

PENGEMBANGAN PERANGKAT LUNAK PEMBANGKIT GEOMETRI FRAKTAL BERBASIS BILANGAN KOMPLEKS (PLFRAKOM)

Jaidan Jauhari

Fakultas Ilmu Komputer Universitas Sriwijaya

Email : jaidan_j@yahoo.com

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 function is the complex number-based one. The methodology used in developing this software is waterfall model. The design used functional model, that is DFD (Data Flow Diagram). **PLFraKom** software is developed in Microsoft Windows programming environment. Programming language used here is Borland C++ Builder 6.0. This software has produce the desired output, such as fractal image generation.

Keywords: Fractal Geometry, Complex Numbers, *Self Similarity*, *Waterfall Model*

I. PENDAHULUAN

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 dkk,1995:70).

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, misalnya 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, 2004:2).

Pembangkitan fraktal dapat dilakukan dengan melakukan 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 melalui fungsi matematika antara lain fraktal yang berbasis bilangan kompleks, fraktal berbasis fungsi polynomial dan fraktal yang berbasis fungsi transenden. Fraktal berbasis bilangan kompleks akan menghasilkan gambar-gambar yang indah dan akan menghasilkan gambar fraktal yang unik. Bentuk-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 mengembangkan sebuah

perangkat lunak yang dapat membangkitkan gambar-gambar fraktal berbasis bilangan kompleks.

Penelitian ini ditulis dengan tujuan untuk melakukan studi terhadap geometri fraktal dan merancang pengembangan perangkat lunak pembangkit fraktal berbasis bilangan kompleks 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.

II. TINJAUAN PUSTAKA

Fraktal berasal dari bahasa latin, dari kata kerja *frangere* yang berarti membelah atau kata sifat *fractus* yang artinya tidak teratur atau terfragmentasi (Mandelbrot, 1992:5).

Beberapa pakar yang lain mengatakan dalam 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:

- a. Gambar primitif sebagai blok pembangun, yang jika diduplikasi

dengan berbagai ukuran dan dikomposisikan dapat membentuk gambar; dan

- b. Aturan rekursif yang mendefinisikan posisi relatif dari gambar primitif dengan berbagai ukuran.

Himpunan Fraktal menurut Falconer (1992:40) mempunyai 5 karakter, yaitu:

- a. Merupakan struktur halus, walaupun diperbesar seberapapun;
- b. Bersifat terlalu tidak teratur, jika digambarkan dengan bahasa geometri biasa;
- c. Mempunyai *Self-similarity*, mungkin secara pendekatan maupun secara statistic;
- d. Dimensi fraktal biasanya lebih besar dari dimensi topologinya; dan
- e. Umumnya dapat didefinisikan secara sederhana, mungkin secara rekursif.

Secara umum dari pendapat-pendapat di atas dapat disimpulkan sifat-sifat fraktal ada 2 macam, yaitu:

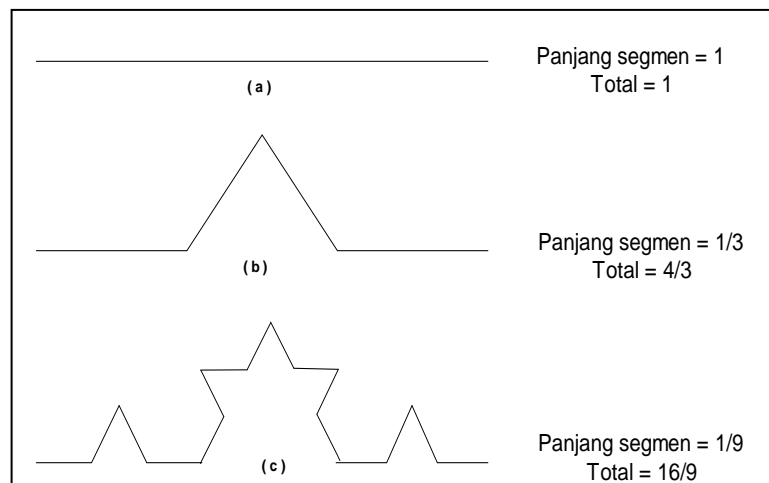
1. *Self-similarity*

Fraktal adalah objek yang memiliki kemiripan dengan dirinya sendiri (*Self-similarity*) namun dalam skala yang berbeda, ini artinya objek fraktal terdiri dari bagian-bagian yang memiliki sifat seperti objek tersebut. Setiap bagian objek tersebut bila diperbesar akan identik dengan objek tersebut.

2. *Dimension*

Fraktal adalah obyek 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. Objek Fraktal

Gambar 1 menunjukkan panjang dari objek fraktal tersebut bertambah $4/3$ setiap tahap. Sehingga panjang objek fraktal tersebut = $4/3 \times 4/3 \times 4/3 \times \dots$

Objek fraktal tersebut memiliki panjang tak berhingga.

Dalam geometri fraktal, fraktal adalah sebuah titik di dalam ruang metrik. Ruang metrik disimbolkan dengan X , adalah himpunan titik-titik yang disertai dengan fungsi $d: X \times X \rightarrow \mathbb{R}$ 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 $d: X \times X \rightarrow \mathbb{R}$ yang mengukur jarak antara dua titik x dan y pada X . Fungsi ini memiliki aksioma sebagai berikut:

- (1) Simetri $d(x, y) = d(y, x) \quad \forall x, y \in X$
- (2) Positif $0 < d(x, y) < \infty \quad \forall x, y \in X, x \neq y$
- (3) Jarak ke diri sendiri $d(x, x) = 0, \quad \forall x \in X$
- (4) Ketaksamaan segitiga $d(x, y) \leq d(x, z) + d(z, y) \quad \forall x, y, z \in X$

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:

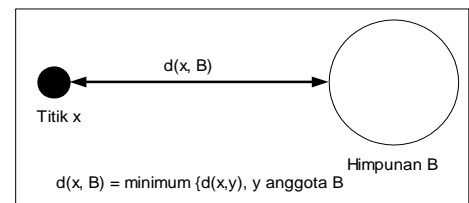
$$c_1 d_1(x, y) \leq d_2(x, y) \leq c_2 d_1(x, y) \quad \forall (x, y) \in X \times X$$

Sebuah titik dalam geometri fraktal dapat berupa gambar hitam putih, yaitu himpunan bagian yang padat dari ruang X . Dalam geometri fraktal ruang dimana 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 $x \in X$ dan $B \in F(X)$, maka jarak antara titik x dengan himpunan B dalam ruang metrik adalah:

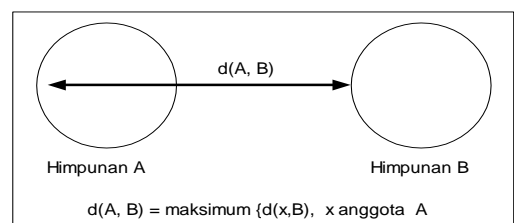
$$d(x, B) = \text{minimum}\{d(x, y), y \in B\}$$



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

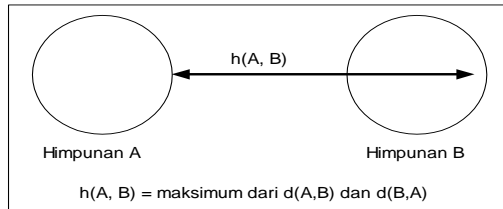
$$d(A, B) = \text{maksimum}\{d(x, B), x \in A\}$$



Gambar 3. Jarak Himpunan ke Himpunan

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

$$h(A,B) = \text{maksimum}\{d(A,B), d(B,A)\}.$$



Gambar 4. Jarak Hausdorff

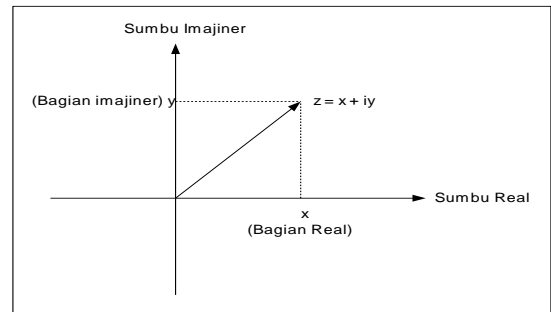
Bilangan Kompleks

Definisi 6. Bilangan kompleks adalah suatu pasangan terurut yang dinyatakan oleh (x,y) atau $x + iy$ dengan $x, y \in \mathbb{R}$, dengan

$$i^2 = -1$$

Untuk memudahkan, biasanya penulisan bilangan kompleks ditulis dengan huruf tunggal z , jadi dapat ditulis $z = x + iy$. Bilangan riil x disebut bagian riil z dan bilangan riil y disebut bagian imajiner z , masing-masing dinyatakan sebagai $\text{Re}(z)$ dan $\text{Im}(z)$.

Bila bilangan kompleks dinyatakan secara geometris dalam sistem koordinat $-xy$, maka sumbu- x disebut sumbu riil, sumbu- y disebut sumbu imajiner dan bidangnya disebut bidang kompleks. Untuk lebih jelas dapat dilihat pada gambar di bawah ini :



Gambar 5. Bidang Kompleks

Jika $y = 0$, maka bilangan kompleks $x + iy$ mengecil menjadi $x + i0$ yang cukup dituliskan sebagai x . Jadi untuk sembarang bilangan riil x , $a = a + i0$, sehingga dengan demikian bilangan riil dapat dipandang sebagai bilangan kompleks dengan bagian imajiner nol. Secara geometris, bilangan riil bersesuaian dengan titik pada sumbu riil. Sebaliknya jika $x = 0$, maka $x + iy$ mengecil menjadi $0 + iy$ yang biasanya hanya ditulis iy . Bilangan-bilangan kompleks ini, yang bersesuaian dengan titik-titik pada sumbu imajiner disebut bilangan imajiner murni.

III. FRAKTAL BERBASIS BILANGAN KOMPLEKS

Secara umum obyek fraktal dibangkitkan dari proses pengulangan sebuah fungsi transformasi khusus ke titik-titik dalam suatu daerah tertentu. Jika P_0 adalah titik awal yang dipilih, maka setiap iterasi dari sebuah fungsi transformasi F akan membangkitkan obyek baru dengan ukuran atau tempat kedudukan yang berbeda. Hal ini

berlangsung terus-menerus hingga tingkat iterasi tertentu, jadi :

$$P1 = F(P0)$$

$$P2 = F(P1)$$

$$P3 = F(P2)$$

.

.

$$Pn = F(Pn-1)$$

Dimana fungsi transformasi tersebut dapat berupa fungsi transformasi geometri klasik yaitu penskalaan, translasi dan rotasi. Pada bagian ini akan dibahas tentang fungsi pembangkit fraktal yang berbasis bilangan kompleks, yaitu Fraktal Mandelbrot, Julia, Dragon dan Barnsley.

Fraktal Mandelbrot didapatkan dengan melakukan iterasi atas fungsi berikut :

$$z_n = z_{n-1}^2 + c$$

dengan z dan c adalah bilangan kompleks.

Fraktal Julia didapatkan dengan melakukan iterasi atas persamaan yang sama dengan persamaan yang digunakan untuk membangkitkan fraktal Mandelbrot. Perbedaannya terletak pada nilai bilangan kompleks c . Jika pada fraktal Mandelbrot nilai c ditentukan oleh kedudukan titik $x + iy$, maka pada fraktal Julia nilai c ditentukan oleh suatu bilangan $P_{val} + iQ_{val}$ yang besarnya tetap untuk setiap titik kompleks $x + iy$ yang diuji. Perbedaan lainnya adalah pada penentuan titik z_0 . Pada fraktal ini nilai z_0 ditentukan sebagai $z_0 = x + iy$.

Sedangkan Fraktal Dragon diperoleh dengan melakukan iterasi atas fungsi sebagai berikut :

$$z_n = c(1-z_{n-1})^2$$

Seperti pada fraktal Julia, konstanta c pada fraktal Dragon ditentukan oleh suatu bilangan kompleks $P_{val} + iQ_{val}$, yang besarnya sama untuk setiap titik kompleks $x + iy$ yang diuji. Nilai awal z_n ditentukan sebagai $z_0 = x + iy$.

Fraktal Barnsley diperkenalkan oleh Michael Barnsley dari Georgia Institute of Technology. Fraktal Barnsley juga diperoleh dengan melakukan iterasi atas suatu fungsi bilangan kompleks. Berdasarkan fungsi penghasil fraktal, dikenal adanya fraktal Barnsley pertama, kedua, ketiga dan seterusnya. Pada bagian ini hanya akan dibahas sampai fraktal Barnsley ketiga. Fraktal Barnsley pertama didapat dari iterasi atas fungsi berikut :

$z_n = cz_{n-1} - c$, jika bagian riil dari $z_{n-1} \geq 0$

dan

$z_n = cz_{n-1} + c$, jika bagian riil dari $z_{n-1} < 0$

Fungsi yang akan diiterasi untuk menghasilkan fraktal Barnsley kedua sama dengan fungsi untuk menghasilkan fraktal Barnsley pertama, tetapi berbeda pada bagian pengujiannya, yaitu :

$z_n = cz_{n-1} - c$, jika bagian imajiner dari $cz_{n-1} \geq 0$

dan

$$z_n = cz_{n-1} + c, \text{ jika bagian}$$

imajiner dari $z_{n-1} < 0$

Sedangkan untuk Fraktal Barnsley ketiga diperoleh dengan melakukan iterasi atas fungsi :

$$z_n = z_{n-1}^2 - 1,0, \text{ jika bagian}$$

riil dari $z_{n-1} \geq 0$

dan

$$z_n = z_{n-1}^2 - 1,0 + px, \text{ jika}$$

bagian riil dari $z_{n-1} < 0$

dengan p adalah bagian riil dari c dan x adalah bagian riil dari z_{n-1}

Untuk ketiga variasi dai fraktal di atas z dan c adalah bilangan kompleks, Nilai awal z_n ditentukan dari suatu titik kompleks yang diuji, dan nilai c diawali oleh suatu konstanta bilangan kompleks $P_{val} + iQ_{val}$ yang besarnya sama untuk setiap titik yang diuji.

IV. METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah metodologi pengembangan perangkat lunak *Waterfall Model* (Pressman, 1997:50). Tahapan-tahapan pengembangannya, yaitu

- Analisa dan perancangan perangkat lunak
Tahap analisa adalah tahapan pengumpulan kebutuhan-kebutuhan dari semua elemen sistem perangkat lunak yang akan di bangun.

Tujuan tahap analisa adalah:

- Menjabarkan kebutuhan pemakai;
- Meletakkan dasar-dasar untuk tahap perancangan perangkat lunak; dan
- Mendefinisikan semua kebutuhan pemakai sesuai dengan lingkup kontrak 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:

- Merealisasikan hasil tahap analisa ke dalam bentuk rancangan sistem yang lebih rinci;
 - Mendefinisikan bentuk antar muka pemakai pada bagian masukan dan keluaran;
 - Mendefinisikan proses pengolahan data atau informasi secara detil; dan
 - 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 6.0.
 - Pengujian perangkat lunak
Menguji perangkat lunak dengan menggunakan beberapa kasus uji dan mengevaluasi kinerja perangkat lunak apakah sudah sesuai dengan kebutuhan yang diinginkan.

V. HASIL DAN PEMBAHASAN

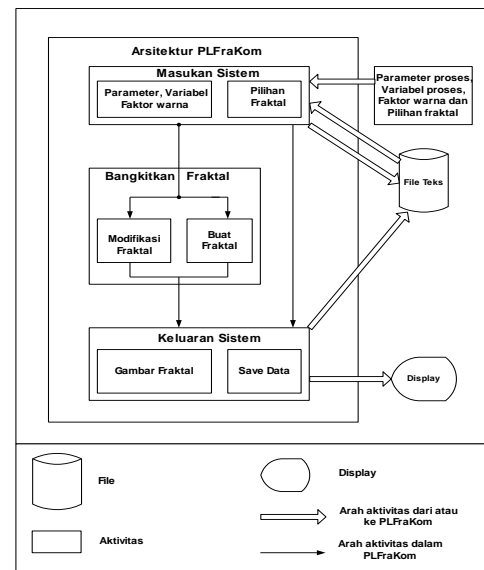
A. Hasil Analisis Dan Perancangan Perangkat Lunak

Fasilitas-fasilitas yang ada di dalam perangkat lunak yang diberi nama *PLFraKom* ini adalah sebagai berikut:

1. Pembangkitan Fraktal yang terdiri dari
 - Membangkitkan fraktal baru, dengan adanya fasilitas ini maka dapat dibuat sebuah fraktal dengan masukan berupa file teks.
 - Modifikasi Fraktal, dengan fasilitas ini sebuah 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, dengan fasilitas ini suatu fraktal yang sudah dibangkitkan atau dimodifikasi dapat disimpan ke dalam direktori tertentu atau ke dalam list fraktal.
3. Membuka data fraktal, dengan adanya fasilitas ini pengguna dapat membuka suatu fraktal dari *list* fraktal dengan cara memilih suatu *file* yang akan di buka.
4. Menampilkan *About*, dengan adanya fasilitas ini pengguna dapat menampilkan *form* yang berisi informasi mengenai perangkat lunak ini.

B. 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 *PLFraKom* secara umum disajikan seperti gambar 5.



Gambar 5. Arsitektur Perangkat Lunak **PLFraKom**

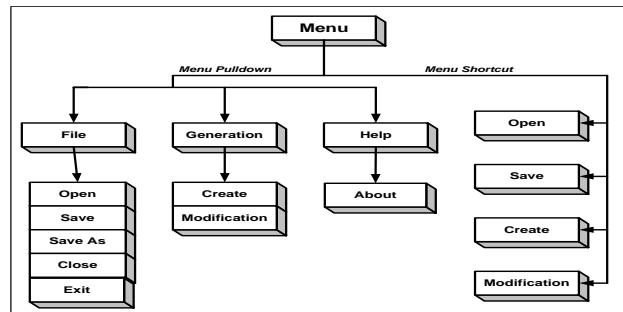
C. Perancangan Antar Muka

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 Gambar 6.



Gambar 6. Struktur Menu *PLFraKom*

Selain menu *pull down*, ada beberapa tombol (*button speed*) yang merupakan *shortcut* ke beberapa proses yaitu tombol *Open*, *Save*, *Create* dan *Modification*.

Perancangan Layar Saji

Pengguna berinteraksi dengan perangkat lunak *PLFraKom* 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 parameter-parameter untuk membangkitkan fraktal.

3. Jendela *About*

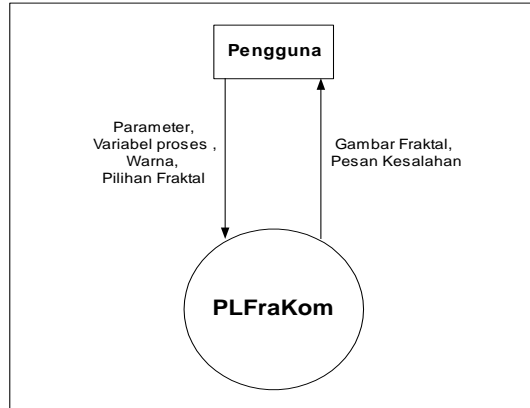
Jendela ini menampilkan informasi mengenai nama perangkat lunak, pembuatnya, versi pembuatan, tahun pembuatan, dan nama lembaga.

D. Model Fungsional

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

Diagram Konteks

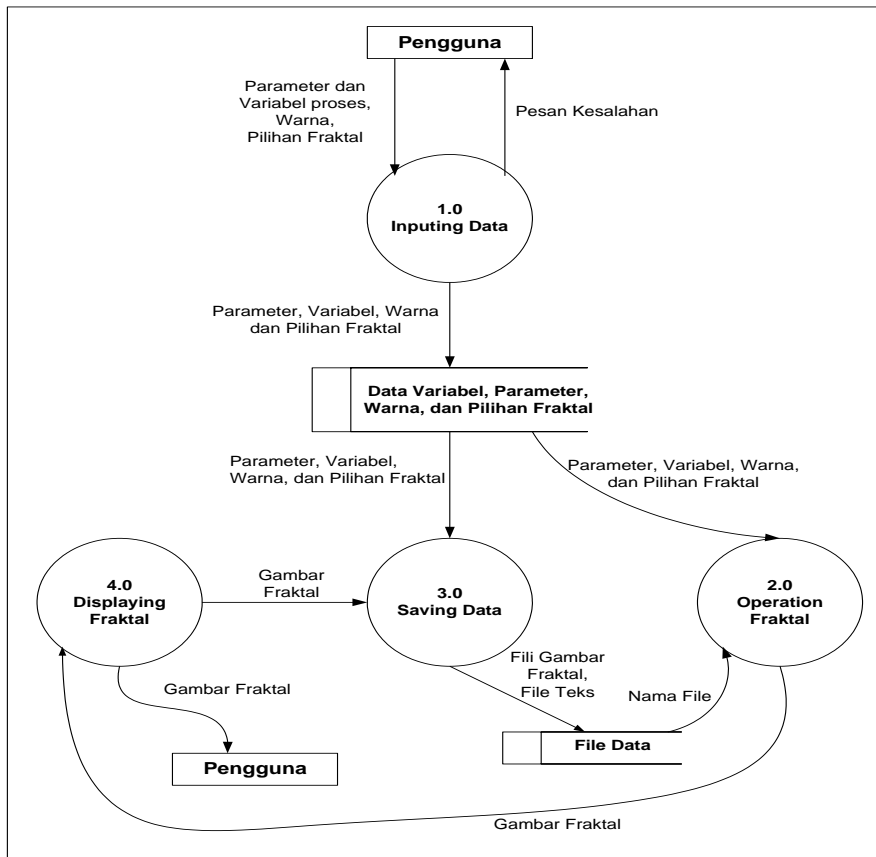
Diagram Konteks *PLFraKom* digambarkan pada Gambar 7. Masukan ke sistem berupa parameter-parameter masukan, sedangkan keluarannya gambar fraktal dan pesan kesalahan.



Gambar 7. Diagram Konteks (DFD level 0) *PLFraKom*

DFD Level 1

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



Gambar 8. DFD level 1 *PLFraKom*

E. 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 *PLFraKom* dibangun dalam lingkungan pemrograman *Microsoft Windows*. Bahasa pemrograman yang dipakai adalah *Borland C++ Builder 6.0*. Adapun alasan pemilihan lingkungan ini adalah :

- Jika dibandingkan dengan lingkungan berbasis mode teks seperti DOS, lingkungan Windows mempunyai kemampuan untuk memberikan antar muka yang lebih

baik dan mudah;

- Bahasa pemrograman C++ hemat ruang memori, sehingga relatif lebih cepat menjalankannya, meskipun aplikasi yang dibuat melakukan proses terhadap gambar; dan
- Penggunaan C++ Builder karena aplikasi ini cukup potensial untuk pemrograman visual dan pembuatan antar muka yang memadai.

Implementasi Jendela Utama

Selengkapnya hasil implementasi jendela utama dapat dilihat pada Gambar 9.



Gambar 9. Jendela Utama *PLFraKom*

VI. KESIMPULAN DAN SARAN

Dari Pembahasan sebelumnya, dapat ditarik beberapa kesimpulan dan saran sebagai berikut:

1. Geometri fraktal dapat dibangkitkan dengan cara melakukan iterasi terhadap suatu fungsi matematika tertentu. Salah satu adalah yang berbasis bilangan kompleks.
2. Perangkat lunak *PLFraKom* dibangun dengan menggunakan metode perancangan berorientasi aliran data (DFD) dan diimplementasi dengan perangkat pengembangan Borland C++ Builder versi 6.0, dan telah mencapai tujuan yang diinginkan yaitu dapat membangkitkan fraktal.
3. 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.
4. Perangkat lunak *PLFraKom* ini telah dapat menghasilkan keluaran seperti yang diinginkan, yaitu pembangkitan gambar-gambar fractal Mandelbrot, Julia, Dragon dan Barnsley.

Untuk proses pengembangan lebih lanjut, maka beberapa saran yang perlu dipertimbangkan adalah 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: Mathematical Foundation & Applications*. New York. John-Wiley & Sons.
- Jauhari, Jaidan. 2004. *Perangkat Lunak Pembangkit Fraktal*. Makalah disampaikan pada Seminar Nasional *Research and Studies* Proyek TPSDP tanggal 11-13 Februari 2004 di Yogyakarta.
- Jauhari, Jaidan. 2008. Perangkat Lunak Pembangkit Fraktal Berbasis Fungsi Polynomial. *Jurnal Ilmiah Generic Fakultas Ilmu Komputer Unsri*. Volume 2, No 2
- Jauhari, Jaidan. 2008. Perangkat Lunak Pembangkit Fraktal Berbasis Fungsi Transenden. *Jurnal MIPA dan Pembelajarannya Universitas Negeri Malang*. Volume 31, No 1
- Mandelbrot, B.B. 1992. *The Fractal Geometry of Nature*. W.H. Freeman Company.
- Mujiono. 2002. Tentang Fraktal. *Harian Umum Kompas*. Tanggal 10 Mei 2002
Online
www.kompas.com/kompas_cetak/0205/10/iptek/tent34.htm [diakses tanggal 15 Oktober 2006].

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

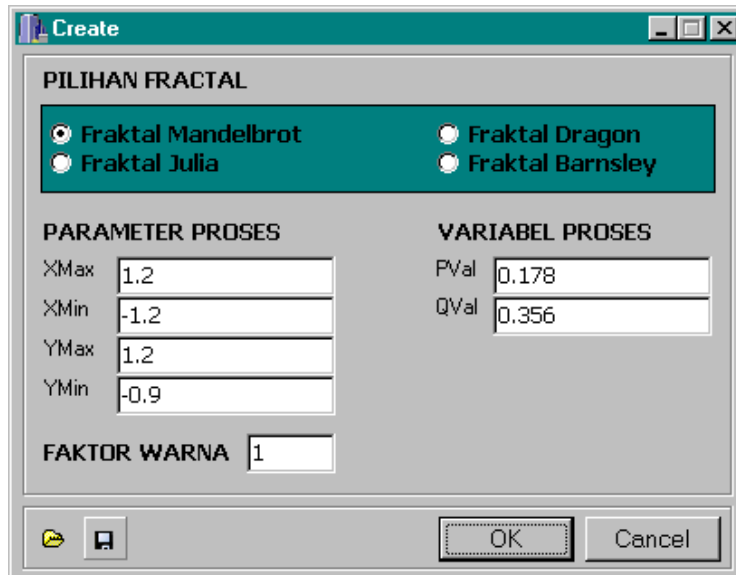
Pressman, R.S. 1997. *Software Engeneering : A Practitioner's Approach* (4thed). New York. Mc Graw Hill.

Purcell, E.J. dan Dale Verberg. 1993. *Kalkulus dan Geometri Analitis* (Jilid 2). Jakarta. Erlangga.

Stevens, R.T. 1990. *Advanced Fractal Programming in C*. Redwood City, CA : M&T Books.

LAMPIRAN

Contoh Keluaran yang dihasilkan adalah sebagai berikut:



Hasilnya :

