JCAR 6 (2) (2024)

Journal of Classroom Action Research

http://jppipa.unram.ac.id/index.php/jcar/index

Computational Physics Education through AI-Assisted Media: Improving Pascal Programming Skills for Sixth-Semester Physics Education Students at PMIPA FKIP UNRAM

Muhammad Taufik[1*], Hikmawati[2]

[1,2] Phgysics education program, Faculty of Teacher Training and Education University of Mataram, Mataram, Indonesia

DOI: https://doi.org/10.29303/jppipa.v6i2.8225

Received : 20 Februari 2024 Revised : 24 Mei 2024 Accepted : 30 Mei 2024

Abstract: This study investigates the integration of AI-assisted media to enhance Pascal programming skills among sixth-semester Physics Education students at PMIPA FKIP UNRAM within the domain of computational physics education. Utilizing a classroom action research methodology over two cycles, the intervention employed AI-powered programming assistants and interactive platforms to enhance students' comprehension and application of Pascal programming. Initial findings from Cycle 1 highlighted ongoing challenges despite technological integration, with a mean score of 61.13 (SD=6.67). However, refinements in AI-assisted feedback and practice activities during Cycle 2 resulted in a significant improvement in mean scores to 78.06 (SD=7.6), accompanied by a substantial normalized gain (mean=0.4423, SD=0.1265). Detailed median analyses from Mood's Median Test for Cycle 2 further underscored improvements across various score ranges, with an overall median of 60.0 indicating enhanced student performance. These results underscore the transformative potential of AI-assisted educational interventions in advancing Pascal programming skills and enriching computational physics education at PMIPA FKIP UNRAM.

Keywords: AI-Assisted Media, Classroom Action Research, Computational Physics Education, Pascal Programming Skills, PMIPA FKIP UNRAM.

Introduction

Programming skills are crucial for undergraduate Physics Education students, significantly enhancing their ability to analyze data, model physical systems, and solve complex problems (Fidan & Tuncel, 2019). Mastery of programming fosters critical thinking and creativity, essential for conducting experiments and interpreting results (Cargas, et al., 2017). Integrating programming into the physics curriculum allows students to simulate impractical classroom experiments, deepening their understanding of physical concepts. According to Pendergast (2006), teaching introductory programming, despite challenges like Java, is fundamental for building a strong career foundation.

Incorporating AI in programming education further enhances both teaching and learning experiences (Chen et al., 2020). AI tools in programming curricula offer personalized learning paths, addressing diverse learning styles and proficiency levels (Tapalova & Zhiyenbayeva, 2022). These intelligent systems provide instant feedback and error correction, promoting a deeper understanding of programming concepts (Kotsiantis et al., 2024) and bridging theoretical knowledge with practical application (Vaithilingam et al., 2023). AI-powered programming tools significantly improve learning effectiveness in areas like code comprehension and debugging.

Moreover, AI-driven virtual laboratories in physics education enhance experimental skills and data

Email: taufik@unram.ac.id

analysis abilities (Sibonghanoy et al., 2024). Recent research by Taufik et al. (2024) focuses on enhancing numerical literacy through Pascal programming for physics education students. AI-assisted programming platforms have been reported to improve coding proficiency and self-regulated learning (Rajamani, 2022). However, empirical research on AI-assisted media's impact on Pascal programming skills in physics education remains limited.

This study addresses this gap by implementing an AI-assisted intervention to enhance Pascal programming abilities in numerical integration tasks, aligning with evolving needs in computational physics education and contributing to technology-enhanced learning in STEM disciplines. By leveraging AI-powered programming assistants and interactive learning platforms, this research aims to provide insights into effective pedagogical strategies that benefit computational physics education and the broader field of STEM.

Method

This research employs a classroom action research (CAR) approach, structured over two cycles: planning, action, observation, and reflection (Kemmis et al., 2014). The participants consist of 31 sixth-semester Physics Education students at FKIP UNRAM. AI-assisted learning tools, specifically designed to teach Pascal programming within the context of computational physics, are seamlessly integrated into the curriculum. Data is meticulously collected through a combination of pre- and post-tests, student surveys, and direct observations to gauge the effectiveness of the intervention and the progression of students' skills.

In Cycle 1, In the planning phase, learning materials were developed from computational physics textbooks and literature sources to teach Pascal programming concepts related to numerical integration methods, including the trapezoidal rule, Simpson's rule, and Gaussian quadrature.

During the action phase, face-to-face classes were conducted, where theoretical explanations of the numerical integration methods were provided, accompanied by practical examples implemented in Excel and Pascal programming. Students were given access to the code examples and exercises from the computational physics textbooks and literature sources.

In the observation phase, students were monitored as they worked on inputting and testing the code from the textbook sources. Any errors encountered were identified, and students were guided to debug and correct their programs. This phase was conducted over two 50-minute sessions, alternating between theoretical explanations and practical coding exercises.

The reflection phase involved assessing student performance and understanding of the numerical integration methods and Pascal programming concepts based on the textbook examples. Areas requiring improvement were identified, and feedback was gathered to refine the teaching materials and approach for the subsequent cycle.

Cycle 2 In the planning phase, the teaching materials and methods were adjusted based on the feedback and observations from Cycle 1. Additionally, AI-assisted learning tools were integrated to provide a wider variety of programming exercises and examples as follows.

```
program NumericalQuadrature;
uses Math;

type
  Func = function(x: Double): Double;

function f(x: Double): Double;
begin
  // The example of the function to be integrated, for instance f(x) =
x^2
  f := x * x;
end;

function TrapezoidalRule(a, b: Double; n: Integer; func: Func):
Double;
var
  h, sum, x: Double;
  i: Integer;
begin
  h := (b - a) / n;
  sum := (func(a) + func(b)) / 2.0;
  for i := 1 to n - 1 do
  begin
    x := a + i * h;
    sum := sum + func(x);
  end;
  TrapezoidalRule := sum * h;
end;

function SimpsonsRule(a, b: Double; n: Integer; func: Func): Double;
var
  h, sum, x: Double;
  i: Integer;
begin
  if n mod 2 = 1 then
    n := n + 1; // n must be an even on  Simpson's Rule
  h := (b - a) / n;
  sum := func(a) + func(b);
  for i := 1 to n - 1 do
  begin
    x := a + i * h;
    if i mod 2 = 0 then
      sum := sum + 2 * func(x)
    else
      sum := sum + 4 * func(x);
  end;
  SimpsonsRule := sum * h / 3.0;
end;
```

```
function GaussianQuadrature(a, b: Double; func: Func): Double;
const
  x: array[1..2] of Double = (-0.5773502691896257, 0.5773502691896257);
  w: array[1..2] of Double = (1.0, 1.0);
var
  xm, xr, sum: Double;
  i: Integer;
begin
  xm := 0.5 * (b + a);
  xr := 0.5 * (b - a);
  sum := 0.0;
  for i := 1 to 2 do
    sum := sum + w[i] * func(xm + xr * x[i]);
  GaussianQuadrature := sum * xr;
end;

var
  a, b: Double;
  n: Integer;
  result: Double;
begin
  a := 0.0; // Lower bound
  b := 1.0; // Upper  bound
  n := 100; // The number of sub-intervals

  // Using Trapezoidal Rule
  result := TrapezoidalRule(a, b, n, @f);
  WriteLn('Trapezoidal Rule: ', result:0:10);

  // Using Simpson''s Rule
  result := SimpsonsRule(a, b, n, @f);
  WriteLn('Simpson''s Rule: ', result:0:10);

  // Using Gaussian Quadrature
  result := GaussianQuadrature(a, b, @f);
  WriteLn('Gaussian Quadrature: ', result:0:10);
end.
```

During the action phase, students were tasked with modifying the program functions from Cycle 1 using the AI-assisted tools. For example, instead of implementing a quadratic function for numerical integration from the textbook examples, students were prompted by the AI system to modify the code to integrate cubic or logarithmic functions.

In the observation phase, students' understanding was evaluated based on their ability to successfully modify and execute the programs with the new functions using the AI-assisted tools. Peer discussions were facilitated to encourage collaborative learning and problem-solving.

The reflection phase involved assessing the extent of concept mastery among students based on their performance with the AI-assisted exercises. Those who demonstrated a solid understanding of the numerical integration methods and Pascal programming were identified, while others who needed additional support were noted.

Throughout both cycles, students were actively engaged in understanding the syntax and structure of Pascal programming, debugging errors, and modifying programs to incorporate different functions and applications. In Cycle 1, the code examples and exercises were sourced from computational physics textbooks and literature, while in Cycle 2, the AI-assisted learning tools provided a rich and varied set of exercises, enabling students to practice and reinforce their skills in an adaptive and supportive learning environment.

Result and Discussion

The research findings and discussions will be presented in the following separate section, focusing on the enhancement of Pascal programming skills among sixth-semester physics education students at PMIPA FKIP UNRAM through the implementation of AI-assisted media in computational physics education.

The results of a proficiency test assessing programming skills in numerical integration across both Cycle 1 and Cycle 2 are presented in Figure 1.

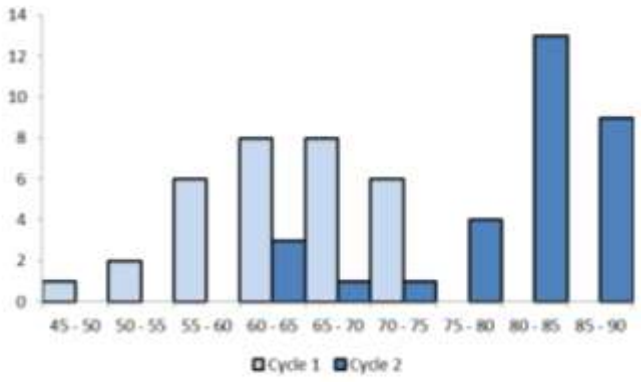

Figure 1. Histogram data test cycle 1 and cycle 1

The descriptive statistics of the numerical integration material test in Cycle 1 and Cycle 2 are presented in Table 1.

Table 1. Descriptive Statistics: Cycle 1, Cycle 2, N-Gain

Variable	N	Mean	StDev	Minimum	Median	Maximum
Cycle 1	31	61.13	6.67	45	60	70
Cycle 2	31	78.06	7.6	60	80	85
N-Gain	31	0.4423	0.1265	0.11	0.44	0.63

The normality data of test in numerical integration of the students on cycle 1 is presented on Figure 2.

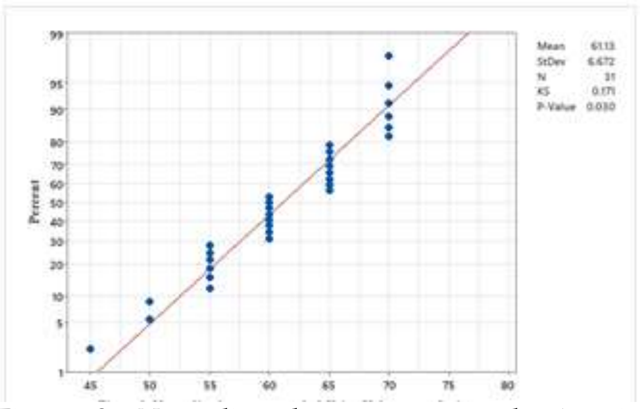

Figure 2. Normality data test on cycle 1 using kolmogrov-smirnov.

The Kolmogorov-Smirnov (K-S) test was used to assess the normality of the data distribution for Cycle 1, with results showing a mean of 61.13, standard deviation of 6.672, and a sample size of 31. The K-S test statistic was 0.171, with a p-value of 0.030. Given the significance level of 0.05, the p-value indicates that we reject the null hypothesis, concluding that the data for Cycle 1 does not follow a normal distribution. Consequently, non-parametric tests should be considered for further statistical analysis to ensure the accuracy and reliability of the interpretations regarding students' performance improvements.

The normality data of test in numerical integration of the students on cycle 2 is presented on Figure 3.

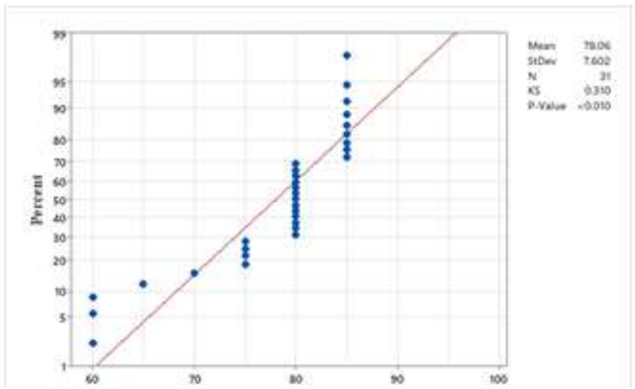

Figure 3. Normality data test on cycle 1 using kolmogrov-smirnov.

The Kolmogorov-Smirnov (K-S) test was conducted to evaluate the normality of the Cycle 2 data, revealing a mean of 78.06, a standard deviation of 7.602, a sample size of 31, a K-S test statistic of 0.310, and a p-value of less than 0.010. Given the significance level of 0.05, the p-value indicates a rejection of the null hypothesis, confirming that the data for Cycle 2 does not follow a normal distribution. Consequently, non-parametric tests should be employed for further statistical analysis to ensure the accuracy and reliability of the interpretations regarding the improvements in students' performance.

Given that the data in Cycle 1 and Cycle 2 are not normally distributed, a non-parametric statistical analysis using Mood's Median Test was conducted to compare the test results between the two cycles. The outcomes are presented in Table 2.

Table 2. Mood's Median Test: Cycle 1 versus Cycle 2

Cycle 2	Median	N <= Overall Median	N > Overall Median	Q3 – Q1	95% Median CI
60	50.0	3	0	5.0	(50, 55)
65	45.0	1	0	*	(45, 45)
70	55.0	1	0	*	(55, 55)
75	57.5	4	0	5.0	(55, 60)
80	65.0	6	7	7.5	(60, 66.5768)
85	65.0	2	7	7.5	(61.1395, 70)
Overall	60.0				

Levels with < 6 observations have confidence < 95.0%

The descriptive statistics from data table 2 using Mood's Median Test for Cycle 2 reveal a detailed median analysis across various score ranges. For a score of 60, the median was 50.0, with three students scoring at or below the overall median and none scoring above it, with an interquartile range (Q3–Q1) of 5.0 and a 95% median confidence interval (CI) of (50, 55). For a score of 65, the median was 45.0, with one student scoring at or below the overall median, none above, and a 95% median CI of (45, 45). At a score of 70, the median was 55.0, with one student scoring at or below the overall median, none above, and a 95% median CI of (55, 55). For a score of 75, the median was 57.5, with four students scoring at or below the overall median, none above, an interquartile range of 5.0, and a 95% median CI of (55, 60). For a score of 80, the median was 65.0, with six students scoring at or below the overall median and seven above, an interquartile range of 7.5, and a 95% median CI of (60, 66.5768). Lastly, for a score of 85, the median was 65.0, with two students scoring at or below the overall median and seven above, an interquartile range of 7.5, and a 95% median CI of (61.1395, 70). The overall median for Cycle 2 was 60.0, indicating an upward trend in student performance, though levels with fewer than six observations had confidence intervals below 95.0%. This data suggests a significant improvement in numerical integration skills among the students, reflecting the efficacy of the AI-assisted educational intervention.

The study initially identified a notable disparity in Pascal programming proficiency among sixth-semester Physics Education students at PMIPA FKIP UNRAM during Cycle 1. However, significant progress was evident in Cycle 2, where students displayed enhanced programming skills and a better grasp of applying these skills to solve computational physics problems. This improvement is corroborated by descriptive statistics from data table 2, utilizing Mood's Median Test for Cycle 2, which provided a comprehensive analysis of median scores across different ranges. For instance, scores at 60 exhibited a median of 50.0, with three students scoring at or below this median and none above, alongside a 95% median confidence interval (CI) of (50, 55). Similarly, scores at 65 had a median of 45.0, with one student below the median and a 95% median CI of (45, 45). Higher scores, such as 80 and 85, showed medians of 65.0, with an interquartile range (Q3–Q1) of 7.5 and 95% median CIs of (60, 66.5768) and (61.1395, 70), respectively. Overall, the Cycle 2 median of 60.0 indicates a positive trajectory in student performance, underscoring the effectiveness of AI-assisted educational interventions in bolstering numerical integration skills and fostering student engagement in computational physics education.

These findings highlight the transformative potential of AI-based educational technologies in delivering personalized and adaptive learning experiences. By enhancing understanding of both theoretical principles and their practical applications, these tools effectively bridge the gap between academic learning and real-world problem-solving in computational physics. The integration of AI-assisted methods not only enhances educational outcomes but also equips students with essential skills that are pertinent to future academic pursuits and professional endeavors in computational sciences. Continued exploration of advanced AI applications and longitudinal studies will further elucidate the sustained impact of these technologies on educational practices and student achievement in physics and related disciplines.

Conclusion

This study investigates the impact of AI-assisted media on Pascal programming skills among sixth-semester Physics Education students at PMIPA FKIP UNRAM, employing a classroom action research (CAR) methodology over two cycles. The integration of AI-based learning tools aimed to enhance students' understanding and application of programming concepts in computational physics education. Analysis using Minitab software revealed a substantial average N-Gain of approximately 0.44, indicating significant improvement post-intervention. The findings are supported by descriptive statistics from data table 2 using Mood's Median Test for Cycle 2, which detailed median analyses across score ranges. Notably, scores at 60 had a median of 50.0, with an interquartile range (Q3–Q1) of 5.0 and a 95% median confidence interval (CI) of (50, 55). Similarly, scores at 65 had a median of 45.0, with a 95% median CI of (45, 45), and scores at 80 and 85 showed medians of 65.0, with interquartile ranges of 7.5 and 95% median CIs of (60, 66.5768) and (61.1395, 70), respectively. The overall Cycle 2 median of 60.0 indicates an upward trend in student performance, emphasizing the efficacy of AI-assisted educational interventions in enhancing numerical integration skills and fostering deeper engagement in computational physics education.

Furthermore, AI tools provided tailored learning experiences that facilitated mastery of complex programming tasks and application of numerical integration methods such as trapezoidal, Simpson, and Gaussian quadrature (Fitria, 2021). This innovative approach effectively addressed conventional teaching challenges, creating a more engaging and effective learning environment. Looking ahead, future research should explore the enduring impacts of AI-assisted learning interventions and their broader applications in physics education. Longitudinal studies could provide insights into sustained learning outcomes and the evolving role of AI in enhancing teaching methodologies. Continuous advancements in AI technologies present opportunities to refine educational practices, equipping students to meet the evolving demands of computational physics and related disciplines effectively.

References

Cargas, S., Williams, S., & Rosenberg, M. (2017). An approach to teaching critical thinking across disciplines using performance tasks with a common rubric. *Thinking Skills and Creativity*, *26*, 24-37.

Chen, L., Chen, P., & Lin, Z. (2020). Artificial intelligence in education: A review. *Ieee Access*, *8*, 75264-75278.

Fidan, M., & Tuncel, M. (2019). Integrating augmented reality into problem based learning: The effects on learning achievement and attitude in physics education. *Computers & Education*, *142*, 103635.

Fitria, T. N. (2021, December). Artificial intelligence (AI) in education: Using AI tools for teaching and learning process. In *Prosiding Seminar Nasional & Call for Paper STIE AAS* (Vol. 4, No. 1, pp. 134-147).

Kemmis, S., McTaggart, R., & Nixon, R. (2014). The Action Research Planner: Doing Critical Participatory Action Research. Springer Singapore. https://doi.org/10.1007/978-981-4560-67-2

Kotsiantis, S.; Verykios, V.; Tzagarakis, M. AI-Assisted Programming Tasks Using Code Embeddings and Transformers. Electronics, 2024, 13, 767. https://doi.org/10.3390/electronics13040767

Nazaretsky, T., Ariely, M., Cukurova, M. & Alexandron, G. (2022). Teachers' trust in AI-powered educational technology and a professional development program to improve it. British Journal of Educational Technology, 53, 914–931. https://doi.org/10.1111/bjet.13232

P. Vaithilingam et al., "Towards More Effective AI-Assisted Programming: A Systematic Design Exploration to Improve Visual Studio IntelliCode's User Experience," 2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Melbourne, Australia, 2023, pp. 185-195, doi: 10.1109/ICSE-SEIP58684.2023.00022

Paul Tschisgale, Peter Wulff, and Marcus Kubsch. Integrating artificial intelligence-based methods into qualitative research in physics education research: A case for computational grounded theory, Phys. Rev. Phys. Educ. Res. 19, 020123. https://doi.org/10.1103/PhysRevPhysEducRes.19.020123

Pendergast, M.O. (2006). Teaching Introductory Programming to IS Students: Java Problems and Pitfalls. Journal of Information Technology Education: Research, 5(1), 491-515. Informing Science Institute. Retrieved June 21, 2024 from https://www.learntechlib.org/p/111559/.

Rajamani, Sriram. AI Assisted Programming, Proceedings of the 15th Annual ACM India Compute Conference November 2022 Pages 5. https://doi.org/10.1145/3561833.3568496

Sibonghanoy, Elma., Groenewald, Kumar, Nand., Avinash, Irfan, Shahrukh., Yerasuri, Santosh. Virtual Laboratories Enhanced by AI for hands-on Informatics Learning. Journal of Informatics Education and Research ISSN: 1526-4726 Vol 4 Issue 1(2024)

Tapalova, O., & Zhiyenbayeva, N. (2022). Artificial intelligence in education: AIEd for personalised learning pathways. *Electronic Journal of e-Learning*, *20*(5), 639-653.

Taufik, M., Rokhmat, J., & Zuhdi, M. (2024). Improving Students' Numerical Literacy Through Project-Based Learning (PjBL) in Pascal Programming Course . International Journal of Contextual Science Education, 1(1), 6–10. https://doi.org/10.29303/ijcse.v1i1.549

Wong, M.-F.; Guo, S.; Hang, C.-N.; Ho, S.-W.; Tan, C.-W. Natural Language Generation and Understanding of Big Code for AI-Assisted Programming: A Review. Entropy 2023, 25, 888. https://doi.org/10.3390/e2506088