

**IMPLEMENTASI JARINGAN SYARAF TIRUAN *BACKPROPAGATION*  
DENGAN ALGORITMA *CONJUGATE GRADIENT*  
UNTUK KLASIFIKASI KONDISI RUMAH  
(Studi Kasus di Kabupaten Cilacap Tahun 2018)**

**Johanes Roisa Prabowo<sup>1</sup>, Rukun Santoso<sup>2</sup>, Hasbi Yasin<sup>3</sup>**

<sup>1,2,3</sup>Departemen Statistika FSM Universitas Diponegoro

Email: [rukunsantoso25@gmail.com](mailto:rukunsantoso25@gmail.com)

**ABSTRACT**

House is one aspect of the welfare of society that must be met, because house is the main need for human life besides clothing and food. The condition of the house as a good shelter can be known from the structure and facilities of buildings. This research aims to analyze the classification of house conditions is livable or not livable. The method used is artificial neural networks (ANN). ANN is a system information processing that has characteristics similar to biological neural networks. In this research the optimization method used is the conjugate gradient algorithm. The data used are data of Survei Sosial Ekonomi Nasional (Susenas) March 2018 Kor Keterangan Perumahan for Cilacap Regency. The data is divided into training data and testing data with the proportion that gives the highest average accuracy is 90% for training data and 10% for testing data. The best architecture obtained a model consisting of 8 neurons in input layer, 10 neurons in hidden layer and 1 neuron in output layer. The activation function used are bipolar sigmoid in the hidden layer and binary sigmoid in the output layer. The results of the analysis showed that ANN works very well for classification on house conditions in Cilacap Regency with an average accuracy of 98.96% at the training stage and 97.58% at the testing stage.

**Keywords:** House, Classification, Artificial Neural Networks, Conjugate Gradient

**1. PENDAHULUAN**

Salah satu indikator kesejahteraan rakyat adalah rumah dan lingkungan hidup yang sehat. Rumah merupakan kebutuhan yang sangat mendasar bagi kehidupan setiap orang atau suatu rumah tangga yang harus dipenuhi, disamping pangan dan sandang. Dalam Undang-Undang Nomor 1 Tahun 2011 tentang Perumahan dan Kawasan Permukiman disebutkan dalam Pasal 1 Ayat 7 bahwa rumah adalah bangunan gedung yang berfungsi sebagai tempat tinggal yang layak huni, sarana pembinaan keluarga, cerminan harkat dan martabat penghuninya, serta aset bagi pemiliknya.

Namun kenyataan yang ada di masyarakat yaitu masih banyak rumah yang kondisinya belum memenuhi syarat rumah layak huni. Menurut data Badan Pusat Statistik pada tahun 2015, jumlah rumah tak layak huni sekitar 2,51 juta unit<sup>[4]</sup>. Hal tersebut dikarenakan faktor pendapatan masyarakat dan keterbatasan pengetahuan tentang fungsi rumah dan sarana prasarana yang dapat menyebabkan semakin sulit terwujudnya rumah layak huni. Oleh karena itu, pemerintah membuat sebuah program untuk membantu perbaikan rumah yang tidak layak huni. Tentu saja untuk memberikan bantuan agar tepat sasaran, pemerintah harus menentukan seseorang memiliki kondisi rumah yang layak huni atau tidak layak huni.

Salah satu metode yang dapat digunakan untuk mengklasifikasi kondisi rumah layak huni atau tidak layak huni yaitu jaringan syaraf tiruan (JST). JST merupakan sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan syaraf pada makhluk hidup<sup>[3]</sup>. Metode ini sering digunakan karena memiliki kemampuan tingkat keakuratan yang tinggi. Kecepatan dan ketepatan pelatihan jaringan dipengaruhi oleh inisialisasi bobot, arsitektur jaringan, nilai *error*, serta pemilihan metode optimasi<sup>[9]</sup>. Salah satu metode optimasi yang dapat digunakan adalah *conjugate gradient*. Metode optimasi ini mampu mempercepat kecepatan proses pelatihan dan meningkatkan akurasi pelatihan<sup>[7]</sup>.

## 2. TINJAUAN PUSTAKA

### 2.1. Rumah

Rumah merupakan ruang dimana keluarga beraktivitas, harus dapat memberi suasana kondusif bagi pembentukan generasi bangsa yang berkualitas<sup>[1]</sup>. Oleh karena itu, masalah rumah yang layak huni perlu mendapatkan perhatian khusus demi terciptanya kehidupan yang sejahtera. Rumah layak huni adalah rumah tangga yang memenuhi seluruh kriteria atau paling sedikit enam kriteria berikut ini<sup>[2]</sup>:

1. Luas lantai perkapita lebih dari 7,2 m<sup>2</sup>.
2. Jenis atap rumah terluas bukan ijuk atau rumbia.
3. Jenis dinding terluas bukan bambu atau lainnya.
4. Jenis lantai terluas bukan tanah.
5. Ada akses sanitasi layak.
6. Menggunakan penerangan listrik (PLN dan Non-PLN).
7. Mempunyai akses air minum layak.

### 2.2. Klasifikasi

Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia<sup>[6]</sup>. Pengukuran kinerja klasifikasi dilakukan dengan matriks konfusi (*confusion matrix*) seperti berikut:

**Tabel 1. Matriks Konfusi**

		Kelas Hasil Prediksi ( <i>j</i> )	
		Kelas = 1	Kelas = 0
Kelas asli ( <i>i</i> )	Kelas = 1	$f_{11}$	$f_{10}$
	Kelas = 0	$f_{01}$	$f_{00}$

Untuk menghitung akurasi dan laju error digunakan formula:

$$\text{Akurasi} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

### 2.3. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) merupakan sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi<sup>[3]</sup>. JST adalah sebuah mesin yang dirancang untuk memodelkan cara kerja otak manusia dalam mengerjakan tugas-tugas tertentu. Mesin ini memiliki kemampuan menyimpan pengetahuan berdasarkan pengalaman dan menjadikan pengetahuan yang sudah tersimpan menjadi bermanfaat<sup>[5]</sup>.

Secara garis besar JST mempunyai dua tahap pemrosesan informasi yaitu tahap pelatihan dan tahan pengujian. Tahap pelatihan (*training*) dimulai dengan memasukkan pola-pola pelatihan (data latih) ke dalam jaringan. Pola-pola pelatihan yang dimasukkan akan mengubah bobot yang menjadi penghubung antar *neuron*. Tahap pengujian (*testing*) merupakan pengujian suatu pola masukan yang belum pernah dilatihkan sebelumnya (data uji) menggunakan bobot-bobot yang telah dihasilkan pada tahap pelatihan.

*Neuron-neuron* dalam jaringan syaraf tiruan tersusun dalam bentuk lapisan-lapisan (*layer*) dan memiliki pola keterhubungan yang baik dalam satu lapisan maupun antar lapisan. Susunan dari *neuron-neuron* dalam lapisan dan pola keterhubungan antar lapisan disebut

sebagai arsitektur jaringan. JST memiliki beberapa bagian utama yang menyusunnya antara lain *input*, bias, fungsi aktivasi, *hidden layer*, dan *output*.

*Input layer* yaitu lapisan yang menerima masukan (*input*) dari jaringan luar dan mendistribusikan sinyal ke semua *neuron* pada *hidden layer*. *Hidden layer* yaitu lapisan yang berfungsi untuk menyalurkan informasi dari *input layer* ke *output layer* dimana sebelumnya informasi tersebut sudah dilakukan pemrosesan untuk memberikan *output* yang optimal. Hasil keluaran proses pada *hidden layer* disebut *output hidden layer*. Informasi *output hidden layer* ini selanjutnya diteruskan ke *output layer*. *Output* merupakan hasil dari semua perhitungan *input* dan bobot-bobot yang diberikan pada neuron yang saling berhubungan dengannya. Pada dasarnya *output* merupakan hasil akhir pemrosesan.

Fungsi aktivasi merupakan sebuah fungsi transfer yang berguna untuk menyesuaikan *output* yang dihasilkan oleh sistem jaringan syaraf sesuai dengan target, sehingga jaringan menjadi lebih kuat dalam mengenali pola<sup>[9]</sup>. Fungsi aktivasi yang biasa digunakan pada JST adalah sebagai berikut:

- Fungsi Sigmoid Biner

Fungsi sigmoid biner bernilai antara 0 sampai 1, fungsi sigmoid biner digunakan apabila *output* yang diinginkan merupakan bilangan riil antara 0 sampai 1. Fungsi sigmoid biner dinyatakan sebagai berikut:

$$f(x) = \frac{1}{1+e^{-x}} \quad \text{untuk } -\infty < x < \infty \quad \text{dengan } f'(x) = f(x)[1 - f(x)]$$

- Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, yang membedakannya adalah nilai *output* jaringan dari fungsi ini terletak pada interval -1 sampai 1. Fungsi sigmoid bipolar dirumuskan sebagai berikut:

$$f(x) = \frac{1-e^{-x}}{1+e^{-x}}, \text{ untuk } -\infty < x < \infty \quad \text{dengan } f'(x) = \frac{1}{2}[1 + f(x)][1 - f(x)]$$

- Fungsi Linear

Fungsi linear akan menghasilkan *output* yang sama dengan *input*nya. Fungsi ini sering dipakai jika diinginkan *output* jaringan berupa sebarang bilangan riil. Fungsi linear dirumuskan sebagai berikut:

$$f(x) = x, \text{ untuk } -\infty < x < \infty \quad \text{dengan } f'(x) = 1$$

## 2.4. Pre-Processing

Sebelum dilakukan pelatihan terhadap jaringan syaraf, seringkali diperlukan penskalaan pada *input* dan target sehingga data masuk pada *range* tertentu. *Pre-processing* merupakan proses mengubah skala data sebelum dilakukan tahap pelatihan agar data dapat diproses sesuai dengan fungsi aktivasi yang digunakan<sup>[9]</sup>. Pada penulisan tugas akhir ini, persamaan yang digunakan dalam *pre-processing* akan menghasilkan nilai yang berada pada interval [0.1, 0.9]. Mengingat bahwa fungsi aktivasi yang digunakan adalah fungsi sigmoid biner<sup>[8]</sup>. *Pre-processing* data dirumuskan dalam persamaan berikut:

$$x' = \frac{0,8(x-a)}{(b-a)} + 0,1$$

dimana  $x'$  : data hasil *pre-processing*

$x$  : data asli

$a$  : nilai minimum dari data asli

$b$  : nilai maksimum dari data asli

## 2.5. Backpropagation

*Backpropagation* merupakan suatu teknik pembelajaran atau pelatihan terawasi yang paling banyak digunakan. Metode ini merupakan salah satu metode yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks.

Proses pelatihan *backpropagation* meliputi tiga fase. Fase pertama adalah fase maju, dimana pada fase ini pola masukan dihitung maju mulai dari layar masukan hingga layar

keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur, dimana pada fase ini dilakukan perhitungan kesalahan. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi<sup>[9]</sup>.

Arsitektur jaringan *backpropagation* terdiri p unit pada *input layer*, m unit pada satu *hidden layer*, satu bias, dan satu *output*. Selain itu terdapat bobot penghubung antara *input nodes* dan *hidden nodes*, antara bias dan *hidden nodes*, serta antara *hidden nodes* dan *output*. Berikut adalah model jaringan *backpropagation* dengan m unit pada satu *hidden layer*:

$$y = \psi_o(w_{b1} + \sum_{j=1}^p w_{j1} \psi_h(v_{bj} + \sum_{i=1}^n v_{ij} x_i))$$

dengan:

$v_{bj}$ : bobot bias pada lapisan *input* ke lapisan tersembunyi ke-  $j$

$w_{b1}$ : bobot bias pada lapisan tersembunyi ke lapisan *output*

$v_{ij}$ : bobot pada lapisan *input* ke-  $i$  ke lapisan tersembunyi ke-  $j$

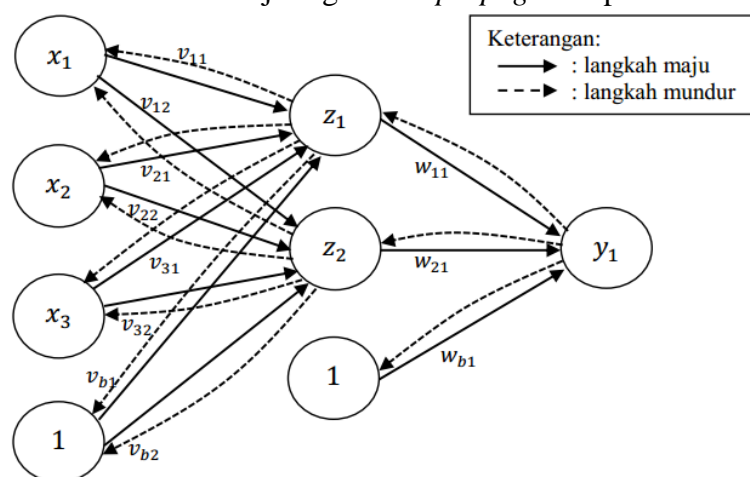
$w_{j1}$ : bobot pada lapisan tersembunyi ke-  $j$  ke lapisan *output*

$\psi_h$ : fungsi aktivasi yang digunakan pada lapisan tersembunyi

$\psi_o$ : fungsi aktivasi yang digunakan pada lapisan *output*

$x_i$ : *input* jaringan ke-  $i$

Berikut disajikan ilustrasi arsitektur jaringan *backpropagation* pada Gambar 1.



**Gambar 1. Arsitektur Jaringan Backpropagation**

Berikut algoritma dari jaringan syaraf *backpropagation*<sup>[9]</sup>:

Langkah 0 : Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil)

Langkah 1 : Jika kondisi pemberhentian belum terpenuhi ( $epoch < maximum\ epoch$  dan  $error\ target < MSE$ ), lakukan langkah 2-9

Langkah 2 : Untuk setiap pelatihan, lakukan langkah 3-8

**Feed Forward**

Langkah 3 : Tiap unit *input* ( $X_i, i = 1,2,3, \dots, n$ ) menerima sinyal *input*  $x_i$  dan meneruskan ke semua unit pada *hidden layer*.

Langkah 4 : Tiap unit pada *hidden layer* ( $Z_j, j = 1,2,3, \dots, p$ ) menjumlahkan bobot *input*

$$z\_in_j = v_{bj} + \sum_{i=1}^n x_i v_{ij}$$

menggunakan fungsi aktivasi untuk menghitung sinyal *output*

$$z_j = f(z\_in_j)$$

dan mengirimkan sinyal ini ke setiap unit *output*.

Langkah 5 : Unit *output* ( $Y_k, k = 1,2,3, \dots, m$ ) menjumlahkan bobot sinyal masuk

menggunakan fungsi aktivasi untuk menghitung sinyal *output*

$$y_k = f(y_{in_k})$$

### Error dari *backpropagation*

Langkah 6 : Unit *output* ( $Y_k, k = 1, 2, 3, \dots, m$ ) menerima pola yang sesuai dengan pola pelatihan *input*, menghitung informasi *error*

$$\delta_k = (t_k - y_k) f'(y_{in_k})$$

menghitung bobot terkoreksi (digunakan untuk memperbaharui  $w_{jo}$ )

$$\Delta w_{jk} = \eta \delta_k z_j$$

menghitung bias terkoreksi

$$\Delta w_{bk} = \eta \delta_k$$

dan mengirimkan  $\delta$  ke unit pada *layer* dibawahnya.

Langkah 7 : Tiap unit *hidden* ( $Z_j, j = 1, 2, 3, \dots, p$ ) menjumlahkan delta *input*

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

dikalikan dengan turunan fungsi aktivasi untuk menghitung *error*

$$\delta_j = \delta_{in_j} f'(z_{in_j})$$

hitung koreksi bobot

$$\Delta v_{ij} = \eta \delta_j x_i$$

hitung koreksi bias

$$\Delta v_{bj} = \eta \delta_j$$

### Perbaharui bobot dan bias

Langkah 8 : Unit *output* ( $Y_k, k = 1, 2, 3, \dots, m$ ) memperbaharui bobot dan bias

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$$

Tiap unit *hidden* ( $Z_j, j = 1, 2, 3, \dots, p$ ) memperbaharui bobot dan bias

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$$

Langkah 9 : Kondisi tes berhenti

## 2.6. Algoritma *Conjugate Gradient*

Pada algoritma *conjugate gradient*, pengaturan bobot disesuaikan dengan arah konjugasinya. Algoritma ini menggunakan pendekatan penemuan bobot optimal sepanjang arah gradien dengan fungsi *line search* untuk mencari arah gradien fungsi kinerja. Fungsi *line search* tersebut digunakan untuk menempatkan titik minimum  $\alpha$  sehingga dapat meminimumkan fungsi kinerja selama arah pencarian. Algoritma *conjugate gradient* dapat dideskripsikan sebagai berikut<sup>[9]</sup>:

Langkah 1 : a. Inisialisasi bobot dengan bilangan bobot acak kecil

b. Inisialisasi  $epoch = 0$  dan  $error = 0$

c. Tetapkan *maximum epoch* dan target *error*

Langkah 2 : Hitung  $g_j = \frac{\partial e}{\partial w_j}$ ,  $g_{b2} = \frac{\partial e}{\partial w_{b2}}$ ,  $g_{ij} = \frac{\partial e}{\partial v_{ij}}$ , dan  $g_{b1j} = \frac{\partial e}{\partial w_{b1j}}$

dengan  $e$  adalah selisih antara target dan *output* yang dihasilkan, sedangkan  $g_0$  merupakan vektor *gradient* fungsi kinerja pada waktu  $t$  yang berisi

$$g_{j(0)}, g_{b2(0)}, g_{ij(0)} \text{ dan } g_{b1j(0)}$$

Langkah 3 : Hitung arah pencarian awal  $d_0 = -g_0$

Langkah 4 : Hitung  $\alpha_t = -\frac{g_t^T d_t}{d_t^T Q d_t}$ , dengan  $d_t = -g_t$

Didefinisikan  $Q$  adalah matriks definit positif berukuran  $p \times p$  dengan  $Q^T = Q$ .

Langkah 5 : Ubah bobot vektor baru dengan rumus:

$$w_{t+1} = w_t + \alpha_t d_t \text{ dengan } d_t = -g_t$$

Langkah 6 : Jika  $MSE < Target Error$  atau  $epoch = \text{maksimum } epoch$ , maka kondisi penghentian terpenuhi. Jika tidak, lakukan langkah 7

Langkah 7 : Jika  $t + 1 \leq \text{maksimum } epoch$ , maka

a.  $t = t + 1$

b. Hitung  $g_{t+1}$  yang merupakan vektor berisi  $g_{j(t+1)}, g_{b2(t+1)}, g_{ij(t+1)}$  dan  $g_{b1j(t+1)}$

c. Masukkan nilai  $\beta_t$  dengan rumus:

$$\beta_t = \frac{g_{t+1}^T [g_{t+1} - g_t]}{g_t^T g_t}$$

d. Hitung arah pencarian yang baru dengan persamaan:

$$d_{t+1} = -g_{t+1} + \beta_t d_t$$

lanjutkan ke langkah 4.

### 3. METODOLOGI PENELITIAN

Data yang digunakan dalam tugas akhir ini merupakan data sekunder yaitu data Survei Sosial Ekonomi Nasional (Susenas) Maret 2018 Kor Keterangan Perumahan yang diambil dari Badan Pusat Statistik Provinsi Jawa Tengah untuk Kabupaten Cilacap. Variabel yang digunakan dibagi menjadi 2 yaitu variabel respon dan variabel prediktor.

1. Variabel respon:

Y = Klasifikasi kondisi rumah

2. Variabel-variabel prediktor:

X<sub>1</sub> = Luas lantai

X<sub>2</sub> = Jenis atap

X<sub>3</sub> = Jenis dinding

X<sub>4</sub> = Jenis lantai

X<sub>5</sub> = Fasilitas buang air besar

X<sub>6</sub> = Sumber penerangan

X<sub>7</sub> = Tempat pembuangan akhir tinja

X<sub>8</sub> = Sumber air minum

Software yang digunakan sebagai alat hitung adalah *Microsoft Excel 2013* dan *Matlab 2015a*. Adapun langkah-langkah dalam analisis data adalah sebagai berikut:

1. Menyiapkan data penelitian.
2. Membagi data menjadi 2, yaitu data latih dan data uji.
3. Memasukkan data *input* dan data target.
4. Melakukan *pre-processing*.
5. Menentukan jumlah *neuron* pada *hidden layer*.
6. Menentukan fungsi aktivasi.
7. Menentukan bobot awal, *max epoch* dan *goal*.
8. Membangun arsitektur jaringan.
9. Melakukan proses *training* dengan algoritma *conjugate gradient*.
10. Menentukan apakah MSE sudah lebih kecil dari *goal* yang ditetapkan atau jumlah *epoch* maksimal tercapai.
  - a. Jika iya, maka dilanjutkan dengan proses selanjutnya.
  - b. Jika tidak, maka kembali pada proses *training*.
11. Menentukan bobot akhir.
12. Melakukan proses *testing*.
13. Melakukan evaluasi hasil klasifikasi

### 4. ANALISIS DAN PEMBAHASAN

Sebelum pembelajaran *backpropagation* dilakukan, data dibagi menjadi dua terlebih dahulu yaitu data *training* dan data *testing*. Proporsi pembagian pada penelitian ini menggunakan 90% data *training* (894 data) dan 10% data *testing* (99 data). Sedangkan penentuan jumlah *neuron* pada *hidden layer* dilakukan dengan cara melakukan percobaan mulai dari 1 *neuron* sampai 10 *neuron*. Fungsi aktivasi yang digunakan yaitu fungsi sigmoid

bipolar pada *hidden layer* dan fungsi sigmoid biner pada *output layer*. Masing-masing jumlah *neuron* dicoba dengan 10 kali pengulangan.

**Tabel 2. Penentuan Jumlah Neuron pada Hidden Layer**

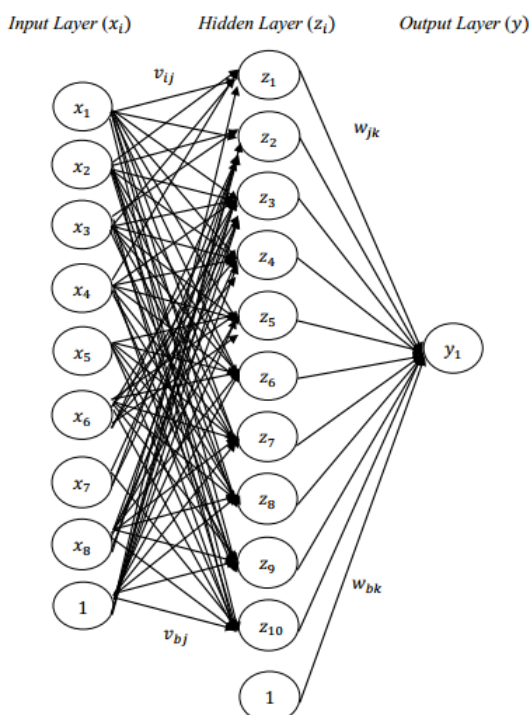
Jumlah Neuron	Rata-Rata MSE Training	Rata-Rata Akurasi Training
1	0.065996	0.93400
2	0.052237	0.94776
3	0.040828	0.95917
4	0.035570	0.96443
5	0.024944	0.97506
6	0.017785	0.98221
7	0.016219	0.98378
8	0.019463	0.98054
9	0.013423	0.98658
<b>10</b>	<b>0.010403</b>	<b>0.98960</b>

Berdasarkan Tabel 2, model terbaik dipilih berdasarkan rata-rata nilai akurasi yang terbesar yaitu pada *neuron* tersembunyi yang berjumlah 10 *neuron* dengan rata-rata nilai akurasi tertinggi 98,96%.

#### 4.1. Pembentukan Arsitektur Jaringan

Untuk membangun jaringan dengan algoritma *conjugate gradient* yang akan digunakan untuk klasifikasi kondisi rumah menggunakan perintah pada MATLAB berikut: `net=newff(minmax(pn), [10 1], {'tansig' 'logsig'}, 'traincgf')`

Pada perintah tersebut, terdapat Arsitektur model terbaik yang digunakan untuk klasifikasi kondisi rumah, dibangun dari 10 *neuron* pada *hidden layer* 1 *output layer*. Fungsi aktivasi yang digunakan pada *hidden layer* adalah sigmoid bipolar dan pada *output layer* adalah fungsi sigmoid biner. Arsitektur jaringan dapat dilihat pada Gambar 2.



**Gambar 2. Arsitektur Jaringan**

## 4.2. Tabulasi Hasil *Training* dan *Testing* Jaringan

Berikut ini disajikan tabel hasil *training* dan *testing* jaringan menggunakan algoritma *conjugate gradient* dengan 10 kali pengulangan.

**Tabel 3. Nilai Akurasi *Training* Jaringan**

Pengulangan	MSE <i>Training</i>	Akurasi <i>Training</i>
1	0.0078	0.9922
2	0.0246	0.9754
3	0.0123	0.9877
4	0.0067	0.9933
5	0.0101	0.9899
6	0.0145	0.9855
7	0.0067	0.9933
8	0.0101	0.9899
9	0.0022	0.9978
10	0.0089	0.9911
Rata-rata	0,0104	0,9896

**Tabel 4. Nilai Akurasi *Testing* Jaringan**

Pengulangan	MSE <i>Testing</i>	Akurasi <i>Testing</i>
1	0.0202	0.9798
2	0.0101	0.9899
3	0.0202	0.9798
4	0.0404	0.9596
5	0.0101	0.9899
6	0.0303	0.9697
7	0.0303	0.9697
8	0.0303	0.9697
9	0.0303	0.9697
10	0.0202	0.9798
Rata-rata	0,0242	0,9758

Berdasarkan Tabel 3 dan Tabel 4, diperoleh nilai rata-rata MSE *training* sebesar 0,0104 dan rata-rata akurasi *training* sebesar 0,9896 atau 98,96%. Sedangkan proses *testing* diperoleh nilai rata-rata MSE *testing* sebesar 0,0242 dan rata-rata akurasi *testing* sebesar 0,9758 atau 97,58%. Sehingga dapat dikatakan data kondisi rumah dapat diklasifikasikan dengan baik menggunakan jaringan syaraf tiruan *backpropagation* dengan algoritma *conjugate gradient*.

## 4.3. Penentuan Model

Arsitektur model terbaik dari jaringan syaraf tiruan *backpropagation* dengan algoritma *conjugate gradient* yang digunakan untuk klasifikasi kondisi rumah, dibangun dari 8 *neuron* pada *input layer*, 10 *neuron* pada *hidden layer* dan 1 *neuron* pada *output layer*. Fungsi aktivasi yang digunakan pada *hidden layer* adalah sigmoid bipolar dan pada *output layer* adalah fungsi sigmoid biner. Sehingga diperoleh persamaan model jaringan sebagai berikut:

$$y = \psi_o(w_{b1} + \sum_{j=1}^{10} w_{j1} \psi_h(v_{bj} + \sum_{i=1}^8 v_{ij} x_i))$$

dengan:

$v_{bj}$ : bobot bias pada lapisan *input* ke lapisan tersembunyi ke-  $j$

$w_{b1}$ : bobot bias pada lapisan tersembunyi ke lapisan *output*

$v_{ij}$ : bobot pada lapisan *input* ke-  $i$  ke lapisan tersembunyi ke-  $j$

$w_{j1}$ : bobot pada lapisan tersembunyi ke-  $j$  kelapisan *output*



$\psi_h$ : fungsi aktivasi yang digunakan pada lapisan tersembunyi  
 $\psi_o$ : fungsi aktivasi yang digunakan pada lapisan *output*  
 $x_i$ : *input* jaringan ke-  $i$

#### 4.4. Ketepatan Klasifikasi

Kinerja klasifikasi menggunakan jaringan syaraf tiruan dapat diukur dengan melihat akurasi yang dihasilkan, yaitu dengan membuat matriks konfusi. Berdasarkan isi dari matriks konfusi, dapat diketahui yaitu jumlah data yang diklasifikasikan secara tepat sebagai rumah layak huni (0) dan rumah tidak layak huni (1), maupun secara tidak tepat sebagai rumah layak huni (0) dan rumah tidak layak huni (1). Pengukuran ketepatan klasifikasi pada penelitian ini dilakukan pada data *testing*. Tabel 17 merupakan contoh matriks konfusi hasil klasifikasi.

**Tabel 5.** Matriks Konfusi Hasil Klasifikasi

Observasi	Prediksi	
	Tidak Layak Huni	Layak Huni
Tidak Layak Huni	9	3
Layak Huni	0	87

Berdasarkan Tabel 5, diperoleh nilai akurasi sebagai berikut:

$$\text{Akurasi} = \frac{87 + 9}{9 + 3 + 0 + 87} = \frac{96}{99} = 0,9697 \text{ atau } 96,97\%$$

## 5. KESIMPULAN

Berdasarkan hasil analisis dan pembahasan pada bab sebelumnya, dapat diambil beberapa kesimpulan bahwa arsitektur model terbaik yang digunakan untuk klasifikasi kondisi rumah, dibangun dari 8 *neuron* pada *input layer*, 10 *neuron* pada *hidden layer* dan 1 *neuron* pada *output layer*. Berdasarkan model tersebut, jaringan syaraf tiruan *backpropagation* dengan algoritma *conjugate gradient* mampu mengklasifikasikan kondisi rumah dalam kategori layak huni atau tidak layak huni dengan hasil yang baik, yaitu diperoleh rata-rata ketepatan klasifikasi sebesar 98,96% pada proses *training* dan 97,58% pada proses *testing*.

## DAFTAR PUSTAKA

- [1] [BPS] Badan Pusat Statistik. (2018). *Indikator Perumahan dan Kesehatan Lingkungan 2018*. Jakarta: BPS.
- [2] [BPS] Badan Pusat Statistik. (2018). *Statistik dan Peranan Penyediaan Perumahan dan Pemukiman yang Layak Terhadap Kemiskinan di Jawa Tengah*. Diakses 18 Agustus 2019, dari [http://biroinfrasda.jatengprov.go.id/files/uploads/2018/02/Statistik-PERUMAHAN-2017-jateng\\_rev.pdf](http://biroinfrasda.jatengprov.go.id/files/uploads/2018/02/Statistik-PERUMAHAN-2017-jateng_rev.pdf)
- [3] Fausett, L. (1994). *Fundamentals of Neural Networks: Architecture, Algorithms and Applications*. New Jersey: Prentice-Hall.
- [4] Fitra, S. (2016). *Pemerintah Akan Pangkas 610 Ribu Rumah Tak Layak Huni Hingga 2019*. Diakses dari katadata.co.id: <https://katadata.co.id/berita/2016/10/10/pemerintah-akan-entaskan-24-persen-rumah-tak-layak-huni-hingga-2019>
- [5] Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. New York: Macmillan Publishing Company.
- [6] Prasetyo, E. (2012). *Data Mining: Konsep dan Aplikasi Menggunakan Matlab*. Yogyakarta: Andi.
- [7] Sari, Y. (2016). *Optimasi Conjugate Gradient pada Algoritma Backpropagation Neural Network untuk Prediksi Kurs Time Series*. *GEMA AKTUALITA*, Vol. 5 No. 1, 86-90.
- [8] Siang, J. J. (2005). *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta: Andi.
- [9] Warsito. (2009). *Kapita Selekta Statistika Neural Network*. Semarang: BP Universitas Diponegoro.