15 (1) (2026) 22-40
Unnes Science Education Journal
Accredited Sinta 2
https://journal.unnes.ac.id/journals/usej

Applying the Computational Thinking for Science Framework to Enhance Students' Computational Thinking in Renewable Energy Topics

Dian Purnama✉**, Lilik Hasanah, Irma Rahma Suwarma**

DOI: http://dx.doi.org/10.15294/usej.v13i1.39854

Universitas Pendidikan Indonesia, Indonesia

Article Info

Submitted 2025-12-24
Revised 2026-01-26
Accepted 2026-04-08

Keywords

Alternative Energy; Computational thinking; CT-S framework; Rasch Model; VPA

Abstract

This study investigates the effectiveness of the Computational Thinking for Science (CT-S) framework in improving high school students' computational thinking (CT) skills in renewable energy learning. Results indicate that CT-S-based instruction resulted in marked improvement in students' computational thinking performance. The study employed an embedded mixed-method design in which qualitative data served as the primary source of evidence, supported by quantitative analysis. The quantitative component used a pre-experimental one-group pretest-posttest design. Rasch modelling was applied to examine changes in item difficulty, person ability, and CT ability levels. Qualitative data were collected through Verbal Protocol Analysis (VPA) to capture students' cognitive processes during learning activities. CT-S-aligned activities engaged students in data analysis, modelling simulations, solar-panel experimentation, and the design of an automated solar-tracking system. Qualitative findings show progressive development of CT components. Abstraction dominated early data interpretation, while decomposition and algorithmic thinking emerged during modelling and experimentation. Advanced problem-solving became evident during the engineering design activity. Quantitative results confirmed substantial improvement. Mean item difficulty decreased from 0.25 to -0.33 logits, mean person ability increased from 0.04 to 0.56 logits, and the number of students reaching the highest CT level increased from 10 to 31, with no decline observed. Overall, the findings demonstrate that CT-S-based instruction effectively strengthens students' conceptual and procedural computational thinking skills in science learning.

How to Cite

Purnama, D., Hasanah, L., & Suwarma, I. R. (2026). Applying the Computational Thinking for Science Framework to Enhance Students' Computational Thinking in Renewable Energy Topics. *Unnes Science Education Journal*, 15(1), 22-40.

✉ Correspondence Author:
E-mail: purnamadian@upi.edu

p-ISSN 2252-6617
e-ISSN 2502-6232

INTRODUCTION

Rapid technological and societal developments have shifted educational priorities toward higher-order skills essential for navigating modern scientific and technological environments (Silva, 2009; Trilling, 2009). Within this shift, computational thinking (CT) has emerged as a core 21st-century competency due to its central role in scientific inquiry and computational practices (Wing, 2006, p. 33, 2011, pp. 2–4), a position reinforced by frameworks such as the Next Generation Science Standards, which explicitly include CT as one of eight scientific practices (NGSS Lead States, 2013). CT is inherently tied to science learning because computational tools deepen conceptual understanding and strengthen the interplay between computation, mathematics, and science (Weintrop et al., 2016). However, in Indonesia, the integration of CT remains limited and inconsistently implemented. Students often have minimal prior exposure to CT or programming (Khasyyatillah & Osman, 2019). Teachers' conceptual understanding of CT is also low, with fewer than five percent able to define CT accurately (Irvani et al., 2024). In addition, the absence of a unified national framework leads to fragmented interpretations across regions and subjects (Budiyanto et al., 2025, p. 10). Assessment practices rely heavily on self-reports and mathematics-based tasks that fail to capture core CT dimensions such as abstraction and debugging (Budiyanto et al., 2025; Sondakh et al., 2022), while limited teacher readiness and technological constraints further hinder classroom enactment (Irvani et al., 2024). These conditions reveal a persistent gap between policy aspirations and classroom realities, signaling the urgent need for structured CT-oriented instructional designs and valid assessment tools in Indonesian science education.

Despite its importance, computational thinking (CT) has been defined in diverse ways across the literature. Wing (2006) describes CT as a way of solving problems, designing systems, and understanding human behaviour using concepts from computer science. Hemmendinger (2010) emphasizes the use of computation as a way of thinking and acting within disciplinary contexts. CT has also been described as the process of formulating problems and solutions that can be executed by an information-processing agent (Cuny et al., 2010, as cited in Wing, 2011). Other perspectives emphasize recognizing computational structures in natural and artificial systems (Furber, 2012) or engaging in programming practices (Brennan & Resnick, 2012). Other scholars emphasize abstraction and automation as key aspects of CT (Yadav et al., 2014). Additional perspectives highlight logical reasoning (Csizmadia et al., 2015) and creative problem solving (Cansu & Cansu, 2019). Correspondingly, CT is often conceptualized as two broad categories. The first involves computational problem-solving skills grounded in general and domain-specific reasoning. The second includes technical computational skills related to programming and computing concepts (Tang et al., 2020). This definitional variation highlights the lack of consensus that complicates the development of coherent CT pedagogies and assessments, particularly in science education.

Although numerous frameworks describe the core components of computational thinking (CT), most conceptualize CT as universal problem-solving or computer science domains (Angeli et al., 2016; Barr & Stephenson, 2011; Brennan & Resnick, 2012; Csizmadia et al., 2015; Lee et al., 2011; Selby & Woollard, 2013; Yadav et al., 2014). In contrast, CT for science and mathematics frameworks conceptualize as embedded within disciplinary practices, emphasizing competencies such as data, modelling and simulation, computational problem solving, and systems thinking practices (Weintrop et al., 2016). More recent science-specific formulations extend this view by situating CT within core scientific activities (data collection, data processing, modelling, and problem-solving) mediated through reflective use, design, and evaluation of computational tools (Hurt et al., 2023). These distinctions highlight the need for a CT framework explicitly aligned with the epistemic and practical demands of science education.

Interest in integrating computational thinking (CT) into science education has grown substantially in recent years (Tekdal, 2021). In physics learning, CT-oriented interventions have been shown to strengthen students' conceptual understanding and problem-solving skills, whether through CT-based science modules (Lapawi et al., 2020), related to developing module in physics (Khasyyatillah & Osman, 2019), to improve and assess physics and engineering integrated CT skills through developing maker activities and assessment (Yin et al., 2020), or integrating CT into science learning trough modelling and programming (Aksit & Wiebe, 2020; Hutchins et al., 2020). Robot programming improved Newtonian physics learning faster than traditional lessons while boosting teamwork and skills in coding (Ferrarelli & Iocchi, 2021). Although these studies demonstrate the benefits of integrating CT into physics, most focus on specific tools, programming platforms, or isolated physics concepts. As a result, fewer studies examine how a science-specific framework such as CT-S can structure learning across a sequence of scientific and engineering activities. In addition, little research has explored the application of CT-S within renewable energy contexts, where students must reason about complex energy systems, interpret real-world datasets, and design technology-based solutions.

Integrating computational thinking into science learning can be supported by the Computational Thinking for Science (CT-S) framework, which organizes learning activities around scientific practices such as data collection, data processing, modelling, and problem solving while encouraging students to apply computational thinking skills during these activities (Hurt et al., 2023). The framework integrates two complementary dimensions: Scientific Activities comprising Data Collection, Data Processing, Modelling, and Problem Solving and Cognitive Processes, including Reflective Use, Design, and Evaluation, which together specify how learners interact with computational tools to construct and refine scientific reasoning. Evidence shows that CT-S reliably predicts students' science learning across diverse populations and is supported by strong psychometric validity grounded in reflective use, design, and evaluation processes (Cannady et al., 2025). However, even as CT becomes more widely integrated into physics instruction, little is known about whether a science-specific framework such as CT-S can enhance students' general computational thinking skills in interdisciplinary contexts. Renewable energy learning provides a particularly relevant setting because students must analyze energy data, model energy transformations, and design technological solutions such as solar tracking systems. This context offers opportunities to examine how CT-S activities support computational reasoning within authentic energy-system problems. This question is particularly important in topics like alternative energy, where learners must decompose complex systems, identify patterns, abstract meaningful relationships, and apply algorithmic logic. To investigate this relationship, the present study adopts the general computational thinking components proposed by Selby and Woollard (2013), which includes decomposition, pattern recognition, abstraction, and algorithmic thinking. While the CT-S framework structures the learning activities, these CT components are used to evaluate students' computational thinking performance.

Understanding how CT-S supports students' computational reasoning in physics requires an analytic method that captures real-time cognitive processes. Verbal Protocol Analysis (VPA), which examines think-aloud verbalizations during problem-solving tasks, offers direct insight into underlying reasoning and has long been used in physics education, from classic analyses of problem-solving strategies to contemporary studies of expert–novice thinking to systematic reviews highlighting think-aloud methods as central to studying physics problem solving (Docktor & Mestre, 2014). Recent studies have further used protocol analysis to document computationally mediated reasoning in physics-based programming tasks (Taub et al., 2015). Accordingly, this study investigates how the CT-S framework supports the development of high school students' computational thinking in alternative energy physics learning. Specifically, it examines: (1) Specifically, it examines: (1) the extent to which students' computational thinking (CT) skills improve after learning activities implemented using the CT-S framework, and (2) the characteristics of students' cognitive processes when engaging with CT-S tasks, analyzed through Verbal Protocol Analysis. In this study, the CT-S framework functions as the instructional structure for learning activities, while computational thinking (CT) refers to the cognitive skills demonstrated by students and measured through the assessment instrument.

METHOD

This study Applied a mixed-methods explanatory design, integrating quantitative measurement of learning outcomes with qualitative analysis of students' cognitive processes. An embedded mixed-method design was employed, in which qualitative data served as the primary source of evidence, while quantitative data were embedded to support, reinforce, and extend the qualitative findings (Creswell, 2012). The quantitative component followed a pre-experimental one-group pretest–posttest design (Sugiyono, 2013, p. 74) to examine the effect of the intervention on students' computational thinking (CT). A total of 45 Grade 10 students from a public senior high school in an urban area participated.

The learning intervention was implemented in a renewable-energy physics unit. The lessons progressed from contextual problems about fossil-fuel dependence to the optimization of solar energy using an automated sun-tracking system. Across three to four 60-minute sessions, instruction followed the Computational Thinking for Science (CT-S) framework (Hurt et al., 2023), emphasizing reflective use of computational tools as students engaged with Tinkercad, PhET simulations, and Arduino block-based coding. Five learning activities were designed based on the CT-S framework, including data collecting, data processing, modelling, and problem solving. These activities were intended to stimulate core computational thinking components such as abstraction, decomposition, algorithmic thinking, evaluation, and generalization. (Dagiene et al., 2017; Selby & Woollard, 2013). Students worked in small groups during modelling tasks, with individual cognitive-prompt responses collected via Google Forms to trace real-time reasoning each activity. Data Collecting occurred in Activities 1, 2, and 4 through analysis of national fossil-fuel datasets, exploration of renewable-energy potential, and mini solar-cell measurements; Data Processing in Activities 2 and 4 involved comparing, organizing, and interpreting datasets; Modelling in Activity 3 required constructing simplified representations of energy-conversion processes; and Problem Solving in Activity 5 engaged students in generating pseudocode for an automatic solar-tracking system. All sessions were supported by web-capable student devices, and audio plus static video recordings captured interactions during modelling, with cognitive processes later examined through verbal protocol analysis (Ericsson & Simon, 1993; Taub et al., 2015).

Students' computational thinking was assessed using a 13-item multiple-choice instrument with reasoning components. The instrument measured five CT components based on Selby and Woollard (2013, p. 5), which were further operationalized through the ten from fourteen indicators proposed by Dagiene et al. (2017, p. 37), as detailed in Table 1. The distribution of items across indicators was not intended to be uniform because each indicator represents different levels of conceptual complexity. Some indicators were assessed using a single item when the task required a focused response targeting a specific computational thinking skill. Other indicators were represented by multiple items to capture more complex reasoning processes and to increase measurement stability within the Rasch framework. This variation in item distribution is consistent with Rasch measurement principles, where item difficulty and conceptual coverage are more important than equal item counts across categories (Bond & Fox, 2015; Boone et al., 2014). The assessment had undergone expert validation and empirical testing with 95 students who had previously studied renewable energy and engaged in Arduino-based learning. The CT pre-test was administered before the intervention, and the post-test was conducted afterward under equivalent conditions. Each indicator was operationalized through one or more items depending on the complexity of the targeted computational thinking process.

Table 1. Computational Thinking Aspects, Indicators, and Item Distribution Used in the Assessment Instrument

Computational Thinking Aspect	Indicator	Item
Abstraction	AB.1. Removing unnecessary skills	S08
	AB.2. Spotting key element	S01
Algorithmic thinking	AT.1. Thinking in terms of sequences and rules	S11
	AT.2. Creating an algorithm	S07, S12
Decomposition	DC.1. Breaking down tasks	S06
	DC.2. Thinking about problems in terms of component parts	S03
Evaluation	EV.1. Finding best solution	S04
	EV.2. Making decisions about good use of resources	S05, S13
Generalization	GN.1. Identifying patterns as well as similarities and connections	S02, S10
	GN.2. Utilizing the general solution	S09

Dagiene et al. (2017)

Quantitative data were analysed using the Rasch model to examine students' computational thinking performance across the pre- and post-measures. Prior to the stacking and racking analyses, item calibration was conducted to ensure that the assessment instrument functioned consistently within the Rasch measurement framework. Item fit was evaluated using infit and outfit mean square statistics following commonly accepted Rasch criteria (Bond & Fox, 2015; Boone et al., 2014). All items demonstrated acceptable fit, indicating that the instrument provided reliable measurement of students' computational thinking abilities.

After confirming item functioning, stacking and racking procedures were applied to characterise changes in student performance across the pre- and post-measures. The racking analysis examined shifts in item difficulty, allowing each CT aspect to be mapped in terms of increased or decreased challenge following instruction. The stacking analysis enabled a more granular evaluation of individual students' improvement by comparing person estimates across the two measurement points. Person and item reliability indices were also examined to ensure the stability of the measurement model.

Qualitative data were analysed using Verbal Protocol Analysis (VPA). Throughout the learning sessions, students articulated their reasoning in response to targeted cognitive prompts delivered through group modelling activities were captured using audio and static video recordings and individual responds using Google Forms. All qualitative data were transcribed, segmented into cognitive units, and manually coded according to CT-S Activity and CT's aspect categories. Triangulation across think-aloud responses, transcripts, worksheets, and digital artefacts (Tinkercad solar-tracking models) ensured analytic credibility and enabled identification of reasoning patterns associated with the reflective use of computational tools.

RESULT AND DISCUSSION

The qualitative findings are based on Verbal Protocol Analysis (VPA) of classroom interactions during five CT-S–aligned activities, captured through a static video camera for whole-class discourse and an audio recorder attached to the teacher for guided group interactions. All verbalizations were transcribed, segmented, and indexed with speaker labels to enable detailed tracing of computational thinking processes. To complement the transcripts, students completed reflection prompts via Google Form after each activity, providing individual evidence of emerging abstraction, algorithmic thinking, decomposition, evaluation, and Generalization. This combination of classroom VPA and individual reflections aligns with recommended approaches for capturing in-situ cognitive processes in computational thinking research.

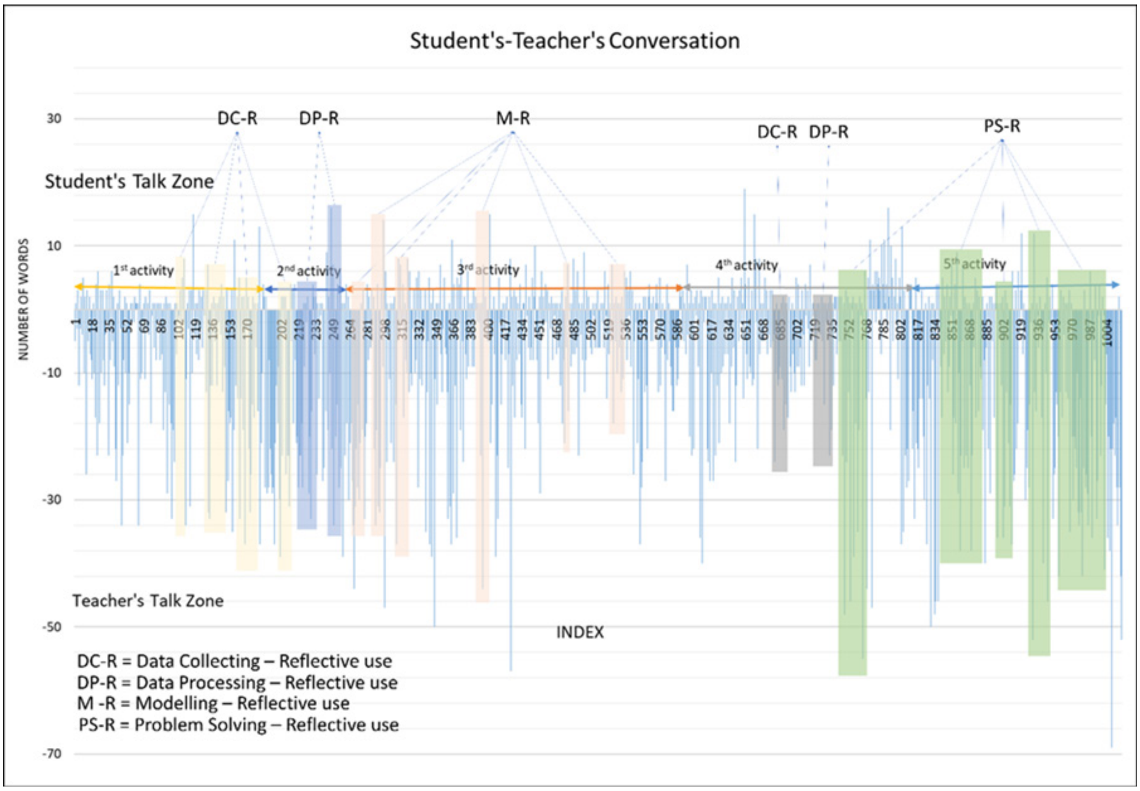

Figure 1. Distribution of teacher and student talk across CT-S learning activities

Figure 1 illustrates how classroom discourse patterns shift throughout the CT-S learning sequence, reflecting varying levels of teacher scaffolding and student engagement across different instructional phases. The figure presents distribution of teacher and student talk across five CT-S activities: Data Collecting-Reflective Use (DC-R), Data Processing-Reflective Use (DP-R), Modelling-Reflective Use (M-R), and Problem Solving-Reflective Use (PS-R). Each activity segment is color-coded to highlight how participation patterns change as learning progresses from conceptual exploration toward modelling and problem solving. Student discourse increases during the Data Collecting, Data Processing, and Modelling phases, whereas teacher guidance becomes more prominent during the Problem Solving phase, reflecting the higher cognitive and technical demands associated with this stage of CT-S instruction. These patterns illustrate how teacher scaffolding supports students' engagement with computational thinking processes during different phases of the CT-S learning sequence.

Activity 1 introduced the fossil-energy problem through teacher-guided discussion and initial engagement with Indonesia's primary energy and emission context. Classroom talk patterns (Indices 1–184) showed strong teacher dominance (20–35 words per segment) with short student responses (1–8 words), reflecting an early problem-orientation phase where scaffolding is needed to frame the issue and activate prior knowledge (Hmelo-Silver et al., 2007; Weintrop et al., 2016). Classroom talk Indices 188–250 in activity 2 reflects a continuation of teacher-led scaffolding, with teacher utterances ranging from 15 to 45 words and student responses between 1 and 10 words. Given the limited session time, much of the data-collection work shifted to group worksheets, while the brief whole-class exchanges provided essential conceptual grounding before students engaged independently with renewable-energy datasets (Hmelo-Silver et al., 2007; Weintrop et al., 2016)

In Activity 3, classroom discourse Indices 263–584 reflects a modelling-focused phase in which the teacher again dominated talk (around 15–60 words per segment), while student utterances remained short (1–10 words) but more frequent than in previous activities. This pattern is typical of a Modelling–Reflection stage, where extended teacher explanations are needed to establish core ideas about energy forms, energy flow, and conservation before students work more independently with the PhET Energy Forms and Changes simulation (Lehrer & Schauble, 2015; Sengupta et al., 2013). Classroom talk Indices 589–732 in activity 4 shows short, dense interaction patterns that reflect a compressed experimental session on mini solar panels. Teacher utterances (about 10–40 words per segment) again dominated, as the teacher provided technical instructions, partial experimental data (light intensity and current), and initial guidance on interpreting results. Student contributions remained brief (1–8 words) but frequent, mostly clarifications, confirmations of measurements, or naming components, indicating active but highly focused participation under time constraints.

In Activity 5, the automatic solar-tracker project, classroom talk across Indices 735–1008 showed the most intensive and fluctuating interaction pattern in the sequence. Teacher utterances (about 10–70 words per segment) still dominated, reflecting the high technical complexity of integrating LDR sensors, servo motors, Arduino code, and circuit wiring in students' first experience with such a system. Student contributions were short (1–10 words) but frequent, mostly in the form of questions about wiring errors, sensor readings, and servo movement, which is typical of design–debug–iterate phases in technology-rich problem solving (Bers et al., 2014; Weintrop et al., 2016). Time constraints meant that many groups only reached the simulation stage, with relatively few completing a fully functional physical prototype, so discourse concentrated on making the system work rather than on systematic comparison between static and tracking panels.

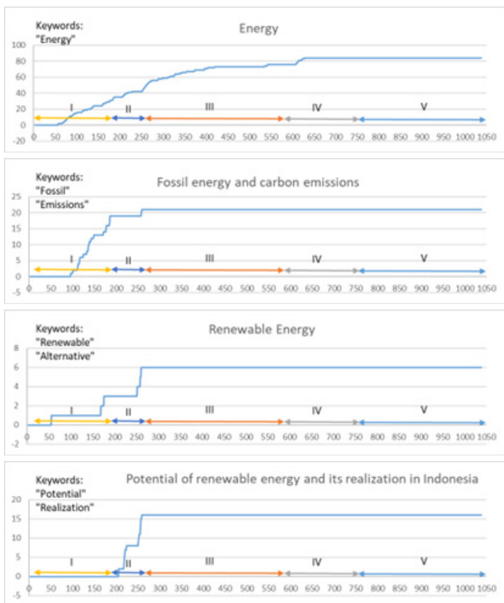

Figure 2. Keyword distribution from classroom discourse during early CT-S activities (Activities 1–3)

Figure 2. highlights dominant conceptual terms used by teachers and students while discussing fossil energy, renewable energy potential, and energy transformation processes. For Activity 1, Figure 2 shows that the keyword analysis reinforces the conceptual framing of the discussion. The term energy appeared approximately 30 times, while fossil energy and carbon emissions each appeared around 14 times. In contrast, references to alternative energy, solar energy, solar panel, and solar tracker occurred only 1–3 times. This distribution indicates that classroom discourse was primarily centered on characterizing fossil-energy problem, while only beginning to introduce cues toward renewable-energy contexts that would become more prominent in subsequent activities (Sengupta et al., 2013). For Activity 2, Figure 2 shows a clear shift in keyword patterns compared to Activity 1. The term energy appeared about 10 times, while potential and realization occurred 16 times, energy change 6 times, and solar energy 5 times. This distribution indicates a transition from framing fossil-energy problem toward early engagement with renewable-energy opportunities and emerging solar-focused project phase.

For Activity 3, keyword analysis in Figure 2 shows energy remained dominant, appearing about 40 times, while alternative energy appeared 3 times. Figure 3 further shows energy change appeared 13 times, solar energy 16 times, and solar panel 9 times. Together, these frequencies indicate that classroom discourse was strongly oriented toward understanding energy transformation and explicitly linking it to solar-energy systems, thereby bridging abstract scientific ideas with technological context needed subsequent activities.

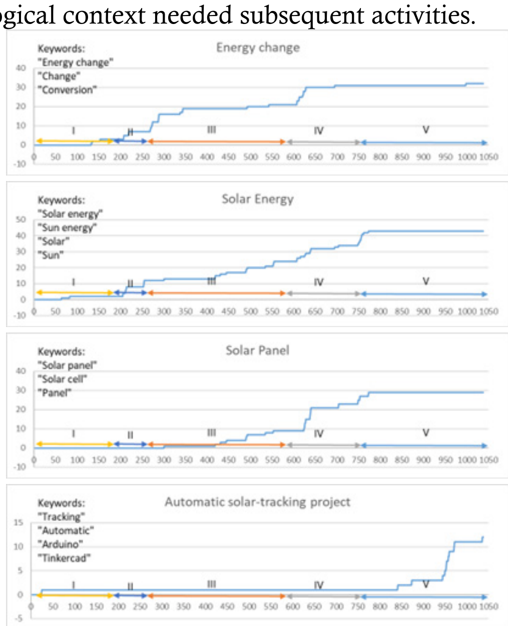

Figure 3. Keyword distribution from classroom discourse during later CT-S activities (Activities 4–5)

Figure 3. reflects a shift toward technical and design-oriented terms associated with solar energy systems and the solar-tracking project.

For Activity 4, Figure 3 shows that energy changes appeared about 12 times, solar energy 10 times, and solar panel 12 times, indicating that classroom discourse was concentrated on interpreting transformation processes and connecting them to the operational components of the solar-tracking system. In comparison, Figure 2 shows that the broader keyword energy decreased to around 10 occurrences, reflecting a shift away from general conceptual framing toward more specific, system-level reasoning. Together, these patterns suggest that students' attention was increasingly directed toward the concrete mechanisms of solar-energy conversion and the functioning of the experimental apparatus, even though aspects of the data-collection procedure were partially pre-structured by the teacher.

For Activity 5, Figure 3 shows that solar energy and solar panel appeared 6 and 7 times, respectively, indicating that conceptual discussions about energy had largely been established in earlier activities. In contrast, Figure 4 shows that solar tracking project emerged as the most frequent term (11 occurrences), reflecting a shift toward system implementation and technical problem solving. This distribution is consistent with algorithmic problem-solving phases where language becomes dominated by technical devices, control rules, and troubleshooting rather than by basic science concepts (Bers et al., 2014; Weintrop et al., 2016).

The keyword patterns also illustrate how classroom discourse evolved throughout the CT-S learning sequence. Early activities emphasized conceptual understanding of fossil and renewable energy systems, while later activities increasingly involved technical and design-oriented language related to solar panels and automated tracking systems. This progression suggests that CT-S activities gradually shifted students' reasoning from conceptual exploration toward system-level and engineering-oriented problem solving.

In Activity 1, evidence of CT-S implementation emerged clearly in the transcript, while students' responses revealed the development of several computational thinking processes. A representative example from Index 110 shows the teacher explaining why fossil fuels are non-renewable ("formed through geological processes that take millions of years"), prompting students to reinterpret their earlier statements about electricity and transportation needs (Indices 98–117). This marks the start of Reflective Use, where students refine conceptual understanding before working with numerical data (Hurt et al., 2023). Later, as students examined the primary-energy graph, reflective data collection became explicit. Students identified key categories a grouped them meaningfully. Teacher prompts drew attention to dominant patterns ("fossil sources increase consistently each year"), enabling students to summarize system-level trends verbally rather than read off exact values. Across these segments, students demonstrated core forms of abstraction: (1) removing irrelevant detail (focusing on major energy groups rather than minor categories), (2) identifying key system components (dominance of fossil fuels, limited renewables), and (3) generating verbal representations (e.g., "our fossil-energy use keeps rising"), consistent with CT frameworks by Selby and Woollard (2013) and Dagiene et al. (2017).

98	*T: Now I want to ask-some of you have already filled it in, I saw earlier. The first question: What will happen if the world runs out of fossil energy?*
99	*S: Darkness.*
100	*T: Darkness, okay. Whose answer was that? What's your name?*
104	*S: A resource crisis.*
105	*T: Why a resource crisis?*
106	*S: Because it would be shocking, Sir. Suddenly it's all gone.*
107	*T: Any other answers?*
108	*S: Transportation will be significantly affected because it needs fuel.*
110	*T: Question number one: Can fossil energy run out? Yes or no?*
111	*S: Yes.*
112	*T: Because its source comes from what?*
113	*S: Fossils, remains of life.*
111	*S: Yes.*
112	*T: Because its source comes from what?*
113	*S: Fossils, remains of life.*
114	*T: Yes, from remains of ancient life. It can form again, but the process takes millions of years. So, it absolutely can run out. Then, the second question: What's the answer? Why is fossil energy used so widely?*
115	*S: Because it is very useful.*
116	*T: Useful, okay. Why is it widely used and useful? What's the reason?*
117	*S: Because nowadays it's needed. For example, if there's no electricity, how would we charge our phones.*

Triangulation data from 42 Google Form responses reinforced these findings. For Question 1, 38 students demonstrated CT by articulating the causal chain linking fossil-energy consumption, rising emissions, and global warming. For Question 2, 35 students extracted the core problem—fossil dependence, rising emissions, limited renewable adoption. For Question 3, 24 students revealed CT in recognizing the interpretive challenges of reading graphs. For Question 4, 35 students produced coherent cause–effect explanations involving pollution, greenhouse gases, and climate change. These written responses mirror the abstraction processes observed in the transcript and align with research showing that visual representations and causal explanations help students articulate principled, computationally relevant reasoning (Lehrer & Schauble, 2015; Taub et al., 2015).

Overall, Activity 1 effectively instantiated Data Collecting – Reflective Use within the CT-S framework. Teacher scaffolding, conceptual prompts, and visual representations supported students in constructing a system-level understanding of fossil-energy trends and environmental impacts. Convergent evidence from talk patterns, keyword distributions, transcript indices, and individual reflections shows that abstraction emerged as the dominant CT skill, forming the conceptual and cognitive foundation for the more data-intensive, modelling, and problem-solving activities that followed.

In Activity 2, transcript evidence illustrates how Data Collection–Reflection and Data Processing–Reflection emerged. Students began extracting structured information from articles and graphs, identifying energy categories, units, and distinguishing between potential and actual utilization. In Indices 210–212, the teacher supported students' interpretation of megawatt units and guided them to make sense of the substantial gap between potential and realization, aligning with the CT-S emphasis on reflection as constructing meaning from partially structured data (Hurt et al., 2023). Computational thinking became evident in students' generalization and evaluation, they identified patterns linking high potential to low realization and leveraged these patterns to make informed judgments about which renewable sources should be prioritized, consistent with CT indicators described by Selby & Woollard (2013) and Dagiene et al. (2017). A follow-up review in Indices 242–250 further consolidated these processes. Students readily recalled solar energy as the source with the highest national potential and explained the magnitude of the potential–realization gap. The teacher then extended the reflection by linking this disparity to Indonesia's ongoing fossil-fuel dependency (Indices 248–250), prompting evaluative reasoning about energy-transition priorities (E2).

210	*T: I'll show you the table. This is what you need to fill in. Here, this part needs to be completed. If you can't fill it in directly, just write it on paper first and take a photo later. What you need to find is the information.*
211	*S: What is MW, Sir?*
212	*T: MW is megawatt, a unit. Megawatt.*
242	*T: Yesterday each group filled this in, right? What table was it? Potential of what?*
243	*S: Energy, renewable energy in Indonesia, and how its utilization is being realized.*
244	*T: Okay. Yesterday you also filled out a Google Form, right? Now, what do you think is the energy with the largest potential in Indonesia?*
245	*S: Solar energy.*
246	*T: According to you, solar energy? Solar energy. Why solar energy, you?*
247	*S: Because its national potential is around 20,700 MW, while only 123 MW has been utilized.*
248	*T: Okay, so the potential is large, but the realization is small, right? Then before talking about this potential, what was the earlier problem related to—our dependence on what energy source?*
249	*S: Fossil energy. We depend on fossil energy.*
250	*T: We can see that in Indonesia, primary energy use is mostly fossil energy. Then in 2018, it began to shift, transitioning toward renewable energy. That's why the realization of renewable energy utilization has increased, as shown in the graph at the beginning.*

Survey responses from 36 students triangulate these findings. For Question 1, 22 students demonstrated CT by explaining how tables or graphs helped them detect patterns or justify the use of specific representations. For Question 2, 28 students correctly generalized that solar energy has the highest potential and related this to Indonesia's geographic conditions. For Question 3, 23 students used data to evaluate renewable-energy priorities, and Question 4 yielded a similar count (23 students) articulating conclusions about the potential–realization gap and the urgency of energy transition. Responses lacking CT were typically short, lacked data references, or drifted toward non-analytical statements.

Overall, Activity 2 strengthened CT-S implementation by engaging students in reflective data collection and processing using national renewable-energy datasets. Conceptually, discourse shifted from fossil-energy concerns (fossil energy, carbon emissions) to renewable-energy opportunity structures (potential, realization, solar energy). Computationally, students demonstrated the capacity to generalize patterns across datasets and evaluate evidence-based priorities for Indonesia's energy transition. These emerging skills form a crucial bridge to subsequent CT-S activities that require deeper modelling and problem-solving engagement.

In Activity 3, transcript data illustrate how Modelling–Reflection and decomposition emerged in tandem. A representative segment in Indices 379–396 shows the teacher guiding students to unpack the geothermal power-plant system in the simulation: students identify and order components such as heat source, steam, turbine, generator, and load (for example, "heat ⟶ steam ⟶ turbine ⟶ generator ⟶ electricity ⟶ house"). Through repeated prompts like "what does this part represent?" and "from what energy to what energy does it change?", students are pushed to connect visual elements to underlying physics concepts, rather than merely naming objects. This interaction exemplifies decomposition: breaking the system into functional subsystems, identifying critical components such as the turbine and generator, and reassembling them into a coherent energy-flow model. Later, when clouds are moved across the simulated sky, students recognize that the cloud regulates light intensity and thus output energy, showing that they can treat "cloud–light–panel–output" as an interacting subsystem in the solar-energy model.

379	*T: (To Group 5) Finished? Is it done yet? What do you think about this?*
380	*S: It's simulated into this, simulated, Sir.*
381	*T: You want Activity Two, right? Activity One or Two?*
382	*S: One.*
383	*T: Where is it? Is the picture the same?*
384	*S: The same.*
385	*T: Okay. Go ahead. What has become? From what form of energy?*
386	*S: From kinetic energy.*
387	*T: Before that? Before this? You eat— "Feed me." So, what becomes?*
388	*S: Chemical, kinetic.*
389	*T: And then what becomes?*

390	*S: Electricity.*
391	*T: And finally?*
392	*S: Heat.*
393	*T: Good. That's the first one. The second one is this. Turbines and geothermal—geothermal energy. Geothermal is converted from what energy into what? The heat becomes.*
394	*S: Steam.*
395	*T: Then using a generator, for example when it's delivered to a house, it becomes what? It becomes a fan, right? This becomes System One. This is done by the first group, this one. Screenshot or take a photo. The second one is the edited one. The first one has steam, the second has turbines. Which one do you think is number two?*
396	*S: I think here, this picture. (points to the windmill)*

Individual Google Form responses from 38 students provide converging evidence that Activity 3 primarily developed decomposition and modelling–reflection. For the task on ordering energy changes in the bike–generator system, 22 students correctly decomposed the process into a logical sequence from chemical to kinetic to electrical to thermal energy, while 16 struggled with ordering or omitted stages. In building the geothermal power-plant (PLTP) model, 31 students successfully selected and arranged key components into a coherent system, whereas 7 showed errors in component choice or ordering. Matching the simulation-based PLTP model to a more complex real-world schematic proved challenging: 20 students did not complete this accurately, highlighting the representational jump needed for higher-level. In contrast, all 38 students correctly constructed the simpler solar power-plant (PLTS) model, indicating that their decomposition skills are robust for linear, low-complexity systems but less stable when realism and detail increase. On the reflective items, 15 students identified contextually appropriate factors affecting solar-cell output (for example, light intensity, angle, panel area, temperature), 11 showed clear procedural decomposition in describing simulation steps, 19 evaluated specific limitations of the simulation model relative to reality, and 23 articulated why understanding stepwise energy conversion is crucial for analyzing efficiency and losses.

Taken together, these findings show that Activity 3 implemented the Modelling–Reflection component of the CT-S framework. During this activity, students demonstrated computational thinking processes, particularly decomposition and modelling reasoning. Students progressed from recognizing individual components to decomposing complex systems into ordered subprocesses and, in many cases, reassembling them into coherent, causally plausible energy-flow models. At the same time, their difficulties with mapping between simulation and real-world schematics underscore the need for continued scaffolding when system complexity and abstraction demand increase. Overall, Activity 3 plays a pivotal role in the CT-S sequence by strengthening decomposition and modelling–reflection as foundational skills for later problem-solving tasks.

In activity, Transcript evidence illustrates how Data Collection–Reflection and Data Processing–Reflection were enacted alongside the development of decomposition and algorithmic thinking. In Indices 623–632, the teacher first led students to identify and name core elements of the energy system (source, cables, switch, load), and then to specify the function of each part. This interaction required students to decompose the circuit into functional components rather than treating it as a single undifferentiated object. In Indices 680–691, the focus shifted to a simple solar-energy setup (panel, LED, multimeter). Students were guided to distinguish positive and negative terminals, orient the LED correctly, and use the multimeter in a fixed sequence: select the measurement mode, connect probes to the correct terminals, check orientation, then read voltage. This explicit stepwise guidance fostered algorithmic thinking, since students had to follow and internalize a systematic procedure and understand that correct output depends on ordered, non-random actions. Later, students interpreted differences in LED brightness and voltage readings under varying light conditions. Here they began to relate physical variables (light intensity, panel position) to numerical outputs, engaging in reflective data processing by explaining why different groups obtained different values.

623	*T: Next, in the other box, there is this. This is what we call a solar panel.*
624	*S: Oh, this is a solar panel?*
625	*T: I've already attached the sensors. The actual solar panel is only this black part. Now, this device works to convert light into…*
626	*S: Electricity.*
627	*T: Okay. I'll bring it closer. This functions like a battery. Do you know what a battery is?*
628	*S: Yes, Sir.*

629	*T: How can we make this lamp turn on using a battery?*
630	*S: Plug it in.*
631	*T: Okay. The positive to positive, and the negative to…?*
632	*S: Negative.*
680	*T: Look, even without using the halogen lamp, it still lights up, right? That means a little sunlight is already being converted directly. We don't know how big the value is.*
681	*T: Now look here. Earlier this was already plugged into negative and positive, right?*
682	*S: Yes.*
683	*T: Where is the negative one? Plug this into the negative rail of the breadboard. Not to the LED leg.*
684	*T: Remove the negative leg of the LED. Connect it to the negative rail of the breadboard. Understand?*
685	*T: Now look forward. Do you find this? (shows a multimeter)*
686	*S: Yes.*
687	*T: This is for measuring electrical voltage. Batteries have voltage. How much do you usually make?*
688	*S: Nine volts.*
689	*T: Some are nine volts, some are 1.5. Now let's check the output voltage of the solar panel. There's a yellow wire—see it?*
690	*S: Yes.*
691	*T: The black one is the ground. Connect black to negative, yellow to positive.*

Google Form responses from 42 students triangulate these observations and show how Activity 4 supported the development of computational thinking. For variable identification, 24 students correctly distinguished independent, dependent, and control variables (for example, light intensity as independent, voltage or current as dependent, and the load as controlled), indicating successful decomposition of the experiment into functional elements, while 18 still confused or misassigned variables. For pattern recognition and causal reasoning, 28 students identified a consistent relationship between light intensity and input/output power, often expressing it as "the higher the intensity, the higher the input and output power", which reflects early algorithmic reasoning about rule-like covariation. The same number, 28 students, produced short conclusions that followed an if–then–therefore structure, showing that they could translate data patterns into concise, logically ordered statements. On efficiency-related items, 22 students decomposed the solar-panel system into multiple relevant factors (for example, light intensity, panel angle, temperature, cleanliness, shading), whereas others remained at a single-factor or vague level. Regarding procedural reconstruction, 21 students were able to describe the experiment as a coherent sequence of steps (assemble, adjust light, measure, record, conclude), and 25 students proposed plausible, ordered strategies to improve efficiency (for example, adjusting orientation, cleaning the panel, avoiding shade).

Overall, Activity 4 implemented Data Collection–Reflection and Data Processing–Reflection within the CT-S framework in a compressed experimental context by shifting some of the burden of raw data collection to teacher-provided datasets and reallocating student effort toward measurement, pattern identification, and interpretation. Conceptually, students consolidated their understanding of how solar-energy systems convert light into electrical output and which variables matter for that process. During this CT-S activity, students demonstrated computational thinking processes, particularly decomposition and algorithmic thinking, by following and reconstructing measurement procedures and formulating rule-like relationships and improvement strategies. The findings also indicate that while many students were able to express these processes explicitly, a substantial subset still required more explicit scaffolding to consistently reflect on variables, procedures, and system-improvement strategies in clear algorithmic terms.

In activity 5, transcript segments illustrate how Problem Solving-Reflective Use unfolded through decomposition, algorithmic thinking, and early debugging. In Indices 735-766 the teacher led students from problem formulation (*"how can a panel stay aligned with the moving Sun?"*) to representational abstraction, reframing light-dark differences into LDR resistance values that can be treated as numerical inputs. Students instructed to decompose the solar-tracking system into core components and began to reason in if–then terms: different light on each LDR should produce different servo movements. Later, using Tinkercad, the circuit was decomposed into repeated sensor subsystems (LDR + resistor + analog input), making the overall system manageable as a pattern of similar blocks. In Indices 889–910, students and teacher co-constructed an explicit comparison algorithm: read A0 and A1, compute the difference, apply a threshold, decide whether the servo should move and in which direction.

Questions like "what if the values are equal?" or "what does a positive difference mean?" pushed students to articulate conditional logic and small modifications to the rule, indicating more mature algorithmic reasoning. Final segments shifted to execution and hardware debugging: copying code, uploading to Arduino, checking power and ground rails, and correcting wiring when the servo failed to respond, reinforcing the idea that algorithms must be embodied consistently in both code and hardware.

735	*T: Now we are moving on to the problem in the project you will be working on. The project is this: if the sun moves… does the sun move?*
736	*S: It moves.*
737	*T: If you had a solar panel at home, which direction should it face?*
738	*S: South.*
739	*S: Face the sun.*
740	*T: Right, but the sun moves. So how can we keep it optimal?*
741	*S: By following the sun.*
742	*T: Following the sun, can we do that, Caro***?*
743	*S: Using sensors.*
744	*T: Correct, using sensors. Which sensor?*
746	*T: (repeating, emphasizing) Let's repeat the key idea: it must follow the sun. The sun can be detected not by heat, but by light. So, what sensor do we need?*
747	*S: Light.*
748	*T: Light. A light sensor, right? The simplest light sensor is called an LDR. Does anyone know what LDR stands for? LDR is this one here, on the tip. I've ordered two of them, called LDR sensors: Light Dependent Resistor. repeat*
749	*S: Light Dependent Resistor.*
889	*T: Later, connect it to pin 9. Done, right? Look at the picture. I've already connected it to pin 9, correct? Then here are the variables: the variable "compares" and the variable "angle". What is the function of "compare", try to explain?*
890	*S: To make them different.*
891	*T: What is being differentiated? Which values?*
892	*S: The sensor.*

893	*T: The sensors. Right, look—this one receives light. If I cover it, which one becomes brighter?*
894	*S: The end zone.*
895	*S: This one move, see.*
896	*T: So, I want to compare those values, the lightness and darkness of the sensors. If there is light here, which one is brighter? This one is brighter, right? And this one is darker. Logically, it always wants to turn toward which one?*
897	*S: The brighter one.*
898	*T: So that the values of both become what?*
899	*S: The same.*
900	*T: Correct. So, we evaluate this value. If the values are the same, what happens to the servo?*
901	*S: It doesn't move.*
902	*T: But if this one is larger, the servo moves either to the right or to the left. We will adjust this "compare" value. Look at the code, it's there, right? The top part only displays sensor values.*
903	*T: (explaining the blocks) "Compare = analog0 – analog1", for example. Analog 0 minus 1. This is 0, this is 1, roughly. So, if the value is 0, or equal, what happens?*
904	*S: The servo doesn't move.*
905	*T: But if the value is positive, which one is bigger, A0 or A1?*
906	*S: A0.*
907	*T: Let's assume this is A0, okay. When A0 is bigger, what happens to the angle? No—its angle changes, increases by 5 degrees. So, it changes by 5.*
908	*T: Suppose the positive direction is this way. Zero is like this, then 90, then 180. If originally, it's like this, and this side is brighter, then the compare result is positive. The angle increases by 5 degrees, so it moves.*
909	*T: If the compare value is still positive, it will continue until it finds a compare value equal to 0.*
910	*T: Because the values have a range, I don't set it to 0, but to 10. If the compare value is greater than 10, then it moves. If the value is less than 10, it does not move yet. Does that make sense?*

Google Form responses from 42 students triangulate these qualitative findings. For Question 1 (pseudocode), 24 of 42 students (about 57 percent) produced structurally correct algorithms that included reading two LDRs, looping, and using if–else rules to decide servo direction based on light differences, consistent with algorithmic thinking indicators (Dagiene et al., 2017; Selby & Woollard, 2013). The remaining 18 showed typical novice issues – inverted logic, duplicated conditions, omission of one sensor, or incomplete sequences. For Question 2, all 42 students described clear decomposition of the project into subtasks such as wiring LDRs, arranging the algorithm, testing the simulation, and checking system behavior, showing strong systemic decomposition at the group-work level. For Question 3 (debugging), 24 students identified valid sources of mismatch between expected and observed behavior, for example pin mapping errors, incorrect if–else structure, or loop problems, and suggested plausible corrections, while 18 gave non-algorithmic or overly generic answers, in line with reports that debugging is a particularly demanding CT skill (Grover & Pea, 2013). For Question 4 (LDR–servo mechanism), 25 of 43 recorded responses explained the mechanism in an explicitly algorithmic way – two readings, comparison, decision rule, and a “no movement” condition when values are equal – whereas 18 remained vague, omitted the comparison step, or referred only to one sensor.

Taken together, Activity 5 represents the peak of CT-S implementation in the sequence. Conceptually, students moved from understanding solar energy and sensing in earlier activities to specifying and enacting a concrete solution: a solar tracker that aligns a panel with maximum light. Within CT-S, the activity integrated problem formulation, representational abstraction of sensor data, and reflective use of computational tools as students interpreted how if–else rules and sensor differences drive servo motion (Hurt et al., 2023). Computationally, many students demonstrated advanced algorithmic thinking (through pseudocode and mechanism explanations), robust decomposition of both system structure and project workflow, and emerging debugging skills, although these capabilities were uneven across the cohort. The findings suggest that project-based work with microcontrollers can effectively foster higher-level CT, while also highlighting the need for extended scaffolding and longer test–debug–reflect cycles to consolidate algorithmic understanding for all learners.

Quantitative analysis used Rasch racking to examine shifts in item difficulty across CT indicators following CT-S–based instruction (Bond & Fox, 2015; Wright & Masters, 1982). All items showed negative shifts in logit values from pre- to posttest (Table 2), indicating that every item became easier relative to student ability. This pattern reflects a general improvement in CT proficiency consistent with Rasch principles (Boone et al., 2014).

Table 2. Item Measure for Pre and Post-test

Item	Pre-test	Post-test	Difference	Sign
S01A2	-0.03	-0.77	-0.74	-
S02G1	-0.16	-0.77	-0.61	-
S03D2	-0.05	-0.67	-0.62	-
S04E1	-0.07	-0.7	-0.63	-
S05E2	0.31	-0.18	-0.49	-
S06D1	0.29	0.19	-0.10	-
S07B3	0.24	-0.25	-0.49	-
S08A1	0.41	0.07	-0.34	-
S09G3	0.63	-0.36	-0.99	-
S10G1	0.43	-0.44	-0.87	-
S11B1	0.47	0.03	-0.44	-
S12B3	0.61	0.26	-0.35	-
S13E2	0.51	-0.01	-0.52	-

According to Table 2, at pretest, the most difficult item was S09G3 (0.63 logits), indicating substantial challenge in generalization indicator 3. The easiest item was S02G1 (–0.16 logits), suggesting that generalization indicator 1 was accessible to most students even before the intervention.

Posttest results revealed a more coherent hierarchy of difficulty, with S12B3 (0.26 logits) emerging as the hardest item and S02G1 (–0.77 logits) remaining the easiest. The shift in rank order from 02–04–03–01–07–06–05–08–10–11–13–12–09 (pretest) to 12–06–08–11–13–05–07–09–10–03–04–01–02 (posttest) indicates that gains were not uniform across CT indicators and that the relative complexity of skills changed as students progressed through instruction.

The largest improvement was observed for S09G3, with a 0.99-logit decrease, indicating major gains in advanced generalization. Two additional items—S10G1 and S01A2—showed substantial improvements ranging from –0.99 to –0.74 logits, highlighting strong development in pattern generalization and higher-order abstraction. These outcomes align with the nature of

CT-S activities, which required students to interpret energy-system datasets, construct causal models, and translate these into algorithmic rules—tasks known to stimulate generalization and advanced abstraction (Sengupta et al., 2013; Weintrop et al., 2016).

In contrast, three items showed relatively small logit changes: S06D1, S08A1, and S12B3, with differences ranging from –0.10 to –0.35 logits. Limited gains on basic decomposition and introductory abstraction suggest possible ceiling effects, where many students had already mastered these foundational skills at pretest. Meanwhile, the modest change for S12B3—an algorithmic indicator—supports the literature noting that procedural reasoning, rule formulation, and flow-control logic tend to develop more slowly and require extended cycles of guided practice (Lye & Koh, 2014; Selby & Woollard, 2013).

Taken together, the racking results demonstrate that the CT-S instructional sequence effectively supported the development of students' computational thinking abilities across all indicators, but with varying magnitudes aligned to the cognitive complexity of each indicator. The strongest gains emerged in advanced generalization and abstraction, which were heavily activated through iterative modelling and reasoning tasks. More modest gains in decomposition and algorithmic execution reflect skills that typically stabilize early (for decomposition) or require longer engagement for deeper mastery (for algorithmic reasoning), consistent with developmental patterns reported across CT research (Grover & Pea, 2013; Weintrop et al., 2016).

To examine changes in students' computational thinking (CT) proficiency at the person level, pretest and posttest responses were stacked onto a common Rasch item-difficulty scale. In this approach, both response sets are calibrated against an identical item metric so that shifts in person measures reflect true change rather than changes in item characteristics (Boone et al., 2014; Linacre, 2002). Interpretation of CT proficiency was then carried out using Rasch-based level classification following Boone's procedure.

Level boundaries were established using the 50% probability thresholds from the Item Structure File generated in Winstep (Boone et al., 2014, p. 351). For each item, the logit value at 0.50 response probability was extracted, interpreted as the minimum ability required to have an even chance of success. The mean of these 50% thresholds was then used to define stable cut points between adjacent levels. This process yielded three empirically grounded boundaries: –0.33 logits between Level 1 and Level 2, 0.10 logits between Level 2 and Level 3, and 0.44 logits between Level 3 and Level 4. These cut points were applied to person measures and visualised on the Wright Map (Figure 4.) to classify students into four CT proficiency levels.

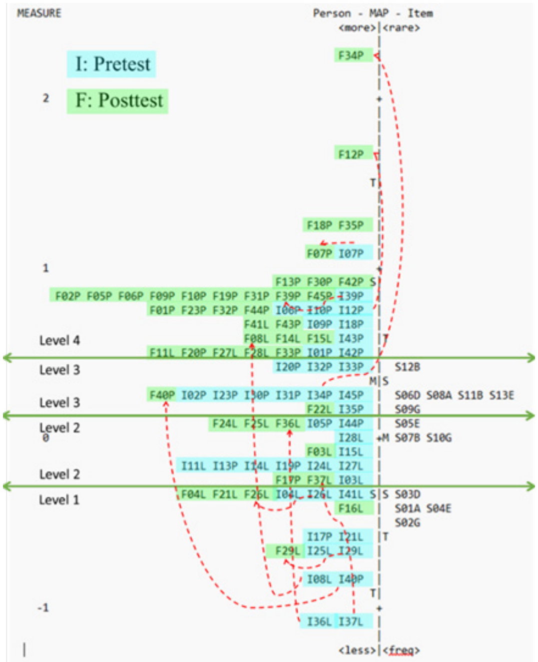

Figure 4. Person Wright Map

The Person Wright Map displays at Figure 4, pretest abilities in blue and posttest abilities in green. None of the students showed a decline in logit ability, consistent with Rasch expectations that upward movement reflects genuine growth rather than a measurement artefact (Boone et al., 2014). Two students (29L, 26L) remained in Level 1, indicating that the intervention did not sufficiently shift their ability beyond the –0.33 logit threshold. Conversely, two students (39P, 07P) consistently remained in Level 4, demonstrating stable high-level CT proficiency. Such stability at the upper range is typical in Rasch analysis and often reflects a ceiling effect caused by items that are not sufficiently challenging to discriminate among high-ability learners (Bond & Fox, 2015).

Beyond these stable cases, several students demonstrated substantial ability gains. Student 08L progressed from Level 1 to Level 4, while 40P moved from Level 1 to Level 3, indicating accelerated CT development within a relatively short intervention. Among the lowest initial performers, students 36L and 37L improved from Level 1 to Level 2, reflecting meaningful true growth on the logit scale (Linacre, 2002). Higher-ability students also showed movement, with learners such as 34P shifting from Level 3 to Level 4, while others like 12P remained at Level 4, suggesting that item difficulty was insufficient to differentiate among the highest performers (Wright & Masters, 1982).

Overall person statistics further support the pattern of improvement. The mean person measure rose from 0.04 logits (pretest) to 0.56 logits (posttest), a gain of 0.52 logits. The minimum ability increased from –1.08 to –0.66 logits (Δ = 0.42), showing growth even among the weakest learners. Meanwhile, the maximum ability increased from 1.08 to 2.23 logits (Δ = 1.15), indicating that advanced students also benefited substantially from the CT-S intervention. These shifts are consistent with Rasch theory, where increases on the linear logit scale represent true growth independent of raw score distributions (Bond & Fox, 2015; Boone et al., 2014).

Changes in level distributions further illustrate the strength of the intervention. At pretest, students were relatively evenly distributed across the four CT levels (9 in Level 1; 13 in Level 2; 10 in Level 3; 10 in Level 4). Following instruction, only one student remained at Level 1, nine were in Level 2, and the number of students in Level 4 rose sharply from 10 to 31. Nearly all Level 1 students progressed at least one level, and most of them reached the highest CT level. This distribution suggests considerable strengthening of higher-order CT components including generalization, evaluation, and abstraction, in alignment with established CT frameworks (Hurt et al., 2023; Selby & Woollard, 2013).

Group-level patterns complement these findings. Groups with heterogeneous initial abilities tended to show the largest gains, supporting Vygotskian interpretations of the Zone of Proximal Development (Vygotskij & Cole, 1981), where peer interaction facilitates upward movement through shared problem solving and scaffolded reasoning. In this study, such collaborative advantages emerged during tasks involving data interpretation, pseudocode development, and automated system design.

Viewed through the lens of the CT-S framework (Hurt et al., 2023), the stacking and Wright Map analyses suggest that learning activities centered on data collection, data processing, sensor-based experimentation, and pseudocode-driven automation effectively supported students' Reflective Use of Computational Tools. Students who progressed from Levels 1–2 to Levels 3–4 not only improved their success probabilities on CT items but also demonstrated evidence of higher-order CT processes such as pattern generalization, data-based evaluation, and rule-based algorithm construction (Dagiene et al., 2017; Selby & Woollard, 2013)

To clarify the relationship between qualitative and quantitative findings, Table 3 summarizes how each CT-S activity supported the development of specific computational thinking processes observed in the verbal protocol analysis and how these processes relate to improvements identified in the Rasch results.

As shown in Table 3, the sequence of CT-S activities corresponds to the progression of computational thinking processes observed in the study. Early activities emphasized abstraction and pattern recognition during data interpretation, while later activities promoted decomposition, algorithmic reasoning, and system design during modelling, experimentation, and engineering tasks. These qualitative patterns align with the Rasch analysis, which shows the largest improvements in higher-order CT indicators such as generalization and abstraction.

This study extends previous CT research in several ways. Earlier studies have often integrated computational thinking through programming activities, robotics, or isolated physics modules. In contrast, the present study applies the CT-S framework within a renewable energy learning sequence that integrates data interpretation, system modelling, experimentation, and engineering design. The renewable energy context requires students to reason about complex systems and real-world sustainability challenges, thereby providing a meaningful context for applying computational thinking beyond traditional programming-focused activities.

The findings also have implications for broader systemic challenges in implementing computational thinking in Indonesian education. Previous studies have highlighted limited teacher readiness and inconsistent integration of computational thinking in classroom practice (Budiyanto et al., 2025; Irvani et al., 2024). The CT-S instructional sequence used in this study demonstrates how computational thinking can be embedded within existing science topics such as renewable energy through structured activities involving data analysis, modelling, experimentation, and engineering design. This suggests that CT integration does not necessarily require extensive programming instruction, but can be implemented through carefully designed science learning activities. The CT-S framework may offer a practical pathway for supporting teachers in integrating computational thinking into science classrooms within existing curricular structures.

Table 3. Integration of CT-S Activities, Qualitative CT Processes, and Quantitative Outcomes

CT-S Activity	Learning Focus	Dominant CT Processes (VPA Evidence)	Representative Classroom Evidence	Quantitative Evidence (Rasch Results)
Activity 1 – Data Collecting	Understanding fossil-energy problems	Abstraction	Students identified key energy categories and summarized trends in energy graphs	Improvement in abstraction items (S01A2, S08A1)
Activity 2 – Data Processing	Interpreting renewable-energy datasets	Generalization, Evaluation	Students identified patterns between potential and realization of renewable energy sources	Largest Rasch gains in generalization indicators (S09G3, S10G1)
Activity 3 – Modelling	Simulation of energy systems	Decomposition	Students broke down geothermal and solar energy systems into functional components	Moderate gains in decomposition items (S03D2, S06D1)
Activity 4 – Experimentation	Solar panel measurements	Algorithmic reasoning	Students followed structured measurement procedures and identified variable relationships	Improvement in algorithmic indicators (S07B3, S11B1)
Activity 5 – Problem Solving	Solar tracking design	Algorithmic thinking, Debugging	Students constructed pseudocode and debugged sensor–servo interactions	Increase in higher CT ability levels (Level 4 growth)

Several limitations should be acknowledged. The quantitative sample size (N = 45) represents a single class from one school, which may limit the generalizability of the findings to broader student populations. In addition, the study employed a one-group pretest–posttest design without a control group, making it difficult to fully rule out alternative explanations for the observed improvements, such as maturation or testing effects. Therefore, the results should be interpreted as evidence of improvement within the instructional context rather than as definitive causal proof of the intervention's effectiveness.

The instructional implementation also faced practical constraints. Due to limited classroom time and the technical complexity of the solar-tracking project, not all student groups were able to complete a fully functional physical prototype, with some groups reaching only the simulation stage. As a result, part of the analysis focused on students' design reasoning, algorithmic development, and simulation outcomes rather than fully implemented hardware systems.

However, the purpose of the Rasch analysis in this study was primarily to examine changes in student ability and item functioning within the instructional context rather than to generalize results statistically to a larger population. To strengthen the interpretation of the findings, the study employed an embedded mixed-method design in which qualitative evidence from verbal protocol analysis provided detailed insights into students' cognitive processes during CT-S activities. This qualitative evidence complements the quantitative results by illustrating how computational thinking processes developed throughout the learning sequence. Future studies employing larger samples, multi-school settings, and controlled or quasi-experimental designs would help strengthen the generalizability and causal interpretation of these findings.

CONCLUSION

The findings demonstrate that implementing the Computational Thinking for Science (CT-S) instructional framework across a sequence of Data Collecting, Data Processing, Modelling, and Problem-Solving activities effectively strengthened students' computational thinking skills, both in terms of cognitive processes and measurable learning gains. Qualitative VPA analysis showed that each activity elicited distinct CT components: abstraction emerged strongly in Activities 1–2, decomposition and algorithmic reasoning developed in Activities 3–4, and advanced problem-solving and algorithmic rule construction became prominent in Activity 5, with reflective use consistently observed as students interpreted data, constructed if–else rules, and connected computational representations to physical phenomena. Quantitatively, students' abilities improved substantially: mean item difficulty decreased (from 0.25 to –0.33 logits), indicating that CT indicators became easier after instruction; mean person ability increased from 0.04 to 0.56 logits ($\Delta = 0.52$); minimum ability rose from –1.08 to –0.66, and maximum ability from 1.08 to 2.23, reflecting growth among both low- and high-performing students. Level distributions also shifted markedly, with Level 4 students increasing from 10 to 31, while nearly all Level 1 students moved upward. Taken together, the qualitative and quantitative results show that CT-S–based instruction improved students' conceptual and procedural computational thinking skills. The findings also highlight renewable energy learning as an effective interdisciplinary context for applying the CT-S framework and developing computational thinking through authentic energy-system problems and technology-based design tasks.

ACKNOWLEDGMENT

The authors gratefully acknowledge the Lembaga Pengelola Dana Pendidikan (LPDP) for financial support. Sincere thanks are extended to the supervising lecturer for guidance throughout the study, as well as to the lecturers, teachers, and students involved in the research. Appreciation is also given to SEAMEO QITEP in Science for support in completing this study.

REFERENCES

Aksit, O., & Wiebe, E. N. (2020). Exploring Force and Motion Concepts in Middle Grades Using Computational Modeling: A Classroom Intervention Study. *Journal of Science Education and Technology,* 29(1), 65–82. https://doi.org/10.1007/s10956-019-09800-z

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., & Zagami, J. (2016). *A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge*. 19(3), 47–57.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is Involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54. https://doi.org/10.1145/1929887.1929905

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Bond, T. G., & Fox, C. M. (2015). *Applying the Rasch model: Fundamental measurement in the human sciences (Third edition)*. Routledge, Taylor and Francis Group.

Boone, W. J., Staver, J. R., & Yale, M. S. (2014). Rasch Analysis in the Human Sciences. *Springer Netherlands*. https://doi.org/10.1007/978-94-007-6857-4

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking.

Budiyanto, C. W., Wahyudi, M. N. A., Widiastuti, I., & Latifah, R. (2025). Teaching Computational Thinking in Indonesia: Trend in Conception, Methods, Technology, and Evaluation. *Educational Process: International Journal*, 17(e2025308). https://doi.org/10.22521/edupij.2025.17.308

Cannady, M. A., Collins, M. A., Hurt, T., Montgomery, R., Greenwald, E., & Dorph, R. (2025). Computational Thinking for Science Positions Youth to Be Better Science Learners. Education Sciences, 15(1), 105. https://doi.org/10.3390/educsci15010105

Cansu, F. K., & Cansu, S. K. (2019). An Overview of Computational Thinking. *International Journal of Computer Science Education in Schools*, 3(1), 17–30. https://doi.org/10.21585/ijcses.v3i1.53

Creswell, J. W. (2012). Qualitative Inquiry and Research Design: Choosing Among Five Approaches. *SAGE Publications*. https://books.google.co.id/books?id=jMfVyU8ida4C

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking—A guide for teachers. *Computing At School*. http://eprints.soton.ac.uk/id/eprint/424545

Dagiene, V., Sentance, S., & Stupurien, G. (2017). Developing a Two-Dimensional Categorization System for Educational Tasks in Informatics. *Informatica*, 28(1), 23–44. https://doi.org/10.15388/Informatica.2017.119

Docktor, J. L., & Mestre, J. P. (2014). Synthesis of discipline-based education research in physics. *Physical Review Special Topics - Physics Education Research*, 10(2), 020119. https://doi.org/10.1103/PhysRevSTPER.10.020119

Ericsson, K. A., & Simon, H. A. (1993). *Protocol Analysis: Verbal Reports as Data (2nd ed.).* The MIT Press. https://doi.org/10.7551/mitpress/5657.001.0001

Ferrarelli, P., & Iocchi, L. (2021). Learning Newtonian Physics through Programming Robot Experiments. *Technology, Knowledge and Learning,* 26(4), 789–824. https://doi.org/10.1007/s10758-021-09508-3

Furber, S. (2012). *Shut down or restart? The way forward for computing in UK schools (p. 29)*. The Royal Society.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43. https://doi.org/10.3102/0013189X12463051

Hemmendinger, D. (2010). A plea for modesty. *ACM Inroads*, 1(2), 4–7. https://doi.org/10.1145/1805724.1805725

Hmelo-Silver, C. E., Duncan, R. G., & Chinn, C. A. (2007). Scaffolding and Achievement in Problem-Based and Inquiry Learning: A Response to Kirschner, Sweller, and Clark (2006). *Educational Psychologist*, 42(2), 99–107. https://doi.org/10.1080/00461520701263368

Hurt, T., Greenwald, E., Allan, S., Cannady, M. A., Krakowski, A., Brodsky, L., Collins, M. A., Montgomery, R., & Dorph, R. (2023). The computational thinking for science (CT-S) framework: Operationalizing CT-S for K–12 science education researchers and educators. *International Journal of STEM Education*, 10(1), 1. https://doi.org/10.1186/s40594-022-00391-7

Hutchins, N. M., Biswas, G., Maróti, M., Lédeczi, Á., Grover, S., Wolf, R., Blair, K. P., Chin, D., Conlin, L., Basu, S., & McElhaney, K. (2020). C2STEM: A System for Synergistic Learning of Physics and Computational Thinking. *Journal of Science Education and Technology*, 29(1), 83–100. https://doi.org/10.1007/s10956-019-09804-9

Irvani, A. I., Rochintaniawati, D., Riandi, R., & Sinaga, P. (2024). Analyzing the Integration of Computational Thinking in Science and Physics Education within the Indonesian Curriculum. *Kasuari: Physics Education Journal (KPEJ)*, 7(1), 182–194. https://doi.org/10.37891/kpej.v7i1.620

Khasyyatillah, I., & Osman, K. (2019). The Development of CT-S Learning Module on The Linear Motion Topic to Promote Computational Thinking Thinking. *Journal of Educational Sciences,* 3(3), 270. https://doi.org/10.31258/jes.3.3.p.270-280

Lapawi, N., Husnin, H., & The National University of Malaysia (UKM). (2020). The Effect of Computational Thinking Module on Achievement in Scienceonal Thinking Modules on Achievement in Science. *Science Education International,* 31(2), 164–171. https://doi.org/10.33828/sei.v31.i2.5

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1), 32–37. https://doi.org/10.1145/1929887.1929902

Lehrer, R., & Schauble, L. (2015). *The Development of Scientific Thinking. In R. M. Lerner (Ed.), Handbook of Child Psychology and Developmental Science (1st ed., pp. 1–44).* Wiley. https://doi.org/10.1002/9781118963418.childpsy216

Linacre, J. M. (2002). *What do Infit and Outfit, Mean-square and Standardized mean?* https://www.rasch.org/rmt/rmt162f.htm

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61. https://doi.org/10.1016/j.chb.2014.09.012

NGSS Lead States. (2013). *Next Generation Science Standards: For States, By States (p. 18290).* National Academies Press. https://doi.org/10.17226/18290

Selby, C. C., & Woollard, J. (2013). *Computational Thinking: The Developing Definition*.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies,* 18(2), 351–380. https://doi.org/10.1007/s10639-012-9240-x

Silva, E. (2009). Measuring Skills for 21st-Century Learning. *Phi Delta Kappan*, 90(9), 630–634. https://doi.org/10.1177/003172170909000905

Sondakh, D. E., Pungus, S. R., & Putra, E. Y. (2022). Indonesian Undergraduate Students' Perception of Their Computational Thinking Ability. *CogITo Smart Journal*, 8(1), 68–80. https://doi.org/10.31154/cogito.v8i1.387.68-80

Sugiyono. (2013). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D. In Metode penelitian pendidikan: (Pendekatan kuantitatif, kualitatif dan R & D).* Alfabeta. https://books.google.co.id/books?id=0xmCnQAACAAJ

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 103798. https://doi.org/10.1016/j.compedu.2019.103798

Taub, R., Armoni, M., Bagno, E., & Ben-Ari, M. (Moti). (2015). The effect of computer science on physics learning in a computational science environment. *Computers & Education*, 87, 10–23. https://doi.org/10.1016/j.compedu.2015.03.013

Tekdal, M. (2021). Trends and development in research on computational thinking. *Education and Information Technologies*, 26(5), 6499–6529. https://doi.org/10.1007/s10639-021-10617-w

Trilling, B. (with Fadel, C. & Partnership for 21st Century Skills). (2009). *21st century skills: Learning for life in our times (1st ed)*. Jossey-Bass.

Vygotskij, L. S., & Cole, M. (1981). *Mind in society: The development of higher psychological processes (Nachdr.).* Harvard Univ. Press.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). 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*. Communications of the ACM, 49(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2011). *Research Notebook: Computational Thinking—What and Why?* The Link Magazine, (6), 20–23.

Wright, B. D., & Masters, G. N. (1982). *Rating scale analysis: Rasch measurement*. Mesa Pr.

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), 1–16. https://doi.org/10.1145/2576872

Yin, Y., Hadad, R., Tang, X., & Lin, Q. (2020). Improving and Assessing Computational Thinking in Maker Activities: The Integration with Physics and Engineering Learning. *Journal of Science Education and Technology*, 29(2), 189–214. https://doi.org/10.1007/s10956-019-09794-8.