MIKAILALSYS
Journal of Multidisciplinary Sciences

e-ISSN : 2987-2286
p-ISSN : 2987-3924
Index: Harvard, Boston, Sydney, Copernicus, Sinta 4, Scilit, Semantic, Google, etc

https://doi.org/10.58578/mikailalsys.v4i3.12688

Development of an Android-Based Practicum Module for the Object-Oriented Programming Subject for Grade XI Software and Game Development Students at SMK Negeri 2 Padang

Mutiara Ningsih, Sofia Edriati, Ami Anggraini Samudra
Universitas PGRI Sumatera Barat, Indonesia
mutiaran348@gmail.com; sofiaedriati81@gmail.com

Submitted:	Revised:	Accepted:	Published:
Oct 11, 2026	Nov 9, 2026	Nov 20, 2026	Nov 25, 2026

Abstract

Although the growing use of mobile technology in vocational education offers opportunities for flexible learning, practical programming resources that support independent learning remain limited. This study developed an Android-based practicum module for object-oriented programming and evaluated its validity and practicality among eleventh-grade students enrolled in the Software and Game Development program at SMK Negeri 2 Padang. The study employed a research and development approach using an iterative System Development Life Cycle (SDLC). The evaluation involved three media experts, two subject-matter experts, one teacher, and 25 students. Data were collected through observations, interviews, functional testing, and questionnaires and analyzed using descriptive quantitative techniques. Alpha testing yielded a functionality score of 100%, while expert validation scores for media and content reached 85.12% and 86.98%, respectively, both classified as highly valid. Practicality scores from the teacher and students were 82.92% and 81.60%, respectively, both classified as highly practical. By integrating instructional materials, guided programming activities, and assessments, the module offers a valid, practical, and accessible supplementary resource for independent learning in Java programming. Further empirical research is needed to establish its effectiveness in improving students' programming skills.

Keywords: Android-Based Learning; Object-Oriented Programming; Practicum Module; System Development Life Cycle; Vocational Education

Mikailalsys is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

INTRODUCTION

The growing integration of information and communication technology (ICT) into education has changed how learning resources are developed, delivered, and accessed. Digital technology offers opportunities to extend learning beyond conventional classrooms, particularly in vocational education, where students are expected to acquire both conceptual knowledge and practical skills relevant to their future professions. The United Nations Educational, Scientific and Cultural Organization UNESCO (2023) emphasizes that educational technology can improve access to learning and support educational quality when its implementation responds to learners' needs and is supported by appropriate teaching practices. This perspective highlights the importance of developing digital learning resources that do more than provide convenient access to instructional content. Vocational students need opportunities to observe procedures, apply knowledge, identify errors, and reflect on their work. Mobile learning, particularly through Android-based applications, offers a promising approach to supporting these activities, provided that the learning environment is designed around the competencies students are expected to develop.

Indonesia's vocational education system has increasingly emphasized flexible, meaningful, and competency-oriented learning through the implementation of the Merdeka Curriculum. The curriculum encourages teachers to select and develop instructional resources that accommodate students' learning needs while providing opportunities for independent exploration and practical application. Maulida (2022) explains that teaching modules developed within the Merdeka Curriculum should integrate learning materials, instructional activities, and assessment procedures into a coherent learning experience. More specifically, the national learning outcomes for vocational software engineering include the application of structured and object-oriented programming, software modelling, graphical user interfaces, and mobile application development (Badan Standar, Kurikulum, dan Asesmen Pendidikan (Badan Standar dan Asesmen Pendidikan, 2025). These expectations require vocational schools to provide learning experiences that connect programming concepts with authentic development tasks. The availability of appropriate instructional resources is therefore important for helping students move beyond memorizing programming syntax toward understanding how software applications are designed, constructed, tested, and improved.

Object-Oriented Programming (OOP) is an important component of software development education because it introduces students to a programming approach in which data and related operations are organized into interacting objects. Learning OOP requires an understanding of concepts such as classes, objects, encapsulation, inheritance, and polymorphism, alongside the ability to translate these concepts into functioning programs. Hasanah & Untari (2021) emphasize the importance of systematic software development knowledge in preparing learners to understand programming structures and software engineering processes. These competencies cannot be adequately developed through theoretical explanations alone. Students must engage in repeated coding activities, examine program execution, interpret error messages, and revise their solutions. Pratama & Sujatmiko (2024) demonstrated the educational value of practice-oriented programming media in a vocational school setting, reporting improved web development competencies among students who used the developed learning environment. Their findings suggest that programming instruction benefits from resources that encourage students to participate directly in the development process rather than merely reproduce completed examples.

Preliminary observations at SMK Negeri 2 Padang revealed several challenges in the implementation of OOP practicum activities among Grade XI students in the Software and Game Development (PPLG) program. Students generally demonstrated satisfactory performance when assessed through their submitted programming projects. Their apparent achievement, however, did not always reflect an equally strong understanding of the processes involved in producing those projects. Classroom observations and teacher guidance indicated that some students struggled to explain the logic of their programs, identify the causes of coding errors, or modify existing code independently. A number of students also preferred obtaining solutions from generative artificial intelligence tools, including ChatGPT, rather than consistently following the instructional materials and video tutorials provided by their teachers. Although these tools helped students complete programming tasks, the resulting code was sometimes inconsistent with the structures and procedures introduced during lessons. Difficulties became particularly apparent when students were required to troubleshoot programs without external assistance. These observations indicate a discrepancy between the quality of submitted projects and students' understanding of the programming processes underlying their work.

The widespread use of internet-connected devices among Indonesian adolescents creates opportunities to make programming instruction more accessible outside scheduled

laboratory sessions. Data from the Indonesian Internet Service Providers Association indicated that internet penetration among individuals aged 13–18 reached 98.2% in 2023 (Asosiasi Penyelenggara Jasa Internet Indonesia, 2023). Although internet access does not necessarily indicate that every student owns an Android smartphone, this level of connectivity provides a relevant context for considering mobile learning as a supplementary instructional approach. Faqih (2020) identified the educational potential of Android-based mobile learning in supporting more flexible learning experiences. Similarly, Ghafara et al. (2022) developed Android-based instructional media for vocational students that incorporated learning materials, video tutorials, and interactive activities accessible through smartphones. Such resources can help reduce students' dependence on classroom-based instruction by allowing them to revisit explanations, examine worked examples, and review learning procedures according to their individual needs. For programming subjects, this flexibility is particularly valuable because students often require additional time to understand unfamiliar concepts and resolve errors encountered during practical assignments.

Previous studies have provided empirical support for the development of Android-based learning resources in Indonesian vocational education. Maharani et al. (2023) developed Android-based instructional media for an informatics subject at SMK Nusatama Padang and reported that the resulting product met established validity and practicality criteria. F. N. Hasanah & Untari (2021); S. U. Hasanah et al. (2024) similarly developed Android-based learning media for visual communication design students at SMK Negeri 1 Pasaman, demonstrating the feasibility of mobile applications as supplementary learning resources in vocational classrooms. Amalia et al. (2024) examined Android-based mobile learning developed using Kodular, highlighting its potential to support students' interest in learning. Research by Imamah & Hasanah (2023) extended this development approach to an Android-based electronic module for network operating systems, while Rahni et al. (2025) reported favourable validity and practicality assessments for a project-based electronic module in vocational informatics education. Collectively, these studies indicate that digital modules and mobile applications can provide suitable instructional resources across different vocational subjects. Their primary emphasis, however, has generally been on media development, content delivery, product feasibility, or user responses rather than the specific learning processes required for mastering object-oriented programming.

Practical learning environments have also received attention in studies examining digital modules and programming education. Putri & Jayanta (2023) developed a digital

practicum module for integrated science learning, demonstrating how structured digital resources can support the organization of practical learning activities. Pratama & Sujatmiko (2024) focused more directly on programming instruction by developing web-based practice media for vocational students. These studies provide a useful basis for considering how digital resources can connect instructional explanations with practical tasks. The emergence of generative artificial intelligence, however, introduces another dimension to programming education. Sun et al. (2024) found that ChatGPT-assisted programming influenced students' programming behaviours, including debugging activities and code reuse, although differences in programming performance between the experimental groups were not statistically significant. Humble et al. (2024) also discussed both the educational opportunities and assessment challenges associated with ChatGPT in programming education. These findings suggest that AI-supported learning should not be treated simply as beneficial or harmful. Its educational value depends partly on whether students are encouraged to examine generated solutions, understand their underlying logic, and remain actively involved in problem-solving activities.

Existing research provides a basis for Android-based instructional development, yet a more specific need remains for practicum modules that address the learning characteristics of OOP among Grade XI PPLG students. The studies reviewed have investigated mobile learning across vocational subjects, electronic modules for informatics, digital practicum materials, and technology-supported programming activities. Less attention has been directed toward integrating these elements into an Android-based OOP practicum module that emphasizes sequential programming procedures, conceptual explanations, guided exercises, and systematic error analysis within a single learning resource. The present study addresses this need by developing a module using Flutter, a framework that can support the creation of accessible mobile application interfaces (Husain et al., 2025). Its instructional design is informed by the principles of electronic module development, particularly the systematic organization of learning content, activities, and learner guidance discussed by (Najuah et al., 2020). Consideration of software quality is equally important because the educational usefulness of a digital module depends on its functionality, usability, and reliability, as reflected in the software quality evaluation discussed by (Ratnaduhita et al., 2023). The proposed contribution therefore lies in combining the accessibility of Android-based learning with structured programming practice and explicit attention to students' understanding of the code-development process.

The present study aims to develop an Android-based practicum module for the Object-Oriented Programming subject among Grade XI Software and Game Development students at SMK Negeri 2 Padang and to examine its suitability as a supplementary learning resource. The module is intended to provide organized explanations of OOP concepts, practical programming guidance, illustrative examples, and learning activities that encourage students to examine program structures and identify coding errors independently. The study also seeks to establish a basis for evaluating the module's validity and practicality in relation to students' learning needs and classroom conditions. By focusing on the programming process rather than solely on completed project outcomes, this research is expected to contribute to the development of vocational learning resources that encourage conceptual understanding, reflective problem-solving, and greater learner independence in an increasingly technology-supported educational environment.

METHODS

This study employed a Research and Development (R&D) approach with descriptive quantitative evaluation to develop and assess an Android-based practicum module for the Object-Oriented Programming (OOP) subject among Grade XI Software and Game Development (PPLG) students at SMK Negeri 2 Padang. The R&D approach was selected because the study focused on producing an instructional application and evaluating its validity and practicality, consistent with the characteristics of educational development research described by Okpatrioka (2023). The development process followed an iterative System Development Life Cycle (SDLC), which enables progressive refinement through repeated cycles of requirements analysis, design and development, testing, and implementation (Samudra & Maharani, 2023). Initial data were gathered through classroom and laboratory observations to identify instructional difficulties, existing learning resources, and the functional and non-functional requirements of the application. The system was designed using Unified Modeling Language (UML) diagrams and developed with Flutter and Dart, while Firebase was used for database management. Development was organized into two iterations: the first focused on essential learning functions, including instructional materials, practicum guidance, videos, programming activities, and evaluations; the second addressed improvements to learning-content and user-management functions. Each iteration involved testing, identifying technical problems, and making necessary revisions before the application proceeded to further evaluation.

The research involved Grade XI PPLG students and selected evaluators with relevant expertise in instructional media, programming content, and vocational education. The research documentation listed 31 respondent positions, comprising three university lecturers serving as media validators, two subject-matter validators consisting of one lecturer and one vocational teacher, one teacher evaluating classroom practicality, and 25 students assessing the application from the learners' perspective. Participants were involved according to their expertise or their role as intended users of the developed module. Data were collected through preliminary observations, expert validation questionnaires, and user-response questionnaires administered after participants had examined or used the application. The evaluation included alpha testing through white-box and black-box procedures to examine internal program logic, execution paths, and functional performance. Media validity was assessed using a 27-item questionnaire adapted from the ISO/IEC 25010 software quality framework, covering functional suitability, performance efficiency, usability, reliability, compatibility, security, maintainability, and portability (Ratnaduhita et al., 2023) . Material validity was examined using 15 items addressing content suitability, presentation, instructional material, and language. Beta testing involved teacher and student practicality assessments using the End-User Computing Satisfaction (EUCS) framework, with 16 and 15 questionnaire items, respectively, covering content, accuracy, format, ease of use, and timeliness (Putra et al., 2023).

The collected data were analyzed using descriptive quantitative techniques to determine the functional validity, content validity, and practicality of the developed application. Alpha-testing results were evaluated according to whether the tested functions operated as intended, with complete functional compliance required to meet the stated validity criterion. Expert validation and practicality responses were measured using a four-point Likert scale, ranging from 1 (disagree) to 4 (strongly agree), following the scoring approach described by (Sugiyono, 2019). The scores obtained from each instrument were converted into percentages by dividing the total observed score by the maximum possible score and multiplying the result by 100. Media and material validity percentages were classified as very valid (81–100%), valid (61–80%), moderately valid (41–60%), less valid (21–40%), or invalid (0–20%). The same percentage intervals were applied to practicality assessment, using the categories very practical, practical, moderately practical, less practical, and impractical, respectively (Widodo et al., 2023). These analyses were used to establish the

extent to which the application met the specified technical and instructional requirements and could be used appropriately as a supplementary resource for OOP practicum activities.

RESULTS

The study resulted in an Android-based practicum module designed to support Object-Oriented Programming (OOP) instruction for Grade XI Software and Game Development (PPLG) students at SMK Negeri 2 Padang. The developed application focused on Java programming operators and integrated instructional materials, practical exercises, learning videos, a simple Java programming environment, and learning assessments. The development process followed an iterative System Development Life Cycle (SDLC), encompassing requirements analysis, system design and development, testing, and implementation. The research findings are presented according to the outcomes of each development stage, followed by software testing, expert validation, and practicality assessments involving teachers and students.

1. Results of Needs Analysis and Iterative Development

The initial needs analysis was conducted through classroom observations, teacher interviews, and documentation of existing instructional practices. Observations revealed that programming instruction continued to rely on conventional learning resources, particularly video tutorials that offered limited opportunities for direct interaction. Students experienced difficulties when attempting to connect the explanations provided in instructional videos with practical programming activities. Interviews with the subject teacher further indicated that the available resources did not adequately accommodate students' needs for flexible learning and independent programming practice in the classroom. These findings provided the basis for developing an Android application that could bring instructional explanations, practical guidance, and programming activities into a single learning environment.

The development process involved two iterations, each addressing specific limitations identified during the design and evaluation of the application. The first iteration focused on improving the organization and accessibility of the user interface. Initial testing revealed inconsistencies in the arrangement of navigation elements and the presentation of the practicum module. Revisions were consequently made to improve the alignment, proportions, and visual consistency of the main menu and to provide clearer access to

practicum materials. Figure 1 presents the interface before and after these revisions, illustrating the changes made to the home screen and practicum module display.



Figure 1. Interface Improvements During the First Development Iteration

Based on figure 1 panels (a) and (b) show the main menu before and after revision, while panels (c) and (d) present changes to the practicum module interface. The second iteration focused on refining learning content and practical activities. Evaluation of the instructional materials identified an incomplete explanation of ternary operators, particularly the conditions under which ternary expressions should or should not be used. Additional explanations were incorporated to clarify the appropriate application of these operators.

Revisions were also made to the presentation of the Java practicum and evaluation interfaces. The revised practicum screen displayed programming examples and execution results more clearly, while the evaluation interface was reorganized to distinguish learning exercises from quizzes. Figure 2 illustrates the changes introduced during this development stage.

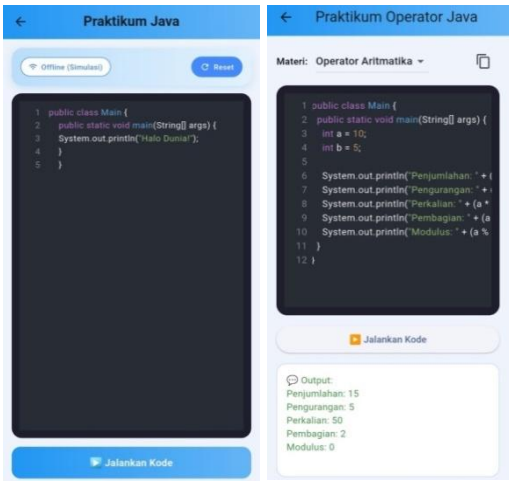

Figure 2. Revisions to the Java Practicum and Evaluation Interfaces During the Second Iteration

Based on figure 2 panels (a) and (b) illustrate the Java practicum interface before and after revision. Panels (c) and (d) show the corresponding changes to the evaluation interface. The two development iterations resulted in a more organized application structure and revised instructional content. Each revision was followed by testing to examine whether the modified functions operated according to the intended system requirements.

2. Results of Android-Based Practicum Module Development

The completed application was developed using the Flutter framework and Dart programming language, with Firebase supporting database management. The application provided several interconnected learning functions, including user registration, authentication, instructional guidance, learning outcomes, instructional materials, practicum modules, learning videos, programming exercises, and assessments. The main navigation interface is presented in Figure 3.

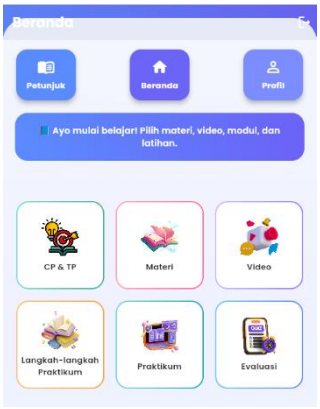

Figure 3. Main Navigation Interface of the Android-Based Practicum Module

Based on figure 3 the home screen provides access to application instructions, learning outcomes and objectives, instructional materials, practicum modules, learning videos, Java programming activities, and evaluations. The application begins with a landing page that introduces the learning environment and directs users to the login or registration page. Students register using their email addresses and passwords before accessing the main learning interface. After authentication, they can select the instructional components according to their learning needs. The application instructions explain how to navigate the available features, while the learning outcomes and objectives section presents the competencies intended to be addressed over four instructional meetings.

Instructional materials are accessible in PDF format, whereas the practicum module provides an electronic guide containing sequential instructions for completing programming activities. The learning video section offers additional explanations of the material, allowing students to review programming procedures when necessary. The Java practicum feature provides a simple programming environment in which students can work with Java code and view execution results. Figure 4 presents the programming interface developed for this purpose.

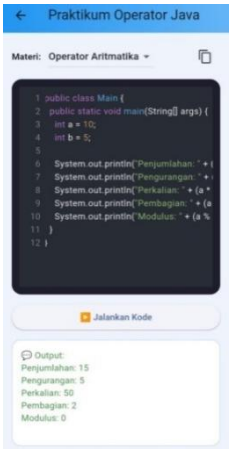

Figure 4. Java Programming Practicum Interface

Based on figure 4 the practicum interface displays Java programming code and provides a function for executing simple programs and viewing their output. The evaluation component consists of exercises and quizzes. The exercises allow students to complete assignments and submit practicum reports, while the quizzes provide questions presented through an interactive, game-like interface. Figure 5 shows the evaluation screens included in the developed application.



Figure 5. Evaluation Interface Featuring Learning Exercises and Interactive Quizzes

Based on figure 5 the evaluation menu provides two forms of assessment: exercises for completing learning tasks and quizzes organized around Java programming operator topics. These features constituted the final instructional components of the application before it underwent expert validation and classroom practicality testing.

3. Results of Alpha Testing

Alpha testing was conducted to examine the technical functionality of the developed application before its use in classroom learning activities. The evaluation included white-box testing to examine internal program execution paths and black-box testing to verify system functions from the user's perspective.

a. White-Box Testing Results

White-box testing was applied to six application components: login, home page, learning materials, practicum module, programming activities, and evaluation. The testing procedure employed basis-path analysis to identify independent execution paths within each component. Cyclomatic complexity was calculated using the following equation:

$$\mathbf{CC = E - N + 2} \quad (1)$$

where CC represents cyclomatic complexity, E denotes the number of edges, and N represents the number of nodes in the program flow graph. For example, the login function was reported to contain four edges and four nodes, resulting in a cyclomatic complexity of 2. This value identified two independent execution paths to be examined during testing. The white-box testing results are summarized in Table 1.

Table 1. Summary of White-Box Testing Results

Application component	Cyclomatic complexity	Independent paths	Testing result
Login page	2	2	Successful
Home page	2	2	Successful
Learning materials	2	2	Successful
Practicum module	2	2	Successful
Java practicum	2	2	Successful
Evaluation page	2	2	Successful
Total	—	12	All reported paths successful

Table 1 shows that all six application components had a reported cyclomatic complexity value of 2, resulting in 12 independent execution paths. The documented test cases were reported as successful across the six components. The login tests examined authentication-related execution paths, while the home page, materials, and module tests examined navigation and content accessibility. The practicum tests covered programming activity execution and the handling of execution outcomes, whereas the evaluation tests

examined task completion and submission procedures. These findings document the successful execution of the test cases included in the white-box assessment.

b. Black-Box Testing Results

Black-box testing was conducted to verify whether the visible functions of the application operated according to the expected requirements. Twelve functional components were examined, covering access to the application, navigation, learning resources, programming activities, assessment features, and logout procedures. The results of the black-box testing are presented in Table 2.

Table 2. Results of Black-Box Functional Testing

No.	Tested component	Expected functional outcome	Result
1	Splash screen	Displays the application logo and opening screen	Valid
2	Login page	Authenticates users with valid credentials	Valid
3	Main menu	Provides access to the available learning features	Valid
4	Instructions page	Displays application usage instructions	Valid
5	Profile page	Displays the application developer's profile	Valid
6	Learning outcomes and objectives	Displays the intended learning outcomes and objectives	Valid
7	Learning materials	Opens and displays the available instructional materials	Valid
8	Practicum module	Provides access to the electronic practicum module	Valid
9	Learning videos	Displays and plays the available instructional videos	Valid
10	Java practicum	Supports simple Java programming activities and displays execution results	Valid
11	Evaluation page	Provides access to exercises and quizzes	Valid
12	Logout	Ends the user session and returns to the login page	Valid
	Total valid functions	12 of 12	100%

Table 2 indicates that all 12 functional components met the expected outcomes recorded during black-box testing. The testing covered the complete user navigation sequence, from application access and authentication to learning activities, evaluation, and logout. The overall black-box validity percentage was reported as 100%, indicating that all functions included in the testing procedure satisfied the specified functional criteria. The combined alpha-testing assessment also produced a reported validity score of 100%. The application subsequently proceeded to expert validation and user practicality assessment.

4. Results of Media Expert Validation

Media validation was conducted by three lecturers from Universitas PGRI Sumatera Barat who evaluated the technical characteristics and software quality of the developed application. The assessment used eight quality dimensions derived from the ISO/IEC 25010 framework: functional suitability, performance efficiency, usability, reliability, compatibility, security, maintainability, and portability. The results of the three validators' assessments are presented in Table 3.

Table 3. Results of Media Expert Validation

Assessment aspect	V1 (%)	V2 (%)	V3 (%)	Mean (%)	Category
Functional suitability	90.00	80.00	100.00	90.00	Very valid
Performance efficiency	81.25	75.00	87.50	81.25	Very valid
Usability	87.50	75.00	87.50	83.33	Very valid
Reliability	91.67	75.00	91.67	86.11	Very valid
Compatibility	75.00	75.00	87.50	79.20	Valid
Security	83.33	83.33	83.33	83.33	Very valid
Maintainability	100.00	75.00	91.67	88.89	Very valid
Portability	91.66	75.00	100.00	88.89	Very valid
Overall mean	—	—	—	85.12	Very valid

Table 3 shows that the developed application received an overall media validity score of 85.12%, placing it in the *very valid* category according to the assessment criteria. Functional suitability obtained the highest mean score at 90.00%, followed by maintainability and portability, both at 88.89%. Reliability received a mean score of 86.11%, while usability and security obtained 83.33% each.

Performance efficiency received a score of 81.25%, which was also classified as very valid. Compatibility obtained the lowest assessment score, at 79.20%, and was classified as valid. The compatibility result differed from the other seven dimensions, which were all classified as very valid. The assessment therefore identified compatibility as the lowest-rated technical aspect of the application, although it remained within the established validity criteria.

5. Results of Material Expert Validation

Material validation was undertaken by two evaluators: one lecturer from Universitas PGRI Sumatera Barat and one vocational subject teacher from SMK Negeri 2 Padang. The assessment examined four aspects of instructional quality: content suitability, presentation suitability, instructional material, and language. The results are presented in Table 4.

Table 4. Results of Material Expert Validation

Assessment aspect	V1 (%)	V2 (%)	Mean (%)	Category
Content suitability	87.50	100.00	93.75	Very valid
Presentation suitability	75.00	83.33	79.17	Valid
Instructional material	81.25	93.75	87.50	Very valid
Language suitability	87.50	87.50	87.50	Very valid
Overall mean	—	—	86.98	Very valid

As shown in Table 4, the overall material validity score was 86.98%, corresponding to the very valid category. Content suitability received the highest mean score of 93.75%, with the two validators assigning scores of 87.50% and 100.00%, respectively. Instructional material and language suitability each obtained a mean score of 87.50%. Presentation suitability received the lowest mean score of 79.17%, placing it in the valid category. The individual scores for this dimension were 75.00% and 83.33%. These results indicate that three of the four material assessment aspects met the very valid criterion, while presentation suitability met the valid criterion. The overall expert assessment supported the suitability of the instructional content for the intended Java programming practicum activities.

6. Results of Beta Testing and Practicality Assessment

Beta testing was conducted to examine the practicality of the developed module in the intended learning environment. The assessment involved one vocational teacher and 25 students from Grade XI PPLG 1 at SMK Negeri 2 Padang. Teacher and student responses were collected through questionnaires adapted from the End-User Computing Satisfaction (EUCS) framework.

The questionnaires assessed five dimensions: content, accuracy, format, ease of use, and timeliness. The teacher completed a 16-item questionnaire, while the students completed a 15-item questionnaire. The resulting scores are presented in Table 5.

Table 5. Teacher and Student Practicality Assessment Results

Assessment aspect	Teacher (%)	Teacher category	Students (%)	Student category
Content	83.33	Very practical	80.00	Practical
Accuracy	81.25	Very practical	85.68	Very practical
Format	83.33	Very practical	80.33	Practical
Ease of use	91.67	Very practical	78.34	Practical
Timeliness	75.00	Practical	83.67	Very practical
Overall mean	82.92	Very practical	81.60	Very practical

The teacher practicality assessment produced an overall score of 82.92%, which was classified as very practical. Ease of use received the highest score, at 91.67%, followed by

content and format, both at 83.33%. Accuracy obtained a score of 81.25%, while timeliness received the lowest score at 75.00%. Four of the five assessment dimensions were classified as very practical, whereas timeliness was classified as practical.

The student practicality assessment, completed by 25 learners, yielded an overall score of 81.60%, placing the application in the very practical category. Accuracy received the highest score of 85.68%, followed by timeliness at 83.67%. Content obtained 80.00%, while format received 80.33%. Ease of use received the lowest student assessment score of 78.34%.

The distribution of student responses differed from that of the teacher assessment. Students assigned their highest score to accuracy, whereas the teacher assigned the highest score to ease of use. The student assessment classified accuracy and timeliness as very practical, while content, format, and ease of use were classified as practical. Both respondent groups nevertheless produced overall mean scores within the very practical category.

7. Results of Classroom Implementation

The developed Android-based practicum module was implemented in Grade XI PPLG learning activities at SMK Negeri 2 Padang after the application had undergone technical testing and expert assessment. The classroom implementation involved the use of the module as a supplementary resource for teaching Java programming operators.

During the implementation, the teacher used the application to present instructional materials and support practical activities across several learning meetings. Students accessed the instructional materials, electronic practicum modules, and learning videos through the application. The Java practicum feature also provided a space for completing simple programming activities, while the evaluation component allowed students to complete exercises and quizzes associated with the learning content.

The implementation records documented the use of the application in classroom instruction and students' access to its learning and assessment features. Quantitative evidence from the subsequent practicality assessments showed an overall score of 82.92% from the teacher and 81.60% from students. Both results were within the very practical category. Together with the reported media validity score of 85.12% and material validity score of 86.98%, these findings established the product's validity and practicality according to the assessment criteria applied in the study. The results describe the quality and usability of the developed module; they do not constitute a direct measurement of improvements in students' programming achievement or debugging proficiency.

DISCUSSION

The findings demonstrate that the Android-based practicum module developed for Object-Oriented Programming (OOP) at SMK Negeri 2 Padang met the established criteria for technical validity, instructional suitability, and practical use. The development process, which employed an iterative System Development Life Cycle (SDLC), allowed the application to be refined through repeated design, testing, and revision activities. Improvements made during the first iteration addressed interface organization and navigation, while the second iteration focused on instructional content and programming activities, particularly the explanation and application of ternary operators in Java. These revisions illustrate the value of an iterative approach in responding to problems identified during product development. The findings are consistent with Samudra & Maharani (2023), who discussed the application of systematic software development processes in educational technology. The present study extends this approach to a vocational practicum module that brings instructional guidance, learning materials, programming exercises, and assessments together within an Android application.

The alpha-testing results indicated that the developed application fulfilled the functional requirements examined during testing. White-box testing successfully covered 12 documented independent execution paths across six application components, while black-box testing confirmed that all 12 tested functions operated as expected, resulting in a reported functional validity score of 100%. These findings suggest that the application was technically suitable for proceeding to classroom trials. The successful execution of the tested functions, however, should not be interpreted as evidence that the application is entirely free from software defects. The results reflect compliance with the specific test scenarios included in the study. Media expert validation provided further evidence of software quality, producing an overall score of 85.12%, categorized as very valid. Functional suitability received the highest score (90.00%), whereas compatibility obtained the lowest (79.20%). The relatively lower compatibility rating indicates an aspect requiring further attention, particularly when the application is used across Android devices with different technical specifications. This assessment aligns with Ratnaduhita et al. (2023), who emphasized the relevance of ISO/IEC 25010 quality characteristics in evaluating educational software.

The material validation results provided additional support for the instructional suitability of the developed module, with an overall validity score of 86.98%. Content

suitability received the highest rating (93.75%), indicating favourable expert assessments of the relevance and appropriateness of the instructional materials. Presentation suitability, however, obtained a lower score (79.17%), suggesting that the organization and delivery of learning content could be further refined. These findings highlight the distinction between the suitability of instructional content and the manner in which it is presented to learners. Maharani et al. (2023) similarly examined Android-based learning media in vocational informatics education, while Imamah & Hasanah (2023) developed an Android-based electronic module for network operating systems. Both studies provide relevant examples of how mobile instructional resources can be adapted to technical learning contexts. The present study contributes to this area by focusing on practical Java programming activities and integrating instructional explanations with opportunities to execute simple programs.

The practicality assessments revealed generally favourable responses from teachers and students, although their evaluations differed across particular dimensions. The teacher assessment yielded an overall score of 82.92%, with ease of use receiving the highest rating (91.67%) and timeliness the lowest (75.00%). The student assessment produced an overall score of 81.60%, with accuracy receiving the highest rating (85.68%) and ease of use the lowest (78.34%). These differences suggest that the application's usability may be experienced differently by teachers and learners. While the teacher considered the module relatively straightforward to operate, students indicated greater room for improvement in its ease of use. This finding is important because students are the primary users of the practicum activities and must navigate the learning resources independently. The results are relevant to the work of Hasanah et al. (2024), who developed Android-based instructional media for vocational students, and (Abuna et al., 2025), who examined students' satisfaction with Android-based interactive learning media. The findings reinforce the importance of incorporating user responses into the refinement of digital learning resources rather than relying exclusively on expert validation.

The educational contribution of the developed module lies in its integration of programming instruction and practical learning opportunities within a mobile environment. The application provides access to instructional materials, step-by-step practicum guidance, learning videos, Java programming activities, and interactive evaluations. These features offer students opportunities to revisit explanations and complete simple programming exercises beyond conventional classroom instruction. Faqih (2020) identified the potential of Android-based mobile learning to provide more flexible learning experiences, while Ghafara et al.

(2022) highlighted the relevance of Android-based instructional media for vocational education. The present study builds on these developments by providing a practicum-oriented application specifically designed around Java programming operators. Its contribution is particularly relevant to vocational learning environments where students require repeated exposure to programming procedures and opportunities to connect theoretical explanations with practical tasks. The module also provides teachers with an additional resource for organizing programming instruction and supporting students who need further guidance during independent learning.

The findings also have implications for programming education in an environment where students increasingly use generative artificial intelligence to complete coding assignments. Preliminary observations at SMK Negeri 2 Padang indicated that some students relied on AI-generated solutions without fully understanding the underlying programming procedures. Research by Sun et al. (2024) demonstrated that ChatGPT-assisted programming can influence students' programming behaviours, while Humble et al. (2024) discussed the opportunities and challenges associated with generative AI in programming education. The developed module offers a complementary instructional resource by organizing programming activities around explanations, examples, and practical exercises that students can examine directly. Its relevance lies in providing a structured environment for engaging with programming procedures rather than concentrating solely on completed code. Nevertheless, the present study did not directly measure students' conceptual understanding, debugging proficiency, or dependence on AI-generated solutions. Any potential contribution to these learning outcomes therefore requires further empirical investigation.

Several limitations should be considered when interpreting the findings. The study was conducted at a single vocational school and involved 25 students, one teacher for the practicality assessment, and a limited number of expert validators. The instructional content focused primarily on Java programming operators rather than the full range of object-oriented programming concepts. Moreover, the evaluation relied on expert judgments, functional testing, and user-response questionnaires without a controlled comparison or pretest–posttest assessment of learning achievement. Consequently, the findings establish the module's validity and practicality within the research setting but do not demonstrate its effectiveness in improving programming performance. Future studies should involve larger and more diverse student populations, examine application compatibility across different Android devices, and employ experimental or quasi-experimental designs to investigate

changes in conceptual understanding, debugging skills, learning independence, and programming achievement. Such investigations would provide stronger evidence of the educational value of Android-based practicum modules in vocational programming instruction.

CONCLUSION

Based on the research findings and discussion, this study concludes that the Android-based practicum module developed for the Object-Oriented Programming subject among Grade XI Software and Game Development students at SMK Negeri 2 Padang met the established criteria for validity and practicality. The module was developed using the iterative System Development Life Cycle (SDLC), which allowed continuous refinement through planning, requirements analysis, design, development, testing, and implementation. The resulting application integrates instructional materials, practicum guidelines, learning videos, simple Java programming exercises, and interactive assessments into a single mobile learning environment. Alpha testing demonstrated that all tested functions operated as expected, achieving a functional validity score of 100%. Expert assessments yielded a media validity score of 85.12% and a material validity score of 86.96%, both classified as very valid. Practicality assessments also produced favourable results, with scores of 82.92% from the teacher and 82.60% from students, placing the module in the very practical category.

These findings indicate that the developed module is a suitable supplementary learning resource for supporting Java programming practicum activities in vocational education. By providing accessible instructional guidance and opportunities for simple programming practice, the application offers students greater flexibility in reviewing learning materials and completing practical tasks. The study contributes to the development of mobile-based vocational learning resources by integrating instructional content, programming activities, and assessment features within an Android application. Nevertheless, the findings are limited to product validity and practicality and do not establish improvements in students' programming achievement or conceptual understanding. Further research is recommended to evaluate the module's effectiveness through experimental studies involving larger student populations, particularly in relation to programming proficiency, debugging skills, and independent learning. Additional improvements to

application compatibility, instructional presentation, and usability may also strengthen its implementation across different vocational learning environments.

REFERENCES

Abuna, G., Mulyanto, A., Ashari, S. A., Rohandi, M., Dwinanto, A., & Suwandi, I. (2025). Analisis Tingkat Kepuasan Siswa terhadap Penggunaan Media Pembelajaran Interaktif Berbasis Aplikasi Android di SMK Negeri 1 Limboto. *Inverted: Journal of Information Technology Education, 5*(2), 171–183. https://ejurnal.ung.ac.id/index.php/inverted/article/view/32058

Asosiasi Penyelenggara Jasa Internet Indonesia. (2023). *Survei Penetrasi & Perilaku Internet 2023*. https://survei.apjii.or.id/survei/2023

Badan Standar, Kurikulum, dan Asesmen Pendidikan. (2025). *Keputusan Kepala Badan Standar, Kurikulum, dan Asesmen Pendidikan Kementerian Pendidikan Dasar dan Menengah Nomor 046/H/KR/2025 tentang Capaian Pembelajaran pada Pendidikan Anak Usia Dini, Jenjang Pendidikan Dasar, dan Jenjang Pendidikan Menengah*. Kementerian Pendidikan Dasar dan Menengah. https://kurikulum.kemendikdasmen.go.id/file/1753929861_manage_file.pdf

Faqih, M. (2020). Efektivitas Penggunaan Media Pembelajaran Mobile Learning Berbasis Android dalam Pembelajaran Puisi. *Konfiks: Jurnal Bahasa, Sastra dan Pengajaran, 7*(2), 27–34. https://doi.org/10.26618/konfiks.v7i2.4556

Ghafara, S. T., Simatupang, W., Ambiyar, Muskhir, M., & Irfan, D. (2022). Pengembangan Media Pembelajaran Berbasis Android – Pembelajaran Inovatif untuk Meningkatkan Kompetensi Siswa SMK. *ZONAsi: Jurnal Sistem Informasi, 4*, 1–17.

Hasanah, F. N., & Untari, R. S. (2021). *Buku Ajar Rekayasa Perangkat Lunak*. Umsida Press. https://doi.org/10.21070/2020/978-623-6833-89-6

Hasanah, S. U., Irsyadunas, & Novita, R. (2024). Pengembangan Media Pembelajaran Berbasis Android pada Mata Pelajaran Dasar-Dasar Desain Komunikasi Visual Kelas X DKV di SMK Negeri 1 Pasaman. *Jurnal Ilmiah Sains Teknologi dan Informasi, 2*(2), 1–9. https://jurnal.alimspublishing.co.id/index.php/JITI/article/view/704

Humble, N., Boustedt, J., Holmgren, H., Milutinovic, G., Seipel, S., & Östberg, A.-S. (2024). Cheaters or AI-enhanced learners: Consequences of ChatGPT for programming education. *Electronic Journal of e-Learning, 22*(2), 16–29. https://doi.org/10.34190/ejel.21.5.3154

Husain, S. M., Herdiansah, A., Prianggodo, D. Y., & Jonni, M. (2025). Pemanfaatan Flutter dalam Pengembangan Sistem Media Informasi Teknik Informatika Universitas Muhammadiyah Tangerang. *JIKA (Jurnal Informatika), 9*(1), 111–117. https://jurnal.umt.ac.id/index.php/jika/article/view/13403

Imamah, N., & Hasanah, F. N. (2023). Android-based e-module development on network operating system material: Pengembangan E-Modul Berbasis Android pada Materi Sistem Operasi Jaringan. *Indonesian Journal of Education Methods Development, 18*(2). https://ijemd.umsida.ac.id/index.php/ijemd/article/view/737

Maharani, Rini, F., & Pratama, A. (2023). Pengembangan Media Pembelajaran Berbasis Android pada Mata Pelajaran Informatika Kelas X di SMK Nusatama Padang. *Jurnal*

Pustaka Data (Pusat Akses Kajian Database, Analisa Teknologi, dan Arsitektur Komputer), 3(1), 19–24. https://jurnal.pustakagalerimandiri.co.id/index.php/pustakadata/article/view/416

Maulida, U. (2022). Pengembangan Modul Ajar Berbasis Kurikulum Merdeka. *Tarbawi: Jurnal Pendidikan dan Pemikiran Islam, 5*(2), 130–138. https://ejournal.ibi.ac.id/Tarbawi/article/view/392

Najuah, Lukitoyo, P. S., & Wirianti, W. (2020). *Modul Elektronik: Prosedur Penyusunan dan Aplikasinya*. Yayasan Kita Menulis. https://digilib.unimed.ac.id/id/eprint/49006/

Pratama, D., & Sujatmiko, B. (2024). Rancang Bangun Media Praktik Pemrograman Berbasis Web untuk Meningkatkan Kompetensi Web Developer pada Siswa SMK. *IT-Edu: Jurnal Information Technology and Education, 9*(1), 101–109. https://ejournal.unesa.ac.id/index.php/it-edu/article/view/59006

Putra, J., Indah, D. R., & Firdaus, M. A. (2023). Analisis Kepuasan Pengguna pada E-Learning Menggunakan Metode End User Computing Satisfaction. *Jurnal Sisfokom (Sistem Informasi dan Komputer), 12*(1), 45–52. https://jurnal.atmaluhur.ac.id/index.php/sisfokom/article/view/1575

Putri, N. L. P. S. U., & Jayanta, I. N. L. (2023). Digital practicum module in integrated science for elementary school subjects. *Journal of Education Technology, 7*(2), 343–350.

Rahni, W. S., Novita, R., & Mulyono, H. (2025). Pengembangan E-Modul Berbasis Project Based Learning pada Pelajaran Informatika Kelas X DKV di SMK Negeri 4 Payakumbuh. *PeTeKa: Jurnal Penelitian Tindakan Kelas dan Pengembangan Pembelajaran, 8*(2), 816–828. https://jurnal.um-tapsel.ac.id/index.php/ptk/article/view/21631

Ratnaduhita, N., Sudianto, Y., & Kusumawati, A. (2023). ISO/IEC 25010: Analisis Kualitas Sistem E-Learning sebagai Media Pembelajaran Online. *Journal of Information System, Graphics, Hospitality and Technology, 5*(1), 8–20. https://doi.org/10.37823/insight.v5i1.302

Samudra, A. A., & Maharani, A. D. (2023). SINOPE – Sebagai Media Pembelajaran Inovasi Pembelajaran Biologi Berbasis Website pada Program Studi Pendidikan Biologi Universitas PGRI Sumatera Barat. *Jurnal Edik Informatika: Penelitian Bidang Komputer Sains dan Pendidikan Informatika, 9*(1).

Sugiyono. (2019). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Alfabeta.

Sun, D., Boudouaia, A., Zhu, C., & Li, Y. (2024). Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study. *International Journal of Educational Technology in Higher Education, 21*, Article 14. https://doi.org/10.1186/s41239-024-00446-5

UNESCO. (2023). *Global education monitoring report 2023: Technology in education: A tool on whose terms?* https://doi.org/10.54676/UZQV8501

Widodo, S., Ladyani, F., Asrianto, L. O., Rusdi, Khairunnisa, Lestari, S. M. P., Wijayanti, D. R., Devriany, A., Hidayat, A., Dalfian, Nurcahyati, S., Sjahriani, T., Armi, Widya, N., & Rogayah. (2023). *Buku Ajar Metode Penelitian*. CV Science Techno Direct. https://eprints.poltekkesadisutjipto.ac.id/id/eprint/2780/