International Journal of

# Applied Sciences and Smart Technologies

Volume 01, Issue 02, December 2019



p-ISSN 2655-8564 & e-ISSN 2685-9432

# International Journal of Applied Sciences and Smart Technologies

p-ISSN : 2655 - 85
e-ISSN : 2655 - 94


HOME ABOUT LOGIN REGISTER SEARCH CURRENT ARCHIVES ANNOUNCEMENTS

Home > About the Journal > **Editorial Team**

# Editorial Team




**Publisher** : Fakultas sains dan Teknologi

**Society/Institution** : Universitas Sanata Dharma


OPEN JOURNAL SYSTEMS

Journal Help

**USER**

Username
Password
Remember me
Login

**NOTIFICATIONS**

- View
- Subscribe

**LANGUAGE**

Select Language
English
Submit

**JOURNAL CONTENT**

Search

Search Scope
All
Search

**Browse**

- By Issue
- By Author
- By Title
- Other Journals

**FONT SIZE**

**INFORMATION**

- For Readers
- For Authors
- For Librarians

Article Template

International Journal of Applied Sciences and Smart Technologies



# Volume 01, Issue 02, December 2019

DOI: https://doi.org/10.24071/ijasst.v1i2

IJASST Volume 1 No.2 Desember, 2019

## Full Issue

View or download the full issue PDF

## Table of Contents

### Articles



OPEN JOURNAL SYSTEMS

Journal Help

**USER**
Username
Password
Remember me
Login

**NOTIFICATIONS**
- View
- Subscribe

**LANGUAGE**
Select Language
English
Submit

**JOURNAL CONTENT**
Search
Search Scope
All
Search

**Browse**
- By Issue
- By Author
- By Title
- Other Journals

**FONT SIZE**

**INFORMATION**
- For Readers
- For Authors
- For Librarians

Article Template

KALBIScientia
Jurnal Sains dan Teknologi

ISSN 2356 - 4393

# Testing Algebraic Structures Using A Computer Program

**Ricky Aditya[1], Muhammad Taufiq Zulfikar[2], Ngarap Imanuel Manik[3]**

Department of Mathematics and Statistics, BINUS University,
Jalan K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480
[1] *Email: riaditya@binus.edu*

***Abstract:*** *Abstract algebra is a branch of mathematics which is studying algebraic structures of sets with respect to some operations on them. In abstract algebra, some structures such as groups, rings, integral domains and fields, are studied. Each structure has its own axioms and properties. In some cases, to test what kind of structure is a given set and operations is difficult to be done manually. To help this, an application program to test algebraic structures is developed. In this article, an application program to test some derived structures of rings such as commutative rings, integral domains, fields and finite fields is created by using Java, an open-source based programming language. This application program provides testing of various input set such as integers, matrices and alphabets. By this application, testing of algebraic structures can be done faster than the manual one and it has accurate results.*

***Keywords:*** *abstract algebra, ring theory, structure testing*

***Abstrak:*** *Aljabar abstrak merupakan suatu cabang ilmu matematika yang mempelajari struktur aljabar dari suatu himpunan terhadap operasi-operasi pada himpunan tersebut. Dalam aljabar abstrak, dipelajari sejumlah struktur seperti grup, ring, daerah integral dan lapangan. Dalam sejumlah kasus, pengujian jenis struktur dari suatu himpunan terhadap suatu operasi sulit untuk dikerjakan secara manual. Untuk membantu hal ini, dikembangkan sebuah program aplikasi untuk menguji struktur aljabar. Dalam paper ini, dibuat sebuah program aplikasi untuk menguji sejumlah struktur terkait dengan ring, antara lain ring komutatif, daerah integral, lapangan dan lapangan hingga, menggunakan Java, sebuah bahasa pemrograman berbasis open-source. Program aplikasi ini menyediakan pengujian dari himpunan input yang bervariasi, antara lain bilangan bulat, matriks dan alfabet. Dengan aplikasi ini, pengujian struktur aljabar dapat dikerjakan lebih cepat daripada pengujian manual dan hasilnya akurat.*

***Kata Kunci :*** *aljabar abstrak, teori ring, pengujian struktur*

## I. INTRODUCTION

Abstract algebra, sometimes also called as modern algebra, is a branch of mathematics studying algebraic structures, such as groups, rings and fields [1]. Algebraic structures generally refers to a set with one or more finitary operations defined on it. An algebraic structure is characterized by properties of its operations. Because of its abstract characteristics, algebraic structures are not easy to learn. Therefore, an application in a computer program to help the testing of algebraic structures is worth to be developed.

There are many algebraic structures that are studied in abstract algebra. In group theory, it is also studied about Abelian group, cyclic group, subgroup, normal subgroup, factor group, group homomorphism, etc. In ring theory, there are ring, commutative ring, integral domain, division ring, field, subring, subfield, ideal, ring of quotient, ring homomorphism, etc. [1]. Each structure has its own definition and characterizations and there are also relations among them. A set with respect to some operations might also be more than one structures.

Previously, an application program to test rings and fields has been developed [2]. The application is able to test rings, commutative rings, division rings, fields, subrings, ideals and ring homomorphisms. In that application, the set that wants to be tested is inputted by inputing its elements, and also define the operations on that set. After that, the program will analyze the structure of the inputed set. As an output, it is shown what properties hold in the set, and what kind of algebraic structures is the set.

Based on that application, in this paper, the application is also delevoped by adding more algebraic structures to test. In the developed application, testing for integral domain, finite field and subfield are added. Also, the input is also added by matrix and alphabet input. In the previous work, the input of the set is limited to integers modulo n. By this addition, some various examples of algebraic structures can be tested. Various results of testing can also be obtained in this application program and the difference among some algebraic structures can be shown clearer.

In this section, some basic concepts and theories about abstract algebra will be discussed. Specifically, ring, commutative ring, integral domain, division ring and field will be discussed. A ring is an algebraic structure of a set that consists of two binary operations: addition and multiplication, in which the set is an Abelian group with respect to the addition, a semigroup with respect to the multiplication and the multiplication is distributive to the addition. In other words, a set $R$ with two binary operations: addition ( $+$ ) and multiplication ( $\cdot$ ) is a ring if these following conditions hold for any $a,b,c \in R$:

- $a+b \in R$ and $a \cdot b \in R$ (closed property of addition and multiplication)
- $(a+b)+c = a+(b+c)$ and $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ (associative property of addition and multiplication)
- There exists $0 \in R$ such that $0+a = a+0 = a$, for any $a \in R$ (existence of the identity element of addition).
- There exists $(-a) \in R$ such that $a+(-a) = (-a)+a = 0$ (existence of additional inverse)
- $a+b = b+a$ (commutative property of addition)
- $a \cdot (b+c) = a \cdot b + a \cdot c$ and $(a+b) \cdot c = a \cdot c + b \cdot c$ (distributive property)

Some examples of rings are the set of integers with respect to usual addition and multiplication, the set of all real matrices with respect to matrix addition and multiplication, and the set of integers modulo () with respect to addition and multiplication modulo . By adding some conditions, some special classes of rings can be defined. This will be discussed in the next sub-sections.

## Commutative Rings, Rings with Unity and Integral Domains

A ring is an Abelian (commutative) group with respect to addition, because it satisfies all axioms of addition group and its addition operation is commutative. A ring is not always a multiplicative group. A ring is just a semigroup with respect to multiplication, i.e. it only satisfies closed and associative property. Multiplication in a ring is not always commutative, as multiplication in ring of all real matrices is not commutative. Some rings might have identity element of multiplication (also called by unity), usually denoted by 1, such that $1 \cdot a = a \cdot 1 = a$ , for any , but some others might not. As an example has 1 as its multiplication identity, but the ring of even integers does not have any multiplicative identity. The cancellation law also does not always hold. For example in ring , but . This is possible because a ring might have zero divisors, i.e. a nonzero element that if it is multiplied by another nonzero element, the result will be 0. For example, in , , so that 2 and 3 are both zero divisors. In other hand, in the ring of integer , its multiplication is commutative, it has multiplicative unity and it does not have zero divisor. From this fact, three new classes of rings are defined:

- A ring is said to be a commutative ring if its multiplication is commutative, i.e. $a \cdot b = b \cdot a$, for any $a,b \in R$.
- A ring is said to be a ring with unity if it has identity element of multiplication.
- A ring is said to be an integral domain if it is commutative ring, it has multiplication unity and it has no zero divisors.

Examples of commutative rings and rings with unity are and . The set is an integral domain, but in general is not always integral domain. The set is an integral domain if and only if is a prime number. The set of all real matrices is a ring with unity (the identity matrix), but not a commutative ring.

## Division Rings and Fields

Another axiom of multiplicative group that does not always hold in a ring is that every element has multiplicative inverse. This is because even the multiplicative unity of a ring does not always exists. However, in some ring with unity, it can be found that any nonzero element has multiplicative inverse, as in the ring of all rational numbers $Q$ and ring of all real numbers $R$, and in some others not, as in the ring of all integers $Z$. In other side, existence of multiplicative inverse has no correlation with the commutativity of the multiplication. Therefore it can be defined two more classes of rings:

A ring is said to be a division ring if it is a ring with unity (1) and any its nonzero element has multiplicative inverse, i.e. if $a \in R$ and $a \neq 0$, then it exists $a^{-1} \in R$ such that $a \cdot a^{-1} = a^{-1} \cdot a = 1$.

A ring is said to be a field if it is a division ring and a commutative ring. A finite field is a field that has finitely many elements.

Examples of fields (and also be examples of division rings) is the ring of all rational numbers $Q$ and ring of all real numbers $R$ with respect to usual addition and multiplication. The set of all integers $Z$ is not a division ring, and so not a field. It is a little bit difficult to find an example of a division ring that is not a field, but it will be showed later.

## II. METHOD

In designing the application program, the Waterfall method model is used, as in previous work [3], with the stages as follows [4]: (1) Design of Screen Display*:* there are four screen displays made on the stage is designing the application program of algebraic structure testing. The draft of the screen display design is as follows; (2) Design of Prologue/Opening Screen Display: this is what users see when the program running. The prologue display contains program title, user's identity, supervising lecturer's identity, and Jbutton. Jbutton is useful to close the prologue display and open the main display; (3) Design of Algebraic Structure Testing Display: the display provides users to perform ring, commutative ring, integral domain, division ring and field testing. On the screen, there are three main sub-tabs: Data Input, Analysis of Cayley Table and Results Analysis. Data Input sub-tab is used to input the elements of tested set and fill the Cayley table. Analysis of Cayley Table sub-tab allows the user to see the testing result of the Cayley table. Result Analysis sub-tab shows conclusions of Cayley table testing results, that is what kind of structures is the tested set; and (4) Module Design (Pseudocode): in its development, the application program was built by forming the program modules. The are many modules in this application program. Some of them are shown in this paper.

**Module OperasiTertutup (for checking closed property)**

```
Begin
 Untuk setiap sel, ulangi
 Begin
  Jika ada isi sel yang bukan anggota himpunan
  Begin
     flag = salah
       End
       Selain itu
       Begin
          flag= benar
       End
    End
End
```

**Module OperasiAsosiatif (for checking associative property)**

```
Begin
  Count=0
  Dari i=1, sampai i=jumlah anggota, ulangi
  Begin
   Dari j=1, sampai j=jumlah anggota, ulangi
    Begin
    Dari k=1, sampai k=jumlah anggota, ulangi
     Begin
     temp = anggota baris ke-j, kolom ke-k
     lokasi = posisi kolom temp
 kiri = anggota baris ke-i, kolom ke-lokasi
  temp = anggota baris ke-i, kolom ke-j
     lokasi = posisi baris temp
kanan =anggota baris ke-lokasi, kolom ke-k
                Jika kiri=kanan
                  Begin
                   Count=count+1
                  End
                End
            End
     End
     Jika count = jumlah anggota himpunan
     Begin
          flag=benar
     End
     Selain itu
     Begin
                flag=salah
     End
     End
```

**Modul OperasiKomutatif (for checking commutative property)**

```
Begin
 Count=0
 Dari i=1, sampai i=jumlah anggota, ulangi
     Begin
   Dari j=1, sampai j=jumlah anggota, ulangi
         Begin
      Kiri = anggota baris ke-i, kolom ke-j
     Kanan = anggota baris ke-j, kolom ke-i
             Jika kiri=kanan
             Begin
                   Count = count+1
             End
         End
     End
     Jika count = jumlah anggota himpunan
     Begin
         flag=benar
     End
     Selain itu
     Begin
         flag =salah
     End
End
```

**Module OperasiUnity (for checking existence of unity)**

```
Begin
    flag =salah
  Dari i=1, sampai i=jumlah anggota, ulangi
     Begin
     Jika baris ke-i = baris header dan
     kolom ke-i = kolom header
     Begin
     unkes = anggota himpunan ke-i
             flag = benar
         End
     End
End
```

**Modul OperasiInvers (for checking inverse existence property)**

```
Begin
 Jika operasiUnity = benar
 Begin
        Count=0
    Dari i=1, sampai i=jumlah anggota, ulangi
              Begin
    Dari j=1, sampai j=jumlah anggota, ulangi
                     Begin
  Jika anggota baris ke-i, kolom ke-j = unkes
                            Begin
       Invers anggota ke-i = anggota ke-j
                        Count=count+1
                            End
                     End
                End
        Jika count=jumlah anggota himpunan
             Begin
                   flag=benar
             End
      End
End
```

**Module OperasiPembagiNol (for checking existence of zero divisors)**

```
Begin
      flag=salah
      Begin
   Dari i=1, sampai i=jumlah anggota, ulangi
              Begin
   Dari j=1, sampai j=jumlah anggota, ulangi
                     Begin
 Jika  angg  baris  ke-i,kolom  ke-j=elemen
identitas
                            Begin
                                   flag=benar
                            End
                     End
              End
      End
End
```

**Module OperasiDistributif (for checking distributive property)**

```
Begin
      Count kiri = 0
      Count kanan = 0
  Dari i=1, sampai i=jumlah anggota, ulangi
       Begin
   Dari j=1, sampai j=jumlah anggota, ulangi
             Begin
   Dari k=1, sampai k=jumlah anggota, ulangi
              Begin
   temp = anggota baris ke-j, kolom ke-k
   tabel operasi tambah
                lokasi = posisi kolom temp
  kiri = anggota baris ke-i, kolom ke-lokasi
                   temp1 = anggota baris
      ke-i, kolom ke-j tabel operasi kali
   temp2= anggota baris ke-i, kolom ke-k
   tabel operasi kali
   kanan= anggota baris ke-temp1, kolom ke-
temp2 tabel operasi tambah
                Jika kiri=kanan
                Begin
                Count kiri = count kiri +1
                   End
                 End
         End
  Dari i=1, sampai i=jumlah anggota, ulangi
      Begin
  Dari j=1, sampai j=jumlah anggota, ulangi
          Begin
  Dari k=1, sampai k=jumlah anggota, ulangi
              Begin
temp = anggota baris ke-i, kolom ke-j tabel
operasi tambah
      lokasi = posisi baris temp
  kiri  = anggota baris  ke-temp,  kolom ke-k
temp1 = anggota baris ke-i, kolom ke-j tabel
operasi kali
temp2 = anggota baris ke-i, kolom ke-k tabel
operasi kali
kanan  =  anggota  baris  ke-temp1,  kolom  ke-
temp2 tabel operasi tambah
             Jika kiri=kanan
             Begin
             Count kanan = count kanan +1
             End
               End
          End
      End
      Jika  count  kiri   =  jumlah  anggota
      himpunan dan count kiri = count kanan
      Begin
             distributif  = benar
      End
      Selain itu
      Begin
             distributif  =salah
      End
End
```

**Module ProsesAnalysisCayley (for Cayley table analysis)**

```
Begin
      Jika operasiPembagiNol = benar
      Begin
        Kesimpulan = mempunyai pembagi nol
      End
      Selain itu
      Begin
   Kesimpulan = tidak mempunyai pembagi nol
      End
      Jika   operasiTertutup   =   benar,
      operasiAsosiatif      =      benar,
      operasiKomutatif = benar, operasiUnity
      = benar, operasiInvers = benar dan
      distributif=benar  terhadap  operasi
      (+)
      Begin
             Kesimpulan = RING
             Jika operasiTertutup = benar,
             operasiAsosiatif   =   benar,
             operasiKomutatif   =    benar
             Terhadap operasi (*)
             Kesimpulan = RING KOMUTATIF
      Selain itu,
```

```
      Begin
      Kesimpulan = Bukan RING KOMUTATIF
      End
      Jika    operasiUnity    =    benar,
      operasiInvers  =  benar   Terhadap
      operasi (*)
      Begin
            Kesimpulan=DIVISION RING
      End
      Selain itu,
      Begin
            Kesimpulan  =  Bukan  DIVISION
      RING
      End
      Jika operasiUnity=benar,operasiInvers
      = benar , operasiKomutatif Terhadap
      operasi (*)
      Begin
            Kesimpulan = FIELD
      End
      Selain itu,
      Begin
            Kesimpulan = Bukan FIELD
      End
      End
      Selain itu,
      Begin
            Kesimpulan = bukan RING
      End
      Jika kesimpulan = field
      Begin
            Jika cekPrima=2
            Begin
            Kesimpulan = finite field
      End
End
```

Based on that design method, an application program to test algebraic structures has been developed. We will see the result of this program in the next section.

## III. RESULTS AND DISCUSSION

In this section, we will see some examples of algebraic structure testings using a computer programdescribed in previous section. The program is a development from previous work [3], so that its basic design and specifications are same. The first step in running this program is inputing the set that will be tested together with defining some operations, in this case addition and multiplication, on it. After that, by clicking Jbutton “Analysis”, we will obtain the analysis of Cayley table of our inputed set and the program will show what operation properties are satisfied in our set. Then in sub-tab “Result Analysis”, it is shown what kind of structures is our set.

We will see those processes in three examples: $Z_3$ with respect to addition and multiplication modulo 3, a set of ternary matrices $\left\{\begin{bmatrix}0 & 0\\0 & 0\end{bmatrix},\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix},\begin{bmatrix}2 & 2\\2 & 2\end{bmatrix}\right\}$ with respect to matrix addition and multiplication modulo 2, and a set of alphabet with designed Cayley table.

### Testing $Z^3$

First the user input the set and define addition and multiplication modulo 3:

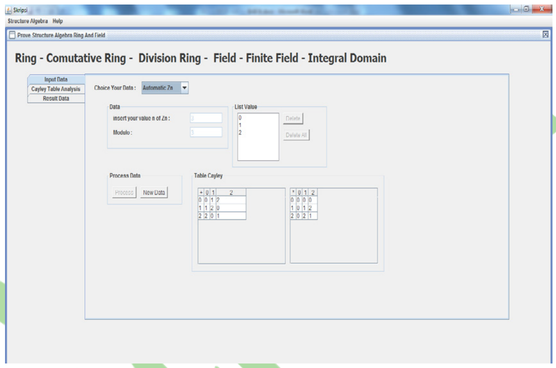


*Figure 1. Input set $Z_3=\{0,1,2\}$ by user.*

Then we execute the Cayley table analysis:

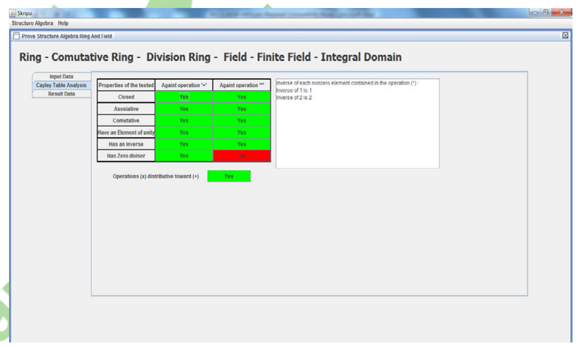


*Figure 2. Cayley table analysis of Z3={0,1,2}*

As the results, the program shows that our inputed set is a ring, a commutative ring, a division ring, a field, a finite field and an integral domain.

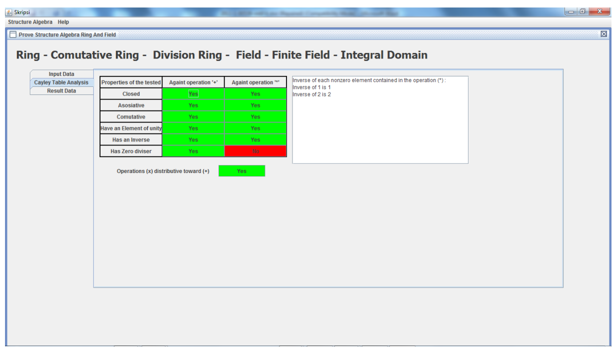


*Figure 3. Conclusion of set of alphabet with designed Cayley table.*

### Testing set of matrices

First the user input the set

$$\left\{\begin{bmatrix}0 & 0\\0 & 0\end{bmatrix},\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix},\begin{bmatrix}2 & 2\\2 & 2\end{bmatrix}\right\}$$

and define matrix addition and multiplication mod 3:

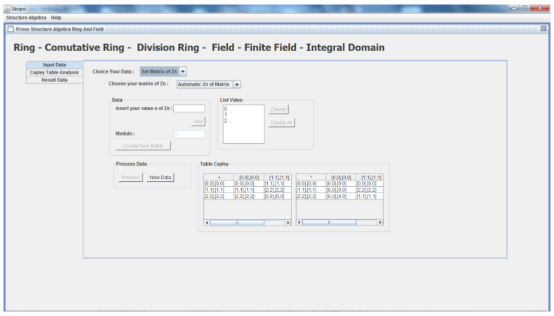


*Figure 4. Input set of matrices by user.*

Then we execute the Cayley table analysis:

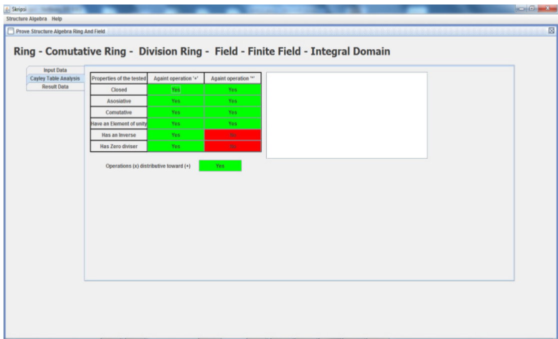


*Figure 5. Cayley table analysis of set of matrices.*

As a result, the program shows that that set of matrices is a ring and a commutative ring, but not a division ring, not a field, not a finite field and not an integral domain.

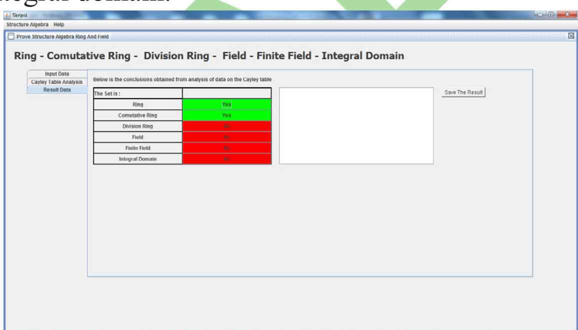


*Figure 6. Conclusion of set of matrices.*

**Testing set of alphabet with designed Cayley table**

In this case, we see the testing of a set of alphabets with designed Cayley table. Cayley tables of addition and multiplication are given below:

| + | x | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|
| x | x | a | b | c | d | e | f |
| a | a | b | c | d | e | f | x |
| b | b | c | d | e | f | x | a |
| c | c | d | e | f | x | a | b |
| d | d | e | f | x | a | b | c |
| e | e | f | x | a | b | c | d |
| f | f | x | a | b | c | d | e |

| + | x | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|
| x | x | x | x | x | x | x | x |
| a | x | a | b | c | d | e | f |
| b | x | b | c | a | e | f | d |
| c | x | c | a | b | f | d | e |
| d | x | d | f | e | a | c | b |
| e | x | e | d | f | b | a | c |
| f | x | f | e | d | c | b | a |

The user input the set of alphabet and the designed Cayley table:

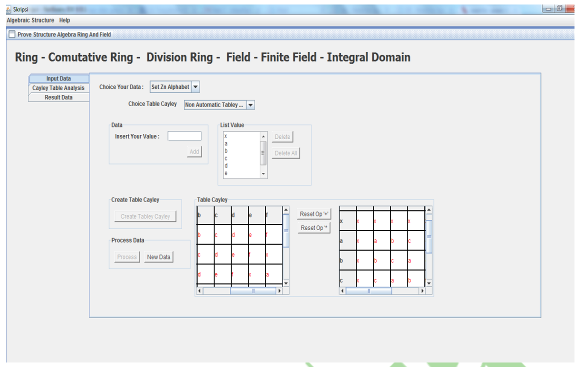


*Figure 7. Input set of alphabet with designed Cayley table.*

Then we execute the Cayley table analysis:

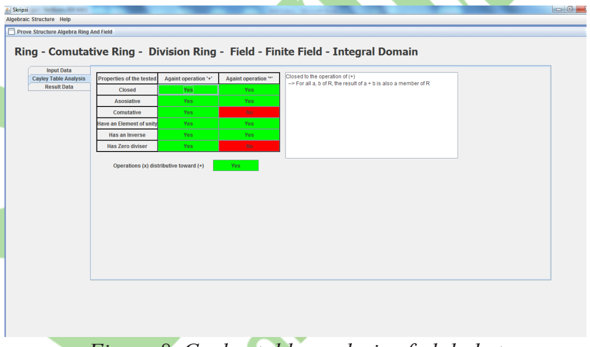


*Figure 8. Cayley table analysis of alphabet with designed Cayley table.*

As a result, the program shows that the inputted set is a ring and a division ring, but not a commutative ring, not a field, not a finite field and not an integral domain.

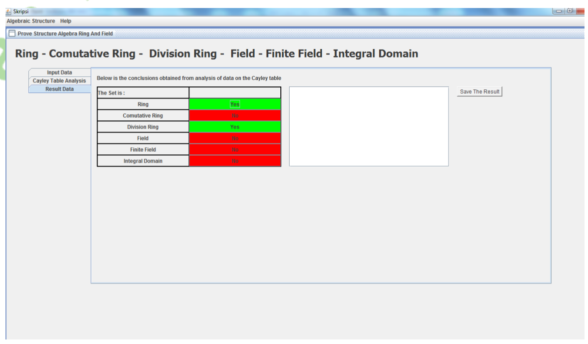


*Figure 9. Conclusion of set of alphabet with designed Cayley table.*

Why the set need to be tested? Because by testing , it will be found out that its addition and multiplication are always commutative. Thus any tested set will result in commutative ring. To give an example of a set that is a ring but not commutative, different kind of input has to be given. Therefore, matrix multiplication has to be considered, not the commutative. However, to create a non-commutative division ring, it is difficult if a set of matrices are used because not all nonzero matrix has multiplicative inverse. So, to provide a non-commutative division ring, the addition and multiplication have to be defined by defining their Cayley tables manually.

## IV. CONCLUSION

From previous section, it can be seen that the application program is worked properly for some various examples. It gave same results as manual testing based on modern algebra concepts. In checking associative and commutative properties, it spend less time than manual testing, especially for testing set with many elements. Comparing with previous program, our developed program has some advantages such as: (1) There are some new algebraic structure added, such as integral domain and division ring. (2) Input of the application program are added by matrix and alphabet input, so it has more variation of set to be tested. (3) By more various input set, we can test some non-commutative structures, which are not able to be tested if the input is limited to integer modulo n. However the application program also has disadvantages such as the input set is still limited to finite set. There are some structures of infinite set with well-defined operations and this program cannot test them. For example, the set of integers $Z$ with usual addition and multiplication, is an integral domain but not a field. This program cannot test it.

## V. REFERENCES


[1]. L. Gilbert & J. Gilbert, (2009). *Elements of Modern Algebra*. Belmont: Brooks/Cole.

[2]. D. Arifin. (2011). Perancangan Pengembangan Program Aplikasi Pengujian Struktur Aljabar Ring, Ring Komutatif, Field, Sub Ring, Ideal. Bina Nusantara University, Jakarta.

[3]. N.I. Manik. et al. (2014). Testing of Rings and Fields Using a Computer Program.*Journalof Software, 9 (5),*1141-1150.

[4]. R. S. Pressman. (2010). *Software Engineering: A Practitioners Approach*. New York: McGraw.

[5]. J. B. Fraleigh (2003). *A First Course in Abstract Algebra*. Boston: Addison-Wesley.