

PEMBUATAN *RULE L-SYSTEM* MENGGUNAKAN ALGORITMA *CANNY EDGE DETECTION*

PRODUCTION L-SYSTEM RULE WITH EDGE DETECTION

Agita Chrisna¹, Purba Daru Kusuma, Dr., S.T., M.T.², Anton Siswo Raharjo Ansori, S.T., M.T.³

¹²³Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom

¹agita.chrisna99@gmail.com, ²purbodaru@gmail.com, ³raharjo@telkomuniversity.ac.id

ABSTRAK

Pada tahun 1968, Aristid Lindenmayer memperkenalkan *Lindenmayer system* yang sering disebut *L-system*. *L-system* adalah bahasa formal yang disusun dengan sistem penulisan ulang untuk menggambarkan fraktal, memodelkan, dan mensimulasikan perkembangan tanaman dalam bidang dua dimensi. Dengan mendefinisikan simbol, parameter, aksiom, aturan pertumbuhan (*rule*). Tugas akhir ini memaparkan pembuatan *rule L-system* menggunakan algoritma *edge detection*. Tanaman yang diteliti aturan produksinya adalah tanaman kacang kedelai.

Tanaman tersebut akan diambil gambar percabangan batangnya minimal saat tanaman tersebut memasuki fase berbunga. Gambar yang telah diambil akan diproses dengan algoritma *edge detection* agar hanya bagian batangnya saja yang akan diteliti aturan produksinya. Lalu, ditelusuri percabangan batangnya untuk mendapatkan aturan produksi dari tanaman tersebut. Hasil dari tugas akhir ini akan memberikan data analisis aturan (*rule*) *L-system* dari percabangan batang kacang kedelai tersebut dan memodelkannya.

Kata kunci: *L-system, canny edge detection, python, kamera, kacang kedelai*

ABSTRACT

In 1968, Aristid Lindenmayer introduced *Lindenmayer system* are often called *L-system*. *L-system* are formal languages that are arranged with a rewrite to describe the system of fractal, model, and simulate the development of plant in two dimensional field. By defining parameters, symbols, axioms, rules of growth. This final project presents the making of the *rule L-system* using the algorithm of *edge detection*. The plant which examined the rules of production was soy bean plant.

The plant will be taken pictures of its trunk branching is minimal when the plant enter a phase of flowering. Pictures that have been taken will be processed with *edge detection* algorithm for that only the stalks are meticulous in production rules. Then, traced branching stems to get the production rules from those plant. The result of the final project will provide data analysis of the *rule L-system* of branching stems the soy beans and modeling it.

Keywords: *L-system, canny edge detection, python, camera, soybeans*

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi saat ini sudah semakin pesat, terutama dalam bidang dunia digital. Salah satunya dengan adanya pengolahan citra. Pengolahan citra merupakan teknik pengolahan dan analisis citra yang dilihat dari sudut pandang visual. Pengolahan citra tersebut akan mengolah suatu gambar yang diambil secara dua dimensi dan melakukan transformasi citra menjadi citra digital dengan menggunakan teknik tertentu.

Pengolahan citra dapat dimanfaatkan sebagai pengenalan pola pada perkembangan tanaman. Setiap jenis tanaman mempunyai ciri khas masing-masing, walaupun jenis yang sama setiap fase pertumbuhan tanaman tersebut tidak sama persis satu dengan lainnya. Jadi, setiap tanaman mempunyai aturan pertumbuhan sendiri.

Pada tahun 1968, Aristid Lindenmayer memperkenalkan *Lindenmayer system* yang sering di sebut *L-system*. *L-system* adalah bahasa

formal yang disusun dengan sistem penulisan ulang untuk menggambarkan fraktal, memodelkan, dan mensimulasikan perkembangan tanaman dalam bidang dua dimensi. Dengan mendefinisikan simbol, parameter, aksiom, dan aturan pertumbuhan (*rule*). Dikarenakan hal tersebut, maka dilakukan penelitian untuk pembuatan *rule L-System* menggunakan algoritma *edge detection* agar dapat membantu memberikan informasi data aturan pertumbuhan tanaman. Jadi, manfaat dari hasil penelitian ini akan membuat suatu model pertumbuhan tanaman kacang kedelai dari *rule L-System* yg didapatkan dari tanaman tersebut. [10]

1.2 Rumusan Masalah

Dalam penelitian pembuatan *rule L-System* ini mengacu pada latar belakang. Oleh karena itu, ada beberapa masalah dalam penelitian ini, yaitu :

- Bagaimana software dapat membuat *rule L-System* tanaman tersebut?
- Bagaimana mengolah gambar tanaman dengan menggunakan algoritma *canny edge detection* agar hanya mengambil bagian percabangan batang saja dari gambar tanaman tersebut?
- Bagaimana algoritma *canny edge detection* dapat melacak pola pergerakan dari percabangan batang pada tiap piksel gambar tanaman?
- Bagaimana penulisan ulang tanaman dari masukan *rule L-system*?

1.3 Tujuan Penelitian

Mengacu pada rumusan masalah yang ada, maka tujuan dari pembuatan tugas akhir ini adalah:

- Dapat memberikan data aturan pertumbuhan (*rule L-System*) dari tanaman yang diteliti.
- Mengimplementasikan aplikasi yang dapat mendeteksi tepi gambar.
- Menganalisis untuk mengetahui pertumbuhan *rule L-system* pada tanaman tersebut.
- Membuat model tanaman tersebut dari *rule L-System*.

1.4 Batasan Masalah

Batasan masalah dalam pengerjaan tugas akhir ini adalah sebagai berikut :

- Aplikasi berjalan pada sistem operasi *desktop*.
- Aplikasi diimplementasikan pada tanaman kacang kedelai.
- Mengambil gambar dari satu sisi saja.
- Hanya mengambil bagian percabangan batang saja (tanpa daun dan buah).
- Tanaman yang akan diteliti adalah tanaman kacang kedelai.
- Tanaman memiliki percabangan batang 5 cabang atau lebih.

II. DASAR TEORI

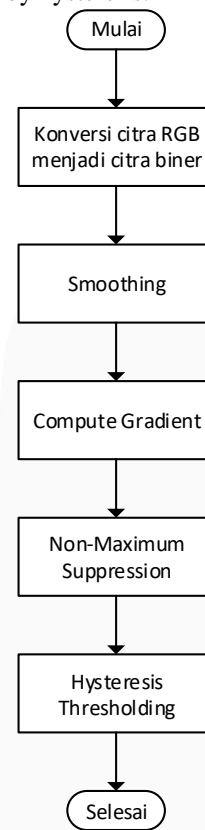
2.1 Lindenmayer System (*L-System*)

Lindenmayer system atau sering disebut *L-system* adalah bahasa formal yang disusun dengan sistem penulisan ulang untuk menggambarkan fraktal, memodelkan, dan mensimulasikan perkembangan tanaman dalam bidang dua dimensi. Dengan mendefinisikan simbol, parameter, aksiom, aturan pertumbuhan (*rule*). [10]

2.2 Canny Edge Detection

Algoritma *canny edge detection* merupakan salah satu teknik *edge detection* yang cukup populer didalam pengolahan citra. Salah satu alasannya adalah karena ketebalan *edge* yang bernilai satu piksel ini. Algoritma ini dapat digunakan untuk mendeteksi semua bagian *edge* atau garis-garis yang membentuk objek gambar. Citra RGB dari sebuah gambar akan diubah menjadi citra biner.[6]

Dengan menggunakan algoritma ini, suatu ciri-ciri tanaman dapat dikenali. Beberapa langkah untuk melakukan proses algoritma ini yaitu *image smoothing*, *finding gradient*, *non-maximum suppression*, *double thresholding*, dan *edge tracking by hysteresis*.

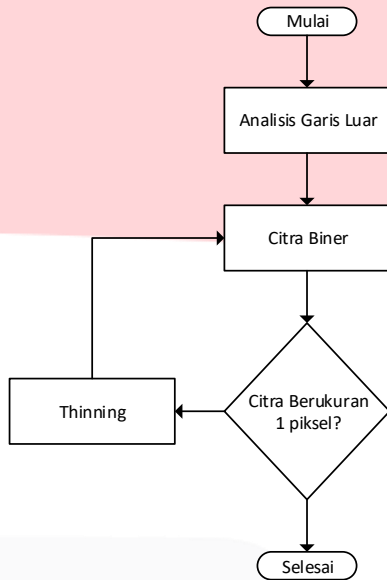


Gambar 2.1 Diagram Alir *Canny Edge Detection*

2.3 Thinning

Thinning merupakan metode yang digunakan untuk *skeletonizing* yang salah satu manfaat kegunaannya dalam adalah dalam aplikasi pengenalan pola. Banyak sekali algoritma untuk *thinning* suatu citra dengan tingkat kompleksitas, efisiensi, dan akurasi yang berbeda-beda. Citra yang digunakan adalah citra biner yang hanya bernilai 0 atau 1. Nilai 0 adalah *background points*, biasanya bukan merupakan bagian dari sesungguhnya. Sedangkan nilai 1 adalah *region points*, yaitu bagian dari citra

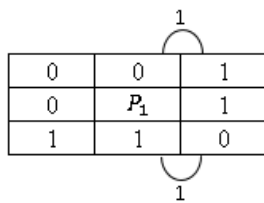
sebenarnya. Hasil citra dari algoritma *thinning* seringkali disebut dengan *skeleton* (terlihat seperti kerangka). Metode *thinning* yang digunakan yaitu algoritma *thinning* Zhang-Suen. Keuntungan metode ini memiliki komputasi waktu cepat.



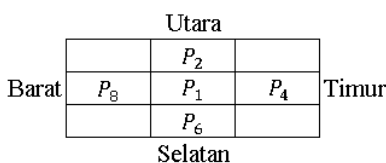
Gambar 2.2 Diagram Alir *Thinning*

Algoritma ini merupakan algoritma iteratif dan menghapus semua titik luar gambar kecuali ini menjadi kerangka. Metode ini digunakan untuk ekstraksi kerangka citra terdiri dari menghapus semua poin kontur gambar kecuali poin milik kerangka. Untuk mempertahankan kerangka citra maka proses ini akan membagi setiap iterasi menjadi dua-subiterasi. Pada subiterasi pertama, poin kontur P_1 dihapus dari pola jika memenuhi kondisi:

- $2 \leq B(P_1) \leq 6$
- $A(P_1) = 1$
- $P_2 * P_4 * P_6 = 0$
- $P_4 * P_6 * P_8 = 0$



Gambar 2.3 Hitung Pola 01



Gambar 2.4 Poin Yang Akan Dipertimbangkan dan Lokasi

Dari kondisi diatas, $A(P_1)$ adalah nomor dari pola 01 kemudian menentukan P_2, P_3, \dots, P_8 merupakan 8 tetangga dari P_1 seperti pada gambar 2.3. Nilai $B(P_1)$ merupakan jumlah tetangga tidak nol dari P_1 , adalah:

$$B(P_1) = P_2 + P_3 + P_4 + \dots + P_9$$

Jika semua kondisi tidak terpenuhi, maka nilai $P_2 + P_3 + P_4 + \dots + P_9$ adalah $A(P_1) = 2$, dan P_1 tidak dihapus dari citra.

Pada iterasi kedua, hanya kondisi (c) dan (d) yang berubah seperti pada gambar 2.4 berdasarkan:

- $P_2 * P_4 * P_6 = 0$
- $P_4 * P_6 * P_8 = 0$

Proses ini dilakukan berulang hingga tidak ada lagi poin yang akan dihapus. Berdasarkan kondisi (c) dan (d) pada subiterasi pertama hanya menunjukkan bahwa hanya menghilangkan poin bagian titik batas selatan-timur dan sudut utara-barat poin yang bukan kerangka ideal. [13]

III. PERANCANGAN DAN IMPLEMENTASI

3.1 Deskripsi Umum Sistem

Pada penelitian ini akan dirancang aplikasi yang dapat menggambar ulang gambar percabangan batang suatu tanaman menggunakan metode *L-system*. Tanaman yang digunakan adalah kacang kedelai minimal fase berbunga. Dengan menggunakan bahasa pemrograman *python* versi 2.7.5 dan metode *L-system* maka dapat menggambar ulang pada percabangan batang tanaman kacang kedelai sehingga arah jalur percabangannya akan ditentukan secara otomatis.

3.2 Sistem Alur Kerja

Alur sistem pembuatan *rule L-system* percabangan batang tanaman kacang kedelai akan dijelaskan sebagai berikut. Tahap 1 pengambilan gambar tanaman kacang kedelai menggunakan kamera smartphone dengan jarak ± 40 cm serta sudut pandang pola percabangan batang tanaman terlihat jelas yang menghasilkan citra RGB. Tahap 2 Deteksi Tepi dilakukan untuk mendapatkan hasil tepi dari citra tersebut dan mengubah citra RGB (.jpeg) menjadi citra biner(.bmp). Tahap 3 Penebalan dilakukan untuk menghubungkan piksel percabangan batang yang tidak saling terhubung karena saat pengolahan deteksi tepi terkadang ada beberapa bagian piksel percabangan yang terputus. Karena menyimpan gambar yang berada di kanvas *turtle python* maka tipe data citra ini akan menjadi .ps lalu dikonversi

menjadi .jpeg menggunakan photoshop. Tahap 4 *Thinning* dilakukan untuk mengecilkan piksel citra sebelumnya yang tebal agar berukuran satu piksel sehingga memudahkan dalam tahap pembuatan *rule L-system*. Tipe data citra ini diubah dari .jpeg menjadi .bmp. Tahap 5 Pembuatan *Rule L-system* dilakukan untuk mencari pola pertumbuhan tanaman secara otomatis dengan menggunakan metode *L-system*. Tahap 6 Penulisan Ulang *Rule L-system* dilakukan untuk melihat akurasi penggambaran ulang tanaman menggunakan *rule L-system* tersebut.

3.3 Perancangan Sistem

Perancangan sistem dilakukan karena aplikasi ini terdiri dari beberapa rancangan sistem. Oleh karena itu, untuk menunjang jalannya aplikasi maka dibutuhkan perangkat keras dan lunak sebagai penunjang.

a. Pengambilan Gambar

Dengan menggunakan kamera smartphone Lenovo a6000 berukuran 8MP untuk mengambil gambar percabangan batang tanaman kacang kedelai sebagai input pertama kali. Hasil dari pengambilan gambar berupa citra RGB berekstensi JPEG. Lalu citra tersebut selanjutnya akan diproses menggunakan laptop Asus a4551 dengan CPU Intel Core i5-5200u @2.20GHz dan 4GB RAM menggunakan bahasa pemrograman python.

b. Canny Edge Detection

Deteksi tepi *canny* digunakan untuk mengonversi citra RGB menjadi citra biner. Setelah itu akan dideteksi bagian tepi gambar dengan.

c. Penebalan

Hasil dari deteksi tepi *canny*, akan dilakukan penebalan. Karena ada beberapa piksel yang tidak saling terhubung. Penebalan dilakukan dengan menggunakan fitur *turtle* pada *python*. *Turtle* akan menggambar titik berukuran 10 piksel pada tiap koordinat piksel yang terdeteksi.

d. Thinning

Thinning dilakukan untuk mengecilkan piksel dari hasil penebalan. *Thinning* akan menghasilkan citra yang berukuran 1 piksel yang terlihat seperti kerangka.

e. Pembuatan *Rule L-system*

Setelah mendapatkan hasil *thinning*, maka selanjutnya dilakukan pembuatan *rule L-system*. Input awal adalah koordinat piksel putih paling bawah. Lalu dengan menggunakan rumus pencarian tetangga piksel putih untuk menentukan arah jalur pertumbuhan percabangan batang tanaman kacang kedelai.

Jika:

- Tetangga piksel berada di kiri, maka menghasilkan output --F++
- Tetangga piksel berada di kiri atas, maka menghasilkan output -F+
- Tetangga piksel berada di atas, maka menghasilkan output F
- Tetangga piksel berada di kanan atas, maka menghasilkan output +F-
- Tetangga piksel berada di kanan, maka menghasilkan output ++F--

f. Penulisan Ulang *Rule L-system*

Hasil dari output pembuatan *rule L-system* akan dilakukan penulisan kembali *rule L-system* untuk melihat pola percabangan batang tanaman kacang kedelai dengan syarat sebagai berikut.

- d : 45°
- F : bergerak ke depan sepanjang satu satuan l dan menggambar garis;
- $+$: berbelok kekanan dengan sudut d ;
- $-$: berbelok kekirinya dengan sudut d .
- $[$: menyimpan nilai posisi dan sudut *turtle* saat ini
- $]$: kembali ke nilai posisi dan sudut *turtle* sebelumnya.

BAB IV PENGUJIAN DAN ANALISIS

4.1 Parameter Pengujian

Pengujian dilakukan dengan menguji kinerja dari sistem, maka dibutuhkan beberapa parameter pengujian untuk mengetahui performa dari setiap aspek yang diuji. Dengan nama gambar.jpg yang berukuran 528x3036 piksel sebagai input pertama maka akan dilakukan pemrosesan *canny edge detection* untuk mendeteksi tepi citra dan melakukan pengecilan citra sebesar 0,175 kali ukuran citra original yang menjadi 92x531 piksel dan tipe data citra tersebut menjadi bmp. Ukuran citra yang diperkecil tersebut dilakukan agar citra dapat dilakukan pengujian didalam kanvas *turtle python*. Berikut adalah beberapa pengujian yang dilakukan.

4.2 Pengujian Penebalan




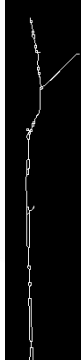
Pengujian ini bertujuan untuk mengetahui apakah program dapat menebalkan gambar piksel yang terputus dengan mengganti ukuran dot piksel pada *turtle* dari ukuran 1 sampai 25. Hasil dari pengujian ini dapat disimpulkan bahwa semakin besar nilai *dot turtle* maka akan menghasilkan gambar percabangan batang yang lebih tebal juga. Hasil pengujian dan analisis data dilihat pada lampiran A.


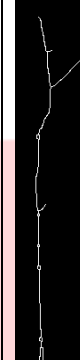






Dilihat dari no. 1 - 3, hasil penebalan masih kurang tebal, karena belum menutupi bagian gambar percabangan batang. Sehingga pada saat *men-thinning*, menghasilkan gambar percabangan batang yang memiliki 2 lajur piksel percabangan batang.







Dilihat dari no. 4 - 6, hasil penebalan sudah menutupi bagian percabangan batang sehingga saat *men-thinning* dapat menghasilkan gambar yang hanya memiliki 1 lajur piksel percabangan batang.

Dilihat dari no. 7 - 9, hasil penebalan sudah menutupi bagian percabangan batang tetapi saat *men-thinning* hanya tersisa 2 cabang, karena cabang lain yang kecil sudah tertutup oleh *dot turtle* yang lebih tebal. Sehingga cabang terlihat mendekati seperti bagian batang saat dilakukan proses *thinning*.

Tabel 4.1 Ukuran *Dot Turtle* Untuk Penebalan dan *Thinning*

| No. | Ukuran Dot Turtle | Hasil Penebalan | Hasil Thinning |
|-----|-------------------|---|---|
| 1. | 1 |  |  |
| 2. | 2 |  |  |



| | | | |
|----|---|---|---|
| 3. | 3 |  |  |
| 4. | 4 |  |  |
| 5. | 5 |  |  |
| 6. | 6 |  |  |

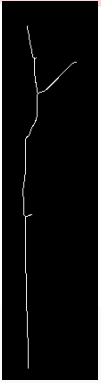
| | | | |
|----|----|--|--|
| 7. | 12 |  |  |
| 8. | 13 |  |  |
| 9. | 14 |  |  |

4.3 Pengujian Pembuatan Rule L-system

Pengujian ini bertujuan untuk mengetahui apakah program dapat membuat *rule L-system* dengan menggunakan keluaran gambar dari *thinning*.

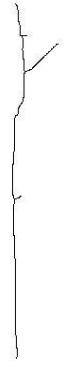


| No. | Thinning | Pembuatan Rule L-system |
|-----|----------|-------------------------|
|-----|----------|-------------------------|

| | | |
|----|--|--|
| 1. |  | <pre> F+F-FFFFFFFFF-F+FFFFFFF FFFFFFF-F+FFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFF F-F+FFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFF FF-F+FFFFFFFFFFFFFFFFFFFF FFFFFFF-F+FFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFF+F- FFFFF[+F--+F--+F--+F--+ ++F--F++F--F++F--F+F-][-F+ FFF-F+FFFFFFFFFFFFFFFFF- F+FFFFFFFFFFFFFFFFFFFFF- FFFFFFFFFFFFFFFFFFFFF+F-FFFFF FFFFFFFFFFFFFFFFFFFFFFF FFFFFFF+F-F+F--+F--+F--+F- FFF+F-FF+F-FF++F--F+F+F+ +F--F+F-F+F-FF++F--FFFFFFF FFFFFFFFFFFFFFFFFFFFF+F- FF[+F--+F--+F--+F--+F--+ +F--+F--+F--+F--+F--+F--+ +F--F++F--F++F--F++F--F+ +F--F++F--F++F--F++F--F++F --F++F--F++F--F++F--F++F- +F--+F--+F--+F--+F--+F--+ +F--+F--+F--+F--+F--+F--+ ++F--+F--+][-F+FFFFFFF- F+FFFFF-F+FFFFFFFFF- F+F+F-FFFFFFFFF-F+FF[-- F++-F+FFFFFFF-F+FFF- F+FFF-F+FFFFFFF-F+FFFF- F+FFFF-F+F-F+F][+F--+F--+ ++F--+F--+F--+F--+F--+] </pre> |
| 2. |  | <pre> FFFFFFFFF-F+FFFFFFFFF FFFFFFFFFFFFFFFFF-F+FFFFF FFFFFFFFFFFFFFFFFFFFFFF F-F+FFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFF- F+FFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFF F[+F--+F--+F--+F--+F--+F- --+F-F+F-][-F+ F+FFFFFFF FFFFFFFFF-F+FFFFFFFFF FFFF+F-FFFF++F--FFFFFFF +F-FFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFFFFFFF+ +F--F++F--+F--+F--+F-FF+F- FFF++F--FF+F-F++F--F+F- F+F--+F-FF+F-FFFFFFFFF FFFFFFFFFFFFFFFFF+F-FFF[+ +F--+F--+F--+F--+F--+F--+ +F--+F--+F--+F--+F--+F--+ +F--+F--+F--+F--+F--+F--+ -F+F--+F--+F--+F--+F--+F- +F--+F--+F--+F--+F--+F- +F--+F--+F--+F--+F--+F- +F--+F--+F--+F--+F--+F- F+FFFFFFFFF-F+FFFFF-F+F FFFF-F+FFFFFFFFFFFFFFF </pre> |

| | | |
|----|--|---|
| | | FFF[++F---F--F-][F+-- F++FFFFF-F+FFFFF-F+FFF- F+FFFF-F+FFFFF-F+FFFF-- F++FF-F+-F+] |
| 3. |  | FFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF-F+FFFFFFFFFF FFFFFFFFFFFFFFFFFFFFFFFF- F+FFFFFFFFFFF FFFFFFFFFFFFFFFF- F+FFFFFFFFFFF FFFFFFFFFFFFFFFF[++F--++F--F--++F--++F-- F++F--++F--][F+- F+FFFFFFFFFFF F+FFFFFFFFFFF-F-FFFFF+F- FFFFFFFF+F- FFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF+F- F++F--F++F--F-F+F-FF+F- FF+F-F+F-F++F--F+F-F+F- FF++F--FFF+F- FFFFFFFFFFFFFFFF FFF+F-F[++F---F-++F---F-- +F++F---++F---+F++F-- F++F--F++F--F++F--F++F-- F++F--F++F--F++F--F++F-- F++F--F++F--F++F--F++F-- F++F--F++F--F++F--F++F-- +F++F--F++F--F++F--F++F-- +F++F--F++F--F++F--F++F-- +F++F--F++F--F++F--F++F--][F- F+FFFFF-F+FFFFF-- F++FFFFF- F+FFFFFFFFFFF[F --F++F+FFF-F+FFFFF- F+FFF-F+FFFFF-F+FFFFF- F+FFF-F+FFFFF][++F--F-] |

4.4 Pengujian Penulisan Rule L-system

Pengujian ini bertujuan untuk mengetahui apakah program dapat menulis ulang *rule L-system* menggunakan keluaran dari pembuatan *rule L-system*. Hasil pengujian saat menggambar ulang *rule L-system* sudah dapat menggambar ulang bagian percabangan.

| No. | Pembuatan Rule L-system | Penulisan Rule L-system |
|-----|-------------------------|---|
| 1. | 1. |  |
| 2. | 2. |  |
| 3. | 3. |  |

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan dari penelitian tugas akhir ini adalah sebagai berikut.

- 1) Penebalan bagian piksel dilakukan secara 25 kali dengan mengganti ukuran *dot turtle* dari 1 – 25. Penebalan menggunakan ukuran dot turtle dari 4 – 11 adalah penebalan yang baik, karena menghasilkan gambar yang sudah menutupi bagian percabangan batang sehingga saat melakukan proses *thinning* hanya memiliki 1 lajur piksel.
- 2) Dari 25 gambar hasil *thinning* dengan ukuran *dot turtle* yang berbeda, tingkat keberhasilan program dapat membaca

bagian percabangan batang tanaman dengan baik yaitu 88% karena pada table hasil penulisan ulang *rule L-system* di lampiran A tidak sesuai dengan hasil *thinning* pada no. 6, 11, dan 15.

- 3) Pola pertumbuhan tanaman dapat dianalisis menggunakan *L-system*.

5.2 Saran

Hasil dari penelitian tugas akhir ini masih terdapat beberapa kekurangan. Oleh karena itu, diperlukan saran untuk pengembang lebih lanjut adalah sebagai berikut.

- 1) Pembuatan *rule L-system* untuk percabangan batang yang mempunyai 8 tetangga piksel, yaitu tetangga kiri, kiri atas, atas, kanan atas, kanan, kanan bawah, bawah, dan kiri bawah.
- 2) Menambah model percabangan batang tanaman lain.

DAFTAR PUSTAKA

- [1] Aborisade, D.O., (2011). Novel Fuzzy Logic Based Edge Detection Technique. *International Journal of Advanced Science and Technology*, 29, 76-78.
- [2] Alshennawy, Abdallah A., & Aly, Ayman A. (2009). Edge Detection in Digital Images Using Fuzzy Logic Technique. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 3(3), 540-546.
- [3] Bowo, S.A.A., Hidayatno, A., Isnanto, R.R. (2010). Analisis Deteksi Tepi Untuk Mengidentifikasi Pola Daun. *Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro*, 1, 1-5.
- [4] Cai, L., Ma, Y., Yuan, T, Wang, H., Xu, T. (2015). An Application of Canny Edge Detection Algorithm to Rail Thermal Image Fault Detection. *Journal of Computer and Communications*, 3, 20-24.
- [5] Fakhrine, F.A., Rahmadwati, Wijono. (2016). Thinning Zhang-Suen Dan Stentiford Untuk Menentukan Ekstraksi Ciri (Minutiae) Sebagai Identifikasi Pola Sidik Jari Whorl Dan Loop. *Majalah Ilmiah Teknik Elektro*.128.
- [6] Kurniawati, I.D., Kusumawardhani, A. (2017). Implementasi Algoritma Canny dalam Pengenalan Wajah menggunakan Antarmuka GUI Matlab. *Institution of Engineering and Technology*.1-3.
- [7] Liu, T. S., Liu, R.X., Zeng, P., Pan, S.W. (2014). Improved Canny Algorithm for Edge Detection of Core Image. *The Open Automation and Control Systems Journal*, 6, 426-432.
- [8] Liu, W., Hildebrandt, T. H., Cavin, R. (1989). Hamiltonian Cycles in the Shuffle-Exchange Network. *IEEE Transactions On Computers*, 38 (5), 741-745.
- [9] Madan, L., Anand, K., Bhushan, B. (2014). Bresenham's Lines Algorithm. *International Journal of Research in Science And Technology*, 4, 55-62.
- [10] Prusinkiewicz, Przemyslaw & Lindenmayer, Aristid. (1990). *The Algorithmic Beauty of Plants*. Regina, Canda : Springer Verlag. 6-11.
- [11] Rong, W., Li, Z., Zhang, W., Sun, L. (2014). An Improved Canny Edge Detection Algorithm. *International Conference on Mechatronics and Automation*, 1, 577-582.
- [12] Sun, R., Jia, J., Jaeger, M. (2009). Intelligent Tree Modeling Based on L-system. *IEEE*, 1, 1096-1100.
- [13] Zurnawita, Suar, Z. (2009). Algoritma Image Thinning. *Staf Pengajar Teknik Elektro Politeknik Negeri Padang*. 31-33.