EXPLORATION OF COMPLEX NUMBER VISUALIZATION IN JULIA SET-BASED FRACTAL GEOMETRY

Anggraeni Maha Dewi[1], Laili Khamdanah[2], Lila Dalalah[3], Umi Mahmudah[4]
UIN K.H. Abdurrahman Wahid Pekalongan
e-mail: anggraenimahadewi@mhs.uingusdur.ac.id

ABSTRAK

Penelitian ini mengeksplorasi visualisasi bilangan kompleks dalam konteks geometri fraktal, terutama fokus pada Fraktal Julia Set. Tujuan penelitian adalah untuk memahami bagaimana parameter-parameter dalam fungsi iterasi memengaruhi struktur fraktal, dengan perhatian khusus pada variasi warna, bentuk, dan detail geometris. Metode penelitian yang digunakan adalah deskriptif kualitatif dengan menggunakan studi pustaka sebagai sumber data, termasuk buku, jurnal ilmiah, dan literatur lainnya. Untuk menjaga keabsahan data, teknik triangulasi sumber data digunakan, yaitu mengumpulkan data dari berbagai sumber seperti buku, jurnal ilmiah, dan literatur lainnya untuk memperkuat validitas temuan. Teknik analisis data yang diterapkan meliputi analisis tematik untuk mengidentifikasi pola-pola dan tema-tema utama yang muncul dari literatur yang diselidiki. Hasil penelitian menunjukkan bahwa sifat-sifat bilangan kompleks memiliki kontribusi signifikan dalam pembentukan pola fraktal yang kompleks dan menarik secara visual.

Kata kunci *:* Bilangan kompleks, geometri fraktal, Julia set.

ABSTRACT

This research explores the visualization of complex numbers in the context of fractal geometry, particularly focusing on fractal Julia Sets. The purpose of the research is to understand how the parameters in the iteration function affect the fractal structure, with special attention to the variations in color, shape, and geometric details. The research method used is descriptive qualitative using literature studies as data sources, including books, scientific journals, and other literature. To maintain the validity of the data, a data source triangulation technique was used, collecting data from various sources such as books, scientific journals, and other literature to strengthen the validity of the findings. The data analysis techniques applied included thematic analysis to identify patterns and main themes that emerged from the literature investigated. The results show that the properties of complex numbers *have a significant contribution in the formation of complex and visually appealing fractal patterns.*

INTRODUCTION

Fractal geometry offers a new way to understand and describe the complexity of nature's irregular and repetitive shapes, going beyond the limitations of classical geometry. According to Mandelbrot, classical or Euclid geometry is not suitable in describing natural forms such as clouds, mountains, coastlines or trees. Clouds are not spherical, mountains are not conical, and coastlines are not circular. So Mandelbrot conceived and developed a new natural geometry and implemented its use in various fields, namely fractal geometry (Palma & Nadiasari, 2022). Fractal Geometry is the formal study of self-similar structures and is the conceptual core of understanding the complexity of nature (Wahyuningsih & Hernadi, 2020).

The term “fractal” was first introduced by Benoit Mandelbrot in 1975 in his book entitled A Theory of Fractal Sets to describe objects that have the property of similar shapes at different scales (Jaya & Aliansa, 2017). Benoit Mandelbrot introduced the term 'fractal' to describe objects that have self-similarity and do not have clearly defined dimensions. So it can be concluded that fractal geometry can be defined as a field of science that studies irregular shapes and has no definite dimensions, but shows similar patterns at different scales (Rohmah & Hernadi, 2020).

The existence of fractal geometry shows that mathematics is not a dry, flat and monotonous science, but a beautiful science that can produce works that have high artistic and intellectual value. (Romadiastri, 2013). Fractals not only provide deep theoretical insights, but also have wide practical applications. In technology, fractal concepts are used in image compression, antenna design, and traffic pattern modeling. The visual beauty of fractals is also appreciated in digital art and computer graphics.

One of the most well-known types of fractals is the Julia Set. A Julia Set is a mathematical object that displays the property of self-similarity, where the same pattern repeats itself at various scales. These fractals are generated from complex polynomial functions, which is a mathematical process that allows the creation of very intricate and interesting shapes (Kodri & Titaley, 2017). The Julia Set provides a clear illustration of how a seemingly simple iteration process can result in a very complex set and can give rise to images that are incredibly stunning and full of detail (Rohmah & Hernadi, 2020).

In the context of Julia Set, visualization of complex numbers allows us to observe the patterns resulting from the iteration of complex functions directly. By using appropriate visualization techniques, we can show how fractal patterns are formed, including the small details hidden in them. The visual characteristics of the Julia Set fractals, such as color, shape, and geometric details, are strongly influenced by the parameters in the iteration function used.

Based on the description above, researchers are interested in exploring the visualization of complex numbers in the context of fractal geometry, specifically Julia Set. The purpose of the research is to understand how the parameters in the iteration function affect the fractal structure of the Julia Set, with special attention to the variations in color, shape, and geometric details.

To achieve the purpose of this research, a descriptive qualitative research method was used by using literature studies as data sources, including books, scientific journals, and other literature. To maintain the validity of the data, a data source triangulation technique was used, which is collecting data from various sources such as books, scientific journals, and other literature to strengthen the validity of the findings.

Once the data has been collected, the data will be analyzed using thematic analysis. Thematic analysis involves the process of identifying, analyzing, and reporting the main patterns or themes that emerge from the data obtained from the literature (Heriyanto, 2018). These themes will then be interpreted and synthesized to answer the research questions and provide new insights into the relationship between the iteration function parameters and the visual characteristics of Julia Set fractals.

DISCUSSION

Fractal Geometry provides a mathematical construction framework for studying disordered sets (Widodo, 2021). As we know classical geometry or Euclidean geometry lacks the ability to describe the naturalness or irregularity of these objects. While fractal geomatery is the right way to present natural objects. Fractals have the main characteristic of self-similarity. This characteristic makes fractals have the ability to model complex and irregular natural objects. Using this characteristic, fractals are also able to determine the dimensions of an object (Putra,2010). Dimensions contain a lot of information about the geometric properties of a set. In Euclid geometry, dimensions will always be integers, such as dimensioned flat geometry and dimensioned space geometry While fractal dimensions do not have to be integers, because fractals consist of objects that have irregular shapes (Wahyuningsih & Hernadi, 2020).

Fractals are divided into two, natural fractals and artificial fractals. Natural fractals are fractals that form naturally, such as in the patterns found in the leaves and twigs of trees, in broccoli vegetables, in white cloud clusters, in the ripples of waves, in the details that we can see in snowflakes, and many more if we try to pay close attention around us (Hasang & Supardjo, 2012). While artificial fractals are man-made (Faseha et al., 2019). Fractals are created by humans through the use of algorithms and computation, as opposed to natural fractals that occur spontaneously in nature.

A fractal is generated by repeating a pattern, usually in a recursive or iterative process (Romadiastri, 2013). In this process, the initial shape of the fractal is iteratively changed by a transformation, resulting in a new shape composed of several parts. Each part of the new shape is a scaled-down version of the original fractal shape, with a certain predetermined scale. Scale is the main parameter in fractal construction that affects the overall size and proportion of details in the fractal. Regardless of the scale used to construct a fractal, the inherent property of fractals is self-similarity (Wahyuningsih & Hernadi, 2020).

Some well-known examples of fractals are Mandelbrot sets, Julia sets, Cantor sets, Sierpinski triangles, and Von Koch curves. Cantor sets, Sierpinski triangles, and Von Koch curves are examples of classical fractals that belong to real number fractals. Mandelbrot sets and Julia sets are two very famous examples of fractals, which belong to complex number fractals. Both sets are related to each other. The Julia set was discovered first than the Mandelbrot set (Rohmah & Hernadi, 2020).

Julia Set is a fractal defined on complex numbers. The Julia Set is built from iterations of the complex function with itself (Faseha et al., 2019). This means that complex numbers are not merely an output as in the Mandelbrot Set, but also a parameter that influences the fractal shape. This shows how the nature of complex numbers that have both real and imaginary components can create complex and compelling fractal structures.

Julia Set discovered by Gaston Julia is a two-dimensional fractal related to complex numbers. The Julia Set is generated from an iterated function mapping $\boldsymbol{f_c(z)}: \mathbb{C} \rightarrow \mathbb{C}$ which is defined:

$$\boldsymbol{f_c(z) = z_n^2 + c}$$

Where $\boldsymbol{z}$**,** $\boldsymbol{c} \in \mathbb{C}$ and $\boldsymbol{c}$ is a constant. The series of complex numbers $\boldsymbol{z}$**,** $\boldsymbol{f_c(z), {f_c}^2(z), \ldots, {f_c}^n(z)}$, formed are called orbits (Solar et al., 2021). Some of these orbits are convergent (finite) and some are divergent (infinite). These convergent orbits will form the Julia Set. This highlights the importance of convergent orbits in forming the complex and amazing fractal geometry structure of the Julia Set.

One of the most interesting aspects of Julia Set fractals is the visual variety in the resulting patterns. Although these patterns are created through the iteration of simple functions on complex numbers, the results display an incredible richness of colors, shapes, and geometric details. These visual variations are not only aesthetically pleasing, but also provide insight into the properties of fractals and the complexity inherent in the underlying mathematical system.

Manually iterating the process to form a Julia Set is considered cumbersome, so the assistance of a computer application is essential. The chosen computer application should be able to facilitate the iteration process efficiently and provide a clear visualization of the resulting Julia Set, one of which is Matlab.

By using Matlab application, researchers can explore and create very diverse and visually appealing Julia Set fractal visualizations. Matlab provides built-in functions to create and visualize Julia Set fractals, and provides flexibility in setting parameters such as iteration function, constant value, and color scheme.

Here are the steps to construct the Julia Set using Matlab:

1. Take a function from the Julia Set, namely the complex quadratic function of the form $\boldsymbol{f_c = z^2 + c}$, where $\boldsymbol{z, c} \in \mathbb{C}$.
2. Select the desired value of c.
3. Determine the number of iterations to be performed, i.e. determine the value of $\boldsymbol{k}$ so that $\boldsymbol{f^k}(\boldsymbol{z}) \to \infty$.
4. Perform calculations according to the specified iteration, then visualize (Rohmah & Hernadi, 2020).

The following M-file (Matlab script) can be used to create a Julia Set visualization:

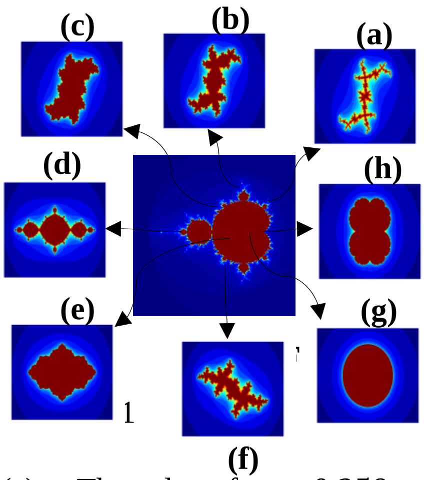
```
width = 800;
height = 800;
xmin = -2;
xmax = 2;
ymin = -1.5;
ymax = 1.5;

juliaSet = zeros(height, width);

c = a + bi;

for y = 1:height
    for x = 1:width
        zx = xmin + (x - 1) *
        (xmax - xmin) / (width –
        1);
        zy = ymin + (y - 1) *
        (ymax - ymin) / (height –
        1);
        z = zx + zy * 1i;
        iteration = 0;
        maxIteration = k;
        while abs(z) < 2 &&
              iteration <
              maxIteration
              z = z^2 + c;
              iteration =
              iteration + 1;
        end
        juliaSet(y, x) =
        iteration;
    end
end

imagesc(juliaSet);
colormap(jet);
axis off;
```

Description:

1. Initialize Image Size and Coordinate Boundaries:
   "width" and ***"height"*** to specify the image resolution, which is 800×800 pixels. "***xmin***", "***xmax***", "***ymin***", and "***ymax***" to determine the coordinate range on the complex plane to be visualized.
2. Initialize Matrix to Store Iteration Values:
   "***juliaSet***" is an 800×800 matrix filled with zeros. This matrix will store the iteration value for each pixel.
3. Complex Constants for Julia Set:
   "***c***" is a complex number used in iteration to form a Julia Set. The number ***a*** is a real number and ***b*** is an imaginary number.
4. Two Loops to Iterate Each Pixel in the Image:
   The first loop "***for y = 1:height***" iterates through the rows (y-coordinates).
   The second loop "***for x = 1:width***" iterates through the columns (x-coordinates).
5. Calculating the Complex Coordinates for Each Pixel:
   "***zx***" and "***zy***" calculate the x and y values on the complex plane based on the pixel position.
   "***z***" is the complex coordinate calculated from "***zx***" and "***zy***".
6. Iteration to Determine Whether Complex Values Converge or Diverge:
   "***iteration***" stores the number of iterations performed.
   "***maxIteration***" is the maximum limit of "***k***" iterations.
   The "***while***" loop evaluates whether the absolute value of "***z***" is less than 2 and the iteration is less than "maxIteration".
   If "***z***" remains within the boundary, the "***z***" value is updated using the formula "***z = z^2 + c***".
7. Storing Iteration Results in the Matrix
   The resulting "***iteration***" value is stored in the "***juliaSet***" matrix at position "***(y, x)***".
8. Display Julia Set Image:
   "***imagesc(juliaSet)***" to display the juliaSet matrix as an image.
   "***colormap(jet)***" to set the jet color scheme for the image.

Here are the results of the Julia Set visualization based on the complex number $\boldsymbol{c} = \boldsymbol{a} + \boldsymbol{bi}$, with "$\boldsymbol{k}$" iterations :

1. The parameter $\boldsymbol{c}$ is different with the same iteration of 20:



1 ılia Set with 20 iterations.

(a) The value of $\boldsymbol{c} = 0{,}359 + 0{,}599i$.

(b) The value of $\boldsymbol{c} = 0{,}276 + 0{,}536i$.

(c) The value of $\boldsymbol{c} = 0{,}198 + 0{,}49i$.

(d) The value of $\boldsymbol{c} = -1$.

(e) The value of $\boldsymbol{c} = -0{,}5$

(f) The value of $\boldsymbol{c} = -0{,}50125 - 0{,}5925i$.

(g) The value of $\boldsymbol{c} = 0$.

(h) The value of $\boldsymbol{c} = 0{,}25$.

The selection of different $\boldsymbol{c}$ values in the iteration function $\boldsymbol{f_c} = \boldsymbol{z^2} + \boldsymbol{c}$ will result in a very diverse Julia Set fractal pattern. The c value acts as an initial parameter that determines the structure and complexity of the fractal pattern formed.

Some values of $\boldsymbol{c}$ can produce regular, symmetrical, and relatively simple fractal patterns. For example in figure (g), when $\mathbf{c} = \mathbf{0}$, the resulting fractal pattern is a simple circle with radial symmetry. This circle is formed because all the iterated complex numbers lead to or diverge from the center point of the circle.

Other $\boldsymbol{c}$ values can produce fractal patterns with more complex geometric shapes but are still organized and symmetrical. For example in figure (d), when

c = -1, the fractal pattern formed resembles a flower with several petals arranged symmetrically. This pattern still has a certain regularity and symmetry, although it is more complex than a simple circle.

Even small changes in the value of ***c*** can result in large changes in the fractal structure of the Julia Set formed. This indicates a high sensitivity to the initial parameters and reflects the typical nature of the non-linear dynamical system underlying fractal formation. By understanding the effect of choosing different ***c*** values, researchers can explore and create a wide variety of Julia Set fractal patterns, ranging from simple shapes to very complex and irregular shapes.

2. Same ***c*** parameters ($c = 0{,}359 + 0{,}599i$) with different iterations:

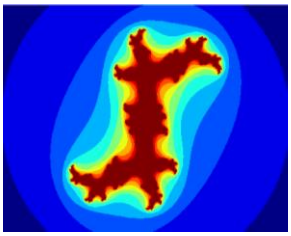


Figure 2. Julia Set with 10 iterations.

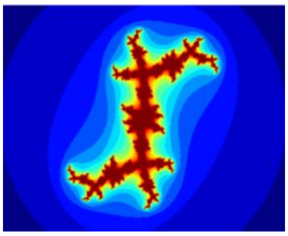


Figure 3. Julia Set with 15 iterations.

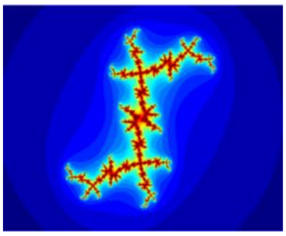


Figure 4. Julia Set with 25 iterations.

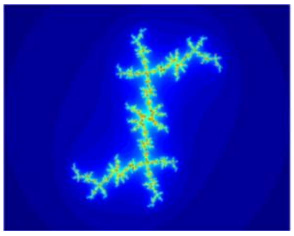


Figure 5. Julia Set with 50 iterations.

From the visualization results, there is a significant change in the fractal pattern of Julia Set as the number of iterations increases. As the number of iterations increases, the resulting fractal structure becomes more complex and the geometric details become finer. Increasing the number of iterations allows the appearance of previously unseen fractal details, revealing the complexity inherent in the mathematical system underlying the formation of Julia Set fractals.

At first, with a low number of iterations such as figure 2 with 10 iterations, the resulting Julia Set fractal pattern appears simple and only displays the main structure or basic framework. However, when the number of iterations is increased as in figure 3, figure 4, and figure 5, new details start to appear in the fractal structure. The more iterations performed, the finer and more complex the geometric details produced.

This phenomenon demonstrates the self-similarity of fractal geometry, where the same pattern keeps repeating itself on a smaller and smaller scale as the iterations increase. Each part of the fractal structure is a scaled-down version of the overall structure, with the same details but at a smaller scale. Constant iteration reveals these details, creating increasingly complex and visually appealing fractal patterns..

This self-similarity property is a very important property of fractal geometry, if a part of the fractal object is enlarged in a certain scale, then the enlarged part of the fractal object will be similar to its overall shape (Wahyuningsih & Hernadi, 2020).

3. Same parameter **c** ($c = 0{,}359 + 0{,}599i$), same iteration 20 times, but with different functions $\boldsymbol{f_c = z^n + c}$, where $\boldsymbol{z, c} \in \mathbb{C}$ and $n = 1, 2, 3, \cdots$.

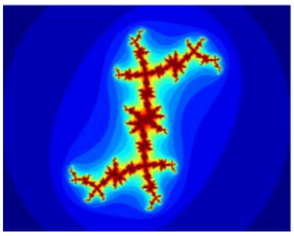


Figure 6. Julia Set with function $\boldsymbol{f_c = z^2 + c}$**.**

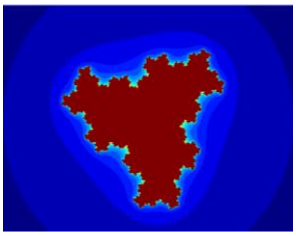


Figure 7. Julia Set with function $\boldsymbol{f_c = z^3 + c}$.

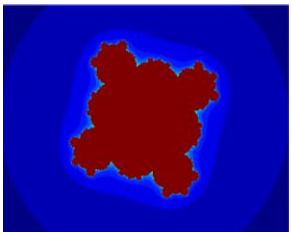


Figure 8. Julia Set with function $\boldsymbol{f_c = z^4 + c}$.

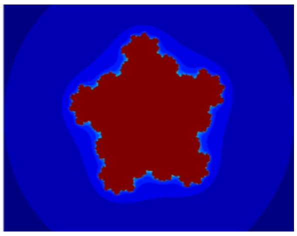


Figure 9. Julia Set with function $\boldsymbol{f_c = z^5 + c}$.

From this visualization, it can be seen that changes in the iteration function, namely the value of n in $\boldsymbol{f_c = z^n + c}$ can produce very diverse and complex Julia Set fractal patterns. The larger the value of $\boldsymbol{n}$, the more complex and random the resulting fractal structure, even with the same parameter $\boldsymbol{c}$ and number of iterations.

When the value of $\boldsymbol{n = 2}$ as in Figure 6, the iteration function becomes $\boldsymbol{f_c = z^2 + c}$, which is the standard function for Julia Set fractals. The resulting fractal pattern has a complex structure with branches that are interconnected and form irregular patterns. The resulting geometric details are already quite refined and visually appealing.

However, when the value of n increases to 3 as in Figure 7, the iteration function changes to $\boldsymbol{f_c = z^3 + c}$. The resulting fractal pattern becomes more complicated and unsymmetrical. There are more branches and finer details, creating a more complex and visually appealing pattern compared to the previous iteration of the function.

The larger the value of $\boldsymbol{n}$, the more the complexity of the resulting fractal pattern increases. When $\boldsymbol{n} = \mathbf{4}$ as in Figure 8, the iteration function becomes $\boldsymbol{f_c = z^4 + c}$, and the resulting fractal pattern has a very complex and random structure. There is a collection of spirals, circles and branches that are randomly interconnected, creating a very unique and visually appealing pattern. The resulting geometric details become more subtle and complex.

The complexity of the fractal pattern reaches its peak when $\boldsymbol{n} = \mathbf{5}$, with the iteration function $\boldsymbol{f_c = z^5 + c}$. The structure is very random and irregular, with branches that interconnect and form very intricate patterns. The geometric details are very fine and complex, creating a visually appealing fractal visualization..

This shows that the larger the value of $\boldsymbol{n}$ in the iteration function $\boldsymbol{f_c = z^n + c}$, the more complex and random the resulting Julia Set fractal pattern is. This happens even with the same parameter $\boldsymbol{c}$ and number of iterations.

CONCLUSION

Based on the research results obtained, it can be concluded that the Julia Set is an example of a fractal built from the iteration of a complex function with itself. Through visualization using Matlab, several results were obtained. First, the Julia Set has a very interesting visual variety, with patterns rich in color, shape, and complex geometric details. Second, the selection of different values of the constant $\boldsymbol{c}$ in the function $\boldsymbol{f_c = z^2 + c}$ will produce diverse fractal patterns of Julia Set, ranging from regular geometric shapes such as circles and flowers, to irregular patterns. Third, increasing the number of iterations will result in increasingly complex fractal structures and finer geometric details, showing the self-similarity property of fractal geometry. The results of this study show that the properties of complex numbers have a significant contribution in the formation of complex and visually appealing fractal patterns.

REFERENCES

Faseha, N., Helmi, & Kiftiah, M. (2019). Visualisasi Variasi Motif Songket Sambas Menggunakan Metode L-System dan Himpunan Julia. *Angewandte Chemie International Edition, 6(11), 951–952.*, *08*(4), 969–978.

Hasang, S., & Supardjo, S. (2012). Geometri fraktal dalam rancangan arsitektur. *Media Matrasain*, *9*(2), 111–124. https://ejournal.unsrat.ac.id/index.php/jmm/article/view/665

Heriyanto, H. (2018). Thematic Analysis sebagai Metode Menganalisa Data untuk Penelitian Kualitatif. *Anuva*, *2*(3), 317. https://doi.org/10.14710/anuva.2.3.317-324

Jaya, A. K., & Aliansa, N. (2017). Orbit Fraktal Himpunan Julia. *Jurnal Matematika, Statistika, Dan Komputasi*, *13*(2), 162–170.

Kodri, R. F., & Titaley, J. (2017). Variasi Motif Batik Minahasa Berbasis Julia Set. *Jurnal MIPA*, *6*(2), 81. https://doi.org/10.35799/jm.6.2.2017.17815

Palma, D. I., & Nadiasari, E. (2022). Tinjauan Filsafat Matematika Geometri Fraktal dan Implikasinya dalam Pembelajaran Matematika. *PRISMA: Prosiding Seminar Nasional Matematika*, *5*, 14–20. https://journal.unnes.ac.id/sju/index.php/prisma/

Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi

Rohmah, A. F., & Hernadi, J. (2020). Iterasi Fungsi Kuadrat Kompleks Dan Konstruksi Himpunan Julia. *Edupedia*, *4*(1), 55–64. https://doi.org/10.24269/ed.v4i1.429

Romadiastri, Y. (2013). *BATIK FRAKTAL : PERKEMBANGAN APLIKASI GEOMETRI*. 158–164.

Solar, F. Y., Titaley, J., & Rindengan, A. J. (2021). Penerapan Geometri Fraktal Dalam Membuat Variasi Motif Batik Nusantara Berbasis Julia Set. *D'CARTESIAN*, *9*(2), 189. https://doi.org/10.35799/dc.9.2.2020.29968

Wahyuningsih, S., & Hernadi, J. (2020). Sistem Fungsi Iterasi Dan Dimensi Fraktal Pada Himpunan Serupa Diri. *Euclid*, *7*(2), 108. https://doi.org/10.33603/e.v7i2.2941

Widodo, W. (2021). *Geometri Frakatal*. Yogyakarta: Gadjah Mada University Press