**_Prisma Sains: Jurnal Pengkajian Ilmu dan Pembelajaran Matematika dan IPA IKIP Mataram_**
_http://ojs.ikipmataram.ac.id/index.php/prismasains/index_
_e-mail: prismasains.pkpsm@gmail.com_

_July 2026, Vol. 14, No. 3_
_p-ISSN: 2338-4530_
_e-ISSN: 2540-7899_
_pp. 1300-1317_

# Computational Thinking in Linear Programming Based on Prior Mathematical Ability

***Reza Kusuma Setyansah, Swasti Maharani**
Mathematics Education Department, Universitas PGRI Madiun, Kota Madiun, Indonesia
*Corresponding Author e-mail: reza.mathedu@unipma.ac.id

Received: March 2026; Revised: April 2026; Published: July 2026

**Abstract**

Computational thinking (CT) plays an important role in supporting students' mathematical problem-solving abilities, particularly in technology-assisted learning environments. This study aimed to analyze students' computational thinking profiles in solving linear programming problems using the simplex method based on their prior mathematical ability and to examine their responses to the use of POM-QM for Windows as a learning support tool. A qualitative descriptive approach was employed involving 21 undergraduate students in a mathematics education program. Three students representing high, moderate, and low levels of prior mathematical ability were purposively selected for in-depth analysis. Data were collected through linear programming problem-solving tasks assisted by POM-QM for Windows, semi-structured interviews, and student response questionnaires. Data analysis was conducted through data reduction, data display, and conclusion drawing based on four computational thinking components: decomposition, pattern recognition, abstraction, and algorithmic thinking. The findings showed that students' computational thinking profiles differed according to their prior mathematical ability. Students with high prior mathematical ability demonstrated all CT components consistently through systematic problem decomposition, accurate mathematical modeling, effective pattern identification, and appropriate application of simplex procedures. Students with moderate prior mathematical ability showed adequate performance but experienced difficulties in pattern recognition and algorithmic implementation. Meanwhile, students with low prior mathematical ability encountered challenges in identifying relevant information, constructing mathematical representations, and developing structured solution procedures. Students generally responded positively to the use of POM-QM for Windows, particularly in simplifying calculations and supporting the problem-solving process. These findings indicate that prior mathematical ability influences students' computational thinking performance and highlight the importance of integrating technological tools with mathematical reasoning in learning activities.

**Keywords:** Computational thinking; Linear programming; Prior mathematical ability; POM-QM for windows; Simplex method

**How to Cite:** Setyansah, R. K., & Maharani, S. (2026). Computational Thinking in Linear Programming Based on Prior Mathematical Ability. _Prisma Sains: Jurnal Pengkajian Ilmu Dan Pembelajaran Matematika Dan IPA IKIP Mataram_, _14_(3), 1300–1317. https://doi.org/10.33394/j-ps.v14i3.19950

https://doi.org/10.33394/j-ps.v14i3.19950



## INTRODUCTION

The rapid transformation of education in the era of the Industrial Revolution 4.0 has fundamentally reshaped the goals and practices of teaching and learning. Education today is no longer limited to strengthening conceptual understanding but also emphasizes the development of twenty-first century competencies such as critical thinking, problem-solving, digital literacy, and computational reasoning (Siregar et al., 2024). Consequently, various pedagogical innovations have been introduced, including blended learning, flipped classrooms, problem-based learning (PBL), and the integration of STEM–TPACK approaches. These innovations have been shown to improve learning outcomes while supporting the development of higher-order cognitive skills required in modern education (Kuwat et al., 2025). In this context, digital learning is not merely a technological adaptation, but a strategic pedagogical transformation aimed at improving the quality of education.

Despite these developments, many educational challenges remain rooted in pedagogical practices and the limited effectiveness of instructional implementation. Previous studies indicate that innovative learning approaches such as blended learning can reduce students' misconceptions and improve conceptual understanding (Kesuma et al., 2020). Furthermore, student engagement plays a significant role in improving learning outcomes, highlighting the importance of collaborative and participatory learning environments (Umar & Ko, 2022). Effective learning therefore requires instructional designs that position students as active participants in the learning process (Fauziyah & Pujiastuti, 2020). However, the implementation of such pedagogical innovations often faces practical barriers, including large class sizes, limited technological infrastructure, insufficient teacher training, and weak institutional support (Aksoy, 2025). As a result, many instructional practices remain suboptimal.

One important competency that has gained increasing attention in mathematics education is computational thinking (CT). Computational thinking enables learners to solve complex problems systematically by applying processes such as decomposition, pattern recognition, abstraction, and algorithmic thinking (Irawati & Hadi, 2025; Maharani et al., 2023). These processes support the development of logical, analytical, and structured reasoning abilities that are essential for mathematical problem solving. Empirical studies have shown that the integration of CT in mathematics learning can enhance students' critical and creative thinking skills while improving their ability to solve complex problems (Kaswar & Nurjannah, 2024). Moreover, appropriate instructional strategies can facilitate the development of CT as part of higher-order thinking skills (HOTS) (Syari & Meiliasari, 2024). The four CT indicators - decomposition, pattern recognition, abstraction, and algorithmic thinking have been widely recognized as core dimensions for examining students' computational reasoning and problem-solving processes in mathematics education (Darmawan et al., 2024). In technology-supported mathematical environments, students are expected not only to understand mathematical concepts but also to formulate solutions that can be executed and evaluated through computational tools (Borkulo & Utrecht, 2020).

Nevertheless, the integration of computational thinking into mathematics instruction remains a significant challenge. Studies indicate that CT has not yet been optimally implemented in classroom practice, leading to variations in students' mastery of computational thinking skills (Maharani et al., 2019). In addition, students' CT performance is influenced by several internal factors, including self-efficacy and prior knowledge (Istofany et al., 2025). This suggests that students' prior mathematical ability may play an important role in determining the quality of computational thinking during problem solving. However, the relationship between prior mathematical ability and computational thinking has not been sufficiently explored in specific mathematical contexts. Therefore, investigating CT through the lens of students' prior mathematical ability may provide a deeper understanding of how computational reasoning develops during mathematical problem solving.

One mathematical topic that is particularly relevant for examining computational thinking is linear programming. Linear programming requires students to analyze contextual problems, construct mathematical models, and apply systematic procedures to determine optimal solutions. Previous research has reported that students demonstrate varying levels of success when solving linear programming problems using computational thinking approaches (Nuraisa et al., 2019). However, many students still experience difficulties when translating contextual problems into mathematical models, which indicates weaknesses in conceptual understanding and procedural reasoning (Suratih & Pujiastuti, 2020). Errors frequently occur during the problem-solving process, especially in identifying constraints, formulating objective functions, and performing algorithmic procedures (Fauziyah & Pujiastuti, 2020). The use of optimization software provides a meaningful context for examining plugged computational thinking in linear programming. Among the available tools, POM-QM for Windows enables students to model optimization problems, input constraints, execute simplex procedures, and interpret computational outputs. These activities require students to engage in decomposition, pattern recognition, abstraction, and algorithmic thinking while interacting with technology. Therefore, solving linear programming problems using POM-QM for Windows offers an appropriate setting for investigating how plugged computational thinking is manifested across different levels of prior mathematical ability.

Despite the growing body of literature on computational thinking in mathematics education, research focusing specifically on plugged computational thinking remains limited. Existing studies predominantly investigate CT in general mathematical problem-solving contexts or programming-based activities, with little attention given to how students demonstrate computational thinking while interacting with technological tools during mathematical problem solving. Although several studies have examined the integration of technology in mathematics learning to enhance students' mathematical (Suprapto et al., 2021), limited empirical evidence explains how different levels of prior mathematical ability shape students' plugged CT processes. This gap is particularly evident in linear programming, where students are required to integrate mathematical reasoning, algorithmic procedures, and technological tools to obtain optimal solutions. Consequently, the relationship among prior mathematical ability, plugged computational thinking, and the use of optimization software remains insufficiently understood.

Therefore, this study aims to analyze students' plugged computational thinking (Plugged CT) profiles in solving linear programming problems using the simplex method based on their prior mathematical ability levels and to explore students' responses to the use of POM-QM for Windows as a learning support tool.

Based on these objectives, the research questions are formulated as follows:

1) How do students' plugged computational thinking profiles differ based on their prior mathematical ability levels?
2) How do students respond to the use of POM-QM for Windows in solving linear programming problems?

This study contributes to the literature in three ways. First, it extends computational thinking research by examining plugged computational thinking within the specific context of linear programming supported by optimization software. Second, it provides a qualitative account of how prior mathematical ability influences the manifestation of decomposition, pattern recognition, abstraction, and algorithmic thinking during technology-supported problem solving. Third, it offers empirical evidence regarding the educational potential of POM-QM for Windows as a tool for facilitating students' computational reasoning and mathematical problem-solving processes.

## METHOD

This study employed a qualitative approach on the grounds that the nature of the problem under investigation required an in-depth understanding of students' cognitive processes within their natural context. In qualitative research, the researcher functions as the primary instrument in collecting and interpreting data, thereby enabling a comprehensive and contextual exploration of the phenomenon under study (Fadli, 2021). Consistent with previous findings (Juwantara, 2019), qualitative inquiry aims to generate rich and detailed descriptions of a particular phenomenon—in this case, students' computational thinking skills. The research design was descriptive in nature, focusing on identifying and elaborating the characteristics and indicators of computational thinking that emerged during the problem-solving process.

The study was conducted in the Mathematics Education Study Program at Universitas PGRI Madiun, involving seventh-semester students in the 2025/2026 academic year. A total of twenty-one students participated in the study, from whom three were purposively selected to represent varying levels of mathematical problem-solving ability. Data were collected through problem-solving tasks and in-depth interviews. The primary instrument consisted of a single problem-solving task in the Linear Programming course, specifically on the simplex method, supported by POM-QM for Windows. The task was designed to elicit indicators of computational thinking. Data analysis was carried out systematically through the stages of data collection, data reduction, data display, and conclusion drawing and verification.

The grouping of students' prior ability levels was based on the criterion-referenced assessment standards commonly used at Universitas PGRI Madiun (UNIPMA) and referred to the theoretical score distribution. The justification for using these ranges was based on UNIPMA's Academic Regulations regarding passing grades and competency mastery. The determination of category boundaries was established as follows.

**Table 1.** Score Range Indicators

| Category | Description |
|---|---|
| High | – Scores of 80 and above, equivalent to A or A- (very good). |
| Medium | – Scores ranging from 60 to 79, within the range of B to C (sufficient/moderate). |
| Low | – Scores below 60 require intensive remediation as they have not yet achieved the minimum mastery standard expected for higher-order thinking skills (HOTS). |

Prior to analyzing students' computational thinking profiles, the prior mathematical ability of the 21 research subjects was mapped. This mapping was based on the average scores from previous linear programming coursework and the pre-test scores administered. A summary of the distribution of students' prior ability scores is presented below.

From this grouping, three research subjects were selected—each representing one category—for an in-depth analysis of their computational thinking profiles.

**Table 2.** Distribution of Students' Prior Ability Levels

| Category | Score Range | Frequency |
|---|---|---|
| High | ≥ 80 | 5 |
| Medium | 60–79 | 12 |
| Low | < 60 | 4 |

Three subjects were selected for in-depth analysis. Subject 1 (S-1) represented the high-ability category, with a course grade of 86 and a pre-test score of 85. Subject 2 (S-2) represented the medium-ability category, with a course grade of 73 and a pre-test score of 70. Subject 3 (S-3) represented the low-ability category, with a course grade of 55 and a pre-test score of 52.

The research procedure comprised three stages. First, participants were assigned a linear programming problem related to the simplex method using POM-QM for Windows. They were required to complete their solutions in written form using Microsoft Word and submit them in PDF format through the university’s LMS (eLMA). Second, students’ written responses were analyzed based on predetermined computational thinking indicators. The indicators applied in this study included: (1) decomposition, defined as the ability to break down complex problems into simpler components; (2) pattern recognition, referring to the ability to identify and classify patterns or characteristics within a problem; (3) abstraction, defined as the ability to select essential information and represent the problem in a simplified model; and (4) algorithmic thinking, referring to the ability to design systematic solution steps leading to accurate results and conclusions. A detailed description of the computational thinking indicators used in this study is presented in Table 3, Table 4, and Table 5.

**Table 3.** Computational Thinking Indicators

| No. | Indicators | Description |
|---|---|---|
| 1 | Decomposition | – Breaking down a problem into smaller components and representing them in the form of a mathematical model. |
| 2 | Pattern recognition | – Classifying problems according to their respective types or characteristics. |
| 3 | Abstraction | – Deciding whether to use or reject information by identifying which information is important and which is not. |
| 4 | Algorithmic thinking | – At the final stage, designing systematic solution procedures using POM-QM for Windows and formulating appropriate conclusions. |

**Table 4.** Indicators, Cognitive Levels, and Question Descriptions

| No | Indicator | Cognitive Level | Question Description |
|---|---|---|---|
| 1 | Decomposition | C3 | – Decompose the "Mentari Bersinar" case into linear programming components (decision variables, objective function, constraints) and explain how each is input into POM-QM for Windows. |
| 2 | Pattern Recognition | C4 | – Identify whether the case is a maximization or minimization problem, explain the pattern of constraints (minimum percentage and maximum limits), and describe how these constraints are represented in POM-QM for Windows. |
| 3 | Abstraction | C4 | – Simplify the three-variable model (materials A, B, C) into two variables using the total production of 2000 pounds. Explain which information is retained and omitted and how this affects the graphical analysis in POM-QM for Windows. |
| 4 | Algorithmic Thinking | C5 | – Design systematic steps to determine the minimum cost using POM-QM for Windows, from variable input, objective function, and constraints to obtaining optimization results. Explain how to draw conclusions from the output. |

**Table 5.** Rubric for Computational Thinking Assessment

| Indicator | Level 0 | Level 1 | Level 2 | Level 3 |
|---|---|---|---|---|
| Decomposition | Cannot identify problem components. | Identifies partial components (e.g., variables only), incomplete. | Identifies variables, objective function, and some constraints with reasonable accuracy. | Identifies all components (variables, objective function, constraints) completely, accurately, ready for POM-QM modelling. |
| Pattern Recognition | Cannot recognize problem patterns. | Limited pattern recognition, inaccurate (e.g., wrong optimization type/constraints). | Recognizes problem patterns (minimization, constraints) but inconsistent. | Accurately identifies problem patterns (cost minimization, percentage & maximum constraints) and relates them to POM-QM constraint forms. |
| Abstraction | Cannot simplify the model. | Simplifies model irrelevantly or with major errors. | Able to simplify model (e.g., variable reduction) but suboptimal. | Simplifies model appropriately, selects essential information, explains impact on analysis (graphical/ POM-QM). |
| Algorithmic Thinking | Cannot formulate solution steps. | Formulates unsystematic steps, not following procedures. | Formulates fairly systematic steps but incomplete (e.g., input only, no interpretation). | Formulates complete, systematic steps (input–process–output in POM-QM) and draws conclusions from optimization results. |

In the third stage, data triangulation was conducted through semi-structured interviews to validate and further deliberate the findings derived from the written analysis. These interviews were specifically designed to probe the participants' mental processes, aligning with (Wing, 2006) conceptualization of computational thinking as a fundamental abstraction and algorithmic reasoning that transcends mere procedural execution. By employing a semi-structured guideline, the researcher explored how students navigate the transition from problem decomposition to the formulation of generalized mathematical models. Furthermore, the interview protocol was adapted from (Weintrop et al., 2016) taxonomy, specifically targeting computational problem-solving practices. This allowed for an in-depth exploration of how students identify modularity within linear programming constraints and evaluate the efficiency of their chosen Simplex iterations. Through this rigorous procedure, the analysis captured the underlying cognitive architecture of the students' reasoning, thereby systematically strengthening the validity, credibility, and theoretical depth of the research findings.

## RESULTS AND DISCUSSION

The research instrument, consisting of linear programming questions, was developed to assess students' Computational Thinking (CT) achievement. The format of the questions administered is presented below.

*Sebuah perusahaan penyedia sarapan, "Mentari Bersinar", diharuskan memproduksi sebanyak **1 ton (2000 pon)** sereal manis setiap hari untuk memenuhi permintaan pasar. Dalam proses produksinya, perusahaan menggunakan tiga jenis bahan baku, yaitu bahan A, bahan B, dan bahan C, dengan rincian biaya sebagai berikut: bahan A sebesar **Rp.4.000.000,- per pon**, bahan B sebesar **Rp.3.000.000,- per pon**, dan bahan C sebesar **Rp.2.000.000,- per pon**.*

*Berdasarkan regulasi pemerintah, komposisi sereal yang diproduksi harus memenuhi ketentuan berikut:*

1. *Mengandung **sekurang-kurangnya 10% bahan A** dari total produksi,*
2. *Mengandung **sekurang-kurangnya 20% bahan B** dari total produksi,*
3. *Penggunaan bahan C **tidak boleh melebihi 800 pon per ton produksi**, guna menjaga kualitas rasa produk.*

*Berdasarkan permasalahan tersebut, mahasiswa diminta untuk:*

1. ***Menyusun model matematis program linier** yang meliputi:*
   - *Penentuan variabel keputusan,*
   - *Perumusan fungsi tujuan untuk meminimalkan biaya produksi,*
   - *Penentuan fungsi kendala sesuai dengan ketentuan yang diberikan.*
2. ***Melakukan pengujian terhadap fungsi kendala** yang telah disusun untuk memastikan bahwa model memenuhi seluruh batasan yang ditetapkan.*
3. ***Menentukan solusi optimal** berupa kombinasi penggunaan bahan baku yang menghasilkan **biaya minimum**, baik melalui pendekatan analitik maupun dengan bantuan perangkat lunak POM-QM for Windows.*
4. ***Menginterpretasikan hasil solusi** dalam konteks permasalahan nyata serta menjelaskan implikasinya terhadap efisiensi biaya produksi perusahaan.*

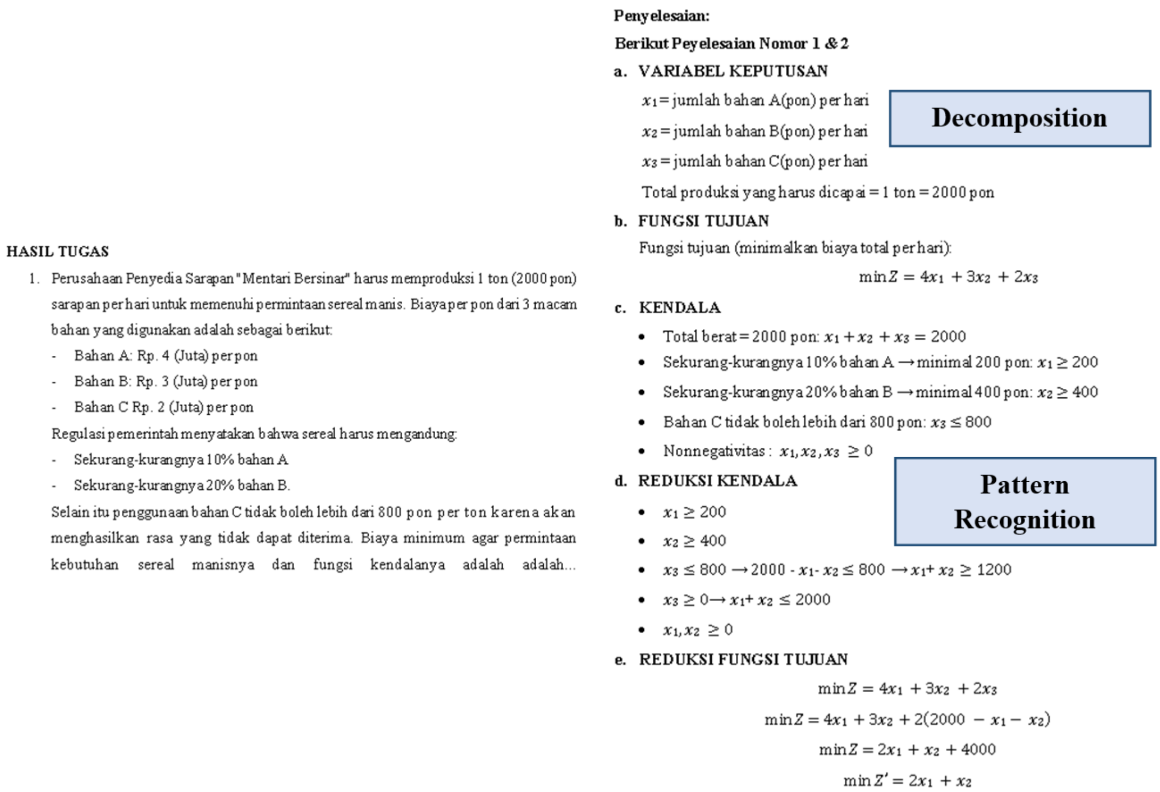
The data in this study consisted of written test results based on computational thinking tasks supported by QM software, interviews with students and lecturers, and student response questionnaires. These four data sources were analyzed integratively to identify the profile of students' computational thinking skills according to the four core indicators: decomposition, pattern recognition, abstraction, and algorithmic thinking. The analysis aimed to map each participant's level of achievement across the indicators while simultaneously identifying aspects that had not been optimally attained. The computational thinking test results of 21 students had a minimum score of 20, a maximum score of 95, a mean of 67.38, a standard deviation of 15.38, and a variance of 236.55.

### Computational Thinking Profile of a High Prior Mathematical Ability Student

To ensure the validity and trustworthiness of this qualitative inquiry, the study applied the criteria of credibility, transferability, dependability, and confirmability. Data credibility was established through technical triangulation, comparing students' written problem-solving results in linear programming tasks assisted by QM software with in-depth semi-structured interviews. Furthermore, persistent observation of students' responses and peer debriefing with mathematics education faculty members at Universitas PGRI Madiun (UNIPMA) were conducted to refine the data analysis. Transferability was addressed by providing a thick description of the decomposition, pattern recognition, abstraction, and algorithmic thinking profiles across the three subject categories. Finally, dependability and confirmability were maintained through a systematic audit trail of the research procedures—from data collection via the university's LMS (eLMA) to final verification—to ensure that the findings purely reflected the subjects' cognitive processes without researcher bias.

The analysis of the high prior mathematical ability student shows that the participant successfully fulfilled all computational thinking indicators. In the decomposition stage, the student was able to clearly identify the decision variables, objective function, and constraints involved in the problem. The mathematical model constructed by the student accurately represented the problem situation. In the pattern recognition stage, the student demonstrated an understanding of the procedural structure of the simplex method. The participant was able to recognize the pattern of iterations and apply the appropriate steps to determine the pivot column and pivot row.

In the abstraction stage, the student effectively reduced the contextual problem to a well-defined mathematical representation. Pertinent information was accurately identified and retained, whereas extraneous information was omitted. Subsequently, in the algorithmic thinking stage, the student exhibited systematic and logical reasoning when executing the simplex method procedures until the optimal solution was achieved. Samples of student responses with high prior mathematical ability are shown in Figure 1 and Figure 2.



**HASIL TUGAS**

1. Perusahaan Penyedia Sarapan "Mentari Bersinar" harus memproduksi 1 ton (2000 pon) sarapan per hari untuk memenuhi permintaan sereal manis. Biaya per pon dari 3 macam bahan yang digunakan adalah sebagai berikut:
   - Bahan A: Rp. 4 (Juta) per pon
   - Bahan B: Rp. 3 (Juta) per pon
   - Bahan C Rp. 2 (Juta) per pon

   Regulasi pemerintah menyatakan bahwa sereal harus mengandung:
   - Sekurang-kurangnya 10% bahan A
   - Sekurang-kurangnya 20% bahan B.

   Selain itu penggunaan bahan C tidak boleh lebih dari 800 pon per ton karena akan menghasilkan rasa yang tidak dapat diterima. Biaya minimum agar permintaan kebutuhan sereal manisnya dan fungsi kendalanya adalah adalah...

**Penyelesaian:**

**Berikut Peyelesaian Nomor 1 & 2**

a. **VARIABEL KEPUTUSAN**

$x_1$ = jumlah bahan A(pon) per hari

$x_2$ = jumlah bahan B(pon) per hari

$x_3$ = jumlah bahan C(pon) per hari

Total produksi yang harus dicapai = 1 ton = 2000 pon

b. **FUNGSI TUJUAN**

Fungsi tujuan (minimalkan biaya total per hari):

$$\min Z = 4x_1 + 3x_2 + 2x_3$$

c. **KENDALA**

- Total berat = 2000 pon: $x_1 + x_2 + x_3 = 2000$
- Sekurang-kurangnya 10% bahan A → minimal 200 pon: $x_1 \geq 200$
- Sekurang-kurangnya 20% bahan B → minimal 400 pon: $x_2 \geq 400$
- Bahan C tidak boleh lebih dari 800 pon: $x_3 \leq 800$
- Nonnegativitas : $x_1, x_2, x_3 \geq 0$

d. **REDUKSI KENDALA**

- $x_1 \geq 200$
- $x_2 \geq 400$
- $x_3 \leq 800 \rightarrow 2000 - x_1 - x_2 \leq 800 \rightarrow x_1 + x_2 \geq 1200$
- $x_3 \geq 0 \rightarrow x_1 + x_2 \leq 2000$
- $x_1, x_2 \geq 0$

e. **REDUKSI FUNGSI TUJUAN**

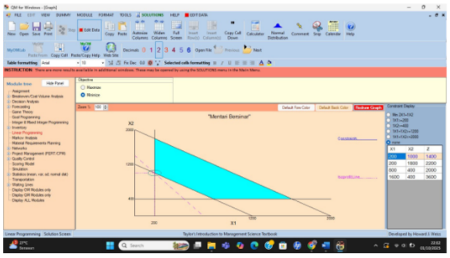
$$\min Z = 4x_1 + 3x_2 + 2x_3$$
$$\min Z = 4x_1 + 3x_2 + 2(2000 - x_1 - x_2)$$
$$\min Z = 2x_1 + x_2 + 4000$$
$$\min Z' = 2x_1 + x_2$$



**Figure 1.** Answers provided by students with high prior mathematical ability (Decomposition & Pattern Recognition)

## Abstraction

**f. DAERAH FEASIBEL**



**g. ANALISIS HASIL**

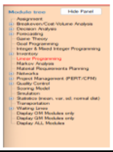
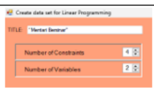
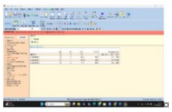
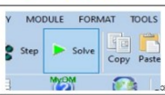
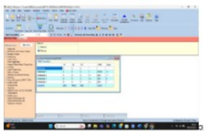
Hasil optimasi pada perangkat lunak **POM-QM** menunjukkan bahwa model linear programming yang dibangun untuk meminimalkan pnggunaan bahan. Ada dua jenis bahan (bahan A sebagai X1 dan bahan B sebagai X2 dengan batasan per pon).

Grafik yang dihasilkan oleh POM-QM memperlihatkan daerah feasible berwarna biru muda, yaitu area yang memenuhi seluruh kendala secara simultan. Garis-garis pembatas yang tampak adalah representasi matematis dari masing-masing kendala sumber daya. Titik-titik pada sudut daerah feasible menunjukkan kombinasi produksi yang memenuhi batasan sumber daya secara tepat.

Grafik POM-QM menunjukkan bahwa kombinasi terbaik untuk meminimalkan keuntungan adalah **$X_1$=200 (bahan A)** dan **$X_2$=1000 (bahan B)** dengan penggunaan minimum 1400 pon per hari.. Daerah biru pada grafik adalah kombinasi yang feasible, dan titik merah muda (garis isoprofit) menunjukkan lokasi keuntungan maksimum yang bersinggungan dengan daerah feasible.

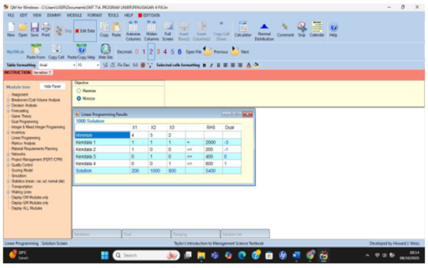
## Algorithmic Thinking

**k. LANGKAH-LANGKAH PENGGUNAAN APLIKASI POM-QM (Metode Simpleks)**

| No | Langkah-langkah | Gambar |
|---|---|---|
| 1. | Instal aplikasinya, lalu buka dengan mengeklik ex pada aplikasi |  |
| 2. | Klik pada pilihan menu 'Linear Programing' |  |
| 3. | Masukkan Title sesuai dengan program masalah yang anda pilih |  |
| 4. | Klik option 'Constraint 1, Constraint 2, Constraint 3,…" |  |
| 5. | Pilih 'Number of Constraints(kendala sesuai model matematis dalam soal)' dan 'Number of Variabels (disesuaikan dengan banyaknya variable pada soal)' |  |
| 6. | Klik option "Minimize" |  |
| 7 | Klik "Ok" |  |
| 8 | Masukkan data sesuai dengan kendala yang sudah dirumuskan ke dalam model matematis |  |
| 9. | Klik 'Solve'<br>Lalu pilih 'Interation' |  |
| 10. | Hasil |  |

## Algorithmic Thinking

**i. METODE SIMPLEKS**



**j. LANGKAH-LANGKAH PENGGUNAAN APLIKASI POM-QM**

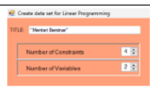
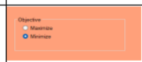
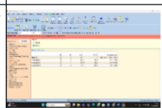
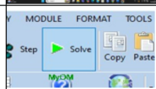
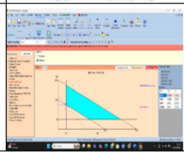
| No | Langkah-langkah | Gambar |
|---|---|---|
| 1. | Instal aplikasinya, lalu buka dengan mengeklik ex pada aplikasi |  |
| 2. | Klik pada pilihan menu 'Linear Programing' |  |
| 3. | Masukkan Title sesuai dengan program masalah yang anda pilih |  |
| 4. | Klik option 'Constraint 1, Constraint 2, Constraint 3,…" |  |
| 5. | Pilih 'Number of Constraints(kendala sesuai model matematis dalam soal)' dan 'Number of Variabels(disesuaikan dengan banyaknya variable pada soal)' |  |
| 6. | Klik option "Minimize" |  |
| 7 | Klik "Ok" |  |
| 8 | Masukkan data sesuai dengan kendala yang sudah dirumuskan ke dalam model matematis |  |
| 9. | Klik 'Solve'<br>Lalu pilih 'Graph' |  |
| 10. | Hasil Grafik |  |

**Figure 2.** Answers provided by students with high prior mathematical ability (Abstraction & Algorithmic Thinking)

Figure 1 and Figure 2 illustrates the solution process of a student with high prior mathematical ability in solving the linear programming problem using the simplex method. In the decomposition stage, the student successfully identified all essential components of the problem, including the decision variables, objective function, and constraints. The variables were clearly defined, and the mathematical model was formulated correctly. In the pattern recognition stage, the student demonstrated the ability to recognize the procedural structure of the simplex method. The student systematically arranged the initial simplex tableau and correctly identified the pivot column and pivot row during each iteration. This indicates that the student understood the repetitive pattern involved in the simplex optimization process.

At the abstraction stage, the student was able to simplify the contextual problem into a clear mathematical representation. The objective function and constraints were expressed concisely, reflecting the student's ability to filter relevant information from the problem context. Finally, in the algorithmic thinking stage, the student applied a sequence of logical and systematic steps to reach the optimal solution. The row operations were performed accurately, and the final tableau correctly indicated the optimal solution. This demonstrates that the student possessed well-developed computational thinking skills and a strong conceptual understanding of the simplex method.

**Computational Thinking Profile of a Moderate Prior Mathematical Ability Student**

The analysis of the student with moderate prior mathematical ability shows that the participant demonstrated a partially developed computational thinking process. In the decomposition stage, the student was able to identify several important elements of the problem, such as the decision variables and the objective function. However, some inaccuracies appeared in the formulation of the constraints. Although the student attempted to represent the problem mathematically, certain coefficients and inequality symbols were not consistently written, indicating that the decomposition process was not yet fully accurate.

In the pattern recognition stage, the student demonstrated an emerging understanding of the procedural structure of the simplex method. The participant was able to construct the initial simplex tableau and showed awareness of the iterative nature of the algorithm. Nevertheless, difficulties were observed when determining the pivot column and pivot row during the optimization process. This suggests that although the student recognized the general pattern of the simplex procedure, the criteria for selecting pivot elements had not been fully mastered.

At the abstraction stage, the student attempted to simplify the contextual problem into a mathematical model. The representation of the objective function and constraints indicated that the student was able to filter some essential information from the problem. However, the abstraction process was not entirely precise because certain elements of the model were still expressed ambiguously. This condition reflects that the student had begun to develop abstraction skills but required further reinforcement to represent mathematical relationships more accurately.

In the algorithmic thinking stage, the student followed several steps of the simplex method but showed inconsistencies in performing row operations and updating tableau values. Although the student attempted to continue the iterative process toward an optimal solution, several intermediate steps contained calculation errors or procedural inaccuracies. This indicates that the student's algorithmic thinking ability had started to develop but was not yet fully stable or systematic. Overall, the computational thinking process of the student in this category can be considered moderately developed, although improvements are still needed in recognizing procedural patterns and implementing accurate algorithmic steps. Responses of students with medium prior mathematical ability are presented in Figure 3 and Figure 4.

**Decomposition**

**HASIL PENGERJAAN TUGAS**

1. Perusahaan Penyedia Sarapan "Mentari Bersinar" harus memproduksi 1 ton (2000 pon) sarapan per hari untuk memenuhi permintaan sereal manis. Biaya per pon dari 3 macam bahan yang digunakan adalah sebagai berikut. Bahan A Rp.4.000.000,- per pon, bahan B Rp.3.000.000,- per pon, dan bahan C Rp.2.000.000,- per pon. Regulasi pemerintah menyatakan bahwa sereal mengandung setidaknya 10% bahan A dan 20% bahan B. Penggunaan bahan C lebih dari 800 pon per ton akan menghasilkan rasa yang tidak dapat diterima. Biaya minimum agar permintaan kebutuhan sereal manisnya adalah…
   A. Maks : Z = 4x + 3y + 2z
   B. Min : Z = 4x + 3y + 2z
   C. Min : Z = x + y + z
   D. Maks : Z = 200x + 400y + 800z
   E. Min : Z = 200x + 400y + 800z

2. Fungsi Kendala Pada No.1 adalah…

**PENYELESAIAN:**

**Dari soal diketahui:**

Perusahaan **"Mentari Bersinar"** harus memproduksi 1 ton (2000 pon) sereal manis per hari dengan 3 bahan:

- Bahan A: biaya $4/pon
- Bahan B: biaya $3/pon
- Bahan C: biaya $2/pon

Ketentuan:

- Kandungan bahan A minimal 10% dari total (≥ 200 pon).
- Kandungan bahan B minimal 20% dari total (≥ 400 pon).
- Kandungan bahan C maksimal 800 pon (≤ 800 pon).
- Total bahan = 2000 pon

**A. Model Matematis**

**1. Variabel Keputusan**

Misalkan:

- $x$ = jumlah bahan A (pon)
- $y$ = jumlah bahan B (pon)
- $z$ = jumlah bahan C (pon)

**2. Fungsi Tujuan**

$Min\ Z = 4x + 3y + 2z$ (karena biaya harus diminimalkan)

**3. Kendala (Costraints)**

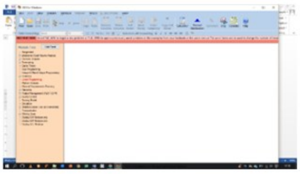
- Jumlah total Campuran: $x + y + z = 2000$
- Kandungan minimum bahan A: $x \geq 200$
- Kandungan minimum bahan B: $y \geq 400$
- Kandungan maksimal bahan C: $z \leq 800$
- $x, y, z \geq 0$

**Figure 3** . Answers provided by students with moderate prior mathematical ability (Decomposition & Pattern Recognition)

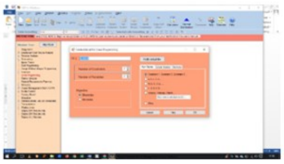
**Algorithmic Thinking**

**B. Langkah penggunaan POM-QM (disertai tangkapan layar).**

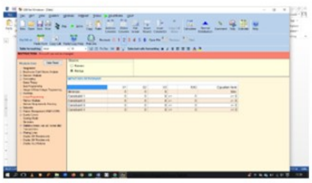
1. Buka POM-QM lalu pilih Linier Programing



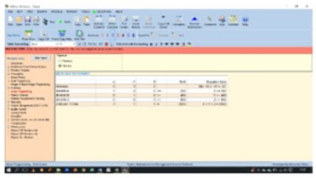
2. Pada bagian title di isi judul, setelah itu pada bagian *number of constrains* isi sesuai dengan banyaknya kendala pada masalah, pada bagian *number of variabel* isi sesuai dengan banyaknya variable, setelah itu pada bagian *objective* pilih yang minimize dan klik oke.

3. 

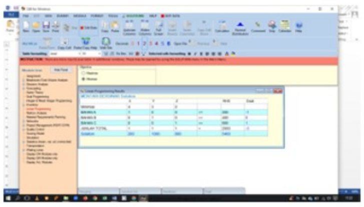
4. Pada bagian $x_1, x_2, x_3$ sesuai dengan permisalan dan setelah itu diisi kolom *minimize* dengan koefisien fungsi tujuan.



5. Pada bagian *constrains* diganti dengan beberapa kendala pada masalah, masukkan angka pada kendala, setelah semua data sudah dimasukkan klik *solve* untuk melihat hasilnya.



6. Setelah itu akan muncul hasil nilai optimal pada table. Dan klik pada bagian Iterasi



7. Akan muncul data seperti pada gambar

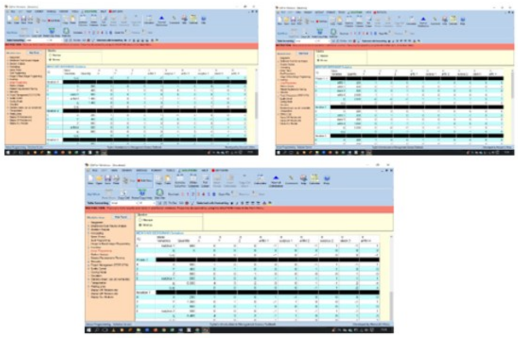


**Figure 4** . Answers provided by students with moderate prior mathematical ability (Algorithmic Thinking)

**Computational Thinking Profile of a Low Prior Mathematical Ability Student**

The student with low prior mathematical ability demonstrated substantial difficulties across nearly all computational thinking indicators. In the decomposition stage, the participant struggled to identify the essential components of the problem. The decision variables were not clearly defined, and the objective function was either incomplete or incorrectly formulated. As a result, the mathematical model constructed by the student did not accurately represent the problem situation. In the pattern recognition stage, the student showed limited understanding of the procedural structure of the simplex method. The participant attempted to construct a solution but did not follow the appropriate format of the simplex tableau. This indicates that the student was not yet able to recognize the repetitive pattern involved in the simplex optimization procedure. The lack of familiarity with the procedural steps prevented the student from progressing through the solution systematically.

During the abstraction stage, the student experienced difficulties in transforming the contextual information into a simplified mathematical model. Several important elements of the problem were either omitted or incorrectly interpreted. This suggests that the student had not yet developed the ability to filter relevant information and represent it effectively in mathematical form. In the algorithmic thinking stage, the student was unable to organize a clear sequence of steps to solve the problem. The procedures used were incomplete and unsystematic, and the student was not able to reach an optimal solution. The solution process appeared to rely more on trial-and-error rather than on a structured understanding of the simplex algorithm. These findings indicate that the student's computational thinking skills were still at an early stage of development and required substantial instructional support, particularly in understanding the structure of linear programming problems and the systematic procedures used to solve them. Responses of students with low prior mathematical ability are presented in Figure 5.

**Decomposition**
(Algorithm not achieved)

**Model Matematis**

1. Variabel Keputusan
   $x$ = jumlah pon bahan A yang digunakan
   $y$ = jumlah pon bahan B yang digunakan
   $z$ = jumlah pon bahan C yang digunakan
2. Parameter
   a. Kapasitas produksi: 2000 pon per hari
   b. Harga bahan:
      Bahan A: Rp4 (juta) per pon
      Bahan B: Rp3 (juta) per pon
      Bahan C: Rp2 (juta) per pon
   c. Regulasi:
      Bahan A minimal 10% dari total → $x \geq 0.1 \times 2000 = 200$
      Bahan B minimal 20% dari total → $y \geq 0.2 \times 2000 = 400$
      Bahan C maksimal 800 pon → $z \leq 800$
3. Fungsi Tujuan
   Minimisasi biaya produksi:
   $\text{Min } Z = 4x + 3y + 2z$
4. Kendala
   a. Total produksi
      $x + y + z = 2000$
   b. Proporsi bahan
      $x \geq 200$
      $y \geq 400$
      $z \leq 800$
   c. Non-negativitas
      $x, y, z \geq 0$
5. Reduksi Kendala
   $x + y + z = 2000$
   $x \geq 200$
   $y \geq 400$
   $z \leq 800$
   - $x, y, z \geq 0$

**Langkah-langkah Penggunaan Aplikasi POM-QM**

| No | Langkah-Langkah |
|---|---|
| 1 | Klik *Linear Programming* pada *Module Tree* maka akan muncul seperti pada gambar tersebut, ubah title menjadi Mentari Bersinar, masukkan *Number of Constraints* menjadi 4 dan *Number of Variables* menjadi 2. Dan ubah *Objective* menjadi *Minimize*, pastikan *Row Names* dipilih *Constraint 1, Constraint 2, Constraint 3* setelah klik "OK". |
| 2 | Masukkan data sesuai dengan kendala yang sudah dirumuskan ke model matematika |
| 3 | Setelah memasukkan semua fungsi kendala, lalu klik ikon "Solve" diatas dan pilih "*Graph*". |
| 4 | Maka akan mendapatkan hasil grafik sebagai berikut. |
| 5 | Setelah memasukkan semua fungsi kendala, lalu klik ikon "Solve" diatas dan pilih "Iterations". |
| 6 | Maka akan mendapatkan hasil metode simpleks sebagai berikut. |

***Penyelesaian***

1. Ubah persamaan garis untuk mencari titik potong
2. Tentukan titik pojok (vertex) daerah feasible

Titik pojok muncul pada potongannya garis batas dan sumbu:

d) Origin: . (jelas feasible)

d) Intersep sumbu : kedua kendala memberi batas dan . Jadi . Titik pojok pada sumbu

d) Intersep sumbu : kedua kendala memberi batas dan . Jadi . Titik pojok pada sumbu $y$

d) Potong antara kedua garis kendala: selesaikan system

Substitusi nilai x ke persamaan

Jadi, titik pojok yang perlu diperiksa adalah:

3. Hitung nilai Z pada tiap titik pojok

Jadi nilai maksimun Z pada titik pojok adalah 5

4. Periksan jawaban

***Penyelesaian***

**Langkah-langkah Pengunaan Aplikasi POM-QM**

| No | Langkah-Langkah |
|---|---|
| 1 | Klik *Linear Programing* pada *Module Tree* maka akan muncul seperti pada gambar tersebut, ubah title menjadi Mentari Bersinar, masukkan *Number of Constraints* menjadi 4 dan *Number of Variables* menjadi 2. Dan ubah *Objective* menjadi *Minimize*, pastikan *Row Names* dipilih *Constraint 1, Constraint 2, Constraint 3* setelah klik 'OK'. |
| 2 | Masukkan data sesuai dengan kendala yang sudah dirumuskan ke model matematis. |
| 3 | Setelah memasukkan semua fungsi kendala, lalu klik ikon '*Solve*' diatas dan pilih '*Graph*'. |
| 4 | Maka akan mendapatkan hasil grafik sebagai berikut. |
| 5 | Setelah memasukkan semua fungsi kendala, lalu klik ikon '*Solve*' diatas dan pilih 'Iterations'. |
| 6 | Maka akan mendapatkan hasil metode simpleks sebagai berikut. |

**Kesimpulan:**

Berdasarkan hasil analisis terhadap dua permasalahan program linear yang diselesaikan menggunakan metode simpleks, dapat disimpulkan bahwa metode ini mampu memberikan solusi optimal secara sistematis dan efisien, baik pada kasus minimisasi maupun maksimisasi fungsi tujuan. Pada kasus pertama, metode simpleks melalui dua fase iterasi berhasil menemukan solusi optimal dengan nilai minimum fungsi tujuan sebesar Z = 2.000 pada kombinasi $x_1 = 800$ dan $x_2 = 400$, sedangkan pada kasus kedua diperoleh nilai maksimum Z = 5 dengan kombinasi $x_1 = 2$ dan $x_2 = 0$. Nilai slack pada tiap kendala menggambarkan tingkat pemanfaatan sumber daya secara optimal.

Jika dibandingkan dengan metode grafik, keduanya memiliki peran dan keunggulan masing-masing dalam penyelesaian masalah program linear. Metode grafik lebih sederhana dan mudah dipahami karena menyajikan representasi visual daerah feasible dan titik-titik pojok sebagai kandidat solusi, namun hanya terbatas pada dua variabel. Sebaliknya, metode simpleks lebih unggul karena dapat digunakan untuk kasus dengan banyak variabel dan kendala, serta menghasilkan solusi secara efisien melalui proses iteratif berbasis aljabar. Dengan demikian, dapat disimpulkan bahwa metode grafik lebih cocok untuk pembelajaran konseptual dan pemahaman dasar optimasi linear, sedangkan metode simpleks lebih tepat digunakan dalam konteks praktis dan kompleks yang memerlukan ketelitian perhitungan dan efisiensi penyelesaian. Kedua metode tersebut saling melengkapi dalam memberikan pemahaman menyeluruh terhadap konsep dan penerapan program linear.

**Figure 5** . Answers provided by students with low prior mathematical ability (Decomposition)

Based on the data obtained, the results of open coding, axial coding, and categorization of students' computational thinking are presented in Table 6 and elaborated below.

**Table 6.** Results of Open Coding, Axial Coding, and Categorization of Student Computational Thinking

| Subject (Ability Level) | Raw Data | Open Coding | Axial Coding | CT Category |
|---|---|---|---|---|
| 1. S-1 (High) | "x1 = amount of A, x2 = amount of B, x3 = amount of C... total = 2000: x1+x2+x3=2000" | Defined quantity variables and total constraint functions. | Broke textual components into variables and linear equations. | Decomposition |
| 2. S-2 (Medium) | "Pivot: x1 enters basis replacing art2. Reason: x1 has most negative coefficient in W row" | Identified entering/ leaving rule for basis variables. | Understood iteration patterns and key selection criteria. | Pattern Recognition |
| 3. S-1 (High) | "Constraint reduction: x3≤800→ 2000−x1−x2≤800 → x1+x2≥1200" | Performed substitution, reduced variables from 3 to 2. | Filtered non-dominant variables to simplify graphical analysis. | Abstraction |
| 4. S-3 (Low) | "Mathematical Model: x = A, y = B, z = C... total x+y+z=2000" | Wrote basic mathematical model. | Identified initial decomposition but did not proceed to algorithmic QM iteration. | Decomposition (Algorithm not achieved) |
| 5. S-1 (Medium) | "Open POM-QM, select Linear Programming... input coefficients... enter constraints, click solve" | Detailed systematic procedures for QM software use. | Designed sequential step-by-step instructions for computerized algorithm. | Algorithmic Thinking |

## The Impact of POM-QM Usage on Students' Computational Thinking

Drawing on data from questionnaires and interviews, the use of POM-QM for Windows yielded differential impacts according to students' levels of prior mathematical ability. High-ability students employed POM-QM for Windows primarily as a validation tool. They were able to connect manual analysis results with graphical visualizations of the feasible region, thereby reinforcing their comprehension of abstraction and decomposition. This suggests that for high-ability students, technology serves to deepen conceptual understanding by providing visual confirmation of analytical processes.

Medium-ability students benefited from the software's capacity to reduce computational burdens, particularly during simplex method iterations. The iteration feature enabled them to discern patterns in solution procedures, thereby supporting pattern recognition and algorithmic thinking. For this group, POM-QM for Windows functioned as a scaffold that facilitated the development of procedural understanding.

In contrast, low-ability students demonstrated suboptimal engagement with the software. Although they were able to obtain final answers, they encountered significant difficulties during the modeling stage. This finding indicates that weak decomposition skills constrain the effective utilization of technology, limiting its potential to support higher-order thinking. In conclusion, the effectiveness of POM-QM for Windows is contingent upon students' prior mathematical abilities. Technology proves most effective as a cognitive support tool for students who already possess a strong foundation in conceptual understanding.

**Discussion**

To ensure the validity of data in this qualitative study, the researcher applied tests of credibility, transferability, dependability, and confirmability based on the criteria established by Lincoln and Guba. Data credibility was pursued through technical triangulation by comparing the results of written tests on linear programming problem-solving using POM-QM for Windows with the results of in-depth interviews based on semi-structured guidelines. Additionally, persistent observation of students' worksheet responses and focused discussions with peer lecturers from the Mathematics Education Study Program at Universitas PGRI Madiun (UNIPMA) were conducted to sharpen data analysis. Transferability was fulfilled by presenting a thick description of the decomposition, pattern recognition, abstraction, and algorithmic thinking profiles of subjects at each level. Finally, dependability and confirmability were ensured through an external audit of the entire research process—from task collection in the eLMA learning management system to drawing conclusions—to guarantee that the research findings originated purely from the subject data without researcher subjectivity bias.

The results of data analysis using the Strauss and Corbin coding model revealed a gradation in the achievement of computational thinking pillars that was strongly influenced by students' prior mathematical ability. At the high-ability level, students were able to perform high-level decomposition and abstraction by reducing mathematical variables so that the objective function could be solved more simply using graphical or simplex methods. At the medium-ability level, students demonstrated strengths in pattern recognition and algorithmic thinking. Riani was able to systematically follow the manual algorithm for using the POM-QM for Windows application and understood the pattern of artificial variables exiting toward optimal conditions. Meanwhile, at the low-ability level, their cognitive mapping only reached the basic decomposition pillar, namely identifying decision variables without being accompanied by the ability to perform advanced abstraction or validated POM-QM for Windows iteration procedures.

Descriptive analysis of the computational thinking test results shows a considerable variation in students' abilities when solving linear programming problems. The scores ranged from 20 to 95, with a mean score of 67.38 and a standard deviation of 15.38. The relatively high range and standard deviation indicate that students' computational thinking abilities are heterogeneous. Although some students demonstrated strong computational reasoning, others experienced substantial difficulties in organizing systematic solution steps. The mean score, which remains significantly lower than the maximum possible score, suggests that overall students' computational thinking skills are at a moderate level and still require further development.

The variability observed in this study can be understood through the role of prior mathematical ability in supporting computational thinking processes. Computational thinking is widely defined as a set of problem-solving skills that include decomposition, pattern recognition, abstraction, and algorithmic thinking (Wing, 2006). These processes require learners to structure problems logically and design systematic procedures for obtaining solutions. In mathematics education, computational thinking is closely related to students' prior conceptual understanding because effective problem solving requires the integration of conceptual knowledge and procedural reasoning (Grover & Pea, 2013).

The findings of this study demonstrate that students with higher prior mathematical ability tend to exhibit more mature computational thinking processes. Students in this category were able to systematically decompose the problem by identifying decision variables, objective functions, and constraints accurately. This ability reflects a well- developed conceptual schema that allows students to organize problem elements effectively. Schema theory suggests that prior knowledge helps learners' structure new information more efficiently, enabling them to solve complex problems through organized cognitive frameworks (Sweller et al., 2011).

Furthermore, students with high prior ability demonstrated strong pattern recognition skills during the simplex procedure. They were able to recognize the iterative structure of the algorithm and correctly determine pivot columns and pivot rows during optimization. Pattern recognition is a critical element of computational thinking because it enables learners to identify recurring structures in problem-solving procedures (Barr & Stephenson, 2011). The ability to recognize such procedural patterns supports the development of efficient problem-solving strategies (Wang et al., 2023). At the abstraction stage, high-ability students successfully transformed contextual problems into precise mathematical representations. They were able to filter relevant information while ignoring unnecessary details. This finding supports previous research indicating that abstraction is a key component of computational thinking because it allows learners to represent complex problems in simplified forms that can be processed algorithmically (Weintrop et al., 2016).

In addition, students with high prior mathematical ability demonstrated well-developed algorithmic thinking. They were able to organize systematic solution steps using the simplex method until reaching the optimal solution. Algorithmic thinking involves designing clear and logical procedures that guide the problem-solving process (Shute et al., 2017). Students who possess strong conceptual understanding are more capable of constructing such procedures because they understand the relationships between each step in the algorithm. In contrast, students with moderate prior mathematical ability demonstrated partially developed computational thinking skills. In the decomposition stage, they were able to identify the main components of the problem, such as variables and objective functions, but some inaccuracies appeared in representing the constraints. These findings suggest that although the students possessed basic conceptual understanding, their knowledge structures were not yet fully stable.

During the pattern recognition stage, moderate-ability students showed an emerging understanding of the simplex procedure. They were able to construct the initial tableau correctly but encountered difficulties when selecting pivot elements during the iteration process (Marcelino et al., 2018). This indicates that the students recognized the existence of procedural patterns but had not fully mastered the criteria required to implement them correctly (Manches et al., 2020). The abstraction process among moderate-ability students was also less precise. Although they attempted to represent the problem mathematically, certain elements of the model were expressed ambiguously (Román-González et al., 2017). Research on computational thinking indicates that abstraction often becomes a challenging stage for learners because it requires the ability to generalize problem structures and identify essential relationships between variables (Angeli & Giannakos, 2020).

Students with low prior mathematical ability demonstrated the greatest difficulties across all computational thinking indicators. In the decomposition stage, they struggled to identify decision variables and construct appropriate mathematical models. As a result, the solutions they produced did not accurately represent the problem situation. These findings indicate that limited conceptual understanding can significantly hinder students' ability to engage in structured problem-solving processes (Kong & Wang, 2021). The difficulties continued in the pattern recognition stage, where students were unable to identify the procedural structure of the simplex method. Instead of following systematic steps, they often relied on trial-and-error approaches. This lack of procedural awareness prevented them from recognizing the algorithmic patterns necessary for solving linear programming problems.

Similarly, the abstraction process among low-ability students was incomplete because several essential elements of the problem were omitted or incorrectly interpreted. Without the ability to abstract relevant information, students were unable to construct meaningful mathematical representations (García-Peñalvo & Mendes, 2018). Consequently, their algorithmic thinking processes were also weak, as they could not organize coherent steps to reach optimal solutions. Overall, the findings confirm that prior mathematical ability plays a significant role in shaping students' computational thinking skills. Students with higher prior ability demonstrated more structured reasoning, while those with lower ability experienced difficulties integrating conceptual understanding with procedural implementation. These findings are consistent with previous research suggesting that the development of computational thinking is strongly influenced by students' prior knowledge and learning experiences (Román-González et al., 2018).

Another important finding is that the most pronounced differences between ability levels appeared in the indicators of pattern recognition and algorithmic thinking. These two indicators require the integration of conceptual understanding and procedural accuracy. Students who lack strong prior mathematical foundations often struggle to organize algorithmic steps systematically. Therefore, instructional strategies in linear programming courses should explicitly scaffold the stages of computational thinking, particularly in helping students recognize procedural patterns and design algorithmic solution steps. From a pedagogical perspective, the results highlight the importance of integrating computational thinking explicitly into mathematics instruction. Structured problem-solving activities, guided modeling practices, and the use of technological tools can support students in developing stronger computational reasoning skills. By gradually guiding students through the stages of decomposition, pattern recognition, abstraction, and algorithmic thinking, educators can help learners develop more systematic approaches to solving complex mathematical problems.

## CONCLUSION

Our findings indicate a clear stratification of computational thinking (CT) skills based on students' prior mathematical levels. While high-ability students manifest a comprehensive mastery of CT components—namely decomposition, pattern recognition, abstraction, and algorithms—moderate-ability students show only partial development with notable procedural gaps. Low-ability students, however, struggle with the foundational abstraction and modeling required for linear programming. This evidence suggests that prior mathematical ability is a primary precursor to the development of sophisticated computational reasoning.

The analysis of open coding, axial coding, and categorization of students' computational thinking reveals that prior mathematical ability influences the achievement of computational thinking pillars. High-ability students (S-1) demonstrated comprehensive mastery across decomposition, abstraction, and the ability to simplify complex problems. Medium-ability students (S-2 and S-1 medium) showed competence in pattern recognition and algorithmic thinking, particularly in identifying iteration patterns and designing systematic procedures for POM-QM for Windows. In contrast, low-ability students (S-3) were only able to perform initial decomposition and did not progress to algorithmic thinking stages. These findings suggest a positive correlation between prior mathematical ability and the level of computational thinking achievement.

Therefore, mathematics instruction, particularly in linear programming courses, should incorporate structured learning strategies that explicitly scaffold the stages of computational thinking. Lecturers are recommended to provide guided modeling activities, structured problem-solving exercises, and technology-supported learning environments to help students progressively develop decomposition, pattern recognition, abstraction, and algorithmic thinking skills. Such instructional approaches are expected to support students with moderate and low prior ability in improving their computational thinking competence more effectively.

**ACKNOWLEDGMENT**
The authors would like to thank the students who participated in this study and the Mathematics Education Study Program of Universitas PGRI Madiun for supporting the research process.

**FUNDING INFORMATION**
This research did not receive external funding.

**AUTHOR CONTRIBUTIONS STATEMENT**

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reza Kusuma Setyansah | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | |
| Swasti Maharani | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | | ✓ | ✓ | | | |

**CONFLICT OF INTEREST STATEMENT**
Authors state no conflict of interest.

**INFORMED CONSENT**
All participants were informed about the purpose of the study, the voluntary nature of their participation, the anonymity and confidentiality of their responses, and their right to decline or withdraw without consequence before they provided consent to complete the questionnaire.

**ETHICAL APPROVAL**
The researchers meticulously followed ethical protocols throughout the research process, adhering to the principles outlined in the Declaration of Helsinki.

**DATA AVAILABILITY**
The data that support the findings of this study are available from the corresponding author upon reasonable request.

## REFERENCES

Aksoy, E. (2025). Perceptions of active learning among faculty in student-centered universities. *Language Teaching and Educational Research, 8*(1), 1–12. https://doi.org/10.35207/later.1674855

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior, 105*, 106185. https://doi.org/10.1016/j.chb.2019.106185

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12. *ACM Inroads, 2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Darmawan, P., Rofiki, I., Mutia, T., Slamet, S., Nugroho, C. M. R., Dewi, V. M., Pramudya, S. S., & Alaiya, S. V. (2024). Eksplorasi computational thinking calon guru matematika dalam menyelesaikan soal pola bilangan. *Jurnal Pendidikan MIPA, 14*(4), 1049–1059. https://doi.org/10.37630/jpm.v14i4.2049

Fauziyah, R. S., & Pujiastuti, H. (2020). Analisis kesalahan siswa dalam menyelesaikan soal cerita program linear berdasarkan prosedur Polya. *UNION: Jurnal Ilmiah Pendidikan Matematika, 8*(2), 253–264. https://doi.org/10.30738/union.v8i2.7747

García-Peñalvo, F. J., & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior, 80*, 407–411. https://doi.org/10.1016/j.chb.2017.12.005

Grover, S., & Pea, R. (2013). Computational thinking in K–12. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Irawati, L., & Hadi, M. S. (2025). Computational thinking dalam pengembangan berpikir matematis di sekolah dasar. *JIIP - Jurnal Ilmiah Ilmu Pendidikan, 8*(2), 2358–2364. https://doi.org/10.54371/jiip.v8i2.7106

Istofany, M. A. B., Afifurrahman, & Negara, H. R. P. (2025). Kemampuan computational thinking siswa pada materi pola bilangan ditinjau dari self-efficacy. *GAUSS: Jurnal Pendidikan Matematika, 8*(1), 64–77. https://doi.org/10.30656/gauss.v8i1.10655

Kaswar, A. B., & Nurjannah, N. (2024). Keefektifan computational thinking dalam meningkatkan kemampuan pemecahan masalah matematika siswa. *SIGMA: Jurnal Pendidikan Matematika, 16*(1), 109–120. https://doi.org/10.26618/sigma.v16i1.14574

Kesuma, G. C., Diani, R., Hasanah, N., & Fujiani, D. (2020). Blended learning model: Can it reduce students' misconception in physics? *Journal of Physics: Conference Series, 1467*(1), Article 012044. https://doi.org/10.1088/1742-6596/1467/1/012044

Kong, S. C., & Wang, Y. Q. (2021). Item response analysis of computational thinking practices: Test characteristics and students' learning abilities in visual programming contexts. *Computers in Human Behavior, 122*, 106836. https://doi.org/10.1016/j.chb.2021.106836

Kuwat, R., Fadhilah, H., & Mulyono, Y. (2025). Transformasi strategi pembelajaran di era digital: Inovasi menuju pendidikan abad 21. *Jurnal Metabio, 7*(1), 1–8. https://doi.org/10.36985/5btv4v84

Maharani, S., Kholid, M. N., Pradana, L. N., & Nusantara, T. (2019). Problem solving in the context of computational thinking. *Infinity Journal, 8*(2), 109–116. https://doi.org/10.22460/infinity.v8i2.p109-116

Maharani, S., Mu'arif, A. N., & Andari, T. (2023). Computational thinking: Vocational students abstraction in solve the geometric pattern problem. *AL-ISHLAH: Jurnal Pendidikan, 15*(4). https://doi.org/10.35445/alishlah.v15i4.2478

Manches, A., McKenna, P. E., Rajendran, G., & Robertson, J. (2020). Identifying embodied metaphors for computing education. *Computers in Human Behavior, 105*, 105859. https://doi.org/10.1016/j.chb.2018.12.037

Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2018). Learning computational thinking and Scratch at distance. *Computers in Human Behavior, 80*, 470–477. https://doi.org/10.1016/j.chb.2017.09.025

Nuraisa, D., Azizah, A. N., Nopitasari, D., & Maharani, S. (2019). Exploring students computational thinking based on self-regulated learning in the solution of linear program problem. *JIPM (Jurnal Ilmiah Pendidikan Matematika), 8*(1), 30–36. https://doi.org/10.25273/jipm.v8i1.4871

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior, 72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Román-González, M., Pérez-González, J.-C., Moreno-León, J., & Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior, 80*, 441–459. https://doi.org/10.1016/j.chb.2017.09.030

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

Siregar, N. S., Siregar, P. S., & Gusmaneli, G. (2024). Transformasi pendidikan agama Islam di era Revolusi Industri 4.0: Strategi menghadapi tantangan teknologi digital dan inovasi. *Populer: Jurnal Penelitian Mahasiswa, 3*(2), 1–9. https://doi.org/10.58192/populer.v3i2.2071

Suprapto, E., Setyansah, R. K., & Devina, D. (2021). GeoGebra application based tutorial materials to improve spatial mathematics abilities in vocational high schools. *Jurnal Pendidikan Teknologi dan Kejuruan, 27*(2), 175–181. https://doi.org/10.21831/jptk.v27i2.39099

Suratih, S., & Pujiastuti, H. (2020). Analisis kesalahan siswa dalam menyelesaikan soal cerita program linear berdasarkan Newman's error analysis. *Pythagoras: Jurnal Pendidikan Matematika, 15*(2). https://doi.org/10.21831/pg.v15i2.30990

Sweller, J., Ayres, P., & Kalyuga, S. (2011). *Cognitive load theory*. Springer. https://doi.org/10.1007/978-1-4419-8126-4

Syari, A. K., & Meiliasari, M. (2024). Systematic literature review: Peningkatan kemampuan berpikir komputasional matematis siswa. *ALGORITMA: Journal of Mathematics Education, 6*(2), 111–123. https://doi.org/10.15408/ajme.v6i2.42816

Umar, M., & Ko, I. (2022). E-learning: Direct effect of student learning effectiveness and engagement through project-based learning, team cohesion, and flipped learning during the COVID-19 pandemic. *Sustainability, 14*(3), 1724. https://doi.org/10.3390/su14031724

Wang, X., Li, L., Tan, S. C., Yang, L., & Lei, J. (2023). Preparing for AI-enhanced education: Conceptualizing and empirically examining teachers' AI readiness. *Computers in Human Behavior, 146*, 107798. https://doi.org/10.1016/j.chb.2023.107798

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