

Jaringan Syaraf Tiruan Pengenalan Pola Rambu Lalu Lintas Dengan Metode Kohonen

Dara Phonna *, Dr.Zulfian Azmi,St.,M.Kom**,
Ardianto Pranata,S.Kom.,M.Kom**

* Program Studi Sistem Informasi, STMIK Triguna Dharma

** Program Studi Sistem Informasi, STMIK Triguna Dharma

Article Info	ABSTRACT
Article history:	<i>Rambu lalu lintas memberikan arahan penggunaan jalan raya, gambar yang ditunjukkan dapat menurunkan tingkat kecelakaan jalan raya. Gambar rambu lalu lintas yang kurang jelas karena sudah using berisiko menimbulkan kecelakaan. Sehingga dibutuhkan sebuah pengetahuan yang dapat memberikan penjelasan gambar/pola rambu lalu lintas yang sudah kurang baik.</i>
Keyword: Jaringan Syaraf Tiruan Rambu Lalu Lintas Algoritma Kohonen	<p><i>Dari permasalahan yang terjadi, dengan menggunakan keilmuan jaringan saraf tiruan yang menerapkan metode kohonen diharapkan dapat membantu dalam penyelesaian pengenalan pola rambu lalu lintas dengan membangun sebuah sistem berbasis android yang dapat digunakan perangkat smartphone Android.</i></p> <p><i>Hasil dari penelitian ini menciptakan system berbasis android yang mudah dan efisien dalam penggunaannya karena tidak membutuhkan paket data atau koneksi internet.</i></p>
Corresponding Author: Nama :Dhara Phonna Program Studi : Sistem Informasi STMIK Triguna Dharma Email :daraphonna626@gmail.com	<p style="text-align: right;"><i>Copyright © 2020 STMIK Triguna Dharma. All rights reserved.</i></p>

1. PENDAHULUAN

Transportasi merupakan salah satu penunjang aktivitas manusia masa kini yang semakin pesat berkembang akan transportasi, maka semakin efektif manusia dalam pemenuhan kebutuhan hidupnya. Namun, disisi lain meningkat pula tingkat kecelakaan di jalan raya ini yang dapat mengakibatkan masyarakat kurang memiliki pengetahuan tentang rambu lalu lintas seperti yang tertera di undang-undang [1]. Ada banyak rambu lalu lintas yang sudah tidak layak pasang di jalanan sehingga menyebabkan ketidak tahuan para pengendara dalam melakukan perjalanan. Dengan mengetahui tanda rambu lalu lintas diharapkan kendaraan bisa memberikan informasi kepada pengemudi mengenai rambu-rambu yang ada disekitarnya, sehingga dapat mengurangi pelanggaran lalu lintas yang tidak disengaja [2]. Oleh karena itu dibutuhkan suatu sistem computer dan kamera yang diletakkan pada kendaraan pengemudi. Peletakan kamera bisa didesain sesuai dengan arah mana yang ingin ditangkap gambarnya. Proses pendeteksian gambar rambu-rambu lalu lintas dilakukan dengan memanfaatkan ciri atau fitur yang ada pada gambar tersebut yang terlihat dominan dengan menggunakan algoritma jaringan syaraf tiruan.

Didalam teknologi informasi jaringan syaraf tiruan adalah sistem perangkat keras atau perangkat lunak pengenalan pola yang cara kerjanya meniru cara kerja otak manusia. JST disusun

oleh elemen-elemen pemroses yang berada pada lapisan-lapisan yang berhubungan dan diberi bobot[1]. Teknologi ini umumnya digunakan untuk pemrosesan sinyal yang kompleks dalam pengenalan pola. Ada beberapa metode JST yang dapat digunakan untuk pengenalan karakter. Salah satu metode yang dapat digunakan adalah metode kohonen. Kohonen merupakan salah satu tipe metode dengan melakukan pengenalan pola dengan cara mengelompokkan pola-pola yang hamper sama.

2. METODE PENELITIAN

2.1 Jaringan Syaraf Tiruan

Jaringan saraf tiruan (JST) merupakan sistem pemrosesan informasi yang memiliki karakteristik serupa dengan jaringan saraf biologis dan terdiri dari sejumlah besar elemen pemrosesan sederhana yang disebut *neuron* unit, sel, atau *node*[2]. Setiap *neuron* terhubung ke *neuron* lain dengan sarana hubungan komunikasi terarah, masing-masing dengan bobot yang saling terkait. Bobot merupakan informasi yang digunakan oleh jaringan untuk memecahkan masalah". Jaringan Saraf Tiruan (JST) merupakan suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai Jaringan Saraf Biologis (JSB)[14]. JST tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia (*human cognition*).

2.1 Rambu Lalu Lintas

Rambu lalu lintas adalah salah satu dari perlengkapan jalan, berupa lambang, huruf, angka, kalimat dan/atau perpaduan diantaranya sebagai peringatan, larangan, perintah atau petunjuk bagi pemakai jalan[6].

2.3 Metode Kohonen

Self Organizing Map (SOM) atau yang sering disebut dengan kohonen merupakan suatu metode jaringan saraf tiruan yang diperkenalkan oleh Professor Teuvo Kohonen[15]. Jaringan Kohonen merupakan salah satu bentuk topologi dari *Unsupervised Artificial Neural Network* dimana dalam proses pelatihannya tidak memerlukan pengawasan. Jaringan *Kohonen* merupakan salah satu teknik *Neural Network*[3]. Pada jaringan ini neuron-neuron pada suatu lapisan akan menyusun dirinya sendiri berdasarkan nilai input tertentu dalam suatu cluster[4]. Jaringan kohonen digunakan untuk mengelompokkan (*clustering*) data berdasarkan karakteristik/fitur-fitur data.

Langkah-langkah metode kohonen adalah sebagai berikut :

1. Initialization: Initialize the cluster center $Y_{i,0}$ ($1 \leq i \leq c$), learning rate $\beta_{ij,t}$

($0 \leq \beta_{ij,t} \leq 1$), threshold ϵ ($\epsilon > 0$) and topological neighborhood parameters. Set $t = 1$ and maximum iteration limit t_{max} .

2. Selection of winner : Calculate the squared Euclidean distance :

$$d^2_{ij,t} = \|I_{ru,j} - Y_{i,t}\|^2 \text{ for } j = 1, 2, \dots, n \text{ and } i = 1, 2, \dots, c$$

Winning output neuron is decided by

$$\min \{ d^2_{ij,t} \} \text{ for } j = 1, 2, \dots, n \text{ and } i = 1, 2, \dots, c$$

3. Weight Updation : The weight of the winner neuron is updated by

$$Y_{i,t} = Y_{i,t-1} + \beta_{ij,t} (I_{ru,j} - Y_{i,t-1})$$

Where $\beta_{ij,t}$ is learning rate.

4. If $\|Y_{i,t} - Y_{i,t-1}\| > \epsilon$ set $t = t + 1$, update learning rate go to step 2 else stop.

Berikut adalah terjemahan dari algoritma tersebut :

1. Inisialisasi : Inisialisasi *Cluster center* $Y_{i,0}$ ($1 \leq i \leq c$), *learning rate* $\beta_{ij,t}$

($0 \leq \beta_{ij,t} \leq 1$), *threshold* ϵ ($\epsilon > 0$) dan parameter topologi hubungan ketetangaan. Atur $t = 1$ dan batas maksimum iterasi t_{max} .

2. Pemilihan pemenang : Hitung menggunakan Euclidean distance :

$$d_{ij,t}^2 = \|I_{fu,j} - Y_{i,t}\|^2 \text{ for } j = 1, 2, \dots, n \text{ dan } i = 1, 2, \dots, c$$

Neuron keluaran pemenang ditentukan dengan

$$\min \{ d_{ij,t}^2 \} \text{ for } j = 1, 2, \dots, n \text{ dan } i = 1, 2, \dots, c$$

3. Perbarui bobot : Bobot neuron pemenang diperbarui dengan

$$Y_{i,t} = Y_{i,t-1} + \beta_{ij,t} (I_{fu,j} - Y_{i,t-1})$$

Dimana $\beta_{ij,t}$ adalah *learning rate*.

4. Jika $\|Y_{i,t} - Y_{i,t-1}\| > \epsilon$ atur $t = t + 1$, perbarui *learning rate*. Ulangi langkah ke-2 sampai kondisi berhenti.

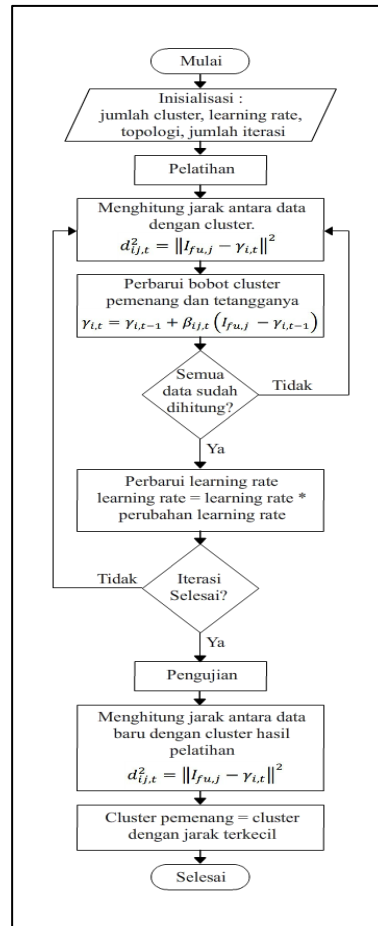
3. ANALISA DAN HASIL

3.1 Analisa Permasalahan

Pada saat kita berkendara, banyak rambu lalu lintas yang sudah tidak layak untuk dipasang dijalanan, sehingga menyebabkan ketidaktahuan para pengendara dalam melakukan perjalanan. Hal tersebut dapat merugikan orang lain. Dan disini lain meningkatnya tingkat kecelakaan dijalan raya ini terjadi karena kurangnya pengetahuan masyarakat tentang peraturan rambu lalu lintas yang tertera di undang-undang. Oleh karena itu, dibutuhkan suatu system yang berupa system computer dan kamera yang diletakkan pada kendaraan pengemudi. Rambu lalu lintas merupakan salah satu alat pelengkap jalan dalam bentuk tertentu dengan lambing, huruf, angka, kalimat atau perpaduan diantaranya. Di Indonesia, rambu lalu lintas digunakan sebagai peringatan, larangan, perintah dan petunjuk bagi pengguna jalan. Hal ini banyak menyebabkan kendaraan lalu lalang dijalan raya sering tidak memperhatikan rambu lalu lintas yang menyebabkan seringnya terjadi kecelakaan.

3.2 Algoritma Sistem

Ada beberapa tahapan agar aplikasi yang akan dibuat dapat melakukan proses pengenalan pola rambu lalu lintas. Dalam flowchart, proses akan diperlihatkan penggambaran secara grafik langkah-langkah proses pelatihan dan pengujian pada metode kohonen. Berikut adalah flowchart algoritma kohonen.



Gambar 3.2 flowchart proses pelatihan dan pengujian metode kohonen

Berikut adalah tahapan-tahapan dalam metode kohonen :

1. Tahap pengolahan citra, sebelum melakukan perhitungan, citra harus diubah kedalam bentuk biner terlebih dahulu, berikut adalah tahapannya :
 - a. *Contrast/Brightness*, merupakan proses mengatur pencahayaan citra
 - b. *Grayscale*, merupakan proses mengubah citra warna menjadi citra hitam putih
 - c. *Monokrom*, merupakan proses mengubah citra menjadi hanya memiliki dua kemungkinan piksel yaitu hitam dan putih.
 - d. *Cropping*, merupakan proses memotong bagian kosong citra
 - e. *Resize*, merupakan proses merubah ukuran citra, proses ini dibutuhkan karena ukuran citra yang terlalu besar dapat memperlambat proses pengenalan karakter.
 - f. *Threshold*. Merupakan proses untuk mengubah citra menjadi *array* biner.
2. Tahap pelatihan, tahap ini setiap data rambu lalu lintas yang akan dilatih dan dihitung menggunakan algoritma pelatihan kohonen. Dalam pelatihan ini huruf yang mirip akan dikelompokkan (Cluster). Bobot cluster hasil dari pelatihan tersebut akan digunakan dalam tahap pengujian. Dari beberapa rambu akan diambil 2 saja yang akan menjadi contoh pelatihan. Berikut adalah contohnya :

$$\begin{aligned} \text{cluster}(5,3) &= 0.50 + 0.60(0 - 0.50) = 0.20 \\ \text{cluster}(5,4) &= 0.50 + 0.60(0 - 0.50) = 0.20 \\ \text{cluster}(5,5) &= 0.50 + 0.60(0 - 0.50) = 0.20 \\ \text{cluster}(5,400) &= 0.50 + 0.60(0 - 0.50) = 0.20 \end{aligned}$$

jarak data 2 ke:

$$\begin{aligned} \text{cluster ke-1} &= (0.2 - 0)^2 + (0.2 - 0)^2 + (0.2 - 0)^2 + \dots + (0.2 - 0)^2 = 55.6 \\ \text{cluster ke-2} &= (0.2 - 0)^2 + (0.2 - 0)^2 + (0.2 - 0)^2 + \dots + (0.2 - 0)^2 = 55.6 \\ \text{cluster ke-3} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \\ \text{cluster ke-4} &= (0.2 - 0)^2 + (0.2 - 0)^2 + (0.2 - 0)^2 + \dots + (0.2 - 0)^2 = 55.6 \\ \text{cluster ke-5} &= (0.2 - 0)^2 + (0.2 - 0)^2 + (0.2 - 0)^2 + \dots + (0.2 - 0)^2 = 55.6 \\ \text{cluster ke-6} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \\ \text{cluster ke-7} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \\ \text{cluster ke-8} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \\ \text{cluster ke-9} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \end{aligned}$$

jarak terkecil ke cluster 1

Perbarui cluster ke 1 2 4 5

$$\begin{aligned} \text{cluster}(1,1) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(1,2) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(1,3) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(1,4) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(1,5) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(1,400) &= 0.20 + 0.60(0 - 0.20) = 0.08 \end{aligned}$$

$$\begin{aligned} \text{cluster}(2,1) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(2,2) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(2,3) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(2,4) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(2,5) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(2,400) &= 0.20 + 0.60(0 - 0.20) = 0.08 \end{aligned}$$


$$\begin{aligned} \text{cluster}(4,1) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(4,2) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(4,3) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(4,4) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(4,5) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(4,400) &= 0.20 + 0.60(0 - 0.20) = 0.08 \end{aligned}$$

$$\begin{aligned} \text{cluster}(5,1) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(5,2) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(5,3) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(5,4) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(5,5) &= 0.20 + 0.60(0 - 0.20) = 0.08 \\ \text{cluster}(5,400) &= 0.20 + 0.60(0 - 0.20) = 0.08 \end{aligned}$$

Table 3.2 Hasil Perhitungan Iterasi Ke-1

Data	Cluster								
	1	2	3	4	5	6	7	8	9
1	100	100	100	100	100	100	100	100	100
2	55.6	55.6	100	55.6	55.6	100	100	100	100
3	68.9	68.9	100	68.9	68.9	100	100	100	100
4	51.5	51.5	100	51.5	51.5	100	100	100	100

Keterangan :

 Jarak terkecil

$Learning\ rate = 0.60 * 0.81 = 0.49$

Table 3.3 Hasil Perhitungan Iterasi Ke-2

Data	Cluster								
	1	2	3	4	5	6	7	8	9
1	54.9	54.9	100	54.9	54.9	100	100	100	100
2	52.5	52.5	100	52.5	52.5	100	100	100	100
3	53.0	53.0	100	53.0	53.0	100	100	100	100
4	44.9	44.9	100	44.9	44.9	100	100	100	100

$Learning\ rate = 0.49 * 0.81 = 0.39$

Setelah iterasi ke 20 maka didapatkan bobot akhir *cluster* sebagai berikut :

cluster ke-1 :

0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.24 0.00 0.00 0.00 0.00 0.00 0.26 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.24 0.24 0.24 0.00 0.00 0.00 0.26 0.26 0.26 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.24 0.24 0.24 0.24 0.24 0.00 0.26 0.26 0.26 0.26 0.26 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.24 0.24 0.51 0.24 0.24 0.24 0.51 0.26 0.26 0.26 0.26 0.49 0.26 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.27 0.51 0.24 0.24 0.00 0.00 0.00 0.26 0.26 0.26 0.22 0.22 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.27 0.27 0.78 0.78 0.51 0.49 0.49 0.49 1.00 0.73 0.73 0.22 0.22 0.22 0.00 0.00
 0.00 0.00 0.00 0.27 0.27 0.78 0.78 0.78 0.49 0.74 0.74 1.00 1.00 0.73 0.22 0.22 0.22 0.00 0.00
 0.00 0.00 0.00 0.27 0.27 0.78 0.78 0.78 1.00 1.00 0.74 1.00 1.00 0.73 0.22 0.22 0.22 0.00 0.00
 0.00 0.00 0.00 0.27 0.78 0.51 0.51 0.73 1.00 1.00 0.78 0.78 0.51 0.22 0.22 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.78 0.51 0.51 0.73 1.00 1.00 0.78 0.78 0.51 0.22 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.51 0.51 0.51 0.73 1.00 0.76 0.78 0.78 0.51 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.51 0.51 0.51 0.22 0.49 0.76 0.78 0.78 0.51 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.26 0.26 0.26 0.22 0.49 0.49 0.51 0.78 0.51 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.26 0.26 0.26 0.22 0.49 0.49 0.51 0.51 0.24 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.26 0.26 0.26 0.22 0.49 0.49 0.51 0.51 0.24 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

cluster ke-2 :

0.00
 0.00 0.00 0.00 0.00 0.00 0.00 0.24 0.00 0.00 0.00 0.00 0.00 0.26 0.00 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.24 0.24 0.24 0.00 0.00 0.00 0.26 0.26 0.26 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.24 0.24 0.24 0.24 0.24 0.00 0.26 0.26 0.26 0.26 0.26 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.24 0.24 0.45 0.24 0.24 0.24 0.51 0.26 0.26 0.26 0.26 0.55 0.26 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.21 0.45 0.24 0.24 0.00 0.00 0.00 0.26 0.26 0.26 0.29 0.29 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.21 0.21 0.71 0.71 0.45 0.49 0.49 0.49 1.00 0.79 0.79 0.29 0.29 0.29 0.00 0.00 0.00
 0.00 0.00 0.00 0.21 0.21 0.71 0.71 0.71 0.49 0.74 0.74 1.00 1.00 0.79 0.29 0.29 0.29 0.29 0.00 0.00
 0.00 0.00 0.00 0.21 0.21 0.71 0.71 0.71 1.00 1.00 0.74 1.00 1.00 0.79 0.29 0.29 0.29 0.00 0.00 0.00
 0.00 0.00 0.00 0.21 0.71 0.51 0.51 0.79 1.00 1.00 0.71 0.71 0.51 0.29 0.29 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.71 0.51 0.51 0.79 1.00 1.00 0.71 0.71 0.51 0.29 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.51 0.51 0.51 0.79 1.00 0.76 0.71 0.71 0.51 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.51 0.51 0.51 0.29 0.49 0.76 0.71 0.71 0.51 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.26 0.26 0.26 0.29 0.49 0.49 0.45 0.71 0.51 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.26 0.26 0.26 0.29 0.49 0.49 0.45 0.45 0.24 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.26 0.26 0.26 0.29 0.49 0.49 0.45 0.45 0.24 0.00 0.00 0.00 0.00 0.00 0.00
 0.00 0.00 0.00 0.00 0.00 0.26 0.26 0.26 0.29 0.49 0.49 0.45 0.45 0.24 0.00 0.00 0.00 0.00 0.00 0.00
 0.00

0.00
 0.00

Setelah bobot akhir didapatkan, maka proses selanjutnya adalah mengelompokkan data rambu yang sebelumnya dilatih. Proses ini dilakukan dengan cara menghitung jarak antar *cluster* dengan data rambu yang sebelumnya dilatih.

Jarak data 1 rambutikungan tajam kekiri (A) ke :

$$\begin{aligned} \text{cluster ke-1} &= (0.0 - 0)^2 + (0.0 - 0)^2 + (0.0 - 0)^2 + \dots + (0.0 - 0)^2 = 24.0 \\ \text{cluster ke-2} &= (0.0 - 0)^2 + (0.0 - 0)^2 + (0.0 - 0)^2 + \dots + (0.0 - 0)^2 = 27.4 \\ \text{cluster ke-3} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \\ \text{cluster ke-4} &= (0.0 - 0)^2 + (0.0 - 0)^2 + (0.0 - 0)^2 + \dots + (0.0 - 0)^2 = 28.1 \\ \text{cluster ke-5} &= (0.0 - 0)^2 + (0.0 - 0)^2 + (0.0 - 0)^2 + \dots + (0.0 - 0)^2 = 29.2 \\ \text{cluster ke-6} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \\ \text{cluster ke-7} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \\ \text{cluster ke-8} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \\ \text{cluster ke-9} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \end{aligned}$$

Jarak terkecil ke *cluster*1, maka *cluster* 1 termasuk rambu (A)

Jarak data 2 rambutikungan tajam kekanan (B) ke :

$$\begin{aligned} \text{cluster ke-1} &= (0.0 - 0)^2 + (0.0 - 0)^2 + (0.0 - 0)^2 + \dots + (0.0 - 0)^2 = 38.4 \\ \text{cluster ke-2} &= (0.0 - 0)^2 + (0.0 - 0)^2 + (0.0 - 0)^2 + \dots + (0.0 - 0)^2 = 33.8 \\ \text{cluster ke-3} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \\ \text{cluster ke-4} &= (0.0 - 0)^2 + (0.0 - 0)^2 + (0.0 - 0)^2 + \dots + (0.0 - 0)^2 = 40.5 \\ \text{cluster ke-5} &= (0.0 - 0)^2 + (0.0 - 0)^2 + (0.0 - 0)^2 + \dots + (0.0 - 0)^2 = 40.3 \\ \text{cluster ke-6} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \\ \text{cluster ke-7} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \\ \text{cluster ke-8} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \\ \text{cluster ke-9} &= (0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0 \end{aligned}$$

Jarak terkecil ke *cluster*2, maka *cluster* 2 termasuk rambu (B)

Berikut adalah hasil dari pengelompokan data :

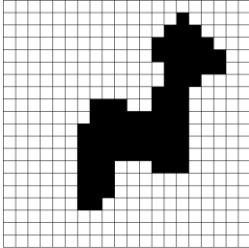
Tabel 3.4 Hasil Pengelompokan Data

No.	Rambu	Cluster
1	Rambu tikungan tajam kekiri (A)	1
2	Rambu tikungan tajam kekanan (B)	2
3	Rambu tikungan ganda, tikungan pertama kekiri (C)	4
4	Rambu tikungan ganda, tikungan pertama kekanan (D)	5

3. Tahap Pengujian

Tahap pengujian adalah proses mengenali rambu yang sudah dilatih tetapi memiliki bentuk yang berbeda. Pengujian dilakukan dengan cara menghitung jarak antara *cluster* akhir dari tahap pelatanganan data rambu yang akan diuji. Berikut adalah data rambu yang akan diuji :

Tabel 3.5 Rambu Yang Akan Diuji

No.	Rambu	Gambar	Array Biner
1.	Rambu tikungan ganda, tikungan pertama kekanan		<pre> 00000000000000000000 000000000000010000 000000000000111000 000000000000111100 000000000000111110 000000000000111110 000000000000111000 000000000000110000 000000111001110000 000000111111110000 00000111111110000 00000111111110000 00000111111110000 00000111000110000 00000111000000000 00000111000000000 00000110000000000 00000000000000000 00000000000000000 00000000000000000 </pre>

Untuk menghitung jarak antara data baru dengan cluster digunakan rumus :

$$d^2_{ij,t} = \|I_{fu,j} - Y_{i,t}\|^2$$

Hasil ditentukan oleh cluster dengan jarak terkecil.

Jarak data rambu baru ke:

- clusterke-1 = $(0.0 - 0)^2 + (0.0 - 0)^2 + (0.0 - 0)^2 + \dots + (0.0 - 0)^2 = 41.9$
- clusterke-2 = $(0.0 - 0)^2 + (0.0 - 0)^2 + (0.0 - 0)^2 + \dots + (0.0 - 0)^2 = 40.8$
- clusterke-3 = $(0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0$
- clusterke-4 = $(0.0 - 0)^2 + (0.0 - 0)^2 + (0.0 - 0)^2 + \dots + (0.0 - 0)^2 = 40.7$
- clusterke-5 = $(0.0 - 0)^2 + (0.0 - 0)^2 + (0.0 - 0)^2 + \dots + (0.0 - 0)^2 = 37.5$
- clusterke-6 = $(0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0$
- clusterke-7 = $(0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0$
- clusterke-8 = $(0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0$
- clusterke-9 = $(0.5 - 0)^2 + (0.5 - 0)^2 + (0.5 - 0)^2 + \dots + (0.5 - 0)^2 = 100.0$

Jarak terkecil ke cluster5.

Berdasarkan pengelompokan data pada tahap pelatihan, clusterke-5 adalah Rambu tikungan ganda, tikungan pertama kekanan. Maka rambu ini berhasil dikenali walaupun memiliki bentuk yang berbeda.

4.1 ANALISA DAN PERANCANGAN

4.1 Kebutuhan Sistem

Dalam perancangan jaringan syaraf tiruan untuk merancang aplikasi rambu lalu lintas membutuhkan beberapa fasilitas pendukung. Berikut ini fasilitas pendukung yang dibutuhkan oleh sistem, baik perangkat keras maupun perangkat lunak.

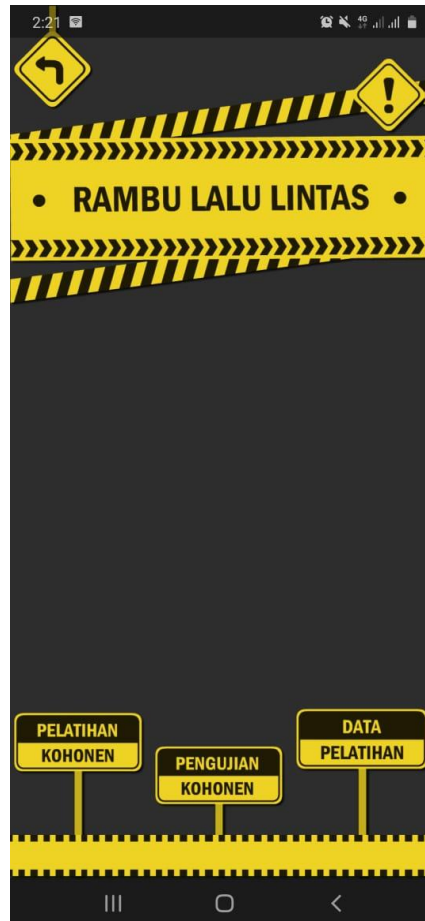
1. Perangkat Lunak(*Software*)
 - a. Sistem Operasi *Windows 7, Windows 8*
 - b. *Android Studio*
 - c. *Emulator*
2. Perangkat Keras(*Hardware*)
 - a. Komputer, Laptop, HP android (OS Nugget)
 - b. *Processor Minimal Intel Dual Core Processor*
 - c. Sistem Operasi Minimal *Windows 7*
 - d. *Ram Minimal 4 Gb*
 - e. *Harddisk Minimal 500 Gb*

4.2 Implementasi Sistem

Implementasi merupakan langkah yang digunakan untuk mengoperasikan sistem yang akan dibangun. Dalam bab ini akan dijelaskan bagaimana menjalankan sistem yang telah dibangun tersebut. Dibawah ini merupakan tampilan dari implementasi Jaringan Saraf Tiruan dengan Metode Kohonen untuk pengenalan rambu lalu lintas.

1. Tampilan Utama Aplikasi

Halaman menu utama berfungsi sebagai tempat menu dan halaman *default* suatu aplikasi yang dibangun, halaman ini memiliki beberapa menu untuk memanggil halaman lainnya. Berikut ini adalah tampilan halaman menu utama:

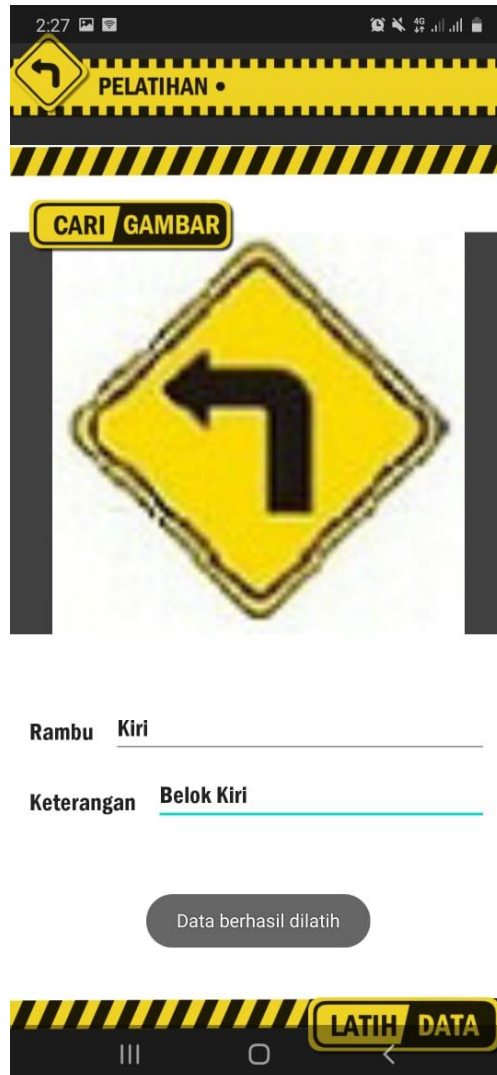


Gambar 5.1 Tampilan Menu Utama

Tampilan menu utama terdiri dari beberapa menu, Pelatihan Kohonen, Pengujian Kohonen, Data Pelatihan dan Informasi Aplikasi.

2. Tampilan Halaman Pelatihan Kohonen

Halaman ini memiliki fungsi sebagai pelatihan gambar rambu lalu lintas yang gambar tersebut nantinya akan diuji dengan gambar yang pada halaman pengujian. Berikut ini adalah tampilan halaman pelatihan Kohonen:

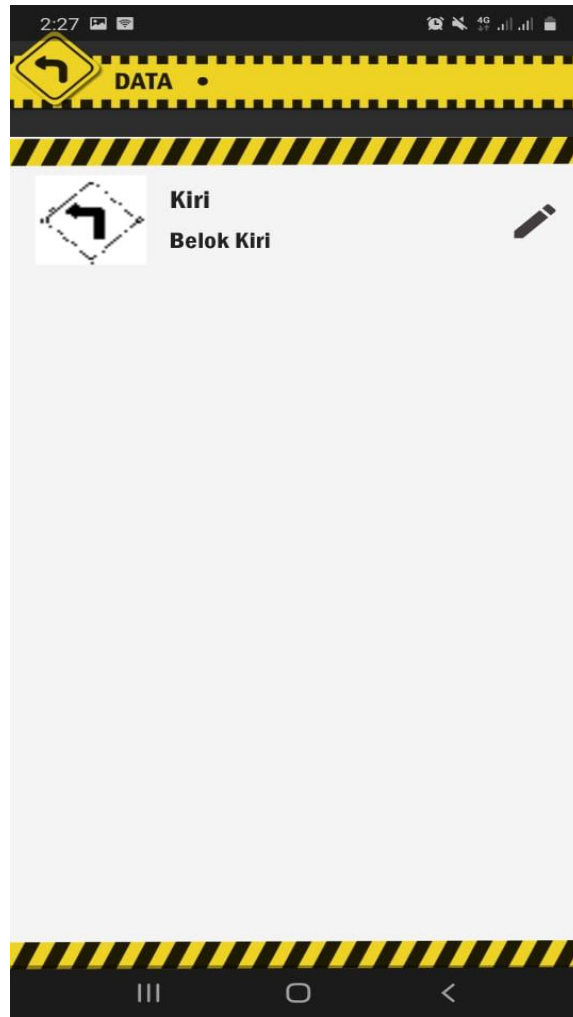


Gambar 5.2 Tampilan Menu *Input Data Alternatif*

Pada halaman ini, klik Cari Gambar untuk mencari gambar rambu yang akan dilatih, kemudian input nama rambu dan keterangan, lalu klik latih data untuk melakukan pelatihan pada gambar yang sudah ditentukan.

3. Tampilan Halaman Data Pelatihan

Halaman ini berfungsi untuk menampilkan dan menghapus data pelatihan yang sudah dilatih pada form pelatihan sebelumnya, berikut tampilan form data yang sudah dilakukan pelatihan :



Gambar 5.3 Tampilan Data Pelatihan

4. Tampilan Halaman Pengujian

Halaman ini berfungsi untuk menampilkan hasil pengujian rambu lalu lintas. Gambar yang digunakan merupakan gambar lalu lintas yang sudah kurang baik gambarnya, kemudian akan dilakukan pengujian terhadap data gambar rambu lalu lintas yang sudah dilatih pada halaman sebelumnya. Berikut tampilan halaman Pengujian Kohonen:



Hasil

Dari hasil pengujian, gambar tersebut adalah rambu :

Kiri
dengan persentase (6.8%)

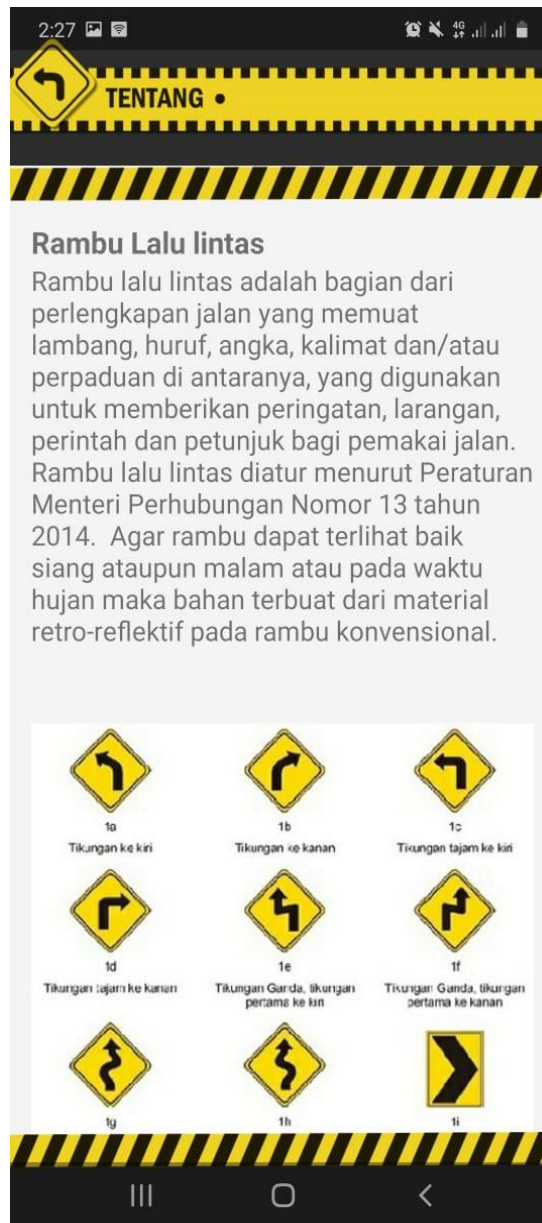


Gambar 5.4 Tampilan Pengujian Kohonen

Dari gambar yang sudah dilatih, kemudian mengambil gambar yang akan diuji, dengan metode kohonen maka dapat hasil pengujian dengan persentase 6,8 % yang menyatakan gambar tersebut adalah rambu Kiri.

5. Tampilan Halaman Informasi

Halaman ini digunakan untuk menampilkan informasi mengenai penjelasan rambu lalu lintas dan beberapa gambar rambu lalu lintas. Berikut tampilan halaman informasi pada aplikasi:



Gambar 5.5 Halaman Informasi

a. KESIMPULAN

Dari hasil penelitian yang dilakukan maka dapat di ambil beberapa kesimpulan sebagai berikut:

1. Kohonen dapat mengenali perbedaangambar rambu dengan cara menghitung jarak antara pola gambar rambu tersebut dengan pola yang sudah dikelompokkan pada saat pelatihan.
2. Dalam merancang sistem pengenalan pola rambu lalulintas menggunakan metode kohonen dibutuhkan pengolahan citra yang berfungsi untuk mengubah citra menjadi *array* biner, dimana *array* biner ini yang akan di hitung menggunakan algoritma kohonen.
3. Cara kerja metode kohonen pada sistem pengenalan pola rambu yaitu dengan cara mengelompokkan semua pola rambu yang sama, sehingga apabila ditemukan pola rambu yang sama tetapi memiliki bentuk yang berbeda maka kohonen masih dapat mengenalinya.
4. Bahasa pemrograman android dapat digunakan untuk membuat aplikasi pengenalan pola rambu lalulintas menggunakan metode kohonen.



UCAPAN TERIMA KASIH

Terima kasih kepada dosen pembimbing bapak Jaka Prayudha, S.Kom., M.Kom, dan bapak Drs. Sobirin, S.H., M.Si, dan beserta pihak – pihak yang lainnya yang mendukung penyelesaian jurnal skripsi ini.

DAFTAR PUSTAKA

1. Tim Pengajar Universitas Sumatera Utara. *Landasan Teori*. Diunduh tanggal 5 Januari 2013, <http://repository.usu.ac.id/bitstream/123456789/23794/4/Chapter%20II.pdf>.
2. Aji, S. 2016. Jaringan Syaraf Tiruan untuk memprediksi laju pertumbuhan penduduk menggunakan metode Backpropagation (Studi Kasus Di kota Bengkulu). *Jurnal Media Informa*, 12(1):61-69.
3. Azmi, Z., Taufik, F. and Susilo, B., 2018. Implementasi Jaringan Kohonen Dalam Pengenalan Citra Huruf Aksara Jawa. *Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer)*, 17(2), pp.214-217.
4. Edy, I., & Faisal, M. 2005. *Advanced Clustering Teoridan Aplikasi*. Sleman: Deepublish.
5. Heru, M. 2016. Pengenalan pola HIV dan AIDS menggunakan algoritma Kohonen pada Jaringan Saraf Tiruan Backpropagation. *Jurnal Nasional Informatika dan Teknologi Jaringan*, 1(1):65-69.

BIBLIOGRAFI PENULIS

	<p>Data Diri</p> <p>Nama : dara phonna Tempat/Tanggal Lahir : tg.morawa 24 april 1997 Jenis Kelamin : Perempuan Agama : Islam Status : Belum Menikah Pendidikan Terakhir : Sekolah Menengah kejuruan Kewarganegaraan : Indonesia E-mail : daraphonna626@gmail.com</p>
	<p>Nama : dr.zulfian azmi,s.t.,m.kom NIDN : 0116067304 Jabatan : Dosen E-mail : zulfian.azmi@gmail.com</p>



Nama	: Ardianto Pranata,S.Kom.,M.Kom
NIDN	: 0112029101
Jabatan	: Dosen
E-mail	: ardianto_pranata@gmail.com00