ISSN 2085-4552

Design and Evaluation of an AI-Driven Gamified Intelligent Tutoring System for Fundamental Programming Using the Octalysis Framework

Dzaky Fatur Rahman[1], Fenina Adline Twince Tobing[2], Cian Ramadhona Hassolthine[3]
[1,2] Informatics Department, Universitas Multimedia Nusantara, Tangerang, Indonesia
[3]PJJ Informatika, Universitas Siber Asia, Jakarta, Indonesia
email: dzaky.fatur@student.umn.ac.id [1], fenina.tobing@umn.ac.id[2], cianhassolthine@lecturer.unsia.ac.id[3]

Accepted 04 January 2026
Approved 21 January 2026

***Abstract*— Gamification and Intelligent Tutoring Systems (ITS) are independently proven to enhance student engagement, yet their integration remains fragmented in fundamental programming education. Existing systems often feature either robust gamification with static content or adaptive AI with limited motivational design, leaving a gap in comprehensive frameworks that address both intrinsic motivation and cognitive support simultaneously. This study designs and evaluates "Starcoder," an adaptive learning environment that synthesizes the Octalysis Framework with a Generative AI-driven ITS. The system architecture integrates the Next.js framework with the Gemini AI API to deliver real-time, context-aware feedback and remedial learning paths. The study employs a Research and Development (R&D) methodology, culminating in a quantitative evaluation using the Hedonic-Motivation System Adoption Model (HMSAM). Data from 54 undergraduate respondents were analyzed using descriptive statistics to measure seven core constructs of user experience. The results indicate a positive reception of the integrated approach, with the platform achieving high approval ratings in Perceived Usefulness (86.44%) and Curiosity (85.56%). Comparative analysis reveals a notable increase in Behavioral Intention to Use (+15.56%) relative to the traditional classroom baseline. These findings suggest that coupling narrative-driven gamification with adaptive AI agents effectively fosters student engagement, offering a promising model for next-generation educational technologies.**

Index Terms*— Artificial Intelligence (AI); Gamification; Hedonic-Motivation System Adoption Model (HMSAM); Intelligent Tutoring System (ITS); Octalysis Framework.

I. INTRODUCTION

Fundamental programming education presents significant pedagogical challenges, particularly in the initial years of informatics study. The curriculum demands a grasp of abstract concepts such as algorithms, logic, and data structures, which are often difficult for novice learners to visualize [1]. Traditional instructional methods, characterized by linear delivery and rigid classroom structures, frequently struggle to foster the intrinsic motivation and engagement necessary for mastering these complex skills [2]. Consequently, students often experience demotivation and a lack of active participation, leading to suboptimal learning outcomes [3].

To address these engagement deficits, gamification has emerged as a powerful pedagogical strategy. Recent meta-analyses indicate that integrating game mechanics into educational settings significantly enhances student motivation and behavioral engagement [4]. Among various gamification models, the Octalysis Framework has proven particularly effective in educational contexts by leveraging eight core drives of human motivation to create immersive learning experiences [5]. However, while gamification improves engagement, it does not inherently address the need for personalized cognitive support. This is where Intelligent Tutoring Systems (ITS) become critical. AI-driven ITS can provide adaptive feedback and tailored learning paths, which have been shown to consistently improve academic achievement by addressing individual learning gaps [6].

Despite the proven efficacy of both gamification and ITS independently, the seamless integration of these two paradigms remains a significant research gap. A systematic review of recent literature suggests that while "personalized gamification" is a crucial technological approach, comprehensive frameworks combining motivational theory (like Octalysis) with adaptive AI systems are scarce [7]. Existing implementations often suffer from a dichotomy: they either feature advanced AI with superficial gamification or robust gamification with static, non-adaptive content. Recent studies confirm that current gamified systems frequently fail to personalize motivational cues dynamically, limiting their long-term effectiveness [8].

This research proposes "Starcoder," a novel learning platform that bridges this gap by integrating the Octalysis Framework with a generative AI-based ITS. The system was developed using the Next.js framework to ensure high performance and scalability [9], incorporating the Gemini AI API to function as an adaptive mentor named "M.E.C.H.A." The primary objective is to design a cohesive environment where the eight core drives of Octalysis, such as Epic Meaning and Scarcity, are supported by real-time, context-aware AI feedback. The study employs a Research and Development (R&D) methodology, culminating in a comparative evaluation using the Hedonic-Motivation System Adoption Model (HMSAM) with 54 respondents to measure improvements in engagement and perceived usefulness against traditional learning methods.

A. Research Gap

Despite the proven efficacy of both gamification and ITS independently, the seamless integration of these two paradigms remains a significant research gap. A systematic review of recent literature suggests that while "personalized gamification" is a crucial technological approach, comprehensive frameworks combining motivational theory (like Octalysis) with adaptive AI systems are scarce [7]. Existing implementations often suffer from a dichotomy: they either feature advanced AI with superficial gamification (points and badges only) or robust gamification with static, non-adaptive content [19]. Recent studies confirm that current gamified systems frequently fail to personalize motivational cues dynamically, limiting their long-term effectiveness [8]. There is a critical lack of theoretical models and tested implementations that systematically combine the eight core drives of Octalysis with Generative AI-based ITS architectures.

B. Research Objectives

To bridge this gap, this study aims to achieve the following objectives:

1. To design and build "Starcoder," a web-based learning platform that systematically integrates the Octalysis Gamification Framework with an Intelligent Tutoring System architecture.
2. To implement a generative AI agent (M.E.C.H.A.) capable of providing real-time, context-aware feedback and adaptive remedial learning paths.
3. To measure and analyze the difference in user experience—specifically regarding motivation, curiosity, and perceived usefulness—between the proposed gamified platform and traditional classroom learning methods.

C. Research Questions

Based on the identified problems and objectives, this study addresses the following research questions:

1. How can a web-based learning platform be designed to effectively integrate the eight core drives of the Octalysis Framework with an AI-driven Intelligent Tutoring System?
2. To what extent does the proposed gamified system improve user perception, specifically in terms of Behavioral Intention to Use and Curiosity, compared to traditional non-gamified learning methods?

II. METHOD

This study adopts a Research and Development (R&D) approach utilizing the ADDIE Model (Analysis, Design, Development, Implementation, Evaluation) to ensure a systematic development process. The stages are defined as follows:

1. Analysis: Identifying motivational gaps in current programming pedagogy.
2. Design: Mapping the eight Octalysis core drives to specific system features and architecting the AI-driven ITS logic.
3. Development: Constructing the "Starcoder" platform using Next.js and integrating the Google Gemini API for the "M.E.C.H.A" agent.
4. Implementation: Deploying the system to a pilot cohort of undergraduate students.
5. Evaluation: Assessing user perception and system acceptance using the HMSAM framework.

The platform's frontend is constructed using the Next.js framework, selected for its superior performance in educational web applications through Server-Side Rendering (SSR) and Static Site Generation (SSG). Recent studies indicate that Next.js significantly optimizes load times and SEO, which are critical factors for maintaining learner engagement in digital environments [9], [10]. For the intelligent component, the system integrates the Google Gemini API to power "M.E.C.H.A.," an AI agent capable of natural language processing and code analysis. This integration leverages Generative AI to provide context-aware explanations and adaptive feedback, a method proven to enhance the efficacy of Intelligent Tutoring Systems compared to static rule-based agents [11], [12].

A. System Architecture and Technologies

The operational flow of the "Starcoder" platform is summarized in the mission flowchart shown in Fig. 1. This workflow outlines the process of learning through missions to the adaptive remedial mechanisms triggered by the Intelligent Tutoring System.

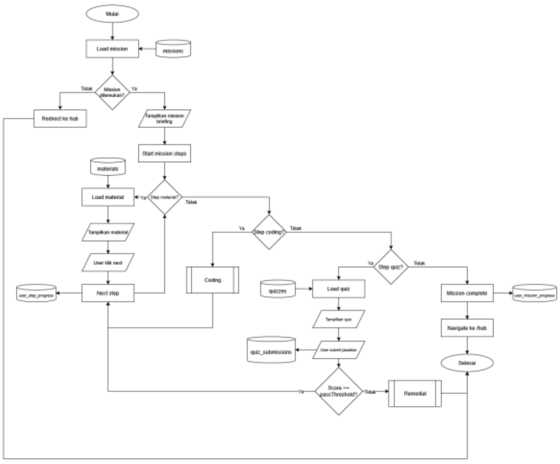

Fig. 1. Mission Flow

The detailed stages in the website are as follows:

1. *User Initialization:* The system begins with user authentication managed by Supabase, ensuring secure session management. Upon registration, a "User Dossier" is automatically generated, initializing gamification metrics such as Experience Points (XP), level, and a daily streak counter.
2. *Gamified Navigation (Nebula Path):* Users access the "Nebula Path," a visual representation of the curriculum where learning modules are depicted as planetary nodes. This interface restricts access to advanced content until prerequisite nodes are successfully completed, ensuring a structured learning trajectory.
3. *Intelligent Instruction:* Upon selecting a mission, the user enters the "Mission Interface." Here, the AI agent "M.E.C.H.A." operates in Tutor Mode, providing contextual explanations of programming concepts via the Gemini API. This interaction model is designed to build trust and facilitate scaffolded learning.
4. *Assessment and Diagnostic:* The learning process culminates in a practical coding challenge. User code is submitted to a secure sandbox environment powered by the Judge0 API, which executes the code against pre-defined test cases to verify functional correctness.
5. *Adaptive Branching:* The system analyzes the assessment results to determine the subsequent learning path:
   a. *Success Scenario:* If the user passes the test cases, the system unlocks the next node, awards XP (Octalysis Core Drive 2), and updates the leaderboard (Core Drive 5).
   b. *Remedial Scenario:* If the user fails, the ITS logic triggers a "System Diagnostic Alert." The system identifies the specific knowledge gap and recommends a "Foundational Mission" (remedial path). This ensures that students master prerequisites before re-attempting the main challenge.
6. *Feedback Loop:* All interactions, including success rates and remedial attempts, are logged in the database to refine future AI recommendations and update the user's "Starchart Log," providing a sense of ownership over their learning progress.

B. Octalysis Gamification Implementation

The gamification strategy for the "Starcoder" platform is grounded in the Octalysis Framework, which categorizes human motivation into eight core drives. Unlike superficial gamification that relies solely on points and badges, Octalysis emphasizes a holistic approach that targets both extrinsic and intrinsic motivation [14].

To ensure sustained engagement, the platform implements specific features corresponding to each of the eight drives. The mapping between the theoretical core drives and their practical implementation in the system is detailed in Table I.

TABLE I. IMPLEMENTATION OF OCTALYSIS CORE DRIVES

Octalysis Core Drive	**Implementation, Risks, and Impact on Evaluation**
Epic Meaning & Calling	Implementation: Users act as "Starcoders" saving a digital galaxy. Impact: This narrative wrapper contributed to the 74.52% Focused Immersion (FI) score, helping students enter a flow state by contextualizing abstract code as mission-critical tools.
Development & Accomplishment	Implementation: Visual progression bars and immediate feedback loops via the AI agent, rather than just static points. Impact: Supported the 86.44% Perceived Usefulness (PU) score by providing tangible evidence of skill acquisition.
Empowerment of Creativity	Implementation: A sandbox coding environment where the AI (M.E.C.H.A.) encourages debugging rather than giving answers. Impact: Directly correlates to the 75.56% Control (CTL) score, as users felt agency over their problem-solving process.
Ownership & Possession	Implementation: Unlockable Profile Pictures (PFPs) and a customizable "User Dossier". Impact: Enhances intrinsic value; students maintain the account not just for grades but to preserve their virtual identity.
Social Influence & Relatedness	Implementation: "Hall of Legends" leaderboard.

	Risk Analysis: While motivating for top performers, leaderboards can induce performance anxiety or demotivation for lower-ranked students. This risk is mitigated by focusing on "Top 3" highlighting rather than shaming low ranks.
Scarcity & Impatience	Implementation: "Daily Challenges" are locked to a 24-hour cycle. Impact: This artificial scarcity contributed to the +15.56% increase in Behavioral Intention to Use (BIU), as students felt a "fear of missing out" (FOMO) on the daily opportunity.
Unpredictability & Curiosity	Implementation: Randomized "Gacha" (Lootbox) rewards for quiz completion. Impact: This mechanic was the primary driver for the +12.47% surge in Curiosity (CUR) compared to the baseline, proving that variable rewards are more engaging than fixed rewards.
Loss & Avoidance	Implementation: A visual "Streak Counter" that resets upon inactivity. Impact: Acts as a retention hook; while effective for short-term engagement (BIU), reliance on this drive must be balanced to avoid burnout.

C. *AI-Driven Intelligent Tutoring System (ITS)*

The platform's cognitive architecture is built upon the classic four-component ITS model—Domain, Student, Pedagogical, and Interface models—enhanced by Generative AI [17]. To ensure pedagogical effectiveness, the system's design is grounded in two core educational theories: Instructional Scaffolding and Mastery Learning.

1. AI Companion Modes: The AI agent, "M.E.C.H.A." (Mission Enhanced Coding Helper & Assistant), does not merely function as a chatbot but operates within the Zone of Proximal Development (ZPD). The agent dynamically adjusts its support level through three context-aware modes to provide appropriate scaffolding:
    a. Tutor Mode (Explicit Instruction): Active during material consumption, this mode lowers Cognitive Load for novice learners by breaking down abstract concepts into digestible analogies and providing direct instruction on syntax and logic.
    b. Co-Pilot Mode (Scaffolding): Active during coding challenges, this mode implements Instructional Scaffolding by offering "faded support." Instead of revealing solutions, the AI provides progressive hints and debugging clues, encouraging students to bridge the gap between their current ability and the target competency independently.
    c. Observer Mode (Formative Assessment): Monitors student inputs during quizzes without interfering, collecting real-time data on misconceptions to update the Student Model.
2. Adaptive Remedial Sequencing (Mastery Learning): The Pedagogical Model implements Bloom's Mastery Learning strategy. The system assumes that students must achieve a high level of proficiency (mastery) in prerequisite concepts before moving to more advanced tasks.
    - Mechanism: When a student fails a core mission (e.g., scoring below the threshold or failing critical test cases), the system triggers a "System Diagnostic Alert."
    - Intervention: Instead of simply allowing a retry, the ITS identifies the specific knowledge gap and activates a "Remedial Training Protocol." This redirects the student to a foundational mission designed to reinforce basic concepts.
    - Outcome: Upon successful completion of the remedial path, the original mission is unlocked. This dynamic adjustment ensures that no student is left behind due to accumulating knowledge gaps.

The flow is as illustrated in Fig. 2.

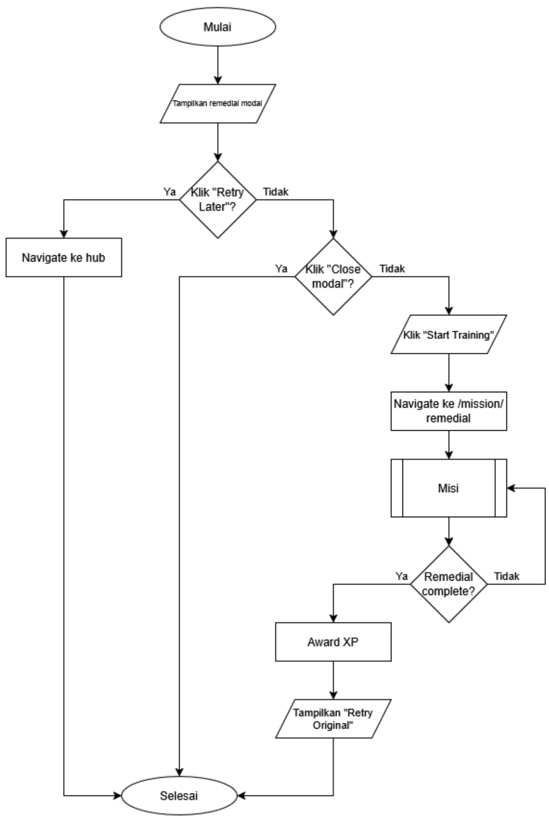

Fig. 2. Remedial Flow

3. Automated Code Assessment via Judge0: To validate practical skills, the system integrates the Judge0 API for secure, real-time code execution. As illustrated in Fig. 3, the live coding workflow follows a strict validation protocol. Upon code submission, the system performs initial validation before sending a batch request to the Judge0 service. The results are compared against pre-defined test cases to provide immediate pass/fail feedback. The live coding flow is as follows:

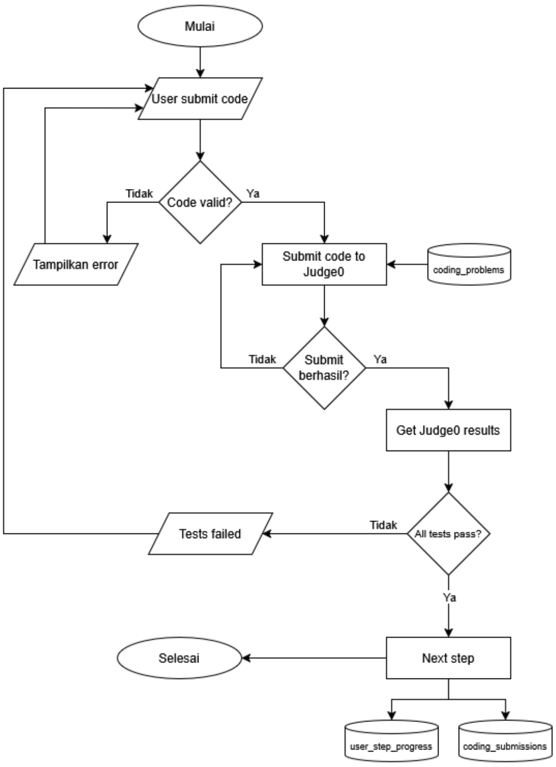

Fig. 3. Live Coding Flow

Upon code submission, the system performs initial validation before sending a batch request to the Judge0 service. The system utilizes a polling mechanism to check submission status asynchronously. Once execution is complete, results are compared against pre-defined test cases . This automated assessment provides immediate pass/fail feedback, which serves as the primary trigger for the system's adaptive logic.

D. Evaluation Design and Instruments

To quantitatively assess the platform's impact on student motivation and engagement, this study employs the Hedonic-Motivation System Adoption Model (HMSAM). Unlike traditional Technology Acceptance Models (TAM) that focus primarily on utilitarian aspects, HMSAM is specifically designed to evaluate systems where intrinsic motivation and enjoyment are critical, making it ideal for gamified learning environments [16], [20].

1. Study Design: This research employs a comparative within-subjects design to evaluate the impact of the proposed system on user perception. The same group of respondents (N=54) evaluated two distinct learning conditions to allow for a direct paired comparison:
   - Condition A (Baseline): The traditional non-gamified classroom instruction (standard lectures and IDEs) that the students had previously experienced or were currently undertaking.
   - Condition B (Experimental): The gamified, AI-supported "Starcoder" platform developed in this study.
2. Measurement of Baseline: To establish a valid baseline for comparison, respondents were explicitly asked to rate their engagement levels regarding their *standard classroom experience* using the same HMSAM constructs prior to evaluating the Starcoder platform. This retrospective evaluation method ensures that the comparison of "Intention to Use" and "Curiosity" is relative to their existing educational context, providing a quantifiable gap between traditional methods and the proposed AI-gamified approach.
3. Data Collection: The survey was distributed to undergraduate students who had completed or were currently enrolled in the Fundamental Programming course. The data collection focused on verifying whether the integration of Octalysis drives and AI support resulted in a statistically observable improvement in user perception compared to the established baseline.
4. Measurement Constructs: The evaluation instrument consists of a questionnaire utilizing a 5-point Likert Scale (1 = Strongly Disagree to 5 = Strongly Agree). The instrument measures seven key constructs:
   - Perceived Ease of Use (PEOU): The degree to which using the system is free of effort.
   - Perceived Usefulness (PU): The degree to which the system enhances learning performance.
   - Curiosity (CUR): The extent to which the system evokes user curiosity.
   - Joy (JOY): The perceived enjoyment and fun derived from the system.
   - Control (CTL): The user's sense of agency over the learning interaction.

- Behavioral Intention to Use (BIU): The likelihood of the user continuing to use the platform.
- Focused Immersion (FI): The level of deep engagement and flow experienced.

5. Data Analysis: The study involved respondents from undergraduate programming courses. The collected data was analyzed using a percentage score formula to determine the agreement level for each construct. The percentage score (PS) is calculated as follows:

$$PS = \frac{\sum(R \times F)}{M \times N} \times 100\%$$

Where:

- R = Weight of the choice (1 to 5).
- F = Frequency of the choice.
- M = Maximum score per item (5).
- N = Total number of respondents.

A score above 80% is categorized as "Strongly Agree" (Very Good), while scores between 61-80% are categorized as "Agree" (Good). This quantitative approach allows for a direct statistical comparison between the gamified "Starcoder" platform and the baseline traditional classroom method.

E. Data Analysis and Validity

To ensure the scientific rigor of the findings, the research instrument underwent validity and reliability testing prior to descriptive analysis.

Internal consistency was measured using Cronbach's Alpha (α) for both the Baseline (Classroom) and Experimental (Starcoder) conditions. A threshold of $\alpha \geq 0.70$ was set as the standard for high reliability, while $0.60 \leq \alpha < 0.70$ was considered acceptable for exploratory psychological constructs in early-stage research.

The analysis of the respondent data (N=54) yielded the following reliability statistics for the Starcoder platform:

- High Reliability: The constructs for Perceived Ease of Use (α=0.927), Behavioral Intention to Use (α=0.908), Perceived Usefulness (α=0.870), and Curiosity (α=0.833) demonstrated excellent internal consistency, confirming that the core acceptance metrics are highly reliable.
- Moderate Reliability: The constructs for Joy (α=0.681) and Control (α=0.668) fell within the acceptable range for exploratory affective measures, indicating consistent enough patterns to draw conclusions about user sentiment.
- Variable Reliability: The Focused Immersion (α=0.573) construct showed lower consistency in the experimental condition compared to the baseline (α=0.768). This suggests that the "flow state" experienced by students in the gamified environment was more variable or subjective than in the traditional classroom, likely due to varying levels of familiarity with game mechanics among the respondents.

Following validation, the data was analyzed using descriptive statistics. The percentage scores were calculated to determine the comparative gap between the Baseline and Experimental conditions, as presented in the Results section.

III. Result and Discussions

A. Platform User Interface Implementation

To validate the functional implementation of the design, the key interfaces of the platform were deployed and tested by the respondents. The visual implementation of the gamified elements and AI support is presented below.

To drive exploration and provide a sense of progression, the Nebula Path replaces traditional module lists with an interactive galactic map, allowing students to visualize their learning trajectory as a space exploration mission.

Fig. 4. Nebula Path Navigation Interface

The core learning experience utilizes a clean, mission-focused interface where learning materials are presented as mission briefings to maintain narrative immersion while delivering technical concepts.

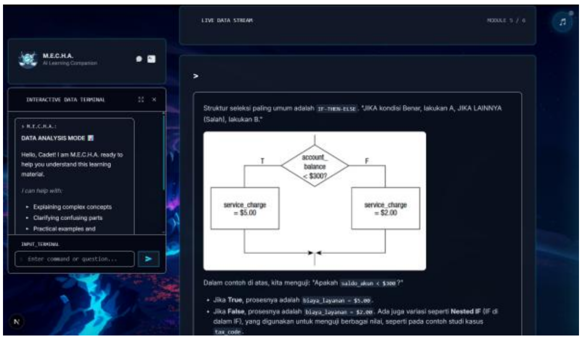

Fig. 5. Mission Learning Material Interface

Interactive assessments are integrated directly into the flow; the Quiz Interface provides immediate feedback on conceptual understanding, reinforcing knowledge before practical application.

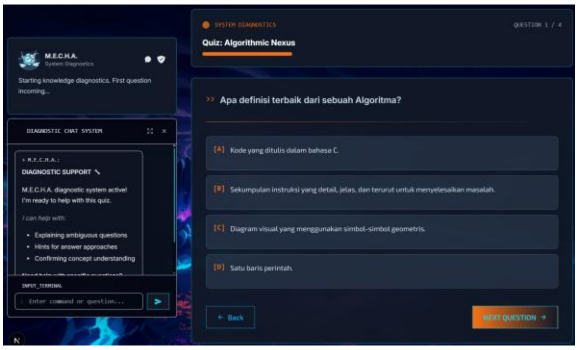

Fig. 6. Interactive Quiz Interface

To validate practical skills, the Live Coding Environment features a split-screen layout with a code editor and real-time execution results, allowing students to write and test code within the browser.

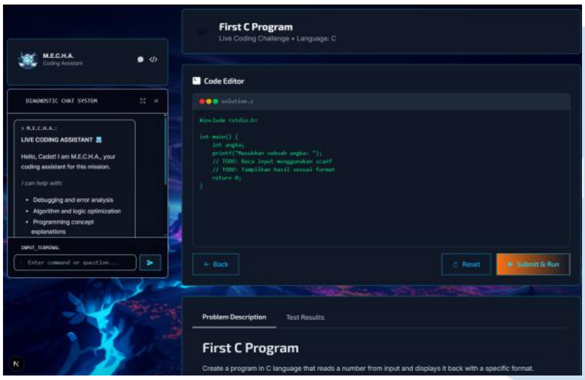

Fig. 7. Live Coding Environment

The AI agent, M.E.C.H.A., is embedded directly within the workspace to offer contextual guidance, answering questions and providing debugging assistance without requiring the user to leave the learning environment.

Fig. 8. AI Companion Chat Interface

Finally, when performance thresholds are not met, the system automatically triggers a Remedial Protocol, directing users to foundational exercises to ensure prerequisite knowledge is mastered before proceeding.

Fig. 9. Adaptive Remedial Prompt

B. *Comparative Analysis of HMSAM Constructs*

The effectiveness of the "Starcoder" platform was evaluated through a quantitative study involving 54 respondents (N=54). The participants were students who had previously taken or were currently enrolled in a Fundamental Programming course. The data was collected using a standard HMSAM questionnaire, ensuring a valid comparison between the traditional classroom experience (Baseline) and the gamified AI-driven platform.

The demographic composition of the respondents provides context for the evaluation results. As illustrated in Fig. 4, the gender distribution was predominantly male (85.2%) compared to female (14.8%), a ratio typical of students. Despite the gender imbalance, the sample size is statistically sufficient to draw initial conclusions regarding system acceptance.

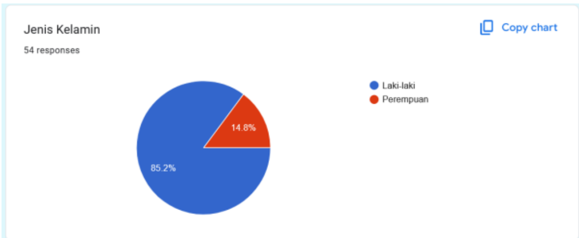

Fig. 10. Gender Distribution of Respondents

In terms of age distribution, shown in Fig. 5, the majority of respondents fell into the 18–20 year age group (61.1%), followed by the 21–23 year group (35.2%). This indicates that the primary evaluators were in the early-to-mid stages of their university education, which aligns perfectly with the target audience for an introductory programming platform. This age group is generally considered "digital natives," possessing a high baseline familiarity with both gaming mechanics and digital learning interfaces.

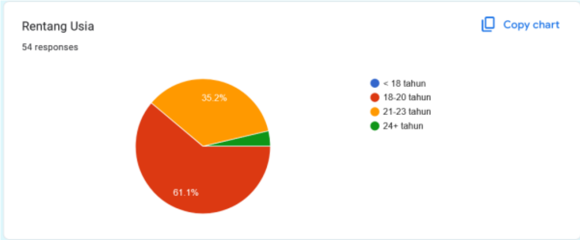

Fig. 11. Age Distribution of Respondents

Furthermore, the study achieved a degree of institutional diversity. While the majority of respondents originated from the host university (87%), there was participation from external institutions (13%), including Binus University and Universitas Terbuka, as shown in Fig. 6.

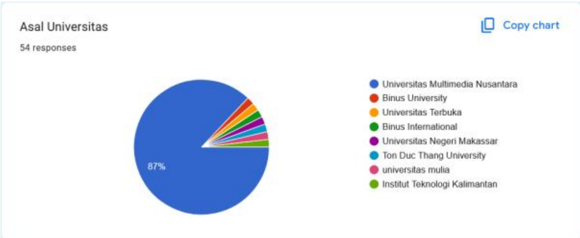

Fig. 12. University Origin of Respondents

The core of the evaluation rests on the comparison between the baseline classroom experience and the "Starcoder" platform across seven HMSAM constructs. The summary of these findings is presented in Table II.

TABLE II. COMPARISON OF EVALUATION SCORES

HMSAM Constructs	Classroom (Baseline)	Starcoder	Increase	Result Category
Behavioral Intention to Use (BIU)	62.84%	78.40%	+15.56%	Good (Agree)
Curiosity (CUR)	73.09%	85.56%	+12.47%	Very Good (Strongly Agree)
Perceived Usefulness (PU)	75.33%	86.44%	+11.11%	Very Good (Strongly Agree)
Control (CTL)	66.42%	75.56%	+9.14%	Good (Agree)
Perceived Ease of Use (PEOU)	76.39%	84.63%	+8.24%	Very Good (Strongly Agree)
Focused Immersion (FI)	67.31%	74.52%	+7.20%	Good (Agree)
Joy (JOY)	67.47%	73.58%	+6.11%	Good (Agree)

The descriptive analysis reveals a consistent positive trend across all metrics, with the "Starcoder" platform receiving higher approval ratings compared to the traditional baseline in every category. The most notable difference was observed in Behavioral Intention to Use (BIU), which showed a difference of 15.56% between the two conditions. In the baseline study, students rated their intention to continue with traditional methods at a moderate 62.84%. However, after interacting with Starcoder, this metric rose to 78.40%. This suggests that the combination of gamification and AI support may help convert a passive learning obligation into an active desire to engage with the material.

The second highest difference was observed in Curiosity (CUR), which was 12.47% higher in the experimental condition, reaching a "Strongly Agree" level of 85.56%. While traditional programming lectures are informative, the data suggests they may struggle to maintain the element of anticipation. By contrast, Starcoder's use of "Black Hat" gamification techniques, specifically Octalysis Core Drive 7 (Unpredictability) via randomized rewards, appears to successfully stimulate student curiosity based on the self-reported responses.

Perceived Usefulness (PU) also saw a substantial increase of 11.11%, achieving the highest overall score of 86.44%. This validates the integration of the AI agent "M.E.C.H.A." unlike static textbooks or pre-recorded videos, the AI provided immediate, context-aware assistance. The ability of the system to diagnose specific errors and offer remedial paths (as detailed in the Methodology) directly contributed to students feeling that the system was practically useful for their learning goals.

The results of this study align with recent literature suggesting that "Personalized Gamification" is superior to generic approaches [17], [20]. The moderate gains in *Control (CTL)* (+9.14%) and *Joy (JOY)* (+6.11%) suggest that the gamified environment provided a stronger sense of agency. The "Nebula Path" allowed students to navigate at their own pace, while the AI agent provided support without taking over the task.

Furthermore, the increase in Focused Immersion (FI) (+7.20%) highlights the potential effectiveness of the narrative wrapper (Epic Meaning). By framing code compilation as a "mission" to repair a digital universe, the platform reportedly induced a deeper state of engagement than standard IDEs. This confirms that wrapping technical tasks in a thematic narrative is a viable strategy for reducing the perceived cognitive load often associated with introductory programming [13], [15].

C. Limitations

While the results indicate a strong preference for the gamified system, several limitations must be acknowledged to contextualize the findings:

1. Descriptive Nature of Analysis: The comparative analysis relies on descriptive statistics (percentage scores) rather than inferential statistical tests (e.g., t-tests). Consequently, while the differences in means

are distinct, statistical significance ($p<0.05$) was not calculated. The findings should be interpreted as indicative of user preference rather than a confirmed causal effect.

2. Self-Reported Measures: The data is derived entirely from the HMSAM questionnaire, which measures perceived experience. This introduces potential perceptual bias, where a "fun" system is rated highly even if learning gains are not proportional.

3. Lack of Objective Learning Outcomes: This study did not measure cognitive learning gains through pre-test and post-test scores. While Behavioral Intention to Use is a strong predictor of engagement, it does not guarantee improved academic performance (grades). Future research will require longitudinal studies with control groups to validate whether this increased motivation translates into measurable programming proficiency.

IV. Conclusions

This study designed and evaluated "Starcoder," an adaptive learning platform integrating the Octalysis Gamification Framework with a Generative AI-driven Intelligent Tutoring System. The development process confirmed that combining modern web technologies (Next.js) with Large Language Model APIs (Gemini) is a technically viable approach for creating responsive, gamified educational environments.

Evaluation results from the pilot cohort (N=54) indicate a positive reception of the system. Descriptive analysis of the HMSAM constructs reveals that the gamified platform received higher approval ratings compared to the traditional classroom baseline, particularly in *Perceived Usefulness* (86.44%) and *Curiosity* (85.56%). The data suggests that the integration of narrative elements and adaptive AI support may effectively foster *Behavioral Intention to Use*, which showed a 15.56% difference relative to the baseline. These findings support the premise that addressing intrinsic motivation through "White Hat" (Meaning, Accomplishment) and "Black Hat" (Scarcity, Curiosity) gamification drives is a promising strategy for enhancing student engagement.

However, these conclusions must be interpreted within the context of the study's limitations. The findings rely on self-reported perception data rather than objective learning outcomes, and the sample size prevents broad generalization. Consequently, future research should prioritize the following areas to validate the pedagogical efficacy of the system:

- Objective Performance Measurement: Future studies must implement a quasi-experimental design with Pre-Test and Post-Test assessments to measure actual cognitive gains and programming proficiency, rather than relying solely on perceived usefulness.
- Longitudinal and Scaled Analysis: To rule out the "novelty effect," longitudinal studies with a larger, multi-institutional sample size are required to determine if the increased motivation translates into sustained long-term engagement and retention.
- Technical Evolution: Beyond validation, technical development should focus on enabling the AI to dynamically adjust gamification parameters (e.g., difficulty curves, reward probabilities) in real-time based on the learner's performance data, moving from a static rule-based system to a fully adaptive motivational engine.

References

[1] L. Christopher and A. Waworuntu, "Java programming language learning application based on octalysis gamification framework," IJNMT (International Journal of New Media Technology), vol. 8, no. 1, pp. 65-69, 2021.

[2] E. Ratinho and C. Martins, "The role of gamified learning strategies in student's motivation in high school and higher education: A systematic review," Heliyon, vol. 9, 2023.

[3] N. A. Pratama, A. C. Sari, and P. K. N. Bara, "The effect of online learning on student motivation and achievement during the pandemic," Edumaniora: Jurnal Pendidikan dan Humaniora, vol. 3, no. 2, pp. 33-38, 2024.

[4] T. Karakose, R. Yirci, S. Papadakis, T. T. Ozdemir, M. Demirkol, and H. Polat, "Examining the effectiveness of gamification as a tool promoting teaching and learning in educational settings: a meta-analysis," Frontiers in Psychology, vol. 14, p. 1253549, 2023.

[5] E. Rivera and C. Garden, "Gamification for student engagement: a framework," Journal of Further and Higher Education, vol. 45, pp. 999–1012, 2021.

[6] A. Létourneau, M. Dupuis, and S. Tremblay, "A systematic review of ai-driven intelligent tutoring systems: Effects on k-12 learning and retention," International Journal of Artificial Intelligence in Education, vol. 35, no. 2, pp. 215-238, 2025.

[7] S. D. Ristianto, A. Putri, and Y. Rosmansyah, "Personalized gamification a technological approach for student education: A systematic literature review," IEEE Access, vol. 13, pp. 55 712-55 726, 2025.

[8] F. Naseer, M. Khan, A. Addas, Q. Awais, and N. Ayub, "Game mechanics and artificial intelligence personalization: A framework for adaptive learning systems," *Education Sciences*, vol. 15, p. 301, 2025.

[9] R. Pati, N. Sharma, and J. Lee, "Building scalable intelligent tutoring systems with next.js and ai microservices," Journal of Web Engineering, vol. 19, no. 2, pp. 123-145, 2025.

[10] Aghaei and S. Kumar, "Performance and seo benefits of next.js in educational web applications," in *Proceedings of the International Conference on Educational Technology*, 2024, pp. 78-89.

[11] H. Almetnawy, A. Orabi, A. Alneyadi, T. Ahmed, and A. Lakas, "An adaptive intelligent tutoring system powered by generative ai," 2025, pp. 1-10.

[12] M. Mukherjee, J. Le, and Y. Chow, "Generative ai-enhanced intelligent tutoring system for graduate cybersecurity programs," *Future Internet*, vol. 17, no. 4, p. 154, 2025.

[13] G. Papagni, J. de Pagter, S. Zafari, M. Filzmoser, and S. T. Koeszegi, "Artificial agents' explainability to support trust: considerations on timing and context," *AI & SOCIETY*, vol. 38, no. 2, pp. 947–960, 2022.

[14] Octalysis Group, "Framework," Sep 2023. [Online]. Available: https://octalysisgroup.com/framework/

[15] F. Marisa, S. S. S. Ahmad, Z. I. M. Yusoh, A. L. Maukar, R. D. Marcus, and A. A. Widodo, "Evaluation of student core drives on e-learning during the covid-19 with octalysis gamification framework," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 11, pp. 104-116, 2020.

[16] Suwarno, D. Deli, and M. Christian, "Exploring the impact of octalysis gamification in japanese m-learning using the technology acceptance model," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 9, no. 1, pp. 77-88, 2024.

[17] P. Sharma and M. Harkishan, "Designing an intelligent tutoring system for computer programing in the pacific," *Education and Information Technologies*, vol. 27, pp. 6197 – 6209, 2022.

[18] T. Shah, "Ai-powered personalised learning plans for intelligent tutoring systems," *International Journal for Research in Applied Science and Engineering Technology*, vol. 12, no. 5, pp. 5450-5462, 2024.

[19] F. Keshtkar, N. Rastogi, S. Chalarca, and S. Ahmad Chan Bukhari, "Ai tutor: Student's perceptions and expectations of ai-driven tutoring systems: A survey-based investigation," in The International FLAIRS Conference Proceedings, vol. 37, 2024.

[20] R. Pati, N. Sharma, and J. Lee, "Building scalable intelligent tutoring systems with next.js and ai microservices," Journal of Web Engineering, vol. 19, no. 2, pp. 123-145, 2025