Bidang Fokus/Unggulan : Teknologi Informasi
Fakultas : MIPA

LAPORAN AKHIR

RISET TERAPAN UNGGULAN UNSRAT (RTUU)

PEMBANGKITAN SEGITIGA SIERPINSKI DAN KURVA *KOCH SNOWFLAKE* DENGAN METODE L-SISTEM PADA DESAIN GRAFIS

TIM PENGUSUL

Ketua Peneliti
Jullia Titaley, S.Pd, M.Si - 0018077204

Anggota Peneliti
Prof. Dr. Benny Pinontoan, M.Sc - 0004066603
Drs. Jantje D. Prang, M.Si – 0020125801

UNIVERSITAS SAM RATULANGI
SEPTEMBER 2019

Daftar Isian Pelaksanaan Anggaran (DIPA) Universitas Sam Ratulangi.
Kementerian Riset, Teknologi dan Pendidikan Tinggi
Nomor SP DIPA-042.01.2.400959/2019, tanggal 05 Desember 2018

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS SAM RATULANGI
LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT
Alamat : Kampus UNSRAT Manado
Telp : (0431) 827560, Fax. (0431) 827560
Email : lppm@unsrat.ac.id Laman : http://lppm.unsrat.ac.id

HALAMAN PENGESAHAN LAPORAN AKHIR
RTUU

Judul Kegiatan **Pembangkitan Segitiga Sierpinski dan Kurva Koch Snowflake Dengan Metode L-Sistem pada Desain Grafis**

Ketua Peneliti

Nama Lengkap : JULLIA TITALEY
Perguruan Tinggi : Universitas Sam Ratulangi
NIP/NIK : 197207182000032001
NIDN : 0018077204
Jab. Fungsional : Lektor
Unit Kerja : Matematika
Nomor HP :
Alamat Email : july_titaley@unsrat.ac.id
Usulan Biaya : 60.000.000
Biaya Maksimum : 51.000.000
Lama Penelitian : 6 bulan

Anggota Peneliti (1)

Nama Lengkap : BENNY PINONTOAN
NIP : 196606041995121001
NIDN : 0004066603
Perguruan Tinggi : Universitas Sam Ratulangi

Anggota Peneliti (1)

Nama Lengkap : JANTJE DENNY PRANG
NIP : 195812201986021001
NIDN : 0020125805
Perguruan Tinggi : Universitas Sam Ratulangi

Mengetahui
Dekan FMIPA UNSRAT

Prof. Dr. Benny Pinontoan, M.Sc
NIP 196606041995121001

Manado, 21 Oktober 2019
Ketua Peneliti

JULLIA TITALEY
NIP 197207182000032001

Menyetujui
Ketua LPPM Universitas Sam Ratulangi

Prof. Dr. Ir. Charles L. Kaunang, MS
NIP 195910181986031002

RINGKASAN

Fraktal *Koch snowflake* dihasilkan oleh prosedur geometris sederhana yang dapat diiterasikan tak terbatas dengan membagi segmen segitiga samasisi menjadi tiga bagian yang sama. Fraktal *Koch snowflake* atau bunga salju memiliki pola yang sangat bagus apabila terus menerus diiterasikan. Di lain pihak segitiga *Sierpinski* adalah fraktal linier yang mempunyai sifat keserupaan diri identik sampai pada iterasi tak-hingga. Pembangkitannya diawali dengan segitiga sama sisi yang berisi warna tertentu. Kemudian titik tengah masing-masing sisinya dihubungkan untuk memperoleh segitiga dengan ukuran setengahnya dan terletak di tengah segitiga awal. Segitiga yang terletak di tengah lalu dihilangkan atau dikosongkan dari segitiga awal. Selanjutnya, pada ketiga segitiga berisi dengan ukuran setengah dari segitiga awal dilakukan proses serupa untuk mendapatkan segitiga dengan ukuran setengahnya lagi. Algoritma seperti ini dilakukan sampai pada iterasi tertentu.

Tahapan pelaksanaan penilitian ini meliputi, membangkitkan kurva Koch Snowflake dan segitiga Sierpinski. Kedua, menentukan pola kedua fractal ini dengan menggunakan transformasi geometri. Beberapa transformasi yang digunakan diantaranya translasi, dilatasi, rotasi, dan refleksi. Ketiga, hasil pembangkitan kedua fractal ini dimodifikasi dalam desain grafis.

PRAKATA

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Kuasa atas berkat dan rahmatNya sehingga seluruh rangkaian kegiatan Penelitian ini dapat diselesaikan dengan baik dan tepat waktu.

Kegiatan Riset Terapan Unggulan Universitas Sam Ratulangi (RTUU) dapat dilaksanakan berkat adanya bantuan dan kerjasama yang sangat baik dari semua pihak yang terlibat. Pada kesempatan ini kami mengucapkan terima kasih sebesar-besarnya kepada :

1. Universitas Sam Ratulangi yang telah memberikan dana untuk pelaksanaan kegiatan penelitian ini
2. Ketua LPPM Unsrat yang telah memberikan persetujuan untuk melaksanakan kegiatan penelitian ini
3. Semua pihak yang tidak dapat disebutkan namanya satu persatu, yang telah membantu terlaksananya penelitian ini.

Kami menyadari bahwa apa yang telah kami lakukan dan hasilkan selama melaksanakan kegiatan penelitian ini masih jauh dari sempurna. Oleh karena itu, kami mengharapkan saran dan kritik yang konstruktif dari semua pihak demi penyempurnaan laporan akhir penelitian ini.

Manado, September 2019

Tim Peneliti

DAFTAR ISI

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

DAFTAR LAMPIRAN

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Kegiatan mengamati keindahan pola telah ada hampir sama lamanya dengan kegiatan mengamati keindahan pemandangan. Keindahan pola yang ada di alam telah banyak diaplikasikan dalam berbagai kesenian di berbagai penjuru dunia. Batik, adalah salah satu dari budaya-budaya kesenian Indonesia yang memiliki unsur keindahan berpola. Bentuk-bentuk yang berpola adalah bahasan utama dari cabang ilmu Geometri, yaitu Fraktal. Penelitian ini berfokus pada pemodelan dan visualisai batik fraktal dengan menggunakan simulasi computer grafis. Pemodelan desain grafis ini bertujuan untuk mengembangkan keragaman pola desain sekaligus sebagai upaya melestarikan kebudayaan. Fraktal berasal dari kata *fractus,* yang berarti pecah. Dalam definisi secara umum, fraktal dapat diartikan sebagai pengulangan bentuk geometri yang dibentuk dari bentuk primitif geometri tersebut yang dipecah atau dibagi ke dalam bentuk dengan skala dan posisi tertentu. Fraktal bermula dari sebuah *chaos* yaitu sebuah bentuk geometri yang memiliki sifat acak, gangguan atau tidak teratur. Bentuk *chaos* tersebut membentuk sebuah pengulangan yang memiliki keteraturan. Sebagai contoh di alam adalah bentuk gunung, awan, pohon, dan lainnya. Konsep fraktal dapat menguraikan sifat fisis yang rumit menjadi elemen yang lebih sederhana. Proses yang lama-kelamaan membentuk suatu keteraturan tertentu, yakni s*elf-similarity* yang menunjukkan bahwa fraktal terdiri dari bagian-bagian yang berbentuk serupa satu sama lain, *self-affinity* menggambarkan bahwa fraktal disusun atas bagianbagian geometri yang saling terangkai satu sama lain, *self-inverse* berarti terdapat satu bagian dalam geometri fraktal yang merupakan susunan yang terbalik dari susunan lainnya, dan *self-squaring* yang dapat diartikan bahwa suatu bentuk geometri fraktal merupakan peningkatan kerumitan dari bagian sebelumnya. Keempatnya merupakan konsep dasar dari geometri fraktal. Berdasarkan aspek geometrisnya, fraktal dapat dikaji melalui keindahan bentuknya sehingga dapat dikembangkan untuk berbagai desain. Konsep fraktal yang demikian yang membuat cabang ilmu Matematika ini berkembang pesat, karena penerapannya yang beragam ke berbagai disiplin ilmu. Salah satu konsep fractal ini dapat diterapkan dan dikembangkan pada desain grafis.

Pekerjaan desain grafis erat hubungannya dengan seni. Seorang desainer juga merangkap seorang seniman. Banyak arti mengenai seni (bergantung pada sudut mana kita melihat). Arti seni secara umum adalah suatu usaha penciptaan bentuk yang menyenangkan (*sense of beauty*) dan harmoni bentuk yang baik. Herbert Read menyebutkan bahwa seni adalah menciptakan plus mengekspresikan bentuk-bentuk yang menyenangkan dan bentuk-bentuk itu menciptakan keindahan. Akan timbul kenikmatan bagi si penikmat seni yang kemudian akan memberikan penghargaan mulai dari empati sampai dengan apresiasi. Seni erat hubungannya dengan keindahan, kreativitas, dan keterampilan.

Salah satu bagian kesenian yang penerapannya berbentuk dua atau tiga dimensi, dikenal dengan istilah seni rupa. Seni rupa merupakan ungkapan gagasan dan perasaan manusia yang diwujudkan melalui pengolahan media dan penataan elemen (yang meliputi unsur *titik, garis, warna, bidang, tekstur, gelap terang)* serta prinsip-prinsip desain. Seni rupa merupakan realisasi dari sebuah imajinasi tanpa batas dan tidak ada batasan, sejatinya, dalam berkarya seni tidak akan kehabisan ide dan imajinasi.

Manfaat Desain Grafis Jika dilihat dan segi kepentingan suatu organisasi (perusahaan, partai, dan sebagainya), desain gratis sangat bermanfaat untuk menyampaikan suatu informasi dan misi sehingga dapat tercapai suatu tujuan yang telah ditetapkan pada saat penetapan tujuan. Suatu organisasi dapat dikatakan berhasil jika pesan yang disampaikan dalam bentuk grafis telah meningkatkan nilai tujuan yang akan dicapai.

Terdapat dua macam kategori grafis yang terdapat pada komputer yaitu grafis berbasis bitmap dan vektor. Kita dapat menggunakan kedua tipe grafis tersebut dan mengolahnya menggunakan aplikasi yang telah disediakan oleh vendorvendor software grafis. Grafis Berbasis Bitmap Bitmap tersusun dan titik-titik kecil dengan sebutan pixel di mana unsur terkecil tersebut akan membentuk suatu kesan yang ditangkap oleh mata sebagai sebuah gambar. Semakin besar pixel yang membentuk grafis berbasis bitmap maka semakin tinggi tingkat kerapatannya yang akan membuat gambar semakin halus atau kualitasnya semakin bagus. Akan tetapi, hal ini akan menyebabkan ukuran file semakin besar. Jika gambar tersebut diperbesar sampai batas maksimal maka akan terlihat jelas unsur dasar gambar berupa satuan kotak-kotak dengan wama-wama yang berbeda. File gambar yang merupakan grafis berbasis bitmap adalah file dengan ekstensi.bmp,.jpg,.gif,.tif,.pcx, dan.pix.

Desain grafis tipe bitmap adalah desain yang sentuhan desainnya dibuat dengan rumus-rumus matematika yang dikerjakan dengan teknologi komputer. Pertama-tama, pola segigia Sierpinski dan Kurva Koch Snowflake ditransformasikan dalam rumus matematika fraktal dengan menggunakan *L-System*. Rumus tersebut kemudian dimodifikasi dengan mengubah parameter-parameternya sehingga menghasilkan rumus yang lebih kompleks dan rumit. Rumus ini akan menghasilkan gambar pola yang berbeda dari pola asli dengan mengubah parameter dalam rumus tersebut (Kudiya, 2009).

Dari uraian di atas, pada penelitian ini penulis akan mengembangkan pola desain grafis dengan membangkitkan desain fraktal *Koch Snowflake* beserta variasinya menggunakan metode *L-System* dan segitiga Sierpinski serta memanfaatkan teknik-teknik transformasi geometri dengan menggunakan *software* Matlab

1.2 Tujuan Khusus

Secara spesifik penelitian ini bertujuan sebagai berikut :

1. Membangkitkan pola segitiga Sierpinski dengan tranformasi
2. Membangkitkan pola Kurva Koch Snowflake dengan metode L-Sistem
3. Menerapkan kedua pola diatas pada desain grafis.

1.3 Target Luaran

Akhir dari penelitian diharapkan dapat dipergunakan oleh para desain grafis lokal untuk menambah kreasi desain yang baru. Selain itu target luaran lain yang ingin dicapai diakhir penelitian ini disajikan pada tabel 1.1 dibawah ini:

Tabel 1.1 Target Luaran

No	Luaran	Target dicapai pada bulan "November 2019"
1	Publikasi Ilmiah di Jurnal Nasional (ber-ISSN)	Submitted
2	Pemakalah dalam Temu Ilmiah Nasional	Terdaftar
3	Karya Seni	Produk
4	HKI	Terdaftar

BAB 2
TINJAUAN PUSTAKA

2.1 Fraktal

Fraktal berasal dari kata *fractus,* kata sifat dalam bahasa Latin yang bersesuaian dengan kata kerjanya *frangere* yang berarti pecah, untuk menciptakan potonganpotongan tak beraturan (Mandelbrot, 1983:4). Fraktal adalah sebuah kajian dalam ilmu matematika yang mempelajari mengenai bentuk atau geometri yang didalamnya menunjukkan sebuah proses pengulangan tanpa batas. Geometri yang dilipat-gandakan tersebut memiliki kemiripan bentuk satu sama lain (*self similiarity*), dan pada penyusunan pelipat-gandaannya tersebut tidak terikat pada satu orientasi, bahkan cenderung meliuk-liuk dengan ukuran yang beragam mulai dari kecil hingga besar.

Berbagai jenis fraktal awalnya dipelajari sebagai benda-benda matematis (Hasang dan Supardjo, 2012). Sifat *self-similarity* ada dua macam fraktal yaitu *regular fractal* dan *random fractal. Regular fractal* mempunyai sifat *exactly self-similarity* yaitu setiap bagian dari objek fraktal menyerupai secara persis dengan bentuk objek secara keseluruhan jika dilihat dari berbagai skala. Contoh objek fraktal yang mempunyai sifat *exactly self-similarity* adalah struktur daun pakis, segitiga Sierpinski, himpunan Cantor. Sedangkan *random fractal* mempunyai sifat *statistically self-similarity* yaitu setiap bagian dari objek fraktal tidak menyerupai secara persis dengan bentuk objek secara keseluruhan. Contoh objek fraktal yang mempunyai sifat *statistically self-similarity* adalah himpunan Julia Set,Mandelbrot dan garis pantai (Addison, 1997:7).

Perkembangan metode matematika dibalik fraktal pertama kali dimulai pada abad ke-17 ketika seorang matematikawan Leibniz melakukan suatu penelitian mengenai bentuk perulangan (*recursive*) bangun yang serupa (*self-affinity*). Namun, Leibniz melakukan sebuah kesalahan dengan memberikan sebuah perkiraan bahwa hanya garis lurus yang memiliki sifat *self-similiar* dalam kasus ini. Hingga pada tahun 1872 ketika Karl Wieerstrass memberikan contoh sebuah fungsi dengan sifat *non-intuitif* yang memiliki kekontiunitasan tetapi tidak *differentiable*. Pada tahun 1904, Helge von Koch yang tidak puas dengan teori dari Wieerstrass dan menyebutnya sangat abstrak serta definisi yang terlalu analitik, memberikan sebuah definisi secara geometris terhadap fungsi yang serupa, yang kemudian dikenal dengan *Koch Snowflake*. Pada tahun 1915, Waclaw Sierpinski membuat sebuah geometri yang disebut dengan Karpet Sierpinski (Mandelbrot, 1983:5). Ide terhadap konsep kurva *self-similiar* dikembangkan lebih lanjut lagi oleh Paul Pierre Levy yang pada tahun 1938 dalam jurnalnya *Plane or Space Curves and Surfaces Consisting of Parts to the Whole* menjelaskan mengenai bentuk kurva fraktal baru yaitu *Levy C Curve*. George Cantor juga memberikan contoh dari sebuah himpunan yaitu *Cantor Set* yang juga termasuk fraktal. Pada tahun 1960, Benoit B. Mandelbrot memulai investigasinya mengenai *self-similarity* pada jurnalnya *HowLong is The Coast of Britain? Statistical Self-Similarity and Fractional Dimension.* Akhirnya Mandelbrot (1983:15) memberikan sebuah kesimpulan mengenai definisi fraktal sebagai bentuk geometri yang memiliki nilai dimensi Haussdorff-Besicovitch lebih tinggi daripada nilai dimensi topologisnya.

2.2 Fraktal *Koch Snowflake*

Koch Snowflake atau yang sering disebut bunga salju Koch ditemukan oleh matematikawan dari Swedia, Helge Von Koch pada tahun 1904.

Kurva *Koch* mempunyai karakteristik menarik, yaitu :

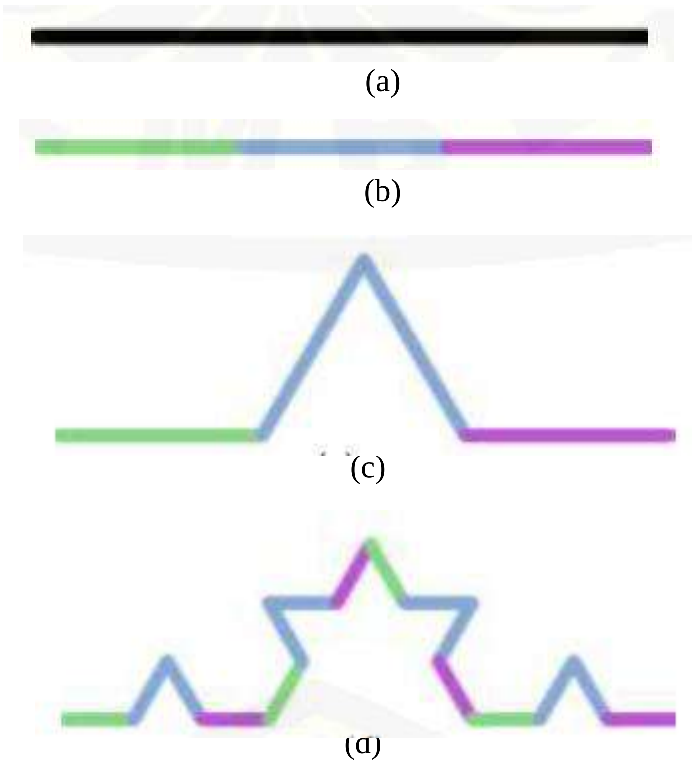
a. Masing-masing segmen adalah pertambahan panjang $\frac{4}{3}$ dari sebuah factor. Oleh karena itu, K_{n+1} adalah $\frac{4}{3}$ sepanjang K_n dan K_i mempunyai total panjang $\left(\frac{4}{3}\right)^i$

b. Ketika *n* bertambah besar, kurva masih tampak mempunyai bentuk yang sama.

c. Ketika *n* menjadi tanpa batas, kurva mempunyai suatu panjang tanpa batas, sedangkan areanya terbatas

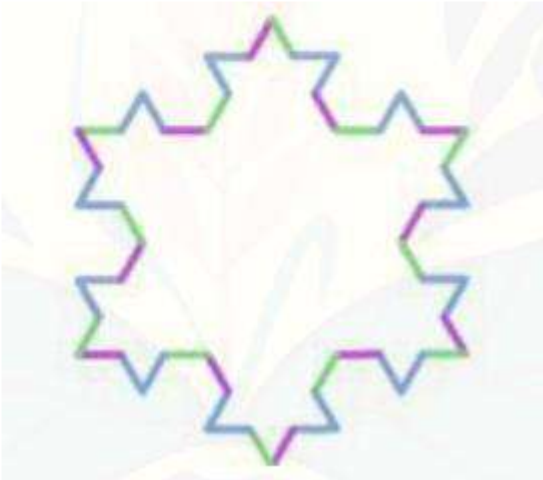
(Kusuwawati *et al*, 2009)

Fraktal *Koch Snowflake* merupakan hasil dari modifikasi kurva *Koch* yang mana didasarkan pada garis-garis yang mempunyai arah tertentu dan dihubungkan satu sama lain, sehingga terbentuk suatu garis yang sangat panjang pada suatu daerah yang terbatas. Langkah-langkah pembentukan kurva *Koch* dimulai dengan sebuah garis lurus (Gambar 3.1 a). Untuk membentuk kurva *Koch* orde satu, K_1, garis tersebut dibagi menjadi tiga bagian (Gambar 3.1 b), kemudian bagian tengah diubah menjadi segitiga samasisi tanpa alas sehingga menjadi bentuk bangun dengan empat buah segmen garis (Gambar 3.1 c). Selanjutnya untuk membentuk kurva *Koch* orde dua K_2, lakukan hal yang sama seperti sebelumnya pada setiap segmen garis dari K_1. Yaitu membagi setiap segmen menjadi tiga bagian baru, lalu bagian tengah diubah menjadi segitiga samasisi tanpa alas sehingga akan terbentuk bangun dengan empat segmen garis baru di setiap segmen garis pada bangun K_1 (Gambar 3.1 d). Dengan cara yang sama, kurva *Koch* untuk orde yang lebih tinggi (K_i) bisa diperoleh dari modifikasi kurva *Koch* orde sebelumnya(K_{i-1}).



Gambar 3.1 Fraktal *Koch Snowflake*

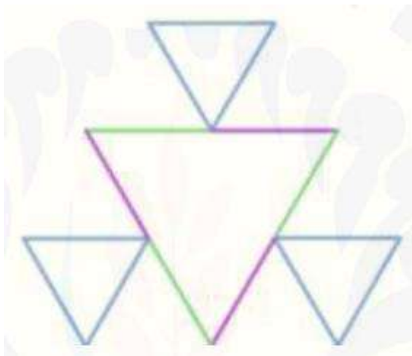
Fraktal *Koch Snowflake* (Gambar 3.2) dibangun dari kurva *Koch* yang dibangkitkan pada sisi-sisi segitiga samasisi, didasarkan pada garis-garis yang mempunyai arah tertentu dan dihubungkan satu sama lain. Variasi *Koch Snowflake* dapat dilakukan dengan cara menggabungkan antara inisiator (bentuk dasar) dan generatornya (bentuk perulangan) (Kamil, 2004).



Gambar 3.2 Fraktal *Koch Snowflake*

2.3 Pola Bentuk *Koch Snowflake*

Pada dasarnya Pada dasarnya variasi dari *Koch Snowflake* ini dapat dilakukan dengan cara membangkitkan berbagai generator (hasil dari *production rule*) pada sebarang segmen garis pada sisi bidang inisiator (bentuk awal). Pengembangan variasi *Koch Snowflake* diantara jenisnya adalah: luas bertambah, luas berkurang, dan luas tetap (Kamil, 2004). Variasi *Koch Snowflake* dengan luas bertambah dilakukan dengan cara membangkitkan generatornya ke arah luar dari sisi-sisi inisiator seperti tampak pada Gambar 3.3. Variasi *Koch Snowflake* dengan luas berkurang dikenal juga dengan sebutan *Anti Koch Snowflake,* sesuai dengan namanya variasi ini dibangkitkan dengan cara yang berlawan dengan variasi luas bertambah. Yaitu dengan cara membangkitkan generator pada sisi bidang inisiator ke arah dalam atau pusat seperti dalam Gambar 3.4. Selanjutnya adalah variasi *Koch Snowflake* dengan mempertahankan luas (luas tetap), variasi yang merupakan perpaduan antara *Koch Snowflake* luas bertambah dan *Koch Snowflake* luas berkurang (Gambar 3.5).



Gambar 3.3 Variasi *Koch Snowflake*
Luas Bertambah dengan inisiator dan generator segitiga samasisi

2.4 Segitiga Sierpinski

Segitiga Sierpinski adalah fraktal linier yang mempunyai sifat keserupaan diri identik sampai pada iterasi tak-hingga. Pembangkitannya diawali dengan segitiga sama sisi yang berisi warna tertentu. Kemudian titik tengah masing-masing sisinya dihubungkan untuk memperoleh segitiga dengan ukuran setengahnya dan terletak di tengah segitiga awal. Segitiga yang terletak di tengah lalu dihilangkan atau dikosongkan dari segitiga awal. Selanjutnya, pada ketiga segitiga berisi dengan ukuran setengah dari segitiga awal dilakukan proses serupa untuk mendapatkan segitiga dengan ukuran setengahnya lagi. Algoritma seperti ini dilakukan sampai pada iterasi tertentu. Pada setiap iterasi didapatkan fakta bahwa satu segitiga dibagi menjadi empat segitiga (dengan ukuran sisi setengahnya) yang terdiri atas 3 segitiga berisi warna dan 1 segitiga kosong. Dengan rumusan ini, luas segitiga Sierpinski pada iterasi ke-*n* adalah $\left(\frac{3}{4}\right)^n$ dari luas awalnya. Jika prosesnya diteruskan sampai iterasi mendekati tak-hingga, luas segitiga Sierpinski akan mendekati nol.

Penelitian ini bertujuan membahas algoritma pembangkitan segitiga Sierpinski dengan memanfaatkan transformasi affine dalam bentuk dilasi dan translasi. Dalam hal ini segitiga yang dibangkitkan untuk mengisi bentuk dasar segitiga Sierpinski terdiri atas dua macam, yaitu segitiga yang berisi warna dan segitiga kosong. Pada masing-masing segitiga ini dilakukan dilasi dan translasi untuk mendapatkan segitiga Sierpinski. Pada bagian selanjutnya akan dilakukan beberapa modifikasi segitiga Sierpinski dengan mengganti bentuk dasar segitiga sama sisi dengan beberapa benda geometris lainnya

Algoritma pembangkitan segitiga Sierpinski akan dilakukan dalam dua cara. Pertama, dilakukan transformasi affine pada benda geometris yang terisi warna. Pada algoritma ini dalam tiap iterasi segitiga Sierpinski yang terbentuk akan didilasi ½ dan kemudian ditranslasi pada titik tengah sisi yang menghubungkan titik sudut bentuk dasar dengan titik pusat dilasi. Kedua, dilakukan transformasi affine pada bagian yang tidak berwarna atau kosong. Dalam hal ini, algoritma kedua hanya akan diterapkan pada bentuk dasar dan benda geometris segitiga.

Algoritma pertama akan digunakan untuk membangkitkan segitiga Sierpinski dengan berbagai variasi bentuk dasar dan benda geometris. Misalkan diberikan bentuk dasar segitiga Sierpinski dan benda geometris dengan warna tertentu. Bentuk dasar dan benda geometris ini dinyatakan dengan titik sudut tertentu. Pembangkitan segitiga Sierpinski dengan memanfaatkan transformasi affine dalam bentuk dilasi dan translasi. Dilasi dilakukan dengan skala ½ dan menjadikan salah satu titiknya sebagai pusat dilasi. Diasumsikan pusat dilasinya di titik (0,0). Translasi dilakukan sesuai dengan bentuk dasar yang dipilih. Dalam hal ini ada dua kasus yang akan dibahas:

a. Bentuk dasar segitiga dengan benda geometris segitiga;
b. Bentuk dasar segitiga dengan benda geometris segiempat;

2.5 *L-System*

L-System adalah sebuah metode penulisan secara paralel yang dikembangkan oleh Aristid Lindenmayer (1925-1989) seorang peneliti biologi dan botani di Hungaria pada tahun 1968. *L-System* dapat juga disebut sebagai sebuah *formal grammar* yang terdiri dari beberapa simbol dan aturan. *L-System* secara umum digunakan untuk membentuk model proses pertumbuhan pada sebuah tanaman, namun dapat juga digunakan sebagai morfologi dari varietas makhluk hidup. *L-System* juga dapat digunakan untuk membuat *self-similar fractal* dan merupakan salah satu metode untuk menghasilkan sebuah fraktal (Prusinkiewicz dan Lindenmayer, 1990). Secara umum *L-System* adalah bentuk notasi dari sebuah perulangan tulisan dimana ide dasarnya adalah membentuk sebuah objek dengan menukar atau mengganti beberapa bagian pada sebuah aturan melalui mekanisme perulangan.

Pengulangan pada aturan *L-System* merujuk kepada sebuah *self-similarity* dan untuk itu bentuk fraktal dapat dibuat dengan mudah menggunakan *L-System*. Tata bahasa atau *grammar L-System* hampir serupa dengan *semi-Thue grammar* dan juga sekarang lebih dikenal sebagai parametrik *L-System* yang diartikan sebagai *tuple*.

Berikut adalah *Grammar* dari *L-System.*

$G = \{V, S, \omega, P\}$,

dimana:

a. ***V*** (the *alphabet*) adalah himpunan dari beberapa simbol variabel yang mengandung elemen yang dapat diganti oleh variabel lain;
b. ***S*** adalah himpunan dari beberapa simbol yang konstan, yang tidak dapat diganti oleh simbol lain;
c. *ω (start, axiom atau initiator*) adalah sebuah inisial awal dari system berupa string yang mengandung V dan S
d. ***P*** adalah sebuah himpunan dari *production rules* yang menjelaskan bagaimana setiap variabel dapat diubah dengan kombinasi dari variabel lain, mengandung dua buah string yaitu *predecessor* dan *successor.*

Aturan pada *L-System* diterapkan secara berulang dimulai dari sebuah pernyataan awal (*intial state*). *Rule* tersebut diulang sesuai dengan jumlah iterasi yang diinginkan *user*. *L-System* adalah sebuah *context-free grammar* dimana setiap *production rule* hanya berlaku untuk satu symbol saja pada sebuah set. Simbol yang lain tidak terpengaruh dengan *production rule* tersebut. Hal ini disebut kelas *D*0*L-System* (*Deterministic and 0-contex*t /*context-free*) (Dickau, 1996). Sebagai contoh, ada dua buah variabel *A* dan *B* dimana untuk setiap variable tersebut diberikan sebuah aturan produksi atau *production rule*. Aturan tersebut adalah *A→AB* dan *B→A*, maksudnya adalah untuk setiap perulangan huruf *A* akan diganti dengan *AB*, sedangkan huruf *B* akan diganti oleh huruf *A*. Sebuah pernyataan awal (*initial state*) disebut *axiom*. Pada langkah pertama, asumsikan terdapat *axiom* dengan huruf *A* saja. Kemudian pada perulangan huruf tersebut diganti dengan *AB* merujuk pada aturan *A→AB*. Langkah berikutnya, huruf *B* tersebut diganti dengan *A* sesuai aturan $B \rightarrow A$. Kedua huruf tersebut pada langkah selanjutnya akan diganti sesuai aturan yang telah dibuat, dan proses tersebut berlangsung terus secara berulang sesuai dengan jumlah perulangan yang diinginkan.

2.6 Desain Grafis

Kata *desain* memiliki arti merancang atau merencanakan. Kata *grafis* sendiri mengandung dua pengertian: (1) *graphien* (Latin = garis, marka) yang kemudian menjadi *graphic arts* atau komunikasi grafis, (2) *graphise vakken* (Belanda = pekerjaan cetak) yang di Indonesia menjadi grafika, diartikan sebagai percetakan. Jadi, pengertian desain grafis adalah pekerjaan dalam bidang komunikasi visual yang berhubungan dengan grafika (cetakan) dan/ atau pada bidang dua dimensi, dan statis (tidak bergerak dan bukan *time-based image*). Secara khusus, desain grafis adalah keahlian menyusun dan merancang unsur visual menjadi informasi yang dimengerti publik/masyarakat. Bidang profesi desain grafis menangani konsep komunikasi grafis, merancang, dan meyelaraskan unsur yang ditampilkan dalam desain (huruf, gambar, dan atau foto, elemen grafis, warna) sesuai dengan tujuan komunikasi, dan mengawasi produksi (cetak). Dalam kerjanya, desainer grafis memberi *brief* dan pengarahan kepada ilustrator atau fotografer agar hasil yang diperoleh sesuai dengan rancangan desainnya. Bidang profesi desain grafis meliputi kegiatan penunjang dalam kegiatan penerbitan (*publishing house*), media massa cetak koran dan majalah, dan biro grafis (*graphic house*, *graphic boutique*, *production house*). Selain itu, desain grafis juga menjadi penunjang pada industri nonkomunikasi (lembaga swasta/pemerintah, pariwisata, hotel, pabrik/manufaktur, usaha dagang) sebagai *inhouse graphics* di departemen promosi ataupun tenaga grafis pada departemen hubungan masyarakat perusahaan.

BAB 3
TUJUAN DAN MANFAAT PENELITIAN

3.1 Tujuan Khusus

Secara spesifik penelitian ini bertujuan sebagai berikut :

1. Membangkitkan pola segitiga Sierpinski dengan tranformasi
2. Membangkitkan pola Kurva Koch Snowflake dengan metode L-Sistem
3. Menerapkan kedua pola diatas pada desain grafis.

3.2 Manfaat Penelitian

Manfaat dari hasil akhir penelitian ini diharapkan dapat memberikan manfaat kepada pembatik dan masyarakat umum sebagai berikut :

1. Memperkaya desain grafis Indonesia, dan
2. Memperkenalkan fraktal sebagai salah satu motif baru dalam desain grafis

BAB 4
METODE PENELITIAN

4.1. Alat dan Sumber Data

a. Data yang digunakan untuk mendapatkan pengetahuan tentang pola-pola desain grafis yang ada di Sulawesi Utara.

b. Alat yang digunakan

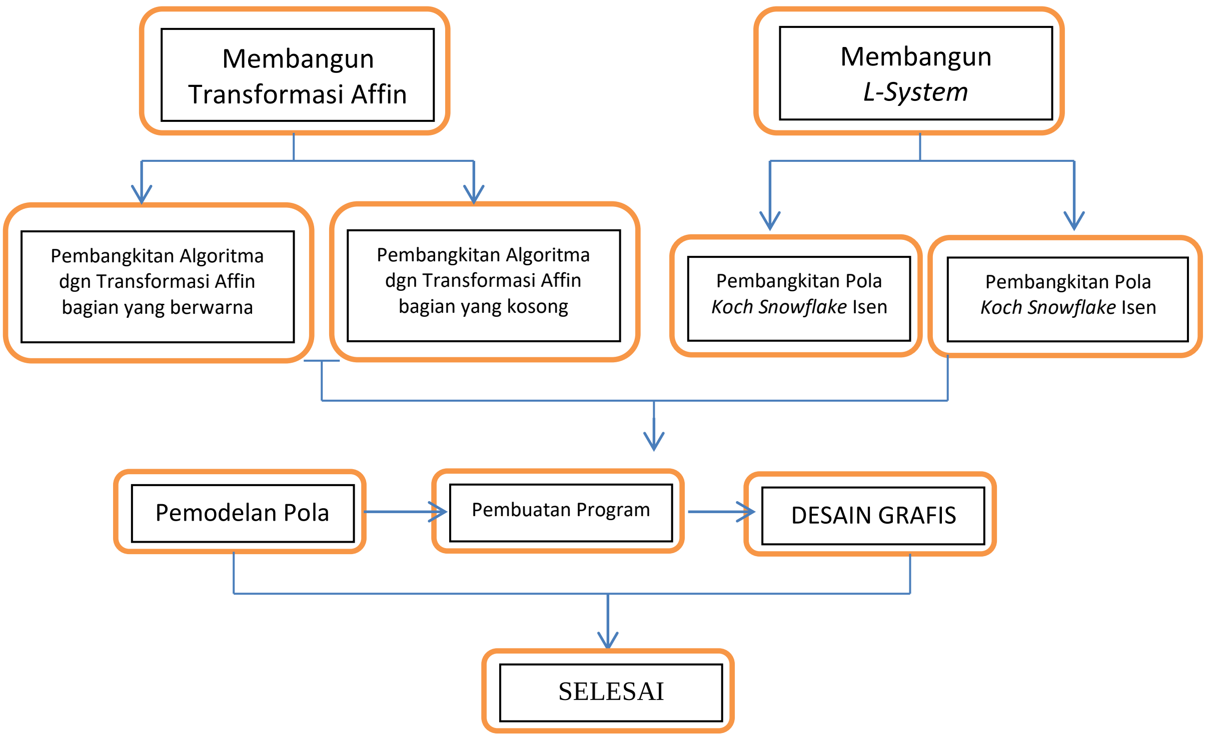
Untuk kegiatan penelitian ini, alat-alat yang digunakan adalah sebagai berikut :

1. Perangkat keras : Sebuah notebook dengan spesifikasi prosesor minimal Intel inside Core i5, 1.66 GGB, 500Gb HDD, 1 GB DDR2
2. Perangkat lunak : MatLab, Software Jbatik

4.2. Tahapan Penelitian

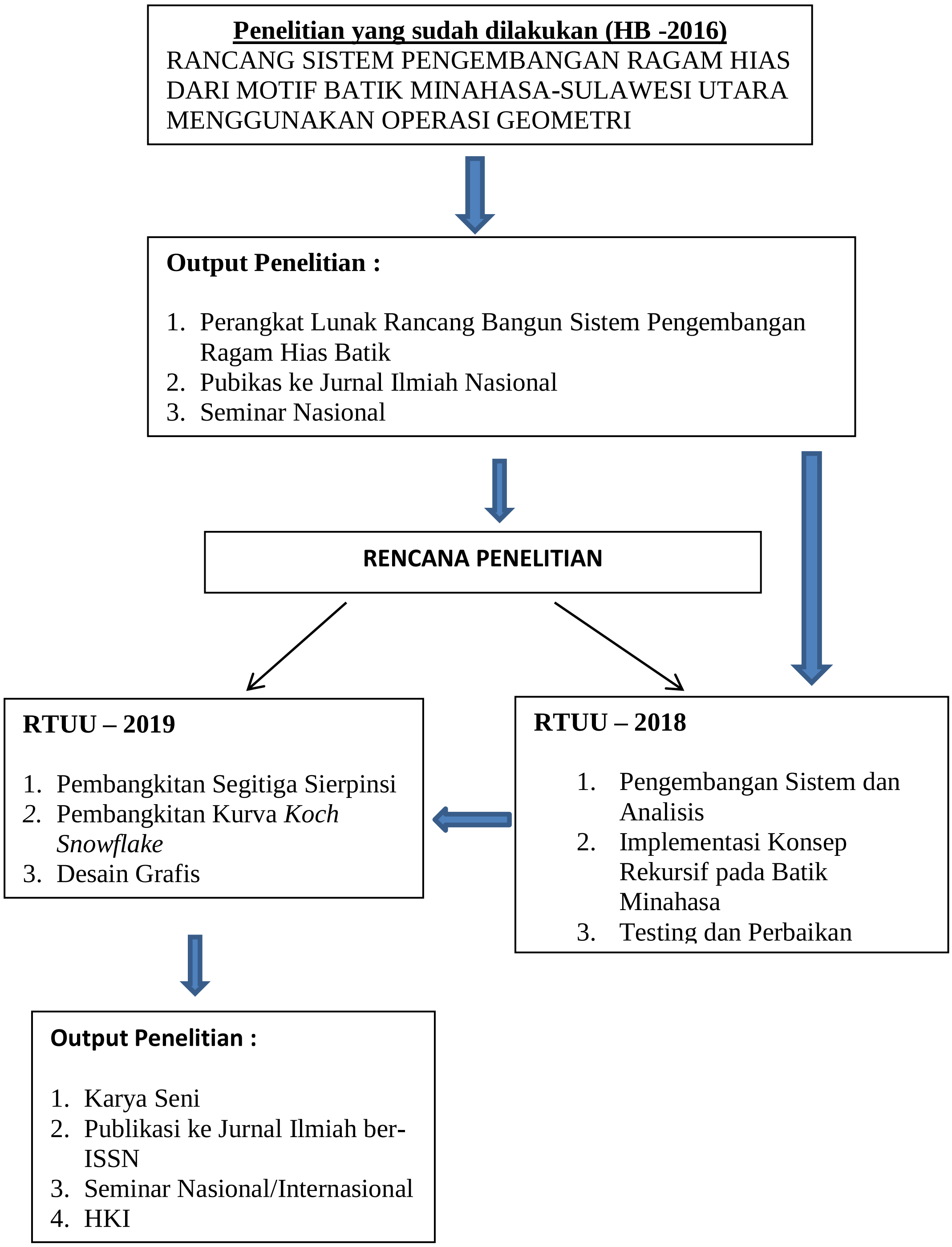
a. Metode dalam penelitian ini dilakukan dalam beberapa tahap yaitu :

Berikut ini merupakan skema metode penelitian untuk menggabungkan pola Segitiga Sierpienski dan fraktal *Koch Snowflake* dengan beberapa jenisnya untuk membentuk sebuah Pola Desain Grafis (lihat Gambar 4.1).



Gambar 4.1. Tahapan Penelitian

b. Diagram Tahapan Penelitian

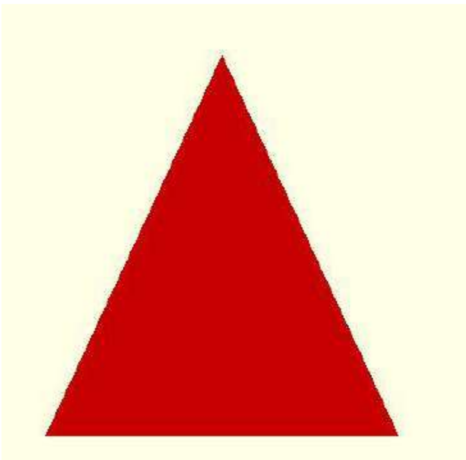


BAB 5
HASIL DAN PEMBAHASAN

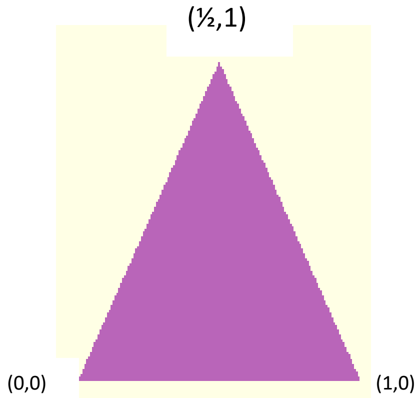
5.1 Algoritma Pembangkitan Sierspinskie

Algoritma pembangkitan sierpinskie akan dilakukan dalam dua cara. Pertama dilakukan transformasi affine pada segitiga sama sisi yang terisi warna. PadaPertama, dilakukan transformasi affine pada benda geometris yang terisi warna. Pada algoritma ini dalam tiap iterasi segitiga Sierpinski yang terbentuk akan didilasi ½ dan kemudian ditranslasi pada titik tengah sisi yang menghubungkan titik sudut bentuk dasar dengan titik pusat dilasi. Kedua, dilakukan transformasi affine pada bagian yang tidakberwarna atau kosong. Dalam hal ini, algoritma kedua hanya akan diterapkan pada bentukdasar dan benda geometris segitiga.

Untuk mengilustrasikan perumusan algoritma pertama, pandang bentuk dasar segitiga dengan benda geometris juga segitiga. Misalkan pada Gambar 1 diberikan bentuk dasar segitiga dengan titik sudut (0,0), (1,0), dan (½,1). Sebuah benda geometris berbentuk segitiga berwarna (dalam hal ini hijau) akan digunakan mengisi bentuk dasar ini.

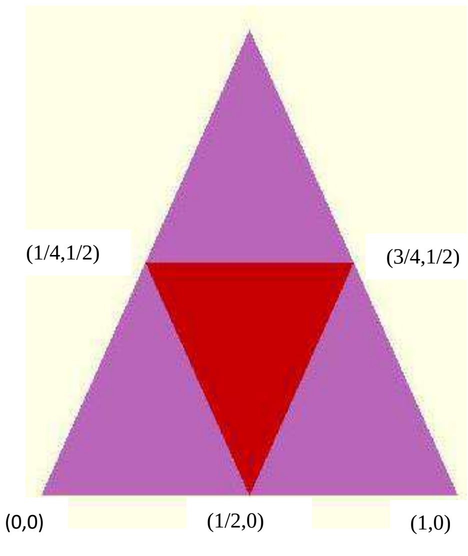


Gambar 5.1. Bentuk Dasar Segitiga



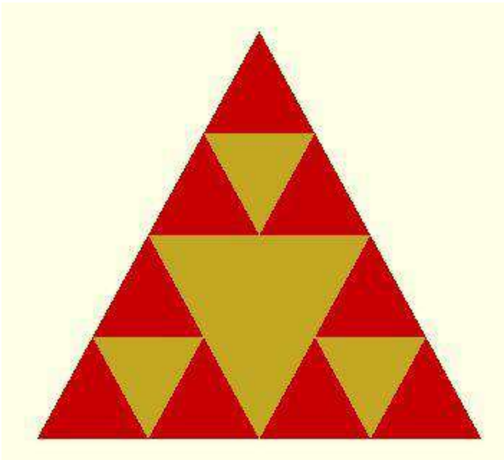
Gambar 5.2. Benda Geometris Segitiga

Benda geometris pada gambar 2 didilasi berpusat di (0,0) dengan skala dilasi ½. Kemudian hasil dilasi ini diduplikasi menjadi dua segitiga lain dengan translasi $\begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}$ dan $\begin{pmatrix} \frac{1}{4} \\ \frac{1}{2} \end{pmatrix}$. Diperoleh benda geometris baru pada gambar 3 yang disebut sebagai segitiga Sierpinskie pada iterasi 1 yang mengisi bentuk dasar pada gambar 1.



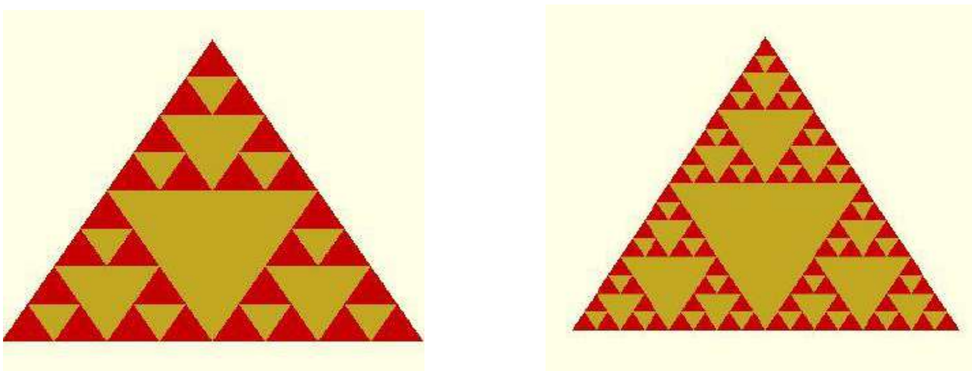
Gambar 5.3. Segitiga Sierspinskie pada iterasi 1

Benda geometris (yaitu segitiga Sierpinski pada iterasi 1) pada Gambar 3 kemudian didilasi lagi dengan skala ½ sehingga menghasilkan segitiga Sierpinski dengan ukuransetengahnya. Kemudian diduplikasi lagi untuk mendapatkan dua segitiga Sierpinski lain dengan translasi $\begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}$ dan $\begin{pmatrix} \frac{1}{4} \\ \frac{1}{2} \end{pmatrix}$. Hasilnya adalah segitiga Sierpinski pada iterasi 2 (gambar 4).



Gambar 5.4. Segitiga Sierspinskie pada iterasi 2

Demikian seterusnya. Segitiga sierspinskie pada iterasi 3 dan 4 dapat dilihat pada gambar 5.



Gambar 5.5. Segitiga Sierspinskie pada iterasi 3 dan itearsi 4

Dengan mengacu pada langkah-langkah di atas, dapat dirumuskan algoritma pertama untuk membangkitkan segitiga Sierpinski berikut.

a. Berikan bentuk dasar segitiga yang akan dijadikan acuan serta benda geometris segitiga berwarna yang akan mengisi bentuk dasar tersebut. Nyatakan segitiga berwarna sebagai himpunan titik-titik (x,y)

b. Berikan faktor skala dilasi x dan vector translasi $\binom{u_x}{u_y}$ dan $\binom{v_x}{v_y}$. Dalam hal ini untuk segitiga sierpinskie konvensional diambil

c. Tentukan benda geometris segitiga baru hasil dilasi dengan pusat di (0,0) dan factor skala s, sehingga titik *(x,y)* akan ditransformasikan menjadi *(sx, sy).*

d. Tentukan duplikasi benda geometris segitiga pada langkah c dengan vector translasi pada langkah b, yaitu didapatkan titik ($sx + u_x$, $sy + u_y$) dan ($sx + v_x$, $sy + v_y$)

e. Definisikan segitiga Sierpinski yang merupakan gabungan segitiga dari benda geometris baru pada langkah c dan d

f. Ulangi langkah c sampai pada iterasi yang diinginkan untuk mendapatkan segitiga Sierpinski pada iterasi *n*

Pembangkitan segitiga kosong yang ada dalam segitiga sierspinkie secara umum dapat dirumuskan sebagaimana pada Tabel 2. Dalam hal ini diasumsikan bahwa panjang sisi dari bentuk dasar segitiga sama sisi adalah satu satuan.

Ta**bel 2**. Beberapa ukuran transformasi affine pada segitiga Sierpinski

Iterasi	Nama Segitiga Kosong	Jumlah Segitga Kosong	Ukuran sisi segitiga kosong	Skala Dilasi	Vektor Translasi pertama	Vektor Translasi pertama
1	Δ_1	1	1/2	Tidak ada	Tidak ada	Tidak ada
2	Δ_1	1	$\frac{1}{2}$	Tidak ada	Tidak ada	Tidak ada
	$\Delta_{11}, \Delta_{12}, \Delta_{13}$	3	$\frac{1}{2^2}$	$\frac{1}{2}$	$\begin{pmatrix}\frac{1}{2}\\0\end{pmatrix}$	$\begin{pmatrix}\frac{1}{2^2}\\\frac{1}{2}\end{pmatrix}$
3	Δ_1	1	$\frac{1}{2}$	Tidak ada	Tidak ada	Tidak ada
	$\Delta_{11}, \Delta_{12}, \Delta_{13}$	3	$\frac{1}{2^2}$	$\frac{1}{2}$	$\begin{pmatrix}\frac{1}{2}\\0\end{pmatrix}$	$\begin{pmatrix}\frac{1}{2^2}\\\frac{1}{2}\end{pmatrix}$
	$\Delta_{11i}, \Delta_{12i}, \Delta_{13i}$	3^2	$\frac{1}{2^3}$	$\frac{1}{2^2}$	$\begin{pmatrix}\frac{1}{2^2}\\0\end{pmatrix}$	$\begin{pmatrix}\frac{1}{2^2}\\\frac{1}{2^2}\end{pmatrix}$
…	…	…	…	…	…	…

Berdasarkan pada Tabel 2 dapat dirumuskan langkah-langkah membangkitkan segitiga Sierpinski berikut. Algoritma kedua ini bertujuan membangkitkan segitiga kosong pada segitiga Sierpinski.

1. Berikan bentuk dasar segitiga berwarna (beri nama Δ_0) yang akan diisi dengan segitiga kosong.
2. Tentukan segitiga kosong (atau warna putih) yang titik sudutnya adalah titik tengah masing-masing sisi segitiga berwarna. Beri nama segitiga ini dengan Δ_1. Hasil dari Δ_0 - Δ_1 membentuk segitiga sierpinski iterasi 1.
3. Lakukan dilasi pada segitiga kosong Δ_1 dengan skala ½ maka akan didapatkan Δ_{11} dengan ukuran sisi $\frac{1}{2^2}$.
4. Lakukan tranlasi pertama dan kedua Δ_{11} dengan vector tranlasi $\begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}$ dan $\begin{pmatrix} \frac{1}{2^2} \\ \frac{1}{2} \end{pmatrix}$ untuk mendapatkan segitiga kosong Δ_{12} dan Δ_{13}. Jika gabungan segitiga kosong ini diberi nama $\Delta_{11} + \Delta_{12} + \Delta_{13}$, maka hasil dari Δ_0 - Δ_1 – ($\Delta_{11} + \Delta_{12} + \Delta_{13}$) membentuk segitiga Sierpinski iterasi 2.
5. Lakukan dilasi pada segitiga kosong Δ_1 dengan skala $\frac{1}{2^2}$, maka akan di dapatkan Δ_{111} dengan ukuran sisi $\frac{1}{2^3}$.
6. Lakukan translasi pada Δ_{111} dengan vector translasi $\begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}$ dan $\begin{pmatrix} \frac{1}{2^2} \\ \frac{1}{2} \end{pmatrix}$ untuk mendapatkan segitiga kosong Δ_{112} dan Δ_{113}.
7. Translasikan gabungan segitiga kosong $\Delta_{111} + \Delta_{112} + \Delta_{113}$, maka hasil dari perhitungannya membentuksegitiga Sierpinski iterasi ke 3
8. Demikian seterusnya dilakukan sampai iterasi ke-n

Dari algoritma pertama dan kedua yang dirumuskan diatas terlihat bahwa langkah-langkah pada algoritma pertama lebih sederhana. Skala dilasi dan vector translasi yang digunakan pada algoritma pertama juga lebih mudah diingat karena selalu tetap untuk membangkitkan segitiga manapun dalam tiap iterasi. Sedangkan dalam tiap iterasi algoritma kedua besaran tersebut selalu berubah.

5.2. **Algoritma Segitiga Sierpinski Menggunakan OpenScad**

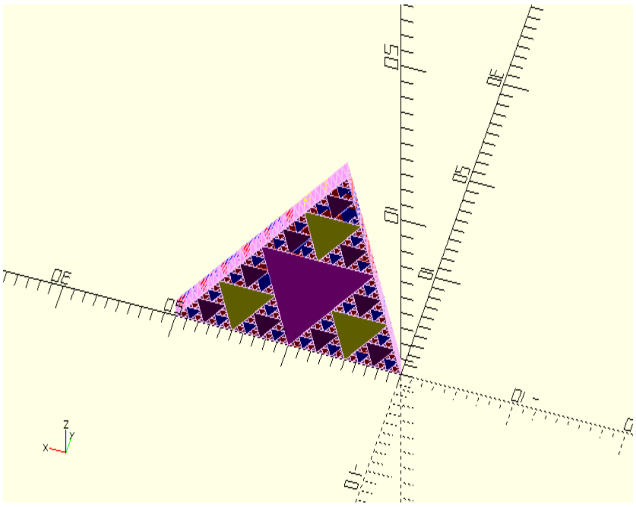
Berikut disajikan algoritma Segitiga Sierpinski menggunakan software gratis Openscad. Software ini dapat diunduh gratis pada situs :

github.com/openscad/openscad

Output dari software ini adalah bentuk geometri Dimensi tiga.

```
coding menggunakan "Openscad"
$fn segitiga sierspinski
p1 = [0,0];
p2 = [20,0];
p3 = [10,sqrt(300)];
module triangle(p1,p2,p3,hgt,cl) {
    color(cl)
    linear_extrude(hgt)
    polygon([p1,p2,p3]);
}
function mid(p1,p2) = [(p1[0] + p2[0])/2,(p1[1] + p2[1])/2];
module sierpinski(p1,p2,p3,hgt,degree) {
    c =
["violet","red","blue","purple","yellow","magenta","yellow"]
;
    triangle(p1,p2,p3,hgt,c[degree]);
    if (degree > 0) {
        sierpinski(p1,mid(p1,p2),mid(p1,p3),hgt,degree -1);
        sierpinski(p2,mid(p1,p2),mid(p2,p3),hgt,degree -1);
        sierpinski(p3,mid(p1,p3),mid(p2,p3),hgt,degree -1);
    }
}
//Main
```

```
sierpinski(p1,p2,p3,1,5);
```



```
$fn = 6;

module bar(v1, v2, diam) {
    p = v2 - v1;
    R = sqrt(p.x * p.x + p.y * p.y);
    L = sqrt(R * R + p.z * p.z);
    translate(v1)
        rotate(atan2(R, p.z), [-p.y, p.x, 0])
            cylinder(h=L, d=diam);
}

Aa = [1, 0, 0];
Ba = [-1/2, sqrt(3) / 2, 0];
Ca = [-1/2, -sqrt(3) / 2, 0];
Da = [0, 0, sqrt(2)];
center = 0.25 * (Aa + Ba + Ca + Da);
A = Aa - center;
B = Ba - center;
C = Ca - center;
```

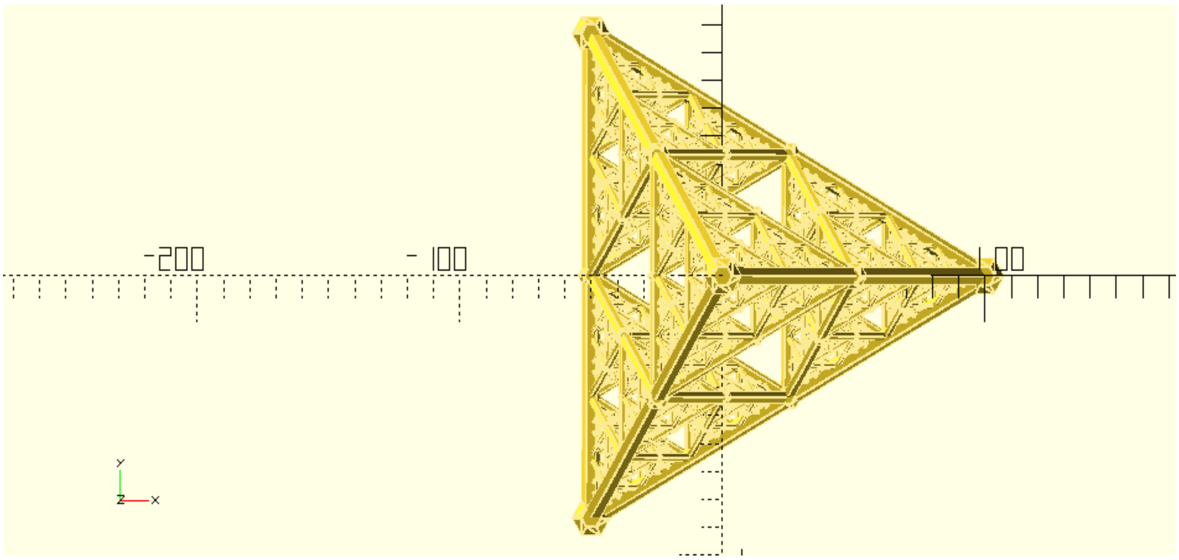
```
D = Da - center;

module tetrahedron(S, dim) {
    v1 = S * A;
    v2 = S * B;
    v3 = S * C;
    v4 = S * D;
    translate(v1) sphere(d=dim);
    translate(v2) sphere(d=dim);
    translate(v3) sphere(d=dim);
    translate(v4) sphere(d=dim);
    bar(v1, v2, 0.5 * dim);
    bar(v1, v3, 0.5 * dim);
    bar(v1, v4, 0.5 * dim);
    bar(v2, v3, 0.5 * dim);
    bar(v2, v4, 0.5 * dim);
    bar(v3, v4, 0.5 * dim);
}

module koch curve(S, dim, depth) {
    if (depth > 0) {
        tetrahedron(S, dim);
        translate((S / 2) * A) {
            koch(S / 2, 0.7 * dim, depth - 1);
        }
        translate((S / 2) * B) {
            koch(S / 2, 0.7 * dim, depth - 1);
        }
        translate((S / 2) * C) {
            koch(S / 2, 0.7 * dim, depth - 1);
```

```
        }
        translate((S / 2) * D) {
            koch(S / 2, 0.7 * dim, depth - 1);
        }
    }
}

koch(100, 15, 5);
```



Lampiran coding program menggunakan "Matlab"

```
%[membuat daerah gambar x,y]
vx = 2000;
vy = 2000;
x = linspace(-2,2,vx);
y = linspace(-2,2,vy);
c = zeros(length(y),length(x));
xn= 0;
yn= 0;
xnew = 0;
ynew = 0;
lenx = length(x);
leny = length(y);
```

```
zval = zeros(lenx,leny);
%[perintah iterasi]
if 1

%[memasukkan nilai c=a+bi dan iterasi]
a =2;
b =-0.5;
iter = 50;
for n=1:leny
c(n,:)=y(n)+1i*x(:);
end
for n = 1:lenx*leny
k = 1;
xn = real(c(n));
yn = imag(c(n));
while ((k < iter) &&((xn*xn + yn*yn)<4))
xnew = xn*xn-yn*yn + a;
ynew = 2*xn*yn + b;
xn = xnew;
yn = ynew;
k = k+1;
end
zval(n)= k;
end

cmap =  flipud(colormap(gray));
%cmap = colormap(gray);
colormap(cmap);

%mengatur ukuran gambar
plot(x,y,'-r');
imagesc(zval);
imresize(zval,[7000 7000]);
hold off;
axis equal;
axis off;
end
```

5.3 Luaran Yang Di Capai

Akhir dari penelitian diharapkan dapat dipergunakan oleh para desain grafis lokal untuk menambah kreasi desain yang baru dan luaran yang dicapai adalah HKI berupa program komputer.

BAB 6
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dalam tulisan ini algoritma yang dikembangkan untuk membangkitkan segitiga Sierpinski melalui pembangkitan segitiga berwarna lebih sederhana dibandingkan melalui pembangkitan segitiga kosong. Namun demikian, perlu dikaji aspek komputasi kedua algoritmanya.

Dalam tulisan ini juga dikembangkan beberapa bentuk dasar segitiga dan objek geometris. Pada iterasi yang cukup besar kesemua bentuk dasar dan benda geometris tersebut memberikan bentuk geometris yang mirip dengan segitiga Sierpinski.

6.2 Saran

Dalam tulisan ini perlu lagi kajian yang mendalam dari aspek algoritmanya sehingga bias mendapatkan bentukdesain segitiga yang indah

DAFTAR PUSTAKA

Barnxley, M. And Demko, S. (1985). Iterated Function System and the Global Construction of Fractals. *The Proceeding of the Royal Society of London A399, 243-275.*

Barnsley, M. (1986). *Fractal Function and Interpolation*. Constructive Approximation 2, 303-329.

Barnsley, M. (1988). *Fractal Everywhere*. Boston: Academic Press Inc

Falkoner, K. (1990). *Fractal Geometry: Mathematical Foundation and Applications*. New York: John Wiley & Sons

Mandelbrot, B. (1982). *The Fractal Geometry of Nature*. San Fransisco: W.H. Freeman and Co

Susanta, B., Soemantri,R., Widodo, Aryati, L., Hendarto, J and Suprapt. (1993). *Perkenalan dengan Geometri Fraktal, Laporan Penelitian, Didanai World Bank XXI, LOAN No:3311-IND, FMIPA UGM.*