Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Application of Generalization of Eisenstein Criterion to Verify Irreducible Polynomial over Z[x] using MATLAB GUI

Leonardus Reinaldy Christianto, Irawati Irawati
leonardusrenaldi@gmail.com, irawati@math.itb.ac.id
Bandung Institute of Technology, Indonesia

Article Information

Submitted : 7 Dec 2022
Reviewed: 22 Dec 2022
Accepted : 30 Dec 2022

Keywords

Generalization of Eisenstein Criterion, Irreducible, Matlab Gui, Polynomial

Abstract

Technology plays an important role in education development to prepare students to answer global challenges. Programming is one of the skills students need to have to adapt to advanced technology. Before preparing students for that skill, teachers need to be experts in programming into the subject they teach. Polynomial is a compulsory course for a master's degree in teaching mathematics. It prepares the master's student about the concept of an irreducible over any field. Each polynomial in the field C with degree one is an irreducible polynomial, while an irreducible polynomial in R[x] is of degrees one and two. However, for every polynomial in the field Q of degree n∈Z, there is any irreducible polynomial, so it is hard to decide whether the polynomial with any degree n is irreducible or not. This study aims to develop the application of teaching and learning Polynomial using MATLAB GUI will be presented to check the irreducibility of a polynomial over Q[x] by using theorem on the Generalization of the Eisenstein Criterion. This application can be applied by lecturers and students of universities, moreover, the algorithm and steps of making an application can be adjusted to make the other application in mathematics.

A. Introduction

Polynomial is one of the compulsory courses in the master's degree in the teaching of mathematics. This polynomial concept is very important for a teacher to master before teaching their students. The concept studied in polynomials is the irreducibility of a polynomial on the field. Let $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, where $0 \leq i \leq n$ and $i \in \mathbb{Z}$. The coefficient a_i is a number from several fields, such as $\mathbb{Q}, \mathbb{R}, \mathbb{C}$, or $\mathbb{Z}_p$, for $p \in$ prime. [7] defines that let F be a field and $f(x), g(x), h(x) \in F[x]$ with $\delta(f(x)) > \delta(g(x))$ and $\delta(f(x)) > \delta(h(x))$, so that $f(x) = g(x)h(x)$, then the polynomial $f(x)$ is reducible in F. We can say that the polynomial is not constant $f(x)$ is irreducible if it does not meet these conditions.
[4] states that each irreducible polynomial in $\mathbb{R}[x]$ has either one or two degrees. Meanwhile, for every polynomial of degree one in $\mathbb{C}[x]$, it is an irreducible polynomial. In contrast, it is found that there are infinitely many irreducible monic polynomials for every positive degree $n \in \mathbb{Z}$ in $\mathbb{Q}[x]$.
In a polynomial in $\mathbb{Q}[x]$ of degree two, you can use completing perfect squares method or determine the discriminant value of the polynomial. If the discriminant value $(d) = k.k$ for $k \in \mathbb{Q}$ is obtained, then the polynomial is reducible in $\mathbb{Q}[x]$. So, if the polynomial does not meet these conditions, then the polynomial of degree two is irreducible in $\mathbb{Q}[x]$. In the polynomial in $\mathbb{Q}[\ x]$ degree three can use the Cardano method.

Since it has been proven that there is an irreducible polynomial in $\mathbb{Q}[x]$ for every degree n [4], it will be difficult to determine whether a polynomial is reducible or not at a sufficiently high degree, for example, whether $f(x) = x^{20} + x^{18} - 7x^2 + 5x - 1$ is a reducible polynomial in $\mathbb{Q}[x]$ or not. It has been proven that every polynomial in $\mathbb{Q}[x]$, can be expressed as a polynomial $\mathbb{Z}[x]$, so to check the irreducibility of a polynomial in $\mathbb{Q}[x]$ can use a method that is quite often used, namely the Eisenstein Criterion popularized by Gotthold Eisenstein in 1850 [9]. However, there are still many polynomials that do not meet Eisenstein Criterion which are irreducible polynomials, for example in the polynomial $f(x) = 4x^7 + 2x^6 + x^5 + x^4 + 2x + 12 \in \mathbb{Z}[x]$, because $p|a_7$ the polynomial does not meet the Eisenstein Criterion. Then the interest was to develop another method involving the concept of Newton Diagrams by Gustave Dumas into a method called the Dumas Criterion, introduced in 1906 [2].
Tests for polynomials that meet the Eisenstein Criterion are carried out against the Dumas Criterion, it is found that every irreducible polynomial that meets the Eisenstein Criterion also meets the Dumas Criterion, so the Dumas Criterion is also often referred to as Generalizations of the Eisenstein Criterion. However, some polynomials cannot be verified by either the Dumas Criterion or Eisenstein Criterion, namely the polynomial $f(x) = x^2 + p^2 \in \mathbb{Z}[x]$ for p is a prime number. This is what motivated the emergence of the Generalization of the Eisenstein Criterion published by Akash Jena in 2016 in a journal entitled "Revisiting Eisenstein-type criterion over integer" [5].

In the use of research methodology, this study uses a computational approach to determine a polynomial in $\mathbb{Z}[x]$ is an irreducible polynomial by applying the Generalization of the Eisenstein Criterion. MATLAB GUI will be used to build an application to determine the irreducibility of a polynomial in $\mathbb{Z}[x]$ so that the reader only needs to enter the coefficient value of the polynomial, then the program is run and shows the conclusion whether the polynomial is an irreducible polynomial or the polynomial does not meet the Generalization Criterion Eisenstein. The study making applications using MATLAB GUI has been done for several research, for example, "Calculus Problem Solution and Simulation Using GUI of Matlab" [10] and "Physics calculator application with MatLab as a learning media to thermodynamics concept" [11]. This study is not only to make an application to verify irreducible polynomials, but also to demonstrate the steps and algorithms of the application. so that teachers or lecturers get a new insight to create MATLAB GUI applications by applying theorems in mathematics.

B. Generalization of Eisentein Criterion

To check for more irreducible polynomials in $\mathbb{Z}[x]$, you can use the Generalization of the Eisenstein Criterion by Jena and Sahoo [5]:

Theorem 1: Let $g(x) = a_n x^n + a_{n-1}\, x^{n-1} + \cdots + a_1 x + a_0 \in Z[x]$ and k be a positive integer that is relatively prime with n. Suppose there is a prime number p such that $p \nmid a_n, p^k | a_j$ with $0 \le j \le n-1$ and $p^{k+1} \nmid a_0$, then $g(x)$ is an irreducible polynomial in $\mathbb{Z}[x]$.

In this study, we will only use the value of $k = \{1,2,3,4\}$

For $k = 1$

For $k = 1$ is the Eisenstein Criterion

Theorem 2: Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \in \mathbb{Z}[x]$ be a polynomial with coefficients on integers such that the coefficients a_n is not divisible by prime number p, while coefficient $a_0, \dots, a_{n-1}$ is divisible by p but a_0 is not divisible by p^2. Then $f(x)$ is an irreducible polynomial in $\mathbb{Z}[x]$.

An example of applying theorem 2, as follows, suppose $f(x) = x^{20} + 2x^{18} - 8x^2 + 2x - 2 \in \mathbb{Q}[x]$, choose $p = 2$ so that $p \nmid a_{20}$ and $p | a_i$ with $0 \le i < 20$ for $i \in \mathbb{Z}$, and $p^2 \nmid a_0$. It is concluded that $f(x)$ satisfies the Eisenstein Criterion so that $f(x)$ is an irreducible polynomial in $\mathbb{Z}[x]$.

Another example is in cyclotomic polynomials, let $f(x) = \frac{x^p - 1}{x - 1} = x^{p-1} + x^{p-2} + \cdots + x + 1$, with p as a prime number, is an irreducible polynomial because it satisfies theorem 2.

For $k = 2$

Theorem 3: Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \in \mathbb{Z}[x]$. Suppose there is a prime number p such that $p \nmid a_n, p | a_i$ for $i \le n-1, p^2 | a_j$ for $j \le \left\lfloor \frac{n}{2} \right\rfloor$ and $p^3 \nmid a_0$. Applicable

- If n is an odd number, then $f(x)$ is an irreducible polynomial in $\mathbb{Z}[x]$
- If n is an even number, then either $f(x)$ is an irreducible polynomial in $\mathbb{Z}[x]$, or $f(x)$ is the product of exactly two non-reduced polynomials in $\mathbb{Z}[x]$ with the same highest degree and the Eisenstein Criterion applies.

An example of theorem 3 is as follows, suppose $f(x) = x^3 + 3x^2 - 54x + 27 \in \mathbb{Z}[x]$ is an irreducible polynomial in $\mathbb{Z}[x]$ because it satisfies theorem 3 on n is an odd number. Meanwhile, for a polynomial where n is an even number, two possibilities are found, namely an irreducible or reducible polynomial, for example, let's say $f(x) = x^2 + p^2 \in \mathbb{Z}[x]$, where p is a prime number that satisfies theorem 3, and $f(x)$ cannot be expressed in terms of the product of two irreducible polynomials, it is concluded that $f(x)$ is an irreducible polynomial in $\mathbb{Z}[x]$, while for the polynomial $g(x) = x^2 - p^2 \in \mathbb{Z}[\,x]$ satisfies theorem 3, but $g(x)$ can be expressed as a product of two irreducible polynomials, namely $g(x) = (x - p)(x + p)$, so it can be concluded that $g(x)$ is a reducible polynomial.

For $k = 3$

Theorem 4: Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \in \mathbb{Z}[x]$, where n is not divisible by 3. Suppose there is a number prime p such that the coefficient a_n is not divisible by p, a_i is divisible by p^3 by $0 \le i < n$, and a_0 is not divisible by p^4. Then $f(x)$ is an irreducible polynomial in $\mathbb{Z}[x]$.

An example of applying theorem 4, as follows, suppose $f(x) = x^2 + p^3 \in \mathbb{Z}[x]$ where p is a prime number, satisfies the conditions in Theorem 4. It can be concluded that $f(x)$ is an irreducible polynomial in $\mathbb{Z}[\,x]$.

For $k = 4$

Theorem 5: Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \in \mathbb{Z}[x]$, where $(n, 4) = 1$. Suppose there is a prime number p such that a_n is not divisible by p, a_i is divisible by p^4 by $0 \le i < n$, and a_0 is not divisible by p^5. Then $f(x)$ is an irreducible polynomial in $\mathbb{Z}[x]$.

An example of applying theorem 5 is as follows, suppose $f(x) = x^3 - p^4 \in \mathbb{Z}[x]$ where p is a prime number, satisfies theorem 5. It can be concluded that $f(x)$ is an irreducible polynomial in $Z[x]$.

C. GUI of MATLAB

Matlab (Matrix Laboratory) which consists of 5 main components namely [10]:

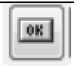
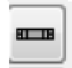
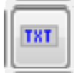
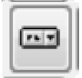
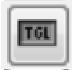
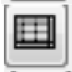
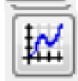
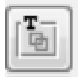
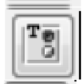
a. Toolbar contains various control tools for Matlab
b. A current folder is a place of folders or files that have been stored and connected with Matlab.
c. Windows Command as the main worksheet in Matlab where users build regular scribes direct execution.
d. Workspace as a workspace where files or variables are stored in Command Windows.
e. Graphical User Interface (GUI) is one of the components of Matlab to create a more efficient and appealing mathematical problem-solving interface.

A graphical user interface (GUI) is a pictorial interface to a program. A good GUI can make programs easier to use by providing them with a consistent appearance with intuitive controls such as pushbuttons, edit boxes, list boxes, sliders, and menus. There are three principal elements requires to create a MATLAB graphical user interface are [12]:

1. Each item on a MATLAB GUI (pushbuttons, labels, edit boxes, etc.) is a graphical component. The types of components include graphical controls (pushbuttons, toggle buttons, edit boxes, lists, sliders, etc.) static elements (text boxes), menus, toolbars, and axes. The types of graphical component is

described in Table 1 below. but in this study not all components are used. The components used in the development of this application that is *Push Button, EditText, Pop-upMenu, StaticText* [12].

Table 1. Graphical Component Description

Name	Icons	Function
Push Button		As a button (process, delete, etc.
Slider		Minimize the screen display
RadioButton		Implement mutually exclusive choice
CheckBox		To put a choice
EditText		As a place of input or output
StaticText		As the property label is used
Pop-upMenu		Same with Check Box
ListBox		Output in a large number of strings
ToggleButton		Same with Push Button
Table		Make output in tabular form
Axes		To draw a graph/histogram
Panel		Unify attributes in one group
BtnGroup		Unify attributes in one group
Activex Control		Brings up some important controls.

2. Containers. The components of a GUI must be arranged within a container, which is a window on the computer screen. The most common container is a figure. A figure is a window on the computer screen that has a title bar along the top and that can optionally have menus attached. Figures have been created automatically whenever we plotted data. However, empty figures can be created with the function figure, and they can be used to hold any combination of components and other containers.
   The other types of containers are panels and button groups. Panels can contain components or other containers, but they do not have a title bar and cannot have menus attached. Button groups are special panels that can manage groups of radio buttons or toggle buttons to ensure that no more than one button in the group is on at any time [12].

3. Callbacks. Finally, there must be some way to perform an action if a user clicks a mouse on a button or types information on a keyboard. A mouse click or a key press is an event, and the MATLAB program must respond to each event if the program is to perform its function. For example, if a user clicks on a button, then that event must cause the MATLAB code that implements the function of the button to be executed. The code executed in response to an event is known as a callback. There must be a callback to implement the function of each graphical component on the GUI [12].

D. Results and Discussion

The construction of an application to verify the irreducibility of a polynomial in $\mathbb{Z}[x]$ by using the theorem on the generalization of the Eisenstein Criterion begins with making an algorithm from the Matlab GUI that will be built. To build an algorithm, what needs to be done is to study the theorem of generalization of the Eisenstein criterion further, and start registering the components needed to support the running of the theorem. Two main components are needed for this theorem to work, namely degree of polynomial and prime numbers. After that, re-examine the theorem and determine the primary conditions for the theorem to work and not work. For example, because in the generalization theorem of the Eisenstein criterion, when the value of a_n is divisible by prime numbers p, the polynomial does not meet the theorem, otherwise if the value of a_n is not divisible by p, the program will proceed to the next stage. The next condition is that each coefficient a_i must be divisible by p with $0 \leq i < n$ and $i \in \mathbb{Z}$, so the program will continue, but if it is not divisible, the polynomial does not meet the generalization of the Eisenstein criterion. The complete results of the algorithm construction can be seen in Figure 1.

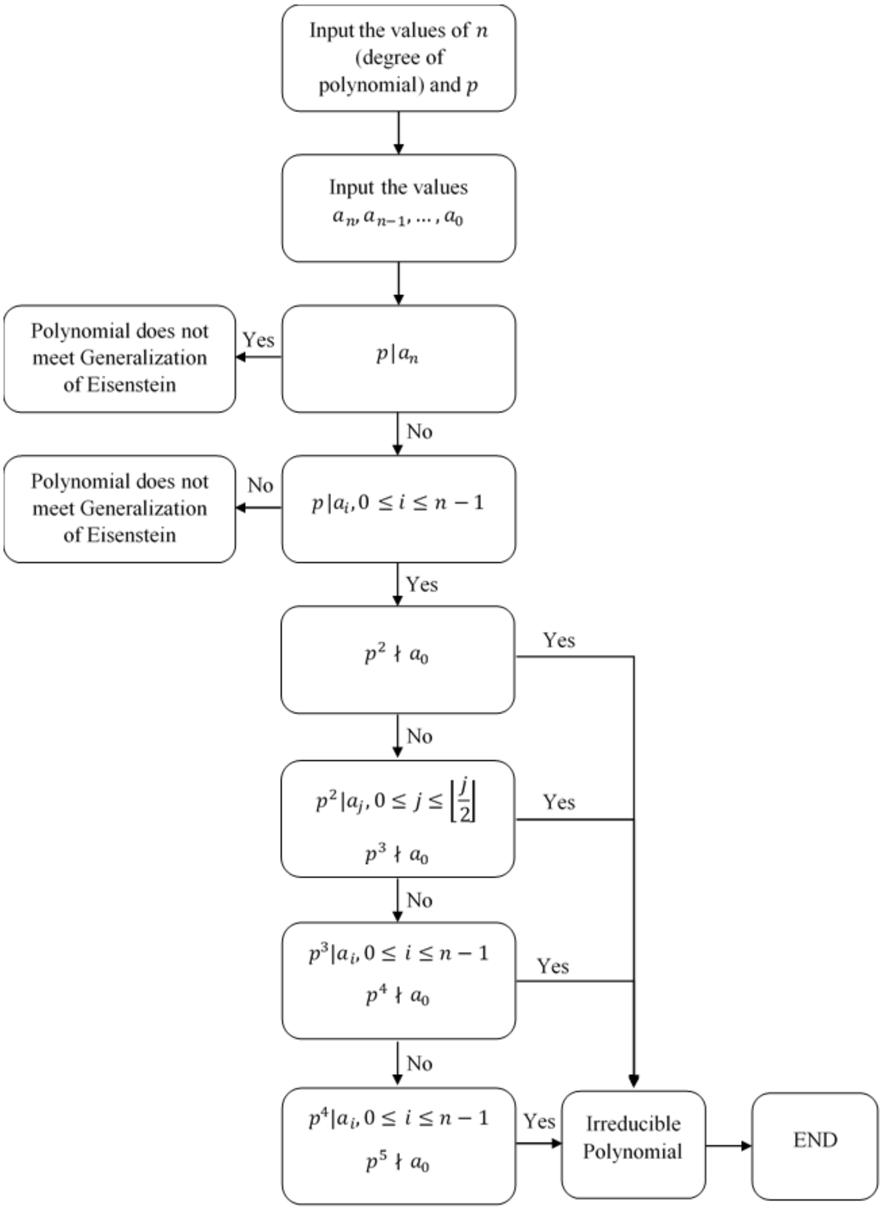


Figure 1. Algorithm of Generalization Eisenstein Criterion MATLAB GUI

After the algorithm is formed, the next step is to create a design or user experience for the Matlab GUI to be built. In this application, two types of displays are connected, namely, the initial display or so-called degree of polynomial selection display to select the degree of polynomial to be used, for example, the polynomial to be tested is $x^5 + 3x^4 - 6x^3 + 3x^2 - 12x + 3$, so that the degree of the polynomial is 5. After selecting the degree of the polynomial, the display will change to the input display of the coefficient value and prime number p, then by clicking the *OK* button, the algorithm will work and give the result of whether the polynomial is irreducible or not meet the generalization of the Eisenstein criterion.

When the design or user experience has been completed, the next step is to identify the components needed to build the application. In the degree of polynomial selection display, *Statictext* is needed to write the application title, then it requires a *PushButton* to select the degree of polynomial to be used, and a close button to close the page. After the degree of the polynomial is selected, the display will switch to the input display. In this view, it requires an *EditText* to enter the value of the prime number to be used and the coefficient to be tested, and in the panel or output section an *EditText* is given to display the results. Additionally, it requires three *PushButtons* for Execute, Clear, and Close options. Because there are 20 degrees of polynomial options, namely at degrees 2 to 21, twenty input displays will be made that represent each degree.

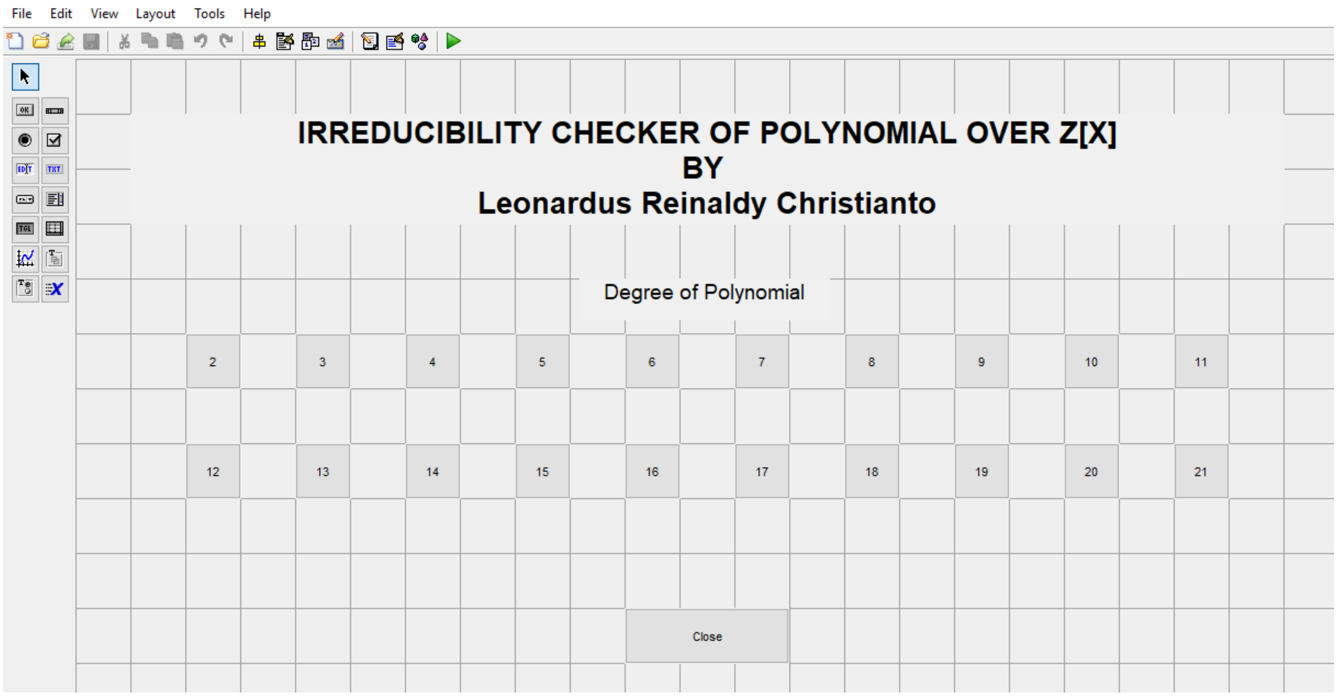


Figure 2. Design of Degree of Polynomial Selection Display.

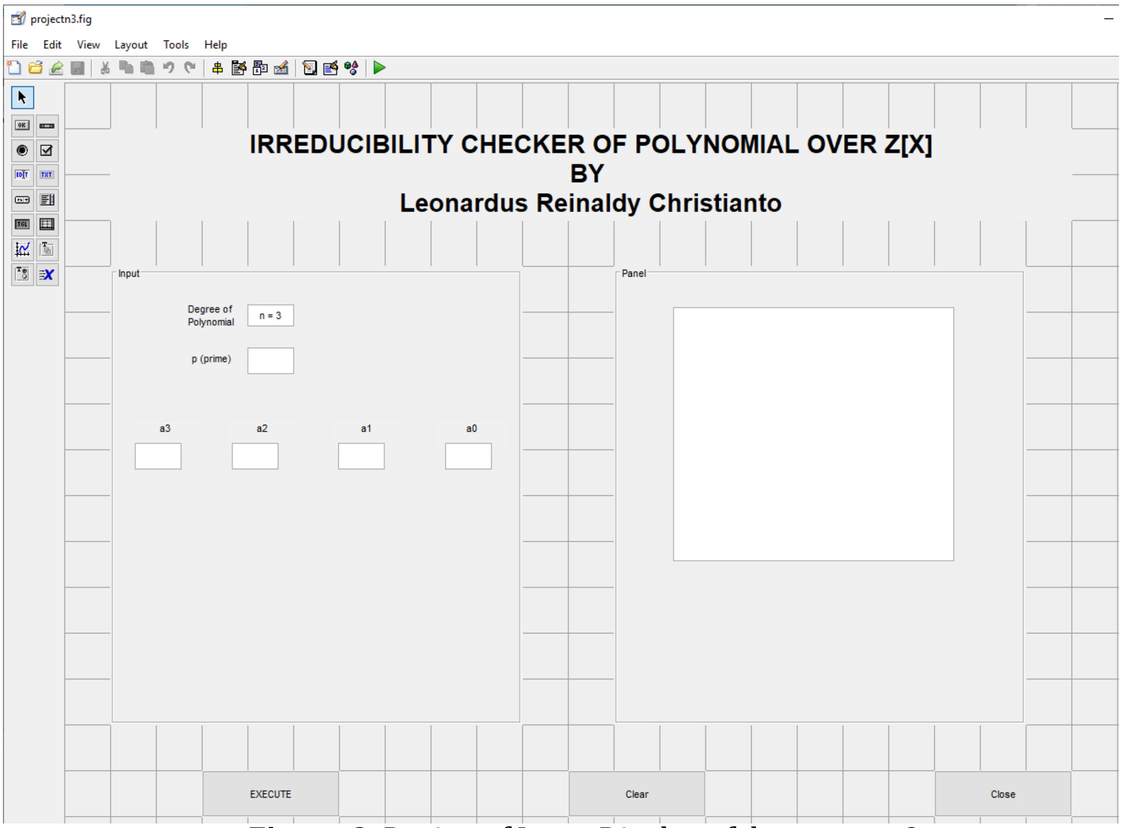


Figure 3. Design of Input Display of degree $n = 3$

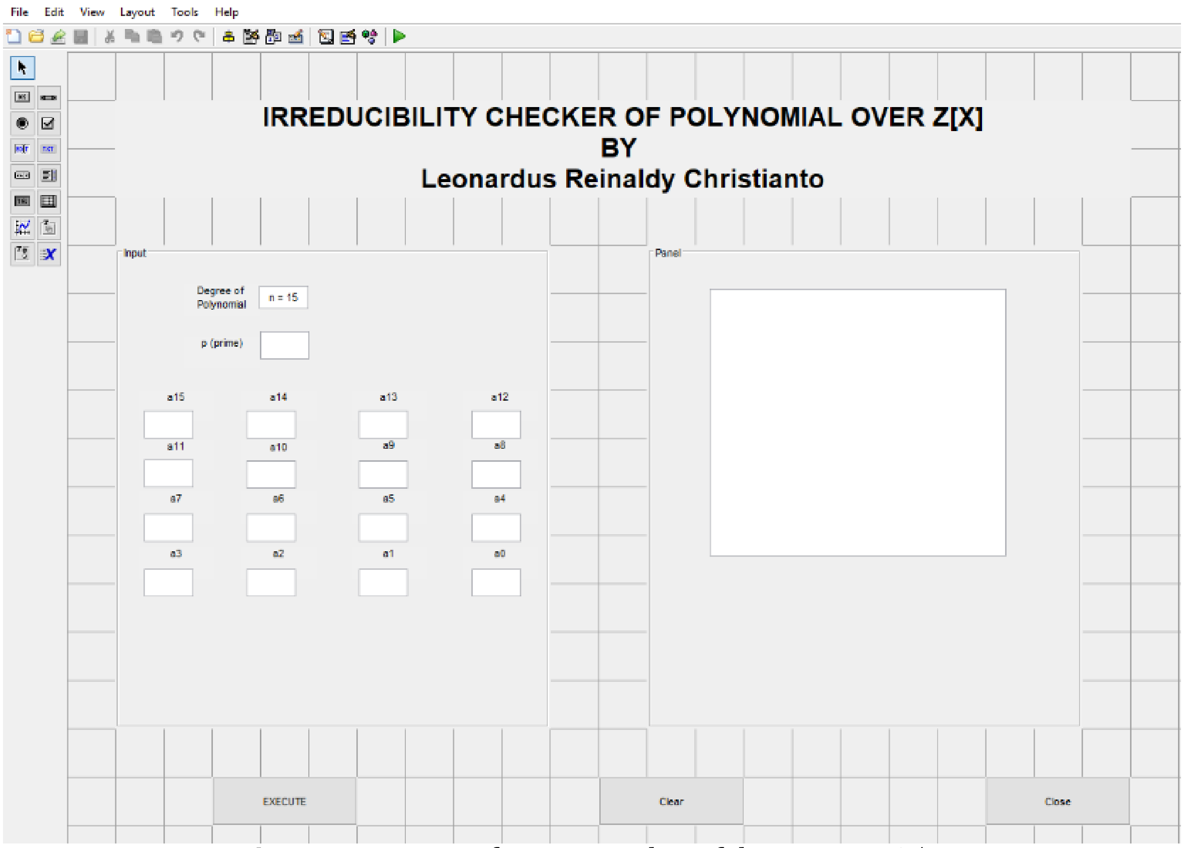


Figure 4. Design of Input Display of degree $n = 15$.

Done with designing the user experience display, the next step is to enter the program or callback on each component. It begins with giving the program to the *PushButton* of degree of polynomial. When the user clicks on one of the *PushButtons*, the display will become the input display Figure 3 or Figure 4, so that the *PushButton* callback will hyperlink to the appropriate input page. To perform a hyperlink, you only need to type the file name of the intended display input, for example, the file name is saved with the name "project", so the callback program becomes as follows:

```
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
project
```

We give Callback for each pushbutton. On the Degree of Polynomial Display, there is a "Close” button, so when you click on "Close", the application will automatically close. The callback for the pushbutton is as follows:

```
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Close
```

We're done with giving a Callback to the Degree of Polynomial Selection Display. Next is giving a callback to the Input Display. In this section the algorithm starts to work. Beginning with the identification of the prime numbers and the entered coefficients. For example on a polynomial with degree $n = 2$, so the Callback becomes as follows:

```
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
prime=str2num(get(handles.edit3,'string'));
a2=str2num(get(handles.edit4,'string'));
a1=str2num(get(handles.edit5,'string'));
a0=str2num(get(handles.edit6,'string'));

residuea2=mod(a2,prime);
residuea1=mod(a1,prime);
residuea12=mod(a1,prime^3);
residuea0=mod(a0,prime);
residuea02=mod(a0,prime^2);
residuea03=mod(a0,prime^3);
residuea04=mod(a0,prime^4);
```

After providing the identification of prime numbers and coefficients, the next step is to use if logic to build an algorithm in Figure1 which is shown as follows:

```
if residuea2==0
    set(handles.edit7,'string','Generalisation of Eisenstein Criterion is not applicable
for this polynomial');
elseif (residuea0<0 || residuea0>0)
    set(handles.edit7,'string','Generalisation of Eisenstein Criterion is not applicable
for this polynomial');
elseif (residuea1<0 || residuea1>0)
    set(handles.edit7,'string','Generalisation of Eisenstein Criterion is not applicable
for this polynomial');
elseif (residuea02==0 && a1==2*prime)
```

```
    set(handles.edit7,'string','Generalisation of Eisenstein Criterion is not applicable
for this polynomial');
elseif (residuea2<0 || residuea2>0 && residuea1==0 && residuea02==0)
    set(handles.edit7,'string','Generalisation of Eisenstein Criterion is not applicable
for this polynomial');
elseif (residuea2<0 || residuea2>0 && residuea1==0 && residuea02<0 || residuea02>0)
    set(handles.edit7,'string','Irreducible Polynomial');
elseif  (residuea2<0  ||  residuea2>0  &&  residuea12==0  &&  residuea0==0  &&  residuea04<0
||residue>0)
    set(handles.edit7,'string','Irreducible Polynomial');
else (residuea2<0|residuea2>0 & residuea1==0 & residuea02==0);
    set(handles.edit7,'string','Irreducible Polynomial');
end
```

The results of the conclusions obtained from the logic will be displayed in the output section or panel.

On the input display, there is also a "Clear" button, which is used to clear information, so that the user can rewrite the value to be tested. The callback from the "Clear" button is as follows:

```
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.edit3,'string','')
set(handles.edit4,'string','')
set(handles.edit5,'string','')
set(handles.edit6,'string','')
set(handles.edit7,'string','')
```

The program creation process has been completed, then a trial will be carried out on the entered values.

For example, it will be tested whether the polynomial $x^3 + 2x^2 + 2x + 2$ is irreducible with the prime number $p = 2$, so that the result is as in Figure5, namely, the polynomial is irreducible.

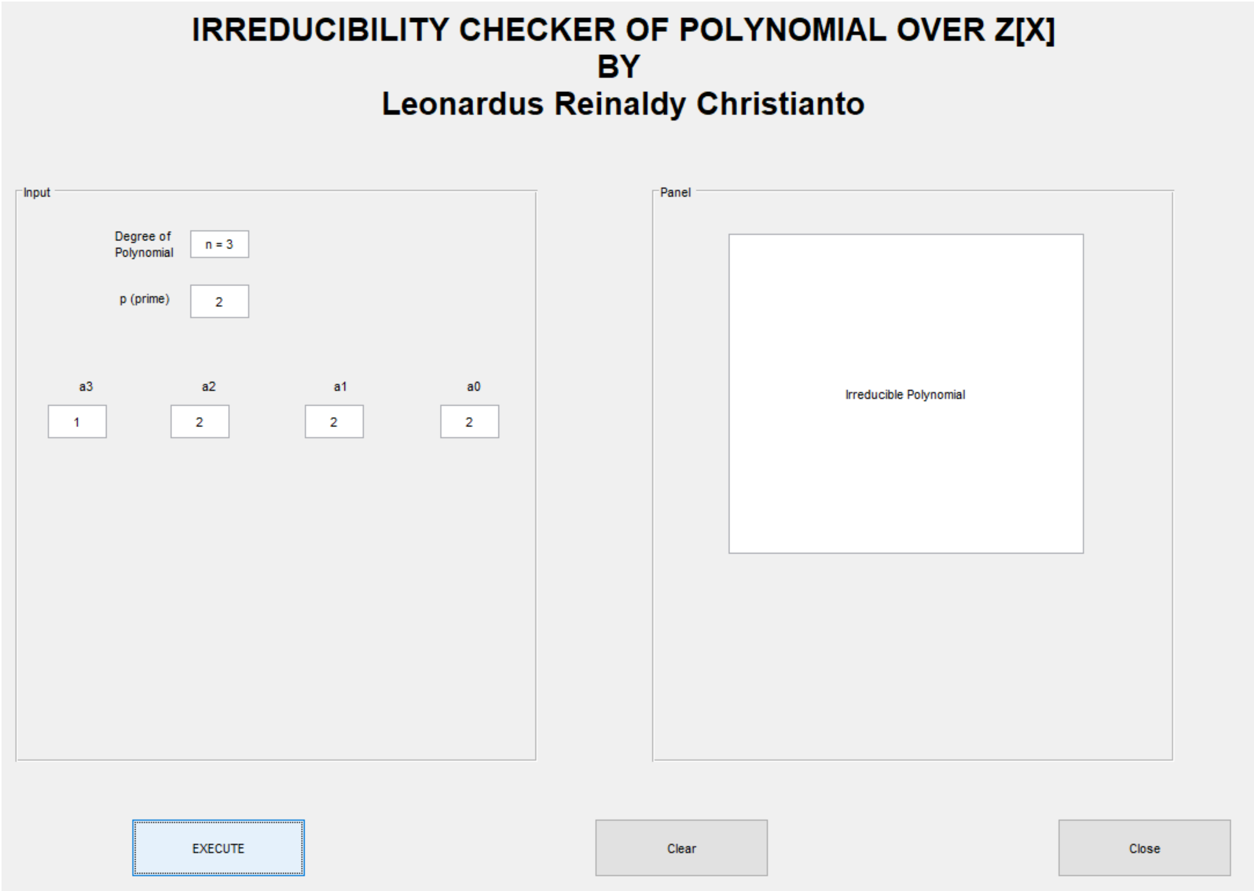


Figure 5. Test of $x^3 + 2x^2 + 2x + 2$

Next, it will be tested whether the polynomial $x^3 + 2x^2 + 2x + 4$ is irreducible with the prime number $p = 2$, so that the results are as in Figure6, namely the generalization of Eisenstein criterion is not applicable for this polynomial.

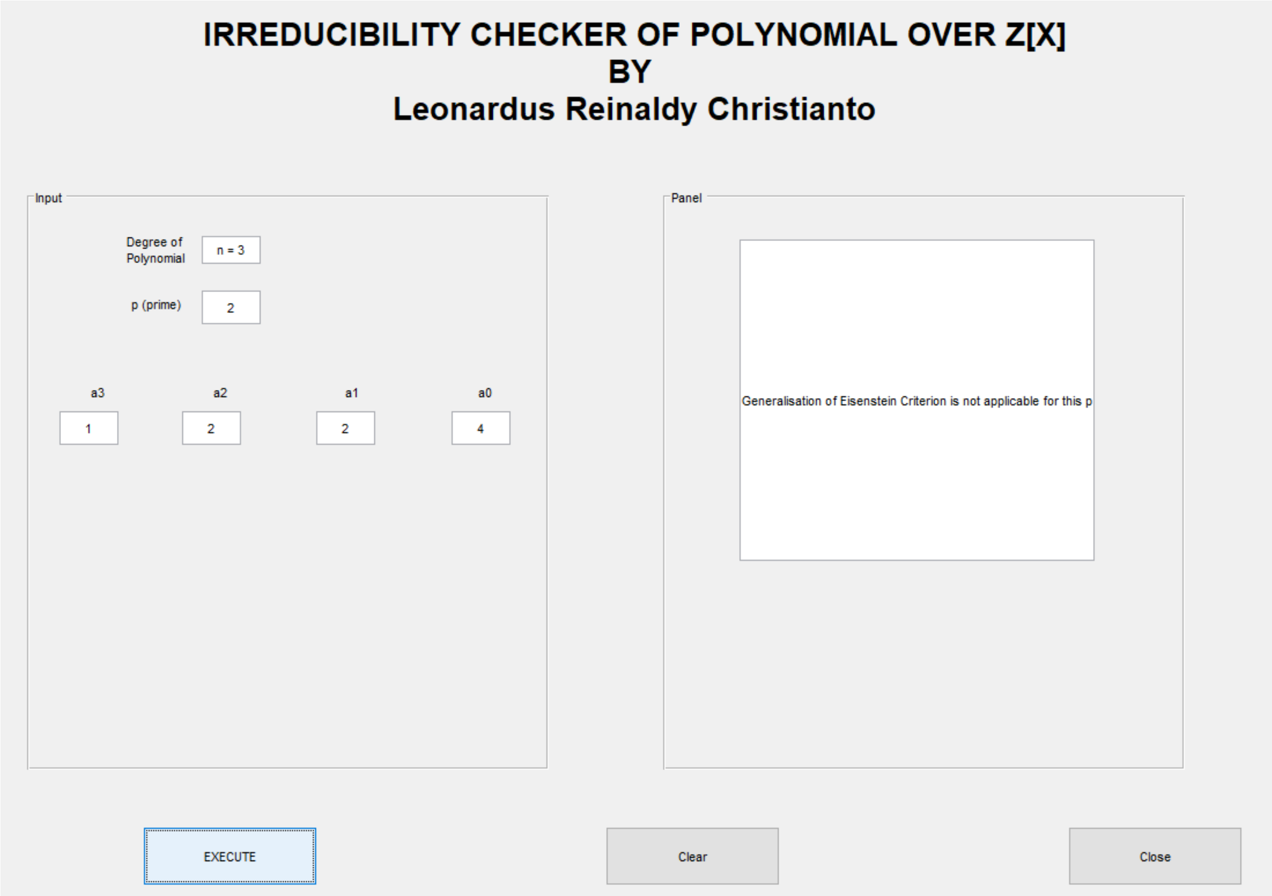


Figure 6. Test of $x^3 + 2x^2 + 2x + 4$

E. Conclusion

Based on the explanation above, it can be concluded that the steps to create an application using Matlab GUI are as follows:

1. Build an algorithm
2. Create a design or user experience for the application
3. Determine the components needed to make the application
4. Provide a callback to each component that is needed
5. Simulating whether the program works well or not

This step is quite easy for teachers or lecturers to follow in exploring other theorems to be made into applications using the MATLAB GUI. This study only explains the process of making an application using MATLAB GUI to verify the irreducibility of a polynomial. Therefore, future studies should statistically identify the success rate of this application.

F. Acknowledgment

Acknowledgment to Bandung Institute of Technology who have provided support for the research.

G. References

[1] Chapman, Stephen, J., *MATLAB Programming for Engineers*. Canada: Thomson, 2008.
[2] Dumas, G., "Sur quelques ca d'irreducibilite des polynomes a coefficients rationels," Journal De Mathematiques Pures Ey Appliquees, pp. 191-258, 1906.
[3] Gao, S., "Absolute Irreducibility of Polynomials via Newton Polytopes," Journal of Algebra, pp. 501-520, 2001.
[4] Irving, Ronald S., *Integers, Polynomials, and Rings*. United States of America: Springer, 2000.
[5] Jena, A., Sahoo, B. K. "Revisiting Eisenstein-type criterion over integers," The Mathematics Student 86, pp. 77-86, 2017
[6] Oleinikov, V., "Irrationality and Irreducibility," Kvanta Selecta: Algebra and Analysis, II, pp. 95-104, 2009.
[7] Prasolov, V.V., *Algorithms, and Computation in Mathematics: Polynomials*. Russia: Springer, 2001.
[8] Salman, Nassir H., Hadi, Gullanar M., "Integrated Image Processing Functions using MATLAB GUI," Journal of Advanced Computer Science and Technology Research, Volume 3, pp. 31-38, 2013.
[9] Sury, B., "Polynomials with Integer Values," Resonance, pp. 46-60, 2001.
[10] Syaharuddin, Negara, P. R. H., Mandalina, V., Sucipto, L., "Calculus Problem Solution and Simulation Using GUI of Matlab," International Journal of Scientific & Technology Research, Volume 6, September 2017.
[11] Utari, Kintan., Mulyaningsih, Nenden N., Astuti, Irnin A.D., Bhakti, Budi Y., Zulherman, "Physics calculator application with MatLab as a learning media to thermodynamics concept," Momentum: Physics Education Journal, pp. 101-110, 2021.
[12] Wilson, Howard B., Turcotte, Louis H., Halpern, D., *Advance Mathematics and Mechanics Application Using MATLAB*. Florida: CRC Press Company, 2003.