Indonesian Journal of Educational Development (IJED)
Volume 7, Issue 2, 2026, pp. 875-891
ISSN: 2722-1059 (Online); ISSN: 2722-3671 (Print)
DOI: https://doi.org/10.59672/ijed.v7i2.7020

Cognitive scaffolding module in the Vilokka virtual laboratory for coding and artificial intelligence learning

Gede Saindra Santyadiputra*)[1], I Wayan Santyasa[2], Made Juniantari[3]
[1]Universitas Pendidikan Ganesha, Singaraja, Indonesia; gsaindras@undiksha.ac.id
[2]Universitas Pendidikan Ganesha, Singaraja, Indonesia; santyasa@undiksha.ac.id
[3]Universitas Pendidikan Ganesha, Singaraja, Indonesia; mdjuniantari@undiksha.ac.id
*)Corresponding author: Gede Saindra Santyadiputra; E-mail addresses: gsaindras@undiksha.ac.id

Article Info

Article history:
Received July 21, 2026
Revised August 03, 2026
Accepted August 24, 2026
Available online August 26, 2026

Keywords: ADDIE, Cognitive scaffolding, KKA, Local-LLM, N-Gain, Vilokka, Vocational education

Abstract. Indonesia's Coding and Artificial Intelligence (KKA) curriculum mandate (BSKAP No. 046/H/KR/2025) requires Vocational High School (VHS) students to master Python programming and generative AI, exposing infrastructure inequality and cognitive overload as dual barriers. This study reports five ADDIE phases of developing and evaluating a cognitive scaffolding module integrated with Vilokka, an on-premises virtual laboratory embedding a guardrailed Local Large Language Model (Local-LLM). Participants were recruited through purposive sampling: five KKA teachers (Analysis), three validators (Expert Validation), and 30 VHS students (Implementation and Evaluation). Analysis identified three cognitive barrier patterns: pre-coding abstraction failure, logic-syntax conflation, and AI-induced cognitive passivity. The Design phase produced a 17-module architecture synchronising a three-level scaffolding framework (Sense-making, Process-scaffolding, and Fading) with KKA curriculum elements. The study employed four instruments: a teacher interview guide, an expert validation rubric, pre-test and post-test assessments, and practicality questionnaires. Expert validation yielded a mean validity of 89.72% (Very Valid). The field trial yielded a mean N-Gain of 0.39 (Moderate), confirmed by paired sample t-test ($t(29) = 6.385$, $p < 0.05$; Cohen's $d = 1.17$). Practicality ratings were Very Practical for teachers (88.12%) and students (86.96%). The findings establish a replicable framework for AI-assisted vocational programming instruction, recommended for schools with infrastructure constraints.

Introduction

Indonesia's 2025 national curriculum reform positions Coding and Artificial Intelligence (KKA) as a mandatory subject across VHS programmes. Governed by Ministerial Decree issued by the Indonesian Board of Standards, Curriculum, and Educational Assessment (BSKAP) No. 046/H/KR/2025, the KKA Learning Achievement Component (CP) requires that students, regardless of their vocational specialisation, develop Python programming proficiency and the capacity to apply generative AI tools in relevant vocational tasks (Kemendikdasmen, 2025). This policy decision reflects a broader national imperative: vocational graduates entering Indonesia's digitalising economy must demonstrate applied AI-literacy alongside domain-specific technical competencies.

Digital infrastructure inequality remains a systemic barrier. Indonesia's performance in OECD digital-economy readiness assessments reveals pronounced disparities between urban and rural vocational institutions, with substantial proportions of VHS lacking stable internet connectivity, licensed educational software, or functional technical laboratory facilities (Kaenong et al., 2023; OECD, 2023). For KKA instruction, this creates an immediate operational problem: cloud-based programming environments and commercial generative AI platforms, the tools most naturally associated with the new curriculum, require persistent internet connectivity that many target institutions cannot reliably provide.

Beyond the infrastructure barrier lies a deeper cognitive challenge. Research on Indonesian VHS learners confirms that most students entering KKA instruction lack foundational computational thinking dispositions (Christi & Rajiman, 2023). Programming learning is characterised by a compound cognitive burden: learners must simultaneously manage abstract concepts, syntactic rules, and debugging processes, a load that frequently overwhelms working memory capacity (Hartley et al., 2024; Huang et al., 2025). Huang et al. (2025) characterise the resulting collapse as 'tutorial hell': a pattern in which learners cycle through syntactic corrections without building the conceptual understanding needed to transfer skills to novel problems.

The rapid proliferation of conversational AI tools has introduced a third challenge. Studies consistently document that learners who access LLM-generated code solutions during programming exercises develop cognitive passivity, delegating reasoning to the AI system rather than engaging in productive problem-solving (Mallik & Gangopadhyay, 2023; L. Yan et al., 2024). Xu and Ouyang (2022) frame this through Cognitive Load Theory (CLT) and Vygotsky's Zone of Proximal Development (ZPD): effective scaffolding should reduce extraneous load while preserving generative processing. Scaffolding that eliminates challenge entirely, as code-delivering AI tools tend to do, violates this principle.

Cognitive scaffolding offers a theoretically grounded response to all three challenges. The CLT and ZPD framework introduced above specifies the mechanism: effective scaffolding reduces extraneous load without eliminating the intrinsic load that drives schema formation, and operates within the ZPD window where learner readiness makes support productive. In programming education, frameworks that guide learners through error analysis without providing direct solutions have demonstrated consistent advantages: Zuo et al. (2023); I Made Elia Cahaya et al., (2024) established through meta-analysis that scaffolded online learning produced significantly better outcomes than unscaffolded instruction, and Zeitlhofer et al. (2023); Surya Abadi et al., (2025): Widana & Ratnaya, 2021 found that cognitive and metacognitive prompts improved learning performance. Together, these frameworks prescribe scaffolding calibrated to current learner competence and progressively withdrawn as competence develops, a trajectory embodied in the three-level Sense-making to Fading progression implemented in this study.

Virtual laboratories represent a promising context for deploying such scaffolded instruction in resource-constrained environments. Mercado and Picardal (2023) demonstrated that virtual laboratory simulations produced comparable learning outcomes to physical environments in STEM education. Critically, LLM-augmented virtual environments configured with appropriate pedagogical constraints can deliver context-sensitive scaffolding without triggering cognitive passivity: Ross et al. (2025) and Sunil and Thakkar (2025) showed that constrained LLMs provide guidance without code delivery; J. Yan et al. (2025) demonstrated that system-prompt engineering reliably prevents solution delivery while preserving conceptual explanations; and Zhang (2025) established that guardrailed LLM systems produced superior metacognitive outcomes compared to ungoverned AI access. Building on the authors' prior virtual laboratory research (Santyadiputra et al., 2024, 2025; Santyadiputra & Kustono, 2023), the present study introduces Vilokka (Virtual

Laboratory of KKA): an on-premise virtual laboratory in which learners submit Python code containing intentional errors and receive structured feedback from a guardrailed Local-LLM enforcing three mandatory response sections (Error Category and Location, Theoretical Explanation, and Prompting Question) while absolutely prohibited from delivering code solutions.

Despite this convergence of evidence, a critical gap remained in the literature. Prior studies on scaffolded programming instruction focused on general-purpose coding environments rather than curriculum-specific, on-premise platforms designed for AI-constrained vocational contexts (Ross et al., 2025; Zuo et al., 2023). Studies on guardrailed LLM environments examined system-prompt engineering and metacognitive outcomes J. Yan et al. (2025); Zhang (2025) but did not integrate these with structured instructional modules aligned to a national curriculum framework. No cognitive scaffolding module synchronised with the KKA curriculum had been developed for Vilokka prior to this study, leaving a gap between platform capability and instructional design. The present study is therefore the first to report a complete ADDIE-based development of a cognitive scaffolding module purpose-built for a guardrailed, on-premise LLM virtual laboratory within a nationally mandated AI-coding curriculum, addressing infrastructure constraints, cognitive overload, and AI-induced passivity simultaneously.

This study addresses that gap by reporting all five phases of an ADDIE-based development project. Three research questions guided the study: (1) What cognitive barriers do VHS learners face in KKA Python instruction, and how do they map onto curriculum elements? (2) How can a scaffolding module be designed to address these barriers within the constraints of Vilokka's on-premise, guardrailed-LLM architecture? (3) How effective and practical is the resulting module when implemented in authentic VHS KKA instruction?

Method

Research Method and Design

This study employs a Research and Development (R&D) methodology structured around the ADDIE instructional design model (Analysis, Design, Development, Implementation, Evaluation). ADDIE was selected in preference to Dick and Carey and Hannafin-Peck based on three criteria: (1) its sequential phase structure provides explicit validation gates between phases; (2) its Analysis phase is explicitly separated from Design, supporting the empirical approach in which cognitive barrier data drive design decisions; and (3) it has established precedent in Indonesian vocational education development contexts. This paper reports all five phases of the ADDIE cycle (Branch, 2010).

Participants and Sampling Technique

The study recruited three distinct participant groups through purposive sampling. For the Analysis phase, five VHS teachers who had actively delivered KKA curriculum units at partner schools in Buleleng regency, Bali, participated in structured observation and semi-structured interview sessions. For the Expert Validation phase, three domain-specific validators were recruited: Validator 1 (MAR) holds expertise in KKA subject matter and Python programming pedagogy; Validator 2 (MAU) specialises in instructional design with a focus on cognitive scaffolding and Computational Thinking; and Validator 3 (SAS) holds expertise in educational media and virtual laboratory interface design. For the Implementation and Evaluation phases, participants were 30 VHS students (one intact class) enrolled in KKA instruction at one partner school during 2025/2026; practicality assessment was additionally conducted with four KKA subject-matter teachers from the same institution (Etikan, 2016).

Research Setting and Timeline

The study was conducted at partner VHS representing infrastructure conditions characteristic of public vocational schools. The Implementation phase was integrated within the active learning period (February–April 2026), prior to the end-of-year assessment examination period commencing in May. All 17 Vilokka modules had been fully digitalised and internally tested (Development phase, January–February 2026) before student deployment commenced. Table 1 presents the complete research timeline across all five ADDIE phases.

Table 1. Research Timeline Across All ADDIE Phases

Phase	Period	Activities
Analysis	October -November 2025	Curriculum document analysis (BSKAP No. 046/H/KR/2025); structured observation and semi-structured interviews with 4 KKA teachers at 3 partner VHS
Design	November - December 2025	17-module architecture; three-level scaffolding framework; EAR task format specification; Vilokka Local-LLM integration specification
Expert Validation	December 2025 - January 2026	Structured rubric-based validation by 3 domain-specific validators; documented revisions applied
Development	January -February 2026	Digitalisation of all 17 modules within the Vilokka platform; system prompt deployment and internal testing, completed prior to student deployment
Implementation	February - April 2026	Field trial at one partner VHS: one-group pretest-posttest design (n = 30) across 17 module sessions (two sessions per week); practicality questionnaire administration
Evaluation	April - June 2026	N-Gain analysis; Shapiro-Wilk normality test; Paired Sample t-Test; teacher and student practicality assessment (*V(%)*)

Research Procedures

The study followed a sequential procedure across all five ADDIE phases. In the Analysis phase (October–November 2025), two complementary activities were conducted: structured document analysis of the KKA CP text (BSKAP No. 046/H/KR/2025) and structured observation with semi-structured interviews with the five teacher participants; interviews were audio-recorded, transcribed verbatim, and analysed using inductive thematic analysis.

Image *1* presents the research procedure.

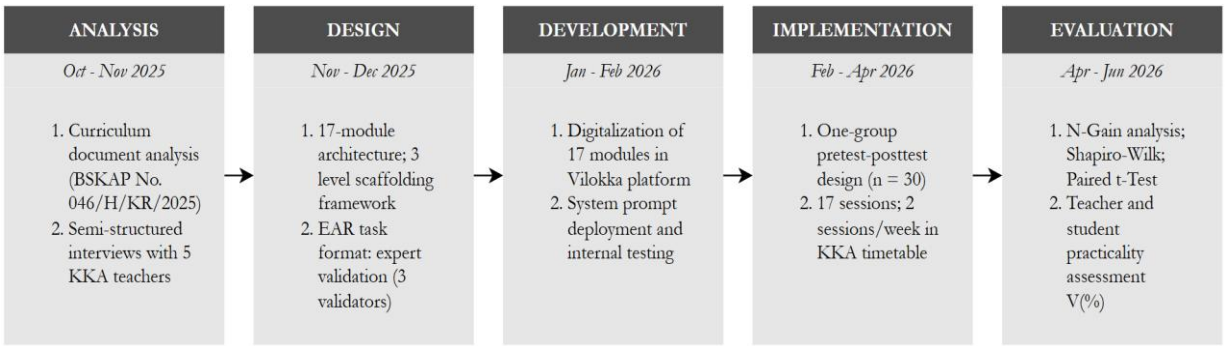

Image 1. Research Procedure Across the Five ADDIE Phases

In the Design phase (November–December 2025), four interconnected outputs were produced: (1) a 17-module architecture mapping each module to a specific CP element and Python topic; (2) a three-level scaffolding framework synchronised with curriculum progression; (3) the EAR task format specification; and (4) a Vilokka integration specification defining Local-LLM system-prompt requirements. Expert validation (December 2025–January 2026) followed immediately after the Design phase, with the compiled module prototype submitted to three domain-specific validators using structured rubric instruments.

In the Development phase (January–February 2026), all 17 modules were systematically digitalised within the Vilokka platform before any student interaction commenced. Each module's four intentional Python errors were encoded in the learner-facing code editor; the three-question EAR worksheet (Q1–Q3) was implemented as a structured digital submission form; and the static Local-LLM system prompt was deployed and tested across all 17 module types. Internal testing was completed with two KKA teachers not participating in the study to verify response consistency and guardrail compliance prior to student deployment in February 2026.

In the Implementation phase (February–April 2026), we used a one-group pretest-posttest design. Prior to the first module session, all 30 participants completed a Python programming competency test (pretest). Across 17 sessions delivered at two sessions per week within the normal KKA timetable during the active learning period of 2025/2026, and completed prior to the end-of-year assessment examination period, students engaged with EAR exercises on Vilokka and received structured feedback from the guardrailed Local-LLM. After completing all 17 sessions in April 2026, participants completed the posttest (parallel-form competency test) and two practicality questionnaires: one for students (14 items) and one for the four KKA teachers (10 items). The researchers conducted statistical analysis in April-June 2026 (Evaluation phase).

Data Collection Techniques and Research Instruments

Four instruments were employed. The Interview Guide for Cognitive Barrier Analysis was administered to the five teacher participants during the Analysis phase. The Expert Validation Rubric was administered to the three validators at the close of the Design phase. Instrument grids are presented in Table 2 and Table 3.

Table 2. Interview Guide Grid: Cognitive Barrier Analysis Instrument

Domain	Indicator	Item No.
Most frequent learner error types by CP element	Syntax and logic error patterns in BK modules (Variables, I/O, Arithmetic)	1-2
Most frequent learner error types by CP element	Error patterns in AP modules (Conditionals, Loops, Functions, OOP)	3-5
Most frequent learner error types by CP element	AI dependency patterns in KA modules (Prompt Engineering)	6
Learner engagement and task completion	Curriculum points where completion rate or engagement declines significantly	7-8
Adaptive instructional strategies	Teacher workarounds and scaffolding strategies developed in response to identified difficulties	9-10

Table 3. Expert Validation Rubric Grid: Module Design Assessment Instrument

Validation Domain (Validator)	Validation Aspect	No. Items	Max Score
Subject Matter Content - Validator 1 (MAR)	Python content accuracy and KKA CP alignment	2	8
Subject Matter Content - Validator 1 (MAR)	Typicality and cognitive representativeness of intentional errors per module level	2	8
Subject Matter Content - Validator 1 (MAR)	Accuracy of Q1-Q3 worksheet prompts relative to Python programming conventions	2	8
Instructional Design - Validator 2 (MAU)	Scaffolding level alignment with CLT and ZPD theoretical framework	2	8
Instructional Design - Validator 2 (MAU)	EAR task format coherence: Q1-Q2-Q3 progression through scaffolding levels	2	8
Instructional Design - Validator 2 (MAU)	ZPD calibration: appropriateness of scaffold level assignment to each CP element	2	8
Educational Media/Interface - Validator 3 (SAS)	Vilokka interface usability for EAR exercise workflow	2	8
Educational Media/Interface - Validator 3 (SAS)	Clarity and legibility of three-section Local-LLM response layout	2	8
Educational Media/Interface - Validator 3 (SAS)	Navigation and module sequencing design	1	4
Total		17	68

The Python programming competency test (pretest and posttest parallel forms) was developed based on the three KKA CP elements (BK, AP, KA) and administered as a 40-item multiple-choice instrument. The student practicality questionnaire comprised 14 items (A1–A14) on a four-point Likert scale, covering platform accessibility, content and instruction clarity, LLM feedback quality, and motivational impact. The teacher practicality questionnaire comprised 10 items across four assessment aspects on a four-point Likert scale (1 = Poor, 4 = Excellent).

Instrument Validity and Reliability Tests

The Interview Guide was subjected to face validity assessment by a KKA curriculum specialist not participating in the study. Following this review, two interview items were revised. The Expert Validation Rubric was grounded in content validity with reference to Akbar (2013) and established scaffolding assessment frameworks. For qualitative interview data, member checking strengthened analytical reliability. For the competency test, we computed item difficulty and discrimination indices following pilot administration with ten students from a non-participant partner school; items with item difficulty $p < 0.20$ or $p > 0.80$ and discrimination index < 0.25 were revised or replaced. No inter-rater reliability procedure was applied to the Expert Validation Rubric, as each domain was assessed by a single specialist by design, a standard approach in Indonesian educational product validation following the Akbar (2013) framework.

Data Analysis Techniques and Criteria

Two primary data analysis approaches were employed for the first three phases. For the Analysis phase, inductive thematic analysis was applied to interview transcripts (Braun & Clarke, 2006). For the Expert Validation phase, the Akbar (2013) validity formula was applied in Equation 1:

$$V(\%) = (\Sigma S / \Sigma S_{max}) \times 100\% \quad \text{..... (1)}$$

where $V(\%)$ is the percentage of validity, ΣS is the sum of scores obtained, and ΣS_{max} is the maximum possible score. Table 4 presents the validity criteria.

Table 4. Validity Criteria Based on Akbar (2013)

Validity Percentage	Category	Interpretation
85.01% - 100%	Very Valid	Module design is ready for the Development phase
70.01% - 85.00%	Valid	Minor revisions required before the Development phase
50.01% - 70.00%	Less Valid	Major revisions required; re-validation recommended
0% - 50.00%	Not Valid	Design must be fundamentally reconceptualised

For the Evaluation phase, four analytical procedures were applied. Normality of pretest, posttest, and *N-Gain* distributions was tested using the Shapiro-Wilk procedure (Royston, 1992) at $a = 0.05$. Learning effectiveness was quantified through *N-Gain* analysis in Equation 2 (Hake, 1998):

$$g = (\text{posttest} - \text{pretest}) / (100 - \text{pretest}) \quad \text{..... (2)}$$

with categories High ($g \geq 0.70$), Moderate ($0.30 \leq g < 0.70$), and Low ($g < 0.30$). Hypothesis testing employed a paired sample t-test with effect size measured by Cohen's d. Practicality was assessed using the $V(\%)$ formula Akbar (2013) applied separately to teacher and student questionnaire data. All statistical analyses were performed using IBM SPSS Statistics version 25.

Product Specifications

The product developed in this study is a cognitive scaffolding KKA module integrated with the Vilokka on-premise virtual laboratory. Product specifications: (1) Module count: 17 modules spanning the full KKA curriculum: 3 BK modules (Sense-making, Modules 1-3), 12 AP modules (Process-scaffolding, Modules 4-15), and 2 KA modules (Fading, Modules 16-17). (2) Content per module: four intentional Python errors per EAR exercise, accompanied by a three-question structured worksheet. (3) LLM platform: Local-LLM deployed on-premises within the Vilokka infrastructure. (4) System prompt: single static system prompt governing all 17 modules; specifies three mandatory response sections and an absolute prohibition on code block delivery. (5) Scaffolding differentiation: achieved through the cognitive complexity of selected error content. (6) Validity threshold: minimum 85.01% (Very Valid) per Akbar (2013).

Results and Discussion

Cognitive Barrier Profile

The Analysis phase encompassed two components. First, structured document analysis of the KKA CP text (BSKAP No. 046/H/KR/2025) identified the three CP element clusters (Basic Algorithms/BK, Programming Algorithms/AP, and Artificial Intelligence/KA), 17 Python topics, and the performance indicators that informed module scope. Second, semi-structured interviews with five KKA teachers at three partner VHS, analysed using inductive thematic analysis Braun &

Clarke (2006), yielded three cognitive barrier categories, one per CP cluster, that served as the primary design input for the module architecture. Table 5 presents these findings.

Table 5. Cognitive Barrier Profile by KKA Curriculum Element

CP Element	Predominant Error Patterns	Cognitive Barrier Category
Basic Algorithms (BK) Modules 1-3	Variable undeclared before use; indentation misconstruction; invalid value assignment to literals	Pre-coding abstraction failure: learners cannot form a mental model of program state before translating intention into syntax
Programming Algorithms (AP) Modules 4-15	Logic-syntax conflation in conditionals and loops; incorrect function scoping; misapplied data structure operations	Logic-syntax conflation: learners confuse algorithmic intent with syntactic expression, producing structurally plausible but semantically erroneous code
Artificial Intelligence (KA) Modules 16-17	AI-generated code accepted uncritically; superficial error correction without conceptual understanding	AI-induced cognitive passivity: LLM access without guardrails creates solution-dependency that bypasses generative reasoning required for schema formation

Three cognitive barrier categories were identified, each associated with a specific KKA curriculum element and scaffolding level. At the BK level, pre-coding abstraction failure was most prevalent: teachers reported that learners attempted to write Python syntax before forming any mental model of programme intent, manifesting in undeclared variables and indentation errors, consistent with Huang et al. (2025) characterisation of early learners as syntax-first thinkers. At the AP level, logic-syntax conflation was documented: learners conflated algorithmic intent with syntactic expression, producing structurally plausible but semantically erroneous code in conditionals, loops, and function scoping. At the KA level, AI-induced cognitive passivity was observed, consistent with Mallik and Gangopadhyay's (2023) and L. Yan et al. (2024) evidence that ungoverned LLM access correlates negatively with metacognitive monitoring. The three categories constitute a developmental sequence that the module architecture must address progressively and cumulatively. These three categories formed the empirical foundation for the Design phase: the scaffolding-level assignment (Sense-making for BK; Process-scaffolding for AP; Fading for KA), the EAR task format, and the Vilokka Local-LLM guardrailing configuration were each derived directly from a corresponding barrier category.

Module Architecture

The Design phase produced a 17-module architecture synchronising the KKA CP structure with the three-level scaffolding framework and Vilokka's implementation capabilities. Table 6 presents the full architecture.

Table 6. Cognitive Scaffolding Module Architecture Synchronised with KKA Curriculum and Vilokka

CP Element	Scaffolding Level	Modules	Python Topic	Vilokka Implementation
Basic Algorithms (BK)	Level 1 Sense-making	1-3	Variables & data types; basic I/O; arithmetic operators	LLM delivers Error Category and Location + conceptual analogy (no code); EAR Q1 targets why the error occurred conceptually

CP Element	Scaffolding Level	Modules	Python Topic	Vilokka Implementation
Programming Algorithms (AP)	Level 2 Process-scaffolding	4-15	Conditionals; iteration; functions; lists/tuples/dicts; file I/O; modules; exception handling; OOP fundamentals	LLM Prompting Question targets procedural analysis; Q2 asks how to resolve the error step-by-step
Artificial Intelligence (KA)	Level 3 Fading	16-17	Advanced OOP (classes, generators); prompt engineering for vocational AI tasks	EAR tasks: Prompting Question demands design-rationale reasoning; prompt engineering tasks: LLM restricted to output-validation only

The scaffolding fading trajectory was calibrated through structural fading: the prescriptive depth of Vilokka's Prompting Question decreases progressively through the cognitive complexity of selected error content rather than through dynamic prompt reconfiguration. At Level 1, errors are syntactically isolated and concrete; at Level 2, errors embed logical ambiguity requiring multi-step diagnosis; at Level 3, errors are nested within architecturally complex OOP structures. This content-driven fading produces a de facto reduction in LLM support without changing the static system prompt, consistent with J. Yan et al. (2025) demonstrating that meaningful pedagogical differentiation can emerge from error complexity within a single prompt configuration.

Error-Analysis-Reflection (EAR) Task Format

The EAR task format is the instructional core of each module. Every EAR exercise presents a Python programme containing exactly four intentional errors characteristic of that module's scaffold level. After interacting with Vilokka, students complete a three-question worksheet: Q1 asks why the error occurred conceptually (Sense-making); Q2 asks how to resolve it procedurally (Process-scaffolding); Q3 asks what design decision would prevent the error class entirely (Fading). A representative EAR exemplar from Module 4 (Conditionals, AP Level 2) illustrates the format: the exercise presents a programme with four intentional errors (missing colon, incorrect == operator, capitalised AND, and misplaced print block). Vilokka's guardrailed response follows the mandatory three-section structure: Error Category and Location identify the SyntaxError on line 3; Theoretical Explanation provides a language-based account of Python's block structure without delivering code; and the Prompting Question asks 'In your view, what would happen to the programme's flow if Python did not know where that block of instructions begins?'

The rationale for four intentional errors per exercise reflects two design constraints. First, four errors provide sufficient breadth to expose learners to multiple instantiations of a module's target error type within a single contact hour. Second, four errors support the three-question worksheet structure with a metacognitive sorting task embedded in every EAR exercise at all scaffolding levels.

LLM Guardrailing and the Static System Prompt

Vilokka's scaffolding differentiation across 17 modules is achieved without a dynamic or level-specific system prompt. A single static system prompt governs all LLM interactions, defined by: (1) an absolute prohibition on code block delivery; (2) a mandatory three-section response format; (3) a persona as VHS programming teacher; and (4) the respectful Indonesian address '*Nak*' in the Theoretical Explanation. This architecture aligns with J. Yan et al. (2025) finding that system-prompt engineering reliably constrains LLM behaviour without dynamic prompt construction.

Preliminary system-prompt testing with two Local-LLM variants prior to the Design phase confirmed that dual specification (what the LLM must produce and must not produce) achieved consistent guardrail compliance across diverse error types and complexity levels.

Expert Validation Results

Expert validation results were computed using the *V(%)* formula (Akbar, 2013) and analysed using SPSS Descriptive Statistics. Table 7 presents item-level *V*(%) scores across all 17 validation items, and Table 8 summarises the SPSS output per domain.

Table 7. Item-Level *V*(%) Scores by Domain

No	Validation Aspect	Items	*V(%)*	Domain *V(%)*
1	Python content accuracy & KKA CP alignment	1	75	
		2	75	
2	Typicality of intentional errors per level	3	100	
		4	100	
3	Q1–Q3 prompt accuracy	5	75	
		6	100	87.50
4	Scaffolding level alignment with CLT and ZPD	7	100	
		8	75	
5	EAR task format coherence: Q1–Q2–Q3 progression	9	100	
		10	75	
6	ZPD calibration: scaffold level assignment to each CP element	11	100	
		12	100	91.67
7	Vilokka interface usability for EAR workflow	13	75	
		14	100	
8	Clarity & legibility of LLM three-section response layout	15	75	
		16	100	
9	Navigation and module sequencing design	17	100	90.00

Table 8. SPSS Descriptive Statistics of *V(%)* per Validation Domain

Validation Domain	Validator	*N*	*Min*	*Max*	*Sum*	*V (%)*	Category
Subject-matter Content	MAR	6	75	100	525	87.50	Very Valid
Instructional Design	MAU	6	75	100	550	91.67	Very Valid
Media and Interface	SAS	5	75	100	450	90.00	Very Valid
Overall Mean	-	17	75	100	1525	89.72	Very Valid

Note: V(%) per domain = Mean of item-level V(%) scores. Overall from SPSS Descriptive Statistics (N=17). Validity category based on Akbar (2013): Very Valid ≥ 85.01%.

The overall mean validity of 89.72% confirms that the module design is structurally sound across all three evaluated domains. Subject-matter validation (87.50%) reflected the validator's recommendation for revisions in three areas: theoretical justification for generator syntax in Module 17, precision of tuple immutability explanation in Module 12, and Q3 in Module 10, which should more clearly target metacognitive reasoning. Instructional design validation achieved the highest score (91.67%), indicating strong alignment between the EAR task format and the scaffolding framework's theoretical basis. Media validation (90.00%) confirmed Vilokka's interface usability. These results compare favourably with related studies: Zuo et al. (2023) meta-analysis confirmed the consistent advantage of scaffolded online learning, and Zeitlhofer et al. (2023)

cognitive prompts research provides direct empirical support for the Prompting Question mechanism. The validated module design thus served as the direct specification for the Development phase: each module's four intentional Python errors, EAR worksheet questions, and Vilokka system prompt configuration were encoded in strict accordance with the validated architecture.

Development Phase

The Development phase (January–February 2026) successfully digitalised all 17 cognitive scaffolding modules within the Vilokka platform prior to student deployment. Each module's EAR exercise was encoded with four intentional Python errors in the learner-facing code editor interface; the three-question worksheet was integrated as a structured digital submission form; and the static system prompt was deployed to the Vilokka Local-LLM. Internal testing, completed in February 2026 before the Implementation phase commenced, confirmed consistent three-section response output and guardrail compliance across all 17 module types and error categories. Internal testing required no content revisions and produced a complete, deployment-ready module set within the Vilokka platform. Completion of Development thus cleared all technical prerequisites for student deployment, and the full 17-module set was made available on the Vilokka platform before the Implementation phase commenced in February 2026.

Implementation and Evaluation Results

Normality Testing

The Implementation phase commenced in February 2026, following completion of the Development phase. Prior to hypothesis testing, Shapiro-Wilk tests were conducted on the pretest, posttest, and N-Gain distributions (n = 30; a = 0.05). Results are presented in Table 9.

Table 9. Shapiro-Wilk Normality Test Results (n = 30)

	Statistic	*df*	*Sig.*
Pretest	0.950	30	0.174
Posttest	0.943	30	0.108
N-Gain	0.940	30	0.089

All three variables satisfied the normality assumption (p > 0.05 for all variables), confirming the appropriateness of the paired sample t-test for hypothesis testing.

Learning Effectiveness: N-Gain Analysis

The mean pretest score was 50.67 (SD = 15.07), and the mean posttest score was 72.33 (*SD* = 13.05), representing a gain of 21.66 points. The mean *N-Gain* was $\bar{g}$ = 0.39 (*SD* = 0.32), which falls within the Moderate category (0.30 ≤ g < 0.70) according to Hake's (1998) criteria. Distribution of individual *N-Gain* categories is presented in Table 10.

Table 10. Distribution of N-Gain Categories (n = 30)

Category	Range	*n*	Percentage (%)
High	$g \geq 0.70$	6	20.0
Moderate	$0.30 \leq g < 0.70$	17	56.7
Low	$g < 0.30$	7	23.3
Total	-	30	100.0

Of the 30 participants, 6 (20.0%) achieved the High category ($g \geq 0.70$), 17 (56.7%) fell within the Moderate category, and 7 (23.3%) remained in the Low category ($g < 0.30$). The predominance of

Moderate-category learners is consistent with the intermediate cognitive demands of a first implementation: learners traversed all three scaffolding levels, but the absence of a student activity log in Vilokka v0.1 prevented teachers from identifying and providing real-time intervention for struggling learners, the primary constraint contributing to the 23.3% Low-category proportion. The Moderate N-Gain is consistent with Zuo et al. (2023) meta-analytic findings that first-implementation scaffolded online environments produce moderate learning gains relative to unscaffolded conditions, and it shows that the EAR task format successfully operationalised productive struggle without reducing challenge to zero.

Hypothesis Testing: Paired Sample t-Test

The paired sample *t-test* yielded $t(29) = 6.385$, $p < 0.05$, indicating a statistically significant difference between pretest and posttest scores. H_0 (no significant improvement) was rejected; H_1 (application of the Vilokka-integrated module produces significant improvement in KKA learning outcomes) was accepted. *Cohen's d* = 1.17 indicates a large effect size ($d \geq 0.80$), confirming that the observed improvement is not only statistically significant but educationally meaningful. Table 11 presents the paired sample t-test results.

Table 11. Paired Sample t-Test Results

		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		*t*	*df*	*Sig.* (2-tailed)
					Lower	Upper			
Pair 1	Posttest - Pretest	21.66667	18.58500	3.39314	14.72691	28.60642	6.385	29	0.000

Practicality Assessment: Teacher Evaluation

Four KKA teachers evaluated the module across 10 items covering four assessment aspects. Results are presented in Table 12.

Table 12. Teacher Practicality Assessment Results (n = 4 raters; 10 items)

Aspect	Item	ΣS	S_{max}	V (%)	Category
Aspect 1	Item 1	15	16	93.75	Very Practical
Aspect 1	Item 2	15	16	93.75	Very Practical
Aspect 1	Item 3	15	16	93.75	Very Practical
Aspect 2	Item 4	14	16	87.50	Very Practical
Aspect 2	Item 5	13	16	81.25	Practical
Aspect 2	Item 6	13	16	81.25	Practical
Aspect 3	Item 7	13	16	81.25	Practical
Aspect 3	Item 8	13	16	81.25	Practical
Aspect 3	Item 9	15	16	93.75	Very Practical
Aspect 4	Item 10	15	16	93.75	Very Practical
Overall	10 Items	141	160	88.12	Very Practical

Overall teacher $V(\%)$ = 88.12%, placing the module in the Very Practical category (≥ 85.01%) by Akbar's (2013) criteria. $V(\%)$ per item ranged from 81.25% (Practical) to 93.75% (Very Practical), with no item below the Practical threshold. Items achieving V = 93.75% corresponded to aspects of platform accessibility, LLM response section clarity, and session-time appropriateness, dimensions directly related to the static system prompt and on-premise infrastructure design. Items scoring 81.25% were predominantly related to implementation flexibility and instructional pacing,

suggesting that refined time-allocation guidance and instructor onboarding materials would strengthen these dimensions in subsequent versions.

Practicality Assessment: Student Evaluation

Thirty students evaluated the module across 14 items (A1–A14). Results are presented in Table 13.

Table 13. Student Practicality Assessment Results (n = 30)

Item	ΣS	S_{max}	V (%)	Category
A1	108	120	90.00	Very Practical
A2	102	120	85.00	Practical
A3	106	120	88.33	Very Practical
A4	104	120	86.67	Very Practical
A5	107	120	89.17	Very Practical
A6	102	120	85.00	Practical
A7	103	120	85.83	Very Practical
A8	103	120	85.83	Very Practical
A9	107	120	89.17	Very Practical
A10	104	120	86.67	Very Practical
A11	102	120	85.00	Practical
A12	104	120	86.67	Very Practical
A13	105	120	87.50	Very Practical
A14	104	120	86.67	Very Practical
Overall	1461	1680	86.96	Very Practical

Overall student $V(\%)$ = 86.96% (Very Practical). $V(\%)$ per item ranged from 85.00% to 90.00%, demonstrating high consistency across all 14 questionnaire dimensions. Item A1 (platform accessibility; V = 90.00%) confirmed that Vilokka v0.1's on-premise infrastructure operates reliably within the partner school's existing hardware environment, a critical validation for deployment in infrastructure-constrained VHS. Items A2, A6, and A11 scored at the minimum V = 85.00%, each within the Practical threshold, identifying specific refinement opportunities for Vilokka v0.2. The Very Practical rating from both teachers and students aligns with Ross et al. (2025) and Zhang's (2025) evidence that structured LLM-augmented environments designed for pedagogical constraints are well received in vocational education contexts.

Discussion

The mean N-Gain of 0.39, classified as Moderate by Hake's (1998) criteria, is consistent with the broader scaffolding literature. Zuo et al. (2023), in a meta-analysis of scaffolding interventions in online learning, reported a pooled effect of g = 0.58, and noted that moderate gains are characteristic of first-cycle implementations where fading mechanisms are not yet fully adaptive. The large effect size (*Cohen's d* = 1.17) further corroborates findings from Hartley et al. (2024), who reported large effects when AI-assisted tools were designed to support independent student learning through structured guidance rather than direct answer provision. Comparably, Huang et al. (2025) found that GenAI-generated coding hints, when constrained to the problem at hand, produced significant pretest-to-posttest gains in student coding performance, a result attributable to the same guardrailing principle operationalised in Vilokka v0.1.

The EAR task format and LLM guardrailing architecture extend two converging lines of prior research. Sunil and Thakkar (2025) demonstrated that structuring LLM interactions around Socratic error-correction sequences, where students diagnose, explain, and reflect rather than receive answers, produced measurable gains in conceptual understanding among introductory CS learners. The EAR format operationalises an analogous sequence (Error identification, root-cause analysis, and reflection on cognitive process) adapted to the three KKA barrier categories. Simultaneously, Ross et al. (2025) showed that supervised fine-tuning of LLMs with pedagogically constrained response templates reduces off-task AI use by students, a risk explicitly addressed by the static system prompt in Vilokka. Zeitlhofer et al. (2023) additionally found that the effectiveness of cognitive scaffolding is moderated by the specificity of scaffolding prompts, a finding that validates the per-module system prompt customisation adopted in the present design.

At the platform level, Vilokka extends the virtual laboratory paradigm reviewed by Mercado and Picardal (2023) by introducing on-premise LLM integration as a scaffolding layer, a feature absent from the biotechnology and science simulators surveyed in that review. The on-premise deployment addresses the data-privacy constraint Zhang (2025) identifies as a structural barrier to AI adoption in vocational education institutions, particularly in contexts where cloud-based AI services are constrained by institutional bandwidth and policy. The practicality ratings (teacher V = 88.12%; student V = 86.96%, both Very Practical) align with L. Yan et al. (2024), who found that learner-facing AI tools with practicality ratings above 85% were associated with sustained voluntary engagement beyond mandated sessions.

Three aspects of novelty distinguish the present work from prior studies. First, the cognitive barrier taxonomy, comprising pre-coding abstraction failure, logic-syntax conflation, and AI-induced cognitive passivity, constitutes the first empirically derived, curriculum-mapped barrier profile for the Indonesian KKA subject, providing a domain-specific diagnostic instrument absent from the existing literature. Second, integrating a guardrailed on-premise Local-LLM within a purpose-built virtual laboratory offers a novel architectural response to the data-privacy and infrastructure constraints of Indonesian VHS, distinguishing Vilokka from cloud-dependent AI tutoring systems reviewed by Xu and Ouyang (2022). Third, the EAR task format constitutes a new instructional unit design for combined coding-and-AI curricula, extending the error-analysis approach beyond single-language programming contexts to the dual-domain knowledge structure of KKA.

Several limitations constrain the scope of the present findings. The sample (n = 30, single institution, purposive) limits statistical generalisation; replication across multiple VHS sites with diverse socioeconomic profiles is required before broad applicability can be claimed. The pre-experimental design (one-group pretest-posttest) precludes causal attribution of observed gains to the module alone, as it used no control condition. Vilokka v0.1 also lacks a student activity log, preventing real-time teacher monitoring of learner progress and limiting the scaffolding system's adaptive capacity during live sessions. Finally, subject-matter experts validated the LLM's static system prompt, but the study did not empirically calibrate it against actual student–LLM interaction transcripts, leaving the guardrailing efficacy partially unverified in ecological conditions.

Despite these limitations, the practical and policy implications of the present work are substantive. The validated module set provides VHS KKA teachers with a ready-to-deploy, curriculum-aligned resource addressing a documented gap between CP performance indicator requirements and available instructional materials (BSKAP No. 046/H/KR/2025). The Vilokka platform demonstrates the technical feasibility of on-premise AI-guardrailed virtual laboratories for vocational contexts, a proof-of-concept with direct relevance to the broader OECD (2023) agenda for quality, equity, and efficiency in digital education. The cognitive barrier taxonomy and EAR

task format are transferable to other programming and AI literacy subjects, offering curriculum designers a replicable framework for scaffolded error-based instruction beyond the KKA context.

Conclusion

This study applied the ADDIE model to design, develop, and evaluate a cognitive scaffolding module for the Indonesian KKA subject, integrated with the Vilokka on-premise virtual laboratory. In response to RQ1, systematic curriculum analysis and teacher interviews identified three cognitive barrier categories, such as pre-coding abstraction failure, logic-syntax conflation, and AI-induced cognitive passivity, which each mapped to a specific KKA CP element. In response to RQ2, the resulting module architecture, comprising 17 scaffolding modules structured around the Error-Analysis-Reflection task format and a guardrailed Local-LLM, achieved expert validation at the Very Valid level across all three evaluated domains. In response to RQ3, field implementation demonstrated that the module produced a statistically significant improvement in student learning outcomes at a large effect size and was rated Very Practical by both teacher and student evaluators. These results affirm the viability of curriculum-aligned, on-premise AI-guardrailed scaffolding as a response to the documented cognitive barriers in KKA instruction. Future work will prioritise student activity log integration and adaptive fading mechanisms as the core development agenda for Vilokka v0.2.

Acknowledgements

This research was supported by the PPKR Research Grant, Universitas Pendidikan Ganesha (Contract No. 578/UN48.16/PT.01.03/2026). The Analysis and Design phases were conducted as preliminary activities during the grant proposal preparation period in late 2025, consistent with standard research practice at Universitas Pendidikan Ganesha; the formal research contract supported the Expert Validation, Development, Implementation, and Evaluation phases. The authors thank the VHS partner teachers, the expert validators, and the student participants who contributed their time and expertise to this study.

Bibliography

Akbar, S. (2013). *Instrumen perangkat pembelajaran* (Instruments for learning materials). PT. Remaja Rosdakarya.

Branch, R. M. (2010). Instructional design: The ADDIE approach. In *Instructional Design: The ADDIE Approach*. Springer US. https://doi.org/10.1007/978-0-387-09506-6

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Christi, S. R. N., & Rajiman, W. (2023). Pentingnya berpikir komputasional dalam pembelajaran matematika (The importance of computational thinking in mathematics learning). *Journal on Education*, *5*(4), 12590–12598. https://doi.org/10.31004/joe.v5i4.2246

Etikan, I. (2016). Comparison of convenience sampling and purposive sampling. *American Journal of Theoretical and Applied Statistics*, *5*(1), 1. https://doi.org/10.11648/j.ajtas.20160501.11

Hake, R. R. (1998). Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics*, *66*(1), 64–74. https://doi.org/10.1119/1.18809

Hartley, K., Hayak, M., & Ko, U. H. (2024). Artificial intelligence supporting independent student learning: An evaluative case study of ChatGPT and learning to code. *Education Sciences*, *14*(2), 120. https://doi.org/10.3390/educsci14020120

Huang, A. Y. Q., Lin, C. Y., Su, S. Y., & Yang, S. J. H. (2025). The impact of GenAI-enabled coding hints on students' programming performance and cognitive load in an SRL-based Python course. *British Journal of Educational Technology*, *56*(5), 1942–1972. https://doi.org/10.1111/bjet.13589

I Made Elia Cahaya, I. M. E., Suryaningsih, N. M. A., Parwata, I. M. Y., Poerwati, C. E. (2024). The influence of a guided inquiry learning model in improving students' creative thinking abilities: A meta-analysis study. (2024). *Indonesian Journal of Educational Development (IJED)*, *5*(3), 376–384. https://doi.org/10.59672/ijed.v5i3.4211

Kaenong, H. A., Alexandri, M. B., & Sugandi, Y. S. (2023). Analysis projection of the fulfillment of priority facilities and infrastructures for vocational high school using system dynamic to increase school participation rates in central kalimantan province, indonesia. *Sustainability (Switzerland)*, *15*(24), 16696. https://doi.org/10.3390/su152416696

Kemendikdasmen. (2025). *Keputusan kepala badan standar, kurikulum, dan asesmen pendidikan kementerian pendidikan dasar dan menengah nomor 046/H/KR/2025 tentang capaian pembelajaran* (Decree of the head of the standards, curriculum, and educational assessment board of the ministry of basic and secondary education number 046/H/KR/2025 on learning outcomes). https://guru.kemendikdasmen.go.id/dokumen/74r6Yln0zK

Mallik, S., & Gangopadhyay, A. (2023). Proactive and reactive engagement of artificial intelligence methods for education: A review. *Frontiers in Artificial Intelligence*, *6*. https://doi.org/10.3389/frai.2023.1151391

Mercado, J. C., & Picardal, J. P. (2023). Virtual laboratory simulations in biotechnology: A systematic review. *Science Education International*, *34*(1), 52–57. https://doi.org/10.33828/sei.v34.i1.6

OECD. (2023). *Shaping digital education: Enabling factors for quality, equity and efficiency*. OECD Publishing. https://doi.org/10.1787/bac4dc9f-en

Ross, E., Kansal, Y., Renzella, J., Vassar, A., & Taylor, A. (2025). *Supervised fine-tuning LLMs to behave as pedagogical agents in programming education*. http://arxiv.org/abs/2502.20527

Royston, P. (1992). Approximating the Shapiro-Wilk W-test for non-normality. *Statistics and Computing*, *2*(3), 117–119. https://doi.org/10.1007/BF01891203

Santyadiputra, G. S., & Kustono, D. (2023). An analysis of cybersecurity subject and Vilanets learning media towards vocational school students' digital skills. *Letters in Information Technology Education (LITE)*, *6*(1), 16–21. https://doi.org/10.17977/UM010V6I12023P16-21

Santyadiputra, G. S., Purnomo, Kamdi, W., Patmanthara, S., & Nurhadi, D. (2024). Vilanets: An advanced virtual learning environments to improve higher education students' learning achievement in computer network course. *Cogent Education*, *11*(1). https://doi.org/10.1080/2331186X.2024.2393530

Santyadiputra, G. S., Purnomo, P., Kamdi, W., & Patmanthara, S. (2025). *Pengaruh virtual project based learning dan direct instruction terintegrasi SAMR dan TPACK terhadap prestasi belajar dan literasi teknologi peserta didik sekolah menengah kejuruan* (The effect of virtual project based learning and direct instruction integrated with SAMR and TPACK on learning achievement and technological literacy of vocational high school students) [Doctoral dissertation]. Universitas Negeri Malang.

Sunil, K., & Thakkar, A. (2025). *SocraticAI: Transforming LLMs into guided CS tutors through scaffolded interaction*. http://arxiv.org/abs/2512.03501

Surya Abadi, I.B.G., Widiana, I. W., Septiari, N. K., Putu Listyana, I.G.A.A., Ari Rahayu, N. K. (2025). The impact of metacognitive-based learning strategies on enhancing students' decision-making and cognitive dissonance. (2025). *Indonesian Journal of Educational Development (IJED)*, *6*(1), 241–253. https://doi.org/10.59672/ijed.v6i1.4805

Widana, I. W. & Ratnaya, I. G. (2021). Relationship between divergent thinking and digital literacy on teacher ability to develop HOTS assessment. *Journal of Educational Research and Evaluation, 5*(4), 516-524. https://doi.org/10.23887/jere.v5i4.35128

Xu, W., & Ouyang, F. (2022). The application of AI technologies in STEM education: A systematic review from 2011 to 2021. *International Journal of STEM Education*, *9*(1), 59. https://doi.org/10.1186/s40594-022-00377-5

Yan, J., Tian, H., Sun, X., & Song, L. (2025). Role of artificial intelligence in enhancing competency assessment and transforming curriculum in higher vocational education. *Frontiers in Education*, *10*. https://doi.org/10.3389/feduc.2025.1551596

Yan, L., Sha, L., Zhao, L., Li, Y., Martinez-Maldonado, R., Chen, G., Li, X., Jin, Y., & Gašević, D. (2024). Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology*, *55*(1), 90–112. https://doi.org/10.1111/bjet.13370

Zeitlhofer, I., Hörmann, S., Mann, B., Hallinger, K., & Zumbach, J. (2023). Effects of cognitive and metacognitive prompts on learning performance in digital learning environments. *Knowledge*, *3*(2), 277–292. https://doi.org/10.3390/knowledge3020019

Zhang, Y. (2025). AI-driven transformation of vocational education. *International Journal of Knowledge Management*, *21*(1), 1–20. https://doi.org/10.4018/IJKM.394819

Zuo, M., Kong, S., Ma, Y., Hu, Y., & Xiao, M. (2023). The effects of using scaffolding in online learning: A meta-analysis. *Education Sciences*, *13*(7), 705. https://doi.org/10.3390/educsci13070705