

# Penerapan Rekursi dalam Pembuatan Segitiga Sierpinski dengan Menggunakan ActionScript 3

Adin Baskoro Pratomo - 13513058

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13513058@std.stei.itb.ac.id

**Abstrak**—Fractal adalah pola bentuk yang berulang. Terdapat berbagai cara untuk membuat *fractal*, baik manual maupun dengan bantuan alat digital seperti komputer. *Fractal* biasanya dibuat dengan metode rekursif, karena sifat-sifatnya. Salah satu contoh *fractal* adalah Segitiga Sierpinski. Pada makalah ini, akan dibahas penerapan rekursi dalam pembuatan Segitiga Sierpinski dengan bahasa pemrograman ActionScript 3

**Kata Kunci**—ActionScript 3, Fractal, Rekursif, Segitiga Sierpinski.

## I. PENDAHULUAN

*Fractal* merupakan pola bentuk yang berulang, dan memiliki bentuk yang mirip atau sama dengan bentuk pendahulunya, sedangkan *Fractal Art* adalah salah satu bentuk karya seni dari *Fractal* yang digemari orang banyak. *Fractal Art* digemari karena bentuknya yang indah untuk dilihat dan unik. Banyak seniman-seniman ternama yang hanya berfokus pada *fractal art* sebagai karyanya.

*Fractal* sebenarnya bukan hanya pola buatan manusia. Di alam terdapat berbagai macam pola yang berulang, yang terbentuk secara alami, missal pohon, awan, sungai, dan lain sebagainya. Oleh karenanya, *Fractal* sudah tidak asing bagi kebanyakan orang.

Ada banyak cara untuk membuat fraktal. Namun, kebanyakan seniman membuatnya dengan bantuan program. Program tersebut akan menerima masukan fungsi atau algoritma dari seniman, kemudian membentuknya menjadi gambar.

Untuk membuat fractal selain dengan menggunakan program yang khusus untuk membuat fractal, kita dapat pula membuat sendiri algoritma untuk membuat fractal tersebut yang kemudian diimplementasikan dengan menggunakan bahasa pemrograman tertentu.

Segitiga Sierpinski merupakan salah satu objek *fractal*. Segitiga tersebut sangat menarik karena polanya yang berulang. Meski diperbesar berapa kali pun tetap akan muncul pola yang sama. Dapat dikatakan bahwa Segitiga Sierpinski adalah salah satu bentuk dari *fractal*

*art* sederhana.

Pada makalah ini, penulis akan membahas dan menerapkan algoritma untuk membuat Segitiga Sierpinski. Penulis akan menggunakan bahasa pemrograman ActionScript 3 sebagai media untuk implementasi algoritma, sehingga dihasilkan gambar *fractal* dari algoritma tersebut.

## II. DASAR TEORI

### 1. Rekursi

Rekursi merupakan proses sebuah objek mendefinisikan dirinya sendiri di dalam terminologi dirinya sendiri. Objek-objek yang melakukan rekursi disebut objek rekursif. Rekursi memungkinkan seseorang mendefinisikan suatu objek secara tak terbatas dengan pernyataan yang terbatas, atau melakukan operasi sebanyak tak hingga, meski tanpa repetisi yang eksplisit[1].

Di dalam bidang informatika, rekursi digunakan dalam pendefinisian fungsi rekursif. Fungsi rekursif adalah suatu fungsi yang akan memanggil dirinya sendiri ketika fungsi tersebut dijalankan, sehingga akan dihasilkan perulangan akibat pemanggilan fungsi yang sama secara berulang-ulang. Fungsi rekursif banyak dimanfaatkan sebagai alternatif dari iterasi pada program, dan biasanya digunakan apabila suatu objek yang akan diproses memiliki pola tertentu yang mirip dengan sebagian dari objek tersebut



**Gambar 2-1** Rekursi pada perangkat lunak perekam layar.

Sumber :

[http://upload.wikimedia.org/wikipedia/commons/thumb/b/b3/Screenshot\\_Recursion\\_via\\_vlc.png/220px-Screenshot\\_Recursion\\_via\\_vlc.png](http://upload.wikimedia.org/wikipedia/commons/thumb/b/b3/Screenshot_Recursion_via_vlc.png/220px-Screenshot_Recursion_via_vlc.png)

Suatu fungsi yang rekursif, terdiri dari dua bagian utama, basis dan rekurens. Basis merupakan bagian yang berisi nilai secara eksplisit dan berfungsi untuk menghentikan perulangan yang dilakukan rekurens. Sedangkan rekurens merupakan bagian dari objek yang mendefinisikan fungsi itu sendiri. Suatu rekurens harus mengarah ke basis, dengan kata lain, rekurens pasti berhenti ketika sudah bertemu basis. Bila tidak, akan terjadi *infinite loop* di dalam fungsi tersebut yang tentu tidak diinginkan.

Salah satu contoh sederhana dari fungsi rekursif adalah barisan Fibonacci. Barisan Fibonacci adalah bilangan yang nilai suku ke- $n$  nya adalah jumlah dari dua suku sebelumnya. Oleh karena itu, barisan Fibonacci dapat didefinisikan sebagai fungsi rekursif, yaitu:

$$f_n = \begin{cases} 0 & , n = 0 \\ 1 & , n = 1 \\ f_{n-1} + f_{n-2} & , n > 1 \end{cases}$$

Dari contoh tersebut, terlihat bahwa barisan Fibonacci terdiri dari dua basis. Nilai suku ke- $n$  akan bernilai 0 jika  $n = 0$ , dan akan bernilai 1 ketika  $n = 1$ . Dari bagian rekurens, terlihat bahwa nilai suku ke- $n$  bergantung pada nilai dua suku sebelumnya, suku  $n - 1$  dan suku  $n - 2$ . Pada bagian rekurens nilai suku ke  $n$  ketika  $n > 1$  didefinisikan sebagai jumlah dari nilai suku ke  $n - 1$  dan nilai dari suku ke  $n - 2$ . Dari bagian rekurens terlihat pula bahwa rekurens akan menuju ke basis, karena nilai  $n$  selalu dikurangi dengan 1. Ketika  $n$  sudah mencapai 0, basis sudah tercapai dan fungsi akan berhenti memanggil dirinya sendiri. Sebagai contoh, untuk mencari suku ke-4 dari barisan Fibonacci, urutan proses yang dilakukan adalah sebagai berikut:

$$\begin{aligned} f_4 &= f_3 + f_2 \\ f_3 &= f_2 + f_1 \\ f_2 &= f_1 + f_0 = 1 + 0 = 1 \\ f_3 &= f_2 + f_1 = 1 + 1 = 2 \\ f_4 &= f_3 + f_2 = 2 + 1 = 3 \end{aligned}$$

Proses dilakukan secara terus menerus hingga mencapai basis. Setelah mencapai basis, nilai dari basis tersebut digunakan untuk mendapatkan nilai dari fungsi rekurens tersebut.

## 2. Fractal

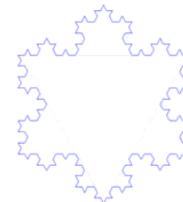
Fractal adalah suatu fenomena atau himpunan matematika yang memiliki ciri-ciri berupa pola

berulang untuk setiap skala. Pola tersebut dapat persis sama untuk setiap skala,

Kata fractal pertama kali digunakan oleh Mandelbrot pada tahun 1975. Asal mula kata fractal yaitu dari kata dalam bahasa latin: *fractus* yang berarti patah.

Fractal berbeda dari betuk geometri biasa. Ketika seluruh sisi dari sebuah polygon panjangnya diubah menjadi dua kali lipat, luasnya akan berubah menjadi empat kali lipat. Namun pada objek fractal, ketika seluruh panjang sisinya diubah menjadi dua kali lipat, luasnya belum tentu menjadi empat kali lipat sebelumnya, namun menjadi perpangkatan dua dari suatu bilangan. Bilangan tersebut belum tentu bilangan bulat dan disebut sebagai *fractal dimension*.

Bagi kebanyakan orang, kata *fractal* sendiri memiliki konotasi yang berbeda dibandingkan bagi metematikawan. Fractal lebih diidentikkan dengan fractal art, yaitu karya seni rupa yang dihasilkan dari objek-objek fractal. Definisi fractal secara matematis cukup rumit untuk dipahami oleh kebanyakan orang. Meski demikian, orang awam dapat memahami ciri-ciri dari objek fractal yang berupa kesamaan pola untuk setiap skala. Beberapa contoh dari objek fractal adalah keeping salju Koch, Segitiga Sierpinski, dan Himpunan Mandelbrot.



**Gambar 2-2** Keeping Salju Koch merupakan objek *fractal*

. Sumber :

[http://upload.wikimedia.org/wikipedia/commons/thumb/f/fd/Von\\_Koch\\_curve.gif/170px-Von\\_Koch\\_curve.gif](http://upload.wikimedia.org/wikipedia/commons/thumb/f/fd/Von_Koch_curve.gif/170px-Von_Koch_curve.gif)

Fractal memiliki aplikasi dalam berbagai macam bidang, misal antenna fractal dan transistor fractal (elektronika dan telekomunikasi), Kompresi citra dan sinyal (informatika), pergerakan saham (ekonomi), Fractal Art (Seni), dan lain-lain.

## 3. Fractal Art

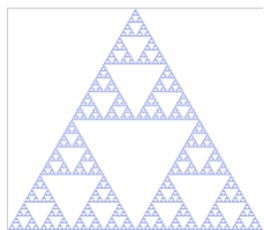
Fractal Art merupakan karya seni yang dibuat dengan menggunakan sifat-sifat fractal, seperti memiliki kesamaan pola ketika diperbesar. Seniman biasanya menggunakan perangkat lunak spesifik untuk membuat pola fractal. Perangkat lunak tersebut akan menghasilkan gambar dari masukan yang diberikan oleh seniman.

Fractal Art sangat populer karena pola yang unik dan proses pembuatannya yang tak lazim dengan menggunakan algoritma, tak seperti karya seni pada umumnya. Selain itu, pada fractal art, nampak terlihat keseimbangan antara simetri, dan kekacauan atau *chaos*, yang juga dapat ditemukan dari lukisan-lukisan Cina dan bonsai.

Pada umumnya, proses pembuatan fractal art dibantu dengan menggunakan computer, karena banyaknya proses perhitungan yang harus dilakukan, yang hanya dimungkinkan dengan penggunaan komputer. Dalam proses pembuatannya, terkadang algoritma yang digunakan digabung dengan *evolutionary algorithm* yang dibantu oleh manusia untuk memberikan hasil yang lebih baik dan indah untuk dilihat.

#### 4. Segitiga Sierpinski

Salah satu contoh objek fractal adalah Segitiga Sierpinski. Segitiga berupa segitiga sama sisi, dan terdiri dari sekumpulan segitiga yang bentuknya sama, dan berulang.



**Gambar 2-2** Segitiga Sierpinski  
. Sumber :

[http://upload.wikimedia.org/wikipedia/commons/thumb/b/4/45/Sierpinski\\_triangle.svg/220px-Sierpinski\\_triangle.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/b/4/45/Sierpinski_triangle.svg/220px-Sierpinski_triangle.svg.png)

Segitiga Sierpinski pertama kali dideskripsikan oleh Waclaw Sierpiński pada tahun 1915. Meski demikian, pola Segitiga Sierpinski sudah ada sejak bertahun-tahun sebelumnya, di beberapa tempat di Italia.

Terdapat berbagai cara untuk membangun sebuah segitiga Sierpinski, diantaranya:

##### a. Menghapus segitiga

Proses ini diawali dengan sebuah segitiga sama sisi. Kemudian, bagi segitiga tersebut menjadi 4 segitiga sama sisi, kemudian hapus bagian tengah dari segitiga tersebut. Ulangi langkah tersebut untuk segitiga yang terbentuk dari langkah sebelumnya.



**Gambar 2-2** Proses pembentukan Segitiga

##### Sierpinski dengan menghapus segitiga

. Sumber :

[http://en.wikipedia.org/wiki/File:Sierpinski\\_triangle\\_evolution.svg](http://en.wikipedia.org/wiki/File:Sierpinski_triangle_evolution.svg)

##### b. Penyusutan dan Penduplikasian

Proses ini diawali dengan sebuah bentuk dalam sebuah bidang. Bentuk yang digunakan tidak perlu berbentuk segitiga. Kemudian kecilkan panjang dan tinggi bentuk tersebut menjadi setengah dari aslinya. Lalu salin bentuk tersebut, tempatkan sedemikian rupa agar ujung bentuk bertemu dengan ujung segitiga lain. Ulangi proses tersebut dengan segitiga yang semakin disusutkan.



**Gambar 2-2** Proses pembentukan Segitiga Sierpinski dengan penyusutan dan penduplikasian

. Sumber :

[http://upload.wikimedia.org/wikipedia/commons/thumb/b/c/c9/Sierpinski\\_triangle\\_evolution\\_square.svg/512px-Sierpinski\\_triangle\\_evolution\\_square.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/b/c/c9/Sierpinski_triangle_evolution_square.svg/512px-Sierpinski_triangle_evolution_square.svg.png)

##### c. Chaos game

Pertama, tentukan 3 titik di sebuah bidang yang membentuk segitiga. Kemudian tentukan sebuah titik secara acak di dalam segitiga. Lalu pilih sebuah titik sudut segitiga yang telah dibentuk secara acak. Berpindahlah sejauh setengah jarak dari posisi sebelumnya dan titik sudut yang dipilih. Tandai posisi yang telah dicapai. Ulangi memilih titik sudut secara acak, dan bergerak sejauh setengah jarak dari posisi yang sedang ditempati ke titik sudut.

Apabila titik yang pertama ditandai terdapat di dalam Segitiga Sierpinski, maka seluruh titik yang ditandai setelahnya pun akan berada dalam segitiga Sierpinski. Sebaliknya, bila titik tidak berada dalam segitiga Sierpinski, titik-titik berikutnya juga tidak akan membentuk segitiga Sierpinski.

Proses pembuatan Segitiga Sierpinski yang akan diimplementasikan menggunakan ActionScript 3 pada makalah ini serupa dengan proses penghapusan segitiga, namun terdapat perbedaan dalam cara membentuk segitiga.

#### 5. ActionScript 3

ActionScript adalah bahasa pemrograman berorientasi objek yang awalnya dikembangkan oleh Macromedia Inc dan kini dikembangkan oleh Adobe System. ActionScript digunakan sebagian besar untuk pengembangan website dan aplikasi yang menggunakan platform Adobe Flash Player. Aplikasi

yang dihasilkan berbentuk file SWF dan di-embed ke dalam halaman situs. ActionScript merupakan bahasa yang open-source, dan tersedia pula *compiler* yang open-source seperti Apache flex. Hingga saat ini, versi ActionScript sudah mencapai ActionScript 3, dan Flash Player sudah mencapai versi 16 untuk windows dan OSX, serta versi 11.2 untuk Linux.

ActionScript 3 atau yang biasa disebut AS3 merupakan hasil pengembangan lebih lanjut dari ActionScript 2 (AS2). ActionScript 3 menggunakan ActionScript Virtual Machine yang telah ditulis ulang dari awal, sehingga tidak kompatibel dengan versi sebelumnya. Namun, ActionScript 3 lebih cepat hingga sepuluh kali lipat bila dibandingkan dengan pendahulunya, karena adanya tambahan *Just-In-Time Compiler*.

Secara sintaks, ActionScript 3 tidak memiliki banyak perbedaan dengan pendahulunya, ActionScript 2. Perbedaan yang jelas terlihat terletak pada API untuk membuat objek. Selain itu, ActionScript 3 juga lebih mendekati bahasa dengan paradigma pemrograman berorientasi objek dibandingkan dengan ActionScript 2.

### III. ALGORITMA

Seperti yang telah disebutkan pada bagian II, proses pembuatan segitiga yang akan dibahas serupa dengan proses penghapusan segitiga. Meski demikian, proses yang akan dibahas sedikit berbeda, karena segitiga hanya akan digambar garis luar / *outline* nya saja. Oleh karena itu, langkah penghapusan segitiga menjadi tidak diperlukan.

Langkah pertama yang perlu dilakukan yaitu menentukan titik sudut dari segitiga yang paling besar. Lalu gambar garis yang menghubungkan titik-titik tersebut menjadi segitiga. Segitiga ini akan menjadi dasar untuk segitiga yang lebih kecil. Setelah itu, bagi dua setiap sisi dari segitiga tersebut, kemudian hubungkan setiap titiknya, sehingga terbentuk empat segitiga yang lebih kecil, dan serupa dengan segitiga yang menjadi dasar. Ulangi langkah tersebut sampai tingkat yang diinginkan. Semakin tinggi tingkatnya, semakin rapat dan detail hasil yang didapat.

Berikut adalah pseudocode dari algoritma yang digunakan:

```
procedure draw(x1, y1, x2, y2, x3, y3:int){
    //menggambar mulai dari titik (x1,y1)
    moveto(x1,y1)
    //Gambar garis ke (x2,y2)
    lineto(x2,y2)
    //Gambar garis ke (x3,y3)
    lineto(x3,y3)
    //Panggil prosedur rekursif
```

```

    //Titik berikutnya ada pada tengah-tengah sisi
    drawrecursive(1, (x1 + x2) / 2, (y1 + y2) / 2,
    (x1 + x3) / 2, (y1 + y3) / 2, (x2 + x3) / 2,
    (y2 + y3) / 2)
}
procedure drawrecursive(n,x1, y1, x2, y2, x3, y3:int){
    //menggambar mulai dari titik (x1,y1)
    moveto(x1,y1)
    //Gambar garis ke (x2,y2)
    lineto(x2,y2)
    //Gambar garis ke (x3,y3)
    lineto(x3,y3)

    //jika n < tingkat rekursi, lakukan rekursi.
    if (n < depth){
        // karena segitiga dibagi empat, perlu
        //Ada tiga rekursi penggambaran.
        drawrecursive (n+1,(x1 + x2) / 2 + (x2
        - x3) / 2, (y1 + y2) / 2 + (y2 - y3) /
        2,(x1 + x2) / 2 + (x1 - x3) / 2,(y1 + y2)
        / 2 + (y1 - y3) / 2,(x1 + x2) / 2, (y1
        + y2) / 2)
        drawrecursive (n+1, (x3 + x2) / 2 + (x2
        - x1) / 2, (y3 + y2) / 2 + (y2 - y1) / 2,
        (x3 + x2) / 2 + (x3 - x1) / 2 (y3 + y2) /
        2 + (y3 - y1) / 2, (x3 + x2) / 2, (y3 +
        y2) / 2

        drawrecursive (n+1, (x1 + x3) / 2 + (x3
        - x2) / 2, (y1 + y3) / 2 + (y3 - y2) / 2,
        (x1 + x3) / 2 + (x1 - x2) / 2, (y1 + y3) /
        2 + (y1 - y2) / 2, (x1 + x3) / 2, (y1 +
        y3) / 2)
    }
}

```

Pada pseudocode tersebut, basis yang digunakan untuk menghentikan rekurens adalah argumen parameter *n*. parameter tersebut menandakan tingkat rekursi yang sedang dijalankan. Ketika parameter tersebut sudah lebih besar dari batas yang ditentukan, rekurens akan berhenti.

Untuk setiap langkah rekursif yang dilakukan, program akan mencari titik tengah dari sisi-sisi segitiga yang akan dibagi, kemudian titik-titik tersebut dihubungkan dengan garis. Rekursi perlu dilakukan sebanyak tiga kali, karena pada setiap tahapan rekursi akan dihasilkan empat segitiga baru, salah satu segitiga yang menghadap ke bawah tidak akan diproses sementara tiga lainnya yang menghadap ke atas akan diproses kembali secara rekursif.

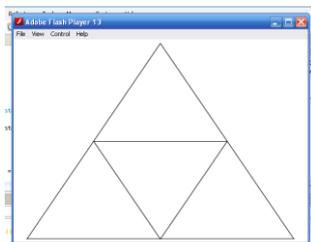
### IV. IMPLEMENTASI DENGAN ACTIONSCRIPT 3

Penulis memilih ActionScript 3 sebagai bahasa pemrograman untuk implementasi pembuatan segitiga Sierpinski karena sudah familiar dengan bahasa tersebut, dan dalam bahasa ActionScript 3 sudah tersedia library

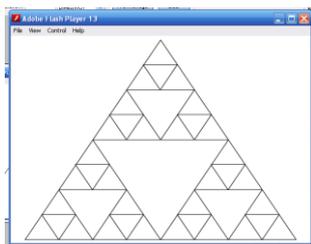
untuk menggambar bentuk-bentuk sederhana seperti garis, kurva, dan lain sebagainya. Library tersebut akan memudahkan untuk implementasi algoritma yang telah dibuat, karena pada algoritma tersebut diperlukan penggambaran garis yang sangat banyak.

Sintaks *source code* yang dibuat kurang lebih serupa dengan yang terdapat pada pseudocode di bagian III. Oleh karena itu, translasi dan implementasi ke bahasa AS3 cukup mudah. Lingkungan untuk implementasi kode sumber adalah Windows XP SP3 32-bit, Flex SDK 4.6, FlashDevelop 4.6.1, dan Flash Player 13.

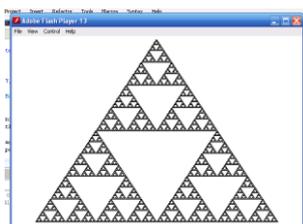
Berikut adalah hasil dari pembuatan Segitiga Sierpinski dengan menggunakan bahasa AS:



**Gambar 4-1** Segitiga Sierpinski yang terbentuk dengan tingkat rekursi 1

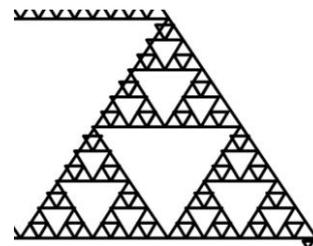


**Gambar 4-2** Segitiga Sierpinski yang terbentuk dengan tingkat rekursi 3



**Gambar 4-3** Segitiga Sierpinski yang terbentuk dengan tingkat rekursi 7

Hasil dari pembuatan Segitiga Sierpinski dengan menggunakan AS3 dapat terlihat dari gambar diatas. Gambar-gambar tersebut menunjukkan Segitiga Sierpinski dengan tingkat rekursi atau jumlah langkah rekursi yang dilakukan. Semakin tinggi tingkat rekursinya, maka segitiga tersebut akan terlihat semakin padat, dan semakin detail. Jika gambar tersebut diperbesar, maka akan muncul pola yang serupa dengan pola segitiga yang lebih besar. Ini menunjukkan bahwa Segitiga Sierpinski merupakan *fractal*.



**Gambar 4-3** Segitiga Sierpinski dengan tingkat rekursi 7 yang diperbesar

Dari hasil yang didapat, nampak ujung segitiga kecil yang tidak sesuai dengan sisi dari segitiga besar. Hal tersebut diakibatkan oleh ketebalan garis luar dari segitiga. Program yang dibuat tidak memperhitungkan ketebalan garis luar, sehingga hasil yang kurang rapi tersebut muncul.

## V. KESIMPULAN

Kita dapat menggambar fractal secara digital dengan menggunakan bahasa pemrograman ActionScript 3.

Fractal merupakan objek rekursif, sehingga dapat digambar secara rekursif. Oleh karena itu, algoritma yang digunakan untuk menggambar suatu objek fractal bekerja secara rekursif.

Untuk menghasilkan segitiga yang rapi, ketebalan dari tiap-tiap segitiga juga harus diperhitungkan.

## REFERENSI

- C Mook, Essential ActionScript 3.0. O'Reilly, 2007, pp. 629–632.
- Mandelbrot, Benoît B. The fractal geometry of nature. (1983)..  
<http://www.cut-the-knot.org/triangle/Tremsas.shtml> diakses 9 Desember 2014
- <http://www.cut-the-knot.org/ctk/Sierpinski.shtml> diakses 10 Desember 2014
- <http://lodev.org/cgtutor/sierpinski.html> diakses 9 Desember 2014
- [http://help.adobe.com/en\\_US/AS3LCR/Flash\\_10.0/](http://help.adobe.com/en_US/AS3LCR/Flash_10.0/) diakses 10 Desember 2014
- <http://natureofcode.com/book/chapter-8-fractals>

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 10 Desember 2014

Adin Baskoro Pratomo - 13513058