



**MODIFIKASI *CHAOS GAME* DENGAN VARIASI ROTASI  
PADA SEGIEMPAT**

**SKRIPSI**

Oleh

**Indy Larasati  
NIM 151810101022**

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS JEMBER  
2019**



# **MODIFIKASI *CHAOS GAME* DENGAN VARIASI ROTASI PADA SEGIEMPAT**

## **SKRIPSI**

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

Oleh

**Indy Larasati**  
**NIM 151810101022**

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS JEMBER  
2019**

## **PERSEMBAHAN**

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

1. Ibunda Widi Asmarani dan Ayahanda Ir. Indra Widarmadi, M.T. yang telah membesarkan, mendidik, mendoakan, serta memotivasi dengan penuh kasih sayang dan pengorbanan selama ini;
2. Adik Dyas Restu Palupi dan Kaniagara Winata tersayang yang telah mendoakan dan memberi dukungan kepada penulis;
3. Guru-guru TK Pertiwi, SDN Padurenan VI, SMPN 2 Bekasi, SMAN 9 Bekasi, dan dosen-dosen khususnya Jurusan Matematika FMIPA Universitas Jember yang telah memberikan ilmu, memotivasi, dan membimbing dengan penuh kesabaran;
4. Almamater tercinta Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

## MOTTO

Dia menentukan rahmat-Nya kepada siapa yang Dia kehendaki. Dan Allah memiliki karunia yang besar.

(Q.S Āli ‘Imrān: 74)<sup>1</sup>

Kalau kita hidup dari apa yang kita cintai, maka kita akan mencintai hidup kita.

(Raditya Dika)<sup>2</sup>

---

<sup>1</sup> Kementrian Agama Republik Indonesia. 2018. *Al-Qur'an Tafsir Perkata*. Bandung: Al-Hambra.

<sup>2</sup> Raditya Dika. 2018. *Ubur-ubur Lembur*. Jakarta: GagasMedia.

## **PERNYATAAN**

Saya yang bertanda tangan dibawah ini:

Nama : Indy Larasati

NIM : 151810101022

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul: “Modifikasi *Chaos Game* dengan Variasi Rotasi pada Segiempat” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun dan bukan merupakan 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 manapun serta bersedia mendapat sanksi akademik bila ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Mei 2019

Yang menyatakan,

Indy Larasati

NIM 151810101022

# **SKRIPSI**

## **MODIFIKASI *CHAOS GAME* DENGAN VARIASI ROTASI PADA SEGIEMPAT**

Oleh

**Indy Larasati**

**NIM 151810101022**

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 Variasi Rotasi pada Segiempat”  
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

Ika Hesti Agustin, S.Si., M.Si.  
NIP. 198408012008012006

Abduh Riski, S.Si., M.Si.  
NIP. 199004062015041001

Mengesahkan,

Dekan

Drs. Sujito, Ph.D.  
NIP. 196102041987111001

## RINGKASAN

**Modifikasi *Chaos Game* dengan Variasi Rotasi pada Segiempat;** Indy Larasati; 151810101022; 2019; 90 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

*Chaos game* merupakan bentuk permainan menggambar suatu titik dalam suatu bangun geometri dengan aturan tertentu yang dilakukan berulang-ulang secara iteratif. *Chaos game* dapat dimodifikasi dengan memvariasikan beberapa aturannya, misalnya rasio kompresi, lokasi simpul, rotasi, iterasi, modifikasi aturan *random* dan *non-random*, dan lain sebagainya. Penelitian sebelumnya telah mengembangkan modifikasi aturan *chaos game* pada segitiga, segilima, hingga segi- $n$ . Modifikasi yang dilakukan berupa modifikasi aturan *random*, *non-random*, variasi rasio kompresi, dan lain sebagainya. Pada penelitian ini dikaji pengembangan modifikasi *chaos game* dengan variasi rotasi yang dilakukan pada segiempat dengan aturan *random* dan beberapa modifikasinya, yaitu titik terakhir yang telah dipilih tidak boleh dipilih kembali, titik yang berhadapan tidak boleh dipilih, titik sebelah kiri tidak boleh dipilih, dan titik sebelah kanan tidak boleh dipilih. Variasi rotasi juga dilakukan pada aturan *non-random* dengan beberapa modifikasi, yaitu *non-random* 1 dengan 4 pola pemilihan titik yang teratur dan berurutan ( $A_1 - A_2 - A_3 - A_4$ ), *non-random* 2 dengan 16 pola pemilihan titik, yaitu pada kelipatan 4 berulang dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_3 - A_4 - A_1 - A_2 - A_2 - A_3 - A_4 - A_1 - A_1 - A_2 - A_3 - A_4 - A_4$ ), *non-random* 3 dengan 20 pola pemilihan titik, yaitu pada kelipatan 5 melompati satu titik setelahnya dan pada iterasi  $t$  yaitu  $t \equiv 4 \pmod{5}$  titik yang dipilih adalah titik yang berhadapan dengan titik terakhirnya ( $A_1 - A_2 - A_3 - A_1 - A_3 - A_4 - A_1 - A_2 - A_4 - A_2 - A_3 - A_4 - A_1 - A_3 - A_1 - A_2 - A_3 - A_4 - A_2 - A_4$ ), *non-random* 4 dengan 24 pola pemilihan titik, yaitu pada kelipatan 6 dan pada iterasi  $t$  yaitu  $t \equiv 5 \pmod{6}$  berulang satu kali seperti titik sebelumnya, kemudian pada  $t \equiv 4 \pmod{6}$  melompati satu titik setelah titik terakhirnya ( $A_1 - A_2 - A_3 -$

$A_3 - A_3 - A_1 - A_2 - A_3 - A_4 - A_4 - A_4 - A_2 - A_3 - A_4 - A_1 - A_1 - A_1 - A_3 - A_4 - A_1 - A_2 - A_2 - A_2 - A_4$ ), dan *non-random* 5 dengan 28 pola pemilihan titik, pada iterasi  $t$  yaitu  $t \equiv 4 \bmod (7)$  melompati satu titik dari titik terakhir, pada iterasi  $t \equiv 5 \bmod (7)$  berulang seperti titik terakhir yang dipilih, pada iterasi  $t \equiv 6 \bmod (7)$  melompati satu titik dari titik terakhir lagi, dan pada kelipatan 7 berulang lagi satu titik dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_1 - A_1 - A_3 - A_3 - A_4 - A_1 - A_2 - A_4 - A_1 - A_2 - A_4 - A_4 - A_2 - A_2 - A_3 - A_4 - A_1 - A_3 - A_3 - A_1 - A_1 - A_2 - A_3 - A_4 - A_2 - A_2 - A_4 - A_4$ ).

Simulasi program pada semua aturan dilakukan dengan rasio kompresi bernilai 2 ( $r = 2$ ) yang berarti jarak titik hasil bentukannya adalah  $1/2$  dari titik tumpunya ( $\alpha = 1/2$ ), dan iterasi yang digunakan adalah 10.000 untuk aturan *random*, dan 5.000 untuk aturan *non-random* agar hasilnya terlihat jelas. Simulasi pada aturan *random* dan seluruh modifikasinya dilakukan dengan putaran rotasi setiap nilai sudut  $10^\circ$ . Hasil dari percobaan didapatkan visualisasi bentuk-bentuk yang menyerupai segiempat dalam bagian yang diperkecil dan menyebar pada setiap titik acuan. Dengan demikian, variasi rotasi yang dilakukan dengan aturan *random* dan seluruh modifikasinya merupakan bentuk fraktal karena membentuk objek tertentu yang identik dan memiliki kemiripan bentuk (*self-similarity*).

Simulasi pada aturan *non-random* dan seluruh modifikasinya dilakukan dengan perputaran sudut setiap  $30^\circ$ . Hasil dari seluruh percobaan didapatkan titik-titik hasil bentukan *chaos game* dengan aturan *non-random* berjauhan dan tidak menyebar ke semua titik sudut yang merupakan titik acuan, melainkan berkelompok pada titik koordinat tertentu yang merupakan titik sudut acuannya. Dengan demikian, objek-objek yang dihasilkan aturan *non-random chaos game* dan seluruh rotasinya bukan merupakan objek fraktal, namun memiliki karakteristik yaitu kumpulan titik hasil bentukannya “konvergen” berulang pada titik koordinat tertentu sebanyak pola pemilihan titik yang ditentukan.

## PRAKATA

Puji dan syukur penulis kepada Allah SWT, karena atas limpahan rahmat, kasih sayang, serta anugerah-Nya maka penulis dapat menyelesaikan skripsi yang berjudul “Modifikasi *Chaos Game* dengan Variasi Rotasi pada Segiempat”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata 1 (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember. Penyusunan skripsi ini tidak terlepas dari dukungan dan arahan dari beberapa pihak, baik yang dilakukan secara langsung maupun tidak langsung. Pada kesempatan ini penulis ingin menyampaikan terima kasih 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 meluangkan waktu dan tenaganya untuk memberi ilmu, arahan, dan dukungan untuk membimbing penulis dengan penuh kesabaran untuk menyelesaikan skripsi;
2. Ika Hesti Agustin, S.Si., M.Si., Abduh Riski, S.Si., M.Si., serta almarhum Drs. Rusli Hidayat, M.Sc., selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun dalam proses penyusunan skripsi;
3. Drs. Sujito, Ph.D. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
4. Seluruh dosen dan staf karyawan Fakultas Matematika dan Ilmu Pengetahuan Alam;
5. Ibu Widi Asmarani, Bapak Ir. Indra Widarmadi, M.T., Adik Dyas Restu Palupi dan Kaniagara Winata tercinta yang selalu mendoakan dan mendukung penulis;
6. Teman-teman satu angkatan SIGMA '15 yang telah membantu dan mendukung penulis selama masa studi;
7. Pengurus HIMATIKA “Geokompstat” masa bakti 2017 dan 2018 yang senantiasa memberi dukungan;
8. Pengurus IKAHIMATIKA Indonesia khususnya Divisi Kewirausahaan Wilayah V periode 2017-2019 yang telah membantu penulis dalam berproses;

9. UKMS TITIK khususnya teman satu *band* “G11SOPATI”, Tiloy, Omesh, Kobot, Kuro yang memberi banyak pengalaman serta dukungan;
10. Teman-teman paguyuban IMADA (Ikatan Mahasiswa DKI Jaya) yang turut berjuang bersama di Jember dan memotivasi penulis;
11. Teman-teman KKN 262 khususnya Amalia dan Mas Emon yang selalu memberi semangat;
12. Kak Nina Novita yang telah menginspirasi untuk memilih Jurusan Matematika;
13. Kakak tingkat dan adek tingkat serta alumni, Mas Haris, Massay yang telah membantu dan memotivasi penulis;
14. Teman satu bidang skripsi fraktal Novita, Ellen, Mitha, Nanda, Dwi, Iza, Jafna yang memberi semangat kepada penulis;
15. Sahabat tersayang BEHAMBLO Tika, Ayu, Nanda, dan Yuli yang selalu mendukung penulis;
16. Sahabat SMP Aldi, Nadya, Asas, Virgin, sahabat SMA “Gabuters” dan Hesti, Winda, Ulfa yang memberi semangat kepada penulis;
17. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulisan skripsi ini masih jauh dari kesempurnaan sehingga diharapkan adanya kritik dan saran untuk perbaikan selanjutnya. Penulis berharap semoga skripsi ini dapat bermanfaat untuk semua pihak.

Jember, Mei 2019

Penulis

## DAFTAR ISI

|                                                                                           | Halaman    |
|-------------------------------------------------------------------------------------------|------------|
| <b>HALAMAN JUDUL .....</b>                                                                | <b>i</b>   |
| <b>HALAMAN PERSEMBAHAN.....</b>                                                           | <b>ii</b>  |
| <b>HALAMAN MOTTO .....</b>                                                                | <b>iii</b> |
| <b>HALAMAN PERNYATAAN.....</b>                                                            | <b>iv</b>  |
| <b>HALAMAN PEMBIMBINGAN.....</b>                                                          | <b>v</b>   |
| <b>HALAMAN PENGESAHAN .....</b>                                                           | <b>vi</b>  |
| <b>RINGKASAN .....</b>                                                                    | <b>vii</b> |
| <b>PRAKATA .....</b>                                                                      | <b>ix</b>  |
| <b>DAFTAR ISI.....</b>                                                                    | <b>xi</b>  |
| <b>DAFTAR GAMBAR.....</b>                                                                 | <b>xii</b> |
| <b>BAB 1. PENDAHULUAN.....</b>                                                            | <b>1</b>   |
| <b>1.1 Latar Belakang .....</b>                                                           | <b>1</b>   |
| <b>1.2 Rumusan Masalah .....</b>                                                          | <b>3</b>   |
| <b>1.3 Batasan Masalah.....</b>                                                           | <b>3</b>   |
| <b>1.4 Tujuan Penelitian .....</b>                                                        | <b>3</b>   |
| <b>1.5 Manfaat.....</b>                                                                   | <b>3</b>   |
| <b>BAB 2. TINJAUAN PUSTAKA.....</b>                                                       | <b>4</b>   |
| <b>2.1 Fraktal .....</b>                                                                  | <b>4</b>   |
| <b>2.2 Chaos Game .....</b>                                                               | <b>7</b>   |
| <b>2.3 Transformasi.....</b>                                                              | <b>9</b>   |
| <b>BAB 3. METODE PENELITIAN .....</b>                                                     | <b>13</b>  |
| <b>BAB 4. HASIL DAN PEMBAHASAN .....</b>                                                  | <b>18</b>  |
| <b>4.1 Modifikasi Chaos Game dengan Variasi Rotasi pada Segiempat..</b>                   | <b>19</b>  |
| 4.1.1 Aturan <i>Random</i> .....                                                          | 19         |
| 4.1.2 Modifikasi 1: Titik Terakhir yang Telah Dipilih Tidak Boleh<br>Dipilih Kembali..... | 21         |
| 4.1.3 Modifikasi 2: Titik yang Berhadapan Tidak Boleh Dipilih....                         | 23         |
| 4.1.4 Modifikasi 3: Titik Sebelah Kiri Tidak Boleh Dipilih.....                           | 25         |

|                                                                                                                                       |           |
|---------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 4.1.5 Modifikasi 4: Titik Sebelah Kanan Tidak Boleh Dipilih.....                                                                      | 28        |
| 4.1.6 Aturan <i>Non-Random</i> .....                                                                                                  | 30        |
| 4.1.7 Aturan Rotasi <i>Chaos Game</i> Segiempat.....                                                                                  | 33        |
| <b>4.2 Simulasi Program Menggunakan Aturan <i>Random</i> dan<br/><i>Non-Random Chaos Game</i> dengan Variasi Sudut Rotasinya.....</b> | <b>35</b> |
| 4.2.1 Aturan <i>Random</i> .....                                                                                                      | 37        |
| 4.2.2 Modifikasi 1: Titik Terakhir yang Telah Dipilih Tidak Boleh<br>Dipilih Kembali.....                                             | 38        |
| 4.2.3 Modifikasi 2: Titik yang Berhadapan Tidak Boleh Dipilih....                                                                     | 39        |
| 4.2.4 Modifikasi 3: Titik Sebelah Kiri Tidak Boleh Dipilih.....                                                                       | 40        |
| 4.2.5 Modifikasi 4: Titik Sebelah Kanan Tidak Boleh Dipilih.....                                                                      | 41        |
| 4.2.6 Aturan <i>Non-Random</i> .....                                                                                                  | 42        |
| 4.2.7 Aturan Rotasi <i>Chaos Game</i> Segiempat.....                                                                                  | 45        |
| <b>4.3 Pembahasan.....</b>                                                                                                            | <b>49</b> |
| 4.3.1 Aturan <i>Random</i> .....                                                                                                      | 49        |
| 4.3.2 Modifikasi 1: Titik Terakhir yang Telah Dipilih Tidak Boleh<br>Dipilih Kembali.....                                             | 52        |
| 4.3.3 Modifikasi 2: Titik yang Berhadapan Tidak Boleh Dipilih....                                                                     | 55        |
| 4.3.4 Modifikasi 3: Titik Sebelah Kiri Tidak Boleh Dipilih.....                                                                       | 58        |
| 4.3.5 Modifikasi 4: Titik Sebelah Kanan Tidak Boleh Dipilih.....                                                                      | 61        |
| 4.3.6 Aturan <i>Non-Random</i> .....                                                                                                  | 64        |
| <b>BAB 5. KESIMPULAN DAN SARAN.....</b>                                                                                               | <b>75</b> |
| 5.1 Kesimpulan.....                                                                                                                   | 75        |
| 5.2 Saran.....                                                                                                                        | 75        |
| <b>DAFTAR PUSTAKA.....</b>                                                                                                            | <b>76</b> |
| <b>LAMPIRAN.....</b>                                                                                                                  | <b>78</b> |

## DAFTAR GAMBAR

|                                                                          | Halaman |
|--------------------------------------------------------------------------|---------|
| 2.1 Himpunan per-tiga tengah Cantor.....                                 | 5       |
| 2.2 Kurva Von Koch.....                                                  | 5       |
| 2.3 Segitiga Sierpinski.....                                             | 5       |
| 2.4 Debu Cantor.....                                                     | 6       |
| 2.5 Himpunan Mandelbrot.....                                             | 6       |
| 2.6 Bentuk fraktal alamiah.....                                          | 7       |
| 2.7 Sierpinski <i>hexagon</i> .....                                      | 8       |
| 2.8 Pembangunan Sierpinski <i>hexagon</i> .....                          | 8       |
| 2.9 Rotasi segitiga Sierpinski.....                                      | 11      |
| 2.10 "Film" fraktal.....                                                 | 12      |
| 3.1 Skema penelitian.....                                                | 13      |
| 3.2 <i>Chaos game</i> segiempat.....                                     | 16      |
| 4.1 Segiempat dengan titik sudut $A_1, A_2, A_3$ , dan $A_4$ .....       | 19      |
| 4.2 Titik awal $T_0$ di dalam segiempat.....                             | 20      |
| 4.3 Pemilihan titik sudut sebagai titik acuan secara acak.....           | 20      |
| 4.4 Aturan <i>random chaos game</i> pada iterasi ke-1.....               | 20      |
| 4.5 Aturan <i>random chaos game</i> pada iterasi ke-2.....               | 21      |
| 4.6 Segiempat dengan titik sudut $A_1, A_2, A_3$ , dan $A_4$ .....       | 21      |
| 4.7 Titik awal $T_0$ di dalam segiempat.....                             | 22      |
| 4.8 Pemilihan titik sudut sebagai titik acuan secara acak.....           | 22      |
| 4.9 Modifikasi 1 aturan <i>random chaos game</i> pada iterasi ke-1.....  | 22      |
| 4.10 Modifikasi 1 aturan <i>random chaos game</i> pada iterasi ke-2..... | 23      |
| 4.11 Modifikasi 1 aturan <i>random chaos game</i> pada iterasi ke-4..... | 23      |
| 4.12 Segiempat dengan titik sudut $A_1, A_2, A_3$ , dan $A_4$ .....      | 23      |
| 4.13 Titik awal $T_0$ di dalam segiempat.....                            | 24      |
| 4.14 Pemilihan titik sudut sebagai titik acuan secara acak.....          | 24      |
| 4.15 Modifikasi 2 aturan <i>random chaos game</i> pada iterasi ke-1..... | 24      |
| 4.16 Modifikasi 2 aturan <i>random chaos game</i> pada iterasi ke-2..... | 25      |

|                                                                                         |    |
|-----------------------------------------------------------------------------------------|----|
| 4.17 Modifikasi 2 aturan <i>random chaos game</i> pada iterasi ke-5.....                | 25 |
| 4.18 Segiempat dengan titik sudut $A_1, A_2, A_3$ , dan $A_4$ .....                     | 26 |
| 4.19 Titik awal $T_0$ di dalam segiempat.....                                           | 26 |
| 4.20 Pemilihan titik sudut sebagai titik acuan secara acak.....                         | 26 |
| 4.21 Modifikasi 3 aturan <i>random chaos game</i> pada iterasi ke-1.....                | 27 |
| 4.22 Modifikasi 3 aturan <i>random chaos game</i> pada iterasi ke-2.....                | 27 |
| 4.23 Modifikasi 3 aturan <i>random chaos game</i> pada iterasi ke-4.....                | 27 |
| 4.24 Segiempat dengan titik sudut $A_1, A_2, A_3$ , dan $A_4$ .....                     | 28 |
| 4.25 Titik awal $T_0$ di dalam segiempat.....                                           | 28 |
| 4.26 Pemilihan titik sudut sebagai titik acuan secara acak.....                         | 29 |
| 4.27 Modifikasi 4 aturan <i>random chaos game</i> pada iterasi ke-1.....                | 29 |
| 4.28 Modifikasi 4 aturan <i>random chaos game</i> pada iterasi ke-2.....                | 29 |
| 4.29 Modifikasi 4 aturan <i>random chaos game</i> pada iterasi ke-4.....                | 30 |
| 4.30 Segiempat dengan titik sudut $A_1, A_2, A_3$ , dan $A_4$ .....                     | 31 |
| 4.31 Titik awal $T_0$ di dalam segiempat.....                                           | 31 |
| 4.32 Pemilihan titik sudut sebagai titik acuan secara acak.....                         | 31 |
| 4.33 Aturan <i>non-random chaos game</i> pada iterasi ke-1.....                         | 32 |
| 4.34 Aturan <i>non-random chaos game</i> pada iterasi ke-2.....                         | 32 |
| 4.35 Aturan <i>non-random chaos game</i> pada iterasi ke-5.....                         | 32 |
| 4.36 Segiempat dengan titik sudut $A_1, A_2, A_3$ , dan $A_4$ .....                     | 33 |
| 4.37 Titik awal $T_0$ di dalam segiempat.....                                           | 33 |
| 4.38 Pemilihan titik sudut sebagai titik acuan secara acak.....                         | 34 |
| 4.39 Aturan rotasi <i>chaos game</i> pada iterasi ke-1.....                             | 34 |
| 4.40 Aturan rotasi <i>chaos game</i> pada iterasi ke-2.....                             | 34 |
| 4.41 Aturan rotasi <i>chaos game</i> pada iterasi ke-5.....                             | 35 |
| 4.42 Tampilan GUI MATLAB R2015b.....                                                    | 37 |
| 4.43 <i>Chaos game</i> segiempat dengan aturan <i>random</i> pada sudut $0^\circ$ ..... | 38 |
| 4.44 <i>Chaos game</i> segiempat dengan aturan modifikasi 1 pada sudut $0^\circ$ .....  | 39 |
| 4.45 <i>Chaos game</i> segiempat dengan aturan modifikasi 2 pada sudut $0^\circ$ .....  | 40 |
| 4.46 <i>Chaos game</i> segiempat dengan aturan modifikasi 3 pada sudut $0^\circ$ .....  | 41 |
| 4.47 <i>Chaos game</i> segiempat dengan aturan modifikasi 4 pada sudut $0^\circ$ .....  | 42 |

|                                                                                                |    |
|------------------------------------------------------------------------------------------------|----|
| 4.48 <i>Chaos game</i> segiempat dengan aturan <i>non-random</i> 1 pada sudut $0^\circ$ .....  | 43 |
| 4.49 <i>Chaos game</i> segiempat dengan aturan <i>non-random</i> 2 pada sudut $0^\circ$ .....  | 43 |
| 4.50 <i>Chaos game</i> segiempat dengan aturan <i>non-random</i> 3 pada sudut $0^\circ$ .....  | 44 |
| 4.51 <i>Chaos game</i> segiempat dengan aturan <i>non-random</i> 4 pada sudut $0^\circ$ .....  | 44 |
| 4.52 <i>Chaos game</i> segiempat dengan aturan <i>non-random</i> 5 pada sudut $0^\circ$ .....  | 44 |
| 4.53 <i>Chaos game</i> segiempat dengan aturan <i>random</i> pada sudut $30^\circ$ .....       | 45 |
| 4.54 <i>Chaos game</i> segiempat dengan aturan modifikasi 1 pada sudut $30^\circ$ .....        | 46 |
| 4.55 <i>Chaos game</i> segiempat dengan aturan modifikasi 2 pada sudut $30^\circ$ .....        | 46 |
| 4.56 <i>Chaos game</i> segiempat dengan aturan modifikasi 3 pada sudut $30^\circ$ .....        | 46 |
| 4.57 <i>Chaos game</i> segiempat dengan aturan modifikasi 4 pada sudut $30^\circ$ .....        | 47 |
| 4.58 <i>Chaos game</i> segiempat dengan aturan <i>non-random</i> 1 pada sudut $30^\circ$ ..... | 47 |
| 4.59 <i>Chaos game</i> segiempat dengan aturan <i>non-random</i> 2 pada sudut $30^\circ$ ..... | 47 |
| 4.60 <i>Chaos game</i> segiempat dengan aturan <i>non-random</i> 3 pada sudut $30^\circ$ ..... | 48 |
| 4.61 <i>Chaos game</i> segiempat dengan aturan <i>non-random</i> 4 pada sudut $30^\circ$ ..... | 48 |
| 4.62 <i>Chaos game</i> segiempat dengan aturan <i>non-random</i> 5 pada sudut $30^\circ$ ..... | 48 |
| 4.63 Variasi rotasi sudut positif dengan aturan <i>random</i> .....                            | 49 |
| 4.64 Variasi rotasi sudut negatif dengan aturan <i>random</i> .....                            | 51 |
| 4.65 Variasi rotasi sudut positif pada modifikasi 1 aturan <i>random</i> .....                 | 52 |
| 4.66 Variasi rotasi sudut negatif pada modifikasi 1 aturan <i>random</i> .....                 | 54 |
| 4.67 Variasi rotasi sudut positif pada modifikasi 2 aturan <i>random</i> .....                 | 55 |
| 4.68 Variasi rotasi sudut negatif pada modifikasi 2 aturan <i>random</i> .....                 | 57 |
| 4.69 Variasi rotasi sudut positif pada modifikasi 3 aturan <i>random</i> .....                 | 58 |
| 4.70 Variasi rotasi sudut negatif pada modifikasi 3 aturan <i>random</i> .....                 | 60 |
| 4.71 Variasi rotasi sudut positif pada modifikasi 4 aturan <i>random</i> .....                 | 61 |
| 4.72 Variasi rotasi sudut negatif pada modifikasi 4 aturan <i>random</i> .....                 | 63 |
| 4.73 Variasi rotasi sudut positif <i>chaos game</i> dengan aturan <i>non-random</i> 1.....     | 64 |
| 4.74 Variasi rotasi sudut positif <i>chaos game</i> dengan aturan <i>non-random</i> 2.....     | 65 |
| 4.75 Variasi rotasi sudut positif <i>chaos game</i> dengan aturan <i>non-random</i> 3.....     | 66 |
| 4.76 Variasi rotasi sudut positif <i>chaos game</i> dengan aturan <i>non-random</i> 4.....     | 67 |
| 4.77 Variasi rotasi sudut positif <i>chaos game</i> dengan aturan <i>non-random</i> 5.....     | 68 |

|                                                            |    |
|------------------------------------------------------------|----|
| 4.78 Koordinat titik baru aturan <i>non-random</i> 1 ..... | 69 |
| 4.79 Koordinat titik baru aturan <i>non-random</i> 2 ..... | 70 |
| 4.80 Koordinat titik baru aturan <i>non-random</i> 3 ..... | 71 |
| 4.81 Koordinat titik baru aturan <i>non-random</i> 4 ..... | 72 |
| 4.82 Koordinat titik baru aturan <i>non-random</i> 5 ..... | 73 |

## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Fraktal menurut kamus Webster (1975) didefinisikan sebagai potongan yang tidak rata, salah satu variasi kurva yang tidak beraturan dan mengulangi dirinya sendiri pada skala tertentu. Konsep fraktal pertama kali diusulkan oleh Benoît B. Mandelbrot yang merupakan seorang matematikawan berkebangsaan Perancis dan Amerika Serikat. Mandelbrot dianggap sebagai orang yang memperkenalkan dan memopulerkan istilah "fraktal". Istilah tersebut berasal dari kata sifat bahasa latin yaitu *fractus*. Kata *fractus* artinya memecah-mecah, membuat bagian-bagian yang tidak beraturan. Sebenarnya, dibalik bentuk tak beraturan fraktal ini terdapat keteraturan dalam bentuk “kemiripan diri” (*self-similarity*), yaitu bentuk fraktal sebenarnya berasal dari suatu bentuk dasar yang teratur.

Menurut Romadiastri (2013), fraktal adalah benda geometris yang kasar pada segala skala, dan terlihat dapat "dibagi-bagi" dengan cara yang radikal. Beberapa fraktal dapat dipecah menjadi beberapa bagian yang semuanya mirip dengan fraktal aslinya. Geometri fraktal adalah cabang matematika yang mempelajari sifat-sifat dan perilaku berbagai jenis fraktal. Berbagai jenis fraktal pada awalnya dipelajari sebagai benda-benda matematis yang dapat diukur dengan perhitungan matematis biasa. Ada banyak bentuk matematis yang merupakan bentuk fraktal, misalnya seperti *Sierpinski Triangle*, *Koch Snowflake*, *Peano Curve*, *Mandelbrot Set*, *Lorenz Attractor*, dan *Chaos Game*. Fraktal juga banyak dijumpai pada objek-objek di dunia nyata, seperti awan, pegunungan, turbulensi, dan garis pantai, yang mempunyai bentuk geometri yang rumit. Secara umum fraktal bentuknya tidak teratur, merupakan bentuk yang tidak berdasarkan linearitas seperti pada bentuk matematis umumnya, jadi bukan termasuk benda yang terdefinisikan oleh geometri tradisional atau lebih dikenal dengan geometri *Euclid*.

*Chaos game* merupakan bentuk permainan menggambar suatu titik dalam suatu 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. Pembangkitan *chaos game* pada segiempat, segilima, *hexagon*, hingga segi- $n$  pada dasarnya menggunakan prinsip yang sama.

Segitiga Sierpinski merupakan contoh dari objek fraktal. Salah satu metode untuk membangkitkannya adalah *chaos game*. Segitiga Sierpinski adalah jenis fraktal linier yang mempunyai sifat *self-similarity* (keserupaan diri) yang identik dari satu iterasi ke iterasi berikutnya. 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).

Pola fraktal seperti segitiga Sierpinski dan kurva Koch telah dikenal selama bertahun-tahun oleh para ahli Matematika. *Chaos game* dengan berbagai banyak variasinya memberikan kesempatan bagus untuk menggabungkan ide-ide dasar dari geometri, aljabar linier, probabilitas, dan topologi dengan beberapa matematika yang cukup kontemporer. Devaney (2003) telah memodifikasi segitiga Sierpinski dengan variasi rotasi yang membentuk “film” fraktal. Modifikasi segitiga Sierpinski dapat dilakukan dengan memvariasikan rotasi, rasio kompresi, atau lokasi dari simpul untuk masalah *chaos game*. Berdasarkan penelitian yang telah dilakukan oleh Devaney, penelitian ini akan membahas tentang pembentuk *chaos game* segiempat yang dirotasi sudutnya dengan menggunakan transformasi affine dengan rasio kompresi tertentu. Aturan produksinya yaitu dengan menggunakan aturan *random* dan *non-random*, dimana aturan *random* dilakukan dengan pemilihan titik pembentuk *chaos game* secara acak yang dapat dimodifikasi dengan syarat tertentu, sedangkan aturan *non-random* pemilihan titiknya dilakukan secara teratur dan berurutan sebanyak pola titik yang diinginkan. Penelitian dilakukan untuk menganalisis apakah *chaos game* segiempat yang divariasikan rotasinya dengan aturan tertentu akan tetap membentuk fraktal seperti segitiga Sierpinski.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka masalah yang dibahas dalam penelitian ini adalah bagaimana bentuk fraktal yang didapatkan dengan memvariasikan rotasi pada *chaos game*?

## 1.3 Batasan Masalah

Penelitian ini akan memvariasikan rotasi *chaos game* segiempat dengan rasio kompresi yang telah ditentukan, yaitu  $r = 2$  yang berarti jarak titik hasil bentukannya adalah  $1/2$  dari titik tumpunya ( $\alpha = 1/r = 1/2$ ).

## 1.4 Tujuan

Tujuan yang ingin dicapai pada penelitian ini yaitu untuk mengetahui variasi hasil bentukan *chaos game* segiempat dengan memvariasikan rotasinya apakah membentuk fraktal atau tidak.

## 1.5 Manfaat

Manfaat dari penelitian ini adalah memberikan informasi tentang pengembangan *chaos game* yang dimodifikasi dengan variasi rotasi sehingga dapat bermanfaat sebagai referensi dalam penelitian-penelitian selanjutnya.

## BAB 2. TINJAUAN PUSTAKA

### 2.1 Fraktal

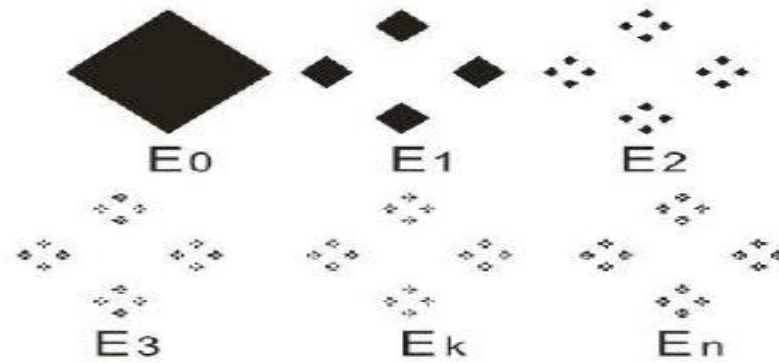
Kata fraktal pertama kali dicetuskan oleh Benoît Mandelbrot pada tahun 1975, ketika makalahnya yang berjudul “*A Theory of Fractal Set*” dipublikasikan. Bahasa Inggris dari fraktal adalah *fractal*. Sedangkan akar kata fraktal berasal dari kata latin *frangere* yang berarti terbelah menjadi fragmen-fragmen yang tidak teratur. Berdasarkan beberapa definisi mengenai fraktal, maka diambil pengertian bahwa fraktal adalah sebuah kajian dalam ilmu matematika yang mempelajari mengenai bentuk atau geometri yang didalamnya menunjukkan sebuah proses pengulangan tanpa batas. Geometri yang dilipatgandakan tersebut memiliki kemiripan bentuk satu sama lain (*self-similarity*), dan pada penyusunan pelipatgandaannya tersebut tidak terikat pada suatu aturan orientasi, bahkan cenderung meliuk-liuk dengan ukuran yang beragam mulai dari kecil hingga besar (Hasang dan Supardjo, 2012).

Fraktal didefinisikan sebagai bentuk kompleks tanpa batas, karena memiliki rincian jumlah yang tak terbatas dalam volume ruang yang terbatas. Bentuk ini akan selalu terlihat identik seperti pola yang tak terbatas. Namun, jika sebuah benda itu terbuat dari partikel dasar, maka rincian jumlah bentuk geometrisnya adalah terbatas. Jumlah ini ditentukan oleh jumlah partikel dasar dan posisi relatif mereka. Jumlahnya mungkin sangat besar, tetapi selalu terbatas (Shenker, 1994).

Menurut Romadiastri (2013), fraktal adalah konsep matematika yang membahas kesamaan pola pada semua skala. Pola fraktal yang sudah diterjemahkan dalam rumus fraktal dapat dimodifikasi dengan bantuan teknologi komputer sehingga menghasilkan desain pola baru yang sangat beragam dan juga unik. Keragaman desain ini dapat dilihat dari grafis, warna, ukuran, sudut dan perulangannya.

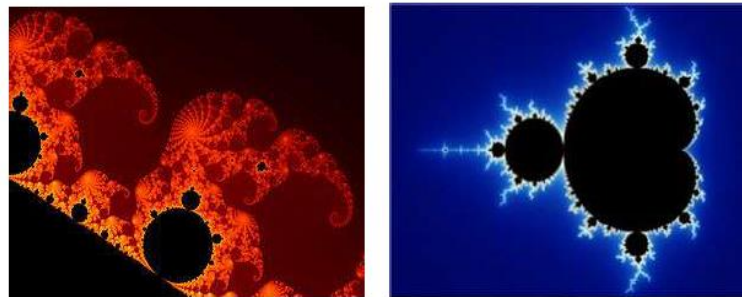
Bentuk fraktal memiliki variasi yang bermacam-macam. Beberapa contoh diantaranya adalah himpunan per-tiga tengah Cantor, kurva Von Koch, segitiga Sierpinski, dan debu Cantor. Contoh bentuk fraktal tersebut dapat dilihat pada Gambar 2.1 sampai dengan Gambar 2.4 (Sulistiyantoko, 2008).





Gambar 2.4 Debu Cantor

Benda-benda yang merupakan bentuk fraktal bisa ditemukan di alam nyata. Benda-benda tersebut menunjukkan struktur fraktal yang kompleks pada skala tertentu. Contohnya adalah awan, gunung, jaringan sungai, dan sistem pembuluh darah. Pohon dan pakis, juga merupakan contoh fraktal di alam dan dapat dimodelkan pada komputer dengan menggunakan algoritma rekursif. Sifat rekursifnya bisa dilihat dengan mudah yaitu dengan mengambil satu cabang dari suatu pohon dan akan terlihat bahwa cabang tersebut adalah miniatur dari pohonnya secara keseluruhan (tidak sama persis, tapi mirip) (Romadiastri, 2013).

Gambar 2.5 Himpunan Mandelbrot  
(Sumber: Hasang dan Supardjo, 2012)

Gambar 2.5 menunjukkan bentuk Himpunan Mandelbrot yang merupakan salah satu contoh bentuk fraktal. Fraktal ini banyak ditemukan di alam, seperti pada pola yang terdapat di daun dan ranting pohon, pada sayur brokoli, di gugusan awan putih, dalam riak ombak, pada detail yang bisa kita lihat di kepingan salju, dan banyak lagi. Contoh bentuk fraktal alamiah yang dapat ditemukan di alam ditunjukkan pada Gambar 2.6.



Gambar 2.6 Bentuk fraktal alamiah  
(Sumber: Hasang dan Supardjo, 2012)

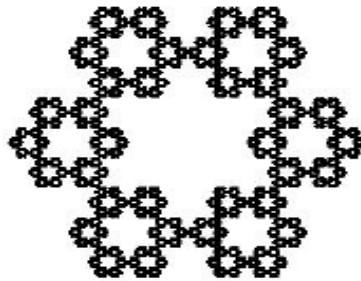
Riyadi (2007) mengklasifikasikan fraktal yang didasarkan oleh kemiripannya dengan diri sendiri atau *self-similarity*. Ada tiga jenis *self-similarity* dalam fraktal: *Exact self-similarity*, *Quasi-self-similarity*, *Statistical self-similarity*. Fraktal *exact self-similarity* tampak terlihat sama persis dengan ukuran dan rasio yang berbeda. Fraktal *quasi-self-similarity* lebih lemah jika dibandingkan dengan fraktal *exact self-similarity*. Fraktal *quasi-self-similarity* tampak mirip, namun tidak sama persis dengan ukuran dan rasio yang berbeda. Sedangkan *statistical self-similarity* merupakan fraktal *self-similarity* yang paling lemah diantara ketiganya. Pada umumnya, fraktal *statistical self-similarity* merupakan bentuk fraktal yang paling dasar. Hal ini dikarenakan fraktal *statistical self-similarity* memiliki ukuran numerik atau statistik yang dipertahankan pada ukuran dan rasio yang berbeda.

## 2.2 Chaos Game

*Chaos game* merupakan salah satu metode untuk membangkitkan segitiga Sierpinski. *Chaos game* juga dapat berupa segiempat, segilima, *hexagon*, hingga segi- $n$ . Sundbye (1997) membangun segitiga Sierpinski dengan perangkat *spinner* untuk mendapatkan angka acak dan segitiga yang telah diberi tanda pada titik sudutnya. *Chaos game* dibangkitkan dengan menggunakan rasio kompresi yang ditentukan, yaitu batas antara titik acak dan titik sudut. Rasio kompresi dituliskan dengan simbol  $r$  dan hasilnya jarak titik baru  $Z_{n+1}$  dari titik yang tepat adalah  $1/r$  dari jarak sebelumnya  $Z_n$  ke titik sudut (Milley, 2011).

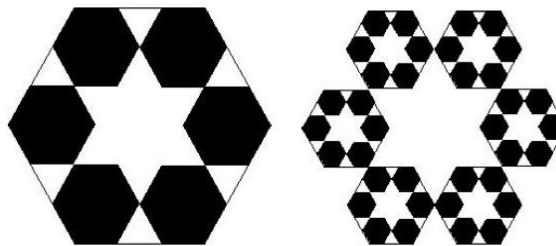
Menurut Yunaning (2018), *chaos game* merupakan suatu permainan titik yang masih dapat dikembangkan aturan produksinya, misal bagaimana jika mengubah jumlah titik sudutnya atau mengubah jarak antara titik awal dengan titik sudut yang

ditentukan, dan masih banyak lainnya. Devaney (2003) memodifikasi aturan produksi *chaos game* dan menghasilkan Sierpinski *hexagon*. Langkah-langkah *chaos game* yang dilakukan sama seperti pada segitiga Sierpinski, namun yang berbeda adalah jarak antara titik sudut dengan titik awal yaitu  $1/3$  jarak aslinya.



Gambar 2.7 Sierpinski *hexagon*  
(Sumber: Devaney, 2003)

Gambar 2.7 menunjukkan bentuk Sierpinski *hexagon* yang dibangun dengan rasio kompresi yang bernilai  $r = 3$ . Berikut pembangunannya akan ditunjukkan pada Gambar 2.8.



Gambar 2.8 Pembangunan Sierpinski *hexagon*  
(Sumber: Devaney, 2003)

Zohuri (2015) mengatakan bahwa bila titik awal pada *chaos game* terletak pada segitiga Sierpinski, maka titik yang ditandai berikutnya juga akan berada pada segitiga Sierpinski. Namun, sebenarnya titik yang dibentuk dari *chaos game* tidak selalu tepat terletak pada segitiga Sierpinski. Edgar (2008) telah menunjukkan bahwa pola segitiga Sierpinski mempunyai kemiripan dengan segitiga Pascal. Dalam hal ini susunan titik-titik pada segitiga Sierpinski bisa didapatkan dengan menggantikan bilangan ganjil dalam segitiga Pascal dengan titik hitam; sedangkan bilangan genapnya dihilangkan. Semakin banyak baris yang digambarkan dalam segitiga Pascal, susunan titik-titik tersebut semakin mendekati bentuk segitiga Sierpinski (Purnomo *et al.*, 2016).

### 2.3 Transformasi

Menurut Sulastri (2007), transformasi dapat diartikan sebagai suatu metode yang dapat digunakan untuk memanipulasi lokasi sebuah titik. Apabila transformasi dikenakan terhadap sekumpulan titik yang membentuk sebuah benda (objek) maka benda tersebut akan mengalami perubahan. Perubahan dalam hal ini adalah perubahan dari lokasi awal suatu benda menuju lokasi yang baru dari benda tersebut, biasanya dituliskan dalam bentuk koordinat. Koordinat dapat diartikan sebagai tata keseimbangan yang membantu di dalam menentukan suatu kondisi dengan nilai dan batas dalam konteks geometri. Koordinat diciptakan dengan tujuan untuk membantu manusia agar dapat dengan lebih mudah mengenal, menguasai serta mengendalikannya.

Metode yang paling umum digunakan dalam grafika komputer adalah metode transformasi affine. Transformasi affine adalah suatu transformasi yang menggunakan matriks dalam menghitung posisi objek yang baru. Keuntungan menggunakan matriks adalah transformasi yang berbeda-beda dapat digabungkan dengan mengalikan matriks-matriks tersebut sehingga diperoleh satu matriks transformasi.

Sebuah benda dengan titik  $P$  ditransformasikan ke titik  $Q$  menggunakan rumus-  
rumus tertentu, sehingga titik  $Q$  merupakan lokasi baru dari titik  $P$ . Pemetaan titik koordinat  $P = (P_x, P_y)$  ke  $Q = (Q_x, Q_y)$  dapat dilakukan dengan menggunakan rumus:

$$Q_x = aP_x + cP_y + Tr_x \quad (2.1)$$

$$Q_y = bP_x + dP_y + Tr_y \quad (2.2)$$

Transformasi dua dimensi menggunakan dua buah sumbu, yaitu sumbu  $x$  dan sumbu  $y$ . Sumbu  $x$  mewakili ukuran panjang, sedangkan sumbu  $y$  mewakili ukuran lebar.  $P_x$  adalah lokasi awal di sumbu  $x$  dan  $P_y$  adalah lokasi awal di sumbu  $y$ .  $Q_x$  adalah lokasi baru di sumbu  $x$  dan  $Q_y$  adalah lokasi baru di sumbu  $y$ . Kemudian  $a, b, c, d$  adalah konstantanya.  $Tr_x$  adalah translasi yang dilakukan terhadap sumbu  $x$ , sedangkan  $Tr_y$  adalah translasi yang dilakukan terhadap sumbu  $y$ . Perbedaan antara transformasi dua dimensi dengan transformasi tiga dimensi adalah satu buah

sumbu tambahan yang mewakili jarak pandang kedalaman yang terdapat pada transformasi tiga dimensi. Sumbu ini biasanya ditulis dengan perwakilan huruf  $z$ . Pada transformasi tiga dimensi, sumbu  $z$  berfungsi untuk menampilkan kesan kedalaman yang akan dilihat oleh mata manusia.

Transformasi dasar pada objek (dua dimensi) dapat berupa:

### 1. Translasi

Translasi dapat diartikan dengan perubahan lokasi dari lokasi awal menuju ke lokasi baru. Titik  $P(x, y)$  digeser sejumlah  $Tr_x$  pada sumbu  $x$  dan digeser sejumlah  $Tr_y$  pada sumbu  $y$ . Sehingga akan didapatkan rumus umum sebagai berikut:

$$Q(x, y) = P(x, y) + Tr = P(x + Tr_x, y + Tr_y) \quad (2.3)$$

### 2. Skala

Skala dapat diartikan dengan sebuah lokasi awal dikalikan dengan besaran yang di inginkan. Lokasi asli dikalikan dengan besaran  $S_x$  pada sumbu  $x$  dan  $S_y$  pada sumbu  $y$ . Sehingga akan diperoleh rumus umum sebagai berikut:

$$Q(x, y) = P \times S = P(x, y) \times S(x, y) = P(x \times S_x, y \times S_y) \quad (2.4)$$

### 3. Rotasi

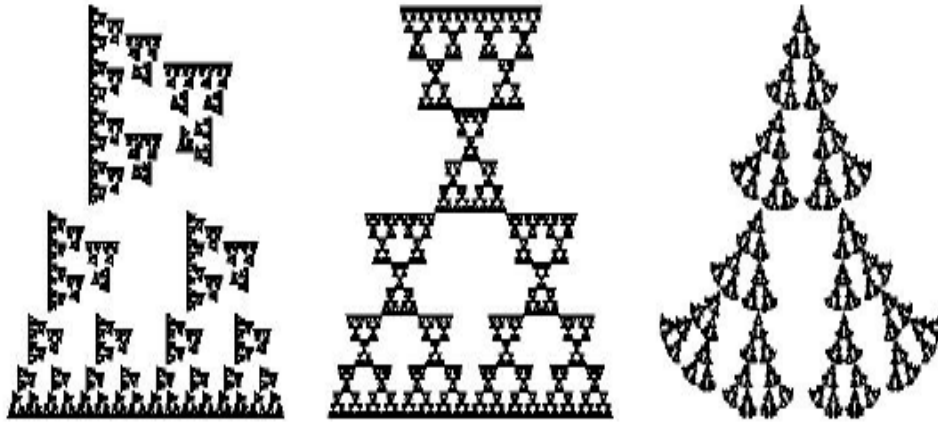
Rotasi dapat diartikan dengan perpindahan lokasi awal dari sebuah benda (objek) dengan cara diputar pada sudut  $\theta$  terhadap sumbu  $x$  atau sumbu  $y$ , dengan asumsi bahwa titik pusat  $(0,0)$  dengan sudut positif berlawanan dengan arah jarum jam dan sudut negatif searah dengan jarum jam. Rumus yang digunakan untuk melakukan rotasi adalah sebagai berikut:

$$x' = x \cos \theta - y \sin \theta \quad (2.5)$$

$$y' = x \sin \theta + y \cos \theta \quad (2.6)$$

Devaney (2003) memodifikasi segitiga Sierpinski dengan memanfaatkan transformasi geometri yaitu rotasi. Pembangkitan dimulai dengan titik pada sebuah segitiga, seperti dalam kasus ini adalah segitiga Sierpinski. Dua titik bawah dipindahkan ke titik setengah jarak dimana titik itu disebut. Titik paling atas dipindahkan ke titik separuh jarak ke titik tersebut, kemudian putar titik 90 derajat searah jarum jam dengan pusat perputaran titik puncak tersebut. Titik juga dapat diputar 180 derajat dan 20 derajat searah jarum jam. Rotasi dilakukan dengan rasio

kompresi yang bernilai  $r = 2$ , yang berarti jarak titik hasil bentukannya adalah  $1/2$  dari titik tumpunya. Hasil perputarannya dapat dilihat pada Gambar 2.9.



Gambar 2.9 Rotasi segitiga Sierpinski  
(Sumber: Devaney, 2003)

Modifikasi *chaos game* lainnya adalah pembuatan “film” fraktal. *Chaos game* sejauh ini ditentukan dengan memberikan lokasi simpul, rasio kompresi, dan rotasi. Modifikasi dapat dilakukan secara perlahan dengan mengubah beberapa rotasi, rasio kompresi, atau lokasi dari simpul untuk membuat “film” fraktal dengan banyak simetri. Sundbye (1997) dalam satu dimensi, persamaan linear berikut adalah transformasi affine:

$$w(x) = ax + b \quad (2.7)$$

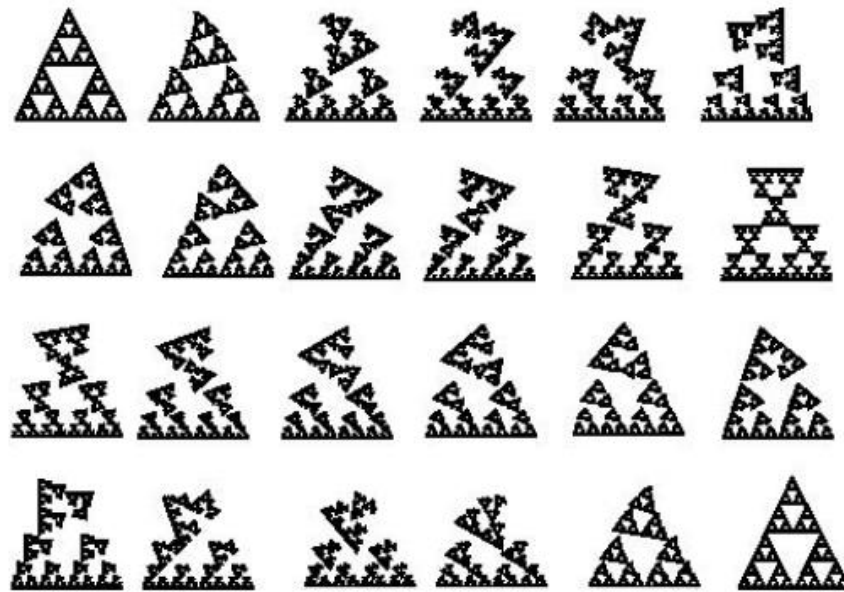
Rumus dalam dua dimensi adalah sebagai berikut:

$$w \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \quad (2.8)$$

Aturan-aturan transformasi affine dengan prinsip aljabar linier untuk membangun sebuah “film” fraktal dirumuskan:

$$T \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1/r & 0 \\ 0 & 1/r \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \quad (2.9)$$

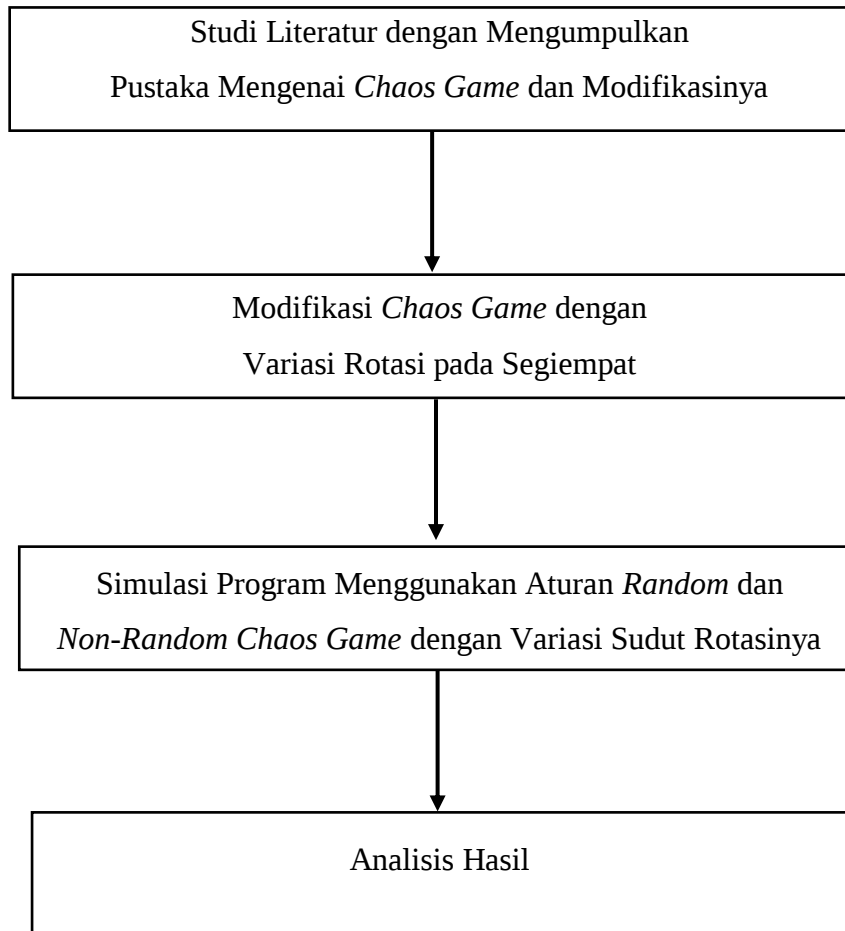
Transformasi affine memadukan perhitungan matriks translasi, skala dan rotasi dengan rasio kompresi yang dituliskan sebagai  $r$ , dimana  $\alpha$  merupakan jarak yang nilainya  $1/r$ .  $(x_0, y_0)$  adalah *vertex*,  $\alpha$  adalah jarak, dan  $\theta$  adalah sudut rotasi. Penggambaran pola perputaran “film” fraktal pada segitiga Sierpinski dengan metode transformasi affine dapat dilihat pada Gambar 2.10.



Gambar 2.10 “Film” fraktal  
(Sumber: Devaney, 2003)

### BAB 3. METODE PENELITIAN

Prosedur yang akan digunakan dalam penelitian ini akan digambarkan secara skematik sebagai berikut.



Gambar 3.1 Skema penelitian

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

- a. Studi Literatur dengan Mengumpulkan Pustaka Mengenai *Chaos Game* dan Modifikasinya

Langkah awal penelitian ini yaitu dengan studi literatur, yaitu melakukan pencarian informasi dari berbagai sumber tertulis seperti jurnal, buku, dan skripsi yang terkait dengan pustaka *chaos game* sebagai sumber yang relevan.

Sumber informasi yang berhubungan dengan masalah penelitian tersebut kemudian dianalisis untuk digunakan sebagai referensi penelitian.

b. Modifikasi *Chaos Game* dengan Variasi Rotasi pada Segiempat

Aturan rotasi pada *chaos game* segiempat sama dengan aturan yang digunakan untuk merotasikan segitiga Sierpinski dengan menggunakan rumus (2.9). Variasi yang akan dilakukan dalam penelitian ini terdapat pada pemilihan sudut dengan rasio kompresi yang ditentukan. Rasio kompresi yang akan digunakan pada penelitian ini adalah  $r = 2$  yang berarti jarak titik hasil bentukannya adalah  $1/2$  dari titik tumpunya ( $\alpha = 1/2$ ) untuk setiap sudut yang dilakukan untuk variasi rotasi.

c. Simulasi Program Menggunakan Aturan *Random* dan *Non-Random Chaos Game* dengan Variasi Sudut Rotasinya

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

1) *Input*

Tahap ini dilakukan dengan memasukkan titik awal dan iterasi yang dilakukan pada *chaos game* untuk membangun segiempat dengan aturan *random* dan *non-random*. Selanjutnya *input* rasio kompresi yang telah ditentukan, yaitu  $r = 2$  untuk setiap sudut dalam variasi rotasi. Sudut yang dimasukkan berupa variasi sudut positif dan negatif.

2) *Proses*

- a) Membuat *chaos game* segiempat dengan aturan *random* atau *non-random*.
- b) Menentukan titik awal untuk membangun *chaos game* segiempat.
- c) Memasukkan nilai rasio kompresi yang telah ditentukan, yaitu  $r = 2$ .
- d) Menentukan arah rotasi beserta nilai sudutnya.
- e) Menentukan jumlah iterasi maksimal.
- f) Membuat titik baru dengan aturan rotasi *chaos game*.
- g) Mengulang langkah (f) hingga iterasi mencapai jumlah iterasi maksimalnya.

- h) Terbentuk *chaos game* segiempat yang telah dirotasi.
- i) Ulangi langkah (d) sampai (h) hingga terbentuk berbagai *chaos game* segiempat yang telah dirotasi sesuai sudut dan iterasi yang diinginkan.

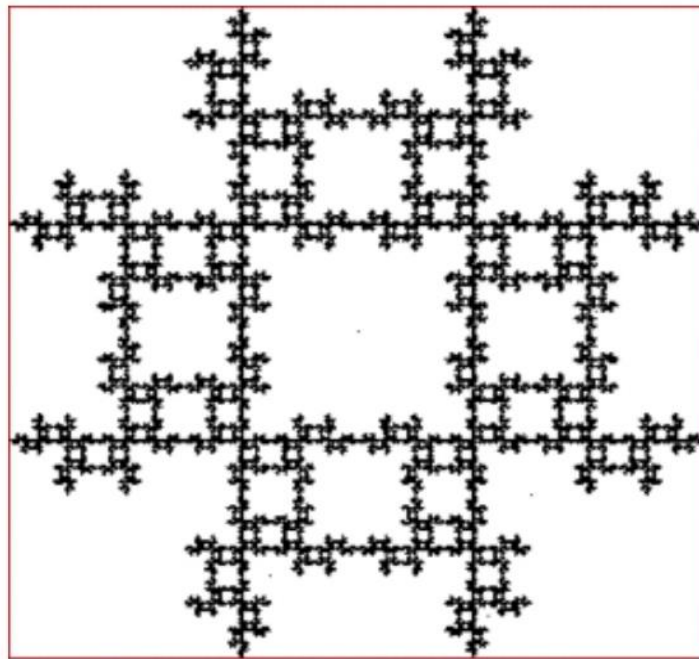
### 3) Output

*Output* yang dihasilkan dari simulasi ini adalah visualisasi bentuk *chaos game* segiempat yang dimodifikasi dengan variasi rotasi.

### d. Analisis Hasil

Modifikasi *chaos game* yang telah dirotasi akan ditunjukkan pada bagian analisis hasil. Pembangkitan dilakukan dengan menggunakan aturan *random* dan *non-random* untuk dibandingkan. Pembangkitan *chaos game* menggunakan aturan *random* dilakukan dengan beberapa modifikasi, modifikasi 1 yaitu titik terakhir yang telah dipilih tidak boleh dipilih kembali, modifikasi 2 yaitu titik yang berhadapan tidak boleh dipilih, modifikasi 3 yaitu titik sebelah kiri tidak boleh dipilih, dan modifikasi 4 yaitu titik sebelah kanan tidak boleh dipilih. Pada aturan *non-random* juga dilakukan dengan beberapa modifikasi, yaitu *non-random* 1 dengan 4 pola pemilihan titik yang teratur dan berurutan ( $A_1 - A_2 - A_3 - A_4$ ), *non-random* 2 dengan 16 pola pemilihan titik, yaitu pada kelipatan 4 berulang dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_4 - A_1 - A_2 - A_3 - A_4 - A_1 - A_2 - A_3 - A_4 - A_1 - A_2 - A_3 - A_4$ ), *non-random* 3 dengan 20 pola pemilihan titik, yaitu pada kelipatan 5 melompati satu titik setelahnya dan pada iterasi  $t$  yaitu  $t \equiv 4 \pmod{5}$  titik yang dipilih adalah titik yang berhadapan dengan titik terakhirnya ( $A_1 - A_2 - A_3 - A_4 - A_1 - A_3 - A_4 - A_1 - A_2 - A_4 - A_2 - A_3 - A_4 - A_1 - A_3 - A_1 - A_2 - A_3 - A_4 - A_2 - A_4$ ), *non-random* 4 dengan 24 pola pemilihan titik, yaitu pada kelipatan 6 dan pada iterasi  $t$  yaitu  $t \equiv 5 \pmod{6}$  berulang satu kali seperti titik sebelumnya, kemudian pada  $t \equiv 4 \pmod{6}$  melompati satu titik setelah titik terakhirnya ( $A_1 - A_2 - A_3 - A_4 - A_3 - A_1 - A_2 - A_3 - A_4 - A_4 - A_4 - A_2 - A_3 - A_4 - A_1 - A_1 - A_1 - A_3 - A_4 - A_1 - A_2 - A_2 - A_2 - A_4$ ), dan *non-random* 5 dengan 28 pola pemilihan titik, pada iterasi  $t$  yaitu  $t \equiv 4 \pmod{7}$  melompati satu titik dari titik terakhir, pada iterasi  $t \equiv 5 \pmod{7}$  berulang seperti titik terakhir yang dipilih, pada iterasi  $t \equiv 6 \pmod{7}$  melompati satu

titik dari titik terakhir lagi, dan pada kelipatan 7 berulang lagi satu titik dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_1 - A_1 - A_3 - A_3 - A_4 - A_1 - A_2 - A_4 - A_1 - A_2 - A_4 - A_4 - A_2 - A_2 - A_3 - A_4 - A_1 - A_3 - A_3 - A_1 - A_1 - A_2 - A_3 - A_4 - A_2 - A_2 - A_4 - A_4$ ). Bentuk awal *chaos game* segiempat yang telah dimodifikasi aturannya ditunjukkan pada Gambar 3.2. Modifikasi yang dimaksud yaitu aturan *random* dengan titik terakhir yang telah dipilih tidak boleh dipilih kembali.



Gambar 3.2 *Chaos game* segiempat  
(Sumber: [https://en.wikipedia.org/wiki/Chaos\\_game](https://en.wikipedia.org/wiki/Chaos_game))

Penelitian ini akan memvariasikan rotasi *chaos game* segiempat dengan melakukan beberapa sudut rotasi, misalnya:  $20^\circ$ ,  $40^\circ$ ,  $60^\circ$ ,  $90^\circ$ ,  $120^\circ$ ,  $150^\circ$ , dan seterusnya hingga kembali ke bentuk semula yaitu  $360^\circ$ . Rotasi juga akan dilakukan pada sudut yang bernilai negatif misalnya  $-20^\circ$ ,  $-40^\circ$ ,  $-60^\circ$ ,  $-90^\circ$ ,  $-120^\circ$ ,  $-150^\circ$ , dan seterusnya hingga  $-360^\circ$ . Pusat rotasi akan dilakukan dengan titik sudut segiempat sebagai titik tumpunya. Hasilnya akan dianalisis apakah pada *chaos game* segiempat yang sudutnya dimasukkan untuk variasi rotasi akan mengalami perubahan yang signifikan. Hipotesis awal yaitu variasi rotasi yang dilakukan pada *chaos game* segiempat

akan menghasilkan bentuk-bentuk yang berbeda. Bentuk-bentuk tersebut kemudian akan dianalisis apakah termasuk bentuk fraktal yang memiliki kemiripan bentuk satu sama lain (*self-similarity*) ataukah berubah tidak beraturan sehingga tidak dapat dikatakan sebagai fraktal.

## BAB 4. HASIL DAN PEMBAHASAN

Pada bab ini akan dijabarkan beberapa subbab mengenai modifikasi *chaos game* segiempat dengan variasi rotasi. Simulasi program akan ditunjukkan untuk mengetahui bentuk yang dihasilkan dari rotasi *chaos game* segiempat. Analisis dilakukan dengan melihat bentuk yang dihasilkan pada *input* nilai sudut yang berbeda-beda. Perbandingan dilakukan pada hasil bentuk yang didapatkan dengan aturan *random* dan *non-random chaos game* dengan nilai sudut positif dan negatif, serta titik tumpu rotasi berada pada titik sudut segiempat. Aturan dengan proses *random* dilakukan dengan pemilihan titik sudut acak tanpa syarat serta dengan beberapa syarat lainnya, yaitu titik terakhir yang telah dipilih tidak boleh dipilih kembali, titik yang berhadapan tidak boleh dipilih, titik sebelah kiri dari titik terakhir yang telah dipilih tidak boleh dipilih kembali, serta titik sebelah kanan dari titik terakhir yang telah dipilih tidak boleh dipilih kembali. Aturan *non-random* juga dilakukan dengan lima aturan modifikasi, yaitu *non-random 1* dengan 4 pola pemilihan titik yang teratur dan berurutan ( $A_1 - A_2 - A_3 - A_4$ ), *non-random 2* dengan 16 pola pemilihan titik, yaitu pada kelipatan 4 berulang dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_3 - A_4 - A_1 - A_2 - A_2 - A_3 - A_4 - A_1 - A_1 - A_2 - A_3 - A_4 - A_4$ ), *non-random 3* dengan 20 pola pemilihan titik, yaitu pada kelipatan 5 melompati satu titik setelahnya dan pada iterasi  $t$  yaitu  $t \equiv 4 \mod (5)$  titik yang dipilih adalah titik yang berhadapan dengan titik terakhirnya ( $A_1 - A_2 - A_3 - A_1 - A_3 - A_4 - A_1 - A_2 - A_4 - A_2 - A_3 - A_4 - A_1 - A_3 - A_1 - A_2 - A_3 - A_4 - A_2 - A_4$ ), *non-random 4* dengan 24 pola pemilihan titik, yaitu pada kelipatan 6 dan pada iterasi  $t$  yaitu  $t \equiv 5 \mod (6)$  berulang satu kali seperti titik sebelumnya, kemudian pada  $t \equiv 4 \mod (6)$  melompati satu titik setelah titik terakhirnya ( $A_1 - A_2 - A_3 - A_3 - A_3 - A_1 - A_2 - A_3 - A_4 - A_4 - A_4 - A_2 - A_3 - A_4 - A_1 - A_1 - A_1 - A_3 - A_4 - A_1 - A_2 - A_2 - A_2 - A_4$ ), dan *non-random 5* dengan 28 pola pemilihan titik, pada iterasi  $t$  yaitu  $t \equiv 4 \mod (7)$  melompati satu titik dari titik terakhir, pada iterasi  $t \equiv 5 \mod (7)$  berulang seperti titik terakhir yang dipilih, pada iterasi  $t \equiv 6 \mod (7)$  melompati satu titik dari titik terakhir lagi, dan pada kelipatan 7 berulang lagi satu titik dari

titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_1 - A_1 - A_3 - A_3 - A_4 - A_1 - A_2 - A_4 - A_1 - A_2 - A_4 - A_4 - A_2 - A_2 - A_3 - A_4 - A_1 - A_3 - A_3 - A_1 - A_1 - A_2 - A_3 - A_4 - A_2 - A_2 - A_4 - A_4$ ). Bentuk-bentuk yang didapatkan akan dianalisis apakah termasuk fraktal atau tidak, dengan melihat kemiripan bentuk satu sama lainnya (*self-similarity*).

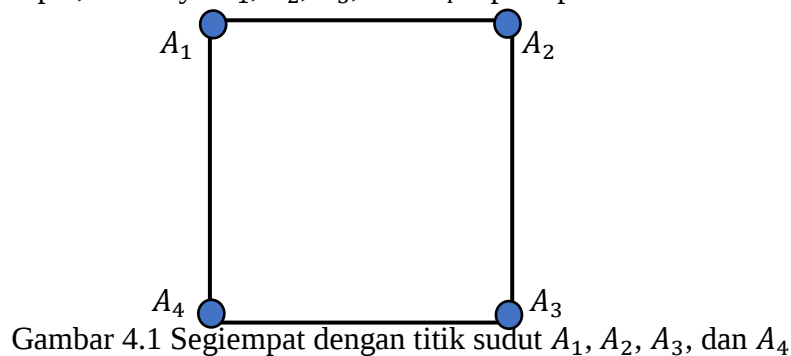
#### 4.1 Modifikasi *Chaos Game* dengan Variasi Rotasi pada Segiempat

Langkah awal yang dilakukan untuk merotasikan *chaos game* segiempat yaitu dengan aturan pemilihan titik sudut secara *random*, pemilihan titik sudut secara *random* bersyarat, serta aturan pemilihan titik sudut secara *non-random*. Nilai sudut dimasukkan bersamaan dengan rasio kompresi yang telah ditentukan, serta iterasi yang diinginkan. Aturan rotasi yang digunakan yaitu dengan rumus transformasi affine pada persamaan (2.9). Berikut aturan-aturan pembangkitan *chaos game* segiempat yang akan digunakan pada penelitian ini.

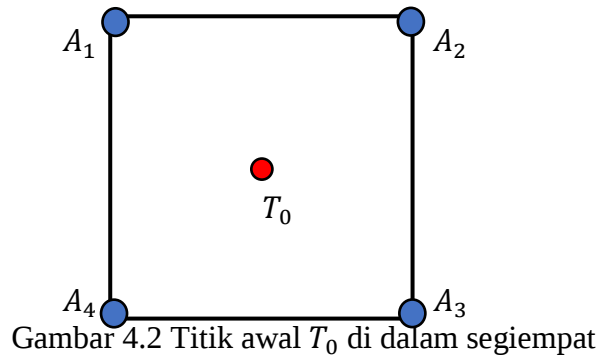
##### 4.1.1 Aturan *Random*

Misalkan titik-titik acuan diberi nama  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$ . Pemilihan titik sudut sebagai titik acuan secara acak boleh dilakukan dengan adanya perulangan pemilihan titik. Pemilihan titik dilakukan sampai iterasi yang diinginkan. Berikut adalah algoritma aturan *chaos game* pada segiempat yang dilakukan secara *random* dapat dilihat pada Gambar 4.1 sampai dengan Gambar 4.5.

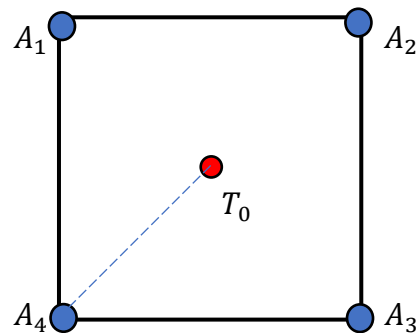
- 1) Membuat sebuah segiempat dan memberi nama pada setiap titik sudut segiempat, misalnya  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$  seperti pada Gambar 4.1.



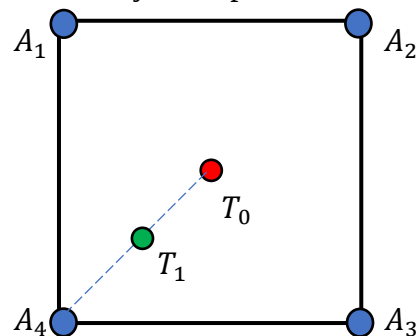
- 2) Memilih titik awal secara acak dan diberi nama  $T_0$  seperti pada Gambar 4.2.



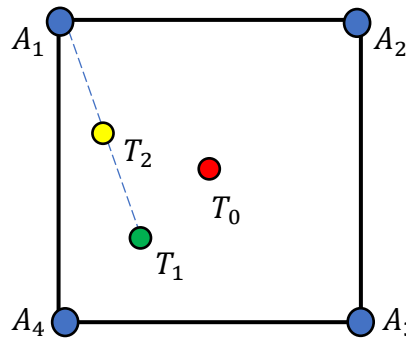
- 3) Memilih salah satu titik sudut pada segiempat untuk dihubungkan dengan titik awal  $T_0$  untuk membuat *chaos game* seperti pada Gambar 4.3.



- 4) Menentukan titik baru yang diberi nama  $T_1$  dan jaraknya ( $\alpha$ ) adalah setengah jarak titik sudut ke titik awal sesuai rasio kompresi yang telah ditentukan, yaitu  $r = 2$ . Berikut ditunjukkan pada Gambar 4.4.



- 5) Mengulang langkah (3) dengan titik awal adalah titik baru yang didapat dari iterasi sebelumnya dan boleh dilakukan perulangan pemilihan titik sudut sebagai titik acuan seperti pada Gambar 4.5.



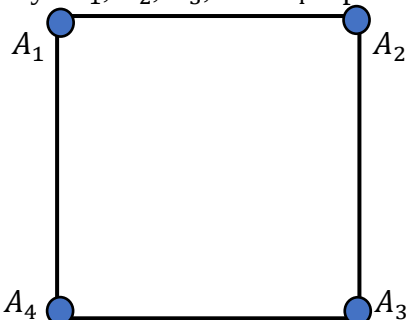
Gambar 4.5 Aturan *random chaos game* pada iterasi ke-2

6) Melakukan banyaknya pembentukan titik hingga iterasi yang diinginkan.

#### 4.1.2 Modifikasi 1: Titik Terakhir yang Telah Dipilih Tidak Boleh Dipilih Kembali

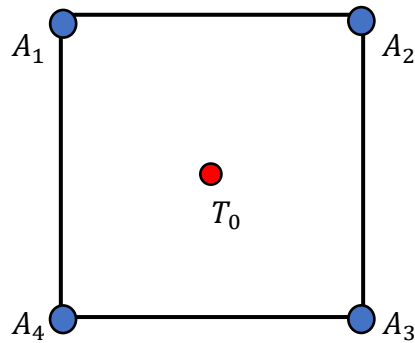
Misalkan diberikan titik-titik acuan  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$ . Modifikasi 1 aturan *random* dilakukan dengan pemilihan titik sudut secara acak, namun titik terakhir yang telah dipilih tidak boleh dipilih kembali. Misalkan titik terakhir yang dipilih adalah titik  $A_1$ , maka iterasi selanjutnya titik yang boleh dipilih adalah titik  $A_2$ ,  $A_3$ ,  $A_4$ , begitu pula seterusnya. Langkah pembentukan *chaos game* segiempat dengan aturan *random* titik terakhir yang telah dipilih tidak boleh dipilih kembali dapat dilihat dari Gambar 4.6 sampai dengan Gambar 4.11.

1) Membuat sebuah segiempat dan memberi nama pada setiap titik sudut segiempat, misalnya  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$  seperti pada Gambar 4.6.



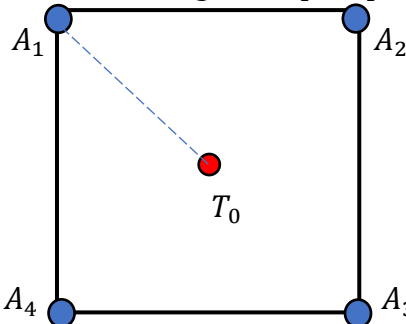
Gambar 4.6 Segiempat dengan titik sudut  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$

2) Memilih titik awal secara acak dan diberi nama  $T_0$  seperti pada Gambar 4.7.



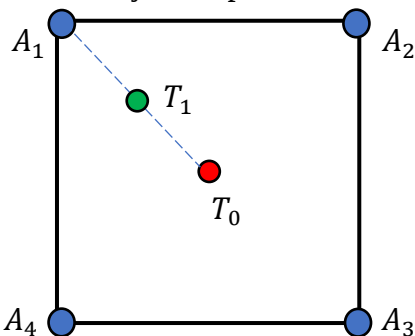
Gambar 4.7 Titik awal  $T_0$  di dalam segiempat

- 3) Memilih salah satu titik sudut pada segiempat untuk dihubungkan dengan titik awal  $T_0$  untuk membuat *chaos game* seperti pada Gambar 4.8.



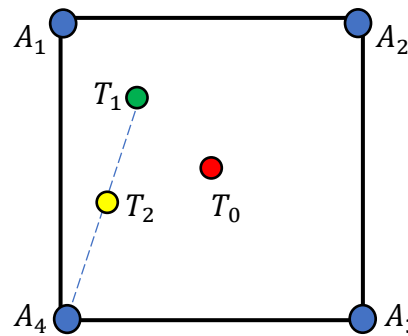
Gambar 4.8 Pemilihan titik sudut sebagai titik acuan secara acak

- 4) Menentukan titik baru yang diberi nama  $T_1$  dan jaraknya ( $\alpha$ ) adalah setengah jarak titik sudut ke titik awal sesuai rasio kompresi yang telah ditentukan, yaitu  $r = 2$ . Berikut ditunjukkan pada Gambar 4.9.

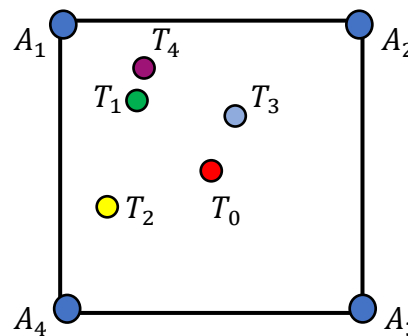


Gambar 4.9 Modifikasi 1 aturan *random chaos game* pada iterasi ke-1

- 5) Mengulang langkah (3) dengan titik awal adalah titik baru yang didapat dari iterasi sebelumnya namun pemilihan titik sudut yang digunakan sebagai acuan pada iterasi selanjutnya tidak boleh sama dengan titik terakhir yang telah dipilih sebelumnya. Pola pemilihan titik sudut dengan syarat tersebut dapat dilihat pada Gambar 4.10 dan Gambar 4.11.



Gambar 4.10 Modifikasi 1 aturan *random chaos game* pada iterasi ke-2



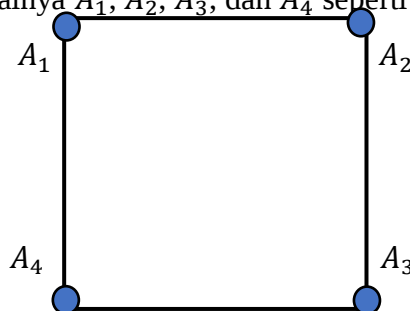
Gambar 4.11 Modifikasi 1 aturan *random chaos game* pada iterasi ke-4

- 6) Melakukan banyaknya pembentukan titik hingga iterasi yang diinginkan.

#### 4.1.3 Modifikasi 2: Titik yang Berhadapan Tidak Boleh Dipilih

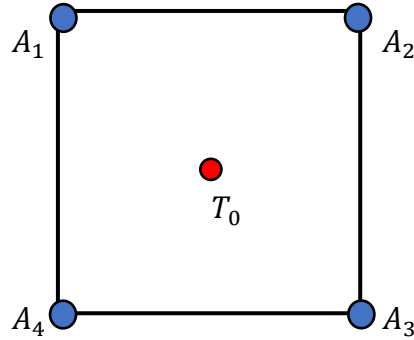
Modifikasi 2 aturan *random* dilakukan dengan pemilihan titik sudut secara acak, namun titik yang berhadapan pada segiempat dari titik terakhir yang dipilih tidak boleh dipilih kembali. Misalkan titik terakhir yang dipilih adalah titik  $A_1$ . Titik  $A_1$  berhadapan dengan titik  $A_3$ , maka iterasi selanjutnya titik  $A_3$  tidak boleh dipilih, begitu pula seterusnya. Langkah pembentukan *chaos game* segiempat dengan aturan ini dapat dilihat dari Gambar 4.12 sampai dengan Gambar 4.17.

- 1) Membuat sebuah segiempat dan memberi nama pada setiap titik sudut segiempat, misalnya  $A_1, A_2, A_3$ , dan  $A_4$  seperti pada Gambar 4.12.



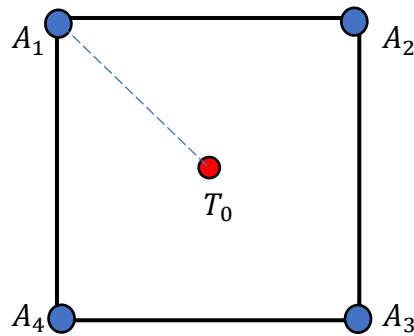
Gambar 4.12 Segiempat dengan titik sudut  $A_1, A_2, A_3$ , dan  $A_4$

- 2) Memilih titik awal secara acak dan diberi nama  $T_0$  seperti pada Gambar 4.13.



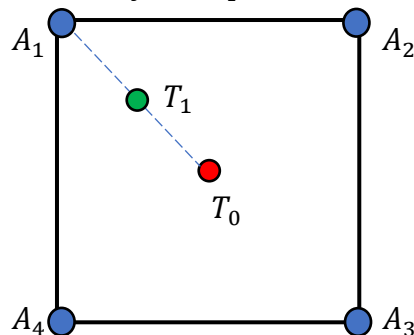
Gambar 4.13 Titik awal  $T_0$  di dalam segiempat

- 3) Memilih salah satu titik sudut pada segiempat untuk dihubungkan dengan titik awal  $T_0$  untuk membuat *chaos game* seperti pada Gambar 4.14.



Gambar 4.14 Pemilihan titik sudut sebagai titik acuan secara acak

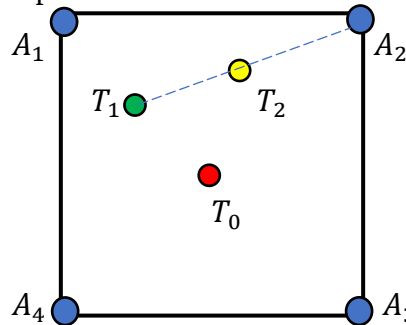
- 4) Menentukan titik baru yang diberi nama  $T_1$  dan jaraknya ( $\alpha$ ) adalah setengah jarak titik sudut ke titik awal sesuai rasio kompresi yang telah ditentukan, yaitu  $r = 2$ . Berikut ditunjukkan pada Gambar 4.15.



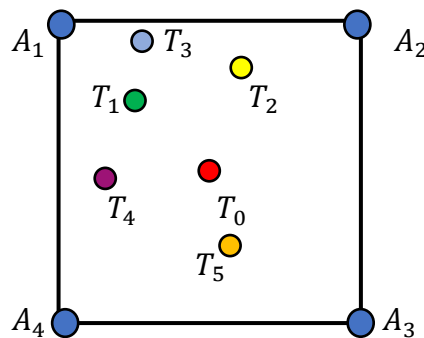
Gambar 4.15 Modifikasi 2 aturan *random chaos game* pada iterasi ke-1

- 5) Mengulang langkah (3) dengan titik awal adalah titik baru yang didapat dari iterasi sebelumnya namun pemilihan titik sudut yang digunakan sebagai acuan pada iterasi selanjutnya tidak boleh titik yang berhadapan dengan titik

terakhir telah dipilih sebelumnya. Pola pemilihan titik sudut dengan syarat ini dapat dilihat pada pada Gambar 4.16 dan Gambar 4.17.



Gambar 4.16 Modifikasi 2 aturan *random chaos game* pada iterasi ke-2



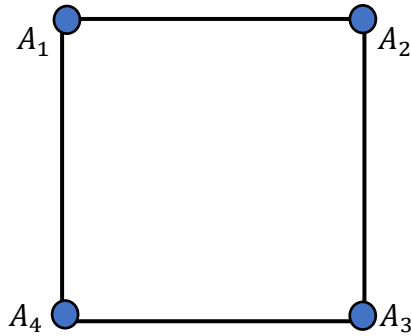
Gambar 4.17 Modifikasi 2 aturan *random chaos game* pada iterasi ke-5

- 6) Melakukan banyaknya pembentukan titik hingga iterasi yang diinginkan.

#### 4.1.4 Modifikasi 3: Titik Sebelah Kiri Tidak Boleh Dipilih

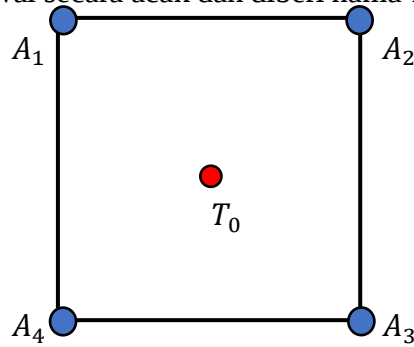
Misalkan diberikan titik-titik acuan  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$ . Modifikasi 3 aturan *random* dilakukan dengan pemilihan titik sudut secara acak, namun titik sebelah kiri dari titik terakhir yang telah dipilih tidak boleh dipilih. Titik sebelah kiri dilihat dari sudut pandang titik yang letaknya berada sebelum titik yang terakhir dipilih ( $A_1$ ,  $A_2$ ,  $A_3$ ,  $A_4$ ). Misalkan titik terakhir yang dipilih adalah titik  $A_2$ . Titik  $A_2$  sebelah kirinya (sebelumnya) adalah titik  $A_1$ , maka iterasi selanjutnya titik  $A_1$  tidak boleh dipilih, begitu pula seterusnya. Langkah pembentukan *chaos game* segiempat dengan aturan *random* titik sebelah kiri tidak boleh dipilih dapat dilihat dari Gambar 4.18 sampai dengan Gambar 4.23.

- 1) Membuat sebuah segiempat dan memberi nama pada setiap titik sudut segiempat, misalnya  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$  seperti pada Gambar 4.18.



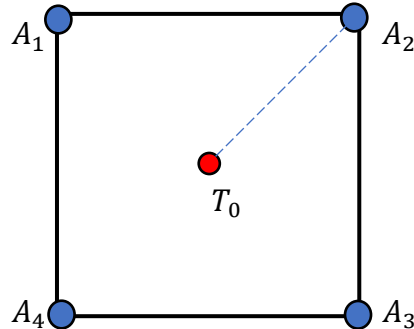
Gambar 4.18 Segiempat dengan titik sudut  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$

- 2) Memilih titik awal secara acak dan diberi nama  $T_0$  seperti pada Gambar 4.19.



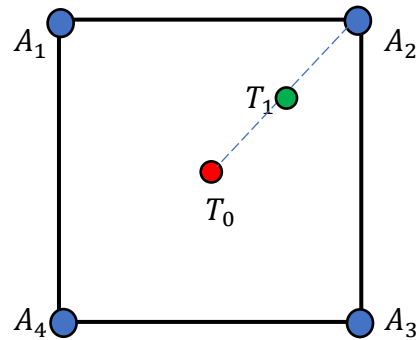
Gambar 4.19 Titik awal  $T_0$  di dalam segiempat

- 3) Memilih salah satu titik sudut pada segiempat untuk dihubungkan dengan titik awal  $T_0$  untuk membuat *chaos game* seperti pada Gambar 4.20.



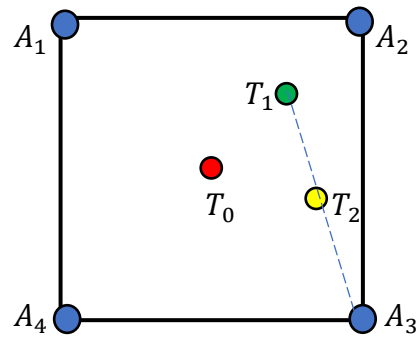
Gambar 4.20 Pemilihan titik sudut sebagai titik acuan secara acak

- 4) Menentukan titik baru yang diberi nama  $T_1$  dan jaraknya ( $\alpha$ ) adalah setengah jarak titik sudut ke titik awal sesuai rasio kompresi yang telah ditentukan, yaitu  $r = 2$ . Berikut ditunjukkan pada Gambar 4.21.

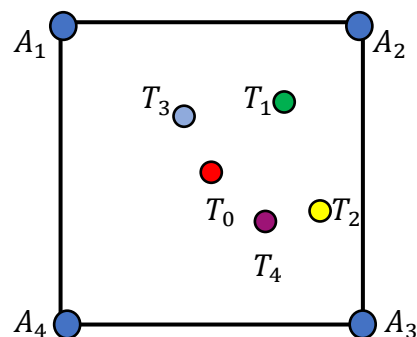


Gambar 4.21 Modifikasi 3 aturan *random chaos game* pada iterasi ke-1

- 5) Mengulang langkah (3) dengan titik awal adalah titik baru yang didapat dari iterasi sebelumnya namun pemilihan titik sudut yang digunakan sebagai acuan pada iterasi selanjutnya tidak boleh titik sebelah kiri (sebelumnya) dari titik yang telah dipilih sebelumnya. Pengecualian pada titik  $A_3$  dimana titik sebelah kirinya adalah titik  $A_4$ , maka setelah memilih titik  $A_3$  tidak boleh memilih titik  $A_4$ . Pola pemilihan titik sudut dengan syarat ini dapat dilihat pada pada Gambar 4.22 dan Gambar 4.23.



Gambar 4.22 Modifikasi 3 aturan *random chaos game* pada iterasi ke-2



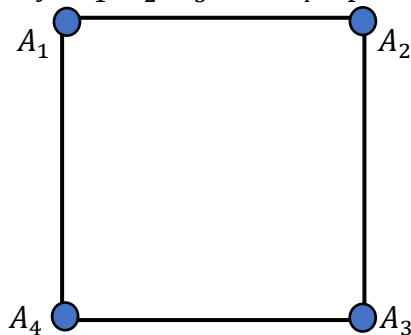
Gambar 4.23 Modifikasi 3 aturan *random chaos game* pada iterasi ke-4

- 6) Melakukan banyaknya pembentukan titik hingga iterasi yang diinginkan.

#### 4.1.5 Modifikasi 4: Titik Sebelah Kanan Tidak Boleh Dipilih

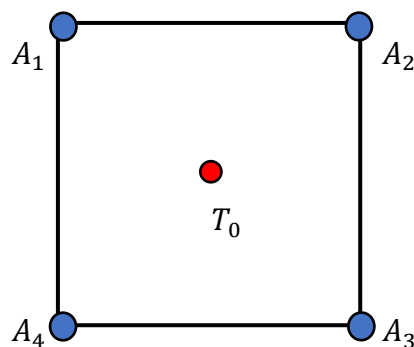
Misalkan diberikan titik-titik acuan  $A_1, A_2, A_3$ , dan  $A_4$ . Modifikasi 4 aturan *random* dilakukan dengan pemilihan titik sudut secara acak, namun titik sebelah kanan dari titik terakhir yang telah dipilih tidak boleh dipilih. Modifikasi aturan ini pada dasarnya mirip seperti modifikasi aturan 3 yang telah dijelaskan sebelumnya, hanya saja titik sebelah kanan dilihat dari sudut pandang titik yang letaknya berada setelah titik yang terakhir dipilih ( $A_1, A_2, A_3, A_4$ ). Misalkan titik terakhir yang dipilih adalah titik  $A_1$ . Titik  $A_1$  sebelah kanannya (setelahnya) adalah titik  $A_2$ , maka iterasi selanjutnya titik  $A_2$  tidak boleh dipilih, begitu pula seterusnya. Langkah pembentukan *chaos game* segiempat dengan aturan *random* titik sebelah kanan tidak boleh dipilih dapat dilihat dari Gambar 4.24 sampai dengan Gambar 4.29.

- 1) Membuat sebuah segiempat dan memberi nama pada setiap titik sudut segiempat, misalnya  $A_1, A_2, A_3$ , dan  $A_4$  seperti pada Gambar 4.24.



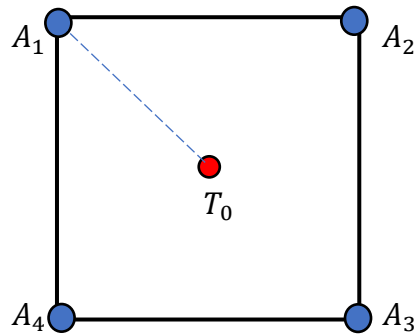
Gambar 4.24 Segiempat dengan titik sudut  $A_1, A_2, A_3$ , dan  $A_4$

- 2) Memilih titik awal secara acak dan diberi nama  $T_0$  seperti pada Gambar 4.25.



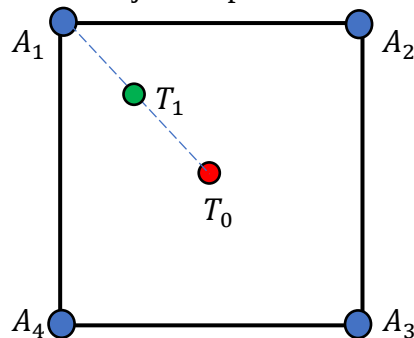
Gambar 4.25 Titik awal  $T_0$  di dalam segiempat

- 3) Memilih salah satu titik sudut pada segiempat untuk dihubungkan dengan titik awal  $T_0$  untuk membuat *chaos game* seperti pada Gambar 4.26.



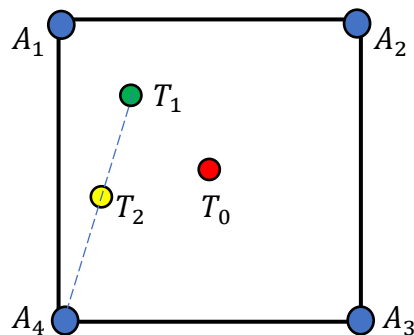
Gambar 4.26 Pemilihan titik sudut sebagai titik acuan secara acak

- 4) Menentukan titik baru yang diberi nama  $T_1$  dan jaraknya ( $\alpha$ ) adalah setengah jarak titik sudut ke titik awal sesuai rasio kompresi yang telah ditentukan, yaitu  $r = 2$ . Berikut ditunjukkan pada Gambar 4.27.

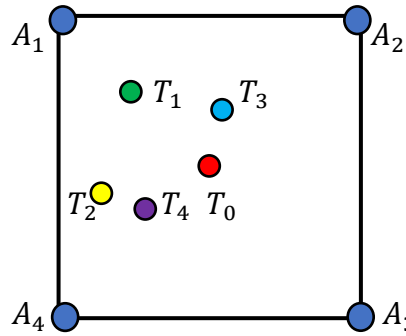


Gambar 4.27 Modifikasi 4 aturan *random chaos game* pada iterasi ke-1

- 5) Mengulang langkah (3) dengan titik awal adalah titik baru yang didapat dari iterasi sebelumnya, namun pemilihan titik sudut yang digunakan sebagai acuan pada iterasi selanjutnya tidak boleh titik sebelah kanan (setelahnya) dari titik yang telah dipilih sebelumnya. Pengecualian pada titik  $A_4$  dimana titik sebelah kanannya adalah titik  $A_3$ , maka setelah memilih titik  $A_4$  tidak boleh memilih titik  $A_3$ . Pola pemilihan titik sudut dengan syarat ini dapat dilihat pada Gambar 4.28 dan Gambar 4.29.



Gambar 4.28 Modifikasi 4 aturan *random chaos game* pada iterasi ke-2



Gambar 4.29 Modifikasi 4 aturan *random chaos game* pada iterasi ke-4

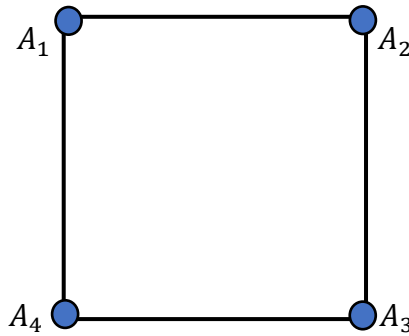
6) Melakukan banyaknya pembentukan titik hingga iterasi yang diinginkan.

#### 4.1.6 Aturan *Non-Random*

Misalkan diberikan titik-titik acuan  $A_1, A_2, A_3$ , dan  $A_4$ . Aturan *non-random* dilakukan dengan cara menentukan pemilihan titik sudut sebagai titik acuan secara berurutan dari iterasi satu hingga iterasi yang diinginkan. Aturan *non-random* dilakukan dengan beberapa modifikasi, yaitu *non-random 1* dengan 4 pola pemilihan titik yang teratur dan berurutan ( $A_1 - A_2 - A_3 - A_4$ ), *non-random 2* dengan 16 pola pemilihan titik, yaitu pada kelipatan 4 berulang dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_3 - A_4 - A_1 - A_2 - A_2 - A_3 - A_4 - A_1 - A_1 - A_2 - A_3 - A_4 - A_4$ ), *non-random 3* dengan 20 pola pemilihan titik, yaitu pada kelipatan 5 melompati satu titik setelahnya dan pada iterasi  $t$  yaitu  $t \equiv 4 \text{ mod } (5)$  titik yang dipilih adalah titik yang berhadapan dengan titik terakhirnya ( $A_1 - A_2 - A_3 - A_1 - A_3 - A_4 - A_1 - A_2 - A_4 - A_2 - A_3 - A_4 - A_1 - A_3 - A_1 - A_2 - A_3 - A_4 - A_2 - A_4$ ), *non-random 4* dengan 24 pola pemilihan titik, yaitu pada kelipatan 6 dan pada iterasi  $t$  yaitu  $t \equiv 5 \text{ mod } (6)$  berulang satu kali seperti titik sebelumnya, kemudian pada  $t \equiv 4 \text{ mod } (6)$  melompati satu titik setelah titik terakhirnya ( $A_1 - A_2 - A_3 - A_3 - A_3 - A_1 - A_2 - A_3 - A_4 - A_4 - A_4 - A_2 - A_3 - A_4 - A_1 - A_1 - A_1 - A_3 - A_4 - A_1 - A_2 - A_2 - A_2 - A_4$ ), dan *non-random 5* dengan 28 pola pemilihan titik, pada iterasi  $t$  yaitu  $t \equiv 4 \text{ mod } (7)$  melompati satu titik dari titik terakhir, pada iterasi  $t \equiv 5 \text{ mod } (7)$  berulang seperti titik terakhir yang dipilih, pada iterasi  $t \equiv 6 \text{ mod } (7)$  melompati satu titik dari titik terakhir lagi, dan pada kelipatan 7 berulang lagi satu titik dari

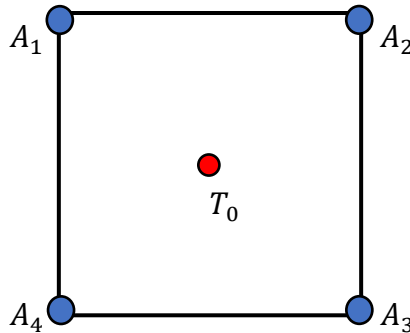
titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_1 - A_1 - A_3 - A_3 - A_4 - A_1 - A_2 - A_4 - A_1 - A_2 - A_4 - A_4 - A_2 - A_2 - A_3 - A_4 - A_1 - A_3 - A_3 - A_1 - A_1 - A_2 - A_3 - A_4 - A_2 - A_2 - A_4 - A_4$ ). Langkah pembentukan *chaos game* segiempat dengan aturan ini dapat dilihat dari Gambar 4.30 sampai dengan Gambar 4.35.

- 1) Membuat sebuah segiempat dan memberi nama pada setiap titik sudut segiempat, misalnya  $A_1, A_2, A_3$ , dan  $A_4$  seperti pada Gambar 4.30.



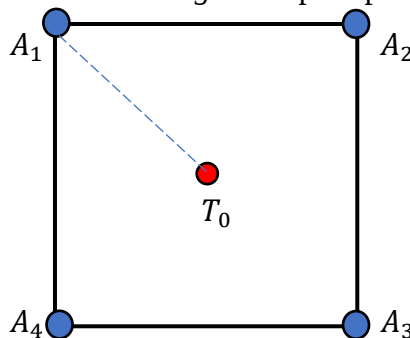
Gambar 4.30 Segiempat dengan titik sudut  $A_1, A_2, A_3$ , dan  $A_4$

- 2) Memilih titik awal secara acak dan diberi nama  $T_0$  seperti pada Gambar 4.31.



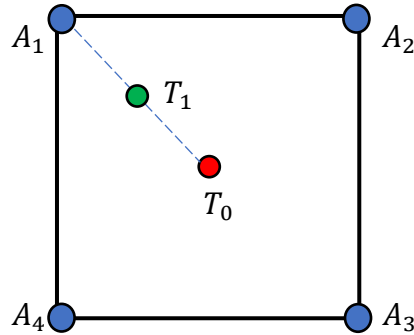
Gambar 4.31 Titik awal  $T_0$  di dalam segiempat

- 3) Memilih salah satu titik sudut pada segiempat untuk dihubungkan dengan titik awal  $T_0$  untuk membuat *chaos game* seperti pada Gambar 4.32.



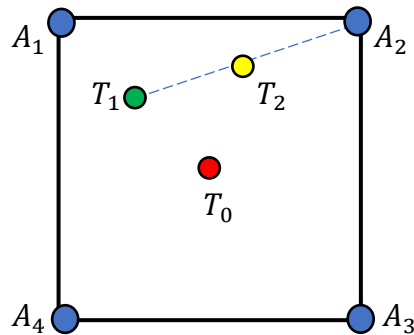
Gambar 4.32 Pemilihan titik sudut sebagai titik acuan secara acak

- 4) Menentukan titik baru yang diberi nama  $T_1$  dan jaraknya ( $\alpha$ ) adalah setengah jarak titik sudut ke titik awal sesuai rasio kompresi yang telah ditentukan, yaitu  $r = 2$ . Berikut ditunjukkan pada Gambar 4.33.

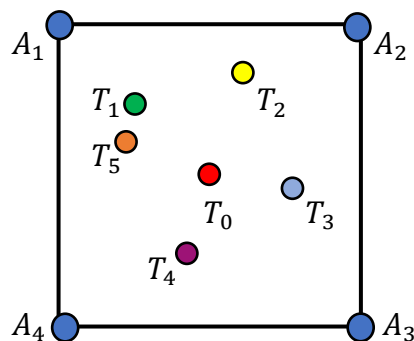


Gambar 4.33 Aturan *non-random chaos game* pada iterasi ke-1

- 5) Mengulang langkah (3) dengan titik awal adalah titik baru yang didapat dari iterasi sebelumnya namun pemilihan titik sudut diatur dengan pola berurutan, misalnya  $A_1 - A_2 - A_3 - A_4$ . Pola pemilihan titik sudut beraturan sebagai titik acuan seperti pada Gambar 4.34 dan Gambar 4.35.



Gambar 4.34 Aturan *non-random chaos game* pada iterasi ke-2



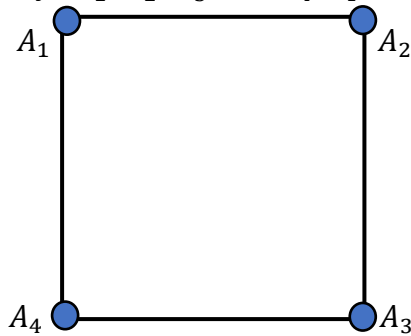
Gambar 4.35 Aturan *non-random chaos game* pada iterasi ke-5

- 6) Melakukan banyaknya pembentukan titik hingga iterasi yang diinginkan.

#### 4.1.7 Aturan Rotasi *Chaos Game* Segiempat

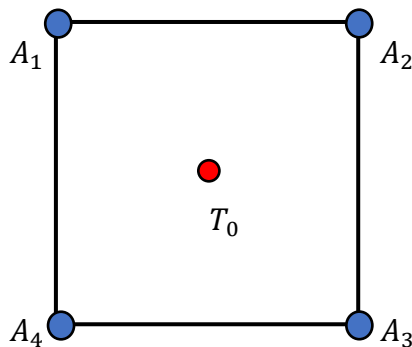
Misalkan diberikan titik-titik acuan  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$ . Proses rotasi dilakukan dengan titik tumpu rotasi berada pada titik acuan sudut segiempat, yaitu  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$  yang telah diberikan. Titik yang dipilih sebagai tumpuan berdasarkan aturan yang ingin digunakan, misalnya *chaos game* dengan aturan *random* tanpa syarat, maka titik sudut rotasinya acak mengikuti aturannya. Jika yang dipilih aturan modifikasi *random* yang telah diberikan syarat atau aturan *non-random*, maka titik tumpu sudut rotasinya juga mengikuti perintah aturannya. *Chaos game* dirotasi sesuai dengan sudut yang diinginkan sejak awal pembentukannya. Rotasi dengan sudut positif bergerak berlawanan dengan arah jarum jam, sedangkan rotasi dengan sudut negatif bergerak searah jarum jam. Proses pembentukannya ditunjukkan pada Gambar 4.36 sampai dengan 4.41.

- 1) Membuat sebuah segiempat dan memberi nama pada setiap titik sudut segiempat, misalnya  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$  seperti pada Gambar 4.36.



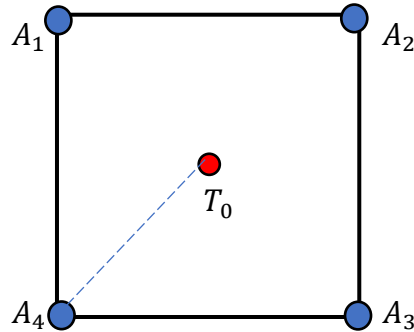
Gambar 4.36 Segiempat dengan titik sudut  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$

- 2) Memilih titik awal secara acak dan diberi nama  $T_0$  seperti pada Gambar 4.37.



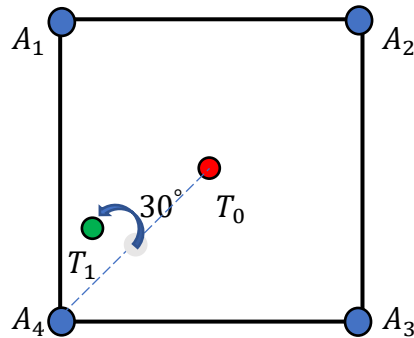
Gambar 4.37 Titik awal  $T_0$  di dalam segiempat

- 3) Memilih salah satu titik sudut pada segiempat untuk dihubungkan dengan titik awal  $T_0$  untuk membuat rotasi *chaos game* seperti pada Gambar 4.38.



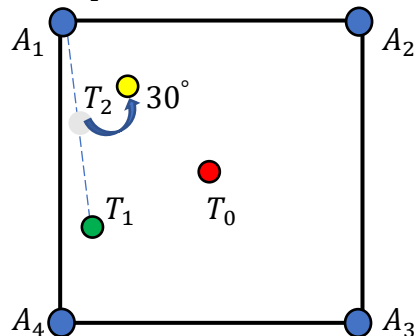
Gambar 4.38 Pemilihan titik sudut sebagai titik acuan secara acak

- 4) Menentukan titik baru yang diberi nama  $T_1$  dan jaraknya ( $\alpha$ ) adalah setengah jarak titik sudut ke titik awal sesuai rasio kompresi yang telah ditentukan, yaitu  $r = 2$ . Misalkan rotasi dilakukan dengan aturan *random* dengan sudut rotasi yang diberikan  $30^\circ$  dan titik tumpunya adalah  $A_4$ . Berikut prosesnya ditunjukkan pada Gambar 4.39.

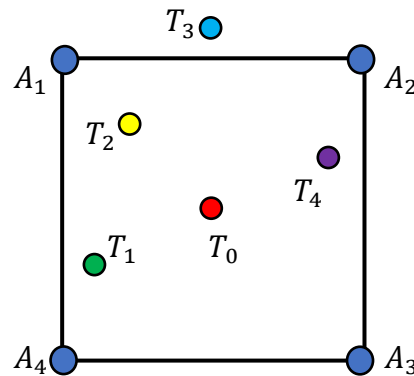


Gambar 4.39 Aturan rotasi *chaos game* secara *random* pada iterasi ke-1

- 5) Mengulang langkah (3) dengan titik awal adalah titik baru yang didapat dari iterasi sebelumnya dan boleh dilakukan perulangan pemilihan titik sudut sebagai titik acuan seperti Gambar 4.40 dan Gambar 4.41.



Gambar 4.40 Aturan rotasi *chaos game* secara *random* pada iterasi ke-2



Gambar 4.41 Aturan rotasi *chaos game* secara *random* pada iterasi ke-4

- 6) Melakukan banyaknya pembentukan titik hingga iterasi yang diinginkan.

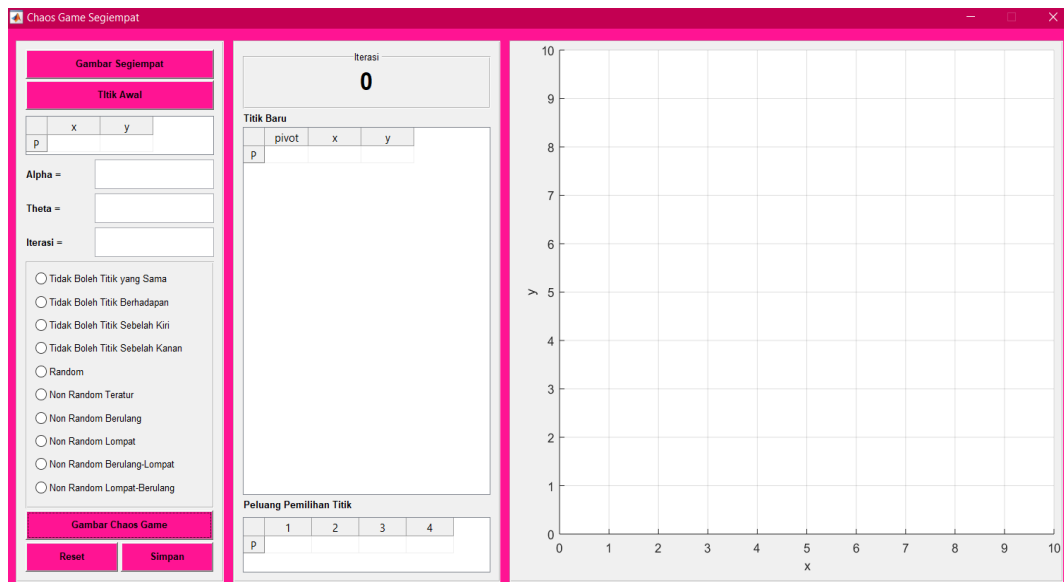
#### 4.2 Simulasi Program Menggunakan Aturan *Random* dan *Non-Random Chaos Game* dengan Variasi Sudut Rotasinya

Simulasi program pada penelitian ini dilakukan dengan menggunakan *software* MATLAB R2015b. Penelitian dilakukan dengan enam algoritma untuk memvisualisasikan bentuk objek yang akan dihasilkan untuk dianalisis apakah termasuk bentuk fraktal. Algoritma tersebut diantaranya menggunakan aturan *random*, modifikasi 1 aturan *random* dengan syarat titik terakhir yang telah dipilih tidak boleh dipilih kembali, modifikasi 2 aturan *random* dengan syarat titik yang berhadapan tidak boleh dipilih, modifikasi 3 aturan *random* dengan syarat titik sebelah kiri dari titik terakhir yang telah dipilih tidak boleh dipilih, modifikasi 4 aturan *random* dengan syarat titik sebelah kanan dari titik terakhir yang telah dipilih tidak boleh dipilih, serta lima macam aturan *non-random*.

Tampilan GUI MATLAB R2015b untuk melakukan simulasi program *chaos game* segiempat terdiri dari beberapa bagian. GUI terdiri dari *uipanel* “input”, *uipanel* “output”, *edit text*, *push button*, *radio button*, tabel dan *axes*. Masing-masing dari bagian tersebut memiliki fungsinya tersendiri untuk menjalankan program pembentukan *chaos game* segiempat beserta variasi rotasinya. Pada *uipanel* “input” terdapat beberapa bagian, yaitu *edit text* untuk menentukan jarak ( $\alpha$ ) titik pada *chaos game* segiempat, *edit text* untuk menentukan sudut rotasi ( $\theta$ ), *edit text* untuk menentukan iterasi, *push button* untuk menggambar segiempat, *push*

*button* untuk menentukan titik awal, *push button* untuk memulai proses *chaos game*, *push button* untuk *reset*, *push button* untuk menyimpan hasil *chaos game* dalam format gambar (jpg), *radio button* untuk menentukan aturan-aturan *chaos game* yaitu aturan *random*, modifikasi aturan *random* dengan syarat tidak boleh titik yang sama (titik terakhir yang telah dipilih tidak boleh dipilih kembali), modifikasi aturan *random* dengan syarat tidak boleh memilih titik yang berhadapan, modifikasi aturan *random* dengan syarat tidak boleh memilih titik sebelah kiri, serta modifikasi aturan *random* dengan syarat tidak boleh memilih titik sebelah kanan, modifikasi aturan *non-random* 1 dengan 4 pola pemilihan titik yang teratur dan berurutan ( $A_1 - A_2 - A_3 - A_4$ ), *non-random* 2 dengan 16 pola pemilihan titik, yaitu pada kelipatan 4 berulang dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_4 - A_1 - A_2 - A_2 - A_3 - A_4 - A_1 - A_1 - A_2 - A_3 - A_4 - A_4$ ), *non-random* 3 dengan 20 pola pemilihan titik, yaitu pada kelipatan 5 melompati satu titik setelahnya dan pada iterasi  $t$  yaitu  $t \equiv 4 \text{ mod } (5)$  titik yang dipilih adalah titik yang berhadapan dengan titik terakhirnya ( $A_1 - A_2 - A_3 - A_1 - A_3 - A_4 - A_1 - A_2 - A_4 - A_2 - A_3 - A_4 - A_1 - A_3 - A_1 - A_2 - A_3 - A_4 - A_2 - A_4$ ), *non-random* 4 dengan 24 pola pemilihan titik, yaitu pada kelipatan 6 dan pada iterasi  $t$  yaitu  $t \equiv 5 \text{ mod } (6)$  berulang satu kali seperti titik sebelumnya, kemudian pada  $t \equiv 4 \text{ mod } (6)$  melompati satu titik setelah titik terakhirnya ( $A_1 - A_2 - A_3 - A_3 - A_3 - A_1 - A_2 - A_3 - A_4 - A_4 - A_4 - A_2 - A_3 - A_4 - A_1 - A_1 - A_1 - A_3 - A_4 - A_1 - A_2 - A_2 - A_2 - A_4$ ), dan *non-random* 5 dengan 28 pola pemilihan titik, pada iterasi  $t$  yaitu  $t \equiv 4 \text{ mod } (7)$  melompati satu titik dari titik terakhir, pada iterasi  $t \equiv 5 \text{ mod } (7)$  berulang seperti titik terakhir yang dipilih, pada iterasi  $t \equiv 6 \text{ mod } (7)$  melompati satu titik dari titik terakhir lagi, dan pada kelipatan 7 berulang lagi satu titik dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_1 - A_1 - A_3 - A_3 - A_4 - A_1 - A_2 - A_4 - A_1 - A_2 - A_4 - A_4 - A_2 - A_2 - A_3 - A_4 - A_1 - A_3 - A_3 - A_1 - A_1 - A_2 - A_3 - A_4 - A_2 - A_2 - A_4 - A_4$ ). Pada *ui panel* “*output*” terdapat tabel dan *axes* dimana tabel pertama adalah untuk menampilkan koordinat titik awal yang dipilih, tabel kedua untuk menampilkan koordinat titik baru yang dibentuk dalam proses *chaos game*, dimana *pivot* adalah titik tumpunya, serta tabel ketiga yang menunjukkan peluang

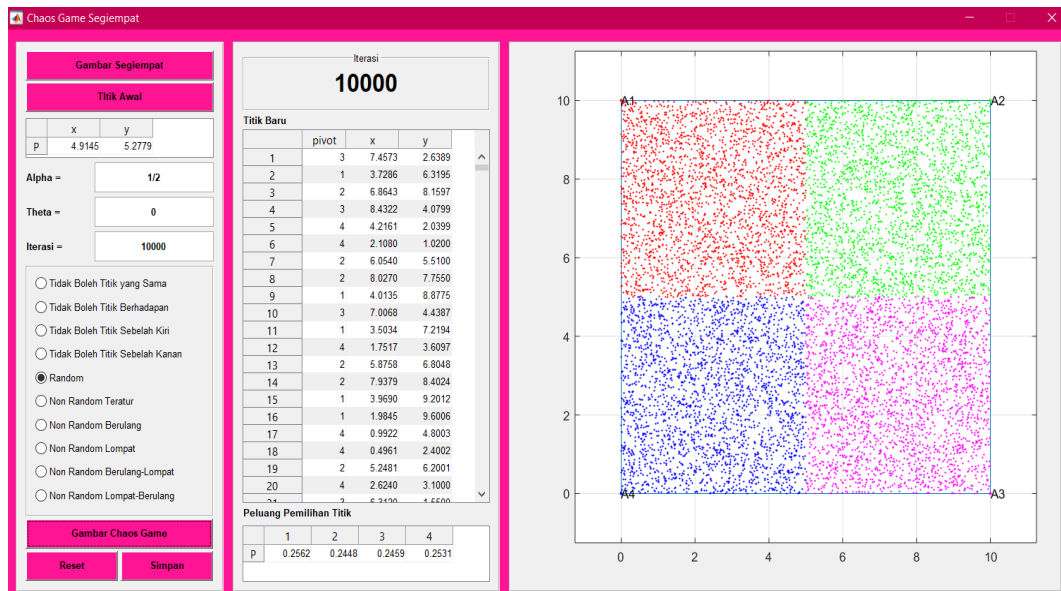
terpilihnya titik  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$ . Axes berfungsi untuk menampilkan gambar segiempat, membuat titik awal, serta memvisualisasikan hasil *chaos game* segiempat yang terbentuk sesuai dengan jarak ( $\alpha$ ), sudut ( $\theta$ ), iterasi, dan aturan *chaos game* yang dipilih. Berikut tampilan GUI MATLAB R2015b ditunjukkan pada Gambar 4.42.



Gambar 4.42 Tampilan GUI MATLAB R2015b

#### 4.2.1 Aturan *Random*

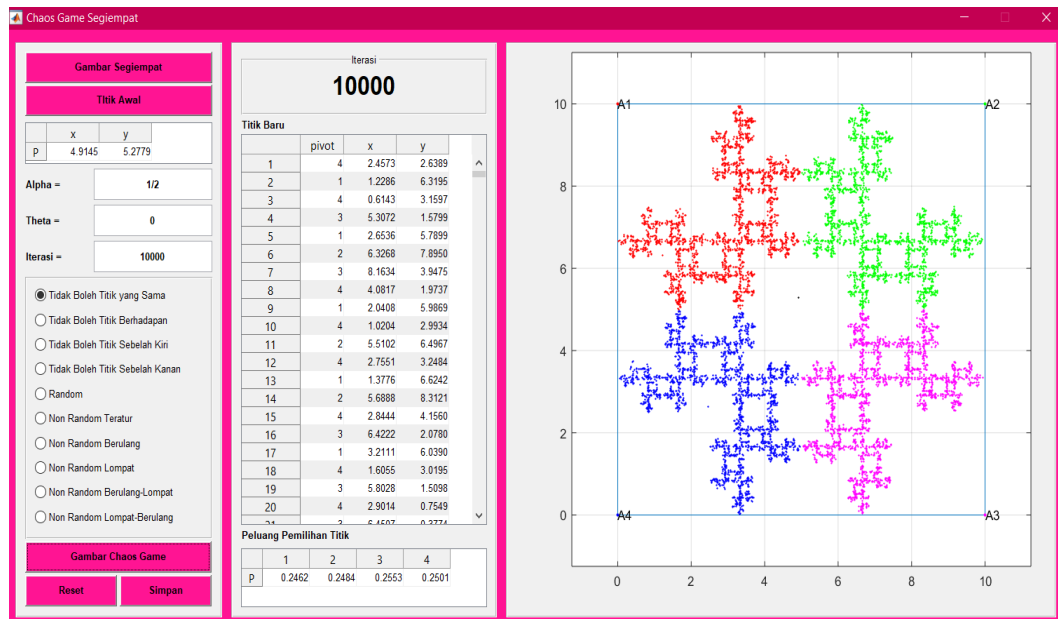
Langkah pertama yang dilakukan dalam simulasi program dengan menggunakan aturan *random* adalah menggambar segiempat dan menentukan titik awalnya. Selanjutnya menentukan jarak ( $\alpha$ ), sudut ( $\theta$ ), iterasi, dan aturan *chaos game* pada *radio button* aturan *random*. Jarak adalah  $1/r$ , dimana  $r$  adalah rasio kompresi yang telah ditentukan, yaitu 2. Maka jarak yang akan terbentuk pada *chaos game* adalah  $1/2$ , yang artinya  $\alpha$  adalah setengah jarak sudut ke titik awal. Sudut ( $\theta$ ) percobaan misalnya dilakukan pada nilai  $\theta = 0^\circ$ , dan iterasi yang diinginkan misalnya 10.000 iterasi agar hasilnya terlihat jelas. Visualisasi *Chaos game* segiempat yang dihasilkan seperti ditunjukkan pada Gambar 4.43.



Gambar 4.43 *Chaos game* segiempat dengan aturan *random* pada sudut  $0^\circ$

#### 4.2.2 Modifikasi 1: Titik Terakhir yang Telah Dipilih Tidak Boleh Dipilih Kembali

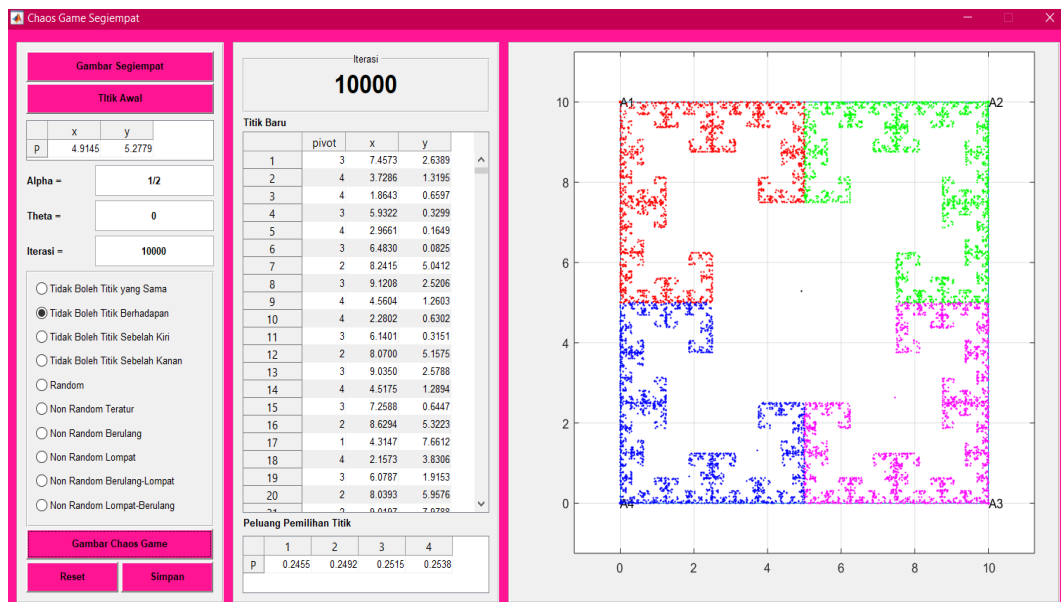
Langkah pertama yang dilakukan dalam simulasi program dengan menggunakan modifikasi 1 aturan *random* dengan syarat titik terakhir yang telah dipilih tidak boleh dipilih kembali pada dasarnya sama seperti sebelumnya, yaitu menggambar segiempat dan menentukan titik awalnya. Selanjutnya menentukan jarak ( $\alpha$ ), sudut ( $\theta$ ), iterasi, dan aturan *chaos game* pada *radio button* aturan *random* tidak boleh titik yang sama. Jarak adalah  $1/r$ , dimana  $r$  adalah rasio kompresi yang telah ditentukan, yaitu 2. Maka jarak yang akan terbentuk pada *chaos game* adalah  $1/2$ , yang artinya  $\alpha$  adalah setengah jarak titik sudut ke titik awal. Sudut ( $\theta$ ) percobaan misalnya dilakukan pada nilai  $\theta = 0^\circ$ , dan iterasi yang diinginkan misalnya 10.000 iterasi agar hasilnya terlihat jelas. Pemilihan titik sudutnya misalnya titik terakhir yang dipilih adalah titik  $A_2$ , maka iterasi selanjutnya titik yang boleh dipilih adalah titik  $A_1$ ,  $A_3$ ,  $A_4$ , begitu pula seterusnya. Bentuk *chaos game* segiempat yang dihasilkan dengan aturan ini seperti ditunjukkan pada Gambar 4.44.



Gambar 4.44 *Chaos game* segiempat dengan aturan modifikasi 1 pada sudut  $0^\circ$

#### 4.2.3 Modifikasi 2: Titik yang Berhadapan Tidak Boleh Dipilih

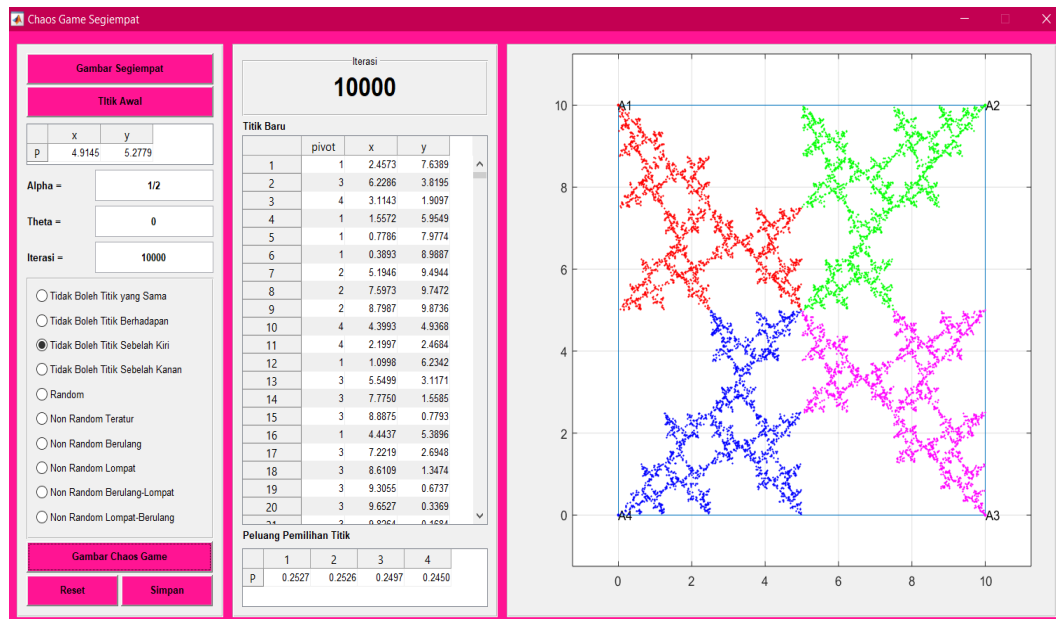
Langkah pertama yang dilakukan dalam simulasi program dengan menggunakan modifikasi 2 aturan *random* dengan syarat titik yang berhadapan tidak boleh dipilih pada dasarnya sama seperti sebelumnya, yaitu menggambar segiempat dan menentukan titik awalnya. Selanjutnya menentukan jarak ( $\alpha$ ), sudut ( $\theta$ ), iterasi, dan aturan *chaos game* pada *radio button* aturan *random* tidak boleh titik berhadapan. Jarak adalah  $1/r$ , dimana  $r$  adalah rasio kompresi yang telah ditentukan, yaitu 2. Maka jarak yang akan terbentuk pada *chaos game* adalah  $1/2$ , yang artinya  $\alpha$  adalah setengah jarak titik sudut ke titik awal. Sudut ( $\theta$ ) percobaan misalnya dilakukan pada nilai  $\theta = 0^\circ$ , dan iterasi yang diinginkan misalnya 10.000 iterasi agar hasilnya terlihat jelas. Pemilihan titik sudutnya misalnya titik yang berhadapan dari titik terakhir yang telah dipilih tidak boleh dipilih kembali. Misalkan titik terakhir yang dipilih adalah titik  $A_4$ . Titik  $A_4$  berhadapan dengan titik  $A_2$ , maka iterasi selanjutnya titik  $A_2$  tidak boleh dipilih. Titik selanjutnya yang boleh dipilih yaitu titik  $A_1$ ,  $A_3$ , atau  $A_4$ , begitu pula seterusnya dilihat dari titik yang berhadapannya. Bentuk *chaos game* segiempat yang dihasilkan dengan aturan ini seperti ditunjukkan pada Gambar 4.45.



Gambar 4.45 *Chaos game* segiempat dengan aturan modifikasi 2 pada sudut  $0^\circ$

#### 4.2.4 Modifikasi 3: Titik Sebelah Kiri Tidak Boleh Dipilih

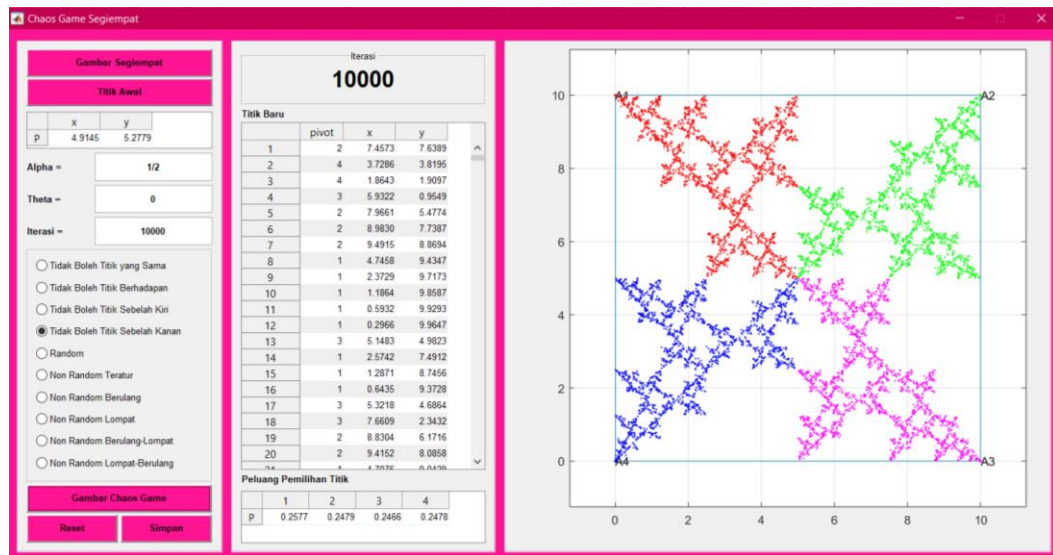
Langkah pertama yang dilakukan dalam simulasi program dengan menggunakan modifikasi 3 aturan *random* dengan syarat titik sebelah kiri tidak boleh dipilih pada dasarnya sama seperti sebelumnya, yaitu menggambar segiempat dan menentukan titik awalnya. Selanjutnya menentukan jarak ( $\alpha$ ), sudut ( $\theta$ ), iterasi, dan aturan *chaos game* pada *radio button* aturan *random* tidak boleh titik sebelah kiri. Jarak adalah  $1/r$ , dimana  $r$  adalah rasio kompresi yang telah ditentukan, yaitu 2. Maka jarak yang akan terbentuk pada *chaos game* adalah  $1/2$ , yang artinya  $\alpha$  adalah setengah jarak titik sudut ke titik awal. Sudut ( $\theta$ ) percobaan misalnya dilakukan pada nilai  $\theta = 0^\circ$ , dan iterasi yang diinginkan misalnya 10.000 iterasi agar hasilnya terlihat jelas. Pemilihan titik sudutnya misalkan titik terakhir yang dipilih adalah titik  $A_2$ . Titik  $A_2$  sebelah kirinya (sebelumnya) adalah titik  $A_1$ , maka iterasi selanjutnya titik  $A_1$  tidak boleh dipilih, begitu pula seterusnya. Pengecualian pada titik  $A_3$  dimana titik sebelah kirinya adalah titik  $A_4$ , maka setelah memilih titik  $A_3$  tidak boleh memilih titik  $A_4$ . Bentuk *chaos game* segiempat yang dihasilkan dengan aturan ini seperti ditunjukkan pada Gambar 4.46.



Gambar 4.46 *Chaos game* segiempat dengan aturan modifikasi 3 pada sudut  $0^\circ$

#### 4.2.5 Modifikasi 4: Titik Sebelah Kanan Tidak Boleh Dipilih

Langkah pertama yang dilakukan dalam simulasi program dengan menggunakan modifikasi 3 aturan *random* dengan syarat titik sebelah kanan tidak boleh dipilih pada dasarnya sama seperti sebelumnya, yaitu menggambar segiempat dan menentukan titik awalnya. Selanjutnya menentukan jarak ( $\alpha$ ), sudut ( $\theta$ ), iterasi, dan aturan *chaos game* pada *radio button* aturan *random* tidak boleh titik sebelah kanan. Jarak adalah  $1/r$ , dimana  $r$  adalah rasio kompresi yang telah ditentukan, yaitu 2. Maka jarak yang akan terbentuk pada *chaos game* adalah  $1/2$ , yang artinya  $\alpha$  adalah setengah jarak titik sudut ke titik awal. Sudut ( $\theta$ ) percobaan misalnya dilakukan pada nilai  $\theta = 0^\circ$ , dan iterasi yang diinginkan misalnya 10.000 iterasi agar hasilnya terlihat jelas. Pemilihan titik sudutnya berkebalikan dengan cara pemilihan titik sudut sebelah kiri tidak boleh dipilih pada subbab sebelumnya, yaitu misalkan titik terakhir yang dipilih adalah titik  $A_1$ . Titik  $A_1$  sebelah kanannya (setelahnya) adalah titik  $A_2$ , maka iterasi selanjutnya titik  $A_2$  tidak boleh dipilih, begitu pula seterusnya. Pengecualian pada titik  $A_4$  dimana titik sebelah kanannya adalah titik  $A_3$ , maka setelah memilih titik  $A_4$  tidak boleh memilih titik  $A_3$ . Bentuk *chaos game* segiempat yang dihasilkan dengan aturan ini seperti ditunjukkan pada Gambar 4.47.

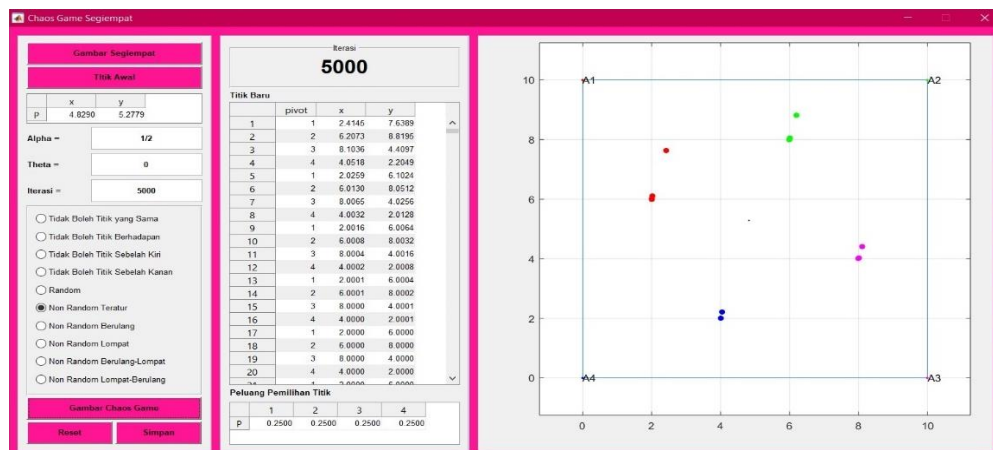


Gambar 4.47 *Chaos game* segiempat dengan aturan modifikasi 4 pada sudut  $0^\circ$

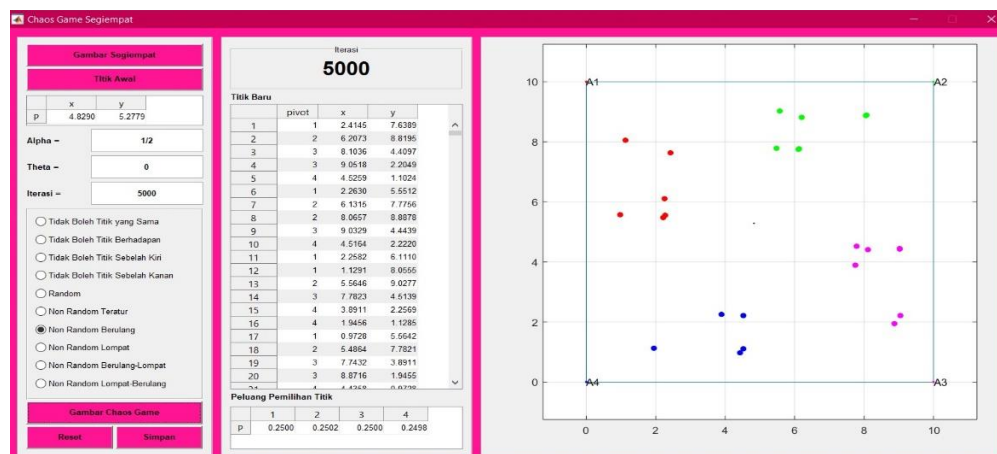
#### 4.2.6 Aturan *Non-Random*

Langkah pertama yang dilakukan dalam simulasi program dengan menggunakan aturan *non-random* pada dasarnya sama seperti sebelumnya, yaitu menggambar segiempat dan menentukan titik awalnya, menentukan jarak ( $\alpha$ ), sudut ( $\theta$ ), iterasi, dan aturan *chaos game* pada *radio button* aturan *non-random*. Jarak adalah  $1/r$ , dimana  $r$  adalah rasio kompresi yang telah ditentukan, yaitu 2. Maka jarak yang akan terbentuk pada *chaos game* adalah  $1/2$ , yang artinya  $\alpha$  adalah setengah jarak titik sudut ke titik awal. Sudut ( $\theta$ ) percobaan misalnya dilakukan pada nilai  $\theta = 0^\circ$ , dan iterasi yang diinginkan misalnya 5.000 iterasi. Aturan *non-random* dilakukan dengan beberapa modifikasi, yaitu *non-random 1* dengan 4 pola pemilihan titik yang teratur dan berurutan ( $A_1 - A_2 - A_3 - A_4$ ), *non-random 2* dengan 16 pola pemilihan titik, yaitu pada kelipatan 4 berulang dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_3 - A_4 - A_1 - A_2 - A_2 - A_3 - A_4 - A_1 - A_1 - A_2 - A_3 - A_4 - A_4$ ), *non-random 3* dengan 20 pola pemilihan titik, yaitu pada kelipatan 5 melompati satu titik setelahnya dan pada iterasi  $t$  yaitu  $t \equiv 4 \pmod{5}$  titik yang dipilih adalah titik yang berhadapan dengan titik terakhirnya ( $A_1 - A_2 - A_3 - A_1 - A_3 - A_4 - A_1 - A_2 - A_4 - A_2 - A_3 - A_4 - A_1 - A_3 - A_1 - A_2 - A_3 - A_4 - A_2 - A_4$ ), *non-random 4* dengan 24 pola pemilihan titik, yaitu pada kelipatan 6 dan pada iterasi  $t$  yaitu  $t \equiv 5 \pmod{6}$

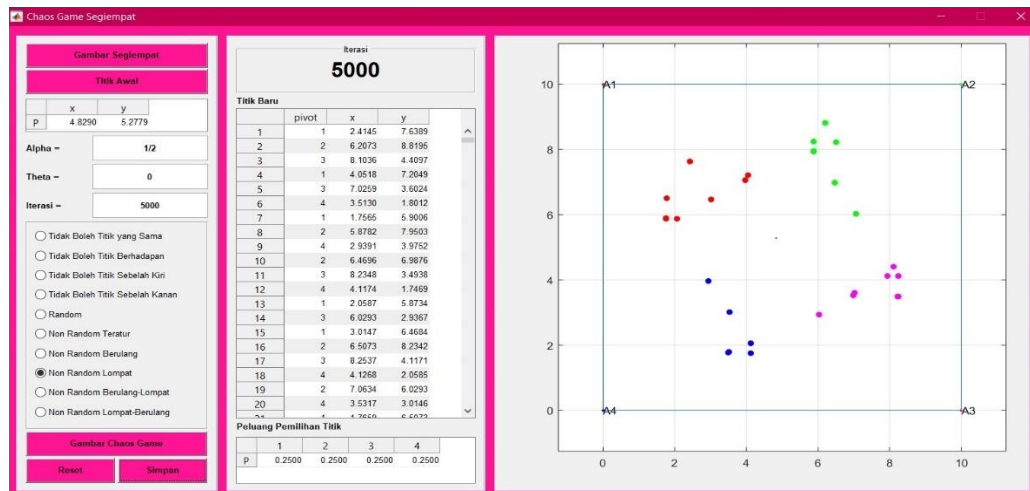
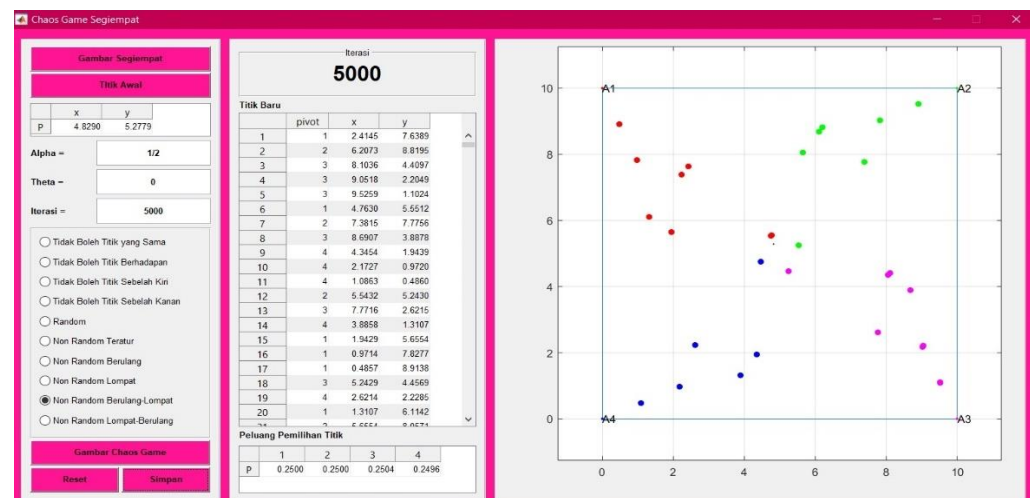
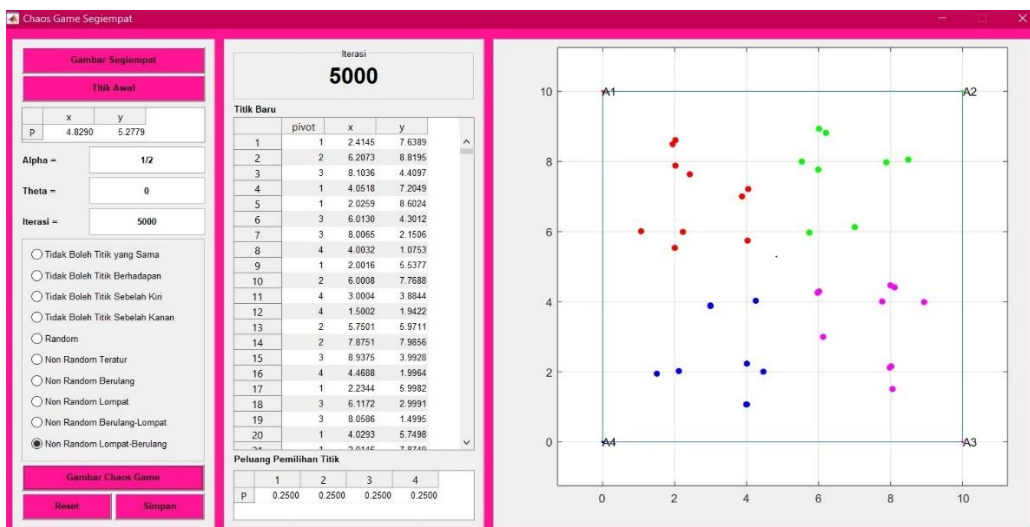
berulang satu kali seperti titik sebelumnya, kemudian pada  $t \equiv 4 \bmod (6)$  melompati satu titik setelah titik terakhirnya ( $A_1 - A_2 - A_3 - A_3 - A_3 - A_1 - A_2 - A_3 - A_4 - A_4 - A_4 - A_2 - A_3 - A_4 - A_1 - A_1 - A_1 - A_3 - A_4 - A_1 - A_2 - A_2 - A_2 - A_4$ ), dan *non-random* 5 dengan 28 pola pemilihan titik, pada iterasi  $t$  yaitu  $t \equiv 4 \bmod (7)$  melompati satu titik dari titik terakhir, pada iterasi  $t \equiv 5 \bmod (7)$  berulang seperti titik terakhir yang dipilih, pada iterasi  $t \equiv 6 \bmod (7)$  melompati satu titik dari titik terakhir lagi, dan pada kelipatan 7 berulang lagi satu titik dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_1 - A_1 - A_3 - A_3 - A_4 - A_1 - A_2 - A_4 - A_1 - A_2 - A_4 - A_4 - A_2 - A_2 - A_3 - A_4 - A_1 - A_3 - A_3 - A_1 - A_1 - A_2 - A_3 - A_4 - A_2 - A_2 - A_4 - A_4$ ). Bentuk *chaos game* segiempat yang dihasilkan seperti ditunjukkan pada Gambar 4.48 sampai dengan 4.52.



Gambar 4.48 *Chaos game* segiempat dengan aturan *non-random* 1 pada sudut  $0^\circ$



Gambar 4.49 *Chaos game* segiempat dengan aturan *non-random* 2 pada sudut  $0^\circ$

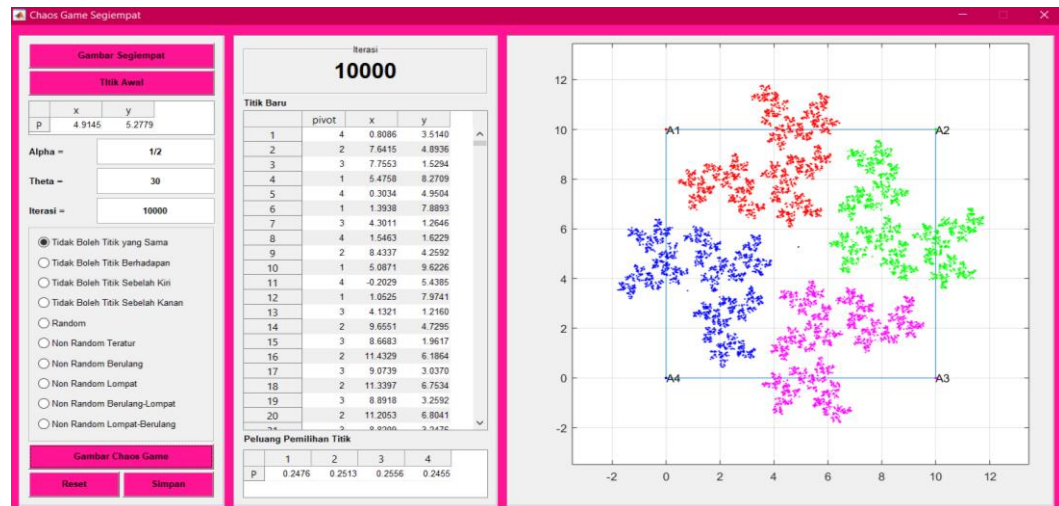
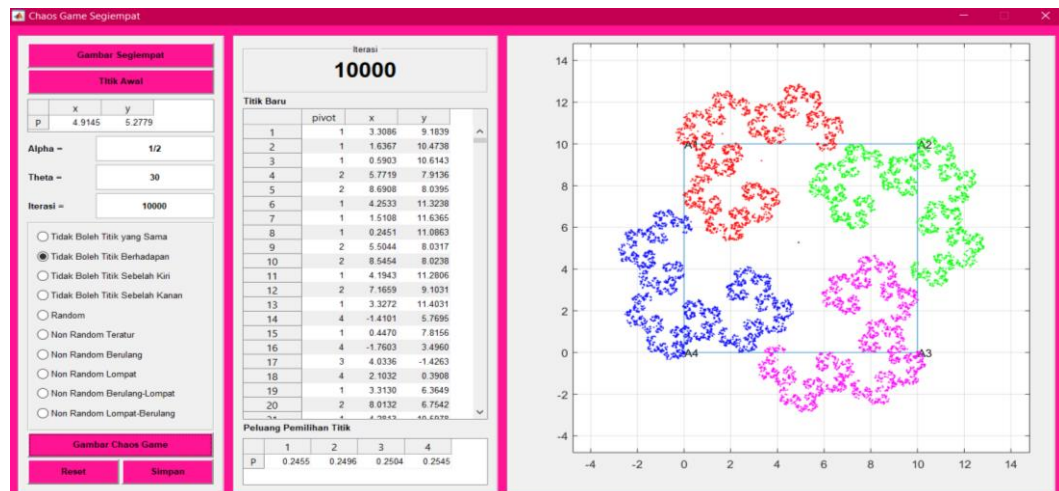
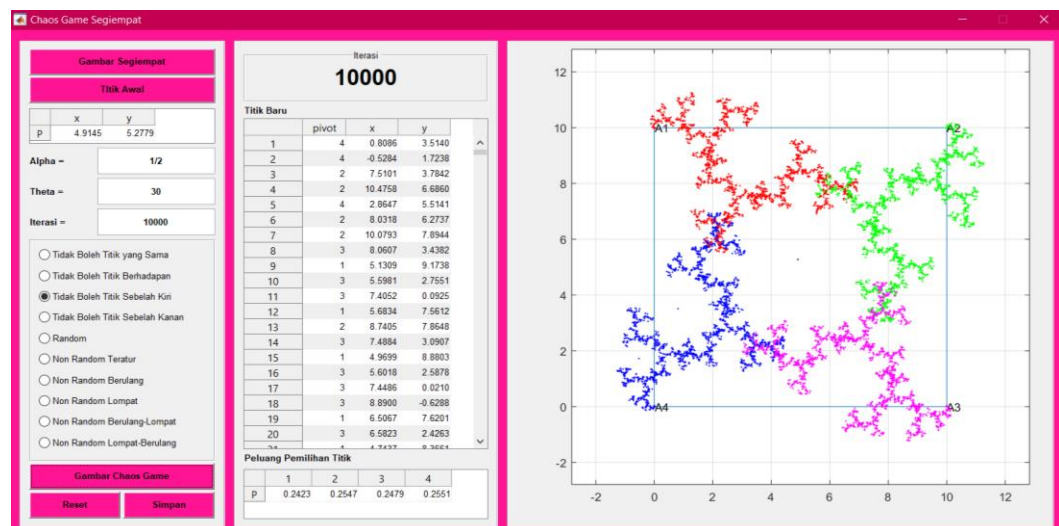
Gambar 4.50 Chaos game segiempat dengan aturan non-random 3 pada sudut  $0^\circ$ Gambar 4.51 Chaos game segiempat dengan aturan non-random 4 pada sudut  $0^\circ$ Gambar 4.52 Chaos game segiempat dengan aturan non-random 5 pada sudut  $0^\circ$

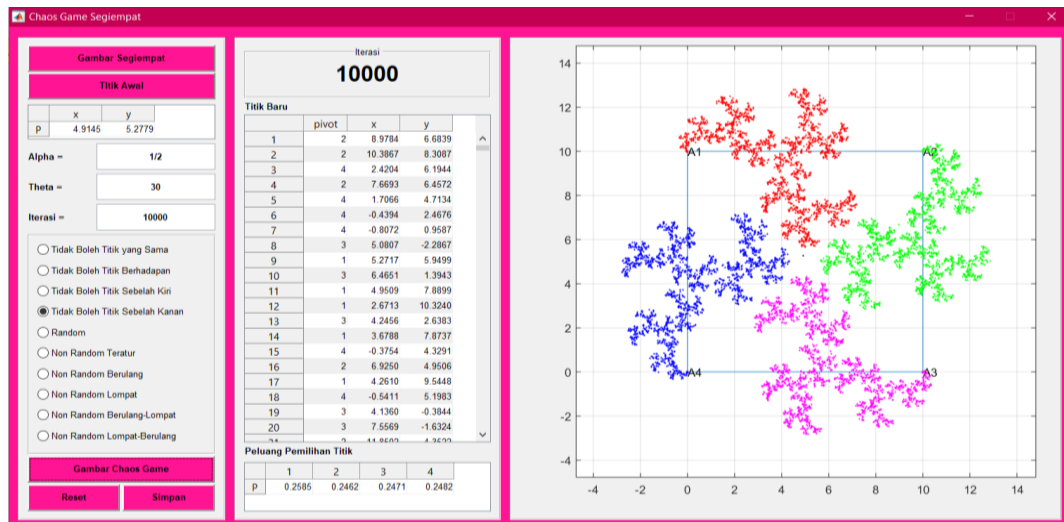
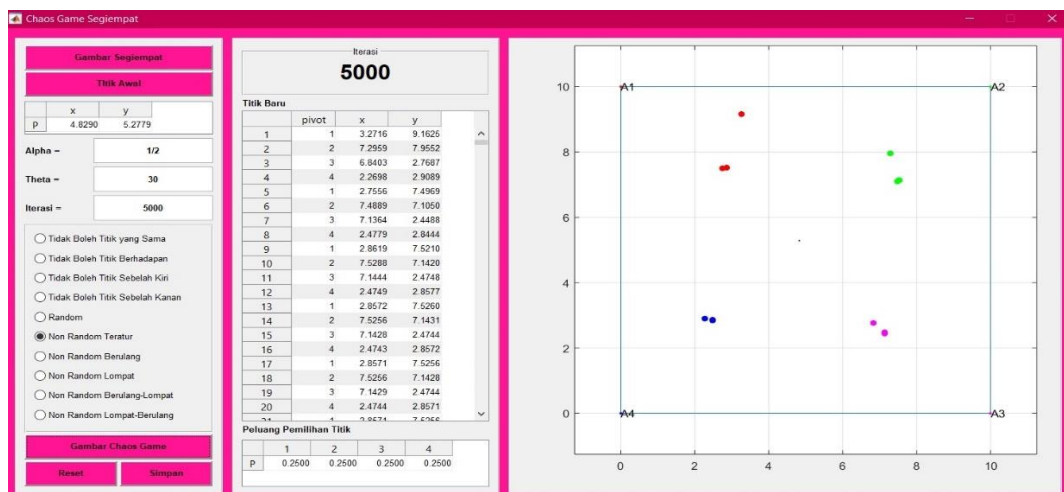
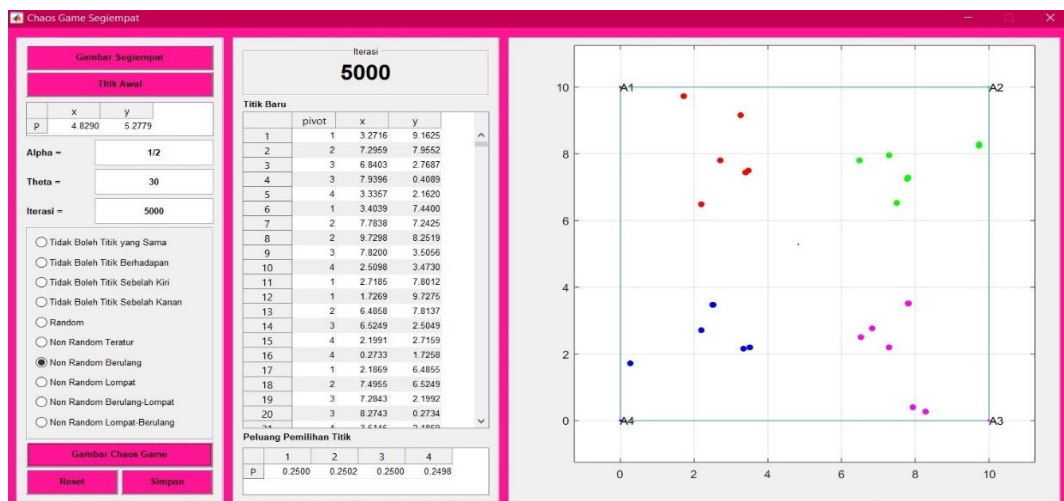
#### 4.2.7 Aturan Rotasi *Chaos Game* Segiempat

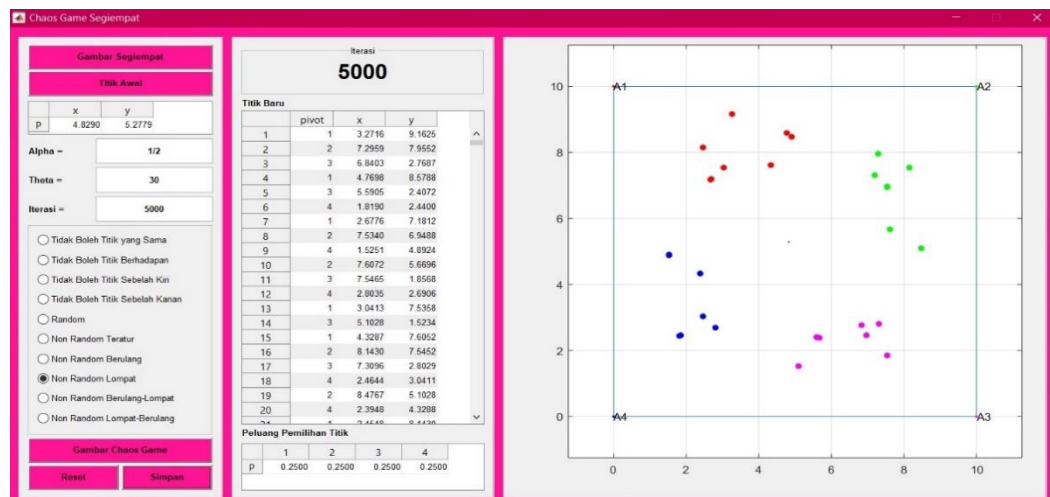
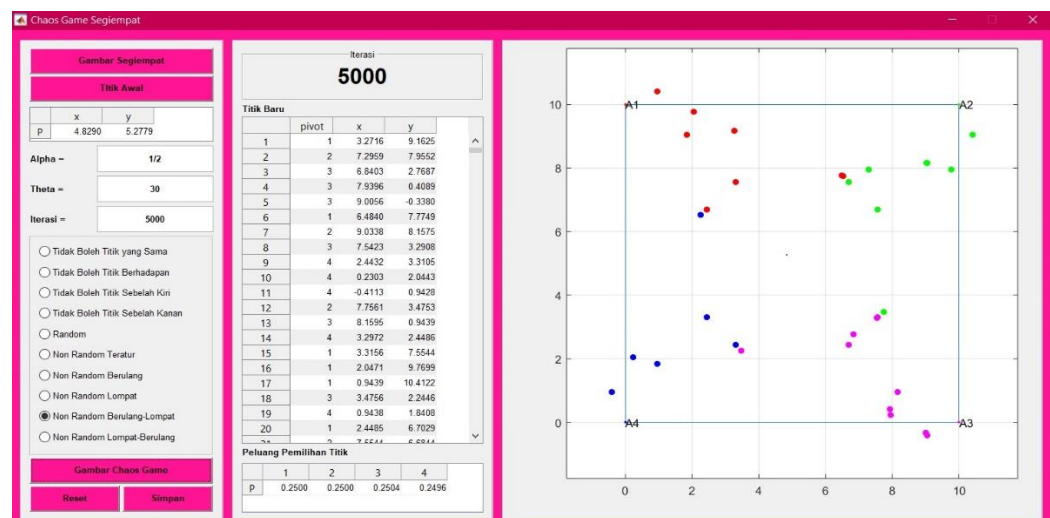
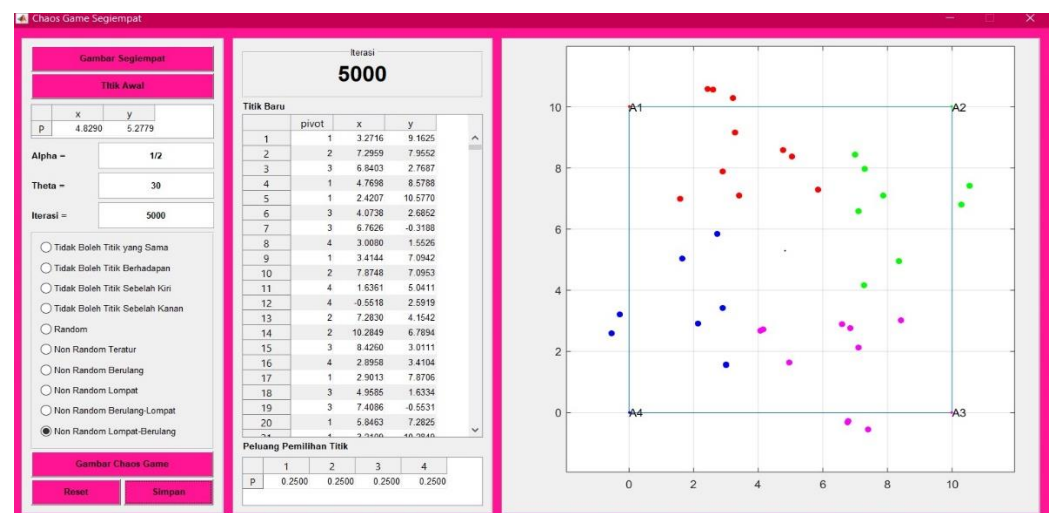
Langkah pertama yang dilakukan dalam simulasi program dengan menggunakan aturan rotasi *chaos game* segiempat adalah menggambar segiempat dan menentukan titik awalnya. Selanjutnya sama seperti aturan-aturan yang telah dijelaskan sebelumnya, yaitu menentukan jarak ( $\alpha$ ), sudut ( $\theta$ ), iterasi, dan aturan *chaos game* pada *radio button* aturan *random*. Jarak adalah  $1/r$ , dimana  $r$  adalah rasio kompresi yang telah ditentukan, yaitu 2. Maka jarak yang akan terbentuk pada *chaos game* adalah  $1/2$ , yang artinya  $\alpha$  adalah setengah jarak titik sudut ke titik awal. Sudut ( $\theta$ ) rotasi dilakukan pada nilai sudut positif dan sudut negatif dengan beberapa variasi perputarannya. Rotasi dilakukan dengan aturan *random* beserta empat modifikasinya, serta aturan *non-random* dengan lima aturan modifikasi. Bentuk yang dihasilkan digunakan untuk analisis apakah termasuk bentuk fraktal seperti yang dilakukan pada segitiga Sierpinski ataukah tidak membentuk fraktal. Pada simulasi ini misalnya dilakukan pada nilai  $\theta = 30^\circ$  untuk semua aturan *chaos game*, dan iterasi yang diinginkan misalnya 10.000 iterasi agar hasilnya terlihat jelas. Maka rotasi *chaos game* segiempat yang dihasilkan seperti ditunjukkan pada Gambar 4.53 sampai dengan Gambar 4.62.



Gambar 4.53 *Chaos game* segiempat dengan aturan *random* pada sudut  $30^\circ$

Gambar 4.54 *Chaos game* segiempat dengan aturan modifikasi 1 pada sudut  $30^\circ$ Gambar 4.55 *Chaos game* segiempat dengan aturan modifikasi 2 pada sudut  $30^\circ$ Gambar 4.56 *Chaos game* segiempat dengan aturan modifikasi 3 pada sudut  $30^\circ$

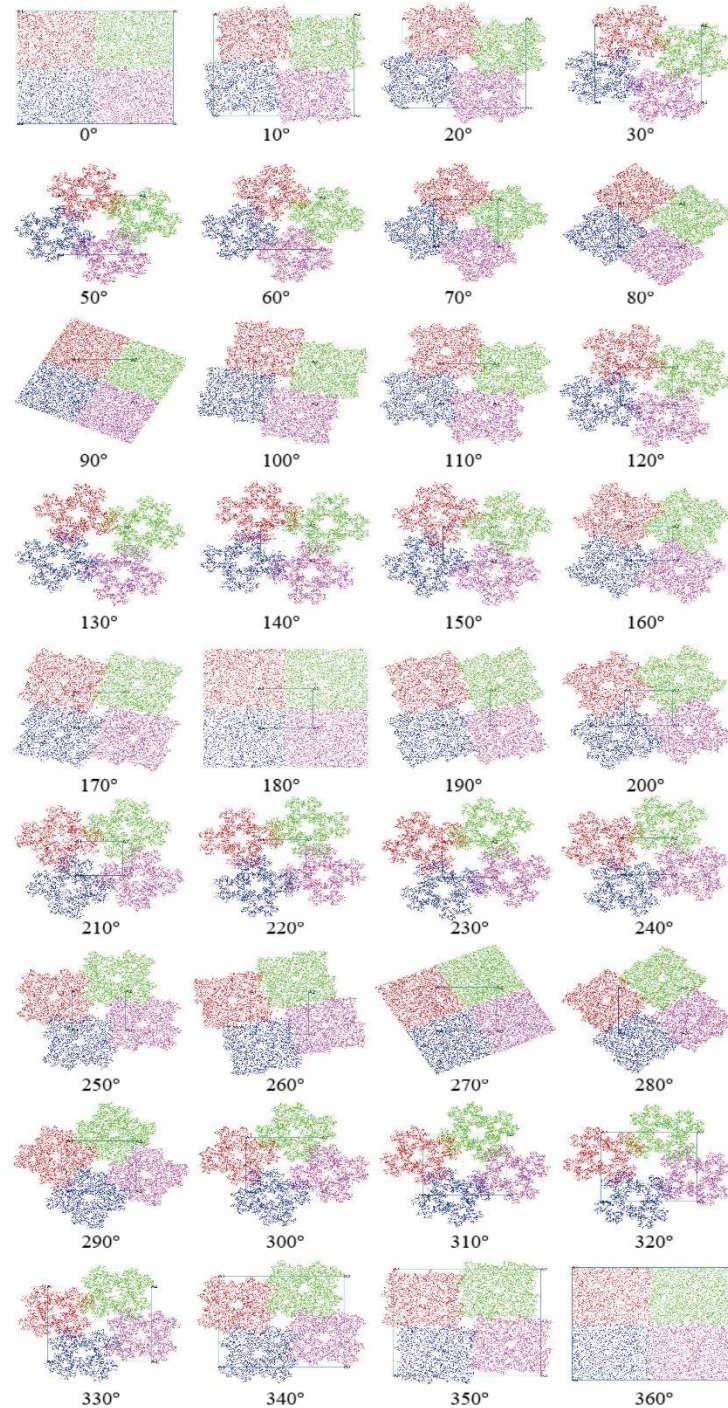
Gambar 4.57 Chaos game segiempat dengan aturan modifikasi 4 pada sudut  $30^\circ$ Gambar 4.58 Chaos game segiempat dengan aturan non-random 1 pada sudut  $30^\circ$ Gambar 4.59 Chaos game segiempat dengan aturan non-random 2 pada sudut  $30^\circ$

Gambar 4.60 Chaos game segiempat dengan aturan non-random 3 pada sudut  $30^\circ$ Gambar 4.61 Chaos game segiempat dengan aturan non-random 4 pada sudut  $30^\circ$ Gambar 4.62 Chaos game segiempat dengan aturan non-random 5 pada sudut  $30^\circ$

### 4.3 Pembahasan

#### 4.3.1 Aturan *Random*

Hasil visualisasi rotasi *chaos game* segiempat dengan perputaran sudut setiap  $10^\circ$ ,  $a = 1/2$  dan iterasi 10.000 ditunjukkan pada Gambar 4.63.

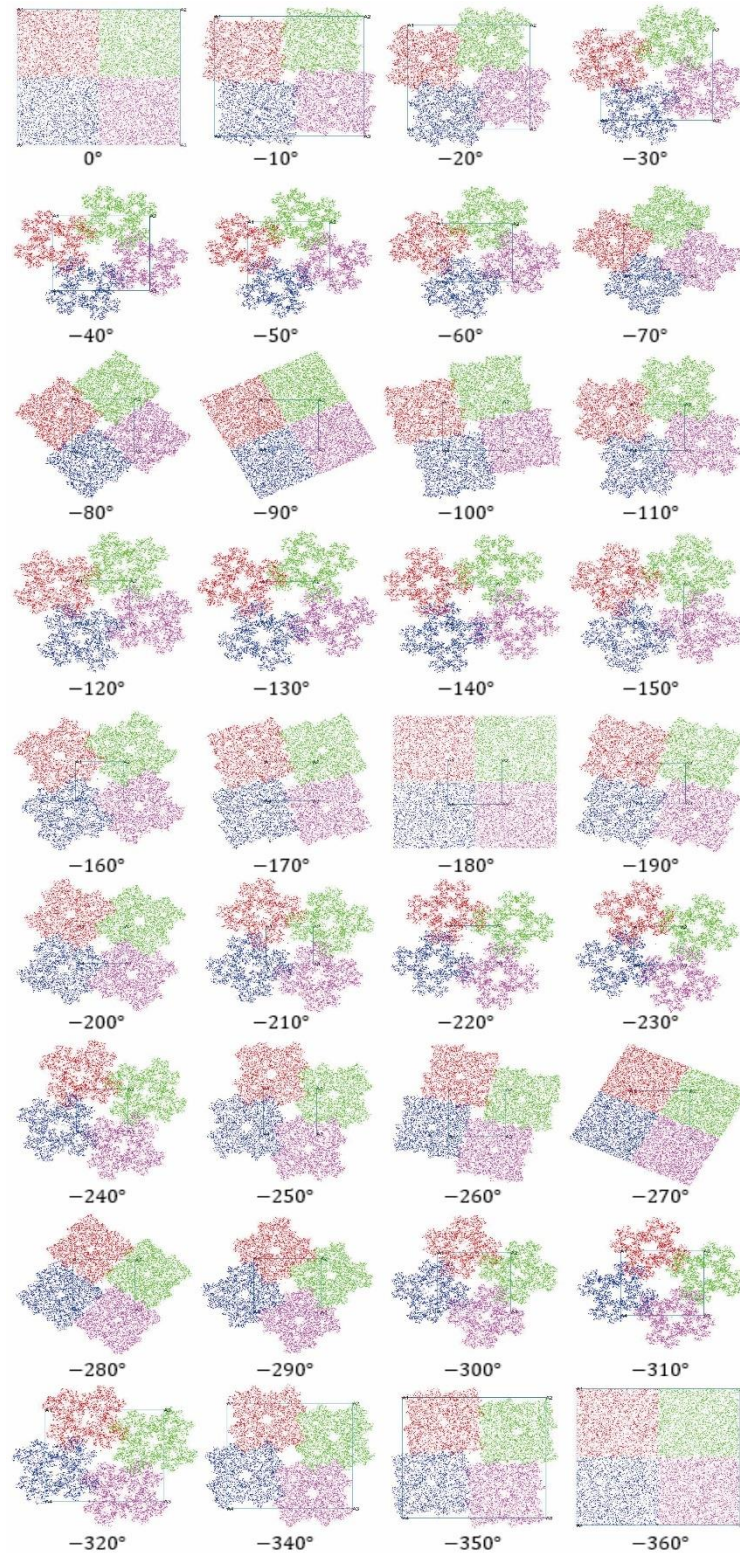


Gambar 4.63 Variasi rotasi sudut positif dengan aturan *random*

Pada visualisasi variasi rotasi *chaos game* di atas, diketahui bahwa titik-titik hasil bentukan *chaos game* segiempat menyerupai bangun segiempat dan menyebar pada masing-masing titik sudutnya. Hasil bentukan variasi rotasinya pada keseluruhannya menunjukkan bahwa di dalam segiempat terdapat bangun yang menyerupai segiempat yang lebih kecil dimana segiempat tersebut menyebar pada masing-masing titik sudut. Hasil perputaran dalam variasi rotasinya dapat dikatakan memiliki bentuk yang identik. Objek hasil bentukan *chaos game* dengan aturan *random* ini memiliki empat bagian warna yang berbeda ke setiap titik sudutnya yang digunakan sebagai titik tumpu untuk merotasikan *chaos game* segiempat. Kumpulan titik-titik yang berwarna merah merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_1$ , titik-titik yang berwarna hijau merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_2$ , titik-titik yang berwarna biru merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_3$ , serta titik-titik yang berwarna *magenta* merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_4$ . Pemilihan titik *random* pada simulasi program ini didasarkan pada perintah “*pivot=ceil(rand\*4)*” pada *m-file software* MATLAB R2015b, yang artinya titik tumpu diacak dengan perintah *random* yang merupakan distribusi seragam (*uniform*) yang hasilnya akan berada diantara nilai 0 sampai dengan 4 dengan pembulatan ke atas (*ceil*), sehingga peluang terpilihnya titik  $A_1$ ,  $A_2$ ,  $A_3$ , dan  $A_4$  akan seimbang dan tidak berkumpul pada satu titik saja.

Berdasarkan variasi rotasi di atas, seluruh objek yang dihasilkan pada rotasi sudut positif dapat dikatakan sebagai objek fraktal karena memiliki kemiripan bentuk satu sama lain (*self-similarity*) dan titik-titik hasil bentukan yang dihasilkan *chaos game* dengan aturan *random* ini menyerupai bentuk segiempat yang menyebar pada setiap titik acuannya. Perputaran pada sudut negatifnya juga merupakan objek fraktal. Perbedaan antara perputaran sudut positif dan negatifnya berada pada arah rotasinya, dimana rotasi sudut positif berputar berlawanan arah jarum jam, sedangkan rotasi sudut negatif berputar searah jarum jam. Berikut

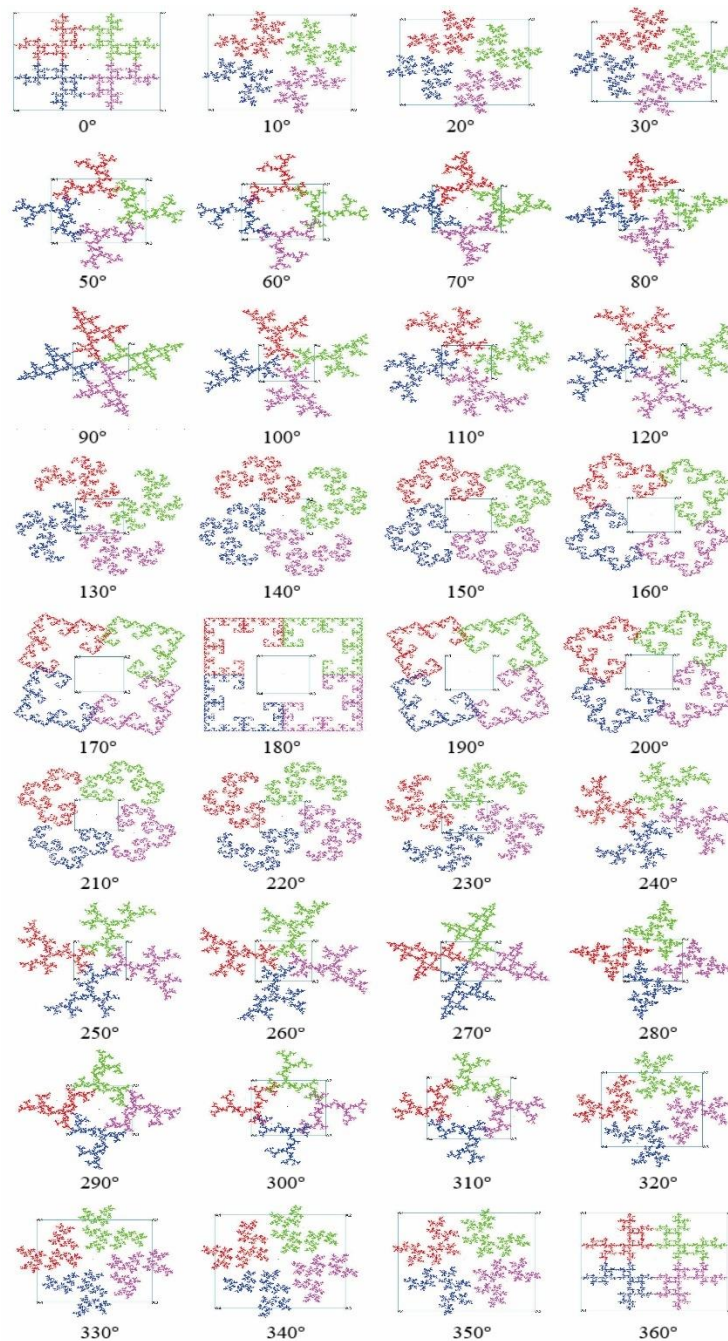
perputaran *chaos game* segiempat yang dilakukan pada sudut negatif setiap perputaran  $10^\circ$  dengan  $\alpha = 1/2$  dan iterasi 10.000 ditunjukkan pada Gambar 4.64.



Gambar 4.64 Variasi rotasi sudut negatif dengan aturan *random*

#### 4.3.2 Modifikasi 1: Titik Terakhir yang Telah Dipilih Tidak Boleh Dipilih Kembali

Subbab ini akan menunjukkan hasil bentukan *chaos game* segiempat yang aturannya telah dimodifikasi, yaitu titik terakhir yang telah dipilih tidak boleh dipilih kembali. Hasil visualisasi rotasi *chaos game* segiempat menggunakan aturan modifikasi 1 dengan perputaran sudut setiap  $10^\circ$ ,  $a = 1/2$  dan iterasi 10.000 ditunjukkan pada Gambar 4.65.

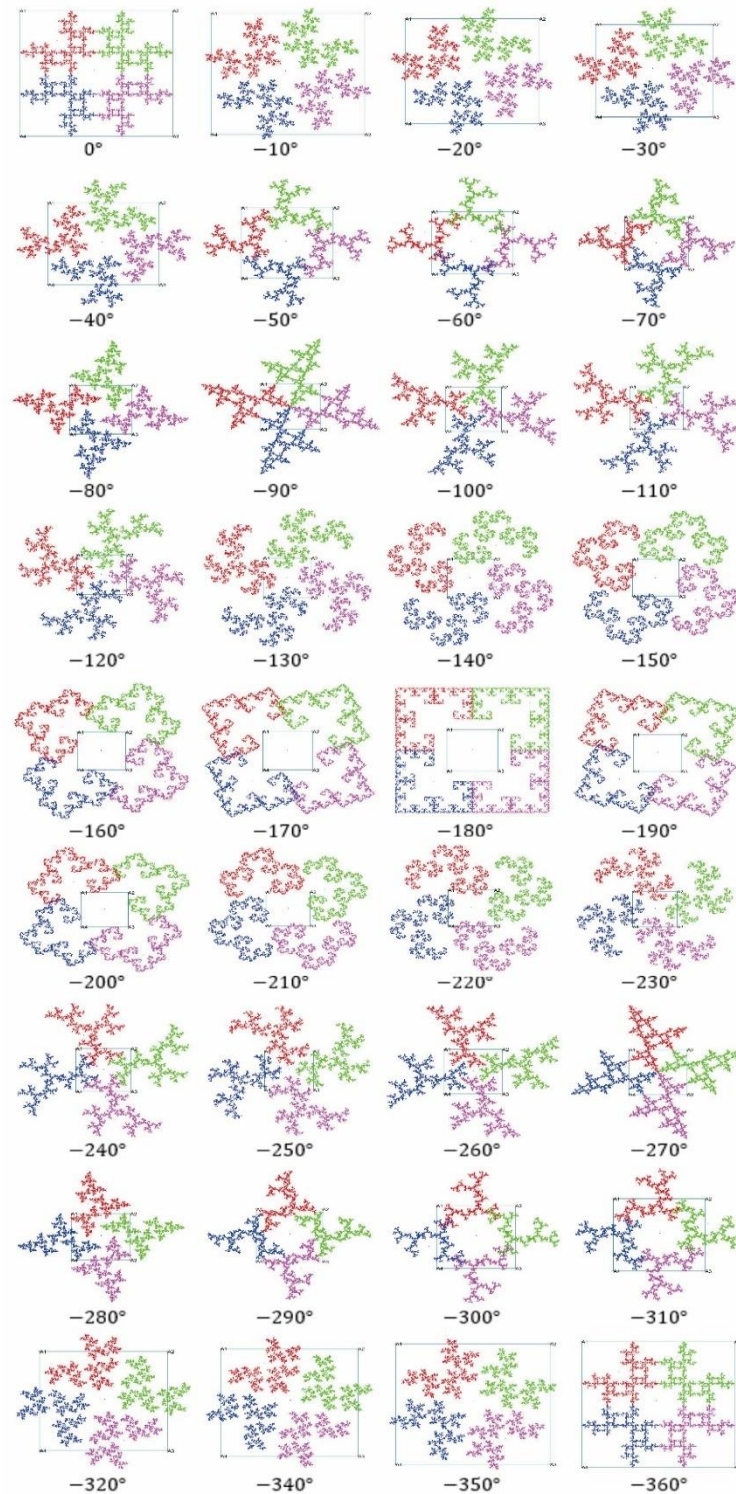


Gambar 4.65 Variasi rotasi sudut positif pada modifikasi 1 aturan *random*

Pada visualisasi variasi rotasi *chaos game* di atas, diketahui bahwa titik-titik hasil bentukan *chaos game* segiempat menyerupai bangun segiempat dan menyebar pada masing-masing titik sudutnya. Hasil bentukan variasi rotasinya pada keseluruhannya mirip dengan aturan sebelumnya, yaitu menunjukkan bahwa di dalam segiempat terdapat bangun yang menyerupai segiempat yang lebih kecil dimana segiempat tersebut menyebar pada masing-masing titik sudut. Hasil perputaran dalam variasi rotasinya dapat dikatakan memiliki bentuk yang identik seperti hasil variasi *chaos game* yang dilakukan dengan aturan *random*. Objek hasil bentukan *chaos game* dengan modifikasi 1 aturan *random* ini juga memiliki empat bagian warna yang berbeda ke setiap titik sudutnya yang digunakan sebagai titik tumpu untuk merotasikan *chaos game* segiempat. Kumpulan titik-titik yang berwarna merah merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_1$ , titik-titik yang berwarna hijau merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_2$ , titik-titik yang berwarna biru merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_3$ , serta titik-titik yang berwarna *magenta* merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_4$ . Pemilihan titik untuk simulasi program ini telah ditunjukkan contohnya pada Gambar 4.10 dan 4.11.

Berdasarkan variasi rotasi di atas, seluruh objek yang dihasilkan pada rotasi sudut positif dapat dikatakan sebagai objek fraktal karena memiliki kemiripan bentuk satu sama lain (*self-similarity*) dan titik-titik hasil bentukan yang dihasilkan *chaos game* dengan modifikasi 1 aturan *random* ini menyerupai bentuk segiempat yang menyebar pada setiap titik acuannya, sama seperti halnya pada aturan *random*. Perputaran pada sudut negatifnya juga merupakan objek fraktal karena juga memiliki bentuk yang identik. Perbedaan antara perputaran sudut positif dan negatifnya pun sama seperti aturan sebelumnya, yaitu berada pada arah rotasinya.

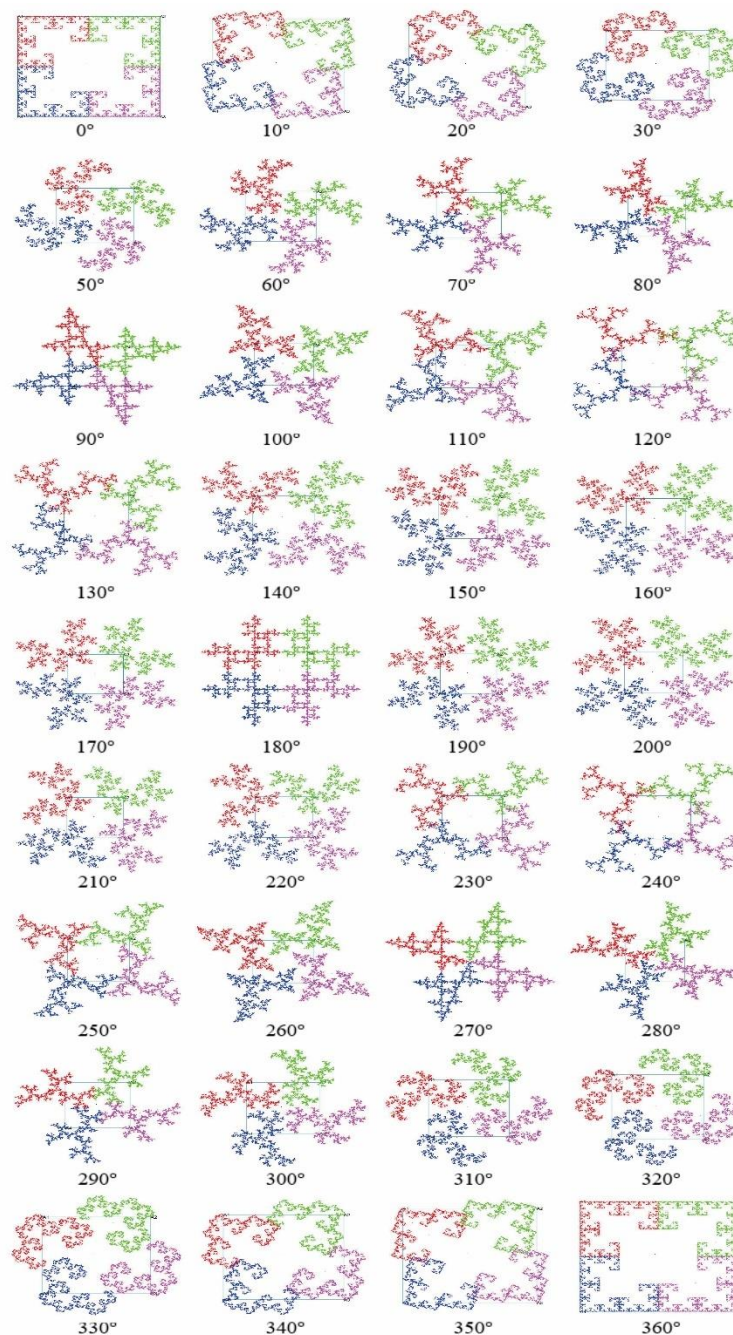
Berikut perputaran *chaos game* segiempat menggunakan modifikasi 1 aturan *random* yang dilakukan pada sudut negatif setiap perputaran  $10^\circ$  dengan  $\alpha = 1/2$  dan iterasi 10.000 ditunjukkan pada Gambar 4.66.



Gambar 4.66 Variasi rotasi sudut negatif pada modifikasi 1 aturan *random*

#### 4.3.3 Modifikasi 2: Titik yang Berhadapan Tidak Boleh Dipilih

Subbab ini akan menunjukkan hasil bentukan *chaos game* segiempat dengan modifikasi 2 aturan *random*, yaitu titik yang berhadapan tidak boleh dipilih. Hasil visualisasi rotasi *chaos game* segiempat menggunakan aturan modifikasi 2 dengan perputaran sudut setiap  $10^\circ$ ,  $a = 1/2$  dan iterasi 10.000 ditunjukkan pada Gambar 4.67.

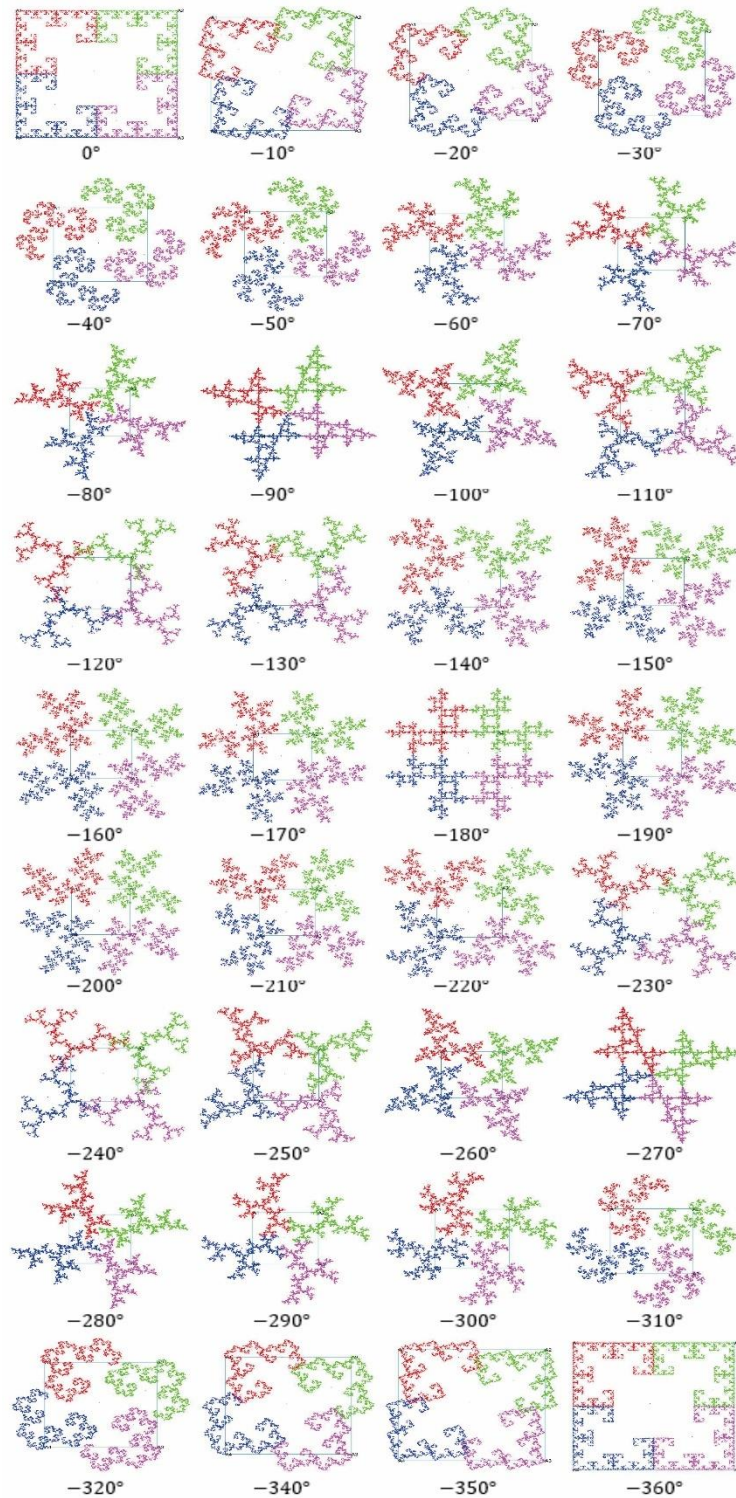


Gambar 4.67 Variasi rotasi sudut positif pada modifikasi 2 aturan *random*

Hasil visualisasi variasi rotasi gambar di atas pada keseluruhannya mirip dengan aturan-aturan sebelumnya, yaitu menunjukkan bahwa di dalam segiempat terdapat bangun yang menyerupai segiempat yang lebih kecil dimana segiempat tersebut menyebar pada masing-masing titik sudut. Hasil perputaran dalam variasi rotasinya dapat dikatakan memiliki bentuk yang identik sama seperti *chaos game* yang dibentuk menggunakan aturan *random* dan modifikasi 1 aturan *random* yaitu titik terakhir yang telah dipilih tidak boleh dipilih kembali. Namun objek pertama pada sudut  $0^\circ$  yang dihasilkan aturan modifikasi 2 ini dimulai dengan bentuk yang sama seperti objek rotasi modifikasi 1 yang ada pada perputaran  $180^\circ$ , sedangkan bentuk *chaos game* modifikasi 2 pada sudut  $180^\circ$  seperti pada objek yang dihasilkan aturan modifikasi 1 pada sudut  $0^\circ$ . Objek hasil bentukan *chaos game* dengan modifikasi 2 aturan *random* ini juga memiliki empat bagian warna yang berbeda ke setiap titik sudutnya yang digunakan sebagai titik tumpu untuk merotasikan *chaos game* segiempat. Kumpulan titik-titik yang berwarna merah merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_1$ , titik-titik yang berwarna hijau merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_2$ , titik-titik yang berwarna biru merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_3$ , serta titik-titik yang berwarna *magenta* merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_4$ . Pemilihan titik untuk modifikasi 2 aturan *random* pada simulasi program ini telah ditunjukkan contohnya pada Gambar 4.16 dan 4.17.

Berdasarkan bentuk yang dihasilkan, seluruh objek yang dihasilkan pada rotasi sudut positif dapat dikatakan sebagai objek fraktal karena memiliki kemiripan bentuk satu sama lain (*self-similarity*) dan titik-titik hasil bentukan yang dihasilkan *chaos game* dengan modifikasi 2 aturan *random* ini menyerupai bentuk segiempat yang menyebar pada setiap titik acuannya. Perputaran pada sudut negatifnya juga merupakan objek fraktal. Perbedaan antara perputaran sudut positif dan negatifnya pun sama seperti aturan sebelumnya, yaitu berada pada arah rotasinya, dimana rotasi sudut positif berputar berlawanan arah jarum jam, sedangkan rotasi sudut

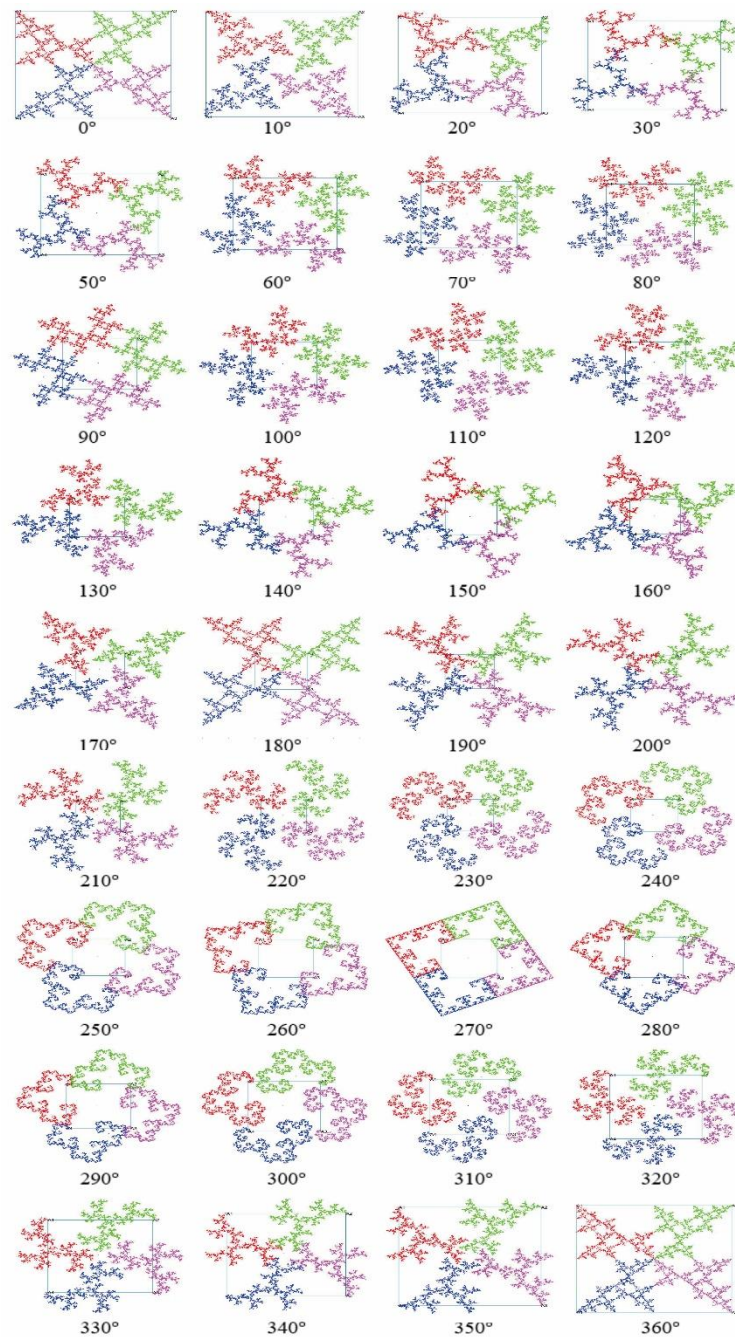
negatif sebaliknya. Berikut perputaran *chaos game* segiempat menggunakan modifikasi 2 aturan *random* yang dilakukan pada sudut negatif setiap perputaran  $10^\circ$  dengan  $\alpha = 1/2$  dan iterasi 10.000 ditunjukkan pada Gambar 4.68.



Gambar 4.68 Variasi rotasi sudut negatif pada modifikasi 2 aturan *random*

#### 4.3.4 Modifikasi 3: Titik Sebelah Kiri Tidak Boleh Dipilih

Modifikasi 3 aturan *random chaos game* segiempat akan menunjukkan hasil bentukan *chaos game* dengan syarat titik sebelah kiri tidak boleh dipilih. Hasil visualisasi rotasi *chaos game* segiempat menggunakan aturan modifikasi 3 dengan perputaran sudut setiap  $10^\circ$ ,  $a = 1/2$  dan iterasi 10.000 ditunjukkan pada Gambar 4.69.

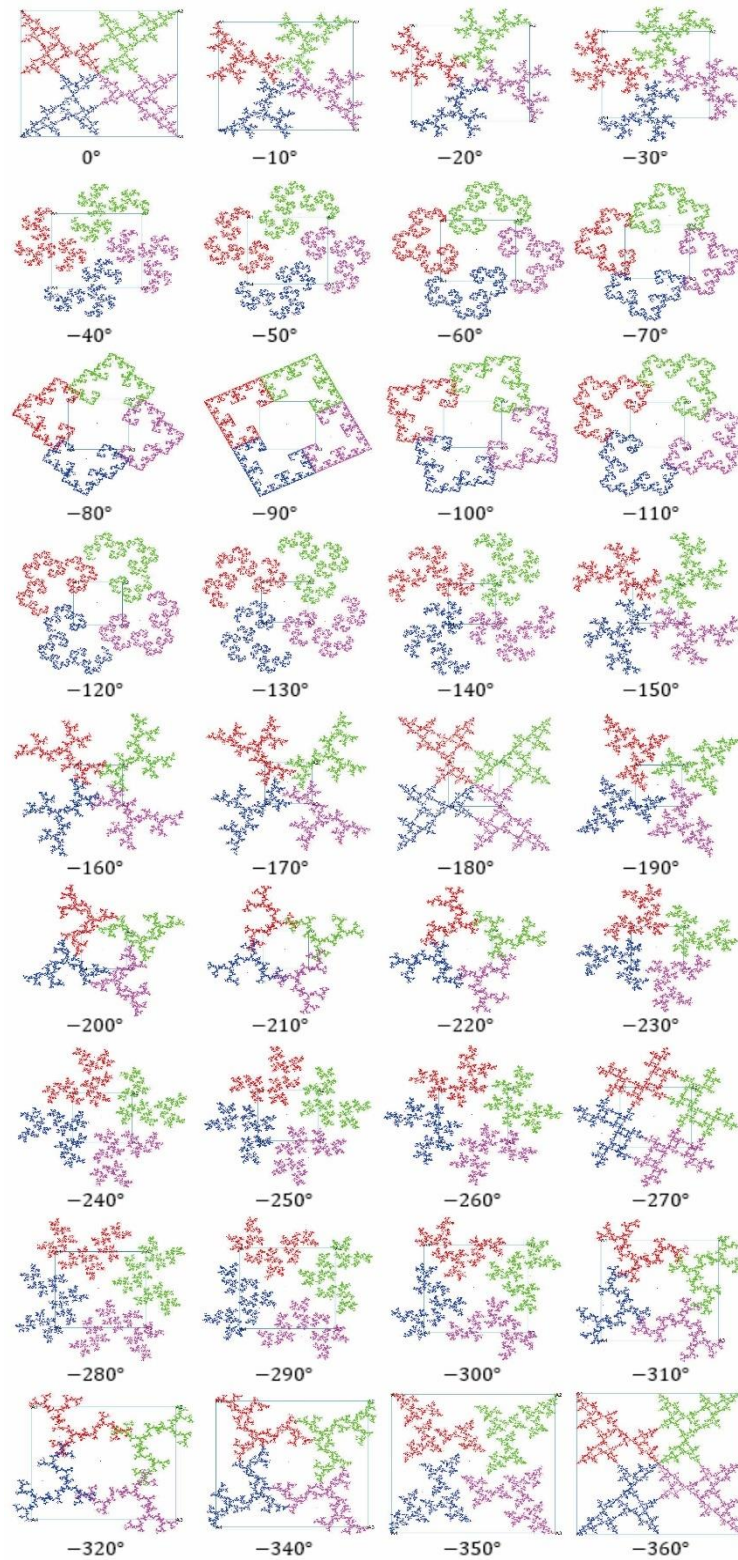


Gambar 4.69 Variasi rotasi sudut positif pada modifikasi 3 aturan *random*

Hasil visualisasi variasi rotasi gambar di atas pada keseluruhannya mirip dengan aturan-aturan sebelumnya, yaitu pada dasarnya menunjukkan bahwa di dalam segiempat terdapat bangun yang menyerupai segiempat yang lebih kecil dimana segiempat tersebut menyebar pada masing-masing titik sudut. Hasil perputaran dalam variasi rotasinya dapat dikatakan memiliki bentuk yang identik sama seperti *chaos game* yang dibentuk menggunakan aturan *random*, modifikasi 1 aturan *random* yaitu titik terakhir yang telah dipilih tidak boleh dipilih kembali, dan modifikasi 2 aturan *random* yaitu titik yang berhadapan tidak boleh dipilih. Objek pertama pada sudut  $0^\circ$  yang dihasilkan aturan modifikasi 3 ini dimulai dengan bentuk yang sama seperti objek rotasi modifikasi 1 yang ada pada perputaran  $90^\circ$ , hanya saja pada modifikasi 3, titik-titiknya berada di dalam segiempat. Objek hasil bentukan *chaos game* dengan modifikasi 3 aturan *random* ini juga memiliki empat bagian warna yang berbeda ke setiap titik sudutnya yang digunakan sebagai titik tumpu untuk merotasikan *chaos game* segiempat. Kumpulan titik-titik yang berwarna merah merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_1$ , titik-titik yang berwarna hijau merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_2$ , titik-titik yang berwarna biru merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_3$ , serta titik-titik yang berwarna *magenta* merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_4$ . Aturan pembentukannya telah dijelaskan pada subbab sebelumnya, tepatnya pada Gambar 4.22 dan Gambar 4.23.

Berdasarkan bentuk yang dihasilkan, seluruh objek yang dihasilkan pada rotasi sudut positif dapat dikatakan sebagai objek fraktal karena memiliki kemiripan bentuk satu sama lain (*self-similarity*) dan titik-titik hasil bentukan yang dihasilkan *chaos game* dengan modifikasi 3 aturan *random* ini menyerupai bentuk segiempat yang menyebar pada setiap titik acuannya. Perputaran pada sudut negatifnya juga merupakan objek fraktal. Perbedaannya pada dasarnya pun sama, yaitu pada arah perputaran sudut positif dan negatifnya. Berikut perputaran *chaos game* segiempat

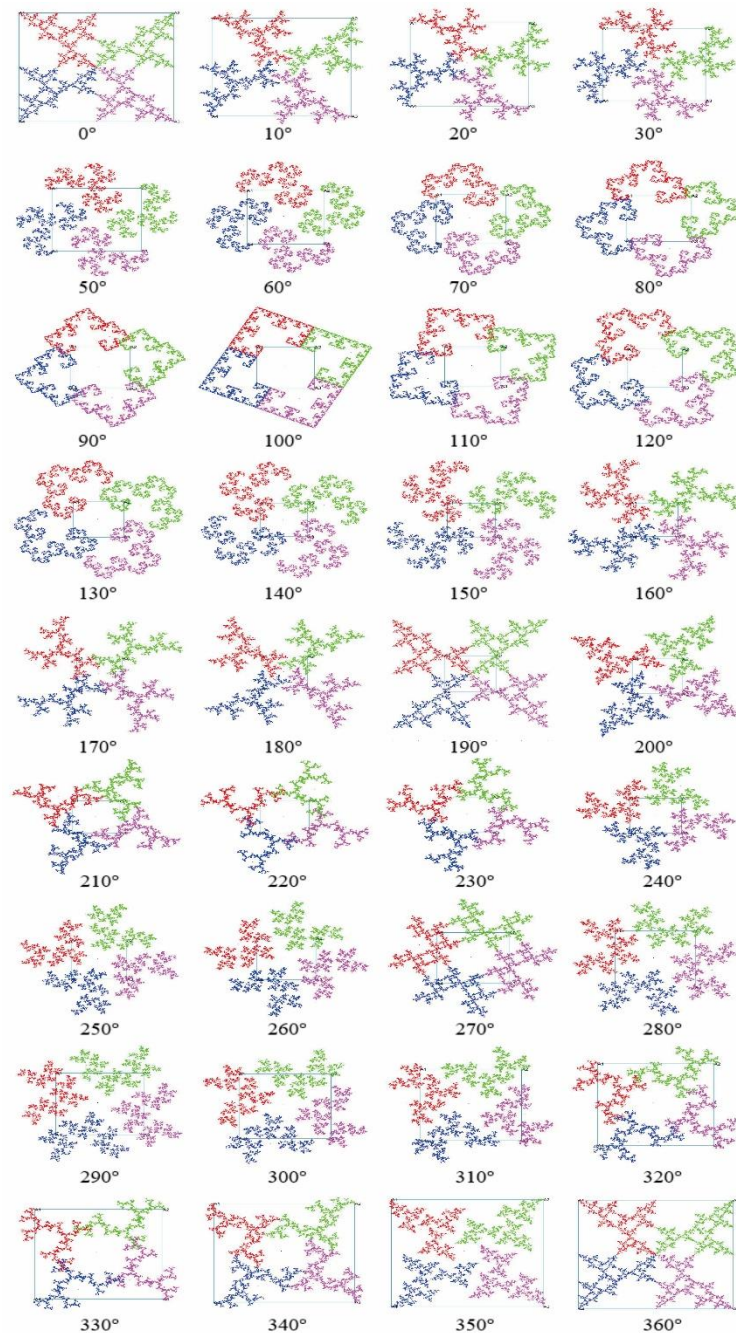
menggunakan modifikasi 3 aturan *random* yang dilakukan pada sudut negatif setiap perputaran  $10^\circ$  dengan  $\alpha = 1/2$  dan iterasi 10.000 ditunjukkan pada Gambar 4.70.



Gambar 4.70 Variasi rotasi sudut negatif pada modifikasi 3 aturan *random*

#### 4.3.5 Modifikasi 4: Titik Sebelah Kanan Tidak Boleh Dipilih

Modifikasi 4 aturan *random chaos game* segiempat akan menunjukkan hasil bentukan *chaos game* dengan syarat titik sebelah kanan tidak boleh dipilih. Hasil visualisasi rotasi *chaos game* segiempat menggunakan aturan modifikasi 4 dengan perputaran sudut setiap  $10^\circ$ ,  $a = 1/2$  dan iterasi 10.000 ditunjukkan pada Gambar 4.71.

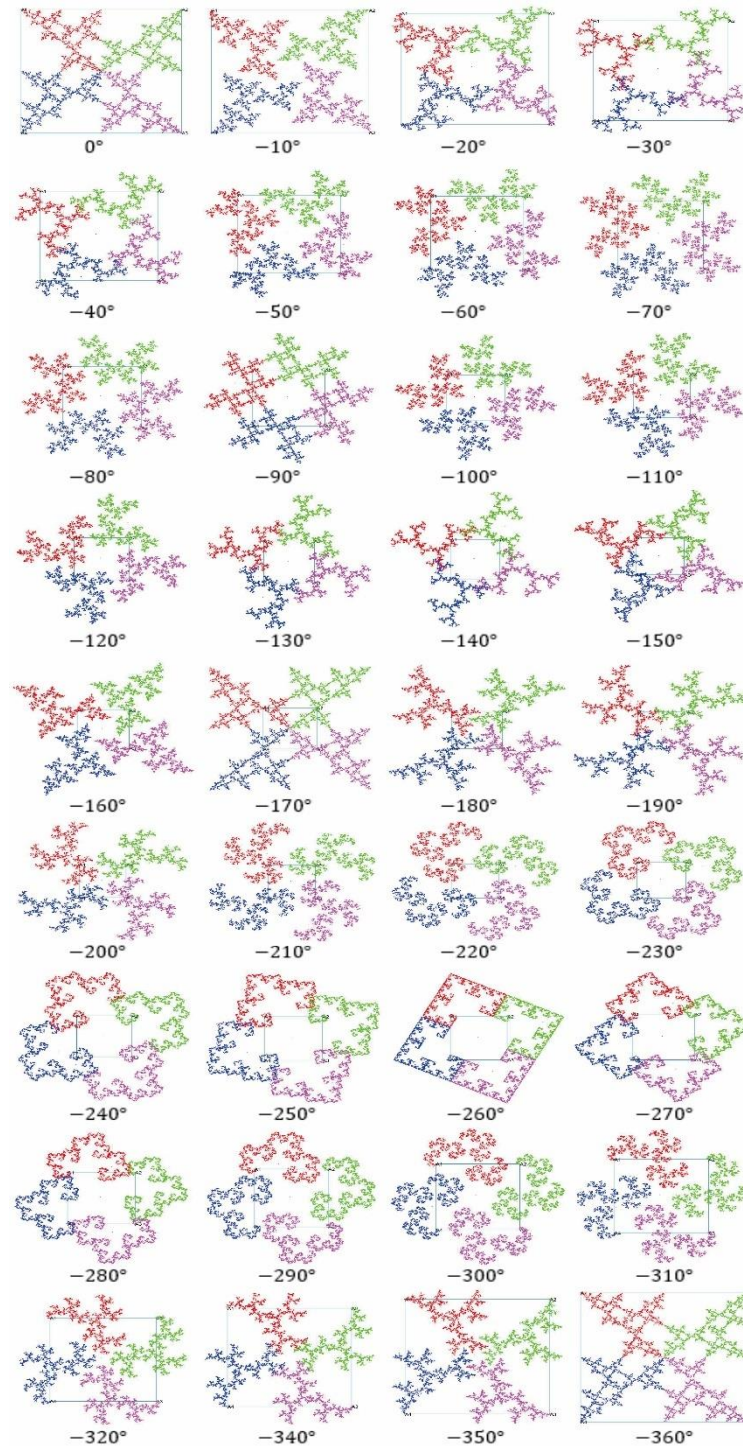


Gambar 4.71 Variasi rotasi sudut positif pada modifikasi 4 aturan *random*

Hasil visualisasi variasi rotasi gambar di atas pada keseluruhannya mirip dengan aturan-aturan sebelumnya, terutama pada modifikasi 3 aturan *random*. Variasi rotasinya pada dasarnya menunjukkan bahwa di dalam segiempat terdapat bangun yang menyerupai segiempat yang lebih kecil dimana segiempat tersebut menyebar pada masing-masing titik sudut. Hasil perputaran dalam variasi rotasinya dapat dikatakan memiliki bentuk yang identik sama seperti *chaos game* yang dibentuk menggunakan aturan *random*, modifikasi 1 aturan *random* yaitu titik terakhir yang telah dipilih tidak boleh dipilih kembali, modifikasi 2 aturan *random* yaitu titik yang berhadapan tidak boleh dipilih, dan modifikasi 3 aturan *random* yaitu titik sebelah kiri tidak boleh dipilih. Objek pertama pada sudut  $0^\circ$  yang dihasilkan aturan modifikasi 4 ini dimulai dengan bentuk yang mirip seperti bentuk awal yang dihasilkan modifikasi 3, hanya saja arahnya yang berbeda. Pada modifikasi 3, bentuknya lebih mengarah ke sebelah kiri, sedangkan pada modifikasi 4 lebih mengarah ke sebelah kanan. Objek hasil bentukan *chaos game* dengan modifikasi 4 aturan *random* ini juga memiliki empat bagian warna yang berbeda ke setiap titik sudutnya yang digunakan sebagai titik tumpu untuk merotasikan *chaos game* segiempat. Kumpulan titik-titik yang berwarna merah merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_1$ , titik-titik yang berwarna hijau merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_2$ , titik-titik yang berwarna biru merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_3$ , serta titik-titik yang berwarna *magenta* merupakan titik hasil bentukan *chaos game* segiempat dengan titik acuan atau titik tumpu rotasi yang berada pada  $A_4$ . Aturan pembentukannya telah dijelaskan pada subbab sebelumnya, tepatnya pada Gambar 4.28 dan Gambar 4.29.

Berdasarkan bentuk yang dihasilkan, seluruh objek yang dihasilkan pada rotasi sudut positif dapat dikatakan sebagai objek fraktal karena memiliki kemiripan bentuk satu sama lain (*self-similarity*) dan titik-titik hasil bentukan yang dihasilkan *chaos game* dengan modifikasi 4 aturan *random* ini menyerupai bentuk segiempat yang menyebar pada setiap titik acuannya. Perputaran pada sudut negatifnya juga

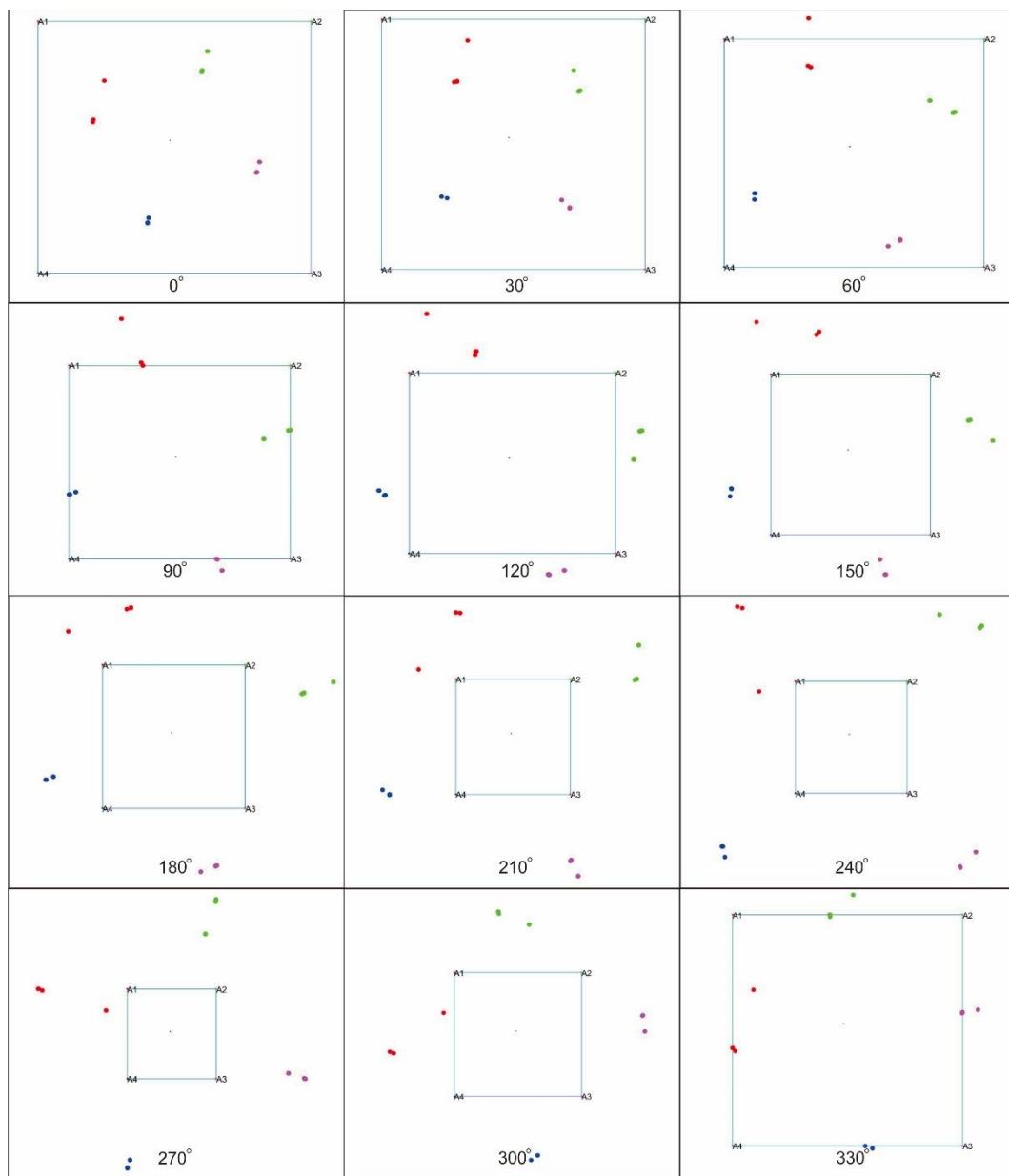
merupakan objek fraktal. Perbedaannya pada dasarnya pun sama, yaitu pada arah perputaran sudut positif dan negatifnya. Perputaran *chaos game* segiempat menggunakan modifikasi 4 aturan *random* yang dilakukan pada sudut negatif setiap perputaran  $10^\circ$  dengan  $\alpha = 1/2$  dan iterasi 10.000 ditunjukkan pada Gambar 4.72.



Gambar 4.72 Variasi rotasi sudut negatif pada modifikasi 4 aturan *random*

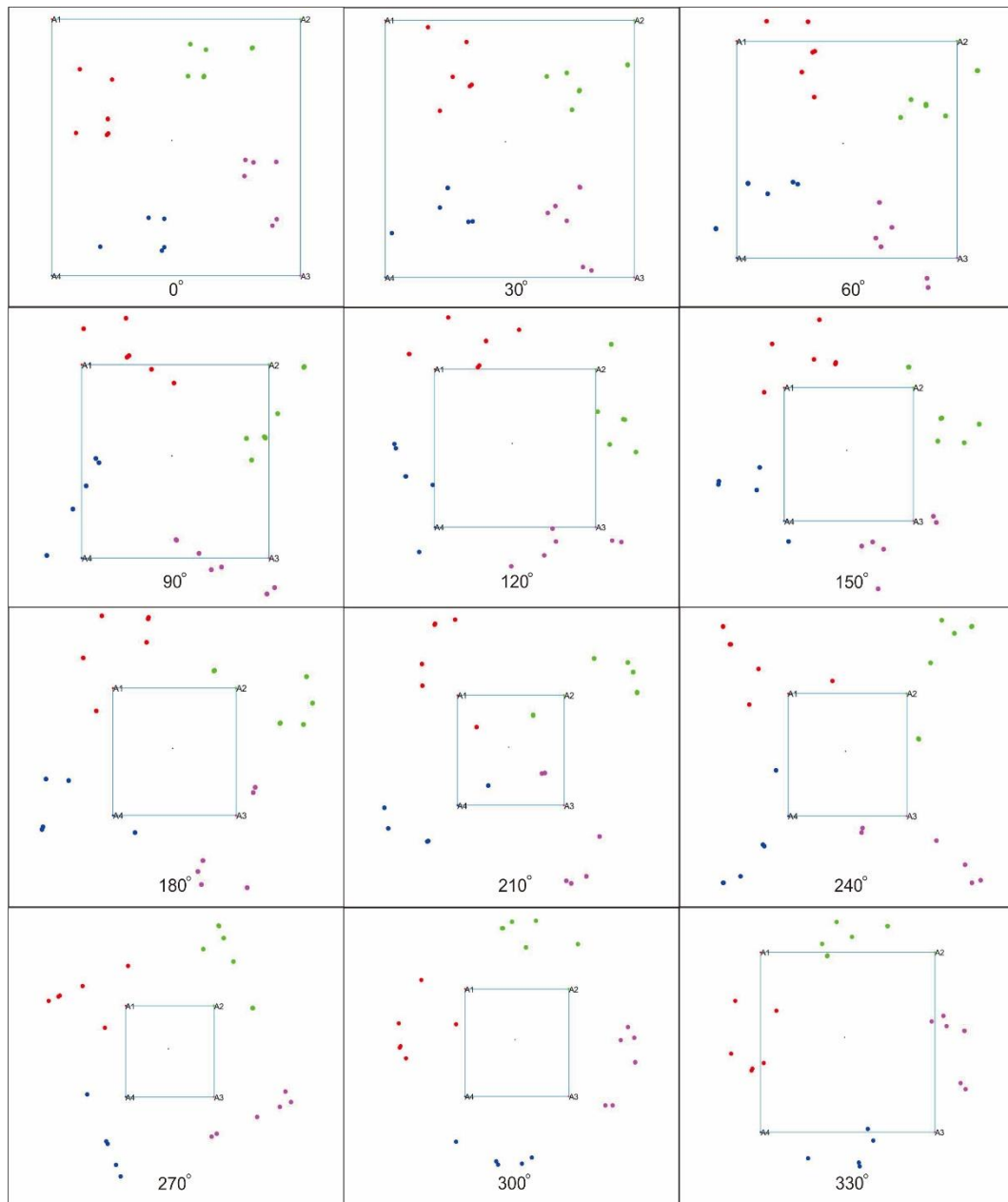
#### 4.3.6 Aturan *Non-Random*

Simulasi program yang telah dijelaskan pada subbab sebelumnya didapatkan hasil visualisasi rotasi *chaos game* segiempat dengan  $a = 1/2$ , iterasi 5.000, dan telah dilakukan pada perputaran sudut positif setiap  $30^\circ$  ternyata tidak menghasilkan bentuk yang jelas, melainkan hanya seperti titik-titik yang berjauhan. Berikut visualisasi hasil simulasi variasi sudut positif *chaos game* dengan aturan 4 pola pemilihan titik yang teratur dan berurutan ( $A_1 - A_2 - A_3 - A_4$ ) ditunjukkan pada Gambar 4.73.



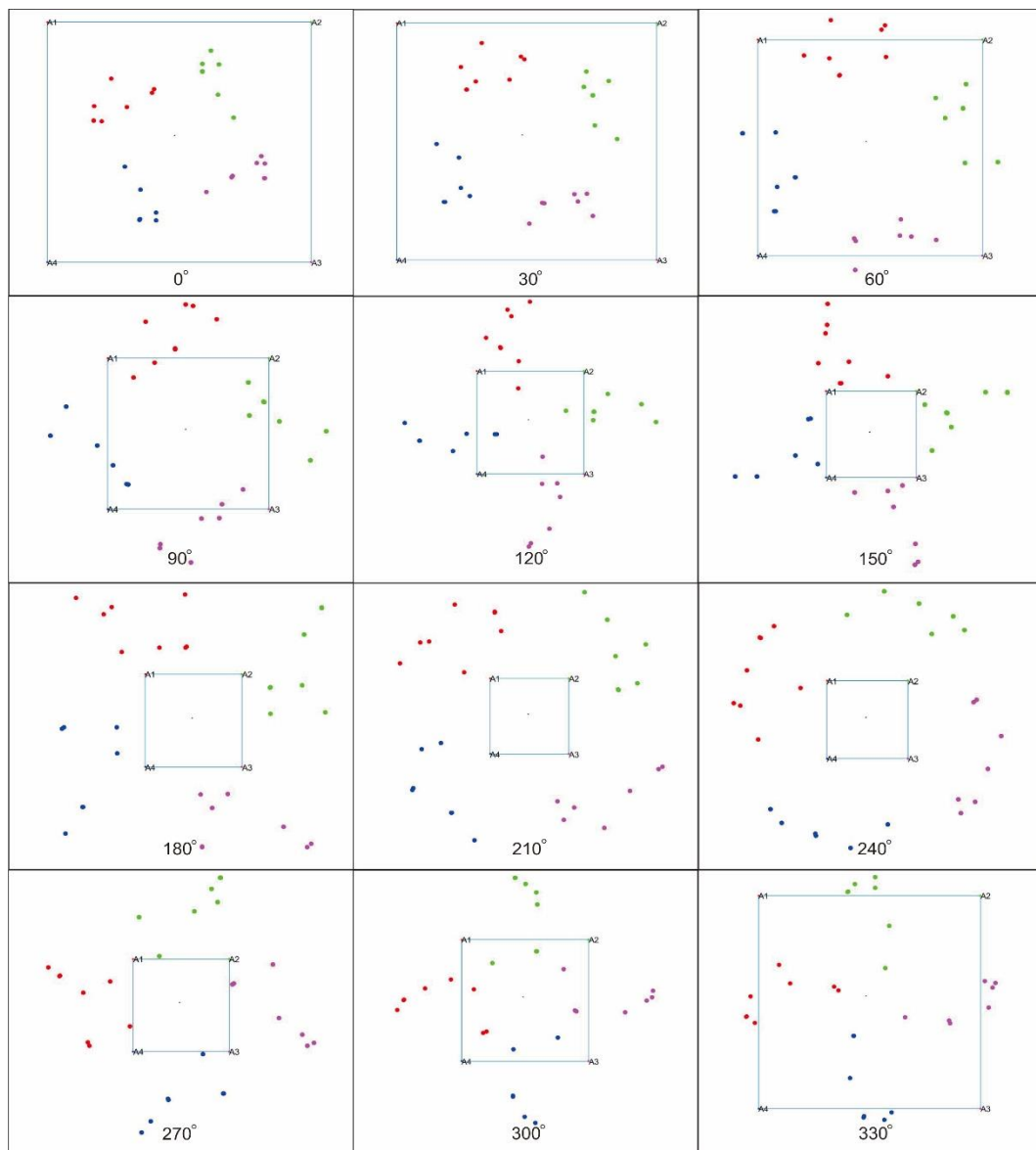
Gambar 4.73 Variasi rotasi sudut positif *chaos game* dengan aturan *non-random* 1

Simulasi juga dilakukan pada aturan *non-random 2* dengan 16 pola pemilihan titik, yaitu pada kelipatan 4 berulang dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_3 - A_4 - A_1 - A_2 - A_2 - A_3 - A_4 - A_1 - A_1 - A_2 - A_3 - A_4 - A_4$ ). Ketentuan yang digunakan sama seperti sebelumnya,  $a = 1/2$ , iterasi 5.000, dan telah dilakukan pada perputaran sudut positif setiap  $30^\circ$ . Hasil variasi rotasinya juga tidak menunjukkan bentuk fraktal. Berikut visualisasi hasil simulasi variasi sudut positif *chaos game* dengan aturan *non-random 2* ditunjukkan pada Gambar 4.74.



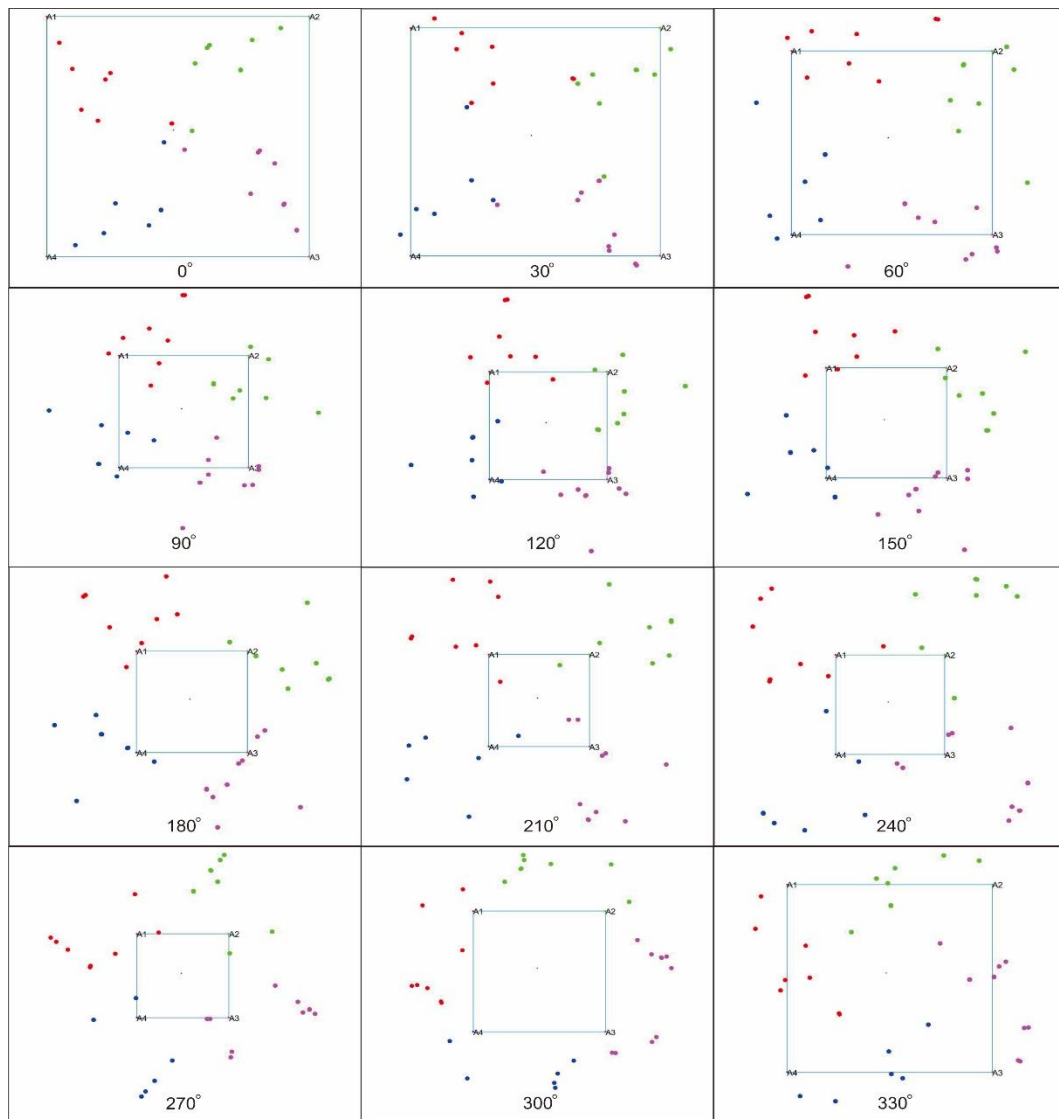
Gambar 4.74 Variasi rotasi sudut positif *chaos game* dengan aturan *non-random 2*

Selanjutnya pada aturan *non-random* 3 dengan 20 pola pemilihan titik, yaitu pada kelipatan 5 melompati satu titik setelahnya dan pada iterasi  $t$  yaitu  $t \equiv 4 \bmod (5)$  titik yang dipilih adalah titik yang berhadapan dengan titik terakhirnya ( $A_1 - A_2 - A_3 - A_1 - A_3 - A_4 - A_1 - A_2 - A_4 - A_2 - A_3 - A_4 - A_1 - A_3 - A_1 - A_2 - A_3 - A_4 - A_2 - A_4$ ). Ketentuan yang digunakan sama seperti sebelumnya,  $a = 1/2$ , iterasi 5.000, dan telah dilakukan pada perputaran sudut positif setiap  $30^\circ$ . Hasil variasi rotasinya juga tidak menunjukkan bentuk fraktal. Berikut visualisasi hasil simulasi variasi sudut positif *chaos game* dengan aturan *non-random* 3 ditunjukkan pada Gambar 4.75.



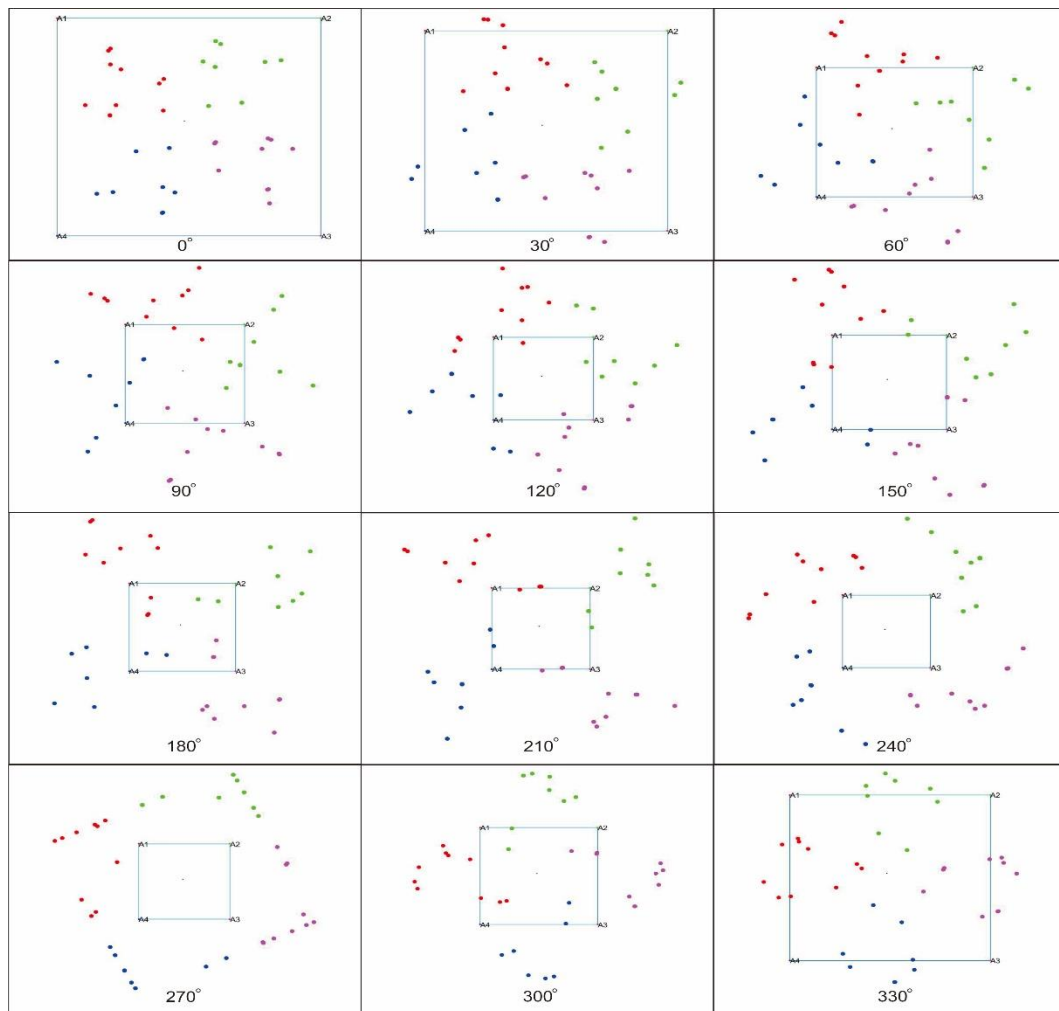
Gambar 4.75 Variasi rotasi sudut positif *chaos game* dengan aturan *non-random* 3

Selanjutnya simulasi pada aturan *non-random 4* dengan 24 pola pemilihan titik, yaitu pada kelipatan 6 dan pada iterasi  $t$  yaitu  $t \equiv 5 \bmod (6)$  berulang satu kali seperti titik sebelumnya, kemudian pada  $t \equiv 4 \bmod (6)$  melompati satu titik setelah titik terakhirnya ( $A_1 - A_2 - A_3 - A_3 - A_3 - A_1 - A_2 - A_3 - A_4 - A_4 - A_4 - A_2 - A_3 - A_4 - A_1 - A_1 - A_1 - A_3 - A_4 - A_1 - A_2 - A_2 - A_2 - A_4$ ). Ketentuan yang digunakan sama seperti sebelumnya,  $a = 1/2$ , iterasi 5.000, dan telah dilakukan pada perputaran sudut positif setiap  $30^\circ$ . Hasil variasi rotasinya juga tidak menunjukkan bentuk fraktal, namun titik yang terbentuk semakin banyak. Berikut visualisasi hasil simulasi variasi sudut positif *chaos game* dengan aturan *non-random 4* ditunjukkan pada Gambar 4.76.



Gambar 4.76 Variasi rotasi sudut positif *chaos game* dengan aturan *non-random 4*

Selanjutnya simulasi pada *non-random 5* dengan 28 pola pemilihan titik, pada iterasi  $t$  yaitu  $t \equiv 4 \bmod (7)$  melompati satu titik dari titik terakhir, pada iterasi  $t \equiv 5 \bmod (7)$  berulang seperti titik terakhir yang dipilih, pada iterasi  $t \equiv 6 \bmod (7)$  melompati satu titik dari titik terakhir lagi, dan pada kelipatan 7 berulang lagi satu titik dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_1 - A_1 - A_3 - A_3 - A_4 - A_1 - A_2 - A_4 - A_1 - A_2 - A_4 - A_4 - A_2 - A_2 - A_3 - A_4 - A_1 - A_3 - A_3 - A_1 - A_1 - A_2 - A_3 - A_4 - A_2 - A_2 - A_4 - A_4$ ). Ketentuan yang digunakan sama seperti sebelumnya,  $a = 1/2$ , iterasi 5.000, dan telah dilakukan pada perputaran sudut positif setiap  $30^\circ$ . Hasilnya menunjukkan semakin banyak titik yang terbentuk, namun tetap tidak membentuk objek fraktal. Berikut visualisasi hasil simulasi variasi sudut positif *chaos game* dengan aturan *non-random 5* ditunjukkan pada Gambar 4.77.



Gambar 4.77 Variasi rotasi sudut positif *chaos game* dengan aturan *non-random 5*

Pada aturan *non-random* 1 sampai dengan *non-random* 5 yang dilakukan dengan  $a = 1/2$ , iterasi 5.000, dan variasi perputaran sudut positif setiap  $30^\circ$  seluruhnya tidak membentuk objek fraktal. Variasi rotasi dengan aturan *non-random* 1 dengan 4 pola pemilihan titik yang teratur dan berurutan ( $A_1 - A_2 - A_3 - A_4$ ) seperti yang ditunjukkan pada Gambar 4.73 menunjukkan titik-titik hasil bentukannya tidak menyebar ke semua titik sudut yang merupakan titik acuan dan letaknya berjauhan satu sama lainnya. Hasilnya juga menunjukkan semakin besar sudut rotasinya, maka titik hasil bentukannya semakin jauh keluar dari segiempat. Titik-titik hasil bentukannya pada sudut yang telah dilakukan tidak membentuk objek fraktal karena tidak ditemukan bentuk yang identik. Namun aturan ini memiliki karakteristik bahwa titik-titik hasil bentukannya “konvergen” sebanyak pola titik yang ditentukan. Tabel titik baru yang dihasilkan setelah simulasi program menunjukkan bahwa semakin lama titik berada pada koordinat yang sama sesuai dengan pola pemilihan titik yang diberikan. Berikut koordinat titik yang dihasilkan dari lima iterasi terakhir aturan *non-random* 1 ditunjukkan pada Gambar 4.78.

| Titik Baru |       |   |   |   |
|------------|-------|---|---|---|
|            | pivot | x | y | z |
| 4980       |       |   |   |   |
| 4981       | 1     | 2 | 6 |   |
| 4982       | 2     | 6 | 8 |   |
| 4983       | 3     | 8 | 4 |   |
| 4984       | 4     | 4 | 2 |   |
| 4985       | 1     | 2 | 6 |   |
| 4986       | 2     | 6 | 8 |   |
| 4987       | 3     | 8 | 4 |   |
| 4988       | 4     | 4 | 2 |   |
| 4989       | 1     | 2 | 6 |   |
| 4990       | 2     | 6 | 8 |   |
| 4991       | 3     | 8 | 4 |   |
| 4992       | 4     | 4 | 2 |   |
| 4993       | 1     | 2 | 6 |   |
| 4994       | 2     | 6 | 8 |   |
| 4995       | 3     | 8 | 4 |   |
| 4996       | 4     | 4 | 2 |   |
| 4997       | 1     | 2 | 6 |   |
| 4998       | 2     | 6 | 8 |   |
| 4999       | 3     | 8 | 4 |   |
| 5000       | 4     | 4 | 2 |   |

Gambar 4.78 Koordinat titik baru aturan *non-random* 1

Variasi rotasi dengan aturan *non-random* 2 dengan 16 pola pemilihan titik, yaitu pada kelipatan 4 berulang dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_3 - A_4 - A_1 - A_2 - A_2 - A_3 - A_4 - A_1 - A_1 - A_2 - A_3 - A_4 - A_4$ ) seperti yang ditunjukkan pada Gambar 4.74 juga menunjukkan titik-titik hasil bentukannya tidak menyebar ke semua titik acuan, tetapi titik “konvergen” yang dihasilkan semakin banyak. Hasilnya juga menunjukkan semakin besar sudut rotasinya, maka titik hasil bentukannya semakin jauh keluar dari segiempat. Titik-titik hasil bentukannya pada sudut yang telah dilakukan tidak membentuk objek fraktal yang memiliki ciri bentuk identik, walaupun lebih banyak titik yang terbentuk dibandingkan dengan menggunakan aturan *non-random* 1. Aturan ini juga memiliki karakteristik sama seperti aturan *non-random* 1, yaitu titik-titik hasil bentukannya “konvergen” sebanyak pola titik yang ditentukan. Tabel titik baru yang dihasilkan setelah simulasi program menunjukkan bahwa semakin lama titik berada pada koordinat yang sama sesuai dengan pola pemilihan titik yang diberikan. Berikut koordinat titik yang dihasilkan dari tiga iterasi terakhir aturan *non-random* 2 ditunjukkan pada Gambar 4.79.

| Titik Baru |       |        |        | Titik Baru |       |        |        | Titik Baru |       |        |        |
|------------|-------|--------|--------|------------|-------|--------|--------|------------|-------|--------|--------|
|            | pivot | x      | y      |            | pivot | x      | y      |            | pivot | x      | y      |
| 4940       |       | 1.1204 | 0.0545 | 4900       |       | 1.9455 | 1.1204 | 4900       |       | 0.9728 | 5.5642 |
| 4941       | 2     | 5.5642 | 9.0272 | 4961       | 1     | 0.9728 | 5.5642 | 4981       | 4     | 4.4358 | 0.9728 |
| 4942       | 3     | 7.7821 | 4.5136 | 4962       | 2     | 5.4864 | 7.7821 | 4982       | 1     | 2.2179 | 5.4864 |
| 4943       | 4     | 3.8911 | 2.2568 | 4963       | 3     | 7.7432 | 3.8911 | 4983       | 2     | 6.1089 | 7.7432 |
| 4944       | 4     | 1.9455 | 1.1284 | 4964       | 3     | 8.8716 | 1.9455 | 4984       | 2     | 8.0545 | 8.8716 |
| 4945       | 1     | 0.9728 | 5.5642 | 4965       | 4     | 4.4358 | 0.9728 | 4985       | 3     | 9.0272 | 4.4358 |
| 4946       | 2     | 5.4864 | 7.7821 | 4966       | 1     | 2.2179 | 5.4864 | 4986       | 4     | 4.5136 | 2.2179 |
| 4947       | 3     | 7.7432 | 3.8911 | 4967       | 2     | 6.1089 | 7.7432 | 4987       | 1     | 2.2568 | 6.1089 |
| 4948       | 3     | 8.8716 | 1.9455 | 4968       | 2     | 8.0545 | 8.8716 | 4988       | 1     | 1.1284 | 8.0545 |
| 4949       | 4     | 4.4358 | 0.9728 | 4969       | 3     | 9.0272 | 4.4358 | 4989       | 2     | 5.5642 | 9.0272 |
| 4950       | 1     | 2.2179 | 5.4864 | 4970       | 4     | 4.5136 | 2.2179 | 4990       | 3     | 7.7821 | 4.5136 |
| 4951       | 2     | 6.1089 | 7.7432 | 4971       | 1     | 2.2568 | 6.1089 | 4991       | 4     | 3.8911 | 2.2568 |
| 4952       | 2     | 8.0545 | 8.8716 | 4972       | 1     | 1.1284 | 8.0545 | 4992       | 4     | 1.9455 | 1.1284 |
| 4953       | 3     | 9.0272 | 4.4358 | 4973       | 2     | 5.5642 | 9.0272 | 4993       | 1     | 0.9728 | 5.5642 |
| 4954       | 4     | 4.5136 | 2.2179 | 4974       | 3     | 7.7821 | 4.5136 | 4994       | 2     | 5.4864 | 7.7821 |
| 4955       | 1     | 2.2568 | 6.1089 | 4975       | 4     | 3.8911 | 2.2568 | 4995       | 3     | 7.7432 | 3.8911 |
| 4956       | 1     | 1.1284 | 8.0545 | 4976       | 4     | 1.9455 | 1.1284 | 4996       | 3     | 8.8716 | 1.9455 |
| 4957       | 2     | 5.5642 | 9.0272 | 4977       | 1     | 0.9728 | 5.5642 | 4997       | 4     | 4.4358 | 0.9728 |
| 4958       | 3     | 7.7821 | 4.5136 | 4978       | 2     | 5.4864 | 7.7821 | 4998       | 1     | 2.2179 | 5.4864 |
| 4959       | 4     | 3.8911 | 2.2568 | 4979       | 3     | 7.7432 | 3.8911 | 4999       | 2     | 6.1089 | 7.7432 |
| 4960       | 4     | 1.9455 | 1.1284 | 4980       | 3     | 8.8716 | 1.9455 | 5000       | 2     | 8.0545 | 8.8716 |

Gambar 4.79 Koordinat titik baru aturan *non-random* 2

Variasi rotasi dengan aturan *non-random* 3 dengan 20 pola pemilihan titik, yaitu pada kelipatan 5 melompati satu titik setelahnya dan pada iterasi  $t$  yaitu  $t \equiv 4 \bmod (5)$  titik yang dipilih adalah titik yang berhadapan dengan titik terakhirnya ( $A_1 - A_2 - A_3 - A_1 - A_3 - A_4 - A_1 - A_2 - A_4 - A_2 - A_3 - A_4 - A_1 - A_3 - A_1 - A_2 - A_3 - A_4 - A_2 - A_4$ ) seperti yang ditunjukkan pada Gambar 4.75 menghasilkan titik “konvergen” yang semakin banyak banyak lagi. Namun hasilnya tetap tidak membentuk objek fraktal karena tidak ada bentuk yang identik satu sama lain. Sama halnya seperti aturan *non-random* 1 dan *non-random* 2 yang dilakukan sebelumnya, yaitu semakin besar sudut rotasinya, maka titik hasil bentukannya semakin jauh keluar dari segiempat. Aturan ini juga memiliki karakteristik sama seperti aturan *non-random* 1 dan *non-random* 2, yaitu titik-titik hasil bentukannya “konvergen” sebanyak pola titik yang ditentukan, dan semakin banyak pola yang ditentukan, semakin banyak titik yang terbentuk menjadi titik “konvergen”. Berikut koordinat titik yang dihasilkan dari tiga iterasi terakhir aturan *non-random* 3 ditunjukkan pada Gambar 4.80.

| Titik Baru |       |        |        | Titik Baru |       |        |        | Titik Baru |       |        |        |
|------------|-------|--------|--------|------------|-------|--------|--------|------------|-------|--------|--------|
|            | pivot | x      | y      |            | pivot | x      | y      |            | pivot | x      | y      |
| 4940       |       |        |        | 4960       |       |        |        | 4980       |       |        |        |
| 4941       | 1     | 1.7659 | 6.5073 | 4961       | 1     | 1.7659 | 6.5073 | 4981       | 1     | 1.7659 | 6.5073 |
| 4942       | 2     | 5.8829 | 8.2537 | 4962       | 2     | 5.8829 | 8.2537 | 4982       | 2     | 5.8829 | 8.2537 |
| 4943       | 3     | 7.9415 | 4.1268 | 4963       | 3     | 7.9415 | 4.1268 | 4983       | 3     | 7.9415 | 4.1268 |
| 4944       | 1     | 3.9707 | 7.0634 | 4964       | 1     | 3.9707 | 7.0634 | 4984       | 1     | 3.9707 | 7.0634 |
| 4945       | 3     | 6.9854 | 3.5317 | 4965       | 3     | 6.9854 | 3.5317 | 4985       | 3     | 6.9854 | 3.5317 |
| 4946       | 4     | 3.4927 | 1.7659 | 4966       | 4     | 3.4927 | 1.7659 | 4986       | 4     | 3.4927 | 1.7659 |
| 4947       | 1     | 1.7463 | 5.8829 | 4967       | 1     | 1.7463 | 5.8829 | 4987       | 1     | 1.7463 | 5.8829 |
| 4948       | 2     | 5.8732 | 7.9415 | 4968       | 2     | 5.8732 | 7.9415 | 4988       | 2     | 5.8732 | 7.9415 |
| 4949       | 4     | 2.9366 | 3.9707 | 4969       | 4     | 2.9366 | 3.9707 | 4989       | 4     | 2.9366 | 3.9707 |
| 4950       | 2     | 6.4683 | 6.9854 | 4970       | 2     | 6.4683 | 6.9854 | 4990       | 2     | 6.4683 | 6.9854 |
| 4951       | 3     | 8.2341 | 3.4927 | 4971       | 3     | 8.2341 | 3.4927 | 4991       | 3     | 8.2341 | 3.4927 |
| 4952       | 4     | 4.1171 | 1.7463 | 4972       | 4     | 4.1171 | 1.7463 | 4992       | 4     | 4.1171 | 1.7463 |
| 4953       | 1     | 2.0585 | 5.8732 | 4973       | 1     | 2.0585 | 5.8732 | 4993       | 1     | 2.0585 | 5.8732 |
| 4954       | 3     | 6.0293 | 2.9366 | 4974       | 3     | 6.0293 | 2.9366 | 4994       | 3     | 6.0293 | 2.9366 |
| 4955       | 1     | 3.0146 | 6.4683 | 4975       | 1     | 3.0146 | 6.4683 | 4995       | 1     | 3.0146 | 6.4683 |
| 4956       | 2     | 6.5073 | 8.2341 | 4976       | 2     | 6.5073 | 8.2341 | 4996       | 2     | 6.5073 | 8.2341 |
| 4957       | 3     | 8.2537 | 4.1171 | 4977       | 3     | 8.2537 | 4.1171 | 4997       | 3     | 8.2537 | 4.1171 |
| 4958       | 4     | 4.1268 | 2.0585 | 4978       | 4     | 4.1268 | 2.0585 | 4998       | 4     | 4.1268 | 2.0585 |
| 4959       | 2     | 7.0634 | 6.0293 | 4979       | 2     | 7.0634 | 6.0293 | 4999       | 2     | 7.0634 | 6.0293 |
| 4960       | 4     | 3.5317 | 3.0146 | 4980       | 4     | 3.5317 | 3.0146 | 5000       | 4     | 3.5317 | 3.0146 |

Gambar 4.80 Koordinat titik baru aturan *non-random* 3

Variasi rotasi dengan aturan *non-random* 4 dengan 24 pola pemilihan titik, yaitu pada kelipatan 6 dan pada iterasi  $t$  yaitu  $t \equiv 5 \bmod (6)$  berulang satu kali seperti titik sebelumnya, kemudian pada  $t \equiv 4 \bmod (6)$  melompati satu titik

setelah titik terakhirnya ( $A_1 - A_2 - A_3 - A_3 - A_3 - A_1 - A_2 - A_3 - A_4 - A_4 - A_4 - A_2 - A_3 - A_4 - A_1 - A_1 - A_1 - A_3 - A_4 - A_1 - A_2 - A_2 - A_2 - A_4$ ) seperti yang ditunjukkan pada Gambar 4.76 menghasilkan titik-titik hasil bentukan yang lebih banyak lagi dibandingkan aturan *non-random* 1, *non-random* 2, dan *non-random* 3. Sama halnya seperti aturan *non-random* 1, *non-random* 2, dan *non-random* 3, yaitu semakin besar sudut rotasinya, maka titik hasil bentukannya semakin jauh keluar dari segiempat. Namun hasilnya tetap tidak juga membentuk objek fraktal karena belum juga membentuk objek yang identik. Titik-titik hasil bentukannya juga “konvergen” sebanyak pola titik yang ditentukan, dan semakin banyak pola yang ditentukan, semakin banyak lagi titik yang terbentuk menjadi titik “konvergen”. Berikut koordinat titik yang dihasilkan dari tiga iterasi terakhir aturan *non-random* 4 ditunjukkan pada Gambar 4.81.

| Titik Baru |       |        |        | Titik Baru |       |        |        |
|------------|-------|--------|--------|------------|-------|--------|--------|
|            | pivot | x      | y      |            | pivot | x      | y      |
| 4920       |       |        |        | 4960       |       |        |        |
| 4921       | 1     | 2.2285 | 7.3786 | 4961       | 1     | 0.4857 | 8.9138 |
| 4922       | 2     | 6.1142 | 8.6893 | 4962       | 3     | 5.2429 | 4.4569 |
| 4923       | 3     | 8.0571 | 4.3446 | 4963       | 4     | 2.6214 | 2.2285 |
| 4924       | 3     | 9.0286 | 2.1723 | 4964       | 1     | 1.3107 | 6.1142 |
| 4925       | 3     | 9.5143 | 1.0862 | 4965       | 2     | 5.6554 | 8.0571 |
| 4926       | 1     | 4.7571 | 5.5431 | 4966       | 2     | 7.8277 | 9.0286 |
| 4927       | 2     | 7.3786 | 7.7715 | 4967       | 2     | 8.9138 | 9.5143 |
| 4928       | 3     | 8.6893 | 3.8858 | 4968       | 4     | 4.4569 | 4.7571 |
| 4929       | 4     | 4.3446 | 1.9429 | 4969       | 1     | 2.2285 | 7.3786 |
| 4930       | 4     | 2.1723 | 0.9714 | 4970       | 2     | 6.1142 | 8.6893 |
| 4931       | 4     | 1.0862 | 0.4857 | 4971       | 3     | 8.0571 | 4.3446 |
| 4932       | 2     | 5.5431 | 5.2429 | 4972       | 3     | 9.0286 | 2.1723 |
| 4933       | 3     | 7.7715 | 2.6214 | 4973       | 3     | 9.5143 | 1.0862 |
| 4934       | 4     | 3.8858 | 1.3107 | 4974       | 1     | 4.7571 | 5.5431 |
| 4935       | 1     | 1.9429 | 5.6554 | 4975       | 2     | 7.3786 | 7.7715 |
| 4936       | 1     | 0.9714 | 7.8277 | 4976       | 3     | 8.6893 | 3.8858 |
| 4937       | 1     | 0.4857 | 8.9138 | 4977       | 4     | 4.3446 | 1.9429 |
| 4938       | 3     | 5.2429 | 4.4569 | 4978       | 4     | 2.1723 | 0.9714 |
| 4939       | 4     | 2.6214 | 2.2285 | 4979       | 4     | 1.0862 | 0.4857 |
| 4940       | 1     | 1.3107 | 6.1142 | 4980       | 2     | 5.5431 | 5.2429 |
| 4941       | 2     | 5.6554 | 8.0571 | 4981       | 3     | 7.7715 | 2.6214 |
| 4942       | 2     | 7.8277 | 9.0286 | 4982       | 4     | 3.8858 | 1.3107 |
| 4943       | 2     | 8.9138 | 9.5143 | 4983       | 1     | 1.9429 | 5.6554 |
| 4944       | 4     | 4.4569 | 4.7571 | 4984       | 1     | 0.9714 | 7.8277 |
| 4945       | 1     | 2.2285 | 7.3786 | 4985       | 1     | 0.4857 | 8.9138 |
| 4946       | 2     | 6.1142 | 8.6893 | 4986       | 3     | 5.2429 | 4.4569 |
| 4947       | 3     | 8.0571 | 4.3446 | 4987       | 4     | 2.6214 | 2.2285 |
| 4948       | 3     | 9.0286 | 2.1723 | 4988       | 1     | 1.3107 | 6.1142 |
| 4949       | 3     | 9.5143 | 1.0862 | 4989       | 2     | 5.6554 | 8.0571 |
| 4950       | 1     | 4.7571 | 5.5431 | 4990       | 2     | 7.8277 | 9.0286 |
| 4951       | 2     | 7.3786 | 7.7715 | 4991       | 2     | 8.9138 | 9.5143 |
| 4952       | 3     | 8.6893 | 3.8858 | 4992       | 4     | 4.4569 | 4.7571 |
| 4953       | 4     | 4.3446 | 1.9429 | 4993       | 1     | 2.2285 | 7.3786 |
| 4954       | 4     | 2.1723 | 0.9714 | 4994       | 2     | 6.1142 | 8.6893 |
| 4955       | 4     | 1.0862 | 0.4857 | 4995       | 3     | 8.0571 | 4.3446 |
| 4956       | 2     | 5.5431 | 5.2429 | 4996       | 3     | 9.0286 | 2.1723 |
| 4957       | 3     | 7.7715 | 2.6214 | 4997       | 3     | 9.5143 | 1.0862 |
| 4958       | 4     | 3.8858 | 1.3107 | 4998       | 1     | 4.7571 | 5.5431 |
| 4959       | 1     | 1.9429 | 5.6554 | 4999       | 2     | 7.3786 | 7.7715 |
| 4960       | 1     | 0.9714 | 7.8277 | 5000       | 3     | 8.6893 | 3.8858 |

Gambar 4.81 Koordinat titik baru aturan *non-random* 4

Variasi rotasi dengan aturan *non-random* 5 dengan 28 pola pemilihan titik, pada iterasi  $t$  yaitu  $t \equiv 4 \bmod (7)$  melompati satu titik dari titik terakhir, pada iterasi  $t \equiv 5 \bmod (7)$  berulang seperti titik terakhir yang dipilih, pada iterasi  $t \equiv 6 \bmod (7)$  melompati satu titik dari titik terakhir lagi, dan pada kelipatan 7 berulang lagi satu titik dari titik terakhir yang dipilih ( $A_1 - A_2 - A_3 - A_1 - A_1 - A_3 - A_3 - A_4 - A_1 - A_2 - A_4 - A_1 - A_2 - A_4 - A_4 - A_2 - A_2 - A_3 - A_4 - A_1 - A_3 - A_3 - A_1 - A_1 - A_2 - A_3 - A_4 - A_2 - A_2 - A_4 - A_4$ ) seperti yang ditunjukkan pada Gambar 4.77 menghasilkan titik-titik hasil bentukan yang banyak seperti aturan *non-random* 4. Sama pula seperti sebelumnya, semakin besar sudut rotasinya, maka titik hasil bentukannya semakin jauh keluar dari segiempat. Namun hasilnya juga tidak menunjukkan objek fraktal karena tidak terdapat kemiripan bentuk. Titik-titik hasil bentukannya juga “konvergen” sebanyak pola titik yang ditentukan, sama seperti seluruh aturan *non-random* yang sebelumnya telah dilakukan. Berikut koordinat titik yang dihasilkan dari dua iterasi terakhir aturan *non-random* 5 ditunjukkan pada Gambar 4.82.

| Titik Baru |  | pivot | x      | y      | Titik Baru |  | pivot | x      | y      |
|------------|--|-------|--------|--------|------------|--|-------|--------|--------|
| 4920       |  |       |        |        | 4960       |  |       |        |        |
| 4921       |  | 1     | 2.0146 | 7.8749 | 4961       |  | 1     | 1.9414 | 8.5005 |
| 4922       |  | 2     | 6.0073 | 8.9374 | 4962       |  | 3     | 5.9707 | 4.2502 |
| 4923       |  | 3     | 8.0037 | 4.4687 | 4963       |  | 3     | 7.9854 | 2.1251 |
| 4924       |  | 4     | 4.0018 | 2.2344 | 4964       |  | 4     | 3.9927 | 1.0626 |
| 4925       |  | 2     | 7.0009 | 6.1172 | 4965       |  | 1     | 1.9963 | 5.5313 |
| 4926       |  | 2     | 8.5005 | 8.0586 | 4966       |  | 2     | 5.9982 | 7.7656 |
| 4927       |  | 4     | 4.2502 | 4.0293 | 4967       |  | 4     | 2.9991 | 3.8828 |
| 4928       |  | 4     | 2.1251 | 2.0146 | 4968       |  | 4     | 1.4995 | 1.9414 |
| 4929       |  | 1     | 1.0626 | 6.0073 | 4969       |  | 2     | 5.7498 | 5.9707 |
| 4930       |  | 2     | 5.5313 | 8.0037 | 4970       |  | 2     | 7.8749 | 7.9854 |
| 4931       |  | 3     | 7.7656 | 4.0018 | 4971       |  | 3     | 8.9374 | 3.9927 |
| 4932       |  | 1     | 3.8828 | 7.0009 | 4972       |  | 4     | 4.4687 | 1.9963 |
| 4933       |  | 1     | 1.9414 | 8.5005 | 4973       |  | 1     | 2.2344 | 5.9982 |
| 4934       |  | 3     | 5.9707 | 4.2502 | 4974       |  | 3     | 6.1172 | 2.9991 |
| 4935       |  | 3     | 7.9854 | 2.1251 | 4975       |  | 3     | 8.0586 | 1.4995 |
| 4936       |  | 4     | 3.9927 | 1.0626 | 4976       |  | 1     | 4.0293 | 5.7498 |
| 4937       |  | 1     | 1.9963 | 5.5313 | 4977       |  | 1     | 2.0146 | 7.8749 |
| 4938       |  | 2     | 5.9982 | 7.7656 | 4978       |  | 2     | 6.0073 | 8.9374 |
| 4939       |  | 4     | 2.9991 | 3.8828 | 4979       |  | 3     | 8.0037 | 4.4687 |
| 4940       |  | 4     | 1.4995 | 1.9414 | 4980       |  | 4     | 4.0018 | 2.2344 |
| 4941       |  | 2     | 5.7498 | 5.9707 | 4981       |  | 2     | 7.0009 | 6.1172 |
| 4942       |  | 2     | 7.8749 | 7.9854 | 4982       |  | 2     | 8.5005 | 8.0586 |
| 4943       |  | 3     | 8.9374 | 3.9927 | 4983       |  | 4     | 4.2502 | 4.0293 |
| 4944       |  | 4     | 4.4687 | 1.9963 | 4984       |  | 4     | 2.1251 | 2.0146 |
| 4945       |  | 1     | 2.2344 | 5.9982 | 4985       |  | 1     | 1.0626 | 6.0073 |
| 4946       |  | 3     | 6.1172 | 2.9991 | 4986       |  | 2     | 5.5313 | 8.0037 |
| 4947       |  | 3     | 8.0586 | 1.4995 | 4987       |  | 3     | 7.7656 | 4.0018 |
| 4948       |  | 1     | 4.0293 | 5.7498 | 4988       |  | 1     | 3.8828 | 7.0009 |
| 4949       |  | 1     | 2.0146 | 7.8749 | 4989       |  | 1     | 1.9414 | 8.5005 |
| 4950       |  | 2     | 6.0073 | 8.9374 | 4990       |  | 3     | 5.9707 | 4.2502 |
| 4951       |  | 3     | 8.0037 | 4.4687 | 4991       |  | 3     | 7.9854 | 2.1251 |
| 4952       |  | 4     | 4.0018 | 2.2344 | 4992       |  | 4     | 3.9927 | 1.0626 |
| 4953       |  | 2     | 7.0009 | 6.1172 | 4993       |  | 1     | 1.9963 | 5.5313 |
| 4954       |  | 2     | 8.5005 | 8.0586 | 4994       |  | 2     | 5.9982 | 7.7656 |
| 4955       |  | 4     | 4.2502 | 4.0293 | 4995       |  | 4     | 2.9991 | 3.8828 |
| 4956       |  | 4     | 2.1251 | 2.0146 | 4996       |  | 4     | 1.4995 | 1.9414 |
| 4957       |  | 1     | 1.0626 | 6.0073 | 4997       |  | 2     | 5.7498 | 5.9707 |
| 4958       |  | 2     | 5.5313 | 8.0037 | 4998       |  | 2     | 7.8749 | 7.9854 |
| 4959       |  | 3     | 7.7656 | 4.0018 | 4999       |  | 3     | 8.9374 | 3.9927 |
| 4960       |  | 1     | 3.8828 | 7.0009 | 5000       |  | 4     | 4.4687 | 1.9963 |

Gambar 4.82 Koordinat titik baru aturan *non-random* 5

Titik-titik hasil bentukan *non-random chaos game* berkelompok di suatu koordinat tertentu dikarenakan pemilihan titik sudut yang berpola teratur secara terus menerus. Titik-titik hasil bentukannya menyebar tidak merata dan hanya “konvergen” ke titik tertentu sebanyak pola yang ditentukan, dan bukan konvergen pada empat titik sudut tumpuannya seperti yang dikatakan pada penelitian sebelumnya. Istilah “konvergen” disini berbeda dengan istilah matematis pada umumnya, dimana “konvergen” berlaku pada *chaos game* yang menunjukkan letak titik baru yang berkumpul pada koordinat tertentu. Hal ini menyebabkan hasil dari *chaos game* dengan aturan *non-random* tidak membentuk fraktal karena tidak memiliki kemiripan bentuk satu sama lain (*self-similarity*). Berbeda dengan aturan *random*, dimana seluruh modifikasinya membentuk objek fraktal dikarenakan pemilihan titiknya yang dilakukan secara acak. Menurut Sulistiyantoko (2008) ini disebabkan karena prinsip utama *chaos game* adalah menggunakan teori *chaos*, yaitu gambaran pergerakan rumit dan perilaku dari sistem dinamis non-linier tertentu yang pergerakannya sangat bergantung pada keadaan awal. Pembentukan *chaos game* dengan pemilihan titik dilakukan secara *non-random* dengan pola teratur yang ditentukan sebanyak apapun tetap tidak akan membentuk fraktal dikarenakan pemilihan titiknya yang teratur dan terus berulang, sehingga titik akan berkumpul pada suatu titik tertentu. Ayuningtyas (2019) pada penelitian sebelumnya yang melakukan penggabungan aturan *random* dan *non-random* juga tidak menunjukkan objek fraktal dikarenakan adanya keterlibatan pola *non-random*, sehingga tetap tidak membentuk fraktal, walaupun ada beberapa pola yang hampir seperti bentuk fraktal, namun kurang identik dan tetap dikatakan bahwa objek tersebut bukan merupakan fraktal.

## BAB 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

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

1. *Chaos game* segiempat yang dilakukan secara *random* dan semua modifikasinya yaitu modifikasi 1 aturan *random* dengan syarat titik terakhir yang telah dipilih tidak boleh dipilih kembali, modifikasi 2 aturan *random* dengan syarat titik yang berhadapan tidak boleh dipilih, modifikasi 3 aturan *random* dengan syarat titik sebelah kiri tidak boleh dipilih, modifikasi 4 aturan *random* dengan syarat titik sebelah kanan tidak boleh dipilih, seluruh variasi rotasinya menghasilkan objek fraktal karena memiliki kemiripan bentuk satu sama lain (*self-similarity*).
2. *Chaos game* segiempat yang dilakukan secara *non-random* dengan beberapa modifikasi dengan pola pemilihan titik berulang setiap 4 pola, 16 pola, 20 pola, 24 pola, dan 28 pola seluruhnya tidak membentuk objek fraktal dikarenakan titik-titik hasil bentukannya saling berjauhan sehingga tidak memiliki kemiripan bentuk satu sama lain (*self-similarity*), tetapi memiliki karakteristik yaitu “konvergen” berulang pada titik koordinat tertentu sebanyak pola pemilihan titik yang ditentukan.

### 5.2 Saran

Pada penelitian ini dilakukan pembentukan variasi rotasi pada *chaos game* segiempat yang dilakukan secara *random* dengan beberapa modifikasinya serta aturan *non-random*. Penelitian selanjutnya diharapkan dapat menunjukkan modifikasi lain seperti perpaduan variasi rotasi dan rasio kompresinya atau menunjukkan suatu objek dikatakan fraktal dengan menggunakan kajian analitik.

## DAFTAR PUSTAKA

- Ayuningtyas, M. H. 2019. *Modifikasi Aturan Chaos Game pada Segilima*. Skripsi. Jember: Fakultas MIPA Universitas Jember.
- Devaney, R. L. 2003. *Fractal Patterns and Chaos game*. Boston: Department of Mathematics Boston University.
- Gerald, Edgar. 2008. *Measure, Topology, and Fractal Geometry*. New York: Springer Science Business Media.
- Hasang, S. dan Supardjo, S. 2012. *Geometri Fraktal dalam Rancangan Arsitektur*. Media Martasain. Manado: Universitas Samratulangi.
- Merriam Webster Dictionary. 1975. *Encyclopaedia Britannica Online*. <https://www.merriam-webster.com/dictionary/fractal#learn-more>. [Diakses tanggal 11 November 2018].
- Miller, C. 2011. *Communicating Mathematics III: Z-Corp* 650. <http://www.maths.dur.ac.uk>. [Diakses 03 April 2019].
- Purnomo, K. D. 2014. Pembangkitan Segitiga Sierpinski dengan Transformasi Affine Berbasis Beberapa Benda Geometris. *Prosiding Seminar Nasional Matematika*. Jember: Fakultas MIPA Universitas Jember.
- Purnomo, K. D., Armana, R. F., dan Kusno. 2016. Kajian Pembentukan Segitiga Sierpinski pada Masalah Chaos game dengan Memanfaatkan Transformasi Affine. *Jurnal Matematika Vol. 6 No. 2*:86-92.
- Riyadi, B. 2007. *Analisis dan Perancangan Perangkat Lunak Generator Gambardan Musik Fraktal dengan Iterated Function System*. Skripsi. Jakarta: Universitas Bina Nusantara.
- Romadiastri, Y. 2013. *Batik Fraktal. Perkembangan Aplikasi Geometri Fraktal*. Semarang: Jurusan Tadris Matematika Fakultas Ilmu Tarbiyah dan Keguruan IAIN Walisongo.
- Shenker, O. R. 1994. *Fractal Geometry is not the Geometry of Nature*. Jerusalem: The Hebrew University of Jerusalem.

- Sulastri. 2007. Transformasi Bangun Ruang Tiga Dimensi Menggunakan Visual Basic 6.0. *Jurnal Teknologi Informasi DINAMIK Volume XII, No. 1:88-100*. Fakultas Teknologi Informasi Universitas Stikubank Semarang.
- 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.
- Sundbye, L. 1997. *Chaos and Fractals on the TI Graphing Calculator*. Department of Mathematical and Computer Sciences: Metropolitan State College of Denver.
- Wikipedia The Free Encylopaedia. [https://en.wikipedia.org/wiki/Chaos\\_game](https://en.wikipedia.org/wiki/Chaos_game). [Diakses tanggal 23 November 2018].
- 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

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

% Last Modified by GUIDE v2.5 12-Apr-2019 10:37:12

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

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

% Update handles structure
guidata(hObject, handles);
clc;
movegui(gcf, 'center');
set(handles.uitable1, 'data', [], 'rowname', 'P');
set(handles.uitable2, 'data', [], 'rowname', 'P', 'columnwidth', {50, 60,
60});
set(handles.uitable3, 'data', [], 'rowname', 'P', 'columnname', 1:4, 'col
umnwidth', {54, 54, 54, 54});
set(handles.edit1, 'string', '');
set(handles.edit2, 'string', '');
set(handles.edit3, 'string', '');
set(handles.radiobutton1, 'value', 0);
set(handles.radiobutton2, 'value', 0);
set(handles.radiobutton3, 'value', 0);
set(handles.radiobutton4, 'value', 0);
set(handles.radiobutton5, 'value', 0);
set(handles.radiobutton6, 'value', 0);
set(handles.radiobutton7, 'value', 0);
set(handles.radiobutton8, 'value', 0);
set(handles.radiobutton9, 'value', 0);
set(handles.radiobutton10, 'value', 0);
set(handles.text5, 'string', '0');
cla(handles.axes1, 'reset');
set(handles.axes1, 'Xlim', [0 10], 'YLim', [0 10]);
xlabel('x'); ylabel('y');
grid on;
% UIWAIT makes MyProgram wait for user response (see UIRESUME)
% uiwait(handles.figure1);

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

```

```

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

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
clc;
set(handles.uitable2,'data',[],'rowname','P');
set(handles.uitable3,'data',[],'rowname','P','columnname',1:4);
set(handles.text5,'string','0');
cla(handles.axes1,'reset');
coor=[0 10 10 0;10 10 0 0];
warna=[1 0 0;0 1 0;1 0 1;0 0 1];
plot([coor(1,:) coor(1,1)],[coor(2,:) coor(2,1)]);
for i=1:4

line(coor(1,i),coor(2,i),'Marker','o','MarkerEdgeColor',warna(i,:),
,'MarkerFaceColor',warna(i,:), 'MarkerSize',2);
    text(coor(1,i),coor(2,i),['A' num2str(i)]);
end
axis image
grid on;
xmin=min(coor(1,:));
xmax=max(coor(1,:));
ymin=min(coor(2,:));
ymax=max(coor(2,:));
xsel=xmax-xmin;
ysel=ymax-ymin;
xlim([xmin-xsel/5 xmax+xsel/5]);
ylim([ymin-ysel/5 ymax+ysel/5]);

% --- 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;
set(handles.uitable2,'data',[],'rowname','P');
set(handles.uitable3,'data',[],'rowname','P','columnname',1:4);
set(handles.text5,'string','0');
cla(handles.axes1,'reset');
coor=[0 10 10 0;10 10 0 0];
warna=[1 0 0;0 1 0;1 0 1;0 0 1];
plot([coor(1,:) coor(1,1)],[coor(2,:) coor(2,1)]);
for i=1:4

line(coor(1,i),coor(2,i),'Marker','o','MarkerEdgeColor',warna(i,:),
,'MarkerFaceColor',warna(i,:), 'MarkerSize',2);
    text(coor(1,i),coor(2,i),['A' num2str(i)]);
end
axis image

```

```

grid on;
xmin=min(coor(1,:));
xmax=max(coor(1,:));
ymin=min(coor(2,:));
ymax=max(coor(2,:));
xsel=xmax-xmin;
ysel=ymax-ymin;
xlim([xmin-xsel/5 xmax+xsel/5]);
ylim([ymin-ysel/5 ymax+ysel/5]);
[x,y]=getpts(handles.axes1);
set(handles.uitable1,'data',[x(1),y(1)],'rowname','P');
line(x(1),y(1),'Marker','o','MarkerEdgeColor','k','MarkerFaceColor',
'k','MarkerSize',1);

```

```

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

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

```

```

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

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

```

```

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

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

```

```

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

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

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

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

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

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

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
set(handles.uitable2,'data',[],'rowname','P');
set(handles.uitable3,'data',[],'rowname','P','columnname',1:4);
set(handles.text5,'string','0');

```

```

cla(handles.axes1, 'reset');
Alpha=str2num(get(handles.edit1, 'string'));
Theta=str2num(get(handles.edit2, 'string'));
Iteration=str2num(get(handles.edit3, 'string'));
rb1=get(handles.radiobutton1, 'value');
rb2=get(handles.radiobutton2, 'value');
rb3=get(handles.radiobutton3, 'value');
rb4=get(handles.radiobutton4, 'value');
rb5=get(handles.radiobutton5, 'value');
rb6=get(handles.radiobutton6, 'value');
rb7=get(handles.radiobutton7, 'value');
rb8=get(handles.radiobutton8, 'value');
rb9=get(handles.radiobutton9, 'value');
rb10=get(handles.radiobutton10, 'value');

coor=[0 10 10 0;10 10 0 0];
warna=[1 0 0;0 1 0;1 0 1;0 0 1];
point=get(handles.uitable1, 'data');

plot([coor(1,:) coor(1,1)], [coor(2,:) coor(2,1)]);
for i=1:4

line(coor(1,i), coor(2,i), 'Marker', 'o', 'MarkerEdgeColor', warna(i,:),
, 'MarkerFaceColor', warna(i,:), 'MarkerSize', 2);
    text(coor(1,i), coor(2,i), ['A' num2str(i)]);
end
axis image
grid on;
xmin=min(coor(1,:));
xmax=max(coor(1,:));
ymin=min(coor(2,:));
ymax=max(coor(2,:));
xsel=xmax-xmin;
ysel=ymax-ymin;
xlim([xmin-xsel/5 xmax+xsel/5]);
ylim([ymin-ysel/5 ymax+ysel/5]);
line(point(1), point(2), 'Marker', 'o', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'k', 'MarkerSize', 1);

ppoint=[]; npoint=[];
for i=1:Iteration
    if rb1==1
        if i==1
            pivot=ceil(rand*4);
        else
            pivot=ceil(rand*4);
            while pivot==ppoint(end)
                pivot=ceil(rand*4);
            end
        end
    elseif rb2==1
        if i==1
            pivot=ceil(rand*4);
        else
            pivot=ceil(rand*4);
            while pivot==mod(ppoint(end)+1, 4)+1

```

```

        pivot=ceil(rand*4);
    end
end
elseif rb3==1
    if i==1
        pivot=ceil(rand*4);
    else
        pivot=ceil(rand*4);
        while pivot==mod(ppoint(end)-2,4)+1
            pivot=ceil(rand*4);
        end
    end
end
elseif rb4==1
    if i==1
        pivot=ceil(rand*4);
    else
        pivot=ceil(rand*4);
        while pivot==mod(ppoint(end),4)+1
            pivot=ceil(rand*4);
        end
    end
end
elseif rb5==1
    pivot=ceil(rand*4);
elseif rb6==1
    pivot=mod(i-1,4)+1;
elseif rb7==1
    if i==1
        pivot=1;
    else
        if mod(i,4)==0
            pivot=ppoint(end);
        else
            pivot=mod(ppoint(end),4)+1;
        end
    end
end
elseif rb8==1
    if i==1
        pivot=1;
    else
        if mod(i,5)==0 || mod(i,5)==4
            pivot=mod(ppoint(end)+1,4)+1;
        else
            pivot=mod(ppoint(end),4)+1;
        end
    end
end
elseif rb9==1
    if i==1
        pivot=1;
    else
        if mod(i,6)==4 || mod(i,6)==5
            pivot=ppoint(end);
        elseif mod(i,6)==0
            pivot=mod(ppoint(end)+1,4)+1;
        else
            pivot=mod(ppoint(end),4)+1;
        end
    end
end

```

```

        end
    elseif rb10==1
        if i==1
            pivot=1;
        else
            if mod(i,7)==0 || mod(i,7)==5
                pivot=ppoint(end);
            elseif mod(i,7)==4 || mod(i,7)==6
                pivot=mod(ppoint(end)+1,4)+1;
            else
                pivot=mod(ppoint(end),4)+1;
            end
        end
    end
    end
    point=(coor(:,pivot)+Alpha*eye(2)*[cosd(Theta) -
sind(Theta);sind(Theta) cosd(Theta)]*(point'-coor(:,pivot)))';
    ppoint=[ppoint pivot];
    npoint=[npoint;point];
    set(handles.uitable2,'data',[ppoint'
npoint'],'rowname','numbered');
    if rb1==1 || rb2==1 || rb3==1 || rb4==1 || rb5==1

line(point(1),point(2),'Marker','o','MarkerEdgeColor',warna(pivot,
:),'MarkerFaceColor',warna(pivot,:), 'MarkerSize',1);
    elseif rb6==1 || rb7==1 || rb8==1 || rb9==1 || rb10==1

line(point(1),point(2),'Marker','o','MarkerEdgeColor',warna(pivot,
:),'MarkerFaceColor',warna(pivot,:), 'MarkerSize',5);
    end
    axis image
    xmin=min(min(coor(1,:)),min(npoint(:,1)));
    xmax=max(max(coor(1,:)),max(npoint(:,1)));
    ymin=min(min(coor(2,:)),min(npoint(:,2)));
    ymax=max(max(coor(2,:)),max(npoint(:,2)));
    xsel=xmax-xmin;
    ysel=ymax-ymin;
    xlim([xmin-xsel/8 xmax+xsel/8]);
    ylim([ymin-ysel/8 ymax+ysel/8]);
    set(handles.text5,'string',num2str(i));
    pause(0.0001);
end
for i=1:4
    piv=find(ppoint==i);
    Peluang(i)=length(piv)/Iteration;
end
set(handles.uitable3,'data',Peluang,'rowname','P','columnname','nu
mbered');

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton4 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
clc;
movegui(gcf,'center');

```

```

set(handles.uitable1,'data',[],'rowname','P');
set(handles.uitable2,'data',[],'rowname','P');
set(handles.uitable3,'data',[],'rowname','P','columnname',1:4);
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.radiobutton1,'value',0);
set(handles.radiobutton2,'value',0);
set(handles.radiobutton3,'value',0);
set(handles.radiobutton4,'value',0);
set(handles.radiobutton5,'value',0);
set(handles.radiobutton6,'value',0);
set(handles.radiobutton7,'value',0);
set(handles.radiobutton8,'value',0);
set(handles.radiobutton9,'value',0);
set(handles.radiobutton10,'value',0);
set(handles.text5,'string','0');
cla(handles.axes1,'reset');
set(handles.axes1,'Xlim',[0 10],'YLim',[0 10]);
xlabel('x'); ylabel('y');
grid on;

```

```

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton5 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
clc;
axes(handles.axes1);
grid off;
frame=getframe(handles.axes1);
image=frame2im(frame);
[name_file,name_path] = uiputfile( ...
    {'*.jpg','File Type JPEG (*.jpg)';
    '*.bmp','File Type BITMAP (*.bmp)';
    '*.tif','File Type TIF (*.tif)';
    '*.png','File Type PNG (*.png)';
    '*.jpg;*.bmp;*.tif;*.png','Files of type
(*.jpg,*.bmp,*.tif,*.png)'},...
    'Save As Image');
if name_file~=0
    imwrite(image,fullfile(name_path,name_file));
end
axes(handles.axes1);
grid on;

```

```

% --- 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.radiobutton6,'value',0);
set(handles.radiobutton7,'value',0);
set(handles.radiobutton8,'value',0);
set(handles.radiobutton9,'value',0);
set(handles.radiobutton10,'value',0);
% 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);
set(handles.radiobutton6,'value',0);
set(handles.radiobutton7,'value',0);
set(handles.radiobutton8,'value',0);
set(handles.radiobutton9,'value',0);
set(handles.radiobutton10,'value',0);
% Hint: get(hObject,'Value') returns toggle state of radiobutton2

```

```

% --- 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.radiobutton6,'value',0);
set(handles.radiobutton7,'value',0);
set(handles.radiobutton8,'value',0);
set(handles.radiobutton9,'value',0);
set(handles.radiobutton10,'value',0);
% 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.radiobutton6,'value',0);
set(handles.radiobutton7,'value',0);
set(handles.radiobutton8,'value',0);
set(handles.radiobutton9,'value',0);
set(handles.radiobutton10,'value',0);
% 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.radiobutton6,'value',0);
set(handles.radiobutton7,'value',0);
set(handles.radiobutton8,'value',0);
set(handles.radiobutton9,'value',0);
set(handles.radiobutton10,'value',0);
% Hint: get(hObject,'Value') returns toggle state of radiobutton5

```

```

% --- Executes on button press in radiobutton6.
function radiobutton6_Callback(hObject, eventdata, handles)
% hObject      handle to radiobutton6 (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',0);
set(handles.radiobutton6,'value',1);
set(handles.radiobutton7,'value',0);
set(handles.radiobutton8,'value',0);
set(handles.radiobutton9,'value',0);
set(handles.radiobutton10,'value',0);
% Hint: get(hObject,'Value') returns toggle state of radiobutton6

```

```

% --- Executes on button press in radiobutton7.
function radiobutton7_Callback(hObject, eventdata, handles)
% hObject      handle to radiobutton7 (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',0);
set(handles.radiobutton6,'value',0);
set(handles.radiobutton7,'value',1);
set(handles.radiobutton8,'value',0);
set(handles.radiobutton9,'value',0);
set(handles.radiobutton10,'value',0);
% Hint: get(hObject,'Value') returns toggle state of radiobutton7

```

```

% --- Executes on button press in radiobutton8.
function radiobutton8_Callback(hObject, eventdata, handles)
% hObject      handle to radiobutton8 (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',0);
set(handles.radiobutton6,'value',0);
set(handles.radiobutton7,'value',0);
set(handles.radiobutton8,'value',1);
set(handles.radiobutton9,'value',0);
set(handles.radiobutton10,'value',0);
% Hint: get(hObject,'Value') returns toggle state of radiobutton8

```

```

% --- Executes on button press in radiobutton9.
function radiobutton9_Callback(hObject, eventdata, handles)
% hObject      handle to radiobutton9 (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',0);
set(handles.radiobutton6,'value',0);
set(handles.radiobutton7,'value',0);
set(handles.radiobutton8,'value',0);
set(handles.radiobutton9,'value',1);
set(handles.radiobutton10,'value',0);
% Hint: get(hObject,'Value') returns toggle state of radiobutton9

```

```

% --- Executes on button press in radiobutton10.
function radiobutton10_Callback(hObject, eventdata, handles)
% hObject      handle to radiobutton10 (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',0);
set(handles.radiobutton6,'value',0);
set(handles.radiobutton7,'value',0);
set(handles.radiobutton8,'value',0);
set(handles.radiobutton9,'value',0);
set(handles.radiobutton10,'value',1);
% Hint: get(hObject,'Value') returns toggle state of radiobutton10
```