Studia Universitatis "Vasile Goldiş", Seria Ştiinţele Vieţii
Vol. 22, issue 3, 2012, pp. 411-418

INTEGRATION OF ARTIFICIAL NEURAL NETWORKS INTO AN LINDENMAYER SYSTEM BASED PLANT MODELING ENVIRONMENT WITH MATHEMATICA

SUHARTONO[1], Mochammad HARIADI[2], Mauridhi Hery PURNOMO[2]
[1]State Islamic University of Malang, Indonesia
[2]Institute Technology Sepuluh Nopember Surabaya, Indonesia

ABSTRACT. Modeling of the plant growth can be visualized using the approach of Lindenmayer System (L-System). To make the condition close to the real environment characteristic, it is required the axiom and syntax grammar for the L-System. In this paper, we propose the use of artificial neural networks together with the L-System method, to model the plant's growth based on the current environment condition. At the beginning of plant growth, let the sprout of plant initially be denoted as axiom. This characteristic rules are illustrated in the reproduction of L-System also occur in nature of plant growth conditions based on the artificial neural networks. The software used in this modeling is Mathematica. In this research, the growth parameters value is given based on the artificial neural networks, the plant growth is visualized with the L-System method, and the 3-Dimension graph is shown as the virtual plant growth.

Keywords: L-System, modeling, artificial neural networks, virtual plant growth, visualization

INTRODUCTION

The modeling of plant growth is initialized by Aristid Lindenmayer, when he introduced the theory of anabaena catenula cell growth using the method called rewriting string. Later on, this rewriting string is known as the Lindenmayer System (L-System) (Lindenmayer, 1990).

Parallel rewrite systems or L-systems provide a useful formal model for the description of developmental processes in organisms. We will give some rudimentary definitions for context-free L-systems with stacking capability. As it is in general very difficult to create an L-system simulating some special growth process we will introduce supporting L-system (Jacob, 1995) inference.

The development of L-System to explain the plant growth based on the environment condition can be seen on (Mech, 1996). It is then improved by (Prusinkiewicz, 2003). L-System methods have been improved for modeling tools at many kinds of plant. The basic concepts of axiom and rules are the basis of how the growth of L-System that works (Viruchpintu, 2005).

At the beginning of plant growth, the sprout of plant initially denoted as axiom. Let axiom be the sprout of plant at the beginning of plant growth. This characteristic rules are illustrated in the further improvement of L-System that follows the plant growth naturally. It is the followed by the growth of stalk, branch, leaf and bloom. The rule of plant growth is represented by the axiom. On the other hand, the value of axiom is generally generated using the probability (Atris, 2010). To make the axiom value closer to the real environment condition, we propose the use of Artificial Neural Networks.

This research is aimed to generate plant growth with L-System method using the growth parameters is given based on the artificial neural networks. Design of plant growth used *Mathematica* programming (Heikki, 2009) on Windows Operating System. The output graphic on 3D reflected on plant growth as a virtual plant growth system.

METHODS

Research is carried out at Multimedia and Networking Laboratory, Institute Technology Sepuluh November. Some references on plant growth are also used. Some data is collected on the plant growth based on the extra inorganic fertilizer (Nitrogen(N), Phosphor(P) and Kalium(K)) and also organic fertilizer.

This research used Zinnia Elegane Jacq, organic fertilizer and inorganic fertilizer (Urea (N), Super Posfat (P), KCl (K), POC NASA and Hormonik with concentration 2 cc/l). Some tools are used, such as electric scale, *leaf area meter*, oven, termohigrometer and luxmeter. Sample data are taken from the plant after 26 days from transplating (HST). In this research, the amount of fertilizer given varies. The field is in Karangploso, Malang, East Java. The data collection is taken between Januarys until May 2011.

The hardware requirements used to model and visualize the plant growth is a personal computer with the specifications Intel Pentium Dual CPU 2.8 GHZ Processor, RAM 1 Giga Byte, 100GB Hard drive, Graphic Card NVIDIA GeForce 8400GS.

The *Mathematica* software under windows operating system is used in this research. Figure 1 shows the research methodology for Integration of Neural Networks into an L-System.

In this research, the observations are carried out in 2 things, the plant growth itself and the environment. The plant growth observations are carried out by collecting data of the plant based on the height, the stalk diameter, leaf (which includes height, width and area), the bloom diameter. On the other hand, the environment observations include putting different combination of fertilizer, temperature, weather, humidity and light intensity.

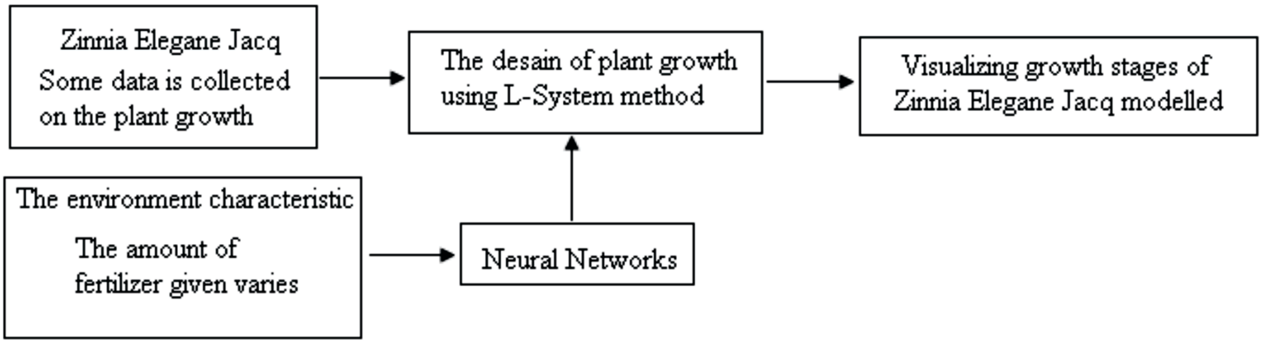

Fig. 1 The research methodology for Integration of artificial neural networks into an L-System

RESULTS AND DISCUSSION

The design of plant growth using L-System method

D0L-systems (D0 meaning: deterministic with no context) are the simplest type of L-systems. Formally a D0L-system L=(∑, α, P, T) , capable of encoding geometrical structures, consists of the following ingredients: a) an alphabet ∑=(δ_1,…, δ_n), each symbol of which stands for a morphological unit, like a sprout, a stalk, a leaf and a bloom b) a start string α, referred to as the axiom, which is an element of ∑, the set of all finite words over the alphabet ∑, c) P = (p_1,…, p_n), a set of productions or rewrite rules, d) a geometrical interpretation T , a 3D semantics, for some of the symbols from ∑, translating a string into a spatial structure, i.e. special symbols represent commands to draw graphic objects like points, lines, polygons etc; this translation is commonly known as turtle geometry interpretation.

Let us consider the following L-system: G
Σ = {a,b,c,d,e}
ω = abc
P_1 a→bc
P_2 c→ae
which generates the following sequence of strings:
Axiom : abc
Iteration 1 : bcbae
Iteration 2 : baebbce
Iteration 3 : bbccebbaee

In this case, some samples of Zinnia Elegane Jacq plant, Figure 7 shows an example L-system describing growth sequences of sprouts, leaves and blooms of virtual plant, together with its graphical interpretation. The D0L-system encodes turtle geometry macros for generating graphical representations of the leaves, blooms and stalks in Figure 2.

$G_{ZinniaEleganeJacq}$ = (Σ, P= {p_1… p_5}, α)
Σ= {FO, PD, PU, RR, *sprout*, *stalk*, *leaf*, *bloom*}
α = *sprout* (4)

P = sprout developing leaves and flower
p_1 = *sprout*(4) → FO *stalk*(3) [PU(60) *leaf*(1)] FO *stalk*(2) RR(90) [PU(60), *leaf*(1)] FO *stalk*(2) RR[90] [PU(60) *leaf*(0)] [PU(30) *sprout*(2)] RR(180) [PU(30) *sprout*(2)] FO *stalk*(2) *bloom*(0)

Ripping sprout:
p_2= *sprout* (t < 4) → *sprout* (t+1)

Stalk elongation:
p_3= *stalk* (t > 0) → FO *stalk* (t-1)

Changing leaf sizes:
p_4= *leaf* (t<4) → *leaf* (t + 1.5)

Growing bloom:
p_5= *bloom* (t<4) → *bloom* (t + 1)

Fig. 2 Example of parameterized D0L-System modeling Zinnia Elegane Jacq stages of plantlike structure

The D0L-System encodes turtle geometry macros for generating graphical representations of the leaves, blooms and stalks. An each symbol, f represent commands to move forward the turtle, b represent commands to move backward the turtle and change the drawing tools orientation by rotation around its longitudinal, lateral, and vertical axes is rl (roll left), rr (roll right), pu (pitch up), pd (pitch down), yl (yaw left) and yr (yaw right) in Table 1, thus translating a one-dimensional string into a 3D geometrical object resembling a plant (some of these commands do not occur in the example L-system).

Table 1

Turtle Rules for Interpreting 3D

Symbol of string	Description
FO[s]	Advances the turtle by a step size of s in the direction and draws a line depending on the current color and line thickness settings.
PU(α)	The turtle is tilted up around its transverse axis y by an angle of α degree.
PD(α)	The turtle is tilted down around its transverse axis y by an angle of α degree.
RR(α)	The turtle is rotated right (clockwise) around its longitudinal axis x by an angle of α degrees.
RL(α)	The turtle is rotated left (clockwise) around its longitudinal axis x by an angle of α degrees.
YR(α)	The turtle is rotated right (clockwise) around its vertical axis Z by an angle of α degree.
YL(α)	The turtle is rotated left (counterclockwise) around its vertical axis Z by an angle of α degree.

The kLSystems package in MathEvolvica (Jacob, 1995) contains definitions for the application of parallel rewrite rules of L-Systems with left and right contexts with arbitrary length. Each rule of the form $l < p > r \rightarrow s$ as described above is represented by a *Mathematica* expression of the form

LRule[LEFT[l], PRED[p], RIGHT[r], SUCC[s]].

Accordingly, we define the production set as an LRULES expression

LRULES [LRule[...], LRule[...], ...]

and an L-system is described as follows:

LSystem[_AXIOM, LRULES[__LRule]]

With this representation we can easily derive the type of the expressions and sub expressions by only looking at sprout symbols. Application kLSystems package in MathEvolvica at plant growth of Zinnia Elegane Jacq (Lydia, 1998).

The Integration Of L-System and Artificial Neural Networks

Here, the artificial neural networks is used to explain input/output relationship for a non linear model. This input/output relationship is generally the interconnections among various elements of a growing structure and the environment. The Neural Networks output is used as values for the growth parameter, such as various elements of a growing structure which are *sprout, stalk, leaf and bloom.*

The experiment is conducted to the plant growth of Zinnia Elegane Jacq with various kinds of treatments. As an example, treatment 1 is without any fertilizers. Treatment 2 is with no organic and 50% amount of inorganic, and so on. The amount of fertilizer that suits the plant growth will generate the growth parameter, as shown in Table 1.

Table 2

Growth Parameter from the treatment of various amount of fertilizer given to Zinnia Elegane Jacq

Code	Growth Parameter				
	Stalk High (cm)	Leaves High (cm)	Leaves Width (cm)	Bloom Diameter (cm)	Plant High (cm)
1	8,3	9,9	5,7	18,7	51.2
2	9,4	10	5,7	19	55
3	10	10,4	5,8	21,1	55
4	9,3	10,3	5,8	23,3	51.8
5	9,2	10,1	5,8	20,7	55
6	10,9	10,4	6	19,2	58.2
7	11	10,4	6	23,2	55
8	11	10,5	6,1	25	55
9	11,4	10,7	6,2	26,7	61.5
10	11,6	11	6,2	27,8	58.2
11	11,8	11,1	6,2	27	58.2
12	12	11,2	6,2	28,9	60.9
13	12,4	11,3	6,3	29,2	61.5
14	12,5	11,4	6,6	31	61.5
15	12,7	11,4	6,6	31	61.5

In this observation, the measurement of plant growth is divided into two parts: a). 15 data is used as the parameters for the learning in the artificial neural networks, as in table 1, b). 5 data is used as the comparator to the output of artificial neural networks (test data).

The modeling of artificial neural networks with *Mathematica* software is using 2 variable inputs which are combination of inorganic fertilizer and organic fertilizer, and one output which is the growth parameter (Stalk Height, Leaves Height, Leaves Width, Bloom Diameter, Length Branch, Number of Branch, Number of Bloom, and Plant Height). The input and output vectors of the architecture artificial neural networks are shown in figure 3.

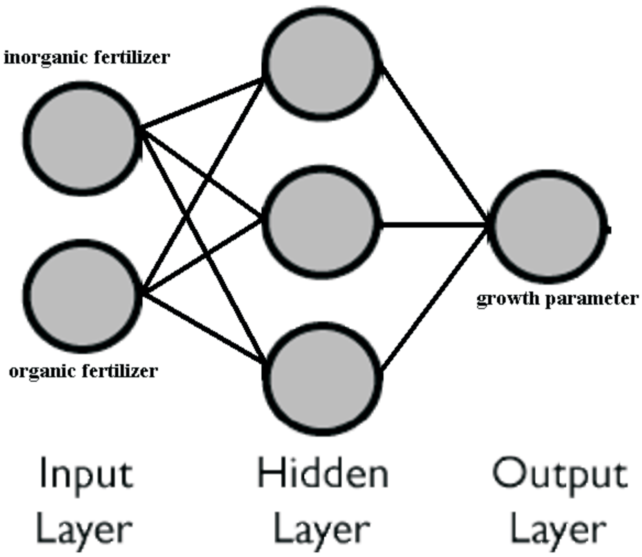

Fig. 3 A two-input and one-output network with three hidden layer.
This network architecture takes a pattern of inputs and creates a binary output.

Individual processing elements receive inputs from many other processing elements (x_i) via weighted connections. We can get the dimensions of the input and output vectors from the architecture artificial neural networks , the input is the first element, the desired output is the second element. We can set the size of the artificial neural networks.

In this figure 5, every processing element receives one or more inputs (x_i) via weighted connections (α_{hidden}). The initial set up in this research, the artificial neural network requires 2-element input (organic fertilizer and inorganic fertilizers), 3 hidden elements and a single output (that's called stalk). From the 3 elements hidden layer, it will be obtained the weights of the inputs to the hidden layer and the weights of the hidden layer to the output layer.

These weighted inputs ($\alpha_{hidden.}x_i$) are summed and the threshold value (θ_j) is then added, generating a single activation level for the processing element (h_{hidden}).

$$h_{hidden} = \sum \alpha_{hiddeni} x_i + \theta_j$$

This activation level constitutes the argument of a transfer function , sigmoid function as in Listing 7. This function then generates the node output . This output is passed to the weighted input connections of many other processing elements. This process is shown for node j in Figure 4.

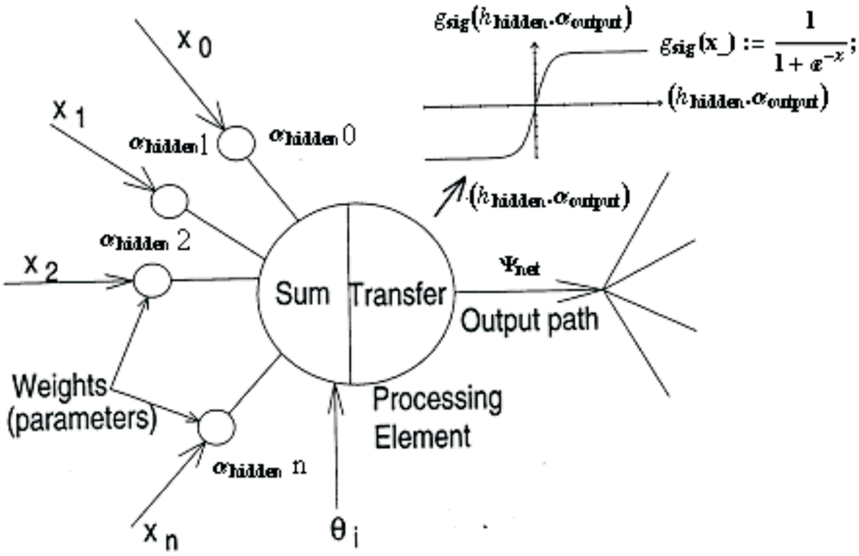

Fig. 4 Operation of a typical processing element.

The Back-propagation networks is the network type most suited to forecasting (Niklas, 1986). The expected relation is "learned" by repeatedly presenting samples of the expected input/output relationship to the network and adjusting the model parameters (i.e. the connection weights) in order to get the best possible match between the historical values and those generated by the model.

To present the training set of input (organic fertilizer and inorganic fertilizers) and stalk output using the *back propagation* method, and Training Set function within *Mathematica* is used (Freeman, 1994). The high level code to run the overall 'artificial neural network' system that is already developed, is shown in Listing 1. In this code, it runs with 12500 at organic fertilizer 0 and inorganic fertilizer 50. For the first 12500 of running this code, the epoch size is plotted as seen in Figure 5. The test result of the program is shown in Listing 2.

```
ws = setupNN[2, 9, 1];
{εTable, ws} = trainNN[ts, ws, 12500, 0.99, 0.00001];
ListPlot[εTable, PlotRange → All]
feedForward[#1, ws] & /@ ts[[All, 1]]
```

List. 1 The code of running it through our network with 12500 at organic fertilizer 0 and inorganic fertilizer 50

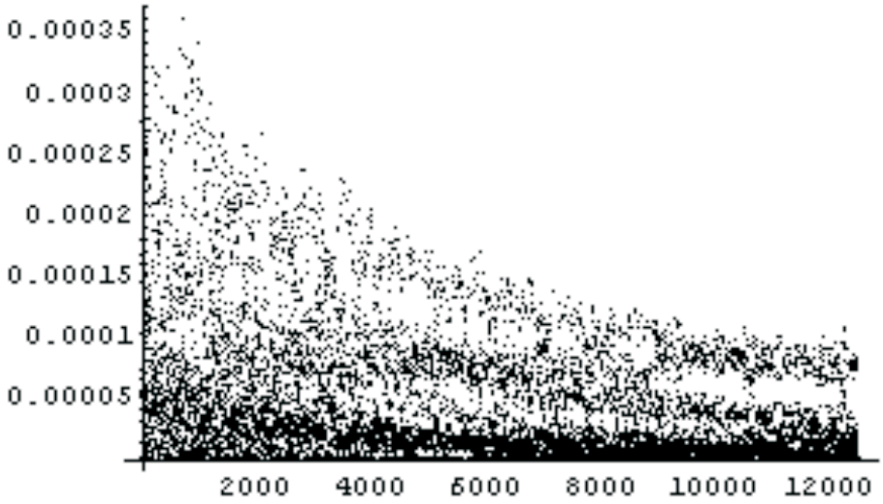

Fig. 5 First 12500 iterations we get the following error

```
In[111]:= new = ( {0.1, 0.75}
                  {0.25, 0.75}
                  {0.50, 0.75}
                  {0.75, 0.75}
                  {0.90, 0.75} );

In[110]:= ({#1, feedForward[#1, ws]}|&) /@ new[[All, 1]]

Out[110]= {{{0.1, 0.75}, {0.0941964}}, {{0.25, 0.75}, {0.0988415}},
  {{0.5, 0.75}, {0.107076}}, {{0.75, 0.75}, {0.115884}}, {{0.9, 0.75}, {0.121417}}}
```

List. 2 The code of the results for the testing at stalk

The output of the artificial neural networks is then compared to the actual data of plant growth in the field. From this figure 6, it can be seen that the difference between artificial neural networks output and the actual plant growth is less than 6% on average.

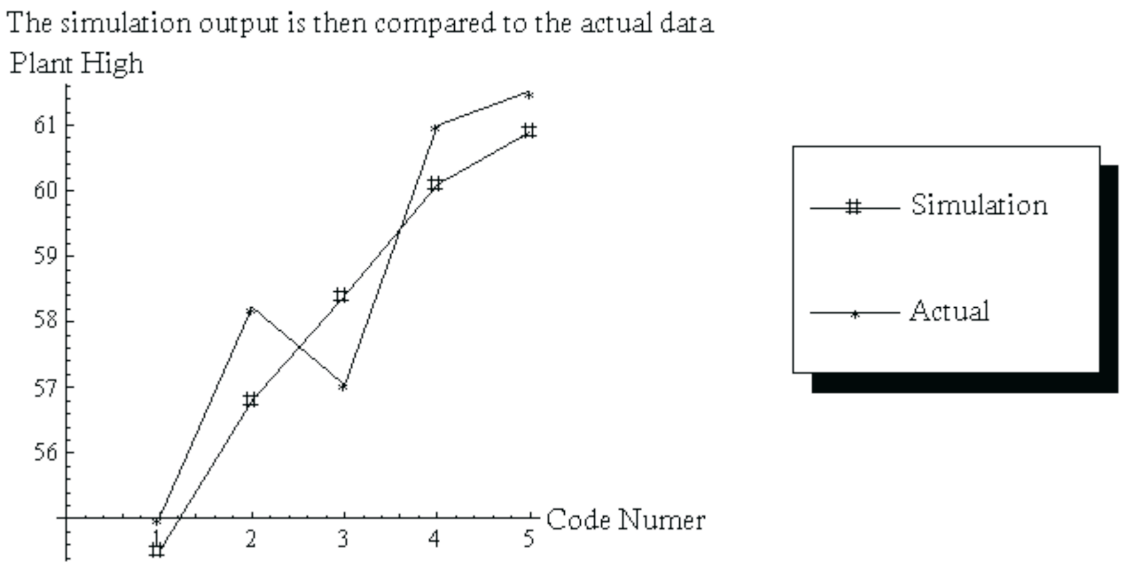

Fig. 6 The comparison of simulation and actual output.

Studia Universitatis "Vasile Goldiş", Seria Ştiinţele Vieţii
Vol. 22, issue 3, 2012, pp. 411-418

Based on the output of the artificial neural networks, it can be derived the growth parameter. It is then applied to the plant, in such a way that it is possible to get the crop as expected. Once the stalk growth is obtained, then the leaf and bloom growth need to be found using the similar method of the artificial neural networks. In order to visualize the plant, the virtual growth of Zinnia Elegane Jacq with the 5th treatment (75% inorganic and organic fertilizer is 100%) is shown in figure 7.

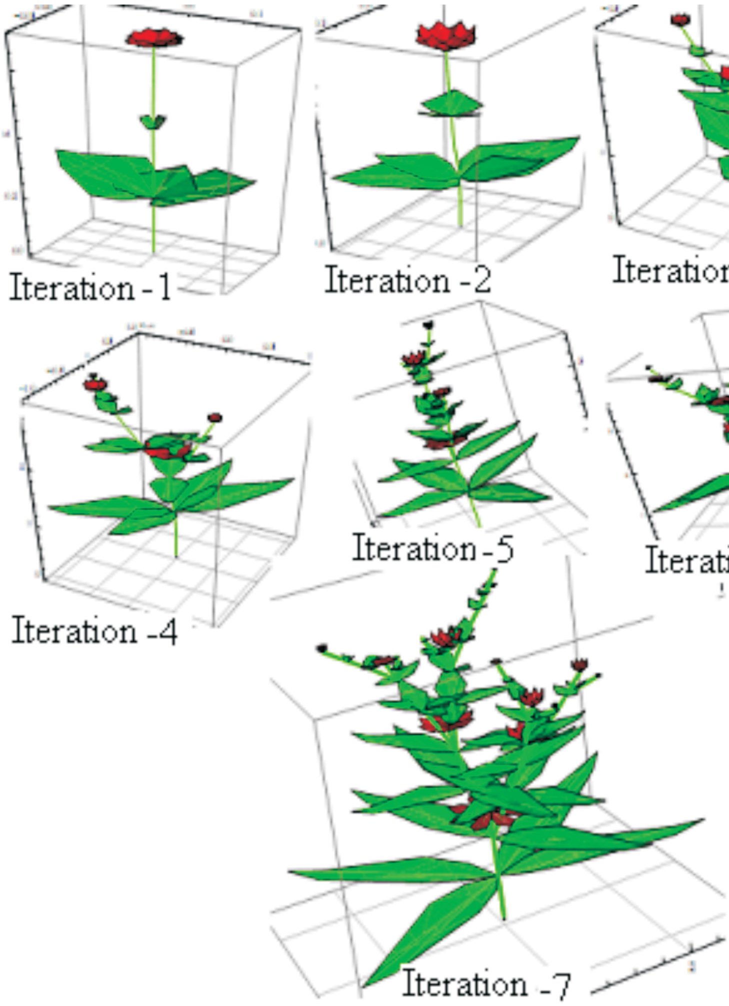

Fig. 8 Growth processes of Zinnia Elegane Jacq plant of Iteration 7

CONCLUSION

This research shows that the Integration of the artificial neural networks into L-system based Plant modeling Environment has been implemented. In this research, it can be demonstrated that the difference between the artificial neural networks output and the actual plant growth is less than 6% on average. It is concluded from this research that the agriculture of Zinnia Elegane Jacq plant could be develop further with the artificial neural networks, which both controlled environment and controlled mixture of fertilizer are used as decision support system.

REFERENCES

Lindenmayer, P. Prusinkiewicz. The Algorithmic Beauty of Plants, Springer Verlag, New York, pp. 227, 1990.

Jacob, C., Genetic L-System Programming: Breeding and Evolving Artificial Flowers with Mathematica, First Int. Mathematica Symposium, Southampton, UK, Computational Mechanics Pub, 1995

R. Mech, P. Prusinkiewicz. Visual Models of Plants Interacting with Their Environment, Proceedings of SIGGRAPH 96, ACM SIGGRAPH, p. 397, 1996.

R. Viruchpintu, N. Khiripet. Real Time 3D Plant Structure Modeling by L-System with Actual Measurement Parameters, National Electronics and Computer Technology Center, Bangkok, 2005

Atris Suyantohadi, Alfiyan, Mochamad Hariadi, Mauridhi Hery Purnomo. Plant Growth Modelling Using L-System Approach and Its Visualization, Jurnal *Makara , Teknologi, Vol. 14, NO. 2, pp. 92-96*, 2010.(In Indonesian).

Heikki Ruskeepaa. Mathematica Navigator, Elsevier Inc, 2009.

Jacob, C. MathEvolvica - Simulierte Evolution von Entwicklungs programmen der Natur, PhD dissertation , Arbeitsberichte des Instituts für mathematische Maschinen und Datenverarbeitung, Band 28, Nummer 10, Erlangen, 1995. .(In Deutch Language).

Lydia Kristi, Nurhajati Ansori. Pengaruh Dosis Dan Waktu Pemberian Uniconazole Terhadap Pertumbuhan Dan Pembungaan Kembang Kertas (Zinnia Elegane Jacq.), Buletin. Agr. Vol. XX. No.1 , 1998.(In Indonesian).

Niklas, K.J. Computer-simulated Plant Evolution, in: Scientific American, 254 (March), pp. 68-75, 1986.

Freeman, JA. *Simulating Neural Networks with Mathematica,* Journal Mathematica Addison-Wesley, Reading MA, 1994.