



**PENERAPAN *STOCHASTIC L-SYSTEMS* PADA PEMODELAN  
PERTUMBUHAN BATANG TANAMAN**

**SKRIPSI**

Oleh

**Chandra Hadi Iswanto  
NIM 061810101083**

**JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS JEMBER  
2011**



**PENERAPAN *STOCHASTIC L-SYSTEMS* PADA PEMODELAN  
PERTUMBUHAN BATANG TANAMAN**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat  
untuk menyelesaikan Program Studi Matematika (S1)  
dan mencapai gelar Sarjana Sains

Oleh

**Chandra Hadi Iswanto**  
**NIM 061810101083**

**JURUSAN MATEMATIKA**  
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**  
**UNIVERSITAS JEMBER**  
**2011**

## **PERSEMBAHAN**

Skripsi ini saya persembahkan untuk:

1. Ayahanda Budi Aman, Ibunda Santik Purjiansih, dan Ibunda Juwariyati yang telah melahirkan dan membesarkan saya dengan kasih sayang, perhatian, dan pengorbanan yang tiada henti, serta doa yang tak pernah putus;
2. Kakak dan adik tersayang, Alifah, Agus, Rindang, dan Firman yang telah memberi segala pengorbanan, dukungan, perhatian, dan doa;
3. Guru-guru saya sejak SD sampai Perguruan Tinggi yang telah memberikan ilmu dan membimbing saya dengan penuh kesabaran;
4. Almater Jurusan Matematika Fakultas MIPA Universitas Jember.

## MOTTO

Kami jelaskan yang demikian itu agar kamu tidak terlalu berduka-cita atas apa-apa yang luput dari kamu, dan tidak juga terlalu gembira atas apa-apa yang diberikan-Nya kepadamu.  
(*Terjemahan Surat Al-Hadid [57] : 23*)<sup>\*)</sup>

---

\*) Khadim Al Haramain Asy Syarifain & Abdullah bin Abdul Aziz Ali Sa'ud. 1971. *Al Qur'an dan Terjemahnya*. Madinah : Mujamma' Al Malik Fahd Li Thiba'at Al Mush-haf Asy Syerif

## PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Chandra Hadi Iswanto

NIM : 061810101083

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Penerapan *Stochastic L-Systems* pada Pemodelan Pertumbuhan Batang Tanaman" adalah benar-benar hasil karya sendiri, kecuali kutipan yang telah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan 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 ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 7 Nopember 2011

Yang menyatakan,

Chandra Hadi Iswanto  
NIM 061810101083

## **SKRIPSI**

### **PENERAPAN *STOCHASTIC L-SYSTEMS* PADA PEMODELAN PERTUMBUHAN BATANG TANAMAN**

Oleh

Chandra Hadi Iswanto  
NIM 061810101083

Pembimbing

Dosen Pembimbing Utama : Drs. Moh. Hasan, M.Sc., Ph.D.

Dosen Pembimbing Anggota : Drs. Rusli Hidayat, M.Sc.

## PENGESAHAN

Skripsi berjudul "Penerapan *Stochastic L-Systems* pada Pemodelan Pertumbuhan Batang Tanaman" telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Jember

Tim Penguji:

Ketua,

Sekretaris,

Drs. Moh. Hasan, M.Sc., Ph.D.  
NIP 196404041988021001

Drs. Rusli Hidayat, M.Sc.  
NIP 196610121993031001

Penguji I,

Penguji II,

Kristiana Wijaya, SSi, MSi  
NIP 197408132000032004

Bagus Juliyanto, SSi  
NIP 198007022003121001

Mengesahkan

Dekan,

Prof. Drs. Kusno, DEA, Ph.D.  
NIP 196101081986021001

## RINGKASAN

**Penerapan *Stochastic L-Systems* pada Pemodelan Pertumbuhan Batang Tanaman;** Chandra Hadi Iswanto, 061810101083; 2011: 51 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

*Lindenmayer systems* atau disebut *L-systems* adalah sebuah sistem penulisan kembali, yaitu teknik membangun objek yang kompleks dari pengulangan bagian objek yang sederhana menggunakan suatu aturan penulisan kembali atau produksi. Penulisan kembali tersebut dilakukan secara rekursif. (J. Mishra dan S. Mishra). Sudah banyak peneliti yang telah menggunakan teori *L-systems* ini baik dalam dimensi dua maupun dalam dimensi tiga untuk memodelkan suatu objek tanaman, namun tidak menyertakan *stochastic L-systems* dalam penelitiannya. Untuk itu peneliti berusaha mengangkat *stochastic L-systems* ini dalam membuat model pertumbuhan batang tanaman dimensi tiga.

Penelitian dilakukan dalam empat tahap. Pertama, pengambilan data penelitian berupa foto tanaman dari dua sisi karena model akan divisualisasikan dalam dimensi tiga. Kedua, menafsirkan grafis secara manual yang dimulai dari tahap identifikasi dan penentuan komponen utama *L-systems* seperti huruf, aksioma dan aturan produksi. Ketiga, tahap simulasi dan visualisasi model. Keempat, menentukan nilai probabilitas yang dilakukan dengan cara pengamatan objek secara langsung. Setelah melakukan empat tahap tersebut akan didapatkan model *L-systems* untuk masing-masing tanaman. Namun dengan model *L-systems* saja dirasa tidak cukup untuk menjelaskan ke-*stochastic*-an. Untuk itu perlu untuk memvisualisasikan beberapa model tanaman tersebut. Dengan menggunakan software Microsoft Visual C++ 6.0 didapatkan hasil visualisasi *stochastic L-systems* dalam pertumbuhan batang tanaman dalam dimensi tiga yang menyertakan ketebalan batang. Dari hasil model yang telah dibuat dapat diketahui bahwa dengan menggunakan *stochastic L-systems* dapat

dihasilkan lebih dari satu model pertumbuhan batang tanaman yang bervariasi dengan komponen *L-systems* yang sama. Sehingga dengan *stochastic L-systems* ini dapat memenuhi pernyataan bahwa tidak ada tanaman yang tumbuh dengan proses percabangan yang sama. Selain itu dengan *stochastic L-systems* dapat dilakukan prediksi beberapa tanaman yang sejenis dengan hanya memanfaatkan satu komponen *L-systems* saja.

## PRAKATA

Puji syukur ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Penerapan *Stochastic L-Systems* pada Pemodelan Pertumbuhan Batang Tanaman”. Skripsi ini disusun untuk memenuhi salah satu syarat untuk menyelesaikan Program Studi Matematika (S1) dan mencapai gelar Sarjana Sains.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Drs. Moh. Hasan, M.Sc., Ph.D., selaku Dosen Pembimbing Utama dan Drs. Rusli Hidayat, M.Sc., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
2. Kristiana Wijaya S.Si., M.Si., dan Bagus Juliyanto, S.Si., selaku Dosen Penguji yang telah memberikan kritik dan saran demi kesempurnaan skripsi ini;
3. teman-teman mahasiswa Jurusan Matematika, khususnya angkatan 2006 yang telah memberi bantuan, dukungan kepada penulis;
4. teman-teman kos Jawa VIII, Arif, Agus, Yayan, Uut, Fian, Anik, Anjar, Jefri, Endang, Jajuli, Farizal, Dio, Hilmi, Wantik, dan bokir yang telah menemani, membantu dan memberi dukungan dalam mengerjakan skripsi ini;
5. semua pihak yang tidak dapat disebutkan satu per satu.

Penulis menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap semoga skripsi ini dapat bermanfaat.

Jember, Nopember 2011

Penulis

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	i
<b>HALAMAN PERSEMBAHAN</b> .....	ii
<b>HALAMAN MOTTO</b> .....	iii
<b>HALAMAN PERNYATAAN</b> .....	iv
<b>HALAMAN PEMBIMBINGAN</b> .....	v
<b>HALAMAN PENGESAHAN</b> .....	vi
<b>HALAMAN RINGKASAN</b> .....	vii
<b>PRAKATA</b> .....	ix
<b>DAFTAR ISI</b> .....	x
<b>DAFTAR TABEL</b> .....	xii
<b>DAFTAR GAMBAR</b> .....	xiii
<b>BAB 1. PENDAHULUAN</b>	
<b>1.1 Latar Belakang</b> .....	1
<b>1.2 Rumusan Masalah</b> .....	2
<b>1.3 Tujuan</b> .....	2
<b>1.3 Manfaat</b> .....	3
<b>BAB 2. TINJAUAN PUSTAKA</b>	
<b>2.1 Definisi Dasar <i>Lindenmayer Systems (L-Systems)</i></b> .....	4
<b>2.2 Jenis-jenis <i>L-Systems</i></b> .....	6
<b>2.3 Penafsiran Grafis pada <i>L-Systems</i></b> .....	9
<b>2.4 Percabangan pada <i>L-Systems</i></b> .....	11
<b>2.5 <i>L-Systems</i> Dimensi Tiga</b> .....	12
<b>2.6 Parametrik <i>L-Systems</i></b> .....	13

<b>BAB 3. METODE PENELITIAN</b>	
<b>3.1 Data Penelitian</b> .....	15
<b>3.2 Langkah-langkah Penelitian</b> .....	16
<b>BAB 4. HASIL DAN PEMBAHASAN</b>	
<b>4.1 Membangunn Penafsiran Grafis <i>L-System</i> Secara Manual</b> .....	20
4.1.1 Identifikasi Komponen Utama <i>L-Systems</i> .....	20
4.1.2 Penamaan Titik .....	23
4.1.3 Parametrik <i>L-Systems</i> .....	25
<b>4.2 Simulasi dan Visualisasi</b> .....	32
<b>4.3 Hasil Pemodelan</b> .....	35
<b>4.4 Pembahasan</b> .....	48
<b>BAB 5. KESIMPULAN DAN SARAN</b>	
<b>5.1 Kesimpulan</b> .....	50
<b>5.2 Saran</b> .....	50
<b>DAFTAR PUSTAKA</b> .....	51

## DAFTAR TABEL

Tabel 2.1 Generasi <i>L-systems</i> .....	6
Tabel 2.2 Generasi <i>stochastic L-systems</i> .....	7
Tabel 2.3 Generasi <i>context sensitive L-systems</i> .....	8
Tabel 2.4 Penafsiran grafis <i>L-systems</i> .....	10
Tabel 2.5 Generasi <i>parametric L-systems</i> .....	14
Tabel 4.1 Beberapa generasi <i>L-systems</i> .....	22
Tabel 4.2 Beberapa generasi <i>parametric L-systems</i> .....	26
Tabel 4.3 Komponen utama aturan produksi dalam program .....	30
Tabel 4.4 Generasi parametrik <i>DOL-systems</i> .....	34
Tabel 4.5 Generasi parametrik <i>stochastic DOL-systems</i> .....	35

## DAFTAR GAMBAR

Gambar 2.1 Penafsiran grafis dari <i>L-systems</i> .....	9
Gambar 2.2 Penafsiran grafis percabangan <i>L-systems</i> .....	11
Gambar 2.3 Penafsiran grafis <i>L-systems</i> pada dimensi tiga .....	12
Gambar 2.4 Visualisasi model non-parametrik <i>stochastic L-systems</i> .....	14
Gambar 3.1 Contoh pohon yang akan dimodelkan .....	15
Gambar 3.2 Skema langkah-langkah penelitian .....	18
Gambar 3.3 Ilustrasi iterasi <i>stochastic L-systems</i> .....	19
Gambar 3.4 Beberapa model <i>stochastic L-systems</i> .....	19
Gambar 4.1 Iterasi kedua <i>L-systems</i> .....	22
Gambar 4.2 Iterasi ketiga <i>L-systems</i> .....	23
Gambar 4.3 Iterasi keempat <i>L-systems</i> .....	23
Gambar 4.4 Penamaan titik pada <i>L-systems</i> .....	24
Gambar 4.5 Identifikasi ketebalan batang.....	27
Gambar 4.6 Identifikasi parametrik <i>L-systems</i> .....	27
Gambar 4.7 Skema penyisipan nilai probabilitas .....	29
Gambar 4.8 Perbandingan input nilai panjang dan ketebalan batang tanaman.....	32
Gambar 4.9 Visualisasi pertumbuhan batang tanaman.....	33
Gambar 4.10 Perbandingan pertumbuhan batang pohon Trembesi asli dengan visualisasi <i>L-systems</i> dalam dimensi tiga.....	34
Gambar 4.11 Parametrik <i>deterministic L-systems</i> dimensi tiga pada pohon Trembesi .....	37
Gambar 4.12 Parametrik <i>stochastic L-systems</i> dimensi tiga pada pohon Trembesi...	38
Gambar 4.13 Parametrik <i>deterministic L-systems</i> dimensi tiga pada pohon Jung Rabab .....	40
Gambar 4.14 Parametrik <i>stochastic L-systems</i> dimensi tiga pada pohon Jung Rabab .....	41

Gambar 4.15 Parametrik <i>deterministic L-systems</i> dimensi tiga pada pohon Cemara.	43
Gambar 4.16 Parametrik <i>stochastic L-systems</i> dimensi tiga pada pohon Cemara .....	44
Gambar 4.17 Parametrik <i>deterministic L-systems</i> dimensi tiga pada pohon Cemara.	46
Gambar 4.18 Parametrik <i>stochastic L-systems</i> dimensi tiga pada pohon Cemara .....	47

## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Pemanfaatan ilmu matematika untuk menyelesaikan suatu permasalahan dalam kehidupan sehari-hari berkembang begitu pesat. Ilmu komputasi merupakan bidang ilmu yang mempelajari tentang penyusunan model matematika dan teknik penyelesaian numerik serta penggunaan komputer untuk menganalisis dan memecahkan masalah. Ilmu ini digunakan untuk menerapkan teori matematika ke dalam aplikasi komputer. Salah satu penerapannya yaitu pemodelan morfologi tanaman dengan menggunakan *Lindenmayer systems* atau disingkat *L-systems*.

Sudah banyak peneliti yang telah menggunakan teori *L-systems* ini baik dalam dimensi dua maupun dalam dimensi tiga untuk memodelkan suatu objek tanaman. Peneliti yang menggunakan *L-systems* dimensi dua antara lain Muzammil (2006) dan Amir (2009). Sedangkan peneliti yang memodelkan tanaman dengan menggunakan *L-systems* dimensi tiga yaitu Santoso (2010) dan Swandana (2010). Muzammil (2006) memodelkan pertumbuhan batang tanaman menggunakan *DOL-systems* atau jenis *L-systems* yang tidak melihat faktor ketetangaan. Model pertumbuhan batang tanaman yang didapat hanya divisualisasikan dalam bentuk dimensi dua dan bentuk pertumbuhan batangnya hanya berupa segmen garis. Sedangkan Amir (2009) memodelkan pertumbuhan batang tanaman menggunakan *context sensitive L-systems*. Bentuk pertumbuhan dan visualisasi pertumbuhan batang tanamannya juga sama dengan penelitian yang dilakukan oleh Muzammil yaitu hanya berupa segmen garis dalam dimensi dua.

Berbeda dengan penelitian yang dilakukan oleh Muzammil dan Amir, penelitian yang dilakukan oleh Santoso merupakan pengembangan dari penelitian yang dilakukan oleh Muzammil (2006). Pengembangan yang dilakukan oleh Santoso terletak pada visualisasi model tanamannya yang dimodelkan dalam dimensi tiga. Sedangkan Swandana (2010) melakukan penelitian untuk mendapatkan model

tanaman tembakau yang mendekati kenyataan. Sedangkan jenis *L-systems* yang digunakan juga sama dengan peneliti-peneliti sebelumnya yaitu menggunakan *deterministic L-systems*.

*Deterministic L-systems* memiliki ciri khusus yaitu semua struktur dan perkembangan tanaman yang dihasilkan adalah sama atau identik (Prusinkiewicz dan Lindenmayer, 1990). Sedangkan pada kenyataannya tidak ada tanaman yang berkembang dengan cara yang sama. Maka dari itu, diperlukan suatu metode untuk mendapatkan visualisasi yang lebih realistis dalam struktur tanaman yang dihasilkan, agar tidak terjadi kemonotonan dalam pemodelan tanaman. Metode yang dapat digunakan untuk menyelesaikan masalah ini yaitu *stochastic L-systems*. *Stochastic L-systems* merupakan salah satu jenis *L-systems* yang menggunakan lebih dari satu aturan produksi untuk satu simbol tertentu. Karena terdapat satu simbol yang dipetakan ke lebih dari satu aturan produksi, maka penggunaannya memanfaatkan faktor probabilitas. Hal ini berarti bahwa dengan menggunakan komponen *L-systems* yang sama dapat menghasilkan model tanaman yang bervariasi.

## 1.2 Perumusan Masalah

Permasalahan yang akan dibahas dalam skripsi ini adalah bagaimana memodelkan pertumbuhan batang tanaman menggunakan *stochastic L-systems* dalam dimensi tiga. Untuk pertumbuhan batang tanamannya, pemodelan ini mengikutsertakan ukuran ketebalan batang.

## 1.3 Tujuan

Tujuan penelitian skripsi ini adalah untuk membuat model pertumbuhan batang tanaman dimensi tiga menggunakan *stochastic L-systems* dan memvisualisasikan simulasi pertumbuhannya.

#### **1.4 Manfaat**

Manfaat yang diperoleh dari penulisan skripsi ini adalah mendapatkan visualisasi pertumbuhan batang tanaman yang mendekati realita dengan menggunakan metode *stochastic L-systems* dalam dimensi tiga. Sehingga nantinya dapat digunakan untuk memprediksi beberapa model pertumbuhan tanaman dengan komponen *L-system* yang sama. Selain itu dengan memanfaatkan *software* dalam visualisasinya diharapkan dapat mempermudah dalam melakukan simulasi pertumbuhan batang tanaman.

## BAB 2 TINJAUAN PUSTAKA

### 2.1 Definisi Dasar *Lindenmayer Systems (L-Systems)*

*Lindenmayer systems* atau sering disebut *L-systems* adalah teknik penulisan kembali yang dilakukan secara berulang-ulang. Dimana ide penulisan kembali ini pada dasarnya digunakan untuk membangun suatu objek kompleks dari suatu objek sederhana. Membangun objek kompleks ini dengan cara mengganti secara bergantian bagian-bagian dari objek sederhana menggunakan seperangkat aturan penulisan kembali atau produksi (Prusinkiewicz dan Lindenmayer, 1990). Dalam konteks sederhana, *L-systems* adalah aturan penulisan kembali yang digunakan untuk membangun suatu objek kompleks dari suatu objek sederhana.

Dalam artikelnya, Dickau (1996) mendefinisikan *L-system* sebagai cara ringkas untuk menggambarkan grafis berulang menggunakan analogi “penyu” atau dikenal dengan istilah “Turtle Implementation”. *L-systems* dibuat dengan suatu aksioma seperti satu segmen garis dan satu atau lebih aturan produksi, yang merupakan pernyataan seperti ”mengganti satu segmen garis dengan berbelok ke kiri, mengganti satu segmen garis dengan berbelok ke kanan, dan mengganti satu segmen garis dengan aturan lainnya”.

Dalam *deterministic L-systems* terdapat beberapa komponen utama yaitu huruf, aksioma, produksi. Berikut ini penjelasan dari komponen-komponen tersebut.

a. Huruf

Huruf adalah himpunan hingga  $V$  dari simbol-simbol formal, misalnya dalam bentuk  $a, b, c$ , dan seterusnya, atau mungkin beberapa huruf (*character*) lainnya.

b. Aksioma

Aksioma (*inisiator*) adalah suatu *string*  $w$  dari simbol-simbol pada  $V$ . Himpunan *string* dari  $V$  dinotasikan  $V^*$ . Jika diberikan  $V = \{a, b, c\}$ , maka

beberapa contoh *string* yang dapat dibuat yaitu :  $a,b,ac,ba,acb,baac$ , dan seterusnya. Panjang  $|w|$  dari suatu *string*  $w$  adalah jumlah simbol dalam string.

c. Produksi

Produksi (aturan penulisan kembali) adalah suatu pemetaan simbol  $a \in V$  ke *string*  $w \in V^*$ . Ditulis dengan notasi :

$$p : a \rightarrow w$$

Jika suatu simbol  $a \in V$  tidak memiliki aturan produksi, maka dapat diasumsikan bahwa simbol tersebut dipetakan pada dirinya sendiri sehingga  $a$  menjadi konstanta *L-systems* (Wright, 1996).

Sedangkan dalam *stochastic L-systems* terdapat komponen lain selain huruf, aksioma, dan produksi, yaitu fungsi  $\pi : P \rightarrow R$  yang merupakan pemetaan barisan produksi ke dalam barisan bilangan real non-negatif yang disebut sebagai *faktor probabilitas*.

Misalkan diberikan *L-systems* dengan komponen sebagai berikut :

$$V = \{a,b,c\}$$

$$w = b$$

$$p_1 : a \rightarrow ca$$

$$p_2 : b \rightarrow ac$$

$$p_3 : c \rightarrow ab$$

Hasil produksi dari *L-systems* didefinisikan sebagai barisan  $\{g_n\}$ ,  $n = 0,1,2,3,\dots,k$ . dimana setiap generasi  $g_n$  adalah sebuah string dalam  $V^*$  yang berkembang dari generasi masing-masing simbol  $g_{n-1}$  dengan menerapkan semua aturan produksi untuk masing-masing simbol  $g_{n-1}$ . Generasi pertama  $g_0$  adalah aksioma  $w$  (Wright, 1996). Beberapa generasi dari sistem ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 Generasi *L-systems*

Generasi	Hasil Produksi
$g_0$	$b$
$g_1$	$ac$
$g_2$	$cab$
$g_3$	$bcaac$
$g_4$	$acbcacab$
$g_5$	$cabacbcabcaac$
$g_6$	$bcaaccabacbcacbcacab$
$g_7$	$acbcacabbcaaccabacbcacabacbcabcaac$

## 2.2 Jenis-Jenis *L-Systems*

Dilihat dari jumlah aturan produksi untuk satu simbol, *L-systems* dibagi menjadi dua yaitu :

### 1. *Deterministic L-Systems*

Adalah jenis *L-systems* yang hanya ada satu aturan produksi untuk satu simbol dan barisan generasi  $g_n$  didefinisikan sebagai elemen  $V^*$ . Contoh dari *L-systems* jenis ini dapat dilihat pada Tabel 2.1.

### 2. *Stochastic L-Systems*

Adalah jenis *L-systems* yang menggunakan lebih dari satu aturan produksi untuk satu simbol tertentu, misalnya  $a \rightarrow w_1$  dan  $a \rightarrow w_2$ . Pada jenis *L-systems* jenis ini diperlukan suatu kriteria untuk menentukan kapan suatu aturan produksi diterapkan.

Jika diberikan komponen *L-systems* :

$$V = \{a, b, c\}$$

$$w = a$$

$$p_1 : a \xrightarrow{0.5} a$$

$$p_2 : a \xrightarrow{0.5} ac$$

$$p_3 : c \rightarrow ca$$

Karena huruf  $a$  pada komponen di atas dipetakan menjadi dua kemungkinan maka untuk menggunakannya diperlukan faktor probabilitas. Penentuan faktor probabilitas tersebut ditentukan berdasarkan pengamatan pada objek. Masing-masing produksi pada contoh tersebut memiliki besar probabilitas yang sama yaitu  $\frac{1}{2}$ .

Beberapa kemungkinan generasi dari sistem tersebut dapat dilihat pada Tabel 2.2 di bawah ini.

Tabel 2.2 Generasi *stochastic L-systems*

Generasi	Hasil Produksi
$g_0$	$a$
$g_1$	$ac$
$g_2$	$aca$
$g_3$	$accaa$
$g_4$	$acacaac$

Dari jenis-jenis *L-systems* tersebut, kemudian dikenal istilah *context free deterministic L-systems* yang disingkat dengan *DOL-systems* (Wright, 1996) dan *context sensitive L-systems* yang disingkat dengan *DIL-systems* (Hanan, 1992).

Sedangkan jika dilihat dari penggunaan simbol untuk aturan produksinya, *L-systems* dibagi menjadi dua, yaitu :

1. *Context Free L-Systems*

*Context free L-systems* merupakan jenis *L-systems* dimana aturan produksinya hanya memperhatikan satu simbol individu dan bukan dari tetangga-

tetangganya. Contoh :  $p: A \rightarrow BA$ . Untuk contoh yang lebih spesifik dapat dilihat pada Tabel 2.1.

## 2. Context Sensitive L-Systems

*Context sensitive L-systems* adalah jenis *L-systems* dimana aturan produksinya berlaku pada suatu simbol dari bentuk awal (*predecessor*) hanya jika simbol tersebut memiliki tetangga.

Misal diberikan komponen *L-systems* :

$$V = \{a, b, c\}$$

$$w = aaa$$

$$p_1 : (a)>a \rightarrow bc$$

$$p_2 : (b)>c \rightarrow ab$$

$$p_3 : a \rightarrow \emptyset$$

$$p_4 : c \rightarrow c$$

Aturan produksi  $p_1$  memiliki arti jika  $a$  memiliki tetangga  $a$  di sebelah kanannya, maka  $a$  diproduksi menjadi  $bc$ . Begitu juga dengan aturan produksi  $p_2$  memiliki arti jika  $b$  memiliki tetangga  $c$  di sebelah kanannya, maka  $b$  diproduksi menjadi  $ab$ . Sedangkan hasil produksi dari  $a$  pada aturan produksi  $p_3$  adalah himpunan kosong. Sehingga hasil produksi beberapa generasi selanjutnya dari system dapat dilihat pada Tabel 2.3 berikut ini.

Tabel 2.3 Generasi *context sensitive L-systems*

Generasi	Hasil Produksi
$g_0$	$aaa$
$g_1$	$bcbc$
$g_2$	$abcabc$
$g_3$	$abcaabc$
$g_4$	$abcbcabc$





## 2.4 Percabangan pada *L-Systems*

Dalam bukunya, Lindenmayer memperkenalkan suatu notasi untuk menggambarkan grafik menggunakan string dengan tanda kurung. Tujuannya adalah untuk menggambarkan struktur percabangan yang ditemukan pada banyak tanaman, mulai dari alga sampai tanaman tingkat tinggi dengan menggunakan kerangka *L-systems*. Sehingga untuk memodelkan struktur percabangan, ada penambahan huruf dalam *L-systems* dengan dua simbol baru yaitu :

[ : menyimpan posisi saat ini dan bergerak sesuai perintah selanjutnya

] : kembali ke posisi semula yang disimpan oleh simbol “[“.

Di bawah ini adalah contoh dari penggunaan simbol percabangan dengan komponen *L-system* :

$V$  :  $\{F,+,[,]\}$

$w$  :  $F$

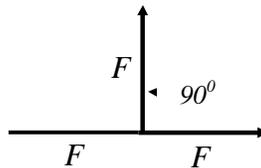
$p$  :  $F \rightarrow F[+F]F$

$\delta$  :  $90^0$

maka akan didapatkan generasi pertama  $g_1$  dengan *string* :

$F[+F]F$

Sedangkan penafsiran grafisnya dapat dilihat pada Gambar 2.2 di bawah ini :



Gambar 2.2 Penafsiran grafis percabangan *L-systems*

Untuk menafsirkan grafis gambar di atas mula-mula menjalankan perintah  $F$  yaitu menggambar segmen garis ke depan satu satuan sebesar  $d$ . Simbol [ digunakan untuk menyimpan posisi saat ini. Sedangkan simbol + digunakan untuk memutar arahnya berlawanan arah jarum jam sebesar  $\delta = 90^0$ . Setelah menjalankan perintah

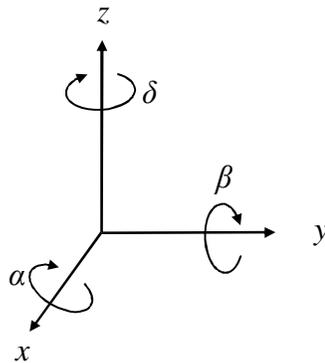
kedua simbol tersebut langkah selanjutnya yaitu menggambar  $F$  sesuai dengan arah yang telah ditentukan sebelumnya. Dilanjutkan menjalankan perintah simbol  $]$  yaitu kembali ke posisi yang disimpan pada simbol  $[$ . Perintah terakhir yaitu menggambar  $F$  sesuai dengan arah sebelum perintah simbol  $[$ . Langkah-langkah tersebut dapat juga diterapkan untuk generasi yang lebih tinggi.

### 2.5 $L$ -Systems Dimensi Tiga

Untuk menafsirkan  $L$ -systems secara grafis dalam dimensi dua hanya membutuhkan sebuah sudut. Akan tetapi dalam dimensi tiga diperlukan sesuatu yang lebih kompleks yaitu dengan menggunakan matriks rotasi  $3 \times 3$ . Dan untuk menentukan perputarannya bisa menggunakan persamaan di bawah ini.

$$[\vec{X}' \ \vec{Y}' \ \vec{Z}'] = [\vec{X} \ \vec{Y} \ \vec{Z}]R$$

dimana  $R$  adalah matriks rotasi. Untuk memutar sudut  $\alpha$ ,  $\beta$ , dan  $\delta$  pada vektor  $\vec{X}$ ,  $\vec{Y}$ ,  $\vec{Z}$  dapat digunakan rotasi matriks standard seperti Gambar 2.3 di bawah ini.



Gambar 2.3 Penafsiran grafis  $L$ -systems pada dimensi tiga

$$R(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \delta & -\sin \delta \\ 0 & \sin \delta & \cos \delta \end{bmatrix}$$

$$R(\beta) = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix}$$

$$R(\delta) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Sehingga simbol-simbol dalam dimensi tiga dapat dijelaskan sebagai berikut.

- + : berputar ke kiri dengan matriks rotasi  $R(\alpha)$
- : berputar ke kanan dengan matriks rotasi  $R(\alpha)$
- & : berputar ke kiri dengan matriks rotasi  $R(\beta)$
- ^ : berputar ke kanan dengan matriks rotasi  $R(\beta)$
- \ : berputar ke kiri dengan matriks rotasi  $R(\delta)$
- / : berputar ke kanan dengan matriks rotasi  $R(\delta)$

## 2.6 Parametrik *L-Systems*

Pada parametrik *L-systems* setiap simbol mempunyai nilai numerik yang disebut sebagai parameter. Sebagai contoh jika sebuah simbol  $A$  mempunyai parameter 5, secara sederhana dapat ditulis  $A(5)$ . Akan tetapi jika ada beberapa parameter, penulisan parameternya dipisahkan oleh koma. seperti  $A(5,2,3,8)$ .

Sebuah produksi parametrik menentukan bagaimana parameter kanan tergantung pada parameter kiri. Sisi kiri memberi nama parameter, sedangkan sebelah kanan diberikan sebagai ekspresi. Di bawah ini merupakan contoh dari penerapan parametrik *L-systems*.

Jika diberikan komponen *L-systems* :

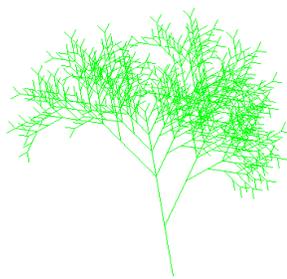
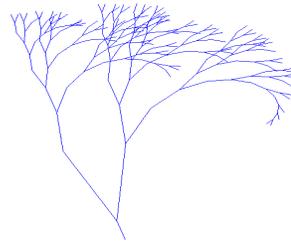
- $w$  :  $A(1)$
- $p_1$  :  $A(x) \rightarrow A(x*2)B(x)$
- $p_2$  :  $B(x) \rightarrow B(x-1)$

maka generasi *string*-nya dapat dilihat pada Tabel 2.5.

Tabel 2.5 Generasi *parametric L-systems*

Generasi	Hasil Generasi
$g_0$	$A(1)$
$g_1$	$A(2)B(1)$
$g_2$	$A(4)B(2)B(0)$
$g_3$	$A(8)B(4)B(1)B(-1)$
$g_4$	$A(16)B(8)B(3)B(0)B(-2)$
$g_5$	$A(32)B(16)B(7)B(2)B(-1)B(-3)$

Selain itu parametrik *L-systems* ini sangat diperlukan dalam memodelkan ketebalan batang tanaman yang mendekati kenyataan. Untuk menyertakan ketebalan batang tanaman dari penelitian yang dilakukan oleh Muzammil (2006), Amir (2009), dan Swandana (2010) seperti yang ditunjukkan secara berturut-turut pada Gambar 2.4 (a), Gambar 2.4 (b), dan Gambar 2.4 (c) di bawah ini, diperlukan suatu input panjang batang dan diameter batang.

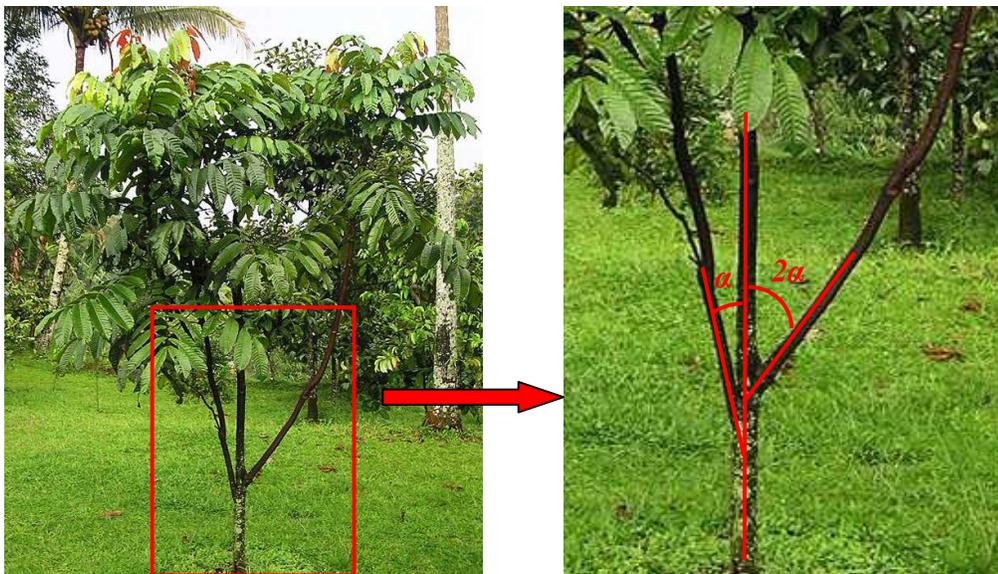
(a) Pohon Kersen  
oleh Muzammil(b) Pohon Mahoni  
oleh Amir(c) Pohon Tembakau  
oleh SwandanaGambar 2.4 Visualisasi model non-parametrik *stochastic L-systems*

Dalam menuliskan ketebalan batang dalam parametrik *L-systems* Hanan (1992) memberikan format penulisan  $A(l,w)$ , yang berarti menggambar segmen garis  $A$  sebesar satu satuan dengan panjang  $l$  dan diameter  $w$ .

## BAB 3. METODE PENELITIAN

### 3.1 Data Penelitian

Data penelitian yang digunakan dalam penelitian ini adalah beberapa foto tumbuhan dengan cabang yang terlihat jelas. Adapun alasan dipilihnya objek tumbuhan yang mempunyai cabang yang terlihat jelas yaitu agar lebih mudah dalam menunjukkan sifat *stochastic L-systems* dalam objek tersebut. Karena pemodelan yang dilakukan dalam dimensi tiga, maka foto objek yang diambil hanya dari dua sisi (sisi depan dan samping). Hal ini dilakukan karena peneliti menganggap foto objek dari dua sisi sudah mencakup dari empat sisi. Gambar 3.1 di bawah ini adalah contoh gambar tumbuhan yang dimodelkan menggunakan *stochastic L-systems*.



(a)

(b)

(a) Foto objek yang dimodelkan; (b) Identifikasi data dari objek yang didapatkan

Gambar 3.1 Contoh pohon yang akan dimodelkan

### 3.2 Langkah-langkah Penelitian

Langkah pertama yang dilakukan untuk memodelkan tanaman menggunakan *Stochastic L-systems* adalah pengambilan gambar objek yang dimodelkan yaitu berupa gambar dimensi dua. Karena pemodelan penelitian ini dilakukan dalam ruang dimensi tiga, maka foto objek diambil dari dua sisi. Setelah dilakukan pengambilan gambar objek yang dimodelkan, langkah selanjutnya yaitu menentukan besar sudut dari setiap percabangan pada tumbuhan tersebut. Dalam menentukan besar sudut yang akan dimasukkan dalam komponen *L-systems*, dipilih besar sudut percabangan yang paling kecil. Setelah itu dapat dilakukan identifikasi komponen *L-systems* yang membangunnya pada aturan produksinya. Adapun identifikasi yang dilakukan yaitu mengenai komponen-komponen dasar *L-systems* yang meliputi pemilihan aksioma, huruf, dan aturan produksi. Setelah tahap tersebut selesai dilakukan, dapat dilanjutkan penafsiran grafisnya.

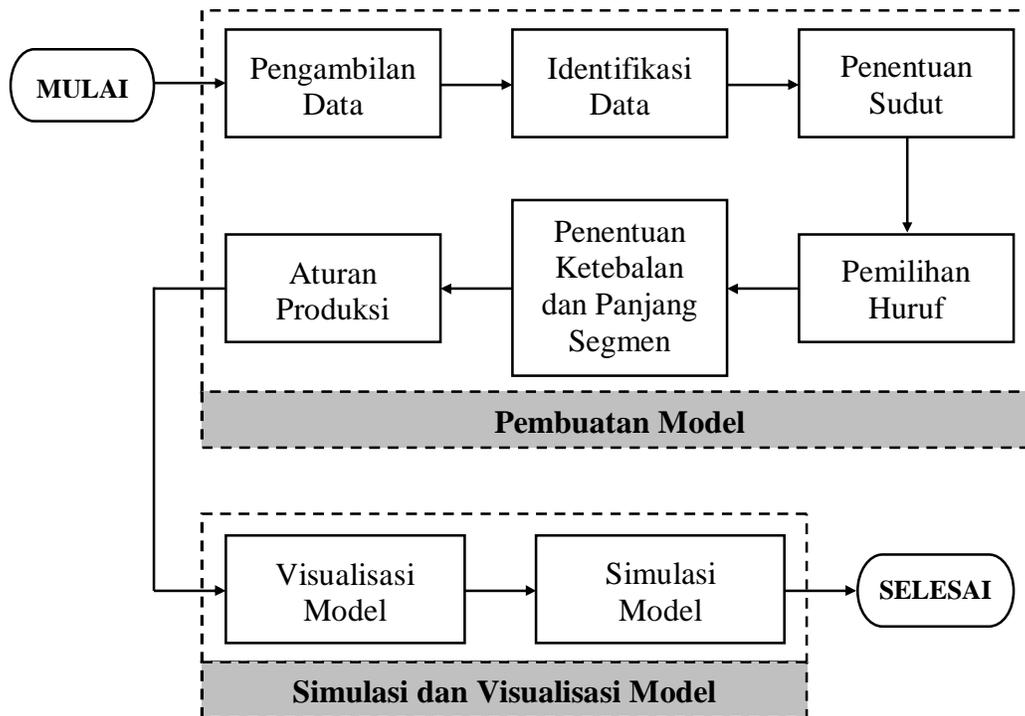
Untuk proses simulasi dan visualisasi model dilakukan dengan cara mengubah jumlah iterasi dan besar probabilitas dengan menggunakan bantuan *open source* yang berbasis bahasa pemrograman Microsoft Visual C++. Perubahan jumlah iterasi bertujuan untuk mengetahui pertumbuhan tanaman beserta percabangannya untuk beberapa generasi. Sedangkan perubahan besarnya probabilitas bertujuan untuk mengetahui kemungkinan hasil pemodelan tumbuhan yang lain (keragaman) dengan komposisi *L-systems* yang sama. Untuk melakukan proses simulasi dan visualisasi, semua komponen *L-systems* harus dimasukkan ke dalam program. Adapun algoritma dari proses simulasi dan visualisasi ini antara lain:

- a. Memasukkan besar sudut percabangan;
- b. Memasukkan besar ketebalan batang;
- c. Memasukkan aksioma awal yang berfungsi sebagai  $g_0$ ;

- d. Memasukkan aturan produksi. Dengan adanya *parametric L-systems*, pada tahap ini dapat juga memasukkan besarnya ketebalan batang, panjang segmen garis, besarnya probabilitas untuk tiap aturan produksi, serta warna batang atau daun.
- e. Memasukkan banyaknya iterasi yang bertujuan untuk mengetahui hasil dari beberapa generasi. Pada tahap ini akan diketahui pertumbuhan batang dan percabangan dari tiap generasi.

Dengan menjalankan kelima langkah tersebut, visualisasi model dalam dimensi tiga yang dibuat akan tampak hasilnya. Kemudian dari hasil tersebut ditinjau ulang apakah ada kekurangan atau sudah memenuhi kriteria model yang diinginkan sesuai dengan foto objek yang didapatkan. Jika terdapat kekurangan dari model yang telah dihasilkan, maka semua langkah di atas diulangi sampai didapatkan model yang mirip dengan foto objek yang didapatkan. Pada tahap simulasi awal, komponen probabilitas tidak perlu dimasukkan. Hal ini bertujuan untuk memudahkan dalam simulasi model tersebut. Komponen probabilitas ini dimasukkan ketika simulasi sudah dianggap memenuhi kriteria model yang diinginkan. Karena pada dasarnya probabilitas ini digunakan untuk mengetahui keragaman dari model awal dengan komposisi *L-systems* yang sama.

Secara garis besar langkah-langkah yang digunakan dalam penelitian ini terdiri atas pengambilan data, analisis data, pembuatan model, simulasi model dan visualisasi model dengan menggunakan *open source* yang berbasis bahasa pemrograman Microsoft Visual C++. Gambar 3.2 menunjukkan gambaran umum dari langkah-langkah penelitian.



Gambar 3.2 Skema langkah-langkah penelitian

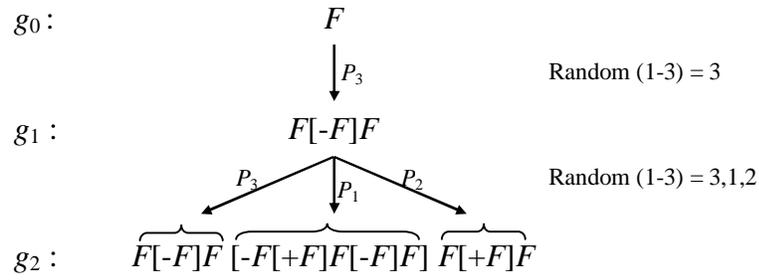
Misalkan diberikan suatu komponen *L-systems*

$$F \rightarrow F[+F]F[-F]F : 50\%$$

$$F \rightarrow F[+F]F : 25\%$$

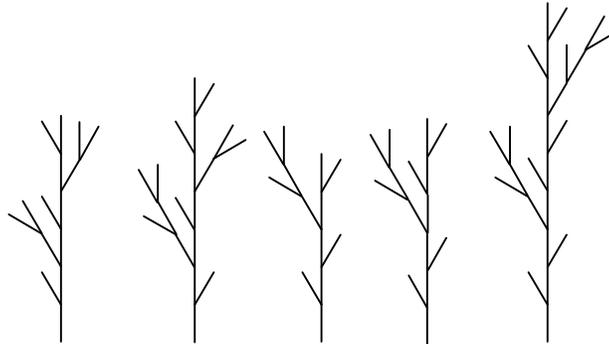
$$F \rightarrow F[-F]F : 25\%$$

Hal ini berarti bahwa distribusi acak antara tiga produksi tersebut adalah 50%, 25%, 25% tetapi kemungkinan terpilihnya tiap aturan produksi tidaklah pasti. Setiap kali simbol  $F$  ditemukan, ada kemungkinan 50% bahwa aturan produksi pertama akan diterapkan. Jika tidak diterapkan, maka ada kemungkinan 50% antara aturan produksi kedua dan ketiga. Gambar 3.3 memberikan ilustrasi contoh di atas.



Gambar 3.3 Ilustrasi iterasi *stochastic L-systems*

Sehingga beberapa kemungkinan model tanaman yang dihasilkan dari komponen *L-systems* di atas dapat dilihat pada Gambar 3.4 di bawah ini :



Gambar 3.4 Beberapa model *stochastic L-systems*

## **BAB 4. HASIL DAN PEMBAHASAN**

Pada bab ini dibahas mengenai pemodelan, simulasi serta visualisasi pertumbuhan batang tanaman dengan menggunakan *stochastic L-systems* sesuai dengan langkah-langkah yang telah diuraikan dalam bab 3 dengan mengasumsikan tidak menyertakan pertumbuhan daun. Untuk mendapatkan model pertumbuhan batang tanaman dengan menggunakan *stochastic L-systems* langkah awal yang dilakukan yaitu mendapatkan model tersebut dengan menggunakan *deterministic L-systems*. Berikut ini adalah beberapa tahap yang dilakukan untuk memodelkan pertumbuhan batang tanaman menggunakan *stochastic L-systems*.

### **4.1 Membangun Penafsiran Grafis *L-system* Secara Manual**

#### **4.1.1 Identifikasi komponen utama *L-systems***

Identifikasi dan penentuan komponen *L-systems* sangat penting dilakukan sebagai langkah awal untuk membangun penafsiran grafis dari suatu model pertumbuhan batang tanaman. Pertama, penggunaan huruf yang akan digunakan dalam aturan produksi sebaiknya digunakan seminim mungkin. Hal ini dilakukan agar aturan produksi yang terbentuk lebih sederhana sehingga lebih mudah dipahami. Huruf yang dapat digunakan yaitu huruf abjad A sampai dengan Z yang dipilih secara acak sesuai dengan kebutuhan peneliti. Setiap huruf tersebut mempresentasikan makna yang sama yaitu menggambar maju dari sebuah segmen garis. Kedua, untuk menentukan aksioma yang digunakan harus digunakan huruf yang telah ditentukan pada tahap sebelumnya. Pada tahap ini merupakan penentuan visualisasi awal seperti apa yang diinginkan oleh peneliti, apakah berupa segmen garis atau berupa tunas tanaman. Ketiga, pembuatan aturan produksi dilakukan bersamaan dengan penentuan



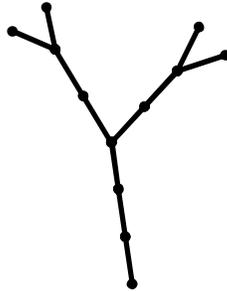
Tabel 4.1 Beberapa generasi *L-systems*

Generasi	Hasil Produksi
$g_0$	$A$
$g_1$	$SF[+++////////B][-\\A]$
$g_2$	$SFF[+++////////SF[---C][+C]][-\\SF[+++////////B][-\\A]]$
$g_3$	$SFFF[+++////////SFF[---SF[+++B][---B]][+++SF[+++B][---B]]$ $[-\\SFF[+++////////SF[---C][+C]][-\\SF[+++////////B]$ $[-\\A]]$

Pada generasi *L-systems* di atas,  $A$  merupakan hasil dari generasi pertama yang juga sebagai aksioma. Huruf ini hanya membentuk satu buah batang tanaman. Selanjutnya pada generasi kedua huruf  $A$  akan membentuk dua cabang yang berlawanan arah yaitu “ $+/B$ ” dan “ $-\\A$ ”. Simbol “ $+$ ” dan simbol “ $-$ ” membuat batang tanaman berputar pada bidang  $xy$ . Simbol “ $/$ ” membuat batang tanaman berputar searah jarum jam pada bidang  $yz$ . Sedangkan simbol “ $\\$ ” membuat batang berputar berlawanan arah jarum jam pada bidang  $yz$ . Hasil penafsiran grafis dari iterasi kedua dapat dilihat pada Gambar 4.1 di bawah ini.

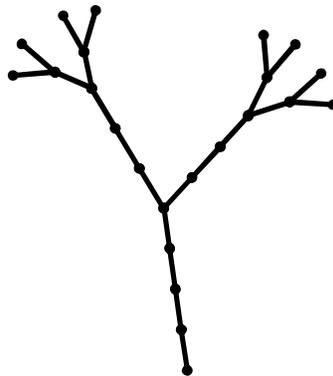
Gambar 4.1 Iterasi kedua *L-systems*

Pada generasi ketiga tanaman bertambah tinggi dan membentuk dua cabang lagi dari tiap-tiap cabang yang terbentuk sebelumnya. Pertambahan tinggi dan cabang pada generasi ketiga dapat dilihat pada Gambar 4.2.



Gambar 4.2 Iterasi ketiga *L-systems*

Dan pada generasi keempat tanaman tumbuh lebih tinggi dan menghasilkan dua cabang lagi dari generasi sebelumnya. Jadi total cabang yang terbentuk sampai generasi keempat adalah empat belas cabang. Gambar 4.3 di bawah ini merupakan hasil gambar dari generasi keempat.

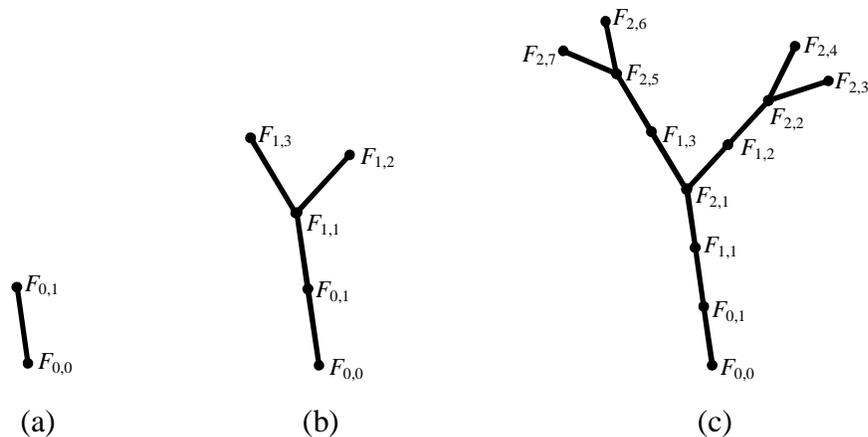


Gambar 4.3 Iterasi keempat *L-systems*

#### 4.1.2 Penamaan Titik

Setelah dilakukan penafsiran grafis, langkah penting selanjutnya yaitu penamaan titik. Hal ini dilakukan untuk mempermudah peneliti dalam mendefinisikan urutan langkah untuk membangun penafsiran grafis secara manual. Sehingga peneliti dapat mengetahui titik-titik yang membangun garis-garisnya pada

tiap generasinya. Seperti yang sudah diungkapkan pada subbab sebelumnya bahwa tiap segmen garis merepresentasikan sebuah segmen garis yang memiliki dua titik yang membangunnya. Begitu juga dengan huruf  $A$  pada contoh  $L$ -systems di atas, masing-masing titik diberi nama  $F_{0,0}$  dan  $F_{0,1}$ . Pada generasi selanjutnya  $A$  diproduksi menjadi  $SF[+/B][-\backslash A]$ . Dari *string* ini dapat diketahui bahwa huruf  $A$  yang diberi nama titik  $F_{0,0}$  dan  $F_{0,1}$  diganti menjadi huruf  $S$  dan dilanjutkan dengan huruf  $F$  yang berfungsi untuk menambah ketinggian pohon. Sehingga terdapat penambahan tiga garis baru pada iterasi kedua ini. Hal ini berarti terdapat tiga titik baru yaitu  $F_{1,1}, F_{1,2}, F_{1,3}$ . Kemudian pada iterasi selanjutnya didapatkan *string*  $SFF[+/SF[-C][+C]][-\backslash SF[+/B][-\backslash A]]$ , dimana pada tahap ini terdapat penambahan tujuh garis baru, sehingga ada tujuh titik baru yang diberi nama  $F_{2,1}, F_{2,2}, F_{2,3}, F_{2,4}, F_{2,5}, F_{2,6}, F_{2,7}$ . Ilustrasi dari penjelasan di atas dapat dilihat pada Gambar 4.4 di bawah ini.



(a) Penamaan titik iterasi pertama; (b) Penamaan titik iterasi kedua;  
(c) Penamaan titik iterasi ketiga

Gambar 4.4 Penamaan titik pada  $L$ -systems

#### 4.1.3 Parametrik *L-Systems*

Langkah-langkah yang dilakukan untuk mendapatkan visualisasi yang mendekati kenyataan diantaranya dengan cara memasukkan nilai probabilitas ataupun dalam mengikutsertakan ketebalan batang dalam memodelkan tanaman. Untuk mencapai hal tersebut peneliti menggunakan parametrik *L-systems*. Dengan adanya parametrik *L-systems* ini, peneliti juga dengan mudah untuk memasukkan sudut percabangan yang diinginkan tanpa harus menulis beberapa simbol yang sama. Misal jika terdapat komponen *L-systems* dengan sudut percabangan  $\delta = 10^\circ$  maka penulisan string “+++++F” memiliki makna berbelok ke kiri sebesar  $50^\circ$  dan dilanjutkan menggambar *F*. Sedangkan dalam parametrik *L-systems* hanya cukup ditulis “+(50)F”. Berikut ini adalah simbol-simbol yang digunakan dalam parametric *L-systems*.

- $F(l)$  : Menggambar ke depan sebesar  $l$  satuan, untuk  $l > 0$ .
- $+(a)$  : Berputar ke kiri dengan matriks rotasi  $R(\alpha)$  sebesar  $a$  derajat.
- $-(a)$  : Berputar ke kanan dengan matriks rotasi  $R(\alpha)$  sebesar  $a$  derajat
- $\&(a)$  : Berputar ke kiri dengan matriks rotasi  $R(\beta)$  sebesar  $a$  derajat
- $\wedge(a)$  : Berputar ke kanan dengan matriks rotasi  $R(\beta)$  sebesar  $a$  derajat
- $/ (a)$  : Berputar ke kiri dengan matriks rotasi  $R(\delta)$  sebesar  $a$  derajat
- $\backslash (a)$  : Berputar ke kanan dengan matriks rotasi  $R(\delta)$  sebesar  $a$  derajat
- $!(x)$  : Menentukan ketebalan garis sebesar  $x$
- $\#(x,x,x)$  : Menentukan warna

Perubahan generasi dari aturan produksi *L-systems* pada Tabel 4.1 setelah memanfaatkan parametrik *L-systems* dapat dilihat pada Tabel 4.2.

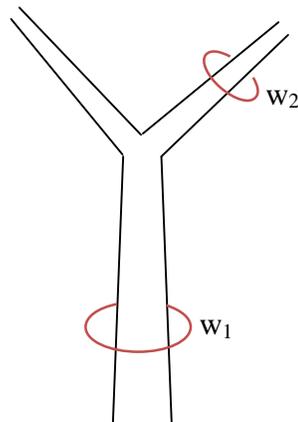
Tabel 4.2 Beberapa generasi *parametric L-systems*

Generasi	Hasil Produksi
$g_0$	$A$
$g_1$	$SF[+(30)/(90)B][- (10)\(90)A]$
$g_2$	$SFF[+(30)/(90)SF[-(30)C][+(10)C]][-(10)\(90)SF[+(30)/(90)B]$ $[-(10)\(90)A]]$
$g_3$	$SFFF[+(30)/(90)SFF[-(30)SF[+(30)B][- (30)B]][+(30)SF[+(30)B]$ $[- (30)B]][[-(10)\(90)SFF[+(30)/(90)SF[-(30)C][+(10)C]]$ $[-(10)\(90)SF[+(30)/(90)B][- (10)\(90)A]]]]$

Untuk menggambarkan ketebalan batang, peneliti mengacu pada definisi yang diungkapkan oleh Honda yang menyatakan bahwa :

- Segmen batang pada pohon adalah lurus
- Segmen ibu menghasilkan dua segmen anak melalui satu proses percabangan.
- Panjang dari dua segmen anak disingkat sebagai rasio konstan  $r_1$  dan  $r_2$ , yang berhubungan dengan segmen ibu.
- Segmen anak bercabang membentuk sudut konstan  $a_1$  dan  $a_2$  yang berhubungan dengan segmen ibu (Prusinkiewicz dan Lindenmayer, 1990).

Selain itu peneliti juga mengacu pada observasi yang dilakukan oleh Leonardo Da Vinci yang mengungkapkan bahwa “Semua cabang pohon di setiap bagian puncaknya bila disatukan akan sama dengan ketebalan pada batang di bawahnya”. Hal ini mengandung makna bahwa cabang induk dengan diameter  $w_1$  membentuk dua cabang anak dengan diameter sebesar  $w_2$ , postulat ini menghasilkan persamaan  $w_1^2 = 2w_2^2$ , yang memberikan nilai untuk  $w_r$  sama dengan  $w_2/w_1 = 1/\sqrt{2} = 0,707$  (Prusinkiewicz dan Lindenmayer, 1990). Penjelasan singkat tentang ketebalan batang dapat dilihat pada Gambar 4.5.



Gambar 4.5 Identifikasi ketebalan batang

Beberapa aturan produksi pohon  $A$  yang diberikan pada contoh hal.21 dapat ditulis sebagai berikut.

$$w = A(1,80)$$

$$p_1 : A(l,w) \rightarrow !(w*0.01)SF(l)[+(a_0)/(a_2)B(l*r_2,w*w_r)][-(a_1)\backslash(90)A(l*r_1,w*w_r)]$$

$$p_2 : B(l,w) \rightarrow !(w*0.01)SF(l)[-(a_0)C(l*r_2,w*w_r)][+(a_1)C(l*r_1,w*w_r)]$$

$$p_3 : C(l,w) \rightarrow !(w*0.01)SF(l)[+(a_0)B(l*r_2,w*w_r)][-(a_0)B(l*r_1,w*w_r)]$$

$$p_4 : S \rightarrow SF$$

Ilustrasi tentang proses identifikasi komponen parametrik  $L$ -systems dapat dilihat pada Gambar 4.6 di bawah ini

Gambar 4.6 Identifikasi parametrik  $L$ -systems

Puncak dari  $A$  menghasilkan ruas  $SF$  dan dua cabang  $B$  dan  $A$  dalam setiap langkah iterasi. Konstanta  $r_1$  dan  $r_2$  menentukan rasio kontraksi dimana  $r_1 = a/b$ ,  $r_2 = c/a$ , sedangkan  $a_0$  dan  $a_1$  adalah sudut percabangan. Dan  $!(w)$  berfungsi menetapkan ketebalan batang tanaman yang nilainya turun dari tiap iterasi dengan adanya faktor  $w_r = 0,707$ .  $A(l,w)$  menghasilkan batang utama dari tanaman dengan panjang dan tebal yang sudah ditentukan. Sedangkan  $B(l,w)$  adalah cabang pertama yang keluar dari batang utama dengan besar sudut percabangan  $a_0$ , ketebalan batang  $w*w_r$ , panjang cabang sebesar  $l*r_2$ . Dalam hal ini  $L$ -systems juga berfungsi sebagai model komputasi paralel.

$$w : A(1,80)$$

$$p_1 : A(l,w) \rightarrow !(w*0.01) \dots A(l*r_1, w*w_r)$$

Hasil perhitungan nilai parameter dari huruf  $A$  dengan  $r_1$  diasumsikan bernilai 0,9 adalah sebagai berikut.

$$A(1,80) \Rightarrow A(0.9,56.6) \Rightarrow A(0.8,39.9) \Rightarrow A(0.7,28.2) \dots$$

Setelah mendapatkan aturan produksi dengan menggunakan *Parametric DOL L-systems*, langkah yang dilakukan berikutnya yaitu memasukkan faktor probabilitas ke dalam aturan produksi tersebut apabila terdapat sebuah huruf yang diproduksi menjadi beberapa *string* yang berbeda. Untuk melakukan perintah tersebut dalam program, peneliti membuat program yang bertipe Boolean. Hal ini bertujuan apabila pengguna (*user*) memasukkan karakter yang salah maka program tidak dapat berjalan. Nilai probabilitas didapat dengan cara memperkirakan aturan produksi/percabangan mana yang sering muncul dalam objek tanaman.

Dari aturan produksi yang sudah didapatkan, peneliti menambahkan komponen kondisi (*cond*) dalam aturan produksi. Hal ini bertujuan untuk menentukan serta membatasi berapa besar parameter yang ingin disertakan dalam model. Sehingga beberapa aturan produksi di atas dapat ditulis sebagai berikut.

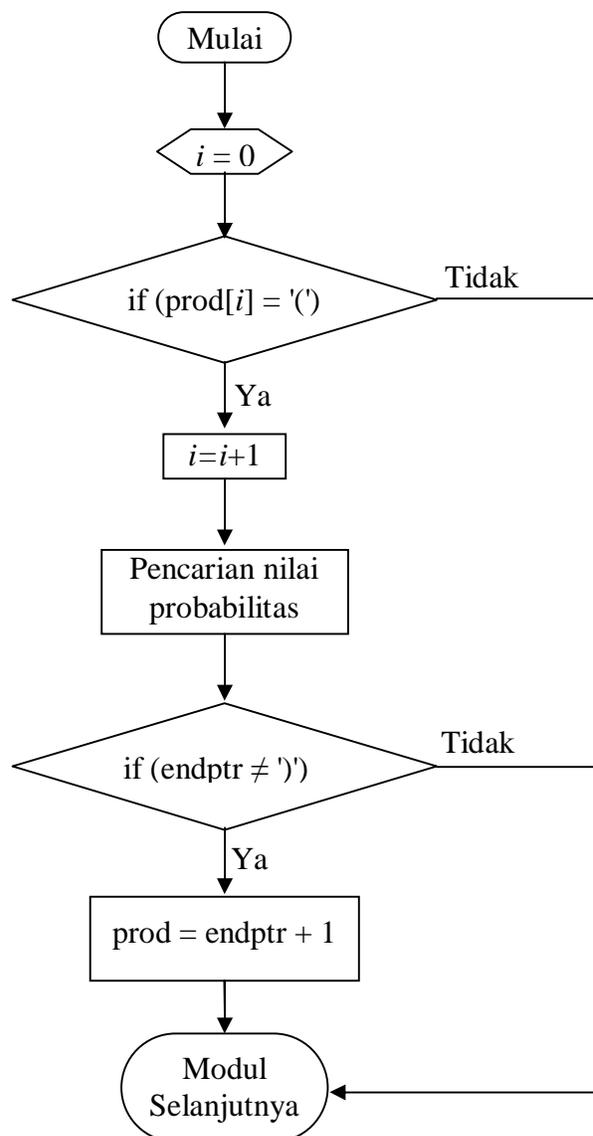
$$p_1 : A(l,w) : (l \geq 0.0) \rightarrow (0.4)!(w \cdot 0.01)SF(l)[+(a0)/(a2)A(l^*r2,w^*w_r)][-(a1)\backslash(90) \\ A(l^*r1,w^*w_r)]$$

$$p_2 : A(l,w) : (l \geq 0.0) \rightarrow (0.6)!(w \cdot 0.01)SF(l)[-(a0)C(l^*r2,w^*w_r)][+(a1)C(l^*r1,w^*w_r)]$$

$$p_3 : C(l,w) \rightarrow !(w \cdot 0.01)SF(l)[+(a0)A(l^*r2,w^*w_r)][-(a0)A(l^*r1,w^*w_r)]$$

$$p_4 : S \rightarrow SF$$

Penyisipan karakter nilai probabilitas dalam program dilakukan dengan menggunakan flowchart pada Gambar 4.7 di bawah ini.



Gambar 4.7 Skema penyisipan nilai probabilitas

Peneliti membagi aturan produksi menjadi empat komponen utama. Hal ini dilakukan untuk mempermudah dalam pembuatan program untuk menyisipkan nilai probabilitas. Contoh komponen utama dari  $p_1$  pada aturan produksi di atas dapat dilihat pada Tabel 4.3 di bawah ini.

Tabel 4.3 Komponen utama aturan produksi dalam program

No	Komponen	String
1	<i>Sym</i> (symbol)	$A(l,w)$
2	<i>Cond</i> (condition)	$(l \geq 0.0)$
3	<i>Prob</i> (probability)	(0.4)
4	<i>Prod</i> (production)	$!(w*0.01)SF(l)[+(a0)/(a2)A(l*r2,w*w_r)]$ $[-(a1)\(90)A(l*r1,w*w_r)]$

Untuk menerapkan probabilitas dalam program, diperlukan sebuah metode yang digunakan untuk memilih aturan produksi yang disubstitusi dari beberapa aturan produksi yang memiliki huruf (*predecessor*) yang sama menurut bobot probabilitasnya. Script program yang digunakan adalah sebagai berikut.

```
float accu = 0;
float rnd = (float)(((double)rand())/RAND_MAX);
BoolExpression be(fe);
```

Kemudian untuk mengiterasikan selama pemilihan aturan produksi digunakan perintah:

```
for (int i=0; i<m_NumProductions; i++)
{
    if ((m_Productions[i].symbol == symbol) &&
        (m_Productions[i].numparam == num))
    {
        accu += m_Productions[i].probability;
        if (accu > rnd)
        {
            pe = m_Productions[i];
            if (m_Productions[i].condition[0] == '\0')
return true;
```

```

        if (be.Evaluate(m_Productions[i].condition) !=
            0) return true;
    }
}
}

```

'accu' digunakan untuk mendapatkan penambahan probabilitas yang ditentukan. Dan dilanjutkan dengan membandingkan *running* total dengan skala nilai acak. Setelah itu melakukan pengulangan substitusi aturan produksi sekaligus melakukan tes fungsi aturan produksi yang memiliki bobot probabilitas beberapa kali. Tujuannya adalah untuk mendapatkan frekuensi masing-masing huruf dalam daftar entry (*production entry*) dari aturan produksi. Frekuensi setiap substitusi *string* yang sudah didapatkan kemudian diurutkan. Berikut ini adalah *script* program yang digunakan.

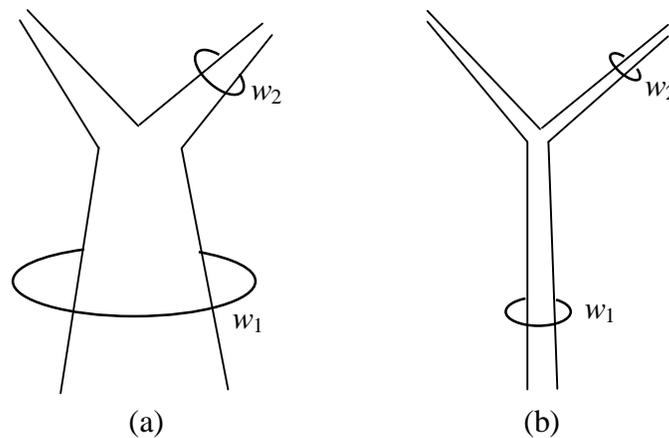
```

    for (int i=0; i<m_NumProductions; i++)
    {
        if ((m_Productions[i].symbol == symbol) &&
            (m_Productions[i].numparam == num))
        {
            pe = m_Productions[i];
            return true;
        }
    }
    int i;
    FuncExpression fe;
    for (i=0; i<numvalues; i++)
    {
        fe.SetVariable(pe.param[i], values[i]);
    }
    for (i=0; i<globalfe.GetNumVariables(); i++)
    {
        float value;
        char *name;
        globalfe.GetVariable(i, name, value);
        fe.SetVariable(name, value);
    }
}

```

## 4.2 Simulasi dan Visualisasi

Untuk mendapatkan hasil akhir pemodelan pertumbuhan batang tanaman dimensi tiga yang mendekati kenyataan, dilakukan beberapa simulasi. Simulasi yang dilakukan antara lain menentukan nilai panjang dan ketebalan yang dimasukkan sebagai aksioma batang yang sesuai dengan objek. Apabila nilai-nilai tersebut tidak sesuai maka hasil visualisasi yang didapat tampak gemuk atau kurus. Gambar 4.8 di bawah ini memberikan ilustrasi dari penjelasan di atas.



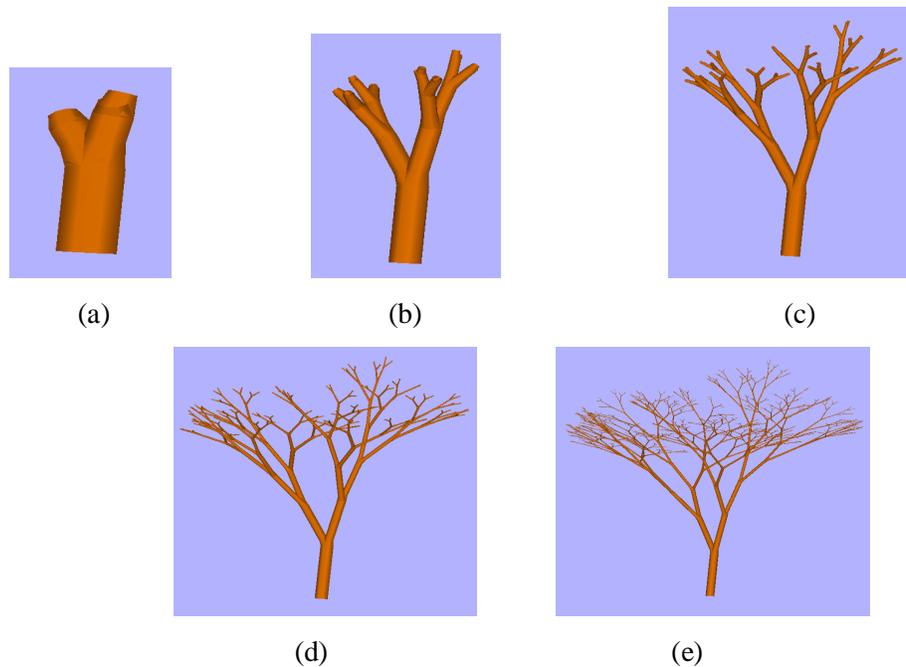
(a) Ilustrasi input ketebalan batang yang terlalu besar;  
 (b) Ilustrasi input ketebalan batang yang terlalu kecil

Gambar 4.8 Perbandingan input nilai panjang dan ketebalan batang tanaman

Dari gambar di atas dapat dijelaskan bahwa batang tanaman sebelah kiri memiliki nilai ketebalan batang yang terlalu besar atau nilai panjang batang yang terlalu kecil, sehingga visualisasi yang didapatkan tampak pohon sangat gemuk. Sedangkan batang tanaman sebelah kanan memiliki nilai ketebalan batang yang terlalu kecil atau nilai panjang batang yang terlalu besar, sehingga visualisasi yang didapatkan tampak pohon sangat kurus. Untuk mengatasi masalah tersebut peneliti menetapkan jumlah iterasi yang digunakan yang sesuai dengan gambar objek yang didapatkan.

Kemudian simulasi lain yang dilakukan yaitu dengan cara mengubah iterasi. Langkah ini bertujuan untuk mengetahui apakah bentuk pertumbuhan cabang batang sudah sesuai dengan data yang didapatkan. Akan tetapi langkah ini tidak cocok

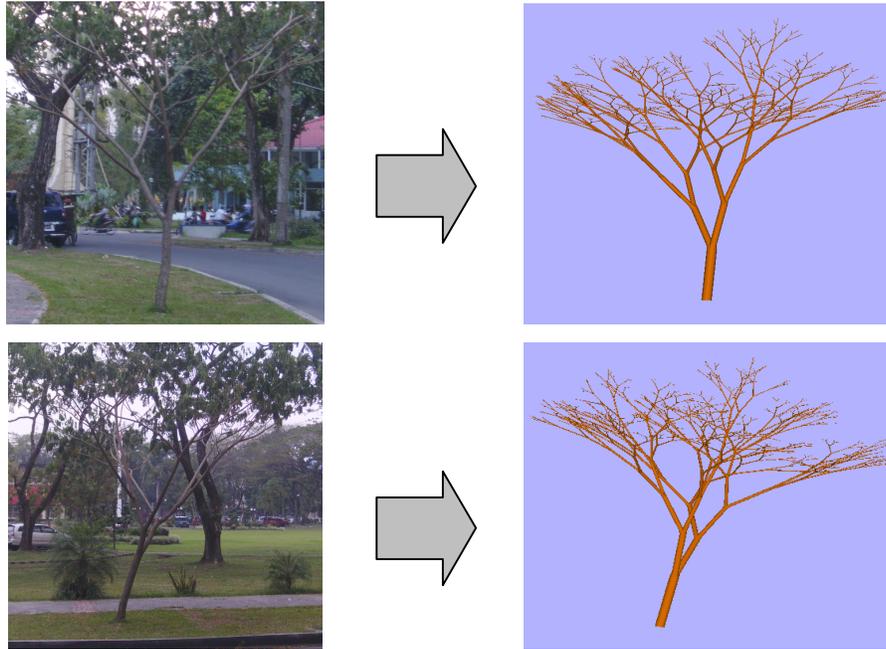
dilakukan untuk mendapatkan nilai panjang dan ketebalan batang. Karena pada tiap iterasi batang akan tumbuh semakin tinggi, sedangkan ketebalan yang diinputkan memiliki nilai tetap. Gambar 4.9 di bawah ini secara berturut-turut merupakan visualisasi pertumbuhan batang tanaman iterasi kedua, keempat, keenam, kedelapan dan kesepuluh.



(a) Visualisasi dari iterasi kedua; (b) Iterasi keempat; (c) Iterasi Keenam; (d) Iterasi kedelapan; (e) Iterasi kesepuluh

Gambar 4.9 Visualisasi pertumbuhan batang tanaman

Gambar 4.9 (e) di atas merupakan visualisasi akhir dari pertumbuhan batang A. Iterasi yang digunakan yaitu sampai iterasi kesepuluh. Kondisi ini disesuaikan dengan gambar asli tanaman tersebut. Sedangkan Gambar 4.10 merupakan perbandingan gambar pohon A asli dengan visualisasi *L-systems* dimensi tiga.



Gambar 4.10 Perbandingan pertumbuhan batang pohon Trembesi asli dengan visualisasi *L-systems* dalam dimensi tiga

Simulasi terakhir yang dilakukan yaitu penentuan nilai probabilitas yang digunakan dalam aturan produksi. Sebelum menentukan nilai probabilitas, peneliti memperkirakan percabangan/aturan produksi yang mana saja yang sering muncul dalam objek. Dari beberapa aturan produksi yang sering muncul tersebut, huruf yang berfungsi sebagai *predecessor* diganti menjadi huruf yang sama. Kemudian diberi nilai probabilitas sesuai dengan tingkat seringnya muncul dalam objek. Tabel 4.4 di bawah ini menjelaskan tentang generasi parametrik *DOL systems*. Sedangkan Tabel 4.5 menjelaskan tentang generasi parametrik *Stochastic L-systems*.

Tabel 4.4 Generasi parametrik *DOL-systems*

<i>Predecessor</i>	<i>Successor</i>
$A(l, w)$	$!(w*0.01)SF(l)[+(a_0)/(a_2)B(l*r_2, w*w_r)][-(a_1)\backslash(90)A(l*r_1, w*w_r)]$
$B(l, w)$	$!(w*0.01)SF(l)[-(a_0)C(l*r_2, w*w_r)][+(a_1)C(l*r_1, w*w_r)]$
$C(l, w)$	$!(w*0.01)SF(l)[+(a_0)B(l*r_2, w*w_r)][-(a_0)B(l*r_1, w*w_r)]$
$S$	$SF$

Tabel 4.5 Generasi parametrik *stochastic L-systems*

<i>Predecessor</i>	<i>Successor</i>
$A(l,w)$	$(0.4)!(w*0.01)SF(l)[+(a_0)/(a_2)A(l*r_2,w*w_r)][-(a_1)\(90)A(l*r_1,w*w_r)]$
$A(l,w)$	$(0.6)!(w*0.01)SF(l)[-(a_0)C(l*r_2,w*w_r)][+(a_1)C(l*r_1,w*w_r)]$
$C(l,w)$	$!(w*0.01)SF(l)[+(a_0)A(l*r_2,w*w_r)][-(a_0)A(l*r_1,w*w_r)]$
$S$	$SF$

### 4.3 Hasil Pemodelan

Setelah melakukan identifikasi komponen utama *L-systems*, penafsiran grafis *L-systems*, penamaan titik, menentukan *parametric DOL-systems*, menentukan nilai *stochastic L-systems*, dan simulasi, langkah selanjutnya yaitu memasukkan semua komponen *L-system* yang didapat ke dalam program. Berikut ini adalah komponen-komponen *L-systems* yang membentuk beberapa tanaman beserta hasil pemodelannya.

#### a). Pohon Trembesi

Komponen-komponen *L-systems*-nya:

$$V = \{r_1, r_2, w_r, l, w, A, C, S, F, *, +, -, /, \backslash, >=, !, (, ), [, ]\}$$

$$\text{Aksioma } w = A(1,80)$$

Jumlah generasi : 10

$$r_1 = 0.9$$

$$r_2 = 0.6$$

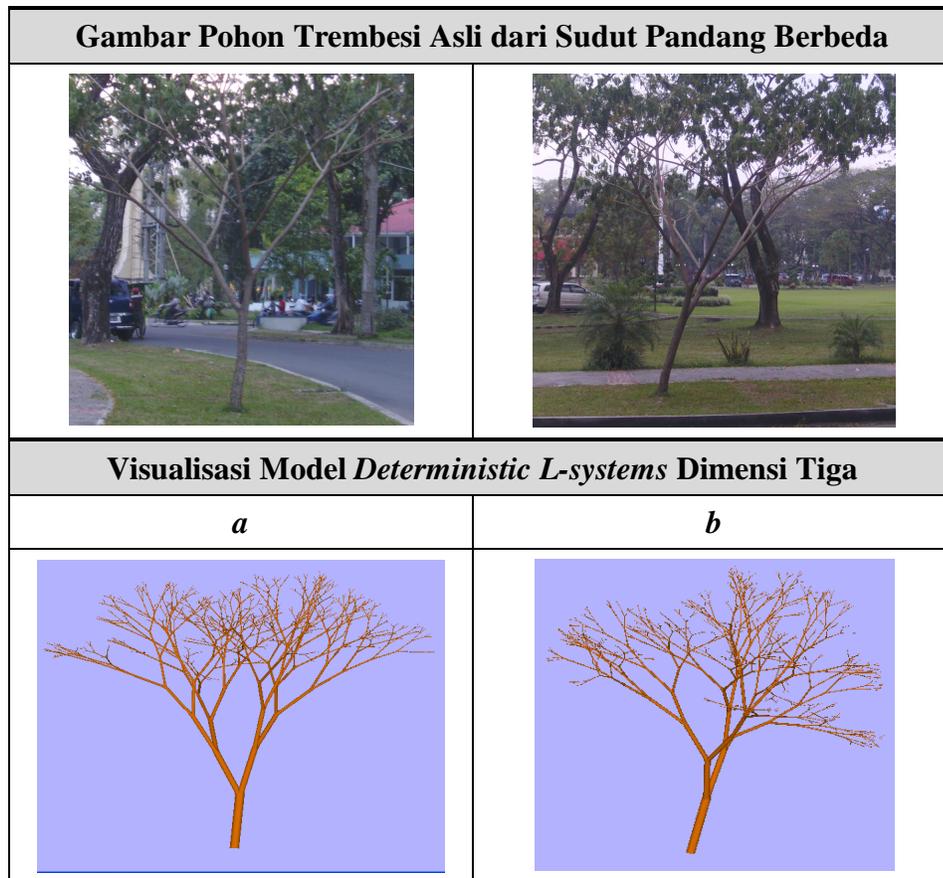
$$w_r = 0.707$$

$$p_1 : A(l,w):(l \geq 0.0) \rightarrow (0.4)!(w*0.01)SF(l)[+(30)/(90)A(l*r_2,w*w_r)][-(10)\(90)A(l*r_1,w*w_r)]$$

$$\begin{aligned}
p_2 & : A(l,w):(l \geq 0.0) \rightarrow (0.6)!(w * 0.01)SF(l)[-(30)C(l * r_2, w * w_r)][+(10) \\
& \quad C(l * r_1, w * w_r)] \\
p_3 & : C(l,w) \rightarrow !(w * 0.01)SF(l)[+(30)A(l * r_2, w * w_r)][-(30)A(l * r_1, w * w_r)] \\
p_4 & : S \rightarrow SF
\end{aligned}$$

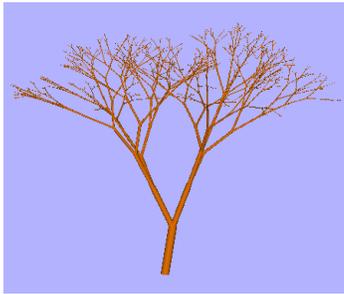
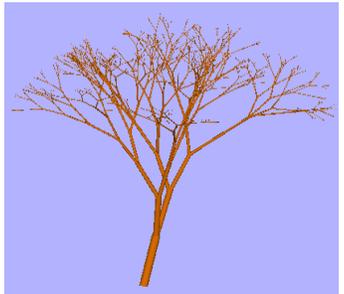
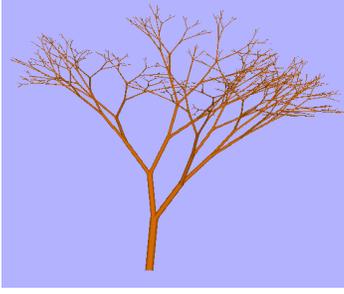
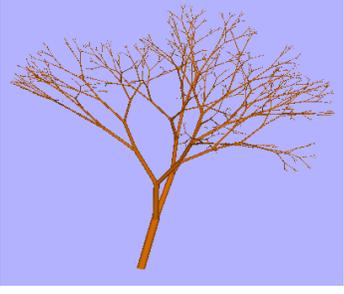
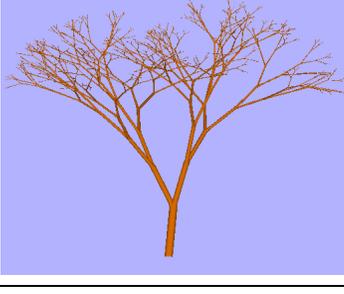
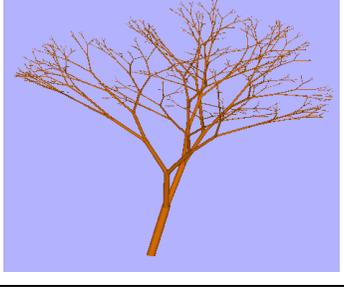
Pada komponen *L-systems* di atas, nilai awal panjang batang yang dimasukkan adalah sebesar 1 sedangkan ketebalan batang awal yang dimasukkan adalah sebesar 80. Dari keempat aturan produksi yang dibuat, hanya aturan produksi pertama dan kedua yang diberi nilai probabilitas. Aturan produksi pertama diberi nilai probabilitas sebesar 0.4 sedangkan aturan produksi kedua diberi nilai probabilitas sebesar 0.6. Hal ini berarti bahwa setiap kali ditemukan huruf A, kemungkinan 40% aturan produksi pertama akan digunakan dan 60% aturan produksi kedua yang digunakan. Pemberian nilai probabilitas aturan produksi yang kedua lebih besar daripada aturan produksi pertama. Hal ini berdasarkan pada pengamatan yang dilakukan oleh peneliti yang menilai bahwa percabangan yang terbentuk dari aturan produksi kedua tersebut cenderung lebih sering muncul dalam foto objek daripada percabangan yang terbentuk dari aturan produksi pertama. Jumlah iterasi yang digunakan yaitu sebanyak 10 iterasi karena pada iterasi ke sepuluh visualisasi gambar yang dihasilkan sudah menyerupai objek asli.

Hasil visualisasi parametrik *deterministic L-systems* dari komponen *L-system* di atas dapat dilihat pada Gambar 4.11 di bawah ini. Gambar 4.11 (a) merupakan visualisasi model dengan putaran 0°. Sedangkan Gambar 4.11 (b) merupakan visualisasi model dengan putaran 90°.



(a) Visualisasi model dengan putaran  $0^\circ$ ; (b) Visualisasi model dengan putaran  $90^\circ$ ;  
Gambar 4.11 Parametrik *deterministic L-systems* dimensi tiga pada pohon Trembesi

Sedangkan hasil visualisasi parametrik *stochastic L-systems* dari komponen *L-systems* di atas dapat dilihat pada gambar 4.12 di bawah ini. Hasil *running* pertama menghasilkan satu visualisasi gambar pohon yang ditunjukkan pada Gambar 4.12 (1). Gambar 4.12 (2) merupakan hasil *running* kedua yang juga menghasilkan 1 visualisasi dan berbeda dengan hasil *running* pertama. Dan Gambar 4.12 (3) merupakan hasil *running* ketiga.

Gambar Pohon Trembesi Asli dari Sudut Pandang Berbeda		
		
Visualisasi Model <i>Stochastic L-systems</i> Dimensi Tiga		
	<i>a</i>	<i>b</i>
1		
2		
3		

( $a_1$ ) Hasil *running* ke-1 dengan putaran  $0^\circ$ ; ( $b_1$ ) Hasil *running* ke-1 dengan putaran  $90^\circ$ ;  
 ( $a_2$ ) Hasil *running* ke-2 dengan putaran  $0^\circ$ ; ( $b_2$ ) Hasil *running* ke-2 dengan putaran  $90^\circ$ ;  
 ( $a_3$ ) Hasil *running* ke-3 dengan putaran  $0^\circ$ ; ( $b_3$ ) Hasil *running* ke-3 dengan putaran  $90^\circ$ ;

Gambar 4.12 Parametrik *stochastic L-systems* dimensi tiga pada pohon Trembesi

## b). Pohon Jung Rabab

Komponen-komponen *L-systems*-nya:

$$V = \{r_1, r_2, w_r, wr_2, l, w, A, C, D, F, G, S, T, Y, *, +, -, /, \backslash, \&, ^, >, =, !, (, ), [, ]\}$$

$$\text{Aksioma } w = (0.5, 0.2, 0) \& (90) / (180) A(1, 50)$$

Banyak iterasi : 9

$$r_1 = 0.9$$

$$r_2 = 0.6$$

$$w_r = 0.707$$

$$w_{r_2} = 0.557$$

$$p_1 : A(l, w) : (l >= 0.0) \rightarrow (0.5)!(w * 0.01)F(4 * l) [ / (30) - (20)D(2 * l * r_1, w * w_r) ]$$

$$[ / (60) ^ (40)D(l * r_2, w * w_{r_2}) ] [ \& (10) + (20)A(l * r_2, w * w_r) ]$$

$$p_2 : A(l, w) : (l >= 0.0) \rightarrow (0.5)!(w * 0.01)SF(l) [ ^ (20) - (20)SC(l * r_2, w * w_r) ]$$

$$p_3 : Y(l, w) \rightarrow !(w * 0.01)SF(l) [ + (10) ^ (10)C(l * r_2, w * w_r) ]$$

$$p_4 : C(l, w) \rightarrow !(w * 0.01)SF(l) [ + (20)C(l * r_1, w * w_r) ] [ / (30) ^ (30)A(l * r_2, w * w_r) ]$$

$$[ \& (20)SG(3 * l * r_2, w * w_r) ]$$

$$p_5 : D(l, w) \rightarrow (0.5)!(w * 0.01)SSF(l) [ \backslash (60) ^ (10) + (10)D(l * r_1, w * w_r) ]$$

$$[ \backslash (20) ^ (30)Y(l * r_1, w * w_{r_2}) ] - (20) \& (5)Y(l * r_1, w * w_{r_2})$$

$$p_6 : D(l, w) \rightarrow (0.5)!(w * 0.01)F(1) [ ^ (5) \backslash (60)D(l * r_1, w * w_r) ]$$

$$p_7 : G(l, w) \rightarrow !(w * 0.01)SF(1) [ / (60) - (20)C(l * r_1, w * w_{r_2}) ]$$

$$[ / (30) ^ (20)C(2 * l * r_2, w * w_{r_2}) ] [ + (10)G(2 * l * r_2, w * w_r) ]$$

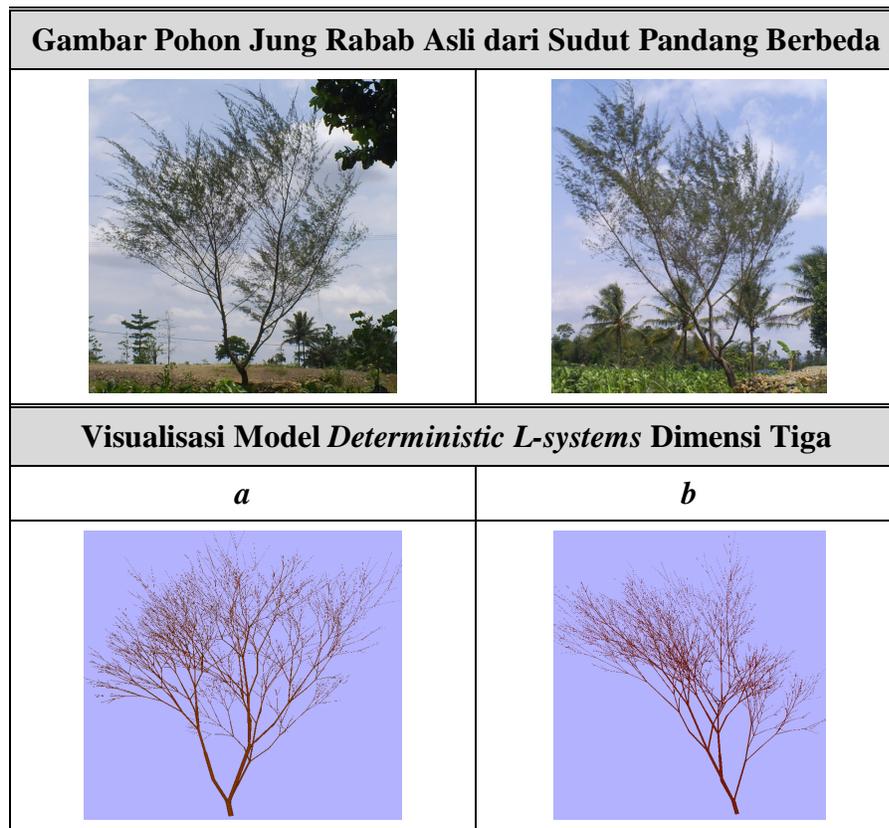
$$p_8 : S \rightarrow SF$$

$$p_9 : T \rightarrow ZF$$

Pada komponen *L-systems* di atas, nilai awal panjang batang yang dimasukan adalah sebesar 1 sedangkan ketebalan batang awal yang dimasukkan adalah sebesar 50. Aturan produksi yang diberi nilai probabilitas yaitu  $p_1$  dan  $p_2$  dengan nilai probabilitas yang sama yaitu sebesar 0.5. Hal ini berarti bahwa setiap kali ditemukan huruf  $D$ , kemungkinan 50% aturan produksi pertama akan digunakan dan 50% aturan produksi kedua yang digunakan. Pemberian nilai probabilitas yang sama terhadap kedua aturan produksi didasarkan pada pengamatan yang dilakukan oleh peneliti yang menilai bahwa percabangan yang terbentuk dari kedua aturan produksi tersebut memiliki kecenderungan muncul yang sama dalam foto objek. Jumlah iterasi yang

digunakan yaitu sebanyak 9 iterasi karena pada iterasi ke-9 visualisasi gambar yang dihasilkan sudah menyerupai objek asli.

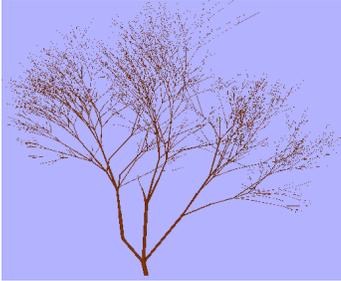
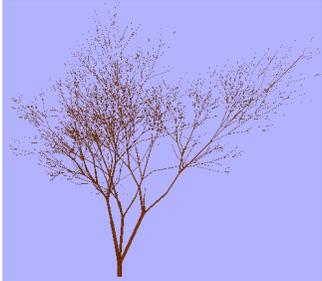
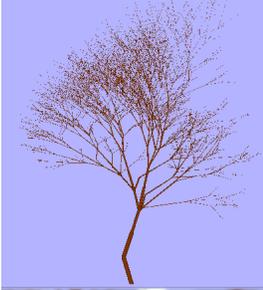
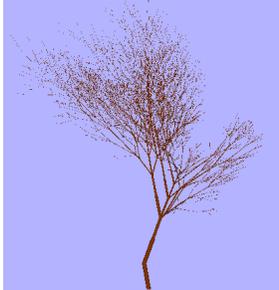
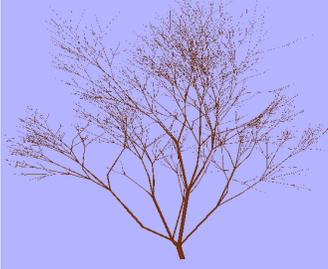
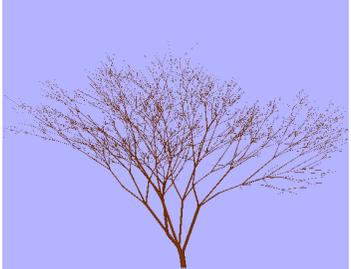
Hasil visualisasi parametrik *deterministic L-systems* dari komponen *L-system* di atas dapat dilihat pada Gambar 4.13 di bawah ini. Gambar 4.13 (a) merupakan visualisasi model dengan putaran  $0^\circ$ . Sedangkan Gambar 4.13 (b) merupakan visualisasi model dengan putaran  $90^\circ$ .



(a) Visualisasi model dengan putaran  $0^\circ$ ; (b) Visualisasi model dengan putaran  $90^\circ$ ;

Gambar 4.13 Parametrik *deterministic L-systems* dimensi tiga pada pohon Jung Rabab

Sedangkan hasil visualisasi parametrik *stochastic L-systems* dari komponen *L-systems* di atas dapat dilihat pada gambar 4.14 di bawah ini. Hasil *running* pertama menghasilkan satu visualisasi gambar pohon yang ditunjukkan pada Gambar 4.14 (1). Gambar 4.14 (2) merupakan hasil *running* kedua yang juga menghasilkan 1 visualisasi dan berbeda dengan hasil *running* pertama. Dan Gambar 4.14 (3) merupakan hasil *running* ketiga.

Gambar Pohon Jung Rabab Asli dari Sudut Pandang Berbeda		
		
Visualisasi Model <i>Stochastic L-systems</i> Dimensi Tiga		
	<i>a</i>	<i>b</i>
1		
2		
3		

( $a_1$ ) Hasil *running* ke-1 dengan putaran  $0^\circ$ ; ( $b_1$ ) Hasil *running* ke-1 dengan putaran  $90^\circ$ ;  
 ( $a_2$ ) Hasil *running* ke-2 dengan putaran  $0^\circ$ ; ( $b_2$ ) Hasil *running* ke-2 dengan putaran  $90^\circ$ ;  
 ( $a_3$ ) Hasil *running* ke-3 dengan putaran  $0^\circ$ ; ( $b_3$ ) Hasil *running* ke-3 dengan putaran  $90^\circ$ ;

Gambar 4.14 Parametrik *stochastic L-systems* dimensi tiga pada pohon Jung Rabab

## c). Pohon Cemara

Komponen-komponen *L-systems*-nya:

$$V = \{r_1, r_2, w_r, wr_2, l, w, A, B, C, D, E, S, F, T, *, +, -, /, \backslash, ^, \&, >, =, !, (, ), [, ]\}$$

$$\text{Aksioma } w = \&(90)/(180)A(1,20)$$

Banyak iterasi : 7

$$r_1 = 0.9$$

$$r_2 = 0.6$$

$$w_r = 0.707$$

$$wr_2 = 0.557$$

$$p_1 : A(l, w) \rightarrow !(w*0.01)SF(7*l)[/(10)\&(50)B(l*r_2, w*wr_2)][/(130)\&(50)B(l*r_1, w*wr_2)][\backslash(130)\&(50)B(l*r_1, w*wr_2)][/(30)SE(l*r_1, w*wr)]$$

$$p_2 : B(l, w) \rightarrow !(w*0.01)SF(l)[+(50)^(10)C(l*r_2, w*wr)][-(50)C(l*r_1, w*wr)] [^(10)B(l*r_2, w*wr)]$$

$$p_3 : C(l, w) \rightarrow !(w*0.01)TF(l)[^(10)D(l*r_2, w*wr)]$$

$$p_4 : D(l, w) \rightarrow !(w*0.01)ZF(l)[^(10)C(l*r_2, w*wr)]$$

$$p_5 : E(l, w): (l >= 0.0) \rightarrow (0.5)!(w*0.01)TF(l)[/(10)\&(l*a_3)B(l*r_2, w*wr_2)] [/(130)\&(l*50)B(l*r_1, w*wr_2)][\backslash(130)\&(l*50)B(l*r_1, w*wr_2)] [/(30)TE(l*r_1, w*wr)]$$

$$p_6 : E(l, w): (l >= 0.0) \rightarrow (0.5)!(w*0.01)ZF(l)[/(10)\&(l^2*50)B(l*r_2, w*wr_2)] [/(130)\&(l^2*50)B(l*r_1, w*wr_2)][\backslash(130)\&(l^2*50)B(l*r_1, w*wr_2)][/(40)SE(l*r_1, w*wr)]$$

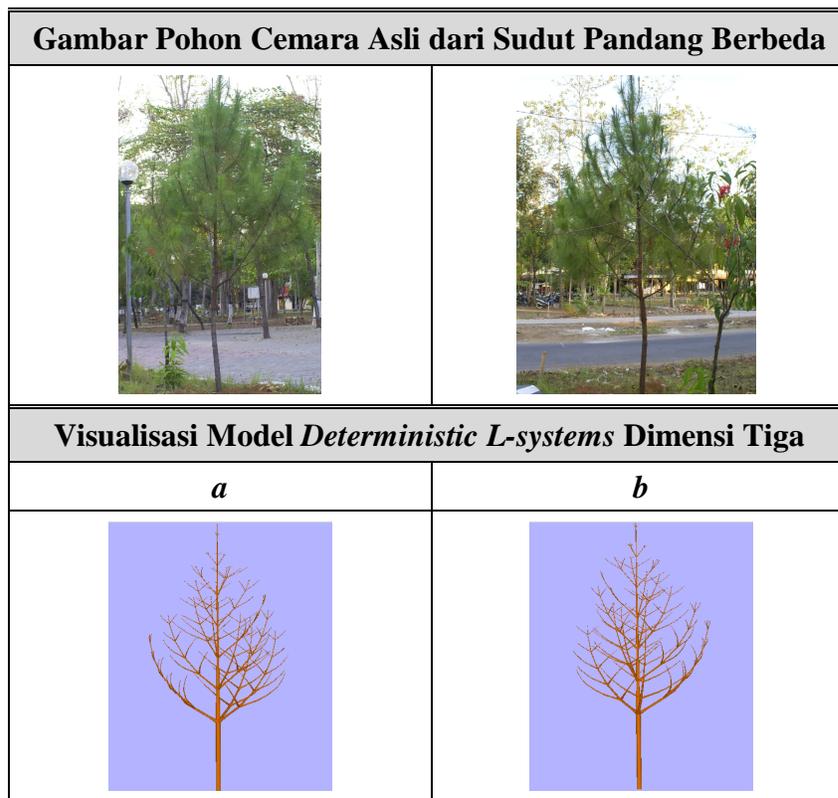
$$p_7 : S \rightarrow TF$$

$$p_8 : T \rightarrow F$$

Pada komponen *L-systems* di atas, nilai awal panjang batang yang dimasukan adalah sebesar 1 sedangkan ketebalan batang awal yang dimasukkan adalah sebesar 20. Nilai probabilitas diberikan pada aturan produksi kelima dan keenam dengan nilai probabilitas yang sama yaitu sebesar 0.5. Hal ini berarti bahwa setiap kali ditemukan huruf *E*, kemungkinan 50% aturan produksi kelima akan digunakan dan 50% aturan produksi keenam yang digunakan. Pemberian nilai probabilitas yang sama terhadap kedua aturan produksi tersebut didasarkan pada pengamatan yang dilakukan oleh peneliti yang menilai bahwa percabangan yang terbentuk dari kedua aturan produksi tersebut memiliki kecenderungan muncul yang sama dalam foto objek. Jumlah iterasi

yang digunakan yaitu sebanyak 7 iterasi karena pada iterasi ke-7 visualisasi gambar yang dihasilkan sudah menyerupai objek asli.

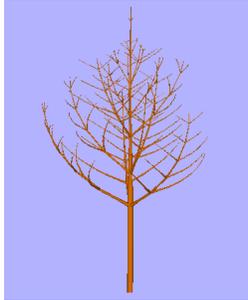
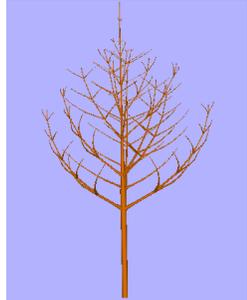
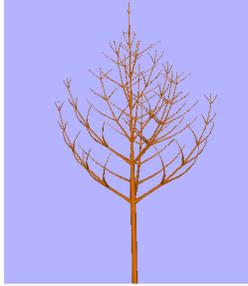
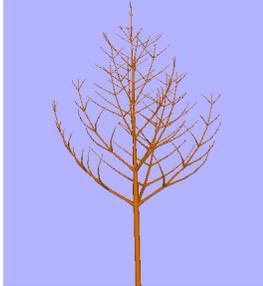
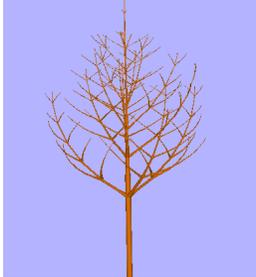
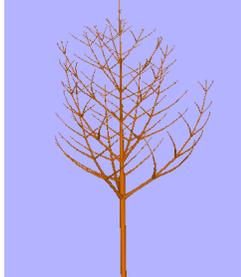
Hasil visualisasi parametrik *deterministic L-systems* dari komponen *L-system* di atas dapat dilihat pada Gambar 4.15 di bawah ini. Gambar 4.15 (a) merupakan visualisasi model dengan putaran  $0^\circ$ . Sedangkan Gambar 4.15 (b) merupakan visualisasi model dengan putaran  $90^\circ$ .



(a) Visualisasi model dengan putaran  $0^\circ$ ; (b) Visualisasi model dengan putaran  $90^\circ$ ;

Gambar 4.15 Parametrik *deterministic L-systems* dimensi tiga pada pohon Cemara

Sedangkan hasil visualisasi parametrik *stochastic L-systems* dari komponen *L-systems* di atas dapat dilihat pada gambar 4.16 di bawah ini. Hasil *running* pertama menghasilkan satu visualisasi gambar pohon yang ditunjukkan pada Gambar 4.16 (1). Gambar 4.16 (2) merupakan hasil *running* kedua yang juga menghasilkan 1 visualisasi dan berbeda dengan hasil *running* pertama. Dan Gambar 4.16 (3) merupakan hasil *running* ketiga.

Gambar Pohon Cemara Asli dari Sudut Pandang Berbeda		
		
Visualisasi Model <i>Stochastic L-systems</i> Dimensi Tiga		
	<i>a</i>	<i>b</i>
1		
2		
3		

( $a_1$ ) Hasil *running* ke-1 dengan putaran  $0^\circ$ ; ( $b_1$ ) Hasil *running* ke-1 dengan putaran  $90^\circ$ ;  
 ( $a_2$ ) Hasil *running* ke-2 dengan putaran  $0^\circ$ ; ( $b_2$ ) Hasil *running* ke-2 dengan putaran  $90^\circ$ ;  
 ( $a_3$ ) Hasil *running* ke-3 dengan putaran  $0^\circ$ ; ( $b_3$ ) Hasil *running* ke-3 dengan putaran  $90^\circ$ ;

Gambar 4.16 Parametrik *stochastic L-systems* dimensi tiga pada pohon Cemara

## d). Pohon Cemara

Komponen-komponen *L-systems*-nya:

$$V = \{r_1, r_2, w_r, wr_2, l, w, A, B, C, D, S, F, T, E, *, +, -, /, \backslash, ^, \&, >=, !, (, ), [, ]\}$$

$$\text{Aksioma } w = \&(90)/(180)A(1,15)$$

Banyak iterasi : 16

$$r_1=0.9$$

$$r_2=0.6$$

$$w_r=0.9$$

$$wr_2=0.707$$

$$p_1 : A(l, w) \rightarrow !(w*0.01)SF(l)[/(5)TE(l*r_1, w*w_r)]$$

$$p_2 : B(l, w):(l>=0.1) \rightarrow !(w*0.01)ZF(l)[+(50)^{(5)}C(l*r_2, w*wr_2)][-(50)C(l*r_1, w*wr_2)][^{(5)}B(l*r_2, w*w_r)]$$

$$p_3 : C(l, w):(l>=0.0) \rightarrow (0.5)!(w*0.01)F(l)[^{(5)}D(l*r_2, w*wr_2)]$$

$$p_4 : C(l, w):(l>=0.0) \rightarrow (0.5)!(w*0.01)F(l)[^{(5)}C(l*r_2, w*wr_2)]$$

$$p_5 : E(l, w) \rightarrow !(w*0.01)F(l)[/(10)\&(l*90)B(l*r_2, w*wr_2)][/(130)\&(l*90)$$

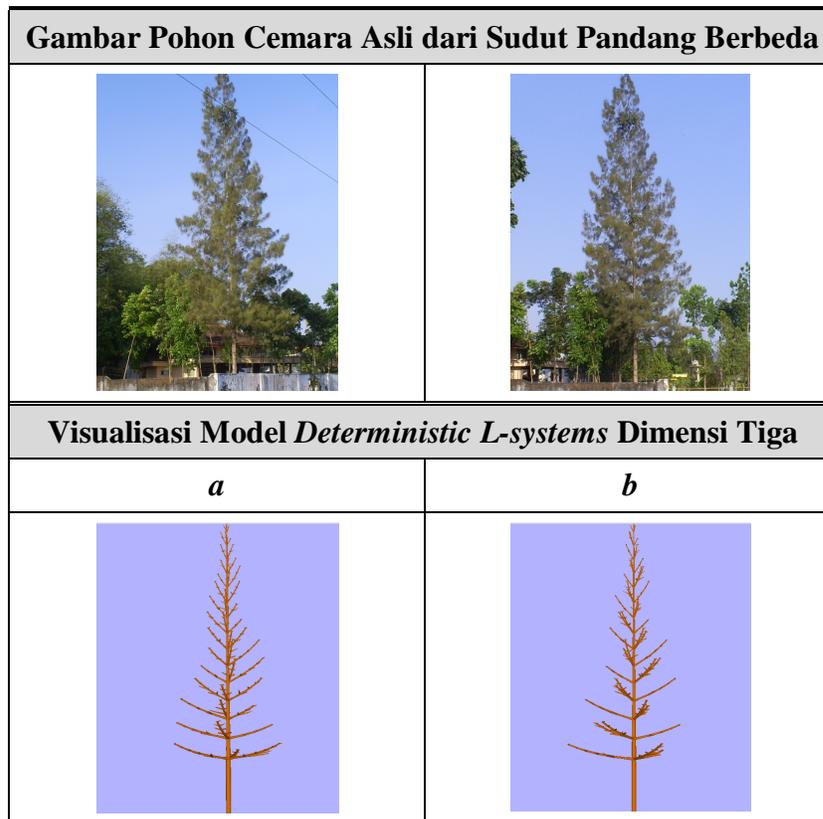
$$B(l*r_1, w*wr_2)][\&(130)\&(l*90)B(l*r_1, w*wr_2)][/(60)TE(l*r_1, w*w_r)]$$

$$p_6 : S \rightarrow TF$$

$$p_7 : T \rightarrow F$$

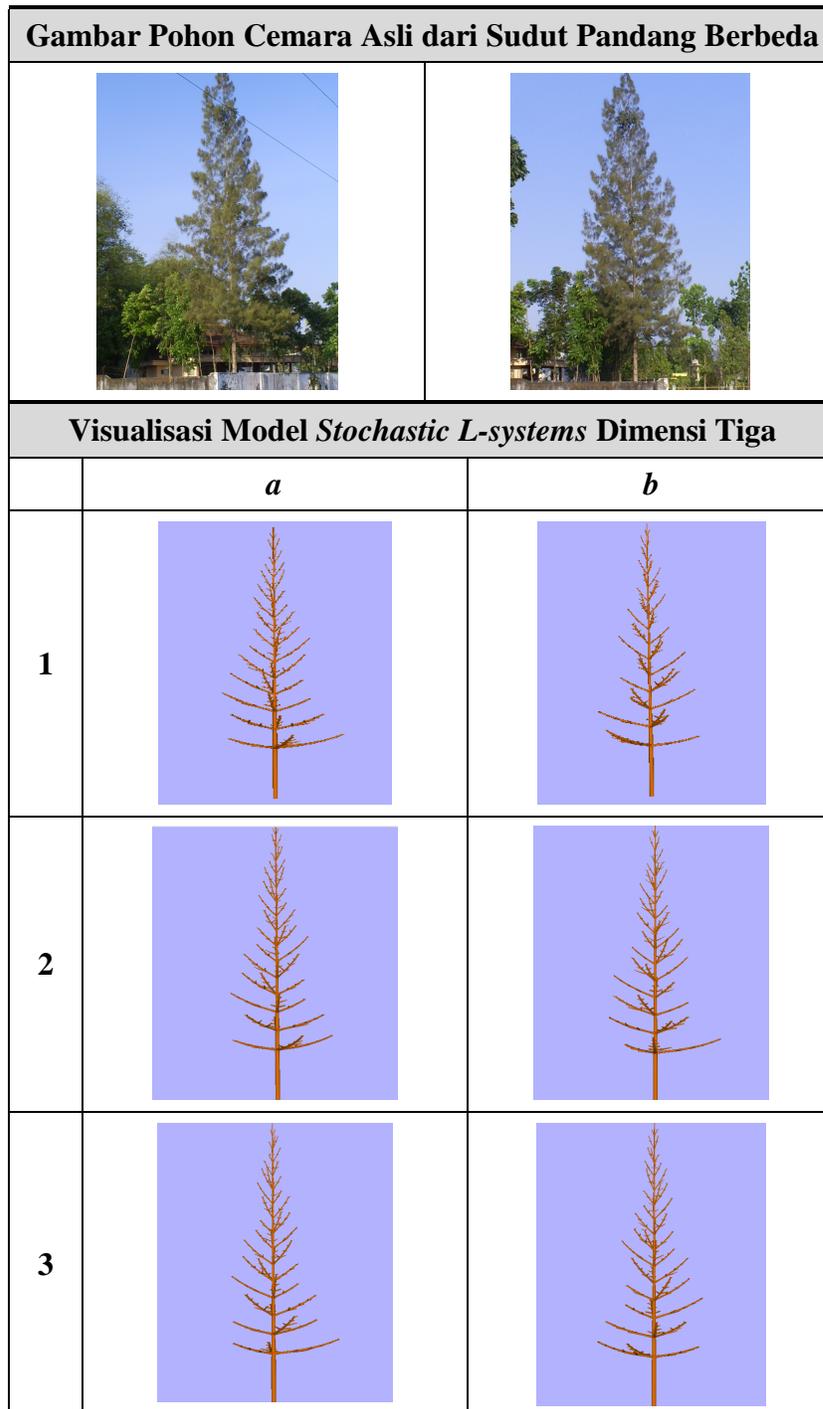
Pada komponen *L-systems* di atas, nilai awal panjang batang yang dimasukan adalah sebesar 1 sedangkan ketebalan batang awal yang dimasukkan adalah sebesar 15. Nilai probabilitas diberikan pada aturan produksi ketiga dan keempat dengan nilai probabilitas yang sama yaitu sebesar 0.5. Hal ini berarti bahwa setiap kali ditemukan huruf *C*, kemungkinan 50% aturan produksi ketiga akan digunakan dan 50% aturan produksi keempat yang digunakan. Pemberian nilai probabilitas yang sama terhadap kedua aturan produksi tersebut didasarkan pada pengamatan yang dilakukan oleh peneliti yang menilai bahwa percabangan yang terbentuk dari kedua aturan produksi tersebut memiliki kecenderungan muncul yang sama dalam foto objek. Jumlah iterasi yang digunakan yaitu sebanyak 16 iterasi karena pada iterasi ke-16 visualisasi gambar yang dihasilkan sudah menyerupai objek asli.

Hasil visualisasi parametrik *deterministic L-systems* dari komponen *L-system* di atas dapat dilihat pada Gambar 4.17. Gambar 4.17 (a) merupakan visualisasi model dengan putaran  $0^\circ$ . Sedangkan Gambar 4.17 (b) merupakan visualisasi model dengan putaran  $90^\circ$ .



(a) Visualisasi model dengan putaran  $0^\circ$ ; (b) Visualisasi model dengan putaran  $90^\circ$ ;  
Gambar 4.17 Parametrik *deterministic L-systems* dimensi tiga pada pohon Cemara

Sedangkan hasil visualisasi parametrik *stochastic L-systems* dari komponen *L-systems* di atas dapat dilihat pada Gambar 4.18. Hasil *running* pertama menghasilkan satu visualisasi gambar pohon yang ditunjukkan pada Gambar 4.18 (1). Gambar 4.18 (2) merupakan hasil *running* kedua yang juga menghasilkan 1 visualisasi dan berbeda dengan hasil *running* pertama. Dan Gambar 4.18 (3) merupakan hasil *running* ketiga.



( $a_1$ ) Hasil *running* ke-1 dengan putaran  $0^\circ$ ; ( $b_1$ ) Hasil *running* ke-1 dengan putaran  $90^\circ$ ;  
 ( $a_2$ ) Hasil *running* ke-2 dengan putaran  $0^\circ$ ; ( $b_2$ ) Hasil *running* ke-2 dengan putaran  $90^\circ$ ;  
 ( $a_3$ ) Hasil *running* ke-3 dengan putaran  $0^\circ$ ; ( $b_3$ ) Hasil *running* ke-3 dengan putaran  $90^\circ$ ;

Gambar 4.18 Parametrik *stochastic L-systems* dimensi tiga pada pohon Cemara

#### 4.4 Pembahasan

Pemodelan pertumbuhan batang tanaman menggunakan *stochastic L-systems* merupakan pengembangan dari *deterministic L-systems*. Dalam *stochastic L-systems* terdapat penambahan komponen faktor probabilitas pada struktur aturan produksinya. Oleh karena itu dalam penggunaannya diawali dengan menggunakan *deterministic L-systems*. Langkah awal yang penting dilakukan yaitu penentuan komponen utama *L-systems*, diantaranya penentuan huruf yang akan digunakan dalam aturan produksi. Huruf yang digunakan adalah huruf kapital  $A$  sampai  $Z$  kecuali huruf  $F$ ,  $f$ ,  $Z$ , dan  $z$  karena dalam program masing-masing huruf tersebut secara berturut-turut sudah ditetapkan untuk merepresentasikan gambar garis ke depan sebesar satu satuan, maju ke depan tanpa menggambar sebesar satu satuan, menggambar garis ke depan sebesar setengah satuan, maju ke depan tanpa menggambar sebesar setengah satuan.

Karena tanaman dalam pemodelan ini divisualisasikan dalam dimensi tiga maka hasil visualisasi ditampilkan dari dua sisi, yaitu sisi depan (putaran  $0^\circ$ ) dan sisi samping (putaran  $90^\circ$ ). Peneliti menilai bahwa hanya dengan memvisualisasikan dari dua sisi saja sudah cukup karena jika visualisasi dari depan (putaran  $0^\circ$ ) sudah menyerupai foto aslinya maka secara otomatis jika dilihat dari sisi belakang juga menyerupai foto aslinya. Demikian juga jika visualisasi dari sisi samping (putaran  $90^\circ$ ) sudah menyerupai foto aslinya maka secara otomatis jika dilihat dari sisi samping yang lain (putaran  $270^\circ$ ) juga menyerupai foto aslinya. Hasil visualisasi tersebut dapat dilihat pada Gambar 4.11, Gambar 4.13, Gambar 4.15, dan Gambar 4.17.

Setelah hasil visualisasi yang didapatkan sudah menyerupai objek yang didapat, langkah selanjutnya adalah perubahan *deterministic L-systems* menjadi *stochastic L-systems*. Nilai probabilitas hanya diberikan pada huruf (*predecessor*) yang sama. Besar nilai probabilitas yang diberikan berdasarkan perkiraan sering munculnya suatu aturan produksi dalam foto objek yang didapatkan, sehingga ada

kemungkinan akan terjadi perbedaan dalam pemberian nilai probabilitas antara peneliti dengan orang lain yang menggunakan *stochastic L-systems* ini.

Pemanfaatan komputer sangat membantu dalam proses visualisasi model tanaman, baik dalam hal proses rekursif dari aturan produksi yang sudah didefinisikan maupun pengacakan beberapa aturan produksi yang telah diberi nilai probabilitas. Meskipun demikian terdapat kekurangan ketika memasukkan iterasi dengan jumlah yang banyak (>20 iterasi) karena program akan berjalan lambat dalam memvisualisasikan model. Selain itu hasil yang didapat tidak menyerupai foto asli tanaman karena cabang yang terbentuk terlalu banyak sehingga tanaman tampak jauh lebih besar dan ranting yang terlalu lebat jika dibandingkan dengan foto asli tanaman tersebut. Untuk memodelkan pohon Trembesi pada Gambar 4.11, pohon Jung Rabab pada Gambar 4.13, pohon Cemara pada Gambar 4.15 dan 4.17 secara berturut-turut digunakan iterasi sebanyak 10 iterasi, 9 iterasi, 7 iterasi, dan 16 iterasi.

Pemodelan pertumbuhan batang tanaman menggunakan parametrik *deterministic L-systems* dapat menghasilkan visualisasi yang menyerupai foto objek asli (Gambar 4.11, Gambar 4.13, Gambar 4.15, dan Gambar 4.17). Sedangkan dengan menggunakan *stochastic L-systems* dapat memprediksi kemungkinan-kemungkinan model yang terbentuk dari foto objek asli. Hal ini dapat dilihat pada Gambar 4.12, Gambar 4.14, Gambar 4.16, dan Gambar 4.18. Dari Gambar 4.12 dapat diketahui bahwa visualisasi model 4.12 (1) berbeda dengan model 4.12 (2) dan model 4.12 (3). Hal ini dikarenakan untuk memodelkan Gambar 4.12 terdapat beberapa aturan produksi yang memiliki huruf (*predecessor*) yang sama yaitu A. Sehingga setiap ditemukan huruf A maka akan diacak dengan probabilitas aturan produksi pertama 40% dan aturan produksi kedua 60%. Proses ini juga berlaku pada Gambar 4.14, gambar 4.16, dan Gambar 4.18.

## **BAB 5. KESIMPULAN DAN SARAN**

### **5.1 Kesimpulan**

Berdasarkan hasil dan pembahasan dapat ditarik kesimpulan sebagai berikut:

- a. Jumlah iterasi untuk setiap pohon berbeda karena tergantung pada kompleksitas percabangan yang ada pada pohon.
- b. Dengan menggunakan *deterministic L-systems*, dapat dihasilkan visualisasi model yang menyerupai foto asli. Sedangkan *stochastic L-systems* dapat memprediksi kemungkinan-kemungkinan model yang terbentuk dari foto asli tanaman.

### **5.2 Saran**

Meskipun model tanaman yang dihasilkan melalui *Stochastic L-systems* sudah mendekati kenyataan, akan tetapi masih banyak terdapat permasalahan bagi pembaca yang ingin menyempurnakan pemodelan pertumbuhan batang tanaman menggunakan parametrik *Stochastic L-systems* ini.

Permasalahan tersebut antara lain:

- a. Pemodelan tanaman masih berupa batang tanaman dan tidak terdapat daun agar tanaman lebih terlihat nyata;
- b. Belum bisa menghasilkan visualisasi segmen batang yang melengkung seperti pada tanaman Bonsai.

## DAFTAR PUSTAKA

- Amir, F. 2009. *Penerapan Sensitive L-Systems Pada Pemodelan Pertumbuhan Batang Tanaman*. Tidak Diterbitkan. Skripsi. Jember: Fakultas MIPA Universitas Jember
- Dickau, R.M. 1996. *Two-Dimensional L-Systems*.  
<http://mathforum.org/advanced/robertd/lsys2d.html> [20 Pebruari 2011]
- Hanan, J.S. 1992. *Parametric L-Systems and Their Application to The Modelling and Visualization of Plants*. Disertasi. Regina: Computer Science University of Regina.
- Muzammil, M. 2006. *Pemodelan Pertumbuhan Batang Tanaman Menggunakan L-Systems*. Tidak Diterbitkan. Skripsi. Jember: Fakultas MIPA Universitas Jember
- Prusinkiewicz, P., & Lindenmayer, A. 1990. *The Algorithmic Beauty of Plants*. New York: Springer-erlag.
- Santoso, 2010. *Pemodelan Pertumbuhan Batang Tanaman dalam Dimensi Tiga Menggunakan L-Systems*. Tidak Diterbitkan. Skripsi. Jember: Fakultas MIPA Universitas Jember.
- Swandana, T.R. 2010. *Pemodelan Tanaman Tembakau Menggunakan L-Systems*. Tidak Diterbitkan. Skripsi. Jember: Fakultas MIPA Universitas Jember.
- Wright, D.J. 1996. *Dynamical Systems and Fractals Lecture*. <http://www.math.okstate.edu/mathdept/dynamics/lecnotes/lecnotes.html> [12 Agustus 2009]

### LAMPIRAN Script Program *L-systems*

```

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>

#include "lsystem.h"
#include "funcexpr.h"
#include "boolexpr.h"

LSystem::LSystem()
{
    m_Productions = NULL;
    m_NumProductions = 0;
    m_SizeProductions = 0;
    m_axiom[0] = '\0';
}

LSystem::~LSystem()
{
    if (m_Productions != NULL) free(m_Productions);
}

void
LSystem::Clear()
{
    m_NumProductions = 0;
}

bool
LSystem::AddProduction(const char *str)
{
    switch (str[0])
    {
        case '#':
        {
            return true;
        }
        case '$':
        {
            char *sstr = strstr(str, "=");

```

```

    if (sstr != NULL)
    {
        char name[128];
        char value[32];
        memset(name, 0, 128);
        memset(value, 0, 32);
        strncpy(name, &str[1], (sstr - str)-1);
        strcpy(value, sstr+1);
        return AddVariable(name, (float)atof(value));
    }
}
default:
{
    char *sstr = strstr(str, "-->");
    if (sstr != NULL)
    {
        char symbol[128];
        char production[1024];
        memset(symbol, 0, 128);
        memset(production, 0, 1024);
        strncpy(symbol, &str[0], (sstr - str));
        strcpy(production, sstr+3);
        return AddProduction(symbol, production);
    }
    else
    {
        if (str[0] == '@') strcpy(m_axiom, &str[1]);
    }
    return false;
}
}
return false;
}

bool
LSystem::AddProduction(const char *symbol, const char *production)
{
    int i;
    char *sstr = NULL;
    char *cond = NULL;
    sstr = strstr(symbol, ":");
    if (sstr != NULL) cond = sstr+1;

```

```

float probability = 1.0f;
char *prod = (char *)production;
i = 0;
if (prod[i] == '(')
{
    i++;
    char *endptr = NULL;
    probability = (float)strtod(&prod[1], &endptr);
    if (*endptr != ')') return false;
    prod = endptr + 1;
}

char variables[MAXPARAM][256];
char *vp[MAXPARAM];
for (int j=0; j<MAXPARAM; j++) vp[j] = variables[j];
memset(variables, 0, MAXPARAM*256);
int vi = 0;
i = 0;
char sym = symbol[i];
i++;
if (symbol[i] == '(')
{
    // parse parameter
    int i1 = i+1;
    while (symbol[i] != ')')
    {
        i++;
        switch (symbol[i])
        {
            case ',':
            case ')':
            {
                strncpy(variables[vi], &symbol[i1], i-i1);
                i1 = i+1;
                vi++;
                break;
            }
        }
    }
}
AddProduction(sym, cond, vp, vi, prod, probability);
return true;
}

```

```

bool
LSystem::AddProduction(const char symbol, const char *condition, char **variables,
int numvariables, const char *production, float probability)
{
    if (m_NumProductions >= m_SizeProductions)
    {
        m_SizeProductions += 4;
        m_Productions = (ProductionEntry *)realloc(m_Productions, m_SizeProductions
* sizeof(ProductionEntry));
    }
    ProductionEntry pe;
    pe.symbol = symbol;
    pe.probability = probability;
    pe.condition[0] = '\0';
    pe.production[0] = '\0';
    if (production != NULL) strcpy(pe.production, production);
    if (condition != NULL) strcpy(pe.condition, condition);
    pe.numparam = numvariables;
    for (int i=0; i<numvariables; i++) strcpy(pe.param[i], variables[i]);
    m_Productions[m_NumProductions] = pe;
    m_NumProductions++;

    return true;
}

bool
LSystem::AddVariable(const char *name, float value)
{
    globalfe.SetVariable(name, value);
    return true;
}

bool
LSystem::GetVariable(const char *name, float &value)
{
    return globalfe.GetVariable(name, value);
}

bool
LSystem::GetProduction(char symbol, int num, ProductionEntry &pe)
{
    for (int i=0; i<m_NumProductions; i++)

```

```

    {
        if ((m_Productions[i].symbol == symbol) && (m_Productions[i].numparam ==
num))
        {
            pe = m_Productions[i];
            return true;
        }
    }
    return false;
}

```

bool

LSystem::GetProduction(char symbol, int num, ProductionEntry &pe,  
FuncExpression \*fe)

```

{
    float accu = 0;
    float rnd = (float)(((double)rand())/RAND_MAX);
    BoolExpression be(fe);
    for (int i=0; i<m_NumProductions; i++)
    {
        if ((m_Productions[i].symbol == symbol) && (m_Productions[i].numparam ==
num))
        {
            accu += m_Productions[i].probability;
            if (accu > rnd)
            {
                pe = m_Productions[i];
                if (m_Productions[i].condition[0] == '\0') return true;
                if (be.Evaluate(m_Productions[i].condition) != 0) return true;
            }
        }
    }
    return false;
}

```

bool

LSystem::Parse(int inpos, int inlen, const char \*in, CharBuf \*ocb)

```

{
    // production: F(x,y,z) --> A(x)B(y)C(z)
    // production: D(u,v) --> E(v,u)
    // in .... F(1,2,3)D(4,5)
    // out ... A(1)B(2)C(3)E(5,4)
    float values[MAXPARAM];

```

```

int vi;
int i0;

int i=inpos;
while (i < inlen)
{
    i0 = i;
    vi = 0;
    char symbol = in[i];
    i++;
    if (in[i] == '(')
    {
        // parse parameter
        int i1 = i+1;
        while ((i < inlen) && (in[i] != ' '))
        {
            i++;
            switch (in[i])
            {
                case ',':
                case ')':
                {
                    char buf[32];
                    memset(buf, 0, 32);
                    strncpy(buf, &in[i1], i-i1);
                    if (!GetVariable(buf, values[vi])) values[vi] = (float)atof(buf);
                    vi++;
                    i1 = i+1;
                    break;
                }
            }
        }
        i++;
    }

    if (!EvalProduction(symbol, values, vi, ocb))
    {
        // for (int j=i0; j<i; j++) out[outpos++] = in[j];
        for (int j=i0; j<i; j++) ocb->Put(in[j]);
    }
}
return true;
}

```

```

bool
LSystem::EvalProduction(char symbol, float *values, int numvalues, CharBuf *ocb)
{
    // production: F(x,y,z) --> A(x)B(y)C(z)
    // in ... F, [1,2,3]
    // out .. A(1)B(2)C(3)

    ProductionEntry pe;
    if (!GetProduction(symbol, numvalues, pe)) return false;

    int i;
    FuncExpression fe;

    for (i=0; i<numvalues; i++)
    {
        fe.SetVariable(pe.param[i], values[i]);
    }

    for (i=0; i<globalfe.GetNumVariables(); i++)
    {
        float value;
        char *name;
        globalfe.GetVariable(i, name, value);
        //MessageBox(0,name,name,MB_OK);
        fe.SetVariable(name, value);
    }

    if (!GetProduction(symbol, numvalues, pe, &fe)) return false;

    const char *in = pe.production;
    int inlen = strlen(in);

    i=0;
    while (i < inlen)
    {
        char symbol = in[i];
        // out[outpos++] = symbol;
        ocb->Put(symbol);
        i++;
        if (in[i] == '(')
        {
            i++;

```

```

//    out[outpos++] = '(';
ocb->Put('(');
// parse parameter
int i1 = i;
while ((i < inlen) && (in[i] != '))
{
    if (in[i] == '(')
    {
        int b=0;
        do
        {
            if (in[i] == '(') b++;
            if (in[i] == ')') b--;
            i++;
        } while ((b>0) && (i < inlen));
    }
    else
    {
        i++;
    }
    switch (in[i])
    {
        case ',':
        case ')':
        {
            char buf[128];
            memset(buf, 0, 128);
            strncpy(buf, &in[i1], i-i1);
            float f = fe.Evaluate(buf);
            i1 = i+1;
            sprintf(buf, "%f", f);
            for (int j=0; j<(int)strlen(buf); j++) ocb->Put(buf[j]);
//            for (int j=0; j<(int)strlen(buf); j++) out[outpos++] = buf[j];
//            out[outpos++] = in[i];
            ocb->Put(in[i]);
            break;
        }
    }
}
i++;
}
return true;

```

```
}

bool
LSystem::Generate(int n, char *in, CharBuf *ocb)
{
    if (ocb == NULL) return false;
    char *ibuf = NULL;
    ocb->Put(in, strlen(in));
    if (m_axiom[0] != '\0')
    {
        ocb->Put(m_axiom, strlen(m_axiom));
    }
    for (int i=0; i<=n; i++)
    {
        ibuf = (char *)realloc(ibuf, ocb->GetSize());
        strncpy(ibuf, ocb->GetBuf(), ocb->GetSize());
        ocb->Clear();
        Parse(0, strlen(ibuf), ibuf, ocb);
        ocb->Put('\0');
    }
    if (ibuf != NULL) free(ibuf);
    return true;
}
```