MODIFIKASI *CHAOS GAME* DENGAN TITIK ACUAN MEMBENTUK POLIGON *NON-CONVEX*

SKRIPSI

Oleh

Nadiya Annisa Dikara
NIM 151810101026

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2019

MODIFIKASI *CHAOS GAME* DENGAN TITIK ACUAN MEMBENTUK POLIGON *NON-CONVEX*

SKRIPSI

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

Oleh

Nadiya Annisa Dikara
NIM 151810101026

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 Lusia Prihatiwi Hermajanti Jasiani dan Ayahanda Bagio Sulistiono tercinta, yang telah membesarkan, mendidik, mendoakan, memotivasi dengan penuh kasih sayang dan pengorbanan selama ini;
2. Kakak Andaka Mahardika dan Adik Talita Nabeela Azka tersayang, yang telah memberikan semangat, doa, dan dukungan kepada penulis;
3. Guru-guru TK BUDI UTOMO, SDN KEPATIHAN 1, SMPN 2 Jember, SMAN 2 Jember, 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

Dan milik Allah-lah apa yang ada di langit dan apa yang ada di bumi. Cukuplah Allah sebagai Pemeliharanya.

(Q.S An-Nisa: 132)

Kementerian Agama Republik Indonesia. 2016. *Al-Qur'an Edisi Terjemah & Penjelasan Ayat tentang Wanita Yasmin*. Solo: PT Tiga Serangkai Pustaka Mandiri.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Nadiya Annisa Dikara

NIM : 151810101026

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul: ”Modifikasi *Chaos Game* dengan Titik Acuan Membentuk Poligon *Non-Convex*” 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,

Nadiya Annisa Dikara

NIM 151810101026

SKRIPSI

MODIFIKASI *CHAOS GAME* DENGAN TITIK ACUAN MEMBENTUK POLIGON *NON-CONVEX*

Oleh

Nadiya Annisa Dikara
Nim 151810101026

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 *Chaos Game* dengan Titik Acuan Membentuk Poligon *Non-Convex*" 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 *Chaos Game* dengan Titik Acuan Membentuk Poligon *Non-Convex*; Nadiya Annisa Dikara; 151810101026; 2019; 44 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Fraktal adalah objek geometris yang didapatkan melalui proses iteratif dan mempunyai sifat *self-similarity* (keserupaan diri). Teknik yang biasa digunakan untuk membuat fraktal diantaranya adalah metode *Iterated Function Systems* (IFS), metode *Iterated Complex Polynomial*, metode L-*system*, dan metode *Strange Attractor*. Terdapat dua metode untuk membangun fraktal IFS, yaitu *Supercopier* dan *Chaos Game*. *Chaos game* merupakan bentuk permainan menggambar titik dalam suatu bidang dengan aturan tertentu yang diulang-ulang secara iteratif. Banyak penelitian sebelumnya mengembangkan aturan *chaos game* pada segitiga, segi empat, dan segi enam. Pada penelitian ini dikaji pengembangan aturan *chaos game* yang diterapkan dengan penentuan titik acuan, dimana titik yang ditentukan membentuk poligon *non-convex* dengan minimal empat titik.

Simulasi program dilakukan dengan memvariasikan jumlah titik yang digunakan pada aturan pemilihan titik acuan secara *random* dan membentuk poligon *non-convex*. Bentuk dasar titik acuan sebagai contoh disajikan sebanyak enam macam bentukan poligon *non-convex*. Keenam titik acuan yang berbentuk poligon *non-convex* tersebut dilakukan perubahan titik menjadi beberapa bentuk poligon baru. Setiap bentuk poligon *non-convex* tersebut memiliki perbedaan yaitu bentuk luar dari titik acuan yang dibuat membentuk sebuah poligon tertentu. Iterasi yang digunakan untuk simulasi adalah 30.000 iterasi. Banyaknya iterasi berpengaruh pada pola titik-titik fraktal *chaos game* yang dihasilkan. Semakin banyak iterasi maka pola titik-titik fraktal *chaos game* semakin jelas terlihat.

Percobaan pertama hingga keenam menghasilkan permasalahan yang hampir sama, perbedaannya pada jumlah titik acuan dan bentuk yang dibuat. Perubahan letak titik-titik pengacau ini lah yang dapat menjelaskan bahwa setiap

bentuk titik acuan yang dibuat yakni *convex* atau *non-convex* sebenarnya selalu membentuk fraktal. Kumpulan titik-titik bentukan poligon *non-convex* pada *chaos game* selalu tertarik pada titik acuannya masing-masing. Pola fraktal yang dihasilkan pada poligon *non-convex* mirip atau ada kaitan dengan poligon *convex* yang dibangun oleh titik terluarnya.

PRAKATA

Puji syukur penulis kepada Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Modifikasi *Chaos Game* dengan Titik Acuan Membentuk Poligon *Non-Convex*". 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. Dian Anggraeni, S.Si., M.Si., selaku Dosen Wali yang memberikan bimbingan dan dukungan selama 3,5 tahun menjalani perkuliahan;
4. Drs. Sujito, PhD selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
5. Seluruh dosen dan staf karyawan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
6. Ibu Lusia Prihatiwi Hermajanti Jasiani, Bapak Bagio Sulistiono, Kakak Andaka Mahardika, dan Adik Talita Nabeela Azka tercinta yang telah memberikan semangat, doa, dan dukungan kepada penulis;
7. Teman-teman satu bidang skripsi fraktal Melati, Mitha, Intan, Rozida, Ingka, Dwi Alfi, Novita, dan Iza yang selalu memberi dukungan dan semangat tanpa henti;

8. Teman-teman seangkatan SIGMA'15 yang telah banyak membantu penulis selama studi;
9. HIMATIKA "Geokompstat" yang telah membantu penulis dalam berproses;
10. Sahabat-sahabat tersayang Adhika Nandiwardhana Dianastya, Yuniar Permatasari, Aisah Intan Purnama, Princess (Annisa, Devi, Rahma, Adinda), dan IS (Eriska, Rivi, Intan, Mitha, Melati, Rika) yang senantiasa memberikan dukungan dan menjadi keluarga kedua penulis;
11. Kakak-kakak tingkat dan adik-adik tingkat yang selalu memotivasi penulis agar terselesaikannya skripsi ini;
12. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang memberikan dorongan bagi penulis selama studi sampai penulisan skripsi selesai.

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

Jember, Januari 2019

Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL i
HALAMAN PERSEMBAHAN ii
HALAMAN MOTTO iii
HALAMAN PERNYATAAN iv
HALAMAN PEMBIMBINGAN v
HALAMAN PENGESAHAN vi
RINGKASAN vii
PRAKATA ix
DAFTAR ISI xi
DAFTAR GAMBAR xiii
BAB I. PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Tujuan Penelitian 3
1.4 Manfaat Penelitian 3
BAB 2. TINJAUAN PUSTAKA 4
2.1 Fraktal 4
2.2 Chaos Game 5
2.3 Koordinat Titik pada Segmen Garis di Bidang 6
2.4 Poligon 7
2.5 Penelitian Sebelumnya 8
BAB 3. METODE PENELITIAN 12
3.1 Kajian Pustaka *Chaos Game* dan Modifikasinya 13
3.2 Kajian *Chaos Game* pada Aturan Pemilihan Titik Acuan Berbentuk Poligon *Convex* 13
3.3 Modifikasi *Chaos Game* pada Aturan Pemilihan Titik Acuan Berbentuk Poligon *Non-Convex* 13
3.4 Simulasi Program 13

3.5 Analisis Hasil 14
BAB 4. HASIL DAN PEMBAHASAN 15
4.1 Modifikasi *Chaos Game* pada Aturan Pemilihan Titik Acuan Berbentuk Poligon *Non-Convex* 15
4.2 Simulasi Program 19
4.2.1 Fasilitas yang Tersedia pada GUI 19
4.2.2 Data Awal yang Harus Ditentukan *User* 21
4.2.3 Tahapan *Chaos Game* 27
4.3 Pembahasan 33
BAB 5. KESIMPULAN DAN SARAN 36
5.1 Kesimpulan 36
5.2 Saran 36
DAFTAR PUSTAKA 37
LAMPIRAN 39

DAFTAR GAMBAR

Halaman

2.1 Mekanisme *chaos game* pada segitiga yang menghasilkan segitiga Sierpinski 6
2.2 Kumpulan titik-titik tengah pada *chaos game* (a) iterasi ke-400 dan (b) ke-30.000 6
2.3 Titik *R* berada di antara titik *P* dan *Q* 7
2.4 Poligon konveks 7
2.5 Poligon tidak konveks 8
2.6 Sierpinski *hexagon* 9
2.7 Fraktal *chaos game* dengan empat titik acuan 11
2.8 Fraktal *chaos game* dengan lima titik acuan 11
3.1 Skema metode penelitian 12
4.1 Poligon *non-convex* segi empat 15
4.2 Poligon *non-convex* segi lima kasus ke-1 16
4.3 Poligon *non-convex* segi lima kasus ke-2 17
4.4 Poligon *non-convex* segi enam kasus ke-1 17
4.5 Poligon *non-convex* segi enam kasus ke-2 18
4.6 Poligon *non-convex* segi enam kasus ke-3 18
4.7 Tampilan menu Bantuan 20
4.8 Tampilan program GUI 21
4.9 Tampilan pada program poligon *non-convex* segi empat 22
4.10 Tampilan pada program poligon *non-convex* segi lima kasus ke-1..23
4.11 Tampilan pada program poligon *non-convex* segi lima kasus ke-2..24
4.12 Tampilan pada program poligon *non-convex* segi enam kasus ke-1 25
4.13 Tampilan pada program poligon *non-convex* segi enam kasus ke-2 26
4.14 Tampilan pada program poligon *non-convex* segi enam kasus ke-3 27
4.15 Hasil *chaos game* yang didapatkan dari perubahan bentuk poligon segi empat *non-convex* dan *convex* 28

4.16 Hasil *chaos game* yang didapatkan dari perubahan bentuk poligon segi lima *non-convex* dan *convex* kasus ke-1 29
4.17 Hasil *chaos game* yang didapatkan dari perubahan bentuk poligon segi lima *non-convex* dan *convex* kasus ke-2 30
4.18 Hasil *chaos game* yang didapatkan dari perubahan bentuk poligon segi enam *non-convex* dan *convex* kasus ke-1 31
4.19 Hasil *chaos game* yang didapatkan dari perubahan bentuk poligon segi enam *non-convex* dan *convex* kasus ke-2 32
4.20 Hasil *chaos game* yang didapatkan dari perubahan bentuk poligon segi enam *non-convex* dan *convex* kasus ke-3 33

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Geometri fraktal atau lebih dikenal dengan fraktal merupakan bagian dari ilmu matematika yang telah berkembang pesat dengan adanya teknologi canggih seperti komputer pada saat ini. Pengembangan fraktal dengan menggunakan komputer dapat menghasilkan sebuah karya yang beraneka ragam dan indah bentuknya. Fraktal merupakan sebuah kelas bentuk geometri kompleks yang umumnya mempunyai dimensi pecahan, sebuah konsep yang pertama kali diperkenalkan oleh matematikawan Felix Hausdorff pada tahun 1918. Bentuk-bentuk fraktal bersifat menyerupai diri sendiri (*self-similar*), artinya setiap bagian kecil dalam sebuah fraktal dapat dipandang sebagai replikasi skala kecil dari bentuk keseluruhan. Fraktal berbeda dengan gambar-gambar klasik sederhana atau geometri *Euclid* seperti bujur sangkar, lingkaran, bola, dan sebagainya. Fraktal dapat digunakan untuk menjelaskan banyak objek yang bentuknya tidak beraturan atau fenomena alam yang secara spasial tidak seragam, seperti bentuk pantai atau lereng gunung. Terdapat beberapa contoh objek fraktal yaitu: segitiga Sierpinski, *dragon curve*, Koch *snowflake*, kurva Hilbert, himpunan Cantor, himpunan Mandelbort dan Julia (Mandelbort, 1983).

Teknik yang biasa digunakan untuk membuat fraktal diantaranya adalah metode *Iterated Function Systems* (IFS), metode *Iterated Complex Polynomial*, metode L-*system*, dan metode *Strange Attractor*. *Iterated Function Systems* (IFS) pertama kali dipopulerkan oleh Michael Barnsley dalam bukunya *Fractal Everywhere*. Fraktal IFS terdiri dari gabungan beberapa tiruan dirinya sendiri, yang kemudian ditransformasikan ke dalam suatu fungsi. Terdapat dua metode untuk membangun fraktal IFS, yaitu *Supercopier* dan *Chaos Game* (Riyadi, 2007).

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. *Chaos* terjadi di dalam suatu sistem dinamis, yaitu jika dua buah titik acak yang mendekati titik pemicu (*starting point*) terbagi-bagi secara eksponensial, maka hasil akhirnya menjadi tidak dapat diprediksi. *Chaos game* sendiri merupakan salah satu bentuk penerapan dari teori *chaos* (Riyadi, 2007). *Chaos game* dipresentasikan oleh Barnsley pada tahun 1988. Teori ini tentang Teorema Shannon yang dipresentasikan dengan menggunakan mekanisme *Random Walk* dan dapat diproduksi dengan bantuan fraktal poligonal (Jampour, dkk., 2010).

Menurut Armana (2016), jarak titik baru yang dibentuk oleh metode *chaos game* yaitu titik tengah dari titik terakhir dan titik sudut yang dapat membentuk segitiga Sierpinski. Apabila jarak tersebut diubah menjadi 1/3 dari titik sudut segitiga, maka yang terbentuk adalah sebuah segi lima di bagian dalam segitiga. Pola yang berbeda akan didapat jika merubah bidang yang diteliti namun tetap menggunakan aturan awal *chaos game*. Segitiga Sierpinski merupakan segitiga yang terdiri atas segitiga-segitiga lain yang berbentuk sama tetapi dengan ukuran setengahnya. Apabila bagian kecil dari segitiga tersebut diperbesar, maka akan tampak seperti bagian lain yang lebih besar (Purnomo, 2014). Dalam kaitannya dengan *chaos game*, Zohuri (2015) menyatakan bahwa bila titik acuan awal untuk *chaos game* terletak pada segitiga Sierpinski, maka titik yang ditandai berikutnya juga akan berada pada segitiga Sierpinski. *Chaos game* pada empat titik sudut atau pada persegi menghasilkan bentuk yang berbeda. Hasilnya bukanlah persegi di dalam persegi, tapi sebuah persegi yang terisi dengan banyak titik yang terbentuk secara *random* (Jeffrey dan Joel, 1990). Modifikasi aturan produksi *chaos game* untuk menghasilkan Sierpinski *hexagon* telah dilakukan oleh Devaney (2003). Penelitian ini dilakukan dengan menentukan enam titik awal yang membentuk *hexagon*, kemudian dilakukan aturan *chaos game* pada pembentukan segitiga Sierpinski dengan mengubah jarak antara titik sudut dengan titik awal menjadi sepertiga dari jarak semula. Menurut Yunaning (2018), aturan *non-random chaos game* pada segitiga menghasilkan kumpulan titik yang konvergen di titik koordinat tertentu pada titik sudut yang sama. Kekonvergenan titik ditunjukkan dengan melakukan perhitungan numerik dan perhitungan analitik pada titik-titik yang dihasilkan dari aturan *non-random chaos game*. Pada penelitian Yunaning dikatakan bahwa objek yang dihasilkan dari penggunaan aturan *non-random chaos game* tidak membentuk fraktal.

Berdasarkan uraian di atas, penulis tertarik untuk melakukan penelitian lebih lanjut mengenai aturan *chaos game*. Penulisan ini akan membahas tentang modifikasi *chaos game* dengan titik acuan berupa titik sudut *non-convex* serta visualisasi dengan menggunakan program MATLAB.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, rumusan masalah yang dibahas dalam tugas akhir ini yaitu apakah *chaos game* yang memanfaatkan titik acuan berupa titik sudut poligon segi n $(n \geq 4)$ *non-convex* secara visual dapat membentuk fraktal?

1.3 Tujuan

Berdasarkan rumusan masalah di atas, tujuan dari penulisan tugas akhir ini yaitu untuk mengetahui bentuk fraktal *chaos game* yang memanfaatkan titik acuan berupa titik sudut poligon segi n $(n \geq 4)$ *non-convex* secara visual.

1.4 Manfaat

Manfaat dari penulisan ini adalah memberikan informasi tentang *chaos game* dan mendapatkan beberapa pola geometri yang membentuk fraktal sehingga dapat bermanfaat pada penelitian–penelitian yang berkaitan nantinya.

BAB 2. TINJAUAN PUSTAKA

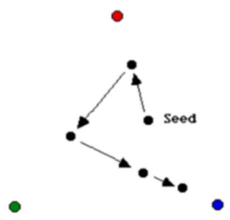
2.1 Fraktal

Fraktal adalah objek geometris yang didapatkan melalui proses iteratif dan mempunyai sifat *self-similarity* (keserupaan diri). Istilah fraktal pertama kali dituliskan oleh Benoit Mandelbrot pada tahun 1977. Sebelumnya, nama yang digunakan untuk menyebut struktur geometri ini adalah "*monster curve*". Fraktal berasal dari bahasa Latin "*fractus*" yang berarti dasar yang tidak beraturan seperti sebuah batu yang pecah. Mandelbrot menggunakan istilah ini pertama kali untuk menjelaskan pola berulang yang ditemukan pada berbagai struktur yang berbeda dalam pengamatannya (Mandelbrot, 1983).

Komponen objek fraktal mirip dengan seluruh objek itu dan diproduksi dengan prosedur berulang. Jika menghitung dimensi benda-benda ini tidak seperti objek garis yang hanya memiliki satu dimensi atau pelat yang dimensinya dua. Dimensi objek fraktal yaitu berupa pecahan (Kigami, 2001). Salah satu objek fraktal adalah segitiga Sierpinski. Segitiga Sierpinski merupakan fraktal linier yang mempunyai bentuk geometris yang identik hingga pada iterasi tak hingga. Salah satu cara untuk membangkitkan segitiga Sierpinski yaitu dengan menentukan terlebih dahulu segitiga sama sisi yang telah diberi warna tertentu, kemudian titik tengah masing-masing sisinya dihubungkan sehingga diperoleh segitiga baru yang berukuran setengahnya serta terletak di tengah segitiga awal. Metode untuk membangkitkan segitiga Sierpinski adalah *chaos game*. Analisis *chaos game* memanfaatkan koordinat segitiga Sierpinski pada level 3. Level segitiga menunjukkan banyaknya ukuran segitiga yang ada di dalam segitiga. Dengan menggunakan bantuan koordinat titik tengah pada masing-masing iterasi, dapat dibuktikan bahwa apabila diambil titik sembarang yang terletak pada, maupun di luar segitiga Sierpinski selalu ada titik serupa ke arah 3 titik sudut. Setiap percobaan *chaos game* dengan aturan *random* menghasilkan segitiga Sierpinski, meskipun titik awal tidak selalu pada segitiga, yaitu di dalam ataupun di luar segitiga. Hal ini dikarenakan ketika titik awal di dalam ataupun di luar segitiga, saat sudah mencapai iterasi ke-n, titik bentukan *chaos game* selanjutnya mendekati letak segitiga Sierpinski, sehingga titik awal pada, di dalam, atau pun di luar segitiga akan tetap bisa membangun segitiga Sierpinski dengan *chaos game* (Armana, 2016).

2.2 *Chaos Game*

Chaos game merupakan bentuk permainan menggambar suatu titik dalam segitiga sama sisi dengan aturan tertentu yang dilakukan berulang-ulang secara iteratif. Titik yang digambar tersebut merupakan titik tengah dari jarak titik awal atau sebelumnya dengan salah satu titik sudut segitiga yang diambil secara acak. Jika penggambaran titik tengah tersebut dilakukan pada jumlah iterasi kecil, maka kumpulan titik tersebut terkesan berbentuk *chaos* (kacau). Namun demikian, jika itu dilakukan pada jumlah ribuan iterasi, maka kumpulan titik-titik tengah tersebut akan mendekati bentuk segitiga Sierpinski (Purnomo, dkk., 2016). Untuk melakukan mekanisme *chaos game* pada segitiga, prosedurnya yaitu titik yang teracak adalah yang terpilih. Titik tersebut merupakan titik awal. Langkah kedua merupakan angka yang teracak dalam lingkup 1 sampai 3 dipilih. Didapat tiga titik yaitu titik A, titik B, dan titik C. Jika 1 dipilih itu berarti puncak A, jika 2 dipilih artinya puncak B, dan akhirnya jika 3 dipilih itu berarti puncak C dalam segitiga. Kemudian, dari titik sekarang melangkah ke tengah jalan menuju terpilih puncak dan dibuat titik baru. Proses dilanjutkan dengan membuat suatu kebetulan pilihan dan diulangi proses yang sama untuk beberapa kali (contoh 50.000 kali). Cara ini membentuk gambar *Sierpinski triangle*. Pada Gambar 2.1 empat langkah mekanisme *chaos game* ditampilkan. Pada iterasi yang sedikit hanya akan terdapat kumpulan titik-titik yang tidak beraturan, namun jika iterasi yang dilakukan semakin banyak, maka akan terlihat bentuk yang menyerupai segitiga Sierpinski (Gambar 2.2).



Gambar 2.1 Mekanisme *chaos game* pada segitiga yang menghasilkan segitiga Sierpinski
(Sumber: Jampour, dkk., 2010)

(a) (b)

Gambar 2.2 Kumpulan titik-titik tengah pada *chaos game* (a) iterasi ke-400 dan (b) iterasi ke-30.000
(Sumber: Zohuri, 2015)

2.3 Koordinat Titik pada Segmen Garis di Bidang

Menurut Kusno (2010), untuk mendefinisikan koordinat suatu titik di R^2 diperlukan dua garis berarah yang saling tegak lurus. Kedua garis tersebut terdiri atas garis mendatar yaitu sumbu X dan garis tegak yaitu sumbu Y. Letak sebuah titik pada koordinat Cartesius didefinisikan dengan pasangan sumbu X dan sumbu Y yang biasa ditulis dengan (x, y).

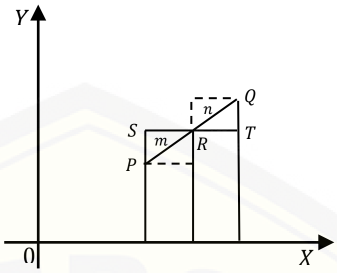
Pada Gambar 2.3 terdapat $\Delta SRP \sim \Delta RQT$ dengan segmen garis $\overline{PQ}$. Segmen garis $\overline{PQ}$ didefinisikan oleh $P(x_1, y_1)$ dan $Q(x_2, y_2)$, dengan $|\overline{PR}| = m$ dan $|\overline{RQ}| = n$. Perbandingan antara m dan n adalah $1 : 1$, jika $R(x, y)$ merupakan titik tengah dari segmen garis $\overline{PQ}$, sehingga koordinat $R(x, y)$ dapat dicari dengan menggunakan persamaan berikut:

$$x = \frac{x_2 + x_1}{2}$$

$$y = \frac{y_2 + y_1}{2}$$

dapat dinyatakan sebagai

$$R\left(\frac{x_2 + x_1}{2}, \frac{y_2 + y_1}{2}\right)$$



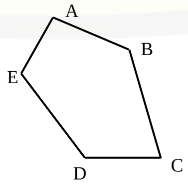
Gambar 2.3 Titik R berada di antara titik P dan Q

2.4 Poligon

Poligon merupakan salah satu istilah dalam geometri. Poligon adalah himpunan titik-titik $P_1, P_2, P_3, \dots, P_{n-1}, P_n$ dengan ruas garis-ruas garis $\overline{P_1P_2}$, $\overline{P_2P_3}$, $\overline{P_3P_4}, \dots, \overline{P_{n-1}P_n}$, $\overline{P_nP_1}$, sedemikian hingga jika dua dari sebarang ruas garis tersebut berpotongan, titik potongnya adalah salah satu dari titik $P_1, P_2, P_3, \dots, P_{n-1}, P_n$ dan tidak ada titik lainnya. Titik $P_1, P_2, P_3, \dots, P_{n-1}, P_n$ disebut titik-titik sudut poligon, sedangkan ruas-ruas garis $\overline{P_1P_2}$, $\overline{P_2P_3}$, $\overline{P_3P_4}, \dots, \overline{P_{n-1}P_n}$, $\overline{P_nP_1}$ disebut sisi poligon. Suatu poligon dinamakan dengan menyebutkan semua titik-titik sudutnya secara berurutan dengan searah jarum jam atau berlawanan arah jarum jam (Kusno, 1995).

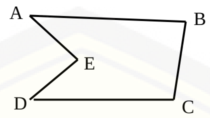
Terdapat beberapa istilah yang berkaitan dengan poligon, diantaranya adalah sebagai berikut:

1. Poligon konveks adalah poligon yang masing-masing sudutnya lebih kecil dari sudut lurus (Gambar 2.4).



Gambar 2.4 Poligon konveks
(Sumber: Kusno, 1995)

2. Poligon tidak konveks (*non-convex*) adalah poligon yang mempunyai minimal satu sudut dalam yang lebih besar dari sudut lurus (Gambar 2.5).

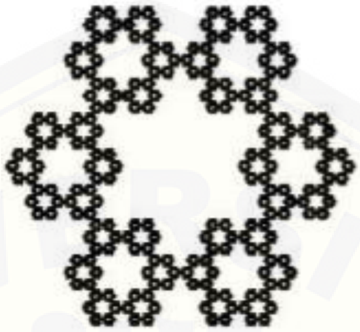


Gambar 2.5 Poligon tidak konveks
(Sumber: Kusno, 1995)

2.5 Penelitian Sebelumnya

Penelitian menggunakan titik acuan sebanyak empat titik (persegi) oleh Jeffrey pada tahun 1990 telah diuji coba dan ditulis dalam makalahnya yang berjudul "*Chaos Game Representation of Gene Structure*". Isi dari makalah Jeffrey yaitu menyajikan metode baru untuk mewakili urutan DNA. Hal ini memungkinkan representasi dan investigasi dalam urutan pola secara visual. Berdasarkan teknik dari dinamika kacau bahwa sebuah metode menghasilkan gambar urutan gen yang ditampilkan dengan pola lokal maupun global. Gambar-gambar memiliki struktur kompleks yang bervariasi tergantung pada urutan. Metode ini disebut *Chaos Game Representation* (CGR). *Chaos game* adalah algoritma yang memungkinkan seseorang untuk menghasilkan gambar struktur fraktal, menggunakan kertas, pensil, atau komputer. Dalam bentuk yang paling sederhana, hasil dari langkah-langkah menggunakan metode *chaos game* biasa disebut dengan "*Sierpinski triangle*". Melihat hasil tersebut timbul sebuah pertanyaan yang mengharuskan dilakukan percobaan dengan titik awal lebih dari tiga. Namun, dengan empat titik awal hasilnya berbeda yaitu tidak kotak dalam kotak seperti yang diharapkan. Faktanya tidak ada pola sama sekali. *Chaos game* pada empat titik menghasilkan sebuah persegi seragam dan secara acak diisi dengan titik-titik.

Penelitian lain oleh Robert L. Devaney pada tahun 2003 yang berjudul "*Fractal Patterns and Chaos Game*" tidak menggunakan bangun segitiga, melainkan segi enam (*hexagon*) yang menjadi bentuk titik acuan awal. Namun, pada aturannya sedikit diubah yaitu dalam menentukan jarak titik acuan dengan sebuah titik sembarang menjadi sepertiga aslinya. Hasil yang didapat setelah melakukan ribuan kali iterasi yaitu berupa bentuk Sierpinski *hexagon*.



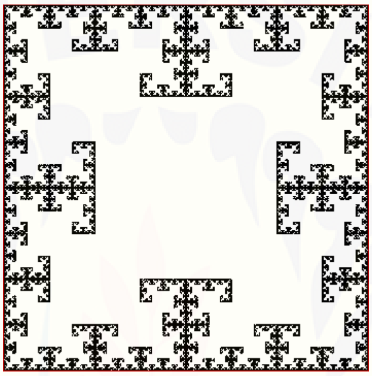
Gambar 2.6 Sierpinski *hexagon*
(Sumber: Devaney, 2003)

Pada Gambar 2.6 terdiri dari enam potong yang sama, masing-masing persis seperti sepertiga ukuran dari segi enam Sierpinski penuh.

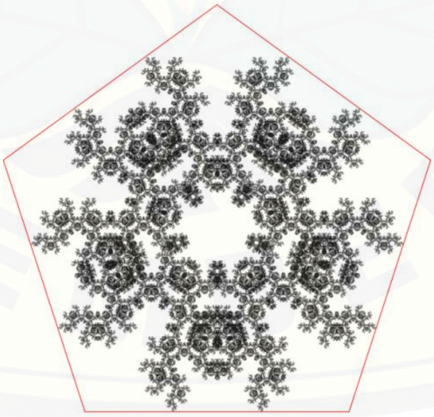
Penelitian oleh Friska Yunaning pada tahun 2018 membahas tentang aturan *non-random chaos game* pada segitiga. Titik-titik bentukan *non-random chaos game* hanya berkelompok di koordinat tertentu pada masing-masing pemilihan titik sudut yang sama. Dengan demikian, objek bentukan *non-random chaos game* tidak dapat dikatakan sebagai fraktal karena objek tersebut tidak membentuk suatu bentuk tertentu dan tidak memiliki salah satu sifat penting fraktal yaitu *self similarity*. Pemilihan titik sudut dipilih secara urut secara terus menerus mengakibatkan tidak ada perulangan pemilihan titik sudut dari iterasi satu ke iterasi selanjutnya. Akibatnya, letak titik-titik bentukan *non-random chaos game* tidak berbeda signifikan sehingga pergerakan titik-titik tersebut hanya berputar secara terus menerus pada lingkup tertentu. Perbedaan urutan pemilihan titik sudut hanya akan merubah arah putaran pergerakan titik-titik bentukan *non-random chaos game*. Objek yang dihasilkan aturan *non-random chaos game* tidak membentuk suatu fraktal. Perhitungan numerik *non-random chaos game* menghasilkan nilai jarak antar dua titik pada pemilihan titik sudut yang sama bernilai semakin kecil dari iterasi rendah ke iterasi tinggi. Artinya, pada iterasi tinggi titik-titik yang dihasilkan *non-random chaos game* akan menuju pada satu titik tertentu. Sedangkan perhitungan analitik aturan *non-random chaos game* menghasilkan nilai jarak antar dua titik pada pemilihan titik sudut T_0, T_1, T_2, konvergen di titik koordinat tertentu. Segitiga baru dengan titik pembentuknya adalah titik terakhir dari kumpulan titik bentukan *non-random chaos game* di masing-masing pemilihan titik sudut T_0, T_1, T_2 tidak sebangun dengan segitiga awal untuk memulai *non-random chaos game*.

Segitiga juga digunakan dalam penelitian mengenai modifikasi aturan *chaos game* dengan memanfaatkan titik berat segitiga oleh Eka Nofiatur Ratna pada tahun 2018. Pada penelitian tersebut dikaji pengembangan aturan *chaos game* yang diterapkan dengan memanfaatkan titik berat segitiga, modifikasi yang dilakukan yaitu dengan aturan pemilihan titik acuan secara *random* dan aturan pemilihan secara *non-random*. Percobaan dengan menambahkan satu titik acuan, yakni satu titik berat atau satu titik sebarang pada segitiga menghasilkan visualisasi objek dengan titik-titik yang teratur, saling mengikuti titik acuan dan berbentuk segitiga pada setiap titik acuan sehingga menyerupai segitiga Sierpinski. Apabila menambahkan lebih dari satu titik acuan pada segitiga juga menghasilkan visualisasi objek dengan titik-titik yang teratur, saling mengikuti titik acuan dan berbentuk segitiga yang berbeda warna pada setiap titik acuan, tetapi tidak menyerupai segitiga Sierpinski. Bentuk segitiga yang dihasilkan pada setiap titik acuan sesuai jumlah titik yang ditambahkan. Titik bentukan *chaos game* dengan aturan pemilihan titik acuan secara random dapat membentuk suatu bentuk tertentu yang menyerupai segitiga Sierpinski serta menyebar ke semua titik acuan dan memiliki salah satu sifat fraktal yaitu *self similarity*. Dengan demikian, objek *random chaos game* dengan memanfaatkan titik berat segitiga dapat dikatakan sebagai fraktal. Objek *non-random chaos game* dengan memanfaatkan titik berat segitiga tidak dapat dikatakan sebagai fraktal karena tidak membentuk suatu bentuk tertentu dan tidak memiliki salah satu sifat fraktal yaitu *self similarity*. Objek *non-random chaos game* dengan memanfaatkan titik berat segitiga memiliki karakteristik, dimana kumpulan titik bentukan *non-random* konvergen ke koordinat tertentu.

Beberapa contoh gambar fraktal dengan aturan *chaos game* seperti pada Gambar 2.7 yaitu sebuah titik di dalam persegi berulang kali melompat setengah dari jarak menuju titik yang dipilih secara acak, tetapi titik yang dipilih saat ini tidak dapat menjadi 2 tempat dari titik yang dipilih sebelumnya. Pada Gambar 2.8 yaitu sebuah titik di dalam *pentagon* berulang kali melompat setengah dari jarak menuju titik yang dipilih secara acak, tetapi titik yang dipilih saat ini tidak dapat sama dengan titik yang dipilih sebelumnya.



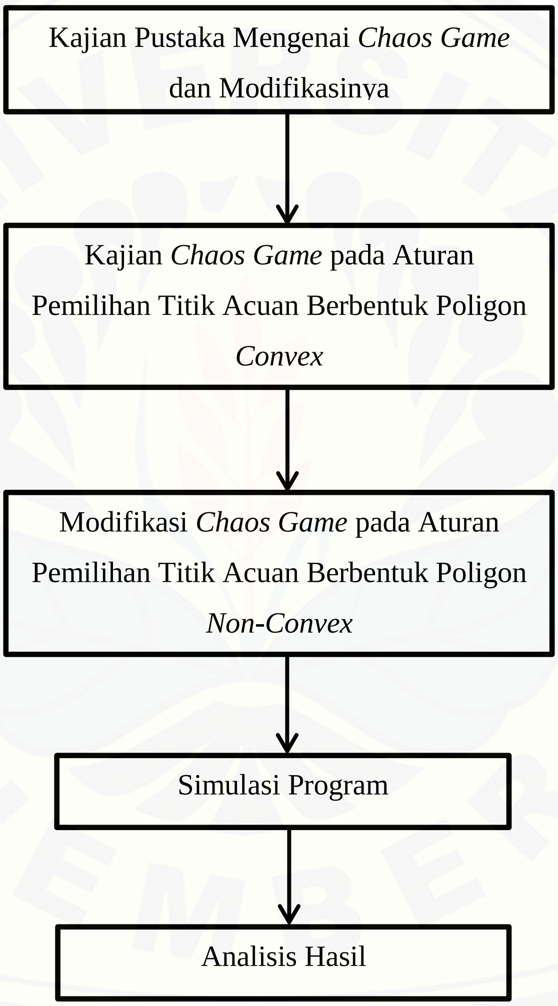
Gambar 2.7 Fraktal *chaos game* dengan empat titik acuan
(Sumber: https://en.wikipedia.org/wiki/Chaos_game, 2018)



Gambar 2.8 Fraktal *chaos game* dengan lima titik acuan
(Sumber: https://en.wikipedia.org/wiki/Chaos_game, 2018)

BAB 3. METODE PENELITIAN

Berdasarkan rumusan masalah, pada bab ini akan dibahas tentang langkah-langkah yang akan digunakan untuk menyelesaikan penelitian ini. Secara skematik, langkah-langkah yang akan dilakukan dalam penelitian ini dapat digambarkan dengan diagram alir pada Gambar 3.1



Gambar 3.1 Skema metode penelitian

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

3.1 Kajian Pustaka *Chaos Game* dan Modifikasinya

Beberapa informasi atau pengetahuan diperlukan untuk dapat menunjang penelitian. Kajian pustaka mengenai aturan *chaos game* serta modifikasi *chaos game* dilakukan dengan pencarian informasi lebih banyak melalui buku, jurnal, dan skripsi.

3.2 Kajian *Chaos Game* pada Aturan Pemilihan Titik Acuan Berbentuk Poligon *Convex*

Beberapa informasi diperlukan untuk dapat menunjang tugas akhir ini. Kajian *chaos game* pada aturan pemilihan titik acuan berbentuk poligon *convex* dilakukan dengan pencarian informasi lebih banyak melalui beberapa penelitian terdahulu yaitu pada segitiga (*triangle*), segi empat (*square*), dan segi enam (*hexagon*).

3.3 Modifikasi *Chaos Game* pada Aturan Pemilihan Titik Acuan Berbentuk Poligon *Non-Convex*

Random merupakan aturan *chaos game* yang urutan pemilihan titik acuannya secara acak dari iterasi satu ke iterasi selanjutnya. Misalnya iterasi pertama dipilih titik acuan T_0, iterasi kedua dan seterusnya dapat ditentukan titik acuan antara T_0, T_1, T_2, T_3, …, T_n secara acak dan boleh berulang hingga iterasi yang diinginkan. Pada tahap ini modifikasi *chaos game* yang digunakan yaitu dengan pemilihan titik acuan atau titik sudut membentuk poligon *non-convex*.

3.4 Simulasi Program

Pada simulasi program ini digunakan *software* MATLAB R2015b, dengan prosedur sebagai berikut:

1. *Input*

Iterasi yang dilakukan pada *chaos game* diinputkan pada tahap ini titik acuan dan titik awal untuk memulai *chaos game*. Penentuan titik-titiknya dilakukan secara manual, artinya dibuat sembarang titik sebanyak n titik $(n \geq 4)$ yang membentuk poligon *non-convex*.

2. Proses

a) Membuat beberapa titik acuan atau titik sudut yang membentuk poligon *non-convex*.

b) Menentukan titik awal secara acak.

c) Memilih titik acuan atau titik sudut untuk dihubungkan dengan titik awal yang dipilih secara *random*.

d) Menentukan sebuah titik baru yang berjarak setengah dari titik acuan.

e) Menjadikan titik baru dari langkah (d) menjadi titik awal.

f) Ulangi langkah (c) dengan urutan boleh berulang sampai iterasi yang diinginkan.

3. *Output*

Output yang dihasilkan dari simulasi ini berupa visualisasi penerapan modifikasi *chaos game* dari kumpulan titik tengah yang membentuk objek fraktal. Titik acuan selalu menarik titik-titik yang berada di sekitarnya, sehingga mempengaruhi bentuk fraktal yang dihasilkan.

3.5 Analisis Hasil

Hasil yang diperoleh dari simulasi program merupakan visualisasi dari penelitian menggunakan *chaos game*. Pada tahap ini akan ditunjukkan modifikasi *chaos game* pada pemilihan titik acuan poligon *non-convex* dapat mempengaruhi persebaran titik sehingga membentuk sebuah fraktal dengan melihat sifat keserupaan dirinya. Dengan anggapan bahwa, hasil simulasi program berupa kumpulan titik-titik yang sesuai dengan bentuk titik acuan awalnya dimana posisi titik-titik yang terbentuk selalu mengikuti arah titik acuannya.

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan dari hasil dan pembahasan, dapat ditarik kesimpulan sebagai berikut:

1. Semua simulasi yang telah dilakukan dengan berbagai macam bentuk titik acuan dapat membentuk fraktal. Apabila titik acuannya berbentuk *non-convex* maka titik acuan yang membentuk *non-convex* akan bersifat sebagai pengacau.
2. Pola fraktal yang dihasilkan pada poligon *non-convex* secara visual sama dengan poligon *convex* yang dibangun oleh titik terluarnya.
3. Kumpulan titik-titik bentukan poligon *non-convex* pada *chaos game* selalu tertarik pada titik acuannya masing-masing.

5.2 Saran

Pada penelitian ini dilakukan modifikasi *chaos game* dengan titik acuan yang membentuk poligon *non-convex* dengan aturan sesuai pada *chaos game* yaitu secara *random*. Penentuan titik acuan yang dilakukan menggunakan beberapa contoh titik acuan yaitu sebanyak enam contoh. Penggunaan beberapa contoh tersebut memberikan dugaan kuat bahwa aturan *chaos game* dapat membentuk fraktal. Diharapkan pada penelitian selanjutnya dapat membangkitkan objek fraktal dengan titik acuan yang membentuk poligon *non-convex* dengan aturan *chaos game* secara *non-random* dan melakukan pembuktian secara analitik bahwa *chaos game* dengan titik acuan berbentuk poligon *convex* maupun *non-convex* dapat membentuk fraktal.

DAFTAR PUSTAKA

Armana, R.F. 2016. Kajian Geometri Analitik Pada Masalah *Chaos Game*. *Skripsi*. Jember: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

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

Jampour, M., Javidi, M.M., Nejad, A.L., Ashourzadeh, M., dan Yaghoobi, M. 2010. A New Technique in saving Fingerprint with low volume by using Chaos Game and Fractal Theory. *International Journal of Artificial Intelligence and Interactive Multimedia.* 1(3):28-32.

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

Kigami, J. 2001. *Analysis on Fractals*. Cambridge : Univ. Press.

Kontributor. Wikipedia. 2018. *Chaos Game*. https://en.wikipedia.org/wiki/ [Diakses pada 9 Oktober 2018].

Kusno. 1995. *Geometri Algoritmik*. Jember: FKIP Universitas Jember.

Kusno. 2010. *Geometri Rancang Bangun Studi tentang Desain dan Pemodelan Benda dengan Kurva dan Permukaan Berbantu Komputer*. Jember: Fakultas MIPA Universitas Jember.

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

Purnomo, K.D. 2014. Algoritma Pembangkitan Segitiga Sierpinski dengan L-System. *Prosiding Seminar Nasional Matematika, Jurusan Matematika FMIPA Universitas Jember:* 365-375.

Purnomo, K.D., Armana, R.F., dan Kusno. 2016. Kajian Pembentukan Segitiga Sierpinski Pada Masalah *Chaos Game* dengan Memanfaatkan Transformasi Affine. *Jurnal Matematika*. Jember: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana.

Ratna, E.N. 2018. Modifikasi Aturan Chaos Game Dengan Memanfaatkan Titik Berat Segitiga. *Skripsi*. Jember: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Riyadi, B. 2007. Analisis dan Perancangan Perangkat Lunak Generator Gambar dan Musik Fraktal dengan Iterated Function System. *Skripsi*. Jakarta: Universitas Bina Nusantara.

Yunaning, F. 2018. Kajian Aturan *Non-Random Chaos Game* Pada Segitiga. *Skripsi*. Jember: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

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

LAMPIRAN

1. Membangkitkan *chaos game*

File **chaos_nr.m**

```
function [x1 y1 terpilih warna jarak]=chaos_nr(x,y,acak,iter)
n=length(x);
t=0;
for j=1:n
    wrn(j,:)=rand(1,3);
end
for i=1:iter
    t=t+1;
    terpilih(i)=randi(n);
    warna(i,:)=wrn(terpilih(i),:);
    x1(i)=(acak(1)+x(terpilih(i)))/2;
    y1(i)=(acak(2)+y(terpilih(i)))/2;
    acak=[x1(i) y1(i)];
    if t>=n
        t=0;
    end
end

for z=1:iter-3
    jarak(z)=sqrt((x1(z+3)-x1(z))^2+(y1(z+3)-y1(z))^2);
end
```

2. Menentukan peluang setiap titik acuan

File **frekuensi.m**

```
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;
```

3. Tampilan GUIDE

File **help.m**

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

% Last Modified by GUIDE v2.5 12-Nov-2018 07:13:24

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

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

% Update handles structure
guidata(hObject, handles);

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


% --- Outputs from this function are returned to the command line.
function varargout = help_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 during object creation, after setting all
properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
guidata(hObject,handles)
a=imread('konveks1.jpg');
imshow(a)
% Hint: place code in OpeningFcn to populate axes1


% --- Executes during object creation, after setting all
properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```

```
% handles    empty - handles not created until after all
CreateFcns called
guidata(hObject,handles)
a=imread('non_konveks1.jpg');
imshow(a)
% Hint: place code in OpeningFcn to populate axes2
```

File **chaos_non_konveks.m**

```
set(handles.pushbutton1,'enable','off')
prompt={'Jumlah Titik :'};
name='Titik';
numlines=1;
answer=inputdlg(prompt,name,numlines);
a=str2double(answer);
if a<4
    errordlg('Titik Acuan Harus lebih dari 4','ERROR');
    set(handles.pushbutton1,'enable','on')
    return
end
for j=1:a+1
    if j>a
        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<=a

plot(TS(j,1),TS(j,2),'o','markersize',10,'markerfacecolor','y');
        text(TS(j,1),TS(j,2)+0.2,['\color{black}T' num2str(j-
1)],'fontsize',11,'fontweight',...

'bold','fontname','cambria','HorizontalAlignment','center','Vertic
alAlignment','baseline')
     else
        for z=1:a
            if z<a
                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,'xtick',[0:20],'ytick',[0:20],'xgrid','on','ygri
d','on',...
```

```
    'xlim',[0 20],'ylim',[0 20]);

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



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'));
nr=get(handles.uitable1,'userdata');
[x1 y1 terpilih warna jarak]=chaos_nr(x,y,acak,iter);
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)={['A0']};
pjg=length(x1);
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)) ' %']};
    matdata(i,:)={a4,a5};
    rname(i)={['T' num2str(i-1)]};
end
set(handles.uitable1,'data',[matdata
a7'],'rowname',rname,'columnname',{'x','y','Peluang'},'columnwidth
',{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)={['          T' num2str(mat(i,3)-1)]};
    end
    matacak(i,:)={a1,a2};
    rownames(i)={['A' num2str(i-1)]};
    if i<=pjg
        plot(x1(i),y1(i),'.','color',warna(i,:),'markersize',3)
    end
end
waktu=toc(waktu);
set(handles.uitable2,'data',[matacak
a3'],'rowname',rownames,'columnname',{'x','y','Terpilih'},...
```

```
    'columnwidth',{70})
set(handles.axes1,'xtick',[0:20],'ytick',[0:20],'xgrid','on','ygri
d','on',...
    'xlim',[0 20],'ylim',[0 20]);
xlabel('x')
ylabel('y')
for i=1:m-1
    text(ttk(i,1),ttk(i,2)+0.2,['\color{black}T' num2str(i-
1)],'fontsize',11,'fontweight',...

'bold','fontname','cambria','HorizontalAlignment','center','Vertic
alAlignment','baseline')
end
for z=1:m-1
            if z<m-1
                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(ttk(1:m-1,1),ttk(1:m-
1,2),'o','markersize',10,'markerfacecolor','y');
```