e-ISSN 2549-7472
Edumatic: Jurnal Pendidikan Informatika
Volume 10, Issue 2, **420-429.**
https://e-journal.hamzanwadi.ac.id/index.php/edumatic
DOI: 10.29408/edumatic.v10i2.35647

Human-in-the-Loop LLM Assessment for Programming Education: Design and Empirical Validation

Ahmad Muyassar Ibrahim [1,*], Runal Rezkiawan [2]

[1] UIN Alauddin Makassar, Indonesia
[2] Universitas Muhammadiyah Makassar, Indonesia
[*] Corresponding author: Ahmad Muyassar Ibrahim, UIN Alauddin Makassar, Indonesia
✉ ahmad.muyassar@uin-alauddin.ac.id

Received: 20 June 2026 | Revised: 06 July 2026 | Accepted: 14 August 2026 | Published: 25 August 2026

Abstract
Programming instructors face the challenge of providing prompt and consistent feedback, yet manual grading becomes unsustainable in large classes. While Large Language Models (LLMs) offer grading assistance, most research is based on English-language contexts and offline assessments, creating uncertainty about their dependability and the extent of human oversight needed. This study aimed to create and assess an evaluation ecosystem that integrates learning management, AI-driven task creation, LLM grading, and human review for programming courses taught in Indonesian. Employing an ADDIE-based Research and Development approach, the system was implemented for 109 students. For grading validation, instructors independently evaluated 50 assignments without access to AI predictions. The agreement was substantial, with a mean absolute error (MAE) of 4.14, a Pearson correlation of 0.986 within a 95 percent confidence interval ranging from 0.975 to 0.992, and an intraclass correlation coefficient (ICC) of 0.977. Instructor adjustments were more frequent for open-ended tasks (34.9 percent) compared to quizzes (20.2 percent). These results contributed to the development of the task-dependent human calibration (TDHC) model. The system attained a System Usability Scale (SUS) score of 88.5 and cut grading time by 87.5 percent, facilitating focused instructor review in LLM-supported programming assessments.

Keywords: automated programming assessment; empirical validation; human-ai collaboration; large language models; programming education

To cite this article: Ibrahim, A. M., & Rezkiawan, R. (2026). Human-in-the-Loop LLM Assessment for Programming Education: Design and Empirical Validation. *Edumatic: Jurnal Pendidikan Informatika*, *10*(2), 420–429. https://doi.org/10.29408/edumatic.v10i2.35647

INTRODUCTION

In higher education, the application of artificial intelligence is becoming more prevalent in areas such as teaching, learning, and assessment (Zawacki-Richter et al., 2019). Assessment warrants special focus because the value of a score hinges not only on its precision but also on the transparency and amendability of the scoring process when needed (Messick, 1995; Zhai et al., 2021). This is particularly pertinent in programming courses, where evaluation involves concurrent assessments of accuracy, structure, completeness, and adherence to sound programming practices (Pressman & Maxim, 2020). To ensure the responsible use of LLM-based grading, it is crucial that the scores are backed by evidence of reliability and that there is a defined role for human oversight.

This concern is significant for education. Formative feedback is most beneficial when provided to students promptly, allowing them to amend their work and manage their learning independently (Black & Wiliam, 1998; Yan & Pastore, 2022; Zimmerman, 2002). In programming classes, traditional grading systems can complicate this process. Feedback might be delivered too late for students to make iterative improvements, and scoring can differ between assessors or grading periods (Bernik et al., 2025). Additionally, learning-management systems like Moodle are primarily built to handle course activities rather than semantically analyse source code (Mohamed et al., 2025). With increasing class sizes, educators thus encounter a practical challenge in balancing the quality of assessments with the need to provide timely feedback.

Research has indicated that LLMs can achieve programming scores nearly equivalent to human evaluations, although comparing these results is complex. Various studies have employed distinct human benchmarks and assessment methods: some align the model with a single instructor (Mohamed et al., 2025), others rely on teaching assistant evaluations (Poličar et al., 2025), and some assess scores at the class level (Chiang et al., 2024). Additional research has evaluated model accuracy without implementing them in classrooms (Bernik et al., 2025) or explored their role as teaching assistants and feedback mechanisms (Denny et al., 2024). Numerous assessments emphasise correlation or exact match, indicating whether two score sets align but not fully capturing their absolute concordance. Consequently, metrics such as intraclass correlation and limits of agreement are valuable for determining whether AI-generated scores are sufficiently similar to human scores for effective formative assessment.

Another constraint of the existing evidence relates to language. Most available research has concentrated on evaluating LLM grading in English, despite programming courses being delivered and evaluated in various other languages. The language employed in rubrics, task instructions, student explanations, and feedback can influence the model's interpretation of a response. The reliability confirmed in English does not necessarily extend to other contexts. In Indonesia, recent AI applications in programming education have primarily targeted learning support, such as tutoring chatbots (Waleska et al., 2025), whereas the reliability of AI-driven assessments has been less explored.

Research has indicated that LLMs have the potential to mimic human grading in programming education. However, there is still a lack of comprehensive evidence regarding their dependability in non-English teaching environments and the specific scenarios where human evaluation is indispensable. Most assessments focus on general accuracy or score alignment, offering limited understanding of how closely they match instructor evaluations and how assessment features affect the necessity for human involvement. This raises a crucial question: Can LLM-based grading deliver reliable scores in Indonesian-language programming education while ensuring adequate instructor supervision across various task types? To tackle this question, it is essential to not only confirm the alignment between LLM and instructor scores but also to investigate when and how extensively instructors modify AI-generated evaluations.

This study was designed to gather empirical data on the dependability and practical application of assessments based on LLMs in the context of Indonesian-language programming education. This was done by evaluating the alignment between scores given by LLMs and those assigned by instructors, examining how instructors modify scores across different task types, and considering the implications for human oversight. Through a blind validation process that uses complementary metrics for score correlation and absolute agreement, this study provides empirical evidence. It also shows how instructor modifications can signal the need for human intervention and develops these observations into the Task-Dependent Human Calibration (TDHC) model. By associating the reliability of LLM scoring with task-specific human

evaluation, this study offers a data-driven framework for collaboration between humans and AI in programming assessments.

METHOD

In the Even Semester of 2026/2027, the ADDIE model (Branch, 2009; Adeoye, 2024) was employed within a Research and Development framework at the Department of Informatics Engineering, UIN Alauddin Makassar. The selection of ADDIE was due to its structured phases, which facilitated the development, deployment, and assessment of the system within a single academic semester while also permitting phase-by-phase revisions.

The Analysis phase identified four essential needs: session-based organisation of course materials, automated feedback post-submission, self-directed quizzes, and AI-supported assignment creation from existing materials. These requirements were transformed in the Design phase into a three-tier architecture: a Presentation Layer (comprising a student portal and instructor CMS), an Application Layer (featuring a serverless REST API), and an AI Layer utilising the DeepSeek-V3 model. Subsequently, seven modules were constructed during the Development phase. In the Implementation phase, the finalised platform, materiku.dosengpt.com, was deployed over 16 sessions for each course. The Evaluation phase assessed functional performance, the correlation between AI and instructor scores, instructor override tendencies, usability, and the efficiency of grading.

Upon a student's submission, the server compiled task context, scoring rubric, and student code into a prompt and sent it to the model in JSON-mode with temperature 0.1. The model evaluated five weighted criteria: instruction alignment (25%), conceptual understanding (25%), implementation quality (20%), completeness (15%), and originality (15%). The server calculated the final score from these weights rather than a single total from the model. Feedback was provided in Indonesian. Instructors could adjust scores, and the system retained both the initial AI score and instructor-approved score, enabling tracking of modification frequency, direction, and magnitude. A function generated assessment tasks at a temperature of 0.7. Figure 1 illustrates the process from submission to instructor review.

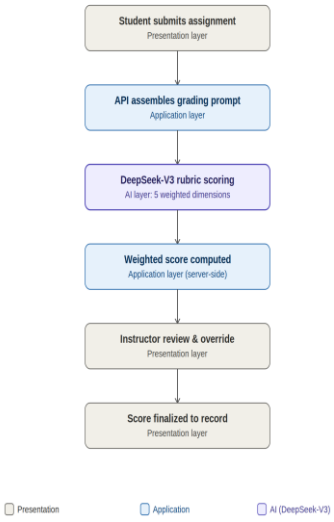

Figure 1. System architecture and end-to-end AI grading data flow

The research included 109 students: 77 in web programming, 59 in mobile programming, and 28 in both courses. Each completed the System Usability Scale (SUS) questionnaire (Brooke, 1996). During the semester, 186 open-ended assignments were submitted; 49 (26.3%) were identified as AI-generated and 137 were not. Because the screening method lacked

validation, the flag was used only to establish the validation sampling frame, not as proof of plagiarism or academic dishonesty. For grading precision, 50 unflagged assignments were randomly chosen and instructor-assessed without AI scores; 27 were from web programming and 23 from mobile programming. Evaluation involved a 35-scenario black-box checklist, blind comparison of AI and instructor scores, and the 10-item SUS with a 5-point scale.

The model used provider's "deepseek-chat" alias, not a fixed version, so reliability results pertain to study-period model. Figures omitted names and numbers; academic records were used. AI and instructor scores were compared using mean absolute error (MAE), root-mean-square error (RMSE), Pearson and Spearman correlations, and a two-way random-effects intraclass correlation for absolute agreement [ICC(2,1)]. Bland–Altman analysis assessed mean difference, 95% limits of agreement, and a regression test for proportional bias. A paired-samples t-test and Cohen's d examined differences. Instructor overrides were summarized by frequency, direction, and size per task type. SUS responses were scored by procedure, reporting mean and standard deviation. Grading efficiency was based on time per submission before and after deployment.

RESULTS AND DISCUSSION

Results

The results followed the ADDIE sequence described in the Methods section. The four requirements identified during the Analysis guided the system design, resulting in a three-layer architecture with seven modules for course materials, quizzes, assignments, task generation, automated grading, grade management, and a glossary. The platform was deployed with 109 students across two courses. Figures 2 and 3 present the two main AI components: an assignment generator that extracts content from course materials and a grading interface that provides Indonesian-language code analysis, scores, and instructor override functionality.

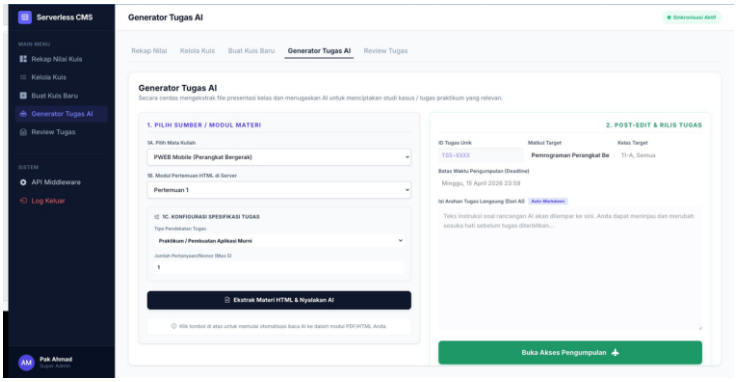

Figure 2. AI Task Generator for creating assignments from uploaded course materials

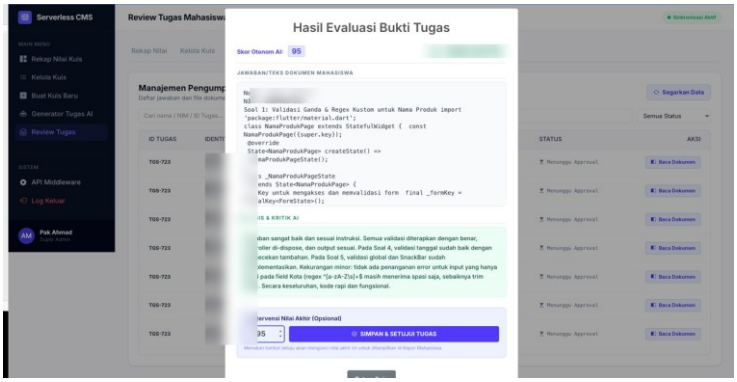

Figure 3. AI grading output showing a score of 95/100 and detailed flutter code analysis in Indonesian

In black-box testing, 35 scenarios were evaluated, focusing on authentication, content management, quizzes, the AI task generator, and the grading module. Of these, 34 scenarios were successful, resulting in a pass rate of 97.1% (Table 1). The sole failure was due to a student submission containing unusual Unicode characters, which led to a malformed grading prompt. This issue was resolved by cleaning the input before it was processed by the model. As the failure was linked to input handling and not the scoring process itself, it did not impact the grading-validation data used in subsequent analyses.

Table 1. Black-box testing results summary

No.	Feature Tested	Scenarios	Pass	Fail	(%)
1	Student & CMS authentication	6	6	0	100%
2	Course material display & navigation	5	5	0	100%
3	Quiz creation & execution	7	7	0	100%
4	Assignment submission & AI grading	8	7	1	87.5%
5	AI Task Generator	5	5	0	100%
6	Grade dashboard & export	4	4	0	100%
Total		35	34	1	97.1%

Throughout the semester, the AI Task Generator created three assignments (TGS-655, TGS-642, and TGS-723), each comprising eleven questions focused on implementation and analysis. These assignments were derived from the course materials provided and utilised in class sessions without any modifications by the instructor. Consequently, the generated tasks were applicable for classroom assessments. Nonetheless, the limited number of assignments produced does not allow for a comprehensive assertion regarding the quality of the items without further independent assessment. Additionally, the system identified 49 out of 186 open-ended submissions (26.3%) as AI-generated. Since the accuracy of this screening was not assessed, this statistic is presented solely as an observation from the deployment.

Table 2. AI vs. instructor grading accuracy comparison (n=50 real assignments)

Course	n	MAE	RMSE	Pearson r [95% CI]	ICC (2,1)	≤5 pts
Web Programming	27	4.11	4.79	0.986 [0.970, 0.994]	0.978	70.4%
Mobile	23	4.17	5.07	0.984 [0.963, 0.993]	0.975	69.6%
Combined	50	4.14	4.92	0.986 [0.975, 0.992]	0.977	70.0%

Table 2 presents a comparison of grading between AI and instructors without revealing their identities. For the 50 assignments evaluated, the mean absolute error was 4.14 points, and the RMSE was 4.92 points on a 100-point scale. The correlation between the AI and instructor scores was notably strong, with a Pearson correlation coefficient of 0.986 (95% CI [0.975, 0.992], $p<0.001$) and a Spearman's rho of 0.965. The level of absolute agreement was also high, as indicated by ICC(2,1) of 0.977 (95% CI [0.914, 0.990]). In 70% of the cases, the scores differed by no more than five points, and none exceeded a 10-point difference. Figure 4 illustrates the relationship between the scores for each course. The Bland–Altman analysis shown in Figure 5 indicates a mean difference of -3.10 points, with 95% limits of agreement ranging from -10.67 to 4.47. The regression test did not find evidence of proportional bias (slope=-0.015, $p=0.540$). However, a paired comparison revealed a statistically significant average difference ($t=-5.68$, $p<0.001$; Cohen's $d=-0.80$), although the mean difference was

minor on a 100-point scale. As shown in Figure 4, the observations from both courses were close to the identity line. The consistent pattern across Web and Mobile Programming suggests that the strong correlation in scores was not solely due to one of the courses.

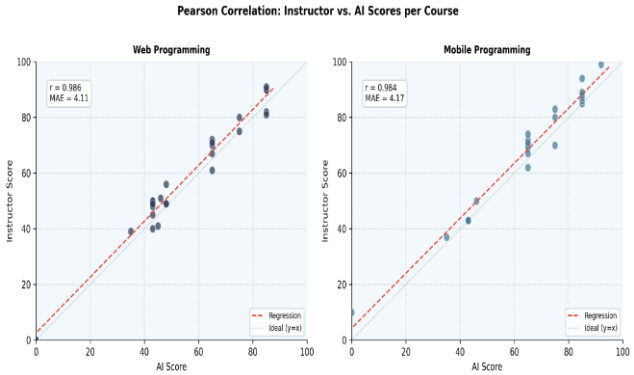

Figure 4. Pearson 'scorrelation scatter plots per course

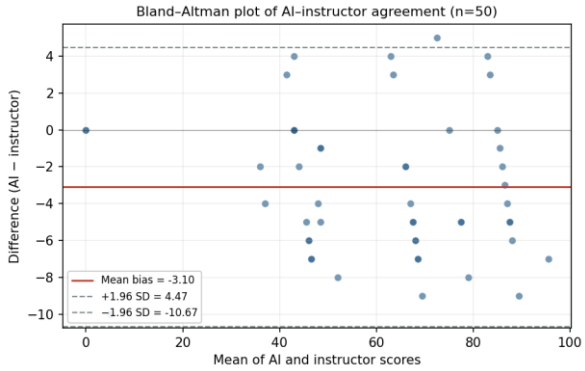

Figure 5. Bland-Altman plot of AI-instructor score agreement (n=50)

Figure 5 shows the same comparison using absolute agreement. The majority of score differences fell within the 95% limits of agreement, with a negative mean difference indicating that the AI generally assigned slightly lower scores than the instructor. The regression test did not reveal any significant proportional bias throughout the entire score range. However, the analysis of score bands indicated that absolute errors were somewhat greater for submissions with higher scores.

The grading error size varied across different instructor score bands. For scores ranging from 0 to 40, the mean absolute error was 3.17 points; for scores between 41 and 70, it was 3.74 points; and for scores from 71 to 100, it was 4.86 points. In 36 out of 50 instances (72.0%), the AI assigned a lower score than the instructor, matched the instructor's score in seven instances (14.0%), and gave a higher score in another seven instances (14.0%). The five largest discrepancies all involved AI scores being lower than those of the instructor, with four of these occurring in submissions where instructor scores ranged from 74 to 94. Consequently, while no statistical proportional bias was found, the most significant absolute disagreements were observed in stronger submissions.

Table 3. Instructor override activity for quiz scoring (n=203 submissions)

Quiz	n	Overridden	Avg. Adjustment	Direction (↑/↓)
KUIS-PPB-1	101	30 (29.7%)	+3.43	Mostly increased
KUIS-PPB-2	102	11 (10.8%)	+2.27	Mostly increased
Combined	203	41 (20.2%)	+3.12 (avg.); 5.12 (abs.)	87.8% ↑ / 12.2% ↓

Throughout the two quizzes, there were 203 submissions recorded by the system, with an average score of 92.5, as shown in Table 3. Instructors modified 41 of the AI-generated quiz scores, accounting for 20.2% of the total. Most of these modifications resulted in higher scores, with 87.8% of changes being increases, and the average absolute change was 5.12 points. In contrast, open-ended programming assignments showed a different trend: 65 out of 186 scores were altered, representing 34.9%, with an average absolute change of 12.18 points, and downward adjustments were more frequent than in the quizzes, occurring at 24.6% compared to 12.2%. The quizzes themselves also varied. The quiz involving free-form data-passing logic had a higher override rate of 29.7%, while the more structured quiz had a rate of 10.8%. These variations imply that instructor intervention was more prevalent as the response space expanded, a topic further explored in the Discussion section.

Table 4. Overall system usability scale results

Sample	n	SUS Average	Std. Dev.	Category
Overall	109	88.5	3.8	Excellent

All 109 students participated in the SUS questionnaire, achieving an average score of 88.5 with a standard deviation of 3.8, which falls within the Excellent range (Bangor et al., 2009; Table 4). To ensure that the 28 students enrolled in both courses were not counted twice, Table 4 consolidates the 109 unique respondents into a single sample. The time required for grading each submission was reduced from approximately eight minutes to just one minute, representing an 87.5% decrease. For a total of 203 submissions, this reduction translates to saving approximately 23.7 hours of instructor time. These metrics were collected during regular classroom activities, reflecting the performance of the system in a real-world setting rather than in a laboratory environment.

Discussion

The results indicate that grading using LLMs can closely align with instructor evaluations in Indonesian-language programming courses, provided that the assessment follows a clear rubric. The strong Pearson correlation ($r = 0.986$) and ICC(2,1) (0.977) reflect a high level of agreement between the AI-generated and instructor-assigned scores. Meanwhile, the MAE of 4.14 points shows a relatively minor average difference on a 100-point scale. Nonetheless, the notable paired difference and Bland–Altman mean difference of −3.10 points revealed that a high correlation does not equate to complete parity between AI and instructor scores. This distinction is crucial because educational assessments must be both consistent and interpretable, allowing for adjustments when discrepancies arise (Messick, 1995; Zhai et al., 2021). Consequently, the findings imply that the observed agreement is due not only to the LLM's capabilities but also to the structured rubric and the separation between model scoring and final score determination.

The pattern of disagreement sheds additional light on the circumstances in which LLM grading might be less dependable. While the Bland–Altman regression did not reveal a notable proportional bias, the MAE rose from 3.17 points for submissions with lower scores to 4.86 points for those with higher scores. In 72.0% of the blind-graded instances, the LLM awarded lower scores than the instructor, with the most significant discrepancies found in stronger submissions. This could be because high-quality programming solutions often feature alternative implementations and qualitative differences that demand more contextual understanding. A rubric-based model might adhere more strictly to predefined criteria compared to an instructor who can evaluate overall competence. Consequently, the findings

suggest that the reliability of LLM grading may depend on the specific characteristics of student responses rather than being consistent across all score levels.

These instructor-override outcomes support this interpretation. Instructors altered 20.2% of quiz scores, whereas 34.9% of scores for open-ended assignments were modified, with the latter experiencing significantly larger changes. The contrast between the two quizzes further demonstrated that the open-ended task led to more overrides than the restricted task. These trends imply that the necessity for human judgment increases when the response space is more adaptable and multiple correct programming solutions exist. This observation aligns with earlier research showing that LLM-based grading can closely mimic human evaluation but is still influenced by task specifics, model behaviour, and evaluation conditions (Chiang et al., 2024; Mohamed et al., 2025; Poličar et al., 2025). Consequently, human oversight should be considered a task-sensitive element of assessment rather than a standard final verification step.

These results from the empirical foundation for the Task-Dependent Human Calibration (TDHC) model introduced in this study. The TDHC model views human participation as a variable element in the evaluation process, with increased scrutiny applied to tasks where disagreements are more probable. This approach advances earlier studies by shifting the focus from whether a large language model (LLM) can replicate instructor scores to when instructor intervention is still needed. The aim is not to replace instructor judgment but to offer an evidence-based justification for more selective human review. This strategy could enable institutions to maintain the scalability of automated grading while ensuring that instructor judgment is applied to assessments that demand more interpretive flexibility.

This study expands on existing findings by integrating various grading validation methods. While earlier studies have shown that LLMs can closely mimic instructor grading in programming education (Chiang et al., 2024; Mohamed et al., 2025; Poličar et al., 2025), this study uniquely combines correlation, ICC, MAE, and Bland–Altman analysis with blind instructor grading and a systematic review of instructor overrides. This methodology differentiates between score correlation and absolute agreement, revealing not only score discrepancies but also their direction and practical significance. Additionally, the study's focus on the Indonesian-language context broadens the empirical reach of LLM-based programming assessment, suggesting that grading reliability must be confirmed within the specific language, rubric, and assessment setting where an LLM is utilised.

The practical implications are evident from the high usability score (SUS = 88.5) and 87.5% decrease in grading time, illustrating the practicality of using LLMs for assessments in typical classroom settings. Nonetheless, this efficiency should not be considered a reason to eliminate the participation of instructors. Instead, the results advocate for selective human oversight, especially in open-ended tasks where disagreements are more probable, whereas prompt assessments can enhance the feedback and learning processes (Yan & Pastore, 2022). These findings should be considered within the study's limitations: it was conducted at a single institution, involved two programming courses, one LLM, and a validation sample of 50 assignments. Because task openness was observed rather than experimentally controlled, the TDHC pattern should be viewed as an empirically based approach rather than a universally validated threshold model for creativity. Additional research across various institutions, languages, programming fields, and LLMs is required to determine their broader applicability.

CONCLUSION

In this study, the potential of LLM-based grading to yield consistent scores in Indonesian-language programming education was explored alongside the variation in instructor review necessity based on task type. Within the blind-grading sample, there was a strong correlation with instructor scores (MAE=4.14; r=0.986; ICC=0.977). However, instructors adjusted a higher percentage of scores for open-ended tasks compared to quizzes. This trend aligns with

the Task-Dependent Human Calibration (TDHC) model, which suggests that human review should intensify as tasks permit a broader spectrum of valid answers. The method was applied in an operational classroom system that demonstrated high usability (SUS=88.5) and significantly decreased grading time. The results advocate for a targeted form of human oversight, focusing instructor attention on tasks and submissions where discrepancies are more probable. Additional research across various institutions, languages, and LLMs is necessary to assess the generalisability of this calibration pattern.

REFERENCES

Adeoye, M. A. (2024). Revolutionizing education: Unleashing the power of the ADDIE model for effective teaching and learning. *Jurnal Pendidikan Indonesia*, *13*(1), 202–209. https://doi.org/10.23887/jpiundiksha.v13i1.68624

Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, *4*(3), 114–123.

Bernik, A., Radošević, D., & Čep, A. (2025). A comparative study of large language models in programming education: Accuracy, efficiency, and feedback in student assignment grading. *Applied Sciences*, *15*(18), 10055.

Black, P., & Wiliam, D. (1998). Assessment and classroom learning. *Assessment in Education: Principles, Policy & Practice, 5*(1), 7-74. https://doi.org/10.1080/0969595980050102

Branch, R. M. (2009). *Instructional design: The ADDIE approach*. Springer. https://doi.org/10.1007/978-0-387-09506-6

Brooke, J. (1996). SUS: A 'quick and dirty' usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry* (pp. 189–194). Taylor & Francis.

Chiang, C.-H., Chen, W.-C., Kuan, C.-Y., Yang, C., & Lee, H.-Y. (2024). Large language model as an assignment evaluator: Insights, feedback, and challenges in a 1000+ student course. *Computers & Education: Artificial Intelligence*, *7*, Article 100274. https://doi.org/10.1016/j.caeai.2024.100274

Denny, P., Leinonen, J., Prather, J., Luxton-Reilly, A., Amarouche, T., Becker, B. A., & Reeves, B. N. (2024). Prompt problems: A new programming exercise for the generative AI era. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education* (pp. 296–302). ACM. https://doi.org/10.1145/3626252.3630909

Messick, S. (1995). Validity of psychological assessment: Validation of inferences from persons' responses and performances as scientific inquiry into score meaning. *American Psychologist, 50*(9), 741-749. https://doi.org/10.1037/0003-066X.50.9.741

Mohamed, K., Yousef, M., Medhat, W., Mohamed, E. H., Khoriba, G., & Arafa, T. (2025). Hands-on analysis of using large language models for the auto evaluation of programming assignments. *Information Systems*, *130*, Article 102523. https://doi.org/10.1016/j.is.2024.102523

Parameswara, D. A. D., Berlilana, B., & Saputro, R. E. (2026). Utilitarian vs human-centered AI acceptance: Explaining students' adoption of ChatGPT in higher education. *Edumatic: Jurnal Pendidikan Informatika, 10*(1), 190–199. https://doi.org/10.29408/edumatic.v10i1.34218

Poličar, P. G., Stražar, M., & Zupan, B. (2025). Automated assignment grading with large language models: Insights from a bioinformatics course. *Bioinformatics, 41*(Suppl. 1), i21–i29. https://doi.org/10.1093/bioinformatics/btaf216

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Suria, O. (2024). A statistical analysis of System Usability Scale (SUS) evaluations in online learning platform. *Journal of Information Systems and Informatics, 6*(2), 992–1007.

https://doi.org/10.51519/journalisi.v6i2.750

Waleska, R. F., Asnal, H., Rahmiati, R., & Gunadi, G. (2025). Aplikasi chatbot interaktif pembelajaran bahasa pemrograman PHP dengan algoritma NLP berbasis BERT. *Edumatic: Jurnal Pendidikan Informatika, 9*(2), 609–618. https://doi.org/10.29408/edumatic.v9i2.31427

Yan, Z., & Pastore, S. (2022). Assessing teachers' strategies in formative assessment: The Teacher Formative Assessment Practice Scale. *Journal of Psychoeducational Assessment, 40*(5), 592–604. https://doi.org/10.1177/07342829221075121

Yasmine, Y. S., & Hikmawan, R. (2025). ChatGPT sebagai alat bantu dalam penulisan karya ilmiah mahasiswa: Analisis keterlibatan dan kreativitas. *Edumatic: Jurnal Pendidikan Informatika, 9*(1), 99–108. https://doi.org/10.29408/edumatic.v9i1.29496

Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education–where are the educators?. *International journal of educational technology in higher education*, *16*(1), 39. https://doi.org/10.1186/s41239-019-0171-0

Zhai, X., Krajcik, J., & Pellegrino, J. W. (2021). On the validity of machine learning-based next generation science assessments: A validity inferential network. *Journal of Science Education and Technology, 30*(2), 298–312. https://doi.org/10.1007/s10956-020-09879-9

Zimmerman, B. J. (2002). Becoming a self-regulated learner: An overview. *Theory Into Practice, 41*(2), 64–70. https://doi.org/10.1207/s15430421tip4102_2