eISSN : 2621-4970
pISSN : 2621-4962

Volume 6 | No 1
Januari | **2023**

Konstruksi Pola Fraktal Berdasarkan Bentuk Dasar Persegi Menggunakan Transformasi *Affine*

Erwin Eko Wahyudi[1], Janoe Hendarto[2], Nur Rokhman[3], Andika Rahim Darusalam[4]
[1,2,3,4]Fakultas Matematika dan Ilmu Pengetahuan Alam, Departemen Ilmu Komputer dan Elektronika
Universitas Gadjah Mada
erwin.eko.w@ugm.ac.id

Abstrak

Geometri fraktal, juga dikenal sebagai "geometri alam", adalah jenis geometri yang mempelajari geometri tidak beraturan. Karakteristik utama dari geometri fraktal adalah *self-similarity*, yaitu bagian lain dari fraktal memiliki bentuk yang serupa pada skala yang berbeda. Penelitian ini bertujuan untuk membangun pola fraktal berdasarkan bentuk dasar persegi dan menggunakan dua jenis transformasi *affine*, yaitu dilatasi dan translasi. Parameter yang dapat diubah untuk transformasi adalah skala. Implementasi pembuatan program dilakukan dengan menggunakan bahasa pemrograman Python. Dengan membandingkan hasil dari enam iterasi untuk skala 0,5 dan 0,45, diperoleh perbedaan secara visual baru terlihat jelas dari iterasi 3.
Kata kunci: geometri fraktal, transformasi, persegi

Abstract

Fractal geometry, also known as "natural geometry", is a type of geometry that studies irregular geometries. The main characteristic of fractal geometry is self-similarity, i.e. other parts of the fractal have a similar shape at different scales. This study aims to build a fractal pattern based on a basic shape of a square and use two types of affine transformations, which are dilation and translation. The parameter that can vary for the transformation is the scale. The implementation of making the program is carried out using the Python programming language. By comparing the results of the six iterations for a scale of 0.5 and 0.45, the visual differences are only clearly visible from iteration 3.
Keywords: ***fractal geometry, transformation, square***

PENDAHULUAN

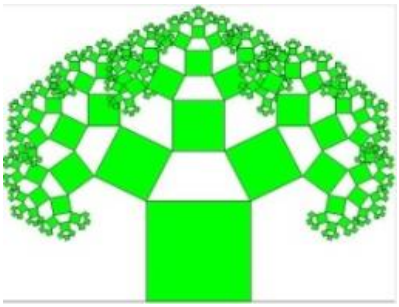
Banyak bentuk-bentuk di alam yang terbentuk dari unsur-unsur geometri seperti garis, lingkaran, poligon, dan lainnya. Bentuk-bentuk tersebut dapat dinyatakan dalam persamaan matematika dalam konsep geometri klasik. Namun, terdapat juga bentuk-bentuk yang tidak beraturan sehingga tidak mudah untuk dituliskan dalam suatu persamaan geometri klasik. Salah satu cara untuk memodelkan bentuk-bentuk yang tidak beraturan adalah dengan menggunakan geometri fraktal (Garg et al., 2014). Geometri fraktal, juga dikenal sebagai "geometri alam", adalah jenis geometri yang mempelajari geometri tidak beraturan. Geometri tidak beraturan mengacu pada aturan yang terlihat kacau dan mengikuti hukum internal tertentu, seperti garis pantai yang berkelok-kelok dan pegunungan yang bergulir (Tian et al., 2019).

Penelitian terkait geometri fraktal bermula pada 1918 yang mana seorang matematikawan dari Prancis bernama Gaston Maurice Julia, melakukan observasi terkait dengan iterasi dari suatu polinomial pada bilangan kompleks. Dari proses iterasi tersebut didapatkan bentuk visual dari bilangan kompleks yang terbentuk, dan bentuk yang diperoleh dari proses ini dinamakan *Julia Set* (Mahanta et al., 2016). Hal ini selanjutnya dikembangkan menjadi Mandelbrot Set oleh Benoit Mandelbrot.

Geometri fraktal adalah cabang matematika yang mempelajari sifat-sifat dan perilaku berbagai jenis fraktal. Berbagai jenis fraktal pada awalnya dipelajari sebagai benda-benda matematis yang dapat diukur dengan perhitungan matematis biasa. Ada banyak bentuk matematis yang merupakan bentuk fraktal, misalnya seperti *Sierpinski triangle*, *Koch snowflake*, Peano *curve*, *Mandelbrot set*, dan *Lorenz attractor* (Romadiastri, 2017).

Contoh penggunaan dari geometri fraktal adalah pembentukan pola batik, seperti yang dilakukan oleh Anggraini (2019). Pola-pola geometri fraktal yang digunakan (segitiga Sierpinski, kurva Hilbert/Peano, Koch *Snowflake*, dan himpunan Mandelbrot) yang telah dibangkitkan sampai beberapa iterasi kemudian ditransformasikan, dan selanjutnya disatukan dengan menggunakan aplikasi sehingga terbentuk motif batik sekar jagad. Adapun contoh lainnya adalah penggunaan geometri fraktal pada bidang arsitektur, seperti pada artikel yang ditulis oleh Yazyeva et al. (2019). Para arsitek secara intuitif menggunakan prinsip-prinsip fraktal dalam bangunan mereka karena sensasi keindahan diberikan kepada mereka pada tingkat genetik dan mereka merasakan secara spiritual keindahan alam di sekitar mereka. Prinsip fraktal lebih sering diwujudkan dalam permukaan fraktal bangunan yang membentuk fasad dan elemen dekoratifnya, tetapi fraktal juga dapat diamati dalam organisasi spasial (internal dan eksternal) bangunan dan struktur.

Pola fraktal yang dibentuk berdasarkan satu bentuk dasar sangat menarik untuk diteliti. Salah satu upaya untuk membentuk pola fraktal adalah mengusulkan pola berikutnya atau aturan fraktal, dan dicari transformasi yang memenuhi aturan yang diinginkan. Transformasi yang dapat digunakan adalah dilatasi, rotasi, atau translasi. Gambar *1* adalah contoh pola fraktal yang disusun berdasarkan transformasi rotasi dan dilatasi.



Gambar 1 Fraktal dengan transformasi rotasi dan dilatasi (Purnomo et al., 2022)

Rumusan Masalah

Permasalahan yang dibahas dalam penelitian ini adalah bagaimana pola fraktal dapat dibentuk berdasarkan bentuk dasar persegi beserta pencarian transformasi yang sesuai berdasarkan pola yang diusulkan.

Tujuan Penelitian

Tujuan dari penelitian adalah untuk membuat program yang dapat digunakan untuk proses konstruksi pola fraktal berdasarkan parameter yang diinginkan oleh pengguna. Bentuk geometri dasar yang

akan digunakan dalam membentuk pola adalah persegi, dengan parameter yang dapat diubah untuk transformasi adalah skala.

Manfaat Penelitian

Manfaat dari penelitian ini adalah memperkaya literatur terkait konstruksi pola fraktal beserta transformasi yang digunakan.

Tinjauan Pustaka

Beberapa pola fraktal yang populer adalah Koch Snowflake, Segitiga Sierpinski, dan Mandelbrot Set. Beberapa penelitian yang berdasarkan pola-pola tersebut telah dilakukan.

Purnomo (2014) membuat pola Segitiga Sierpinski dengan menggunakan beberapa macam benda geometris segiempat sebagai dasarnya. Adapun transformasi yang digunakan adalah dilatasi dan translasi. Segitiga yang dihasilkan juga bisa berupa segitiga siku-siku ataupun segitiga sama kaki.

Selain Segitiga Sierpinski, pola Koch Snowflake dapat dimanfaatkan untuk membuat desain batik (Riwansia, 2016). Pola Koch Snowflake akan disusun sedemikian sehingga membentuk desain batik, seperti ornamen di sisi atas, bawah, kiri, dan kanan dari Koch Snowflake utama. Desain batik tersebut kemudian akan diberi warna sesuai dengan keinginan pengguna. Implementasi dari pembuatan desain batik menggunakan perangkat lunak Matlab.

Anggraini (2019) membangkitkan pola-pola fraktal yang sudah populer, seperti segitiga Sierpinski, kurva Hilbert/Peano, Koch Snowflake, dan himpunan Mandelbrot, dengan menggunakan beberapa iterasi. Hasil dari pembangkitan pola tersebut disatukan dengan menggunakan aplikasi sehingga terbentuk pola batik sekar jagad. Implementasi dari pembangkitan pola fraktal menggunakan perangkat lunak Maple.

Purnomo et al. (2022) membuat pola pohon fraktal tiga cabang, yang terinspirasi dari pohon Pythagoras. Contoh dari hasil dari penelitian ini dapat dilihat pada Gambar *1*.

Beberapa penelitian tidak menggunakan pola-pola terkenal yang telah disebutkan pada paragraf sebelumnya, seperti penelitian yang dilakukan oleh Yuan et al. (2019) dan Tian et al. (2019).

Yuan et al. (2019) membuat pola fraktal yang membentuk kupu-kupu. Elemen dasar bentuk kupu-kupu yang akan dihasilkan didasarkan pada kurva Rhodonea atau mawar yang merupakan sinusoidal yang diplot dalam koordinat kutub.

Tian et al. (2019) membuat pola fraktal yang membentuk bunga. Elemen dasar yang digunakan dibagi berdasarkan bagian yang terdapat pada bunga, yaitu daun bunga, kelopak, putik, dan inti bunga. Masing-masing elemen memiliki persamaan matematis.

Implementasi dari penelitian Yuan et al. (2019) menggunakan perangkat lunak Matlab, sedangkan impementasi dari penelitian Tian et al. (2019) menggunakan bahasa pemrograman Python.

LANDASAN TEORI

Geometri Fraktal

Geometri fraktal adalah jenis geometri yang mempelajari geometri tidak beraturan. Fraktal dibagi ke dalam dua jenis, yaitu *fractal sets* (himpunan-himpunan fraktal) dan *natural fractal* (fraktal alami) (Hasang

& Suparjo, 2012). Selanjutnya, himpunan fraktal diklasifikan ke dalam dua tipe, yaitu linier dan non-linier (Navarro et al., 2014) Fraktal linier mempunyai bentuk geometris yang sama persis pada skala berapapun dan pada iterasi berapapun (tak-hingga). Contoh fraktal linier adalah Segitiga Sierpinski dan kurva Koch. Sedangkan fraktal non-linier dapat dibangkitkan dengan menggunakan fungsi dinamik non-linier. Himpunan Mandelbrot dan himpunan Julia merupakan beberapa contoh fraktal non-linier. Berbeda dengan himpunan fraktal, fraktal alami merupakan bentuk yang berada di alam, seperti pohon, daun, pakis, gunung, garis pantai (Purnomo et al., 2022).

Karakteristik utama dari geometri fraktal adalah *self-similarity*, yaitu bagian lain dari fraktal memiliki bentuk yang serupa pada skala yang berbeda (Riwansia, 2016). Selain itu, Purnomo et al. (2022) menyebutkan bahwa sifat lain dari fraktal adalah *infinite detail*, yang artinya apabila objek diperbesar maka detail-detailnya akan terlihat dengan jelas.

Transformasi *Affine*

Dalam Byrd (1999), sebuah transformasi *affine* dinyatakan dalam Persamaan (1),

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \quad (1)$$

dengan (x, y) merupakan titik awal, (x', y') merupakan titik hasil transformasi, dan nilai-nilai a, b, c, d, e, dan f merupakan sebarang bilangan real yang menyusun transformasi *affine*.

Terdapat beberapa macam transformasi *affine*, di antaranya adalah dilatasi dan translasi. Dilatasi adalah transformasi yang mengubah ukuran bentuk awal tanpa mengubah bentuk aslinya, seperti memperbesar atau memperkecil dengan skala tertentu. Sedangkan translasi merupakan transformasi yang mengubah posisi dari bentuk awal tanpa mengubah ukuran bentuk aslinya.

Metode *Iterated Function System* (IFS)

Metode *Iterated Function System* (IFS) merupakan salah satu metode yang paling umum digunakan untuk membuat pola fraktal. Metode ini dipopulerkan oleh Barsnley et al. (1988).

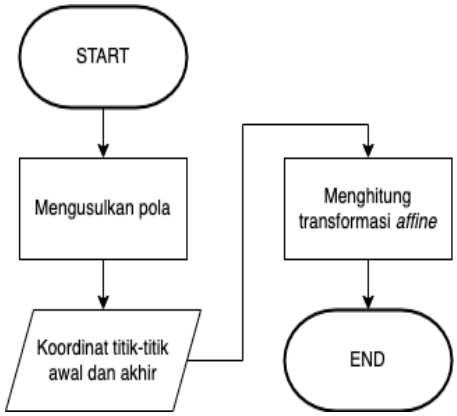
Terdapat n buah transformasi yang digunakan, yaitu w_1, w_2, ..., w_n, yang mana umumnya masing-masing w_i berbentuk transformasi *affine*. Transformasi ini akan diaplikasikan pada pola di iterasi ke-i untuk membentuk pola di iterasi ke-(i+1), sehingga dinamakan metode IFS. Transformasi yang digunakan merupakan *contracting transformation*, yaitu transformasi yang menyebabkan jarak dari hasil transformasi menjadi lebih kecil dibandingkan jarak aslinya (Byrd, 1999).

Terdapat dua jenis metode IFS, yaitu deterministik dan acak. Metode IFS deterministik yaitu semua transformasi akan diterapkan pada setiap titik di gambar awal, tanpa perlu ada pemilihan transformasi mana yang akan digunakan. Sedangkan metode IFS acak hanya memilih satu transformasi yang akan diaplikasikan pada satu titik pada gambar awal.

METODE PENELITIAN

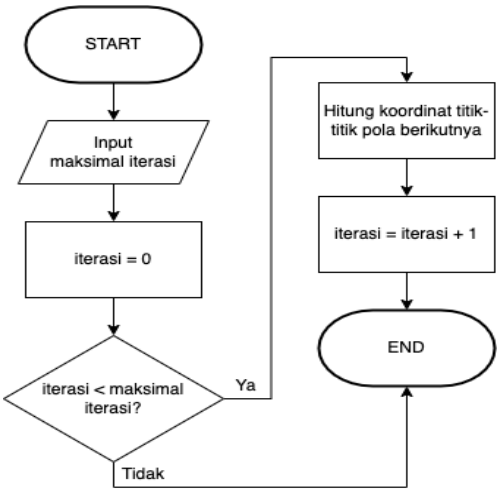
Penelitian ini memiliki beberapa tahapan. Tahapan pertama dimulai dari menentukan pola fraktal, dan dilanjutkan dengan menentukan letak koordinat-koordinat titik ujung dari bentuk awal dan bentuk akhir. Setelah diperoleh titik-titik koordinat dari kedua bentuk, akan dicari

transformasi *affine* yang sesuai. Transformasi *affine* yang telah diperoleh akan digunakan dalam penggunaan metode IFS deterministik. Gambar *2* menyatakan ringkasan dari tahapan pertama penelitian.



Gambar 2 Tahapan pertama penelitian

Tahapan kedua dalam penelitian ini adalah pembuatan pola berdasarkan iterasi yang diinputkan pengguna. Koordinat dari titik-titik awal sudah ditentukan di awal, dan juga transformasi *affine* sudah diperoleh dari tahapan pertama penelitian. Gambar *3* menyatakan ringkasan dari tahapan kedua penelitian.



Gambar 3 Tahapan kedua penelitian

ANALISIS DAN PERANCANGAN

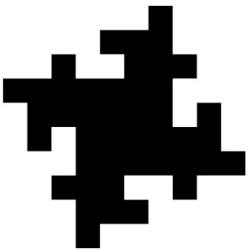
Pengusulan Pola

Proses penelitian dimulai dari pengusulan pola fraktal yang akan dibuat. Berikut adalah usulan polanya:

- Iterasi 0, bentuk awal

- Iterasi 1, sebuah persegi akan ditransformasikan menjadi empat persegi baru, dengan panjang sisi yang baru berukuran 0,5 kali panjang sisi yang lama, dan diletakkan di masing-masing sisi sehingga membentuk mirip gasing. Transformasi ini memanfaatkan dua jenis transformasi *affine*, yaitu dilatasi dan translasi.

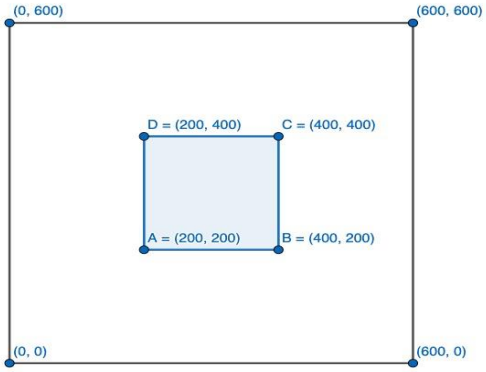
- Iterasi 2, masing-masing persegi akan ditransformasikan lagi menjadi empat persegi baru.



Penentuan Koordinat

Diasumsikan bahwa ukuran layar yang digunakan untuk menampilkan pola adalah 600 x 600. Apabila digambarkan dalam bentuk koordinat, maka layar akan memiliki koordinat (0,0) hingga (600,600). Kemudian, diasumsikan pula ukuran bentuk awal adalah 200 x 200, dan terletak di tengah-tengah layar. Sehingga, koordinat

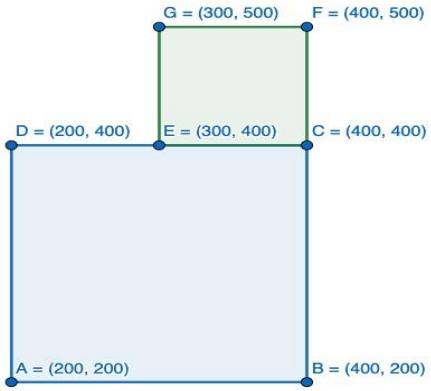
dari bentuk awal adalah (200,200) hingga (400,400). Ilustrasi koordinat dapat dilihat pada Gambar *4*, dengan persegi ABCD merupakan bentuk awal.



Gambar 4 Koordinat layar dan bentuk awal

Perhitungan Transformasi *Affine*

Berdasarkan pola yang diusulkan, terdapat empat transformasi *affine* yang harus dihitung. Dipilih salah satu dari empat persegi baru untuk dihitung transformasinya, misalkan persegi atas. Gambar *5* merupakan koordinat pola pada iterasi 1 untuk persegi atas.



Gambar 5 Koordinat pola pada iterasi 1 untuk persegi atas

Pada Gambar *5*, bentuk awal persegi ABCD akan ditransformasikan menjadi persegi ECFG, dengan pemetaan sebagai berikut:

- titik A(200,200) ke titik E(300,400),
- titik B(400,200) ke titik C(400,400),
- titik C(400,400) ke titik F(400,500), dan
- titik D(200,400) ke titik G(300,500).

Setelah menentukan koordinat dari bentuk awal ABCD dan bentuk hasil transformasi ECFG, keempat pasangan koordinat ini bisa digunakan untuk menghitung nilai-nilai *a*, *b*, *c*, *d*, *e*, dan *f*, pada Persamaan (1). Dengan menyelesaikan sistem persamaan linear, diperoleh

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0{,}5 & 0 \\ 0 & 0{,}5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 200 \\ 300 \end{bmatrix}. \quad (2)$$

Dengan cara yang sama, dapat diperoleh pula persamaan untuk tiga transformasi *affine* lainnya:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0{,}5 & 0 \\ 0 & 0{,}5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 100 \\ 0 \end{bmatrix}, \quad (3)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0{,}5 & 0 \\ 0 & 0{,}5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 200 \end{bmatrix}, \quad (4)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0{,}5 & 0 \\ 0 & 0{,}5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 300 \\ 100 \end{bmatrix}. \quad (5)$$

Persamaan (2)-(5) merupakan empat transformasi *affine* yang diperlukan untuk membuat pola pada iterasi berikutnya. Keempat persamaan ini merupakan transformasi yang memiliki skala bernilai tetap, yaitu 0,5.

Perhitungan Transformasi *Affine* untuk Skala Bebas

Agar pembuatan pola lebih dinamis, nilai skala dibuat sebagai parameter. Persamaan (2)-(5) dimodifikasi agar skala bisa bernilai tidak hanya 0,5. Hasil dari modifikasi Persamaan (2)-(5) dapat dilihat pada Persamaan (6)-(9),

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 200 - 200s \\ 200 - 400s \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 200 - 400s \\ 400 - 400s \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 400 - 200s \\ 200 - 200s \end{bmatrix} \quad (8)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 400 - 400s \\ 400 - 200s \end{bmatrix} \quad (9)$$

dengan *s* menyatakan nilai skala.

IMPLEMENTASI & PEMBAHASAN

Bahasa pemrograman yang digunakan dalam penelitian ini adalah Python, dengan bantuan pustaka numpy (Harris et al., 2020) dan matplotlib (Hunter, 2007). Dalam proses implementasi, terdapat beberapa proses yang dimulai dari deklarasi variabel. Kode untuk proses deklarasi variabel dapat dilihat pada Gambar *6*. Variabel `layar` dan `layar_copy` merupakan *array* berukuran 600x600 dan digunakan untuk menyimpan nilai pixel, `iter` dan `max_iter` digunakan untuk melakukan perulangan, variabel `scale` digunakan untuk menyimpan skala, dan variabel `a`, `b`, `c`, `d`, `e`, dan `f` merupakan *array* dengan 4 elemen dan digunakan untuk menyimpan nilai dari transformasi *affine* yang digunakan.

```
import numpy as np

layar = np.zeros((600, 600))
layar_copy = np.zeros((600, 600))
iter = 0
max_iter = 2
scale = 0.5
a = [0]*4
b = [0]*4
c = [0]*4
d = [0]*4
e = [0]*4
f = [0]*4
```

Gambar 6 Kode untuk deklarasi variable

Proses berikutnya adalah inisialisasi variabel, yaitu memberikan nilai awal pada variabel. Kode untuk proses inisialisasi variabel dapat dilihat pada Gambar *7*. Variabel `layar` akan memiliki nilai 0 di semua pixel, kecuali di pixel antara koordinat (200,200) hingga (400,400) memiliki nilai 1.

```
for i in range(600):
  for j in range(600):
    if i>=200 and i<=400 and j>=200
and j<=400:
      layar[i,j] = 1
    else:
      layar[i,j] = 0
```

Gambar 7 Kode untuk inisialisasi variabel

Setelah variabel dideklarasi dan diinisialisasi, nilai dari `max_iter` dan `scale` dibaca berdasarkan masukan dari *user*. Nilai dari variabel `scale` akan digunakan untuk menentukan nilai-nilai dari keempat transformasi *affine*. Kode untuk proses ini dapat dilihat pada Gambar *8*.

```
scale = float(input("Masukkan
ukuran skala: "))
max_iter = int(input("Masukkan
maksimal perulangan: "))

a[0] = a[1] = a[2] = a[3] = scale
b[0] = b[1] = b[2] = b[3] = 0
c[0] = c[1] = c[2] = c[3] = 0
d[0] = d[1] = d[2] = d[3] = scale

e[0] = 200 - 200*scale
f[0] = 200 - 400*scale

e[1] = 200 - 400*scale
f[1] = 400 - 400*scale

e[2] = 400 - 200*scale
f[2] = 200 - 200*scale

e[3] = 400 - 400*scale
f[3] = 400 - 200*scale
```

Gambar 8 Kode untuk membaca input dan transformasi

Proses utama dalam penelitian ini adalah pembuatan pola. Kode untuk proses ini dapat dilihat pada Gambar *9*.

```
while iter < max_iter:
  layar_copy = layar.copy()
  for i in range(600):
    for j in range(600):
      if layar[i,j] == 1:
        for t in range(4):
          ii = int(a[t]*i + b[t]*j
+ e[t])
          jj = int(c[t]*i + d[t]*j
+ f[t])
          layar_copy[ii,jj] = 1

  layar = layar_copy.copy()
  iter += 1
```

Gambar 9 Kode untuk membuat pola

Setelah pola berhasil dibuat sesuai dengan iterasinya, hal terakhir yang dilakukan adalah menampilkan pola. Kode untuk menampikan pola dapat dilihat pada Gambar *10*.

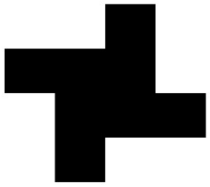
```
from matplotlib import pyplot as
plt

plt.imshow(layar, cmap='gray_r')
plt.show()
```

Gambar 10 Kode untuk menampilkan pola

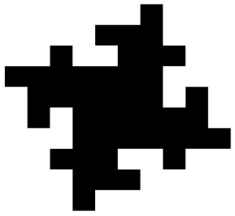
Berikut adalah hasil yang diperoleh untuk nilai skala 0,5:

- Bentuk awal
  Bentuk dasar yang digunakan adalah persegi.

- Iterasi 1
  Terdapat empat persegi kecil di setiap sisinya, yang masing-masing sisinya memiliki panjang 0,5 dari panjang sisi awal.



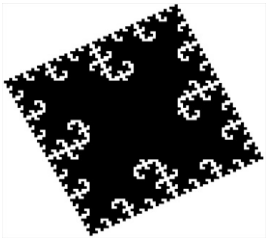
- Iterasi 2
  Setiap persegi akan memiliki empat persegi baru pada masing-masing sisinya, namun pada iterasi ini hanya terlihat tiga persegi baru karena satu persegi baru beririsan dengan persegi lama. Panjang sisi dari persegi baru adalah $0{,}5^2$ dari panjang sisi awal.



- Iterasi 3
  Sama seperti pola di iterasi 2, hanya terlihat 3 persegi baru dengan panjang sisinya $0{,}5^3$ dari panjang sisi awal.

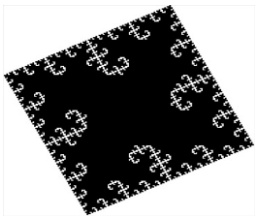
- Iterasi 4
  Sama seperti pola di iterasi 2 dan 3, hanya terlihat 3 persegi baru dengan panjang sisinya $0{,}5^4$ dari panjang sisi awal.

- Iterasi 5
  Panjang sisi dari persegi baru semakin kecil, yaitu $0{,}5^5$ dari panjang sisi awal.



- Iterasi 6

Panjang sisi dari persegi baru lebih kecil dibandingkan pola pada iterasi 5, yaitu $0{,}5^6$ dari panjang sisi awal.

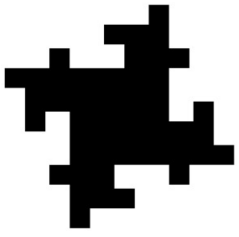


Selanjutnya, hasil untuk skala 0,5 akan dibandingkan dengan hasil untuk skala 0,45. Berikut adalah hasil yang diperoleh untuk nilai skala 0,45:

- Bentuk awal
  Bentuk dasar yang digunakan adalah persegi.

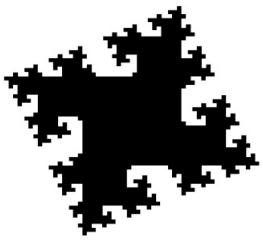
- Iterasi 1
  Terdapat empat persegi kecil di setiap sisinya, yang masing-masing sisinya memiliki panjang 0,45 dari panjang sisi awal.

- Iterasi 2
  Setiap persegi akan memiliki empat persegi baru pada masing-masing sisinya, namun pada iterasi ini hanya terlihat tiga persegi baru karena satu persegi baru beririsan dengan persegi lama. Panjang sisi dari persegi baru adalah $0{,}45^2$ dari panjang sisi awal.

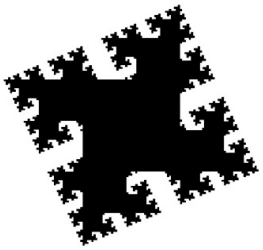


- Iterasi 3
  Sama seperti pola di iterasi 2, hanya terlihat 3 persegi baru dengan panjang sisinya $0{,}45^3$ dari panjang sisi awal.

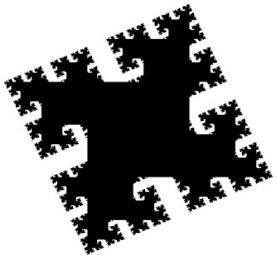
- Iterasi 4
  Sama seperti pola di iterasi 2 dan 3, hanya terlihat 3 persegi baru dengan panjang sisinya $0{,}45^4$ dari panjang sisi awal.



- Iterasi 5
  Panjang sisi dari persegi baru semakin kecil, yaitu $0{,}45^5$ dari panjang sisi awal.



- Iterasi 6
  Panjang sisi dari persegi baru lebih kecil dibandingkan pola pada iterasi 5, yaitu $0{,}45^6$ dari panjang sisi awal.



Dengan menganalisis pola secara visual, pola di iterasi 1 dan iterasi 2 tidak memiliki perbedaan yang mencolok meskipun menggunakan skala yang sedikit berbeda, yaitu 0,5 dan 0,45. Namun, pada iterasi 3 hingga 6, perbedaan pola semakin terlihat jelas. Pada skala 0,45, pembuatan persegi baru menjadi semakin kecil dan daerah putih akan lebih luas, dibandingkan dengan skala 0,5.

KESIMPULAN

Pola fraktal berdasarkan bentuk awal persegi telah berhasil dibuat. Tidak hanya skala 0,5, namun nilai skala bisa dinamis berdasarkan *input* dari *user*. Telah dicontohkan pola dari bentuk awal hingga iterasi 6 untuk skala 0,5 dan skala 0,45, yang mana perbedaan secara visual baru terlihat jelas dari iterasi 3.

Saran

Pada penelitian ini, jenis transformasi yang dibahas hanyalah dilatasi dan translasi. Jenis transformasi yang lain juga dapat diterapkan pada pembuatan pola fraktal, seperti rotasi dan *shear*.

DAFTAR PUSTAKA

Anggraini, L.D.F. (2019). *Geometri Fraktal dan Transformasi Geometri Sebagai Dasar Pengembangan Motif Batik Sekar Jagad*. Transformasi: Jurnal Pendidikan Matematika dan Matematika, 3, 1, 1-14.

Barnsley, M.F., Devaney, R.L., Mandelbrot, B.B., Peitgen, H., Saupe, D., Voss, R.F. (1988). *The Science of Fractal Images*. Berlin: Springer.

Byrd, C. (1999). *Fractals in Nature: An introduction to IFS and L-System Fractals*. https://web.cs.wpi.edu/~matt/courses/cs563/talks/cbyrd/pres1.html, diakses pada 23 November 2022.

Garg, A., Agrawal, A., & Negi, A. (2014). *A Review on Natural Phenomenon on Fractal Geometry*. International Journal of Computer Applications, 86, 4, 1-7.

Hasang, S. & Suparjo, S. (2012). *Geometri Fraktal dalam Rancangan Arsitektur*. Media Matrasain, 9, 1, 111-124.

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. (2020). *Array programming with NumPy*. Nature, 585, 357–362.

Hunter, J.D. (2007). *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 9, 3, 90-95.

Mahanta, A., Sarmah, H., Paul, R. & Choudhury, G. (2016). *Julia Set and Some of Its Properties*. International Journal of Applied Mathematics & Statistical Sciences (IJAMSS), 5, 2, 99-124.

Navarro, C.F., Garcia, J.C., & Tavares, M.M. (2014). *Main Objects of Fractal Geometry and Computer Graphical Generation*. Research Journal of Computation and Mathematics, 2, 2, 14-26.

Purnomo, K.D. (2014). *Pembangkitan Segitiga Sierpinski dengan Transformasi Affine*. Prosiding Seminar Nasional Matematika Jurusan Matematika FMIPA Universitas Jember, 365-375.

Purnomo, K.D., Wahyuningtyas, D., & Ubaidillah, F. (2022). *Pembangkitan Pohon Fraktal Tiga Cabang dengan Metode Iterated Function System*. Jurnal ILMU DASAR, 23, 1, 9-16.

Riwansia, R.R. (2016). *Pengembangan Desain Batik melalui Penggunaan Geometri Fraktal Koch Snowflake*. (Skripsi). Universitas Jember, Jember.

Romadiastri, Y. (2017). *Batik Fraktal: Perkembangan Aplikasi Geometri Fraktal*. Delta: J. Ilm. Pendidik. Matematika, 1, 158–164.

Tian, G., Yuan, Q., Hu, T., & Shi, Y. (2019). *Auto-Generation System Based on Fractal Geometry for Batik*

Pattern Design. Applied Sciences, 9, 11, 2383.

Yazyeva, S., Mayatskaya, I., Kashina, I. & Nesterova, A. (2019). *The manifestation of fractality in the architecture of buildings and structures*. IOP Conference Series: Materials Science and Engineering, 698, 033046.

Yuan, Q., Lv, J. & Huang, H. (2019). *Auto-Generation Method of Butterfly Pattern of Batik Based on Fractal Geometry*. International Journal of Signal Processing, Image Processing and Pattern Recognition, 9, 4, 369-392.