Testing of Rings and Fields Using A Computer Program

Ngarap Im. Manik [manik@binus.edu] 1), Fransisca Fortuanatadewi 2), Don Tasman 3)
1, 2, 3 Mathematics Department – Faculty of Computer Science, Bina Nusantara University
Jl.Kebon Jeruk Raya 27 Jakarta 11530, Indonesia

***Abstract*—Abstract algebra is the study of algebraic structures. Some branches of algebraic structures, such as groups, rings, fields are difficult to learn and less desirable due to the abstract characteristic. To help facilitate the learning process of algebraic structures so that it becomes more attractive, an application of a computer program is developed to help the testing of algebraic structures. By this application, testing of algebraic structures is expected to be easier, faster, and more accurate than manual testing. The application uses Cayley table as a bridge between users and the program. The application limited to testing of algebraic structures of rings, commutative rings, division rings, fields, sub-rings, ideals, homomorphisms, epimorphisms, monomorphisms, and isomorphisms by using Java, an open-source based programming language. Test results of the application program for the topic showed correct results with a relatively short processing time compared to manual testing.**

***Index Terms*—ring, field, ideal, homomorphism, abstract algebra , Cayley**

I. INTRODUCTION

Mathematics is a broad field of study, which studies properties and interactions between ideal objects. Several field studies that have been widely known are logic, calculus, algebra, optimization, probability, and statistics [4]. Algebra as one of the central branches of mathematics studies the rules of operations and relations on sets, as well as the possibility of formations and concepts that emerge from these rules. While the abstract algebra, also called modern algebra, is one branch of algebra which specifically studies algebraic structures, such as groups, rings, and fields [1,7].

Because of the abstract characteristic, algebraic structures are not easy to learn so that they are less desirable. Regarding this condition, an application of a computer program that can help the testing of algebraic structures is developed. With the help of a computer program, studying algebraic structures could be easier. Hence, common people will be interested in learning algebraic structures because the calculation process can be made easier, faster, and more accurate than manual testing [8]. Given the scope of algebraic structures is very large, this paper tested the scope of problems which only covers limited algebraic structures, including rings, division rings (sub-ring, commutative ring, division ring, ring homomorphism, ring epimorphism, embedding ring/ring monomorphism, ring isomorphism), and fields [9].

The purpose of this design is the production of an application program that can perform algebraic structure testing along with the details of the results obtained with the set of members in the form of character input. The program is expected to be a tool in the algebraic structure testing of rings, division rings (sub-ring, commutative ring, division ring, ring homomorphism, ring epimorphism, embedding ring/ring monomorphism, ring isomorphism), and fields. Thus, it can simplify, accelerate, and improve the thoroughness in algebraic structure testing. Moreover, the application program can be a reference for further research and can be developed.

A. Rings

A ring is an algebraic structure consisting of two binary operations: addition and multiplication, in which the structure is an Abelian group to the addition, a semigroup to the multiplication; and the multiplication is distributive to the addition. A ring (R, +, ×) is a nonempty set R with the binary operations of addition (+) and multiplication (×) on R which fulfills the following axioms.

a. To the addition (+)
 Closed: for every a,b ε R, then a + b ε R.
 Associative: for every a,b,c ε R, then (a + b) + c = a + (b + c).
 Has an element of unity: the existence of the identity element α, that a + α = α + a = a.
 Has inverse: for every a ε R, found b, that a + b = b + a = α.
 Commutative: for every a,b ε R, then a + b = b + a.

b. To the multiplication (×)
 Closed: for every a,b ε R, then a × b ε R.
 Associative: for every a,b,c ε R, then (a × b) × c = a × (b × c).

c. Distributive of the multiplication (×) to the addition (+)
 For every a,b,c ε R, if it fulfills:
 Left Distributive: for every a,b,c ε R qualifies a × (b + c) = (a × b) + (a × c),
 Right Distributive: for every a,b,c ε R qualifies (a + b) × c = (a × c) + (b × c),
 then R is distributive multiplication to the addition [6].

doi:10.4304/jsw.9.5.1141-1150

B. Commutative Rings

A commutative ring is a ring, in which the structure is an Abelian group to the addition, a commutative semigroup to the multiplication; and the multiplication is distributive to the addition. A commutative ring (R, +, ×) is a nonempty set R with the binary operations of addition (+) and multiplication (×) on R which fulfills the following axioms.

a. To the addition (+)
   Closed: for every a, b ε R, then a + b ε R.
   Associative: for every a, b, c ε R, then (a + b) + c = a + (b + c).
   Has an element of unity: the existence of the identity element α, that a + α = α + a = a.
   Has inverse: for every a ε R found b, that a + b = b + a = α.
   Commutative: for every a, b ε R, then a + b = b + a.
b. To the multiplication (×)
   Closed: for every a,b ε R, then a × b ε R.
   Associative: for every a,b,c ε R, then (a × b) × c = a × (b × c).
   Has an element of unity: found identity element β, that a × β = β × a = a.
   Commutative: for every a,b ε R, then a × b = b × a.
c. Distributive of the multiplication (×) to the addition (+)
   To every a, b, c ε R, if it fulfills:
   Left Distributive: for every a, b, c ε R qualifies a × (b + c) = (a × b) + (a × c),
   Right Distributive: for every a,b,c ε R qualifies (a + b) × c = (a × c) + (b × c), then R is distributive multiplication to the addition [6].

C. Fields

A field is an algebraic structure consisting of two binary operations: addition and multiplication, in which the set of the addition the structure is an Abelian group, the set of non-zero to the multiplication operation is an Abelian group, and the multiplication operation is distributive to the addition. A field (R, +, ×) is a nonempty set R with the binary operations of addition (+) and multiplication (×) on R which fulfills the following axioms.

a. R to the addition (+)
   Closed: for every a,b ε R, then a + b ε R.
   Associative: for every a,b,c ε R, then (a + b) + c = a + (b + c).
   Has an element of unity: the existence of the identity element α, that a + α = α + a = a.
   Has inverse: for every a ε R found b, that a + b = b + a = α.
   Commutative: for every a,b ε R, then a + b = b + a.
b. R of non-zero to the multiplication (×)
   Closed: for every a,b ε R, then a + b ε R.
   Associative: for every a,b,c ε R, then (a × b) × c = a × (b × c).
   Has an element of unity: the existence of the identity element β, that a × β = β × a = a.
   Has inverse: for every a ε R-{0} found b, that a × b = b × a = β.
   Commutative: for every a,b ε R, then a × b = b × a.
c. Distributive of the multiplication (×) to the addition (+).
   For every a, b, c ε R, if it fulfills:
   Left Distributive: for every a,b,c ε R qualifies a × (b + c) = (a × b) + (a × c),
   Right Distributive: for every a,b,c ε R qualifies (a + b) × c = (a × c) + (b × c), then R is distributive multiplication to the addition [5].

D. Sub-rings

If (R, +, ×) is a ring, A is a non-zero set which is part of R (A ⊂ R). With the same operation as R, (A, +, ×) forms a ring; the set of A is called a sub-ring of the set R [5].

E. Ideal Sub-rings

Ideal is a sub-ring which has a special characteristic, that is, closed to the multiplication element outside the sub-ring. A sub-ring is ideal if it is a left ideal (closed to the left multiplication element) and a right ideal (closed to the right multiplication element) [2].

F. Division Rings

A division ring is a ring in which the non-zero elements form a group under the operation of multiplication (x) [4].

G. Ring Homomorphisms

If (R,+,×) and (S,(+),(×)) are rings, then the mapping function f:R → S is called a homomorphism if:

a. f(a+b) = f(a) (+) f(b) for every a, b ε R,
b. f(a+b) = f(a) (+) f(b) for every a, b ε R,
c. f(unkes x) = unkes (x) [3].

H. Ring Epimorphisms

If (R,+,×) and (S,(+),(×)) are rings, then the mapping function f:R → S is called a monomorphism if the mapping is a homomorphism mapping and onto (surjective) [3].

I. Ring Epimorphisms (Embedding Rings)

If (R,+,×) and (S,(+),(×)) are rings, then the mapping function f:R → S is called a monomorphism if the mapping is a homomorphism mapping and injective (1-1) [11].

J. Ring Isomorphisms

If (R,+,×) and (S,(+),(×)) are rings, then the mapping function f:R → S is called a monomorphism if the mapping is a homomorphism mapping and injective (1-1) and onto (surjective) [1].

K. Cayley Table

Cayley table is a list made to show the operation between two elements on limited sets. The Cayley table is as Table 1 [2].

TABLE 1.
CAYLEY TABLE OF THE ADDITION OF MODULO 5

$+_5$	**0**	**1**	**2**	**3**	**4**
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

II. METHODS

The process of designing the application program used the Waterfall method model, with the stages as follows [10].

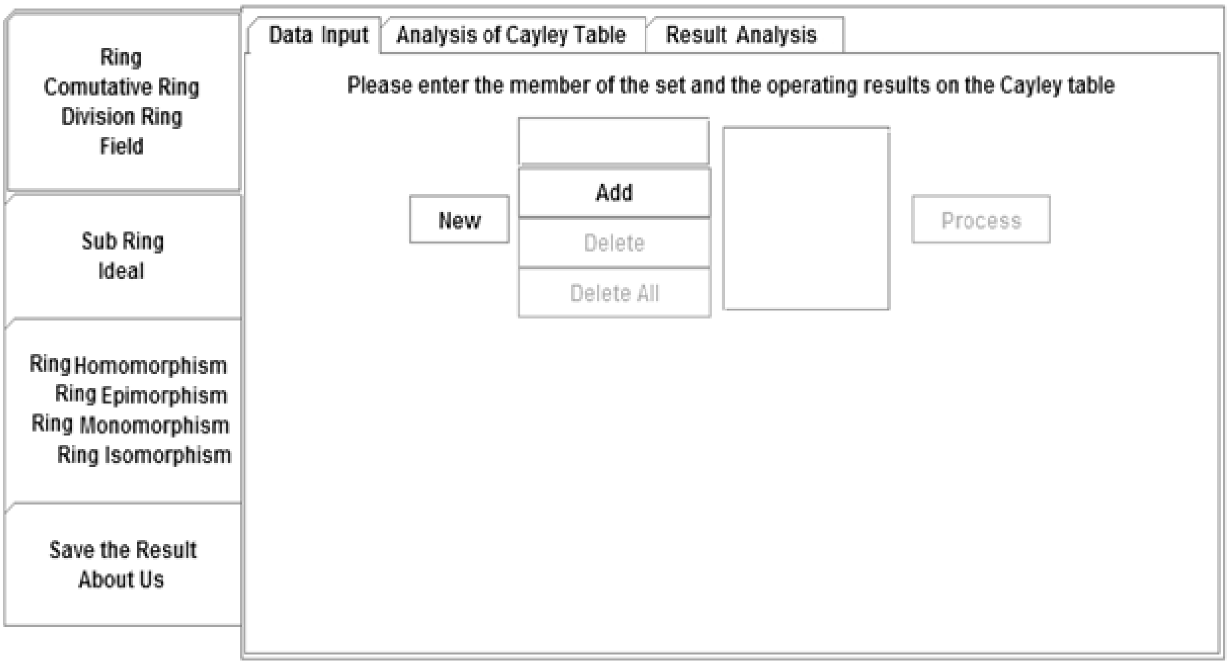

Figure 1. Design of Data Input Sub-tab Display

A. Design of Screen Display

There are four screen displays made on the stage in designing the application program of algebraic structure testing. The draft of the screen display design is as follows.

B. 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.

C. Design of Ring, Commutative Ring, Division Ring, and Field Testing Screen Display

The display provides users to perform ring, commutative ring, division ring, and field testing. On screen, there are three main sub-tabs. Input Data sub-tab allows the user to input member elements and to fill the Cayley table; Analysis of Cayley Table sub-tab allows the user to see the testing results of the Cayley table; and Result Analysis sub-tab shows conclusions of the Cayley table testing results (as shown in figure 1).

D. Design of Sub-ring and Ideal Testing Screen Display

The display provides users to perform sub-ring and ideal testing. There are four main sub-tabs on the screen. Input the Elements sub-tab allows the user to enter member elements of two algebraic structures to test; Fill in the Cayley Table sub-tab lets the user input the Cayley table content for both algebraic structures; Sub-Ring Testing Results sub-tab shows the testing results of the Cayley table along with final conclusions about the relationship between the two algebraic structures. Ideal Testing sub-tab allows the user to fill out the Cayley table for Ideal testing (as shown in figure 2).

E. Design of Ring Homomorphism Testing Screen Display

The display provides users to perform ring homomorphism, ring epimorphism, ring monomorphism, and ring isomorphism testing. On screen, there are three main sub-tabs. Input the Elements sub-tab allows the user to enter the member elements of two algebraic structures to test; Fill In the Cayley Table sub-tab lets the user fill the content of the Cayley table for both algebraic structures; and Testing Result sub-tab shows the testing results of the Cayley table along with final conclusions about the relationship between the two algebraic structures (as shown in figure 3 and figure 4).

F. Module Design (Pseudocode)

In its development, the application program was built by forming the program modules. The modules in this application program are 16 modules. Some of the modules are shown in this paper.

```
Define Public
Var counter1: integer  ; Var counter2: integer  ; Var counter3: integer  ; Var Count: integer  ;Var members: integer
Var temp(): integer  ; Var location(): integer ; Var left(): integer  ; Var right(): integer ; Var Sum(): integer
Var Mul(): integer ; Var Syarat(): integer  ; Var Kesimpulan: String
CekAsosiatifTabelOperasiTambah Module
Begin
        Count=0; counter1=1; counter2=1; counter3=1;
        For Loop counter1 to members
        Begin
                For Loop counter2 to members
                Begin
                For Loop counter3 to members
                  Begin
                  temp = j_row, k_column  ; location = temp_column ; left = i_row, location_column
                  temp = i_row, j_row ; location = temp_row  ; right = location_row, k_column
                            If left = right
                            Begin
                                    Count+=1
                            End
                    End
            End
End
        If Count = = members
        Begin
             Sum (2) = TRUE
        End
        Else
        Begin
           Sum (2) = FALSE
        End
End
```

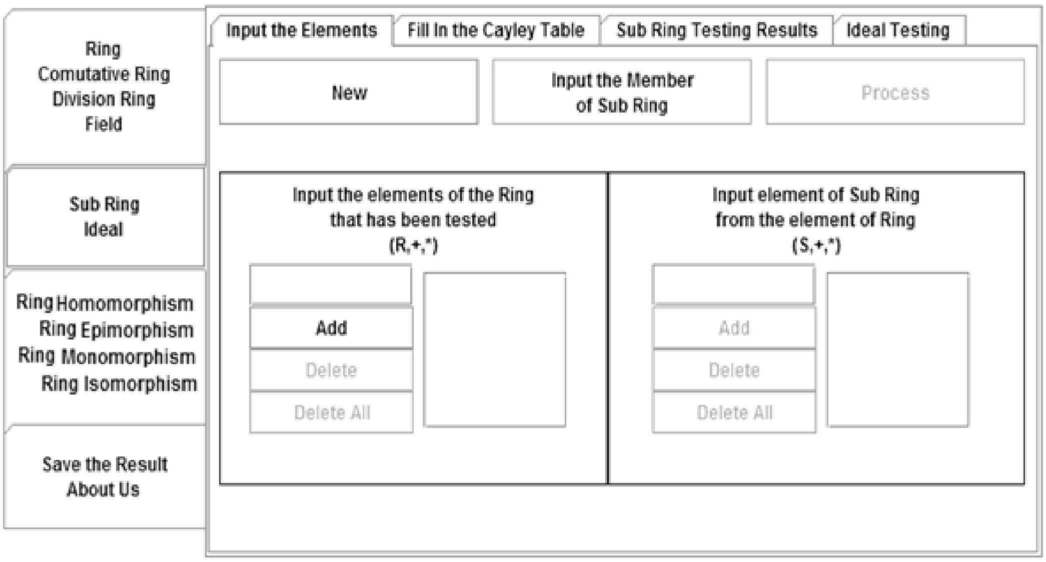

Figure 2. Design of Input the Elements Sub-tab Display

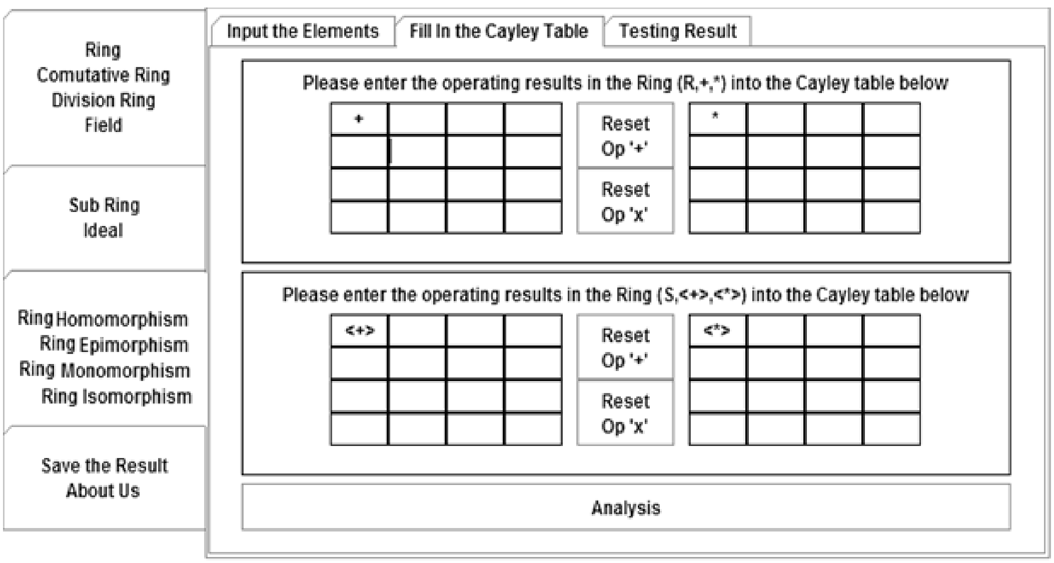

Figure 3. Design of Fill in the Cayley Table Sub-tab Display

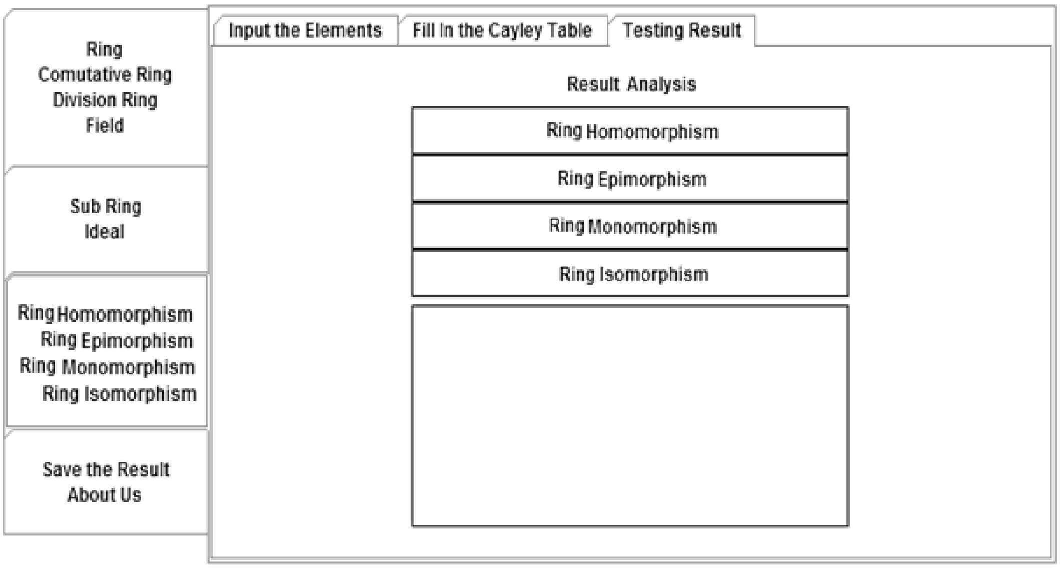

Figure 4. Design of Testing Result Sub-tab Display

CekKomutatifTabelOperasiTambah Module

```
Begin
        Count=0
        For Loop counter1 to members
        Begin
        For Loop counter2 to members
            Begin
                left = i_row, j_column
                right = j_row, i_column
                  If left = right
                  Begin
                          Count +=1
                  End
End
        End
        If Count == members
        Begin
            Sum(3) = TRUE
        End
        Else
        Begin
            Sum (3) = FALSE
        End
End
```

CekRing Module

```
Begin
        If Sum(1, 2, 3, 4, 5) = Mul(1,2) = Distributive =
TRUE
        Begin
                Kesimpulan = RING
                If Mul(3) = TRUE  ;
                Begin
                Kesimpulan = "RING KOMUTATIF"
                End
                Else
                Begin
                Kesimpulan = "Is Not RING
KOMUTATIF"
                End
                If Mul(4, 5) = TRUE
                Begin
                Kesimpulan = "DIVISION RING"
                End
                Else
                Begin
                Kesimpulan = "Is Not DIVISION RING"
                End
                If Mul(3,4,5) = TRUE
                Begin
                Kesimpulan = "FIELD"
                End
                Else
                Begin
                Kesimpulan = "Is Not FIELD"
                End
        End
        Else    Begin
                Kesimpulan = "Is Not RING"
                End
End
```

CekHomomorfis module

```
Begin
        If  homomorfis
        Begin
            Syarat(1) = TRUE
            If surjektif            Begin
            Syarat(2) = TRUE
                                    End
            Else
            Begin
            Syarat(2) = FALSE
            End
            If injektif
            Begin
            Syarat(3) = TRUE
            End
```

```
            Else
            Begin
            Syarat(3) = FALSE
            End
        End
        Else
        Begin
        Syarat(1) = salah
        End
        If syarat(1)=TRUE Begin
        Kesimpulan = "HOMOMORFISMA RING"
        End
        If syarat(1, 2)=TRUE
        Begin
        Kesimpulan = "EPIMORFISMA RING"
        End
        If syarat(1, 3)=TRUE
        Begin
        Kesimpulan = "MONOMORFISMA RING"
        End
        If syarat(1, 2, 3)=TRUE
        Begin
        Kesimpulan = "ISOMORFISMA RING"
        End
End
```

III. Results and discussion

In order the program developed can be used properly, there are some specifications must be fulfilled. The hardware specifications are Intel Pentium Dual-Core CPU T4400 @2.20GHz Processor, 953 MB DDR RAM, 160 GB Hard disk, and Logitech Mouse. While the softwares are Microsoft Windows XP Professional Service Pack 3 operating system and Java Library by installing the Java ™ Standard Edition Development Kit 6 Update 2. For the programming design, writers used Eclipse SDK version 3.7.1 for logic module design and program interface. Then, to run the program, click UjiSA.jar file and select OK. After selecting JButton OK, the screen will show the main menu. Users have four tab options on the left menu. Each of tab menus has three to four sub-tabs, each sub-tab contains a display interface which has functions for each.

On screen (Figure 5), there is JTextField which can be used to input the elements of algebraic structures to test. JButton "Add" is to add the elements in JTextField into JList; JButton "Delete" is to remove the elements; JButton "Delete All" is to empty JList; JButton "new" is to provide a new form for the testing process; and JButton "process" signifies if the user has finished entering the elements of algebraic structures and is ready to fill the Cayley table. Once the user has entered at least 2 elements with the number of elements in a set, she/he can input the content of the Cayley table (as shown in figure 6).

Besides testing results, conclusions of the testing can also be seen on " *Result Analysis*" (see figure 7).

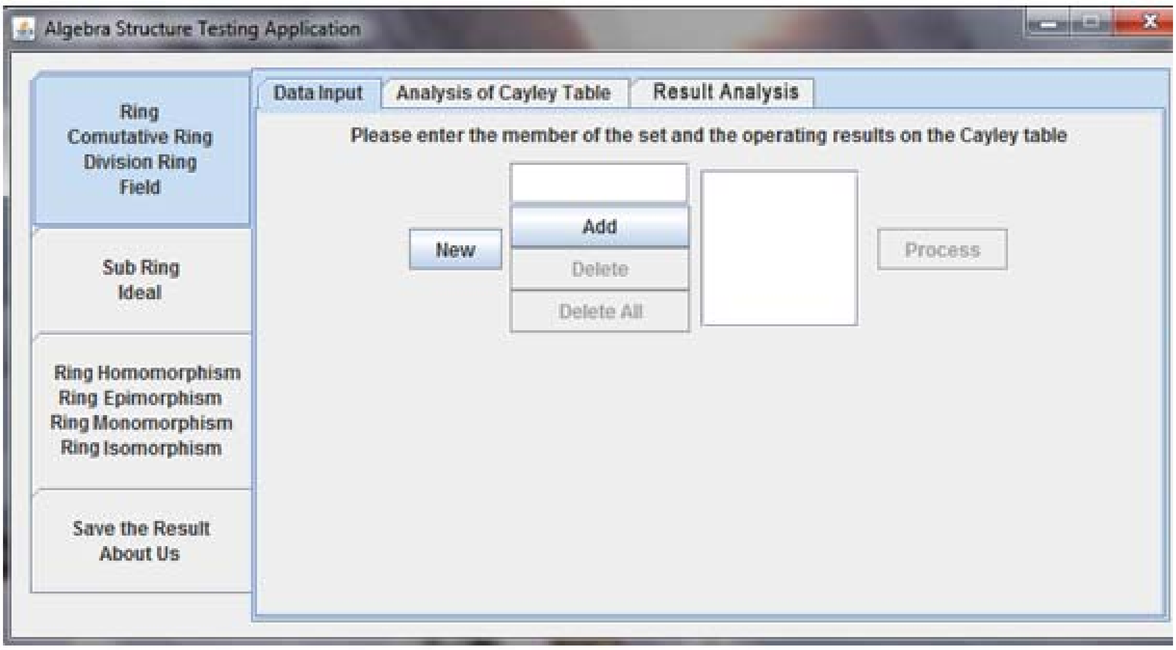

Figure 5. Display Menu of Ring, Commutative Ring, Division Ring, and Field testing – Data Input Tab

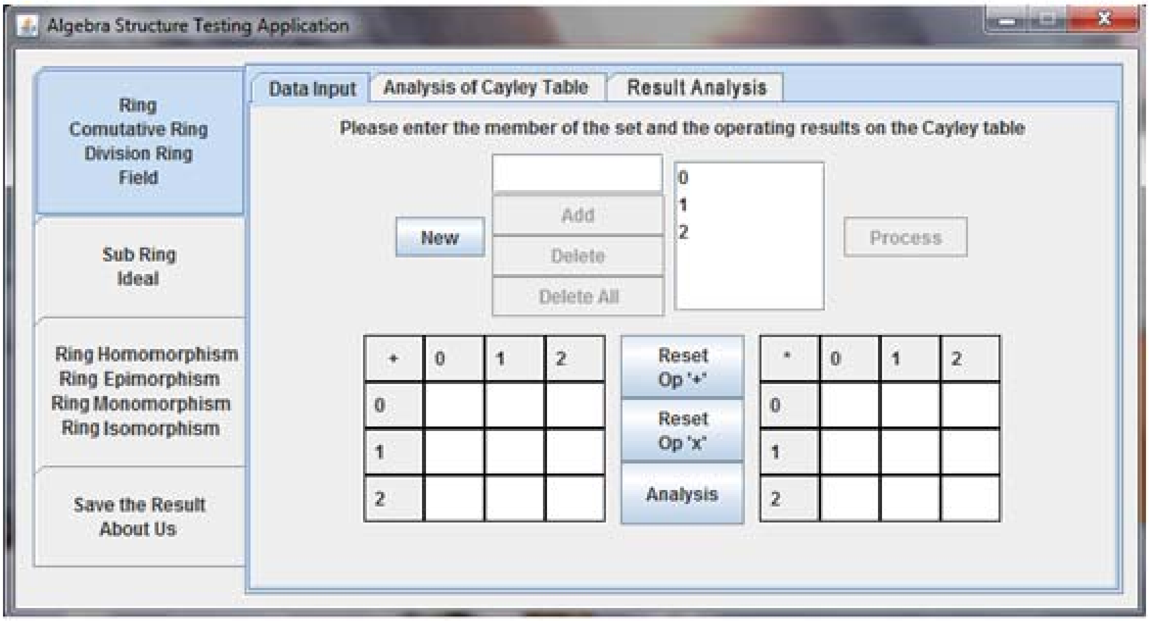

Figure 6. Display of Cayley Table Data Input

Once the user has finished filling the content of the Cayley table, she/he can press JButton "Analysis". Result analysis from the Cayley table contains in "Testing Result" sub-tab. As the two previous tabs, in this tab the user can also access the information of each conclusion by pressing corresponding buttons (see figure 8).

For users who want to print the test results, JButton is available to access the storage of the test results in .txt file. They can be printed via Notepad application program. JButton is on the fourth tab: Save the Result-About Us.

To ensure the program's ability in performing the testing, it is necessary to compare the manual results and the program results. One testing using the Cayley table of Rings, Commutative Rings, Division Rings, Fields (Addition of Modulo 4) was done as shown in Table 2 and Table 3.

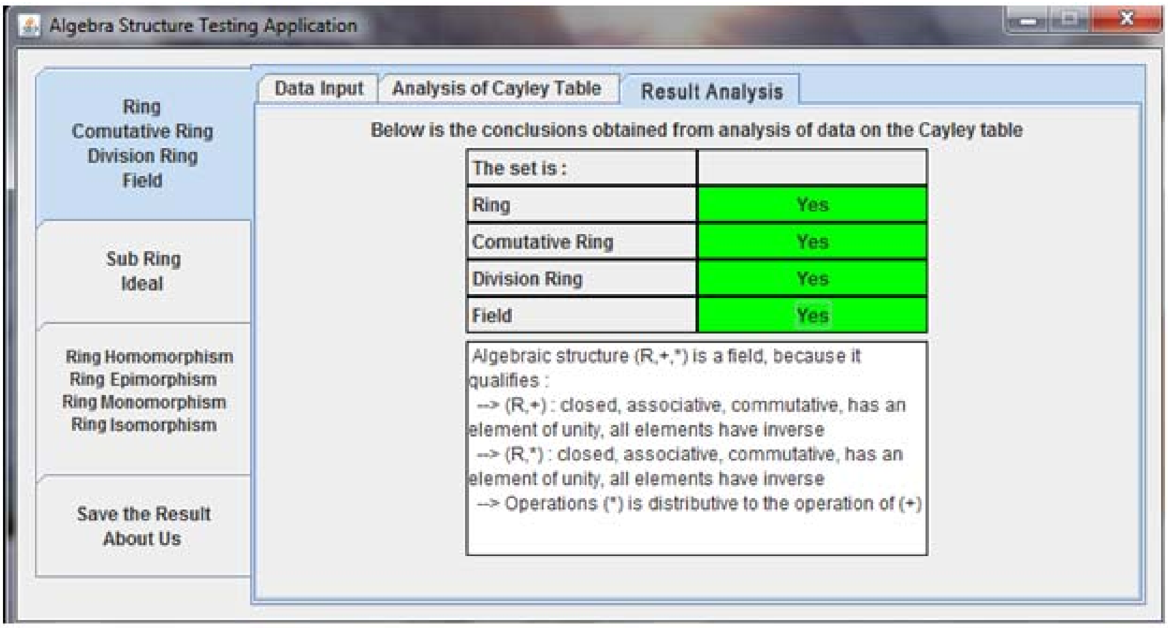

Figure 7. Display of Result Analysis Sub-tab

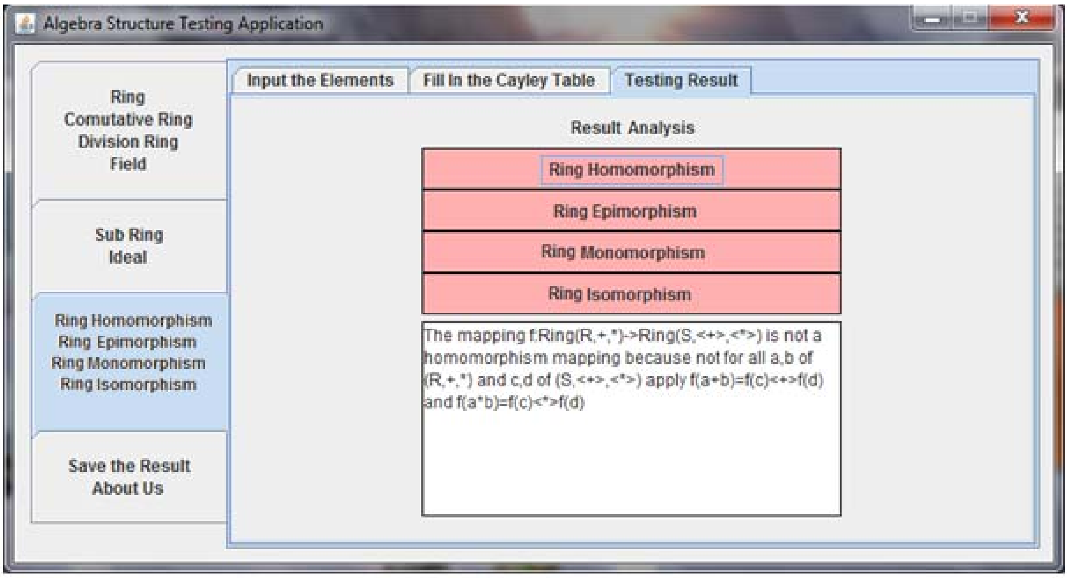

Figure 8. Display of Ring Homomorphism tab on Testing Result sub-tab

TABLE 2.
TESTING OF RINGS, COMMUTATIVE RINGS, DIVISION RINGS, FIELDS (ADDITION OF MODULO 4)

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

TABLE 3.
TESTING OF RINGS, COMMUTATIVE RINGS, DIVISION RINGS, FIELDS (MULTIPLICATION OF MODULO 4)

*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Conclusions derived from manual and program testing give the same result. Table 2 and Table 3 are rings, commutative rings, not division rings and fields. Here is the analysis and the program's testing results.

Result Analysis from Cayley Table (Table 2 & Table 3)

(p1) *Closed to the operation of (+)*
→ For all a, b of R, the result of a + b is also a member of R

The testing result above (p1) stated that based on the Cayley table of the operation of (+) and the pseudocode, the structure qualifies the closed characteristic to the operation of (+).

(p2) *Elements that generate the left and right sides together :*
0+(0+0) = 0 ← is equal to → 0 = (0+0)+0
0+(0+1) = 1 ← is equal to → 1 = (0+0)+1
0+(0+2) = 2 ← is equal to → 2 = (0+0)+2
0+(0+3) = 3 ← is equal to → 3 = (0+0)+3
0+(1+0) = 1 ← is equal to → 1 = (0+1)+0
0+(1+1) = 2 ← is equal to → 2 = (0+1)+1
0+(1+2) = 3 ← is equal to → 3 = (0+1)+2
0+(1+3) = 0 ← is equal to → 0 = (0+1)+3
0+(2+0) = 2 ← is equal to → 2 = (0+2)+0
0+(2+1) = 3 ← is equal to → 3 = (0+2)+1
0+(2+2) = 0 ← is equal to → 0 = (0+2)+2
0+(2+3) = 1 ← is equal to → 1 = (0+2)+3
0+(3+0) = 3 ← is equal to → 3 = (0+3)+0
0+(3+1) = 0 ← is equal to → 0 = (0+3)+1
0+(3+2) = 1 ← is equal to → 1 = (0+3)+2
0+(3+3) = 2 ← is equal to → 2 = (0+3)+3

Associative to the operation of (+)
→ For all a, b, c of R, apply a + (b + c) = (a + b) + c

The testing result above (p2) stated that based on the Cayley table of the operation of (+), the elements qualify the associative characteristic to the operation of (+).

(p3) *Commutative to the operation of (+)*
→ For all a, b of R, apply a + b = b + a

The testing result above (p3) stated that based on the Cayley table of the operation of (+), the elements qualify the commutative characteristic to the operation of (+).

(p4) *Having an element of unity for the operation (+), that is 0*

The testing result above (p4) stated that based on the Cayley table of the operation of (+), the elements have an element of unity to the operation of (+).

(p5) *Inverse of each element contained in the operation of (+)*
Inverse of 0 is 0; Inverse of 1 is 3; Inverse of 2 is 2; Inverse of 3 is 1

The testing result above (p5) stated that based on the Cayley table of the operation of (+), every element has inverse to the operation of (+).

(p6) *Closed to the operation of (*)*
→ For all a, b of R, the result of a * b is also a member of R

The testing result above (p6) stated that based on the Cayley table of the operation of (*), the elements qualify the closed characteristic to the operation of (*).

(p7) *Elements that generate the left and right sides together:*

0*(0*0) = 0 ← is equal to → 0 = (0*0)*0
0*(0*1) = 0 ← is equal to → 0 = (0*0)*1
0*(0*2) = 0 ← is equal to → 0 = (0*0)*2
0*(0*3) = 0 ← is equal to → 0 = (0*0)*3
0*(1*0) = 0 ← is equal to → 0 = (0*1)*0
0*(1*1) = 0 ← is equal to → 0 = (0*1)*1
0*(1*2) = 0 ← is equal to → 0 = (0*1)*2
0*(1*3) = 0 ← is equal to → 0 = (0*1)*3
0*(2*0) = 0 ← is equal to → 0 = (0*2)*0
0*(2*1) = 0 ← is equal to → 0 = (0*2)*1
0*(2*2) = 0 ← is equal to → 0 = (0*2)*2
0*(2*3) = 0 ← is equal to → 0 = (0*2)*3
0*(3*0) = 0 ← is equal to → 0 = (0*3)*0
0*(3*1) = 0 ← is equal to → 0 = (0*3)*1
0*(3*2) = 0 ← is equal to → 0 = (0*3)*2
0*(3*3) = 0 ← is equal to →0 = (0*3)*3

Associative to the operation of (*)
→ For all a, b, c of R, apply a * (b * c) = (a * b) * c

The testing result above (p7) stated that based on the Cayley table of the operation of (*), the elements qualify the associative characteristic to the operation of (*).

(p8) *Commutative to the operation of (*)*
→ For all a, b of R, apply a * b = b * a

The testing result above (p8) stated that based on the Cayley table of the operation of (*), the elements qualify the commutative characteristic to the operation of (*).

(p9) *Having an element of unity for the operation (*), that is 1*

The testing result above (p9) stated that based on the Cayley table of the operation of (*), the elements have an element of unity to the operation of (*).

(p10) *Inverse of each non-zero element contained in the operation of (*):*
Inverse of 1 is 1; Element 2 has no inverse ; Inverse of 3 is 3

The testing result above (p10) stated that based on the Cayley table of the operation of (*), not every element has inverse to the operation of (*). Element 2 has no inverse because there is no column listing the elements of unity of the operation of (*), i.e. 1 in row 2.

(p11) *Checking the left distributive (ld):*

0*(0+0) = 0 ← is equal to → 0 = (0*0)+(0*0)
0*(0+1) = 0 ← is equal to → 0 = (0*0)+(0*1)
0*(0+2) = 0 ← is equal to → 0 = (0*0)+(0*2)
0*(0+3) = 0 ← is equal to → 0 = (0*0)+(0*3)
0*(1+0) = 0 ← is equal to → 0 = (0*1)+(0*0)
0*(1+1) = 0 ← is equal to → 0 = (0*1)+(0*1)
0*(1+2) = 0 ← is equal to → 0 = (0*1)+(0*2)
0*(1+3) = 0 ← is equal to → 0 = (0*1)+(0*3)

→ Left Distributive (ld) Fulfilled

Checking the right distributive (rd):

(0+0)*0 = 0 ← is equal to → 0 = (0*0)+(0*0)
(0+0)*1 = 0 ← is equal to → 0 = (0*1)+(0*1)
(0+0)*2 = 0 ← is equal to → 0 = (0*2)+(0*2)
(0+0)*3 = 0 ← is equal to → 0 = (0*3)+(0*3)
(0+1)*0 = 0 ← is equal to → 0 = (0*0)+(1*0)
(0+1)*1 = 1 ← is equal to → 1 = (0*1)+(1*1)
(0+1)*2 = 2 ← is equal to → 2 = (0*2)+(1*2)
(0+1)*3 = 3 ← is equal to → 3 = (0*3)+(1*3)

→ Right Distributive (rd) Fulfilled

All the elements satisfy the distributive properties of the operations of (*) on the operations of (+) as the fulfillment of left distributive and right distributive.
The testing result above (ld & rd) stated that based on the Cayley table of the operation of (+) and the Cayley table of the operation of (*), the elements qualify the distributive characteristic to the operation of (+).

Conclusion of Result Analysis from Cayley Table (table 2 & table 3)

With members : 0, 1, 2, 3
Algebraic structure (R,+,*) is a ring, because it qualifies :
→ (R,+) : closed, associative, commutative, has an element of unity, all elements have inverse
→ (R,*) : closed, associative
→ Operations (*) is distributive to the operation of (+)

Algebraic structure (R,+,*) is a commutative ring, because it qualifies:
→ (R,+) : closed, associative, commutative, has an element of unity, all elements have inverse
→ (R,*) : closed, associative, commutative
→ Operations (*) is distributive to the operation of (+)

Algebraic structure (R,+,*) is not a division ring, because not every element has inverse in the operation (*).

Algebraic structure (R,+,*) is not a field, because not every element has inverse in the operation of (*).

IV. Conclusion

The application program for ring testing operated properly. It gave results as same as manual testing, but in a shorter time and at higher accuracy as it was done by the computer. The application program of ring testing can be used as a testing tool that makes the testing more effective and efficient. The accuracy of the testing results depends on the thoroughness of data content inputting in the Cayley table. If the user is not careful in the data entry, the testing results are certainly inaccurate.

Reference

[1] Marlow, A First Course in Abstract Algebra, Ring, Groups and Fields, CRC Press, LLC. Taylor & Francis Group, 2005.
[2] D. Arifin, "Perancangan Pengembangan Program Aplikasi Pengujian Struktur Aljabar Ring, Ring Komutatif, Field, Sub Ring, Ideal," Bina Nusantara University, Jakarta, 2011.
[3] D. S. Malik, "Introduction to Abstract Algebra," 2007. [Online]. Available: https://people.creighton.edu/~dsm33733/MTH581/Introduction %20to%20Abstract%20Algebra.pdf. [Accessed 1 September 2012].
[4] E. W. Weisstein, "Division Algebra," 1 September 2012. [Online]. Available: http://mathworld.wolfram.com/Field.html. [Accessed 1 September 2012]
[5] E. W. Weisstein, "Subring," 1 September 2012. [Online]. Available: http://mathworld.wolfram.com/Subring.html. [Accessed 1 September 2012].
[6] J. A. Galian, Abstract Algebra with GAP, Brooklyn: Cole, 2010.
[7] J. J. Rotman, Advanced Modern Algebra, 2nd ed., New Jersey: Prentice Hall, 2003.
[8] N. I. Manik, "Pengujian Struktur Aljabar Grup, Ring, & Field Berbasis Komputer," in *Prosiding SNM-2010*, Depok, 2010.
[9] N. R. Dewi, "Kajian Struktur Aljabar Grup pada Himpunan Matriks yang Invertibel," *Jurnal Penelitian Sains* , vol. 14, no. 1A, pp. 14101-1 - 14101-3, 2011.
[10] R. S. Pressman, Software Engineering: A Practitioners Approach, 6th ed., Singapore: McGraw-Hill, 2005.
[11] W. K. Nicholson, Introduction to Abstract Algebra, Singapore: John Wiley, 2012.