Daengku: Journal of Humanities and Social Sciences Innovation *ISSN: 2775-6165 (online)*
Vol. 6 No. 3 (2026)
https://doi.org/10.35877/454RI.daengku5023

A Computational Thinking Learning Trajectory for Primary Mathematics through Renewable Energy Optimization Projects

Swasti Maharani[a,*], Irna Tri Yuniahastuti[b], Vera Dewi Susanti[a], Ihtiari Prastyaningrum[c], Addy Septyawan[d]

[a]*Faculty of Teacher Training and Education, Universitas PGRI Madiun, Madiun, Indonesia*
[b]*Faculty of Technology, Universitas PGRI Madiun, Madiun, Indonesia*
[c]*Faculty of Technology, Universitas PGRI Madiun, Madiun, Indonesia.*
[d]*Faculty of Teacher Training and Education, Universitas Sebelas Maret, Surakarta, Indonesia.*

Abstract

Despite the increasing emphasis on Computational Thinking (CT) in primary education, limited research has examined how CT can be systematically developed through mathematics learning trajectories situated in authentic sustainability contexts. This study aimed to design and validated a Computational Thinking Learning Trajectory (CTLT) that integrates four CT practices—Data Practices, Modeling and Simulation, Computational Problem-Solving, and Systems Thinking—through renewable energy optimization projects in primary mathematics classrooms. Using a design research approach, the study was conducted in three phases: preliminary design, teaching experiment, and retrospective analysis. Participants were 30 fourth-grade students (aged 9–10 years) from an primary school in Madiun, Indonesia. Quantitative and qualitative data were collected through observations, interviews, student artifacts, and assessment rubrics to investigate the emergence of CT practices during learning activities. The results, derived from rubric attainment scores and supported by qualitative evidence, showed that all CT practices emerged during learning activities, with Modeling and Simulation (93.3%), Computational Problem-Solving (83.3%), Data Practices (80%), and Systems Thinking (70%) demonstrating substantial levels of attainment. Students demonstrated a progression from arithmetic reasoning toward algorithmic and systems-oriented thinking as they analyzed data, constructed models, evaluated alternative renewable energy solutions, and optimized decision-making processes. The validated learning trajectory provides empirical evidence of how sustainability-based mathematical projects can support the development of CT in primary education. These findings contribute to the growing body of research on CT integration in mathematics curricula and offer practical guidance for implementing project-based learning within the Indonesian Merdeka Curriculum.

Keywords: Computational Thinking, Renewable Energy, Mathematics Learning, Science, CT–Math Learning Trajectory (CT-MLT)

Received: 17 March 2026 Revised: 27 May 2026 Published: 30 June 2026

1. Introduction

The Fourth Industrial Revolution (Industry 4.0) has brought about profound transformations in the ways humans work, think, and solve problems through the utilization of digital technologies. In this context, one of the essential competencies required in the twenty-first century is Computational Thinking (CT), a systematic way of reasoning about complex problems by formulating strategies that can be automated through technological tools (Wing, 2006). CT involves key components such as decomposition, abstraction, pattern recognition, and algorithmic design, which are inherently aligned with mathematical reasoning and problem-solving. Integrating CT into education has therefore become increasingly critical, particularly in mathematics learning, as mathematics serves as the foundation for developing logical reasoning, algorithmic processes, and mathematical modelling, all of which constitute the core dimensions of CT itself (Weintrop et al., 2016a). Through mathematics learning, students can be guided to construct mathematical models, design algorithmic solutions, and evaluate outcomes systematically in accordance with the principles of Computational Thinking.

---

[*] Corresponding author.

E-mail address: swasti.mathedu@unipma.ac.id

However, in the Indonesian context, the integration of Computational Thinking (CT) into school mathematics instruction remains limited and has not yet been systematically structured. Several studies indicate that existing CT approaches tend to be partial, focusing primarily on programming activities or the use of digital tools without directly connecting them to the mathematical content being taught (Clark-Wilson et al., 2020; Turgut et al., 2024). This condition creates a gap between national education policies that emphasize the strengthening of technological literacy and classroom mathematics practices that remain conventional and fragmented.

While Computational Thinking has been increasingly discussed across K–12 education, its integration in primary mathematics remains underexplored, particularly for students in upper-primary grades. This study focuses on fourth-grade students (aged 9–10 years), a developmental stage at which learners begin to engage with more structured mathematical reasoning, including measurement, data representation, comparison, and simple optimization tasks. These emerging competencies provide an appropriate foundation for introducing Computational Thinking practices through contextualized mathematical projects.

Addresing this gap, there is a need for a study that explores the integration of Computational Thinking (CT) into mathematics learning through the context of renewable energy, particularly a solar panel optimization project. This context is recognized for its strong educational, applicative, and contextual value. Through the renewable energy optimization project, students engage with age-appropriate mathematical concepts such as angle measurement, comparison of numerical values, data organization, data representation, and simple optimization decisions. These activities provide opportunities for students to collect data, identify patterns, construct models, and develop algorithmic reasoning, which are central components of Computational Thinking. Furthermore, this approach aligns with the vision of the Merdeka Curriculum, which emphasizes project-based learning, strengthening numerical literacy, and developing the Pancasila Student Profile as a holistic educational framework.

Recent studies have increasingly focused on the integration of Computational Thinking (CT) in mathematics education, both conceptually and practically. Research has examined CT integration within STEM learning environments (Shute, Sun, & Asbell-clarke, 2017), the use of programming in mathematical contexts (Yadav et al., 2014), and the implementation of data and algorithm driven projects (Gonzalez-Jimenez et al., 2021; Jiménez et al., 2019). Ye et al., (2023) through a systematic review of 24 studies, emphasized that integrating CT into K–12 mathematics education can strengthen instructional structures through algorithmic and pattern-based reasoning. Their findings suggest that visual programming and data modeling act as key mediators connecting mathematical concepts with CT competencies. In the context of digital learning tools, Kannadass et al. (2023) and Ye et al. (2023b) showed that using GeoGebra in calculus instruction enables students to develop algorithmic thinking and abstract representations core elements of CT although technical challenges remain in effectively integrating such software into classroom practice. Science and mathematics are becoming computational endeavors (Weintrop et al., 2016a; Wing, 2006). This fact is reflected in the recently released Next Generation Science Standards and the decision to include ‘‘computational thinking” as a core scientific practice. With this addition, and the increased presence of computation in mathematics and scientific contexts, a new urgency has come to the challenge of defining computational thinking and providing a theoretical grounding for what form it should take in school science and mathematics classrooms. This paper presents a response to this challenge by proposing a definition of computational thinking for mathematics and science in the form of a taxonomy consisting of four main categories: data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices. In formulating this taxonomy, we draw on the existing computational thinking literature, interviews with mathematicians and scientists, and exemplary computational thinking instructional materials. This work was undertaken as part of a larger effort to infuse computational thinking into high school science and mathematics curricular materials. In this paper, we argue for the approach of embedding computational thinking in mathematics and science contexts, present the taxonomy, and discuss how we envision the taxonomy being used to bring current educational efforts in line with the increasingly computational nature of modern science and mathematics (Lodi & Martini, 2021; Pei et al., 2018; Weintrop et al., 2016b).

Recent research has highlighted that project-based and inquiry-oriented learning environments can effectively support the development of Computational Thinking by engaging students in problem decomposition, algorithmic reasoning, and reflective problem solving (Hsu et al., 2018; Shute, Sun, & Asbell-Clarke, 2017). The study found that project-oriented instructional strategies allow students to engage more actively in systematic and reflective thinking processes. Wehner et al. (2024) added a theoretical perspective by presenting experts' views on CT in mathematics education. They developed the Mathematical Digital Competency framework, which positions CT as an essential component in the design of modern mathematics learning environments.

The integration of Computational Thinking (CT) in more applied contexts has been explored through mathematical modeling projects, as investigated by Yeni et al. (2024), who involved pre-service teachers in constructing and presenting mathematical models based on real-world situations. The findings revealed that participation in authentic projects supported the development of decomposition, abstraction, and model evaluation skills the three core pillars of CT. Similar findings have been reported in project-based STEM contexts where authentic real-world problems promoted students' computational thinking, modeling competence, and conceptual understanding (Pei, 2018; Shin et al., 2021a). Their study demonstrated significant improvements in students' conceptual understanding, critical thinking, and CT competencies.

Khalil et al. (2023) designed a renewable energy–based STEM program for secondary school students, which successfully met quality indicators and received positive feedback from both teachers and learners. These projects connected mathematical concepts such as linear functions, systems of equations, and optimization with global issues while simultaneously introducing elements of Computational Thinking (CT) through data modeling and analysis. In the field of science education, (Shin et al., 2021b) developed a project-based physics unit involving an evaporative cooling model to foster CT development, demonstrating that similar approaches can be effectively adopted in mathematics instruction.

A technological approach was also evident in the study by Mylonas et al. (2023), which utilized an Internet of Things (IoT) laboratory and gamification to enhance students' energy awareness. Although slightly beyond the seven-year review scope, this study demonstrated how real-time data and algorithm-based decision-making can serve as a foundation for developing Computational Thinking (CT). Finally, Sheriff & Sevukan (2023) in their critical review of machine learning algorithms applied to renewable energy prediction, emphasized the importance of computational competence in understanding, selecting, and evaluating complex mathematical models.

Overall, these studies underscore the importance of integrating Computational Thinking (CT) into project-based mathematics learning that is oriented toward real-world issues, particularly those related to renewable energy. Nevertheless, few studies have explicitly developed learning trajectories grounded in socially and ecologically meaningful real-world contexts, especially in primary mathematics education. To address this gap, the present study adopts a theoretical framework that integrates Computational Thinking (CT) and Mathematics Learning Trajectory principles within a contextualized project-based environment. This framework serves as the conceptual foundation of the study, referred to as the Computational Thinking–Mathematics Learning Trajectory (CT–Math LT) Framework. The present study is grounded in the Computational Thinking–Mathematics Learning Trajectory (CT–Math LT) Framework, which integrates the principles of Computational Thinking (CT) (Weintrop et al., 2016a; Wing, 2006) with the Learning Trajectory (LT) approach within Design Research (Sarama & Clements, 2019). This framework posits that a learning trajectory does not merely represent a sequence of instructional activities, but rather the progression of students' ways of thinking as they connect mathematical ideas with computational practices.

Within the CT–Math LT Framework, the four core CT practices (data practices, modeling and simulation, computational problem-solving, and systems thinking) serve as the foundation for designing contextual and progressive mathematics learning activities. Through this framework, students are guided to conceptualize mathematical ideas (conceptualization), develop algorithmic solutions (algorithmization), and reflect on inter-variable relationships systemically (systemic reflection). The CT–Math LT Framework thus functions as both a conceptual foundation and a form of Local Instruction Theory, underpinning the development of a project-based mathematics learning trajectory on renewable energy in this study.

This study aims to address a gap in the literature by developing and implementing a learning trajectory that integrates Computational Thinking (CT) into primary mathematics learning through a renewable energy optimization project. Specifically, it explores how instructional design can foster students' computational thinking skills while deepening their understanding of mathematical concepts within real-world contexts. The novelty of this study lies in the development of a renewable energy–based project learning trajectory that unifies three essential dimensions: (1) contextual strengthening of mathematical concepts, (2) integration of CT through modeling and algorithmic approaches, and (3) education that is relevant to global and sustainability issues. In addition to providing a theoretical contribution to the literature on CT-based mathematics learning design, this study is expected to produce a practical model that can be adopted by teachers, curriculum developers, and educational policymakers. Accordingly, the study holds strategic significance in promoting the meaningful integration of technology within contextual and future-oriented mathematics education. Considering the growing call to integrate Computational Thinking (CT) into mathematics learning through authentic, sustainability-oriented contexts, this study aims to examine how the development and implementation of a

Hypothetical Learning Trajectory (HLT) can embed CT skills within mathematics instruction through a renewable-energy optimization project to foster students' mathematical and computational reasoning.

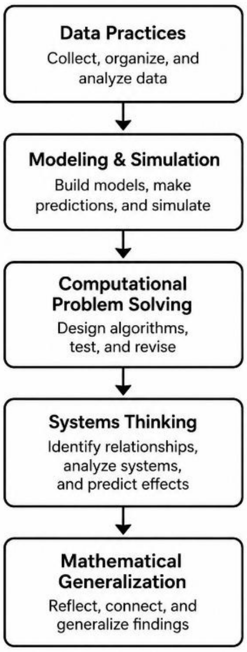

Figure 1. Conceptual Framework of CT-MathLT for Renewable Energy Optimization Projects.

2. Methods

2.1 Research design

This study employed a Design Research approach Akker et al., (2006) and Gravemeijer & Cobb (2006) aimed at developing and testing a learning trajectory that integrates Computational Thinking (CT) into mathematics learning through a renewable energy–based project. The approach comprises three main phases: (1) preliminary design, (2) teaching experiment, and (3) retrospective analysis. This design was selected as it enables an in-depth exploration of students' learning processes in real time while simultaneously examining the effectiveness of the instructional design within an authentic classroom context. In this study, the learning trajectory focused on the mathematical topics of angle measurement and basic data analysis, which were integrated into a solar panel optimization project through data modeling and simple simulation activities.

The learning trajectory was developed based on the principles of Realistic Mathematics Education (RME) and the Computational Thinking Framework proposed by Weintrop et al. (2016), which encompasses four key dimensions: Data Practices, Modeling and Simulation Practices, Computational Problem-Solving Practices, and Systems Thinking Practices. The present study adopted a single-cycle design research approach to explore the initial feasibility of the CT–Math LT framework. While the study was grounded in qualitative design research, quantitative descriptive indicators derived from rubric scores and observation records were used to support the analysis of students' Computational Thinking practices. These quantitative indicators were not intended for statistical inference but rather to provide evidence of the extent to which CT practices emerged during the learning process.

2.2 Preparation for the Experiment

This phase began with a comprehensive review of the curriculum and relevant literature to ensure the alignment of the selected mathematical topics angle measurement and basic data analysis (including tables, mean/median as optional concepts, and value comparison) with the characteristics of fourth-grade students. The CT framework referred specifically to four core practices: Data Practices, Modeling and Simulation Practices, Computational Problem-Solving Practices, and Systems Thinking Practices. The solar panel context was chosen for its relevance to students' everyday experiences and its potential to support activities involving angle measurement (30°–45°–60°) and light intensity data analysis. Based on this review, the learning objectives, achievement indicators, and success criteria for each activity were established.

Subsequently, a Hypothetical Learning Trajectory (HLT) was developed, detailing the sequence of learning activities, specific objectives, anticipated student responses or misconceptions, and corresponding scaffolding strategies. The learning design consisted of four stages that aligned with the core dimensions of Computational Thinking. The first stage, Data Practices, involved collecting light intensity data, checking, and organizing the data by verifying the 1–5 scale range, ensuring consistency of angle and distance, identifying outliers, re-measuring or taking the median, and calculating the mean. The second stage, Modeling and Simulation, engaged students in constructing a foldable solar panel model with hinges to simulate various angles and compare their predictions with the obtained measurements. The third stage, Computational Problem-Solving, required students to formulate an optimization problem and design an algorithm or flowchart to determine the most efficient angle, followed by testing and debugging procedures. Finally, the fourth stage, Systems Thinking, guided students to develop causal diagrams representing the interrelationships among variables (time–angle–intensity and possible interfering factors) and to analyze "if–then" scenarios. To support the implementation, several instructional tools were developed, including student worksheets (containing raw and organized data tables as well as prediction–result comparison tables), observation sheets for assessing CT practices and student engagement, interview guidelines for both students and teachers, algorithm rubrics, systems diagram rubrics, and portfolio rubrics. All instruments underwent expert validation by specialists in mathematics education, primary education, and Computational Thinking to evaluate their clarity, relevance, and feasibility.

The implementation phase was preceded by coordination with the school administration and the completion of ethical approval procedures, including obtaining consent from parents and teachers. The partner teacher received a briefing on the Computational Thinking (CT) framework and the use of instructional materials and tools to be applied during the lessons. Scheduling of classroom space and time was also arranged to ensure the smooth execution of activities. A limited pilot test was then conducted with a small group of students or one class to assess the clarity of instructions, the appropriateness of time allocation for each activity, and the readability of the student worksheets (LKPD). The findings from this pilot study were used to refine the Hypothetical Learning Trajectory (HLT), for instance, by adding angle cards, reference lines on the model, a flowchart template, and a systems diagram template. Readiness criteria for proceeding to the Design Experiment phase were established, covering instrument validation, completeness of instructional materials, feasibility of the time schedule, and the teacher's preparedness to implement the sequence of learning activities.

2.3 Desain Experiment

This phase encompassed the implementation of the sequence of learning activities outlined in the Hypothetical Learning Trajectory (HLT), the monitoring of students' learning processes in an authentic classroom setting, and the systematic collection of data from 30 fourth-grade students at SD Islam Siti Hajar, Madiun. The implementation was carried out over four learning sessions (4 × 60 minutes) across two weeks, with students organized into small groups of four to five members. The flow of activities was structured as follows: Activity 1 focused on Data Practices (data collection, data verification and organization, and calculation of the mean); Activity 2 centered on Modeling and Simulation Practices (construction of a foldable panel model and comparison between predictions and simulation results); Activity 3 emphasized Computational Problem-Solving Practices (problem formulation, algorithm or flowchart design, testing, and debugging); and Activity 4 addressed Systems Thinking Practices (development of a causal diagram representing time–angle–intensity relationships and a brief group presentation). The classroom teacher acted as the primary facilitator, while the researcher observed, documented, and ensured the fidelity of implementation in accordance with the designed learning trajectory.

During the implementation, data were collected using several validated instruments: observation sheets (capturing the occurrence of the four CT practices and aspects of student engagement), student worksheets containing raw and organized data tables as well as prediction–result comparison tables, algorithm and systems diagram rubrics, group portfolios, and post-instruction interview guidelines for both students and the teacher. Observations were conducted at regular intervals (approximately every 5–10 minutes) to record distinctive learning behaviors and key moments when students collected data, constructed models, made algorithmic decisions, and mapped system relationships. Photographs and short video recordings were used as documentation to support data triangulation, while the researcher's field notes captured emerging misconceptions, technical difficulties (such as model hinge instability or inconsistent angle measurements), and scaffolding strategies provided by the teacher.

The design experiment adopted an adaptive structure while maintaining fidelity to the planned design. To monitor implementation consistency, the researcher employed a fidelity checklist that recorded the alignment of instructional steps, the duration of each activity, and the attainment of micro-level objectives, with a minimum fidelity threshold of

80 percent. Minor adjustments were permitted based on formative classroom findings, such as inserting an example of mean calculation when more than one-third of the groups made errors, providing a flowchart template when the algorithmic sequence was incorrect, or adding guiding lines for angle measurement when the simulation lacked precision. After each session, the teacher and researcher conducted a brief reflection to identify necessary improvements for subsequent meetings. All data from the four sessions were then compiled and analyzed retrospectively in the following phase.

2.4 Retrospective analysis

This phase focused on comparing the conjectures outlined in the Hypothetical Learning Trajectory (HLT) with students' actual responses, analyzing data from multiple sources, and formulating design refinements. The analyzed data included observation sheets (documenting the occurrence of CT practices and student engagement), complete student worksheets (containing raw–organized data tables and prediction–result comparisons), algorithm and systems diagram rubrics, group portfolios, and interview transcripts or notes from both teachers and students. To quantify the emergence of Computational Thinking practices, each CT dimension was assessed using analytic rubrics and structured observation protocols. Student performance on each indicator was on a three-level scale (0 = no evidence, 1 = partial evidence, 2 = complete evidence). Percentage attainment was calculated by dividing the total score obtained by students by the maximum possible score and multiplying by 100. These percentages were used solely as descriptive indicators of the degree to which each CT practice emerged and were subsequently interpreted alongside qualitative evidence from observations, interviews, and student artifacts.

The analysis revealed a general alignment between the Hypothetical Learning Trajectory (HLT) and students' actual responses, with several important refinements identified. In the Data Practices component, the procedures for checking and organizing data functioned effectively once the outlier threshold rule was clarified, and an example of mean calculation was provided. A key recommendation is to require two tables (Raw Data and Organized Data) and to add a column for justifying data adjustments. In the Modeling and Simulation component, the use of the foldable panel model proved effective; however, the precision of angle measurements improved when a jig or guiding lines for 30°–45°–60° angles were provided. It is therefore recommended to include a hinge template and step-by-step photo guide. In the Computational Problem-Solving component, most groups were able to construct if–then rules and flowcharts, yet their procedural consistency improved when branching-decision templates were introduced and cross-check sessions between groups were conducted. A short debugging note should be required as evidence of testing and revision. In the Systems Thinking component, establishing relationships among time, angle, and intensity remained challenging. The diagrams became more meaningful when students were given pre-designed icons or symbols and example scenarios illustrating variable changes. Accordingly, it is recommended to add a model systems diagram and more specific reflective guiding questions.

The output of the retrospective phase consisted of a revised Hypothetical Learning Trajectory (R-HLT) and a set of design principles formulated from classroom data. Key revisions included adjustments to the sequence and time allocation of activities, clarification of the outlier threshold rule and the use of the median, incorporation of hinge templates and angle guidelines, the requirement for flowcharts accompanied by testing and revision steps, and the provision of a systems diagram template with guiding scenario questions. The assessment instruments were also refined by sharpening the descriptors in the algorithm and systems diagram rubrics and by adding operational definitions for each CT practice in the observation sheet. The resulting design principles emphasize authentic and contextual problem settings, the explicit articulation of CT practices within each activity, the use of both visual and procedural scaffolding, and authentic assessment integrating both process and product. These findings serve as the basis for planning subsequent iterations or for preparing the final research report.

3. Result and Discussion

2.5 Results

Stage 1: Preparation for the Experiment

The Preparation for the Experiment stage focused on establishing the theoretical rationale, mapping the curriculum, designing the initial Hypothetical Learning Trajectory (HLT), developing, and validating the research instruments, and conducting a limited pilot test. A summary of the outcomes from this stage is presented below.

2.5.1 Curriculum and Literature Review

The curriculum and literature review indicated that the most appropriate mathematics topics for fourth-grade students are angle measurement and basic data processing. At this level, students are expected to recognize and use 30°, 45°, and 60° angles with a protractor, collect information, present it in simple tables or diagrams, and identify patterns (for instance, determining the highest value). To ensure that these skills are meaningfully assessed, the activity design required students to record light intensity data at the three angles, with a minimum of three measurements per angle using a 1–5 scale. The initial data were recorded in a "Raw Data" table and subsequently reviewed and refined: range errors (e.g., entries exceeding 5) were corrected, the consistency of angles and distances was verified, and large deviations (differences of ≥ 2 levels from the other two measurements at the same angle) were marked for remeasurement or summarized using the median. The average value for each angle was then calculated, and an initial pattern was written as the basis for determining the optimal angle.

The literature review on Computational Thinking (CT) identified four CT practices that framed the design. Data Practices guided students in collecting, validating, and refining data to ensure it was suitable for analysis. Modeling and Simulation Practices engaged students in constructing a hinged folding panel model that could be locked at 30°, 45°, and 60°, predicting the outcomes, and comparing them with real experimental results. Computational Problem-Solving Practices required students to formulate an angle optimization problem and develop structured solutions expressed both in "if–then" statements and flowcharts, followed by testing and debugging using their own and other groups' data. Finally, Systems Thinking Practices led students to map the relationships among variables (time, angle, and intensity) and to analyze how changes in one variable affect the overall behavior of the small system they had constructed.

The solar energy context was selected for its relevance to students' everyday experiences and its feasibility for safe, low-cost, and easily prepared hands-on classroom activities. This context provided tangible objects for applying angle measurement, generating analyzable data, and enabling authentic assessments based on both process and product, including the Raw–Clean Data Tables, Prediction vs. Simulation Results Table, algorithms or flowcharts, and system diagrams. With the integration of the four CT practices, the curriculum topic, and clearly defined technical rules for data measurement and refinement, the preparation stage produced an operational design ready for pilot implementation with 30 fourth-grade students at an primary school in Madiun City over four 60-minute sessions.

2.5.2 Participants

Participants consisted of 30 fourth-grade students (aged 9–10 years) from a private Islamic elementary school in Madiun, Indonesia. The school was purposively selected because it had experience implementing project-based learning aligned with the Merdeka Curriculum. The participants represented a typical elementary classroom setting in which renewable energy topics had not previously been integrated with computational thinking practices. Although the study was conducted in a single classroom context, the participants provided a suitable setting for exploring the development of computational thinking through the proposed CT-MathLT.

2.5.3 Designing the initial HLT (Hypothetical Learning Trajectory)

The initial HLT was designed as a sequence of interconnected activities, each with specific micro-goals, anticipated student responses or misconceptions, and corresponding scaffolding strategies. The series of activities was mapped onto the four CT practices: Data Practices, Modeling and Simulation, Computational Problem-Solving, and Systems Thinking. Each activity generated assessable artifacts such as data tables, models or simulations, algorithms or flowcharts, and system diagrams allowing both the learning process and outcomes to be observed and evaluated.

In the Data Practices phase, students worked in groups to measure light intensity at three fixed angles—30°, 45°, and 60°—with three measurements taken per angle using a 1–5 scale. The results were first recorded in the Raw Data Table, then reviewed and refined: value ranges were verified (no entries outside 1–5), the consistency of light source angles and distances was checked, and large deviations (practically defined as differences of ≥ 2 levels from the other two measurements at the same angle) were marked for remeasurement or, if time was limited, summarized using the median. The cleaned data were then transferred to the Refined Data Table, where students calculated the mean value for each angle and recorded the emerging pattern as the basis for prediction. In the Modeling and Simulation phase, students were tasked with constructing a hinged folding panel model that could be locked at 30°, 45°, and 60°, predicting light intensity based on the calculated means, then conducting a real simulation and comparing the predictions with the observed results. Any discrepancies were recorded and analyzed for possible causes, such as imprecise angle settings, hand shadows, or variations in lamp distance.

Students formulated an optimization problem "how to determine the best angle for maximum intensity" and developed structured solutions in two formats: if–then statements and flowcharts incorporating the steps of input, process, decision, and output in the Computational Problem-Solving phase. The constructed algorithms were tested using both their own group's data and that of other groups; any inconsistencies were recorded as debugging findings for revision. The final part of the HLT, Systems Thinking, engaged students in mapping the relationships among time, angle, and intensity, along with possible interfering factors (such as clouds or object shadows), within a system diagram. Students then explored "if… then…" scenarios—for example, if the time shifts from morning to noon, then the recommended angle changes—and wrote concise conclusions describing how the small system they had constructed operates. The initial Hypothetical Learning Trajectory (HLT) table is presented as follows:

Table 1. The initial Hypothetical Learning Trajectory (HLT)

Sequence of Activities / CT Phases	Specific Learning Objectives	Activity Description & Student Tasks	Anticipated Student Responses / Strategies	Teacher Scaffolding Strategies	Produced Products / Student Artifacts
1. Data Practices *(*Data Collection and Refinement*)*	Students can measure light intensity at three angles (30°, 45°, and 60°) and refine the data based on validation rules	Students work in groups to measure light intensity on a 1–5 scale using a flashlight and a panel model. They record the results in a Raw Data Table, check for range errors, and calculate the mean for each angle.	Students identify outlier values but are uncertain about how to determine them. Some students forget how to calculate the mean.	Provide an example of the outlier threshold (≥ 2 scale-level difference). Demonstrate a step-by-step example of mean calculation. Supply a distance jig to ensure measurement precision.	Raw and Refined Data Tables. Preliminary graph of light intensity patterns across angles
2. Modeling & Simulation Practices *(*Constructing and Analyzing the Solar Panel Model*)*	Students can construct a simple foldable panel model and compare the simulation results with their predicted data.	Students construct a hinged cardboard model with angles of 30°, 45°, and 60°, predict the angle that produces maximum light intensity, and then conduct a real simulation to compare the results.	Students notice discrepancies between their predictions and the actual results. They realize possible causes such as inaccurate angle settings or shadow interference during measurement.	Provide angular guidelines on the model base. Demonstrate how to record the difference between predicted and observed results and guide students to analyze the possible causes.	Prediction vs. Simulation Results Table. Photographs of the panel model and corresponding observation notes.
3. Computational Problem-Solving Practices *(*Developing an Optimization Algorithm*)*	Students can formulate a simple algorithm to determine the optimal angle (maximum light intensity).	Based on the collected data, students write *if–then* rules to identify the optimal angle and represent them in a simple flowchart (input–process–decision–output). They then conduct cross-	Some students write a linear algorithm without decision branches. Others struggle to handle cases with equal (tied) results.	Provide a branched flowchart template. Present an example algorithm with two conditionals (equal value) cases.	Optimization algorithm flowchart. Test and revision notes (debug log)

Sequence of Activities / CT Phases	Specific Learning Objectives	Activity Description & Student Tasks	Anticipated Student Responses / Strategies	Teacher Scaffolding Strategies	Produced Products / Student Artifacts
		testing with data from other groups (debugging).		Guide intergroup debugging sessions.	
4. Systems Thinking Practices *(*Analyzing Relationships Among Variables*)*	Students can represent the relationships among variables (time–angle–intensity) and interfering factors in a system diagram	Students create a small system diagram illustrating the effects of time, angle, intensity, and other factors (e.g., clouds, shadows). They analyze conditional changes, such as "if time shifts, then intensity changes."	Students draw only one-way relationships. They often omit interfering factors such as shadows or reflections.	Provide a complete example diagram with two-way arrows. Supply ready-to-use icons or symbols. Pose guiding reflection questions (e.g., "What changes if…?").	System diagram illustrating relationships among time, angle, and light intensity. Short written reflection (3–5 sentences)

2.5.4 Instrument Development and Operational Definitions

The development of research instruments began with the design of a student worksheet (LKPD) package that included all required learning artifacts: the Raw and Refined Data Tables, the Prediction versus Simulation Results Table, sheets for algorithm and flowchart construction, a system diagram sheet, and a brief reflection section. To capture classroom processes, observation sheets were prepared to assess CT practices and student engagement, equipped with clear indicators, and note columns. Meaning making was explored through structured interview guides for both students and teachers, organized according to each CT practice. Product assessment was conducted using specific rubrics: an algorithm rubric, a system diagram rubric, and a portfolio rubric evaluating the completeness of each group's artifacts (worksheet, poster or diagram, experimental documentation, and reflection).

Operational definitions for each CT practice were formulated to ensure consistent use by teachers and researchers during observation and assessment. For Data Practices, the indicators included data completeness (at least three measurements per angle on a 1–5 scale), validity of range and consistency of distance/angle, identification of outlier values using a practical threshold rule (a difference of ≥ two levels from the other two measurements at the same angle, followed by remeasurement or summarization using the median), accuracy of mean calculations, and a clearly written initial pattern statement. For Modeling and Simulation, the indicators included a functioning folding panel model that locks at 30°, 45°, and 60°, the presence of predictions prior to testing, precision in conducting simulations, reasonable prediction–result discrepancies (target average deviation ≤ one scale level), and a logical explanation of the causes of these discrepancies. The indicators comprised a clearly formulated optimization problem, algorithms or flowcharts containing input–process–decision–output components, correct decision branching (including handling of equal values), evidence of cross-testing using data from other groups, and documented revisions (debugging notes) for Computational Problem-Solving. The indicators included a diagram containing at least three main variables (time, angle, and intensity) and one interfering factor, correctly directed influence arrows, and a 3–5 sentence narrative describing the impact of changes in one variable consistent with the data (system thinking).

Table 2. Rubric for Assessing Students' Computational Thinking Practices

CT Practice	Indicator	Score 0	Score 1	Score 2
Data Practices	Collecting and organizing data	No evidence	Partial evidence	Complete evidence
Data Practices	Interpreting data patterns	No evidence	Partial evidence	Complete evidence
Modeling & Simulation	Constructing a mathematical model	No evidence	Partial evidence	Complete evidence
Modeling & Simulation	Comparing prediction and observation	No evidence	Partial evidence	Complete evidence

CT Practice	Indicator	Score 0	Score 1	Score 2
Computational Problem Solving	Developing algorithmic procedures	No evidence	Partial evidence	Complete evidence
Computational Problem Solving	Revising procedures after testing	No evidence	Partial evidence	Complete evidence
Systems Thinking	Identifying relationships among variables	No evidence	Partial evidence	Complete evidence
Systems Thinking	Explaining system interactions	No evidence	Partial evidence	Complete evidence

Score interpretation:

0 = indicator not demonstrated
1 = partially demonstrated
2 = fully demonstrated

The observation sheet used a checklist format indicating whether each behavior was observed or not, accompanied by concise operational behavior definitions for each indicator. The interview guides were condensed into key questions designed to explore students' learning experiences and teachers' considerations during implementation. All instruments were formatted uniformly for readability and ease of duplication, ensuring that both process and product data could be collected consistently during classroom implementation.

2.5.5 Expert Validation and Instrument revision

The expert validation stage involved three validators: a primary mathematics education expert, a Computational Thinking (CT) expert, and an experienced classroom teacher. They evaluated the clarity, relevance, and feasibility of each instrument using a 1–4 rating scale. The Content Validity Index (CVI) was calculated as the proportion of ratings in categories 3 or 4 relative to the total items. The results indicated an average CVI of ≥ 0.86 across all instruments, suggesting that the instruments were suitable for use with minor to moderate revisions based on the validators' feedback.

In detail, the student worksheet (LKPD), consisting of 18 items, obtained a CVI of 0.89. The main recommendation was to clarify the example of data refinement and add a column for the reason behind each correction; accordingly, the LKPD was revised to include an explicit example of an outlier and a dedicated column for justification. The observation sheet (12 items; CVI 0.92) was recommended to sharpen the operational definitions of each CT indicator for greater specificity; revisions were made by adding observable behavior descriptors for each indicator. The algorithm rubric (6 items; CVI 0.88) was advised to emphasize the testing and debugging stages; the revision added criteria for cross-group testing and a notes column for revisions. In the system diagram rubric (6 items; CVI 0.90), validators requested examples of two-way relationships and interfering factors; improvements included the addition of icons/symbols and clear examples of bidirectional relationships. The interview guide (10 items; CVI 0.86) was recommended to be simplified and focused on learning experiences; revisions were made by condensing items and grouping questions according to the four CT practices.

With these results, the instrument package—including the student worksheet (LKPD), observation sheet, algorithm rubric, system diagram rubric, and interview guide—met the criteria for content validity. The revisions ensured clearer operational definitions, more concrete examples and templates, and well-documented evidence of processes such as cross-testing and revision logs. The revised instrument package was therefore ready for use in the implementation stage (Design Experiment).

Table 3. Summary of Expert Validation Results and Revisions

Instrument	Number of Items	CVI	Main Validator Recommendations	Revision Follow-up
LKPD (including tables and guide)	18	0,89	Clarify the example of data refinement; add a column for the reason	Added an example of an outlier and a specific column for refinement reasons
Observation Sheet (CT + engagement)	12	0,92	Define operational indicators of CT more specifically	Added observable behavior descriptors for each indicator
Algorithm Rubric	6	0,88	Emphasize the testing and debugging stages	Added criteria for cross-group testing and a notes column for revisions

Instrument	Number of Items	CVI	Main Validator Recommendations	Revision Follow-up
System Diagram Rubric	6	0,90	Provide examples of two-way relationships and interfering factors	Added icons/symbols and clear examples of bidirectional relationships
Interview Guide	10	0,86	Simplify questions and focus on learning experiences	Simplified and regrouped questions according to CT practices

The average CVI across all instruments was ≥ 0.86, indicating that they were suitable for use with minor to moderate revisions, as summarized in the revision follow-up column.

2.5.6 Limited trial (pilot)

The pilot involved eight students from different classes within the same school. The pilot focused on the clarity of instructions, time allocation, readability of the student worksheet (LKPD), precision of tools, and the smoothness of observation procedures. The initial trial results indicated that most students (85%) were able to follow the activity steps independently after a single demonstration, while the remaining students required assistance, particularly when identifying outlier values. The duration of each phase generally aligned with the planned schedule, except for the Data Practices phase, which required additional time (38 minutes compared to the planned 30 minutes) as students took extra care to maintain a consistent distance between the flashlight and the panel. The suggested adjustment was to extend the time by 5–8 minutes or provide a distance jig to ensure measurement consistency. The stability of the folding-panel models improved after adding guiding lines and reference points at 30°, 45°, and 60°. The data refinement strategy using a simple outlier threshold—marking values differing by ≥2 scale levels from two other measurements at the same angle—proved effective, while using the median served as a more reliable summary measure when time was limited. Furthermore, more detailed operational definitions enhanced inter-observer consistency, with an observation interval of 5–7 minutes per activity deemed sufficient to capture the learning process dynamics.

Based on the findings from the initial trial, several revisions were implemented to enhance the clarity of instructions, time efficiency, and measurability of the learning process. In the student worksheet (LKPD), a brief example of data refinement was added along with a dedicated row for recording the rationale behind data cleaning. Additionally, a three-box flowchart template was included, complete with an example of decision branching, as well as visual icons to support system diagram construction. For the physical model materials, pre-cut hinge patterns and a distance jig were provided to ensure consistent positioning of the flashlight relative to the panel. The observation sheet was also revised by adding behavioral descriptors for each Computational Thinking (CT) indicator and expanding the notes section to allow for more comprehensive observations. In terms of timing, the allocation for the Data Practices phase was adjusted to 35–40 minutes, or maintained at 30–35 minutes when a distance jig was used to expedite measurement. These revisions ensured that the CT–MLT design became more operational, efficient, and feasible for implementation in the subsequent classroom experiment phase.

Stage 2 Results: Design Experiment

2.5.7 Implementation Overview

The implementation was conducted over four sessions (4 × 60 minutes) involving 30 fourth-grade primary students in Madiun City, working in small groups of four to five students. The lesson sequence followed the Hypothetical Learning Trajectory (HLT): Session 1 focused on Data Practices; Session 2 on Modeling and Simulation; Session 3 on Computational Problem-Solving; and Session 4 on Systems Thinking. The classroom teacher facilitated learning activities and provided scaffolding, while the researcher conducted structured observations, documentation, and collected students' worksheets (LKPD), rubrics, and portfolios.

2.5.8 Session Outcomes and Supporting Evidence

Session 1 – Data Practices

Most groups demonstrated strong performance in data processing practices. A total of 90% of the groups successfully completed the Raw Data Table, 86.7% accurately calculated the mean, and 73.3% provided clear rationales for data refinement, including identifying outliers and explaining the corrective actions taken. Nevertheless, several common challenges emerged during the activity, such as inconsistencies in the flashlight-to-surface distance and students' difficulty in identifying outliers when the deviation was only one scale level. To address these issues, several in-class adjustments were made, including the use of a distance jig to maintain consistent measurement spacing, the provision of step-by-step examples for mean calculation, and the application of a simplified outlier threshold rule—specifically, any value differing by two or more scale levels from the other two measurements at the same angle was marked for re-measurement or summarized using the median. These adjustments proved effective in improving measurement consistency while deepening students' understanding of data validation and representation principles, which are central components of the Computational Thinking (CT) practice of Data Practices

Session 2 – Modeling and Simulation

A total of 90% of the foldable panel models constructed by students functioned stably at 30°, 45°, and 60°, indicating that most groups were able to assemble and operate the model as designed. Furthermore, 86.7% of the students' light intensity predictions closely matched the simulation results, with an average deviation of no more than one scale level. Students also successfully identified several causes of discrepancies between predicted and observed results, including inaccuracies in angle alignment, hand shadows interfering with measurements, and light reflections from the table surface. Based on these findings, three key improvements were implemented during classroom activities: the addition of angle guidelines on the model base, the inclusion of photo-based assembly instructions for hinge construction, and the assignment of one group member as a "angle keeper" responsible for maintaining the correct panel alignment. These adjustments enhanced measurement precision and consistency between predictions and simulations, while also strengthening students' understanding of the relationship between geometric variables and light intensity within the Modeling and Simulation practice of Computational Thinking–based learning

Session 3 – Computational Problem-Solving

A total of 83.3% of Students demonstrated increasing proficiency in algorithmic thinking, with an average algorithm rubric score of 1.67 out of 2, indicating that most students consistently developed and tested procedural solutions. Additionally, 76.7% of the flowcharts produced displayed correct decision branching, while 68% included evidence of cross-testing and subsequent revisions (debugging) after applying their algorithms to data from other groups. These results indicate that most students were able to apply algorithmic thinking systematically, including the ability to test and refine their procedures reflectively. Based on these observations, several classroom refinements were introduced, such as providing a branching flowchart template and explicit examples of if–then statements to help students address cases with equivalent results. These adjustments contributed to greater consistency in students' algorithmic representations and reinforced the Computational Problem-Solving dimension within CT-based learning, particularly by cultivating procedural and reflective thinking habits through algorithm testing and revision processes.

Session 4 – Systems Thinking

A total of 70% of Students achieved an average system thinking rubric score of 1.40 out of 2, suggesting that relationships among variables were generally recognized, although explanations of system interactions remained partially developed. This result indicates that most students were able to identify the interconnections among the main variables within the observed phenomenon. The primary challenges encountered included determining the direction of influence between variables in the relationship arrows and maintaining consistency in incorporating external factors, such as cloud cover or hand shadows. To address these difficulties, several refinements were implemented in subsequent classroom sessions, including the provision of ready-to-use icons and symbols, the inclusion of a complete example of a system diagram as a reference, and the use of scenario-based guiding questions such as "If the time shifts from morning to noon, what changes and why?". These strategies helped students better understand causal relationships and enhanced their ability to represent the dynamics among variables systematically. Consequently, these improvements strengthened the application of Systems Thinking within CT-based learning, particularly in developing students' relational analysis skills and systemic understanding of the modeled phenomenon.

2.5.9 Assessment of CT Practices and Percentage Calculation

To assess the emergence of Computational Thinking (CT) practices, students' artifacts, observations, and project outputs were evaluated using a rubric consisting of eight indicators distributed across four CT practices: Data Practices, Modeling & Simulation, Computational Problem-Solving, and Systems Thinking. Each indicator was scored on a three-level scale (0 = no evidence, 1 = partial evidence, 2 = complete evidence). A student was considered to have attained a CT practice when the average score across indicators within that practice reached at least 1.5 (75% of the maximum score). The reported percentages represent the proportion of students who met the attainment criterion for each CT practice. These quantitative results were used descriptively and were triangulated with qualitative evidence obtained from observations, interviews, worksheets, flowcharts, and system diagrams.

2.5.10 Quantitative Summary Across Aspects (n = 30 students)

Based on the CT assessment rubric, 24 of the 30 students (80.0%) met the attainment criterion for Data Practices, 28 students (93.3%) for Modeling & Simulation, 25 students (83.3%) for Computational Problem-Solving, and 21 students (70.0%) for Systems Thinking. These percentages represent the proportion of students demonstrating consistent evidence of the targeted CT practices across project activities and assessment artifacts. The highest attainment was observed in Modeling & Simulation, suggesting that students were particularly successful when working with tangible models and direct experimentation. Systems Thinking showed the lowest attainment, indicating that understanding interconnected relationships among multiple variables remained challenging for many students.

2.5.11 Implementation Fidelity and Adjustments

The implementation fidelity ranged from 86% to 92% per session, exceeding the minimum threshold of 80%. Minor in-situ adjustments were made, including an additional 5–8 minutes for the Data Practices session, provision of distance jigs and angle guides, addition of a flowchart template, and inclusion of system diagram icons. All modifications were documented in the fidelity checklist and served as the basis for revising the Hypothetical Learning Trajectory (HLT).

2.5.12 Implications for HLT Revision

Several instructional refinements were implemented across sessions to enhance the integration of CT practices within the CT–Math LT framework. In Session 1, students were required to use two data tables—Raw Data and Cleaned Data—with an additional column specifying the rationale for data refinement, and they were encouraged to use the median value when remeasurement was not feasible. Session 2 was supported with a ready-to-cut hinge template, 30°–45°–60° angle guides, and the assignment of an "angle keeper" role to maintain model precision. In Session 3, cross-testing and the inclusion of debug notes were made mandatory criteria for evaluating the algorithm improvement process. Meanwhile, Session 4 was strengthened through the provision of a system diagram template, a set of ready-to-use icons, and a list of "if…" scenarios to guide students' analysis of intervariable relationships.

The findings from Phase 2 indicate that the designed activities within the Hypothetical Learning Trajectory (HLT) were effectively implemented and successfully elicited all four Computational Thinking (CT) practices as intended. The areas requiring further refinement are the consistency of data cleaning procedures in the initial phase and the explicit articulation of variable relationships in the Systems Thinking stage. These findings serve as the foundation for the Revised HLT (CT- MathLT) to be applied in the next iteration.

Stage 3. Retrospective Analysis

2.5.13 The Aim and Sources

This phase aimed to compare the initial hypotheses in the HLT with students' actual responses, evaluate the attainment of the four CT practices, and formulate design revisions. Data were collected from observation sheets, student worksheets (raw–cleaned data tables and prediction–result tables), algorithm and system diagram rubrics, group portfolios, field notes, and short student–teacher interviews. Quantitative data were analyzed descriptively using frequencies and percentages, while qualitative data were examined through thematic coding and cross-source triangulation.

2.5.14 Summary of Quantitative Results

The attainment of Computational Thinking (CT) practices during the implementation of the CT–MathLT demonstrated consistency with the previous phase, with observed achievements of 80.0% in Data Practices, 93.3% in Modeling & Simulation, 83.3% in Computational Problem-Solving, and 70.0% in Systems Thinking (n = 30). The most stable indicators were the use of physical models and prediction testing activities, where most groups achieved an average deviation of no more than one scale level between predicted and simulated results. In contrast, relatively lower performance was found in the coherence of inter-variable relationships within system diagrams and the ability to explicitly articulate data-cleaning justifications. Student engagement levels were notably high, reflected in 93.3% group collaboration, 90.0% enthusiasm and communication, and 66.7% of written reflections demonstrating meaningful understanding without additional prompts. These findings indicate that the CT–MathLT effectively fostered active participation and consistent application of CT practices across various dimensions of mathematics learning activities.

2.5.15 Main Qualitative Findings

The implementation results demonstrated an overall improvement in the quality of each CT practice within the CT–MathLT framework. In data examination and refinement, students were better able to identify outliers after being provided with simplified threshold rules and example justifications, as well as by using the median as a summary measure when remeasurement was not feasible. The precision of physical models also improved by 30°–45°–60° angle guides and the assignment of an "angle keeper" role, with groups that assembled their hinges neatly showing smaller gaps between predicted and observed results. In the Computational Problem-Solving practice, writing if–then statements alongside flowcharts helped clarify reasoning processes, while cross-testing with other groups' data encouraged debugging and a deeper understanding of equivalent-value cases. Students' systems thinking also developed using variable icons and time-shift scenarios, which helped them recognize interfering factors such as shadows or lamp positions as components of a small system affecting light intensity.

2.5.16 Evidence of Transition from Arithmetic to Algorithmic Thinking

Evidence of students' transition from arithmetic reasoning to algorithmic thinking emerged most clearly during the Computational Problem-Solving phase. At the beginning of the project, students typically determined the optimal panel angle by directly comparing numerical values and selecting the largest result. Statements such as "45° is the best because it has the highest average intensity" reflected arithmetic reasoning focused on obtaining a single answer. As students engaged in flowchart construction, cross-testing, and debugging activities, their reasoning became increasingly procedural and generalized. Rather than simply identifying the largest value, students began expressing their solutions as reusable decision rules. For example, several groups formulated procedures such as:

> *If the average intensity at 45° is greater than the averages at 30° and 60°, then select 45°; otherwise compare the remaining angles and choose the highest value.*

Students subsequently represented these procedures in flowcharts and tested them using datasets from other groups. The debugging process required them to revise decision branches when equivalent values occurred. This shift from answer-oriented reasoning toward the creation and refinement of generalized procedures provides evidence of emerging algorithmic thinking.

2.5.17 Alignment of the HLT with students' responses

The implementation results indicated that most Computational Thinking (CT) practices within the CT–MathLT developed as initially anticipated, though certain aspects still required design reinforcement. The Data Practices phase proceeded as expected after terminology and procedures were clarified; the primary classroom need identified was the inclusion of concrete examples of data-cleaning rationales and a designated space in the student worksheet (LKPD) for recording them. The Modeling & Simulation phase also aligned with initial predictions, as issues of angular precision were successfully addressed with hinge templates and guiding angle lines. Meanwhile, Computational Problem-Solving demonstrated partial alignment with expectations, as many groups continued to rely on branched flowchart templates and example decision rules to handle cases with equivalent outcomes. In contrast, Systems Thinking fell below the initial expectations, indicating the need for enhanced visual scaffolding and scenario-based guiding questions to help students conceptualize the relationships among time, angle, and light intensity as an interconnected system rather than a linear sequence.

2.5.18 CT-MathLT Revision

Revisions to the CT–MathLT focused on strengthening the structure of learning activities and clarifying scaffolding strategies to ensure that each Computational Thinking (CT) practice was more observable and accessible to students. In the Data Practices phase, two distinct tables—Raw Data and Cleaned Data—were introduced, each equipped with a dedicated column for recording data-cleaning rationales, clearer outlier threshold rules, and a recommendation to use the median when remeasurement was not feasible within limited time. In the Modeling & Simulation phase, pre-cut hinge templates, angle guidelines on the model base, and the assignment of a "angle keeper" role within each group were incorporated to maintain measurement precision. The Computational Problem-Solving phase was enhanced by requiring outputs in the form of branched flowcharts, incorporating cross-testing procedures, and including brief revision notes as debug logs. Meanwhile, in the Systems Thinking phase, a structured system-diagram template with variable icons and examples of bidirectional arrows was employed, accompanied by a set of guiding "if… then…" scenarios designed to stimulate causal reasoning and deepen students' systemic understanding. Revised Hypothetical Learning Trajectory (R–HLT) on the CT–Math Learning Project "Optimasi Energi Terbarukan" is presented as follows:

Tabel 4. CT-MathLT Revision

Sequence of Activities / CT Phase	Specific Learning Objectives	Description of Student Activities & Tasks	Anticipated Student Responses / Strategies	Teacher Scaffolding Strategies	Products / Artifacts Produced
1. Data Practices (Data Collection and Preparation)	Students can measure light intensity at three angles (30°, 45°, 60°) and organize the data based on validation rules.	Students work in groups to measure light intensity on a 1–5 scale using sensors and a panel model. They record the results in a Raw Data Table, check error ranges, identify outliers, and calculate the mean for each angle.	– Students identify extreme values (outliers) but are confused about determining the threshold. – Some students forget how to calculate the mean correctly.	– Provide examples of outlier thresholds (≥2 levels difference). – Demonstrate step-by-step calculations of the mean. – Provide a distance jig to improve measurement precision.	– Raw and cleaned data tables. – A graph of light intensity versus angle.
2. Modeling & Simulation (Modeling and Simulation)	Students can construct a foldable panel model and simulate changes in light intensity at three angles.	Students assemble a panel model using pre-cut hinge patterns, set the angles at 30°, 45°, and 60° with guiding lines, and then compare predicted light intensity values with measured results.	– Students have trouble maintaining angle precision. – Some students forget to record the difference between predicted and observed results.	– Provide hinge-pattern templates and examples of stable models. – Emphasize the role of an “angle keeper.” – Demonstrate how to record prediction–observation differences.	– Photographs of a stable model. – A prediction–results difference table.
3. Computational Problem-Solving (Algorithms and Debugging)	Students can design branching algorithms and test the correctness of procedural steps.	Students write if–then rules to select the optimal angle, construct a branching flowchart, test the algorithm using data from other groups, and document revisions.	– Students initially create linear algorithms without branching. – Difficulty arises when handling cases with equal values.	– Provide examples of branching decision rules. – Supply a 3–5 box flowchart template. – Guide cross-group testing and peer review.	– Final flowchart. – A brief debug log.
4. Systems Thinking (System Diagrams)	Students can represent relationships among time, angle, light intensity, and confounding factors.	Students construct system diagrams using variable icons and bidirectional arrows, then analyze time-based change scenarios (e.g., “if time shifts, then…”)	– Students depict only one-directional relationships. – Difficulty arises in positioning confounding factors.	– Provide examples of bidirectional system diagrams. – Supply a guided list of scenario prompts.	– A complete system diagram including confounding factors. – Scenario-based reflections.

Sequence of Activities / CT Phase	Specific Learning Objectives	Description of Student Activities & Tasks	Anticipated Student Responses / Strategies	Teacher Scaffolding Strategies	Products / Artifacts Produced
5. Assessment & Reflection (Final Reflection)	Students can reflect the process of learning about the data, models, algorithm, and system.	Students write a summary of the process, describing how they verified the data, ensured angle accuracy, refined the algorithm, and developed an understanding of the system.	Students tend to describe procedures without providing underlying reasoning.	-Offer structured reflective prompts. -Provide exemplars of reflective responses. .	- Final Reflection Sheet. - Revised CT Rubric.

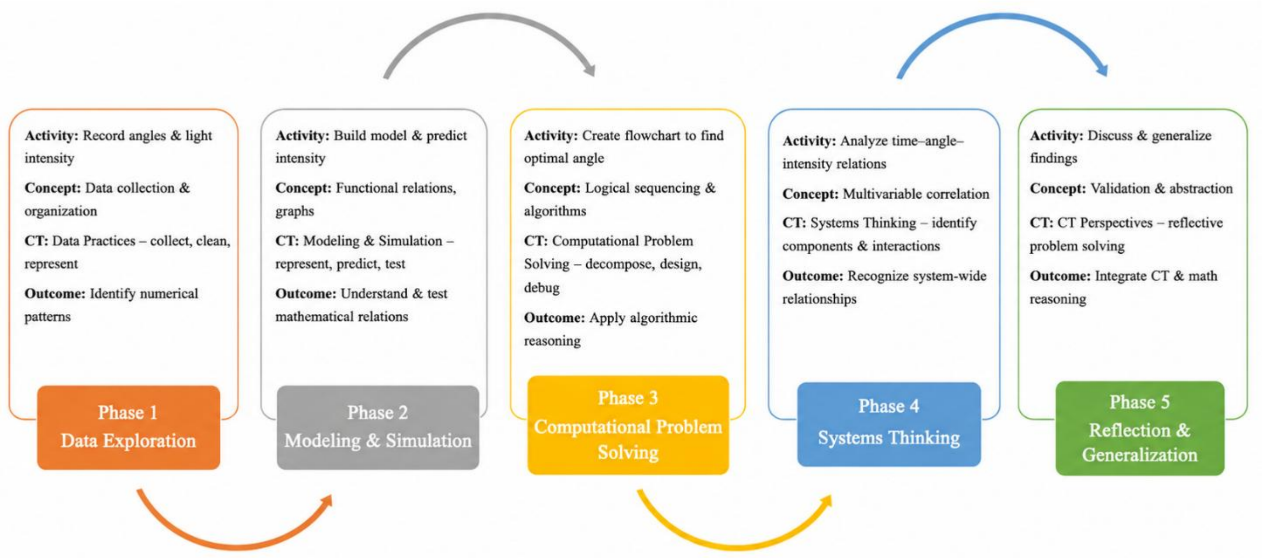

Figure 2. Computational Thinking–Mathematics Learning Trajectory (CT–MathLT)

Figure 1 presents the revised Computational Thinking–Mathematics Learning Trajectory (CT–MathLT) developed through the design research process. The trajectory consists of five interconnected phases: Data Exploration, Modeling and Simulation, Computational Problem Solving, Systems Thinking, and Reflection and Generalization. Each phase was designed to support the progressive development of Computational Thinking practices while simultaneously strengthening mathematical reasoning. The trajectory begins with Data Exploration, where students collect, organize, and interpret light-intensity data obtained from renewable energy investigations. At this stage, students primarily engage in arithmetic reasoning by comparing numerical values, calculating averages, and identifying patterns in the collected data. These activities provide the empirical foundation for subsequent learning phases. In the Modeling and Simulation phase, students transform numerical data into predictive representations by constructing and testing solar panel models. Through comparing predicted and observed outcomes, students move beyond simple numerical comparison and begin to engage in representational and predictive reasoning. This phase serves as an important bridge between arithmetic reasoning and more advanced forms of computational thinking.

The transition toward algorithmic thinking becomes most visible during the Computational Problem-Solving phase. Students formulate optimization procedures, express their reasoning through if then rules and flowcharts, and test these procedures using alternative datasets. Through cross-testing and debugging activities, students refine their decision-making processes and develop generalized procedures that can be applied to different situations. This progression reflects a shift from obtaining individual answers toward constructing reusable solution strategies.

The Systems Thinking phase further extends students' reasoning by encouraging them to examine relationships among time, angle, light intensity, and other influencing variables. Students learn to identify interactions among variables and analyze how changes in one component affect the behavior of the entire system. Finally, the Reflection and

Generalization phase consolidates learning by prompting students to evaluate their findings, abstract key principles, and connect Computational Thinking practices with mathematical reasoning. Taken together, these five phases illustrate a coherent progression from data-based arithmetic reasoning toward algorithmic and systemic forms of thinking, providing empirical support for the development of Computational Thinking within primary mathematics learning.

2.5.19 Learning Design Principles Derived from the Research Findings

The explicit integration of Computational Thinking (CT) practices within each activity was found to enhance the visibility of students' thought processes, enabling teachers to trace the development of their computational strategies more effectively. The provision of visual and procedural scaffolding—through step-by-step examples, templates, icons, and guiding lines—had a tangible impact on improving the quality of student products while accelerating their comprehension of task instructions. Furthermore, the implementation of cross-group testing, and revision logs fostered a habit of data-driven refinement, a defining feature of reflective CT practice. The use of real-world contexts that could be modeled in simple forms also supported students' i transition from observation to system generalization, facilitating a more integrated and meaningful understanding of variable relationships in CT-based mathematics learning.

2.6 Discussion

The findings indicate that the four practices of Computational Thinking (CT) can be meaningfully integrated into primary mathematics learning through the developed Learning Trajectory (LT). The high achievement in Modeling & Simulation and Computational Problem-Solving demonstrates that activities based on visual representation and algorithmic iteration strongly support students' computational ways of thinking. Meanwhile, the relatively lower outcomes in Systems Thinking and Data Practices suggest that these two domains require more explicit conceptual scaffolding.

Conceptually, these findings reveal that the integration of Computational Thinking (CT) in mathematics learning is not uniform across practices. Modeling & Simulation stands out because it is grounded in concrete and contextual activities students can manipulate physical models, make predictions, and directly verify outcomes. From a constructivist perspective Alanazi (2016) and Valente & Blikstein (2019), direct interaction with tangible artifacts stimulates the construction of knowledge through learning by making, where students build and test hypotheses through action. Weintrop et al., (2016) further emphasize that computational representations allow students to "see and inspect" the dynamics of mathematical systems in an exploratory manner, which in this study was manifested through simulations of renewable energy models. Thus, the strength of Modeling & Simulation reflects not only the success of the activity design but also provides empirical validation of constructionist principles in CT-based learning.

In contrast, Data Practices required students to engage with raw data that were often unstructured or inconsistent. Difficulties in maintaining measurement consistency and identifying outliers indicated that the procedural aspects of CT were not yet fully internalized. However, when scaffolding was strengthened with "raw–clean" data templates, threshold rules, and a column for documenting data refinement rationale a marked improvement in the quality of students' outputs was observed. This finding aligns with Vygotsky's concept of the zone of proximal development, suggesting that appropriate external supports (such as visual aids and step-by-step guidance) enable students to operate at higher cognitive levels. Similarly, Grover and Pea (2013) emphasize that at the primary level, procedural scaffolding is an essential condition for helping students internalize computational practices as independent modes of thinking.

The transition from arithmetic reasoning to algorithmic thinking was facilitated by a sequence of learning trajectory components, including pattern identification, flowchart construction, cross-testing, and debugging. Initially, students relied on direct numerical comparisons to identify optimal solutions. Through repeated opportunities to express, test, and revise procedures, they gradually shifted toward constructing generalized decision rules that could be applied across multiple datasets. This progression reflects an important movement from answer-oriented reasoning to process-oriented reasoning, a hallmark of algorithmic thinking. The debugging process not only addressed procedural errors but also fostered metacognitive awareness of problem-solving strategies. This finding aligns with Brennan and Resnick's (2012) view that CT encompasses not only technical skills but also habits of mind such as reflection, refinement, and optimization. Consequently, the flowchart–debug–revise phase within the Learning Trajectory functions as a form of cognitive apprenticeship, allowing students to emulate computational engineering practices in a simple yet authentic manner.

The relatively lower achievement in Systems Thinking reflects a natural cognitive boundary among primary school students. Their tendency to depict linear, one-directional relationships without accounting for multiple feedback loops indicates that systemic reasoning remains at an early developmental stage. This aligns with the findings of Ben-Zvi Assaraf and Orion (2010), who argued that the ability to identify causal and feedback relationships evolves gradually through repeated reflective experiences. Therefore, the revision of the Learning Trajectory by incorporating system diagram templates and guiding scenarios serves as an appropriate form of scaffolding to broaden students' conceptual representations toward a more systemic understanding.

The integration of the four Computational Thinking (CT) practices within this Learning Trajectory not only fosters computational skills but also enriches students' forms of mathematical reasoning. Data Practices cultivate evidence-based reasoning as students interpret data variations and derive quantitative generalizations from their measurements. Modeling & Simulation strengthen representational reasoning by enabling students to map real-world situations into manipulable and testable mathematical models. Meanwhile, Computational Problem-Solving sharpens procedural reasoning the ability to structure solution steps logically and sequentially while engaging in algorithm revision through debugging. Finally, Systems Thinking develops structural reasoning, allowing students to understand inter-variable relationships within a system and predict the effects of change. Collectively, these four CT practices interact synergistically with the three core modes of mathematical thinking reasoning, generalization, and representation (NCTM, 2020) positioning the CT–MathLT as a design that not only cultivates computational proficiency but also deepens the foundation of mathematical reasoning.

Cognitively, these findings indicate that CT practices do not operate in isolation but form an integrated cycle of mathematical thinking. Data Practices provide empirical inputs that guide students toward quantitative abstraction; Modeling & Simulation transform those data into testable mathematical representations; Computational Problem-Solving structures the exploratory process into orderly algorithmic steps; and Systems Thinking enables students to reconnect their models to real-world contexts through inter-variable relationships. This cycle illustrates that computational thinking functions as a mediating mechanism between observation and mathematical generalization—a finding that reinforces Weintrop et al.'s (2016) claim that CT is not a separate domain, but rather a new epistemology for modern mathematics learning.

From the perspective of Indonesian mathematics education, these findings provide preliminary insights for curriculum implementation, which emphasizes project-based learning and the development of the Profil Pelajar Pancasila. Activities integrating measurement, modeling, and systemic reflection foster contextual numerical literacy while cultivating computational dispositions from an early age. Thus, the developed CT–MathLT represents not merely a technical adaptation, but a substantive contribution to the pedagogical transformation of mathematics education in Indonesia—toward a paradigm of data literacy and system-based problem solving.

Methodologically, the high level of implementation fidelity (86–92%) indicates that the revised HLT is stable and feasible for broader application. However, the limited sample size and localized context suggest the need for replication across diverse school environments. Future studies may incorporate simple quantitative measurements (e.g., lux or voltage sensors) to strengthen the connection between CT practices and data-driven mathematical modeling. This direction aligns with the global trend of integrating machine learning into STEM education (Del Ser et al., 2021), emphasizing the importance of cultivating data-driven reasoning from the primary education level.

The findings suggest that the effectiveness of the CT–MathLT did not stem from isolated CT activities but from the deliberate sequencing of learning experiences. Data collection and refinement supported pattern recognition, modeling activities encouraged abstraction, flowchart construction promoted algorithmic reasoning, and system diagrams facilitated relational thinking. Together, these components created a coherent progression through which students moved from concrete arithmetic operations toward increasingly sophisticated forms of computational thinking.

Thus, this study contributes to the CT–Math literature by providing an empirical model demonstrating how the four Computational Thinking (CT) practices can be simultaneously implemented in primary mathematics through an iteratively designed Learning Trajectory (LT). The integration of concrete activities, visual scaffolding, and algorithmic reflection enables learning that not only facilitates conceptual understanding of mathematics but also cultivates systematic, reflective, and data-driven modes of thinking, competencies essential for mathematics in the twenty-first century.

4. Conclusion

This study demonstrates that the four Computational Thinking (CT) practices—Data Practices, Modeling & Simulation, Computational Problem-Solving, and Systems Thinking—can be meaningfully integrated into primary mathematics learning through a structured Computational Thinking–Mathematics Learning Trajectory (CT–MathLT). The findings indicate that students engaged in activities involving data collection, modeling, algorithmic reasoning, and system analysis, providing evidence of emerging progression from arithmetic reasoning toward algorithmic and systems-oriented thinking. These learning experiences also supported mathematical reasoning, representation, and generalization within an authentic renewable energy context.

From a pedagogical perspective, the study highlights the potential of combining project-based learning, visual representations, and procedural scaffolding to support the development of CT in primary mathematics classrooms. The CT–MathLT developed in this study offers a practical design framework that teachers may adapt when integrating CT practices into mathematics learning activities aligned with the goals of the Merdeka Curriculum. Several limitations should be acknowledged. The study was conducted in a single school with a relatively small number of participants, and the findings are therefore context specific. Consequently, the results should be interpreted as design-based evidence rather than as a basis for broad curriculum recommendations. Future research is needed to investigate the implementation of the CT–MathLT across different school contexts, grade levels, and mathematical topics, as well as to examine its long-term impact on students' computational and mathematical thinking development.

References

Akker, J. V. den, Gravemeijer, K., McKenney, S., & Nieveen, N. (Eds.). (2006). *Educational Design Research*. Routledge. https://doi.org/10.4324/9780203088364

Alanazi, A. (2016). A Critical Review of Constructivist Theory and the Emergence of Constructionism. *American Research Journal of Humanities and Social Sciences*, *2*, 1–8. https://doi.org/10.21694/2378-7031.16018

Clark-Wilson, A., Robutti, O., & Thomas, M. (2020). Teaching with digital technology. *ZDM*, *52*(7), 1223–1242. https://doi.org/10.1007/s11858-020-01196-0

Gonzalez-Jimenez, D., del-Olmo, J., Poza, J., Garramiola, F., & Madina, P. (2021). Data-Driven Fault Diagnosis for Electric Drives: A Review. *Sensors*, *21*(12), 4024. https://doi.org/10.3390/s21124024

Gravemeijer, K., & Cobb, P. (2006). Design research from a learning design perspective. In *Educational Design Research*. Routledge.

Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, *126*, 296–310. https://doi.org/10.1016/j.compedu.2018.07.004

Jiménez, D., Delerce, S., Dorado, H., Cock, J., Muñoz, L. A., Agamez, A., & Jarvis, A. (2019). A scalable scheme to implement data-driven agriculture for small-scale farmers. *Global Food Security*, *23*, 256–266. https://doi.org/10.1016/j.gfs.2019.08.004

Kannadass, P., Hidayat, R., Siregar, P. S., & Husain, A. P. (2023). Relationship Between Computational and Critical Thinking Towards Modelling Competency Among Pre-Service Mathematics Teachers. *TEM Journal*, 1370–1382. https://doi.org/10.18421/TEM123-17

Khalil, R. Y., Tairab, H., Qablan, A., Alarabi, K., & Mansour, Y. (2023). STEM-Based Curriculum and Creative Thinking in High School Students. *Education Sciences*, *13*(12), 1195. https://doi.org/10.3390/educsci13121195

Lodi, M., & Martini, S. (2021). Computational Thinking, Between Papert and Wing. *Science & Education*, *30*(4), 883–908. https://doi.org/10.1007/s11191-021-00202-5

Mylonas, G., Paganelli, F., Cuffaro, G., Nesi, I., & Karantzis, D. (2023). Using gamification and IoT-based educational tools towards energy savings—Some experiences from two schools in Italy and Greece. *Journal of Ambient Intelligence and Humanized Computing*, *14*(12), 15725–15744. https://doi.org/10.1007/s12652-020-02838-7

Pei, C. (Yu), Weintrop, D., & Wilensky, U. (2018). Cultivating Computational Thinking Practices and Mathematical Habits of Mind in Lattice Land. *Mathematical Thinking and Learning*, *20*(1), 75–89. https://doi.org/10.1080/10986065.2018.1403543

Pei, C.-H. (2018). Computational thinking in STEM education: A review and future directions. *Eurasia Journal of Mathematics, Science and Technology Education*.

Sarama, J., & Clements, D. H. (2019). *Chapter 2 Learning Trajectories in Early Mathematics Education*. Brill. https://doi.org/10.1163/9789004396449_002

Sheriff, N., & Sevukan, R. (2023). Discovering research data management trends from job advertisements using a text-mining approach. *Journal of Information Science*, 01655515231193845. https://doi.org/10.1177/01655515231193845

Shin, N., Bowers, J., Krajcik, J., & Damelin, D. (2021a). Promoting computational thinking through project-based learning. *Disciplinary and Interdisciplinary Science Education Research*, *3*(1), 7. https://doi.org/10.1186/s43031-021-00033-y

Shin, N., Bowers, J., Krajcik, J., & Damelin, D. (2021b). Promoting computational thinking through project-based learning. *Disciplinary and Interdisciplinary Science Education Research*, *3*(1), 7. https://doi.org/10.1186/s43031-021-00033-y

Shute, V. J., Sun, C., & Asbell-clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Turgut, M., Kohanová, I., & Gjøvik, Ø. (2024). Developing survey-based measures of mathematics teachers' pedagogical technology knowledge: A focus on computational thinking and programming tools. *Research in Mathematics Education*, 1–24. https://doi.org/10.1080/14794802.2024.2401482

Valente, J. A., & Blikstein, P. (2019). Maker Education: Where Is the Knowledge Construction? *Constructivist Foundations*, *14*(3), 252–262.

Wehner, A., Masanek, N., Hellmann, K., Heinz, T., Grospietsch, F., & Glowinski, I. (Eds.). (2024). *Vernetzung von Wissen bei Lehramtsstudierenden: Eine Black-Box für die Professionalisierungsforschung?* Verlag Julius Klinkhardt. https://doi.org/10.35468/6118

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016a). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016b). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Wing, J. M. (2006). Computational Thinking. *Communication of The ACM*, *49*(3), 33–35.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, *14*(1). https://doi.org/10.1145/2576872

Ye, H., Liang, B., Ng, O.-L., & Chai, C. S. (2023a). Integration of computational thinking in K-12 mathematics education: A systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education*, *10*(1), 3. https://doi.org/10.1186/s40594-023-00396-w

Ye, H., Liang, B., Ng, O.-L., & Chai, C. S. (2023b). Integration of computational thinking in K-12 mathematics education: A systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education*, *10*(1), 3. https://doi.org/10.1186/s40594-023-00396-w

Yeni, S., Nijenhuis-Voogt, J., Saeli, M., Barendsen, E., & Hermans, F. (2024). Computational thinking integrated in school subjects – A cross-case analysis of students' experiences. *International Journal of Child-Computer Interaction*, *42*, 100696. https://doi.org/10.1016/j.ijcci.2024.100696