MEMBANGUN FRAKTAL DAUN PAKIS
MENGGUNAKAN *ITERATED FUNCTION SYSTEMS* (IFS)

SKRIPSI

Oleh :

HERY TAUFAN SETYAWAN
NIM : 021810101065

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2007

MEMBANGUN FRAKTAL DAUN PAKIS MENGGUNAKAN *ITERATED FUNCTION SYSTEMS* (IFS)

SKRIPSI

Diajukan Guna Melengkapi Tugas Akhir dan Memenuhi Syarat-syarat
untuk Menyelesaikan Program Sarjana Sains Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember

Oleh:

HERY TAUFAN SETYAWAN
NIM : 021810101065

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2007

MOTTO

"Wahai orang-orang yang beriman, taatilah Allah dan taatilah Rasul-Nya, dan ulil amri (pimpinan) di antara kalian"
(*Terjemahan Surat An-Nisa: 59*).

"Tanda taubat ialah menangis (menyesal) atas perbuatan (dosa) yang telah dilakukan dan takut akan terjatuh kembali ke dalam dosa (tersebut), tidak bergaul dengan orang-orang jahat dan senantiasa bersama dengan orang baik"
(*Siyar A'lam An-Nubala, 9/315*).

"Manusia dapat dihancurkan dan dimusnahkan, tetapi manusia tidak dapat dikalahkan selama ia masih setia pada hati nuraninya"
(*Setia Hati Terate*).

Kupersembahkan Karya Tulis Skripsi ini kepada:

Ayahanda Haryono

Dan

Ibunda Eny Setyaningsih

yang dengan ketulusan dan keikhlasannya telah memberikan kasih sayang dan doa selama ini bagi masa depan dan cita-citaku

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Hery Taufan Setyawan

Nim : 021810101065

menyatakan dengan sesungguhnya bahwa skripsi ini yang berjudul "Membangun Fraktal Daun Pakis Menggunakan *Iterated Function Systems* (IFS)" adalah benar-benar hasil karya sendiri, kecuali jika disebutkan sumbernya dan belum pernah diajukan pada institusi lain manapun, serta bukan 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 adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 11 Januari 2007

Yang menyatakan

Hery Taufan Setyawan

Nim: 021810101065

RINGKASAN

Membangun Fraktal Daun Pakis Menggunakan *Iterated Function Systems* (IFS), Hery Taufan Setyawan*, 021810101065, Skripsi, Januari 2007, 53 Halaman.

Iterated Function Systems (IFS) merupakan suatu cara membangun objek fraktal melalui sebuah rangkaian beberapa transformasi afin linier w_1, w_2, w_3, w_4 yang dipilih dengan bilangan riil random antara 0 dan 1 di R^2. Masalah yang akan dibahas adalah bagaimana membangun dan memvisualisasikan beberapa bentuk objek fraktal daun pakis dengan IFS pada komputer. Dalam membangun bentuk-bentuk objek fraktal daun pakis dapat dilakukan mula-mula menentukan besaran skalar pada empat transformasi afin linier w_1, w_2, w_3, w_4 dengan suatu titik awal (x_0, y_0) sehingga menghasilkan suatu titik koordinat baru (x', y'). Terakhir, rumusan tersebut diaplikasikan ke dalam program dengan memberikan beberapa kemungkinan pada w_1, w_2, w_3, w_4 untuk muncul dalam suatu proses iterasi. Hasilnya dapat digambarkan bentuk objek fraktal daun pakis dengan parameter yang berbeda-beda.

Kata kunci **:** *Iterated Function Systems* (IFS), Transformasi Afin, Daun Pakis.

*** Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.**

PENGESAHAN

Skripsi ini diterima oleh Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember, pada:

Hari : SELASA

Tanggal : 27 FEB 2007

Tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji

Ketua (Dosen Pembimbing Utama)	Sekretaris (Dosen Pembimbing Anggota)
Drs. Moh. Hasan, M.Sc., Ph.D. NIP. 131 759 844	Ahmad Kamsyakawuni, S.Si. NIP. 132 206 038
Anggota 1	Anggota 2
Drs. I Made Tirta, M.Sc., Ph.D. NIP. 131 945 800	Firdaus Ubaidillah, S.Si., M.Si NIP. 132 213 838

Mengesahkan

Dekan FMIPA

Universitas Jember

DEPARTEMEN PENDIDIKAN NASIONAL UNIVERSITAS JEMBER FAKULTAS MATEMATIKA & ILMU PENGETAHUAN ALAM

Ir. Sumadi, MS

NIP. 130 368 784

KATA PENGANTAR

Puji Syukur kehadirat Allah Azza Wajalla, yang telah memberikan petunjuk, rahmat dan hidayah-Nya, sehingga skripsi ini dapat terselesaikan dengan baik. Sholawat dan salam penulis haturkan kepada junjungan kita Nabi Besar Muhammmad SAW, yang mengabarkan kebenaran, kebajikan dan mendoakan ummatnya untuk selalu dilindungi oleh-Nya.

Penulis ingin menyampaikan banyak terima kasih kepada semua pihak yang telah membantu dalam penyusunan karya tulis skripsi ini antara lain:

1. Drs. Mohamad Hasan, M.Sc., Ph.D. dan Ahmad Kamsyakawuni, S.Si, yang telah memberikan petunjuk, dorongan dan semangat untuk menyelesaikan skripsi ini;
2. Drs. I Made Tirta, M.Sc., Ph.D. dan Firdaus Ubaidillah, S.Si., M.Si. yang telah memberikan kritik dan saran;
3. Ayahku Haryono dan Ibuku tercinta Eny Setyaningsih yang telah memberikan banyak pengorbanan dan doanya untuk penyelesaian skripsi ini;
4. Adikku tercinta Riski Dwi;
5. Mas Heru Ratid terima kasih atas segala bantuannya;
6. Sahabatku Anis Dwi C.S terima kasih atas segala perhatiannya;
7. Teman-teman Angkatan 2002 yang telah mendukung penulis selama ini dan teman-teman penulis lainnya yang tidak mungkin untuk disebutkan satu-persatu;
8. Teman-teman kontrakan Griya As-Sunnah terutama dr. Harie Cipta terima kasih atas laptopnya.

Penulis berharap semoga apa yang penulis tuangkan dalam skripsi ini dapat bermanfaat bagi pembaca.

Jember, Januari 2007

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Ilmu pengetahuan dan teknologi saat ini berkembang sangat pesat terutama setelah ditemukannya komputer. Komputer dapat membantu manusia dalam melakukan perhitungan matematika atau perhitungan lainnya. Salah satu bidang matematika yang berkembang sangat pesat dengan adanya komputer adalah geometri fraktal. Geometri fraktal, dalam ketidakteraturannya memiliki bentuk yang teratur yaitu suatu bentuk yang mengulang bentuk keseluruhan dalam skala yang berbeda (Mandelbrot, 1982).

Fraktal pertama kali dikenalkan oleh Mandelbrot pada tahun 1975 dari bahasa latin, yaitu kata sifat "*fractus*" yang berarti memecahkan atau menguraikan, dan dari kata kerja "*frangere*" yang berarti meretakkan atau membagi menjadi kepingan-kepingan (Beek, 1996). Menurut Mandelbrot, fraktal merupakan bentuk geometri yang berupa sebuah obyek yang dapat dibagi lagi pada bagian-bagiannya dengan mengganti suatu objek yang sebangun tetapi berukuran lebih kecil dari asalnya (Mandelbrot, 1982). Suatu keadaan objek yang dibangun secara berulang-ulang dengan mengganti suatu objek yang sebangun tetapi berukuran lebih kecil dari asalnya dinamakan *self-similarity*, dan merupakan sifat khusus dari fraktal.

Penelitian-penelitian tentang fraktal telah banyak dilakukan oleh peneliti. Pada tahun 2004, Abdul Kamil, dalam skripsinya membahas tentang bagaimana menentukan luas permukaan fraktal koch snowflake. Sedangkan pada tahun 2006, Muzammil, dalam skripsinya membahas tentang pemodelan pertumbuhan batang tanaman menggunakan metode *L-systems*, di mana komponen utamanya berupa susunan garis dan titik. Dalam skripsi ini, akan dikaji tentang bagaimana cara membangun suatu objek fraktal menggunakan metode *IFS*, yang secara garis besar hanya dibangun oleh kumpulan titik-titik yang saling mengkontraksi.

Fraktal dapat dibagi menjadi dua jenis, yaitu: himpunan-himpunan fraktal (*fractal sets*) dan fraktal alami (*natural fractal*) (Mandelbrot, 1982). Contoh fraktal jenis pertama antara lain: koch snowflake, cantor dust, sierpinski triangle, Julia sets dan Mandelbrot set. Sedangkan bentuk-bentuk awan, gunung, garis pantai, pohon, dan daun pakis merupakan contoh fraktal yang ada di alam. Adapun beberapa contoh fraktal jenis pertama yang menyerupai daun pakis, dapat dibangun menggunakan transformasi afin yang disebut *Iterated Function Systems* (IFS) (Budhi, 1995).

IFS yang dikembangkan pertama kali oleh seorang ahli matematika Michael Barnsley pada tahun 1985, merupakan suatu cara membangun objek fraktal melalui beberapa transformasi afin linier w_1, w_2, w_3, w_4 yang dipilih dengan bilangan riil random antara 0 dan 1. Diawali dengan (x_0, y_0), kemudian dilanjutkan dengan menghitung koordinat baru $(x', y') = w(x, y)$, sehingga dapat digambar sebuah titik di koordinat (x', y'). Hal ini dilakukan secara berulang-ulang dengan $x = x'$ dan $y = y'$ (Budhi, 1995). Dari uraian di atas menarik untuk dikaji bagaimana cara membangun bentuk-bentuk daun pakis dengan menggunakan IFS pada komputer.

1.2 Perumusan Masalah

Masalah yang akan dibahas adalah bagaimana membangun dan memvisualisasikan bentuk-bentuk objek fraktal daun pakis menggunakan IFS. Objek fraktalnya hanya terbatas dalam dimensi dua dan bentuknya berupa kumpulan titik-titik.

1.3 Tujuan

Tujuan penelitian ini adalah untuk mendapatkan bermacam-macam bentuk daun pakis menggunakan IFS dengan bantuan komputer dan menggunakan program Maple 8.

1.4 Manfaat

Manfaat dari penelitian ini antara lain:

a. memberikan suatu alternatif teknik konstruksi objek daun pakis selain dengan metode *L-Systems*;

b. menambah literatur berbahasa Indonesia yang berkaitan dengan IFS.

BAB 2. TINJAUAN PUSTAKA

2.1 Fraktal

Istilah fraktal pertama kali diperkenalkan oleh Benoit Mandelbrot di tahun 1975. Istilah ini muncul dalam bukunya yang berjudul *A Theory of Fractal Sets*, yang kemudian menjadi buku dan manifestonya, *The Fractal Geometry of Nature*. Mandelbrot memakai istilah fraktal untuk menyamakan dengan istilah yang dipakai sebelumnya.

Pekerjaan Benoit Mandelbrot sebenarnya telah didahului oleh Cantor, von Koch, dan Peano. Satu sifat utama yang ditemukan oleh tiga orang ini adalah adanya sifat yang disebut *self similarity*, yaitu bahwa bentuk dari suatu bagian kecil pada kurva tersebut adalah sama atau mirip dengan bagian lain yang lebih besar. Panjang kurva ini tidak mudah diukur dan ditentukan, dan dimensinya kemungkinan terletak antara garis dan bidang.

Mandelbrot mengamati karakteristik himpunan Julia (*Julia set*), yakni suatu kurva yang didasarkan dari suatu pemetaan atas fungsi:

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

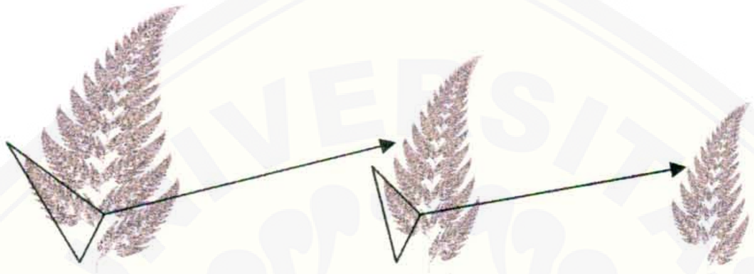
dengan z dan c adalah variabel kompleks. Mandelbrot menemukan cara baru untuk memetakan fungsi ini, yang selanjutnya disebut sebagai himpunan Mandelbrot.

Penyelidikan lebih lanjut yang dilakukan oleh Barnsley tentang himpunan Julia dan berusaha menemukan suatu metode untuk menghasilkan suatu kurva yang lebih bervariasi, sehingga dapat membangkitkan suatu pola yang cocok dengan pola yang ada pada makhluk hidup. Barnsley menemukan suatu metode yang dapat digunakan untuk membangun suatu objek fraktal, yakni metode *Iterated Function Systems* (IFS) (Santosa, 1997).

2.2 *Self Similarity*

Sifat khusus objek fraktal adalah setiap objek fraktal dapat dipecah menjadi beberapa bagian yang mempunyai kesamaan bentuk dengan objek fraktal aslinya pada tingkat perbesaran yang berbeda. Bentuk-bentuk yang mempunyai kesamaan sifat dengan obyek aslinya disebut sebagai *self similarity* (Wikipedia, 2005).

Untuk menjelaskan tentang sifat "self similarity", lihat daun pakis berikut:



(a) (b) (c)

Gambar 2.1 Daun Pakis Barnsley

Jika diambil salah satu helai daun pakis pada Gambar 2.1(a) kemudian diperbesar dan dirotasikan maka akan terbentuk daun pakis seperti Gambar 2.1(b) yang mempunyai kesamaan bentuk dengan Gambar 2.1(a) dengan ukuran yang lebih kecil. Apabila diambil salah satu helai daun pakis pada Gambar 2.1(b) kemudian diperbesar dan dirotasikan maka akan terbentuk daun pakis yang mempunyai kesamaan bentuk dengan Gambar 2.1(b) dengan ukuran yang lebih kecil dan ditunjukkan pada Gambar 2.1(c) (Mandelbrot, 1982).

2.3 *Iterated Function Systems* (IFS)

Menurut Budhi (1995), IFS merupakan transformasi afin yang digunakan untuk membangun suatu objek fraktal. Transformasi afin yaitu transformasi linier yang diikuti dengan translasi, rotasi, dan kontraksi.

Transformasi afin (*affine transformations*) adalah pemetaan $w: R^2 \to R^2$ dalam bentuk

$$w_i\left(\begin{bmatrix} x \\ y \end{bmatrix}\right)=\begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}+\begin{bmatrix} e_i \\ f_i \end{bmatrix}=A_i.p+t_i \qquad (2.1)$$

di mana a_i, b_i, c_i, d_i, e_i, dan f_i adalah besaran-besaran skalar, dan komponen matriks a, b, c, d dari salah satu atraktornya memiliki determinan nol (Anton & Rorres, 2004).

Menurut Wright (1996), IFS adalah sebuah rangkaian terbatas w_1, w_2, w_3, w_4 dari suatu transformasi afin linier di R^2. IFS dapat diterapkan pada dimensi ruang R^n dan pada situasi yang lebih umum lagi, tetapi yang akan dibahas di sini hanyalah pada sistem dimensi dua dan setiap w ditentukan oleh enam besaran skalar.

IFS merupakan suatu metode membangun fraktal melalui suatu himpunan transformasi-transformasi afin dari suatu gambar pada dirinya sendiri. Suatu transformasi $w: x \to x$ pada suatu ruang x dengan jarak s disebut pemetaan kontraksi jika terdapat suatu konstanta $0 \le a,b,c,d < 1$ (disebut faktor kontraktivitas) sedemikian hingga

$$s(w(x),w(y)) \le (a,b,c,d)\cdot s(x,y) \quad \forall x,y \in X . \qquad (2.2)$$

Secara intuitif, jika suatu transformasi w_i kontraktif, luas gambar dari transformasi lebih kecil dari luas gambar aslinya (Beek, 1996).

Berikut akan dijelaskan definisi tentang metrik Haussdorf. Didefinisikan $h(A,B)=\max\{s(A,B),s(B,A)\}$, jika $h(A,B)=h(B,A)$, maka hal ini menunjukkan $A \subset B$ dan $B \subset A$, yang berarti bahwa $A = B$. Sehingga jika $A = B$, maka diperoleh jarak baru $h(A,B)$ yang disebut metrik Haussdorf pada daerah fraktal.

Misalkan (x,s) adalah ruang metrik dan misalkan $(H(x),h)$ menotasikan ruang subset kompak tidak kosong yang terhubung, dengan metrik Hausdorff h suatu fraktal bisa didefinisikan sebagai titik tetap dari suatu transformasi kontraktif pada ruang ini. Jika $w_i : x \to x$ $(i = 1,2,...,n)$ adalah pemetaan kontraksi pada (x,s) dengan faktor kontraktivitas $(a,b,c,d)_i$, dan didefinisikan $w_i : H(x) \to H(x)$ dengan

$$w_i(A) = \{w_i(p) \mid p \in A\} \quad \text{untuk } A \in H(x) \tag{2.3}$$

maka $w: H(x) \rightarrow H(x)$, yang didefinisikan oleh

$$w(A) = \bigcup_{i=1}^{n} w_i(A), \tag{2.4}$$

merupakan suatu pemetaan kontraksi dengan faktor kontraktivitas $(a,b,c,d) = \max\{(a,b,c,d)_i \mid i = 1,2,...,n\}$, dan IFS yang terkait dinotasikan $\{x: w_i,\ i = 1,2,...,n\}$.

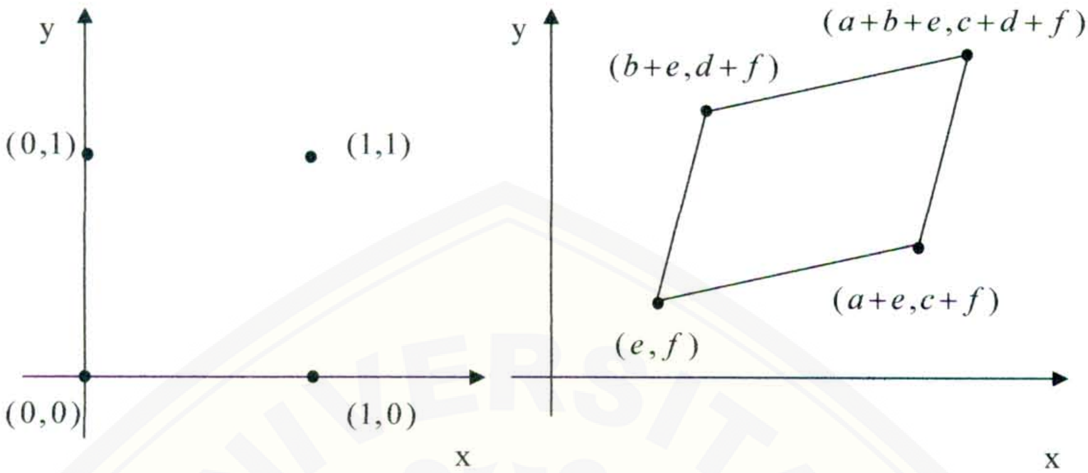
Transformasi yang umum digunakan dalam IFS pada bidang Euclidean adalah transformasi afin, yakni dalam bentuk persamaan (2.1). Secara berulang-ulang mengaplikasikan *w* seperti yang didefinisikan pada persamaan (2.4) ke suatu gambar *finite* A, sehingga bisa mendekati *fixed point A* dari *w* yang diberikan oleh

$$A = \lim_{n \to \infty} w^n\ (A) \tag{2.5}$$

yang disebut dengan atraktor *w*. Atraktor ini tidak tergantung dari gambar A di mana *w* diaplikasikan. Ini berarti atraktor terdefinisi sempurna oleh koefisien *w*, dan bahwa atraktor tersebut bisa direproduksi dengan penerapan IFS secara berulang-ulang pada sebarang titik awal (Wright, 1996).

Sebuah transformasi afin dikatakan mengkontraksi apabila jarak Euclidean di antara sebarang dua titik pada suatu bidang berkurang. Sehingga, untuk sebarang *n* transformasi afin yang mengkontraksi dapat menentukan sebuah himpunan tertutup dan terbatas yang unik s, yang memenuhi persamaan

$$s = w_1(s) \cup w_2(s) \cup w_3(s) \cup \ldots \cup w_n(s) \tag{2.6}$$



(a) Sebelum Transformasi Afin (b) Setelah transformasi Afin

Gambar 2.2 Transformasi afin

Persamaan (2.6) tidak menentukan sebuah himpunan yang saling serupa s, tetapi himpunan yang ditentukannya mempunyai sifat himpunan saling serupa (Anton & Rorres, 2004).

Dengan mengiterasikan atraktor w untuk masing-masing parameter dengan persamaan transformasi afin, akan mendapatkan batasan yang sama, yang membuktikan bahwa $w(x_f)=x_f$. Sifat dari kontraksi itu sendiri dapat digunakan lagi untuk menunjukkan bahwa hanya ada satu titik tetap yaitu w dan seluruh persamaan iterasi dari w bertemu pada titik tetap tersebut. Hal ini menunjukkan bahwa batasan persamaan dari iterasi adalah bebas secara keseluruhan dari titik penyebabnya (Wrigth, 1996).

2.3.1 Pendekatan IFS untuk Gambar Dimensi Dua

Untuk pendekatan gambar dimensi dua menggunakan suatu IFS seperti didefinisikan pada persamaan (2.4), harus membatasi parameter-parameter agar sesuai dengan kondisi kontraktivitas persamaan (2.2), maka dengan A_i seperti pada persamaan (2.1), diperoleh

$$\left|\det A_i\right| < 1 \tag{2.7}$$

untuk memetakan suatu gambar $A \in \left|x_{\min}, x_{\max}\right| \times \left|y_{\min}, y_{\max}\right|$ pada dirinya sendiri, parameter translasinya e_i dan f_i harus dibatasi oleh batas-batas gambar (Beek, 1996).

$$x_{\min} \le e_i \le x_{\max}\,,\; y_{\min} \le f_i \le y_{\max}\,. \tag{2.8}$$

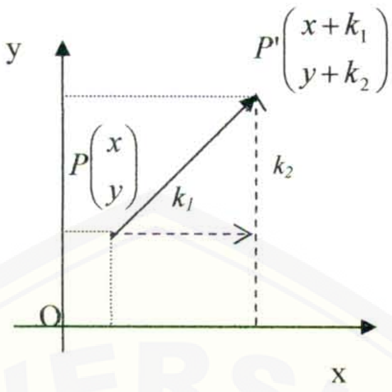
IFS merupakan sejumlah persamaan, yang masing-masing menyajikan suatu kombinasi translasi, rotasi, dan kontraksi. Dengan berawal dari sebuah titik (x_0,y_0) yang secara acak dalam mengaplikasikannya ke himpunan persamaan yang ditentukan oleh empat transformasi afin linier dan sesuai dengan aturan probabilitas tertentu, Barnsley dapat membangkitkan fraktal-fraktal dan kemudian menemukan suatu aturan untuk membangkitkan gambar daun pakis (*fern*) (Santosa, 1997).

2.3.2 Translasi dan Rotasi

Dalam membangun fraktal daun pakis diperlukan juga sistem koordinat. Dalam satu sistem koordinat, sering dilakukan suatu pemindahan obyek dari satu posisi ke posisi lain. Proses ini terkadang dilakukan hanya satu kali perpindahan atau bahkan diperlukan beberapa kali proses perpindahan. Sehingga, diperlukan penjelasan tentang translasi dan rotasi.

a. Translasi

Translasi adalah transformasi w yang memetakan titik $P\begin{pmatrix} x \\ y \end{pmatrix}$ bergeser sejauh k_1 satuan searah sumbu x dan k_2 satuan searah sumbu y sehingga didapat titik bayangan $P'\begin{pmatrix} x' \\ y' \end{pmatrix}$ (Gambar 2.3) (Kusno, 2003).



Gambar 2.3 Transformasi Translasi

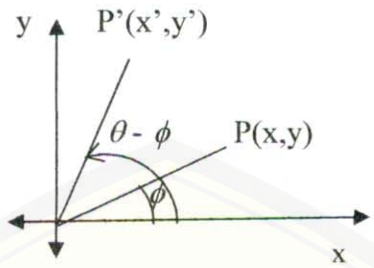
b. Rotasi

Misalkan θ adalah sebuah sudut tetap, dan misalkan $w : R^2 \rightarrow R^2$ adalah suatu transformasi yang memetakan titik $P\begin{pmatrix} x \\ y \end{pmatrix}$ ke $P'\begin{pmatrix} x' \\ y' \end{pmatrix}$ oleh perkalian matriks

$$A = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \tag{2.9}$$

dan didefinisikan sebagai P' = AP, untuk memperoleh persamaan yang diinginkan dapat dilakukan dengan dua cara, yakni menggunakan matriks $A = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$ atau matriks $B = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$, karena kedua matriks tersebut merupakan matriks koefisien yang bersesuaian dengan rotasi titik $P\begin{pmatrix} x \\ y \end{pmatrix}$ ke $P'\begin{pmatrix} x' \\ y' \end{pmatrix}$ terhadap titik pusat O(0,0) sebesar sudut θ berlawanan arah jarum jam (arah trigonometri), maka diperoleh (Kusno, 2003):

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{pmatrix} \tag{2.10}$$



Gambar 2.4 Rotasi Titik terhadap Titik Asal O menurut Arah Trigonometri

2.4 Algoritma Penafsiran Grafis pada IFS

Algoritma adalah urutan langkah instruksi logis yang disusun secara sistematis untuk menyelesaikan suatu permasalahan. Rancangan yang baik untuk suatu algoritma adalah menguraikan prosedur dalam beberapa sub prosedur, dari sub prosedur ini diuraikan lagi menjadi beberapa sub prosedur. Metode ini disebut rancangan algoritma terstruktur yang memberikan kemudahan pemahaman logika algoritma (Munir, 1999).

Berikut merupakan algoritma untuk membangun obyek fraktal daun pakis menggunakan IFS:

1. memilih empat transformasi afin linier;
2. mulai dengan suatu titik awal (x_0, y_0);
3. pilih suatu indeks i;
4. hitung titik berikutnya dengan rumus:

   $x_{n+1} = a_i x_n + b_i y_n + e_i$

   $y_{n+1} = c_i x_n + d_i y_n + f_i$

   dimana a, b, c, d, e, dan f adalah enam besaran skalar yang didefinisikan oleh persamaan (2.1);
5. plot titik baru dan kembali ke langkah tiga.

2.5 Langkah-langkah Penyelesaian

Adapun langkah-langkah yang dilakukan dalam menyelesaikan penelitian ini adalah:

1. menentukan parameter-parameter yang akan digunakan dengan memberikan nilai kemungkinan;
2. memodifikasi bentuk pakis Barnsley menjadi bentuk-bentuk yang lain dengan parameter yang berbeda;
3. memasukkan parameter-parameter yang sudah ditentukan ke dalam rumus transformasi afin dan memplotnya menggunakan program Maple 8.

BAB 4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Fraktal daun pakis dapat dibangun dengan IFS yang terdiri dari empat transformasi afin linier w_1, w_2, w_3, dan w_4. Setiap atraktor w_1, w_2, w_3, dan w_4 memiliki fungsi masing-masing dalam membangun fraktal daun pakis. Atraktor w_1 berfungsi untuk membangun tangkai utama daun pakis, atraktor w_2 berfungsi untuk membangun bentuk keseluruhan daun pakis, atraktor w_3 berfungsi untuk membangun lengan kiri daun pakis, dan atraktor w_4 berfungsi untuk membangun lengan kanan daun pakis. Nilai besaran skalar yang berbeda pada masing-masing atraktor menghasilkan bentuk daun pakis yang berbeda.

4.2. Saran

Masih terbuka peluang bagi pembaca untuk membangun objek-objek fraktal yang lain seperti Koch Snowflake, segi tiga Sierpinski, dan sebagainya dengan menggunakan metode IFS ini.

DAFTAR PUSTAKA

Anton, H. 1997. *Elementary Linear Algebra*: Application Version / Howard Anton, Chris Rorres,--7th.

Anton, H, Rorres, C. 2004. *Aplikasi Aljabar Linear*. Jakarta: Erlangga.

Beek, A. 1996. *Iterated Function Systems Optimization*. dissertations.ub.rug.nl / FILES / faculties / science / 1996 / m.m.lankhorst / c7.pdf.

Budhi, W. S. 1995. *Aljabar Linier*. Jakarta: Gramedia Pustaka Utama.

Kusno. 2003. *Geometri Rancang Bangun Studi Surfas Putar Transformasi Titik dan Proyeksi*. Jember: Fakultas MIPA Universitas Jember.

Mandelbrot, B. 1982. *The Fractal Geometry of Nature*. San Fransisco: Freeman.

Munir, R. 1999. *Algoritma dan Pemrograman Bahasa Pascal dan C*. Bandung: CV. Informatika.

Santosa, P. I. 1997. *Pemrograman Fraktal Resolusi Tinggi untuk Kartu Trident SVGA*. Yogyakarta: Andi.

Wikipedia., 2005. *Self Similarity*. http:id.wikipedia.org.wiki Fractal self-similarity.

Wright, D. J. 1996. *Iterated Function Systems Self-similarity and sets of affine maps*. http: //www.math.okstate.edu/math dept/dynamics/lecnotes/node 40.html.

LAMPIRAN A. *LISTING* PROGRAM UNTUK DAUN PAKIS 1

***Listing* Program Untuk Daun Pakis 1:**

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> Nmax:=50000:
Z:=array(0..Nmax+2):
h:=array(0..Nmax+2):
Z[0]:=[0.5,0.5]:
W:=proc(a,b,e,x,y):
a*x+b*y+e:
end:
fern1:=rand(1..100):

d:=0.25: # Panjang tangkai
b:=0.02: # Kelengkungan Tangkai
c:=-0.02: # Kecondongan badan daun pakis
mm:=1: # Refleksi
xx:=1: # Rotasi
B:=0.28: # Lebar daun

for n from 0 to Nmax do
r:=fern1():
if r<5 then
Z[n+1]:=[W(0,0,0,op(1,Z[n]),op(2,Z[n])),W(0,d,-
0.14,op(1,Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<86 then
Z[n+1]:=[W(0.85,b,0,op(1,Z[n]),op(2,mm*Z[n])),W(c,0.83,1,
op(1,mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<93 then  Z[n+1]:=[W(0.09,-
B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.60,op(1,mm*Z[
n]),op(2,Z[n]))]:h[n+1]:=r:
else Z[n+1]:=[W(-
0.09,B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.70,op(1,
mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
fi:od:
fern1:=[xx*Z[i]$i=1..Nmax]:
ulang:=[h[i]$i=1..Nmax]:
ulang2:=[[h[i],Z[i]]$i=1..Nmax]:
```

```
pointplot(fern1,style=point,symbol=point,axes=NONE,scalin
g=constrained,color=green,title="Daun Pakis 1");
```

LAMPIRAN B. *LISTING* PROGRAM UNTUK DAUN PAKIS 3

***Listing* Program Untuk Daun Pakis 3:**

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> Nmax:=50000:
Z:=array(0..Nmax+2):
Z[0]:=[0.5,0.5]:
W:=proc(a,b,e,x,y):
a*x+b*y+e:
end:
fern3:=rand(1..100):

d:=0.25: # Panjang tangkai
b:=0.002: # Kelengkungan tangkai
c:=-0.002: # Kecondongan daun pakis
mm:=1: # Refleksi
xx:=1: # Rotasi
B:=0.11: # Lebar daun


for n from 0 to Nmax do
r:=fern3():
if r<5 then
Z[n+1]:=[W(0,0,0,op(1,Z[n]),op(2,Z[n])),W(0,d,0.4,op(1,Z[
n]),op(2,Z[n]))]:
elif r<86 then Z[n+1]:=[W(0.95,b,-
0.002,op(1,Z[n]),op(2,mm*Z[n])),W(c,0.93,-
0.50,op(1,mm*Z[n]),op(2,Z[n]))]:
elif r<93 then Z[n+1]:=[W(0.04,-B,-
0.05,op(1,Z[n]),op(2,mm*Z[n])),W(0.27,0.01,-
0.01,op(1,mm*Z[n]),op(2,Z[n]))]:
else Z[n+1]:=[W(-
0.04,B,0.047,op(1,Z[n]),op(2,mm*Z[n])),W(0.27,0.01,-
0.06,op(1,mm*Z[n]),op(2,Z[n]))]:fi:od:
fern3:=[xx*Z[i]$i=1..Nmax]:
pointplot(fern3,style=point,symbol=point,axes=NONE,scalin
g=constrained,color=green,title="Daun Pakis 3");
```

 # sebelum ditambah dan diganti tanda pada komponen *f*-nya di w_1, w_2, w_3, w_4

 # sesudah ditambah dan diganti tanda pada komponen *f*-nya di w_1, w_2, w_3, w_4

Daun Pakis 3

LAMPIRAN C. *LISTING* PROGRAM UNTUK DAUN PAKIS 4

***Listing* Program Untuk Daun Pakis 4:**

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> Nmax:=50000:
Z:=array(0..Nmax+2):
Z[0]:=[0.5,0.5]:
W:=proc(a,b,e,x,y):
a*x+b*y+e:
end:
fern4:=rand(1..100):

d:=0.25: # Panjang tangkai
b:=0.005: # Kelengkungan tangkai
c:=-0.005: # Kecondongan badan daun pakis
mm:=1: # Refleksi
xx:=1: # Rotasi
B:=0.20: # Lebar daun

for n from 0 to Nmax do
r:=fern4():
if r<5 then
Z[n+1]:=[W(0,0,0,op(1,Z[n]),op(2,Z[n])),W(0,d,0.40,op(1,Z
[n]),op(2,Z[n]))]:
elif r<86 then Z[n+1]:=[W(0.95,b,-
0.002,op(1,Z[n]),op(2,mm*Z[n])),W(c,0.93,-
0.50,op(1,mm*Z[n]),op(2,Z[n]))]:
elif r<93 then Z[n+1]:=[W(0.04,-B,-
0.09,op(1,Z[n]),op(2,mm*Z[n])),W(0.16,0.04,-
0.02,op(1,mm*Z[n]),op(2,Z[n]))]:else Z[n+1]:=[W(-
0.04,B,0.09,op(1,Z[n]),op(2,mm*Z[n])),W(0.16,0.04,-
0.12,op(1,mm*Z[n]),op(2,Z[n]))]:
fi:od:
fern4:=[xx*Z[i]$i=1..Nmax]:
pointplot(fern4,style=point,symbol=point,axes=NONE,scalin
g=constrained,color=green,title="Daun Pakis 4");
```

sebelum ditambah dan diganti tanda pada komponen *f*-nya di w_1, w_2, w_3, w_4

sesudah ditambah dan diganti tanda pada komponen *f*-nya di w_1, w_2, w_3, w_4

LAMPIRAN D. *LISTING* PROGRAM UNTUK DAUN PAKIS 1 DENGAN SATU LENGAN

***Listing* Program Untuk Daun Pakis 1 dengan Satu Lengan:**

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> Nmax:=50000:
Z:=array(0..Nmax+2):
h:=array(0..Nmax+2):
Z[0]:=[0.5,0.5]:
W:=proc(a,b,e,x,y):
a*x+b*y+e:
end:
fern1:=rand(1..100):

d:=0.25: # Panjang tangkai
b:=0.02: # Kelengkungan Tangkai
c:=-0.02: # Kecondongan badan daun pakis
mm:=1: # Refleksi
xx:=1: # Rotasi
B:=0.28: # Lebar daun

for n from 0 to Nmax do
r:=fern1():
if r<5 then
Z[n+1]:=[W(0,0,0,op(1,Z[n]),op(2,Z[n])),W(0,d,-
0.14,op(1,Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<86 then
Z[n+1]:=[W(0.85,b,0,op(1,Z[n]),op(2,mm*Z[n])),W(c,0.83,1,
op(1,mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<93 then
Z[n+1]:=[W(0.09,B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09
,0.60,op(1,mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
else Z[n+1]:=[W(-
0.09,B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.70,op(1,
mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
fi:od:
fern1:=[xx*Z[i]$i=1..Nmax]:
ulang:=[h[i]$i=1..Nmax]:
```

```
ulang2:=[[h[i],Z[i]]$i=1..Nmax]:
pointplot(fern1,style=point,symbol=point,axes=NONE,scalin
g=constrained,color=green,title="Daun Pakis 1");
```

 # daun pakis 1 dengan nilai *b* pada w_3 = -0.28

 # daun pakis 1 dengan nilai *b* pada w_3 = 0,28

LAMPIRAN E. *LISTING* PROGRAM UNTUK DAUN PAKIS 1 DENGAN NILAI c PADA ATRAKTOR w_2 LEBIH KECIL

***Listing* Program Untuk Daun Pakis 1 dengan Nilai c pada Atraktor w_2 Lebih Kecil :**

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> Nmax:=50000:
Z:=array(0..Nmax+2):
h:=array(0..Nmax+2):
Z[0]:=[0.5,0.5]:
W:=proc(a,b,e,x,y):
a*x+b*y+e:
end:
fern1:=rand(1..100):

d:=0.25: # Panjang tangkai
b:=0.02: # Kelengkungan Tangkai
c:=-0.08: # Kemiringan badan daun pakis
mm:=1: # Refleksi
xx:=1: # Rotasi
B:=0.28: # Lebar daun

for n from 0 to Nmax do
r:=fern1():
if r<5 then
Z[n+1]:=[W(0,0,0,op(1,Z[n]),op(2,Z[n])),W(0,d,-
0.14,op(1,Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<86 then
Z[n+1]:=[W(0.85,b,0,op(1,Z[n]),op(2,mm*Z[n])),W(c,0.83,1,
op(1,mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<93 then  Z[n+1]:=[W(0.09,-
B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.60,op(1,mm*Z[
n]),op(2,Z[n]))]:h[n+1]:=r:
else Z[n+1]:=[W(-
0.09,B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.70,op(1,
mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
fi:od:
fern1:=[xx*Z[i]$i=1..Nmax]:
```

```
ulang:=[h[i]$i=1..Nmax]:
ulang2:=[[h[i],Z[i]]$i=1..Nmax]:
pointplot(fern1,style=point,symbol=point,axes=NONE,scalin
g=constrained,color=green,title="Daun Pakis 1");
```

 # daun pakis 1 dengan nilai *c* pada w_2 = -0.08

 # daun pakis 1 dengan nilai *c* pada w_2 = -0.02

LAMPIRAN F. *LISTING* PROGRAM UNTUK DAUN PAKIS 1 DENGAN NILAI *b* PADA ATRAKTOR w_2 YANG LEBIH BESAR

Listing **Program Untuk Daun Pakis 1 dengan Nilai *b* Pada Atarktor w_2 Yang Lebih Besar:**

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> Nmax:=50000:
Z:=array(0..Nmax+2):
h:=array(0..Nmax+2):
Z[0]:=[0.5,0.5]:
W:=proc(a,b,e,x,y):
a*x+b*y+e:
end:
fern1:=rand(1..100):

d:=0.25: # Panjang tangkai
b:=0.09: # Kelengkungan Tangkai
c:=-0.02: # Kecondongan badan daun pakis
mm:=1: # Refleksi
xx:=1: # Rotasi
B:=0.28: # Lebar daun

for n from 0 to Nmax do
r:=fern1():
if r<5 then
Z[n+1]:=[W(0,0,0,op(1,Z[n]),op(2,Z[n])),W(0,d,-
0.14,op(1,Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<86 then
Z[n+1]:=[W(0.85,b,0,op(1,Z[n]),op(2,mm*Z[n])),W(c,0.83,1,
op(1,mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<93 then  Z[n+1]:=[W(0.09,-
B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.60,op(1,mm*Z[
n]),op(2,Z[n]))]:h[n+1]:=r:
else Z[n+1]:=[W(-
0.09,B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.70,op(1,
mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
fi:od:
fern1:=[xx*Z[i]$i=1..Nmax]:
```

```
ulang:=[h[i]$i=1..Nmax]:
ulang2:=[[h[i],Z[i]]$i=1..Nmax]:
pointplot(fern1,style=point,symbol=point,axes=NONE,scalin
g=constrained,color=green,title="Daun Pakis 1");
```

 # Daun pakis 1 dengan nilai *b* pada $w_2 = 0.02$

 # Daun pakis 1 dengan nilai *b* pada $w_2 = 0.09$

LAMPIRAN G. *LISTING* PROGRAM UNTUK DAUN PAKIS 2 SETELAH DIREFLEKSIKAN TERHADAP SUMBU Y

Listing **Program Untuk Daun Pakis 2 Setelah Direfleksikan Terhadap sumbu Y:**

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> Nmax:=50000:
Z:=array(0..Nmax+2):
h:=array(0..Nmax+2):
Z[0]:=[0.5,0.5]:
W:=proc(a,b,e,x,y):
a*x+b*y+e:
end:
fern2:=rand(1..100):

d:=0.20: # Panjang tangkai
b:=0.04: # Kelengkungan tangkai
c:=-0.04: # Kecondongan daun pakis
mm:=-1: # Refleksi
xx:=1: # Rotasi
B:=0.24: # Lebar daun

for n from 0 to Nmax do
r:=fern2():
if r<5 then
Z[n+1]:=[W(0,0,0,op(1,Z[n]),op(2,Z[n])),W(0,d,-
0.12,op(1,Z[n]),op(2,Z[n]))]:h[n+1]:=r:
elif r<86 then
Z[n+1]:=[W(0.85,b,0,op(1,Z[n]),op(2,mm*Z[n])),W(c,0.82,1.
60,op(1,mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
elif r<93 then Z[n+1]:=[W(0.15,-
B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.25,0.20,0.29,op(1,mm*Z[
n]),op(2,Z[n]))]:h[n+1]:=r:
else Z[n+1]:=[W(-
0.15,B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.25,0.20,0.68,op(1,
mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
fi:od:
fern2:=[xx*Z[i]$i=1..Nmax]:
```

```
ulang:=[[h[i],Z[i]]$i=1..Nmax]:
ulang2:=[h[i]$i=1..Nmax]:
pointplot(fern2,style=point,symbol=point,axes=NONE,scalin
g=constrained,color=green,title="Daun Pakis 2");
```

 # Daun pakis 2 sebelum direfleksikan terhadap sumbu Y

 # Daun pakis 2 setelah direfleksikan terhadap sumbu Y

LAMPIRAN H. *LISTING* PROGRAM UNTUK DAUN PAKIS 2 SETELAH DI ROTASIKAN SEBESAR 180°

***Listing* Program Untuk Daun Pakis 2 Setelah Dirotasikan Sebesar 180°:**

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> Nmax:=50000:
Z:=array(0..Nmax+2):
h:=array(0..Nmax+2):
Z[0]:=[0.5,0.5]:
W:=proc(a,b,e,x,y):
a*x+b*y+e:
end:
fern2:=rand(1..100):

d:=0.20: # Panjang tangkai
b:=0.04: # Kelengkungan tangkai
c:=-0.04: # Kecondongan badan daun pakis
mm:=-1: # Refleksi
xx:=-1: # Rotasi
B:=0.24: # Lebar daun

for n from 0 to Nmax do
r:=fern2():
if r<5 then
Z[n+1]:=[W(0,0,0,op(1,Z[n]),op(2,Z[n])),W(0,d,-
0.12,op(1,Z[n]),op(2,Z[n]))]:h[n+1]:=r:
elif r<86 then
Z[n+1]:=[W(0.85,b,0,op(1,Z[n]),op(2,mm*Z[n])),W(c,0.82,1.
60,op(1,mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
elif r<93 then Z[n+1]:=[W(0.15,-
B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.25,0.20,0.29,op(1,mm*Z[
n]),op(2,Z[n]))]:h[n+1]:=r:
else Z[n+1]:=[W(-
0.15,B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.25,0.20,0.68,op(1,
mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
fi:od:
fern2:=[xx*Z[i]$i=1..Nmax]:
ulang:=[[h[i],Z[i]]$i=1..Nmax]:
```

```
ulang2:=[h[i]$i=1..Nmax]: .
pointplot(fern2,style=point,symbol=point,axes=NONE,scalin
g=constrained,color=green);
```

 # Daun pakis 2 sebelum dirotasikan

 # Daun pakis 2 setelah dirotasikan

LAMPIRAN I. *LISTING* PROGRAM DAUN PAKIS 1 UNTUK NILAI KOMPONEN a PADA ATRAKTOR w_2 YANG MENDEKATI SATU

***Listing* Program Daun Pakis 1 Untuk Nilai Komponen a Pada Atraktor w_2 Yang Mendekati Satu:**

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> Nmax:=50000:
Z:=array(0..Nmax+2):
h:=array(0..Nmax+2):
Z[0]:=[0.5,0.5]:
W:=proc(a,b,e,x,y):
a*x+b*y+e:
end:
fern1:=rand(1..100):

d:=0.25: # Panjang tangkai
b:=0.02: # Kemiringan Tangkai
c:=-0.02: # Kecondongan daun pakis
mm:=1: # Refleksi
xx:=1: # Rotasi
B:=0.28: # Lebar daun

for n from 0 to Nmax do
r:=fern1():
if r<5 then
Z[n+1]:=[W(0,0,0,op(1,Z[n]),op(2,Z[n])),W(0,d,-
0.14,op(1,Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<86 then
Z[n+1]:=[W(0.99,b,0,op(1,Z[n]),op(2,mm*Z[n])),W(c,0.83,1,
op(1,mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<93 then  Z[n+1]:=[W(0.09,-
B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.60,op(1,mm*Z[
n]),op(2,Z[n]))]:h[n+1]:=r:
else Z[n+1]:=[W(-
0.09,B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.70,op(1,
mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
```

```
fi:od:
fern1:=[xx*Z[i]$i=1..Nmax]:
ulang:=[h[i]$i=1..Nmax]:
ulang2:=[[h[i],Z[i]]$i=1..Nmax]:
pointplot(fern1,style=point,symbol=point,axes=NONE,scalin
g=constrained,color=green);
```

 # daun pakis 1 dengan nilai *a* pada w_2 = 0.85

 # daun pakis 1 dengan nilai *a* pada w_2 = 0.99

LAMPIRAN J. *LISTING* PROGRAM DAUN PAKIS 1 UNTUK NILAI KOMPONEN *a* PADA ATRAKTOR w_2 YANG MENDEKATI NOL

***Listing* Program Daun Pakis 1 Untuk Nilai Komponen *a* Pada Atraktor w_2 Yang Mendekati Nol :**

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> Nmax:=50000:
Z:=array(0..Nmax+2):
h:=array(0..Nmax+2):
Z[0]:=[0.5,0.5]:
W:=proc(a,b,e,x,y):
a*x+b*y+e:
end:
fern1:=rand(1..100):

d:=0.25: # Panjang tangkai
b:=0.02: # Kemiringan Tangkai
c:=-0.02: # Kecondongan daun pakis
mm:=1: # Refleksi
xx:=1: # Rotasi
B:=0.28: # Lebar daun

for n from 0 to Nmax do
r:=fern1():
if r<5 then
Z[n+1]:=[W(0,0,0,op(1,Z[n]),op(2,Z[n])),W(0,d,-
0.14,op(1,Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<86 then
Z[n+1]:=[W(0.50,b,0,op(1,Z[n]),op(2,mm*Z[n])),W(c,0.83,1,
op(1,mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<93 then  Z[n+1]:=[W(0.09,-
B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.60,op(1,mm*Z[
n]),op(2,Z[n]))]:h[n+1]:=r:
else Z[n+1]:=[W(-
0.09,B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.70,op(1,
mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
```

```
fi:od:
fern1:=[xx*Z[i]$i=1..Nmax]:
ulang:=[h[i]$i=1..Nmax]:
ulang2:=[[h[i],Z[i]]$i=1..Nmax]:
pointplot(fern1,style=point,symbol=point,axes=NONE,scalin
g=constrained,color=green);
```

 # daun pakis 1 dengan nilai *a* pada w_2 = 0.50

 # daun pakis 1 dengan nilai *a* pada w_2 = 0.85

LAMPIRAN K. *LISTING* PROGRAM MODIFIKASI DAUN PAKIS 1 DENGAN NILAI KOMPONEN *d* PADA ATRAKTOR w_2 YANG MENDEKATI SATU

***Listing* Program Modifikasi Daun Pakis 1 Untuk Nilai Komponen *d* Pada Atraktor w_2 Yang Mendekati Satu :**

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> Nmax:=50000:
Z:=array(0..Nmax+2):
h:=array(0..Nmax+2):
Z[0]:=[0.5,0.5]:
W:=proc(a,b,e,x,y):
a*x+b*y+e:
end:
fern1:=rand(1..100):

d:=0.25: # Panjang tangkai
b:=0.02: # Kemiringan Tangkai
c:=-0.02: # Kecondongan daun pakis
mm:=1: # Refleksi
xx:=1: # Rotasi
B:=0.28: # Lebar daun

for n from 0 to Nmax do
r:=fern1():
if r<5 then
Z[n+1]:=[W(0,0,0,op(1,Z[n]),op(2,Z[n])),W(0,d,-
0.14,op(1,Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<86 then
Z[n+1]:=[W(0.85,b,0,op(1,Z[n]),op(2,mm*Z[n])),W(c,0.99,1,
op(1,mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<93 then  Z[n+1]:=[W(0.09,-
B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.60,op(1,mm*Z[
n]),op(2,Z[n]))]:h[n+1]:=r:
else Z[n+1]:=[W(-
0.09,B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.70,op(1,
mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
```

```
fi:od:
fern1:=[xx*Z[i]$i=1..Nmax]:
ulang:=[h[i]$i=1..Nmax]:
ulang2:=[[h[i],Z[i]]$i=1..Nmax]:
pointplot(fern1,style=point,symbol=point,axes=NONE,scalin
g=constrained,color=green);
```

 # Daun pakis 1 dengan nilai *d* pada $w_2 = 0.83$

Daun Pakis 1

 # Daun pakis 1 dengan nilai *d* pada $w_2 = 0.99$

Daun Pakis 1

LAMPIRAN L. *LISTING* PROGRAM MODIFIKASI DAUN PAKIS 1 DENGAN NILAI KOMPONEN *d* PADA ATRAKTOR w_2 YANG MENDEKATI NOL

Listing **Program Modifikasi Daun Pakis 1 Untuk Nilai Komponen *d* Pada Atraktor w_2 Yang Mendekati Nol :**

```
> restart;
> with(plots):
Warning, the name changecoords has been redefined

> Nmax:=50000:
Z:=array(0..Nmax+2):
h:=array(0..Nmax+2):
Z[0]:=[0.5,0.5]:
W:=proc(a,b,e,x,y):
a*x+b*y+e:
end:
fern1:=rand(1..100):

d:=0.25: # Panjang tangkai
b:=0.02: # Kelengkungan Tangkai
c:=-0.02: # Kecondongan daun pakis
mm:=1: # Refleksi
xx:=1: # Rotasi
B:=0.28: # Lebar daun

for n from 0 to Nmax do
r:=fern1():
if r<5 then
Z[n+1]:=[W(0,0,0,op(1,Z[n]),op(2,Z[n])),W(0,d,-
0.14,op(1,Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<86 then
Z[n+1]:=[W(0.85,b,0,op(1,Z[n]),op(2,mm*Z[n])),W(c,0.09,1,
op(1,mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r
elif r<93 then  Z[n+1]:=[W(0.09,-
B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.60,op(1,mm*Z[
n]),op(2,Z[n]))]:h[n+1]:=r:
else Z[n+1]:=[W(-
0.09,B,0,op(1,Z[n]),op(2,mm*Z[n])),W(0.30,0.09,0.70,op(1,
mm*Z[n]),op(2,Z[n]))]:h[n+1]:=r:
```

```
fi:od:
fern1:=[xx*Z[i]$i=1..Nmax]:
ulang:=[h[i]$i=1..Nmax]:
ulang2:=[[h[i],Z[i]]$i=1..Nmax]:
pointplot(fern1,style=point,symbol=point,axes=NONE,scalin
g=constrained,color=green);
```

 # Daun pakis 1 dengan nilai *d* pada w_2 = 0.09

 # Daun pakis 1 dengan nilai *d* pada w_2 = 0.99