Journal on Mathematics Education
Volume 15, No. 2, 2024, pp. 495-516

Elevating student engagement and academic performance: A quantitative analysis of Python programming integration in the *Merdeka Belajar* curriculum

Damar Rais[1,*], Zhao Xuezhi[2]

[1]Mathematics Education, Universitas PGRI Yogyakarta, Yogyakarta, Indonesia
[2]School of Mathematical Sciences, Capital Normal University, Beijing, China
*Correspondence: damar2ais@gmail.com

Received: 31 December 2023 | Revised: 5 February 2024 | Accepted: 22 February 2024 | Published Online: 9 March 2024

Abstract

Python programming is widely employed in educational institutions worldwide. Within the *Merdeka Belajar* curriculum context, this programming is recognized as a suitable vehicle for mathematics instruction, significantly influencing students' motivation and learning outcomes, particularly following periods of educational hiatus. This study examines the effectiveness of Python programming in promoting heightened learning outcomes by examining the intricate relationship between student motivation and learning. The study uses quantitative research methodologies to evaluate student learning facilitated through Python programming, encompassing problem-solving assessments and the administration of motivation questionnaires. By engaging in coding practices, students understand the symbols they manipulate, facilitating their ability to juxtapose data derived from mathematical modeling with the resultant programming output. When disparities arise, students are empowered to reassess their work, fostering a more profound comprehension of the subject matter. These exercises serve to augment students' capacity to retain and process information within memory. Furthermore, students demonstrate a favorable disposition, exhibiting persistence in resolving programming challenges by meticulously analyzing error outputs, particularly those pertaining to TypeErrors. Encouraging students to confront errors through thoroughly examining error output manifestations engenders an efficacious learning paradigm. This research proffers invaluable insights for educational institutions contemplating the integration of Python programming as an instructional adjunct.

Keywords: Learning Motivation, *Merdeka Belajar* Curriculum, Process Information, Python Programming

How to Cite: Rais, D., & Xuezhi, Z. (2024). Elevating student engagement and academic performance: A quantitative analysis of Python programming integration in the *Merdeka Belajar* curriculum. *Journal on Mathematics Education, 15*(2), 495-516. http://doi.org/10.22342/jme.v15i2.pp495-516

An educator's role encompasses the imperative task of equipping students with the necessary understanding of technology's manifold benefits, transcending the confines of conventional classroom learning. By instilling this knowledge, students are empowered to navigate and surmount challenges encountered in various facets of life. Embracing technology-based learning methodologies, as elucidated by Mhlongo et al. (2023) and Alyoussef (2022), necessitates a proactive adaptation to contemporary advancements, thereby ensuring alignment with the evolving educational landscape. Notably, students transitioning back to face-to-face learning environments following the pandemic-induced hiatus are confronted with the palpable repercussions of learning disruption, commonly referred to as learning loss, as expounded by Tilak (2021), Jakubowski et al. (2023), and Bertoletti et al. (2023). This phenomenon denotes a regression in academic proficiency attributable to prolonged disruptions in the learning continuum, leading to diminished student competencies. The manifestation of learning loss underscores a departure from historical educational progress trajectories, as highlighted by Hevia et al. (2021) and Donnelly and Patrinos (2022), necessitating concerted efforts to mitigate its adverse effects.

Learning loss ensues from a gradual erosion of knowledge and skills within students' academic development, arising from prolonged interruptions in the educational process, such as extended school closures, absenteeism, or ineffective pedagogical practices, particularly amid the pandemic. Transitioning to post-pandemic face-to-face learning environments inevitably presents significant hurdles characterized by students grappling with learning difficulties. As elucidated by Gül and Köse (2018), these challenges stem from various factors, including limited prior knowledge, misconceptions, diminished motivation, and apprehensions regarding the perceived utility of the subject matter.

Moreover, Abdulsahib (2021) underscores many factors contributing to students' learning challenges, ranging from inadequacies in foundational mathematical comprehension, impaired visual and symbolic cognition, and linguistic impediments to deficits in memory retention and information processing. Additionally, prevalent issues such as waning motivation, deficient mathematical grounding, misconceptions about the complexity of mathematics, excessive reliance on smartphones during remote study, passive engagement during lessons, and struggles in deciphering mathematical symbols further exacerbate the learning milieu. Recognizing the pivotal role of mathematics education, as expounded by Suherman and Vidákovich (2022), Firdaus et al. (2015), and Ramli et al. (2019), in nurturing students' cognitive faculties, including logical, analytical, systematic, critical, and creative thinking skills, underscores the urgency of addressing these challenges through targeted interventions and pedagogical innovations.

The phenomenon of learning loss presents educators with a strategic opportunity to integrate technology into pedagogical practices, notably harnessing the ubiquitous smartphone platform through Python programming to enhance mathematical learning experiences. This choice stems from Python's reputation for accessibility, catering to both novices and seasoned programmers alike, as attested by Ling et al. (2021), and its widespread adoption within educational contexts, as evidenced by Kuriki (2021), Caccavale et al. (2023), and Kozakai et al. (2022). Python's efficacy in bolstering students' problem-solving acumen, as demonstrated by Lee and Chung (2019), can be attributed to its iterative problem-solving approach, encompassing problem analysis, algorithm development, coding, testing, and debugging, complemented by real-time error feedback for syntax corrections.

Within this framework, the instructional focus pivots towards nurturing student competence and fostering collaborative learning environments conducive to programming comprehension. Thus, integrating Python programming into mathematics instruction promises to improve student learning outcomes. To redress learning loss, the Ministry of Education has implemented the *Merdeka Belajar* curriculum, denoting "Independent Learning," which allows educators to craft classroom experiences leveraging technological innovations to cultivate holistic learning outcomes encompassing soft skills and character development.

Educators can effectively address student challenges by devising innovative pedagogical designs incorporating Python programming. Furthermore, integrating Python programming into mathematics instruction not only epitomizes pedagogical innovation but also serves as a potent vehicle for nurturing students' programming and critical thinking proficiencies, as Bao et al. (2021) underscored.

The Minister of Education, in this case, the Ministry of Research and Technology (Isaeni & Nugraha, 2022), is highly conscious that the presence of technology in learning is essential. Python programming is considered appropriate for studying mathematics in the *Merdeka Belajar* curriculum. Students learn Python programming by first understanding mathematical concepts, then being introduced to the basics of programming and creating basic programming ranging from basic math operations to coding for the material studied, specifically the system of linear equations in three-variable (SPLTV).

Python programming is a popular programming language in 2022, so this programming language is widely used as a companion in the learning process (Bascuñana et al., 2023; Koupritzioti & Xinogalos, 2020; Bao, 2021). In addition, Python programming is widely applied in artificial intelligence and quantum computing because of its easy learning curve and great power (Auccahuasi et al., 2018), which has many libraries in the math module that can facilitate coding related to mathematical concepts. In the design of this research, Python programming can be used on smartphones called Pydroid version 3 or Pydroid 3-IDE. Students are led to a visual-logical approach to understanding problem situations (Carević et al., 2019) and concepts from the material that has been learned using Python. The use of technology in education also supports government programs through the *Merdeka Belajar* curriculum. Students learning activities to actively use modern technology are of utmost importance in contemporary education today (Doruk et al., 2013).

Such learner-centric instructional approaches offer a unique opportunity to tailor the learning trajectory to students' individual interests, aptitudes, learning preferences, and existing knowledge base, as delineated by Schacter (1999). The insights gleaned from this research endeavor serve as a potent catalyst for nurturing students' cognitive faculties and cultivating programming thinking skills in mathematics education through Python programming. This holistic approach ensures sustained retention of foundational concepts and theories essential for proficient problem-solving, thereby positively impacting students' mathematical achievement.

By fostering the development of mathematical acumen, logical reasoning, and programming prowess, students are equipped with enhanced cognitive abilities and adeptness in utilizing programming tools, as elucidated by Auccahuasi et al. (2018). The proposed implementation endeavors to elucidate the efficacy of integrating Python programming as a pedagogical tool in augmenting students' arithmetic learning outcomes.

When delving into programming languages, the acquisition of coding skills becomes paramount. An integral aspect of this learning process involves encountering and rectifying errors that occur during code creation. “Error” messages, such as those indicating syntax errors or runtime issues, provide invaluable insights into the execution flow, variable manipulations, and logic constructs inherent within the codebase. This instructional paradigm not only familiarizes students with fundamental programming concepts but also instills problem-solving acumen, as elucidated by Karnalim and Ayub (2017).

The provision of error feedback plays a pivotal role in shaping students' engagement with Python programming, particularly novices embarking on their coding journey. Encountering a TypeError, for instance, prompts students to reevaluate their code structure, syntax usage, and logical reasoning. Common errors observed among novice programmers encompass mismanagement of letter case conventions, ambiguous syntax constructs, and misuse of programming symbols. Upon rectification of errors, students are rewarded with the manifestation of mathematical models and corresponding output, thereby reinforcing their understanding and application of programming principles.

Moreover, the transfer of enthusiasm and mastery from proficient learners to peers grappling with coding intricacies fosters a contagion of excitement for learning new concepts within the classroom environment. Motivation emerges as a cornerstone of the learning process, alleviating cognitive burdens associated with grappling with unfamiliar concepts, fostering systematic thinking, and unlocking latent problem-solving abilities.

Student learning activities facilitated through Python programming can be meticulously documented utilizing screen capture or smartphone screenshot applications. These tools enable the comprehensive monitoring of students' engagement with various tasks, including labeling, symbol notation, coding sequences, and other pertinent activities. The resultant video recordings or image compilations give educators a detailed glimpse into students' work processes, facilitating targeted feedback and instructional interventions. As posited by Fonseca et al. (2016), the act of teacher monitoring exerts a profound positive impact on students' learning outcomes, fostering heightened performance and comprehension across diverse subject areas.

Drawing from many relevant studies, it becomes evident that integrating Python programming into instructional frameworks augments students' mathematical proficiency, irrespective of their prior familiarity with the programming language. Therefore, this research endeavors to elucidate the nexus between Python programming and enhanced student learning motivation, as noted by Ling et al. (2021) and Takefuji (2023). Through rigorous examination and analysis, this study aims to discern the efficacy of Python programming as a catalyst for bolstering student engagement and enthusiasm toward mathematical learning endeavors.

This research initiative has been undertaken across multiple countries where programming languages have become integral to school curricula. Consequently, the study was conducted within several educational institutions situated in Pekanbaru City, which have embraced the innovative *Merdeka Belajar* curriculum. Notably, this research represents a pioneering endeavor within the high school education sector of Pekanbaru City, marking a departure from prior studies in the region.

The absence of previous research endeavors exploring Python programming's impact on high school education in Pekanbaru City has spurred a profound motivation to introduce a novel learning experience centered around Python programming. This initiative is propelled by the conviction that such an intervention can significantly enhance student learning outcomes within the classroom setting. By filling this research gap, the study aims to shed light on the efficacy of Python programming as a transformative educational tool, particularly within the context of high school education in Pekanbaru City.

This research underscores the necessity of integrating educational technology within school settings to augment students' mathematical proficiency. Central to this objective is deploying smartphone applications directly pertinent to students' academic pursuits. Moreover, the study seeks to probe the impact of Python programming on students' motivation levels and mathematical learning outcomes within the classroom milieu.

The research aims to engender heightened student engagement and enthusiasm toward mathematical learning endeavors by introducing a novel learning paradigm. The experimentation with Python programming on smartphones as a pedagogical tool represents a pioneering effort to leverage technology's allure in fostering mathematical comprehension. This initiative is poised to captivate students' interest in mathematics, bolster their learning motivation, and positively influence their mathematical learning outcomes. Harnessing one of the most widely adopted programming languages as a companion for mathematical learning endeavors, this research offers viable solutions to these educational challenges.

METHOD

This study adopts a quantitative approach, delineating three primary variables within its framework. These variables include mathematics learning outcomes, serving as the dependent variable whose status will be influenced by the independent variable—learning facilitated through Python programming installed on students' smartphones. Additionally, the study accounts for the control variable of students' learning motivation (refer to Table 1 for a comprehensive overview of the variables).

Table 1. The Variables Used in this Study

Variable	Type of Variable	Description of Variable
Learning outcome	Ordinal	The result of students' examination in Linear Equation in three-variable
Learning Motivation	Ordinal	Students' motivation in learning (1 = Very Disagreement, 2 = Disagreement, 3 = Agreement, 4 = Very Agreement)
Python Programming	Ordinal	Students' response in use Python Programming (1 = Very Dissatisfied, 2 = Dissatisfied, 3 = Satisfied, 4 = Very Satisfied

Subject Characteristics

This research was conducted at grade X high schools implementing the *Merdeka Belajar* curriculum. Considerations in selecting samples for this study are based on teachers' experience in managing the class and adapting the *Merdeka Belajar* curriculum because in this study, there are only four schools that implement it, and three of the four schools are the population of the sample in this study, with students having homogeneous character and ability related to the homogeneous character and ability of students. The learning process is designed with the teacher as a learning facilitator for students (41 men and 51 women) and then transformed using the Pydroid 3 IDE.

We conduct math learning using Python programming that has been downloaded via a smartphone. The name of the application that represents Python programming on smartphones is Pydroid 3 IDE (logo: Pydroid 3 - IDE for Python 3). The following is the learning design (Rais & Xuezhi, 2023) conducted during the research in this study.

In the introduction to the topic, the teacher gives the students an illustration of the subject matter. Students recall their memory about a system of linear equations. Brainstorming to reinforce students' basic concepts. Then, through the Pydroid guidebook, students start trying to execute simple math operation programming. In the question phase, the teacher gives questions based on the cognitive domains of analysis, evaluation, and creation that are done individually in their Exercise books. Facilitator of the students' activities phase, to help students who are struggling, the teacher probes for clarification on any subjects that are unclear in questions. Based on the mathematical model, students create SPLSV syntax by using "replace ()" to perform operations from the grouped variables and symbolize "z.real, -z.imag" as a rational fraction to find the solution of SPLSV. Creating SPLDV and SPLTV syntax, students activate the Pydroid Library for sympy to symbolize variables x, y, and z or other variables simultaneously, which they use when creating mathematical models. The teacher sees student progress and facilitates if there are students who ask questions. After the students finish the programming, they execute the program. When the output has an error, the teacher instructs the students to understand the redaction, be careful, and check the coding. Presentation in a manual way phase: Students present their answers and have a brief discussion if there are students who ask questions. Then, all students look at the program output for conformity with the results of the student's answers in the exercise book. All student program activities are sent to the class group through screen capture or record files. Here is the students' activity through Python programming (see Figure 1).

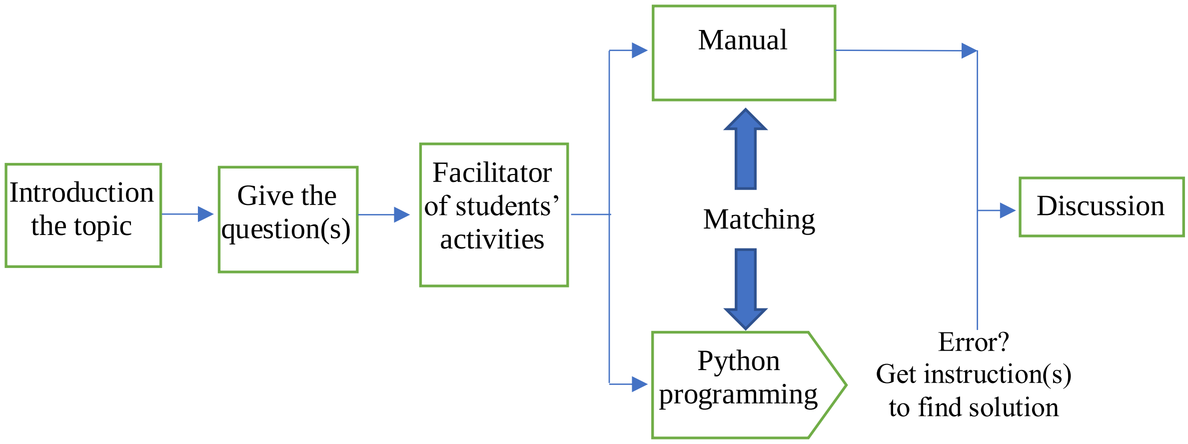

Figure 1. Design Learning Mathematics with Python Programming

Data Sources and Research Data

The data sources in this study were students who were the research subjects. The data source produced data in the form of written answers about the system of linear equations in three-variable, screen-capture, or recording of their works through Pydroid 3-IDE on smartphones (Rais & Xuezhi, 2023). Conduct learning activities for four meetings plus one meeting to test learning outcomes, and then collect students' motivation questionnaires through learning using Python programming.

Data Analysis Technique

The data in this study were analyzed using non-parametric methods. Data from questionnaires, Python programming assignments, and student math learning results were evaluated. The method for calculating learning motivation and the usage of Python programming using the formula (Leifheit et al., 2019) and learning outcomes with the evaluation framework (see Figure 2). This non-parametric analysis replaces data processing that cannot be done parametrically (Frunza, 2016) and does not require prerequisite tests. The Mann-Whitney U Test measured significant differences in the mean and median of math learning outcomes. The Wilcoxon Signed Rank Test was used to measure learning outcomes on motivation and the use of Python programming. This test produces data in descriptive statistics, pretest and posttest rankings, and the Wilcoxon Signed Rank Test results, which results in hypothesis testing with significance level $\alpha = 0.05$.

RESULTS AND DISCUSSION

The primary objective of this investigation is to explore the relationship between students' motivation levels and their learning outcomes within the realm of Python programming. The ensuing discourse delineates the analytical insights organized in alignment with the research inquiries. To introduce programming concepts effectively to students with limited prior experience, it is imperative to utilize straightforward activities that serve as foundational frameworks for comprehending programming principles. Presented herewith is the inaugural activity meticulously designed to acquaint students with Python programming.

The initial phase of the activity entails students revisiting fundamental mathematical operations such as addition, subtraction, multiplication, division, modulo, and exponentiation. Subsequently, students are prompted to assign random symbols as placeholders for these operations within a programming context.

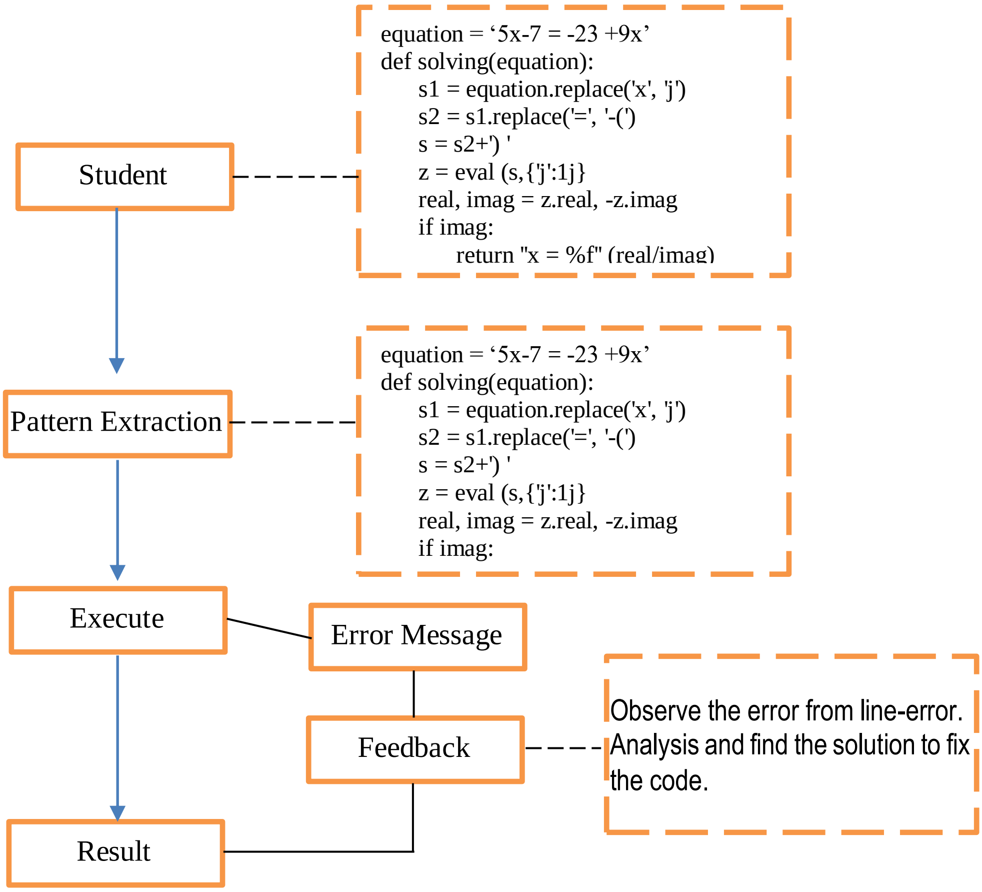

Figure 2. Design Learning Mathematics with Python Programming

They are then tasked with executing commands corresponding to the mathematical operations they've input, thereby generating the desired output (refer to Figure 3 for a visual depiction).

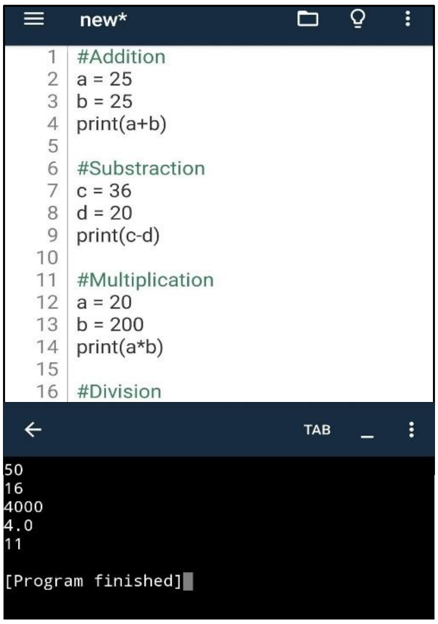

Figure 3. Trial Operational Mathematics on Pydroid 3-IDE

Upon grasping these rudimentary commands, students' progress to a more advanced stage wherein they are presented with a case study involving systems of linear equations with a single variable. Here, students are prompted to recall the general form of such equations. The teacher and students collaboratively identify coefficients, variables, and constants based on the provided example. This collaborative exercise serves to deepen students' comprehension of algebraic concepts within the context of Python programming. After students understand this basic concept, the teacher guides them in making a command call using replace() on Pydroid 3-IDE.

```
new*
equation = '5x - 7 = 9x - 23'

def solve(equation):
  s1 = equation.replace('x', 'j')
  s2 = s1.replace('=','-(')
  s = s2+')'
  z = eval(s, {'j': 1j})
  real, imag = z.real, -z.imag

  if imag:
    return"x = %f" %(real/imag)
  else:
    if real:
      return "has no solutions"
    else:
      return "infinite solutions"
print(solve(equation))
```

Figure 4. Student activities on the System of Linear Variable One-Variable

The way to work using this programming is the same as the manual way, but because it is related to the coding command (see Figure 4),

equation = "5x – 7 = 9x - 23

step 1: use replace() in python to replace "=" with "-(" and replace "x" with "j".

if: 5x - 7 = 9x – 23

then after using replace() it becomes:

5j - 7 – (9j – 23

step 2: then the string is added with "+)" to complete the statement.

5j - 7 – (9j – 23)

step 3: then {"j" : 1j} has successfully converted the equation into a format that is easily evaluated by the eval() function. At this stage, all Steps are evaluated in X or imaginary forms.

(16 – 4j)

step 4: Then, the evaluated expression is broken into real and imaginary parts. If the imaginary part exists or x is true and is not zero, the answer is printed; otherwise, if the imaginary part is 0 and the real part is true, there is no solution, or there are infinite solutions. Therefore, we get the

Output: j = 4 in other words the result of

5x – 7 = 9x – 23

x = 4

The solution to the problem is 4.

In this activity, students feel burdened by understanding and running coding in programming. However, the teacher facilitates students who experience problems in coding and gives instructions to students to teach programming. The next lesson is the system of linear equations of two variables. Students understand the system of linear equations of two variables (SPLDV) by solving it through elimination, substitution, and mixed methods (elimination and substitution). The teacher gives an example problem based on the general form, and students complete the given data. The following are the results of students' answers, as shown in Figure 5.

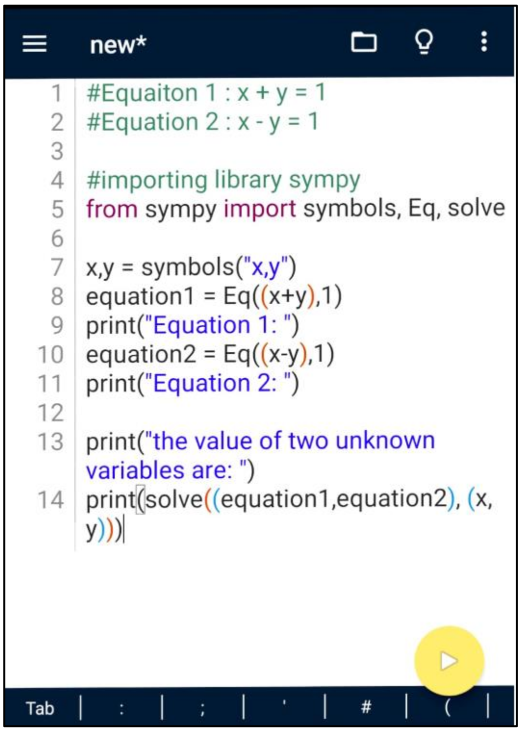

Figure 5. Student activities on the System of Linear Variable Two-Variable

The problem given is to determine the value of variables x and y. First of all, students do it manually, which they can write in their math notebooks. After students understand the basic concept, they are directed to use a library in the form of SymPy, which can be installed in the Pydroid 3 IDE application. By activating SymPy, students activate variables (in this case, using the x and y variables) that refer to the equations in the two-variable general form they created. Then, students input the equation starting with eq1 (representation of the first) and eq2 (representation of the second). This section was created to remember information on SPLDV lessons that students have studied manually. By making eq1 and eq2 directly, students will go to the equation they assumed at the beginning. In the process of absorbing information into memory, it still refers to two variables; later in the output, you will see the equation that has been made. To find out the solution to the equation that has been created, solve ((eq1, eq2), (x,y)) is used, which refers to the command to find a solution to the input equation involving the variables x and y. In the results section, students can flash back the input they have done in the program.

For further work, students do the same for a system of linear equations in three-variable (SPLTV). The teacher challenges students by working on questions in the student handbook with one solution, no solution, and many solutions. Then, the teacher takes one SPLTV question, which students do manually and by writing code in programming (see Figure 6).

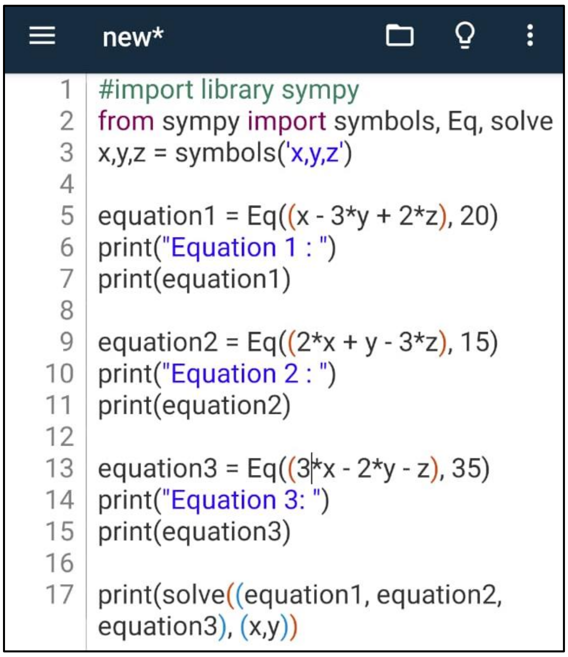

Figure 6. Student activities on the System of Linear Variable Three-Variable

Students are given a problem found in the math book (for students). The SPLTV problem instructs to determine the value of the variables x, y, and z, respectively, from the equations 3x + 2y + z = 390, x + 3y + 2z = 460, and 2x - z = 0. First of all, students work manually. Students create a mathematical model of the given story problem using the variables x, y, and z. and find the solution for each variable. After students finalize the answer and understand the concept of SPLTV, students are directed to solve it using Python programming. Similar to solving SPLDV material, students use a library in the form of SymPy. By activating SymPy, students can use variables (in this case, using x, y, and z variables) that refer to the SPLTV general form equation they created. Then, students enter the equations starting with eq1 (representation of the first equation), eq2 (representation of the second equation), and eq3 (representation of the third equation). This part is made to help students remember information from SPLTV material that they have learned manually. This is an attempt to process information into students' memory. Because the beginning of learning activates working memory, the repetition of learning activities (as an effort to manipulate extrinsic cognitive load) will facilitate students in processing information.

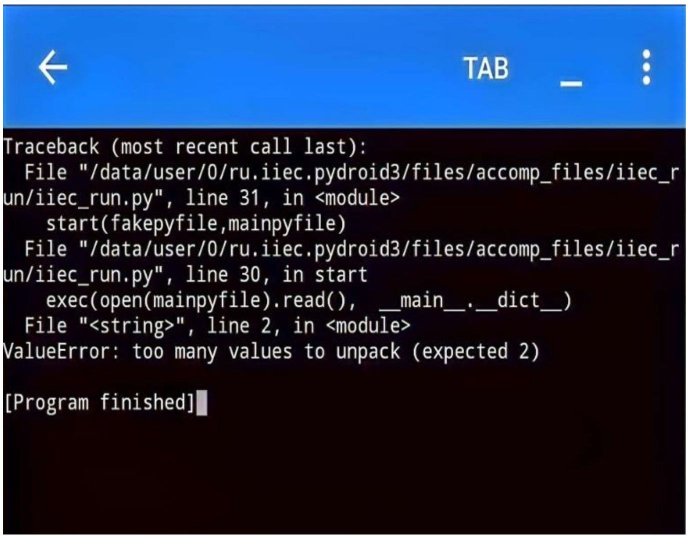

Figure 7. Error Syntax on Python Programming

In Figure 7, there is a syntax error, so it appears that "line 14 cannot determine the truth value of the relationship". Feedback from outputs like this makes it easier for students to find mistakes and learn from them. When students detect errors, instructions will display for the discovered errors. Students frequently make Errors such as replacing ('=', '-('). Students coding flow is not methodical since they forgot the replace () step to complete subtraction (e.g., 5x - 7 = 9x + 23, then 5x - 9x = 23 - (-7)). Then, on the screen of the student's phone, a Type Error editor will emerge, containing the part that is the source of the error. Students focus on the comment in TypeError and replace('=', '-('). Then, with other errors on line 14 (as shown in Figure 7), students only create variables partially.

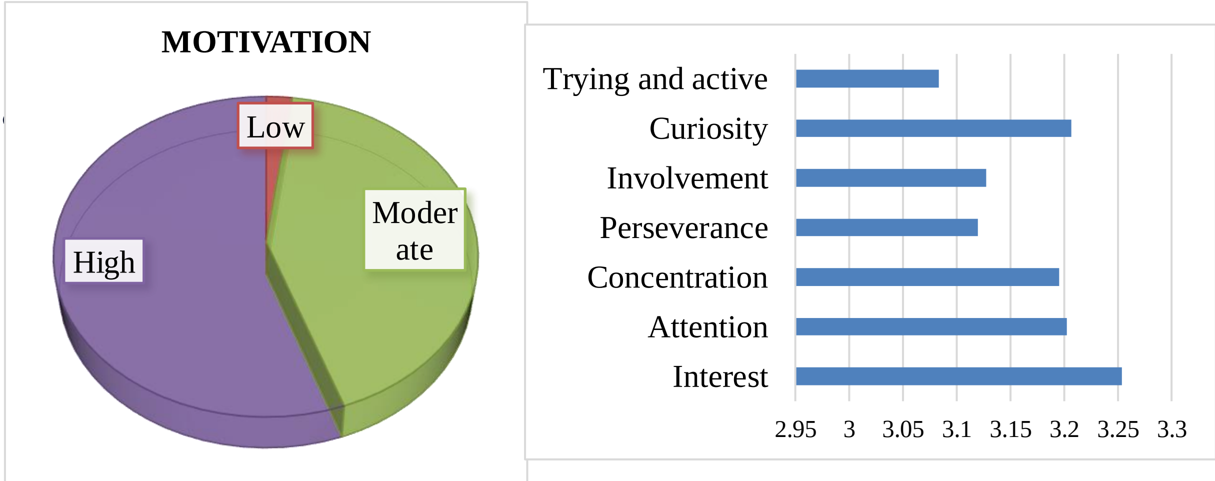

Figure 8. Student motivation in learning math using Python programming

For errors like these, students should have mentioned variable z on the code and constructed the variable z at the last code (in Figure 6), making it impossible to discover faults even after checking the work results using ValueError. The teacher and students checked from the beginning of syntax to the end, and it was discovered that the z variable did not appear to process SPLTV. The interaction of students with Pydroid worksheets and teachers provides a meaningful learning process for students to work systematically and thoroughly in the future.

This finding was influenced by students' motivation to learn mathematics with Pydroid (see Figure 8). Students' interest in the new learning design influenced their motivation to learn and try, and they actively had poor assessment because students found it challenging to correct error codes (if any). The intervention increased student motivation during the classroom learning process (see Table 2).

Table 2. Student learning motivation before and after intervention

	N	Minimum	Maximum	Mean	Std. Deviation
Post_Motivation	92	2.04	3.84	3.1691	.28198
Pre_Motivation	92	2.00	3.76	2.9548	.31397
Valid N (listwise)	92				

The rise in student motivation before and after experienced an appropriate increase. Student learning motivation belongs to the moderate level. Students' interest in using Python programming increased significantly, with a score of 3.25 (very high). Meanwhile, the use of Python programming in math learning is shown in Figure 9.

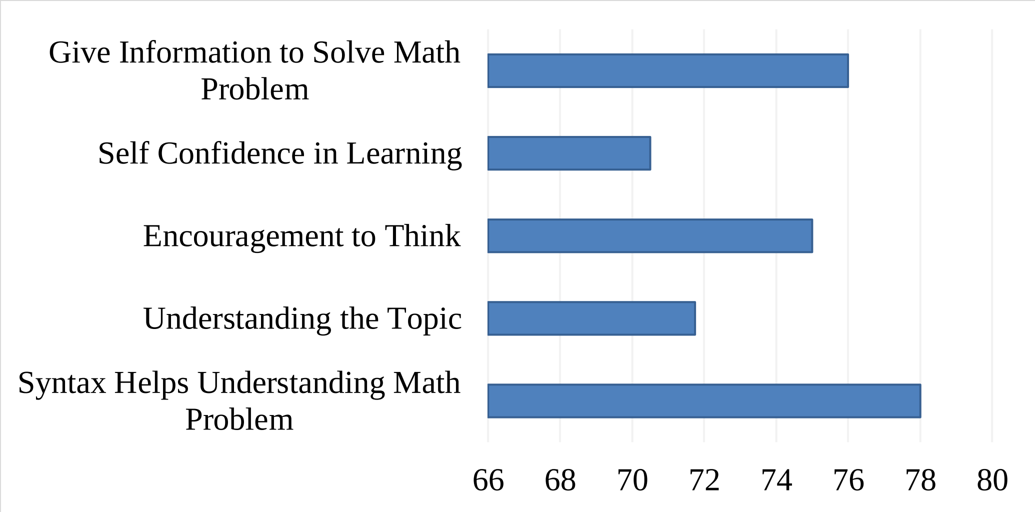

Figure 9. Student's Perception of Learning Python Programming

Students exhibit enhanced comprehension of the foundational concepts underlying systems of linear equations with three variables, including the concepts of coefficients, variables, equations, and constants. Throughout the learning process, students are encouraged to engage in critical thinking, share their perspectives, and participate actively in discussions while concurrently mastering Python programming.

However, in the context of learning mathematics through Python programming, students may initially need more confidence in independently utilizing Python programming tools, necessitating recourse to manuals, peer assistance, and textbooks. Despite efforts to foster independent learning, some students may still require occasional guidance and support to navigate through the complexities of the coding process.

Nevertheless, upon completing their programming tasks and attaining results that address posed questions, students demonstrate remarkable enthusiasm in reporting their findings to the teacher. They eagerly recount the aspects of the coding process that posed challenges, reflecting a proactive approach to overcoming obstacles and a genuine commitment to mastering Python programming within the context of mathematical learning.

In the Wilcoxon Signed Ranks Test conducted to examine the relationship between motivation and the utilization of Python programming in classroom learning, the obtained Wilcoxon's Z value for the comparison between Python programming and learning motivation reveals a significant difference before and after the treatment. Specifically, the calculated Z value is -4.619 with a p-value (Asymp. Sig 2 tailed) of 0.000. Given that the p-value is less than the predetermined significance level (α), which indicates statistical significance, it can be inferred that a substantial disparity exists between Python programming utilization and students' motivation to learn mathematics in school.

This research holds considerable value for students, particularly post-comprehension of Python programming for mathematical learning. Identifying and resolving procedural errors, particularly those pertaining to TypeErrors encountered in Pydroid 3-IDE, engender enhanced student motivation. Consequently, this influences the seamless integration of Python applications within mathematics classrooms, thereby fostering a conducive learning environment conducive to students' academic growth and proficiency.

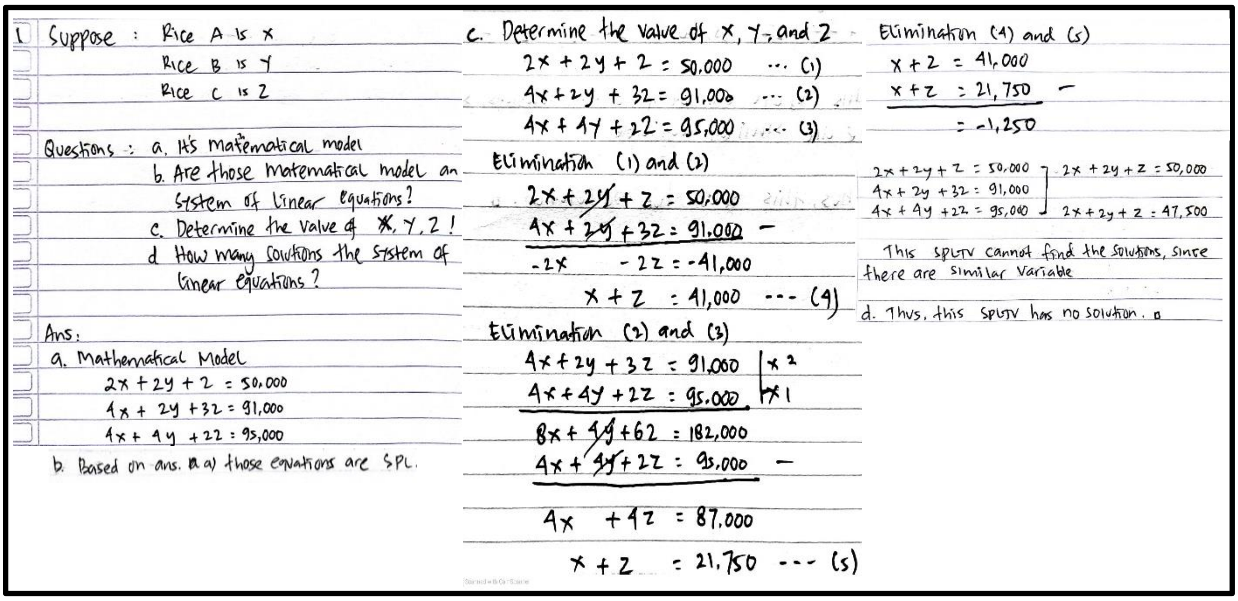

Figure 10. Student Answer Sheet

From Figure 10, students are asked to model the given problem, determine whether the model belongs to a system of linear equations, determine the value of each variable, and determine how many solutions. The results of the students' answers concluded that they had the skills to dissect the contents of the problem. Students can make mathematical models according to the data given and understand the basis of the system of linear equations that have been learned. In finding the solution to the question, students obtain a unique solution, with results that have no solution. Students understand that the existence of equations (4) and (5), if eliminated, will have no results, so the student in this question gets a total score of 10 points. This student's answer analysis is categorized as good, but the results will be more accurate if presented in a graph, so the graph of the solution (see Figure 11).

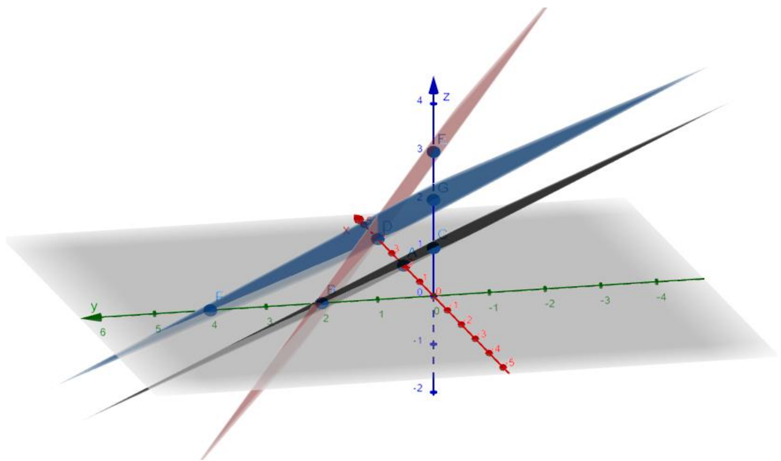

Figure 11. Graph-Solution

Mathematics learning activities using Python programming in the classroom increased 92 participants by 54.87%. The Mann-Whitney U test has Sig. 0.044. Because the p-value < α, there is a difference at a significant level of 5%. Based on the test results (see Table 3), it can be concluded that learning mathematics using Python programming in the classroom effectively improves student learning outcomes in the selected sample.

Table 3. Mann-Whitney U Test for Learning Outcome using Python Programming

Mann-Whitney U	**2234.000**
Z	-2.013
Asymp. Sig. (2-tailed)	0.044
a. Grouping Variable: Class	

The initial introduction of Python programming into mathematics learning poses a cognitive challenge for students, who perceive the learning effort as significantly more demanding than traditional methods. This augmented cognitive load arises from the simultaneous need to comprehend mathematical concepts and the intricacies of Python programming manually. To alleviate students' cognitive burden, a strategy is employed to manage the extrinsic cognitive load by focusing on core mathematical concepts essential for understanding systems of linear equations with three variables (SPLTV). Specifically, students are encouraged to grasp fundamental concepts such as variables, coefficients, and constants, which serve as foundational knowledge for SPLTV learning.

Throughout the instructional process, teachers and students collaboratively engage with story problems, constructing mathematical models that adhere to the general form of SPLTV equations. Although many students initially find it challenging to assimilate information presented as story problems, this activity mitigates intrinsic cognitive load. As noted by Sweller et al. (2019), students who are adept at regulating their information intake, understanding presentation formats, and discerning requisite actions are better positioned to achieve learning competencies. Therefore, this instructional approach not only facilitates comprehension of SPLTV concepts but also empowers students to navigate cognitive demands effectively, ultimately enhancing their learning outcomes.

Learning Outcomes on Motivation

Motivation is a significant determinant of students' actions, guiding their decisions regarding task prioritization and resource allocation (Nurmuiza et al., 2015). When students are imbued with a high level of intrinsic motivation, they are likelier to exhibit exceptional learning outcomes and strive for academic excellence (Filgona et al., 2020; Austin et al., 2018; Peng & Fu, 2021). Motivation catalyzes students to unlock their latent potential and embark on a journey of self-development and growth.

In the context of the novel learning experience introduced in this study—namely, utilizing smartphones for mathematics learning with Python programming—students exhibit heightened interest and attentiveness towards mathematics education. This innovative approach fosters a conducive environment wherein students are motivated to engage actively with Python programming, thereby enhancing their mathematical proficiency and problem-solving skills. This new learning experience empowers students to harness their academic capabilities and achieve remarkable learning outcomes by nurturing a sense of intrinsic motivation.

Indeed, learning motivation plays a pivotal role in shaping students' mathematical learning outcomes within the classroom setting. Examining questionnaire results from the students' perspective makes the profound impact of motivation on math learning evident. It is essential to underscore the significance of repetition as a fundamental learning aid, as repeated exposure to concepts fosters more profound understanding and mastery.

Whether intrinsic or extrinsic, motivation serves as a driving force that influences students' engagement, persistence, and, ultimately, academic achievements. Motivation can emanate from internal sources, such as personal goals, interests, and aspirations, as well as external factors, including supportive learning environments, encouragement from peers and educators, and recognition of accomplishments.

Therefore, fostering a motivating learning environment that encourages active participation, provides opportunities for success, and nurtures students' intrinsic interests is crucial for optimizing math learning outcomes. By acknowledging the multifaceted nature of motivation and its impact on learning, educators can effectively tailor instructional strategies to cultivate and sustain students' motivation, thereby fostering a conducive atmosphere for academic success.

In the integrated learning process, students engage with mathematics through textbooks, smartphones, and Python programming, creating program syntax relevant to the mathematical concepts. This approach fosters heightened motivation among students to comprehend and utilize Python programming tools actively. Initially, students may rely on collaborative learning in groups, seeking peer assistance, but they swiftly transition to independently utilizing Python programming. This shift underscores the learning process's efficacy and students' proficiency in programming. Overall, this multifaceted approach cultivates a dynamic educational environment wherein students explore mathematical concepts while honing their programming skills, leading to a deeper understanding of both disciplines.

Students' tenacity in designing appropriate and efficient algorithms to solve given mathematical problems helps students understand Python programming and improve learning outcomes (Papadakis, 2020; Kong et al., 2020; Hsiao et al., 2023). If students get an error result, students can solve the problem by paying attention to the TypeError on the work screen. Students are satisfied in learning Python programming because they can understand their work conventionally (using paper) by comparing the process and output results of the same program. This student activity trains them how to grasp ideas (Inguva et al., 2021) and solve mathematics problems systematically and logically.

Learning Outcomes for Python Programming

Python programming language training design is applied as learning because Python programming language is a relatively easy programming language for beginners in general. In addition, its syntax is easy to read and understand (Možina & Lazar, 2018; Sakharkar, 2023). When compared to other programming languages, using Python in learning has a significant impact on student learning outcomes (Wainer & Xavier, 2018), and through repeated training on assessment and coding, one relies on self-assistance and peer support to achieve effective learning outcomes.

Python programming allows for innovative teaching that can improve students' learning results with diverse ability backgrounds (Hsiao et al., 2023). Learning math using Python allows students and teachers to transfer the basic concepts learned (Tabet et al., 2016). Many students need help understanding the basics of programming, so they find it difficult to get assignments that require programming skills. In the development of technology, programming languages contribute to the world of education (i.e., mathematics) so that learning mathematics in schools can take advantage of Python programming that runs through smartphones (Fabic et al., 2018), which are close to students and continue to grow. The use of this smartphone is only aimed at learning mathematics by utilizing the Python programming language that the teacher has approved in the classroom. This can be an alternative for teachers in carrying out the mathematics learning process based on the *Merdeka Belajar* curriculum.

Learning Motivation towards the Use of Python Programming

Teaching programs are widely used in schools and universities (Santos et al., 2019). Anyone can learn to teach using Python without a basic understanding of programming. The Z test result in this study is -4,619 with a p-value of 0.000, and learning math in class using Pydroid significantly affects student learning motivation. This can be seen when students are first introduced to the Python programming language with Pydroid, namely by listening to the teacher demonstrate the use of Pydroid, students' willingness to follow the instructions given by the teacher, and students' enthusiasm to try. However, when students try to run coding related to SPLSV, they experience obstacles such as misspellings due to the use of programming languages that are identical to using a lot of English in the use of tools, not familiar with programming logic (Sentance & Csizmadia, 2017), long syntax, and they have to put a long focus to understand it. After students finish coding and executing, the results show "Error." This error section shows students' motivation to determine which part of their code is wrong. If students need help to solve the coding problem, some students ask the teacher and discuss it with friends.

Many students need help coding a system of linear equations of one variable using Python. Despite the difficulties, students can still follow and understand the basic concepts of the lesson. With an average questionnaire score of 79.12, students are well motivated to learn mathematics using Pydroid. Students' conditions when creating codes using Pydroid still really need guidance from fellow teachers to provide an understanding of the use of codes in SPLTV material.

Generally, the type of error students make when carrying out programming syntax is NameError, where students often make typos. Based on Saraswat et al. (2022), Python programming is case-sensitive. The program will treat the uppercase and lower-case letters differently. Hence, students must use the exact case according to the Syntax. The following error is Syntax Errors, where students use inappropriate indentation, erroneous keyword usage, or incorrect operator usage. Syntax errors prohibit the code from running (i.e., replace('=', '-(replace part of the library that functions to replace a specified phrase with another specified phrase). The identification of this error has an impact on the student's work system in solving mathematics must be coherent and systematic. If in doing replace, students must pay attention to the type of number in the coefficient of a variable. If the coefficient is positive, it will become negative if the replacement is done. And vice versa. When working with paper, some students still make mistakes in solving, such as 5x - 7 = 9x - 23; some students still make the solution into 5x - 9x = - 23 – (-7). The teacher providing an explanation of the replace () command benefits students in understanding the systematic derivative of solving mathematics coherently.

Despite making strides in grasping the coding logic related to the material, students in the experimental group still need guidance from teachers and peers to ensure the readability and accuracy of their code. Some students in this group encounter errors during program execution and need help interpreting the feedback provided. However, students have noted that error information displayed on Pydroid enhances their understanding of error locations and the runtime behavior of Python code, as emphasized by Karnalim and Ayub (2017). This acknowledgment underscores the importance of ongoing support and feedback mechanisms in facilitating students' learning journey with Python programming.

The findings of Rodrigues et al. (2021) indicate that learning experiences designed to enhance intrinsic motivation can significantly influence students' comprehension of programming, although the extent of this influence varies depending on students' prior familiarity with programming concepts. Balreira et al. (2022) similarly suggest a positive impact of programming on student motivation. Moreover, research results from Ling et al. (2021) and Pabón and Villegas (2019) demonstrate increased student motivation after implementing classroom learning utilizing Python programming, as evidenced by comparing pretest and posttest assessments.

The Effectiveness of Learning Mathematics Using Python Programming

Learning Python programming can indeed pose initial challenges for students, but with consistent exposure and practice, students can develop a solid understanding of its functionality. Integrating Python programming into student mathematics learning within the *Merdeka Belajar* curriculum not only aligns with the curriculum's objectives but also offers students an opportunity to engage in meaningful projects. While Karnalim and Ayub research (2017) suggests that Python programming may initially seem less intuitive for first-time users, students can navigate its intricacies with the aid of guidebooks and ongoing support. Furthermore, incorporating Python programming into mathematics learning enables students to enhance their mathematical competence, hone problem-solving skills, and cultivate systematic and logical thinking abilities, as evidenced by Fauzan et al. (2023). Leveraging Python programming on smartphones ensures optimal utilization of digital resources, thereby preparing students for careers in the digital age. Overall, Python programming is a valuable tool in fostering students' academic growth and equipping them with essential skills for future success.

CONCLUSION

Python programming has demonstrated its efficacy in bolstering student motivation, confidence in self-learning, and overall enthusiasm for learning, thereby leading to improved learning outcomes. As students embark on their programming journey, they begin to grasp the algorithms inherent in systems of linear equations with three variables (SPLTV) learning, facilitated by the introduction of the programming language. Progressively, students exhibit mastery of SPLTV learning, as evidenced by the analysis of pretest and posttest results. Their enthusiasm for learning is palpable, as observed through their active engagement and collaboration in deciphering the intricacies of Pydroid. Over time, students' intrinsic motivation drives their understanding of programming, albeit the extent of influence may vary based on their prior exposure to programming concepts.

In parallel, students are guided through manual instruction in SPLTV material, with periodic recollection sessions to reinforce previously learned concepts. Upon acquiring a solid understanding of SPLTV fundamentals, students transition to practical problem-solving tasks, attempting to code SPLTV using simplified algorithms. Subsequently, students meticulously assess the results of their programming efforts, comparing them with manual calculations. Discrepancies between the two outputs prompt students to revisit and verify their manual work steps, facilitating a comprehensive understanding of SPLTV concepts. This integrated approach underscores the symbiotic relationship between manual learning and programming, culminating in enriched learning experiences and enhanced comprehension of mathematical concepts.

Despite encountering initial difficulties in learning the coding algorithm, students undergo a process that fosters creativity in problem-solving and solution-finding. Emphasizing manual problem-solving techniques, students learn to read and analyze problems before devising manual solutions, which then serve as the basis for developing program syntax (i.e., solving on TypeError and re-checking student's worksheet). Upon encountering errors in their programming endeavors, students are tasked with analyzing error output comments and proposing improvements, thereby honing their attention to detail and coding proficiency. Notably, even students who have yet to gain prior programming experience can effectively learn Python programming, as its user-friendly features attract beginners and boost their confidence and satisfaction. In light of the positive results and feedback regarding integrating Python programming in mathematics learning, it is a practical and effective tool for enhancing motivation and improving mathematics learning outcomes in classroom settings.

This study acknowledges its limitations, including a limited sample size comprising three schools implementing the *Merdeka Belajar* curriculum. Each school represents a single class at the tenth-grade level in high school. These constraints are primarily due to limitations in sample size, time, and resources available for conducting the study. Despite these limitations, valuable insights were obtained regarding the potential integration of programming languages, particularly Python programming, into the *Merdeka Belajar* curriculum. Teachers' responses indicate a notable need for prior utilization of Python programming in Pekanbaru High School. However, they expressed openness to incorporating Python programming as a tool for learning mathematics or as a reference for introducing new classroom lessons. These findings underscore the potential for further exploring and implementing Python programming within the educational framework, highlighting future research and development avenues in this domain.

Adopting programming languages remains relatively novel for students, necessitating continuous learning and adaptation. Schools implementing this educational paradigm can facilitate students' programming endeavors by downloading resources such as Pydroid 3-IDE and associated libraries from the menu. Furthermore, educators can leverage additional resources available through platforms like freeCodeCamp.org or reach out via email to initiate discussions on advancing students' education and mathematical skills collaboratively. This learning framework offers a comprehensive approach to incorporating Python programming as an educational tool, providing institutions with valuable insights for considering its integration into their educational programs.

Acknowledgments

The authors would like to thank the Chinese Government Scholarship for funding this research, the teachers who gave their classes to be experimented with, and the students in the selected sample who willingly participated in this research.

Declarations

Author Contribution	:	DR: Conceptualization, Investigation, Formal Analysis, Writing Original Draft, Visualization, Writing-Review, and Editing ZX: Resources, Review, and visualization, Writing-Review, and Editing
Funding Statement	:	This paper was funded by the Chinese Government Scholarship (CGS).
Conflict of Interest	:	The authors declare no conflict of interest.

REFERENCES

Abdulsahib, R. H. (2021). Learning difficulties in mathematics and its relationship to cognitive failures among middle school students. *Ilkogretim Online*, *20*(6), 2291-2325. https://doi.org/10.17051/ilkonline.2021.06.211

Alyoussef, I. Y. (2022). Acceptance of a flipped classroom to improve university students' learning: An empirical study on the TAM model and the unified theory of acceptance and use of technology (UTAUT). *Heliyon*, *8*(12). https://doi.org/10.1016/j.heliyon.2022.e12529

Auccahuasi, W., Santiago, G. B., Núñez, E. O., & Sernaque, F. (2018, December). Interactive online tool as an instrument for learning mathematics through programming techniques, aimed at high school students. In *Proceedings of the 6th International conference on information technology: IoT and Smart City* (pp. 70-76). https://doi.org/10.1145/3301551.3301580

Austin, A. C., Hammond, N. B., Barrows, N., Gould, D. L., & Gould, I. R. (2018). Relating motivation and student outcomes in general organic chemistry. *Chemistry Education Research and Practice*, *19*(1), 331-341. https://doi.org/10.1039/C7RP00182G

Balreira, D. G., Silveira, T. L. D., & Wickboldt, J. A. (2022). Investigating the impact of adopting Python and C languages for introductory engineering programming courses. *Computer Applications in Engineering Education*, *31*(1), 47-62. https://doi.org/10.1002/cae.22570

Bascuñana, J., León, S., González-Miquel, M., González, E. J., & Ramírez, J. (2023). Impact of Jupyter Notebook as a Tool to enhance the learning process in Chemical Engineering Modules. *Education for Chemical Engineers*. https://doi.org/10.1016/j.ece.2023.06.001

Bao, Y., Wang, G., & Sun, Z. (2021). Exploration and Research on Integrating Programming Education into Junior Middle School Mathematics Classroom. In *Proceedings of CECNet 2021: The 11th International Conference on Electronics, Communications and Networks (CECNet), November 18-21, 2021* (Vol. 345, p. 356). IOS Press. https://doi.org/10.3233/FAIA210422

Bertoletti, A., Cannistrà, M., Soncin, M., & Agasisti, T. (2023). The heterogeneity of Covid-19 learning loss across Italian primary and middle schools. *Economics of Education Review*, *95*, 102435. https://doi.org/10.1016/j.econedurev.2023.102435

Caccavale, F., Gargalo, C. L., Gernaey, K. V., & Krühne, U. (2023). SPyCE: A structured and tailored series of Python courses for (bio) chemical engineers. *Education for Chemical Engineers*, *45*, 90-103. https://doi.org/10.1016/j.ece.2023.08.003

Carević, M. M., Petrović, M., & Denić, N. (2019). Figurative Numbers Contribution in Perceiving the Legality in Numerous Strings Tasks and Long-term Memory of Numerous Data. *EURASIA Journal of Mathematics, Science and Technology Education*, *15*(4), em1692. https://doi.org/10.29333/ejmste/103387

Donnelly, R., & Patrinos, H. A. (2022). Learning loss during Covid-19: An early systematic review. *Prospects*, *51*(4), 601-609. https://doi.org/10.1007/s11125-021-09582-6

Doruk, B. K., Aktümen, M., & Aytekin, C. (2013). Pre-service elementary mathematics teachers' opinions about using GeoGebra in mathematics education with reference to 'teaching practices'. *Teaching Mathematics and its Applications: An International Journal of the IMA*, *32*(3), 140-157. https://doi.org/10.1093/teamat/hrt009

Fabic, G. V. F., Mitrovic, A., & Neshatian, K. (2018). Investigating the effects of learning activities in a mobile Python tutor for targeting multiple coding skills. *Research and practice in technology enhanced learning*, *13*, 1-24. https://doi.org/10.1186/s41039-018-0092-x

Fauzan, F., Ansori, R. A. M., Dannur, M., Pratama, A., & Hairit, A. (2023). The Implementation of the Merdeka Curriculum (Independent Curriculum) in Strengthening Students' Character in Indonesia. *Aqlamuna: Journal of Educational Studies*, *1*(1), 136-155. https://doi.org/10.58223/aqlamuna.v1i1.237

Filgona, J., Sakiyo, J., Gwany, D. M., & Okoronka, A. U. (2020). Motivation in Learning. *Asian Journal of Education and Social Studies*, *10*(4), 16–37. https://doi.org/10.9734/ajess/2020/v10i430273

Firdaus, F., Kailani, I., Bakar, M. N. B., & Bakry, B. (2015). Developing critical thinking skills of students in mathematics learning. *Journal of Education and Learning (EduLearn)*, *9*(3), 226-236. https://doi.org/10.11591/edulearn.v9i3.1830

Frunza, M, C. (2016). Chapter 2F - Non-Parametric Techniques. *Academic Press*, *2016*, pp. 169-181. https://doi.org/10.1016/B978-0-12-804494-0.00012-7

Fonseca, N. G., Macedo, L., & Mendes, A. J. (2016, June). CodeInsights: Monitoring programming students' progress. In *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016* (pp. 375-382). https://doi.org/10.1145/2983468.2983492

Gül, Ş., & Köse, E. Ö. (2018). Öğretmen Adaylarının Protein Sentezine Yönelik Algıları: Öğrenme Güçlüğüne Karşı Önerilen Çözümler. *Erzincan Üniversitesi Eğitim Fakültesi Dergisi*, *20*(1), 237-250. https://doi.org/10.17556/erziefd.307083

Hevia, F, J., Vergara, L, S., Velásquez, A, D., & Calderón, D. (2021). Estimation of the fundamental learning loss and learning poverty related to COVID-19 pandemic in Mexico. *International Journal of Educational Development*, *88*(1), 102515. https://doi.org/10.1016/j.ijedudev.2021.102515

Hsiao, T. C., Chuang, Y. H., Chang, C. Y., Chen, T. L., Zhang, H. B., & Chang, J. C. (2023). Combining Building Block Process With Computational Thinking Improves Learning Outcomes of Python Programming With Peer Assessment. *SAGE Open*, *13*(4), 21582440231217715. https://doi.org/10.1177/21582440231217715

Inguva, P., Bhute, V. J., Cheng, T. N., & Walker, P. J. (2021). Introducing students to research codes: A short course on solving partial differential equations in Python. *Education for Chemical Engineers*, *36*, 1-11. https://doi.org/10.1016/j.ece.2021.01.011

Isaeni, N., & Nugraha, A. (2022). Teknologi Dalam Transformasi Pembelajaran Kurikulum Merdeka. *Direktorat Guru Pendidikan Dasar, Kemendikbudristek*. https://doi.org/10.4444/jisma.v2i6.736

Jakubowski, M., Gajderowicz, T., & Patrinos, H. A. (2023). Global learning loss in student achievement: First estimates using comparable reading scores. *Economics Letters*, *232*, 111313. https://doi.org/10.1016/j.econlet.2023.111313

Karnalim, O., & Ayub, M. (2017). The use of python tutor on programming laboratory session: Student perspectives. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 327-336. http://dx.doi.org/10.22219/kinetik.v2i4.442

Kong, Q., Siauw, T., & Bayen, A. (2020). *Python programming and numerical methods: A guide for engineers and scientists*. Academic Press. https://doi.org/10.1016/C2018-0-04165-1

Koupritzioti, D., & Xinogalos, S. (2020). PyDiophantus maze game: Play it to learn mathematics or implement it to learn game programming in Python. *Education and Information Technologies*, *25*(4), 2747–2764. https://doi.org/10.1007/s10639-019-10087-1

Kozakai, R., Kobayashi, T., Wenxuan, Z., & Watanabe, Y. (2022). Tendency Analysis of Python Programming Classes for Junior and Senior High School Students. *Procedia Computer Science*, *207*, 4603-4612. https://doi.org/10.1016/j.procs.2022.09.524

Kuriki, M. (2021). Using Python and Google Colab to teach undergraduate microeconomic theory. *International Review of Economics Education*, *38*, 100225. https://doi.org/10.1016/j.iree.2021.100225

Lee, D. Y., & Chung, J. I. (2019). The effects of middle school mathematical statistics area and Python programming STEAM instruction on problem solving ability and curriculum interest. *Journal of the Korea Academia-Industrial Cooperation Society*, *20*(4), 336-344. https://doi.org/10.5762/KAIS.2019.20.4.336

Leifheit, L., Tsarava, K., Moeller, K., Ostermann, K., Golle, J., Trautwein, U., & Ninaus, M. (2019). *Development of a Questionnaire on Self-concept, Motivational Beliefs, and Attitude Towards Programming. Proceedings of the 14th Workshop in Primary and Secondary Computing Education on - WiPSCE'19.* https://doi.org/10.1145/3361721.3361730

Ling, H. C., Hsiao, K. L., & Hsu, W. C. (2021). Can students' computer programming learning motivation and effectiveness be enhanced by learning python language? A multi-group analysis. *Frontiers in Psychology*, *11*, 600814. https://doi.org/10.3389/fpsyg.2020.600814

Mhlongo, S., Mbatha, K., Ramatsetse, B., & Dlamini, R. (2023). Challenges, opportunities, and prospects of adopting and using smart digital technologies in learning environments: An iterative review. *Heliyon*. https://doi.org/10.1016/j.heliyon.2023.e16348

Možina, M., & Lazar, T. (2018). Syntax-Based Analysis of Programming Concepts in Python. In *Artificial Intelligence in Education: 19th International Conference, AIED 2018, London, UK, June 27–30, 2018, Proceedings, Part II 19* (pp. 236-240). Springer International Publishing. https://doi.org/10.1007/978-3-319-93846-2_43

Nurmuiza, I., Maonde, F., & Sani, A. (2015). The Effect of Motivation on Mathematics Learning Outcomes High school student. *Jurnal Pendidikan Matematika*, *6*(2), 113-122. https://doi.org/10.36709/jpm.v6i2.2065

Pabón, O. S., & Villegas, L. M. (2019). Fostering motivation and improving student performance in an introductory programming course: An integrated teaching approach. *Revista EIA*, *16*(31), 65-76. https://doi.org/10.24050/reia.v16i31.1230

Papadakis, S. (2020). *Robots and Robotics Kits for Early Childhood and First School Age*. International Association of Online Engineering.

Peng, R., & Fu, R. (2021). The effect of Chinese EFL students' learning motivation on learning outcomes within a blended learning environment. *Australasian Journal of Educational Technology*, *37*(6), 61–74. https://doi.org/10.14742/ajet.6235

Rais, D & Xuezhi, Z. (2023). Human cognitive: learning mathematics through Python programming to support students' problem-solving skills. *Anatolian Journal of Education*, *8*(2), 85-98. https://doi.org/10.29333/aje.2023.826a

Ramli,I, S., Maat, S, M., & Khalid, F. (2019). Learning Analytics in Mathematics: A Systematics Review. *International Journal Research in Progressive Education and Development*, *8*(4): 436-449. http://dx.doi.org/10.6007/IJARPED/v8-i4/6563

Rodrigues, L., Toda, A. M., Oliveira, W., Palomino, P. T., Avila-Santos, A. P., & Isotani, S. (2021, March). Gamification works, but how and to whom? an experimental study in the context of programming lessons. In *Proceedings of the 52nd ACM technical symposium on computer science education* (pp. 184-190). https://doi.org/10.1145/3408877.3432419

Sakharkar, S. (2023). Systematic Review: Analysis of Coding Vulnerabilities across Languages. *Journal of Information Security*, *14*(4), 330-342. https://doi.org/10.4236/jis.2023.144019

Santos, H., Batista, J., & Marques, R. P. (2019). Digital transformation in higher education: the use of communication technologies by students. *Procedia Computer Science*, *164*, 123-130. https://doi.org/10.1016/j.procs.2019.12.163

Saraswat, S., Keswani, B., Kulshrestha, R., Sharma, S., Verma, N., & Alam, S. (2022). Accuracy assessment of several machine learning algorithms for breast cancer diagnosis. *Mathematical Statistician and Engineering Applications*, *71*(4), 12578-12587. https://doi.org/10.17762/msea.v71i4.2378

Schacter, D. L. (1999). The seven sins of memory: Insights from psychology and cognitive neuroscience. *American psychologist*, *54*(3), 182. https://doi.org/10.1037/0003-066X.54.3.182

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and information technologies*, *22*, 469-495. https://doi.org/10.1007/s10639-016-9482-0

Suherman, S., & Vidákovich, T. (2022). Assessment of mathematical creative thinking: A systematic review. *Thinking Skills and Creativity*, *44*, 101019. https://doi.org/10.1016/j.tsc.2022.101019

Sweller, J., van Merriënboer, J. J., & Paas, F. (2019). Cognitive architecture and instructional design: 20 years later. *Educational psychology review*, *31*, 261-292. https://doi.org/10.1007/s10648-019-09465-5

Tabet, N., Gedawy, H., Alshikhabobakr, H., & Razak, S. (2016, July). From alice to python. Introducing text-based programming in middle schools. In *Proceedings of the 2016 ACM Conference on innovation and Technology in Computer Science Education* (pp. 124-129). https://doi.org/10.1145/2899415.2899462

Takefuji, Y. (2023). An updated tutorial on reproducible PyPI applications for advancing chemometrics and boosting learner motivation. *Chemometrics and Intelligent Laboratory Systems*, *241*, 104941. https://doi.org/10.1016/j.chemolab.2023.104941

Tilak, J. B. (2021). COVID-19 and education in India: A new education crisis in the making. *Social Change*, *51*(4), 493-513. https://doi.org/10.1177/00490857211050131

Wainer, J., & Xavier, E. C. (2018). A controlled experiment on Python vs C for an introductory programming course: Students' outcomes. *ACM Transactions on Computing Education (TOCE)*, *18*(3), 1-16. https://doi.org/10.1145/3152894