files.



Fig. 15 Material List Page and Material Details

3.4.5 Assignments Page and Assignments List Page

In addition to accessing course materials, students can also access assignments from the course through the assignment page. The assignment page consists of an uncollected assignment list page, a collected assignment list page, an

assignment detail page, and a collected assignment page. The uncollected assignment list page is a page that displays a list of assignments that have not been collected by students. Conversely, the collected assignment list page is a page that displays a list of assignments that have been collected by students. The assignment details page is a page that displays detailed assignment descriptions and instructions. On the assignment details page, you can download the assignment file. The assignment collection page is a page used by students to collect assignments.



Fig. 16 Assignment Page

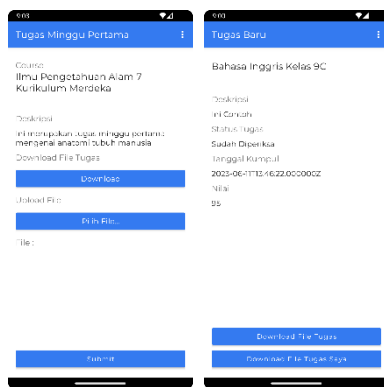


Fig. 17 Assignments List Page

3.4.6 Profile Page

Students can view their profile or biodata through the profile page. The profile page can be accessed through the home page. Through the profile page we can change profile data and exit the application.



Fig. 18 Profile Page

3.5 Discussion of Result

Software testing is applied to validate the behavior of the software system and identify flaws and bugs [12]. Tests carried out using the white box testing method in this study were carried out to test applications with parameters that cause the application to run normally and abnormally. Software testing is crucial for evaluating and assuring the quality and for reducing the risk of failure when a product is released [13]. White-box test generation is a technique used for automatically selecting test inputs using only the code under test [14]. Tests using the whitebox method on several modules that apply the MVVM architecture pattern. The tested modules include the login module, subject module, material module, and assignment module.

Table 1 Login Module Testing Results

No	Function Tested	Test Scenario	Expected Results	Results
1	Student Auth	Enter the username and password according to the data in the database.	Valid, Switch page to home view	Related
2	Student Auth	Username and password input does not match the data in the database	Invalid, and produces the text display output "Invalid"	Related

The results of testing the subject module in the application. The subject module is tested precisely on the function of loading available subject data to be displayed on the subject search page.

Table 2 Courses Module Testing Results

No	Function Tested	Test Scenario	Expected Results	Results
1	Load available subject data	Retrieve subject data with an internet connection.	Valid, get a list of available courses.	True
2	Load available subject data	Mengambil data mata pelajaran dengan kondisi tidak terkoneksi internet.	Invalid, generates 404 error animation	True

Table of test results for the material module in the application. The function tested is the function to load material data on the material list page.

Table 3 Material Module Testing Results

No	Function Tested	Test Scenario	Expected Results	Results
1	Load material data	Retrieve material data with an internet connection.	Valid, getting material data from the internet for the first time and then getting material data from local storage sources.	True
2	Load material data	Retrieve material data without an internet connection.	Valid, gets the latest material data from local storage and generates a toast message.	True
3	Load material data	Refresh material data when not connected to the internet	Invalid, generating an error message output.	True

Test results on the task module. The functions tested in the task module are the functions for submitting tasks and downloading tasks.

Table 4 Task Module Testing Results

No	Function Tested	Test Scenario	Expected Results	Results
1	Submit Task	Send task data when connected to the internet.	Valid, Switch page to success view.	True
2	Submit Task	Send task data without an internet connection.	Invalid, and produces a "Failed" toast output	True
3	Download Task	Download task data when connected to the internet.	Valid, The assignment file was downloaded successfully.	True
4	Download Task	Download assignment data without an internet connection.	Invalid, the file was not downloaded successfully and generated a "Failed" toast output	True

4. CONCLUSION

Through this research, the following conclusions can be drawn:

1. Android-based e-learning application successfully applies MVVM architecture pattern.

2. Android-based e-learning application successfully handles the process of subject search, subject registration, material search, download of assignment material files, collection, and assessment of assignments.

3. Android-based e-learning application can overcome the shortcomings of the e-learning application currently used in one of the junior high schools in the city of Yogyakarta which requires an internet connection to access it. By applying the MVVM architecture pattern, students can access subjects and materials offline without an internet connection.

5. REFERENCES

- [1] E. f. C.-1. a. E. R. T. f. E. E. i. E. Programmes, "Lambert, Chris G.; Rennie, Allan E.W.," Education Sciences, pp. 2-16, 2021.
- [2] E. Dorn, B. Hancock, J. Sarakatsannis and E. Viruleg, "COVID-19 and Learning Loss—Disparities Grow and Students Need Help," McKinsey & Company, vol. 8, pp. 224-228, 2020.
- [3] Reflianto, S. P., K. D. and W. U., "Reading comprehension skills: The effect of online flipped classroom learning and student engagement during the COVID-19 pandemic," European Journal of Educational Research, vol. 10, no. 4, pp. 1613-1624, 2021.
- [4] A. P. Chitra and M. A. Raj, "eLearning," Journal of Applied and Advanced Research, vol. 3, no. 1, pp. 11-13, 2018.
- [5] S. Duggal, "Factors Impacting Acceptance of E-Learning in India: Learners' Perspective," Asian Association of Open Universities Journal, 2022.
- [6] S. Farshidi, S. Jansen and J. Martijn van der Werf, "Capturing software architecture knowledge for pattern-driven design," Journal of Systems and Software, vol. 169, 2020.
- [7] F. Pinciroli, J. L. B. Justo and R. Forradellas, "Systematic mapping study: On the coverage of aspect-oriented methodologies for the early phases of the software development life cycle," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 6, pp. 2883-2896, 2022.
- [8] S. M. Cheema, S. Tariq and I. M. Pires, "A natural language interface for automatic generation of data flow diagram using web extraction techniques," Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 2, pp. 626-640, 2023.
- [9] H. Alshareef, S. Stucki and G. Schneider, "Transforming Data Flow Diagrams for Privacy Compliance," Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development MODELSWARD, vol. 1, pp. 207-215, 2021.
- [10] M. H. S. Al Ashry, "Importance of Data Flow Diagrams and Entity Relationships Diagrams to Data Structures Systems Design in C++ "A Practical Example"," Journal of Management and Strategy, vol. 8, no. 4, 2017.
- [11] D. Felicio, J. Simao and N. Datia, "RapiTest: Continuous Black-Box Testing of RESTful Web APIs," Procedia Computer Science, vol. 219, pp. 537-545, 2023.

- [12] D. Honfi and Z. Micskei, "Automated isolation for white-box test generation," *Information and Software Technology*, vol. 125, 2020.
- [13] A. Abo-eleneen, A. Palliyali and C. Catal, "The role of Reinforcement Learning in software testing," *Information and Software Technology*, vol. 164, 2023.
- [14] R. Blanco, M. Trinidad, M. J. Suarez-Cabal and e. al, "Can gamification help in software testing education? Findings from an empirical study," *Journal of Systems and Software*, vol. 200, 2023.