PERANGKAT LUNAK PEMBANGKIT GEOMETRI FRAKTAL BERBASIS FUNGSI TRANSENDEN

Jaidan Jauhari

Program Studi Pendidikan Matematika FKIP Universitas Sriwijaya
Jl. Raya Palembang-Prabumulih KM 32 Indralaya 30662
e-mail: jaidan_j@ilkom.unsri.ac.id

Abstract: In fractal geometry, object is drawn using iterative algorithm. Fractal has self-similarity by nature, such that each part of fractal is similar to the bigger part, but in different scale. Fractal generation process is done by iteration toward certain mathematic function. One of the mathematic functions is the trancenden function-based one. This research is limited only for trancenden function-based fractals. The methodology used in developing this software is waterfall model. The design used functional model that is DFD (Date Flow Diagram). FracTrans software is developed in Microsoft Windows programming environment. Programming language used here is Borland C^{++} Builder 4.0. This software has produced the desired output, such as fractal image generation.

Kata kunci: geometri fraktal, fungsi transenden, *self-similarity*, *waterfall model*.

Pada umumnya kurva-kurva dan permukaan memiliki struktur yang rumit, tetapi dalam lingkungan yang sederhana bentuknya dapat berupa garis atau bidang. Dalam dunia grafika komputer, grafik bentuk seperti lingkaran, elips, segiempat atau bentuk-bentuk teratur lainnya mudah digambar dengan menggunakan fasilitas fungsi tertentu yang terdapat pada perangkat lunak. Bentuk-bentuk tersebut dalam geometri Euclides dapat dinyatakan sebagai fungsi koordinat. Dengan demikian untuk menampilkan objek dapat digambarkan dengan menggunakan titik-titik koordinat pada koordinat kartesian. Tetapi bentuk objek di alam umumnya tidak beraturan dan kompleks yang tidak mudah didekati dengan rumus matematika geometri Euclidian (Pietronero & Tosatti, 1995).

Geometri fraktal memberikan gambaran dan model matematika kejadian kompleks di alam yang berbeda dengan geometri Euclidian yang dikenal selama ini. Objek dalam geometri Euclidian digambarkan dengan rumus, sedangkan pada geometri fraktal digambarkan dengan suatu algoritma iteratif. Dimensi fraktal memiliki sifat *self-similarity*, yaitu setiap bagian dari fraktal menyerupai keseluruhan bagian yang lebih besar namun dalam skala yang berbeda. Ini artinya, bagian-bagian dari objek akan terlihat identik dengan objek itu sendiri bila dilihat secara keseluruhan. Alam memiliki sifat ini, cabang-cabang pohon menyerupai pohonnya, puncak gunung mempunyai bentuk sama dengan pegunungan, awan kecil mempunyai pola yang sama dengan awan besar, demikian juga dengan struktur atom sama seperti tata surya makro kosmik. Oleh karena itu fraktal sering disebut geometri alam (Jauhari *dkk.*, 2004).

Pembangkitan fraktal dapat dilakukan dengan iterasi baik terhadap fungsi matematika atau dapat juga iterasi atas elemen-elemen dasar penyusun grafik, seperti titik, garis, dan bentuk-bentuk geometri sederhana seperti segitiga, segiempat, dan lain-lain. Fraktal yang terakhir ini dinamakan fraktal bebas, contohnya adalah fraktal plasma dan fraktal pohon. Sedangkan fraktal-fraktal yang dibangkitkan mlalui fungsi matematika antara lain fraktal yang berbasis bilangan kompleks dan fraktal yang berbasis fungsi transenden. Fraktal berbasis fungsi transenden akan menghasilkan gambar yang indah dan akan menghasilkan gambar fraktal yang unik. Bentuk fraktal dari iterasi fungsi matematika semakin menarik, indah, dan bervariasi setelah ditemukan mesin komputer yang sangat membantu komputasi (perhitungan). Selain membantu komputasinya, mesin komputer dengan perkembangan teknologi tampilannya, membantu penampilan bangun fraktal menjadi menakjubkan (Mujiono, 2002).

Berdasarkan latar belakang di atas maka masalah dalam penelitian ini adalah bagaimana membangun sebuah perangkat lunak yang dapat membangkitkan gambar-gambar fraktal berbasis fungsi transenden.

Penelitian ini ditulis dengan tujuan untuk melakukan studi terhadap geometri fraktal dan merancang perangkat lunak pembangkit fraktal berbasis fungsi transenden serta untuk membuat perangkat lunak untuk pembangkit geometri fraktal, dengan mengikuti langkah-langkah metodologi pengembangan perangkat lunak *waterfall* model. Sedangkan manfaat dari penelitian ini adalah dapat menghasilkan gambar fraktal yang bervariasi, tergantung pada parameter, variabel, dan warna masukan.

Fraktal berasal dari bahasa latin, dari kata kerja *frangere* yang berarti membelah atau kata sifat *fractus* yang artinya tidak teratur atau terfragmentasi (Mandelbrot, 1992). Beberapa pakar yang lain mengatakan bahwa fraktal adalah gambar yang secara intuitif berkarakter, yaitu setiap bagian pada sembarang ukuran jika diperbesar secukupnya akan tampak seperti gambar seutuhnya. Dari pengertian tersebut secara tersirat ada dua informasi terkandung di dalamnya: 1) gambar primitif sebagai blok pembangun, yang jika diduplikasi dengan berbagai ukuran dan dikomposisikan dapat membentuk gambar; dan 2) aturan rekursif yang mendefinisikan posisi relatif dari gambar primitif dengan berbagai ukuran.

Himpunan Fraktal menurut Falconer (1992) mempunyai 5 karakter, yaitu: 1) merupakan struktur halus, walaupun diperbesar seberapapun, 2) bersifat terlalu tidak teratur, jika digambarkan dengan bahasa geometri biasa, 3) mempunyai *self-similarity*, mungkin secara pendekatan maupun secara statistik, 4) dimensi fraktal biasanya lebih besar dari dimensi topologinya, dan 5) umumnya dapat didefinisikan secara sederhana secara rekursif.

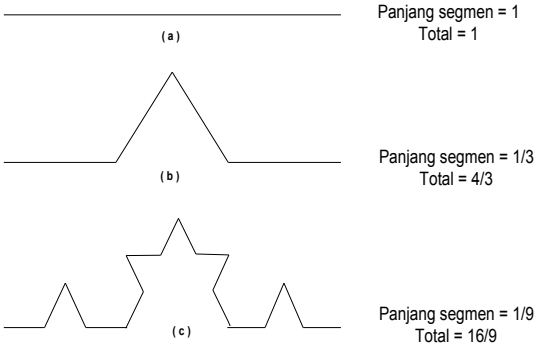
Dari uraian di atas dapat disimpulkan bahwa sifat-sifat fraktal ada 2 macam, yaitu: *self-similarity* dan *dimension*. Fraktal adalah objek yang memiliki kemiripan dengan dirinya sendiri (*self-similarity*), dalam skala yang berbeda, artinya objek fraktal terdiri dari bagian-bagian yang memiliki sifat seperti objek tersebut. Setiap bagian objek bila diperbesar akan identik dengan objek tersebut.

Fraktal adalah objek yang memiliki dimensi bilangan riil. Untuk membandingkan ukuran fraktal diperlukan dimensi fraktal. Dimensi fraktal didefinisikan sebagai kerapatan fraktal menempati ruang metrik.

Panjang sebuah segmen garis (dimensi dua) dapat diketahui dengan mengukur panjang antar dua titik. Namun objek fraktal tidak dapat diukur panjangnya, karena memiliki variasi tak hingga. Gambar 1 menunjukkan panjang dari objek fraktal tersebut bertambah 4/3 setiap tahap. Sehingga panjang objek fraktal tersebut

= 4/3 x 4/3 x 4/3 x ….

Objek fraktal tersebut memiliki panjang tak terhingga.



Gambar 1. Objek Fraktal

Dalam geometri fraktal, fraktal adalah sebuah titik di dalam ruang metrik. Ruang metrik disimbolkan dengan X, adalah himpunan titik-titik yang disertai dengan fungsi d: $X \times X \rightarrow \Re$ yang mengukur jarak antara dua buah titik di ruang tersebut.

Definisi 1. Sebuah ruang X adalah sebuah himpunan. Titik-titik pada ruang adalah anggota-anggota dari himpunan.

Definisi 2. Sebuah ruang metrik (X, d) adalah sebuah ruang X bersama dengan sebuah fungsi riil d: $X \times X \rightarrow \Re$ yang mengukur jarak antara dua titik x dan y pada X. Fungsi ini memiliki aksioma sebagai berikut: 1) Simetri $d(x, y) = d(y, x)$; $\forall x, y \in X$, 2) Positif $0 < d(x, y) < \infty$; $\forall x, y \in X, x \neq y$, 3) Jarak ke diri/kiri sendiri $d(x, x) = 0$; $\forall x \in X$, dan 4) Ketaksamaan segitiga $d(x,y) \leq d(x, z) + d(z, y)$; $\forall x, y, z \in X$.

Fungsi d disebut sebagai metrik.

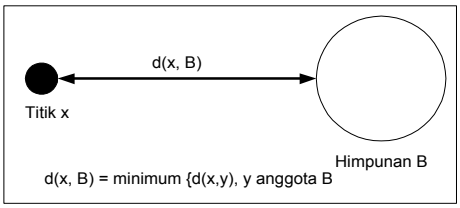
Definisi 3. Dua buah metrik d_1 dan d_2 pada ruang X adalah sama jika ada bilangan konstan $0 < c_1 < c_2 < \infty$ sedemikian hingga:

$$c_1 d_1(x, y) \leq d_2(x, y) \leq c_2 d_1(x, y); \forall (x, y) \in X \times X$$

Sebuah titik dalam geometri fraktal dapat berupa gambar hitam putih, yaitu himpunan bagian yang padat dari ruang X. Dalam geometri fraktal ruang di mana fraktal hidup adalah himpunan bagian dari X dan disimbolkan dengan F.

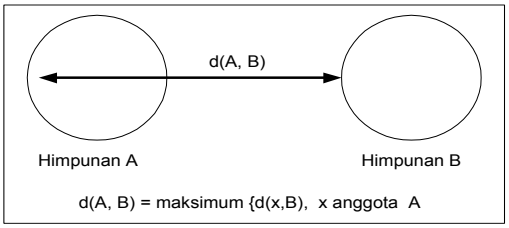
Definisi 4 Misalkan (X, d) adalah ruang metrik, maka F(X) menyatakan ruang yang titik-titiknya adalah himpunan bagian dari X. Titik-titik di ruang F disimbolkan dengan huruf kapital, misalnya A, B, dan lain-lain.

Jika x ∈ X dan B ∈ F(X), maka jarak antara titik x dengan himpunan B dalam ruang metrik adalah: d(x, B) = minimum{d(x, y); y ∈ B}

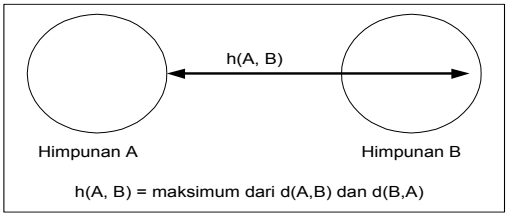


Gambar 2. Jarak Titik ke Himpunan

Sedangkan jarak antara $A \in F(X)$ dan $B \in F(X)$ dalam suatu ruang metrik yang sama didefinisikan oleh $d(A, B)$ = maksimum $\{d(x, B), x \in A\}$



Gambar 3. Jarak Himpunan ke Himpunan



Gambar 4. Jarak Haussdorff

Definisi 5 Misalkan (X, d) adalah ruang metrik lengkap, maka jarak Hausdorf antara titik A dan B di dalam F(X) adalah: h(A, B) = maksimum {d(A, B), d(B, A)}.

Fraktal Berbasis Fungsi Transenden

Fungsi-fungsi yang termasuk dalam keluarga fungsi transenden (Purccell & Verberg, 1995) antara lain adalah fungsi eksponensial e^x, fungsi logaritma asli, fungsi invers dan turunannya, fungsi sinusoidal seperti sin(x), cos(x) dan fungsi hiperbolis seperti sinh(x) dan cosh(x). Dalam penelitian ini fungsi yang dipakai dibatasi hanya untuk fungsi eksponensial dan fungsi sinusoidal seperti sin(x), cos(x), dan fungsi hiperbolis seperti sinh(x) dan cosh(x).

Iterasi atas fungsi dengan menggunakan fungsi transenden akan menghasilkan gambar fraktal yang unik (Stevens, 1990). Fraktal kosinus, sinus, eksponensial, kosinus hiperbolik, dan sinus hiperbolik dapat diperoleh dengan melakukan iterasi terhadap fungsi-fungsi sebagai berikut.

$$z_n = \cos(z_{n-1}) + c$$
$$z_n = \sin(z_{n-1}) + c$$
$$z_n = ce^{z_{n-1}}$$
$$z_n = \cosh(z_{n-1}) + c$$
$$z_n = \sinh(z_{n-1}) + c$$

(Barnsley, 1995) dengan z dan c adalah bilangan kompleks. Nilai c dalam fungsi ini sama dengan kedudukan titik kompleks $x + iy$ dan nilai awal z_0 dapat dipilih $z_0 = 0$ atau $z_0 = c$.

METODE

Metode penelitian yang digunakan dalam penelitian ini adalah metodologi pengembangan perangkat lunak Waterfall Model (Pressman, 1997). Tahapan-tahapan pengembangannya, sebagai berikut.

Analisis dan Perancangan Perangkat Lunak

Tahap analisis adalah tahapan pengumpulan kebutuhan dari semua elemen sistem perangkat lunak yang akan dibangun. Tujuan tahap analisis adalah: 1) menjabarkan kebutuhan pemakai, 2) meletakkan dasar-dasar untuk tahap perancangan perangkat lunak, dan 3) mendefinisikan semua kebutuhan pemakai sesuai dengan lingkup kotrak yang disepakati kedua belah pihak (pengembang dan pengguna).

Metode perancangan yang digunakan adalah model fungsional yaitu *Data Flow Diagram* (DFD). Tujuan dari tahap perancangan adalah: 1) merealisasikan hasil tahap analisa ke dalam bentuk rancangan sistem yang lebih rinci, 2) mendefinisikan bentuk antar muka pemakai pada bagian masukan dan keluaran, 3) mendefinisikan proses pengolahan data atau informasi secara detail, dan 4) membentuk struktur data atau basis data secara logik (*logical database*).

Implementasi Perangkat Lunak

Mengimplementasikan hasil perancangan perangkat lunak ke dalam bahasa pemrograman, yaitu ke dalam bahasa pemrograman Borland C^{++} Builder Versi 4.0.

Pengujian Perangkat Lunak

Menguji perangkat lunak terhadap beberapa kasus uji dan mengevaluasi kinerja perangkat lunak apakah sudah sesuai dengan kebutuhan yang diinginkan.

HASIL DAN PEMBAHASAN

Hasil Analisis dan Perencanaan Perangkat Lunak

Fasilitas-fasilitas yang ada di dalam perangkat lunak yang diberi nama *FracTrans* ini adalah sebagai berikut. 1) Pembangkitan Fraktal yang terdiri dari: a) membangkitkan fraktal baru, agar dapat dibuat sebuah fraktal dengan masukan berupa file teks, dan b) memodifikasi fraktal, agar fraktal yang telah dibentuk dapat dimodifikasi dengan cara memodifikasi parameter-parameternya. Fraktal yang akan dimodifikasi harus sudah ada di dalam suatu *list* fraktal. Di samping itu Fraktal yang sudah dimodifikasi dapat disimpan atau dibatalkan. 2) Menyimpan data fraktal, agar fraktal yang sudah dibangkitkan atau dimodifikasi dapat disimpan ke dalam direktori tertentu atau ke dalam *list* fraktal. 3) Membuka data fraktal, agar pengguna dapat membuka suatu fraktal dari *list* fraktal dengan cara memilih suatu *file* yang akan di buka. 4) Menampilkan *About*, agar pengguna dapat menampilkan *form* yang berisi informasi mengenai perangkat lunak ini.

Arsitektur Perangkat Lunak

Arsitektur perangkat lunak merupakan gambaran secara umum komponen-komponen yang membentuk perangkat lunak. Pengelompokan komponen secara umum dibagi dalam tiga bagian, yaitu komponen masukan sistem, komponen proses, dan komponen keluaran sistem. Arsitektur perangkat lunak *FracTrans* secara umum disajikan seperti Gambar 5.

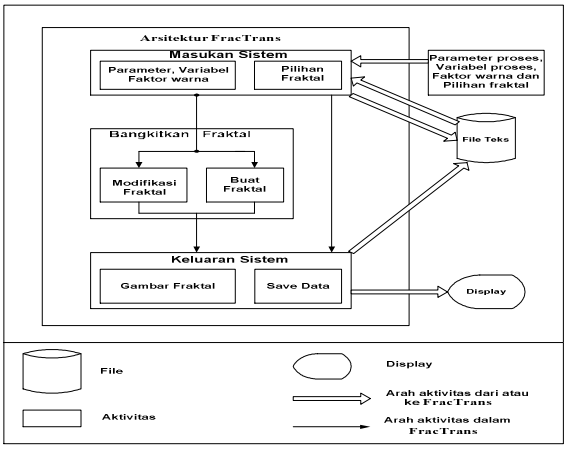
Perancangan Antar Muka dan Perancangan Struktur Menu

Di dalam sebuah perangkat lunak, antarmuka memegang peranan penting karena antarmuka berhubungan langsung dengan pengguna perangkat lunak. Untuk memudahkan pengguna digunakan sistem menu *pull down*. Untuk lebih jelasnya, maka secara lengkap menu dan sub menu diuraikan seperti pada Gambar 6.

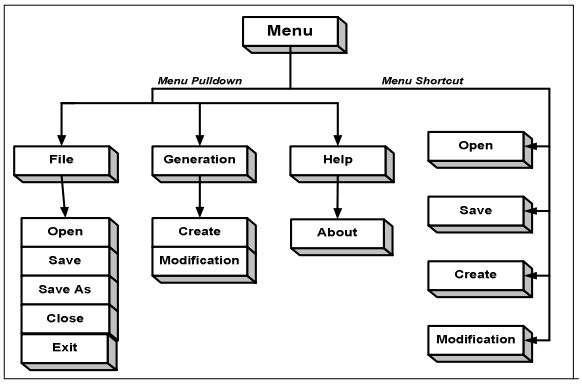
Selain menu *pulldown*, ada beberapa tombol (*buttonspeed*) yang merupakan *shortcut* ke beberapa proses yaitu tombol *Open, Save, Create*, dan *Modification*.

Perancangan Layar Saji

Pengguna berinteraksi dengan perangkat lunak *FracTrans* melalui layar saji dalam bentuk jendela (*Windows*). Layar saji terdiri dari jendela utama, jendela *create* dan jendela *about*: 1) jendela layar utama: jendela layar utama ini adalah layar pertama yang muncul jika perangkat lunak dijalankan. Pada layar ini pengguna dapat memilih menu-menu yang ada, 2) jendela *create*: bila pengguna membuka tombol *Generation* kemudian *Create* pada layar menu utama, maka jendela *Create* akan muncul. Jendela ini akan digunakan oleh pengguna sebagai sarana memasukkan paramater-parameter untuk membangkitkan fraktal, dan 4) jendela *about*: jendela ini menampilkan informasi mengenai nama perangkat lunak, pembuatnya, versi pembuatan, tahun pembuatan, dan nama lembaga.



Gambar 5. Arsitektur Perangkat Lunak *FracTrans*



Gambar 6. Struktur Menu *FracTrans*

Model Fungsional

Model fungsional menggambarkan proses-proses yang terjadi dalam perangkat lunak, masukan serta keluaran yang terjadi pada proses. Model fungsional *FracTrans* akan digambarkan dalam diagram aliran data (DFD). Diagram aliran data untuk *FracTrans* terdiri dari diagram context dan diagram aliran data level-level berikutnya yang merupakan turunan dari diagram konteks dengan proses yang lebih terperinci.

Diagram Konteks

Diagram Konteks *FracTrans* disajikan pada Gambar 7. Masukan ke sistem berupa parameter-parameter masukan, sedangkan keluarannya gambar fraktal dan pesan kesalahan.

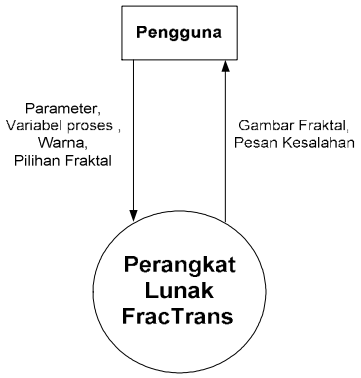
***FracTrans* DFD Level 1**

Perincian proses dari DFD level 1, yang terdiri dari proses-proses dapat dilihat pada Gambar 8.

Implementasi Perangkat Lunak

Perangkat pengembangan sangat menentukan kinerja dari perangkat lunak yang dibangun. Lingkungan ini terbagi ke dalam dua bagian yaitu lingkungan perangkat lunak dan lingkungan perangkat keras.

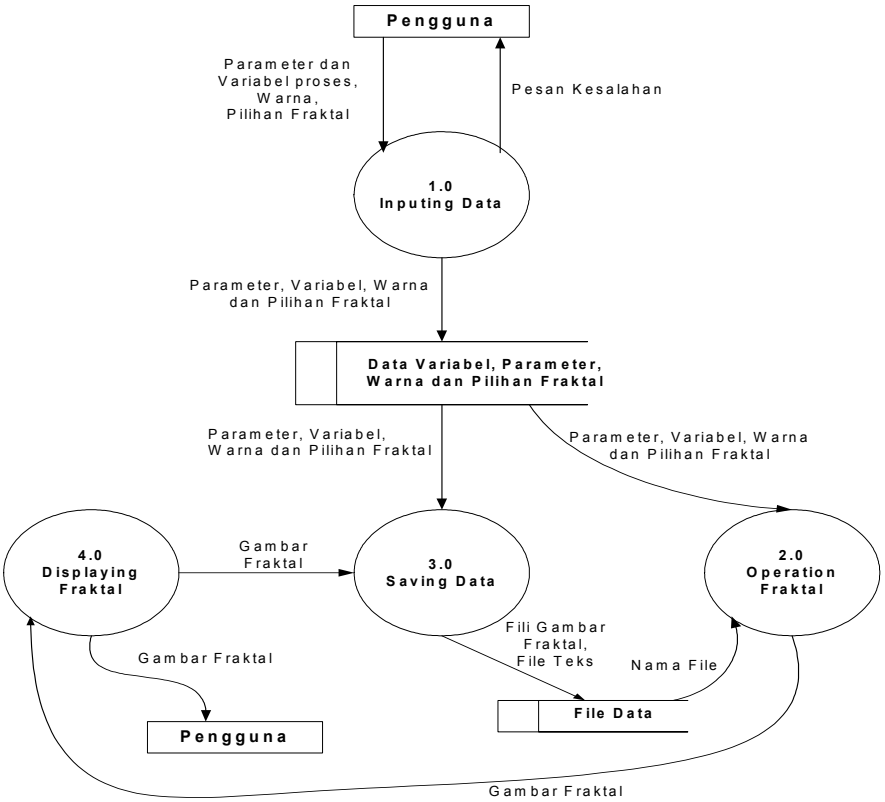
Perangkat lunak *FracTrans* dibangun dalam lingkungan pemrograman *Microsoft Windows*. Bahasa pemrograman yang dipakai adalah *Borland* C^{++} *Builder 4.0.* Adapun alasan pemilihan lingkungan ini adalah: 1) jika dibandingkan dengan lingkungan berbasis mode teks seperti DOS, lingkungan *Windows* mempunyai kemampuan untuk memberikan antar muka yang lebih baik dan mudah, 2) bahasa pemrograman C^{++} hemat ruang memori, sehingga relatif lebih cepat menjalankannya, meskipun aplikasi yang dibuat melakukan proses terhadap gambar,dan 3) penggunaan C^{++} Builder karena aplikasi ini cukup potensial untuk pemrograman visual dan pembuatan antar muka yang memadai.



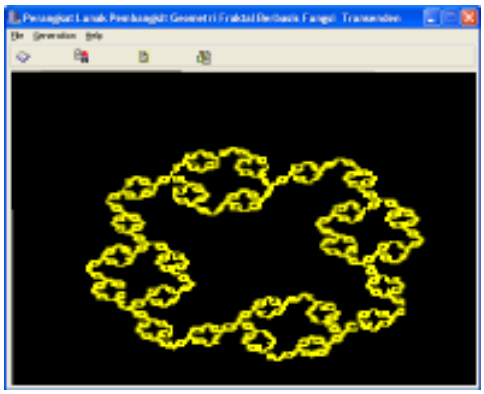
Gambar 7. Diagram Konteks (DFD level 0)

Implementasi Jendela Utama

Selengkapnya hasil implementasi jendela utama dapat dilihat pada Gambar 9.



Gambar 8. DFD Level 1 *FracTrans*



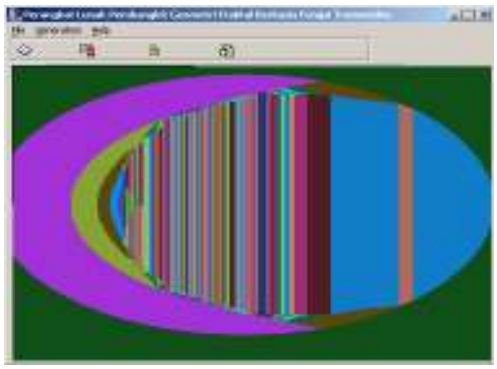
Gambar 9. Jendela Utama *FracTrans*

CONTOH-CONTOH APLIKASI

Contoh-contoh aplikasi geomatri Fraktal dengan menggunakan *FracTrans* adalah sebagai berikut.

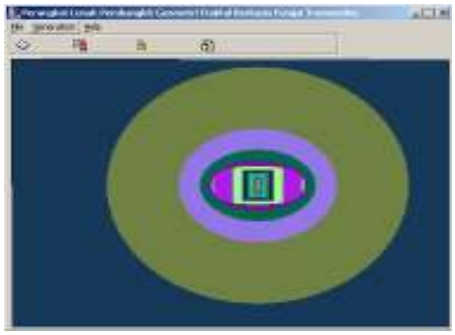
Fraktal Berbasis Fungsi Kosinus

Gambar berikut diperoleh dari masukan:
Xmax = 0,4 Ymax = 1,5
Faktor warna= 4
Xmin = -2,3 Ymin = -1,7



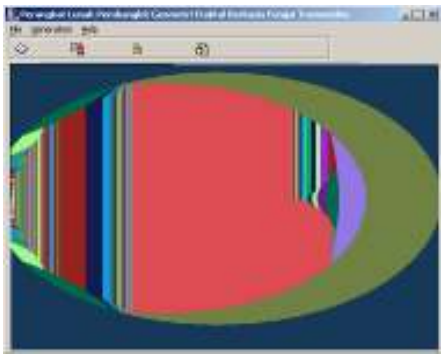
Fraktal Berbasis Fungsi Sinus

Gambar berikut diperoleh dari masukan:
Xmax = 1,8 Ymax = 1,5
Faktor warna= 7
Xmin = -2,3 Ymin = -1,7



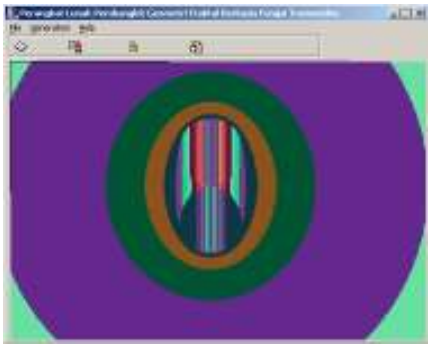
Fraktal Berbasis Fungsi Kosinus Hiperbolis

Gambar berikut diperoleh dari masukan:
Xmax = 0,4 Ymax = 1,5
Faktor warna= 7
Xmin = -2,3 Ymin = -1,7



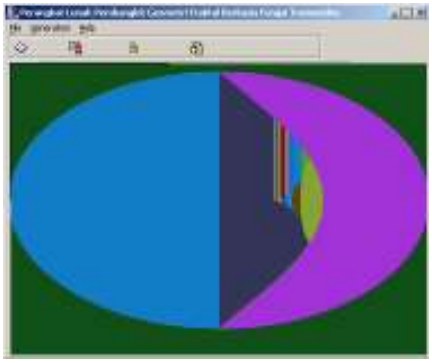
Fraktal Berbasis Fungsi Sinus Hiperbolis

Gambar berikut diperoleh dari masukan:
Xmax = 1,4 Ymax = 0,8
Faktor warna= 9
Xmin = -1,3 Ymin = -1,0



Fraktal Berbasis Fungsi Exponen

Gambar berikut diperoleh dari masukan:
Xmax = 1,4 Ymax = 1,5
Faktor warna= 3
Xmin = -1,4 Ymin = -1,7



KESIMPULAN

Dari pembahasan di atas dapat disimpulkan sebagai berikut.

- Geometri fraktal dapat dibangkitkan dengan cara melakukan iterasi terhadap suatu fungsi matematika tertentu. Salah satu fungsi matematika tersebut adalah fungsi yang berbasis fungsi transenden.
- Perangkat lunak *FracTrans* dibangun dengan menggunakan metode perancangan berorientasi aliran data (DFD) dan diimplementasi dengan perangkat pengembangan Borland C^{++} Builder versi 4.0, dan telah mencapai tujuan yang diinginkan yaitu dapat membangkitkan fraktal.
- Pembuatan antar muka pemakai sangat dibantu oleh pemakaian alat bantu pengembangan perangkat lunak Borland C^{++} Builder, karena Borland C^{++} Builder merupakan alat bantu pemrograman visual.
- Perangkat lunak *FracTrans* dapat menghasilkan keluaran seperti yang diinginkan, yaitu pembangkitan gambar fraktal dengan fungsi sinus, kosinus, sinus hiperbolis, kosinus hiperbolis, dan eksponen.

Untuk proses pengembangan lebih lanjut, maka disarankan perlu adanya fasilitas tambahan ***zoom*** untuk memperbesar tampilan objek fraktal dan penambahan fasilitas ***help*** untuk petunjuk pengoperasian perangkat lunak.

DAFTAR RUJUKAN

Barnsley, M.F. 1995. *Fractals Everywhere*. London: Academic Press Professional.

Falconer, K. 1992. *Fractal Geometry: Mathemetical Foundation & Applications*. New York: John-Wiley & Sons.

Jauhari, J., Kurniawan, R., Fani, H.S. & Rogaiyah. 2004. *Perangkat Lunak Pembangkit Fraktal*. Makalah disampaikan pada Seminar Nasional Research and Studies Proyek TPSDP di Yogyakarta, 11-13 Februari 2004.

Mandelbrot, B.B. 1992. *The Fractal Geometry of Nature*. New York: W.H. Freeman Company.

Mujiono. 10 Mei 2002. Tentang Fraktal. *Harian Umum Kompas*.

Pietronero, L. & Tosatti, E. 1995. *Fractal in Physics*. Proceeding of the Sixth Trieste Int. Symp. On Fractal in Physics, ICTP, Trieste Italy. July 9-12.

Pressman, R.S. 1997. *Software Engeneering: A Practitioner's Approach (4^{th}ed)*. New York: Mc Graw Hill.

Purcell, E.J. & Verberg, D. 1995. *Kalkulus dan Geometri Analitis (Jilid 2)*. Jakarta: Erlangga.

Stevens, R.T. 1990. *Advanced Fractal Programming in C*. New York: Redwood City. M&T Book.