

## Pembangkitan Pohon Fraktal Tiga Cabang dengan Metode Iterated Function System

*Construction of Three Branches Fractal Trees Using Iterated Function System*

Kosala Dwidja Purnomo<sup>\*</sup>), Dita Wahyuningtyas, Firdaus Ubaidillah

*Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Jember*

<sup>\*</sup>E-mail: kosala.fmipa@unej.ac.id

### ABSTRACT

There are two types of fractal: natural fractals and fractals set. The examples of natural fractals are trees, leaves, ferns, mountain, and coastlines. One of the examples of fractals set is Pythagorean tree. In the earlier study, the Pythagorean tree has two branches generated through several affine transformations, i.e dilation and rotation. Here, we developed the Pythagorean tree (or fractal tree) with three branches through dilation, translation, and rotation transformation using Iterated Function System (IFS) method. Some values of height and length parameters were selected to ensure the formation of a fractal tree. These parameters affected the branching angle that can result in different fractal tree shape. Some random values of height and length parameters produced several variations of fractal tree. These values influenced the shape of fractal whether it tended to the left, to the right, or symmetrical shape.

**Keywords:** fractals, fractal tree with three branches, Iterated Function System, affine transformation.

### PENDAHULUAN

Fraktal mempunyai sifat *self-similarity* atau keserupaan diri, yaitu berarti objek fraktal yang dibangun berulang pada skala yang memiliki ukuran lebih kecil akan berbentuk serupa dengan bagian lainnya pada skala lebih besar dari objek aslinya. Sifat lain dari fraktal adalah *infinite detail* yang artinya apabila objek diperbesar maka detail-detailnya akan terlihat dengan jelas. Hasang & Suparjo (2012) menyebutkan bahwa fraktal dibagi menjadi dua jenis, yaitu himpunan-himpunan fraktal (*fractal sets*) dan fraktal alami (*natural fractal*). Navarro, *et al* (2014) mengklasifikasikan himpunan fraktal ke dalam dua tipe, yaitu linier dan non-linier. Fraktal linier mempunyai bentuk geometris yang identik pada skala berapapun sampai iterasi tak-hingga. Segitiga Sierpinski dan kurva Koch adalah contoh fraktal linier.

Fraktal non-linier dapat dibangkitkan melalui fungsi dinamik non-linier. Contoh fraktal non-linier adalah himpunan Mandelbrot dan himpunan Julia. Kedua himpunan tersebut dibangkitkan dari fungsi kuadrat dari variabel kompleks. Fraktal alami meliputi pohon, daun, pakis, gunung, garis pantai. Salah satu pohon yang dapat dibangkitkan dan divisualisasikan dalam fraktal adalah pohon Pythagoras. Matam (2012) menyebutkan bahwa pohon Pythagoras pertama kali dibangun oleh Albert E. Bosman pada tahun 1942. Nama untuk fraktal ini berasal dari fakta bahwa bentuk penggantian secara tradisional digunakan untuk menggambarkan

Teorema Pythagoras. Jenis-jenis fraktal tersebut dapat dibangkitkan melalui metode *Iterated Function System*.

Terdapat beberapa penelitian dalam membangun suatu objek fraktal terkait dengan metode *Iterated Function System* (IFS) atau suatu fungsi iterasi yang terdiri dari sekumpulan transformasi affine, diantaranya Purnomo (2014) membangkitkan segitiga Sierpinski dengan memanfaatkan transformasi affine dalam bentuk dilatasi dan translasi pada segitiga sama sisi sebagai bentuk dasar. Modifikasi tersebut selanjutnya mengganti bentuk dasar segitiga sama sisi dengan benda geometris segiempat. Prasasti (2018) memanfaatkan metode IFS dalam mengembangkan motif anyaman menggunakan beberapa transformasi affine. Kemudian Ubaidillah *et al* (2019) telah membangkitkan berbagai bentuk pohon Pythagoras dengan memanfaatkan metode IFS menggunakan beberapa transformasi affine dilatasi dan rotasi.

Beck, *et al* (2015) telah memodifikasi pohon Pythagoras dengan melakukan pengembangan konsep pohon Pythagoras melalui percabangan lebih dari dua dan sudut percabangan yang divariasikan. Algoritma berdasar pada variasi sudut dan ukuran persegi pembangkitnya dan diimplementasikan menggunakan bahasa C++. Sebelumnya Browne (2007) juga memberikan alternatif pembangkitan percabangan pohon Pythagoras berdasar algoritma Greedy. Dengan algoritma

tersebut dapat diminimalisasi terjadinya tumpukan segmen garis pembentuk pohon Pythagoras. Hal ini ditunjukkan dengan dimensi fraktal yang secara umum lebih besar dibanding dengan pohon Pythagoras standar. Dalam hal ini dimensi fraktal dihitung menggunakan metode *box counting*. Teia (2015, 2016) menggunakan bilangan triple Pythagoras melalui teori “central squares” untuk menjelaskan bagaimana pertumbuhan cabang pohon Pythagoras dan membuat anatomi pelabelan triple Pythagoras untuk memudahkan identifikasi persegi pada setiap iterasi.

Terkait dengan penerapan pohon Pythagoras, Ambarsari (2019) telah menggunakan konsep dasar pohon Pythagoras untuk melakukan konstruksi visualisasi data pada analisis data mining saat mempelajari pola pada big data. Sharma, *et al* (2015) juga telah mengimplementasikan pohon Pythagoras sebagai bentuk antena yang terbukti secara teknis dapat berfungsi dengan baik.

Pada artikel ini pohon Pythagoras standar yang memiliki dua percabangan dikembangkan menjadi tiga percabangan yang selanjutnya disebut dengan pohon fraktal. Pohon fraktal dibangkitkan menggunakan metode IFS, yaitu melalui beberapa transformasi affine dilatasi, translasi, dan rotasi.

Menurut Kusno (2003) dilatasi adalah transformasi yang mengubah ukuran bangun, tetapi tidak mengubah bentuk. Dilatasi dengan pusat  $O(0,0)$  dan faktor skala  $k$  ditulis  $[O,k]$ , yaitu

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (1)$$

Dilatasi dengan pusat  $A(a,b)$  dan faktor skala  $k$  ditulis  $[A(a,b),k]$ , yaitu

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix} \begin{bmatrix} x-a \\ y-b \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \quad (2)$$

Rotasi adalah suatu perpindahan benda pada gerakan melingkar. Pada dimensi dua, benda akan berputar pada pusat rotasi. Jika  $T: R^2 \rightarrow R^2$  adalah suatu transformasi yang memetakan titik  $(x,y)$  ke titik  $(x',y')$  dan misalkan  $\theta$  adalah sebuah sudut tetap maka persamaan rotasi melalui pusat  $P(a,b)$  dengan arah rotasi berlawanan arah jarum jam adalah

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x-a \\ y-b \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \quad (3)$$

dan persamaan rotasi dengan arah rotasi searah jarum jam adalah

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x-a \\ y-b \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \quad (4)$$

Translasi merupakan transformasi yang memetakan titik  $(x,y)$  ke  $(x',y')$  yaitu bergeser sejauh  $p$  satuan searah sumbu  $x$  dan  $q$  satuan searah sumbu  $y$ , sehingga didapatkan persamaan

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} x+p \\ y+q \end{bmatrix} \quad (5)$$

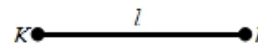
Persamaan (1) sampai (5) secara teknis akan digunakan dalam menghitung koordinat titik atau objek dengan transformasi affine.

## METODE

Tahap pertama membangkitkan persegi sebagai bentuk awal pada koordinat kartesius. Tahap kedua menentukan bentuk dasar percabangan dengan prosedur sebagai berikut:

### 1. Mencari panjang $l$

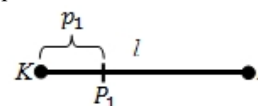
Titik  $K$  dan  $L$  didapatkan dari persegi awal,  $l$  adalah jarak antara titik  $K$  dan  $L$  seperti pada Gambar 1.



Gambar 1. Jarak antara titik  $K$  dan  $L$

### 2. Mencari panjang $p_1$

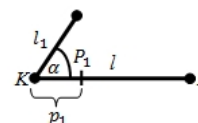
Posisi  $P_1$  ditentukan untuk mengatur panjang  $p_1$ . Jadi,  $p_1$  adalah panjang yang diperoleh dari titik  $K$  ke  $P_1$ . Ilustrasinya pada Gambar 2.



Gambar 2. Kedudukan titik  $P_1$  dan panjang  $p_1$

### 3. Mencari panjang $l_1$

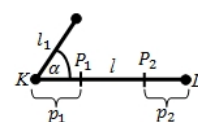
Panjang  $l_1$  didapat dari pembagian  $p_1$  dengan  $\sin(90 - \alpha)$ , letak sudut  $\alpha$  dan  $l_1$  seperti pada Gambar 3.



Gambar 3. Posisi sudut  $\alpha$  dan panjang  $l_1$

### 4. Mencari panjang $p_2$

Posisi  $P_2$  ditentukan untuk mengatur panjang  $p_2$ . Jadi,  $p_2$  adalah panjang yang diperoleh dari  $P_2$  ke titik  $L$ . Ilustrasinya pada Gambar 4.

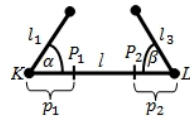


Gambar 4. Kedudukan titik  $P_2$  dan panjang  $p_2$

### 5. Mencari panjang $l_3$

Panjang  $l_3$  didapat dari pembagian  $p_2$  dengan  $\sin(90 - \beta)$ , letak sudut  $\beta$  dan  $l_3$  seperti pada

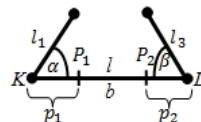
Gambar 5.



Gambar 5. Posisi sudut  $\beta$  dan panjang  $l_3$

6. Mencari panjang  $b$

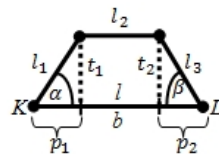
Panjang  $b$  didapat dari selisih  $l$  dengan penjumlahan dari  $p_1$  dan  $p_2$ ,  $b$  adalah panjang dari  $P_1$  ke  $P_2$ . Ilustrasinya seperti pada Gambar 6.



Gambar 6. Ilustrasi  $b$

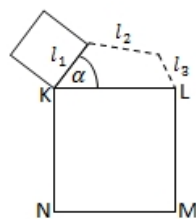
7. Mencari panjang  $l_2$

Panjang  $l_2$  didapat dari akar penjumlahan  $b^2$  dengan  $t_3$ , dengan  $t_3$  diperoleh dari selisih  $t_1$  dan  $t_2$ . Panjang  $l_2$ ,  $t_1$ ,  $t_2$  seperti pada Gambar 7.



Gambar 7. Panjang  $l_2$ ,  $t_1$ ,  $t_2$

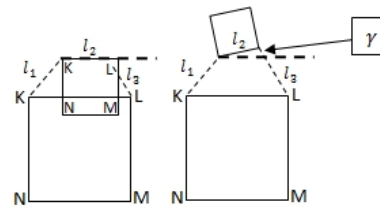
Tahap ketiga membangun bentuk pohon fraktal menggunakan transformasi affine pada IFS. Cabang pertama di sisi  $l_1$  didapat dari persegi awal yang diduplikasi. Selanjutnya, pada hasil duplikasi diterapkan beberapa transformasi affine yaitu dilatasi menggunakan Persamaan (2) yang ditentukan oleh titik pusat dilatasi dan faktor skala dilatasi. Faktor skala dilatasi ( $k_1$ ) didapat dari pembagian  $l_1$  dengan  $l$ . Titik pusat dilatasi pada titik  $K$ . Hasil dari dilatasi kemudian dilakukan operasi rotasi sebesar sudut  $\alpha$  dengan menentukan arah perputaran berlawanan jarum jam menggunakan Persamaan (3) dan titik pusat rotasi berada di titik  $K$ . Operasi dilatasi dan rotasi ini diterapkan secara berulang-ulang pada persegi hingga membentuk pohon fraktal. Gambar 8 merupakan ilustrasi percabangan di  $l_1$ .



Gambar 8. Ilustrasi percabangan pada  $l_1$

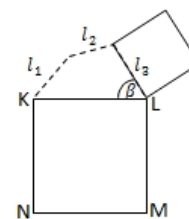
Cabang kedua di sisi  $l_2$  didapat dari persegi awal yang diduplikasi. Selanjutnya, pada hasil duplikasi diterapkan beberapa transformasi affine yaitu dilatasi menggunakan Persamaan (2) yang ditentukan oleh titik pusat dilatasi dan faktor skala dilatasi. Faktor

skala dilatasi ( $k_2$ ) didapat dari pembagian  $l_2$  dengan  $l$ . Titik pusat dilatasi pada titik  $K$ . Hasil dari dilatasi kemudian dilakukan operasi translasi menggunakan Persamaan (5). Hasil dari translasi kemudian dilakukan operasi rotasi sebesar sudut  $\gamma$  dengan menentukan arah perputaran berlawanan jarum jam menggunakan Persamaan (3) atau searah perputaran jarum jam menggunakan Persamaan (4) dan titik pusat rotasi berada di titik  $K$  persegi baru hasil translasi. Gambar 9 merupakan ilustrasi percabangan di  $l_2$ . Operasi dilatasi, translasi dan rotasi ini diterapkan secara berulang-ulang pada persegi hingga membentuk pohon fraktal.



Gambar 9. Ilustrasi percabangan pada  $l_2$

Cabang ketiga di sisi  $l_3$  didapat dari persegi awal yang diduplikasi. Selanjutnya, pada hasil duplikasi diterapkan beberapa transformasi affine yaitu dilatasi menggunakan Persamaan (2) yang ditentukan oleh titik pusat dilatasi dan faktor skala dilatasi. Faktor skala dilatasi ( $k_3$ ) didapat dari pembagian  $l_3$  dengan  $l$ . Titik pusat dilatasi pada titik  $L$ . Hasil dari dilatasi kemudian dilakukan operasi rotasi sebesar sudut  $\beta$  dengan menentukan arah perputaran searah jarum jam menggunakan Persamaan (4) dan titik pusat rotasi berada di titik  $L$ . Gambar 10 merupakan ilustrasi percabangan di  $l_3$ . Operasi dilatasi dan rotasi ini diterapkan secara berulang-ulang pada persegi hingga membentuk pohon fraktal.



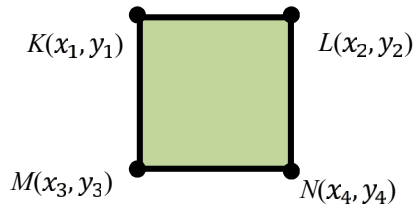
Gambar 10. Ilustrasi percabangan pada  $l_3$ .

**HASIL DAN PEMBAHASAN**

Pada tahap pertama pohon fraktal dibangkitkan dengan bentuk awal berupa persegi pada bidang kartesius. Persegi dibangkitkan pada koordinat kartesius sebanyak jumlah titik sudutnya. Penamaan variabel yang digunakan dapat dilihat pada Gambar 11. Rumus jaraknya adalah

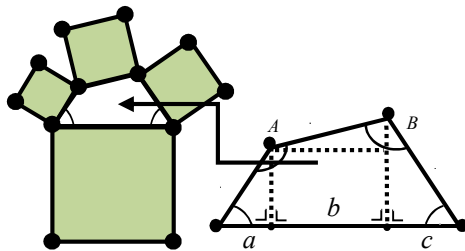
$$s = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

dengan  $s$  menyatakan jarak antara dua titik  $K(x_1, y_1)$  dan  $L(x_2, y_2)$ .



Gambar 11. Penamaan variabel persegi

Pada tahap kedua ditentukan bentuk dasar percabangan.

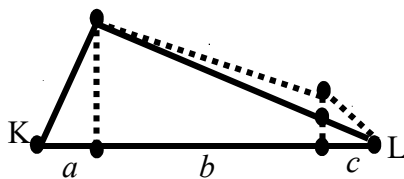


Gambar 12. Bentuk dasar percabangan

Agar kemiringan  $L_1$  dan  $L_2$  tidak melebihi batas digunakan aturan syarat  $0 < p_1 < p_2 < 1$ , dengan batas nilai  $p_1, p_2$  yang diinputkan dari 0,1 sampai 0,9. Batas nilai ini diberikan untuk menjaga kemiringan sisi untuk membangkitkan persegi pada pohon fraktal. Dengan menginputkan nilai  $p_1$  dan  $p_2$  selanjutnya dapat dicari nilai  $a, b$ , dan  $c$  seperti pada Gambar 12.

Setelah didapat nilai  $a, b, c$ , selanjutnya  $t_1, t_2$  diberikan aturan syarat batas atas  $t_1 \leq \sqrt{s - a^2}$  dan  $t_2 \leq \sqrt{s - c^2}$ . Untuk memastikan ukuran cabang pertama, kedua dan ketiga tidak lebih besar dari persegi awal maka panjang  $s$  dibatasi 0,8 kali panjang dari iterasi sebelumnya.

Gambar 13 merupakan ilustrasi syarat batas bawah  $t_2$ .

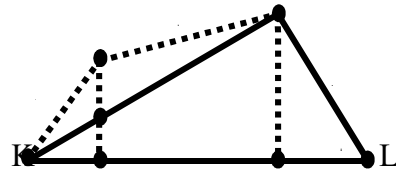


Gambar 13. Ilustrasi syarat batas bawah  $t_2$

Dari Gambar 13 terlihat bahwa  $\tan \beta = \frac{t_1}{b+c} = \frac{t_2}{c}$ . Jadi, nilai  $t_2$  pada iterasi berikutnya agar membentuk percabangan harus memiliki batas bawah

$$t_2 > \frac{t_1}{b+c} c.$$

Gambar 14 merupakan ilustrasi syarat batas bawah  $t_1$ .



Gambar 14. Ilustrasi syarat batas bawah  $t_1$

Terlihat bahwa  $\tan \alpha = \frac{t_1}{a} = \frac{t_2}{a+b}$ . Jadi, nilai  $t_1$  pada iterasi berikutnya agar membentuk percabangan harus memiliki batas bawah

$$t_1 > \frac{t_2}{a+b} a.$$

Jika  $t_1, t_2$  memenuhi syarat, maka besar sudut  $\alpha, \beta$  dapat dicari dengan rumus  $\alpha = a \tan\left(\frac{t_1}{a}\right)$  dan  $\beta = a \tan\left(\frac{t_2}{c}\right)$ .

Syarat batas tersebut untuk mengoreksi percabangan memenuhi segiempat konveks atau tidak. Syarat batas atas dan bawah berlaku untuk pemilihan secara ditentukan atau diinputkan maupun random.

Mencari nilai  $L_1$  dan  $L_3$  dengan menginputkan hasil dari  $a$  dan  $c$  dan dibagi dengan besar sudut  $\alpha$  dan  $\beta$  seperti pada Gambar 12 dengan

$$L_1 = \frac{a}{s \cdot \left(\frac{\pi}{2} - \alpha\right)}, L_3 = \frac{c}{s \cdot \left(\frac{\pi}{2} - \beta\right)}$$

Setelah didapat hasil dari  $L_1$  dan  $L_3$ , maka nilai  $t_1, t_2, t_3$  seperti pada Gambar 12 dapat diperoleh, yaitu

$$t_1 = \sqrt{L_1^2 + a^2},$$

$$t_2 = \sqrt{L_3^2 - c^2},$$

$$t_3 = |t_1 - t_2|.$$

Hasil perhitungan dari  $t_1, t_2$  digunakan dalam program IFS. Selanjutnya dengan menginputkan hasil dari  $b$  dan dikombinasikan dengan  $t_3$ , maka dapat dicari nilai dari  $L_2$ , yaitu

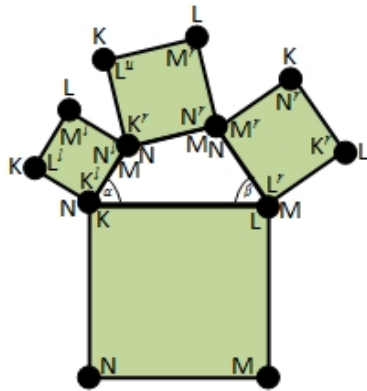
$$L_2 = \sqrt{b^2 + t_3}$$

Setelah diperoleh nilai  $L_2$ , maka selanjutnya besar sudut  $\gamma$  dapat dicari dengan perbandingan  $t_3$  dan  $L_2$  sebagai berikut

$$\gamma = a \sin\left(\frac{t_3}{L_2}\right).$$

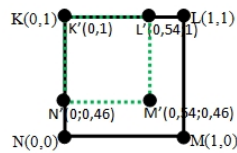
Hasil dari persegi yang dirotasi ke kiri disimbolkan dengan *superscript l*, hasil dari persegi yang ditranslasi dan dirotasi ke atas disimbolkan dengan *superscript u*, dan hasil persegi yang dirotasi ke kanan disimbolkan dengan *superscript r*. Visualisasi perubahan

nama titik hasil rotasi diilustrasikan seperti pada Gambar 15.



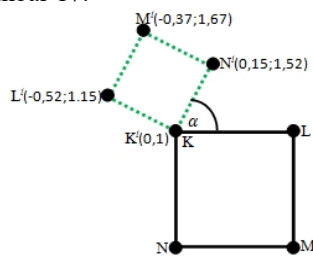
Gambar 15. Ilustrasi Perubahan Nama Titik

Pada tahap ketiga dilakukan pembentukan pohon fraktal dengan metode IFS, yaitu dalam hal ini digunakan transformasi affine. Dengan faktor dilatasi tertentu (misal  $k=0,54$ ), diperoleh hasil dilatasi di titik pusat dilatasi  $K(0,1)$  pada cabang pertama seperti pada Gambar 16.



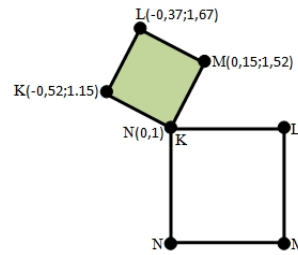
Gambar 16. Hasil dilatasi

Setelah didapat hasil dilatasi, selanjutnya di rotasi dengan arah perputaran berlawanan arah jarum jam sebesar sudut  $\alpha$  di titik pusat rotasi  $K(0,1)$ . Sehingga didapat hasil rotasi seperti pada Gambar 17.



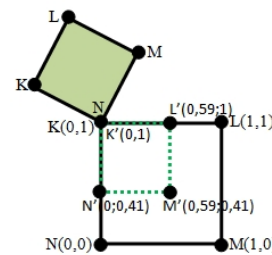
Gambar 17. Hasil Rotasi untuk cabang pertama

Setelah dihasilkan persegi baru dari hasil rotasi penamaan titik diubah seperti Gambar 17, maka didapat penamaan titik  $K^L, L^L, M^L, N^L$ . Selanjutnya penamaan titik-titik baru tersebut dikembalikan seperti penamaan titik pada persegi awal seperti Gambar 18 menjadi  $KLMN$ . Penamaan titik diubah dan menghasilkan cabang pertama seperti pada Gambar 18.



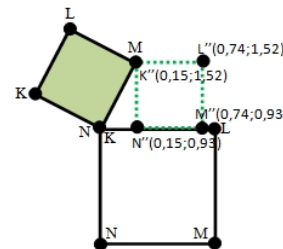
Gambar 18. Hasil dari percabangan pertama

Selanjutnya pada cabang kedua, didapat hasil dilatasi di titik pusat dilatasi  $K(0,1)$  seperti pada Gambar 19.



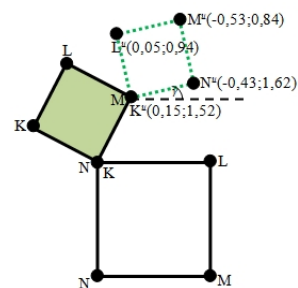
Gambar 19. Hasil dilatasi

Setelah didapat hasil dilatasi, selanjutnya ditranslasi dan didapat hasil seperti pada Gambar 20.



Gambar 20. Hasil translasi

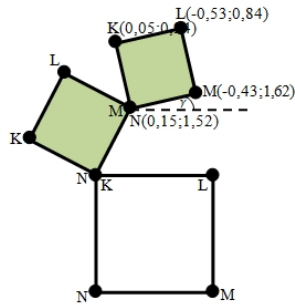
Arah rotasi kemudian diputar berlawanan arah jarum jam sebesar sudut  $\gamma$  di titik pusat rotasi  $K''(0,15;1,52)$ . Sehingga didapat hasil rotasi pada Gambar 21.



Gambar 21. Hasil rotasi

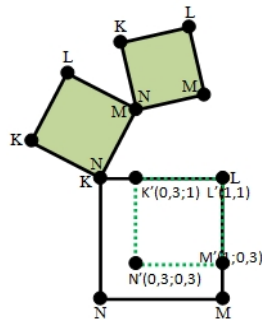
Setelah dihasilkan persegi baru dari hasil rotasi penamaan titik diubah seperti Gambar 21,

maka didapat penamaan titik  $K^u, L^u, M^u, N^u$ . Selanjutnya penamaan titik-titik baru tersebut dikembalikan seperti penamaan titik pada persegi awal seperti Gambar 22 menjadi  $KLMN$ . Penamaan titik diubah dan menghasilkan cabang pertama seperti pada Gambar 22.



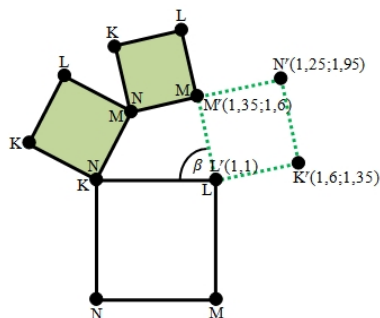
Gambar 22. Hasil dari percabangan kedua

Selanjutnya pada cabang ketiga, didapat hasil dilatasi di titik pusat dilatasi  $L(1,1)$  seperti pada Gambar 23.



Gambar 23. Hasil dilatasi percabangan ketiga

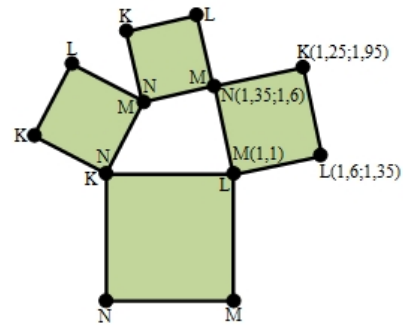
Setelah didapat hasil dilatasi, selanjutnya dirotasi dengan arah perputaran searah jarum jam sebesar sudut  $\beta$  di titik pusat rotasi  $L(0,1)$  dan didapat hasil rotasi seperti pada Gambar 24.



Gambar 24. Hasil rotasi percabangan ketiga

Setelah dihasilkan persegi baru dari hasil rotasi penamaan titik diubah seperti Gambar 19, maka didapat penamaan titik  $K^r, L^r, M^r, N^r$ . Selanjutnya penamaan titik-titik baru tersebut

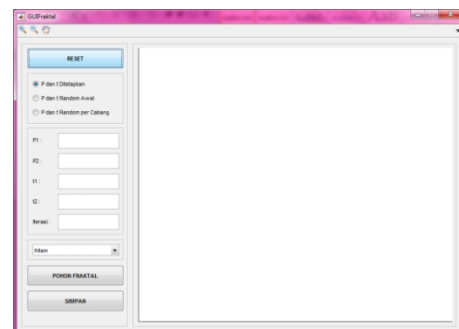
dikembalikan seperti penamaan titik pada persegi awal seperti Gambar 25 menjadi  $KLMN$ . Penamaan titik diubah dan menghasilkan cabang pertama seperti pada Gambar 25.



Gambar 25. Hasil dari percabangan ketiga

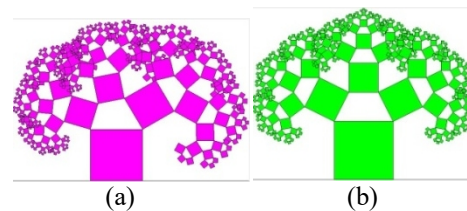
Langkah yang sama dilakukan terus menerus sampai iterasi ke- $n$ .

Simulasi pembuatan pohon fraktal dilakukan dengan membuat Guide Users Interface (GUI) pada MATLAB2015B (32 BIT). Tampilan GUI dapat dilihat di Gambar 26.



Gambar 26. Tampilan GUI pohon fraktal

Dalam hal ini ditampilkan dua hasil pembangkitan pohon fraktal berdasarkan pemilihan tinggi  $t$  dan panjang  $P$  yang ditetapkan secara manual, seperti pada Gambar 27.



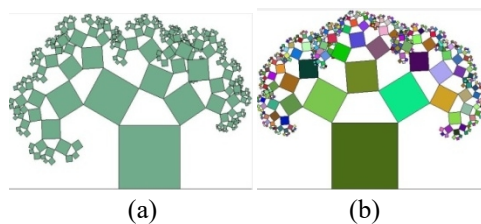
Gambar 27. Hasil visualisasi pemilihan secara ditetapkan

Gambar 27 bagian (a) penulis menginputkan nilai  $P_1 = 0,15, P_2 = 0,65, t_1 = 0,5, t_2 = 0,6$ , diperoleh besar sudut  $\alpha = 73,61^\circ$  dan  $\beta =$



59,53°. Nilai  $t_2$  yang diinputkan lebih panjang dari  $t_1$  dan besar sudut  $\alpha$  lebih besar dari sudut  $\beta$ , sehingga didapat kemiringan  $l_1$  yang lebih miring dari  $l_3$  dan panjang  $l_3 = 0,7$  lebih panjang dari  $l_1 = 0,54$  sehingga diperoleh cabang  $l_2$  lebih condong ke kiri. Gambar 27 bagian (b) peneliti menginputkan nilai  $P_1 = 0,25, P_2 = 0,75$  dan  $t_1 = 0,5, t_2 = 0,5$ , diperoleh besar sudut  $\alpha = \beta$ ,  $l_1$  dan  $l_3$  sama panjang sehingga terlihat bahwa pohon fraktal memiliki bentuk yang simetris.

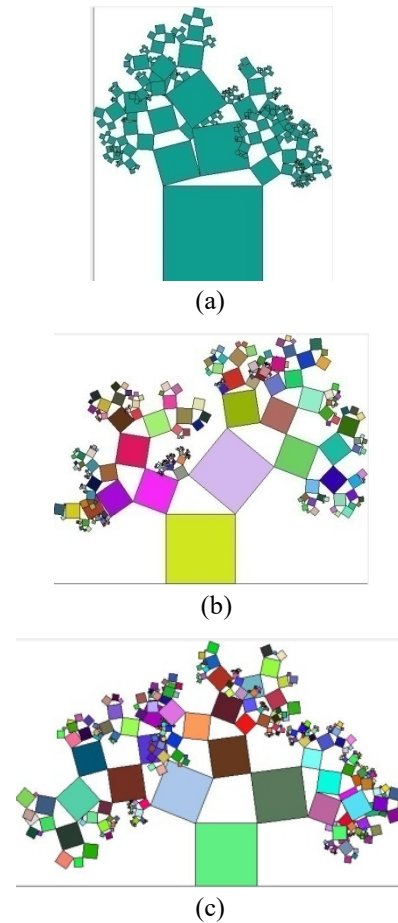
Pemilihan tinggi dan P secara random, terdiri dari dua yaitu random awal dimana bentuk dasar percabangan hanya random diawal percabangan saja, percabangan selanjutnya sampai iterasi ke- $n$  bentuk dasar percabangannya tetap dan sama seperti iterasi ke-1, dan untuk random per cabang P dan t random per cabang, dimana bentuk dasar tiap percabangannya selalu random atau bentuk dasar percabangan pertama sampai percabangan pada iterasi ke- $n$  berbeda-beda. Hasil visualisasi pohon fraktal untuk pemilihan tinggi dan P secara random awal peneliti menampilkan dua hasil seperti pada Gambar 28.



Gambar 28. Hasil visualisasi random awal

Gambar 28 bagian (a) terlihat bahwa panjang  $l_1$  lebih panjang dari  $l_3$ . Hal ini dikarenakan panjang  $t_1$  lebih panjang dari  $t_2$ . Besar sudut  $\alpha$  dan  $\beta$  sama besar, sehingga kemiringan dari  $l_1$  dan  $l_3$  sama. Cabang  $l_2$  diubah condong ke arah kanan. Gambar 28 bagian (b) bentuk dasar percabangan pada iterasi pertama terlihat bahwa letak titik  $P_1$  dan  $P_2$  seimbang, dimana tidak mendekati batas minimum maupun maksimum. Panjang  $t_1$  dan  $t_2$  hampir sama panjang, dan besar sudut  $\alpha, \beta$  sama besar.

Hasil visualisasi pohon fraktal untuk pemilihan tinggi dan P secara random per cabang peneliti menampilkan tiga hasil seperti pada Gambar 29.



Gambar 29. Hasil visualisasi random per cabang

Gambar 29 bagian (a) bentuk dasar percabangan pada iterasi pertama, terlihat bahwa sudut  $\delta$  hampir sebesar  $180^\circ$  (bentuknya hampir tidak konveks). Hal ini dikarenakan panjang  $t_2$  lebih panjang dari  $t_1$  dan sudut  $\beta$  lebih besar dari sudut  $\alpha$ , sehingga mengakibatkan cabang  $l_1$  dan  $l_2$  hampir menempati garis lurus. Gambar 29 bagian (b) bentuk dasar percabangan pada iterasi pertama, terlihat bahwa titik  $P_2$  melewati titik tengah  $l$  dan mendekati titik  $P_1$ . Panjang  $t_2$  lebih panjang dari  $t_1$ , dan besar sudut  $\alpha$  lebih besar dari sudut  $\beta$ , sehingga dihasilkan ukuran cabang  $l_2$  seperti pada Gambar 29 bagian (b). Gambar 29 bagian (c) bentuk dasar percabangan pada iterasi pertama, panjang  $t_2$  lebih panjang dari  $t_1$ , sehingga diperoleh cabang pada  $l_2$  sedikit

condong ke kiri. Dikarenakan titik  $P_2$  mendekati batas maksimum dan panjang  $t_2$  lebih panjang dari  $t_1$ , sehingga didapat besar sudut  $\beta$  hampir tegak lurus (mendekati  $90^\circ$ ).

### KESIMPULAN

Di dalam artikel ini didapatkan sejumlah variasi pohon fraktal, yaitu didapatkan dengan memberikan nilai parameter tinggi dan panjang tertentu yang digunakan untuk menentukan ukuran dan sudut percabangan. Dalam kasus khusus, diperoleh bentuk pohon fraktal yang random (dalam arti dapat bergerak condong ke kanan, ke kiri, atau simetris pada setiap percabangan) pada saat nilai-nilai parameter tersebut diberi nilai random pada interval tertentu.

### DAFTAR PUSTAKA

- Ambarsari, E.W., 2019. Basic Concept Pythagoras Tree for Construct Data Visualization on Decision Tree Learning. *Journal of Applied Engineering Science*. **17**(4): 468-472.
- Beck, F., 2015. Generalized Pythagoras Tree: A Fractal Approach to Hierarchy Visualization. *International Joint Conference on Computer Vision, Imaging and Computer Graphics*. 115-135.
- Browne, C., 2007. Efficient Pythagorean trees: Greed is good. *Computers & Graphics*. **9**(1): 111-124.
- Hasang, S., Suparjo, S. 2012. Geometri Fraktal dalam Rancangan Arsitektur. *Media Matrasain*. **9**(1): 111-124.
- Kusno. 2003. *Geometri Rancang Bangun Studi Surfasi Putar Transformasi Titik dan Proyeksi*. Jember: Fakultas MIPA Universitas Jember.
- Matam, S. 2012. *Pythagoras Trees*. <https://www.codewalk.com/2012/01/pythagoras-trees.html>. [29 Mei 2019]
- 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.
- Prasasti, I. 2018. *Pemanfaatan Metode Iterated Function System Dalam Pengembangan Motif Anyaman*. Skripsi. Jember: Jurusan Matematika FMIPA Universitas Jember.
- Purnomo, K.D. 2014. Pmbangkitan Segitiga Sierpinski dengan Transformasi Affine. *Prosiding Seminar Nasional Matematika Jurusan Matematika FMIPA Universitas Jember*. 365-375.
- Sharma, R., Kokane, D.N.. 2015. Pythagorean Tree Multiband Fractal Antenna. *International Journal of Engineering Research and General Science*. **3**(5): 1031-1035.
- Teia, L., 2015. Pythagoras' triples explained by central squares. *Australian Senior Mathematics Journal*. **29**(1): 7-15.
- Teia, L., 2016. Anatomy of the Pythagoras' tree. *Australian Senior Mathematics Journal*. **30**(2): 38-47.
- Ubaidillah, F., Purnomo, K.D., Sundusia, J.K. 2019. *Pembangkitan Fraktal Pohon Pythagoras Menggunakan Iterated Function System*. Prosiding Seminar Nasional Integrasi Matematika dan Nilai Islam. **3**(1): 55-60.
- Van der Laan, K. 2013. *Pythagoras Trees in PostScript*. <https://www.ntg.nl/maps/44/06.pdf> [28 Mei 2021]