

# Implementasi *Extreme Learning Machine* untuk Pengenalan Objek Citra Digital

Zulfa Afiq Fikriya, Mohammad Isa Irawan, dan Soetrisno  
Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Institut Teknologi Sepuluh Nopember (ITS)  
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia  
*e-mail*: mii@its.ac.id

**Abstrak**— Pengenalan citra digital merupakan bagian yang sangat penting dalam *computer vision* yang menerapkan *pattern recognition*. Pengenalan citra digital bertujuan untuk menduplikasi kemampuan manusia dalam memahami informasi citra sehingga komputer dapat mengenali objek pada citra selayaknya manusia. Salah satu metode *pattern recognition* adalah *Extreme Learning Machine* (ELM). *Extreme Learning Machine* merupakan jaringan syaraf tiruan *feedforward* dengan satu *hidden layer* atau lebih dikenal dengan istilah *single hidden layer feedforward neural networks* (SLFNs). *Extreme Learning Machine* untuk pengenalan objek citra digital pada Tugas Akhir ini terdiri dari 2500 node pada *input layer*, 1250 node pada *hidden layer*, dan 3 node pada *output layer*. Dataset dikelompokkan berdasarkan ukuran objek dalam citra. Hasil uji coba dan evaluasi model dengan data *testing* menghasilkan tingkat akurasi sebesar 57,33% pada citra dengan objek berukuran kecil, 81,33% pada citra dengan objek berukuran sedang, dan 74,67% pada citra dengan objek berukuran besar.

**Kata Kunci**— Pengenalan Objek, *Machine Learning*, *Extreme Learning Machine*

## I. PENDAHULUAN

PERKEMBANGAN teknologi informasi yang sangat cepat di tengah masyarakat membuat data atau informasi tidak hanya disajikan dalam bentuk teks, tetapi juga dapat berupa gambar, audio, dan video. Keempat macam data atau informasi ini sering disebut multimedia. Era teknologi informasi saat ini tidak dapat dipisahkan dari multimedia. Saat ini orang tidak hanya dapat mengirim pesan dalam bentuk teks, tetapi juga dapat mengirim pesan berupa gambar maupun video. Citra (*image*), istilah lain untuk gambar, merupakan salah satu komponen multimedia yang berperan sangat penting sebagai bentuk informasi visual. Sebuah gambar dapat memberikan informasi yang lebih banyak dari pada informasi tersebut disajikan dalam bentuk kata-kata.

Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra sehingga dapat lebih mudah diinterpretasikan oleh mata manusia, namun dengan berkembangnya dunia komputasi dan dengan semakin meningkatnya kapasitas dan kecepatan proses komputer maka muncullah ilmu-ilmu komputasi yang memungkinkan komputer dapat mengambil informasi dari suatu citra untuk keperluan pengenalan objek secara otomatis. Pengolahan citra digital ini sangat erat kaitannya dengan ilmu pengenalan pola (*pattern recognition*) yang umumnya bertujuan untuk mengenali suatu objek dengan cara mengekstrak informasi penting yang terdapat pada suatu citra. Pengenalan pola bisa didefinisikan sebagai cabang

kecerdasan yang menitikberatkan pada metode pengklasifikasian objek ke dalam kelas-kelas untuk menyelesaikan masalah tertentu.

Pengenalan citra digital merupakan salah satu masalah dalam *computer vision* khususnya *pattern recognition* dan *machine learning*. *Machine learning* adalah salah satu disiplin ilmu dari *computer science* yang mempelajari bagaimana membuat komputer atau mesin mempunyai suatu kecerdasan. Agar mempunyai suatu kecerdasan, komputer atau mesin harus dapat belajar. Dengan kata lain *machine learning* adalah suatu bidang keilmuan yang berisi tentang pembelajaran komputer atau mesin untuk menjadi cerdas.

Pada tahun 2016 telah dilakukan penelitian oleh I Wayan Suartika, dkk mengenai salah satu metode *machine learning* yaitu *Convolutional Neural Network* (CNN) yang digunakan untuk klasifikasi citra objek. CNN adalah pengembangan dari *Multilayer Perceptron* yang didesain untuk mengolah data dua dimensi. Hasil uji coba dari klasifikasi objek citra dengan CNN ini menghasilkan tingkat akurasi sebesar 20%-50% [1]. Metode lain dari *machine learning* yang dapat mengatasi permasalahan ini adalah *Extreme Learning Machine* (ELM) dengan tingkat akurasi yang lebih baik [2].

Berdasarkan kelebihan ELM tersebut, dapat diambil kesimpulan bahwa ELM memiliki kemampuan klasifikasi yang diperuntukkan untuk data gambar. Oleh karena itu, pada Tugas Akhir ini akan digunakan *Extreme Learning Machine* untuk pengenalan objek citra digital.

## II. DASAR TEORI

### A. Citra Digital

Citra digital merupakan suatu matriks di mana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar atau piksel) menyatakan tingkat keabuan pada titik tersebut. Untuk sebuah citra digital, setiap piksel memiliki nilai integer yakni *gray level* yang menunjukkan amplitudo atau intensitas dari piksel tersebut. Citra merupakan fungsi dua dimensi yang kedua variabelnya yaitu nilai amplitudo dan koordinatnya merupakan nilai integer [7].

### B. Deteksi Tepi Canny

Salah satu algoritma deteksi tepi modern adalah deteksi tepi dengan menggunakan metode Canny. Deteksi tepi Canny ditemukan oleh Marr dan Hildreth yang meneliti pemodelan persepsi visual manusia. Deteksi tepi Canny dapat mendeteksi tepian yang sebenarnya dengan tingkat error yang minimum [8].

Berikut adalah langkah-langkah dalam melakukan deteksi tepi Canny:

1. Menghilangkan noise yang ada pada citra dengan mengimplementasikan filter Gaussian. Hasilnya citra akan tampak sedikit buram. Hal ini dimaksudkan untuk mendapatkan tepian citra yang sebenarnya. Bila tidak dilakukan maka garis-garis halus juga akan dideteksi sebagai tepian.
2. Melakukan deteksi tepi dengan salah satu operator deteksi tepi seperti Roberts, Prewitt, atau Sobel dengan melakukan pencarian secara horizontal ( $G_x$ ) dan secara vertikal ( $G_y$ ). Hasil dari kedua operator digabungkan untuk mendapatkan hasil gabungan tepi horizontal dan vertikal dengan rumus :

$$|G| = |G_x| + |G_y|$$

3. Menentukan arah tepian yang ditemukan dengan menggunakan rumus :

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

Selanjutnya membagi ke dalam 4 warna sehingga garis dengan arah yang berbeda memiliki warna yang berbeda. Pembagiannya adalah:

- $0^\circ - 22,5^\circ$  dan  $157,7^\circ - 180^\circ$  berwarna kuning
  - $22,5^\circ - 67,5^\circ$  berwarna hijau
  - $67,5^\circ - 157,5^\circ$  berwarna merah
4. Memperkecil garis tepi yang muncul dengan menerapkan non maximum suppression sehingga menghasilkan garis tepian yang lebih ramping.
  5. Binerisasi dengan menerapkan dua buah nilai ambang.

### C. Machine Learning

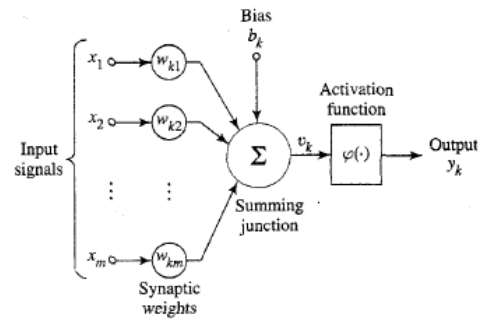
Secara definisi, *machine learning* adalah cabang dari ilmu kecerdasan buatan yang berfokus pada pembangunan dan studi sebuah sistem agar mampu belajar dari data-data yang diperolehnya. Menurut Arthur Samuel, *machine learning* adalah bidang studi yang memberikan kemampuan program komputer untuk belajar tanpa secara eksplisit diprogram.

Untuk bisa mengaplikasikan teknik-teknik *machine learning* maka harus ada data. Tanpa data maka algoritma *machine learning* tidak dapat bekerja. Data yang ada biasanya dibagi menjadi dua, yaitu data *training* dan data *testing*. Data *training* digunakan untuk melatih algoritma, sedangkan data *testing* digunakan untuk mengetahui performa algoritma yang telah dilatih sebelumnya ketika menemukan data baru yang belum pernah dilihat.

### D. Jaringan Syaraf Tiruan

Jaringan syaraf tiruan adalah salah satu cabang dari *machine learning*. JST merupakan teknik yang digunakan untuk membangun program yang cerdas dengan pemodelan yang mensimulasikan cara kerja jaringan syaraf pada otak manusia. Jadi, JST menggunakan konsep kerja dari syaraf otak manusia untuk menyelesaikan perhitungan pada komputer.

Seperti otak manusia, fungsi dari jaringan ditentukan oleh hubungan antar neuron. Hubungan antar neuron ini disebut bobot (*weight*). Untuk mendapatkan fungsi tertentu dapat dilakukan dengan pelatihan (*training*) dengan menyesuaikan nilai bobot dari masing-masing neuron. Satu sel syaraf terdiri dari tiga bagian, yaitu fungsi penjumlahan (*summing function*), fungsi aktivasi (*activation function*), dan keluaran (*output*). Pada **Gambar.1.** dapat dilihat model dari sel syaraf (neuron) [10].



Gambar.1. Sel Syaraf (Neuron)

Pada umumnya, JST dilatih (*trained*) agar *input* mengarah ke target *output* yang spesifik. Jadi jaringan dilatih terus menerus hingga mencapai kondisi di mana *input* sesuai dengan target yang telah ditentukan. Pelatihan di mana setiap *input* diasosiasikan dengan target yang telah ditentukan disebut pelatihan terarah (*supervised learning*).

### E. Extreme Learning Machine

*Extreme Learning Machine* merupakan metode pembelajaran baru dari jaringan syaraf tiruan. ELM merupakan jaringan syaraf tiruan *feedforward* dengan *single hidden layer* atau biasa disebut *single hidden layer feedforward neural networks* (SLFNs). Metode pembelajaran ELM dibuat untuk mengatasi kelemahan-kelemahan dari jaringan syaraf tiruan *feedforward* terutama dalam hal *learning speed* [11]. Huang et al mengemukakan dua alasan mengapa JST *feedforward* mempunyai *learning speed* rendah, yaitu:

1. Menggunakan *slow gradient based learning algorithm* untuk melakukan *training*.
2. Semua parameter pada jaringan ditentukan secara *iterative* dengan menggunakan metode pembelajaran tersebut.

Pada pembelajaran dengan menggunakan *conventional gradient based learning algorithm* seperti *backpropagation* (BP), semua parameter pada JST *feedforward* harus ditentukan secara manual. Parameter yang dimaksud adalah *input weight* dan *hidden bias*. Parameter-parameter tersebut juga saling berhubungan antara layer yang satu dengan yang lain, sehingga membutuhkan *learning speed* yang lama dan sering terjebak pada *local minima*. Sedangkan pada ELM parameter-parameter seperti *input weight* dan *hidden bias* dipilih secara random, sehingga ELM memiliki *learning speed* yang cepat dan mampu menghasilkan *good generalization performance*.

Metode ELM mempunyai model matematis yang berbeda dari jaringan syaraf tiruan *feedforward*. Model matematis dari ELM lebih sederhana dan efektif. Untuk  $N$  jumlah pasangan input dan target output yang berbeda  $(x_i, t_i)$ , dengan  $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in \mathbf{R}^n$  dan  $t_i = [t_{i1}, t_{i2}, \dots, t_{in}]^T \in \mathbf{R}^m$ , standar SLFNs dengan jumlah *hidden nodes* sebanyak  $\tilde{N}$  dan fungsi aktivasi  $g(x)$  dapat dimodelkan secara matematis sebagai berikut [12]:

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(x_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = o_j, \quad (1)$$

$j = 1, 2, \dots, N$

di mana

- a.  $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$  merupakan vektor bobot yang menghubungkan *hidden node* ke- $i$  dan *input nodes*.
- b.  $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$  merupakan vektor bobot yang menghubungkan *hidden node* ke- $i$  dan *output nodes*.
- c.  $b_i$  merupakan *threshold* dari *hidden node* ke- $i$
- d.  $w_i \cdot x_j$  merupakan *inner product* dari  $w_i$  dan  $x_j$

SLFNs standar dengan  $\tilde{N}$  *hidden nodes* dan fungsi aktivasi  $g(x)$  diasumsikan dapat memperkirakan  $N$  sampel ini dengan tingkat *error* 0 yang artinya  $\sum_{j=1}^N \|o_j - t_j\| = 0$ , sehingga terdapat  $\beta_i, w_i$ , dan  $b_i$  sedemikian hingga [9]

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = t_j, \quad j = 1, 2, \dots, N \quad (2)$$

Persamaan di atas dapat dituliskan secara sederhana sebagai [12]:

$$H\beta = T, \quad (3)$$

di mana

$$H = \begin{bmatrix} g(w_{11} \cdot x_1 + b_1) & \dots & g(w_{\tilde{N}1} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(w_{1N} \cdot x_N + b_1) & \dots & g(w_{\tilde{N}N} \cdot x_N + b_{\tilde{N}}) \end{bmatrix},$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix} \text{ dan } T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}$$

$H$  pada persamaan di atas adalah matriks *output hidden layer* dari jaringan syaraf.  $g(w_i \cdot x_j + b_i)$  menunjukkan *output* dari *hidden neuron* yang berhubungan dengan *input*  $x_j$ .  $\beta$  merupakan matriks dari bobot *output* dan  $T$  matriks dari target. Pada ELM, *input weight* dan *hidden bias* ditentukan secara acak, sehingga bobot *output* yang berhubungan dengan *hidden layer* dapat ditentukan dari persamaan (2.5) [13]:

$$\beta = H^\dagger T \quad (4)$$

Algoritma *Extreme Learning Machine* (ELM) ditunjukkan pada **Gambar.2.** berikut [13].

<b>Algoritma Extreme Learning Machine</b>	
<i>Input</i>	: pola <i>input</i> $x_j$ dan pola target <i>output</i> $t_j, j = 1, 2, \dots, N$
<i>Output</i>	: bobot <i>input</i> $w_i$ , bobot <i>output</i> $\beta_i$ dan bias $b_i, i = 1, 2, \dots, \tilde{N}$
Algoritma	:
Langkah 1	: Tentukan fungsi aktivasi ( $g(x)$ ) dan jumlah <i>hidden nodes</i> ( $\tilde{N}$ )
Langkah 2	: Tentukan secara acak nilai dari bobot <i>input</i> $w_i$ dan bias $b_i, i = 1, 2, \dots, \tilde{N}$
Langkah 3	: Hitung nilai matriks <i>output</i> $H$ pada <i>hidden layer</i>
Langkah 4	: Hitung nilai bobot <i>output</i> $\beta$ dengan menggunakan $\beta = H^\dagger T$

Gambar.2. Algoritma *Extreme Learning Machine*

#### F. Moore-Penrose Generalized Inverse

Konsep invers matriks pada umumnya merupakan konsep invers matriks yang terbatas pada matriks persegi berordo  $n \times n$  dan non singular. Matriks yang berordo  $m \times n$

yang singular tidak mempunyai invers. Akan tetapi, terdapat matriks yang seolah-olah menjadi invers untuk matriks yang berordo  $m \times n$  yang singular. Matriks tersebut dinamakan *Moore-Penrose generalized inverse* atau *pseudoinverse* yang ditemukan oleh E.H. Moore pada tahun 1920 dan Roger Penrose pada tahun 1955.

Definisi dari *Moore-Penrose generalized inverse* adalah sebagai berikut [14]:

Jika  $A \in M_{n,m}$  maka terdapat  $A^\dagger \in M_{m,n}$  yang unik dan memenuhi empat kondisi Penrose yaitu:

1.  $AA^\dagger A = A$
2.  $A^\dagger AA^\dagger = A^\dagger$
3.  $A^\dagger A = (A^\dagger A)^*$
4.  $AA^\dagger = (AA^\dagger)^*$

Di mana  $A^*$  menyatakan konjugat transpose dari matriks  $A$ .

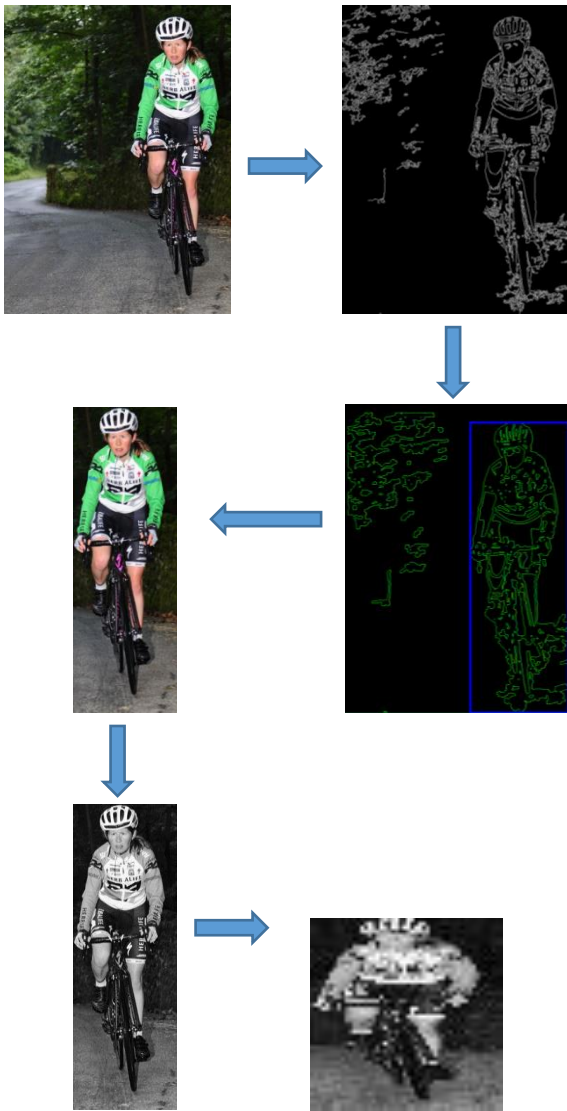
### III. PERANCANGAN SISTEM

#### A. Pengumpulan Dataset

Data yang dikumpulkan adalah data gambar yang di dalamnya terdapat satu objek dari kategori guci, kelinci, mobil, sepatu, dan sepeda. Data gambar tersebut adalah gambar yang berwarna dengan ukuran yang bervariasi. Data gambar ini nantinya akan dibagi ke dalam dua kelompok yaitu data *training* dan data *testing*.

#### B. Praproses Data

Agar dapat diproses oleh *Extreme Learning Machine*, maka data citra yang telah dikumpulkan harus melalui praproses terlebih dahulu. Praproses ini terdiri dari proses deteksi tepi, *cropping*, konversi citra warna ke *grayscale*, dan penyeragaman ukuran citra. Algoritma deteksi tepi yang digunakan adalah metode Canny. Setelah dilakukan deteksi tepi kemudian masing-masing tepi yang terdeteksi akan diberikan *bounding box*. Luas masing-masing *bounding box* tersebut kemudian dihitung dan dicari *bounding box* dengan luas terbesar untuk dilakukan proses *cropping*. Selanjutnya citra berwarna yang telah melalui proses *cropping* akan dikonversi menjadi citra *grayscale* dan dilakukan penyeragaman ukuran citra yaitu menjadi 50x50 piksel. Praproses citra input dapat dilihat pada Gambar.3. berikut.



Gambar.3. Tahap Praproses Citra Input

C. Pemodelan Extreme Learning Machine

ELM mempunyai model matematis yang berbeda dari jaringan syaraf tiruan *feedforward*. Model matematis dari ELM lebih sederhana dan efektif. Dengan 75 jumlah sampel yang berbeda  $(x_i, t_i)$ , jumlah *hidden nodes* sebanyak 1250 dan fungsi aktivasi  $g(x)$  maka berdasarkan Persamaan (1) didapatkan model matematis ELM pada Tugas Akhir ini adalah sebagai berikut

$$\sum_{i=1}^{1250} \beta_i g_i(x_j) = \sum_{i=1}^{1250} \beta_i g(w_i \cdot x_j + b_i) = o_j, \quad j = 1, 2, \dots, 75$$

di mana

- $w_i = [w_{i1}, w_{i2}, \dots, w_{i2500}]^T$  merupakan vektor bobot yang menghubungkan *hidden node* ke- $i$  dan *input nodes*.
- $\beta_i = [\beta_{i1}, \beta_{i2}, \beta_{i3}]^T$  merupakan vektor bobot yang menghubungkan *hidden node* ke- $i$  dan *output nodes*.
- $b_i$  merupakan bias dari *hidden node* ke- $i$
- $w_i \cdot x_j$  merupakan *inner product* dari  $w_i$  dan  $x_j$

Berikut penjabaran model pada setiap *layer* arsitektur jaringan.

1) Input Layer

*Layer* ini merupakan *layer* pertama pada jaringan. *Layer* ini terdiri dari 50x50 atau 2.500 node sesuai dengan ukuran citra input yang merupakan hasil dari praproses data. Satu piksel pada citra input mewakili satu node pada lapisan ini.

*Input layer* dengan *hidden layer* dihubungkan oleh vektor bobot  $w$  dan bias yang nilainya ditentukan secara acak. Fungsi yang dipilih sebagai fungsi aktivasi pada *layer* ini adalah fungsi softsign yaitu [15]:

$$f(x) = \frac{x}{1 + |x|}$$

Fungsi aktivasi ini memberikan batasan keluaran : (5) (-1,1). Berdasarkan Persamaan (5) dapat dirumuskan suatu fungsi umum sebagai berikut:

$$H_{i,j} = g(w_i \cdot x_j + b_i) = \frac{w_i \cdot x_j + b_i}{1 + |w_i \cdot x_j + b_i|} \quad (6)$$

Keterangan variabel dan indeks pada persamaan di atas adalah sebagai berikut:

- $H$  : matriks output pada *hidden layer*
- $w$  : vektor bobot yang menghubungkan *hidden node* dan *input node*.
- $x$  : vektor input
- $b$  : bias yang terhubung dengan *hidden node*
- $i$  : indeks jumlah node pada *hidden layer*
- $j$  : indeks jumlah citra input
- $w \cdot x$  : *inner product* dari  $w$  dan  $x$

2) Hidden Layer

*Layer* ini merupakan *layer* kedua dari jaringan dan terdiri dari 1250 buah node. *Hidden layer* dan *output layer* dihubungkan oleh vektor bobot  $\beta$ . Sehingga berdasarkan Persamaan (3) dengan  $N = 75$  dan  $\tilde{N} = 1250$  dapat dituliskan suatu persamaan sebagai berikut:

$$H\beta = T, \quad (7)$$

di mana

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{1250} \cdot x_1 + b_{1250}) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_{75} + b_1) & \dots & g(w_{1250} \cdot x_{75} + b_{1250}) \end{bmatrix},$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{1250}^T \end{bmatrix} \text{ dan } T = \begin{bmatrix} t_1^T \\ \vdots \\ t_{75}^T \end{bmatrix}$$

Keterangan variabel dan indeks pada persamaan di atas adalah sebagai berikut:

- $H$  : matriks output pada *hidden layer*
- $w$  : vektor bobot yang menghubungkan *hidden node* dan *input node*.
- $x$  : vektor input
- $b$  : bias yang terhubung dengan *hidden node*
- $w \cdot x$  : *inner product* dari  $w$  dan  $x$
- $\beta$  : vektor bobot yang menghubungkan *hidden node* dan *output node*.
- $T$  : matriks target

Sehingga bobot *output* yang berhubungan dengan *hidden layer* dapat ditentukan dari persamaan berikut:

$$\beta = H^\dagger T \quad (8)$$

dengan  $H^\dagger$  merupakan *Moore-Penrose generalized inverse* dari  $H$ .

3) Output Layer

Setiap node yang ada di *hidden layer* dihubungkan dengan *output layer* melalui vektor bobot  $\beta$ . Jumlah node pada *output layer* disesuaikan dengan jumlah kelas dari

objek yang akan dikenali. *Output layer* pada Tugas Akhir ini terdiri dari 3 buah node dan hasil keluaran dari *output layer* ini mewakili kelas dari citra input. Nilai output dari *layer* ini dapat dihitung dengan Persamaan (7).

IV. UJI COBA DAN EVALUASI

Uji coba dalam penelitian ini dilakukan sebanyak 3 kali dengan bobot dan bias yang diperoleh dari 3 kali proses *training*. Citra dibagi ke dalam 3 kategori berdasarkan ukuran objek di dalamnya, yaitu citra dengan objek berukuran kecil, sedang, dan besar. Objek dikategorikan kecil jika luasnya kurang dari 40% dari luas citra, sedang jika luasnya di antara 40% sampai dengan 70% dari luas citra, dan besar jika luasnya lebih dari 70% dari luas citra.

A. Uji Coba 1

1) Uji Coba pada Citra dengan Objek Berukuran Kecil

Rincian perhitungan akurasi model pada citra dengan objek berukuran kecil disajikan pada **Tabel.1.**

Tabel.1.

Perhitungan Tingkat Akurasi Model pada Citra dengan Objek Berukuran Kecil

No	Label	Benar	Salah	Total
1	Guci	1	4	5
2	Kelinci	5	0	5
3	Mobil	3	2	5
4	Sepatu	1	4	5
5	Sepeda	4	1	5
		14	11	25
Akurasi		56%	44%	100%

2) Uji Coba pada Citra dengan Objek Berukuran Sedang

Rincian perhitungan akurasi model pada citra dengan objek berukuran sedang disajikan pada **Tabel.2.**

Tabel.2.

Perhitungan Tingkat Akurasi Model pada Citra dengan Objek Berukuran Sedang

No	Label	Benar	Salah	Total
1	Guci	3	2	5
2	Kelinci	5	0	5
3	Mobil	5	0	5
4	Sepatu	2	3	5
5	Sepeda	5	0	5
		20	5	25
Akurasi		80%	20%	100%

3) Uji Coba pada Citra dengan Objek Berukuran Besar

Rincian perhitungan akurasi model pada citra dengan objek berukuran besar disajikan pada **Tabel.3.**

Tabel.3.

Perhitungan Tingkat Akurasi Model pada Citra dengan Objek Berukuran Besar

No	Label	Benar	Salah	Total
1	Guci	1	4	5
2	Kelinci	5	0	5
3	Mobil	4	1	5
4	Sepatu	2	3	5
5	Sepeda	5	0	5
		17	8	25
Akurasi		68%	32%	100%

B. Uji Coba 2

1) Uji Coba pada Citra dengan Objek Berukuran Kecil

Rincian perhitungan akurasi model pada citra dengan objek berukuran kecil disajikan pada **Tabel.4.**

Tabel.4.

Perhitungan Tingkat Akurasi Model pada Citra dengan Objek Berukuran Kecil

No	Label	Benar	Salah	Total
1	Guci	0	5	5
2	Kelinci	5	0	5
3	Mobil	4	1	5
4	Sepatu	3	2	5
5	Sepeda	4	1	5
		16	9	25
Akurasi		64%	36%	100%

2) Uji Coba pada Citra dengan Objek Berukuran Sedang

Rincian perhitungan akurasi model pada citra dengan objek berukuran sedang disajikan pada **Tabel.5.**

Tabel.5.

Perhitungan Tingkat Akurasi Model pada Citra dengan Objek Berukuran Sedang

No	Label	Benar	Salah	Total
1	Guci	3	2	5
2	Kelinci	5	0	5
3	Mobil	5	0	5
4	Sepatu	3	2	5
5	Sepeda	5	0	5
		21	4	25
Akurasi		84%	16%	100%

3) Uji Coba pada Citra dengan Objek Berukuran Besar

Rincian perhitungan akurasi model pada citra dengan objek berukuran besar disajikan pada **Tabel.6.**

Tabel.6.

Perhitungan Tingkat Akurasi Model pada Citra dengan Objek Berukuran Besar

No	Label	Benar	Salah	Total
1	Guci	3	2	5
2	Kelinci	4	1	5
3	Mobil	4	1	5
4	Sepatu	3	2	5
5	Sepeda	5	0	5
		19	8	25
Akurasi		76%	24%	100%

C. Uji Coba 3

1) Uji Coba pada Citra dengan Objek Berukuran Kecil

Rincian perhitungan akurasi model pada citra dengan objek berukuran kecil disajikan pada **Tabel.7.**

Tabel.7.

Perhitungan Tingkat Akurasi Model pada Citra dengan Objek Berukuran Kecil

No	Label	Benar	Salah	Total
1	Guci	1	4	5
2	Kelinci	3	2	5
3	Mobil	4	1	5
4	Sepatu	1	4	5
5	Sepeda	4	1	5
		13	12	25
Akurasi		52%	48%	100%

2) Uji Coba pada Citra dengan Objek Berukuran Sedang

Rincian perhitungan akurasi model pada citra dengan objek berukuran sedang disajikan pada **Tabel.8.**

Tabel.8.

Perhitungan Tingkat Akurasi Model pada Citra dengan Objek Berukuran Sedang

No	Label	Benar	Salah	Total
1	Guci	3	2	5
2	Kelinci	4	1	5
3	Mobil	5	0	5
4	Sepatu	3	2	5

5	Sepeda	5	0	5
		20	5	25
	Akurasi	80%	20%	100%

[3] Deepa, S. N., Arunadevi, B. (2013). "Extreme Learning Machine for Classification of Brain Tumor in 3D MR Images". *Informatol Vol. 46*, Hal 111-121.

3) Uji Coba pada Citra dengan Objek Berukuran Besar

Rincian perhitungan akurasi model pada citra dengan objek berukuran besar disajikan pada **Tabel.9**.

Tabel.9.

Perhitungan Tingkat Akurasi Model pada Citra dengan Objek Berukuran Besar

No	Label	Benar	Salah	Total
1	Guci	2	3	5
2	Kelinci	5	0	5
3	Mobil	5	0	5
4	Sepatu	3	2	5
5	Sepeda	5	0	5
		20	5	25
	Akurasi	80%	20%	100%

Perbandingan tingkat akurasi model secara keseluruhan disajikan pada **Tabel.10**.

Tabel.10.

Perbandingan Keseluruhan Akurasi

No	Uji Coba	Ukuran Objek		
		Kecil	Sedang	Besar
1	Uji Coba 1	56%	80%	68%
2	Uji Coba 2	64%	84%	76%
3	Uji Coba 3	52%	80%	80%
	Rata-Rata Akurasi	57,33%	81,33%	74,67%

V. KESIMPULAN

Dari uji coba dan evaluasi dapat disimpulkan beberapa hal sebagai berikut:

1. Konstruksi model *Extreme Learning Machine* yang dibangun oleh tiga layer yaitu *input layer*, *hidden layer*, dan *output layer* dengan masing-masing terdiri dari 2500 node, 1250 node, dan 3 node berhasil melakukan pengenalan objek citra digital dengan rata-rata tingkat akurasi mencapai 57,33% pada citra dengan objek berukuran kecil, 81,33% pada citra dengan objek berukuran sedang, dan 74,67% pada citra dengan objek berukuran besar.
2. Keakuratan sistem sangat dipengaruhi oleh ukuran objek dalam citra. Citra dengan objek berukuran sedang menghasilkan tingkat akurasi paling tinggi, sedangkan citra dengan objek berukuran kecil menghasilkan tingkat akurasi paling rendah. Penentuan dataset juga sangat berpengaruh dalam menentukan kemampuan ELM. Selain itu, proses *cropping* yang tidak tepat mengakibatkan citra hasil pra-proses tidak terfokus pada objek yang akan dikenali sehingga hasil pengenalan yang diberikan tidak tepat.
3. Penentuan arsitektur dari ELM juga mempengaruhi keakuratan sistem. Pemilihan yang kurang tepat untuk jumlah node pada *input layer*, *hidden layer*, dan *output layer* akan memberikan hasil pengenalan yang kurang akurat.

DAFTAR PUSTAKA

[1] Suartika, W., Wijaya, A.Y., dan Soelaiman, R. (2016). "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101". *Jurnal Teknik ITS Vol. 5*, Hal. A65-A69.

[2] Agustina, I. D., Anggraeni, W., Mukhlason, A. "Penerapan Metode Extreme Learning Machine untuk Peramalan Permintaan". Jurusan Sistem Informasi, Institut Teknologi Sepuluh Nopember, Surabaya.