Indian Journal of Information Sources and Services
ISSN: 2231-6094 (P) Vol.14, No.3, 2024, pp.45-51

DOI: https://doi.org/10.51983/ijiss-2024.14.3.07

Integrated Framework for the Implementation of Visual Programming Language in Science Experiment for Secondary School

Tang Kuan Shuen[1], Corrienna Abdul Talib[2*], Sharifah Osman[3], Sim Tze Ying[4], Izzah Sakinah Ahmad[5], Subuh Anggoro[6], Maria Erna[7] and Lay Yoon Fah[8]

[1]Faculty of Social Sciences and Humanities, Universiti Teknologi Malaysia, Malaysia
[2*]Faculty of Social Sciences and Humanities, Universiti Teknologi Malaysia, Malaysia
[3]Faculty of Social Sciences and Humanities, Universiti Teknologi Malaysia, Malaysia
[4]School of American Education, Sunway University, Bandar Sunway, Malaysia
[5]Department of Computing and Information System, School of Engineering and Technology, Sunway University, Bandar Sunway, Malaysia
[6]Department of Preservice Elementary Teachers Training, Universitas Muhammadiyah Purwokerto, Indonesia
[7]Department of Chemistry Education, Universitas Riau, Indonesia
[8]Faculty of Psychology and Education, Universiti Malaysia Sabah, Malaysia
E-mail: [1]tangshuen@graduate.utm.my, [2]corrienna@utm.my, [3]sharifah.o@utm.my, [4]tzeyings@sunway.edu.my, [5]122095475@imail.sunway.edu.my, [6]subuhanggoro@ump.ac.id, [7]mariaerna@lecturer.unri.ac.id, [8]layyf@ums.edu.my
ORCID: [1]https://orcid.org/0009-0002-2850-5134, [2]https://orcid.org/0000-0003-2230-3670, [3]https://orcid.org/0000-0003-2896-9377, [4]https://orcid.org/0000-0002-8233-712X, [5]https://orcid.org/0009-0007-7345-6366, [6]https://orcid.org/0000-0002-6762-9079, [7]https://orcid.org/0000-0002-9446-4863, [8]https://orcid.org/0000-0002-5219-6696
(Received 30 April 2024; Revised 11 June 2024, Accepted 15 July 2024; Available online 30 September 2024)

***Abstract* - This study explores the integration of visual programming language in science experiments. The study employs a quasi-experimental design with a one-group pretest-posttest-only design. It examines the perceptions of the students before and after the integration. The experiment also investigates whether visual programming language can be integrated into science experiments or how it can be better integrated into the curriculum. The results indicate that teaching tools, learning outcomes, module design, and teaching approaches as factors that influence the acceptance and implementation of visual programming language in science experiments. Further refinement identified teaching tools, module design and teaching approaches as the core factors of an integrated framework that will help teachers to design the lessons in different levels and conditions. Teachers can use this framework to design their own modules and teaching practices. The framework can also be used in developing teacher preparation programs, as well as STEM education.**

***Keywords:* Visual Programming Languages, Perceptions, Science Experiment, Integrated Framework**

I. INTRODUCTION

A debate has taken place in recent years regarding how science experiments can stimulate problem-solving as students are taught how to replicate a procedure or answer questions based on examination situations (Gericke et al., 2023; Osborne, 2015; Shamsiah, 2014). Experiments are not intended for these practices. An experiment, by definition, involves hands-on activities such as designing a procedure, conducting experiments, and determining the answers to questions arising from the results. The opportunity should be provided for students to plan, analyse, evaluate, and develop strategies to solve science-related problems (Gericke et al., 2023; Millar, 2010; Osborne, 2015; Shamsiah, 2014; Wahyudiati, 2021). New approaches are being introduced to develop problem-solving skills through science experiments. As an example, visual programming language in science experiments have gained importance to enhance problem-solving abilities (Chongo et al., 2021; Iyamuremye & Nsabayezu, 2022; Jancheski, 2017) while, at the same time, reinforcing STEM education design (Loganathan et al., 2019; Talib et al., 2022). Most science teachers have been demanding more empirical and evidence-based studies to gain a deeper understanding of what is needed during implementation and how it can be effectively taught (Chongo et al., 2021) while some researchers call for further study, large sample size and long-term research (Chongo et al., 2021; Iyamuremye & Nsabayezu, 2022; Loganathan et al., 2019; Meerbaum-Salant et al., 2010; Talan, 2020).

Besides that, a number of concerns were raised. One of the main concerns was student and teacher engagement (Jancheski, 2017; Salleh et al., 2013). Several factors contribute to the concern, including, the features of the visual programming language and module design (Jancheski, 2017; Kalogiannakis & Papadakis, 2019; Kurihara et al., 2015; Salleh et al., 2013) while many teachers consistently and continuously reported the challenges they faced which include limited support, professional development, guidelines, training, time constraints and resources (Kalogiannakis & Papadakis, 2019; Rose et al., 2020; Salam, 2022; Talib et al., 2022). Overall, a school should not use sophisticated language features. The features of the language should be clear, evident, and logically structured, with fast feedback mechanisms. Additionally, it should be easy-to-access help and references with carefully-designed exercise (Kurihara et al., 2015). In this light, Scratch was chosen as the visual programming language of choice (Chongo et al., 2021; Iyamuremye & Nsabayezu, 2022; Kalogiannakis & Papadakis, 2019; Loganathan et al., 2019; Rose et al., 2020; Talan, 2020; Talib et al., 2022). The features of this program are easy to manage and master. The user will only need to drag and drop instructions to create a graphical environment or content-based presentation. The techniques were easy to grasp even for novice teachers and students, and the software could be used offline as well (Theisen, 2019). As for the design of the modules, the activities should be aligned with the curriculum which is to provide students with the knowledge and skills of science and technology and enable them to solve problems and making decisions in daily life (Chongo et al., 2021; Malaysia, 2018; Ministry of Education, 2002). Under the context of science experiments, the activities should consider a wide range of methods and procedures for solving a problem as well as ways to plan, monitor and reflect on the problem-solving processes. This includes activities and applications that demonstrate problem-solving skills. For example, students should be given the opportunity to understand, predict, design, and evaluate problem statements. Through the activity, an individual understands that a problem exists, represents it by identifying the gap that needs to be closed, selects a method for addressing it, evaluates those solutions, and consolidates gains in order to understand how the experience and learning (Ronald et al., 2024) can be applied.

Designing such activities may require further study and long-term research due to the fact that such integration is just emerging (Chongo et al., 2021). To design such activities, there is a need to gather the perceptions of the students and teachers. Such findings could be used to form an integrated framework that includes ways to integrate visual programming language and factors influencing implementation and acceptance. At the same time, it should allow teachers to develop competencies in problem-solving at a fine-grained level, construct assessment tools and to investigate how learning (Arasu et al., 2024) progression happens under different conditions. This is because teachers struggle to understand, design, and integrate a lesson whenever a new approach is introduced. Meanwhile, students lose the interest to apply the approach, resulting many teachers call for more proper course, module, instructional guidelines, teachers training and practices (Kalogiannakis & Papadakis, 2019; Muhaimin et al., 2019; Al Muhtasib, 2019; Priemer et al., 2020; Smutny, 1992; Llopiz-Guerra et al., 2024). Therefore, this study also aims to create a framework in clarifying educational practices, (Mathur et al., 2024) developing teaching programs, and structuring the curriculum specification for STEM syllabus relating to visual programming languages and science investigations.

II. Research Objectives

The objectives of this study are:

(i) To identify factors that influence the acceptance and rejection of visual programming languages in science experiments,
(ii) To integrate visual programming languages into science experiments.

III. Research Questions

The research questions of this study are:

(i) What are the factors that influence the acceptance and rejections of visual programming languages in science experiments?
(ii) How to incorporate visual programming language into science experiments?

IV. Research Methodology

This study employs the questionnaire type of descriptive research, and it uses quasi-experimental one-group pretest-posttest-only design as its research design. Quasi-experimental one-group pretest-posttest-only design is used because this study only involves students who use the intervention. Hence, only the experimental group is included. The population consists of all science students and science teachers in all public secondary schools in Malaysia. Among the populations in the study are the secondary school students from Malaysia who are studying science. A pilot study is conducted as a preliminary step and 78 science students from three schools have participated, making a total of 78 samples. Samples are subjected to integrated science experiments through a module. Prior to and after the intervention, a questionnaire about their experiences in visual programming language and science experiments is administered. The questionnaire serves as the only instrument to generate data for the study. The difference between the results is analyzed through descriptive and content analysis. The objective of the analysis is to examine the influence of independent variable on dependent variable as well as the degree to which intervention affects the outcomes. In this light, independent variable refers to the use of visual programming language while dependent variable refers to students' perceptions towards the integration.

V. Results

Descriptive analysis is conducted on the items in the closed-ended questions of the questionnaire while the content analysis is used to analyze the items from the open-ended questionnaires. In the study, Figure 1 shows the bar graph with percentage of students' acceptance of visual programming language for science experiments. 96.2% of students agreed that visual programming languages can be used for science experiments, while 3.8% rejected it. The idea of integrating visual programming language into science experiments was rejected for three reasons. Students found it difficult to understand the technical aspects of adding the blocks to commands. Understanding such concepts requires more time than is normally taught in a classroom. Prior knowledge is necessary to support the approach, which is no different from laboratory skills.

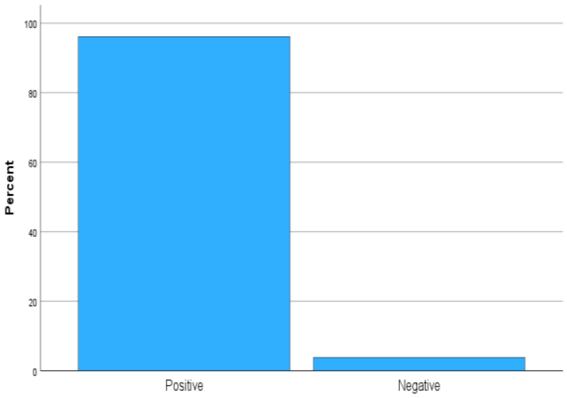

Fig. 1 Student Acceptance of the Use of Visual Programming Language for Science Experiments

Meanwhile there are many reasons that students shared on their acceptance towards the integration. Four students commented that the integration made learning easier. The final products in the form of animation also helped another group of 14 students understand the information better. Seeing the connection in curriculum (3 students) and enhancing scientific investigation skills (13 students) could be two of the possible factors. A total of 19 students commented on the skills they acquired during the integrated science experiments. The students pointed the problem-solving skills and scientific investigation skills as factors in accepting integration. Students commented that this integration is acceptable as long as the experiments discussed in the integration are not difficult to comprehend, as many students enjoyed the new learning experiences (13 students), in which they learned new skills such as computational thinking (7 students). Moreover, a student commented that such an approach allows students to take control of their learning since they can learn on their own from home. Table I displays a summary on why students accept and reject the integration of visual programming language in chemistry experiments.

TABLE I REASONS WHY STUDENTS ACCEPT AND REJECT THE INTEGRATION OF VISUAL PROGRAMMING LANGUAGE IN SCIENCE EXPERIMENTS

Attitude	Reason(s)	Participant	Frequency
Accept (Positive)	make learning easier	1,4,29,57	4
	easy to understand / to be used/ gives illustration	2,6,9,21,37,43,45,46,47,53,60,62,67,73	14
	help students to increase the computational thinking	4,10,15,26,40,60,64	7
	if the experiments are not difficult	12,23,43	3
	support the scientific skills in investigation	15,47,48,49,50,56,58,60,65,69,77,78	13
	supports problem-solving	16,34,39,40,54,77	6
	New experience/learning	18,28,32,33,38,39,42,52,66,67,68,69,71	13
	easy and fast	22,24,27	3
	connection with the curriculum	31,45,76	3
	self-learning	70	1
Reject (Negative)	difficult to construct the procedure into the program	41	1
	takes longer time	44	1
	prior knowledge needed	72	1
Irrelevant comment		2,5,7,8,11,13,14,19,20,30, 35,36,51,61,74,75	16

The content analysis is then used to study the feedback written by the samples in the open-ended questions of the questionnaires. The feedback is categorized into main categories based on statements that appear frequently in the answers. For each statement, positive and negative statements are distinguished. Analyzing these statements could identify the factors influencing the acceptance and implementation of visual programming languages in chemistry experiments and how they can be incorporated into classroom learning. As a result, a framework will be developed.

VI. DISCUSSION

The discussion is generated based on the content analysis from the descriptive data. It is arranged based on the research questions raised in this study.

To Identify the Factors that Influence the Acceptance and Implementation of Visual Programming Language in Science Experiments

The introduction of visual programming language to science practical is method of incorporating computational thinking into science practical (Chongo et al., 2021; Loganathan et al., 2019; Talib et al., 2022) It is evident from the definition of each element in computational thinking skills and problem-solving process of scientific method that the concepts are similar. As a result of comparisons and contrasts, Table illustrates the relationships between computational thinking skills and problem-solving processes.

Teaching tools, learning outcomes, module design, and teaching approaches are identified as the factors that influence the acceptance and implementation of visual programming languages in science experiments.

- Teaching tools: The ability to use electronic devices and visual programming language were two of the factors that led students to accept visual programming as an experiment tool, whereas internet access and availability of electronic devices were identified as the reasons for rejecting the integration. There is a misconception about the need for internet access, which needs to be addressed. An offline version of Scratch 2.0 is available. Meanwhile, the need for an electronic device in this integration is true but Scratch 2.0 is limited to laptop usage at the moment. It needs to be explored how it can be used in tablets.
- Learning outcomes: Students accepted the visual programming language because it was perceived as new knowledge and experience, and they believed they developed computational thinking and problem-solving skills at the same time. It introduces students to different methods of solving a scientific problem while offering an alternative approach to learning science experiments.

  However, a few students commented on the skills set required to understand the visual programming language. The implementation was rejected by some students because of this factor. Because the language feature was uncommon, students perceived it as complex, which hindered their learning and took longer to complete. Some students preferred the hands-on approach as it gave them a more thorough understanding.
- Module design: Module design has been identified as an important factor in acceptance and implementation in three aspects. This includes guidelines, duration of the workshop, and assessment. The pilot studies were conducted three times for three different schools on different timelines, and the module was constantly improved after each pilot study. As a result, the new module is structured in the form of developmental stages with four tasks each, as discussed in Table II.

TABLE II TASKS IN EACH PHASE OF THE MODULE

Phase	Task	Activity	Objective
One	1	Unplugged and plugged in activity (Sudoku and Origami)	To introduce the concepts of visual programming language
Two	2	Create an interactive video as illustration for science concept	To understand how Scratch can be integrated into science
Three	3	Following the guidelines to conduct a science experiment with Scratch	To understand how Scratch can be integrated into science experiment
Four	4	Design your own experiment with Scratch to solve real-life problems	To develop student's problem-solving skills and allow them to take control over the learning during science experiment

- Guidelines: In the first pilot study, students struggled to complete the task, and they did not believe that visual programming language could be incorporated into the science experiment. This is because there were no guidelines given to complete the tasks. Hence, the module was modified with more guidelines to support the integration. Results show that students were more confident to integrate the visual programming into the science experiments. As the module was improvised with guidelines in different developmental stages, students accepted visual programming language integration in science experiments. Through the module, students could make visual representations, such as animations, at different developmental stages, which enhances their understanding and learning. A minority of students, however, commented on the medium of instruction in the guideline. As their medium of instruction in science was Malay, students preferred to complete the module in that language.
- Duration of the integration: A difference existed during the pilot test between a two-day workshop and a three-day workshop. The three-day workshop generally received more positive feedback than the two-day workshop. Furthermore, students believed they would need many rounds of training to master the features of the visual programming language before they could design their own chemistry experiments.
- Assessment: Before moving on to questions with complicated experimental designs, students stressed the importance of simplifying the questions in the module. This was because it required time for students to understand the implementation of the features in the visual programming language before they could be integrated.
- Teaching Approaches: The teaching approaches were identified as another important factor in acceptance and implementation in three aspects. This includes the design, learning activities, and facilitators.
- Design: A few proposals were made by the samples regarding the design of the teaching approach. In the pilot study, the module is designed from the concept of problem-based learning. Students are required to use Scratch 2.0 to explain how to solve a real-life problem together with the expected results. In the questionnaire, group-work, competition, and game-based learning were suggested as alternatives.

  Group work was emphasized as an important factor of acceptance by many students. In this way, they shared ideas, divided tasks, and completed the workshop. They believed the group-work concept would allow them to complete the task more quickly. A consideration needs to be taken if the integration should be conducted as an individual task or group based.

Games and competition were proposed in order to excite the implementations as well as to increase their interest in learning the integration. By using the game or competition approach, students could also assess their understanding of the concept and assess their error. There is a possibility of exploring on the suggestions provided by the samples.

- Learning activities: Some students perceived this learning approach as STEM learning. It was suggested that the hands-on experiment be conducted first, then the integration of the visual programming language as the end-of-term project. Several students believe that visual programming languages should be taught in technology classes as plugged-in activities so that the students can focus on mastering the features of the language while in science classes, they can apply what they have learned. Having such a connection helped them remember the concept better and made them feel connected. However, how can this learning activity be integrated? Similar workshop will be conducted among the science teachers in the next phase and an interview would be conducted to discuss about teacher's perceptions of successful integration.
- Facilitators: Students emphasized the need for detailed explanations and guidance to complete the assignment by the facilitator. In one comment, it was suggested that there should be more than one facilitator to ensure adequate support was provided.

To Develop an Integrated Framework Explaining how Visual Programming Languages can be Integrated Into Science Experiments

This framework will provide the foundation for exploring how visual programming languages can be integrated into science experiments through the factors that influence the acceptance and implementation. The next phase of the study is to expand the framework in relation to effects of the integration towards the development of scientific problem-solving skills. Figure 2 shows the initial phase of the integrated framework to support the integration of visual programming language into science experiments.

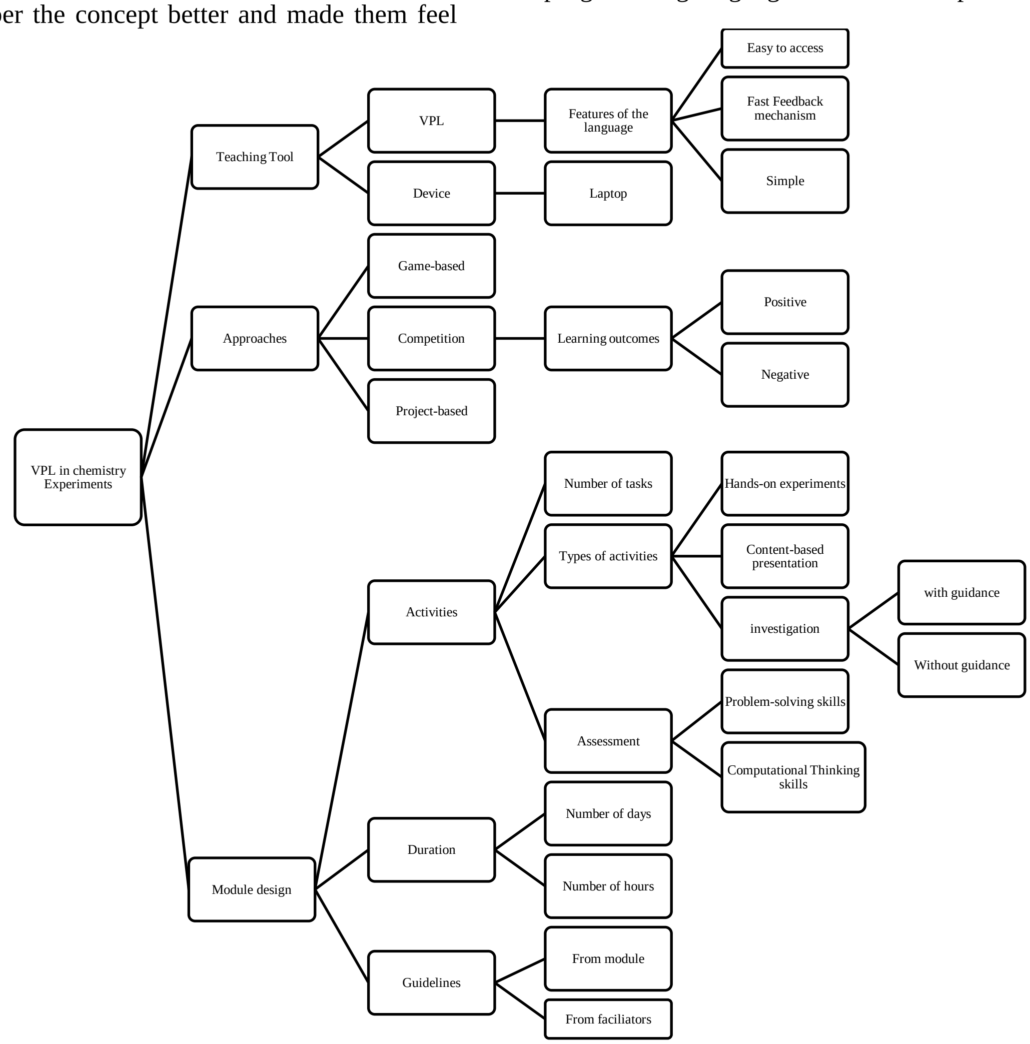

Fig. 2 Integrated Framework to Support the Integration of Visual Programming Language into Chemistry Experiments

VII. CONCLUSION

Three emerging themes are recognized as the elements of the integrated framework for visual programming languages in science experiments. They are teaching tool, approaches and module designs. Each of the elements are bonded with the specific constructs. Through the factors influencing the acceptance and implementations, the study proposes an integrated framework that will help teachers to design the lessons in different levels and conditions. Teachers can use this framework to design their own module and teaching practices.

VIII. SIGNIFICANCE OF THE STUDY

The findings of this study are relevant to people involved in technology and science education. They are educational researchers and science teachers. The integrated framework offers a basic outline for designing lessons related to science experiments and visual programming languages. Science teachers can use the framework to plan and organize the activities of the lesson. They will have an idea of what facilities are to be used, how to modify the lessons, and how to improve the teaching strategies. Educational researchers on the other hand, could use this framework to sort out differing assumptions and explore connections between theory and practice. Besides that, this framework could also be used to develop teacher preparation programs, teaching practices, and curriculum concerning STEM syllabus, visual programming languages, and science experiments.

IX. LIMITATIONS OF THE STUDY

This study has some limitations that need to be addressed in future research. The limitations relate mainly to the sample, instruments, and research design.

The samples only involved secondary students who have exposure to science content and experience in conducting science experiments. Most of the students have little or no experience in using visual programming language. It may disrupt their perceptions of the effectiveness of the integrated curriculum as they need time to learn the features of the language before they learn the integrated curriculum. Future studies should allow students to master visual programming languages in IT courses first before conducting the integration so that students would not be affected in the process.

Further limitations could be the questionnaire instrument. This instrument is based on self-reported knowledge and perceptions of the samples, and it is unclear how reliably the samples are able to report their own perceptions and knowledge. Future research should triangulate self-declaration with other measures, such as lesson observations or performance assessment, to overcome potential biases. Thus, computational thinking and problem-solving skills will have to be tested through computational thinking tests and problem-solving tests, respectively, to show the effects of visual programming language on its development. Furthermore, more in-depth studies on the relationship between visual programming languages and science experiments should be examined as it is important for discriminant validity. Understanding how these aspects interact could be critical to improving technology integration effectively.

Regarding research design, this study only involves the experimental group, and it is conducted by the facilitators who designed the modules. There is a possibility that the outcomes could be biased. Thus, a larger sample size is required for the data to be more reliable. The samples should be divided into control and experimental groups respectively while the assigned facilitators should be teachers who are blind to the hypothesis.

X. RECOMMENDATION FOR FUTURE STUDY

The result of the study shows that the integrated approach, which involves visual programming languages and science experiments, has the potential to enforce STEM education. At the same time, it enhanced students' problem-solving skills. Future in-depth studies are recommended to use this framework so that it can be widely used by all teachers across different syllabi and levels of education. The recommendations for future studies are listed below:

- Since this study only focuses on descriptive study, more in-depth studies are required such as inferential, correlation and phenomenological research, in order to increase the accuracy of the framework.
- In determining the effectiveness of this framework, future research should be expanded to more schools from all over the world so that the reliability and validity of the final framework can be increased. Longitudinal studies are necessary to investigate the interplay and reciprocal effects of the integration over time.
- A variety of approaches could be used towards the integration of visual programming in science experiments, as proposed by the students. A problem-based learning, individual work and group work approach were used in the pilot test, while some students suggested a game-based and competition approach to increase their interest. The difference between such approaches could be explored.
- The module design could also be explored, as many students stressed the need to provide guidelines and assessments in accordance with developmental stages.

XI. ACKNOWLEDGEMENTS

This work was supported/ funded by the Ministry of Higher Education under Fundamental Research Grant Scheme (FRGS/1/2022/SSI07/UTM/02/15).

REFERENCES

[1] Al Muhtasib, A. (2019). The Effect of Interactive Drills Using Dry Lab on the Acquisition of Laboratory Skills in Learning Science among the Ninth-Grade Female Students in Palestine in Light of Their Thinking Style. *Journal of Education and Learning*, *8*(5), 89-99. https://doi.org/10.5539/jel.v8n5p89

[2] Arasu, R., Chitra, B., Anantha, R.A., Rajani, B., Stephen, A.L., & Priya, S. (2024). An E-learning Tools Acceptance System for Higher Education Institutions in Developing Countries. *Journal of Internet Services and Information Security, 14*(3), 371-379.

[3] Chongo, S., Osman, K., & Nayan, N. A. (2021). Impact of the Plugged-in and Unplugged Chemistry Computational Thinking Modules on Achievement in Chemistry. *Eurasia Journal of Mathematics, Science and Technology Education*, *17*(4), 1–21. https://doi.org/10.29333/ejmste/10789

[4] Gericke, N., Högström, P., & Wallin, J. (2023). A systematic review of research on laboratory work in secondary school. *Studies in Science Education*, *59*(2), 245–285. https://doi.org/10.1080/03057267.2022.2090125

[5] Iyamuremye, A., & Nsabayezu, E. (2022). Mathematics and Science Teacher's Conception and Reflection on Computer Programming with Scratch: Technological and Pedagogical Standpoint. *International Journal of Education, Training and Learning*, *6*(1), 11–16. https://doi.org/10.33094/ijetl.v6i1.488

[6] Jancheski, M. (2017). Improving teaching and learning computer programming in schools through educational software. *Olympiads in Informatics*, *11*, 55–75. https://doi.org/10.15388/ioi.2017.05

[7] Kalogiannakis, M., & Papadakis, S. (2019). Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. *International Journal of Technology Enhanced Learning*, *11*(3), 231-246. https://doi.org/10.1504/ijtel.2019.10020447

[8] Kurihara, A., Sasaki, A., Wakita, K., & Hosobe, H. (2015). A programming environment for Visual block-based domain-specific languages. *Procedia Computer Science*, *62*(December), 287–296. https://doi.org/10.1016/j.procs.2015.08.452

[9] Llopiz-Guerra, K., Daline, U. R., Ronald, M. H., Valia, L. V. M., Jadira, D. R. J. N., Karla, R. S. (2024). Importance of Environmental Education in the Context of Natural Sustainability. *Natural and Engineering Sciences*, *9*(1), 57-71.

[10] Loganathan, P., Mohd Alwi, A., Romainor, N., Abdul Talib, C., Hanri, C., Abdul Malik, A. M., & Siang, K. H. (2019). Students' chemistry learning process through visualprogramming language: A preliminary study. *International Journal of Recent Technology and Engineering*, *8*(1C2), 509–514.

[11] Malaysia, K. P. (2018). Kurikulum Standard Sekolah Menengah: Kimia. *Dokumen Standard Kurikulum dan Pentaksiran Tingkatan*, *4*.

[12] Mathur, G., Nathani, N., Chauhan, A. S., Kushwah, S. V., & Quttainah, M. A. (2024). Students' Satisfaction and Learning: Assessment of Teaching-Learning Process in Knowledge Organization. *Indian Journal of Information Sources and Services*, *14*(1), 1–8.

[13] Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2010). Learning computer science concepts with Scratch. *ICER'10 - Proceedings of the International Computing Education Research Workshop*, 69–76. https://doi.org/10.1145/1839594.1839607

[14] Millar, R. (2010). *Analysing practical science activities to assess and improve their effectiveness*. Hatfield: Association for Science Education, 5-11. https://www.researchgate.net/profile/Robin-Millar/publication/264889206_Analysing_practical_science_activities_to_assess_and_improve_theireffectiveness/links/5509777e0cf26ff55f858d46/Analysing-practical-science-activities-to-assess-and-improve-their-effectiveness.pdf

[15] Ministry of Education, M. (2002). Integrated curriculum for primary schools. *In Integrated Curriculum for Secondary Schools, Curriculum Specification, Science Form 1*.

[16] Muhaimin, M., Habibi, A., Mukminin, A., Saudagar, F., Pratama, R., Wahyuni, S., Sadikin, A., & Indrayana, B. (2019). A sequential explanatory investigation of TPACK: Indonesian science teachers' survey and perspective. *Journal of Technology and Education Technology*, *9*(3), 269–281. https://doi.org/https://doi.org/10.3926/jotse.662

[17] Osborne, J. (2015). Practical Work in Science: Misunderstood and Badly Used?. *School Science Review*, *96*(357), 16–24.

[18] Priemer, B., Eilerts, K., Filler, A., Pinkwart, N., Rösken-Winter, B., Tiemann, R., & Zu Belzen, A. U. (2020). A framework to foster problem-solving in STEM and computing education. *Research in Science and Technological Education*, *38*(1), 105–130. https://doi.org/10.1080/02635143.2019.1600490

[19] Ronald, M. H., Walter, A. C. U., Segundo, J. S. T., Silvia, J. A. V., Sara, E. L. O., Ronald, A. M., Dora, E. E. M., & Doris, E. F. (2024). Exploring Software Infrastructures for Enhanced Learning Environments to Empowering Education. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 15*(1), 231-243.

[20] Rose, S. P., Habgood, M. P. J., & Jay, T. (2020). Designing a Programming Game to Improve Children's Procedural Abstraction Skills in Scratch. *Journal of Educational Computing Research*, *58*(7), 1372–1411. https://doi.org/10.1177/0735633120932871

[21] Salam, S. (2022). A systemic review of Problem-Based Learning (PBL) and Computational Thinking (CT) in teaching and learning. *International Journal of Humanities and Innovation (IJHI)*, *5*(2), 46–52. https://doi.org/10.33750/ijhi.v5i2.145

[22] Salleh, S. M., Shukur, Z., & Judi, H. M. (2013). Analysis of Research in Programming Teaching Tools: An Initial Review. *Procedia - Social and Behavioral Sciences*, *103*, 127–135. https://doi.org/10.1016/j.sbspro.2013.10.317

[23] Shamsiah, S. (2014). Teachers ' purposes and practices in implementing practical work at the lower secondary school level. *Procedia - Social and Behavioral Sciences*, *116*, 1016–1020. https://doi.org/10.1016/j.sbspro.2014.01.338

[24] Smutny, J. F. (Ed). (1992). *Preparing the workforce of tomorrow: A conceptual Framework for career and Tecnical education*.

[25] Talan, T. (2020). Investigation of the Studies on the Use of Scratch Software in Education. *Journal of Education and Future*, *18*, 95–111. https://doi.org/10.30786/jef.556701

[26] Talib, C. A., Shuen, T. K., Wahidah, N., Hakim, A., Thoe, N. K., Hermita, N., Ali, M., & Ong, E. T. (2022). Scratchtopia Challenge : From Science Experiment to Coding in Upper Primary School, *0832*(17), 79–88.

[27] Theisen, K. J. (2019). Programming languages in chemistry: A review of HTML5/JavaScript. *Journal of Cheminformatics*, *11*(1), 1–19. https://doi.org/10.1186/s13321-019-0331-1

[28] Wahyudiati, D. (2021). Investigating Problem Solving Skills and Chemistry Learning Experiences of Higher Education Base on Gender and Grade Level Differences. *Journal of Science and Science Education*, *2*(2), 62–67. https://doi.org/10.29303/jossed.v2i2.632