

PERANCANGAN PERANGKAT LUNAK GENERATOR GAMBAR DAN MUSIK FRAKTAL DENGAN METODE *ITERATED FUNCTION SYSTEM*

Tri Djoko Wahjono¹; Syaeful Karim²; Bayu Riyadi³

^{1, 2, 3}Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Bina Nusantara,
Jl. K.H. Syahdan No. 9, Kemanggis/Palmerah, Jakarta Barat 11480,
tri_d_wahjono@yahoo.com

ABSTRACT

Article presents analysis and analyze a software that utilize Geometry Fractal, especially Iterated Function System Fractal, as art. Research method that has been used in this research is by library study and by laboratorium study to test the performance of the software. Result of the research has shown that converted music by Geometry Fractal has various results, which depend on the parameters used in it and type of Geometry Fractal image produced. It can be said that usage of fractal in high iteration can produce clear image fractal and complicated music fractal.

Keywords: *image generator, music fractal, iterated function system*

ABSTRAK

Artikel menjelaskan analisis dan perancangan perangkat lunak yang dapat memanfaatkan geometri fraktal, khususnya fraktal iterated function system, sebagai seni. Metode digunakan dalam penelitian adalah studi pustaka dan studi laboratorium dengan menguji kinerja perangkat lunak tersebut. Hasil penelitian menunjukkan bahwa musik yang dikonversi dari geometri fraktal sangat bervariasi tergantung parameter dan jenis geometri gambar fraktal yang dihasilkan. Disimpulkan, penggunaan jumlah iterasi fraktal yang berjumlah besar menghasilkan gambar fraktal yang jelas dan musik fraktal yang rumit.

Kata Kunci: *generator gambar, musik fraktal, iterated function system*

PENDAHULUAN

Dalam kehidupannya, manusia tidak pernah terlepas dari pengaruh seni, baik yang berbentuk gambar maupun yang berbentuk musik. Seni dikenal sebagai cara manusia mengekspresikan dirinya. Lukisan sebagai suatu bentuk seni dihasilkan melalui sisi kreativitas manusia untuk membayangkan sesuatu dan menuangkannya ke dalam bentuk gambar. Lukisan memiliki berbagai metode dalam proses pembuatannya. Hal yang umum adalah dengan cara melukis dengan sarana cat minyak, cat air, crayon, pensil, bahkan dengan perangkat lunak.

Konsep fraktal memungkinkan pemodelan bentuk natural yang tidak dapat dihasilkan oleh matematika klasik. Konsep itu juga menawarkan bentuk gambar yang tidak memiliki batasan detail. Sementara itu, seni masih bergantung kepada kreativitas manusia untuk menghasilkan musik. Dalam pencarian manusia untuk mendapatkan jenis musik baru, kemunculan komputer memungkinkan

manusia untuk menghasilkan musik yang sangat berbeda. Musik dengan nada yang dihasilkan dari sebuah algoritma. Musik juga dapat dihasilkan dengan mengambil koordinat titik dari sebuah gambar kemudian dikonversi menjadi sebuah melodi.

Hal yang diuraikan tersebut sangat menarik bagi yang menginginkan sebuah bentuk seni gambar dan seni musik baru. Oleh karena itu, dilakukan penelitian dengan judul “Analisis dan Perancangan Perangkat Lunak Generator Gambar dan Musik Fraktal dengan metode Iterated Function System”. Dalam Penelitian ini, batasan penelitian yang akan digunakan sebagai berikut: Gambar yang akan dihasilkan oleh perangkat lunak adalah gambar fraktal 2 dimensi; Musik yang akan dihasilkan oleh perangkat lunak adalah musik *monophonic* berbasis MIDI; Metode algoritma yang akan digunakan adalah metode Iterated Function System; Perangkat lunak akan dikembangkan menggunakan Borland Delphi 6.

Tujuan penelitian adalah memanfaatkan konsep dasar fraktal untuk menghasilkan gambar yang menarik; Memanfaatkan gambar fraktal yang dihasilkan untuk membentuk sebuah komposisi musik; Memanfaatkan *iterated function system* untuk menghasilkan sebuah perangkat lunak yang menghasilkan gambar dan komposisi musik; Meneliti cara kerja algoritma *iterated function system* dalam membangun gambar dan komposisi musik fraktal. Manfaat penelitian adalah menjadi referensi dalam pengembangan perangkat lunak gambar dan musik fraktal yang lebih baru; Menjadi referensi bagi penggemar seni khususnya seni musik dan lukisan sebagai sumber inspirasi.

METODE PENELITIAN

Penelitian ini menggunakan dua metode penelitian, yaitu studi kepustakaan dan studi laboratorium. Studi kepustakaan dilakukan dengan cara mencari sumber dari internet berupa artikel dan dokumentasi serta buku yang berkaitan dengan algoritma yang dibahas dalam penelitian ini. Studi Laboratorium dilakukan dengan cara menguji kecepatan perangkat lunak dalam menghasilkan gambar dan musik fraktal pada dua komputer dengan spesifikasi yang berbeda.

TINJAUAN PUSTAKA

Fraktal memiliki dua arti yang saling berhubungan. Dalam penggunaan sehari-hari, fraktal adalah bentuk yang dibentuk secara berulang atau *self-similar* atau dengan kata lain, sebuah bentuk yang mirip pada semua tingkat pembesaran sehingga dianggap rumit tidak berhingga – *infinitely complex*. Dalam sudut pandang matematika, fraktal adalah objek yang memenuhi spesifikasi teknis tertentu. Spesifikasi itu adalah Dimensi *Hausdorff* > Dimensi Topologi.

Objek fraktal mempunyai sifat dasar yang membedakan objek fraktal dengan objek geometri pada umumnya, yaitu *Self-similarity* atau sifat keserupaan diri dan *Infinite detail* atau detil yang takberhingga.

Dimensi *Euclidean* adalah dimensi dimana sebuah ruang Euclidean \mathbb{R}^n sebagai n -dimensi. Atau, dimensi dari sebuah ruang sama dengan jumlah

parameter real yang diperlukan untuk menunjukkan titik yang berbeda pada ruang.

Sebuah ruang topologi X memiliki dimensi topologi sebesar m jika tiap *covering* \mathcal{C} dari X memiliki *refinement* \mathcal{C}' dan tiap titik dari X berada pada paling besar $m+1$ set pada \mathcal{C}' , dan m adalah paling kecil dari bilangan bulat ini.

Dimensi *Hausdorff*, didefinisikan oleh Felix Hausdorff (1868-1942) adalah dimensi dengan definisi: Untuk objek apapun dengan ukuran (P) yang terdiri dari objek dengan ukuran (p), dan jumlah objek (N) yang dapat dimasukkan ke dalam objek yang lebih besar sama dengan rasio ukuran (P/p) dipangkatkan dimensi *Hausdorff* (d) (Tucek, 2006:1).

$$N = \left(\frac{P}{p}\right)^d \text{ atau } d = \frac{\log n}{\log \left(\frac{P}{p}\right)}$$

Fraktal diklasifikasikan berdasarkan kemiripannya dengan diri sendiri atau *self-similarity*. Ada tiga jenis *self-similarity* dalam fraktal: *Exact self-similarity*, *Quasi-self-similarity*, *Statistical self-similarity*.

Teori *Chaos* adalah adalah teori yang menggambarkan pergerakan rumit dan tidak dapat ditebak atau dinamika sebuah sistem yang mudah berubah dari kondisi inisialnya. Sistem *Chaos* dapat dijelaskan secara matematika karena mengikuti hukum tertentu tetapi karena sifat berubah-ubahnya, akan tampak acak bagi mata awam. Teori *Chaos* merupakan suatu bentuk perkembangan dari teori sistem dinamis (*dynamical system*) yang memfokuskan pembahasan pada gerakan yang sangat kompleks (*highly complex motions*) yang dikenal dengan nama gerakan keotik (*chaotic motions*). Hal itu ditemukan oleh Henri Poincaré sekitar tahun 1890-an.

Iterated Function Systems atau IFS adalah sebuah metode pembentukan fraktal yang divisualisasikan ke dalam bentuknya seperti sekarang oleh John Hutchinson dan di populerkan oleh Michael Barnsley dalam buku *Fractals Everywhere*. Fraktal yang dibentuk dari IFS dapat berada pada dimensi spasial mana pun (pada dimensi apa pun). Akan tetapi, biasanya fraktal IFS dihitung dan digambar pada dua dimensi. Sebuah fraktal IFS adalah hasil dari sebuah persamaan set rekursif. Fraktal IFS terdiri

dari *union* atau gabungan dari beberapa tiruan dirinya sendiri. Masing-masing tiruan itu di transformasikan oleh sebuah fungsi (*function system*).

Kata 'musik' berasal dari bahasa Yunani, *mousike*. Menurut Kamus Besar Miriam-Webster, istilah musik (*music*) berarti suatu komposisi atau kombinasi berbagai bunyi atau suara (*sound*); Seni bunyi-bunyian, atau kumpulan bunyi atau suara.

Komposisi bunyi atau suara tersebut merupakan kombinasi deretan frekuensi bunyi yang berbeda-beda di dalam suatu interval nada yang disebut dengan satu oktaf. Istilah 'oktaf' dapat diartikan 'jarak' antara nada dasar dan nada oktafnya atau nada kedelapannya. Istilah itu digunakan, misalnya untuk menetapkan luas suara piano atau luas suara seseorang. Satu oktaf, terdiri dari delapan (oktal) tingkat nada, yaitu dari nada do sampai nada do yang lebih tinggi. Secara keseluruhan, satu oktaf terdiri dari dua belas nada, nada C, C#, D, D#, E, F, F#, G, G#, A, A#, dan B.

Chord adalah dua atau lebih nada yang digunakan pada waktu sama atau hampir bersamaan. *Chord* terdiri dari sebuah nada dasar yang memberi nama utama *chord* tersebut dan satu atau lebih nada lainnya. Nama kedua dari *chord* ditentukan dari nama nada selain nada dasar pada *chord* tersebut.

Melodi memerlukan rangkaian nada yang memiliki perbedaan panjang dan pendek nada. Pada notasi balok, panjang pendeknya nada oleh bentuk atau wujud notnya dan tiap bentuk not mempunyai nilai tertentu. Nilai not itu dihitung dengan satuan hitungan yang disebut dengan ketukan.

Musik fraktal adalah musik yang merupakan hasil proses rekursif ketika sebuah algoritma diaplikasikan berulang kali untuk memproses *output* sebelumnya. Dalam pandangan yang lebih luas, semua bentuk musik, dalam tingkat mikro maupun makro dapat di buat dengan proses ini.

MIDI adalah singkatan dari Musical Instrument Digital Interface. Dalam ilmu komputer, MIDI adalah standar *serial interface* yang memungkinkan koneksi antara *synthesizer* musik, instrumen musik, dan komputer. Standar MIDI dibuat berdasarkan bagian perangkat keras dan bagian penggambaran cara ketika musik dan suara di-*encode* dan dikomunikasikan antara perangkat MIDI.

State Transition Diagram (STD) atau yang juga dikenal sebagai *behavioral modeling* adalah prinsip operasional untuk semua kebutuhan metode analisis. STD menggambarkan perilaku dari sebuah sistem dengan menunjukkan kondisi (*state*) dan kejadian (*event*) yang menyebabkan sistem berubah kondisi.

Borland Delphi merupakan perangkat lunak yang dikembangkan oleh Borland dan Delphi versi 6.0 itu merupakan pengembangan dari versi sebelumnya. Borland Delphi atau yang lebih sering disebut Delphi, menggunakan bahasa pemrograman Pascal. Delphi merupakan salah satu perangkat lunak yang banyak digunakan oleh para *programmer* dunia saat ini.

PEMBAHASAN

Image atau gambar selama ini dihasilkan oleh manusia dengan cara menggambar pada sebuah media, baik media itu kertas, kanvas, maupun melalui perangkat lunak komputer. Begitu juga dengan komposisi musik yang dihasilkan melalui kreativitas manusia. Manusia akan memanfaatkan semua kreativitasnya untuk menghasilkan komposisi musik.

Pada perangkat lunak yang diteliti, gambar fraktal yang dibuat adalah algoritma *Iterated Function System* atau IFS sedangkan komposisi musik akan dihasilkan dengan memanfaatkan kordinat titik-titik yang ada pada gambar fraktal yang dihasilkan perangkat lunak untuk kemudian dikonversi dengan metode yang dipilih *user*.

IFS dengan *Chaos Game* menggunakan nilai probabilitas *pr* dalam memilih salah satu fungsi afin tertentu untuk melakukan pemetaan terhadap *attractor*-nya sehingga dengan titik awal (*starting point*) yang berbeda dan nilai probabilitas untuk setiap fungsi transformasi afin yang dimasukkan akan menghasilkan gambar fraktal yang sama. Penentuan besar nilai probabilitas dari masing-masing fungsi berada dalam interval $0 < pr < 1$ dan semua nilai probabilitas yang digunakan harus berjumlah 1.

Algoritma Iterasi Acak dengan IFS, dalam bentuk *pseudocode* adalah sebagai berikut.

```
//menentukan titik inisial
void chaosGame(Affine aff[], double pr[], int N){
    RealPoint P = {0.0, 0.0};
    int indeks;
    //memilih fungsi afin berikutnya
    do{
        indeks = chooseAffine(pr, N);
        P = transform(aff[indeks], P);
        //melukis titiknya
        DrawRealDot(P);
    }while(!bored);
}
```

Pseudocode musik fractal dengan metode modulus

```

void musik(int x, int n){
    //definisi variable
    int chord, nilaiNada[5][12];
    int jenChord, polaChord,
nada,counter;
    char CMaj[37];
    counter = 0;

    for(i=0;i<n;i++){
    //membatasi nilai counter
    if(counter>3)counter=0;

    //jika nilai counter = 0, ganti chord
    if(counter==0)chord = x mod 12;

    //penentuan nilaiNada
    jenChord = x mod 5;

    //jika < 2 berarti pola chord mayor dan
    chord minor
    if(jenChord<2)polaChord = x mod 9;
    //selain itu pola chord mayor ke 7,
    minor ke 7 dan mayor minor
    else polaChord = x mod 12;

    //penentuan nada
    nada = chord +
nilaiNada[jenChord][polaChord];
    if(nada>36)nada=nada-37;
    //panggil nada
    tulisNada(CMaj[nada]);
    //tambah nilai counter
    counter++;
    }
}

```

Metode normalisasi menerapkan proses normalisasi terhadap nilai X dan dari koordinatnya menjadi nilai indeks penentuan *array* nada-nya, dengan rumus berikut.

$$\text{NormalisasiX} = (x - x_{\text{Min}}) / (x_{\text{Max}} - x_{\text{Min}})$$

$$\text{NormalisasiY} = (y - y_{\text{Min}}) / (y_{\text{Max}} - y_{\text{Min}})$$

Komponen unit *New Fraktal Music* berupa *form* yang digunakan untuk membuat *file* musik fraktal yang baru dan dapat diakses dengan cara memilih *New* dari *menu File*. Unit itu akan meminta sejumlah *input* dari *user*. *User* harus memilih jenis fraktal IFS yang akan digunakan sebagai dasar pembuatan musik fraktal, metode konversi dari geometri fraktal ke musik fraktal, pemasukan nilai x_0 dan y_0 sebagai titik awal inisialisasi iterasi, jumlah iterasi, jumlah nada, dan juga dari nada ke berapa

musik yang ingin dimainkan. Selain itu, unit tersebut memungkinkan *user* untuk membuat fraktal IFS yang baru, selain yang sudah ada dalam tabel *formula* IFS. Input dari *user* akan diubah menjadi koordinat geometri fraktal dengan metode algoritma IFS. Koordinat itu digunakan untuk menggambar geometri fraktal dan dikonversi menjadi urutan nada musik fraktal.

Komponen unit *Existing Fraktal Music* digunakan untuk membuka *file* musik fraktal dan memungkinkan *user* untuk melakukan penyuntingan nilai parameter dari data musik fraktal yang sudah ada. Unit itu dapat diakses dengan cara memilih *Open* dari *menu File*. Perancangan komponen itu terdiri atas keterangan tentang jenis fraktal IFS yang digunakan oleh *file* musik fraktal yang bersangkutan beserta informasi lain yang berisi nilai parameternya serta dapat mengganti nilai parameter x_0 dan y_0 -nya dengan nilai yang baru jika ingin melakukan penyuntingan. Dengan nilai parameter yang baru, musik fraktal yang dihasilkan kemudian mungkin akan berbeda dari musik fraktal asalnya.

Kebutuhan perangkat keras untuk Perangkat lunak Generator Gambar dan Musik Fraktal cukup rendah. Proses pembuatan dijalankan pada Komputer 1 dan diuji-cobakan pada Komputer 1 dan Komputer 2

- Komputer 1:
 - Prosesor: Pentium IV 3 GHz HT
 - Memori: 1536 MB PC3200
 - Graphic card: ATi Radeon 9550 256MB
 - Monitor: VGA 15" resolusi 1024 x 768
 - Hard Disk: Seagate Barracuda 120GB
 - Soundcard: AC97
 - Lainnya: Keyboard, Mouse, Speaker
- Komputer 2:
 - Prosesor: Intel Pentium III 700 MHz
 - Memori: 128 MB PC133
 - Graphic card: ATi Rage 8MB
 - Monitor: VGA 14" resolusi 1024 x 768
 - Hard Disk: Toshiba 12GB
 - Soundcard: Onboard dengan Microsoft MIDI mapper
 - Lainnya: Keyboard, Mouse, Speaker

Kebutuhan minimum untuk menggunakan Perangkat lunak Generator Gambar dan Musik Fraktal adalah sebagai berikut.

- Prosesor: Intel Pentium II 350 MHz
- Memori: 64 MB
- Graphic card: VGA dengan resolusi minimal 800 x 600
- Hard Disk: 20MB
- Soundcard: MIDI *capable*

Lainnya: Keyboard, Mouse, Speaker
 Pengujian dilakukan dengan menggunakan perangkat lunak sebagai berikut:
 Sistem Operasi: Windows XP Profesional SP 2
 Bahasa Pemrograman: Pascal
 Editor: Borland Delphi 6
 Space Requirement: s20 MB

Pada analisis kinerja perangkat lunak, dilakukan pengujian untuk mengetahui berapa lama waktu dan besar *memory resource* yang dibutuhkan bagi perangkat lunak untuk menghasilkan gambar dan musik *fraktal* pada dua komputer yang berbeda. Pengujian akan dilakukan duabelas kali. Empat pengujian pertama akan dijalankan dengan jumlah iterasi yang kecil (10.000 iterasi). Empat pengujian berikutnya akan dijalankan dengan jumlah iterasi menengah (100.000) dan empat pengujian terakhir akan dijalankan dengan jumlah iterasi yang besar (1.000.000). Masing-masing pengujian akan menggunakan dua gambar fraktal di komputer yang berbeda. Pengujian akan dijalankan dengan parameter berikut (Lihat Tabel 1 dan 2).

Tabel 1 Transformasi *Affine Fern*

Formula IFS: Fern_1

	A	B	C	D	E	F	Probabilitas
	0	0	0	0.16	0	0	0.01
<i>Affine</i>	0.2	-0.26	0.23	0.22	0	1.6	0.07
Transformation	-0.15	0.28	0.26	0.24	0	0.44	0.07
	0.85	0.04	-0.04	0.85	0	1.6	0.85

Affine Transformation amount: 4
Conversion Method: Modulus Method
 X0: 0 Y0: 0 Image Width: 800px Height: 600px
 Scales: C Major
 Duration/Tempo: Andante

Tabel 2 Transformasi *Affine Binary*

Formula IFS: Binary

	A	B	C	D	E	F	Probabilitas
<i>Affine</i>	0.5	0	0	0.5	-2.56348	-3.00E-06	0.333333
Transformation	0.5	0	0	0.5	2.436544	-3.00E-06	0.333333
	0	-0.5	0.5	0	4.873085	7.563492	0.333334

Affine Transformation amount: 3
Conversion Method: Normalization Method
 X0: 15 Y0: 8 Image Width: 800px Height: 700px
 Scales: A Minor
 Duration/Tempo: Allegro

Pengujian waktu konversi musik dilakukan dengan menghitung waktu yang diperlukan dari saat musik mulai di konversi (dengan menekan tombol *generate*) hingga saat pertama musik dimainkan oleh perangkat lunak. Pengujian penggunaan *memory resource* dilakukan dengan memantau *Windows Task Manager* dan mencatat *peak memory usage* dari perangkat lunak. Karena penggunaan parameter

yang sama, masing-masing iterasi menghasilkan gambar dan *generate* musik yang sama di kedua komputer.

Hasil pengujian dengan IFS fern pada komputer 1 dan komputer 2 di tunjukkan pada Tabel 3 dan Tabel 4.

Tabel 3 Hasil Pengujian IFS *Fern* pada *Komputer 1*

Komputer 1:

	10.000 it	100.000 it	1.000.000it
Memory Usage (in Kb)	10,848	10,848	132,072
Generating Time (sec & msec)	149ms	743ms	32s 755ms

Tabel 4 Hasil Pengujian IFS *Fern* pada *Komputer 1*

Komputer 2:

	10.000 it	100.000 it	1.000.000it
Memory Usage (in Kb)	8,372	9,440	64,730
Generating Time (sec & msec)	327ms	1s 084 ms	44s 016ms



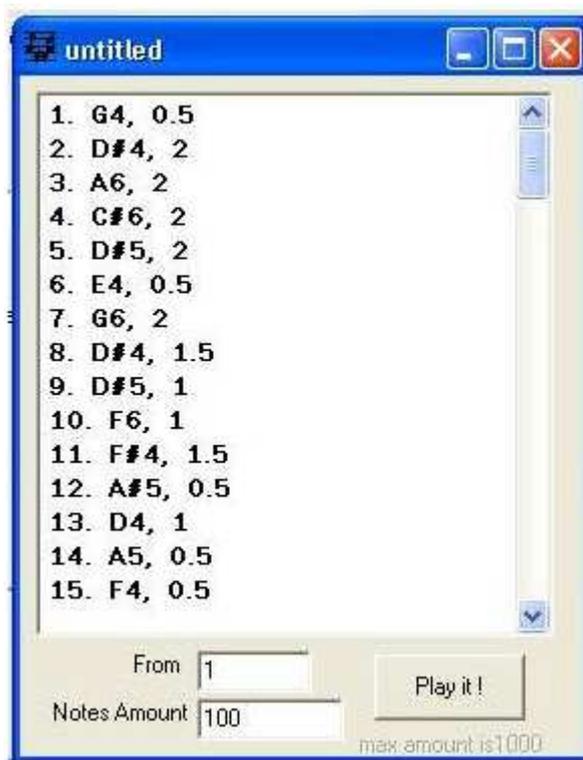
Gambar 1 IFS *Fern* dengan Iterasi 10.0000



Gambar 2 IFS *Fern* Jumlah Iterasi: 100.000



Gambar 3 IFS *Fern* Jumlah Iterasi: 1.000.000



Gambar 4 *Generate IFS fern*

Hasil pengujian dengan IFS *Binary* pada *Komputer 1* dan *Komputer 2* ditunjukkan pada Tabel 5 dan Tabel 6

Tabel 5 Hasil Pengujian IFS *Binary* pada *Komputer 1*

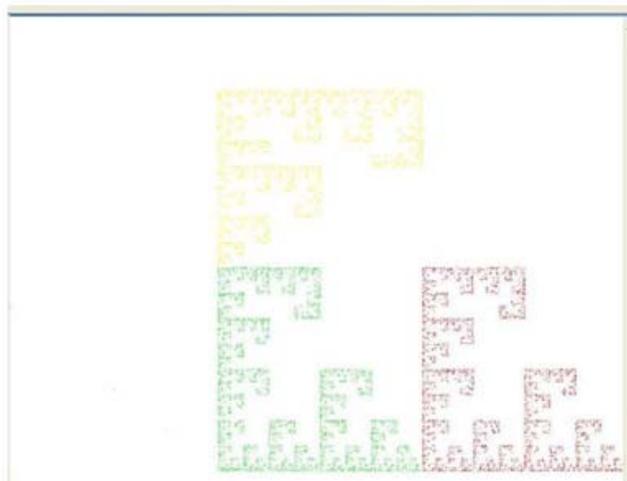
Komputer 1:

	10.000 it	100.000 it	1.000.000it
Memory Usage (in Kb)	10,192	10,776	122,781
Generating Time (sec & msec)	249ms	696ms	32s 529ms

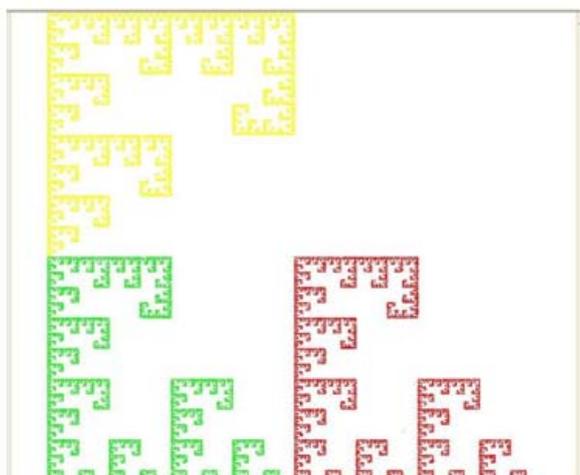
Tabel 6 Hasil Pengujian IFS *Binary* pada *Komputer 2*

Komputer 2:

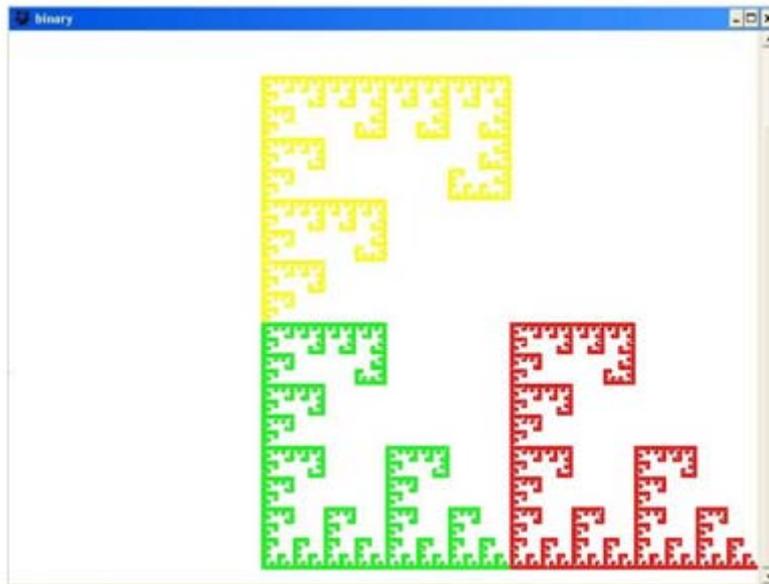
	10.000 it	100.000 it	1.000.000it
Memory Usage (in Kb)	8,424	9,024	80,743
Generating Time (sec & msec)	624ms	1s 232ms	53s 981ms



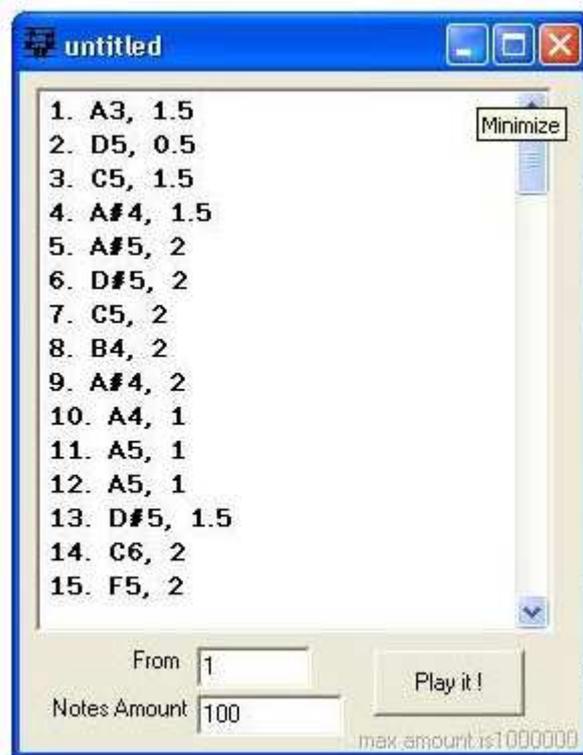
Gambar 5 IFS *Binary* Jumlah Iterasi: 10.000



Gambar 6 IFS *Binary* Jumlah Iterasi: 100.000



Gambar 7 IFS *Binary* Jumlah Iterasi: 1.000.000



Gambar 8 *Generate IFS Binary*

PENUTUP

Simpulan yang diperoleh dari perancangan dan pembuatan Perangkat lunak Generator Gambar dan Musik Fraktal dengan Metode IFS, yaitu perangkat lunak ini menerapkan konsep dasar metode fraktal IFS dalam proses pembentukan geometri fraktalnya yang kemudian dikonversikan ke bentuk musik; Walaupun melibatkan perulangan perhitungan matematis yang sederhana, metode fraktal *Iterated Function System* menampilkan bentuk geometri fraktal yang unik dan menarik dengan sifat fraktalnya yang *self-similar*; Perangkat lunak telah berfungsi dengan baik dan dapat digunakan dengan mudah, sesuai dengan konsep spesifikasi rancangan. Dengan program aplikasi ini, *user* dapat menghasilkan musik fraktal yang unik; Perangkat lunak menyediakan fasilitas penambahan formula IFS baru dan menyimpannya ke dalam basis data sehingga *user* dapat berkreasi menciptakan bentuk fraktal IFS-nya sendiri; Perangkat lunak bersifat fleksibel karena hasil pengujian menunjukkan bahwa tidak banyak perbedaan kinerja pada komputer *high end* maupun *low end*.

Saran untuk penelitian lebih lanjut, yaitu menambahkan komponen pembuatan fraktal IFS tiga dimensi (3D) sehingga musik fraktal yang dihasilkan semakin bervariasi; Menambahkan kemampuan *polyphonic* pada komponen konversi musik; Menerapkan metode fraktal lainnya, seperti *L-System* atau *Iterated Complex Polynomial*, sebagai dasar proses pembangkitan gambar fraktal; Menambahkan fasilitas penyimpanan *file* gambar fraktal ke dalam format *bitmap*; Menambahkan fasilitas penyimpanan *file* musik fraktal ke dalam *file* format MIDI sehingga dapat diputar diperangkat multimedia manapun.

DAFTAR PUSTAKA

- Abraham, Ralph H. 2004. "The Chaos FAQ," Diakses dari <http://www.mathjmiendl.org/chaos/ChaosFAQ.html>
- Anonymous. 2006. "Third Apex to Fractovia," Diakses dari http://www.fractovia.org/what/what_ing4.html
- Armstrong, MA 1983. *Basic Topology*. Springer-Verlag, NYC.
- Barnsley, Michael. 1989. *Fractals Everywhere*. 2nd Ed. San Diego, CA: Academic Press.
- Bahri, K.S dan W. Sjachriyanto. 2005. *Pemrograman Delphi*. Bandung: Informatika.
- Croom, Fred H. 1989. *Principles of Topology*. USA: Saunders College Publishing.
- Devaney, R and L. Keen, eds. 1989. *Chaos, Fractals, and Dynamics*. Menlo Park, CA: Addison-Wesley.
- Hill, F. S., Jr. 2001. *Computer Graphics Using OpenGL*. 2nd Ed. Upper Saddle River: Prentice Hall International.
- Kamien, Roger. 1996. *Music: An Appreciation*. USA: McGraw Hill.
- Oliver, Dick. 1999. *Memandang Realita dengan Fractal Vision*. Edisi Terjemahan. Yogyakarta: Andi.
- Mandelbrot, Benoit M. 1983. *The Fractal Geometry of Nature*. New York: WH Freeman and Company.
- Martina, Ir. Inge. 2000. *36 Belajar Komputer Delphi 5.0*. Jakarta: PT Elex Media Komputindo.
- Merriam-Webster team. 2007. <http://www.m-w.com>
- Microsoft Corporation. 2005. *Encarta Reference Library Premium*. Redmon, WA.: Microsoft Corporation.
- Nugroho, Widodo. 2002. *Tip dan Trik Pemrograman Delphi*. Jakarta: PT Elex Media Komputindo.
- Soeharto, M. 1990. *Belajar Notasi Balok*. Jakarta: PT Gramedia.
- Stevens, Roger T. 1990. *Advanced Fractal Programming in C*. Redwood City, CA: M&T.
- Strohbeen, David. 2004. "Fractals and Chaos - Fractal Basics," Diakses dari <http://www.fractalmusiclab.com/fractals/fractalsandchaos.ht>
- Tucek, Jim. 2006. "Fractals," Diakses dari <http://www.jracademy.com/~jtucek/math/fractals.html>
- Warriner, Alan. 2004. "TMidiGen Help File," Diakses dari <http://www.delphi32.com>