Performance Analysis of Fifth-Semester Physics Education: Assessing an Online Pascal Compiler in the Department of Mathematics and Natural Sciences Education at University of Mataram

Muhammad Taufik* & Syahrial Ayub
Physics Education Study Program, University of Mataram, Indonesia
*Corresponding Author: taufik@unram.ac.id

Received: 29th May 2024; **Accepted**: 25th June 2024; **Published**: 29th June 2024
DOI: https://dx.doi.org/10.29303/jpft.v10i1.7089

Abstract *- This study evaluates the integration of online Pascal compilers in undergraduate physics education at the University of Mataram, focusing on its impact on students' programming skills. Over one semester, three classes (A, B, and C) of physics education students from the Department of Mathematics and Natural Sciences Education (PMIPA), Faculty of Teacher Training and Education (FKIP), University of Mataram, participated in lectures, discussions, and practical exercises using the GDB Online Pascal Compiler. Course content included Pascal fundamentals, arrays, matrices, and applications in physics, evaluated through assignments, mid-semester, and final exams. Quantitative analysis of student performance data revealed significant differences in mean scores and variability among the classes, with Class A achieving the highest average score and Class C exhibiting the greatest variability. Statistical tests confirmed that performance data for all classes deviated significantly from normal distribution, necessitating non-parametric tests. The Mann-Whitney U test indicated no statistically significant differences in median scores between class A vs. class B, class A vs. class C, and class B vs. class C, suggesting consistent performance outcomes across classes post-course. These findings underscore the potential of the online Pascal compiler in enhancing students' programming skills within the context of physics education*

Keywords*: Pascal Compiler; Undergraduate Physics Education; Programming Skills; Quantitative Analysis; Non-Parametric Tests.*

INTRODUCTION

In the ever-evolving digital era, programming skills have become an essential competency for physics undergraduate graduates. The ability to develop, implement, and analyze programming algorithms not only facilitates the modeling and simulation of complex physical phenomena but also opens up career opportunities in diverse fields, such as data analysis, scientific visualization, and scientific software development (Harimurti et.al, 2019, Khairiah, Irmayati.,2023, Munandar, H., Sutrio, S., & Taufik, M. 2018, Sherin, B. L. 2001). However, the effective integration of programming learning into undergraduate physics curricula remains a challenge for many higher education institutions, including the University of Mataram. One promising approach to enhancing programming learning outcomes is the integration of online compilers into the learning process. Online compilers offer web-based integrated development environments (IDEs) that allow students to write, compile, and run their programs without the need for complex local software installations (Robins, A., Rountree, J., & Rountree, N. 2003). This approach offers several advantages, such as increased accessibility, simplified technical requirements, and the ability for real-time collaboration. The interface of the GDP Online Pascal Compiler is illustrated in Figures 1 and 2.

Figure 1 displays the main menu, while Figure 2 shows the workspace. The display in Figure 1 demonstrates how a

Pascal program can be initiated and written in the workspace, as illustrated in Figure 2 and Table 1 for detailed.

The Department of Mathematics and Natural Sciences Education (PMIPA) within the Faculty of Teacher Training and Education (FKIP) at the University of Mataram acknowledged the critical importance of programming skills for students in physics education. To meet this educational need, a one-semester course was developed to introduce students to the fundamentals of programming using the Pascal language. The course was delivered through a combination of face-to-face lectures, discussions, and practical exercises utilizing the GDB Online Pascal Compiler.

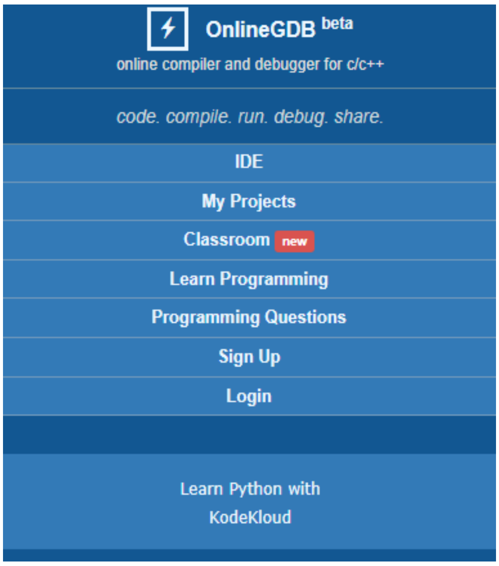

Figure 1. Online GDB Display

The course covered a range of topics, including basic programming concepts in Pascal, arrays, matrices, and their applications in physics. Student assessment was conducted through assignments, a mid-semester exam, and a final exam, providing insights into the success of the course and the suitability of the online Pascal compiler. The ability to create Pascal programs can later be applied and developed for various purposes in physics and its applications, such as earthquake modeling applications (Zuhdi, M., Taufik, M., Sutrio, S., & Ayub, S., 2019).

Numerous studies have highlighted the potential benefits of incorporating online compilers into programming education. According to Krishnamurthi and Fisler (2019), web-based programming environments provide immediate feedback, support exploration, and encourage active learning, which can lead to improved engagement and learning outcomes. Additionally, Lahtinen et al. (2005) found that online practice systems with automated feedback enhance students' programming skills by allowing them to practice at their own pace and receive immediate feedback on their solutions.

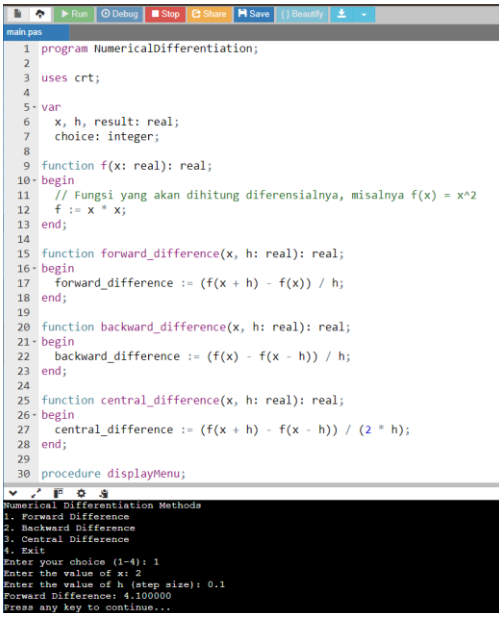

Figure 2. Worksheet Display GDB

However, the effectiveness of online compilers in the specific context of undergraduate physics education has not been extensively explored. While programming is a valuable skill for physics students, the integration of programming learning into physics curricula often faces unique challenges, such as competing demands from other core subjects and the need for contextualized learning experiences (Gomes, A., & Mendes, A. J. 2007). This research aims to bridge this gap by investigating the impact of integrating the GDB Online Pascal Compiler into programming learning for physics education students at the University of Mataram. By adopting a mixed-methods approach, the study seeks to provide a comprehensive understanding of the effects on students' learning outcomes.

Table 1. Numerical Diffentiation Pascal code on GDB

```
program NumericalDifferentiation;
uses crt;
var
 x, h, result: real;
 choice: integer;
function f(x: real): real;
begin
 // Fungsi yang akan dihitung diferensialnya,
misalnya f(x) = x^2
 f := x * x;
end;
function forward_difference(x, h: real): real;
begin
 forward_difference := (f(x + h) - f(x)) / h;
end;
function backward_difference(x, h: real): real;
begin
 backward_difference := (f(x) - f(x - h)) / h;
end;
function central_difference(x, h: real): real;
begin
 central_difference := (f(x + h) - f(x - h)) / (2 * h);
end;
procedure displayMenu;
begin
 writeln('Numerical Differentiation Methods');
 writeln('1. Forward Difference');
 writeln('2. Backward Difference');
 writeln('3. Central Difference');
 writeln('4. Exit');
 write('Enter your choice (1-4): ');
end;
begin
 repeat
  clrscr;
  displayMenu;
  readln(choice);
    if (choice >= 1) and (choice <= 3) then
  begin
   write('Enter the value of x: ');
   readln(x);
   write('Enter the value of h (step size): ');
   readln(h);
  end;
    case choice of
   1: begin
      result := forward_difference(x, h);
      writeln('Forward Difference: ', result:0:6);
     end;
   2: begin
      result := backward_difference(x, h);
      writeln('Backward Difference: ',
result:0:6);
     end;
   3: begin
      result := central_difference(x, h);
      writeln('Central Difference: ', result:0:6);
     end;
   4: writeln('Exiting the program...');
  else
   writeln('Invalid choice. Please select a valid
option.');
  end;
    if (choice >= 1) and (choice <= 3) then
  begin
   writeln('Press any key to continue...');
   readkey;
  end;
   until choice = 4;

end.
```

RESEARCH METHODS

This research utilizes quantitative analysis to assess the effectiveness of integrating the GDP Online Pascal Compiler into programming education for physics students at the University of Mataram over a one-semester period. Student scores on assignments, midterm exams, and final exams were collected following the implementation of the online compiler. Statistical analysis was conducted using Minitab software, in line with Lahtinen et al. (2005), who emphasize the importance of statistical software packages for analyzing educational data and evaluating learning outcomes. Descriptive statistics were calculated, and the normality of data from classes A, B, and C was assessed using the non-parametric Kolmogorov-Smirnov (KS) test (Derrac et al., 2011). Additionally, differences in learning outcomes among the three classes were analyzed using the Mann-Whitney test, a non-parametric method suitable for comparing multiple groups with non-normal distributions (Peng et al., 2019).

RESULTS AND DISCUSSION

Results

The research data consists of final grades in computer programming classes A, B, and C. After calculating the mean and standard deviation, the results are presented in Table 2.

Table 2. Descriptive Statistics: Class A, B, C

Variable	N	N*	Mean	SE Mean	StDev	Minimum	Q1	Median	Q3	Maximum
Class A	20	0	83.25	1.22	5.45	75.00	80.00	80.00	90.00	90.00
Class B	27	0	80.37	1.22	6.34	65.00	80.00	80.00	85.00	90.00
Class C	18	0	81.94	1.72	7.30	70.00	75.00	80.00	90.00	90.00

The provided dataset on table 2 furnishes a descriptive statistical overview concerning the performance or outcomes across three distinct classes, denoted as class A, class B, and class C. Class A exhibits the highest mean score, standing at 83.25, succeeded by class C with an average of 81.94, and class B trailing with a mean of 80.37. Nonetheless, the highest standard deviation (StDev) is recorded in class C (7.30), indicating a greater degree of dispersion within the dataset. First quartile (Q1) and median values appear uniform across all classes, while the third quartile (Q3) tends to vary. This suggests variance in data distribution among the classes. However, further analysis is warranted to elucidate significant disparities between the classes, including potential factors influencing performance or outcomes within each class. The grades of Computer Programming classes A, B, and C were then assessed for normality, and the results are presented in Figures 3, 4, and 5.

The normality test for class A data in Figure 3 was conducted using the Kolmogorov-Smirnov (KS) test, yielding a test statistic of 0.325 and a p-value less than 0.010. Consequently, it can be concluded that the data does not follow a normal distribution at the 0.01 significance level.

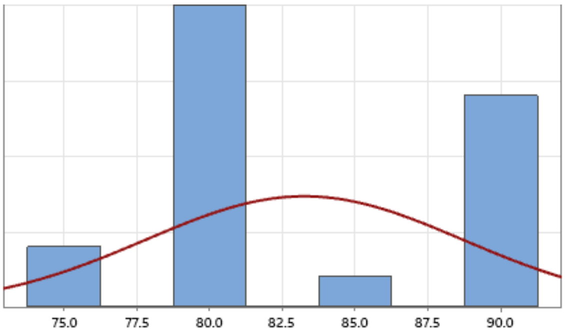

Figure 3. Summary Report for Class A

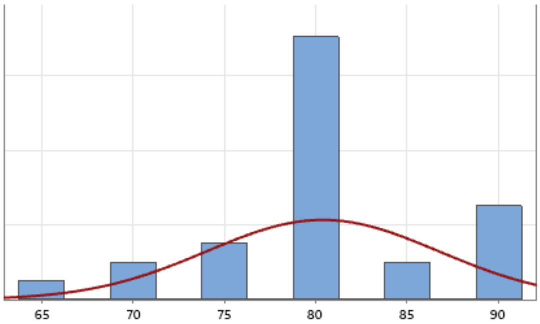

Figure 4. Summary Report for Class B

The normality test for class B data in Figure 4 was conducted using the Kolmogorov-Smirnov (KS) test, resulting in a test statistic of 0.264 and a p-value less than 0.010. This indicates that the data does not follow a normal distribution at the 0.01 significance level. As with the previous analysis on figure 1, a p-value smaller than the chosen significance level (0.010) leads to the rejection of the null hypothesis, which posits that the data is drawn from a normal distribution. Therefore, the final conclusion of the Kolmogorov-Smirnov test for the given data is that it does not follow a normal distribution at the 0.01 significance level.

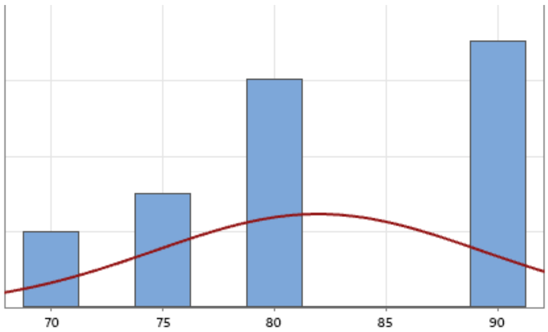

Figure 5. Summary Report for Class C

The normality test for Class C data in Figure 5 was conducted using the Kolmogorov-Smirnov (KS) test, resulting in a test statistic of 0.264 and a p-value less than 0.010. It can be concluded that the data does not follow a normal distribution at the 0.01 significance level. This indicates a significant difference between the observed data distribution and the normal distribution.

The Kolmogorov-Smirnov test results indicate that the data for all three classes, A, B, and C, do not follow a normal distribution. Consequently, non-parametric statistical tests are required for further analysis. The subsequent analysis aims to determine whether there are significant differences in the scores of the three classes after the computer programming course using the Pascal GDB compiler. This was tested using the Mann-Whitney U test, and the results are as follows.

After testing for normality, it was found that the grades of the Computer Programming classes were not normally distributed. Therefore, to test the differences in grades among the three classes, the non-parametric Mann-Whitney test was conducted, and the results are as follows.

The Mann-Whitney U test was employed to determine if there are significant differences between Class A and Class B. The null hypothesis (H_0: $\eta_1 - \eta_2 = 0$) posits that the medians of the two independent samples are equal, while the alternative hypothesis (H_1: $\eta_1 - \eta_2 \neq 0$) suggests a difference in medians. The test yielded a W-value of 541.50. Without adjustment for ties, the p-value was 0.189, and with adjustment for ties, the p-value was 0.154. These results indicate that there is no statistically significant difference between the medians of Class A and Class B at the conventional significance levels.

The Mann-Whitney U test was employed to assess if there are significant differences between Class A and Class C. The null hypothesis (H_0: $\eta_1 - \eta_2 = 0$) posits that the medians of the two independent samples are equal, while the alternative hypothesis (H_1: $\eta_1 - \eta_2 \neq 0$) suggests a difference in medians. The test yielded a W-value of 409.50. Without adjustment for ties, the p-value was 0.579, and with adjustment for ties, the p-value was 0.552. These results indicate that there is no statistically significant difference between the medians of Class A and Class C at the conventional significance levels.

The Mann-Whitney U test was applied to evaluate whether there are significant differences between Class B and Class C. The null hypothesis (H_0: $\eta_1 - \eta_2 = 0$) posits that the medians of the two independent samples are equal, while the alternative hypothesis (H_1: $\eta_1 - \eta_2 \neq 0$) suggests a difference in medians. The test yielded a W-value of 597.00. Without adjustment for ties, the p-value was 0.586, and with adjustment for ties, the p-value was 0.564. These results indicate that there is no statistically significant difference between the medians of Class B and Class C at the conventional significance levels.

Discussion

The dataset presented in Table 2 provides a comprehensive descriptive statistical overview of the performance across three distinct classes: Class A, Class B, and Class C. Class A achieved the highest mean score (83.25), followed by Class C (81.94), and Class B (80.37). However, Class C displayed the highest standard deviation (7.30), indicating a greater degree of variability in performance compared to the other classes. While the first quartile (Q1) and median values were consistent across all classes, variations were observed in the third quartile (Q3), suggesting differences in data distribution among the classes.

To further investigate these differences, the Kolmogorov-Smirnov (KS) test was employed to assess the normality of the data distributions. The KS test results for Class A, Class B, and Class C all indicated significant deviations from normality at the 0.01 significance level, necessitating the use of non-parametric statistical tests for subsequent analysis.

To determine if there were significant differences in performance between the classes following a computer programming course using the Pascal GDB compiler, the Mann-Whitney U test was conducted. Comparing Class A and Class B, the test yielded a W-value of 541.50, with p-values of 0.189 (not adjusted for ties) and 0.154 (adjusted for ties), indicating no statistically significant difference between the medians. Similarly, the comparison between Class A and Class C resulted in a W-value of 409.50, with p-values of 0.579 (not adjusted for ties) and 0.552 (adjusted for ties), again showing no significant difference. Lastly, the comparison between Class B and Class C produced a W-value of 597.00, with p-values of 0.586 (not adjusted for ties) and 0.564 (adjusted for ties), indicating no statistically significant difference between these medians.

Despite the observed differences in mean scores and variability, the Mann-Whitney U test results suggest that there are no significant differences in the performance of the three classes following the programming course.

CONCLUSION

The analysis of the performance data for Class A, Class B, and Class C revealed notable differences in mean scores and variability. Class A had the highest average score, while Class C exhibited the greatest variability. However, the Kolmogorov-Smirnov tests confirmed that the data distributions for all three classes significantly deviated from normality, necessitating the use of non-parametric statistical methods.

Subsequent application of the Mann-Whitney U test found no statistically significant differences between the medians of any pair of classes (class A vs. class B, class A vs. class C, and class B vs. class C) at conventional significance levels. These findings suggest that, despite the observed variations in descriptive statistics, the performance outcomes after the computer programming course using the Pascal GDB compiler were not significantly different across the three classes.

REFERENCES

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation, 1(1), 3-18. https://doi.org/10.1016/j.swevo.2011.02.002

Gomes, A., & Mendes, A. J. (2007). Learning to program - difficulties and solutions. In International Conference on Engineering Education (ICEE-2007), Coimbra, Portugal.

Harimurti, Rina, Ekohariadi, Munoto, Asto B I Gusti Putu, Winanti,Elizabeth Titiek 2019. Analysis Of Programming Skills Concept In Developing Problem Solving Skills. Jurnal Pendidikan Teknologi dan Kejuruan, Vol. 25, No. 1, May 2019, 43-51. DOI: 10.21831/jptk.v25i1.22638

Khairiah, Irmayati.,2023. Algoritma Pemrograman : Studi Pustaka Pemahaman Algoritma Pemrograman. Konstanta : Jurnal Matematika dan Ilmu Pengetahuan Alam,Vol. 1, No. 4 Desember 2023, DOI : https://doi.org/10.59581/konstanta.v1i4.1673

Krishnamurthi, S., & Fisler, K. (2019). Programming paradigms and beyond. In The Cambridge Handbook of Computing Education Research (pp. 182-226). Cambridge University Press. https://doi.org/10.1017/9781108654555.006

Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. ACM SIGCSE Bulletin, 37(3), 14-18. https://doi.org/10.1145/1151954.1067453

Munandar, H., Sutrio, S., & Taufik, M. (2018). Pengaruh Model Pembelajaran Berbasis Masalah Berbantuan Media Animasi Terhadap Kemampuan Berpikir Kritis dan Hasil Belajar Fisika Siswa SMAN 5 Mataram Tahun Ajaran 2016/2017. Jurnal Pendidikan Fisika Dan Teknologi, 4(1), 111–120. https://doi.org/10.29303/jpft.v4i1.526

Peng, C. Y. J., Harwell, M., Liou, S. M., & Ehman, L. H. (2019). Advances in missing data methods and implications for educational research. In M.J.R. David (Ed.), Quantitative Methods and Data Analysis in Education Research (pp. 1-31). University of Kansas. https://doi.org/10.7936/AHVJ-1883

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. Computer Science Education, 13(2), 137-172.

Sherin, B. L. (2001). How students understand physics equations. Cognition and Instruction, 19(4), 479–541. https://doi.org/10.1207/S1532690XCI1904_3

Zuhdi, M., Taufik, M., Sutrio, S., & Ayub, S. (2019). Pemodelan Sesar Pasca Gempa Dengan Inversi Data Gravitasi Time Lapse. Jurnal Pendidikan Fisika Dan Teknologi, 5(1), 156–161. https://doi.org/10.29303/jpft.v5i1.1109