14251/H/02

MILIK PERPUSTAKAAN
ITS

APLIKASI METODE FRAKTAL UNTUK PERBAIKAN CITRA GARIS

TUGAS AKHIR

RSF
621.367
Hid
a-1
2002

Oleh :

WURYAN HIDAYAT
NRP. 2495 100 063

JURUSAN TEKNIK FISIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA

2002

PERPUSTAKAAN
ITS

Tgl. Terima	26-2-2002
Terima Dari	H
No. Agenda Prp.	215210

APLIKASI METODE FRAKTAL
UNTUK PERBAIKAN CITRA GARIS

TUGAS AKHIR

Oleh :

WURYAN HIDAYAT
NRP. 2495 100 063

Surabaya, 30 Januari 2002

Mengetahui / Menyetujui

Pembimbing

Ir. APRIANI K., M.Sc.
NIP. 131 652 210

Ketua Jurusan

Teknik Fisika FTI - ITS

Ir. YA'UMAR, M.T.
NIP. 130 937 708

ABSTRAK

ABSTRAK

Fraktal dapat didefinisikan sebagai suatu kurva. Dengan fraktal dapat dibuat suatu gambar yang dapat membesar tak terbatas yang dihasilkan dari hanya sekumpulan kecil instruksi atau persamaan dan data. Dalam tugas akhir ini dilakukan kajian mengenai pengembangan fraktal untuk perbaikan citra garis pantai berdasarkan data penginderaan jauh dengan menggunakan metode interpolasi fraktal . Fungsi interpolasi fraktal memberikan suatu cara baru untuk memperoleh citra yang lebih alamiah. Dalam pengujiannya, digunakan data koordinat garis pantai (x,y) Pulau Jawa pada daerah 110° - 114,5° BT yang diambil dari data citra bitmap, dengan d_1=0.5 sebagai faktor skala vertikal yang menentukan kekasaran relatif bagian tertentu dari kurva dan dimensi fraktal D_{in} = 1.14. kemudian diperoleh estimasi Dimensi fraktal D_{out} = 1.114217 sehingga diperoleh nisbah kesalahan (error) = 2,56% untuk 75 titik data dan untuk 21 titik data memiliki keluaran dimensi sebesar 1.079676 dengan error 5,29%.

KATA PENGANTAR

KATA PENGANTAR

Syukur Alhamdulillah kami panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya kepada kami sehingga kami dapat menyelesaikan penyusunan laporan Tugas Akhir dengan judul **"APLIKASI METODE FRAKTAL UNTUK PERBAIKAN CITRA GARIS "**

Tugas Akhir merupakan salah satu mata kuliah wajib pada tahap sarjana bagi seluruh mahasiswa jurusan Teknik Fisika dimana juga merupakan penentu bagi tercapainya gelar kesarjanaan. Dengan selesainya penyusunan laporan tugas akhir ini, maka tidak berlebihan jika kami mengucapkan terima kasih kepada pihak-pihak yang telah dengan ikhlas membantu kami, antara lain :

1. Ibu Ir. Apriani Kusumawardhani, Msc. Selaku dosen pembimbing yang di tengah kesibukannya telah berkenan meluangkan waktu untuk memberikan bimbingan dan arahan pada penulis.
2. Bapak Ir. Ya'umar, MT. Selaku Ketua Jurusan Teknik Fisika FTI-ITS.
3. Bapak Ir. Agung Budiono, ME, selaku dosen wali yang telah banyak memberikan nasehat dan sabar saat perwalian selama penulis menempuh studi di Teknik Fisika FTI-ITS.
4. Bapak Hendra Cordova, ST, MT, atas segala masukan dan diskusinya.
5. Bapak Ir. Wiweka, MT, yang telah memberikan ide dan bimbingan awal dalam menyelesaikan tugas akhir ini.

DAFTAR ISI

DAFTAR ISI

LEMBAR JUDUL		i
LEMBAR PENGESAHAN		ii
ABSTRAK		iii
KATA PENGANTAR		iv
DAFTAR ISI		vi
DAFTAR GAMBAR		viii
DAFTAR TABEL		ix
BAB I PENDAHULUAN		1
1.1	Latar Belakang	1
1.2	Permasalahan	2
1.3	Batasan Masalah	2
1.4	Tujuan	2
1.5	Metodologi	3
1.6	Sistematika Penulisan Laporan	3
BAB II TEORI PENUNJANG		4
2.1	Konsep Dasar Geometri Fraktal	4
2.2	Geometri Euclidean	5
2.3	Garis Lurus dan kurva	6
2.4	Dimensi Fraktal	9
2.5	Transformasi Affine dan Iterated Function System	11
2.6	Matrik Transformasi	12
2.7	Transformasi Affine	17

2.8 Iterated Function System 17

2.9 Interpolasi Fraktal 18

BAB III APLIKASI METODE FRAKTAL PADA CITRA GARIS 21

3.1 Analisa Matematis Interpolasi Fraktal 21

3.2 Fraktalisasi Garis Citra 26

3.3 Aplikasi Metode Interpolasi Fraktal pada Citra Garis 27

BAB IV PEMBAHASAN DAN HASIL 31

4.1 Sistem Format berkas Data 31

4.2 Algoritma Program Perbaikan Citra Garis 32

4.3 Analisa dan Pengujian Program 37

BAB V KESIMPULAN DAN SARAN 40

5.1 Kesimpulan 40

5.2 Saran 41

DAFTAR PUSTAKA

LAMPIRAN 1 Hasil Numerik Interpolasi Fraktal

LAMPIRAN 2 Diagram Alir dari Diagram Alir Utama

LAMPIRAN 3. Listing Program

DAFTAR GAMBAR

DAFTAR GAMBAR

Gambar 2.1 Segitiga Sierpinski 7

Gambar 2.2 Kurva Koch 11

Gambar 2.3 Rotasi terhadap sumbu-x 15

Gambar 2.4 Rotasi terhadap sumbu-y 16

Gambar 3.1 Ilustrasi interpolasi fraktal pada citra garis 24

Gambar 3.2 Penentuan lengan pada fraktalisasi garis 27

Gambar 3.3 Data citra 27

Gambar 3.4 Teknik pengambilan data koordinat 28

Gambar 3.5 Kurva titik data citra garis 30

Gambar 4.1 Diagram alir fungsi utama program 32

Gambar 4.2 Contoh hasil proses iterasi 33

Gambar 4.3 Citra garis yang telah mengalami interpolasi 34

Gambar 4.4 Perbandingan kurva dengan 75 titik data 35

Gambar 4.5 Perbandingan kurva dengan 21 titik data 36

DAFTAR TABEL

DAFTAR TABEL

Tabel 2.1 Tabel IFS codes untuk fraktal segitiga sierpienski 18

Tabel 3.1 Tabel titik data kedudukan 29

Tabel 4.1 Perhitungan dimensi untuk 75 titik data 38

Table 4.2 Perhitungan dimensi untuk 21 titik data 38

Tabel 4.3 Nilai error dimensi 39

BAB I

PENDAHULUAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pantai merupakan bentuk geomorfologi yang dinamis, terutama diakibatkan oleh bekerjanya tenaga eksogen seperti abrasi dan akresi. Secara umum proses abrasi pada awalnya terjadi pada bagian pantai yang terkena air laut hingga membentuk suatu lekukan. Lapisan tanah, batu, serta tanaman yang terletak di atas lekukan tersebut secara bertahap akan melapuk lalu runtuh, yang mengakibatkan pantai perlahan-lahan akan melandai dan mundur ke arah daratan, Materi yang terkikis tersebut diangkut oleh arus laut, kemudian diendapkan di bagian-bagian yang landai. Gerakan-gerakan sedimen di sepanjang pantai akan menimbulkan erosi dan sedimentasi. Faktor-faktor yang menyebabkan terjadinya gerakan sedimen sehingga mengakibatkan perubahan kedudukan garis pantai dapat dikelompokkan menjadi dua hal, pertama akibat secara alami dan kedua akibat aktivitas manusia [Suntoyo dkk.,1997].

Perubahan garis pantai akibat proses akresi dan abrasi ini dapat diamati dengan menggunakan data penginderaan jauh yang dikembangkan lebih jauh lagi dengan menggunakan metode Interpolasi fraktal, sehingga diperoleh perbaikan citra garis pantai yang lebih mendekati kondisi sebenarnya. Pantai dalam kajian penginderaan jauh, seringkali dilihat (diolah) sebagai citra garis yang merupakan sekumpulan piksel. Dalam tugas akhir ini, garis pantai yang selanjutnya didefinisikan sebagai citra garis,

Penggunaan geometri fraktal untuk pebaikan citra garis karena deskripsi fraktal mengenai sistem-sistem partikel halus.

1.2 Permasalahan

Citra garis yang diperoleh dengan menggunakan prinsip geometri euclidean tingkat kekasarannya kurang terlihat. Dengan memanfaatkan metode interpolasi fraktal diharapkan dapat memperbaiki (*enhancement*) kontur yang kasar.

1.3 Batasan Masalah

Adapun batasan masalah dalam penulisan tugas akhir ini adalah sebagai berikut :

Metode yang digunakan adalah metode interpolasi fraktal.

- Proses interpolasi menggunakan bahasa pemrograman C.
- Data yang digunakan berupa data citra bitmap.
- Citra garis pantai didefinisikan sebagai citra garis
- Dalam pengolahannya digunakan koordinat piksel pada monitor
- Citra hasil tidak ditransformasikan dalam koordinat geografi
- Untuk validitas data menggunakan error dimensi

1.4 Tujuan

Tujuan dari penelitian tugas akhir ini adalah untuk memperoleh citra garis dengan kontur yang lebih kasar yang diturunkan dari data kedudukan piksel citra garis pantai utara Pulau Jawa dengan pengembangan Geometri fraktal.

1.5 Metodologi

Dalam melakukan penelitian tugas akhir ini, ditempuh tahapan-tahapan sebagai berikut:

- studi literatur untuk memahami secara teoretik penerapan geometri fraktal pada data citra.
- pembangunan perangkat lunak simulasi menggunakan *Turbo C 3.0*
- pengujian dan analisis metoda interpolasi fraktal untuk perbaikan citra garis

1.6 Sistematika Penulisan

Penulisan laporan penelitian dalam rangka tugas akhir ini diawali dari bab pendahuluan yang terdiri dari latar belakang, tujuan, batasan masalah dan metodologi penelitian yang dipilih.

Bab teori penunjang berisikan dari konsep dasar yang berhubungan dengan Geometri Fraktal seperti Dimensi Fraktal, Transformasi Affine, Iterated Function System (IFS) dan sebagainya.

Bab selanjutnya adalah bab metode interpolasi fraktal berisikan materi yang berkaitan dengan pengenalan Metode Interpolasi Fraktal sebagai metode pengolahan data citra agar diperoleh kontur citra garis yang lebih kasar.

Berikutnya adalah bab simulasi dan analisis program data citra secara matematis dan simulasi program sebagai alat bantu mempermudah analisa data.

Bab terakhir adalah bab kesimpulan yang berisikan kesimpulan-kesimpulan yang ditarik dari penelitian dalam rangka tugas akhir ini.

BAB II

TEORI PENUNJANG

BAB II

TEORI PENUNJANG

2.1 Konsep Dasar Geometri Fraktal

Geometri fraktal adalah bahasa baru. Begitu diucapkan, maka dapat digambarkan awan sama persisnya seperti seorang arsitek dapat menggambar sebuah rumah (Barnsley,1988). Kata fraktal dicetuskan pertama kali oleh Mandelbrot pada tahun 1975. Akar kata fraktal berasal dari kata *frangere* yang berarti terbelah menjadi fragmen-fragmen yang tidak teratur. Kata fraktal sendiri digunakan Mandelbrot untuk menyatukan karya-karya ilmuwan yang ada sebelumnya.Terdapat beberapa definisi mengenai fraktal beberapa diantaranya memakai ketentuan dan bahasa matematika dan statistik yang sukar dimengerti oleh orang awam. Barnsley, seorang pakar fraktal, enggan untuk mendefinisikan fraktal. Dia hanya mengatakan bahwa fraktal adalah subset (subhimpunan) dari sebuah set (himpunan). Set biasanya dari geometri eucledian yang sederhana seperti bentuk segitiga banyak, lingkaran, kubus, bola sedangkan subsetnya berbentuk sangat "rumit". Pada saat kita menggambar sebuah garis di atas kertas, gambar tersebut menurut geometri Eucledian dikatakan berdimensi satu yaitu panjang. Namun jika kita memperpanjang garis tersebut, membuatnya berlingkar-lingkar kedepan dan kebelakang tanpa memotong garis itu sendiri sehingga memenuhi seluruh permukaan kertas, menurut Eucledian gambar tersebut tetaplah berdimensi satu. Akan tetapi menurut penglihatan mata, jelas bahwa jika suatu garis mengisi seluruh permukaan kertas harusnya gambar itu berdimensi dua.

Pada sejarah revolusi matematika ruang lebih seabad yang lalu , beberapa ahli matematika seperti Cantor, Peano, Van Koch, Hausdorff dan Besicovitch menggambar suatu kurva yang mereka sebut *monster* dan para ahli matematika tradisional lainnya menyebut "*Pathological*" atau penyakit. Pada saat itu tipe baru dari suatu dimensi mulai diusulkan dimana ada suatu kurva yang mempunyai dimensi pecahan (*fractional*). Namun tanpa komputer digital dengan kecepatan tinggi penggambaran kurva-kurva tersebut akan memakan waktu yang lama hingga dilengkapi dengan komputer modern dan canggih untuk menyelidiki persamaan-persamaan yang beriterasi dan selalu diasumsikan bahwa persamaan sederhana selalu menghasilkan hasil yang sederhana dan persamaan yang kompleks akan menghasilkan hasil yang rumit maka dikatakan hasil itu sebagai *noise* atau kesalahan eksperimen.

2.2 Geometri Eucledian

Sadar atau tidak dalam diri manusia telah tertanam pengertian geometri dari suatu bentuk berdasarkan gagasan yang telah dicetuskan oleh Euclid dari Alexandria (300 SM) dan Descrates dari perancis (permulaan abad 16). Euclid membuat aksioma bahwa garis adalah "panjang yang tak bertebal". Dari aksioma ini kemudian dapat dibuat aturan-aturan logika konsisten yang dapat menerangkan tentang titik, garis lurus, dan bentuk-bentuk sederhana. Descrates memajukan gagasan bahwa alam raya ini seharusnya dapat diukur melalui tiga buah garis yang saling tegak lurus. Dengan tiga garis lurus ini lokasi baris apa saja dapat diketahui dengan tepat sebagai konsekuensinya semua benda dapat dilihat sebagai suatu susunan raksasa dari kubus-kubus yang kecil.

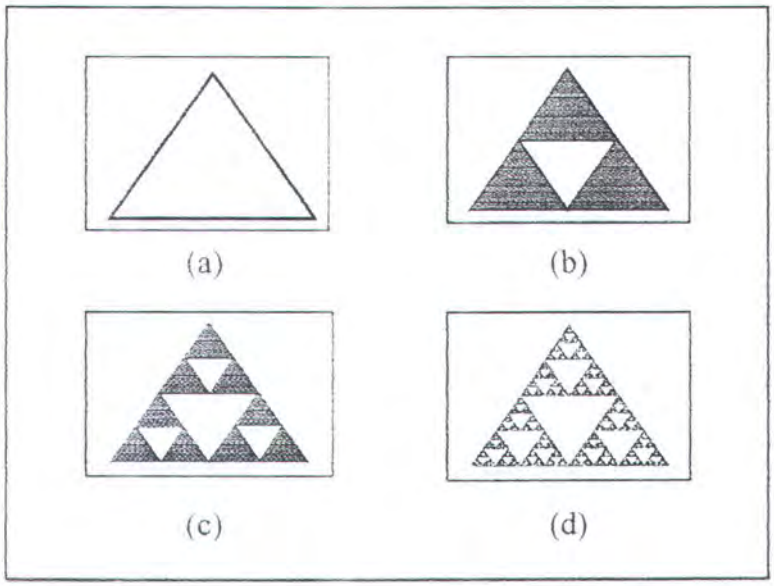
2.3 Garis Lurus dan Kurva

Berdasarkan kesimpulan dari pandangan di atas, Newton dan Leibnitz menemukan kalkulus differensial. Dalam kalkulus differensial semua bentuk lengkung atau kurva berubah menjadi lurus sehingga persamaan linier dapat digunakan untuk kurva. Leibnitz mengajukan gagasan bahwa semua kurva terdiri dari segmen-segmen garis lurus yang kecil tidak berhingga yang disebut sebagai "garis-garis tangen atau differensial". Jika sisi suatu kurva diperbesar semakin terlihat sebagai garis lurus.

Kalkulus memberikan suatu *limit* untuk proses pembesaran ini yaitu kurva akan menyerupai garis lurus pada pembesaran yang tak terhingga. Hingga saat ini semua orang menggunakan teknik differensiasi dan kebalikannya yaitu integrasi untuk merumuskan dan memahami kejadian alam. Walaupun tidak diketahui apakah gagasan Liebnitz itu benar, semua orang tetap bersandar dan percaya bahwa sebuah kurva tidak lebih dan tidak bukan terdiri dari sejumlah tak berhingga segmen-segmen garis lurus.

Waclaw Sierpinski matematikawan Polandia membuat segi tiga sama sisi yang kemudian dibaginya menjadi empat belahan yang berukuran sama dan sebangun seperti yang ditunjukkan oleh gambar 2.1a dan 2.1b. Dengan cara yang sama Sierpinski meneruskan pembagian tersebut untuk segitiga-segitiga lain yang lebih kecil hingga muncul bentuk seperti yang ditunjukkan oleh gambar 2.1d. Jika proses pembagian ini dilanjutkan hingga tak terhingga, sulit kiranya membayangkan bentuk detailnya. Yang jelas kalau salah satu bagian yang gelap diambil dan segitiga diperbesar mendekati tak berhingga akan didapatkan bentuk

seperti bentuk keseluruhannya. Segitiga sierpinski ini merupakan pra-fraktal yang pertama.



Gambar 2.1 Segitiga Sierpinski

Cara lain untuk membuat segitiga Sierpinski adalah dengan mula-mula membuat segitiga yang berisi, kemudian segitiga itu dilubangi tengah-tengahnya dan ketiga bagian sudut-sudutnya dengan segitiga yang berukuran lebih kecil. Selanjutnya proses pelubangan yang sama untuk setiap sisa segitiga yang masih berisi diulangi terus hingga jumlah yang tak berhingga. Sierpinski mempertanyakan apakah luas yang ditutupi oleh bentuk tersebut adalah nol atau tidak ? Inilah teka-teki yang membingungkan Sierpinski yang mirip dengan ketidakpastian Leibnitz akan gagasannya tentang kurva yang menjadi garis lurus pada perbesaran mendekati tak terhingga.

Dalam setiap langkah hanya $\frac{1}{4}$ dari luas daerah berisi saja yang diambil, dan $\frac{3}{4}$ bagian sisanya atau sebagian besar darinya masih berisi. Berapapun

banyaknya proses pelubangan yang dilakukan maka akan tetap diperoleh luas daerah berisi yang lebih besar dari luas yang diambil setiap kalinya. Jadi luas bentuk ini tidak pernah mencapai nol. Namun sesuai dengan aksioma Euclid bahwa garis matematis adalah garis yang tidak memiliki ketebalan maka luas daerah tersebut adalah nol.

- **Definisi fraktal**. Fraktal dapat didefinisikan sebagai suatu kurva. Dengan fraktal dapat dibuat suatu gambar yang dapat membesar tak terbatas yang dihasilkan dari hanya sekumpulan kecil instruksi atau persamaan dan data. Pada gambar fraktal , semakin di zoom (diperbesar) suatu titik atau area dalam gambar tersebut semakin detail gambar yang akan diperoleh.
- **Prinsip ilmu fraktal.** Ada dua prinsip yang paling penting di dalam fraktal, dimana prinsip kedua hanya memperbagus/memperbaiki yang pertama.
  - Dalam sistem-sistem alamiah , struktur dari keseluruhan sistem sering tercermin di dalam setiap bagiannya.
  - Suatu sistem akan tampil sebagai sistem yang *self-similarity* bilamana pada banyak tingkatan skalanya bekerja gaya-gaya yang sama.

Dari prinsip diatas dapat dikatakan bahwa kunci dalam memodelkan sistem fraktal adalah menemukan gaya-gaya yang sekaligus bekerja pada banyak tingkatan.

- **Self Similarity.** Apabila dua bangun geometri memiliki sifat yang dapat diperoleh dari yang lain dengan skala dan ketentuan tertentu maka bangun tersebut memiliki sifat *self similarity.*

Dasar dari penggambaran fraktal adalah dibuat secara rekursif dengan mengganti gambar yang sama tetapi berukuran lebih kecil. Keadaan ini disebut self-similarity dalam bentuk , yang berarti bahwa tiap-tiap bagian dari gambar

tersebut apabila diperbesar tetapi akan mempunyai bentuk yang sama, yaitu mempunyai bentuk keseluruhan dan bila gambar hasil pembesaran tersebut diambil sebagian kecil dan diperbesar maka bentuk seperti keseluruhanlah yang muncul.

2.4 Dimensi Fraktal

Abad XX telah membawa pemikiran manusia pada kebutuhan yang sangat akan suatu cara baru dalam pengukuran ruang dan dimensi. Dua orang matematikawan , Felix Hausdorff dan Abram S. Besicovitch telah mendefinisikan ulang dimensi itu sendiri . Setiap bentuk, oleh karena tradisi yang dilandasi oleh gagasan decrates memiliki dimensi misalnya 0 (titik), 1 (garis lurus dan kurva), 2 (bidang), atau 3 (ruang). Secara teoritis dimensi ini telah diperluas menjadi dimensi-dimensi yang lebih tinggi sehingga sulit untuk dibayangkan dengan memperluas karya Hausdorff, Besicovitch memajukan gagasan bahwa sebuah bentuk sebenarnya dapat memiliki dimensi pecahan seperti 1,5 atau 2,3. Dimensi pecahan ini dapat dihitung dengan tepat berdasarkan pengukuran dari sebuah kurva. Dengan demikian tingkah laku ganjil dari suatu kurva yang lebih kompleks dapat dijelaskan. Dimensi pecahan ini dapat dihitung dengan tepat berdasarkan pengukuran dari sebuah kurva. Dimensi Hausdorff/Besicovitch didefinisikan sebagai nisbah dari logartma jumlah saliinan dan ukuran dari bentuk benih terhadap setap salinan.

Dalam penggambaran geometri fraktal, dapat dimulai dengan sebuah benih seperti pada gambar 2.2 yaitu gambar geometri sederhana yang terdiri dari sejumlah segmen garis yang lain saling dihubungkan, misal sebuah segi tiga,

MILIK PERPUSTAKAAN ITS

bujur sangkar, atau sebuah garis lurus saja sebagai inisiator lalu ditetapkan sebuah generator. Generator adalah runtutan segmen garis untuk menggantikan setiap segmen garis dari benih. Generator terdiri dari K segmen garis masing-masing dengan panjang dimana M adalah faktor pergandaan dari segmen garis yang diulang dan N sebagai jumlah iterasi. Generator adalah sama panjangnya dengan segmen yang diganti. Proses perganian ini diulang sampai waktu yang tak terhingga , masing-masing waktu disetiap waktunya mengganti tiap segmen garis dari kurva pada tingkatan sebelumnya. Dengan demikian dimensi Hausdorff-Besicovitch sebagai akibat dari kurva fraktal dapat dirumuskan sebagai berikut :

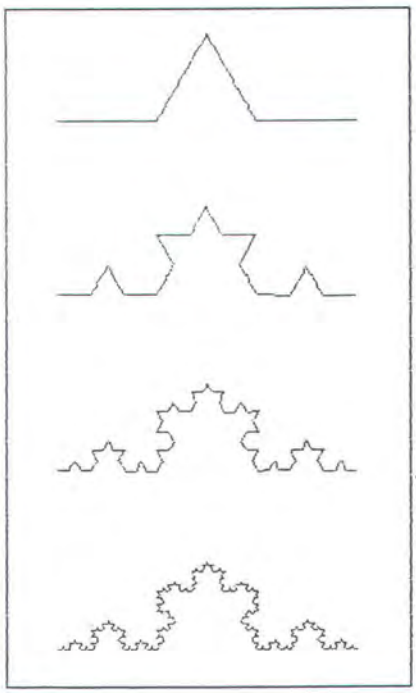
$$D = \frac{\log K}{\log M} \tag{2.1}$$

Sebagai contoh, garis pantai Koch. Dimensi garis pantai Koch karena ada 4 salinan (4 segmen garis) dan setiap salinan memiliki faktor penggandaan $^1/_3$ kali ukuran awal (lihat gambar 2.3), maka menurut definisi ini, dimensi garis pantai Koch adalah D = log (4) / log (3) = 1,2618.

Jika ada dua bentuk yang memiliki dimensi pecahan yang berbeda, misalnya 1,26 dan 1,48 tidak dapat dikatakan bahwa bentuk yang pertama memiliki panjang tak berhingga yang lebih panjang atau mengisi luasan yang lebih kecil tak berhingga lebih banyak dari yang kedua. Sekarang yang dapat dikatakan adalah bahwasanya dimensi bentuk itu, lebih dekat dengan dimensi dua.

Dalam kenyataannya, penggunaan gagasan dimensi pecahan telah melangkah lebih jauh dari apa yang dibayangkan, karena alam berlimpah dalam bentuk-bentuk swa-reflektif seperti garis pantai, maka kebanyakan dari alam dapat dicirikan dengan indeks yang baru ini. Ciri dari sebuah bentuk dapat dilihat dari

dimensinya saja. Garis pantai pulau Sulawesi yang kasar memiliki dimensi pecahan yang lebih besar dari pada garis pantai pulau Bali yang halus. Gumpalan awan mengisi ruang lebih banyak dari kabut tipis dan bangunan indah seperti candi borobudur di magelang memiliki dimensi pecahan yang lebih besar dari pada pencakar langit di Jakarta.



Gambar 2.2 Kurva Koch

Istilah dimensi pecahan kemudian oleh Mandelbrot diganti menjadi "dimensi fraktal" . Dimensi ini jauh lebih penting artinya bagi matematikawan karena mereka dapat mengukur keseluruhan bentuk-bentuk dalam jagad raya , yang sebenarnya tidak dapat diukur.

2.5 Transformasi Affine dan Iterated Function System (IFS)

Pada fungsi matematika, operasi transformasi atau operasi pemetaan, misalnya p = f(q) menyatakan bahwa domain q dipetakan ke p oleh fungsi f.

Dalam fraktal, variabel q mewakili suatu bentuk atau himpunan titik-titik yang akan ditransformasikan, sedangkan p menyatakan bentuk yang tercipta akibat operasi fungsi f terhadap q. Dalam *Iterated Function System* (IFS), fungsi transformasi awal diberikan suatu bentuk pertama yang disebut benih, kemudian bentuk hasil transformasinya diumpanbalikkan ke fungsi yang sama. Proses pengumpanbalikkan ini dilakukan secara terus menerus. Jika fungsi yang diiterasikan memiliki *attractor* maka pada hasil akhirnya adalah berupa bentuk fraktal.

Transformasi yang digunakan untuk membangkitkan bentuk-bentuk fraktal seperti segitiga Sierpinski dan garis pantai Koch merupakan suatu transformasi yang disebut sebagai transformasi affine. Transformasi affine adalah suatu transformasi linier yang terdiri dari tiga operasi dasar yaitu penyekalan (*Scaling*), pemutaran (*rotation*), dan translasi (*translation*). Transformasi dinamakan affine karena terdapat afinitas visual dan struktural antara bentuk lama dan bentk baru. Misalnya, segitiga tetaplah segitiga meskipn dirotasikan, dikecilkan ataupun ditranslasikan.

2.6 Matriks Transformasi

Transformasi titik-titik dalam suatu bidang dapat dinyatakan dengan suatu perkalian matriks transformasi T (2 x 2), dan vektor V (1 x 2) yang mewakili koordinat titik tersebut, yaitu :

$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad V = \begin{bmatrix} x \\ y \end{bmatrix}$$

misalkan sebuah titik asal (x_1,y_1) ditransformasikan ke sebuah titik baru (x_2,y_2) dengan operasi matriks sebagai berikut:

$$V_2 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} ax_1 + by_1 \\ cx_1 + dy_1 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \tag{2.2}$$

jadi untuk sebarang titik (x,y) setelah ditransformasikan oleh matriks T akan menjadi :

$$W = \begin{bmatrix} x' \\ y' \end{bmatrix} = T \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix} \tag{2.3}$$

- **Pergeseran.** Sebarang titik pada bidang *xoy* dapat digeser ke sebarang posisi dengan menambahkan besaran pda arah sumbu *x* dan *y*.

$$Q = P\,I + T \tag{2.4}$$

Sehingga diperoleh hasil pergeseran

$$(Q_x, Q_y) = (P_x + T_x, P_y + T_y) \tag{2.5}$$

dengan

P : posisi titik asal

Q : posisi titik setelah digeser

I : matriks identitas

T : besar pergeseran dengan arah x dan y

dalam matriks dinyatakan

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x + move_x \\ y + move_y \end{bmatrix} \tag{2.6}$$

➢ **Penyekalaan.** Penyekalaan adalah proses untuk memperbesar atau memperkecil gambar, penyekalaan dapat dilaksanakan ke arah sumbu x saja, y ataupun ke arah kedua sumbunya. Penyekalaan sendiri dibentuk oleh matriks diagonal, karena

$$s = \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax \\ dy \end{bmatrix}, \tag{2.7}$$

sehingga elemen diagonal matriks dapat digunakan untuk melakukan penyekalaan pada arah sumbu koordinat. Misalnya titik-titik dapat direnggangkan dalam arah sumbu x dengan faktor 2 (x stretch) melalui perkalian dengan matriks

$$s = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}.$$

Untuk mengerutkan titik-titik dalam arah sumbu y dengan faktor $^1/_5$ (y shrink), kalikan dengan matriks

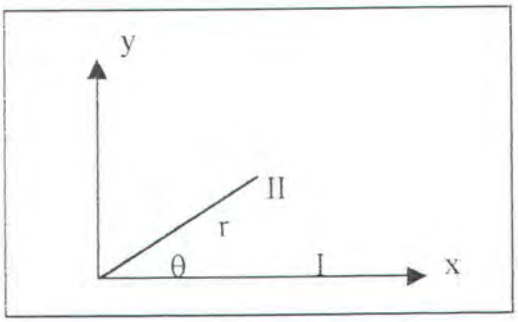
$$s = \begin{bmatrix} 1 & 0 \\ 0 & 0{,}2 \end{bmatrix}$$

Penyekalaan dengan faktor k yang sama dalam arah x,y (pengecilan atau pembesaran) dapat dikalikan dengan matriks

$$s = \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix} \tag{2.8}$$

Jadi transformasi penyekalaan terdiri dari perenggangan dalam arah x (x stretch), y (y shrink) dan pengerutan dalam arah x (x stretch), y (y shrink) serta pengecilan atau pembesaran.

➢ **Rotasi.** Titik-titik dari sebuah bentuk dapat dirotasikan berlawanan arah jarum jam sebagai sudut positif dengan mengalikannya dengan matriks. Rotasi terhadap sumbu-x dapat dijabarkan seperti pada gambar 2.3



Gambar 2.3 Rotasi terhadap sb-x

dengan:

I : Titik awal (sebelum dirotasi)

II : Hasil rotasi terhadap sumbu-x

θ : Rotasi pada komponen x

r : Menskala komponen x

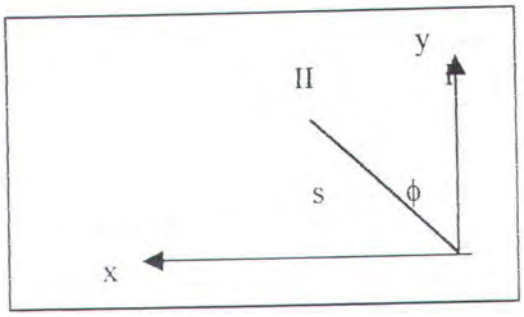
sudut trigonometri : $x = r \cos\theta \approx \cos\theta$

$y = r \sin\theta \approx \sin\theta$

jika titik-titik pada gambar 2.3 dirotasikan terhadap sumbu-x [1,0] maka berdasarkan persamaan (2.3) diperoleh persamaan (2.9).

$$T\overline{v}_1 = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ c \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{2.9}$$

Sedangkan rotasi terhadap sumbu-y dapat digambarkan sebagai berikut :



Gambar 2.4 Rotasi terhadap sb-y

dengan:

I : Titik awal (sebelum dirotasi)

II : Hasil rotasi terhadap sumbu-y

ϕ : Rotasi pada komponen y

s : Menskala komponen y (faktor skala vertikal)

Sudut Trigonometri : $x = -s \sin\phi \approx -\sin\phi$

$y = s\cos\phi \approx \cos\phi$

jika titik-titik pada gambar 2.4 dirotasikan terhadap sumbu-x [0,1] maka berdasarkan persamaan (2.3) diperoleh:

$$T\overline{v_1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} -\sin\phi \\ \cos\phi \end{bmatrix} \tag{2.10}$$

dari persamaan (2.9) dan (2.10) diperoleh persamaan (2.11) untuk operasi Rotasi pada interpolasi fraktal.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\phi \\ \sin\theta & \cos\phi \end{bmatrix} \tag{2.11}$$

2.7 Transformasi Affine

Transformasi yang digunakan untuk mengukur segitiga sierpinski dan bentuk fraktal yang lain adalah transformasi affine yaitu transformasi linier yang terdiri dari tiga operasi yaitu penyekalaan (*scaling*), rotasi dan translasi. Bentuk umum dari transformasi affine adalah sebagai berikut :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \tag{2.10}$$

2.8 Iterated Function System (IFS)

Sistem yang mengiterasikan himpunan atau set transformasi affine kontraktif seperti diatas dinamakan iterated function system (IFS), sedangkan bilangan-bilangan a,b,c,d,e,f disebut sandi IFS (IFS code).Teknik pembentukan fraktal dengan IFS dipelipori oleh Barnsley dkk. Pencarian set transformasi affine dari bentuk geometri fraktal dapat dilakukan dengan metode langsung maupun tidak langsung yaitu melalui penyelesaian n bilangan anu. Metode langsung biasanya digunakan untuk bentuk geometri yang sederhana seperti segitiga sierpinski dan garis pantai koch.

Dengan metode langsung, set transformasi affine untuk segitiga sierpinski dapat diperoleh secara mudah yaitu dengan mula-mula membuat segitiga sama sisi dengan panjang sisi sama dengan 1 (lihat gambar 2.1). transformasi affine pertama, kedua dan ketiga memiliki kesamaan yaitu mula-mula harus memperkecil benih dengan faktor 0.5. ketiga transformasi affine tersebut hanya berbeda pada pemindahan segitiga hasil pengecilan keposisi yang tepat. Transformasi pertama tidak perlu melakukan translasi apapun. Jadi e=f=0.

transformasi kedua harus melakukan translasi arah mendatar sejauh 0.5 . Jadi e=0.5 dan f=0. transformasi ketiga perlu melakukan translasi sejauh 0.5 dalam arah vertikal. Jadi e=0 dan f=0.5 (jika panjang sisi dari benih adalah N satua panjang maka setiap e dan f harus dikalikan dengan N). set lengkap transformasi affine kontraktif untuk segitiga sierpinski dapat dituliskan sebagai berikut :

$$T_1\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0,5 & 0 \\ 0 & 0,5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$T_2\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0,5 & 0 \\ 0 & 0,5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$$

$$T_3\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0,5 & 0 \\ 0 & 0,5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$$

Agar lebih mudah, notasi diatas dapat dituliskan secara lebih singkat dalam bentuk sandi IFS seperti yang diperlihatkan pada tabel 2.1.

Tabel 2.1 . IFS code untuk fraktal segitiga sierpinski.

T	a	b	c	d	e	f
1	0,5	0	0	0,5	0	0
2	0,5	0	0	0,5	0.5	0
3	0,5	0	0	0,5	0	0.5

2.9 Interpolasi Fraktal

Dalam suatu percobaan atau pengamatan biasanya diukur dan dicatat nilai-nilai atau data dari suatu fungsi nyata f(x) sebagai fungsi dari variabel nyata x. dalam percobaan fisika, misalnya f(x) bisa menyatakan tegangan sebagai fungsi

waktu, atau pengamatan bursa saham f(x) bisa menyatakan harga indeks saham gabungan sebagai fungsi dari waktu. Untuk mendapat gambaran yang lebih jelas biasanya terlebih dahulu diplot titik-titik data pada kertas grafik. Kemudian grafik data dianalisis secara geometri. Misalnya dengan menarik sebuah garis lurus yang dianggap merupakan pendekatan atau interpolasi yang terbaik grafik tersebut, atau membentuk sebuah polinomial dengan derajat yang terendah yang paling mungkin sedemikian hingga grafik polinomial tersebut merupakan grafik yang paling sesuai dengan grafik data dalam selang $[x_0, x_N]$. selain polinomial, sering juga digunakan kombinasi linier dari beberapa fungsi elementer seperti misalnya $f(x) = 1+3x+x^2+\sin x+\sqrt{x}$. Tujuan dari proses ini semua sama yaitu untuk mewakili titik-titik data yang dilihat sebagai himpunan bagian dari sebuah bidang dengan entitas geometri yaitu dalam bentuk rumus sederhana sehingga dapat dengan mudah dianalisis dan dikomunikasikan kepada orang lain.

Metode interpolasi tradisional memiliki kekurangan yaitu tidak dapat membuat polinomial pencocok yang paling sesuai dalam kasus dimana data percobaan sangatlah kasar, seperti misalnya temperatur semburan mesin jet sebagai fungsi dari waktu, tegangan listrik pada sebuah titik dalam otak manusia yang dibaca oleh electroencephalograph, data seismik yang diperoleh dari seismograf, harga indeks bursa efek dalam suatu kurun waktu, data temperatur bumi sebagai fungsi waktu dan lain sebagainya. Tidak demikian halnya dengan interpolasi fraktal, fungsi-fungsi interpolasi fraktal dapat melakukannya dengan baik dalam pengertian matrik Hausdorff. Selain dari pada itu dapat dipastikan bahwa dimensi fraktal dari grafik fungsi interpolasi fraktal sesuai dengan dimensi fraktal dari grafik data percobaan dalam suatu jangkauan skala yang sesuai.

Seperti halnya fungsi-fungsi elementer, fungsi-fungsi interpolasi fraktal juga memiliki karakter geometris yaitu dapat diwakili secara ringkas oleh rumus-rumus dan dapat dihitung dengan cepat. Perbedaan utama terletak dari karakter fraktalnya. Misalnya grafik memiliki dimensi fraktal yang bukan bilangan bulat. Jadi fungsi-fungsi interpolasi fraktal memberikan metode baru dalam pencocokan data percobaan.

Selain beberapa kelebihan diatas, grafik dari fungsi interpolasi fraktal dapat pula dipergunakan untuk mendekatkan komponen-komponen citra seperti profil pegunungan, awan, stalaktif yang menggantung dalam gua, garis horison diatas sawah, garis pantai dan lain-lain. Dengan fungsi-fungsi interpolasi fraktal komponen-komponen citra tidak lagi diperlakukan sebagai kumpulan objek-objek terpisah seperti pegunungan, awan,garis pantai, melainkan dimodelkan sebagai satu kesatuan sistem yang saling berhubungan satu sama lain. Komponen-komponen citra seperti ini dapat digambarkan dengan baik oleh fungsi-fungsi interpolasi fraktal, suatu hal yang tidak dapat dilakukan oleh fungsi-fungsi elementer atau fungsi-fungsi grafik Eucledian.

BAB III

APLIKASI METODE FRAKTAL PADA CITRA GARIS

BAB III

APLIKASI METODE FRAKTAL PADA CITRA GARIS

Bentuk-bentuk geometri euclidean, trigonometri, dan kalkulus telah diajarkan elah dikenal seama di sekolah. Hal ini menyebabkan ketika memodelkan bentuk-bentuk yang dilihat, selalu terpola pada bentuk-entuk seperti garis lurus, lingkaran, parabola maupun kurva sederhana lainnya. Sebagai akibatnya, banyak terlihat rancangan bentujk-bentuk gedung, fungsi-fungsi dalam software grafik untuk menggambar titik, garis, kurva, poligon maupun lingkaran merupan bagian dari perujudan pola piker tersebut. Termasuk didalamnya adalah metode interpolasi tradisional yang biasa digunakan untuk menganalisis data yang dikumpulkan dari suatu pengamatan. Interpolasi tradisional dibuat berdasarkan fungs-fungsi elementer seperti sinus, cosinus dan polynomial yang berakar dari geometri euclidean. Demikian halnya pada perolehan bentuk citra garis

Pada kasus seperti di atas, dengan menggunakan fungsi-fungsi pada interpolasi fraktal, dapat dipergunakan untuk mendekatkan profil komponen-komponen citra.

3.1 Analisis Matematis Interpolasi Fraktal

Dalam Tugas Akhir ini untuk memperoleh perbaikan citra garis digunakan metode interpolasi fraktal. Interpolasi ini memerlukan tiga operasi dasar geometri, yaitu: peyekalaan, rotasi dan translasi. Adapun gabungan dari operasi penyekalaan dan rotasi dapat dituliskan sebagai berikut :

$$T_{SR} = T_S * T_R$$

dengan : T_{SR} : Transformasi gabungan

T_S : Transformasi Scalling

T_R : Transformasi rotasi

$$T_{SR} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\phi \\ \sin\theta & \cos\phi \end{bmatrix}$$
$$= \begin{bmatrix} k_1 \cos\theta & -k_1 \sin\phi \\ k_2 \sin\theta & k_2 \cos\phi \end{bmatrix} \quad (3.1)$$

dari persamaan (2.3) maka persamaan (3.1) dapat dituliskan sebagai berikut :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} k_1 \cos\theta & -k_1 \sin\phi \\ k_2 \sin\theta & k_2 \cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = T \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.2)$$

berdasarka definisi bahwa transformasi affine merupakan transformasi linier yang terdiri dari tiga operasi yaitu penyekalaan, rotasi dan translasi, maka dari persamaan (2.6) dan (3.2) diperoleh persamaan sebagai berikut :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = T \begin{bmatrix} x + move_x \\ y + move_y \end{bmatrix} \quad (3.3)$$

jika diambil e = *move*_x dan f = *move*_y maka persamaan (3.3) menjadi :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = T \begin{bmatrix} x + e \\ y + f \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \tag{3.4}$$

dengan :

a = r cos θ

b = - s sin ϕ

c = r sin θ

d = s cos ϕ

e = pergeseran ke arah x

d = pergeseran ke arah y

Dengan diketahuinya persamaan transormasi afine, sehingga fungsi interpolasi fraktal untuk perbaikan citra garis berdasarkan persamaan (3.4) dengan rotasi terhadap sumbu-y tidak disertakan, dapat dituliskan sebagai berikut:

a = r cos θ

b = - s sin ϕ = - s sin 0 = 0

c = r sin θ

d = s cos 0 = s . 1 = s (faktor skala vertikal)

Sehingga persamaan (3.4) menjadi :

$$T_i \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_i & 0 \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix} \tag{3.5}$$

dengan :

I = 1,2,3,.....N

N = (n-1) ; n adalah jumlah data.

Jika misalnya diberikan tiga titik data $(x_0 \quad y_0),(x_1 \quad y_1)$dan $(x_2 \quad y_2)$ maka diperlukan N= 3-1 = 2 transformasi affine untuk menginterpolasikan titik-titik data tersebut. Seperti terlukiskan dalam gambar 3.1.

Gambar 3.1 Ilustrasi Interpolasi Fraktal pada citragaris

Elemen-elemen transformasi a_1, c_1, d_1, e_1 dan f_1 harus dicari dari titik-titik data yang diberikan. Karena transformasi di atas dibatasi oleh data sesuai dengan persamaan :

$$T_i\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} x_{i-1} \\ y_{i-1} \end{bmatrix} \text{ dan } T_i\begin{bmatrix} x_N \\ y_N \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{3.6}$$

maka diperoleh persamaan-persamaan sebagai berikut :

$$\begin{aligned} a_i x_0 + e_i &= x_{i-1}. \\ a_i x_N + e_i &= x_i. \\ c_i x_0 + d_i y_0 + f_i &= y_{i-1}. \\ c_i x_N + {}_i y_N + f_i &= y_i. \end{aligned} \tag{3.7}$$

Dari persamaan (3.7) terdapat 4 persamaan linier dengan 5 bilangan anu a,c,d,e dan f. Jadi dalam setiap transformasi terdapat sebuah parameter yang nilainya

dapat diubah-ubah sesuai dengan keinginan. Dengan memilih d sebagai parameter bebas dan keempat persamaan tersebut dipecahkan maka akan diperoleh persamaan (3.8)

$$a_1 = \frac{(x_i - x_{i-1})}{(x_N - x_0)}$$

$$c_i = \frac{[(y_i - y_{i-1}) - (y_N - y_0)]}{(x_N - x_0)}$$

$$e_i = \frac{(x_N x_{i-1} - x_0 x_i)}{(x_N - x_0)}$$

$$f_i = \frac{(x_N y_{i-1} - x_0 y_i) - d(x_N y_0 - x_0 y_N)}{(x_N - x_0)} \qquad (3.8)$$

Ada dua alasan mengapa d_i dipilih sebagai parameter bebas. Pertama, transformasi T_i adalah transformasi *shear*. Transformasi ini akan memetakan garis-garis yang sejajar dengan sumbu-y kepada garis-garis yang juga sejajar dengan sumbu-y. nisbah panjang. Dengan $T_i\{L\}/L = d_i$ disebut juga dengan faktor penyekalaan vertical dari transformasi T_i. Jadi dengan memilih d_i sebagai parameter bebas, untuk setiap interval diantara dua titik data. Di sini fraktal dapat dinyatakan sebagai faktor penyekalaan vertikal untuk kurva interpolasi fraktal. Fraktol penyekalaan vertikal menentukan kekasaran relatif dari bagian tertentu tertentu dari kurva interpolasi dibandingkan dengan bagan yang lain. Mmisalnya, jika interval pertama memiliki faktor penyekalaan 0.5 dan interval-interval lainnya adalah 0.2, maka bagian paling kiri dari kurva interpolasi akan lebih tinggi dan lebih kasar dari bagian yang lainnya. Faktor penyekalaan negatif seperti –0.3

akan membuat bagian kurva dengan faktor ini akan terbalik. Faktor penyekalaan harus berada di antara –1 dan 1 ($-1<d_i<1$, dengan i = 1,2,3,....N).

Alasan kedua adalah jika titik-titik interpolasi mempunyai interval yang sama dan dengan mengambil asumsi bahwa;

$$\sum_{i=1}^{N} |d_i| > 1 \tag{3.9}$$

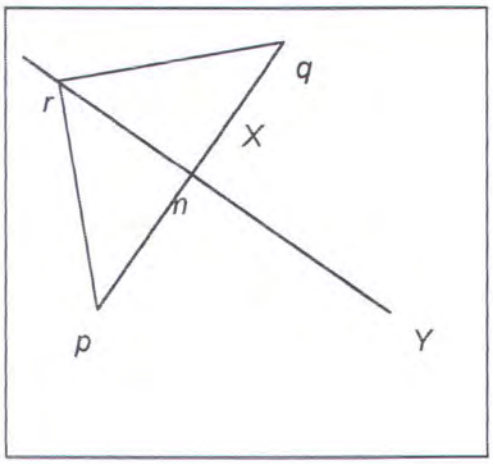
maka dapat dibuktikan bahwa antara d_i dengan dimensi fraktal D dari grafik interpolasi fraktal (*attractor IFS*) terdapat hubungan :

$$\sum_{i=1}^{N} |d_i| > N^{D-1} \tag{3.10}$$

Jadi dengan mengatur d_i dapat ditentukan dimensi fraktal dari grafik fungsi-fungsi interpolasi fraktal antara 1 sampai dengan mendekati 2 ($1<D<2$).

3.2 Fraktalisasi Garis Citra

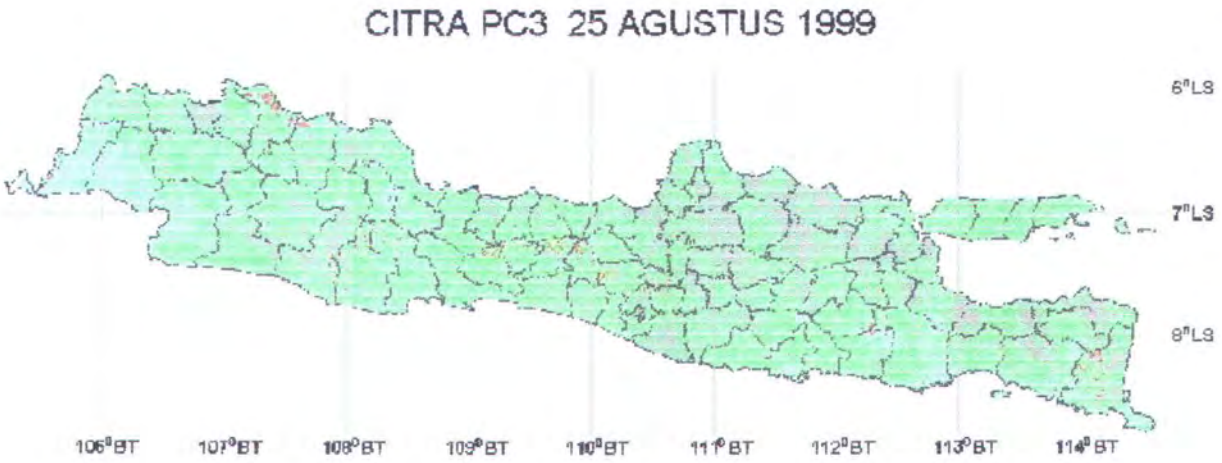
Fraktal yang paling sederhana dibentuk dengan fraktalisasi (*fraktalizing*) sebuah garis. Pada setiap langkah, suatu penggal garis digantikan dengan dua garis lain yang dibentuk secara acak. Gambar 3.1 menunjukkan contoh proses fraktalisasi sebuah garis, garis X, yang mempunyai titik-titik ujung p dan q. Penggal garis tersebut diganti dengan penggal garis dari p ke r, dan dari r ke q. Untuk fraktal kurva, titik r dipilih secara acak pada suatu garis Y yang tegak lurus terhadap X. Sehingga koordinat titik r dapat ditentukan sebagai



Gambar 3.2 Penentuan lengan pada fraktalisasi garis

3.3 Aplikasi Metode Interpolasi Fraktal pada Perbaikan Citra Garis

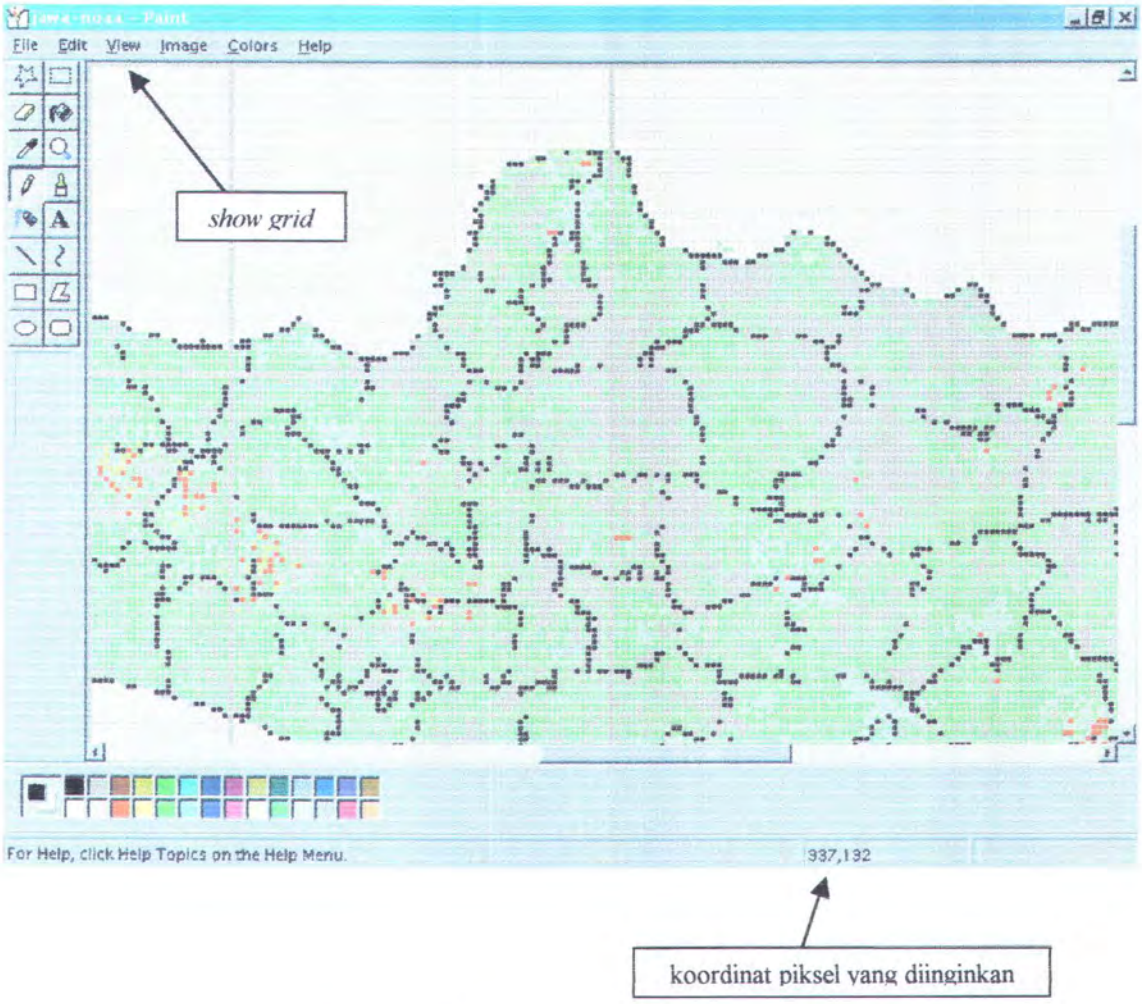
Dalam penerapan metode interpolasi fraktal, faktor yang perlu diperhatikan adalah pengambilan titik data. Titik data ini adalah berupa titik koordinat piksel citra terhadap monitor.



Gambar 3.3

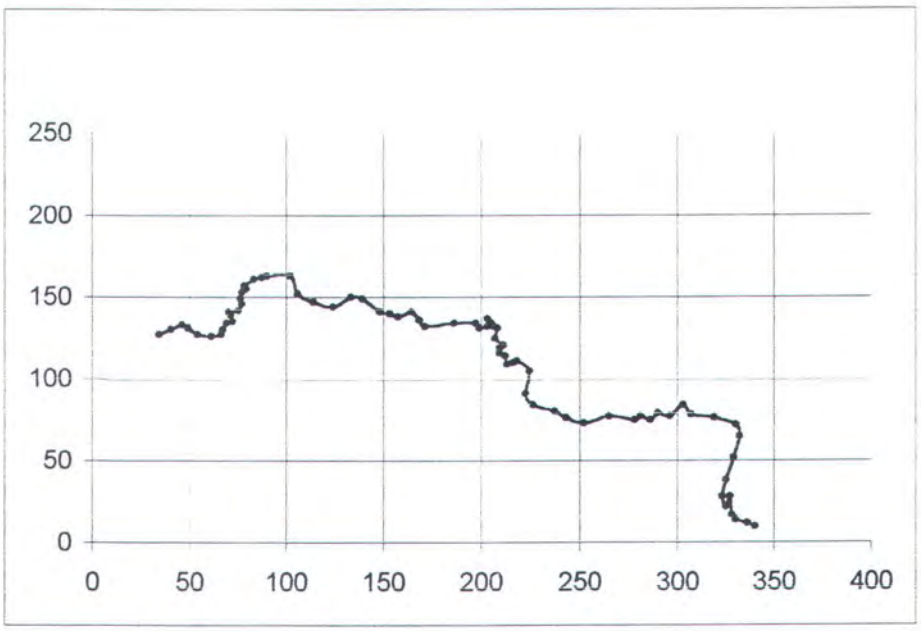
Data Citra (LAPAN Cijantung Jakarta Selatan-Indonesia)

➢ **Pengambilan Data.** Titik data diambil dengan memasukkan/membuka berkas citra pada gambar 3.3 ke dalam program aplikasi *paint* yang tersedia pada *operating system windows*. Setelah itu, dngan mimilih *show grid* pada *menu view* maka citra akan terbagi menjadi kotak-kotak, yang akan membaginya berdasarkan piksel masing-masing citra. Dengan mengarahkan mouse pada piksel yang diinginkan maka akan terlihat koordinat piksel tersebut. Seperti terlihat pada gambar 3.4



Gambar 3.4 Teknik pengambilan data koordinat

Apabila titik-titik data pada tabel 3.1 dikurvakan maka akan terbentuk kurva seperti pada gambar 3.5



Gambar 3.5 Kurva titik data citra garis

BAB IV

PEMBAHASAN DAN HASIL

BAB IV

PEMBAHASAN DAN HASIL

4.1 Sistem Format Berkas Data

Format berkas data untuk menjalankan program perbaikan citra ini adalah FIP, *fractal interpolation program.* Berkas ini dapat diciptakan dengan editor DOS, notepad ataupun editor lainnya yang dapat menghasilkan file ASCII. Nama fiel data boleh diberi nama apa saja. Namun nama penuh dan ekstensinya harus diperikan lengkap ketika diminta oleh program.

Adapun format datanya adalah sebagai berikut :

3			→	jumlah titik data
0.5	0.3		→	jarak antar kisi vertikal jarak antar kisi horizontal
1	6	0.5	→	x_0,y_0 dan faktor skala d_1
3	9	0.3	→	x_1,y_1 dan faktor skala d_2
7	13	0.7	→	x_2,y_2 dan faktor skala d_3

keterangan :

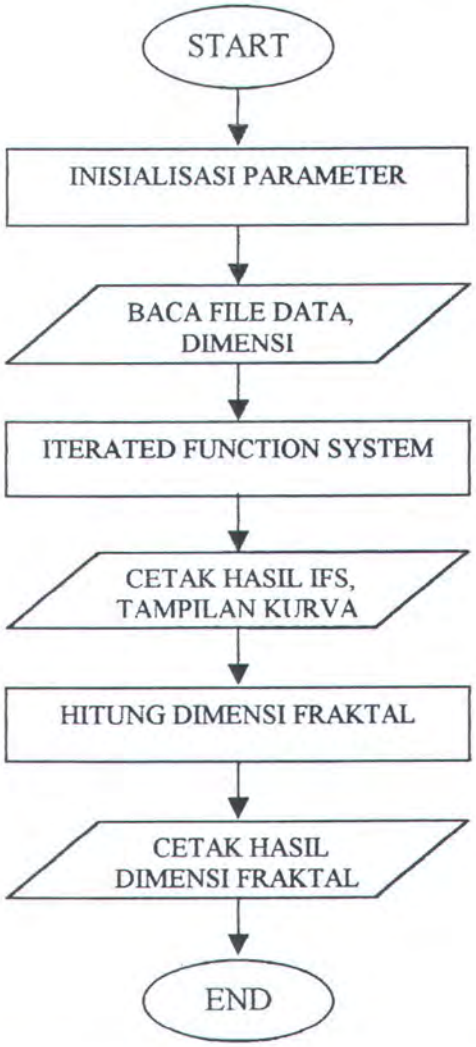
- Tanda panah serta keterangan sesudahnya tidak diperlukan, di sini diberikan hanya untuk penjelasan saja
- Bilangan pertama (jumlah titik data) harus ditempatkan pada baris pertama dari berkas data
- Faktor penyekalaan vertikal yang terakhir tidak mengandung arti apapun karena tidak ada interval setelah titik terakhir. Tetapi sebuah angka

sembarang harus dituliskan di sana karena program harus membaca 3 buah bilangan untuk setiap baris data

4.2 Algoritma Program Perbaikan Citra Garis

Untuk memperoleh citra garis yang diinginkan, maka titik-titik data yang ada dimasukkan ke dalam program untuk dilakukan proses interpolasi. Adapun algoritma pemrosesannya adalah sebagai berikut;

FLOW CHART FUNGSI UTAMA (MAIN)



Gambar 4.1 Diagram alir Fungsi utama program

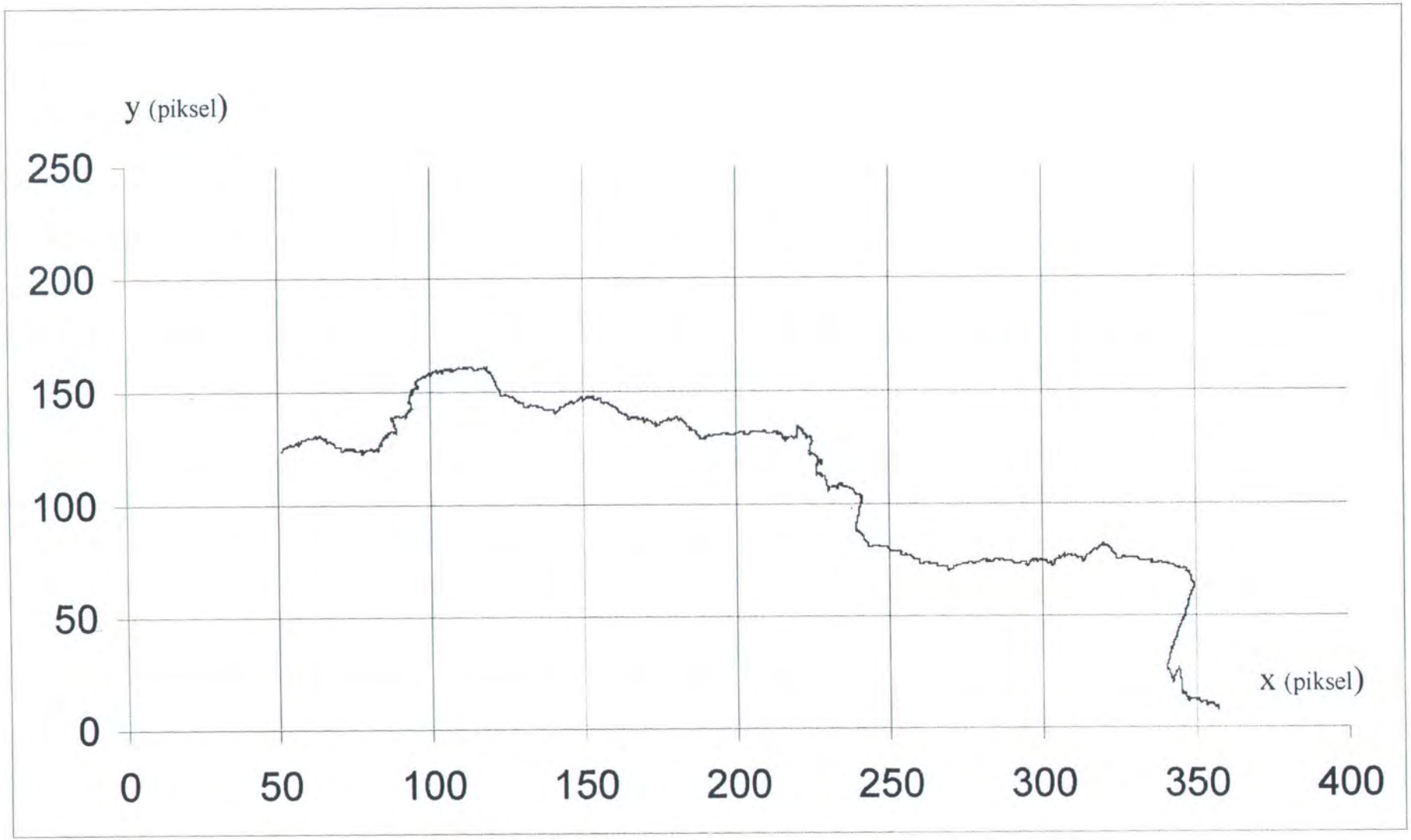
Langkah pertama program akan menginisialisasi parameter yang diprlukan dalam program, selanjutnya akan memasukkan data dengan masukan data berupa file berekstensi FIP dan dimensi fraktal yang diinginkan. Setelah data terbaca program maka akan menghitung kode *iterated function system* (IFS) yang merupakan proses iterasi dari fungsi transformasi. Proses IFS ini akan menghasilkan data numerik nilai-nilai a, c, e, f dan d yang akan digunakan untuk proses interpolasisperti terlihat pada gambar 4.2. Selanjutnya hasil numerik proses transformasi tersebut dicetak. Selengkapnya dapat dilihat pada lampiran 1

HASIL ITERASI FUNCTION SYSTEM

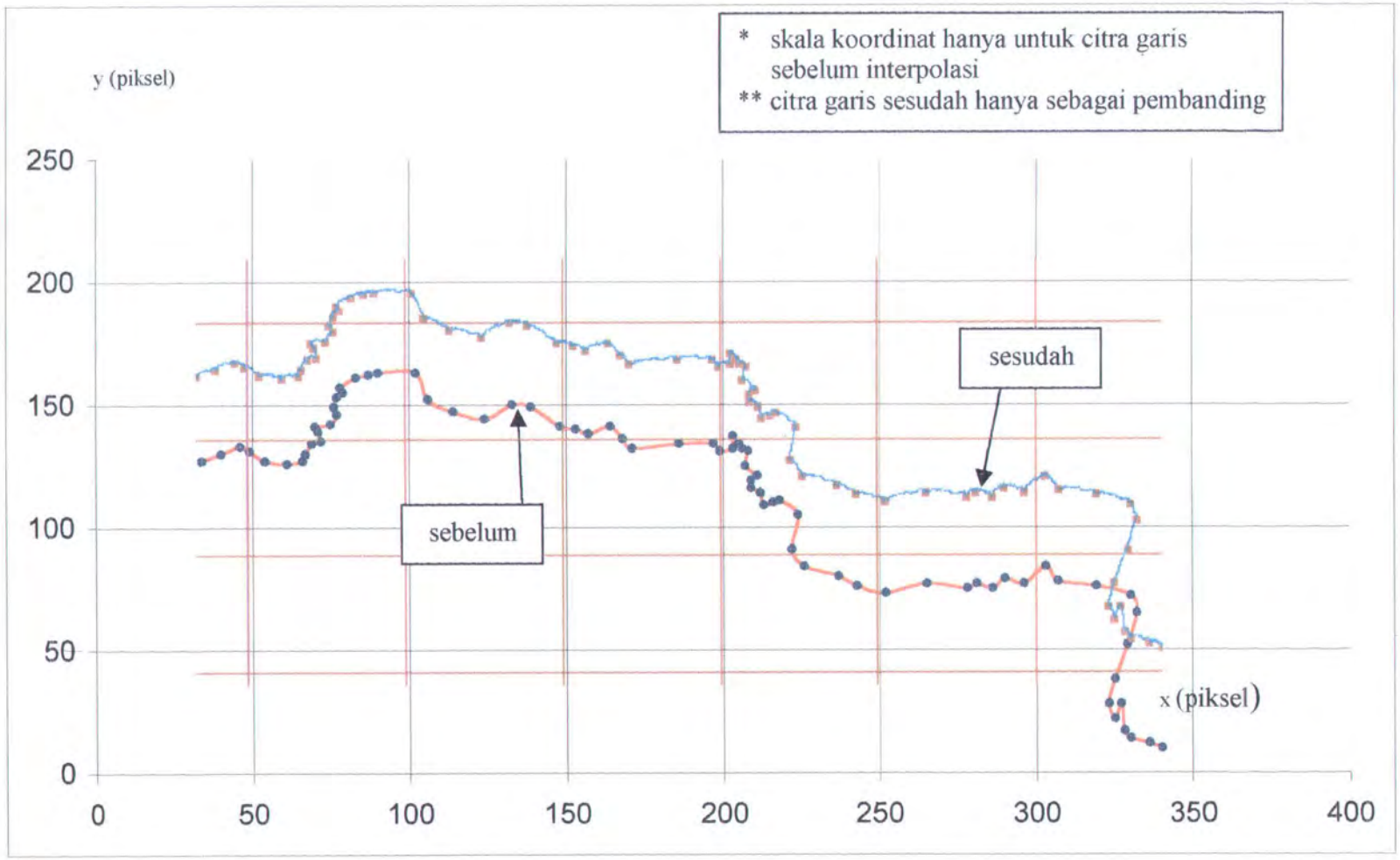
Iter	x	y	a	c	e	f	d
01	34.00	127.00	0.020	0.019	33.33	123.21	0.025
02	40.00	130.00	0.020	0.019	39.33	126.21	0.025
03	46.00	133.00	0.010	0.003	45.67	129.77	0.025
04	49.00	131.00	0.016	-0.004	48.44	127.99	0.025
05	54.00	127.00	0.023	0.006	53.22	123.65	0.025
06	61.00	126.00	0.016	0.013	60.44	122.43	0.025
07	66.00	127.00	0.003	0.019	65.89	123.21	0.025
08	67.00	130.00	0.007	0.023	66.78	126.10	0.025
09	69.00	134.00	0.010	0.013	68.67	130.43	0.025
10	72.00	135.00	-0.003	0.023	72.11	131.10	0.025

Gambar 4.2 Contoh hasil proses iterasi

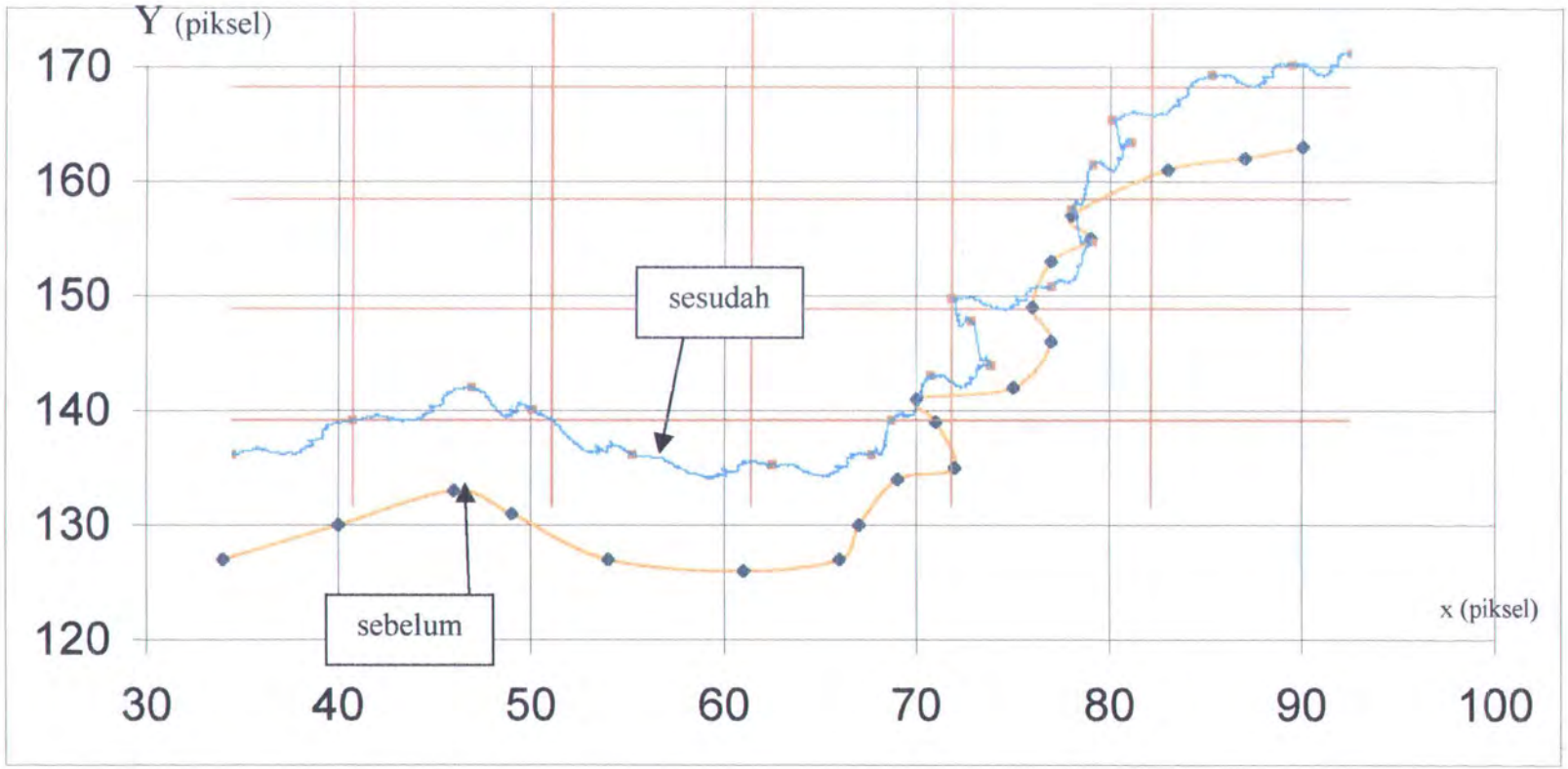
Setelah nilai-nilai transformasi diperoleh, maka program akan menjadikan nilai tersebut menjadi titik-titik data yang saling berhubungan yang kemudian divisualisasikan dalam bentuk gambar kurva (gambar 4.3). Tahap terakhir adalah proses perhitungan dimensi fraktal kurva yang tercetak pada layar sehingga diperoleh dimensi berdasarkan keluaran grafik komputer.roleh selisih dengan dimensi masukan sehingga diketahui errornya.



Gambar 4.3 Citra garis yang telah mengalami interpolasi



Gambar 4.4 perbandingan kurva dengan 75 titik data



* skala koordinat hanya untuk citra garis sebelum interpolasi
** citra garis sesudah hanya sebagai pembanding

Gamabar 4.5 perbandingan kurva dengan 21 titik data data

4.3 Analisa dan Pengujian Program

Pada saat program dijalankan, akan diminta berkas data yang akan difraktalkan beserta dimensi fraktalnya. Dimensi ini diperlukan karena akan menentukan tingkat kekasaran citra yang akan dihasilkan.

Dalam Aplikasi kasus perbaikan citra garis dipakai dimensi fraktal(D_{in})=1.14 dengan faktor penyekalaan vertikal (d_i)=0,5, sehingga diperoleh kurva perbaikan citra garis seperti terlihat pada gambar 4.4. Pada gambar tersebut terlihat hasil citra yang telah mengalami proses transformasi. Apabila kita bandingkan citra data awal (tanpa transformasi) akan terlihat perbedaan pada kontur yang terbentuk. Sekilas memang akan terlihat sama. Namun apabila gambar tersebut dilihat lebih seksama terlihat citra garis tersebut terlihat kekasarannya dibandingkan dengan citra awal sehingga konturnya tampak lebih kasar Yang dimaksud dengan lebih kasar pada hasil di sini adalah, dengan mengambil pengertian pada konsep geometri klasik yang dikenal selama ini, hanya dapat menggambarkan bentuk-bentuk segitiga, kotak, garis lengkung sedangkan pada bentuk-bentuk alam seperti awan, pohon, pegunungan, maupun garis pantai, belum dpat mendeskripsikannya (menggambarkannya). Sedangkan pada geometri fraktal, bentuk-bentuk alam tersebut dapat digambar mendekati bentuk aslinya. Berdasarkan hal inilah bahwa kurva/citra hasil interpolasi menjadi lebih kasar dibandingkan dengan bentuk sebelumnya. Hal ini akan bermanfaat untuk mengetahui perubahan garis pantai suatu kawasan. Sebagai perbandingannya dapat dilihat pada gambar 4.5 yang merupakan gabungan antara gambar 3.5 dengan 4.3. Dari gambar tersebut terlihat bahwa citra garis yang telah mengalami proses interpolasi memiliki kontur garis yang lebih kasar

dibandingakan citra garis sebelumnya. Untuk lebih jelas mengenai tingkat kekasarannya, dapat dilihat pada gambar 4.6. Citra garis ini menggunakan 21 titik data yang terbentang antara 110^0-$111,5^0$ bujur timur. Setelah citra garis mengalami proses interpolasi maka program akan menghitung dimensi dari citra garis yang ada pada layar monitor. Dimensi Fraktal output (D_{out}) sebagai berikut :

Tabel 4.1 Perhitungan dimensi fraktal untuk 75 titik data

Perhitungan ke-	Dimensi Fraktal
1	1.128727
2	1.121995
3	1.091930

Tabel 4.2 Perhitungan dimensi fraktal untuk 21 titik data

Perhitungan ke-	Dimensi Fraktal
1	1.082567
2	1.058802
3	1.097657

Perhitungan dimensi dilakukan berulang karena pada proses perhitungan dimensi citra garis program masih melakukan proses iterasi sehingga perlu dilakukan perhitungan sebanyak tiga kali.Dari ketiga hasil perhitungan diperoleh rata-rata dimensi keluaran (D_{out}). D_{out} ini akan menentukan besarnya error citra yang terbentuk berdasarkan persamaan 4.1.

$$Error = D_{in} - D_{out}$$

$$\text{Prosentase error} = \frac{error}{D_{in}} x100\ \% \qquad (4.1)$$

Tabel 4.3 Nilai error dimensi

Jumlah data	Dimensi Masukan	Dimensi Keluaran	Error (%)
75	1,14	1.114217	2.56
21	1,14	1.079676	5.29

Pada tabel 4.3 diketahui bahwa pada jumlah titik yang lebih banyak maka nilai error dapat diminimalkan/diperkecil. Prosentase error ini mengandung pengertian bahwa dalam mentransformasikan titik-titk data masukan, dengan nilai dimensi yang diinginkan, namun program (komputer) hanya mampu mengiterasikan proses yang ada berkisar pada dimensi keluaran yang dihasilkan. Hal ini dapat terjadi karena program melakukan penyederhanaan desimal terhadap nilai-nilai iterasi.

BAB V

KESIMPULAN DAN SARAN

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Telah dilakukan kajian untuk mendapatkan hasil Perbaikan citra garis pantai menggunakan metode interpolasi fraktal dengan bantuan simulasi program. Dari kajian dan hasil-hasil yang telah diperoleh, maka dapat diambil kesimpulan sebagai berikut :

1. Dengan menggunakan metode interpolasi fraktal citra garis pantai memiliki kontur yang lebih kasar dibandingkan dengan citra sebelumnya.
2. Fraktal dapat dengan mudah merepresentasikan citra garis suatu kawasan pada banyak level skala.
3. Fraktal sering memberikan metode yang lebih sederhana dalam perekaman citra dan data kompleks dibanding vector linier.
4. Secara otomatis dapat menemukan suatu kurva fraktal yang selaras dengan suatu himpunan data (titik data, jumlah titik data, skala, faktor skala vertical dan dimensi fraktal)
5. Dengan dimensi masukan sebesar 1.14, hasil keluaran (output) program diperoleh estimasi dimensi fraktal 1.114217 sehingga diperoleh nisbah kesalahan (error) sekitar 2,56% untuk 75 titik data dan untuk 21 titik data memiliki keluaran dimensi sebesar 1,079676 dengan error 5,29%.
6. Semakin banyak jumlah titik data maka erronya akan semakin kecil
7. Fraktal dapat digunakan untuk membangun model yang berguna dari sistem yang tidak dapat dilakukan dengan persamaan linier.

5.2 Saran

Agar didapatkan hasil yang lebih maksimal, pada Pengembangan fraktal untuk perbaikan citra garis pantai dengan metode Interpolasi Fraktalyang secara prinsip dapat diterapkan pada objek lain untuk tujuan-tujuan tertentu, maka disarankan antara lain sebagai berikut :

1. Dimungkinkannya metode ini dapat dipakai untuk menganalisis perbaikan citra pada suatu objek lain untuk mendapatkan hasil yang lebih alamiah.
2. Dalam pengembangannya dimungkinkannya metode ini dipakai untuk bisa mengetahui/mencari perbedaan garis pantai di suatu kawasan tertentu.
3. Dimungkinkan untuk membuat Simulasi perangkat komputer yang didukung oleh software dengan menggunakan bahasa pemrograman yang lain.

MILIK PERPUS. ITS

DAFTAR PUSTAKA

DAFTAR PUSTAKA

1. Anwar, C, Studi Pengaruh Abrasi dan Sedimentasi di Pantai Kenjeran Surabaya, Laporan Penelitian, FTSP-ITS, Surabaya, 1991
2. Avnir, D, *The Fractal Approach to Heterogeneous Chemistry*, John Wiley and Sons, New York, 1989
3. Falconer, K., *Fractal Geometry, Mathematical Foundations and Application*, John Wiley and Sons, New York, 1990
4. Mikrodata, Vol 6 Seri 11, Penerbit Elex Media Komputindo, Jakarta, 1996
5. Oliver, D., *Fractal Vision: Put Fractal to Work for You,* Sams Publishing, 1992
6. Santoso, I.P., *Pemrograman Fraktal Resolusi Tinggi untuk Kartu Trident VGA,* Penerbit Andi Offset, Yoyakarta, 1997
7. Santoso, I.P, *Grafiika Komputer dan Antarmuka Grafis*, Penerbit Andi Offset, Yoyakarta, 1994
8. Steven, R.T., *Fractal Programming and Ray Tracing with C++*, M&T Publishing Inc, California, 1991
9. Steven, R.T., *Fractal Programming in C*, M&T Publishing Inc, California, 1989
10. Susanta, B et.al., Fraktal Sebagai Grafik Fungsi, Laporan Penelitian Jurusan Matematika FMIPA-UGM, Yogyakarta, 1994

LAMPIRAN

LAMPIRAN 1. Hasil Numerik Interpolasi Fraktal

DATATA.FIP 75 points, skala grafik 50.00 x 50.00 Dimensi 1.140000

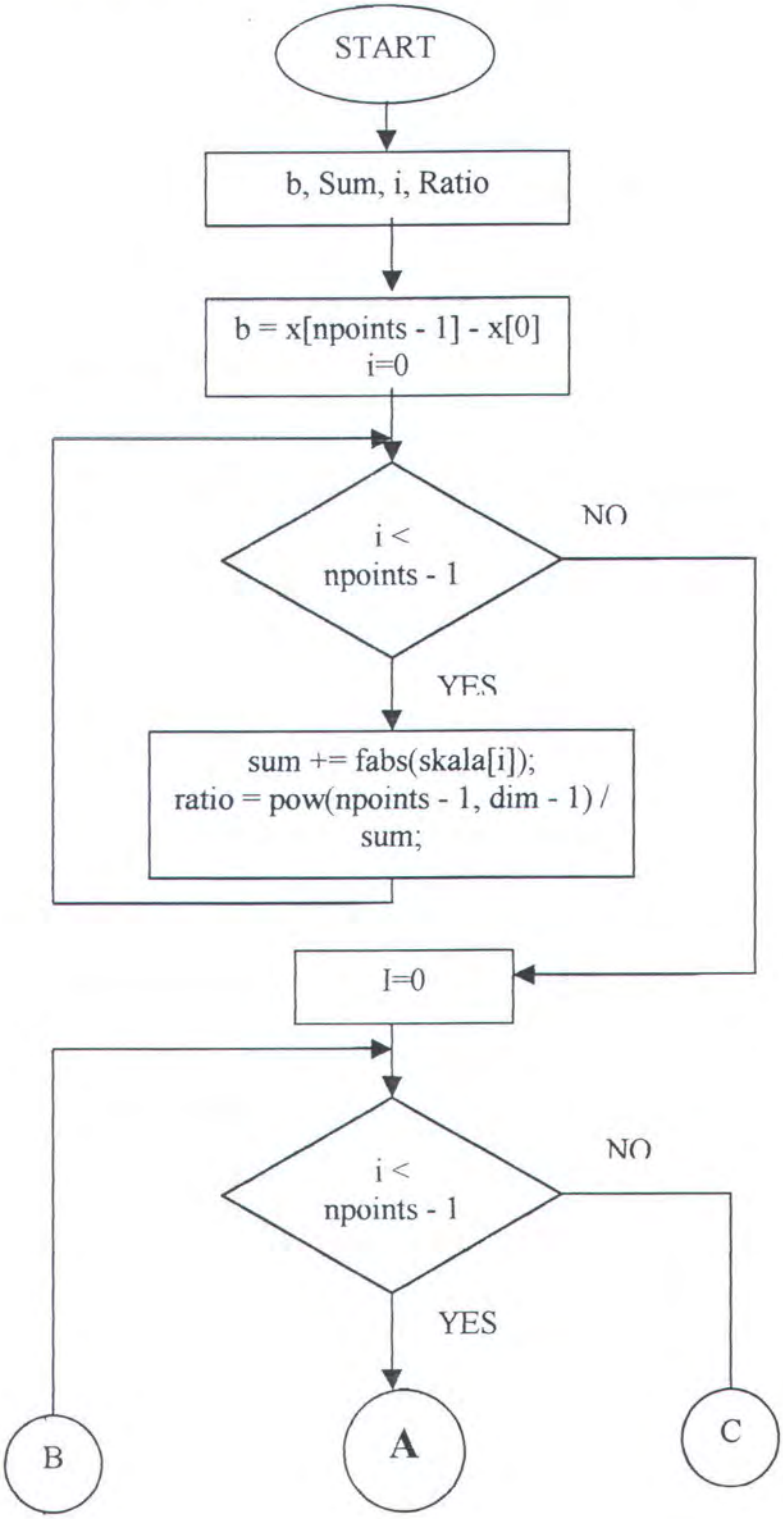
HASIL ITERASI FUNCTION SYSTEM

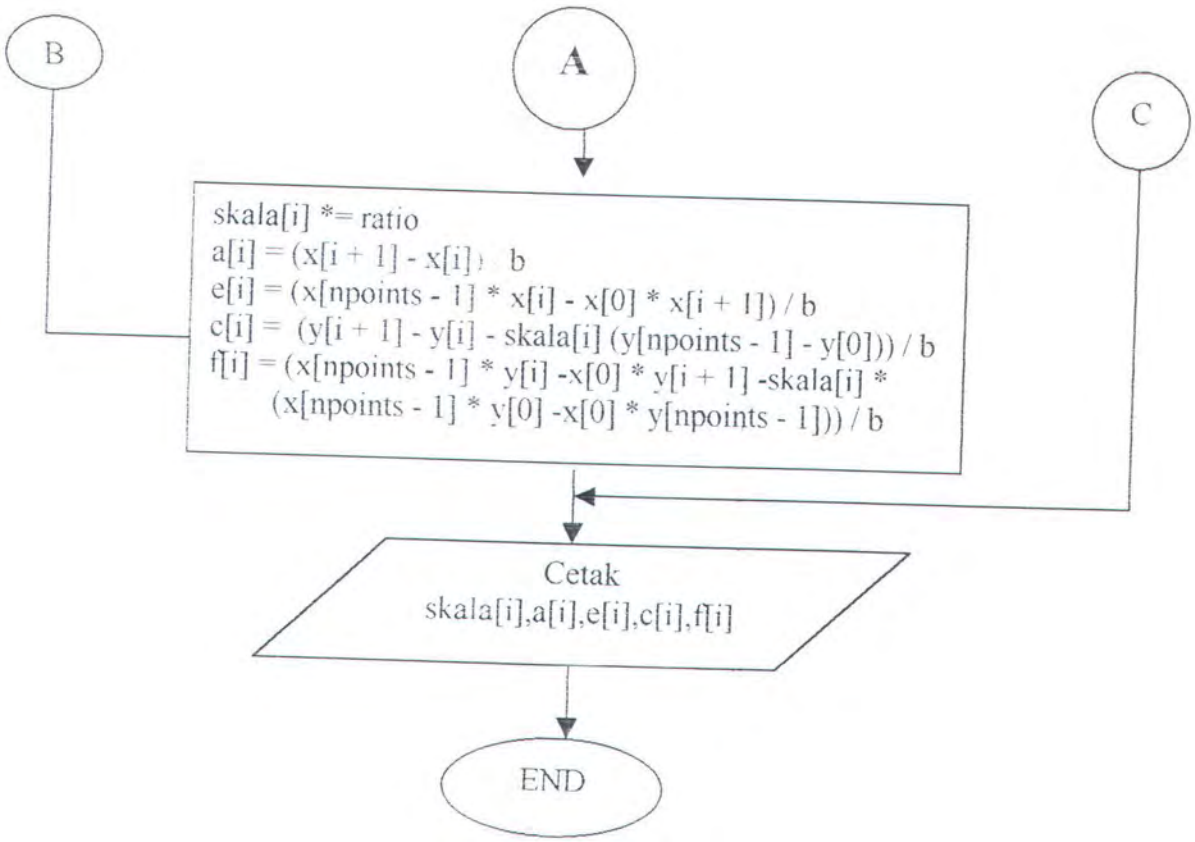
Iter	x	y	a	c	e	f	d
01	34.00	127.00	0.020	0.019	33.33	123.21	0.025
02	40.00	130.00	0.020	0.019	39.33	126.21	0.025
03	46.00	133.00	0.010	0.003	45.67	129.77	0.025
04	49.00	131.00	0.016	-0.004	48.44	127.99	0.025
05	54.00	127.00	0.023	0.006	53.22	123.65	0.025
06	61.00	126.00	0.016	0.013	60.44	122.43	0.025
07	66.00	127.00	0.003	0.019	65.09	123.21	0.025
08	67.00	130.00	0.007	0.023	66.78	126.10	0.025
09	69.00	134.00	0.010	0.013	68.67	130.43	0.025
10	72.00	135.00	-0.003	0.023	72.11	131.10	0.025
11	71.00	139.00	-0.003	0.015	71.11	135.32	0.025
12	70.00	141.00	0.016	0.013	69.44	137.43	0.025
13	75.00	142.00	0.007	0.023	74.78	138.10	0.025
14	77.00	146.00	-0.003	0.019	77.11	142.21	0.025
15	76.00	149.00	0.003	0.023	75.89	145.10	0.025
16	77.00	153.00	0.007	0.016	76.78	149.32	0.025
17	79.00	155.00	-0.003	0.016	79.11	151.32	0.025
18	78.00	157.00	0.015	0.023	77.44	153.10	0.025
19	83.00	161.00	0.013	0.013	82.56	157.43	0.025
20	87.00	162.00	0.010	0.023	86.67	158.43	0.025
21	96.00	163.00	0.020	0.003	88.67	159.54	0.025
22	102.00	163.00	0.013	0.007	101.56	160.77	0.025
23	106.00	152.00	0.026	-0.007	105.11	149.10	0.025
24	114.00	147.00	0.033	0.000	112.89	143.88	0.025
25	124.00	141.00	0.020	0.021	123.00	139.00	0.025
26	133.00	150.00	0.020	0.026	132.33	146.65	0.025
27	139.00	149.00	0.029	0.017	138.00	146.43	0.025
28	140.00	143.00	0.010	0.000	140.00	137.65	0.025
29	153.00	140.00	0.013	0.003	152.56	136.77	0.025
30	157.00	130.00	0.023	0.017	156.22	134.21	0.025
31	164.00	131.00	0.013	0.007	163.44	130.10	0.025
32	168.00	135.00	0.010	-0.003	167.67	132.99	0.025
33	171.00	132.00	0.043	0.010	169.11	128.32	0.025
34	186.00	134.00	0.030	0.004	184.00	130.54	0.025
35	197.00	134.00	0.007	0.000	196.00	130.00	0.025
36	199.00	135.00	0.013	0.013	198.56	127.43	0.025
37	203.00	132.00	0.000	0.000	203.00	129.00	0.025
38	203.00	132.00	0.007	0.000	[illegible]	131.00	0.025
39	205.00	134.00	0.000	0.000	[illegible]	134.00	0.025
40	206.00	132.00	0.000	0.000	[illegible]	132.00	0.025

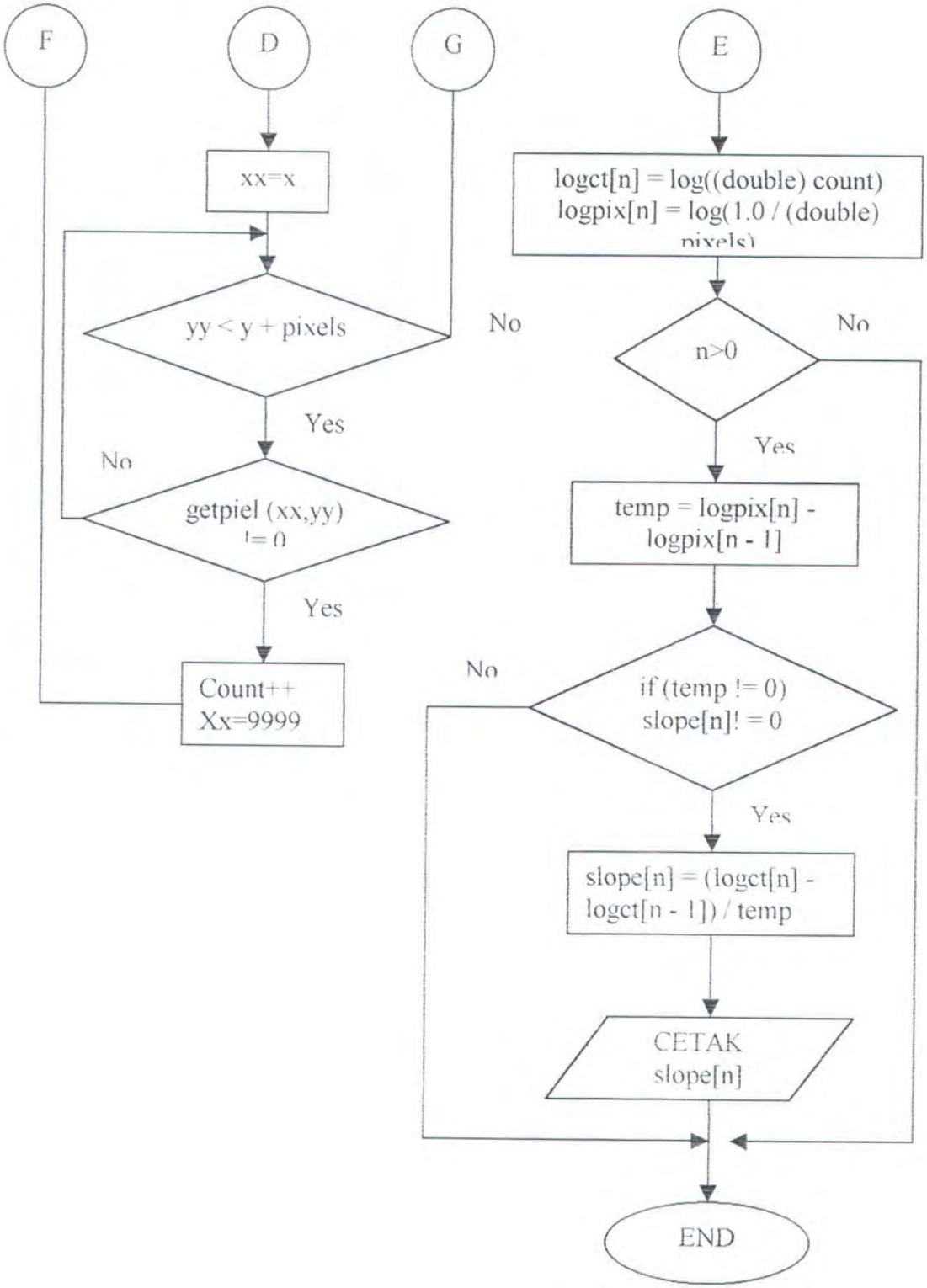
41	208.00	131.00	-0.003	-0.010	208.11	128.21	0.025
42	207.00	125.00	0.013	-0.004	206.56	121.99	0.025
43	211.00	121.00	-0.007	0.003	211.22	117.77	0.025
44	209.00	119.00	0.000	-0.000	209.00	115.00	0.025
45	209.00	116.00	0.010	0.003	208.67	112.77	0.025
46	212.00	114.00	0.003	-0.007	211.89	111.10	0.025
47	213.00	109.00	0.010	0.013	212.67	105.43	0.025
48	216.00	110.00	0.007	0.013	215.78	106.43	0.025
49	218.00	111.00	0.020	-0.010	217.33	108.21	0.025
50	224.00	105.00	-0.007	-0.036	224.22	103.10	0.025
51	222.00	91.00	0.013	-0.013	221.56	88.32	0.025
52	226.00	84.00	0.036	-0.004	224.78	80.99	0.025
53	237.00	80.00	0.020	-0.004	236.33	76.99	0.025
54	243.00	76.00	0.029	-0.000	242.00	72.00	0.025
55	252.00	73.00	0.042	0.023	250.56	69.10	0.025
56	265.00	77.00	0.042	0.003	263.56	73.77	0.025
57	278.00	75.00	0.010	0.016	277.67	71.32	0.025
58	281.00	77.00	0.016	0.003	280.44	73.77	0.025
59	286.00	75.00	0.013	0.023	285.56	71.10	0.025
60	290.00	79.00	0.020	0.003	289.33	75.77	0.025
61	296.00	77.00	0.023	0.032	295.22	72.77	0.025
62	303.00	84.00	0.013	-0.010	302.56	81.21	0.025
63	307.00	78.00	0.039	0.003	305.67	74.77	0.025
64	319.00	76.00	0.036	-0.004	317.78	72.99	0.025
65	330.00	72.00	0.007	-0.013	329.78	69.32	0.025
66	332.00	65.00	-0.010	-0.033	332.33	62.99	0.025
67	329.00	52.00	-0.013	-0.036	329.44	50.10	0.025
68	325.00	38.00	-0.007	-0.023	325.22	35.65	0.025
69	323.00	28.00	0.007	-0.010	322.78	25.21	0.025
70	325.00	22.00	0.007	0.029	324.70	17.00	0.025
71	327.00	28.00	0.003	-0.027	326.89	25.77	0.025
72	328.00	17.00	0.007	-0.000	327.78	13.00	0.025
73	330.00	14.00	0.020	0.003	329.33	10.77	0.025
74	336.00	12.00	0.013	0.003	335.56	8.77	0.025
75	340.00	10.00	0.000	0.000	0.00	0.00	0.500

LAMPIRAN 2 Diagram alir proses dari Diagram alir utama

FLOW CHART ITERATED FUNCTION SYSTEM




B
A
C
skala[i] *= ratio
a[i] = (x[i + 1] - x[i]) / b
e[i] = (x[npoints - 1] * x[i] - x[0] * x[i + 1]) / b
c[i] = (y[i + 1] - y[i] - skala[i] (y[npoints - 1] - y[0])) / b
f[i] = (x[npoints - 1] * y[i] -x[0] * y[i + 1] -skala[i] *
(x[npoints - 1] * y[0] -x[0] * y[npoints - 1])) / b
Cetak
skala[i],a[i],e[i],c[i],f[i]
END


F
D
G
E
xx=x
yy < y + pixels
No
Yes
No
getpiel (xx,yy)
!= 0
Yes
Count++
Xx=9999
logct[n] = log((double) count)
logpix[n] = log(1.0 / (double)
pixels)
n>0
No
Yes
temp = logpix[n] -
logpix[n - 1]
No
if (temp != 0)
slope[n]! = 0
Yes
slope[n] = (logct[n] -
logct[n - 1]) / temp
CETAK
slope[n]
END

LAMPIRAN 3. Listing Program

```
/* INTERPOLASI.C
 *
 * Program ini akan mencari kurva fraktal
 * berdasarkan sekumpulan data berkas ASCII
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

#define MAIN
#include "kepala.h"


#define drawclr 1          /* warna untuk membuat kurva */
#define gridclr 4          /* warna untuk membuat kisi */
#define pointclr 4         /* warna untuk mempuat titik */
#define ESC 27
#define NPOINTS 100        /* jumlah maksimum titik-titik
data */

float x[NPOINTS], y[NPOINTS],/* data dari berkas masukan */
     skala[NPOINTS],          /* faktor skala y */
     a[NPOINTS - 1], b, c[NPOINTS - 1],      /* kode IFS */
     e[NPOINTS - 1], f[NPOINTS - 1];
void hitungdimensi(void);
void main()
{   int vidmode, i, j,k;

    float x1, y1, x2, y2,     /* untuk menampilkan kurva */
       skala_x, skala_y,      * penskalaan layar tampilan */
       maxx, maxy, minx, miny,  /* nilai max dan min x dan
                                   y */
       dim, sum, ratio,      /* untuk menghitung dimensi */
       xgrid, ygrid;

    int xoffset, yoffset, yposisi,     /* posisi layar */
     npoints;            /* jumlah data dalam berkas */
    FILE *diskfile;
    char namafile[20];
    vidmode = detectmode();

    {   clrscr();
```

```
printf("\n\n");
printf("\t\tÉÍÍÍ
```

```
        goto input;
    }

if ((dim > 2.0) || (dim < 1.0))
 {
        printf("Dimensi fraktal harus bernilai antara 1
        dan 2 (misal 1.2).");
        goto dimensi;
 }

fscanf(diskfile, "%d", &npoints);     /* membaca data */
fscanf(diskfile, "%f %f", &xgrid, &ygrid);
for (i = 0; (i < npoints) && (i < NPOINTS); i++)
 fscanf(diskfile, "%f %f %f", x + i, y + i, skala + i);
 fclose(diskfile);
if (setvidmode(vidmode,1) != 1)
  {
        printf("\nTidak bisa masuk ke mode grafik.\n");
        exit(0);
  }

/* Menormalkan data sehingga layarnya cukup */

yposisi = scrny / 20;
maxx = x[0]; minx = x[0];
maxy = y[0]; miny = y[0];
for (i = 1; i < npoints; i++)
  {
        if (x[i] > maxx) maxx = x[i];
        if (y[i] > maxy) maxy = y[i];
        if (x[i] < minx) minx = x[i];
        if (y[i] < miny) miny = y[i];
  }
skala_x = scrnx / (maxx - minx);
skala_y = (scrny / (maxy - miny)) / 2;
xoffset = - (int) (minx * skala_x);
yoffset = scrny - yposisi + (int) (miny * skala_y);

/* Menghitung transformasi skew
   dan menjumlah faktor skala */

b = x[npoints - 1] - x[0];
sum = 0.0;
for (i = 0; i < npoints - 1; i++) sum -=fabs(skala[i]);
ratio = pow(npoints - 1, dim - 1) / sum;
```

```
    for (i = 0; i < npoints - 1; i++)  /* menghitung kode
                                          IFS */
      {
          skala[i] =skala[i]* ratio;
          a[i] = (x[i + 1] - x[i]) / b;
          e[i] = (x[npoints - 1] * x[i] - x[0] * x[i + 1])
          / b;
          c[i] = (y[i + 1] - y[i] - skala[i] *
          (y[npoints - 1] - y[0])) / b;
          f[i] = (x[npoints - 1] * y[i] -
          x[0] * y[i + 1] -
          skala[i] *
          (x[npoints - 1] * y[0] -
          x[0] * y[npoints - 1])) / b;
      }
```

/* mencetak data mentah dan kodenya */

```
    printf("%s  %d points, "
        "skala grafik = %6.2f x %6.2f  Dimensi = %f",
        namafile, npoints, xgrid, ygrid, dim);
    printf("\n\n\n");
    printf("\t\t   HASIL ITERASI FUNCTION SYSTEM ");
    printf("\n\n\n");
    printf(" Iter      x        y       a       c       e
f      d \n");

printf("ÍÍÍ
```

```
/* Membuat garis kisi */

setcolor(gridclr);
xgrid = xgrid * skala_x;  ygrid *=skala_y;
for (x1 = (float) xoffset; x1 < scrnx; x1 += xgrid)
      drawline((int) x1, scrny / 2 - yposisi * 2,
       (int) x1, scrny - 1);
for (y1 = (float) yoffset;
       y1 >= (float) scrny / 2 - yposisi * 2;
       y1 -= ygrid)
       drawline(0, (int) y1, scrnx - 1, (int) y1);


/* Membuat titik data */

for (i = 0; i < npoints; i++)
{
      x1 = x[i] * skala_x + xoffset;
      y1 = yoffset - y[i] * skala_y;
      fillrect((int) x1 - 2, (int) y1 - 2,
      (int) x1 + 2, (int) y1 + 2, pointclr);
}


/* menempatkan kurva pada layar */

x1 = (float) xoffset;
y1 = (float) yoffset;
while(!kbhit())
{
      j = rand() % (npoints - 1);
      x2 = a[j] * x1 + e[j];
      y2 = c[j] * x1 + skala[j] * y1 + f[j];
      x1 = x2, y1 = y2;
      putpixel((int) (x1 * skala_x + xoffset),
      (int) (yoffset - y1 * skala_y), drawclr);
      delay(1);
}
getscreen();
hitungdimensi();
closedown();
}
```

```
/* fungsi untuk menghitung dimensi fraktal */

void hitungdimensi(void)
{   int x, y, pixels, xx, yy, n;
    char a;
    long count;

    /* larik untuk menyimpan perkiraan logaritmik */

    double logct[9], logpix[9], kurva[9], temp;

    /* dimulai dengan kotak berukuran 2 piksel */

    for (pixels = 2, n = 0;
      pixels <= 8;
      pixels = pixels += 2, n++)
    {   count = 0;       /* penghitungan dimulai dari nol */

     putscreen();


     for (x = 0; x < scrnx - pixels; x += pixels)
         for (y = 0; y < scrny - pixels; y += pixels)
         {
          /* memeriksa setiap kotak sampai ditemukan
          piksel yang menyala atau melihat ke setiap
          piksel dalam kotak */

          for (xx = x; xx < x + pixels; xx++)
              for (yy = y; yy < y + pixels; yy++)
              {
                    if (getpixel(xx, yy) != 0)
                    {
                         count++;   /* ada piksel menyala
                    */
                         xx = 9999;
                         break;
                    }
              }

              if (kbhit())
              {
                    a = getch();
                    if (a == ESC) return;
              }
```

```
        }

    /* untuk memperkirakan dimensi fraktal dari hitungan
kotak, maka pencacah diubah itu dalam skala logaritmis. */

        logct[n] = log((double) count);
        logpix[n] = log(1.0 / (double) pixels);

        if (n > 0)
        {
            temp = logpix[n] - logpix[n - 1];
            if (temp == 0) kurva[n] = 0;
            else kurva[n] = (logct[n] - logct[n - 1]) /
        temp;
        }
    }
    settextmode();  /* selesai. kembali ke mode teks. */
    printf("\t\tPERKIRAKAN DIMENSI FRAKTALNYA");
    temp = 0.0;
    for (x = 1; x < n; x++)

    {
        printf("\n\nUkuran kotak %d memberikan dimensi
        fraktal: %f",x * 2, kurva[x]);
        temp += kurva[x];
    }
    printf("\n\nDimensi fraktal rata-rata : %f",temp / 3);
    printf("\n\nTekan sembarang tombol untuk meneruskan.");
    getch();
}
```