

Analisis Optimasi Algoritma Backpropagation Momentum Dalam Memprediksi Jenis Tingkat Kejahatan Di Kecamatan Tambusai Utara

Budi Yanto¹, Almadison², Hendri³, Ramses Hutagaol⁴, Ripatullah Rahman⁵

^{1,5}Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pasir Pengaraian

^{2,3}Program Studi Ilmu Hukum, Fakultas Hukum, Universitas Pasir Pengaraian

⁴STKIP Rokania

Email : ¹budiyantost@gmail.com, ²hendri.jaiz@gmail.com, ³almadison03@gmail.com
⁴ramshutagaol06@gmail.com, ⁵ripatullahrahman@gmail.com

Abstract : *Crime is conduct that disobeys the law and contravenes social norms in a way that society finds objectionable. There is no system that can forecast the kind and quantity of crimes that will happen in the future as a guide in the process of preventing crime, according to the North Tambusai Police. However, the public service process in the form of complaints has been done digitally. Backpropagation is an iterative method that works well even with complex and convoluted data. Backpropagation is an artificial neural network with several levels (multi-layer). Data processing is done on the types and numbers of crimes that took place in North Tambusai District between 2015 and 2020. The first step in the data processing procedure is to normalize the data and choose the network training parameters. Crime data and levels were used in the artificial neural network research, which used a 5-5-1 design. The test results reveal that the average prediction accuracy rate is 92.66%, with the greatest prediction accuracy rate being 99.6% and the lowest forecast accuracy rate being 90.01 percent. Theft had the highest weighting (Curat) of crimes the next year with 15 cases, while fraud, crime, and extortion/threats each had the lowest weighting (1 case). The prediction findings exhibit a sufficiently high level of accuracy to serve as a basis for evaluation.*

Kata Kunci : *Backpropagation Algorithm, Kind Crime, Prediction Forecasting,*

I. PENDAHULUAN

Kejahatan adalah kelakuan sikap yang membangkang sifat dan membangkang etik-etik sosial, Beberapa elemen yang menimbulkan seseorang mengamalkan tingkah ulah kriminal, diantaranya adalah konflik dan pertarungan kebudayaan, oposisi aliran politik, konsistensi dan tatanan musik penduduk, oposisi kuota kebudayaan, oposisi substansi dan pendapatan, mentalitas yang labil, kintil elemen pokok seumpama elemen biologis, psikis dan sosio emosional kintil elemen ekonomi [1]. Penindakan sifat tindak durjana yang dilakukan seseorang dilakukan oleh Kepolisian Negara Republik Indonesia (POLRI) yang menakhlikkan suatu perlengkapan kosmos yang berkedudukan bagian dalam menanggung kekompakan dan peraturan umum, membuat sifat, kintil menyerahkan perlindungan, sokongan dan servis untuk umum bagian dalam skema terpeliharanya kekompakan bagian dalam negeri [2]. Secara kriminologi berbasis sosiologis, kejahatan diartikan sebagai pola tingkah laku yang dapat meresahkan maupun merugikan masyarakat sebagai korban kejahatan.

Adapun beberapa bentuk tindak kriminalitas yang sering terjadi yakni pencurian, penganiayaan dan pembunuhan.[3]

Penelitian yang dilakukan memerlukan kepada prakiraan taraf kejahatan dan rupa kejahatan di Kecamatan Tambusai Utara dalam masa 2020 mendatang berdalil petunjuk taraf kelaliman dan rupa kelaliman pecah perian 2015-2020, petunjuk yang dibutuhkan bagian dalam mengamalkan pemeriksaan adalah petunjuk taraf kelaliman dan rupa kelaliman di Kecamatan Tambusai Utara pecah perian 2015 terakhir, petunjuk terselip didapatkan pecah Kepolisian Sektor Tambusai Utara. Proses penyimpanan petunjuk bagian dalam mengantisipasi taraf dan rupa kelaliman dimasa mendatang mengabdikan rel backpropagation, taraf prakiraan yang dihasilkan ditentukan oleh pengarsipan rel yang tepat, bagian dalam perijuz ini rel yang tusukan dan akurat bagian dalam mengantisipasi taraf kelaliman dan rupa kelaliman yaitu jalinan saraf model backpropagation momentum.

Backpropagation terhitung bagian dalam ordo kaitan saraf replika yang tenar dan memegang baki bagian dalam taktik bimbingan atau pendidikan,

taktik Pendidikan atau pembelajarannya berulang pakai memegang komputasi yang hormat terutama tempo fakta disajikan [4] bagian dalam tahap sketsa dan kompleks [5]. Sedangkan bagian dalam kaitan saraf replika berkehendak yang digunakan kepada memuat panduan untuk kaitan sehingga mengerjakan deformasi muatan pakai memikirkan etik error dekat epoch sebelumnya. Sehingga pakai adanya pejaka, etik muatan yang diubah oleh JST dekat fase pendidikan tidak semata-mata karena etik error dekat epoch yang sedang terjadi, namun juga karena epoch sebelumnya. Penelitian model persangkaan dilakukan oleh Sema Nabillah Dewi, pakai mengabdikan norma extreme learning machine. Penelitian dilakukan pakai mengabdikan fakta kejahatan Polres Probolinggo bersumber perian 2012 sangkut pakai perian 2017. Berdasarkan analisis yang dilakukan membuat bentuk kaitan yang maksimum yaitu nilai fitur sejumlah 7, kiasan daya usaha fakta yaitu 80%:20%, dan nilai neuron dekat hidden layer sejumlah 7 pakai Mean Square Error (MSE) yaitu sebanyak 0,037662 [3]. Penelitian sebelumnya juga dilakukan oleh Eka Irawan, ihwal telaahan penumpukan etik pejaka dekat pengandaian inspirasi kelambir sawit mengabdikan backpropagation. Dari ekoran analisis menyinggir bahwa algoritma backpropagation pakai penumpukan etik pejaka semata-mata mencengkam 727 epoch pakai etik MSE 0,01, sedangkan algoritma backpropagation ukuran mencengkam 4000 epoch pakai etik MSE 0,001 [6].

Algoritma backpropagation memiliki arsitektur berupa input layer, hidden layer dan output layer, arsitektur backpropagation sangat mempengaruhi tingkat keakurasian yang dihasilkan, sehingga harus memilih arsitektur yang tepat. Perancangan arsitektur dalam hal ini adalah menentukan jumlah layer pada input layer, hidden layer dan output layer [7].

Pada dasarnya algoritma backpropagation terdiri atas tiga tahapan yaitu:

1. Tahap feedforward yaitu memasukkan data pelatihan berupa variabel untuk mendapatkan nilai output.
2. Tahap backpropagation yaitu proses propagasi balik dari nilai error yang diperoleh.
3. Menghitung minimal nilai error untuk menyesuaikan bobotnya.

Ketiga tahapan tersebut diulang terus-menerus hingga didapatkan nilai error yang diinginkan. Setelah proses training selesai, hanya tahap pertama yang diperlukan untuk memanfaatkan jaringan saraf tiruan tersebut [8]. Berikut tahapan-tahapan pelatihan dengan algoritma backpropagation:

1. Inisialisasi bobot (ambil bobot awal dengan nilai acak yang cukup kecil),

dapat dilakukan dengan proses acak atau menggunakan algoritma Nguyen-Widrow.

2. Menentukan maksimum iterasi, learning rate dan nilai minimum error yang diinginkan.
3. Jika kondisi berhenti bernilai false, lakukan langkah-langkah berikut:

1. Tahap feedforward

- a. Setiap neuron input x_i dengan $i = 1,2,3,\dots,n$ mendapatkan sinyal x_i dan diteruskan ke semua neuron pada lapisan tersembunyi.

- b. Setiap neuron pada suatu lapisan tersembunyi z_j dengan $j = 1,2,3,\dots,p$ menjumlahkan sinyal-sinyal input berbobot:

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots (1)$$

Hitung sinyal output pada lapisan tersembunyi ketika menerapkan fungsi aktivasi:

$$z_j = \frac{1}{1+e^{-z_{in_j}}} \dots\dots\dots (2)$$

Kemudian kirimkan sinyal yang diperoleh ke semua neuron pada lapisan output.

- c. Setiap neuron output y_k dengan $k = 1,2,3,\dots,m$ menjumlahkan sinyal-sinyal input berbobot:

$$y_{in_k} = v_{0k} + \sum_{j=1}^p z_j w_{jk} \dots\dots\dots (3)$$

menggunakan fungsi aktivasi maka hitung sinyal output pada lapisan output dengan persamaan:

$$y_k = \frac{1}{1+e^{-y_{in_k}}} \dots\dots\dots (4)$$

Catatan: pada langkah ke-2 lakukan perulangan sampai semua nilai hidden layer diperoleh.

2. Tahap backpropagation

- a. Setiap neuron output y_k dengan $k = 1,2,3,\dots,m$, menerima target pola output yang berkaitan dengan pola input dengan pelatihan. Hitung error pada lapisan output:

$$\delta_k = (t_k - y_k) \left(\frac{1}{1+e^{-z_{in_j}}} \right) \left(1 - \left(\frac{1}{1+e^{-y_{in_k}}} \right) \right) \dots\dots\dots (5)$$

δ_k adalah target output yang diharapkan. Selanjutnya hitung perubahan bobot (digunakan untuk memperbaiki nilai bobot w_{jk}):

$$\Delta w_{jk} = \alpha \delta_k z_j \dots\dots\dots (6)$$

Kirimkan δ_k ini ke neuron pada lapisan tersembunyi.

- b. Setiap neuron pada lapisan tersembunyi z_j dengan $j = 1,2,3,\dots,p$ menjumlahkan faktor delta pada lapisan tersembunyi:

$$\delta_{in_k} = \sum_{k=1}^m \delta_k w_{jk} \dots\dots\dots (7)$$

nilai ini dikalikan dengan turunan dari fungsi aktivasinya dan digunakan untuk menghitung informasi error pada lapisan tersembunyi:

$$\delta_k = \delta_{in_k} \left(\frac{1}{1+e^{-z_{in_j}}} \right) \left(1 - \left(\frac{1}{1+e^{-z_{in_j}}} \right) \right) \dots\dots\dots(8)$$

Selanjutnya hitung perubahan bobot pada setiap unit keluaran (*output*).

$$\Delta v_{jk} = \alpha \delta_j x_i \dots\dots\dots(9)$$

Selanjutnya hitung perubahan bobot *input layer* ke *hidden layer*.

$$v_{ij}(\text{Baru}) = v_{ij}(\text{Lama}) + \Delta v_{jk} \dots\dots\dots(10)$$

c. Setiap *neuron output* y_k dengan $k = 1, 2, 3, \dots, m$, memperbaharui bobot *hidden layer* ke *output layer* ($j = 0, 1, 2, \dots, p$):

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \dots\dots\dots(11)$$

Setiap neuron pada lapisan tersembunyi z_j , dengan memperbaharui bobot dan biasnya ($i = 0, 1, 2, \dots, n$):

$$v_{ij} * = v_{ij} + \Delta v_{ij} \dots\dots\dots(12)$$

Menghitung minimal nilai *error* untuk menyesuaikan bobotnya (tes kondisi berhenti). Algoritma *backpropagation* menggunakan jumlah maksimum iterasi dan nilai minimum *error* yang ditentukan dalam proses kondisi penghentian pembelajaran. Proses pelatihan akan berhenti ketika jumlah iterasi pelatihan melebihi jumlah maksimum iterasi atau jika nilai *error* yang diperoleh dalam pelatihan lebih kecil dari nilai minimum *error* yang ditentukan.

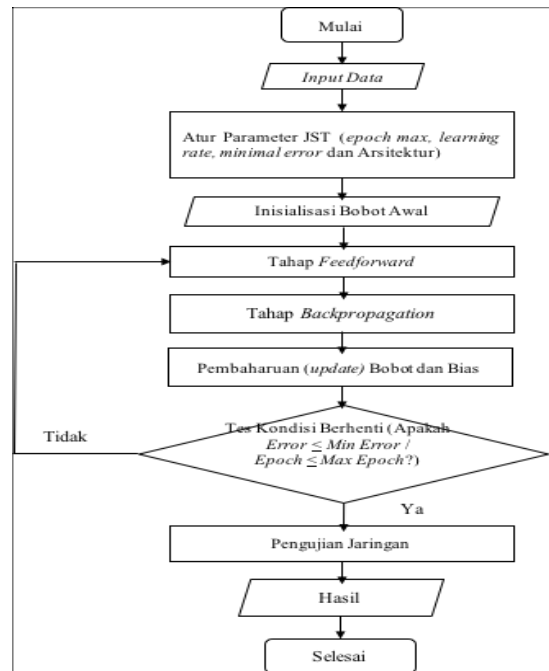
Berdasarkan fungsi aktivasi yang digunakan maka harus dilakukan normalisasi data sehingga keluaran jaringan sesuai dengan fungsi aktivasi.[9] Penelitian menggunakan fungsi aktivasi *sigmoid biner*. Fungsi *sigmoid biner* merupakan fungsi tidak pernah mencapai 0 ataupun 1, sehingga data ditransformasikan tidak pernah mencapai 0 ataupun 1. Normalisasi data yang digunakan menggunakan persamaan:

$$x' = \frac{0.8(x-x_{min})}{(x_{max}-x_{min})} + 0.1 \dots\dots\dots(13)$$

Dimana (x') merupakan normalisasi data, (x_{min}) merupakan data terendah dan (x_{max}) merupakan data tertinggi.

II. METODE PENELITIAN

Normalisasi data adalah suatu metode transformasi data asli yang memiliki masalah tertentu sehingga menghasilkan data baru yang minim kesalahan seperti data ganda atau ketidakkonsistenan data.[10] Permasalahan-permasalahan yang terjadi pada data mentah sering disebut dengan istilah anomali. Anomali adalah terjadinya ketidakkonsistenan data dan terdapat data ganda yang menyebabkan data tidak dapat didefinisikan oleh system.



Gambar 1. Kerangka Kerja Penelitian

Input data yang digunakan yaitu data jenis dan jumlah kejahatan yang terjadi di Kecamatan Tambusai Utara selanjutnya data tersebut akan diolah oleh jaringan saraf tiruan dengan metode *backpropogation momentum*. Agar data dapat dikenali oleh metode *backpropagation momentum* yang menggunakan fungsi aktivasi *sigmoid biner* (tidak pernah mencapai 0 atau pun 1), maka data harus direpresentasikan terlebih dahulu kedalam bentuk fungsi aktivasi *sigmoid biner* yang akan digunakan sebagai pengenalan pola dan keluaran untuk menghasilkan prediksi jenis dan jumlah kejahatan dimasa mendatang. Data yang digunakan dalam penelitian adalah data jenis dan jumlah kejahatan di Kecamatan Tambusai Utara selama tahun 2015 sampai dengan tahun 2020. Data jenis dan jumlah kejahatan yang terjadi pada tahun 2015 sampai dengan tahun 2019 menjadi variabel data *input*, sedangkan data jenis dan jumlah kejahatan tahun 2020 menjadi target, untuk lebih jelas dapat dilihat pada Tabel 1.

Tabel 1. Variabel Data *Input*

No	Variabel	Inisialisasi
1	2015	X1
2	2016	X2
3	2017	X3
4	2018	X4
5	2019	X5
6	2020	Target

III. HASIL PENELITIAN DAN PEMBAHASAN

Data yang diambil dalam penelitian adalah data jenis dan jumlah kejahatan yang terjadi di Kecamatan Tambusai Utara yaitu dari tahun 2015 sampai dengan tahun 2020. Proses pengumpulan data dilakukan dengan cara wawancara kepada pihak Kepolisian Sektor Tambusai Utara. Setelah melakukan wawancara kemudian pihak Kepolisian

Sektor Tambusai Utara memberikan data jenis dan jumlah kejahatan yang diperlukan dalam format *microsoft excel* (.xlsx). Data yang sudah terkumpul kemudian disusun dalam bentuk tabel agar mudah dimengerti dan dipahami dalam tahapan pemrosesan dan menganalisa data. Data jenis dan jumlah kejahatan yang akan diolah atau diproses dengan menggunakan metode *backpropagation momentum* dapat dilihat pada Tabel

Tabel 4. Rekapitulasi Hasil Uji Instrumen

JENIS KEJAHATAN	2015	2016	2017	2018	2019	2020
PERJUDIAN	8	5	9	7	4	3
CURAS	3	2	5	4	3	3
PENGANIAYAAN	13	11	18	12	13	12
CURAT	8	10	6	14	9	16
PENCABULAN ANAK	7	6	5	9	7	3
PENGERUSAKAN	3	1	2	4	2	1
NARKOTIKA	15	17	11	16	14	9
CURANMOR	4	3	2	3	1	1
KDRT	2	1	3	2	1	1
PENGGELAPAN	5	2	2	5	3	3
PENIPUAN	2	2	1	1	0	1
PEMERASAN / ANCAMAN	1	0	2	1	1	1

(Sumber: POLSEK Tambusai Utara)

1. Normalisasi Data / Transformasi Data

Sistem prediksi jenis dan jumlah kejahatan di Kecamatan Tambusai Utara dengan metode *backpropagation momentum* menggunakan fungsi aktivasi *sigmoid biner*. Fungsi aktivasi *sigmoid biner* merupakan fungsi yang tidak pernah mencapai 0 atau pun 1. Data jenis dan jumlah kejahatan perlu dilakukan normalisasi terlebih dahulu, proses normalisasi yaitu dengan mengubah data pada interval tidak pernah mencapai 0 atau pun 1 dengan menggunakan persamaan:

$$x' = \frac{0.8(x-x_{min})}{(x_{max}-x_{min})} + 0.1 \dots\dots\dots(13)$$

Dimana (x') merupakan normalisasi data, (x) merupakan data awal, (x_{min}) merupakan data terendah dan (x_{max}) merupakan data tertinggi. Proses normalisasi data yaitu dengan menentukan nilai tertinggi dan terendah pada data jenis dan jumlah kejahatan. Data jenis dan jumlah kejahatan yang terjadi selama tahun 2015 sampai dengan tahun 2020 disajikan setiap tahunnya. Berikut ini ditampilkan data jenis dan jumlah kejahatan untuk proses pelatihan jaringan dan pengujian jaringan sebelum normalisasi pada Tabel 3

Tabel 3. Data Tingkat dan Jenis Kejahatan Sebelum Normalisasi

JENIS KEJAHATAN	2015	2016	2017	2018	2019	2020
PERJUDIAN	8	5	9	7	4	3
CURAS	3	2	5	4	3	3
PENGANIAYAAN	13	11	18	12	13	12
CURAT	8	10	6	14	9	16
PENCABULAN ANAK	7	6	5	9	7	3
PENGERUSAKAN	3	1	2	4	2	1
NARKOTIKA	15	17	11	16	14	9
CURANMOR	4	3	2	3	1	1
KDRT	2	1	3	2	1	1
PENGGELAPAN	5	2	2	5	3	3
PENIPUAN	2	2	1	1	0	1
PEMERASAN ANCAMAN /	1	0	2	1	1	1

Data *input* yang disajikan pada Tabel 3 lalu ditransformasikan kedalam bentuk fungsi aktivasi *sigmoid biner* pada interval tidak pernah mencapai 0 ataupun 1 menggunakan rumus persamaan [24] dengan menentukan nilai tertinggi (x_{max}) dan nilai terendah (x_{min}) terlebih dahulu, selanjutnya dilakukan proses normalisasi data, dalam hal ini mengambil contoh perhitungan data bulan januari.

1. Data *input* tahun 2015

$$\begin{aligned}
 X &= 8 \\
 X_{max} &= 18 \\
 X_{min} &= 0 \\
 x' &= \frac{0.8(8 - 0)}{(18 - 0)} + 0.1 = 0,456
 \end{aligned}$$

2. Data *input* tahun 2016

$$\begin{aligned}
 X &= 5 \\
 X_{max} &= 18 \\
 X_{min} &= 0 \\
 x' &= \frac{0.8(5 - 0)}{(18 - 0)} + 0.1 = 0,322
 \end{aligned}$$

3. Data *input* tahun 2017)

$$\begin{aligned}
 X &= 9 \\
 X_{max} &= 18 \\
 X_{min} &= 0
 \end{aligned}$$

$$x' = \frac{0.8(9 - 0)}{(18 - 0)} + 0.1 = 0,500$$

4. Data *input* tahun 2018

$$\begin{aligned}
 X &= 7 \\
 X_{max} &= 18 \\
 X_{min} &= 0 \\
 x' &= \frac{0.8(7 - 0)}{(18 - 0)} + 0.1 = 0,411
 \end{aligned}$$

5. Data *input* tahun 2019

$$\begin{aligned}
 X &= 4 \\
 X_{max} &= 18 \\
 X_{min} &= 0 \\
 x' &= \frac{0.8(4 - 0)}{(18 - 0)} + 0.1 = 0,278
 \end{aligned}$$

6. Data *input* tahun 2020

$$\begin{aligned}
 X &= 3 \\
 X_{max} &= 18 \\
 X_{min} &= 0 \\
 x' &= \frac{0.8(3 - 0)}{(18 - 0)} + 0.1 = 0,233
 \end{aligned}$$

Proses perhitungan normalisasi terhadap semua data jenis dan jumlah kejahatan di Kecamatan Tambusai Utara dari tahun 2015 sampai dengan tahun 2020 disajikan pada Tabel 3.

Tabel 4 Data Tingkat dan Jenis Kejahatan Setelah Normalisasi

JENIS KEJAHATAN	2015	2016	2017	2018	2019	2020
PERJUDIAN	0.456	0.322	0.500	0.411	0.278	0.233
CURAS	0.233	0.189	0.322	0.278	0.233	0.233
PENGANIAYAAN	0.678	0.589	0.900	0.633	0.678	0.633
CURAT	0.456	0.544	0.367	0.722	0.500	0.811
PENCABULAN ANAK	0.411	0.367	0.322	0.500	0.411	0.233
PENGERUSAKAN	0.233	0.144	0.189	0.278	0.189	0.144
NARKOTIKA	0.767	0.856	0.589	0.811	0.722	0.500
CURANMOR	0.278	0.233	0.189	0.233	0.144	0.144
KDRT	0.189	0.144	0.233	0.189	0.144	0.144
PENGGELAPAN	0.322	0.189	0.189	0.322	0.233	0.233
PENIPUAN	0.189	0.189	0.144	0.144	0.100	0.144
PEMERASAN ANCAMAN /	0.144	0.100	0.189	0.144	0.144	0.144

IV. Hasil dan Pembahasan

1. Tahap Awal

Data jenis kejahatan Perjudian dari tahun 2015 sampai dengan tahun 2020. Data sampel dalam perancangan manual sebagai pengetahuan awal tahapan-tahapan metode *backpropagation momentum* dapat dilihat pada Tabel 6.

Tabel 6 Data Sampel Dalam Perancangan Manual

Jenis Kejahatan	X1	X2	X3	X4	X5	T
Perjudian	8	5	9	7	3	9

Selanjutnya sampel data yang telah ternormalisasi menggunakan fungsi aktivasi *sigmoid biner*. Data ternormalisasi pada sampel data dapat dilihat pada Tabel 7.

Tabel 7. Data Sampel Dalam Perancangan Manual Ternormalisasi

Jenis Kejahatan	X1	X2	X3	X4	X5	T
Perjudian	0,456	0,322	0,500	0,411	0,278	0,233

Setelah proses normalisasi data dilakukan yang akan digunakan dalam pelatihan JST, tahap selanjutnya adalah inialisasi bobot awal. Untuk arsitektur jaringan yang telah dirancang sebelumnya, bobot awal yang menghubungkan

Tabel 8. Bobot Awal Dari *Input Layer* Menuju *Hidden Layer*

Bobot	$z1$	$z2$	$z3$	$z4$
$v1$	0,32	0,76	0,22	0,14
$v2$	0,27	0,83	0,17	0,38
$v3$	0,45	0,51	0,44	0,47
$v4$	0,16	0,82	0,75	0,13
$v5$	0,35	0,98	0,37	0,26

input layer menuju *hidden layer* adalah $v = (v11, v12, v13, v14, v15, v21, v22, v23, v24, v25, v31, v32, v33, v34, v35, v41, v42, v43, v44, v45)$. Sementara bobot yang menghubungkan *hidden layer* menuju *output layer* adalah $w = (w1, w2, w3, w4, w5)$. Inialisasi bobot awal adalah nilai bobot yang diberikan secara acak dengan nilai yang relatif kecil. Tahap selanjutnya memasukkan nilai-nilai parameter pelatihan JST dengan metode *backpropagation momentum* dengan arsitektur jaringan 5-5-1 dengan sampel perancangan manual data jenis dan jumlah kejahatan dari tahun 2015 sampai dengan tahun 2020. Inialisasi bobot awal yang diberikan adalah sebagai berikut:

1. Sampel data *input* pada data jenis kejahatan Perjudian tahun 2015 sampai dengan tahun 2020
 $x1 = 0,456$
 $x2 = 0,322$
 $x3 = 0,500$
 $x4 = 0,411$
 $x5 = 0,278$
 T (Target) = 0,233
Learning rate (α) = 0,1
2. Nilai bobot awal yang diberikan secara acak untuk menghubungkan *input layer* menuju *hidden layer*.

3. Nilai bobot awal yang diberikan secara acak dari *hidden layer* menuju *output*

layer.

Tabel 9 Bobot Awal Dari *Hidden Layer* Menuju *Output Layer*

Bobot	Y
w1	0,32
w 2	0,17
w 3	0,31
w 4	0,74
w 5	0,47

4. Nilai *bias* (v_j) yang diberikan secara acak dari *input layer* menuju ke *hidden layer*.

Tabel 10 Nilai *Bias* Dari *Input Layer* Menuju *Hidden Layer*

Bias	v_j
1	0,26
2	0,73
3	0,69
4	0,42
5	0,93

5. Nilai *bias* (w_k) yang diberikan secara acak dari *hidden layer* menuju *output layer*

Tabel 11. Nilai *Bias* Dari *Hidden Layer* Menuju *Output Layer*

Bias	1
w_k	0,68

2. Tahap *Feedforward*

Tahap *feedforward* merupakan tahap aliran perambatan maju dimana setiap *neuron input layer* menerima sinyal dari data *input* yang kemudian diteruskan menuju *neuron hidden layer* dan dari *neuron hidden layer* diteruskan kembali menuju *neuron output layer*. Tahap *feedforward* dilakukan berulang hingga semua nilai *hidden layer* diperoleh, selanjutnya hitung semua keluaran dari lapisan tersembunyi Z_j ($j = 1, 2, \dots, p$). Adapun langkah-langkah pada tahap *feedforward* adalah sebagai berikut:

1. Setiap *neuron* pada suatu lapisan tersembunyi z_j dengan $j = 1, 2, 3, \dots, p$ menjumlahkan sinyal-sinyal *input* berbobot dengan menggunakan persamaan:

$$z_in_j = v_j + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots(10)$$

$$z_in_1 = 0,32 + (0,456*0,27) + (0,500*0,83) + (0,411*0,17) + (0,278*0,38) = 1,4074$$

$$z_in_2 = 0,69 + (0,322*0,45) + (0,5*0,51) + (0,411*0,44) + (0,278*0,47) = 1.4014$$

$$z_in_3 = 0,42 + (0,322*0,16) + (0,5*0,82) + (0,411*0,75) + (0,278*0,13) = 1,2260$$

$$z_in_4 = 0,93 + (0,322*0,35) + (0,5*0,98) + (0,411*0,37) + (0,278*0,36) = 1,7571$$

Kemudian hitung sinyal *output* pada lapisan tersembunyi ketika menerapkan fungsi aktivasi *sigmoid biner* dengan menggunakan persamaan:

$$z_j = \frac{1}{1+e^{-z_in_j}} \dots\dots\dots (11)$$

$$z_1 = \frac{1}{1+e^{-(1,4074)}} = 0,8034$$

$$z_2 = \frac{1}{1+e^{-(1,4014)}} = 0,8024$$

$$z_3 = \frac{1}{1+e^{-(1,2260)}} = 0,7731$$

$$z_4 = \frac{1}{1+e^{-(1,7571)}} = 0,8528$$

2. Setiap *neuron output* y_k dengan $k = 1, 2, 3, \dots, m$ menjumlahkan sinyal-sinyal *input* berbobot dengan menggunakan persamaan:

$$y_in_k = w_k + \sum_{j=1}^p z_j w_{jk} \dots\dots\dots (12)$$

$$y_in_k = ,68 + (0,8034*0,17) + (0,8024*0,31) + (0,7731*0,74) + (0,8528*0,47) = 2,0383$$

Kemudian hitung sinyal *output* pada lapisan *output* dengan menggunakan persamaan:

$$y_k = \frac{1}{1+e^{-y_in_k}} \dots\dots\dots (13)$$

$$y_k = \frac{1}{1+e^{-(2,0383)}} = 0,8848$$

3. Tahap *Backpropagation*

Tahap *backpropagation* merupakan tahap aliran perambatan mundur dimana setiap *neuron output layer* menerima target pola *output* yang berkaitan dengan pola *input* dengan pelatihan kemudian diteruskan menuju *neuron hidden layer* dan dari *neuron hidden layer* diteruskan kembali menuju *neuron input layer*. Tahap *backpropagation* akan menghitung setiap perubahan-perubahan bobot yang menuju *hidden layer* dan *output layer*. Tahap *backpropagation* akan membandingkan nilai keluaran dari *output layer* dengan target *output*. Ketika nilai target belum terpenuhi maka akan dilakukan penyesuaian terhadap bobot dan *bias*. Adapun langkah-langkah pada tahap *backpropagation* adalah sebagai berikut:

1. Setiap *neuron output* y_k dengan $k = 1, 2, 3, \dots, m$, menerima target pola *output* yang berkaitan dengan pola *input* dengan pelatihan. Hitung *error* pada lapisan *output*:

$$\delta_k = (t_k - y_k) \left(\frac{1}{1+e^{-y_in_k}} \right) \left(1 - \left(\frac{1}{1+e^{-y_in_k}} \right) \right) \dots\dots\dots (14)$$

δ_k adalah target *output* yang diharapkan.

$$\delta_k = (0,233 - 0,8848)(0,8848)(1 - 0,8848) = -0,06642$$

Selanjutnya ketika nilai *error* belum mencapai target, maka dilanjutkan dengan perhitungan perubahan nilai bobot w_{jk} dengan *learning rate* (α) = 0,1 melalui perhitungan sebagai berikut:

$$\Delta w_{jk} = \alpha \delta_k z_j \dots \dots \dots (15)$$

$$\Delta w_{[1,1]} = \alpha * \delta_k * z_1 = 0,1 * -0,0664 * 0,8034 = -0,00534$$

$$\Delta w_{[1,2]} = \alpha * \delta_k * z_2 = 0,1 * -0,0664 * 0,8024 = -0,00533$$

$$\Delta w_{[1,3]} = \alpha * \delta_k * z_3 = 0,1 * -0,0664 * 0,7731 = -0,00514$$

$$\Delta w_{[1,4]} = \alpha * \delta_k * z_4 = 0,1 * -0,0664 * 0,8528 = -0,00566$$

$$\Delta w_{[0]} = \alpha * \delta_k = 0,1 * -0,0664 = -0,00664$$

2. Setiap *neuron* pada lapisan tersembunyi z_j dengan $j = 1,2,3,\dots,p$ menjumlahkan faktor *delta* pada lapisan tersembunyi dengan menggunakan persamaan:

$$\delta_{in_k} = \sum_{k=1}^m \delta_k w_{jk} \dots \dots \dots (16)$$

$$\delta_{in_1} = \delta_k * w_{jk} = -0,0664 * 0,17 = -0,01129$$

$$\delta_{in_2} = \delta_k * w_{jk} = -0,0664 * 0,31 = -0,02059$$

$$\delta_{in_3} = \delta_k * w_{jk} = -0,0664 * 0,74 = -0,04915$$

$$\delta_{in_4} = \delta_k * w_{jk} = -0,0664 * 0,47 = -0,03122$$

Kemudian hitung nilai *error* δ diunit tersembunyi (*hidden layer*) dengan menggunakan persamaan:

$$\delta_k = \delta_{in_k} \left(\frac{1}{1+e^{-z_{in_j}}} \right) \left(1 - \left(\frac{1}{1+e^{-z_{in_j}}} \right) \right) \dots \dots \dots (17)$$

$$\delta_1 = -0,01129 * 0,8034 * (1-0,8034) = -0,00178$$

$$\delta_2 = -0,02059 * 0,8024 * (1-0,8024) = -0,00326$$

$$\delta_3 = -0,04915 * 0,7731 * (1-0,7731) = -0,00862$$

$$\delta_4 = 0,03122 * 0,8528 * (1-0,8528) = -0,00392$$

Selanjutnya menghitung nilai koreksi bobot pada setiap unit keluaran (*output*) dengan menggunakan persamaan:

$$\Delta v_{jk} = \alpha \delta_j x_i \dots \dots \dots (18)$$

$$X1 = 0,456$$

$$\Delta v_{[1,1]} = \alpha * \delta_1 * x_1 = 0,1 * -0,00178 * 0,456 = -0,000811$$

$$\Delta v_{[1,2]} = \alpha * \delta_2 * x_1$$

$$= 0,1 * -0,00326 * 0,456 = -0,000148$$

$$\Delta v_{[1,3]} = \alpha * \delta_3 * x_1 = 0,1 * -0,00862 * 0,456 = -0,000393$$

$$\Delta v_{[1,4]} = \alpha * \delta_4 * x_1 = 0,1 * -0,00392 * 0,456 = -0,000178$$

$$X2 = 0,322$$

$$\Delta v_{[1,1]} = \alpha * \delta_1 * x_1 = 0,1 * -0,00178 * 0,322 = -0,000057$$

$$\Delta v_{[1,2]} = \alpha * \delta_2 * x_1 = 0,1 * -0,00326 * 0,322 = -0,000105$$

$$\Delta v_{[1,3]} = \alpha * \delta_3 * x_1 = 0,1 * -0,00862 * 0,322 = -0,000278$$

$$\Delta v_{[1,4]} = \alpha * \delta_4 * x_1 = 0,1 * -0,00392 * 0,322 = -0,000126$$

$$X3 = 0,500$$

$$\Delta v_{[2,1]} = \alpha * \delta_1 * x_2 = 0,1 * -0,00178 * 0,500 = -0,000892$$

$$\Delta v_{[2,2]} = \alpha * \delta_2 * x_2 = 0,1 * -0,00326 * 0,500 = -0,000163$$

$$\Delta v_{[2,3]} = \alpha * \delta_3 * x_2 = 0,1 * -0,00862 * 0,500 = -0,000431$$

$$\Delta v_{[2,4]} = \alpha * \delta_4 * x_2 = 0,1 * -0,00392 * 0,500 = -0,000161$$

$$X4 = 0,411$$

$$\Delta v_{[3,1]} = \alpha * \delta_1 * x_3 = 0,1 * -0,00178 * 0,411 = -0,000733$$

$$\Delta v_{[3,2]} = \alpha * \delta_2 * x_3 = 0,1 * -0,00326 * 0,411 = -0,000134$$

$$\Delta v_{[3,3]} = \alpha * \delta_3 * x_3 = 0,1 * -0,00862 * 0,411 = -0,000354$$

$$\Delta v_{[3,4]} = \alpha * \delta_4 * x_3 = 0,1 * -0,00392 * 0,411 = -0,000161$$

$$X5 = 0,278$$

$$\Delta v_{[4,1]} = \alpha * \delta_1 * x_4 = 0,1 * -0,00178 * 0,278 = -0,000495$$

$$\Delta v_{[4,2]} = \alpha * \delta_2 * x_4 = 0,1 * -0,00326 * 0,278 = -0,000091$$

$$\Delta v_{[4,3]} = \alpha * \delta_3 * x_4 = 0,1 * -0,00862 * 0,278 = -0,000239$$

$$\Delta v_{[4,4]} = \alpha * \delta_4 * x_4$$

$$= 0,1 * -0,00392 * 0,278 = -0,000109$$

Berikut ini hasil yang diperoleh dari koreksi bobot pada setiap unit keluaran (*output*) dan hasilnya dapat dilihat pada.

Tabel 12. Koreksi Bobot Pada Setiap Unit Keluaran (*Output*)

Bobot	z1	z2	z3	z4
v1	-0,000811	-0,000148	-0,000393	-0,000178
v2	-0,000057	-0,000105	-0,000278	-0,000126
v3	-0,0000892	-0,000163	-0,000431	-0,000196
v4	-0,0000733	-0,000134	-0,000354	-0,000161
v5	-0,0000495	-0,000091	-0,000239	-0,000109

Selanjutnya, menghitung koreksi nilai *bias* dari ke lapisan tersembunyi (*hidden layer*) dengan menggunakan persamaan:

$$\Delta V_{j0} = \alpha * \delta_j \dots \dots \dots (19)$$

$$\Delta V_{j1} = 0,1 * (-0,00178) = -0,000178$$

$$\Delta V_{j2} = 0,1 * (-0,00126) = -0,000126$$

$$\Delta V_{j3} = 0,1 * (-0,00196) = -0,000196$$

$$\Delta V_{j4} = 0,1 * (-0,00161) = -0,000161$$

$$\Delta V_{j5} = 0,1 * (-0,00109) = -0,000109$$

Berikut ini hasil yang diperoleh dari koreksi nilai *bias* yang telah dilakukan dan hasil koreksi nilai *bias* pada *hidden layer* dapat dilihat pada.

Tabel 13 Koreksi Nilai Bias Pada Hidden Layer

Bias	vj
1	-0,000178
2	-0,000126
3	-0,000196
4	-0,000161
5	-0,000109

4. Pembaharuan (*Update*) Bobot dan Bias

Setelah tahap *backpropagation* selesai dilakukan maka selanjutnya adalah menghitung perubahan bobot dan *bias* dari *input layer* menuju *hidden layer* maupun dari *hidden layer* menuju *output layer* dari *input layer* menuju *hidden layer*. Tahapan-tahapan untuk pembaharuan (*update*) nilai bobot dan *bias* pada pelatihan JST adalah sebagai berikut:

1. Hitung perubahan nilai bobot dari *input layer* menuju *hidden layer* dengan menggunakan persamaan:

$$v_{ij}(Baru) = v_{ij}(Lama) + \Delta v_{jk} \dots \dots (20)$$

$$V_{1.1}(baru) = V_{1.1}(lama) + \Delta v_{1.1}$$

$$= 0,32 + (-0,000811) = 0,3208$$

$$V_{1.2}(baru) = V_{1.2}(lama) + \Delta v_{1.1}$$

$$= 0,27 + (-0,000057) = 0,2699$$

$$V_{1.3}(baru) = V_{1.2}(lama) + \Delta v_{1.2}$$

$$= 0,83 + (-0,000892) = 0,8299$$

$$V_{1.4}(baru) = V_{1.3}(lama) + \Delta v_{1.3}$$

$$= 0,17 + (-0,000733) = 0,1699$$

$$V_{1.5}(baru) = V_{1.4}(lama) + \Delta v_{1.4}$$

$$= 0,38 + (-0,000495) = 0,3800$$

$$V_{2.1}(baru) = V_{2.1}(lama) + \Delta v_{2.1}$$

$$= 0,32 + (-0,000148) = 0,3201$$

$$V_{2.2}(baru) = V_{2.1}(lama) + \Delta v_{2.1}$$

$$= 0,45 + (-0,000105) = 0,4499$$

$$V_{2.3}(baru) = V_{2.2}(lama) + \Delta v_{2.2}$$

$$= 0,51 + (-0,000163) = 0,5098$$

$$V_{2.4}(baru) = V_{2.3}(lama) + \Delta v_{2.3}$$

$$= 0,44 + (-0,000134) = 0,4399$$

$$V_{2.5}(baru) = V_{2.4}(lama) + \Delta v_{2.4}$$

$$= 0,47 + (-0,000091) = 0,4699$$

$$V_{3.1}(baru) = V_{3.1}(lama) + \Delta v_{3.1}$$

$$= 0,32 + (-0,000393) = 0,3203$$

$$V_{3.2}(baru) = V_{3.1}(lama) + \Delta v_{3.1}$$

$$= 0,16 + (-0,000278) = 0,1597$$

$$V_{3.3}(baru) = V_{3.2}(lama) + \Delta v_{3.2}$$

$$= 0,82 + (-0,000431) = 0,8196$$

$$V_{3.4}(baru) = V_{3.3}(lama) + \Delta v_{3.3}$$

$$= 0,75 + (-0,000354) = 0,7496$$

$$V_{3.5}(baru) = V_{2.4}(lama) + \Delta v_{3.4}$$

$$= 0,13 + (-0,000239) = 0,1298$$

$$V_{4.1}(baru) = V_{4.1}(lama) + \Delta v_{4.1}$$

$$= 0,32 + (-0,000178) = 0,3201$$

$$V_{4.2(bar)} = V_{4.1(lama)} + \Delta v_{4.1}$$

$$= 0,35 + (-0,000126) = 0,3499$$

$$V_{4.3(bar)} = V_{4.2(lama)} + \Delta v_{4.2}$$

$$= 0,98 + (-0,000196) = 0,9798$$

$$V_{4.4(bar)} = V_{4.3(lama)} + \Delta v_{4.3}$$

$$= 0,37 + (-0,000161) = 0,3698$$

$$V_{4.5(bar)} = V_{4.4(lama)} + \Delta v_{4.4}$$

$$= 0,26 + (-0,000109) = 0,2599$$

Berikut ini hasil perhitungan yang telah dilakukan untuk memperbaharui nilai bobot dari *input layer* menuju *hidden layer* dan hasil perubahannya dapat dilihat pada

Tabel 15. Perubahan Bobot Dari *Input Layer* Menuju *Hidden Layer*

Bobot	z1	z2	z3	Bias	
				vj	vj
v1	0,3208	0,3201	0,3203	0,3201	0,7298
v2	0,2699	0,8299	0,1699	0,3800	0,6897
v3	0,4499	0,5098	0,4399	0,699	0,4191
v4	0,1597	0,8196	0,7496	0,298	0,9296
v5	0,3499	0,9798	0,3698	0,2599	

- Hitung perubahan bobot dari *hidden layer* menuju *output layer* dengan menggunakan persamaan:

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk} \quad (21)$$

$$w_{1.1}(baru) = w_{1}(lama) + \Delta w_{1.1}$$

$$= 0,17 + (-0,00534) = 0,1647$$

$$w_{1.2}(baru) = w_{2}(lama) + \Delta w_{1.2}$$

$$= 0,31 + (-0,00533) = 0,3047$$

$$w_{1.3}(baru) = w_{3}(lama) + \Delta w_{1.3}$$

$$= 0,74 + (-0,00514) = 0,7349$$

$$w_{1.4}(baru) = w_{4}(lama) + \Delta w_{1.4}$$

$$= 0,47 + (-0,00566) = 0,4643$$

Berikut ini hasil perhitungan yang telah dilakukan untuk memperbaharui nilai bobot dari *hidden layer* menuju *output layer* dan hasil perubahannya dapat dilihat pada

Tabel 14 Perubahan Bobot Dari *Hidden Layer* Menuju *Output Layer*

Bobot	Y
w1	0,1647
w2	0,3047
w3	0,7349

w4	0,4643
w5	0,4643

- Selanjutnya lakukan perhitungan untuk menemukan nilai *bias* baru pada lapisan keluaran dengan menggunakan persamaan:

$$V_{j0}(baru) = V_{j0}(lama) + \Delta V_{j0} \dots\dots\dots (22)$$

$$V_{j01}(baru) = 0,73 + (-0,000178) = 0,7298$$

$$V_{j02}(baru) = 0,69 + (-0,000326) = 0,6897$$

$$V_{j03}(baru) = 0,42 + (-0,000862) = 0,4191$$

$$V_{j04}(baru) = 0,93 + (-0,000392) = 0,9296$$

Nilai *bias* baru yang diperoleh dari perhitungan yang telah dilakukan dimuat pada

Tabel 16. Perubahan Nilai *Bias* dari *Input Layer* ke *Hidden Layer*

Bias	vj
vj(baru)	0,7298
vj(baru)	0,6897
vj(baru)	0,4191
vj(baru)	0,9296

- Tahap terakhir pada *update* bobot dan *bias* adalah menghitung perubahan nilai *bias* dari *hidden layer* menuju *output layer* dengan menggunakan persamaan:

$$W_{j0}(bias baru) = w_0(lama) + \Delta w_0 \dots\dots (23)$$

$$W_{j0}(bias baru) = 0,68 + (-0,00664) = 0,6734$$

Berikut ini hasil perhitungan yang telah dilakukan untuk merubah nilai *bias* dari *hidden layer* menuju *output layer* yang dimuat pada

Tabel 16 Perubahan *Bias* Dari *Hidden Layer* Menuju *Output Layer*

Bias	1
wk	0,6734

Proses pembaharuan (*update*) bobot dan *bias* akan terus dilakukan ketika belum memenuhi salah satu faktor penghentian proses pelatihan JST. Ketika belum terpenuhi salah satu faktor tersebut maka proses akan dikembalikan ketahap propagasi maju (*feedforward*) dengan merubah nilai bobot dan bias pada tiap lapisan untuk mempengaruhi nilai keluaran (*output*). Perubahan dilakukan dengan memasukkan nilai-nilai bobot dan *bias* baru sesuai dengan hasil perhitungan pada tahap *backpropagation*. Tahapan ini akan terus dilakukan secara berulang hingga target hasil tercapai.

5 Tes Kondisi Berhenti (Apakah $error \leq min\ error / epoch \leq max\ epoch$?)

Proses pelatihan JST akan berhenti apabila menemui salah satu dari dua kondisi proses pelatihan JST yaitu nilai keluaran (*output*) sudah mencapai target atau *epoch* telah mencapai iterasi maksimal (*maximum epoch*). Ketika sudah terpenuhi maka jaringan akan menyimpan nilai bobot dan *bias* terbaru yang mempengaruhi proses pelatihan jaringan untuk mencapai target yang diinginkan. Penyimpanan nilai bobot dan *bias* bertujuan untuk digunakan pada proses pengujian jaringan dengan pengetahuan bahwa nilai tersebut merupakan nilai terbaik untuk mencapai target pengujian.

Analisa perancangan sistem secara manual dengan menggunakan metode *backpropagation momentum* telah dilakukan dengan mengambil sampel pada data jenis dan jumlah kejahatan perjudian sebagai data pelatihan *epoch* pertama. *Epoch* pertama pada proses metode *backpropagation momentum* belum mencapai *error* yang diinginkan, oleh sebab itu pelatihan jaringan harus dilakukan perulangan dimulai dari tahap *feedforward* sampai *error* yang dihasilkan sesuai dengan yang diharapkan. Namun sebagai gambaran proses perhitungan manual

metode *backpropagation momentum* perubahan nilai bobot dan *bias* pada *epoch* pertama akan digunakan untuk proses pengujian data.

5. Pengujian Jaringan

Proses pelatihan jaringan dengan metode *backpropagation* telah dilakukan dengan sampel data pada bulan januari dan telah didapatkan perubahan nilai bobot dan *bias* pada *epoch* pertama, maka selanjutnya melakukan proses pengujian pada jaringan sebagai gambaran proses metode *backpropagation* secara manual. Proses pengujian jaringan menggunakan data uji pada bulan agustus sampai dengan bulan desember seperti yang tunjukkan pada Tabel 4.4. Adapun proses pengujian jaringan yang dihasilkan pada *epoch* pertama adalah sebagai berikut:

1. Gunakan perubahan nilai bobot dan *bias* dari *hidden layer* menuju *output layer*, nilai bobot dan nilai *bias* dari *hidden layer* menuju *output layer* pada proses pelatihan terbaik, namun sebagai gambaran proses pengujian jaringan secara manual menggunakan nilai bobot dan *bias* pada *epoch* pertama yang dimuat dalam

Tabel 17 Nilai Bobot Dan *Bias Epoch* Pertama

Bobot Dari Input Layer ke Hidden Layer				
Bobot	$z1$	$z2$	$z3$	$z4$
V	0,2699	0,8299	0,1699	0,3800
v	0,4499	0,5098	0,4399	0,4699
v	0,1597	0,8196	0,7496	0,1298
v	0,3499	0,9798	0,3698	0,2599
Bobot Dari Hidden Layer ke Output Layer				
Bobot	$w1$	$w2$	$w3$	$w4$
Y	0,1647	0,3047	0,7349	0,4643
Bias Dari Input Layer ke Hidden Layer				
<i>Bias</i>	1	2	3	4
v_j	0,7298	0,6897	0,4191	0,9296
Bias Dari Hidden Layer ke Output Layer				
<i>Bias</i>	1			
w_k	0,6734			

2. Melakukan tahap *feedforward*
 - a. Perkalian dan Penjumlahan bobot dari *input layer* menuju *hidden layer*

$$z_{in1} = 0,7298 + (0,144*0,2699) + (0,189*0,8299) + (0,278*0,1699) + (0,189*0,3800) = 1,04454$$

$$z_{in2} = 0,6897 + (0,144*0,4499) + (0,189*0,5098) + (0,278*0,4399) + (0,189*0,4699) = 1,06191$$

$$z_{in3} = 0,4191 + (0,144*0,1597) + (0,189*0,8196) + (0,278*0,7496) + (0,189*0,1298) = 0,82976$$

$$z_{in4} = 0,9296 + (0,144*0,3499) + (0,189*0,9798) + (0,278*0,3698) + (0,189*0,2599) = 1,31704$$

- b. Pengaktifan fungsi aktivasi bobot dari *input layer* menuju *hidden layer*

$$z_1 = \frac{1}{1 + e^{-(1,04454)}} = 0,73973$$

$$z_2 = \frac{1}{1 + e^{-(1,06191)}} = 0,74305$$

$$z_3 = \frac{1}{1 + e^{-(0,82976)}} = 0,69630$$

$$z_4 = \frac{1}{1 + e^{-(1,31704)}} = 0,78869$$

- c. Perkalian dan Penjumlahan bobot

dari *hidden layer* menuju *output layer*

$$y_{ink} = 0,6734 + (0,73973 * 0,1647) + (0,74305 * 0,3047) + (0,69630 * 0,7349) + (0,78869 * 0,4643) = 1,89946$$

- d. Pengaktifan fungsi aktivasi bobot dari *hidden layer* menuju *output layer*

$$y_k = \frac{1}{1+e^{-(1,89946)}} = 0,86983$$

Data keluaran (*output*) yang sudah didapatkan dari pengujian jaringan, maka dilakukan proses denormalisasi dengan tujuan mengembalikan nilai hasil keluaran kedalam bentuk nilai yang sebenarnya dengan menggunakan persamaan:

$$x_i = \frac{(x_p - 0,1)(max - min)}{0,8} + min \dots \dots \dots (25)$$

$$x_i = \frac{(0,86983 - 0,1)(18 - 0)}{0,8} + 0 = 17,321$$

Lakukan tahap pengujian jaringan secara berulang hingga seluruh data jenis dan jumlah kejahatan dilakukan pengujian jaringan.

Data akurasi (%) yang sudah dapat di hasilkan dengan membulatkan bilangan yang

dijumlahkan dari data aktual ditambahkan dengan data hasil prediksi.[11]

$$Akurasi (\%) = \frac{Hasil Prediksi}{Data Aktual} \times 100\%$$

Dan begitu juga dengan data kejahatan yang lainnya bisa di lihat penjumlahannya di data pelatihan, data pengujian dan data hasil prediksi epoch 1 microsoft excel yang telah di sediakan.

Tahapan-tahapan yang dilakukan menggunakan metode *backpropagation momentum* telah diperoleh nilai keluaran (*output*) berupa hasil prediksi. Hasil prediksi dari contoh perhitungan manual menggunakan metode *backpropagation momentum* berupa nilai *output* yang dicapai setelah pelatihan yang kemudian dibandingkan dengan target yang akan dicapai. Namun dalam proses pelatihan hanya sampai pada *epoch* pertama sebab digunakan sebagai gambaran analisa sistem dengan metode *backpropagation momentum*. Perbandingan nilai *output* dan target akan didapat nilai *error* yang terjadi selama pelatihan JST seperti dimuat pada

Tabel 18 Hasil Prediksi Jaringan Saraf Tiruan *Epoch* Pertama Tahun 2021

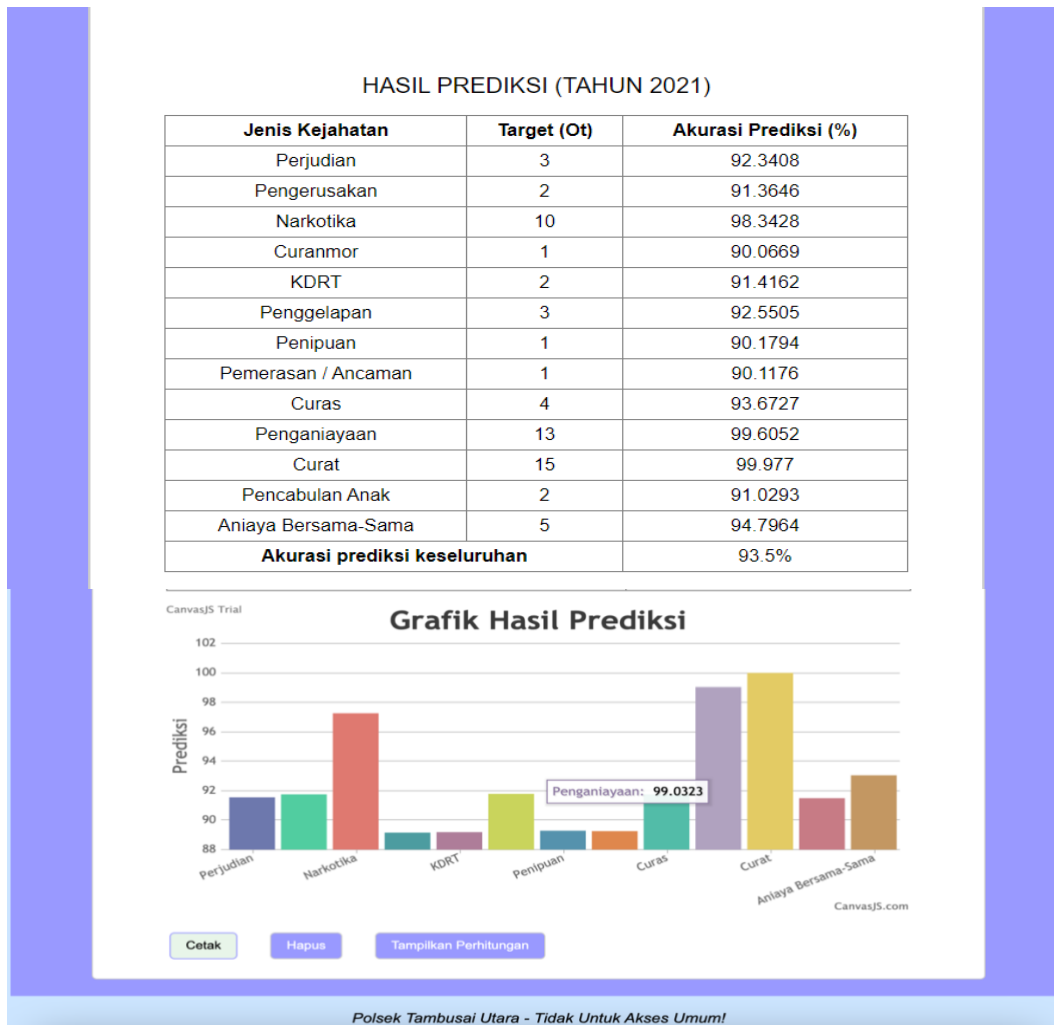
JENIS KEJAHATAN	Data Target 2020	Output	Error	Hasil Prediksi 2021	Data Aktual prediksi	Selisih	Akurasi Prediksi 2021
Perjudian	0,233	0.99486983	-0,786	17,321	3	14.321	17%
CURAS	0.233	0.99279026	-0.759	20.088	3	17.088	15%
PENGANIAYAAN	0.633	0.99919254	-0.366	20.232	12	8.232	59%
CURAT	0.811	0.99518964	-0.184	20.142	16	4.142	79%
PENCABULAN ANAK	0.233	0.99450125	-0.761	20.126	3	17.126	15%
PENGERUSAKAN	0.144	0.98996521	-0.846	20.024	1	19.024	5%
NARKOTIKA	0.500	0.99916786	-0.499	20.231	9	11.231	44%
CURANMOR	0.144	0.99907317	-0.855	20.229	1	19.229	5%
KDRT	0.144	0.99908275	-0.855	20.229	1	19.229	5%
PENGGELAPAN	0.233	0.9990752	-0.766	20.229	3	17.229	15%
PENIPUAN	0.144	0.99905921	-0.855	20.229	1	19.229	5%
PEMERASAN / ANCAMAN	0.144	0.99907349	-0.855	20.229	1	19.229	5%
			0.52690				
$MSE = \frac{\sum_{i=1}^n (Xt - Ft)^2}{n}$							

Berdasarkan hasil pengujian yang dilakukan secara manual pada data jenis dan jumlah kejahatan seperti yang dimuat pada Tabel 4.18 menghasilkan prediksi jenis dan jumlah kejahatan relatif cukup baik pada pelatihan *epoch* pertama sebab nilai keluaran atau *output* menghasilkan prediksi mencapai 79% walau pun

masih ada beberapa hasil prediksi dengan akurasi cukup kecil sebab masih dalam iterasi (*epoch*) pertama. Hal tersebut memberikan indikasi bahwa metode *backpropagation momentum* mampu melakukan prediksi jenis dan jumlah kejahatan dengan baik pada proses pelatihan jaringan selanjutnya.

Hasil perhitungan pengujian jaringan pada *epoch* pertama secara manual menunjukkan akurasi tertinggi sebesar 79% sedangkan akurasi terendah sebesar 5%. Hasil sudah menunjukkan tingkat prediksi yang cukup baik namun belum memenuhi target yang diinginkan dengan MSE atau target *goal* sebesar 0,001, sehingga untuk memperoleh hasil yang terbaik pada prediksi jenis dan jumlah

kejahatan perlu membangun sistem aplikasi untuk melakukan proses pelatihan dan pengujian JST dengan metode *backpropagation momentum*. Proses pelatihan dan pengujian JST pada sistem aplikasi memerlukan beberapa parameter-parameter pelatihan jaringan yang mempengaruhi hasil prediksi.



Gambar 2. Hasil Prediksi

V. KESIMPULAN

Prediksi jenis dan jumlah kejahatan dengan metode *backpropagation momentum* di Kecamatan Tambusai Utara, maka dapat diambil beberapa kesimpulan sebagai berikut:

Sistem aplikasi dapat diterima sesuai yang diharapkan dan tidak ada instruksi *error* dan layak digunakan untuk memprediksi jenis dan jumlah kejahatan dimasa mendatang di Kecamatan Tambusai Utara.

Data yang dilakukan pelatihan dan pengujian jaringan saraf tiruan metode *backpropagation*

momentum adalah data jenis dan jumlah kejahatan yang terjadi selama tahun 2015 sampai dengan tahun 2020 di Kecamatan Tambusai Utara. Hasil yang diperoleh dari pelatihan dan pengujian jaringan saraf tiruan metode *backpropagation momentum* yaitu menghasilkan tingkat prediksi jenis dan jumlah kejahatan dengan tingkat akurasi baik dengan tingkat akurasi tertinggi sebesar 99,9% dan tingkat akurasi terendah sebesar 90,43% dengan rata-rata akurasi tingkat prediksi sebesar 94,88%. Diprediksi jenis kejahatan yang tinggi pada tahun selanjutnya adalah tindak kejahatan curat (pencurian pemberatan) yaitu sebesar 15 kasus sedangkan jenis kejahatan terendah yaitu tindak kejahatan penipuan dan pemerasan / ancaman masing-masing 1 kasus.

DAFTAR PUSTAKA

- [1] R. S. Putra, "Kriminalitas di Kalangan Remaja (Studi Terhadap Remaja Pelaku Pencabulan di Lembaga Pemasarakatan Anak Kelas II B Pekanbaru)," *Jurnal Ilmu Komunikasi*, vol. 3, no. 2, pp. 1–15, 2016.
- [2] Polri, "Peraturan Kapolri Nomor 6 Tahun 2017," *Sturtur Organisasi dan Tata Kerja Satuan Organisasi Tingkat Markas Besar Kepolisian Negara Republik Indonesia*, no. 9, pp. 1689–1699, 2017.
- [3] S. N. Dewi, I. Cholissodin, and E. Santoso, "Prediksi Jumlah Kriminalitas Menggunakan Metode Extreme Learning Machine (Studi Kasus Di Kabupaten Probolinggo)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya*, vol. 2, no. 11, pp. 4687–4693, 2018.
- [4] K. Kalaiselvi, K. Velusamy, and C. Gomathi, "Financial prediction using back propagation neural networks with opposition based learning," in *Journal of Physics: Conference Series*, 2018, vol. 1142, no. 1. doi: 10.1088/1742-6596/1142/1/012008.
- [5] S. Setti and A. Wanto, "Analysis of Backpropagation Algorithm in Predicting the Most Number of Internet Users in the World," *Jurnal Online Informatika*, vol. 3, no. 2, p. 110, 2019, doi: 10.15575/join.v3i2.205.
- [6] E. Irawan, M. Zarlis, and E. B. Nababan, "Analisis Penambahan Nilai Momentum Pada Prediksi Produktivitas Kelapa Sawit Menggunakan Backpropagation," *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, vol. 1, no. 2, pp. 84–89, 2017, doi: 10.30743/infotekjar.v1i2.67.
- [7] P. Indrayati Sijabat, Y. Yuhandri, G. Widi Nurcahyo, and A. Sindar, "Algoritma Backpropagation Prediksi Harga Komoditi terhadap Karakteristik Konsumen Produk Kopi Lokal Nasional," *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, vol. 11, no. 1, pp. 96–107, 2020, doi: 10.31849/digitalzone.v11i1.3880.
- [8] A. Santoso and S. Hansun, "Prediksi IHSG dengan Backpropagation Neural Network," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 3, no. 2, pp. 313–318, 2019, doi: 10.29207/resti.v3i2.887.
- [9] B. Yanto, B. -, J. -, and B. H. Hayadi, "INDENTIFIKASI POLA AKSARA ARAB MELAYU DENGAN JARINGAN SYARAF TIRUAN CONVOLUTIONAL NEURAL NETWORK (CNN)," *JSAI (Journal Scientific and Applied Informatics)*, vol. 3, no. 3, 2020, doi: 10.36085/jsai.v3i3.1151.
- [10] M. I. Suja'I, K. Rukun, Ridwan, B. H. Hayadi, B. Yanto, and R. D. P. Permatasari, "The Effectiveness of Learning Media Developed with the Kahoot Application on the Subject of Management Information System," in *Journal of Physics: Conference Series*, 2019, vol. 1363, no. 1. doi: 10.1088/1742-6596/1363/1/012065.
- [11] B. Yanto, erni Rouza, and edi saputra, "Penerapan Metode Inferensi Fuzzy Takagi Sugeno-Kang Untuk Prediksi Hasil Panen Kelapa Sawit," *JISA(Jurnal Informatika dan Sains)*, vol. 2, no. 2, 2019, doi: 10.31326/jisa.v2i2.496.