MODIFIKASI ATURAN *CHAOS GAME* PADA SEGILIMA

SKRIPSI

Oleh
Melati Hanum Ayuningtyas
NIM 151810101004

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2019

MODIFIKASI ATURAN *CHAOS GAME* PADA SEGILIMA

SKRIPSI

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

Oleh
Melati Hanum Ayuningtyas
NIM 151810101004

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2019

PERSEMBAHAN

Dengan segala kerendahan hati dan puji syukur yang tak terhingga pada Allah SWT, skripsi ini saya persembahkan untuk:

1. Ibunda Winarsih dan Ayahanda Suhartanto tercinta, yang telah membesarkan, mendidik, mendoakan, memotivasi dengan penuh kasih sayang dan pengorbanan selama ini;
2. Adik Tiara Yuniar Maharani tersayang, yang telah memberikan semangat, doa, dan dukungan kepada penulis;
3. Guru-guru TK TUNAS HARAPAN, SDN REJOTANGAN 1, SDN KALIPANG 3, SMPN 1 Sutojayan, SMAN 1 Sutojayan, dan dosen-dosen FMIPA Matematika Universitas Jember yang telah memberikan ilmu dan membimbing penuh dengan kesabaran;
4. Almamater tercinta Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTTO

Di sana, pertolongan itu hanya dari Allah Yang Mahabenar. Dialah (pemberi) pahala terbaik dan (pemberi) balasan terbaik.

(Q.S Al-Kahf: 44)[1]

[1]Departemen Agama Republik Indonesia. 2005. *Al-Quran dan Terjemahanya.* Surabaya. CV. Karya Utama

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Melati Hanum Ayuningtyas

NIM : 151810101004

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul:
”Modifikasi Aturan *Chaos Game* pada Segilima” 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, Januari 2019

Yang menyatakan,

Melati Hanum Ayuningtyas

NIM 151810101004

SKRIPSI

MODIFIKASI ATURAN *CHAOS GAME* PADA SEGILIMA

Oleh

Melati Hanum Ayuningtyas
NIM 151810101004

Pembimbing:

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

Dosen Pembimbing Anggota : Dr. Firdaus Ubaidillah, S.Si., M.Si.

PENGESAHAN

Skripsi berjudul "Modifikasi Aturan *Chaos Game* pada Segilima" 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. NIP. 196908281998021001	Dr. Firdaus Ubaidillah, S.Si., M.Si. NIP. 197006061998031003
Anggota II,	Anggota III,
Dr. Kristiana Wijaya, S.Si., M.Si. NIP. 197408132000032004	Bagus Juliyanto, S.Si., M.Si. NIP. 198007022003121001

Mengesahkan
Dekan,

Drs. Sujito, Ph.D.
NIP. 196102041987111001

RINGKASAN

Modifikasi Aturan *Chaos Game* pada Segilima; Melati Hanum Ayuningtyas; 151810101004; 2019; 99 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Chaos game adalah teknik membangkitkan objek fraktal dengan menganut teori *chaos*. Teori *chaos* adalah teori yang menggambarkan pergerakan rumit dan tidak dapat ditebak atau dinamika sebuah sistem yang mudah berubah dari kondisi inisialnnya. Penelitian sebelumnya telah mengembangkan aturan *chaos game* pada pada bangun segitiga, segiempat dan segienam. Pada penelitian ini dikaji pengembangan aturan *chaos game* yang diterapkan pada bangun segilima, dimana terdapat aturan dan modifikasi aturan yang digunakan yaitu, aturan pemilihan titik sudut secara *random*, modifikasi aturan pemilihan titik sudut secara *random*, aturan pemilihan titik sudut secara *non-random*, serta aturan pemilihan titik sudut secara gabungan *random* dan *non-random*.

Simulasi program pada aturan pemilihan titik sudut secara *random* dilakukan satu kali percobaan. Hasil dari percobaan ini adalah objek visualisasi yang dihasilkan berbentuk menyerupai bangun segilima yang di dalamnya terdapat segilima dengan ukuran lebih kecil serta menyebar pada masing-masing titik sudut sebagai titik acuan. Objek visualisasi percobaan ini memiliki daerah yang berbeda warna ke setiap titik sudut yang digunakan sebagai titik acuan. Dengan demikian, objek visualisasi dari percobaan ini merupakan objek fraktal karena membentuk objek tertentu serta memiliki sifat *self-similarity*.

Simulasi program pada modifikasi aturan pemilihan titik sudut secara *random* dilakukan sebanyak dua percobaan. Untuk percobaan pertama yaitu modifikasi aturan

pemilihan titik sudut secara *random* dimana titik sudut yang dipilih saat ini tidak boleh sama dengan titik sudut yang telah dipilih sebelumnya, untuk percobaan kedua yaitu modifikasi aturan pemilihan titik sudut secara *random* dimana titik sudut yang dipilih saat ini dan sebelumnya tidak boleh dipilih lagi untuk iterasi berikutnya. Pada percobaan pertama objek visualisasi yang dihasilkan memilik bentuk seperti huruf “U” besar yang dikelilingi dengan lima huruf “U” yang lebih kecil lagi, demikian seterusnya. Pada percobaan kedua objek visualisasi yang dihasilkan tidak sama pada setiap pemilihan titik sudut sebagai titik acuan. Dengan demikian, objek visualisasi yang merupakan objek fraktal adalah objek visualisasi pada percobaan pertama sedangkan objek visualisasi pada percobaan kedua bukan merupakan objek fraktal.

Simulasi program pada aturan pemilihan titik sudut secara *non-random* dilakukan sebanyak sepuluh percobaan. Hasil yang didapatkan dari sepuluh percobaan adalah sama, titik-titik hasil bentukan aturan ini tidak menyebar ke semua titik sudut sebagai titik acuan atau hanya berkelompok pada koordinat tertentu yang dekat dengan titik sudut sebagai titik acuan. Dengan demikian, objek hasil modifikasi aturan pemilihan titik sudut secara *non-random* bukan merupakan objek fractal. Meskipun bukan merupakan suatu objek fraktal, namun objek yang dihasilkan aturan *non-random chaos game* memiliki karakteristik, yaitu kumpulan titik hasil bentukan *non-random chaos game* konvergen ke koordinat tertentu.

Simulasi program pada aturan pemilihan titik sudut secara gabungan *random* dan *non-random* dilakukan sebanyak empat percobaan. Hasil visualisasi yang didapatkan dari keempat percobaan adalah sama, objek dari keempat percobaan tidak membentuk suatu bangun tertentu dan semakin kacau (*chaos*) apabila titik yang diberlakukan tetap semakin banyak. Semakin banyak titik-titik hasil bentukan *chaos game* dengan aturan *random* maka objek hasil bentukan titik-titik tersebut semakin mengarah pada fraktal sedangkan semakin banyak titik-titik hasil bentukan *chaos game* dengan aturan *non-random* maka objek hasil bentukan titik-titik tersebut semakin tidak mengarah pada fraktal.

PRAKATA

Puji syukur penulis kepada Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Modifikasi Aturan *Chaos Game* pada Segilima". 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 petunjuk 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 dan Dr. Firdaus Ubaidillah, 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. Kristiana Wijaya, S.Si., M.Si., dan Bagus Juliyanto, S.Si., M.Si., selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun dalam menyelesaikan skripsi;
3. Drs. Sujito, PhD selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
4. Seluruh dosen dan staf karyawan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
5. Ibu Winarsih, Bapak Suhartanto, dan Adik Tiara Yuniar Maharani tercinta yang telah memberikan semangat, doa, dan dukungan kepada penulis;
6. Teman-teman satu bidang skripsi fraktal Nadiya, Mitha, Intan, Rozida, Ingka, Dwi Alfi, Novita, dan Iza yang selalu memberi dukungan dan semangat tanpa henti;
7. Teman-teman seangkatan SIGMA'15 yang telah banyak membantu penulis selama studi;

8. Teman-teman Kos Putri Mastrip 11 dan teman-teman KKN 38 Panti yang telah memberikan semangat dan motivasi;
9. Pengurus HIMATIKA "Geokompstat" masa bakti 2017 yang telah memberikan semangat;
10. Teman-teman paguyuban KEMAPATA (Keluarga Mahasiswa Blitar di Jember) yang telah memberikan semangat dan motivasi;
11. Sahabat-sahabat tersayang Tutut Aprilia Cahyani, Rinaldi Gardhimas, M.F. Akbar, Mery Tatisna, Ulfa Umrhotul, Bridesmaids, SAMRIS, dan IS yang senantiasa memberikan dukungan dan menjadi keluarga kedua penulis;
12. Kakak-kakak tingkat dan adik-adik tingkat yang selalu memotivasi penulis agar terselesaikannya skripsi ini;
13. Serta semua pihak yang tidak dapat penulis sebutkan satu persatu.

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

Jember, Januari 2019

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR LAMPIRAN

Halaman

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Fraktal merupakan bentuk grafik yang mengandung perulangan dirinya sendiri. Jika objek fraktal diperbesar, maka akan memperlihatkan bentuk yang sama dan sebangun dengan ukuran lebih kecil. Terdapat beberapa karakter penting pada fraktal yaitu *self similarity*, *self affine*, *self inverse*, dan *self squaring*. Pada fraktal terdapat istilah dimensi fraktal atau *fractal dimensions* yaitu bilangan-bilangan pecahan atau bilangan tak bulat (*non-integer)* yang mencirikan skala fraktal. Secara umum, di dalam suatu himpunan fraktal terdapat suatu himpunan bagian yang merupakan skala kecil dari keseluruhannya (Armana, 2016).

Ada banyak sekali objek fraktal yang sering dijumpai di alam atau dalam kehidupan manusia, seperti misalnya salju, daun cemara, petir, pola retakan tanah, serta motif batik. Objek-objek fraktal tersebut dapat dibangkitkan oleh beberapa teknik atau metode. Salah satu teknik untuk membangkitkan objek fraktal adalah *chaos game*. *Chaos game* adalah teknik membangkitkan objek fraktal dengan menganut teori *chaos*. Teori *chaos* adalah teori yang menggambarkan pergerakan rumit dan tidak dapat ditebak atau dinamika sebuah sistem yang mudah berubah dari kondisi inisialnya. Sistem *chaos* dapat dijelaskan secara matematika karena mengikuti hukum tertentu tetapi karena sifat berubah-ubahnya, akan tampak acak bagi mata awam (Wahjono dkk., 2007).

Contoh lain dari objek fraktal adalah segitiga Sierpinski. Menurut Purnomo (2014), segitiga Sierpinski adalah fraktal linier yang mempunyai keserupaan diri identik sampai iterasi tak hingga. Segitiga Sierpinski merupakan segitiga yang terdiri dari segitiga-segitiga lain yang berbentuk sama dan berulang dengan skala tertentu. Apabila bagian dari segitiga tersebut diperbesar, maka akan tampak keseluruhan dari segitiga Sierpinski.

Dalam kaitannya dengan *chaos game* telah dilakukan beberapa penelitian antara lain Jefrey (1990) melakukan *chaos game* pada empat titik sudut atau pada persegi dan menghasilkan bentuk yang berbeda, hasilnya bukanlah persegi di dalam persegi, tetapi sebuah persegi yang terisi dengan banyak titik dan terbentuk secara *random*. Kemudian, Devaney (2003) melakukan modifikasi aturan *chaos game* pada segienam (*hexagon*) yaitu dengan merubah jarak titik awal dengan titik sudut menjadi sepertiga. Armana (2016) menggunakan tiga titik sudut pembentuk segitiga untuk membangun segitiga Sierpinski dengan aturan pemilihan titik acuan secara acak (*random*) dan titik awal yang digunakan terletak pada, di dalam serta di luar segitiga. Yunaning (2018) menggunakan tiga titik sudut pembentuk segitiga dengan titik acuan yang digunakan dalam aturan pemilihan dipilih secara urut (*non-random*) tetapi tidak membentuk suatu objek fraktal. Kemudian, Ratna (2018) sama-sama menggunakan tiga titik sudut pembentuk segitiga dengan modifikasi berupa penambahan titik berat untuk membangun segitiga Sierpinski dengan pemilihan titik acuan secara acak (*random*) dan urut (*non-random*).

Berdasarkan uraian di atas, peneliti tertarik untuk melakukan penelitian lebih lanjut mengenai aturan dan modifikasi aturan *chaos game* bukan hanya menggunakan segitiga dan segiempat melainkan pada objek geometri lainnya, yaitu segilima. Diharapkan dengan menggunakan objek segilima dapat diketahui objek geometri selain segitiga yang dapat membentuk objek fraktal apabila dibangkitkan dengan aturan dan modifikasi aturan *chaos game*. Dalam penelitian ini akan dibahas aturan dan modifikasi aturan pemilihan titik sudut sebagai titik acuan pada *chaos game* baik secara *random* maupun *non-random* pada segilima serta visualisasi dengan menggunakan program MATLAB.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, maka rumusan masalah yang dibahas pada penelitian ini adalah apakah aturan *chaos game* secara *random*, modifikasi aturan *chaos game* secara *random*, aturan *chaos game* secara *non-random* serta aturan gabungan *chaos game* secara *random* dan *non-random* pada segilima akan membentuk fraktal?

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian modifikasi aturan *chaos game* adalah segilima konveks.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan dari penelitian ini adalah

a. Mengetahui apakah aturan *chaos game* secara *random*, modifikasi aturan *chaos game* secara *random*, aturan *chaos game* secara *non-random* serta aturan gabungan *chaos game* secara *random* dan *non-random* pada segilima akan membentuk fraktal.

b. Membuat program untuk memvisualisasikan aturan *chaos game* secara *random*, modifikasi aturan *chaos game* secara *random*, aturan *chaos game* secara *non-random* serta aturan gabungan *chaos game* secara *random* dan *non-random* pada segilima.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah memberikan informasi tentang *chaos game* yang dikembangkan dengan memodifikasi aturannya untuk membangun segilima sehingga dapat bermanfaat bagi penelitian-penelitian yang berkaitan selanjutnya.

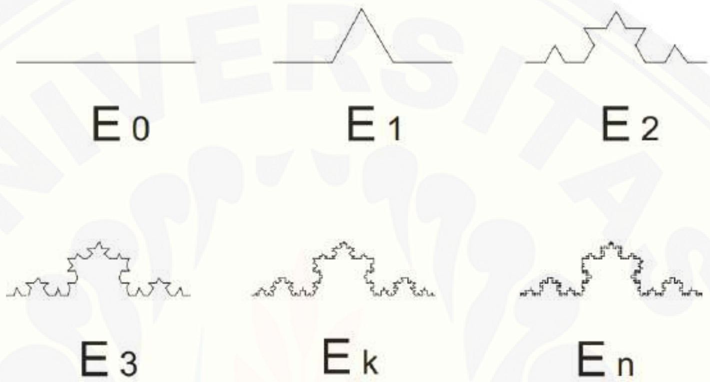
BAB 2. TINJAUAN PUSTAKA

2.1 Fraktal

Konsep mengenai fraktal pertama kali dikenalkan oleh Mandelbrot. Mandelbrot (1983) mengatakan bahwa fraktal berasal dari bahasa latin yaitu *fractus* yang artinya tidak teratur atau terfragmentasi dan *frangere* yang artinya memecah atau membuat fragmen-fragmen yang tidak beraturan. Maksud dari kata fragmen pada fraktal yaitu bentuk dari geometri fraktal yang dapat dibagi menjadi beberapa bagian, bagian-bagian tersebut merupakan tiruan dalam bagian yang ukurannya lebih lebih kecil dari bentuk aslinya atau sama besar dengan bentuk aslinya. Menurut Sulistiyantoko (2008), fraktal adalah sebuah bentuk grafik yang mengandung perulangan atas dirinya sendiri yang bisa dibangkitkan dengan fungsi matematika.

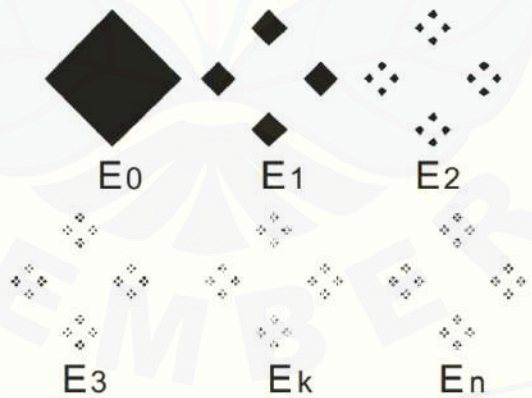
Dalam bentuk fraktal yang abstrak terdapat suatu sifat, dimana sifat ini menunjukkan jika fraktal sebenarnya tersusun dari bagian-bagian dengan bentuk yang sama. Sifat tersebut adalah *self-similarity* atau sifat keserupaan diri. Wahjono dkk. (2007) mengklasifikasikan sifat *self-similarity* fraktal menjadi tiga jenis, yaitu *exact self-similarity*, *quasi self-similarity*, dan *statistical self-similarity*. Sifat fraktal *exact self-similarity* adalah sifat fraktal dimana bentuk fraktal yang tampak persis satu sama lain tetapi dengan ukuran dan rasio berbeda. Sifat fraktal *quasi self-similarity* adalah sifat fraktal dimana bentuk fraktal yang tampak mirip satu sama lain tetapi tidak persis serta dengan ukuran dan rasio yang berbeda. Sifat fraktal *statistical self-similarity* adalah sifat fraktal dimana bentuk fraktal tidak tervisualisasi dengan jelas (bentuk fraktal yang paling dasar) tetapi memiliki ukuran numerik atau statistik yang dipertahankan pada ukuran dan rasio yang berbeda (Armana, 2016).

Jenis-jenis fraktal yang sering dijumpai ada dua, yaitu fraktal alami atau *natural fractal* dan himpunan-himpunan fraktal atau *fractal sets*. Contoh dari fraktal alami antara lain awan-awan, cabang pohon, garis pantai, salju, model fluktuasi mata uang dalam pasar finansial, dan lainnya (Mandelbrot, 1983). Sedangkan contoh dari himpunan fraktal antara lain *Kurva Von Koch* (Gambar 2.1), Debu *Cantor* (Gambar 2.2), Segitiga Sierpinski (Gambar 2.3), dan *Mandelbrot set* (Gambar 2.4).



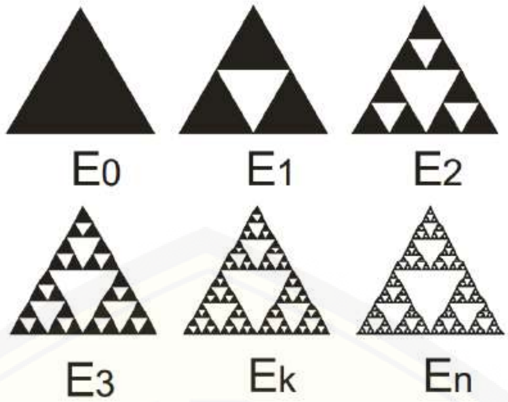
Gambar 2.1 Kurva Von Koch

(Sumber: Sulistiyantoko, 2008)



Gambar 2.2 Debu *Cantor*

(Sumber: Sulistiyantoko, 2008)



Gambar 2.3 Segitiga Sierpinski

(Sumber: Sulistiyantoko, 2008)

Gambar 2.4 *Mandelbrot set*

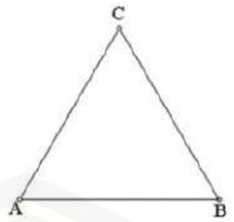
(Sumber: Peitgen dkk., 2004)

2.2 *Chaos Game*

Teori *chaos* dalam fisika dan matematika menjelaskan tentang perilaku dari sistem dinamis nonlinear tertentu yang pergerakannya sangat bergantung kepada kondisi awal. Sebagai hasil dari ketergantungan pada kondisi awal ini adalah bahwa kondisi awal menyebabkan gangguan yang pada akhirnya akan terlihat sebagai suatu yang acak. Sehingga dapat disimpulkan bahwa pergerakan yang akan datang sangat dipengaruhi atau bergantung sepenuhnya kepada kondisi awal (Sulistiyantoko, 2008).

Dalam kaitannya dengan fraktal, menurut Purnomo dkk. (2016) *chaos game* merupakan bentuk permainan menggambar suatu titik dalam segitiga sama sisi dengan aturan tertentu yang dilakukan secara berulang-ulang dan iteratif. Titik yang digambar tersebut adalah titik tengah dari jarak titik awal dengan salah satu titik sudut segitiga yang diambil secara acak. Jika penggambaran titik tengah dilakukan pada jumlah iterasi yang kecil, maka kumpulan titik-titik tersebut terkesan *chaos* (kacau). Namun, jika dilakukan pada jumlah ribuan iterasi maka kumpulan titik-titik tengah tersebut akan mendekati bentuk segitiga Sierpinski.

Salah satu contoh membangkitkan objek fraktal dengan menggunakan *chaos game* adalah pada segitiga Sierpinski. Langkah awal yang dilakukan adalah dengan menggambar tiga buah titik sebagai titik pembentuk segitiga. Selanjutnya menentukan titik awal dan titik sudut secara acak. Kemudian ambil titik baru yang memiliki jarak setengah dari titik awal ke titik sudut yang telah dipilih secara acak tadi. Titik baru hasil dari bentukan *chaos game* inilah yang kemudian menjadi titik awalnya. Titik ini disebut hasil bentukan *chaos game* iterasi pertama, sehingga akan didapatkan titik-titik lain hasil bentukan *chaos game* sesuai dengan iterasi yang ditentukan. Iterasi yang sedikit tidak dapat membangkitkan segitiga Sierpinski karena hanya akan terdapat kumpulan titik-titik yang tidak beraturan, sehingga dibutuhkan iterasi sebanyak mungkin untuk mendapatkan bentuk segitiga Sierpinski (Zohuri, 2015). Tahapan *chaos game* secara berurutan akan dipaparkan pada Gambar 2.5 sampai dengan Gambar 2.9.



Gambar 2.5 Tahap pertama untuk membangkitkan segitiga Sierspinsi

(Sumber:Zohuri,2015)

Gambar 2.6 Tahap untuk membangkitkan segitiga Sierspinsi iterasi ke-30

(Sumber: Zohuri, 2015)

Gambar 2.7 Tahap untuk membangkitkan segitiga Sierspinsi iterasi ke-100

(Sumber: Zohuri, 2015)

Gambar 2.8 Tahap untuk membangkitkan segitiga Sierspinsi iterasi ke-400

(Sumber: Zohuri, 2015)

Gambar 2.9 Tahap membangkitkan segitiga Sierspinsi pada iterasi ke-30.000

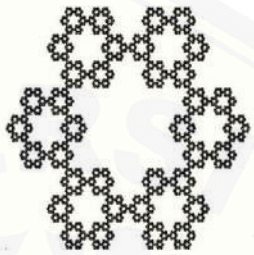
(Sumber: Zohuri, 2015)

2.3 Penelitian Terdahulu

Penelitian terdahulu dirasa sangat penting dalam sebuah penelitian yang akan dilakukan, sebab penelitian terdahulu dapat dijadikan sebagai acuan oleh penulis. Penelitian terdahulu juga dapat memudahkan penulis untuk menentukan langkah-langkah untuk menyusun penelitian, baik dari segi konsep maupun dari segi teori, sehingga teori yang digunakan lebih luas dalam mengkaji suatu penelitian. Penelitian mengenai aturan *chaos game* sudah banyak dilakukan antara lain oleh Jeffrey (1990), Devaney (2003), Armana (2016), Yunaning (2018) dan Ratna (2018).

Jeffrey (1990) telah melakukan penelitian yang dituangkan dalam makalah dengan judul "*Chaos Game Representation of Gene Structure*". Makalah ini menyajikan metode baru untuk mewakili urutan DNA, sehingga memungkinkan representasi dan penyelidikan pola dalam urutan, atau secara visual dapat mengungkapkan struktur yang tidak diketahui sebelumnya. Metode ini disebut dengan *Chaos Game Representation* (CGR). DNA sendiri memiliki urutan yang terdiri dari empat nukleotida, yaitu *adenine* (A), *guanine* (G), *cytosine* (C), dan *thymine* (T), sehingga dalam metode CGR empat nukleotida ini digunakan sebagai label pada setiap titik sudut. Karena pada metode CGR digunakan empat titik sudut maka bidang yang digunakan pada metode ini adalah segi empat. Misal akan menggambarkan enam basa pertama dari kromosom 11 (daerah beta globin manusia), dengan urutan DNA GAATTC. Langkah-langkah metode CGR yaitu menentukan titik awal, yang mana titik awal adalah titik pusat. Kemudian, iterasi pertama dari *chaos game* ini yaitu mengambil jarak setengah dari titik awal ke titik sudut G. Untuk iterasi kedua titik baru bentukan iterasi pertama tersebut mengambil jarak setengah ke titik sudut A. Sedangkan untuk iterasi ketiga, mengambil jarak setengah dari titik baru bentukan iterasi kedua ke titik sudut A. Iterasi keempat mengambil jarak setengah dari titik baru bentukan iterasi ketiga ke titik sudut T. Iterasi kelima mengambil jarak setengah dari titik baru bentukan iterasi keempat ke titik sudut T. Serta iterasi keenam mengambil jarak setengah dari titik baru bentukan iterasi kelima ke titik sudut C. Apabila hanya sedikit iterasi maka hanya terlihat beberapa titik yang menyebar pada bidang segi empat, namun setelah dilakukan banyak iterasi maka terlihat bahwa titik-titik tersebut bukanlah persegi di dalam persegi, tapi sebuah persegi yang terisi dengan banyak titik yang terbentuk secara *random*. Atau bisa dikatakan tidak membentuk fraktal.

Devaney (2003) telah membuat modifikasi terhadap aturan produksi *chaos game* sehingga menghasilkan *Sierpinski hexagon*. Langkah pertama yang dilakukan adalah menempatkan enam titik awal sehingga membentuk sebuah hexagon. Langkah selanjutnya sama seperti langkah-langkah *chaos game* untuk membangkitkan segitiga Sierpinski, tetapi perbedaannya terletak pada jarak antara titik awal dengan titik sudut. Jarak antara titik awal dan titik sudut untuk membangkitkan segitiga Sierpinski adalah setengah jarak aslinya, sedangkan untuk membangkitkan *Sierpinski hexagon* dibutuhkan sepertiga dari jarak aslinya. Atau, bisa dikatakan bahwa dilakukan *compress* rasio dari *game* ini menjadi tiga.



Gambar 2.10 *Sierpinski hexagon*

(Sumber: Devaney, 2003)

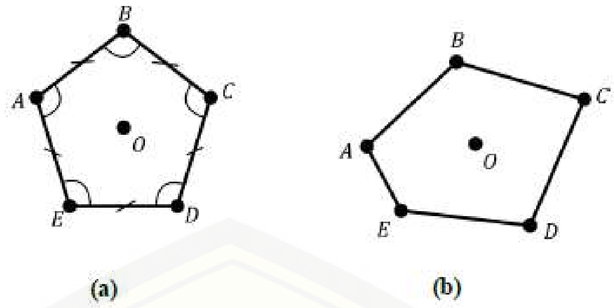
Selain Devaney, Armana (2016) juga melakukan penelitian tentang *chaos game* dengan memanfaatkan koordinat titik tengah segitiga Sierpinski untuk menganalisis *chaos game*. Dalam penelitian yang dilakukan ini letak titik awal baik di luar, di dalam maupun pada segitiga dengan pemilihan titik acuan secara acak (*random*) tetap menghasilkan segitiga Sierpinski, hal tersebut dikarenakan meskipun titik awal terletak sebarang namun pada saat iterasi ke-*n* titik bentukan *chaos game* akan mendekati letak segitiga Sierpinski. Menurut penelitian ini juga, peluang munculnya titik sudut berpengaruh terhadap distribusi pengelompokan titik yang dihasilkan, semakin banyak peluang titik sudut maka akan semakin banyak pula titik yang dihasilkan dari titik sudut tersebut.

Yunaning (2018) juga telah melakukan penelitian berkaitan dengan *chaos game* yaitu mengkaji pengembangan aturan *chaos game* yang disebut aturan *non-random* pada segitiga. Aturan *non-random* adalah aturan pemilihan titik sudut yang dipilih urut secara terus menerus hingga iterasi ke-*n*. Pada penelitian yang dilakukan ini objek segitiga tidak membentuk suatu objek fraktal, hal tersebut dikarenakan titik-titik bentukan *non-random chaos game* tidak dapat menyebar pada seluruh daerah titik sudut atau hanya berkelompok di koordinat tertentu pada masing-masing pemilihan titik sudut yang sama. Titik-titik tersebut berkelompok di suatu koordinat tertentu pada masing-masing titik sudut disebabkan karena pada aturan ini, pemilihan titik sudut dipilih secara urut secara terus menerus mengakibatkan jarak antara titik-titik bentukan *non-random chaos game* tidak berbeda signifikan dari iterasi satu ke iterasi selanjutnya. Meskipun objek bentukan aturan *non-random chaos game* tidak membentuk objek fraktal, tetapi objek tersebut memiliki karakteristik dimana kumpulan titik bentukan aturan *non-random chaos game* pada pemilihan titik sudut yang sama konvergen ke koordinat tertentu.

Ratna (2018) melakukan penelitian dengan mengembangkan aturan *chaos game* yaitu aturan pemilihan titik acuan secara *random* dan *non-random* yang diterapkan dengan memanfaatkan titik berat segitiga. Pada penelitian dengan aturan pemilihan titik acuan secara *random* objek yang dibentuk dapat dikatakan sebagai objek fraktal, sebab dapat membentuk suatu bentuk yang menyerupai segitiga Sierpinski dan titik bentukannya menyebar ke semua titik acuan sehingga terdapat tiga segitiga dengan tiga warna berbeda serta memiliki sifat fraktal yaitu *self-similarity*. Sedangkan untuk aturan pemilihan titik acuan secara *non-random* tidak menghasilkan objek fraktal, sebab titik-titik bentukan tidak dapat menyebar ke semua titik acuan dan hanya berkelompok di koordinat terdekat dengan titik acuan.

2.4 Segilima

Segilima merupakan salah satu contoh dari objek geometri. Segilima dapat diartikan sebagai suatu kurva tertutup yang memiliki lima sisi, setiap sisi tersebut berupa garis lurus yang disebut ruas garis. Selain itu, segilima juga memiliki lima sudut. Segilima atau *pentagon* disebut juga poligon bersisi lima. Segilima dapat dibagi menjadi dua, yaitu segilima beraturan dan segilima tidak beraturan (Morisson, 2015).



(a) Segilima beraturan ABCDE; (b) Segilima tidak beraturan ABCDE

Gambar 2.11 Macam-macam Segilima

(Sumber: Situmorang, 2016)

Poligon beraturan adalah suatu poligon yang ukuran sisi dan sudutnya sama besar, sedangkan poligon tidak beraturan adalah poligon yang ukuran sisi atau sudutnya tidak sama besar (Rich dan Thomas, 2009). Poligon beraturan dapat diilustrasikan pada Gambar 2.11 (a). Dari gambar tersebut diketahui ada lima titik sudut yaitu, titik A, B, C, D, dan E serta terdapat lima sudut yang masing-masing sudut nilainya sama yaitu m∠A= m∠B= m∠C= m∠D= m∠E. Ruas garis poligon beraturan memiliki ukuran panjang yang sama besar yaitu $|\overline{AB}| = |\overline{BC}| = |\overline{CD}| = |\overline{DE}| = |\overline{EA}|$. Selain itu, jarak setiap titik sudut poligon beraturan ke titik pusat O adalah sama besar, yaitu $|\overline{OA}| = |\overline{OB}| = |\overline{OC}| = |\overline{OD}| = |\overline{OE}|$. Sedangkan poligon tidak beraturan dapat diilustrasikan pada Gambar 2.11 (b). Dari gambar diketahui bahwa ada sudut yang memiliki ukuran tidak sama besar yaitu pada m∠A≠ m∠B, terdapat pula panjang ruas garis yang tidak sama besar yaitu $|\overline{AB}| \neq |\overline{BC}|$, serta terdapat jarak titik sudut ke titik pusat O yang memiliki nilai tidak sama besar $|\overline{OA}| \neq |\overline{OB}|$. Atau dapat disimpulkan bahwa jika nilai dari sudut, panjang ruas garis dan jarak titik sudut ke titik pusat O suatu poligon ada yang tidak sama maka poligon tersebut dapat disebut sebagai poligon tidak beraturan.

2.5 Barisan konvergen

Definisi 2.1 Barisan bilangan real adalah suatu fungsi bernilai real (atau suatu barisan di R) atau suatu fungsi pada himpunan N dengan daerah hasil yang termuat di R.

Suatu barisan di R memasangkan masing-masing bilangan asli n=1,2,3,… secara tunggal dengan bilangan real. Bilangan real yang diperoleh tersebut disebut elemen, nilai atau suku dari barisan tersebut. Hal yang biasa untuk menuliskan elemen dari R yang berpasangan dengan $n \in N$, dengan suatu simbol seperti x_n, a_n atau z_n. Jadi bila $X: N \rightarrow R$ suatu barisan, maka menuliskan nilai X di n dengan X_n. Contoh barisan adalah bila $a \in R$, maka barisan $A = (a^n: n \in N)$ adalah barisan $(a^1, a^2, a^2, \ldots, a^n, \ldots)$. Khususnya bila $a = \frac{1}{2}$, maka diperoleh barisan $\left(\frac{1}{2^n}: n \in N\right)$ (Bartle dan Sherbert, 2011).

Definisi 2.2 Misalkan $X = (x_n)$ barisan bilangan real. Suatu bilangan real x dikatakan limit dari (x_n) jika untuk setiap $\varepsilon > 0$ terdapat bilangan asli $K(\varepsilon)$, sedemikian hingga untuk setiap $n \geq K(\varepsilon)$, maka suku-suku x_n terletak di dalam persekitaran- ε, $V_\varepsilon(x)$.

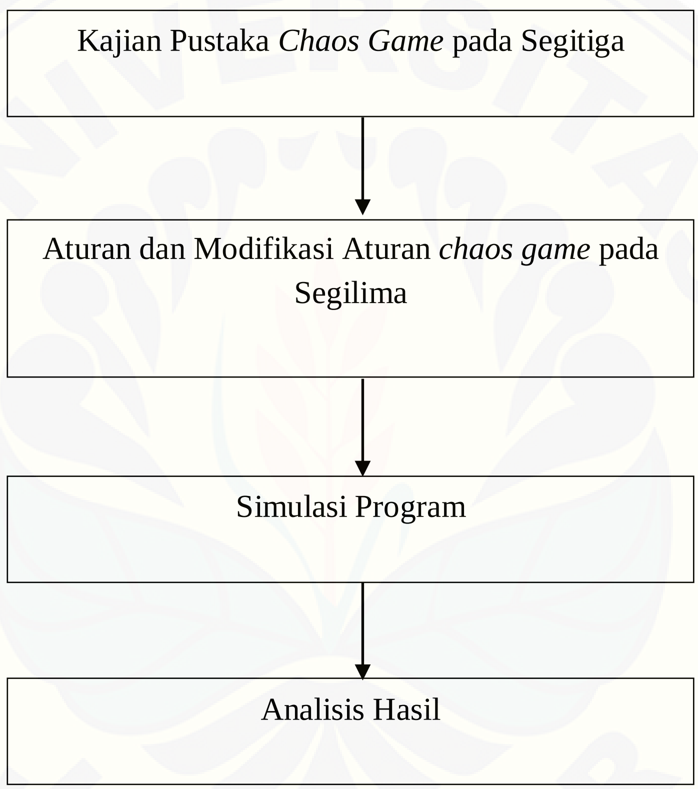
Bila x merupakan suatu limit dari barisan, dapat dikatakan bahwa $X = (x_n)$ konvergen ke x atau mempunyai limit x. Jika suatu barisan mempunyai limit maka barisan tersebut **konvergen**. Sebaliknya jika tidak mempunyai limit maka barisan tersebut **divergen**. Berikut beberapa teorema yang ada pada limit barisan.

Teorema 2.1 Misalkan $X = (x_n)$ barisan bilangan real yang konvergen maka $\lim_{n\to\infty}(k\, x_n) = k \lim_{n\to\infty}(x_n)$.

Teorema 2.2 Jika 0 < b < 1, maka $\lim_{n\to\infty} b^n = 0$

BAB 3. METODE PENELITIAN

Pada bab ini membahas tentang langkah-langkah yang akan digunakan dalam menyelesaikan penelitian ini. Secara skematik, langkah-langkah yang akan dilakukan dalam menyelesaikan penelitian ini dapat digambarkan dengan diagram alir pada Gambar 3.1.



Gambar 3.1 Skema Penelitian

Berdasarkan skema pada Gambar 3.1, langkah-langkah penelitian dapat dijelaskan sebagai berikut:

3.1 Kajian Pustaka *Chaos Game* pada Segitiga

Langkah awal yaitu dengan melakukan kajian pustaka yang memiliki tujuan untuk mendapatkan informasi lebih luas dari jurnal, buku, dan skripsi mengenai Pustaka *Chaos Game* pada penelitian sebelumnya

3.2 Aturan dan Modifikasi Aturan *Chaos Game* pada Segilima

Langkah awal yaitu dengan melakukan aturan dan modifikasi aturan *chaos game* pada segilima yang terdiri dari aturan pemilihan titik sudut secara *random*, modifikasi aturan pemilihan titik sudut secara *random*, aturan pemilihan titik sudut secara *non-random* serta aturan pemilihan titik sudut secara gabungan *random* dan *non-random*.

3.2.1 Aturan Pemilihan Titik Sudut secara *Random*.

Aturan *chaos game* secara random yaitu pemilihan titik sudut sebagai titik acuan yang dilakukan secara acak dari iterasi satu ke iterasi selanjutnya, dimana titik acuan yang dipilih boleh dilakukan perulangan secara berurutan. Misalkan iterasi pertama dipilih titik sudut A_0, kemudian untuk iterasi kedua boleh dipilih titik sudut A_0 lagi atau bisa juga menuju titik sudut lainnya. Berikut adalah algoritma aturan *chaos game* pada segilima yang pemilihan titik sudut sebagai titik acuannya dilakukan secara *random*:

1) Membuat sebuah segilima dan memberi label pada setiap titik sudut segilima, misalnya diberi label A_0, A_1, A_2, A_3, dan A_4.
2) Memilih titik awal secara acak dan beri label T_0.
3) Memilih sebuah titik sudut segilima untuk dihubungkan dengan titik awal yang telah dipilih secara acak.
4) Menentukan titik baru yang berjarak setengah jarak dari titik sudut ke titik awal, beri label T_1 pada titik baru tersebut.

5) Mengulang langkah (3) dengan titik awal adalah titik baru yang didapat dari iterasi sebelumnya dan boleh dilakukan perulangan pada pemilihan titik sudut sebagai titik acuan.
6) Melakukan hingga iterasi yang diinginkan.

3.2.2 Modifikasi Aturan Pemilihan Titik Sudut secara *Random*.

Modifikasi aturan *chaos game* secara *random* yaitu pemilihan titik sudut sebagai titik acuan yang dilakukan secara acak dari iterasi satu ke iterasi selanjutnya, namun titik sudut yang dipilih saat ini tidak boleh sama dengan titik sudut yang telah dipilih sebelumnya atau bisa dikatakan tidak diperbolehkan terdapat perulangan pemilihan titik sudut sebagai titik acuan secara berurutan. Misalkan iterasi pertama dipilih titik sudut A_0, iterasi kedua dipilih titik sudut A_1, maka untuk iterasi selanjutnya tidak boleh dipilih titik A_1 lagi melainkan titik sudut antara A_0, A_2, A_3, dan A_4 lakukan hingga iterasi yang diinginkan. Berikut adalah algoritma hasil modifikasi aturan *chaos game* pada segilima yang pemilihan titik sudut sebagai titik acuannya dilakukan secara *random*:

1) Membuat sebuah segilima dan memberi label pada setiap titik sudut segilima, misalnya diberi label A_0, A_1, A_2, A_3, dan A_4
2) Memilih titik awal secara acak dan beri label T_0.
3) Memilih sebuah titik sudut segilima untuk dihubungkan dengan titik awal yang telah dipilih secara acak.
4) Menentukan titik baru yang berjarak setengah jarak dari titik sudut ke titik awal, beri label T_1 pada titik baru tersebut.
5) Mengulang langkah (3) dengan titik awal adalah titik baru yang didapat dari iterasi sebelumnya namun titik sudut yang digunakan sebagai acuan saat ini tidak boleh sama dengan titik sudut yang telah digunakan sebelumnya.
6) Melakukan hingga iterasi yang diinginkan.

Untuk algoritma modifikasi aturan pemilihan titik sudut secara *random* akan diberlakukan tambahan aturan yaitu pada titik sudut yang dipilih saat ini dan sebelumnya tidak boleh dipilih lagi untuk iterasi berikutnya. Misalkan iterasi pertama dipilih titik sudut A_0, iterasi kedua dipilih titik sudut A_1, maka untuk iterasi ketiga tidak boleh dipilih titik sudut A_0 dan A_1 melainkan titik sudut antara A_2, A_3, dan A_4. Langkah-langkah yang digunakan sama seperti pada langkah nomor 1 sampai dengan 6, namun pada langkah ke-5 titik sudut yang dipilih saat ini dan sebelumnya tidak boleh dipilih lagi untuk iterasi berikutnya.

3.2.3 Aturan Pemilihan Titik Sudut secara *Non-Random*.

Aturan *chaos game* secara *non-random* yaitu pemilihan titik sudut sebagai titik acuan yang dilakukan secara urut dari iterasi satu ke iterasi selanjutnya. Misalkan urutan pemilihan titik sudut sebagai titik acuan adalah A_0, A_1, A_2, A_3, dan A_4, iterasi pertama dipilih titik sudut A_0, iterasi kedua dipilih titik sudut A_1, iterasi ketiga dipilih titik sudut A_2, iterasi keempat dipilih titik sudut A_3, iterasi kelima dipilih titik sudut A_4, kemudian berulang kembali lagi pada titik A_0 dan seterusnya hingga iterasi yang diinginkan. Berikut adalah algoritma hasil aturan *chaos game* pada segilima yang pemilihan titik sudut sebagai titik acuannya dilakukan secara *non-random*:

1) Membuat sebuah segilima dan memberi label pada setiap titik sudut segilima, misalnya diberi label A_0, A_1, A_2, A_3, dan A_4 secara berurutan.
2) Memilih titik awal secara acak dan beri label T_0.
3) Memilih sebuah titik sudut segilima untuk dihubungkan dengan titik awal yang telah dipilih secara acak.
4) Menentukan titik baru yang berjarak setengah jarak dari titik sudut ke titik awal, beri label T_1 pada titik baru tersebut.
5) Mengulang langkah (3) dengan titik awal adalah titik baru yang didapat dari iterasi sebelumnya namun titik sudut yang digunakan pada iterasi berikutnya harus sesuai dengan urutan yang telah ditentukan. Misalkan titik sudut awal yang digunakan adalah titik sudut A_0, kemudian titik sudut selanjutnya yang dipilih adalah titik sudut A_1, titik sudut A_2, titik sudut A_3, dan terakhir titik sudut A_4, lalu

berulang kembali pada titik sudut A_0 serta berjalan sesuai dengan urutan yang telah ditentukan.

6) Melakukan hingga iterasi yang diinginkan.

3.2.4 Aturan Pemilihan Titik Sudut secara Gabungan *Random* dan *Non-Random*.

Akan dilakukan gabungan aturan *chaos game* pada segilima, yaitu aturan *random* dan *non-random*. Misalkan untuk iterasi pertama dipilih titik sudut A_0, kemudian untuk iterasi kedua hingga iterasi kelima boleh dipilih semua titik sudut, yaitu titik sudut A_0, A_1, A_2, A_3, sampai dengan A_4 secara acak. Namun, pada saat iterasi keenam titik sudut yang dipilih harus titik sudut A_0 seperti pada iterasi pertama atau bisa diartikan saat iterasi ke $5n+1$ titik sudut yang dipilih sebagai acuan adalah sama, dengan n merupakan bilangan asli. Berikut adalah algoritma hasil gabungan aturan *chaos game* pada segilima secara *random* dan *non-random*:

1) Membuat sebuah segilima dan memberi label pada setiap titik sudut segilima, misalnya diberi label A_0, A_1, A_2, A_3, dan A_4.
2) Memilih titik awal secara acak dan beri label T_0.
3) Memilih sebuah titik sudut segilima untuk dihubungkan dengan titik awal yang telah dipilih secara acak.
4) Menentukan titik baru yang berjarak setengah jarak dari titik sudut ke titik awal, beri label T_1 pada titik baru tersebut.
5) Mengulang langkah (3) dengan titik awal adalah titik baru yang didapat dari iterasi sebelumnya namun titik sudut yang digunakan sebagai acuan pada iterasi ke $5n+1$ adalah tetap. Dengan n adalah bilangan asli.
6) Melakukan hingga iterasi yang diinginkan.

Untuk algoritma hasil gabungan aturan *chaos game* pada segilima secara *random* dan *non-random* akan diberlakukan pula pemilihan titik sudut sebagai titik acuan saat iterasi ke $5n+2$, $5n+3$, dan $5n+4$ adalah sama, dengan n merupakan bilangan asli. Langkah-langkah yang digunakan sama seperti pada langkah nomor 1 sampai dengan 6, namun pada langkah ke-5 titik sudut sebagai titik acuan saat iterasi ke 5*n*+2, 5*n*+3, dan 5*n*+4 adalah sama. Penulis tidak melakukan pemilihan titik sudut sebagai titik acuan pada saat iterasi ke 5*n*+5, sebab saat menggunakan 5*n*+5 maka aturan *chaos game* menjadi aturan *non-random* saja tanpa melibatkan aturan *random*.

3.3 Simulasi Program

Software yang digunakan untuk membuat program dalam penelitian ini adalah software MATLAB R2015b. Prosedur untuk membuat program ini adalah sebagai berikut:

1. Input

Pada tahap ini diinputkan jumlah iterasi yang akan digunakan pada *chaos game*, lima titik acuan yang membentuk bidang segilima, serta menentukan titik awal untuk memulai *chaos game*.

2. Proses

Pada tahap ini dilakukan pemrosesan program untuk aturan *chaos game* secara *random*, modifikasi aturan *chaos game* secara *random*, aturan *chaos game* secara *non-random*, serta aturan gabungan *chaos game* secara *random* dan *non-random* pada segilima yang telah dijelaskan pada subsubbab 3.2.1 sampai dengan 3.2.4 di atas.

3. Output

Output yang dihasilkan dari simulasi ini berupa visualisasi penerapan aturan *chaos game* secara *random*, modifikasi aturan *chaos game* secara *random*, aturan *chaos game* secara *non-random*, serta aturan gabungan *chaos game* secara *random* dan *non-random* pada segilima.

3.4 Analisis Hasil

Analisis untuk hasil aturan *chaos game* secara *random*, modifikasi aturan *chaos game* secara *random*, aturan *chaos game* secara *non-random*, serta aturan gabungan *chaos game* secara *random* dan *non-random* pada segilima akan dijelaskan di tahapan ini. Hipotesis awal untuk aturan pemilihan titik sudut sebagai titik acuan *chaos game* pada segilima secara *random* akan menghasilkan suatu objek fraktal dengan melihat salah satu sifat fraktal yaitu *self-similarity*. Hipotesis awal untuk modifikasi aturan pemilihan titik sudut sebagai titik acuan *chaos game* pada segilima secara *random* akan menghasilkan suatu objek fraktal dengan melihat salah satu sifat fraktal yaitu *self-similarity*. Hipotesis awal untuk aturan pemilihan titik sudut sebagai titik acuan *chaos game* pada segilima secara *non-random* akan menghasilkan suatu objek yang bukan merupakan objek fraktal karena tidak memiliki sifat *self-similarity*. Serta hipotesis awal untuk aturan pemilihan titik sudut sebagai titik acuan *chaos game* pada segilima secara gabungan *random* dan *non-random* akan menghasilkan suatu objek fraktal dengan melihat salah satu sifat fraktal yaitu *self-similarity*. Akan dilakukan pembuktian apakah hasil dari masing-masing aturan dan modifikasi aturan pemilihan titik sudut sebagai titik acuan tersebut sesuai dengan hipotesis yang telah ditentukan. Untuk aturan *random*, modifikasi aturan *random* serta aturan gabungan *random* dan *non-random* akan dilakukan pembuktian analisis dengan membandingkan setiap iterasi sedangkan untuk aturan *non-random* akan dilakukan pembuktian analisis dengan membandingkan setiap iterasi serta perhitungan analitik untuk menunjukkan titik hasil bentukan *non-random chaos game* pada pemilihan titik sudut yang sama akan konvergen ke titik koordinat tertentu.

BAB 5. KESIMPULAN DAN SARAN

5. 1 Kesimpulan

Berdasarkan hasil dan pembahasan pada bab sebelumnya, maka dapat dibuat kesimpulan sebagai berikut:

1. Aturan *chaos game* secara *random* pada segilima menghasilkan suatu objek yang menyerupai bangun segilima dan merupakan objek fraktal.
2. Modifikasi aturan *chaos game* secara *random* pada segilima dengan titik sudut yang dipilih saat ini tidak boleh sama dengan titik sudut yang telah dipilih sebelumnya menghasilkan suatu objek fraktal, sedangkan modifikasi aturan *chaos game* secara *random* pada segilima dengan titik sudut saat ini dan sebelumnya tidak boleh dipilih untuk iterasi berikutnya tidak menghasilkan suatu objek fraktal.
3. Aturan *chaos game* secara *non-random* pada segilima tidak menghasilkan suatu objek fraktal tetapi kumpulan titik-titik hasil bentukan *non-random chaos game* memiliki karakteristik yaitu pada pemilihan titik sudut yang sama konvergen pada koordinat tertentu.
4. Aturan *chaos game* secara gabungan *random* dan *non-random* pada segilima dimungkinkan menghasilkan suatu objek fraktal. Semakin banyak titik-titik hasil bentukan *chaos game* dengan aturan *random* maka objek hasil bentukan titik-titik tersebut semakin mengarah pada fraktal, sedangkan semakin banyak titik-titik hasil bentukan *chaos game* dengan aturan *non-random* maka objek hasil bentukan titik-titik tersebut semakin tidak mengarah pada fraktal.

5. 2 Saran

Pada penelitian ini dilakukan aturan *chaos game* pada segilima secara *random*, modifikasi aturan *chaos game* secara *random*, aturan *chaos game* secara *non-random* serta aturan gabungan secara *random* dan *non-random*. Diharapkan pada penelitian selanjutnya untuk menunjukkan suatu objek merupakan objek fraktal yaitu dengan menggunakan kajian analitik.

DAFTAR PUSTAKA

Armana, R. F. 2016. Kajian Geometri Analitik pada Masalah *Chaos Game*. Skripsi. Jember: Fakultas MIPA Universitas Jember.

Bartle, R. G dan Sherbert, D. R. 2011. *Introduction to Real Analysis*. Fourth Edition. New York: John Wiley & Sons, Inc.

Devaney, R. L. 2003. *Fractal Patterns and Chaos Game*. Boston: Department of Mathematics Boston University.

Jeffrey, H. J. 1990. *Chaos Game Representation of Gene Structure*. USA: Nothern Illinois University.

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

Morisson, K. D. 2015. *GSCE Mathematics for OCR Higher Student Book*. United Kingdom: Cambridge University Press.

Peitgen, Heins-Otto, Jürgens, Hartmut, and Saupe, Dietmar. 2004. *Chaos and Fractals*. Second Edition. New York: Springer-Verlag New York, Inc.

Purnomo, K. D. 2014. Pembangkitan Segitiga Sierpinski dengan Transformasi Affine Berbasis Beberapa Benda Geometris. *Prosding Seminar Nasional Matematika*. Jurusan Matematika FMIPA Univeristas Jember.

Purnomo, K. D., Armana, Rere F., dan Kusno. 2016. Kajian Pembentukan Segitiga Sierpinski pada Masalah *Chaos Game* dengan Memanfaatkan Transformasi Affine. *Jurnal Matematika*. Vol. 6, No. 2:86-92.

Ratna, E. N. 2018. Modifikasi Aturan *Chaos Game* dengan Memanfaatkan Titik Berat Segitiga. Jember: Fakultas MIPA Universitas Jember.

Rich, B dan Thomas, C. 2009. *Geometry*. Fourth Edition. New York: McGraw-Hill.

Situmorang, E. R. 2016. Jenis-Jenis Segilima-Bola dan Sifat-Sifatnya. Skripsi. Yogyakarta: Fakultas MIPA Universitas Negeri Yogyakarta.

Sulistiyantoko, D. 2008. Aplikasi Sekuensi Deret pada Perhitungan Pembentukan Geometri Fraktal Sederhana. Skripsi. Yogyakarta: Program Studi Pendidikan Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga.

Wahjono, Tri D., Kasim, S., dan Riyadi, B. 2007. Perancangan Perangkat Lunak Generator Gambar dan Musik Fraktal dengan Metode *Iterated Function System*. *Jurnal CommIT*. Vol. 1, No. 2: 140-149.

Yunaning, F. 2018. Kajian Aturan *Non-Random Chaos Game* pada Segitiga. Skripsi. Jember: Fakultas MIPA Universitas Jember.

Zohuri, B. 2015. *Dimensional Analysis and Self-Similarity Methods for Engineers and Scientist*. Switzerland: Springer International Publishing.

LAMPIRAN

A. Hasil Perhitungan Jarak pada Percobaan ke-2 hingga Percobaan ke-10

1. Hasil perhitungan jarak pada percobaan ke-2

	Jarak kedua titik di masing-masing pemilihan titik sudut sebagai titik acuan				
	Titik Sudut A_0	Titik Sudut A_1	Titik Sudut A_2	Titik Sudut A_3	Titik Sudut A_4
Jarak T_n ke T_{n+5}	1,079370	0,067461	0,134921	0,269842	0,539685
$n = 1,2,3, \dots$	0,033730	0,002108	0,004216	0,008433	0,016865
	0,001054	0,000066	0.000132	0,000264	0,000527
	0,000033	0,000002	0,000004	0,000008	0,000016
	0,000001	0,000000	0,000000	0,000000	0,000001
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000

2. Hasil perhitungan jarak pada percobaan ke-3

	Jarak kedua titik di masing-masing pemilihan titik sudut sebagai titik acuan				
	Titik Sudut A_0	Titik Sudut A_1	Titik Sudut A_2	Titik Sudut A_3	Titik Sudut A_4
Jarak T_n ke T_{n+5}	0,560816	1,121631	0,280408	0,140204	0,070102
$n = 1,2,3, \dots$	0,017525	0,035051	0,008763	0,004381	0,002191
	0,000548	0,001095	0,000274	0,000137	0,000068
	0,000017	0,000034	0,000009	0,000004	0,000002
	0,000001	0,000001	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000

	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000

3. Hasil perhitungan jarak pada percobaan ke-4

	Jarak kedua titik di masing-masing pemilihan titik sudut sebagai titik acuan				
	Titik Sudut A_0	Titik Sudut A_1	Titik Sudut A_2	Titik Sudut A_3	Titik Sudut A_4
Jarak T_n ke T_{n+5}	0,282576	1,130304	0,565152	0,141288	0,070644
$n = 1,2,3, \dots$	0,008831	0,035322	0,017661	0,004415	0,002208
	0,000276	0,001104	0,000552	0,000138	0,000069
	0,000009	0,000034	0,000017	0,000004	0,000002
	0,000000	0,000001	0,000001	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000

4. Hasil perhitungan jarak pada percobaan ke-5

	Jarak kedua titik di masing-masing pemilihan titik sudut sebagai titik acuan				
	Titik Sudut A_0	Titik Sudut A_1	Titik Sudut A_2	Titik Sudut A_3	Titik Sudut A_4
Jarak T_n ke T_{n+5}	0,488391	0,244195	0,976781	0,122098	0,061049
$n = 1,2,3, \dots$	0,015262	0,007631	0,030524	0,003816	0,001908

	0,000477	0,000238	0,000954	0,000119	0,000060
	0,000015	0,000007	0,000030	0,000004	0,000002
	0,000001	0,000000	0,000001	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000

5. Hasil perhitungan jarak pada percobaan ke-6

	Jarak kedua titik di masing-masing pemilihan titik sudut sebagai titik acuan				
	Titik Sudut A_0	Titik Sudut A_1	Titik Sudut A_2	Titik Sudut A_3	Titik Sudut A_4
Jarak T_n ke T_{n+5}	0,273549	0,547098	1,094196	0,136775	0,068387
$n = 1,2,3,\ldots$	0,008548	0,017097	0,034194	0,004274	0,002137
	0,000267	0,000534	0,001069	0,000134	0,000067
	0,000008	0,000017	0,000033	0,000004	0,000002
	0,000000	0,000001	0,000001	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000

6. Hasil perhitungan jarak pada percobaan ke-7

	Jarak kedua titik di masing-masing pemilihan titik sudut sebagai titik acuan				
	Titik Sudut A_0	Titik Sudut A_1	Titik Sudut A_2	Titik Sudut A_3	Titik Sudut A_4
Jarak T_n ke T_{n+5}	0,223716	0,055929	0,111858	0,447432	0,027964
$n = 1,2,3, \ldots$	0,006991	0,001748	0,003496	0,013982	0,000874
	0,000218	0,000055	0,000109	0,000437	0,000027
	0,000007	0,000002	0,000003	0,000014	0,000001
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000

7. Hasil perhitungan jarak pada percobaan ke-8

	Jarak kedua titik di masing-masing pemilihan titik sudut sebagai titik acuan				
	Titik Sudut A_0	Titik Sudut A_1	Titik Sudut A_2	Titik Sudut A_3	Titik Sudut A_4
Jarak T_n ke T_{n+5}	0,272829	0,136439	0,068220	0,545758	0,034110
$n = 1,2,3, \ldots$	0,008527	0,004264	0,002132	0,017055	0,001066
	0,000266	0,000133	0,000067	0,000533	0,000033
	0,000008	0,000004	0,000002	0,000017	0,000001
	0,000000	0,000000	0,000000	0,000001	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000

	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000

8. Hasil perhitungan jarak pada percobaan ke-9

	Jarak kedua titik di masing-masing pemilihan titik sudut sebagai titik acuan				
	Titik Sudut A_0	Titik Sudut A_1	Titik Sudut A_2	Titik Sudut A_3	Titik Sudut A_4
Jarak T_n ke T_{n+5} $n = 1,2,3, \ldots$	0,493179	0,123295	0,246589	0,061647	0,986357
	0,015412	0,003853	0,007706	0,001926	0,030824
	0,000482	0,000120	0,000241	0,000060	0,000963
	0,000015	0,000004	0,000008	0,000002	0,000030
	0,000000	0,000000	0,000000	0,000000	0,000001
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000

9. Hasil perhitungan jarak pada percobaan ke-10

	Jarak kedua titik di masing-masing pemilihan titik sudut sebagai titik acuan				
	Titik Sudut A_0	Titik Sudut A_1	Titik Sudut A_2	Titik Sudut A_3	Titik Sudut A_4
Jarak T_n ke T_{n+5} $n = 1,2,3, \ldots$	0,057152	0,114304	0,228607	0,457215	0,914430
	0,001786	0,003572	0,007144	0,014288	0,028576
	0,000056	0,000112	0,000223	0,000446	0,000893
	0,000002	0,000003	0,000007	0,000014	0,000028
	0,000000	0,000000	0,000000	0,000000	0,000001
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000

	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000
	0,000000	0,000000	0,000000	0,000000	0,000000

B. Pembuktian dengan Induksi Matematika dan Pembuktian Kekonvergenan

1. Buktikan benar untuk $m = 1$ berlaku

a) Titik sudut A₀, $n = 5m + 1$

$$x_{T_6} = \left(\frac{x_0 + x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5}{2^5} + x_1\right)\frac{1}{2}$$

$$x_{T_6} = \left(\frac{x_0 + x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5 + 2^5x_1}{2^5}\right)\frac{1}{2}$$

$$x_{T_6} = \left(\frac{x_0 + x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5 + 2^5x_1}{2^6}\right)$$

$$x_{T_{5.1+1}} = \frac{x_0 + \sum_{i=0}^{1} 2^{5i}\, x_1 + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{1-1} 2^{5i}}{2^{5.1+1}}$$

$$x_{T_{5m+1}} = \frac{x_0 + \sum_{i=0}^{m} 2^{5i}\, x_1 + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{m-1} 2^{5i}}{2^{5m+1}}$$

Terbukti benar untuk $m = 1$

b) Titik sudut A₁, $n = 5m + 2$

$$x_{T_7} = \left(\frac{x_0 + x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5 + 2^5x_1}{2^6} + x_2\right)\frac{1}{2}$$

$$x_{T_7}=\left(\frac{x_0+x_1+2x_2+2^2x_3+2^3x_4+2^4x_5+2^5x_1+2^6x_2}{2^6}\right)\frac{1}{2}$$

$$x_{T_7}=\left(\frac{x_0+x_1+2x_2+2^2x_3+2^3x_4+2^4x_5+2^5x_1+2^6x_2}{2^7}\right)$$

$$x_{T_{5.1+2}}=\frac{x_0+(x_1+2x_2)\sum_{i=0}^{1}2^{5i}+(x_3+2x_4+2^2x_5)\sum_{i=0}^{1-1}2^{5i+2}}{2^{5.1+2}}$$

$$x_{T_{5m+2}}=\frac{x_0+(x_1+2x_2)\sum_{i=0}^{m}2^{5i}+(x_3+2x_4+2^2x_5)\sum_{i=0}^{m-1}2^{5i+2}}{2^{5m+2}}$$

Terbukti benar untuk $m=1$

c) Titik sudut A_2, $n=5m+3$

$$x_{T_8}=\left(\frac{x_0+x_1+2x_2+2^2x_3+2^3x_4+2^4x_5+2^5x_1+2^6x_2}{2^7}+x_3\right)\frac{1}{2}$$

$$x_{T_8}=\left(\frac{x_0+x_1+2x_2+2^2x_3+2^3x_4+2^4x_5+2^5x_1+2^6x_2+2^7x_3}{2^7}\right)\frac{1}{2}$$

$$x_{T_8}=\left(\frac{x_0+x_1+2x_2+2^2x_3+2^3x_4+2^4x_5+2^5x_1+2^6x_2+2^7x_3}{2^8}\right)$$

$$x_{T_{5.1+3}}=\frac{x_0+(x_1+2x_2+2^2x_3)\sum_{i=0}^{1}2^{5i}+(x_4+2x_5)\sum_{i=0}^{1-1}2^{5i+3}}{2^{5.1+3}}$$

$$x_{T_{5m+3}}=\frac{x_0+(x_1+2x_2+2^2x_3)\sum_{i=0}^{m}2^{5i}+(x_4+2x_5)\sum_{i=0}^{m-1}2^{5i+3}}{2^{5m+3}}$$

Terbukti benar untuk $m=1$

d) Titik sudut A_3, $n=5m+4$

$$x_{T_9}=\left(\frac{x_0+x_1+2x_2+2^2x_3+2^3x_4+2^4x_5+2^5x_1+2^6x_2+2^7x_3}{2^8}+x_4\right)\frac{1}{2}$$

$$x_{T_9} = \left(\frac{x_0 + x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5 + 2^5x_1 + 2^6x_2 + 2^7x_3 + 2^8x_4}{2^8}\right)\frac{1}{2}$$

$$x_{T_9} = \left(\frac{x_0 + x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5 + 2^5x_1 + 2^6x_2 + 2^7x_3 + 2^8x_4}{2^9}\right)$$

$$x_{T_{5.1+4}} = \frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4)\sum_{i=0}^{1} 2^{5i} + (x_5)\sum_{i=0}^{1-1} 2^{5i+4}}{2^{5.1+4}}$$

$$x_{T_{5m+4}} = \frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4)\sum_{i=0}^{m} 2^{5i} + (x_5)\sum_{i=0}^{m-1} 2^{5i+4}}{2^{5m+4}}$$

Terbukti benar untuk $m = 1$

e) Titik sudut A_4, $n = 5m + 5$

$$x_{T_{10}} = \left(\frac{x_0 + x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5 + 2^5x_1 + 2^6x_2 + 2^7x_3 + 2^8x_4}{2^9} + x_5\right)\frac{1}{2}$$

$$x_{T_{10}} = \left(\frac{x_0 + x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5 + 2^5x_1 + 2^6x_2 + 2^7x_3 + 2^8x_4 + +2^9x_5}{2^9}\right)\frac{1}{2}$$

$$x_{T_{10}} = \left(\frac{x_0 + x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5 + 2^5x_1 + 2^6x_2 + 2^7x_3 + 2^8x_4 + +2^9x_5}{2^{10}}\right)$$

$$x_{T_{5.1+5}} = \frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{1} 2^{5i}}{2^{5.1+5}}$$

$$x_{T_{5m+5}} = \frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{m} 2^{5i}}{2^{5m+5}}$$

Terbukti benar untuk $m = 1$

2. Asumsikan benar untuk $m = q$ berlaku

a) Titik sudut A_0

$$x_{T_{5q+1}} = \frac{x_0 + \sum_{i=0}^{q} 2^{5i}\, x_1 + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5 + 2^5x_1)\sum_{i=0}^{q-1} 2^{5i}}{2^{5q+1}}$$

b) Titik sudut A_1

$$x_{T_{5q+2}} = \frac{x_0 + (x_1 + 2x_2)\sum_{i=0}^{q} 2^{5i} + (x_3 + 2x_4 + 2^2x_5)\sum_{i=0}^{q-1} 2^{5i+2}}{2^{5q+2}}$$

c) Titik sudut A_2

$$x_{T_{5q+3}} = \frac{x_0 + (x_1 + 2x_2 + 2^2x_3)\sum_{i=0}^{q} 2^{5i} + (x_4 + 2x_5)\sum_{i=0}^{q-1} 2^{5i+3}}{2^{5q+3}}$$

d) Titik sudut A_3

$$x_{T_{5q+4}} = \frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4)\sum_{i=0}^{q} 2^{5i} + (x_5)\sum_{i=0}^{q-1} 2^{5i+4}}{2^{5q+4}}$$

e) Titik sudut A_4

$$x_{T_{5q+5}} = \frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{q} 2^{5i}}{2^{5q+5}}$$

3. Maka untuk $m = q + 1$ berlaku

a) Titik sudut A_0, $n = 5(q + 1)+1$

$$x_{T_{5(q+1)+1}} = \frac{1}{2}\left(T_{5q+5} + x_1\right)$$

$$= \frac{1}{2}\left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{q} 2^{5i}}{2^{5q+5}} + x_1\right)$$

$$= \frac{1}{2}\left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{q} 2^{5i} + 2^{5q+5}x_1}{2^{5q+5}}\right)$$

$$= \left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{q} 2^{5i} + 2^{5q+5}x_1}{2^{5q+6}}\right)$$

$$= \left(\frac{x_0 + x_1\sum_{i=0}^{q} 2^{5i} + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{q} 2^{5i} + 2^{5q+5}x_1}{2^{5q+6}}\right)$$

$$= \left(\frac{x_0 + x_1 \sum_{i=0}^{q+1} 2^{5i} + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5) \sum_{i=0}^{q} 2^{5i}}{2^{5q+6}} \right)$$

$$x_{T_{5(q+1)+1}} = \left(\frac{x_0 + x_1 \sum_{i=0}^{q+1} 2^{5i} + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5) \sum_{i=0}^{(q+1)-1} 2^{5i}}{2^{5(q+1)+1}} \right)$$

Terbukti benar untuk $m = q + 1$

b) Titik sudut A_1, $n = 5(q + 1)+2$

$$x_{T_{5(q+1)+2}} = \frac{1}{2}\left(T_{5(q+1)+1} + x_2\right)$$

$$= \frac{1}{2}\left(\frac{x_0 + x_1 \sum_{i=0}^{q+1} 2^{5i} + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5) \sum_{i=0}^{(q+1)-1} 2^{5i}}{2^{5(q+1)+1}} + x_2 \right)$$

$$= \frac{1}{2}\left(\frac{x_0 + x_1 \sum_{i=0}^{q+1} 2^{5i} + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5) \sum_{i=0}^{(q+1)-1} 2^{5i} + 2^{5(q+1)+1}x_2}{2^{5(q+1)+1}} \right)$$

$$= \left(\frac{x_0 + x_1 \sum_{i=0}^{q+1} 2^{5i} + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5) \sum_{i=0}^{(q+1)-1} 2^{5i} + 2^{5(q+1)+1}x_2}{2^{5(q+1)+2}} \right)$$

$$= \left(\frac{\begin{gathered} x_0 + x_1 \sum_{i=0}^{q+1} 2^{5i} + 2x_2 \sum_{i=0}^{(q+1)-1} 2^{5i} + (2^2x_3 + 2^3x_4 + 2^4x_5) \sum_{i=0}^{(q+1)-1} 2^{5i} \\ +2^{5(q+1)+1}x_2 \end{gathered}}{2^{5(q+1)+2}} \right)$$

$$= \left(\frac{x_0 + x_1 \sum_{i=0}^{q+1} 2^{5i} + 2x_2 \sum_{i=0}^{q+1} 2^{5i} + (2^2x_3 + 2^3x_4 + 2^4x_5) \sum_{i=0}^{(q+1)-1} 2^{5i}}{2^{5(q+1)+2}} \right)$$

$$x_{T_{5(q+1)+2}} = \left(\frac{x_0 + (x_1 + 2x_2) \sum_{i=0}^{q+1} 2^{5i} + (2^2x_3 + 2^3x_4 + 2^4x_5) \sum_{i=0}^{(q+1)-1} 2^{5i}}{2^{5(q+1)+2}} \right)$$

Terbukti benar untuk $m = q + 1$

c) Titik sudut A_2, $n = 5(q + 1)+3$

$$x_{T_{5(q+1)+3}} = \frac{1}{2}\left(T_{5(q+1)+2} + x_2\right)$$

$$= \frac{1}{2}\left(\frac{x_0 + (x_1 + 2x_2)\sum_{i=0}^{q+1} 2^{5i} + (2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{(q+1)-1} 2^{5i}}{2^{5(q+1)+2}} + x_3\right)$$

$$= \frac{1}{2}\left(\frac{x_0 + (x_1 + 2x_2)\sum_{i=0}^{q+1} 2^{5i} + (2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{(q+1)-1} 2^{5i} + 2^{5(q+1)+2}x_3}{2^{5(q+1)+2}}\right)$$

$$= \left(\frac{x_0 + (x_1 + 2x_2)\sum_{i=0}^{q+1} 2^{5i} + (2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{(q+1)-1} 2^{5i} + 2^{5(q+1)+2}x_3}{2^{5(q+1)+3}}\right)$$

$$= \left(\frac{x_0 + (x_1 + 2x_2)\sum_{i=0}^{q+1} 2^{5i} + 2^2x_3\sum_{i=0}^{(q+1)-1} 2^{5i} + (2^3x_4 + 2^4x_5)\sum_{i=0}^{(q+1)-1} 2^{5i} + 2^{5(q+1)+3}x_3}{2^{5(q+1)+3}}\right)$$

$$= \left(\frac{x_0 + (x_1 + 2x_2)\sum_{i=0}^{q+1} 2^{5i} + 2^2x_3\sum_{i=0}^{q+1} 2^{5i} + (2^3x_4 + 2^4x_5)\sum_{i=0}^{(q+1)-1} 2^{5i}}{2^{5(q+1)+3}}\right)$$

$$x_{T_{5(q+1)+3}} = \left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3)\sum_{i=0}^{q+1} 2^{5i} + (2^3x_4 + 2^4x_5)\sum_{i=0}^{(q+1)-1} 2^{5i}}{2^{5(q+1)+3}}\right)$$

Terbukti benar untuk $m = q + 1$

d) Titik sudut A$_3$, $n = 5(q + 1)+4$

$$x_{T_{5(q+1)+4}} = \frac{1}{2}\left(T_{5(q+1)+3} + x_4\right)$$

$$= \frac{1}{2}\left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3)\sum_{i=0}^{q+1} 2^{5i} + (2^3x_4 + 2^4x_5)\sum_{i=0}^{(q+1)-1} 2^{5i}}{2^{5(q+1)+3}} + x_4\right)$$

$$= \frac{1}{2}\left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3)\sum_{i=0}^{q+1} 2^{5i} + (2^3x_4 + 2^4x_5)\sum_{i=0}^{(q+1)-1} 2^{5i} + 2^{5(q+1)+3}x_4}{2^{5(q+1)+3}}\right)$$

$$= \left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3)\sum_{i=0}^{q+1} 2^{5i} + (2^3x_4 + 2^4x_5)\sum_{i=0}^{(q+1)-1} 2^{5i} + 2^{5(q+1)+3}x_4}{2^{5(q+1)+4}}\right)$$

$$= \left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3)\sum_{i=0}^{q+1} 2^{5i} + 2^3x_4\sum_{i=0}^{(q+1)-1} 2^{5i} + 2^4x_5\sum_{i=0}^{(q+1)-1} 2^{5i} + 2^{5(q+1)+3}x_4}{2^{5(q+1)+4}}\right)$$

$$= \left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3)\sum_{i=0}^{q+1} 2^{5i} + 2^3x_4\sum_{i=0}^{q+1} 2^{5i} + 2^4x_5\sum_{i=0}^{(q+1)-1} 2^{5i}}{2^{5(q+1)+4}}\right)$$

$$x_{T_{5(q+1)+4}} = \left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4)\sum_{i=0}^{q+1} 2^{5i} + 2^4x_5\sum_{i=0}^{(q+1)-1} 2^{5i}}{2^{5(q+1)+4}}\right)$$

Terbukti benar untuk $m = q + 1$

e) Titik sudut A_4, $n = 5(q + 1)$+5

$$x_{T_{5(q+1)+5}} = \frac{1}{2}\left(T_{5(q+1)+4} + x_5\right)$$

$$= \frac{1}{2}\left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4)\sum_{i=0}^{q+1} 2^{5i} + 2^4x_5\sum_{i=0}^{(q+1)-1} 2^{5i}}{2^{5(q+1)+4}} + x_5\right)$$

$$= \frac{1}{2}\left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4)\sum_{i=0}^{q+1} 2^{5i} + 2^4x_5\sum_{i=0}^{(q+1)-1} 2^{5i} + 2^{5(q+1)+4}x_5}{2^{5(q+1)+4}}\right)$$

$$= \left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4)\sum_{i=0}^{q+1} 2^{5i} + 2^4x_5\sum_{i=0}^{(q+1)-1} 2^{5i} + 2^{5(q+1)+4}x_5}{2^{5(q+1)+5}}\right)$$

$$= \left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4)\sum_{i=0}^{q+1} 2^{5i} + 2^4x_5\sum_{i=0}^{q+1} 2^{5i}}{2^{5(q+1)+5}}\right)$$

$$x_{T_{5(q+1)+4}} = \left(\frac{x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{q+1} 2^{5i}}{2^{5(q+1)+5}}\right)$$

Terbukti benar untuk $m = q + 1$

Karena kelima rumus koordinat tersebut pada langkah 1, 2, dan 3 untuk koordinat x bernilai benar, maka dapat disimpulkan bahwa untuk koordinat y rumus tersebut adalah benar juga, pada semua n sama dengan bilangan asli. Kemudian, akan dilakukan pembuktian kekonvergenan pada masing-masing kumpulan titik pada pemilihan titik sudut sebagai titik acuan yang sama. Langkah ini dilakukan dengan menghitung jarak antara dua titik pada pemilihan titik sudut sebagai titik acuan yang sama dengan memanfaatkan koordinat (x_{T_n}, y_{T_n}) masing-masing titik sudut.

a. Pemilihan Titik Sudut A$_0$

Di bawah ini adalah rumus jarak antara dua titik bentukan *non-random chaos game* pada segilima saat pemilihan titik sudut A$_0$ dimana $n = 5m + 1$.

$$\left|\overline{T_{5(m-1)+1}T_{5m+1}}\right| = \sqrt{\left(x_{T_{5m+1}} - x_{T_{5(m-1)+1}}\right)^2 + \left(y_{T_{5m+1}} - y_{T_{5(m-1)+1}}\right)^2}$$

$$= \sqrt{\left(\frac{x_0 + \sum_{i=0}^{m} 2^{5i}x_1 + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{m-1} 2^{5i}}{2^{5m+1}} - \frac{x_0 + \sum_{i=0}^{m-1} 2^{5i}x_1 + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{(m-1)-1} 2^{5i}}{2^{5(m-1)+1}}\right)^2 + \left(\frac{y_0 + \sum_{i=0}^{m} 2^{5i}y_1 + (2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)\sum_{i=0}^{m-1} 2^{5i}}{2^{5m+1}} - \frac{y_0 + \sum_{i=0}^{m-1} 2^{5i}y_1 + (2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)\sum_{i=0}^{(m-1)-1} 2^{5i}}{2^{5(m-1)+1}}\right)^2}$$

$$= \sqrt{\left(\frac{\begin{array}{c} x_0 + \sum_{i=0}^{m} 2^{5i}x_1 + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{m-1} 2^{5i} - \\ 2^5x_0 + 2^5\sum_{i=0}^{m-1} 2^{5i}x_1 + 2^5(2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{(m-1)-1} 2^{5i} \end{array}}{2^{5m+1}}\right)^2 + \left(\frac{\begin{array}{c} y_0 + \sum_{i=0}^{m} 2^{5i}y_1 + (2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)\sum_{i=0}^{m-1} 2^{5i} - \\ 2^5y_0 + 2^5\sum_{i=0}^{m-1} 2^{5i}y_1 + 2^5(2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)\sum_{i=0}^{(m-1)-1} 2^{5i} \end{array}}{2^{5m+1}}\right)^2}$$

$$= \sqrt{\left(\frac{(1-2^5)x_0 + 2^{5m}x_1 + (1-2^5)\sum_{i=0}^{m-1} 2^{5i}x_1 + 2^{5(m-1)}(2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5) \\ (1-2^5)(2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\sum_{i=0}^{(m-1)-1} 2^{5i}}{2^{5m+1}}\right)^2 + \left(\frac{(1-2^5)y_0 + 2^{5m}y_1 + (1-2^5)\sum_{i=0}^{m-1} 2^{5i}y_1 + 2^{5(m-1)}(2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5) \\ (1-2^5)(2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)\sum_{i=0}^{(m-1)-1} 2^{5i}}{2^{5m+1}}\right)^2}$$

$$= \sqrt{\left(\frac{(1-2^5)x_0 + (2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i})x_1 + (2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i}) \\ (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)}{2^{5m+1}}\right)^2 + \left(\frac{(1-2^5)y_0 + (2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i})y_1 + (2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i}) \\ (2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)}{2^{5m+1}}\right)^2}$$

Setelah mengetahui rumus di atas, maka akan dibuktikan $\left(2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i}\right) = 1$ dan $(2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i}) = 1$ untuk $m > 1$ dengan induksi matematika sebagai berikut.

1. Buktikan benar untuk $m = 2$, maka berlaku

a. $\left(2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i}\right) = 1$

$$\left(2^{5.2} - 31\sum_{i=0}^{2-1} 2^{5i}\right) = 1$$

$$\left(2^{10} - 31(1 + 2^5)\right) = 1$$

$$(2^{10} - 31(33)) = 1$$

$$(1024 - 1023) = 1$$

$$1 = 1$$

Terbukti benar untuk $m = 2$

b. $(2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i}) = 1$

$$(2^{5(2-1)} - 31\sum_{i=0}^{(2-1)-1} 2^{5i}) = 1$$

$$(2^5 - 31(1)) = 1$$

$(32-31)=1$

$1=1$

Terbukti benar untuk $m=2$

2. Asumsikan benar untuk $m=q$, maka berlaku

a. $\left(2^{5m}-31\sum_{i=0}^{m-1}2^{5i}\right)=1$

$$\left(2^{5q}-31\sum_{i=0}^{q-1}2^{5i}\right)=1$$

b. $(2^{5(m-1)}-31\sum_{i=0}^{(m-1)-1}2^{5i})=1$

$(2^{5(q-1)}-31\sum_{i=0}^{(q-1)-1}2^{5i})=1$

3. Maka untuk $m=q+1$ berlaku

a. $\left(2^{5(q+1)}-31\sum_{i=0}^{(q+1)-1}2^{5i}\right)=2^{5q+5}-31\sum_{i=0}^{(q+1)-1}2^{5i}$

$$=2^{5q}.2^{5}-31\left(\sum_{i=0}^{q-1}2^{5i}+2^{5q}\right)$$

$$=2^{5q}.2^{5}-31\sum_{i=0}^{q-1}2^{5i}-31.2^{5q}$$

$$=2^{5q}.32-31.2^{5q}-(2^{5q}-1)$$

$$=2^{5q}-(2^{5q}-1)$$

$$=1$$

Terbukti benar untuk $m=q+1$

b. $\left(2^{5(q+1)-1}-31\sum_{i=0}^{((q+1)-1)-1}2^{5i}\right)=2^{5q}-31\sum_{i=0}^{((q+1)-1)-1}2^{5i}$

$$=2^{5q}-31\left(\sum_{i=0}^{(q-1)-1}2^{5i}+2^{5(q-1)}\right)$$

$$= 2^{5(q-1)}.2^5 - 31\sum_{i=0}^{(q-1)-1} 2^{5i} - 31.2^{5(q-1)}$$

$$= 2^{5(q-1)}.32 - 31.2^{5(q-1)} - \left(2^{5(q-1)} - 1\right)$$

$$= 2^{5(q-1)} - (2^{5(q-1)} - 1)$$

$$= 1$$

Terbukti benar untuk $m = q + 1$

Menurut pembuktian di atas pada $\left(2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i}\right) = 1$ dan $(2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i}) = 1$ untuk $m > 1$ bernilai benar, maka diperoleh rumus umum jarak antara dua titik sebagai berikut

$$\left|\overline{T_{5(m-1)+1}T_{5m+1}}\right| = \sqrt{\left(\frac{\left(1-2^5\right)x_0 + x_1 + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)}{2^{5m+1}}\right)^2 + \left(\frac{\left(1-2^5\right)y_0 + y_1 + \left(2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5\right)}{2^{5m+1}}\right)^2}$$

Selanjutnya akan dibuktikan $\left|\overline{T_{5(m-1)+1}T_{5m+1}}\right|$ untuk konvergen dengan mencari nilai $\lim_{m\to\infty}\left|\overline{T_{5(m-1)+1}T_{5m+1}}\right|$, maka berdasarkan Teorema 2.1 didapatkan seperti di bawah ini.

$$\lim_{m\to\infty}\sqrt{\left(\frac{\left(1-2^5\right)x_0 + x_1 + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)}{2^{5m+1}}\right)^2 + \left(\frac{\left(1-2^5\right)y_0 + y_1 + \left(2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5\right)}{2^{5m+1}}\right)^2}$$

$$= \lim_{m\to\infty}\frac{1}{\sqrt{(2^{5m+1})^2}}\sqrt{\left(\left(1-2^5\right)x_0 + x_1 + (2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\right)^2 + \left(\left(1-2^5\right)y_0 + y_1 + \left(2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5\right)\right)^2}$$

Akan dilakukan pembuktian bahwa $\lim_{m\to\infty}\frac{1}{\sqrt{(2^{5m+1})^2}}$ konvergen berdasarkan Teorema 2.2. Dalam hal ini

$$\frac{1}{\sqrt{(2^{5m+1})^2}} = \frac{1}{2^{5m+1}} = \frac{1}{2^{5m}} \cdot \frac{1}{2}$$

sehingga pembuktiannya seperti berikut.

$$\lim_{m\to\infty} \frac{1}{2^{5m}} = \left(\lim_{m\to\infty} \left(\frac{1}{2}\right)^m\right)^5 = (0)^5 = 0.$$

Karena hasil dari $\lim_{m\to\infty} \left|\overline{T_{5(m-1)+1}T_{5m+1}}\right|$=0, maka dapat disimpulkan bahwa rumus tersebut konvergen. Sehingga, kumpulan titik-titik bentukan *non-random chaos game* pada segilima saat pemilihan titik sudut A_0 sebagai acuan konvergen ke titik koordinat tertentu.

b. Pemilihan Titik Sudut A_1

Di bawah ini adalah rumus jarak antara dua titik bentukan *non-random chaos game* pada segilima saat pemilihan titik sudut A_1 dimana $n = 5m + 2$.

$$\left|\overline{T_{5(m-1)+2}T_{5m+2}}\right| = \sqrt{\left(x_{T_{5m+2}} - x_{T_{5(m-1)+2}}\right)^2 + \left(y_{T_{5m+2}} - y_{T_{5(m-1)+2}}\right)^2}$$

$$= \sqrt{\begin{aligned}&\left(\frac{x_0 + \sum_{i=0}^{m} 2^{5i}(x_1 + 2x_2) + (x_3 + 2x_4 + 2^2x_5)\sum_{i=0}^{m-1} 2^{5i+2}}{2^{5m+2}} - \frac{x_0 + \sum_{i=0}^{m-1} 2^{5i}(x_1 + 2x_2) + (x_3 + 2x_4 + 2^2x_5)\sum_{i=0}^{(m-1)-1} 2^{5i+2}}{2^{5(m-1)+2}}\right)^2 + \\ &\left(\frac{y_0 + \sum_{i=0}^{m} 2^{5i}(y_1 + 2y_2) + (y_3 + 2y_4 + 2^2y_5)\sum_{i=0}^{m-1} 2^{5i+2}}{2^{5m+2}} - \frac{y_0 + \sum_{i=0}^{m-1} 2^{5i}(y_1 + 2y_2) + (y_3 + 2y_4 + 2^2y_5)\sum_{i=0}^{(m-1)-1} 2^{5i+2}}{2^{5(m-1)+2}}\right)^2\end{aligned}}$$

$$= \sqrt{\left(\frac{\begin{gathered}x_0 + \sum_{i=0}^{m} 2^{5i}(x_1 + 2x_2) + (x_3 + 2x_4 + 2^2x_5)\sum_{i=0}^{m-1} 2^{5i+2} - \\ 2^5x_0 + 2^5\sum_{i=0}^{m-1} 2^{5i}(x_1 + 2x_2) + 2^5(x_3 + 2x_4 + 2^2x_5)\sum_{i=0}^{(m-1)-1} 2^{5i+2}\end{gathered}}{2^{5m+2}}\right)^2 + \left(\frac{\begin{gathered}y_0 + \sum_{i=0}^{m} 2^{5i}(y_1 + 2y_2) + (y_3 + 2y_4 + 2^2y_5)\sum_{i=0}^{m-1} 2^{5i+2} - \\ 2^5y_0 + 2^5\sum_{i=0}^{m-1} 2^{5i}(y_1 + 2y_2) + 2^5(y_3 + 2y_4 + 2^2y_5)\sum_{i=0}^{(m-1)-1} 2^{5i+2}\end{gathered}}{2^{5m+2}}\right)^2}$$

$$= \sqrt{\left(\frac{\begin{gathered}(1-2^5)x_0 + (2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i})(x_1 + 2x_2) + (2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i}) \\ (2^2x_3 + 2^3x_4 + 2^4x_5)\end{gathered}}{2^{5m+2}}\right)^2 + \left(\frac{\begin{gathered}(1-2^5)y_0 + (2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i})(y_1 + 2y_2) + (2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i}) \\ (2^2y_3 + 2^3y_4 + 2^4y_5)\end{gathered}}{2^{5m+2}}\right)^2}$$

Karena sudah dibuktikan benar dengan induksi matematika saat pemilihan titik sudut A$_1$ bahwa $\left(2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i}\right) = 1$ dan $\left(2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i}\right) = 1$ untuk $m \in N > 1$, maka didapatkan rumus umum jarak antara dua titik seperti berikut

$$\left|\overline{T_{5(m-1)+2}T_{5m+2}}\right| = \sqrt{\left(\frac{(1-2^5)x_0 + (x_1 + 2x_2) + (2^2x_3 + 2^3x_4 + 2^4x_5)}{2^{5m+2}}\right)^2 + \left(\frac{(1-2^5)y_0 + (y_1 + 2y_2) + (2^2y_3 + 2^3y_4 + 2^4y_5)}{2^{5m+2}}\right)^2}$$

Selanjutnya akan dibuktikan $\left|\overline{T_{5(m-1)+2}T_{5m+2}}\right|$ untuk konvergen dengan mencari nilai $\lim_{m\to\infty}\left|\overline{T_{5(m-1)+2}T_{5m+2}}\right|$, maka berdasarkan Teorema 2.1 didapatkan seperti di bawah ini.

$$\lim_{m\to\infty}\sqrt{\left(\frac{(1-2^5)x_0+(x_1+2x_2)+(2^2x_3+2^3x_4+2^4x_5)}{2^{5m+2}}\right)^2+\left(\frac{(1-2^5)y_0+(y_1+2y_2)+(2^2y_3+2^3y_4+2^4y_5)}{2^{5m+2}}\right)^2}$$

$$=\lim_{m\to\infty}\frac{1}{\sqrt{(2^{5m+2})^2}}\sqrt{\left((1-2^5)x_0+(x_1+2x_2)+(2^2x_3+2^3x_4+2^4x_5)\right)^2+\left((1-2^5)y_0+(y_1+2y_2)+(2^2y_3+2^3y_4+2^4y_5)\right)^2}$$

Akan dilakukan pembuktian bahwa $\lim_{m\to\infty}\frac{1}{\sqrt{(2^{5m+2})^2}}$ konvergen berdasarkan Teorema 2.2. Dalam hal ini

$$\frac{1}{\sqrt{(2^{5m+2})^2}}=\frac{1}{2^{5m+2}}=\frac{1}{2^{5m}}\cdot\frac{1}{2^2}$$

sehingga pembuktiannya seperti berikut.

$$\lim_{m\to\infty}\frac{1}{2^{5m}}=\left(\lim_{m\to\infty}\left(\frac{1}{2}\right)^m\right)^5=(0)^5=0.$$

Karena hasil dari $\lim_{m\to\infty}\left|\overline{T_{5(m-1)+2}T_{5m+2}}\right|$=0, maka dapat disimpulkan bahwa rumus tersebut konvergen. Sehingga, kumpulan titik-titik bentukan *non-random chaos game* pada segilima saat pemilihan titik sudut A_1 sebagai acuan konvergen ke titik koordinat tertentu.

c. Pemilihan Titik Sudut A_2

Di bawah ini adalah rumus jarak antara dua titik bentukan *non-random chaos game* pada segilima saat pemilihan titik sudut A_2 dimana $n=5m+3$.

$$\left|\overline{T_{5(m-1)+3}T_{5m+3}}\right|=\sqrt{\left(x_{T_{5m+3}}-x_{T_{5(m-1)+3}}\right)^2+\left(y_{T_{5m+3}}-y_{T_{5(m-1)+3}}\right)^2}$$

$$= \sqrt{\left(\frac{x_0 + \sum_{i=0}^{m} 2^{5i}(x_1 + 2x_2 + 2^2x_3) + (x_4 + 2x_5)\sum_{i=0}^{m-1} 2^{5i+3}}{2^{5m+3}} - \frac{x_0 + \sum_{i=0}^{m-1} 2^{5i}(x_1 + 2x_2 + 2^2x_3) + (x_4 + 2x_5)\sum_{i=0}^{(m-1)-1} 2^{5i+3}}{2^{5(m-1)+3}}\right)^2 + \left(\frac{y_0 + \sum_{i=0}^{m} 2^{5i}(y_1 + 2y_2 + 2^2y_3) + (y_4 + 2y_5)\sum_{i=0}^{m-1} 2^{5i+3}}{2^{5m+3}} - \frac{y_0 + \sum_{i=0}^{m-1} 2^{5i}(y_1 + 2y_2 + 2^2y_3) + (y_4 + 2y_5)\sum_{i=0}^{(m-1)-1} 2^{5i+3}}{2^{5(m-1)+3}}\right)^2}$$

$$= \sqrt{\left(\frac{x_0 + \sum_{i=0}^{m} 2^{5i}(x_1 + 2x_2 + 2^2x_3) + (x_4 + 2x_5)\sum_{i=0}^{m-1} 2^{5i+3} - 2^5x_0 + 2^5\sum_{i=0}^{m-1} 2^{5i}(x_1 + 2x_2 + 2^2x_3) + 2^5(x_4 + 2x_5)\sum_{i=0}^{(m-1)-1} 2^{5i+3}}{2^{5m+3}}\right)^2 + \left(\frac{y_0 + \sum_{i=0}^{m} 2^{5i}(y_1 + 2y_2 + 2^2y_3) + (y_4 + 2y_5)\sum_{i=0}^{m-1} 2^{5i+3} - 2^5y_0 + 2^5\sum_{i=0}^{m-1} 2^{5i}(y_1 + 2y_2 + 2^2y_3) + 2^5(y_4 + 2y_5)\sum_{i=0}^{(m-1)-1} 2^{5i+3}}{2^{5m+3}}\right)^2}$$

$$= \sqrt{\left(\frac{(1 - 2^5)x_0 + (2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i})(x_1 + 2x_2 + 2^2x_3) + (2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i})(2^3x_4 + 2^4x_5)}{2^{5m+3}}\right)^2 + \left(\frac{(1 - 2^5)y_0 + (2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i})((y_1 + 2y_2 + 2^2y_3) + (2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i})(2^3y_4 + 2^4y_5)}{2^{5m+3}}\right)^2}$$

Karena sudah dibuktikan benar dengan induksi matematika saat pemilihan titik sudut A_1 bahwa $\left(2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i}\right) = 1$ dan $\left(2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i}\right) = 1$ untuk $m \in N > 1$, maka didapatkan rumus umum jarak antara dua titik seperti berikut

$$\left|\overline{T_{5(m-1)+3}T_{5m+3}}\right| = \sqrt{\begin{array}{l}\left(\frac{(1-2^5)x_0 + (x_1 + 2x_2 + 2^2x_3) + (2^3x_4 + 2^4x_5)}{2^{5m+3}}\right)^2 + \\ \left(\frac{(1-2^5)y_0 + (y_1 + 2y_2 + 2^2y_3) + (2^3y_4 + 2^4y_5)}{2^{5m+3}}\right)^2\end{array}}$$

Selanjutnya akan dibuktikan $\left|\overline{T_{5(m-1)+3}T_{5m+3}}\right|$ untuk konvergen dengan mencari nilai $\lim_{m\to\infty}\left|\overline{T_{5(m-1)+3}T_{5m+3}}\right|$, maka berdasarkan Teorema 2.1 didapatkan seperti di bawah ini.

$$\lim_{m\to\infty}\sqrt{\begin{array}{l}\left(\frac{(1-2^5)x_0 + (x_1 + 2x_2 + 2^2x_3) + (2^3x_4 + 2^4x_5)}{2^{5m+3}}\right)^2 + \\ \left(\frac{(1-2^5)y_0 + (y_1 + 2y_2 + 2^2y_3) + (2^3y_4 + 2^4y_5)}{2^{5m+3}}\right)^2\end{array}}$$

$$= \lim_{m\to\infty}\frac{1}{\sqrt{(2^{5m+3})^2}}\sqrt{\begin{array}{l}\left((1-2^5)x_0 + (x_1 + 2x_2 + 2^2x_3) + (2^3x_4 + 2^4x_5)\right)^2 + \\ \left((1-2^5)y_0 + (y_1 + 2y_2 + 2^2y_3) + (2^3y_4 + 2^4y_5)\right)^2\end{array}}$$

Akan dilakukan pembuktian bahwa $\lim_{m\to\infty}\frac{1}{\sqrt{(2^{5m+3})^2}}$ konvergen berdasarkan Teorema 2.2. Dalam hal ini

$$\frac{1}{\sqrt{(2^{5m+3})^2}} = \frac{1}{2^{5m+3}} = \frac{1}{2^{5m}}\cdot\frac{1}{2^3}$$

sehingga pembuktiannya seperti berikut.

$$\lim_{m\to\infty}\frac{1}{2^{5m}} = \left(\lim_{m\to\infty}\left(\frac{1}{2}\right)^m\right)^5 = (0)^5 = 0.$$

Karena hasil dari $\lim_{m\to\infty}\left|\overline{T_{5(m-1)+3}T_{5m+3}}\right|$=0, maka dapat disimpulkan bahwa rumus tersebut konvergen. Sehingga, kumpulan titik-titik bentukan *non-random chaos game* pada segilima saat pemilihan titik sudut A_2 sebagai acuan konvergen ke titik koordinat tertentu.

d. Pemilihan Titik Sudut A_3

Di bawah ini adalah rumus jarak antara dua titik bentukan *non-random chaos game* pada segilima saat pemilihan titik sudut A_3 dimana $n = 5m + 4$.

$$\left|\overline{T_{5(m-1)+4}T_{5m+4}}\right| = \sqrt{\left(x_{T_{5m+4}} - x_{T_{5(m-1)+4}}\right)^2 + \left(y_{T_{5m+4}} - y_{T_{5(m-1)+4}}\right)^2}$$

$$= \sqrt{\begin{array}{c}\left(\dfrac{x_0 + \sum_{i=0}^{m} 2^{5i}(x_1 + 2x_2 + 2^2x_3 + 2^3x_4) + (x_5)\sum_{i=0}^{m-1} 2^{5i+4}}{2^{5m+4}} - \dfrac{x_0 + \sum_{i=0}^{m-1} 2^{5i}(x_1 + 2x_2 + 2^2x_3 + 2^3x_4) + (x_5)\sum_{i=0}^{(m-1)-1} 2^{5i+4}}{2^{5(m-1)+4}}\right)^2 + \\ \left(\dfrac{y_0 + \sum_{i=0}^{m} 2^{5i}(y_1 + 2y_2 + 2^2y_3 + 2^3y_4) + (y_5)\sum_{i=0}^{m-1} 2^{5i+4}}{2^{5m+4}} - \dfrac{y_0 + \sum_{i=0}^{m-1} 2^{5i}(y_1 + 2y_2 + 2^2y_3 + 2^3y_4) + (y_5)\sum_{i=0}^{(m-1)-1} 2^{5i+4}}{2^{5(m-1)+4}}\right)^2\end{array}}$$

$$= \sqrt{\begin{array}{c}\left(\dfrac{\begin{array}{c}x_0 + \sum_{i=0}^{m} 2^{5i}(x_1 + 2x_2 + 2^2x_3 + 2^3x_4) + (x_5)\sum_{i=0}^{m-1} 2^{5i+4} - \\ 2^5x_0 + 2^5\sum_{i=0}^{m-1} 2^{5i}(x_1 + 2x_2 + 2^2x_3 + 2^3x_4) + 2^5(x_5)\sum_{i=0}^{(m-1)-1} 2^{5i+4}\end{array}}{2^{5m+4}}\right)^2 + \\ \left(\dfrac{\begin{array}{c}y_0 + \sum_{i=0}^{m} 2^{5i}(y_1 + 2y_2 + 2^2y_3 + 2^3y_4) + (y_5)\sum_{i=0}^{m-1} 2^{5i+4} - \\ 2^5y_0 + 2^5\sum_{i=0}^{m-1} 2^{5i}(y_1 + 2y_2 + 2^2y_3 + 2^3y_4) + 2^5(y_5)\sum_{i=0}^{(m-1)-1} 2^{5i+4}\end{array}}{2^{5m+4}}\right)^2\end{array}}$$

$$= \sqrt{\begin{array}{c}\left(\dfrac{\begin{array}{c}(1-2^5)x_0 + (2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i})(x_1 + 2x_2 + 2^2x_3 + 2^3x_4) + \\ (2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i})(2^4x_5)\end{array}}{2^{5m+4}}\right)^2 + \\ \left(\dfrac{\begin{array}{c}(1-2^5)y_0 + (2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i})(y_1 + 2y_2 + 2^2y_3 + 2^3y_4) + \\ (2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i})(2^4y_5)\end{array}}{2^{5m+4}}\right)^2\end{array}}$$

Karena sudah dibuktikan benar dengan induksi matematika saat pemilihan titik sudut A1 bahwa $\left(2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i}\right) = 1$ dan $\left(2^{5(m-1)} - 31\sum_{i=0}^{(m-1)-1} 2^{5i}\right) = 1$ untuk $m \in N > 1$, maka didapatkan rumus umum jarak antara dua titik seperti berikut

$$\left|\overline{T_{5(m-1)+4}T_{5m+4}}\right| = \sqrt{\left(\frac{(1-2^5)x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4) + (2^4x_5)}{2^{5m+4}}\right)^2 + \left(\frac{(1-2^5)y_0 + (y_1 + 2y_2 + 2^2y_3 + 2^3y_4) + (2^4y_5)}{2^{5m+4}}\right)^2}$$

Selanjutnya akan dibuktikan $\left|\overline{T_{5(m-1)+4}T_{5m+4}}\right|$ untuk konvergen dengan mencari nilai $\lim_{m\to\infty}\left|\overline{T_{5(m-1)+4}T_{5m+4}}\right|$, maka berdasarkan Teorema 2.1 didapatkan seperti di bawah ini.

$$\lim_{m\to\infty}\sqrt{\left(\frac{(1-2^5)x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4) + (2^4x_5)}{2^{5m+4}}\right)^2 + \left(\frac{(1-2^5)y_0 + (y_1 + 2y_2 + 2^2y_3 + 2^3y_4) + (2^4y_5)}{2^{5m+4}}\right)^2}$$

$$= \lim_{m\to\infty}\frac{1}{\sqrt{(2^{5m+4})^2}}\sqrt{\left((1-2^5)x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4) + (2^4x_5)\right)^2 + \left((1-2^5)y_0 + (y_1 + 2y_2 + 2^2y_3 + 2^3y_4) + (2^4y_5)\right)^2}$$

Akan dilakukan pembuktian bahwa $\lim_{m\to\infty}\frac{1}{\sqrt{(2^{5m+4})^2}}$ konvergen berdasarkan Teorema 2.2. Dalam hal ini

$$\frac{1}{\sqrt{(2^{5m+4})^2}} = \frac{1}{2^{5m+4}} = \frac{1}{2^{5m}}.\frac{1}{2^4}$$

sehingga pembuktiannya seperti berikut.

$$\lim_{m\to\infty}\frac{1}{2^{5m}} = \left(\lim_{m\to\infty}\left(\frac{1}{2}\right)^m\right)^5 = (0)^5 = 0.$$

Karena hasil dari $\lim_{m\to\infty}\left|\overline{T_{5(m-1)+4}T_{5m+4}}\right|$=0, maka dapat disimpulkan bahwa rumus tersebut konvergen. Sehingga, kumpulan titik-titik bentukan *non-random chaos game* pada segilima saat pemilihan titik sudut A_3 sebagai acuan konvergen ke titik koordinat tertentu.

e. Pemilihan Titik Sudut A_4

Di bawah ini adalah rumus jarak antara dua titik bentukan *non-random chaos game* pada segilima saat pemilihan titik sudut A_4 dimana $n = 5m + 5$.

$$\left|\overline{T_{5(m-1)+5}T_{5m+5}}\right| = \sqrt{\left(x_{T_{5m+5}} - x_{T_{5(m-1)+5}}\right)^2 + \left(y_{T_{5m+5}} - y_{T_{5(m-1)+5}}\right)^2}$$

$$= \sqrt{\begin{array}{c}\left(\dfrac{x_0 + \sum_{i=0}^{m} 2^{5i}(x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)}{2^{5m+5}} - \dfrac{x_0 + \sum_{i=0}^{m-1} 2^{5i}(x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)}{2^{5(m-1)+5}}\right)^2 + \\ \left(\dfrac{y_0 + \sum_{i=0}^{m} 2^{5i}(y_1 + 2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)}{2^{5m+5}} - \dfrac{y_0 + \sum_{i=0}^{m-1} 2^{5i}(y_1 + 2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)}{2^{5(m-1)+5}}\right)^2\end{array}}$$

$$= \sqrt{\begin{array}{c}\left(\dfrac{x_0 + \sum_{i=0}^{m} 2^{5i}(x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)}{2^{5m+5}} - \dfrac{2^5x_0 + 2^5\sum_{i=0}^{m-1} 2^{5i}(x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)}{2^{5(m-1)+5}}\right)^2 + \\ \left(\dfrac{y_0 + \sum_{i=0}^{m} 2^{5i}(y_1 + 2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)}{2^{5m+5}} - \dfrac{2^5y_0 + 2^5\sum_{i=0}^{m-1} 2^{5i}(y_1 + 2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)}{2^{5(m-1)+5}}\right)^2\end{array}}$$

$$= \sqrt{\begin{array}{c}\left(\dfrac{(1-2^5)x_0 + (2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i})(x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)}{2^{5m+5}}\right)^2 + \\ \left(\dfrac{(1-2^5)y_0 + (2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i})(y_1 + 2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)}{2^{5m+5}}\right)^2\end{array}}$$

Karena sudah dibuktikan benar dengan induksi matematika saat pemilihan titik sudut A1 bahwa $\left(2^{5m} - 31\sum_{i=0}^{m-1} 2^{5i}\right) = 1$ untuk $m \in N > 1$, maka didapatkan rumus umum jarak antara dua titik seperti berikut

$$\left|\overline{T_{5(m-1)+5}T_{5m+5}}\right| = \sqrt{\begin{matrix}\left(\frac{\left(1-2^5\right)x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)}{2^{5m+5}}\right)^2 + \\ \left(\frac{\left(1-2^5\right)y_0 + (y_1 + 2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)}{2^{5m+5}}\right)^2\end{matrix}}$$

Selanjutnya akan dibuktikan $\left|\overline{T_{5(m-1)+5}T_{5m+5}}\right|$ untuk konvergen dengan mencari nilai $\lim_{m\to\infty}\left|\overline{T_{5(m-1)+5}T_{5m+5}}\right|$, maka berdasarkan Teorema 2.1 didapatkan seperti di bawah ini.

$$\lim_{m\to\infty}\sqrt{\begin{matrix}\left(\frac{\left(1-2^5\right)x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)}{2^{5m+5}}\right)^2 + \\ \left(\frac{\left(1-2^5\right)y_0 + (y_1 + 2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)}{2^{5m+5}}\right)^2\end{matrix}}$$

$$= \lim_{m\to\infty}\frac{1}{\sqrt{(2^{5m+5})^2}}\sqrt{\begin{matrix}\left(\left(1-2^5\right)x_0 + (x_1 + 2x_2 + 2^2x_3 + 2^3x_4 + 2^4x_5)\right)^2 + \\ \left(\left(1-2^5\right)y_0 + (y_1 + 2y_2 + 2^2y_3 + 2^3y_4 + 2^4y_5)\right)^2\end{matrix}}$$

Akan dilakukan pembuktian bahwa $\lim_{m\to\infty}\frac{1}{\sqrt{(2^{5m+5})^2}}$ konvergen berdasarkan Teorema 2.2. Dalam hal ini

$$\frac{1}{\sqrt{(2^{5m+5})^2}} = \frac{1}{2^{5m+5}} = \frac{1}{2^{5m}}\cdot\frac{1}{2^5}$$

sehingga pembuktiannya seperti berikut.

$$\lim_{m\to\infty}\frac{1}{2^{5m}} = \left(\lim_{m\to\infty}\left(\frac{1}{2}\right)^m\right)^5 = (0)^5 = 0.$$

Karena hasil dari $\lim_{m\to\infty}\left|\overline{T_{5(m-1)+5}T_{5m+5}}\right|=0$, maka dapat disimpulkan bahwa rumus tersebut konvergen. Sehingga, kumpulan titik-titik bentukan *non-random* *chaos game* pada segilima saat pemilihan titik sudut A_4 sebagai acuan konvergen ke titik koordinat tertentu.

C. Lampiran Script Tampilan Aturan dan Modifikasi Aturan *Chaos Game* pada Segilima

```
function varargout = chaos_segilima(varargin)
% CHAOS_SEGILIMA MATLAB code for chaos_segilima.fig
%      CHAOS_SEGILIMA, by itself, creates a new CHAOS_SEGILIMA or
raises the existing
%      singleton*.
%
%      H = CHAOS_SEGILIMA returns the handle to a new
CHAOS_SEGILIMA or the handle to
%      the existing singleton*.
%
%      CHAOS_SEGILIMA('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in CHAOS_SEGILIMA.M with the given
input arguments.
%
%      CHAOS_SEGILIMA('Property','Value',...) creates a new
CHAOS_SEGILIMA or raises the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before chaos_segilima_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to chaos_segilima_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
chaos_segilima

% Last Modified by GUIDE v2.5 27-Nov-2018 22:49:14

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @chaos_segilima_OpeningFcn,
...
                   'gui_OutputFcn',  @chaos_segilima_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 chaos_segilima is made visible.
function chaos_segilima_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 chaos_segilima (see
VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes chaos_segilima wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = chaos_segilima_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 radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.radiobutton1,'value',1);
```

```
set(handles.radiobutton2,'value',0);
set(handles.radiobutton3,'value',0);
set(handles.radiobutton4,'value',0);
set(handles.radiobutton5,'value',0);
set(handles.uitable3,'data',[],'columnname','')
% Hint: get(hObject,'Value') returns toggle state of radiobutton1


% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.radiobutton1,'value',0);
set(handles.radiobutton2,'value',1);
set(handles.radiobutton3,'value',0);
set(handles.radiobutton4,'value',0);
set(handles.radiobutton5,'value',0);
prompt={'Non Random :'};
name='Non Random';
numlines=1;
answer=inputdlg(prompt,name,numlines);
a=char(answer);
p=0;
for i=1:length(a)
    if double(a(i))~=32
        p=p+1;
        m(p)=str2double(a(i));
    end
end
set(handles.uipanel1,'Userdata',m);
% Hint: get(hObject,'Value') returns toggle state of radiobutton2


% --- 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)
set(handles.pushbutton1,'enable','off')
for j=1:6
    if j>5
        uiwait(warndlg('Tentukan Titik Acak Awal','Titik
Acak','modal'));
    end
    [x,y]=ginput(1);
    TS(j,:)=[round(x) round(y)];
    [m n]=size(TS);
    hold on

     if j<=5

plot(TS(j,1),TS(j,2),'o','markersize',10,'markerfacecolor','y');
```

```
        text(TS(j,1),TS(j,2)+0.2,['\color{black}A' num2str(j-
1)],'fontsize',11,'fontweight',...

'bold','fontname','cambria','HorizontalAlignment','center','Vertic
alAlignment','baseline')
     else
        for z=1:5
            if z<5
                plot([TS(z,1) TS(z+1,1)], [TS(z,2) TS(z+1,2)],'r-
','linewidth',2);
            else
                plot([TS(z,1) TS(1,1)], [TS(z,2) TS(1,2)],'r-
','linewidth',2);
            end

        end
         plot(x,y,'o','markersize',10,'markerfacecolor','c');
     end
    set(handles.axes1,'XLim',[0 10],'YLim',[0
10],'xtick',[0:10],'ytick',[0:10]);

end
set(handles.figure1,'Userdata',TS);

% --- 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)
clc
ttk=get(handles.figure1,'userdata');
[m n]=size(ttk);
x=ttk(1:m-1,1);
y=ttk(1:m-1,2);
acak=[ttk(m,1) ttk(m,2)];
iter=str2num(get(handles.edit1,'string'));
wrn='rgbkmckmc';
nr=get(handles.uipanel1,'userdata');
pilih1=get(handles.radiobutton1,'value');
pilih2=get(handles.radiobutton2,'value');
pilih3=get(handles.radiobutton3,'value');
pilih4=get(handles.radiobutton4,'value');
pilih5=get(handles.radiobutton5,'value');
if pilih1==1
    [x1 y1 terpilih warna]=chaos_random(x,y,acak,wrn,iter);
elseif pilih2==1
    [x1 y1 terpilih warna jarak]=chaos_nr(x,y,nr,acak,wrn,iter);
elseif pilih3==1
    [x1 y1 terpilih warna]=chaos_mix(x,y,nr,acak,wrn,iter);
elseif pilih4==1
    [x1 y1 terpilih warna]=chaos_random2(x,y,acak,wrn,iter);
elseif pilih5==1
    [x1 y1 terpilih warna]=chaos_random3(x,y,acak,wrn,iter);
end
set(handles.axes1,'NextPlot','replace');
```

```
set(handles.figure1,'CurrentAxes',handles.axes1);
plot(ttk(1:m-1,1),ttk(1:m-
1,2),'o','markersize',10,'markerfacecolor','y')
set(handles.axes1,'NextPlot','add');
plot(ttk(m,1),ttk(m,2),'o','markersize',10,'markerfacecolor','c');
rownames(1)={['T1']};
pjg=length(x1);
wrna={'      Merah','      Hijau','      Biru','      Hitam','
Magenta','      Cyan',' Magenta','      Cyan'};
urutterp=sort(terpilih);
frekterp=frekuensi(urutterp,m-1)*100;

for i=1:m-1
    a4=sprintf('  %10.6f',ttk(i,1));
    a5=sprintf('  %10.6f',ttk(i,2));
    a7(i)={[sprintf('  %6.2f',frekterp(i)) ' %']};
    a6(i)=wrna(i);
    matdata(i,:)={a4,a5};
    rname(i)={['A' num2str(i-1)]};
end
set(handles.uitable1,'data',[matdata a6'
a7'],'rowname',rname,'columnname',{'x','y','Warna','Peluang'},'col
umnwidth',{67})
mat=[acak 0; x1' y1' terpilih'];
waktu=tic;
for i=1:pjg+1
    a1=sprintf('  %10.6f',mat(i,1));
    a2=sprintf('  %10.6f',mat(i,2));
    if mat(i,3)==0
        a3(i)={'             -'};
    else
        a3(i)={['           A' num2str(mat(i,3)-1)]};
    end
    matacak(i,:)={a1,a2};
    rownames(i)={['T' num2str(i-1)]};
    if pilih2==1
        if i<=pjg

plot(x1(i),y1(i),'o','markersize',2,'markerfacecolor',warna(i),'ma
rkeredgecolor',warna(i))
        end
    else
        if i<=pjg

plot(x1(i),y1(i),'o','markersize',1,'markerfacecolor',warna(i),'ma
rkeredgecolor',warna(i))
        end
    end
end
waktu=toc(waktu);
if pilih2==1
    for z=1:length(jarak)
    Jtitik(z)={['   |T' num2str(z) ',T' num2str(z+5) '|']};
    Terpilih(z)={['        A' num2str(terpilih(z)-1)]};
    jrk(z)={sprintf('  %10.6f',jarak(z))};
end
```

```
set(handles.uitable3,'data',[Jtitik' jrk'
Terpilih'],'rowname','','columnname',{'Titik','Jarak','Titik
Sudut'},...
    'columnwidth',{70})
end
set(handles.uitable2,'data',[matacak
a3'],'rowname',rownames,'columnname',{'x','y','Terpilih'},...
    'columnwidth',{70})
set(handles.axes1,'xlim',[0 10],'ylim',[0 10],'Xtick',[0:10],...
    'Ytick',[0:10],'xgrid','on','ygrid','on','box','on');
xlabel('x')
ylabel('y')
for i=1:m-1
    text(ttk(i,1),ttk(i,2)+0.2,['\color{black}A' num2str(i-
1)],'fontsize',11,'fontweight',...

'bold','fontname','cambria','HorizontalAlignment','center','Vertic
alAlignment','baseline')
end
for z=1:5
    if z<5
        plot([ttk(z,1) ttk(z+1,1)], [ttk(z,2) ttk(z+1,2)],'r-
','linewidth',2);
    else
        plot([ttk(z,1) ttk(1,1)], [ttk(z,2) ttk(1,2)],'r-
','linewidth',2);
    end

end
plot(x,y,'o','markersize',10,'markerfacecolor','y');



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


% --- Executes on button press in radiobutton3.
function radiobutton3_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.radiobutton1,'value',0);
set(handles.radiobutton2,'value',0);
set(handles.radiobutton3,'value',1);
set(handles.radiobutton4,'value',0);
set(handles.radiobutton5,'value',0);
set(handles.uitable3,'data',[],'columnname','')
prompt={'Mix Non Random :'};
name='Non Random';
numlines=1;
answer=inputdlg(prompt,name,numlines);
a=char(answer);
p=0;
for i=1:length(a)
    if double(a(i))~=32
        p=p+1;
        m(p)=str2double(a(i));
    end
end
set(handles.uipanel1,'Userdata',m);

% Hint: get(hObject,'Value') returns toggle state of radiobutton3


% --- Executes on button press in radiobutton4.
function radiobutton4_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.radiobutton1,'value',0);
set(handles.radiobutton2,'value',0);
set(handles.radiobutton3,'value',0);
set(handles.radiobutton4,'value',1);
set(handles.radiobutton5,'value',0);
set(handles.uitable3,'data',[],'columnname','')
% Hint: get(hObject,'Value') returns toggle state of radiobutton4


% --- Executes on button press in radiobutton5.
function radiobutton5_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
set(handles.radiobutton1,'value',0);
set(handles.radiobutton2,'value',0);
set(handles.radiobutton3,'value',0);
set(handles.radiobutton4,'value',0);
set(handles.radiobutton5,'value',1);
set(handles.uitable3,'data',[],'columnname','')
    % Hint: get(hObject,'Value') returns toggle state of radiobutton5
```

D. Lampiran Script Aturan *Chaos Game* secara *Random*

```
function [x1 y1 terpilih warna]=chaos_random(x,y,acak,wrn,iter)
n=length(x);
for i=1:iter
    titik=randi(n);
    terpilih(i)=titik;
    warna(i)=wrn(titik);
    x1(i)=(acak(1)+x(titik))/2;
    y1(i)=(acak(2)+y(titik))/2;
    acak=[x1(i) y1(i)];
    end
```

E. Lampiran Script Modifikasi Aturan *Chaos Game* secara *Random*

1. Titik Terakhir Tidak boleh Dipilih Lagi.

```
function [x1 y1 terpilih warna]=chaos_random2(x,y,acak,wrn,iter)
sblm=0;
n=length(x);
for i=1:iter
    titik=randi(n);
    while titik==sblm
        titik=randi(n);
    end
    sblm=titik;
    terpilih(i)=sblm;
    warna(i)=wrn(sblm);
    x1(i)=(acak(1)+x(titik))/2;
    y1(i)=(acak(2)+y(titik))/2;
    acak=[x1(i) y1(i)];
end
```

2. Dua Titik Terakhir Tidak boleh Dipilih Lagi.

```
function [x1 y1 terpilih warna]=chaos_random3(x,y,acak,wrn,iter)
sblm=0;
n=length(x);
for i=1:iter
    titik=randi(n);
    if i<3
```

```
        while titik==sblm
            titik=randi(n);
        end
    else
        while titik==sblm || titik==terpilih(i-2)
            titik=randi(n);
        end
    end
    sblm=titik;
    terpilih(i)=sblm;
    warna(i)=wrn(sblm);
    x1(i)=(acak(1)+x(titik))/2;
    y1(i)=(acak(2)+y(titik))/2;
    acak=[x1(i) y1(i)];
   end
```

F. Lampiran Script Modifikasi Aturan *Chaos Game* secara *Non-Random*

```
function [x1 y1 terpilih warna
jarak]=chaos_nr(x,y,titik,acak,wrn,iter)
n=length(x);
p=1;
for i=1:iter
    terpilih(i)=titik(p);
    x1(i)=(acak(1)+x(terpilih(i)))/2;
    y1(i)=(acak(2)+y(terpilih(i)))/2;
    acak=[x1(i) y1(i)];
    warna(i)=wrn(terpilih(i));
    if p<n
        p=p+1;
    else
        p=1;
    end
end
for z=1:iter-5
    jarak(z)=sqrt((x1(z+5)-x1(z))^2+(y1(z+5)-y1(z))^2);
end
```

G. Lampiran Script Modifikasi Gabungan Aturan *Chaos Game* secara *Random* dan *Non-Random*

```
function [x1 y1 terpilih warna]=chaos_mix(x,y,ttk,acak,wrn,iter)
n=length(x);
p=1;
for i=1:iter
    if p<=length(ttk)
        terpilih(i)=ttk(p);
        x1(i)=(acak(1)+x(terpilih(i)))/2;
        y1(i)=(acak(2)+y(terpilih(i)))/2;
        acak=[x1(i) y1(i)];
        warna(i)=wrn(terpilih(i));
```

```
        sblm=ttk(p);
    else
        titik=randi(n);
        while titik==sblm
            titik=randi(n);
        end
        if p==n
            titik=randi(n);
            while titik==ttk(1) || titik==sblm
                titik=randi(n);
            end
        end
        sblm=titik;
        terpilih(i)=sblm;
        warna(i)=wrn(sblm);
        x1(i)=(acak(1)+x(titik))/2;
        y1(i)=(acak(2)+y(titik))/2;
        acak=[x1(i) y1(i)];
    end
    if p<n
        p=p+1;
    else
        p=1;
    end
    end
```

H. Lampiran Script Menentukan Peluang Terpilihnya Setiap Titik Sudut sebagai Titik Acuan

```
function P=frekuensi(D,m)
n=length(D);
k=1;b=0;
defDat=1:m;
P=zeros(1,m);
for i=2:n
    if D(i-1)==D(i)
        k=k+1;
    else
        b=b+1;
        data(defDat(D(i-1)))=D(i-1);
        P(defDat(D(i-1)))=k/n;
        frek(defDat(D(i-1)))=k;
        k=1;
    end
end
b=b+1;
data(defDat(D(i)))=D(n);
frek(defDat(D(i)))=k;
    P(defDat(D(i)))=k/n;
```