PEMBANGKITAN FRAKTAL *KOCH* ANTI-*SNOWFLAKE* (m, n, c) MENGGUNAKAN METODE TRANSFORMASI AFFINE

SKRIPSI

Oleh

Ellenda Alkhori
NIM 151810101007

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2019

PEMBANGKITAN FRAKTAL *KOCH* ANTI-*SNOWFLAKE* (m, n, c) MENGGUNAKAN METODE TRANSFORMASI AFFINE

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh

Ellenda Alkhori
NIM 151810101007

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2019

PERSEMBAHAN

Dengan menyebut nama Allah yang maha pengasih lagi maha penyayang dan dengan segala kerendahan hati serta puji syukur yang tak terhingga pada Allah SWT, skripsi ini saya persembahkan untuk:

1. Ibuku tercinta Laswati, bapakku tercinta Buayat yang telah membesarkan, mendidik, mendoakan, memotivasi dengan penuh kasih sayang dan pengorbanan selama ini;
2. Kakak Egi Primanta tersayang yang telah memberikan semangat, doa dan dukungan kepada penulis;
3. Laki-laki terkasihku Idharis Surur yang selalu menemani, memberikan semangat, doa dan dukungan kepada penulis;
4. Keluarga besar dari ibuku Laswati dan bapakku Buayat yang selalu memberikan doa dan dukungan kepada penulis;
5. Guru-guru TK Tunas Bangsa, SDN 3 Lemahbang Kulon, SMPN 3 Rogojampi, SMAN 1 Rogojampi, dan dosen-dosen Matematika FMIPA Universitas Jember yang telah memberikan ilmu dan membimbing dengan tulus dan penuh dengan kesabaran;
6. Almamater tercinta Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTTO

"Bertaqwalah kepada Allah, maka Dia akan membimbingmu. Sesungguhnya Allah mengetahui segala sesuatu"

(Q.S Al-Baqarah: 282)

"Persembahkanlah apa yang kamu cita-citakan untuk kedua orang tua dan orang-orang terkasihmu, percayalah pasti Allah akan selalu memberikan jalan yang mudah untukmu meraihnya"

(Ellenda Alkhori)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ellenda Alkhori

NIM : 151810101007

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul: ”Pembangkitan Fraktal *Koch* Anti-*Snowflake* (m, n, c) menggunakan Metode Transformasi Affine” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Juli 2019

Yang menyatakan,

Ellenda Alkhori

NIM 151810101007

SKRIPSI

PEMBANGKITAN FRAKTAL *KOCH* ANTI-*SNOWFLAKE* $(\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{c})$ MENGGUNAKAN METODE TRANSFORMASI AFFINE

Oleh

Ellenda Alkhori
NIM 151810101007

Pembimbing;

Dosen Pembimbing Utama : Kosala Dwidja Purnomo, S.Si., M.Si.

Dosen Pembimbing Anggota : Bagus Juliyanto, S.Si., M.Si.

PENGESAHAN

Skripsi berjudul "Pembangkitan Fraktal *Koch* Anti-*Snowflake* (m, n, c) menggunakan Metode Transformasi Affine" telah diuji dan disahkan pada:

Hari :

Tanggal :

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji:

Ketua,	Anggota I,
Kosala Dwidja Purnomo, S.Si., M.Si.	Bagus Juliyanto, S.Si., M.Si.
NIP. 196908281998021001	NIP. 198007022003121001
Anggota II,	Anggota III,
Dr. Firdaus Ubaidillah, S.Si., M.Si.	Abduh Riski, S.Si., M.Si.
NIP. 197006061998031003	NIP. 199004062015041001

Mengesahkan
Dekan,

Drs. Sujito, Ph.D.
NIP. 196102041987111001

RINGKASAN

Pembangkitan Fraktal *Koch* Anti-*Snowflake* (m, n, c) menggunakan Metode Transformasi Affine; Ellenda Alkhori; 151810101007; 2019; 58 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Fraktal merupakan sebuah bentuk geometri kompleks dan atau sebuah fenomena alam yang dibangun melalui instruksi sederhana yang berulang dengan sifat *self-similarity*. Salah satu bentuk fraktal dalam geometri kompleks adalah *Koch Snowflake* yang merupakan bentuk perkembangan dari fraktal kurva *Koch*. Fraktal *Koch Snowflake* dapat divariasikan dalam bentuk luas berkurang atau yang biasa dikenal dengan sebutan *Koch* Anti-*Snowflake*. Fraktal *Koch* Anti-*Snowflake* dibentuk dengan cara berlawanan dari *Koch Snowflake* luas bertambah yaitu dengan cara membangkitkan generatornya pada setiap sisi inisiator ke arah dalam atau menuju pusat.

Dalam penelitian ini fraktal *Koch* Anti-*Snowflake* akan divariasikan menggunakan (m, n, c), dengan m merupakan inisiator (bentuk dasar) yaitu menggunakan bentuk poligon segi-m $(3 \leq m \leq 6)$ dan n merupakan bentuk generator (perulangan) yang akan dibangkitkan pada segmen tengah di setiap sisi inisiatornya yaitu menggunakan poligon segi-n $(3 \leq n \leq 6)$ dengan panjang dari setiap sisi generatornya tersebut adalah sebesar c. Sedangkan c merupakan ukuran segmen tengah yang akan dihilangkan pada setiap sisi inisiatornya. Setiap iterasi dalam prosedur pembangkitan fraktal *Koch* Anti-*Snowflake* (m, n, c) menggunakan metode transformasi affine dengan beberapa aturan konstruksi yaitu dilatasi, translasi, dan rotasi. Pembangkitan fraktal *Koch* Anti-*Snowflake* (m, n, c) terdiri dari beberapa prosedur yaitu pertama menentukan inisiator (m) dengan bentuk poligon segi-m $(3 \leq m \leq 6)$, kedua menentukan nilai c pada sisi alas inisiator (m) dengan membagi segmen sisi alas inisiator (m) tersebut menjadi tiga bagian, ketiga membangkitkan generator (n) pada sisi alas inisiator (m) atau

pada $\overline{AB}$ yang merupakan panjang segmen dari titik A ke titik B dengan bentuk generatornya yaitu poligon segi-n $(3 \leq n \leq 6)$ pada segmen tengah yang telah dihilangkan dengan panjang dari setiap sisi generatornya tersebut adalah sebesar c, keempat membangkitkan generator (n) *Koch* Anti-*Snowflake* hingga iterasi-i yang akan menghasilkan kurva *Koch* dengan pembangkitan generator (n) tersebut dilakukan pada setiap sisi segmen yang dihasilkan dari iterasi sebelumnya, dan kelima membangkitkan generator (n) atau kurva *Koch* pada sisi inisiator (m) lain sehingga akan menghasilkan bentuk fraktal *Koch* Anti-Snowflake (m, n, c) dengan inisiator (bentuk dasar) poligon segi-m $(3 \leq m \leq 6)$. Sedangkan variasi nilai c dipilih sebanyak tiga variasi yaitu setengah dari batas atas nilai c yang diperbolehkan, ada variasi nilai c yang dipilih yaitu kurang dari setengah batas atas nilai c yang diperbolehkan, dan variasi nilai c yang cukup mendekati batas atas nilai c yang diperbolehkan yaitu sebesar $c - 0{,}00001$.

Berdasarkan prosedur yang telah dilakukan penelitian ini dapat menghasilkan beberapa bentuk, pertama bentuk fraktal *Koch* Anti-*Snowflake* (m, n, c) tidak berpotongan yaitu dengan variasi inisiator menggunakan bentuk poligon segi-m $(3 \leq m \leq 6)$, variasi generator yang dibangkitkan menggunakan bentuk poligon segi-n $(3 \leq n \leq 6)$ dan dengan nilai c yang dipilih yaitu setengah dari batas atas nilai c yang diperbolehkan. Bentuk kedua yaitu fraktal *Koch* Anti-*Snowflake* (m, n, c) berhimpit yaitu dengan variasi inisiator menggunakan bentuk poligon segi-m $(3 \leq m \leq 6)$, variasi generator yang dibangkitkan menggunakan bentuk poligon segi-n $(3 \leq n \leq 6)$ dan ada variasi nilai c yang dipilih yaitu kurang dari setengah batas atas nilai c yang diperbolehkan. Bentuk ketiga yaitu fraktal *Koch* Anti-*Snowflake* (m, n, c) berpotongan terdapat dua variasi, pertama variasi yang menggunakan inisiator poligon segi-m $(m = 3)$ dengan generator yang dibangkitkan menggunakan bentuk poligon segi-n $(3 \leq n \leq 5)$, variasi kedua yaitu menggunakan inisiator poligon segi-m $(3 \leq m \leq 5)$ dengan generator yang dibangkitkan menggunakan bentuk poligon segi-n $(n = 3)$, masing-masing nilai c yang dipilih yaitu cukup mendekati batas atas nilai c yang diperbolehkan sebesar $c - 00{,}00001$.

PRAKATA

Puji syukur penulis kepada Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Pembangkitan Fraktal *Koch* Anti-*Snowflake* (m, n, c) menggunakan Metode Transformasi Affine". Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember. Penyusunan skripsi ini tidak terlepas dari perhatian, bimbingan, motivasi dan bantuan dari beberapa pihak, baik secara langsung maupun tidak langsung. Pada kesempatan ini penulis menyampaikan terimakasih kepada:

1. Kosala Dwidja Purnomo, S.Si., M.Si., selaku Dosen Pembimbing Utama serta Bagus Juliyanto, S.Si., M.Si., selaku Dosen Pembimbing Anggota yang telah ikhlas memberikan ilmu yang bermanfaat dan bersedia meluangkan waktu dan tenaga untuk membimbing penulis dengan penuh kesabaran dalam menyelesaikan skripsi;
2. Dr. Firdaus Ubaidillah, S.Si., M.Si., dan Abduh Riski, S.Si., M.Si selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun dalam menyelesaikan skripsi;
3. Kosala Dwidja Purnomo, S.Si., M.Si., selaku Dosen Wali yang memberikan bimbingan dan dukungan selama menjalani perkuliahan;
4. Drs. Sujito, PhD selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
5. Seluruh dosen dan staf karyawan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
6. Ibu Laswati, Bapak Buayat, Kakak Egi Primanta, dan Idharis Surur yang telah memberikan semangat, doa, dan dukungan kepada penulis;
7. Teman-teman seperjuangan SIGMA'15 yang telah berjuang bersama dan banyak membantu penulis selama studi;

8. Teman-teman satu bidang skripsi Fraktal dan Kalman Indy, Nanda, Mitha, Dyakza, Dwi Alfi, Ayu, dan Nella yang selalu memberi dukungan dan semangat tanpa henti;
9. Teman-teman TETEW (Novita, Retno, dan Rivi) dan teman-teman seperjuangan Banyuwangi (Monika, Citra, Uum, Puput, Nensi, dan Ria) yang selalu memberi keceriaan, dukungan dan semangat tanpa henti;
10. Teman-teman tersayang Eka, Tita, Tika, Yuli, Diah, Amalia, Mahrita, Sheila, Giki, Nuryu, Lia, Vita, dan teman-teman KKN 73 yang menemani selama di Jember, memberi dukungan dan semangat tanpa henti;
11. Kakak-kakak tingkat dan adik-adik tingkat yang selalu memotivasi dan memberi semangat penulis agar terselesaikannya skripsi ini;
12. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang memberikan dorongan bagi penulis selama studi sampai penulisan skripsi selesai.

Penulisan skripsi ini masih jauh dari kesempurnaan sehingga diharapkan adanya saran dan kritik untuk perbaikan selanjutnya. Semoga skripsi ini bermanfaat bagi semua pihak.

Jember, Juli 2019

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR LAMPIRAN

Halaman

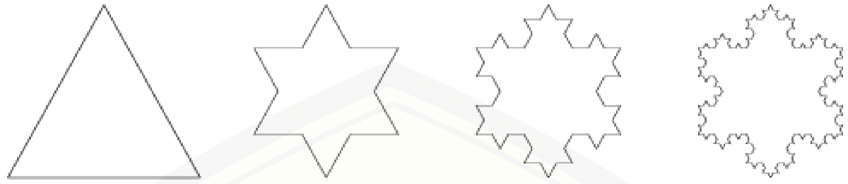
BAB 1. PENDAHULUAN

1.1 Latar Belakang

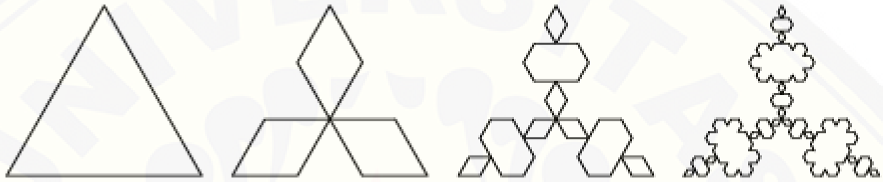
Ilmu matematika di era modern sekarang ini semakin luas berkembang bersamaan dengan teknologi komputasi. Melalui komputasi dapat mempermudah kegiatan manusia dalam melakukan perhitungan matematika atau perhitungan lainnya. Cabang ilmu matematika yang berkembang pesat dengan adanya komputasi salah satunya adalah fraktal.

Fraktal merupakan sebuah bentuk geometri kompleks dan atau sebuah fenomena alam yang dibangun melalui instruksi sederhana yang berulang dengan sifat *self-similarity*. Ciri khas dari sebuah fraktal adalah mempunyai sifat *self-similarity* dan *infinite detail*. *Self-similarity* mempunyai makna bahwa setiap bagian kecil dalam suatu bentuk fraktal dapat dilihat sebagai bagian replikasi dari bentuk keseluruhan dengan skala yang lebih kecil (Purnomo, 2016). Sedangkan *infinite detail* merupakan objek fraktal yang memiliki bentuk dasar yang seakan-akan tidak habis-habis bila diperhatikan. Beberapa bentuk matematis yang termasuk ke dalam fraktal antara lain: segitiga Sierpinski, *Koch Snowflake*, kurva Hilbert dan himpunan Mendelbrot (Romadiastri, 2013).

Salah satu bentuk fraktal dalam geometri kompleks adalah *Koch Snowflake*, pertama kali diperkenalkan oleh Helge von Koch seorang ilmuwan matematika yang berasal dari Swedia pada tahun 1904. Fraktal *Koch Snowflake* merupakan bentuk perkembangan dari fraktal kurva *Koch* yang dibangun dari sebuah inisiator (bentuk dasar) segitiga sama sisi kemudian dibangkitkan dengan generator (perulangan) yaitu segitiga sama sisi dengan ukuran yang lebih kecil dari inisiatornya. Pembangkitan dilakukan dengan melakukan penambahan secara terus menerus bentuk yang sama pada setiap sisi inisiator menggunakan generatornya ke arah luar, sehingga disebut juga bentuk fraktal *Koch Snowflake* dengan luas bertambah. Fraktal *Koch Snowflake* dapat divariasikan dalam bentuk luas berkurang atau yang biasa dikenal dengan sebutan *Koch* Anti-*Snowflake*. Pembangkitan fraktal *Koch* Anti-*Snowflake* dibentuk dengan cara berlawanan dari *Koch Snowflake* luas bertambah yaitu dengan cara membangkitkan generatornya pada setiap sisi inisiator ke arah dalam atau menuju pusat. Bentuk fraktal *Koch Snowflake* dapat dilihat pada Gambar 1.1 dan bentuk fraktal *Koch* Anti-*Snowflake* dapat dilihat pada Gambar 1.2.



Gambar 1.1 *Koch Snowflake*
(Sumber: Weisstein, 2008 http://mathworld.wolfram.com/KochSnowflake.html)



Gambar 1.2 *Koch* Anti-*Snowflake*
(Sumber: Weisstein, 2019 http://mathworld.wolfram.com/KochAntisnowflake.html)

Beberapa objek fraktal dapat dikonstruksikan secara matematis dengan berbagai cara salah satunya menggunakan transformasi affine. Transformasi affine adalah suatu pemetaan dari satu titik ke titik yang lain dalam suatu bentuk geometri menggunakan aturan konstruksi tanpa mengubah bentuk dasarnya. Konstruksi transfromasi affine dapat berupa rotasi, translasi, dilatasi, refleksi dan *shearing* atau pembebanan. Kurva *Koch* Anti-*Snowflake* dapat dibangkitkan menggunakan salah satu aturan konstruksi pada transformasi affine, sehingga akan menghasilkan suatu variasi bentuk pada fraktal *Koch* Anti-*Snowflake*.

Kaleti, T dan Paquatte, E (2010) dalam jurnal penelitiannya dengan judul *The Trouble With Von Koch Curves Built From n Gons*, yaitu dengan variasi (n, c) dalam pembangkitkan fraktal kurva *Koch* yang dibangun dari inisiator berupa segmen garis lurus dengan n sebagai bentuk generatornya yaitu menggunakan poligon segi-n. Sedangkan c merupakan ukuran segmen tengah yang akan dihilangkan sepanjang sisi inisiatornya dan dengan generator yang akan dibangkitkan sebesar c. Penentuan nilai c merupakan hal penting yang diperhatikan agar tidak terjadi tumpang-tindih atau berpotongan dalam setiap pembangkitan generatornya. Serta Purnomo (2016) dalam jurnal penelitiannya mengenai pembangkitan segitiga Sierpinski menggunakan metode transformasi affine berbasis pada beberapa benda geometris yaitu segitiga dan segiempat. Terdapat dua algoritma yang digunakan, yaitu dengan membangkitkan segitiga berwarna yang ditempatkan pada segitiga kosong dan algoritma yang kedua menggunakan segitiga kosong dalam pembangkitannya dan akan ditempatkan pada segitiga berwarna. Pada iterasi yang cukup besar ke semua bentuk dasar dan benda geometris tersebut memberikan bentuk geometris yang mirip dengan segitiga Sierpinski.

Berdasarkan uraian di atas penulis tertarik untuk melakukan penelitian variasi *Koch* Anti-*Snowflake*. *Koch* Anti-*Snowflake* akan divariasikan menggunakan (m, n, c), dengan m merupakan inisiator (bentuk dasar) yaitu menggunakan bentuk poligon segi-m $(m \geq 3)$ dan n merupakan bentuk generator (perulangan) yang akan dibangkitkan pada segmen tengah di setiap sisi inisiatornya yaitu menggunakan poligon segi-n $(n \geq 3)$ dengan panjang dari setiap sisi generatornya tersebut adalah sebesar c. Sedangkan c merupakan ukuran segmen tengah yang akan dihilangkan pada setiap sisi inisiatornya. Setiap iterasi dalam prosedur pembangkitan fraktal *Koch* Anti-*Snowflake* (m, n, c) menggunakan metode transformasi affine dengan beberapa aturan konstruksi yaitu dilatasi, translasi, dan rotasi yang nantinya juga akan dapat dibentuk menggunakan aplikasi komputasi GUI pada matlab.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah yang akan dibahas adalah bagaimana bentuk fraktal *Koch* Anti-*Snowflake* (m, n, c) yang bervariasi menggunakan metode transformasi affine.

1.3 Batasan Masalah

Penelitian ini menggunakan bentuk fraktal *Koch* Anti-*Snowflake* (m, n, c) dengan m merupakan inisiator (bentuk dasar) menggunakan poligon segi-m $(3 \leq m \leq 6)$ dan n merupakan bentuk generator (perulangan) menggunakan poligon segi-n $(3 \leq n \leq 6)$.

1.4 Tujuan Penelitian

Sesuai dengan rumusan masalah dan latar belakang di atas, tujuan yang diharapkan pada penelitian ini adalah mengetahui bentuk fraktal *Koch* Anti-*Snowflake* (m, n, c) menggunakan metode transformasi affine dan akan dapat dibentuk menggunakan aplikasi komputasi yaitu GUI pada matlab.

1.5 Manfaat

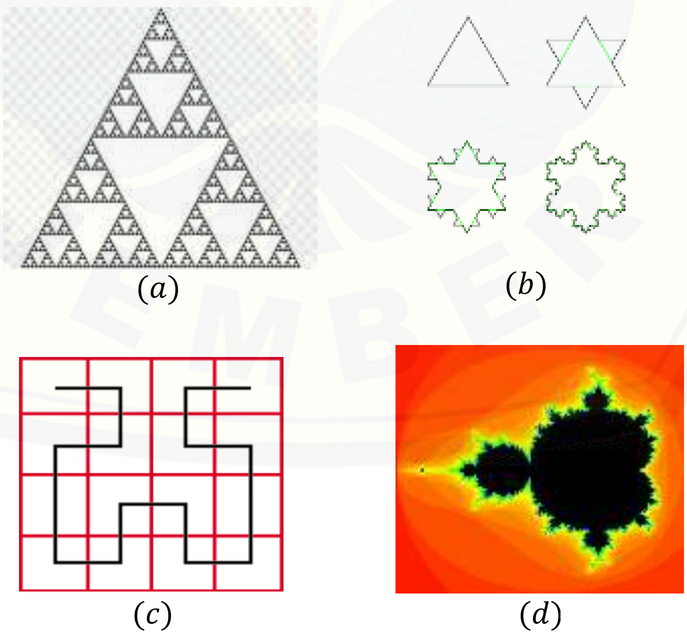
Manfaat yang diperoleh dari penelitian ini adalah memberikan variasi bentuk fraktal *Koch* Anti-*Snowflake* (m, n, c) menggunakan metode transformasi affine yang juga akan dapat ditampilkan dalam aplikasi GUI pada matlab.

BAB 2. TINJAUAN PUSTAKA

2.1 Fraktal

Kata fraktal pertama kali diperkenalkan oleh Mendelbrot pada tahun 1982 dalam bukunya yang berjudul “*The Fractal Geometry of Nature*”. Istilah fraktal tersebut digunakan untuk menunjukkan bentuk-bentuk yang mempunyai sifat *self-similarity* (Santosa, 1994). Falconer (1990) menyebutkan bahwa kata *fractal* diperkenalkan oleh Mendelbrot dari bahasa latin *fractus* yang artinya pecah atau tidak teratur.

Ciri khas dari sebuah fraktal adalah mempunyai sifat *self-similarity* dan *infinite detail*. *Self-similarity* mempunyai makna bahwa setiap bagian kecil dalam suatu bentuk fraktal dapat dilihat sebagai bagian replikasi dari bentuk keseluruhan dengan skala yang lebih kecil (Purnomo, 2016). Sedangkan *infinite detail* merupakan objek fraktal yang memiliki bentuk dasar yang seakan-akan tidak habis-habis bila diperhatikan. Romadiastri (2013) menyebutkan bahwa beberapa bentuk matematis yang termasuk ke dalam bentuk fraktal antara lain: segitiga Sierpinski, *Koch Snowflake*, kurva Hilbert dan himpunan Mendelbrot.



Gambar 2.1 Bentuk-bentuk fraktal (a) Segitiga Sierpinski (b) *Koch Snowflake* (c) Kurva Hilbert (d) Himpunan Mendelbrot (Sumber: Ulfa, 2014)

2.2 Variasi Kurva *Koch*

Kurva *Koch* merupakan salah satu bentuk fraktal yang dibangun dari inisiator (bentuk dasar) berupa segmen garis lurus dan dengan generator (perulangan) yang dibangkitkan yaitu segitiga sama sisi. Fraktal kurva *Koch* dapat divariasikan menggunakan (n, c) dengan n merupakan bentuk generator (perulangan) yang akan dibangkitkan pada segmen tengah di setiap sisi inisiatornya yaitu menggunakan poligon segi-$n \geq 3$ dengan panjang dari setiap sisi generatornya tersebut sebesar c. Sedangkan c merupakan ukuran segmen tengah yang akan dihilangkan sepanjang sisi inisiatornya dan dengan generator (n) yang akan dibangkitkan sebesar c. Penentuan batas nilai c penting diperhatikan agar tidak terjadi saling tumpang-tindih antara pembangkitan generatornya dengan aturan c seperti pada persamaan di bawah ini:

$$c < \frac{\sin^2\left(\frac{\pi}{n}\right)}{\cos^2\left(\frac{\pi}{n}\right)+1} \tag{2.1}$$

untuk generator poligon segi-n genap dengan batas $0 < c < 1$

dan
$$c < 1 - \cos\left(\frac{\pi}{n}\right) \tag{2.2}$$

untuk generator poligon segi-n ganjil dengan batas $0 < c < 1$.

Setiap pembangkitan generator kurva *Koch* (n, c) menggunakan poligon segi-$n \geq 3$ dengan sudut α dan θ sebesar:

$$\alpha = 180° - \theta \tag{2.3}$$

$$\theta = \frac{(n-2)}{n} \times 180° \tag{2.4}$$

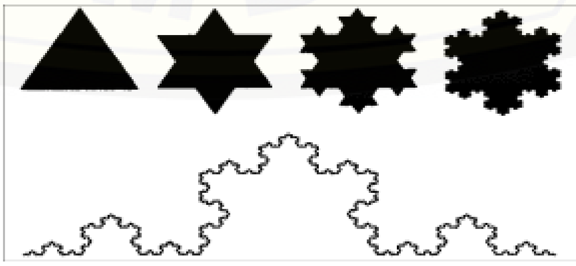
(Kaleti dan Paquatte, 2010).

2.3 Fraktal *Koch Snowflake*

Fraktal *Koch Snowflake* pertama kali diperkenalkan oleh Helge von Koch seorang ilmuwan matematika yang berasal dari negara Swedia pada tahun 1904. *Koch Snowflake* atau yang biasa dikenal dengan sebutan serpihan salju merupakan sebuah fraktal alamiah yang terjadi dari proses kristalisasi air yang kemudian membeku dan jatuh melalui atmosfer bumi. Kepingan salju terjadi dalam berbagai ukuran dan bentuk seperti pada Gambar 2.2 di bawah ini:

Gambar 2.2 Dokumentasi serpihan salju *Koch*
(Sumber: Bentley, 2005-2006 https://id.wikipedia.org/wiki/Serpihan_salju)

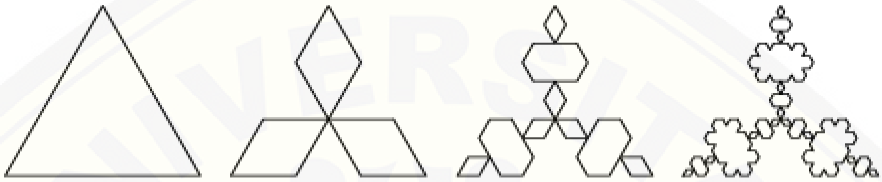
Fraktal *Koch Snowflake* merupakan bentuk perkembangan dari fraktal kurva *Koch* seperti pada Gambar 2.3 yang dibangun dari sebuah inisiator segitiga sama sisi kemudian dibangkitkan dengan membuat penambahan secara terus menerus bentuk yang sama pada setiap sisi inisiatornya ke arah luar. Penambahan dilakukan dengan cara membagi pada masing-masing sisi inisiator menjadi tiga bagian sama panjang terelebih dahulu, selanjutnya menghilangkan bagian segmen tengahnya sebesar $\frac{1}{3}$ dan kemudian membangkitkan generatornya yaitu menggunakan bentuk segitiga sama sisi baru dengan panjang sisi-sisinya sebesar $\frac{1}{3}$ pada segmen tengah yang telah dihilangkan. Sehingga akan menghasilkan segitiga sama sisi baru tanpa alas di setiap sisi inisiatornya. Penambahan akan dilakukan secara terus menerus pada setiap segmen yang dihasilkan hingga iterasi ke-i dengan ukuran segmen tengah yang dihilangkan dan generator yang dibangkitkan masing-masing sebesar $\left(\frac{1}{3}\right)^i$ (Riddle, 2017).



Gambar 2.3 Perkembangan fraktal kurva *Koch*
(Sumber: Walling, 2003 https://www.researchgate.net/figure/The-Koch-snowflake To-the-sides-of-an-equilateral-triangle-add-replicas-with-sides-one_fig1_7488992)

2.4 Fraktal *Koch* Anti-*Snowflake*

Fraktal *Koch* Anti-*Snowflake* merupakan variasi dari fraktal *Koch Snowflake* dengan luas berkurang atau dikenal juga dengan sebutan *Koch* Anti-*Snowflake* yang dibangkitkan dengan cara membangkitkan generator pada setiap sisi inisiator secara berlawanan yaitu dengan pembangkitan generatornya ke arah dalam seperti pada Gambar 2.4:

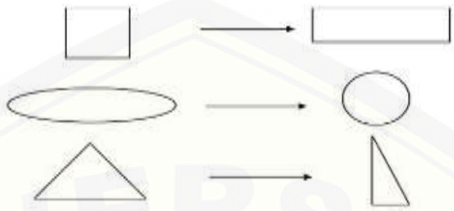


Gambar 2.4 *Koch* Anti-*Snowflake*
(Sumber: Weisstein, 2019 http://mathworld.wolfram.com/KochAntisnowflake.html)

Fraktal *Koch* Anti-*Snowflake* di bangun dari sebuah inisiator segitiga sama sisi kemudian dibangkitkan dengan membuat penambahan secara terus menerus bentuk yang sama pada setiap sisi inisiatornya ke arah dalam. Penambahan dilakukan dengan cara membagi pada masing-masing sisi inisiator menjadi tiga bagian sama panjang terelebih dahulu, selanjutnya menghilangkan bagian segmen tengahnya sebesar $\frac{1}{3}$ dan kemudian membangkitkan generatornya yaitu menggunakan bentuk segitiga sama sisi baru dengan panjang sisi-sisinya sebesar $\frac{1}{3}$ pada segmen tengah yang telah dihilangkan ke arah dalam. Sehingga akan menghasilkan segitiga sama sisi baru tanpa alas di setiap sisi inisiatornya. Penambahan akan dilakukan secara terus menerus pada setiap segmen yang dihasilkan hingga iterasi ke-i dengan ukuran segmen tengah yang dihilangkan dan generator yang dibangkitkan masing-masing sebesar $\left(\frac{1}{3}\right)^i$ (Riddle, 2017).

2.5 Transformasi Affine

Transformasi affine adalah suatu pemetaan dari satu titik ke titik yang lain dalam suatu bentuk geometri menggunakan aturan konstruksi tanpa mengubah bentuk dasarnya. Konstruksi transfromasi affine dapat berupa rotasi, translasi, dilatasi, refleksi dan *shearing* atau pembebanan (Gunawan dan Suwanda, 2013). Sedangkan dalam penelitian ini hanya menggunakan aturan konstruksi berupa rotasi, dilatasi, dan translasi.



Gambar 2.5 Transformasi affine
(Sumber: Gunawan dan Suwanda, 2013)

2.5.1 Translasi

Translasi merupakan transformasi yang memetakan titik (x, y) yang bergeser sejauh e satuan searah sumbu x dan f satuan searah sumbu y dengan persamaan:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} x + e \\ y + f \end{bmatrix} \tag{2.5}$$

(Kusno, 2003).

2.5.2 Dilatasi

Dilatasi adalah suatu transformasi yang mengubah ukuran (memperbesar atau memperkecil) suatu bangun tetapi tidak mengubah bentuk dasarnya. Misalkan diberikan $T: R^2 \to R^2$ adalah suatu transformasi yang memetakan titik $P(x, y)$ ke $P'(x', y')$ dengan sebuah bilangan real positif k,

$$P' = T(P) = kP \text{ , dengan skala } 0 < k \leq 1 \tag{2.6}$$

2.5.3 Rotasi

Rotasi merupakan pemetaan yang memetakan titik (x, y) ke titik (x', y') dengan gerakan melingkar. Pada dimensi dua, benda akan berputar pada pusat rotasi dan misalkan θ adalah sebuah sudut tetap, maka persamaan rotasi melalui pusat $P(a, b)$ dengan arah rotasi berlawanan arah jarum jam adalah:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x - a \\ y - b \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \tag{2.7}$$

Dan persamaan rotasi searah jarum jam adalah:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x-a \\ y-b \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix} \qquad (2.8)$$

(Kusno, 2003).

Transformasi affine menggunakan fungsi iterasi yang fungsinya terdiri atas dirinya sendiri dengan tak terhingga bentuk pengulangan objeknya atau disebut proses iterasi. Proses iterasi dimulai dari nilai awal, selanjutnya menentukan hasil yang akan mengakibatkan fungsi sebagai masukan dengan perulangannya beberapa kali sehingga menghasilkan sebuah bentuk geometri yang kompleks (Falconer, 2003).

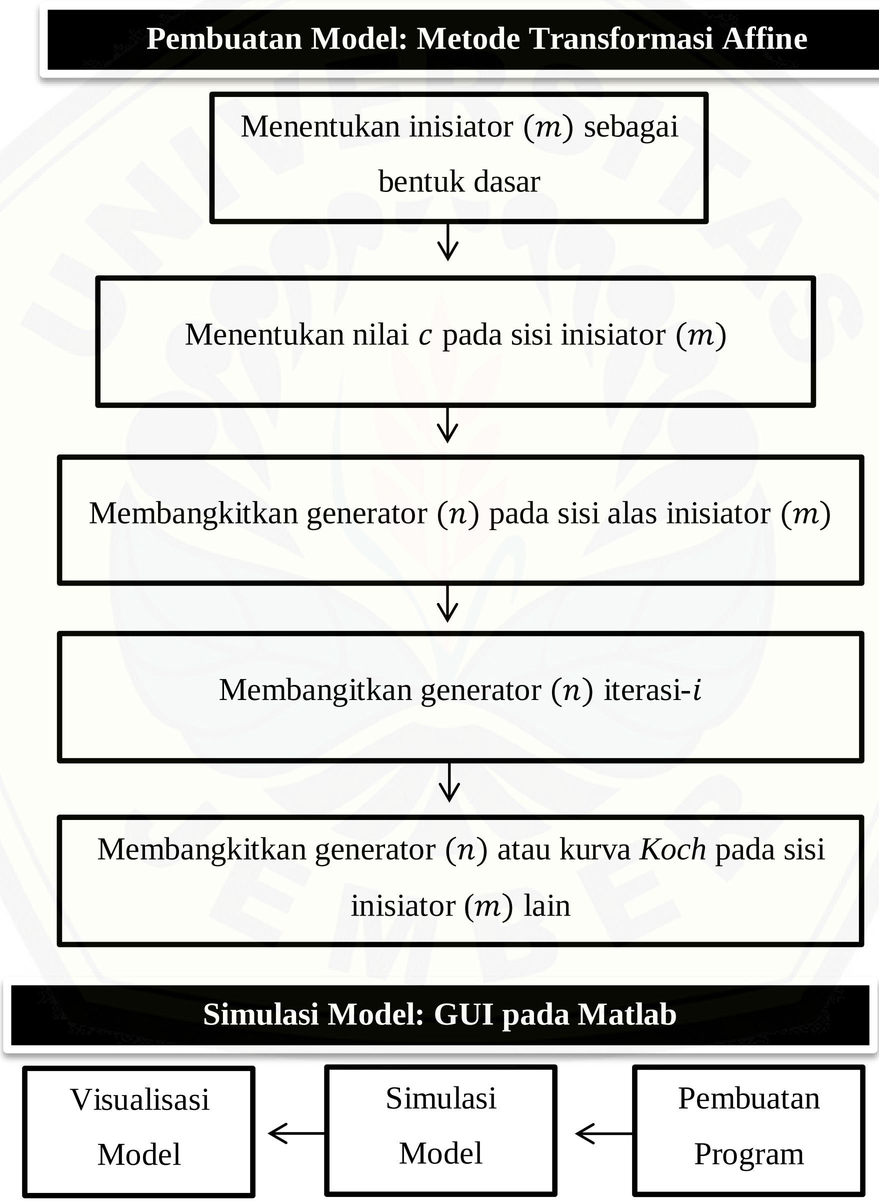
2.6 GUI pada Matlab

Matlab (*Matrix Laboratory*) adalah sebuah ruang lingkup dalam komputasi dan bahasa pemrograman yang dikembangkan oleh Mathwork. Matlab dikembangkan sebagai bahasa pemrograman sekaligus alat visualisasi untuk menyelesaikan berbagai kasus yang berhubungan dengan ilmu matematika. *Graphical User Interface* (GUI) merupakan sebuah tampilan atau visualisasi yang tersedia dalam Matlab yang menyediakan media berupa tampilan grafis sebagai pengganti perintah teks untuk berinteraksi antara user dengan program. Menggunakan GUI dapat menghasilkan tampilan yang lebih menarik dan interaktif serta penggunaan program menjadi lebih efektif (Sugiharto, 2006).

Beberapa komponen standar yang terdapat dalam Matlab untuk membuat tampilan grafis pada GUI antara lain *edit text*, *static text*, *push button*, *axes*, *uitable*, *pop-up menu*, dll. *Edit text* digunakan untuk memasukkan atau memodifikasi suatu input berupa teks atau tulisan sesuai yang diinginkan, sedangkan *static text* hanya dapat menampilkan teks atau tulisan tanpa dapat memodifikasi atau mengubahnya. *Push button* merupakan jenis kontrol berupa tombol tekan yang akan menghasilkan tindakan jika diklik, misalnya tombol OK, Cancel, Hitung, Hasil, dan sebagainya. *Axes* berguna untuk menampilkan sebuah gambar atau grafik, sedangkan *uitable* digunakan untuk menampilkan table. *Pop-up menu* berguna untuk menampilkan beberapa daftar pilihan yang terdefinisi.

BAB 3. METODE PENELITIAN

Bab ini menjelaskan mengenai skema prosedur pembangkitan fraktal *Koch* Anti-*snowflake* (m, n, c) yang terdiri dari pembuatan model menggunakan metode transformasi affine serta simulasi-visualisasi model menggunakan aplikasi GUI pada Matlab:



Gambar 3.1 Skema penelitian

3.1 Membangun Model Transformasi Affine

Berdasarkan skema pada Gambar 3.1, prosedur penelitian dalam membangkitkan fraktal *Koch* Anti-*Snowflake* (m, n, c) menggunakan metode transformasi affine dijelaskan sebagai berikut:

a. Menentukan inisiator (m)

Mengambil titik awal pada sembarang (x, y) yang akan membentuk inisiator (m) sebagai bentuk dasar dengan bentuk poligon segi-m $(3 \leq m \leq 6)$

b. Menentukan nilai c pada sisi alas inisiator (m)

Setelah inisiator (m) terbentuk langkah selanjutnya yaitu menentukan nilai c pada sisi alas inisiator (m) sepanjang L, kemudian membagi segmen sisi alas inisiator (m) tersebut menjadi tiga bagian dengan ukuran segmen tengahnya sebesar c, sehingga akan menghasilkan satu segmen sebesar c pada segmen tengahnya dan dua segmen yang lainnya sebesar a. Kemudian menghilangkan bagian segmen tengahnya tersebut sebagai tempat untuk membangkitkan generator (n).

c. Membangkitkan generator (n) pada sisi alas inisiator (m)

Membangkitkan generator (n) pada sisi alas inisiator poligon-m $(\overline{AB})$ atau akan menghasilkan kurva *Koch*. Generator dibangkitkan pada segmen tengah yang telah dihilangkan menggunakan bentuk generator poligon segi-$n \geq 3$ dengan panjang dari setiap sisi generatornya tersebut adalah sebesar c. Pembangkitan dilakukan menggunakan transformasi affine yaitu dengan aturan konstruksi dilatasi, translasi, dan rotasi sepanjang L:

1. Mendilatasi $\overline{AB}$ sepanjang segmen sisi L di titik A dengan skala dilatasi (diperkecil) sebesar a, sehingga menghasilkan $\overline{A'B'}$.
2. Mendilatasi $\overline{A'B'}$ pada titik B' dengan skala dilatasi (diperkecil) sebesar$\left(\frac{c}{a}\right)$, kemudian hasil dilatasi tersebut dirotasi di titik B' dengan besar sudut $\alpha = (180° - \theta)$ searah jarum jam, sehingga menghasilkan $\overline{A''B''}$.
3. Merotasi $\overline{A''B''}$ sebanyak $(n-2)$ di titik A'' dan B'' secara bergantian dengan arah rotasi berlawanan jarum jam sebesar $\theta = \frac{(n-2)}{n} \times 180°$.

4. Mentranslasi sejauh $(c + a)$ pada arah sepanjang segmen $\overline{A'B'}$ di titik terakhir pada prosedur (3).

d. Membangkitkan generator (n) *Koch* Anti-*Snowflake* iterasi ke-i

Dilakukan proses yang sama pada prosedur affine (b), (c) hingga iterasi-i atau akan menghasilkan kurva *Koch*. Pembangkitan generator (n) dilakukan pada setiap sisi segmen yang dihasilkan dari iterasi sebelumnya dengan skala dilatasi dan translasi yang semakin mengecil.

e. Membangkitkan generator (n) atau kurva *Koch* pada sisi inisiator (m) lain

Setelah pembangkitan generator (n) hingga iterasi-i pada sisi alas inisiator (m) selesai, kemudian bentuk tersebut dibangkitkan pada sisi-sisi inisiator (m) yang lain menggunakan transformasi affine yaitu dengan aturan konstruksi translasi dan rotasi pada $\overline{AB}$ sepanjang L.

3.2 Pembuatan Program, Simulasi-Visualisasi Model

Program yang akan dibuat digunakan sebagai alat untuk membantu dalam simulasi dan juga visualisasi model pembangkitan kurva *Koch* Anti-*Snowflake* (m, n, c) adalah MATLAB R2015b.

a. Pembuatan Program

Program yang akan digunakan dalam penelitian ini adalah menggunakan GUI dalam software MATLAB R2015b.

b. Simulasi Model

Tahap simulasi model yaitu dengan menentukan nilai m untuk mendapatkan bentuk inisiator (m), menentukan nilai n untuk mendapatkan bentuk generator (n), menentukan nilai c untuk ukuran segmen tengah yang akan dihilangkan, dan menentukan banyaknya iterasi yang diinginkan.

c. Visualisasi Model

Selanjutnya pada tahap visualisasi model akan ditampilkan hasil dalam bentuk gambar yang merupakan bentuk fraktal *Koch* Anti-*Snowflake* (m, n, c) dengan inisiator poligon segi-m $(3 \leq m \leq 6)$, menggunakan bentuk generator poligon segi-n $(3 \leq n \leq 6)$ sebesar c yang dibangkitkan pada segmen tengah di setiap sisi inisiator (m).

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang telah dijabarkan pada bab sebelumnya, maka penelitian ini dapat disimpulkan sebagai berikut:

1. Inisiator dengan nilai $3 \leq m \leq 6$ dan generator yang dibangkitkan dengan nilai $3 \leq n \leq 6$ dapat menghasilkan bentuk fraktal *Koch* Anti-*Snowflake* (m, n, c) tidak berpotongan dengan nilai c yang dipilih yaitu setengah dari batas atas nilai c yang diperbolehkan.
2. Inisiator dengan nilai $3 \leq m \leq 6$ dan generator yang dibangkitkan dengan nilai $3 \leq n \leq 6$ dapat menghasilkan bentuk fraktal *Koch* Anti-*Snowflake* (m, n, c) berhimpit dan ada variasi nilai c yang dipilih yaitu c kurang dari setengah batas atas nilai c yang diperbolehkan.
3. Inisiator dengan nilai $3 \leq m \leq 5$ dan generator yang dibangkitkan dengan nilai $n = 3$; inisiator dengan nilai $m = 3$ dan generator yang dibangkitkan dengan nilai $3 \leq n \leq 5$ dapat menghasilkan bentuk fraktal *Koch* Anti-*Snowflake* (m, n, c) berpotongan dengan variasi nilai c yang cukup mendekati dari batas atas nilai c yang diperbolehkan yaitu sebesar $c - 0{,}00001$.

3.2 Saran

Pada penelitian ini telah dilakukan pembangkitan fraktal *Koch* Anti-*Snowflake* (m, n, c) menggunakan metode transformasi affine sebatas menganalisis hasil program. Diharapkan pada penelitian selanjutnya dapat dikembangkan mengenai kajian analitik pembangkitan fraktal *Koch* Anti-*Snowflake* (m, n, c) dalam menentukan nilai (m, n, c) yang dapat menghasilkan bentuk fraktal *Koch* Anti-*Snowflake* berpotongan atau tidak.

DAFTAR PUSTAKA

Bentley, Wilson. 2005-2006. *Serpihan Salju Koch*. Images. Wikipedia: https://id.wikipedia.org/wiki/Serpihan_salju. [Diakses 9 April 2019].

Falconer, K. 1990. *Fractal Geometry*. England: John Wiley and Sons Ltd.

Gunawan, G dan Suwanda. 2013. Transformasi Affine pada Bidang. *Jurnal Matematika*, vol. 12 No.1. Bandung: Universitas Islam Bandung.

Kaleti, T dan Paquette, E. 2010. The Trouble With Von Koch Curves Built From n Gons. *The American Mathematical* Monthly, vol. 117: 124-137.

Kusno. 2003. *Geometri Rancang Bangun Studi Surfas Putar Transformasi Titik dan Proyeksi*. Jember: Fakultas MIPA Universitas Jember.

Mandelbrot, B. B. 1983. *The Fractal Geometry of Nature*. New York: W.H Freeman and Company.

Purnomo, Kosala D. 2016. Kajian Pembentukan Segitiga Sierpinski dengan Transformasi Affine Berbasis Beberapa Benda Geometris. *Prosiding Seminar Nasional Matematika*, vol 6 No 2. Jember: Fakultas Matematika Universitas Jember.

Riddle, L. 2017. *Koch Anti Snowflake*. Images. Georgia: Agness Scott Collage. http://ecademy.agnesscott.edu/~lriddle/ifs/ksnow/kantisnow.htm. [Diakses 10 Januari 2019]

Romadiastri, Y. 2013. Batik Fraktal: Pengembangan Aplikasi Geometri Fraktal. *Jurnal Matematika* Vol. 1(1): 2-25. Semarang: IAIN Walisongo Semarang.

Santosa, P. I. 1994. *Grafika Komputer dan Antarmuka Grafis Teknik Penyusunan Program Aplikasi Berbasis Grafis yang Profesional*. Yogyakarta: CV Andi Offset.

Sugiharto, A. 2006. *Pemrograman GUI dengan Matlab*. Images. Yogyakarta: CV Andi Offset.

Ulfa, H. 2014. *Geometri Fraktal*. Semarang: Semarang State University, Mathematics Education.

Walling, Peter T dan Hicks, Kenneth N. 2003. Dimensions of Consciousness. *Article Proceeding*, vol. 16(2): 162-166. Texas: Baylor University Medical Center.

Wesstein, Eric W. 2008. *Koch Snowflake*. Wolfram Web Resource. Mathworld: http://mathworld.wolfram.com/KochSnowflake.html. [Diakses 20 Mei 2019].

Wesstein, Eric W. 2019. *Koch Anti-Snowflake*. Wolfram Web Resource. Mathworld: http://mathworld.wolfram.com/KochSnowflake.html. [Diakses 2 Januari 2019].

LAMPIRAN

A. Skrip *Koch* Anti-*Snowflake* (m, n, c)

```
function varargout = KochAnti(varargin)
% KOCHANTI MATLAB code for KochAnti.fig
%      KOCHANTI, by itself, creates a new KOCHANTI or raises the
existing
%      singleton*.
%
%      H = KOCHANTI returns the handle to a new KOCHANTI or the
handle to
%      the existing singleton*.
%

%      KOCHANTI('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in KOCHANTI.M with the given input
arguments.
%
%      KOCHANTI('Property','Value',...) creates a new KOCHANTI or
raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before KochAnti_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to KochAnti_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help KochAnti

% Last Modified by GUIDE v2.5 23-Mar-2019 19:33:00

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @KochAnti_OpeningFcn, ...
                   'gui_OutputFcn',  @KochAnti_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
```

```
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before KochAnti is made visible.
function KochAnti_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to KochAnti (see VARARGIN)

% Choose default command line output for KochAnti
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
clc;
movegui(gcf,'center');
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit4,'string','');
set(handles.uitable1,'data',[],'rowname','numbered');
cla(handles.axes1,'reset');
set(handles.axes1,'Fontsize',8,'Fontweight','bold');
% UIWAIT makes KochAnti wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = KochAnti_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
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)
```

```
N1=str2num(get(handles.edit1,'string'));

%Sudut
theta1=(N1-2)*180/N1;

%Poligon
T=[0 1;0 0];
Poly=T;
for i=1:N1-1
    T=[cosd(theta1) sind(theta1);-sind(theta1) cosd(theta1)]*(T-
repmat(T(:,2),1,2))+repmat(T(:,2),1,2);
    T=[T(:,2) T(:,1)];
    Poly=[Poly T(:,2)];
end
Poly(:,end)=Poly(:,1);
set(handles.uitable1,'data',Poly');

% --- Executes on button press in pushbutton2.
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)
Poly=get(handles.uitable1,'data')';
if ~isempty(Poly)
    N1=str2num(get(handles.edit1,'string'));
    N2=str2num(get(handles.edit2,'string'));
    c=str2num(get(handles.edit3,'string'));
    Iterasi=str2num(get(handles.edit4,'string'));

    if mod(N2,2)==0
        ub=sin(pi/N2)^2/(cos(pi/N2)^2+1);
    else
        ub=1-cos(pi/N2);
    end

    if c<=0 || c>=ub
        warndlg(sprintf(['Agar kurva tidak berpotongan/tumpang
tindih' '\n' 'c harus memenuhi 0 < c < ' num2str(ub)]));
    else
        %Sudut
        theta2=(N2-2)*180/N2;
        alpha2=180-theta2;

        %Skala
        s1=(1-c)/2;
        R=1/c;
        s2=2/(R-1);
        KochP=[0;0];
        for i=1:N1

KochP=IFS_KochAnti(Poly(:,i:i+1),N2,s1,s2,theta2,alpha2,Iterasi,Ko
chP);
        end
        KochP(:,end)=KochP(:,1);
```

```
        hold off;
        xl=[min(Poly(1,:)) max(Poly(1,:))];
        yl=[min(Poly(2,:)) max(Poly(2,:))];
        xsel=xl(2)-xl(1);
        ysel=yl(2)-yl(1);
        set(gca,'Xlim',[xl(1)-xsel/10
xl(2)+xsel/10],'Ylim',[yl(1)-ysel/10 yl(2)+ysel/10]);
        set(handles.uitable2,'data',KochP');
    end
end

% --- Executes on button press in pushbutton3.
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)
clc;
frame=getframe(handles.axes1);
image=frame2im(frame);
[name_file,name_path] = uiputfile( ...
        {'*.jpg','File Type JPEG (*.jpg)';
        '*.bmp','File Type BITMAP (*.bmp)';
        '*.tif','File Type TIF (*.tif)';
        '*.png','File Type PNG (*.png)';
        '*.jpg;*.bmp;*.tif;*.png','Files of type
(*.jpg,*.bmp,*.tif,*.png)'},...
        'Save As Image');
if name_file~=0
    imwrite(image,fullfile(name_path,name_file));
end

% --- Executes on button press in pushbutton4.
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)
clc;
movegui(gcf,'center');
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit4,'string','');
set(handles.uitable1,'data',[],'rowname','numbered');
cla(handles.axes1,'reset');
set(handles.axes1,'Fontsize',8,'Fontweight','bold');


function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of
edit1 as a double


% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of
edit2 as a double


% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of
edit3 as a double


% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of
edit4 as a double


% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

B. Skrip metode transformasi affine *Koch* Anti-*Snowflake* (m, n, c)

```
function
KochP=IFS_KochAnti(Segment,n,s1,s2,theta,alpha,iter,KochP)
if iter==0
    KochP=[KochP Segment(:,2)];
    plot(Segment(1,:),Segment(2,:),'b');
    hold on
    axis image
    pause(0.00001);
else
    T=s1*eye(2)*(Segment-
repmat(Segment(:,1),1,2))+repmat(Segment(:,1),1,2);
    Koch=T;
    T=s2*eye(2)*[cosd(alpha) sind(alpha);-sind(alpha)
cosd(alpha)]*(T-repmat(T(:,2),1,2))+repmat(T(:,2),1,2);
    T=[T(:,2) T(:,1)];
    Koch=[Koch T(:,2)];
    for i=1:n-2
        T=[cosd(theta) -sind(theta);sind(theta) cosd(theta)]*(T-
repmat(T(:,2),1,2))+repmat(T(:,2),1,2);
        T=[T(:,2) T(:,1)];
        Koch=[Koch T(:,2)];
    end
    T=1/s2*eye(2)*[cosd(alpha) sind(alpha);-sind(alpha)
cosd(alpha)]*(T-repmat(T(:,2),1,2))+repmat(T(:,2),1,2);
    T=[T(:,2) T(:,1)];
    Koch=[Koch T(:,2)];
    for i=1:size(Koch,2)-1
        KochP=IFS_KochAnti(Koch(:,i:i+1),n,s1,s2,theta,alpha,iter-
1,KochP);
    end
end
```