International Journal of Applied Educational Research (IJAER)
Vol. 4 No. 4 , 2026: 301 - 310

The Use of Gap (Groups, Algorithms, And Programming) Mathematical Software in Group Theory Topics of The Algebraic Structures Course

Donni Andreas Nainggolan[1*], Khairunnisa[2], Hizkia Ronaldus Silalahi[3]
[1,3] Prodi Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas HKBP Nommensen Pematangsiantar, Indonesia.
[2] Prodi Pendidikan Matematika, Fakultas Ilmu Tarbiyah dan Keguruan, Universitas Islam Negeri Sumatera Utara, Indonesia.
Corresponding Author: Donni Andreas Nainggolan, donniandreasngl@gmail.com

A R T I C L E I N F O

Keywords: GAP Software, Group Theory, Learning Media.

Received : 24 May
Revised : 25 July
Accepted: 25 August

A B S T R A C T

Algebraic Structures is a fundamental course for undergraduate students in Mathematics and Mathematics Education programs, where group theory concepts often present challenges due to their abstract nature. This study aims to examine the role of GAP (Groups, Algorithms, and Programming) mathematical software as a computational tool for supporting the exploration and verification of group theory concepts. This research employs a literature review methodology by analyzing relevant studies on computer algebra systems and mathematics learning, followed by a computational demonstration using GAP to solve group theory problems. The findings show that GAP can generate Cayley tables and verify group axioms, including closure, associativity, identity, and inverse elements. The computational results are consistent with manual calculations while providing systematic and accurate procedures. This study concludes that GAP provides valuable computational support for visualizing and analyzing abstract algebraic concepts, particularly in the context of group theory learning.

DOI : https://doi.org/10.59890/ijaer.v4i4.11
ISSN-E: **3026-1872**
https://journalijaer.my.id/index.php/ijaer/index

INTRODUCTION

Algebraic Structures, commonly known as Abstract Algebra, is a fundamental area of mathematics that explores mathematical systems based on sets, operations, and their underlying properties. This subject is a compulsory course for undergraduate students in Mathematics and Mathematics Education programs because it provides essential foundations for understanding advanced mathematical concepts and logical reasoning. Among its topics, group theory represents one of the most important concepts because it introduces students to the formal relationship between mathematical objects, operations, and abstract structures (Isa, 2024).

Despite its fundamental role, group theory is often considered one of the most challenging topics in Algebraic Structures courses. Unlike elementary mathematics, which commonly involves numerical calculations, group theory requires students to reason with abstract objects, axioms, and symbolic representations. Students are expected to understand definitions, apply mathematical properties, and construct rigorous proofs based on logical arguments. The ability to construct and evaluate mathematical proofs depends strongly on students' understanding of definitions, axioms, and the relationships among mathematical concepts (Rojano, 2022). Therefore, difficulties in visualizing abstract structures and verifying mathematical properties may become significant obstacles in achieving conceptual understanding.

One of the central concepts in group theory is the definition of a group, which consists of a non-empty set equipped with a binary operation satisfying four fundamental properties: closure, associativity, the existence of an identity element, and the existence of inverse elements. Although these properties provide a clear formal framework, students frequently experience difficulties when applying these axioms to determine whether a mathematical structure satisfies the requirements of a group. Such difficulties are commonly associated with limited experience in exploring examples, performing abstract reasoning, and connecting symbolic representations with concrete mathematical processes (Manurung et al., 2024).

The development of technology in mathematics education has encouraged the integration of computational tools to support the exploration of abstract mathematical concepts. Computer Algebra Systems (CAS) provide opportunities for students and researchers to perform symbolic calculations, visualize mathematical structures, and verify complex mathematical relationships. One of the open-source CAS tools specifically developed for computational discrete algebra is GAP (Groups, Algorithms, and Programming). GAP enables users to analyze algebraic structures, perform group computations, generate Cayley tables, and verify group axioms through computational procedures.

Previous studies have discussed the use of mathematical software as a supporting tool in mathematics learning; however, most studies focus on general computational assistance or visualization purposes. Limited attention has been given to the specific application of GAP in systematically demonstrating group theory concepts, particularly in verifying the fundamental axioms of groups through computational procedures. Therefore, a clearer investigation is needed regarding how GAP can support the exploration and representation of abstract concepts within Algebraic Structures courses.

This study contributes by presenting a computational demonstration of GAP implementation in solving group theory problems and examining how the software represents mathematical processes that are commonly performed manually. Rather than directly measuring improvements in student learning outcomes, this study focuses on analyzing the computational capabilities of GAP and its potential role as a supporting medium for exploring abstract algebraic concepts. Thus, this research aims to examine the use of GAP (Groups, Algorithms, and Programming) mathematical software in group theory topics within the Algebraic Structures course.

LITERATURE REVIEW

Algebraic Structures Learning and Conceptual Challenges in Group Theory

Algebraic Structures is a fundamental branch of mathematics that investigates mathematical systems consisting of sets, operations, and their underlying properties. Learning algebraic structures requires students to develop the ability to interpret mathematical definitions, symbolic representations, logical relationships, and formal arguments. The interaction between mathematical language and symbolic representation plays an important role in helping students construct meaningful mathematical reasoning and valid proofs (Isa, 2024; Rojano, 2022).

Among various topics in Algebraic Structures, group theory represents one of the most essential concepts because it introduces students to abstract mathematical structures based on axiomatic systems. A group is defined as a non-empty set equipped with a binary operation that satisfies four fundamental properties: closure, associativity, the existence of an identity element, and the existence of inverse elements (Saragih, 2023). Although these properties provide a rigorous mathematical framework, students frequently experience difficulties when applying abstract definitions to specific mathematical objects. The challenges are commonly related to difficulties in visualizing abstract structures, identifying group properties, and constructing deductive proofs (Manurung et al., 2024).

Computational Support in Abstract Mathematics Learning

The abstract characteristics of group theory indicate the need for learning approaches that can provide clearer representations of mathematical concepts. Computational approaches offer opportunities for students to explore mathematical structures through symbolic manipulation, verification processes, and visualization of abstract relationships. Previous studies have indicated that technology-supported learning environments can assist students in understanding complex mathematical concepts by providing interactive representations and systematic problem-solving procedures (Haryani et al., 2024).

One computational approach that is relevant to abstract algebra learning is the utilization of Computer Algebra Systems (CAS). Unlike conventional mathematical calculations, CAS allows users to examine mathematical structures through algorithm-based computation and automated verification. In the context of group theory, computational tools can assist in generating examples, constructing Cayley tables, and testing whether a mathematical structure satisfies group axioms.

GAP Mathematical Software as a Computational Tool for Group Theory

GAP (Groups, Algorithms, and Programming) is an open-source Computer Algebra System specifically designed for computational discrete algebra. The software provides various functions for studying algebraic structures, particularly finite groups, through symbolic computation and algorithmic procedures. GAP enables users to represent group elements, perform operations, generate Cayley tables, and verify fundamental properties such as closure, associativity, identity, and inverse elements.

Previous discussions regarding mathematical software have generally emphasized the role of technology in supporting computation and visualization. However, studies that specifically examine the application of GAP in systematically demonstrating group theory concepts through verification of group axioms remain limited. Therefore, this study positions GAP not merely as a calculation tool, but as a computational medium that supports the exploration and representation of abstract algebraic concepts. Based on the reviewed literature, the conceptual relationship of this study can be summarized as follows:

1. Conceptual Difficulties in Understanding Group Properties
2. Need for Computational Representation
3. Implementation of GAP Mathematical Software
4. Verification and Exploration of Group Axioms

This framework provides the basis for analyzing how GAP can support the computational exploration of group theory concepts within the Algebraic Structures course. However, because this study does not directly measure students' learning outcomes, the role of GAP is discussed as a supporting computational approach rather than as direct evidence of improved student achievement.

RESEARCH RESULTS

In the Algebraic Structures course, group theory concepts form the foundation for students to determine whether a given set equipped with a specific operation satisfies the criteria of a group. Generally, the problem-solving procedure begins with constructing the Cayley table and subsequently verifying the four group axioms: closure, associativity, the existence of an identity element, and the existence of an inverse element. When solving such problems manually, a high degree of precision is required from students to avoid computational errors, given the lengthy and meticulous nature of the calculation process. The following problem statement serves as an example that will be solved using both manual computation and GAP mathematical software.

Problem 1: Let G be the set of integers modulo 9, denoted as $\mathbb{Z}_9$. Show that $\mathbb{Z}_9$ forms a group under the operation of addition, denoted as $(G, +)$.

Manual Solution:
The set is defined as $\mathbb{Z}_9 = \{0,1,2,3,4,5,6,7,8\}$.

Table 1. Cayley Table

$(Z_9, +)$	0	1	2	3	4	5	6	7	8
0	0	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8	0
2	2	3	4	5	6	7	8	0	1
3	3	4	5	6	7	8	0	1	2
4	4	5	6	7	8	0	1	2	3
5	5	6	7	8	0	1	2	3	4
6	6	7	8	0	1	2	3	4	5
7	7	8	0	1	2	3	4	5	6
8	8	0	1	2	3	4	5	6	7

Based on the Cayley table, it is evident that:

1. Closure property: The closure property is satisfied under the operation of addition modulo 9.
2. Associative property: The associative property is satisfied.
3. Identity element: The existence of the identity element ($e = 0$) is verified.
4. Inverse elements: The existence of inverse elements is verified, specifically as follows:

$(0)^{-1} = 0$ $(3)^{-1} = 6$ $(6)^{-1} = 3$
$(1)^{-1} = 8$ $(4)^{-1} = 5$ $(7)^{-1} = 2$
$(2)^{-1} = 7$ $(5)^{-1} = 4$ $(8)^{-1} = 1$
Consequently, $(\mathbb{Z}_9, +)$ forms a group.

As an interactive computer algebra system, GAP mathematical software (Groups, Algorithms, and Programming) operates based on the "read, evaluate, and print" principle. Users simply input text-based commands, and the system immediately processes and displays the results. This capability is highly advantageous for group theory computations, as users can directly define functions and test them with various arguments to understand their operational behavior.

Based on the literature review, it can be concluded that GAP mathematical software is highly beneficial in supporting the learning of Algebraic Structures. For instance, students can utilize this software to analyze sets and determine whether a given set constitutes a group. This application is demonstrated in the preceding example, where GAP mathematical software was employed to verify whether $\mathbb{Z}_9$forms a group under the operation of addition.

GAP mathematical software operates by representing mathematical structures as data structures and providing the corresponding operations and functions. The steps for its utilization are as follows:

1. Installation and Basic Usage of GAP Mathematical Software

   Download and install the GAP mathematical software from the official website (https://www.gap-system.org/). Subsequently, launch the software from the terminal or command prompt to access the interactive environment.
2. Input and Command Execution

   GAP mathematical software utilizes its proprietary programming language to perform computations. Users can directly input commands into the interpreter.
3. Computations in Group Theory and Algebra

   In the context of group theory, GAP mathematical software can be utilized to generate the elements of a Cayley table and to verify whether a given set satisfies the group axioms: the closure property, the associative property, and the existence of identity and inverse elements. The output generated by the software is displayed immediately after the user inputs a command. Fundamentally, the software operates by reading the user's commands, manipulating the underlying mathematical data structures, and returning the results based on the executed operations.

Application of GAP Mathematical Software in Solving Problem 1

Let G be the set of integers modulo 9, denoted as $\mathbb{Z}_9$. Show that $\mathbb{Z}_9$ forms a group under the operation of addition, denoted as $(G, +)$.

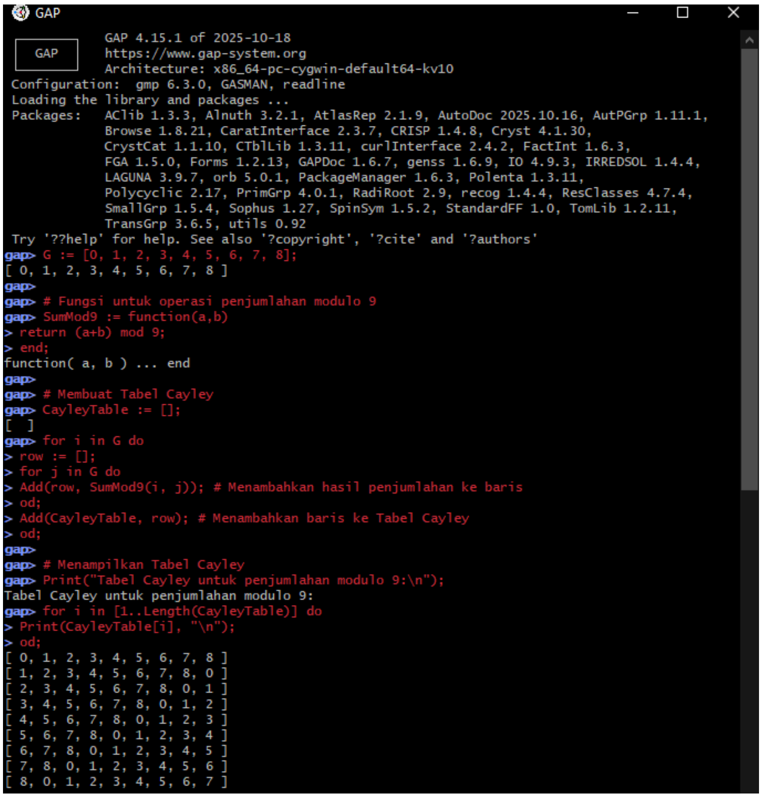

Figure 1. Cayley Table $(Z_9, +)$

Figure 1 illustrates the Cayley table for $(\mathbb{Z}_9, +)$. It is clearly evident that the Cayley table generated by the GAP mathematical software is entirely consistent with the results derived from the manual computation. Subsequently, the demonstration of the closure property using the GAP mathematical software will be presented.

```
gap> G := [0, 1, 2, 3, 4, 5, 6, 7, 8];
[ 0, 1, 2, 3, 4, 5, 6, 7, 8 ]
gap>
gap> Print("a. Tertutup\n");
a. Tertutup
gap> Print("Ambil sebarang nilai dari G\n");
Ambil sebarang nilai dari G
gap> Print("Misal 0, 1, 2, 3, 4, 5, 6, 7, 8 anggota G\n\n");
Misal 0, 1, 2, 3, 4, 5, 6, 7, 8 anggota G

gap>
gap> # Contoh operasi penjjumlahan dalam G
gap> elemen := [1, 1, 1, 1, 1, 1, 1, 1]; # Elemen pertama dalam operasi
[ 1, 1, 1, 1, 1, 1, 1, 1 ]
gap> penjumlahan := [2, 3, 4, 5, 6, 7, 8, 0]; # Elemen kedua dalam operasi
[ 2, 3, 4, 5, 6, 7, 8, 0 ]
gap>
gap> for i in [1..Length(elemen)] do
> hasil := (elemen[i] + penjumlahan [i]) mod 9;
> Print(elemen[i], "+", penjumlahan [i], "=", hasil, "\n");
> od;
1+2=3
1+3=4
1+4=5
1+5=6
1+6=7
1+7=8
1+8=0
1+0=1
gap>
gap> Print("\n Karena hasilnya 0, 1, 3, 4, 5, 6, 7, 8 anggota G maka Tertutup terhadap G\n");

 Karena hasilnya 0, 1, 3, 4, 5, 6, 7, 8 anggota G maka Tertutup terhadap G
```

Figure 2. Closure property

It is clearly evident in Figure 2 that all elements generated by the GAP mathematical software are members of $\mathbb{Z}_9$. Consequently, it can be concluded that the closure property in $\mathbb{Z}_9$is satisfied, which is entirely consistent with the manual solution. Subsequently, the demonstration of the second axiom, namely the associative property, using the GAP mathematical software will be presented.

```
GAP
 GAP        GAP 4.15.1 of 2025-10-18
            https://www.gap-system.org
            Architecture: x86_64-pc-cygwin-default64-kv10
 Configuration:  gmp 6.3.0, GASMAN, readline
 Loading the library and packages ...
 Packages:  AClib 1.3.3, Alnuth 3.2.1, AtlasRep 2.1.9, AutoDoc 2025.10.16, AutPGrp 1.11.1, Br
            CaratInterface 2.3.7, CRISP 1.4.8, Cryst 4.1.30, CrystCat 1.1.10, CTblLib 1.3.11,
            FactInt 1.6.3, FGA 1.5.0, Forms 1.2.13, GAPDoc 1.6.7, genss 1.6.9, IO 4.9.3, IRRE
            LAGUNA 3.9.7, orb 5.0.1, PackageManager 1.6.3, Polenta 1.3.11, Polycyclic 2.17, P
            RadiRoot 2.9, recog 1.4.4, ResClasses 4.7.4, SmallGrp 1.5.4, Sophus 1.27, SpinSym
            TomLib 1.2.11, TransGrp 3.6.5, utils 0.92
 Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> # Mendefinisikan grup Z9 dengan operasi penjumlahan modulo 9
gap> G := [0, 1, 2, 3, 4, 5, 6, 7, 8];
[ 0, 1, 2, 3, 4, 5, 6, 7, 8 ]
gap>
gap> # Mengecek sifat asosiatif untuk a = 2, b = 4, c = 8
gap> a := 2;
2
gap> b := 4;
4
gap> c := 8;
8
gap>
gap> left := ((a+b) mod 9 + c) mod 9;
5
gap> right := (a + (b+c) mod 9) mod 9;
5
gap>
gap> Print("Mengecek sifat asosiatif untuk a = 2, b = 4, c = 8:\n");
Mengecek sifat asosiatif untuk a = 2, b = 4, c = 8:
gap> Print("(", a, " + ", b, ") + ", c ," = ", left, "\n");
(2 + 4) + 8 = 5
gap> Print(a, " + (", b, " + ", c, ") = ", right, "\n");
2 + (4 + 8) = 5
gap>
gap> if left = right and left = 5 then
> Print(" Sifat Asosiatif Terpenuhi: (2 + 4) + 5 = 2 + (4 + 5) = 5\n");
 Sifat Asosiatif Terpenuhi: (2 + 4) + 5 = 2 + (4 + 5) = 5
```

Figure 3. Associative property

The GAP mathematical software output in Figure 3 demonstrates that the associative property is satisfied, thereby confirming that the second group axiom for $(\mathbb{Z}_9, +)$holds. The analysis now proceeds to the third axiom, namely the existence of an identity element.

```
gap> G := [0, 1, 2, 3, 4, 5, 6, 7, 8];
[ 0, 1, 2, 3, 4, 5, 6, 7, 8 ]
gap> # Menentukan Unsur Identitas (e = 0)
gap> e := 0;
0
gap>
gap> Print("Mengecek adanya unsur identitas (satuan) e = 0:\n");
Mengecek adanya unsur identitas (satuan) e = 0:
gap>
gap> identitas_valid := true;
true
gap> for g in G do
> left := (g + e) mod 9;
> right := (e + g) mod 9;
> Print(g, " + ", e, " = ", left, " | ", e, " + ", g, " = ", right, "\n");
>
> if left <> g or right <> g then
> identitas_valid := false;
> fi;
> od;
0 + 0 = 0 | 0 + 0 = 0
1 + 0 = 1 | 0 + 1 = 1
2 + 0 = 2 | 0 + 2 = 2
3 + 0 = 3 | 0 + 3 = 3
4 + 0 = 4 | 0 + 4 = 4
5 + 0 = 5 | 0 + 5 = 5
6 + 0 = 6 | 0 + 6 = 6
7 + 0 = 7 | 0 + 7 = 7
8 + 0 = 8 | 0 + 8 = 8
gap>
gap> if identitas_valid then
> Print(" G memiliki unsur identitas (satuan) yaitu e = 0\n");
 G memiliki unsur identitas (satuan) yaitu e = 0
```

Figure 4. Identity element

It is clearly evident in Figure 4 that the output generated for $(\mathbb{Z}_9, +)$confirms the existence of an identity element, specifically 0, thereby satisfying the third group axiom. Subsequently, the existence of inverse elements will be demonstrated as the fourth group axiom.

```
GAP
od;
^^
gap> G := [0, 1, 2, 3, 4, 5, 6, 7, 8];
[ 0, 1, 2, 3, 4, 5, 6, 7, 8 ]
gap> Print("Adanya unsur invers atau balikan\n");
Adanya unsur invers atau balikan
gap>
gap> # Menyimpan hasil invers untuk setiap elemen
gap> for g in G do
>     for g_inv in G do
>         if (g + g_inv) mod 9 = 0 then
>             Print("Ambil sebarang nilai dari G, misalkan ", g, " anggota G, pilih ", g_inv,
" = G, \n");
>             Print("sehingga ", g, " + ", g_inv, " = 0 = e, maka (", g, ")^-1 = ", g_inv, "\n
\n");
>             break;
>          fi;
>      od;
> od;
Ambil sebarang nilai dari G, misalkan 0 anggota G, pilih 0 = G,
sehingga 0 + 0 = 0 = e, maka (0)^-1 = 0

Ambil sebarang nilai dari G, misalkan 1 anggota G, pilih 8 = G,
sehingga 1 + 8 = 0 = e, maka (1)^-1 = 8

Ambil sebarang nilai dari G, misalkan 2 anggota G, pilih 7 = G,
sehingga 2 + 7 = 0 = e, maka (2)^-1 = 7

Ambil sebarang nilai dari G, misalkan 3 anggota G, pilih 6 = G,
sehingga 3 + 6 = 0 = e, maka (3)^-1 = 6

Ambil sebarang nilai dari G, misalkan 4 anggota G, pilih 5 = G,
sehingga 4 + 5 = 0 = e, maka (4)^-1 = 5

Ambil sebarang nilai dari G, misalkan 5 anggota G, pilih 4 = G,
sehingga 5 + 4 = 0 = e, maka (5)^-1 = 4

Ambil sebarang nilai dari G, misalkan 6 anggota G, pilih 3 = G,
sehingga 6 + 3 = 0 = e, maka (6)^-1 = 3

Ambil sebarang nilai dari G, misalkan 7 anggota G, pilih 2 = G,
sehingga 7 + 2 = 0 = e, maka (7)^-1 = 2

Ambil sebarang nilai dari G, misalkan 8 anggota G, pilih 1 = G,
sehingga 8 + 1 = 0 = e, maka (8)^-1 = 1

gap>
gap> Print("Maka G memiliki unsur invers atau balikan\n");
Maka G memiliki unsur invers atau balikan
```

Figure 5. Inverse elements

Figure 5 clearly illustrates the output from the GAP mathematical software, demonstrating that the fourth group axiom is also satisfied, as evidenced by the existence of inverse elements for $(\mathbb{Z}_9, +)$. Since all four group axioms have been verified, it can be conclusively determined that $(\mathbb{Z}_9, +)$constitutes a group. This finding demonstrates that both the manual computation and the GAP mathematical software yield identical results, thereby providing strong validation for the computational approach. Consequently, this integrated approach will significantly assist students in gaining a deeper understanding of Group Theory concepts within the Algebraic Structures course.

CONCLUSION AND RECOMMENDATION

Based on the research findings, it can be concluded that GAP mathematical software is specifically utilized for computation and problem-solving in group theory. The system not only presents the final solutions but also comprehensively details the computational steps, demonstrating a level of accuracy that is consistent with manual calculations. Furthermore, GAP mathematical software facilitates the verification of group axioms, including the closure property, the associative property, and the existence of identity and inverse elements, by employing binary logic indicators (true/false). This capability significantly assists users in testing the validity of a given set against the criteria for group formation. Consequently, GAP mathematical software has proven to be an effective and essential computational instrument in supporting the exploration and in-depth understanding of group concepts within the Algebraic Structures course.

ADVANCED RESEARCH

Further research should examine the effectiveness of GAP mathematical software through classroom-based experimental studies. Future studies may compare students' conceptual understanding, problem-solving abilities, and learning motivation before and after using GAP. The software may also be applied to more advanced group theory topics, such as subgroups, cyclic groups, permutation groups, homomorphisms, and quotient groups. This development is expected to provide stronger empirical evidence regarding the contribution of GAP to learning abstract algebraic concepts.

ACKNOWLEDGMENT

We would like to express our gratitude to LPPM Universitas HKBP Nommensen Pematangsiantar for its support of this research.

REFERENCES

The GAP Group. (n.d.). *GAP system for computational discrete algebra*. Retrieved July 22, 2026, from https://www.gap-system.org/

Haryani, M., Wahyuningtyas, R., Sakinah, Z. N., & Susilo, B. E. (2024). Studi literatur: Penerapan media pembelajaran augmented reality dalam pembelajaran matematika guna meningkatkan kemampuan pemecahan masalah siswa. *PRISMA, Prosiding Seminar Nasional Matematika, 7*, 359–367. https://proceeding.unnes.ac.id/prisma/article/view/2975

Isa, R. A., & Alisah, E. (2024). Struktur dan sifat-sifat K-aljabar. *Jurnal Riset Mahasiswa Matematika, 3*(4), 200–205. https://doi.org/10.18860/jrmm.v3i4.27187

Lubis, Y. W. (2024). Pembentukan karakter unggul: Analisis optimalisasi pendidikan melalui Organisasi Siswa Intra Madrasah (OSIM) di MAN 2 Deli Serdang. *Bersatu: Jurnal Pendidikan Bhinneka Tunggal Ika, 2*(1), 274–282. https://doi.org/10.51903/bersatu.v2i1.554

Lubis, Y. W., & Ritonga, A. A. (2023). Mobilization school program: Implementation of Islamic religious education teacher preparation in elementary schools. *Jurnal At-Tarbiyat: Jurnal Pendidikan Islam, 6*(1), 144–158. https://doi.org/10.37758/jat.v6i1.632

Manurung, S. L., Mutiara, A. A., Rambe, R. B., & Angraini, R. T. (2024). Penerapan bahasa pemrograman Pascal pada mata kuliah struktur aljabar dalam membuktikan suatu grup. *Jurnal Ilmu Pendidikan dan Pembelajaran, 6*(3), 237–247. https://journalversa.com/s/index.php/jipp/article/view/2444

Ritonga, A. A., Lubis, Y. W., Masitha, S., & Harahap, C. P. (2022). Program Sekolah Penggerak sebagai inovasi meningkatkan kualitas pendidikan di SD Negeri 104267 Pegajahan. *Jurnal Pendidikan, 31*(2), 195–206. https://doi.org/10.32585/jp.v31i2.2637

Rojano, T. (Ed.). (2022). *Algebra structure sense development amongst diverse learners: Theoretical and empirical insights to support in-person and remote learning*. Routledge. https://doi.org/10.4324/9781003197867

Saragih, S. (2023). *Struktur aljabar 1*. Larispa Indonesia.

Setiawan, W., Juniati, D., & Khabibah, S. (2024). Studi literatur: Berpikir kreatif dalam pemecahan masalah matematika. *Jurnal Ilmiah Soulmath: Jurnal Edukasi Pendidikan Matematika, 12*(1), 43–54. https://doi.org/10.25139/smj.v12i1.7548