

Implementasi Jaringan Saraf Tiruan Dalam Prediksi Penjualan Kue pada UD. Mak Kembar Pematang Siantar Dengan Backpropagation

Hendro Sanjaya^{1*}, Harly Okprana¹, Bahrudi Efendi Damanik²

¹Sistem Informasi, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia
Jalan Jendral Sudirman Blok A, No.1,2 & 3, Pematangsiantar, Sumatera Utara, Indonesia

²Komputerisasi Akuntansi, AMIK Tunas Bangsa, Pematangsiantar, Indonesia
Jalan Jendral Sudirman Blok A, No.1,2 & 3, Pematangsiantar, Sumatera Utara, Indonesia
Email: 'hendrosanjaya@gmail.com,

Abstrak—UD.Mak Kembar merupakan toko yang bergerak dalam bidang penjualan Kue dan Roti yang berlokasi di Rambung Merah. Sampai saat ini UD.Mak Kembar hanya menggunakan sistem sederhana dalam pemasaran informasi dan pencatatan, sehingga menimbulkan kelambatan dalam pemasaran informasi dan dalam pencatatan perhitungan penjualan. Berdasarkan permasalahan di atas penulis merancang dan membuat sistem pada UD.Mak Kembar. Sistem ini menerapkan prediksi dalam penjualannya, dan dibangun dengan menggunakan jaringan syaraf tiruan algoritma Backpropagation. Didapatkan hasil dengan arsitektur terbaik 4-5-1 dengan proses perulangan (*epoch*) pada saat pelatihan dengan nilai *epoch* = 75% dan pencapaian SSE pada saat pengujian dengan SSE = 0,23950.

Kata Kunci: *Backpropagation, Penjualan Kue, UD. Mak Kembar, Prediksi*

Abstract—UD.Mak Kembar is a shop engaged in the sale of cakes and bread, located in Rambung Merah. Until now, UD. Mak Kembar only uses a simple system in marketing information and recording, causing delays in marketing information and in recording sales calculations. Based on the problems above, the authors design and create a system at UD.Mak Kembar. This system applies predictions in its sales, and is built using an artificial neural network with the Backpropagation algorithm. The results obtained with the best architecture 4-5-1 with an iterative process (*epoch*) during training with an *epoch* value = 75% and SSE achievement at the time of testing with SSE = 0.23950.

Keywords: *Back-propagation, training function, Neural Network, Optimization, crime victim*

1. PENDAHULUAN

Jaringan Saraf Tiruan yang berupa susunan sel-sel saraf tiruan (neuron) dibangun berdasarkan prinsip-prinsip organisasi manusia, selain itu jaringan saraf tiruan dapat memprediksi kelayakan calon peneriman bantuan renovasi rumah yang dilakukan proses pelatihan menggunakan jaringan saraf tiruan dengan metode Backpropagation. Rumah merupakan salah satu kebutuhan pokok bagi manusia, dimana rumah berperan sebagai wadah untuk bernaung dan tempat berlindung. Selain memiliki fungsi sebagai tempat tinggal, rumah juga berfungsi untuk tempat pembinaan dan bercengkrama suatu keluarga. Setiap manusia pasti berkeinginan memiliki rumah yang layak untuk ditinggali, agar penghuni rumah merasa aman dan nyaman saat menempati rumahnya. Namun keinginan memiliki rumah yang layak bagi beberapa orang harus terpendam disebabkan oleh kemiskinan. Berbagai upaya dilakukan melalui pengembangan inovasi yang dilakukan oleh pengusaha kuliner untuk meningkatkan penjualannya. Banyak faktor yang mempengaruhi tinggi rendahnya tingkat penjualan diantaranya produk itu sendiri, harga, distribusi, promosi dan layanan purna jual. Prakiraan tingkat penjualan yang tepat dapat dijadikan rujukan guna menentukan keberlangsungan usaha dan tingkat keuntungan yang ingin dicapai. Jumlah penjualan dan jumlah transaksi yang dilakukan setiap harinya pasti berbeda-beda. Tetapi, jika jumlah penjualan banyak belum menentukan jumlah transaksi semakin Prediksi sendiri sebenarnya memperkirakan data yang akan dihasilkan pada masa yang akan datang dengan menggunakan data-data yang terjadi sebelumnya. Oleh karena alasan tersebut, maka penulis akan melakukan riset untuk memprediksi penjualan kue di UD. Mak Kembar, di mana jumlah penjualan tersebut sangat mempengaruhi jumlah transaksi penjualan yang akan terjadi. Sehingga UD. Mak Kembar mencari solusi untuk meningkatkan pendapatan penjualan, mulai dari menjual kue serta meningkatkan kualitas bahan pokok pembuatan kue. Akibat dari pola transaksi yang terjadi setiap harinya berbeda-beda, maka diperlukan sistem yang dapat memprediksi kemungkinan transaksi penjualan yang terjadi dimasa yang akan datang, untuk menyelesaikan permasalahan baik dari segi penjualan maupun dari segi mempromosikan sehingga dapat menambah nilai penjualan yang di harapkan dapat berjalan dengan semestinya dan dapat meningkatkan kualitas dari UD.Mak Kembar.

Oleh karena itu perlu adanya sistem yang dapat memantau penjualan kue pada rentang waktu tertentu di UD.Mak Kembar, yaitu dimana sistem yang dapat memperkirakan penjualan kue yang akan terjual perbulannya. Maka penulis mengambil inisiatif untuk melakukan penelitian di UD.Mak Kembar guna untuk membantu kinerjanya agar bisa memantau penjualan kue perbulan agar dapat memaksimalkan angka penjualan kue di setiap bulannya. Ada beberapa hal yang sangat mempengaruhi penjualan kue di UD.Mak Kembar diantaranya adalah jumlah pelanggan, jumlah distribusi kue yang diproduksi serta jumlah kue setiap bulannya. Oleh karena itu, UD.Mak Kembar memerlukan sebuah sistem yang dapat melakukan simulasi untuk memprediksi jumlah penjualan kue perbulan agar UD.Mak Kembar dapat mengambil tindakan untuk dapat mencapai target penjualan kue kesemua pelanggan UD.Mak Kembar perbulannya. Dalam penelitian ini, metode yang akan digunakan untuk melakukan prediksi adalah metode Backpropagation, karena Backpropagation merupakan algoritma pembelajaran yang terawasi dan digunakan dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyi nya sehingga

membuat hasil dari prediksi menjadi lebih baik karena meminimalkan total error dari keluaran yang dihitung oleh jaringan.

2. METODOLOGI PENELITIAN

2.1. Teknik Pengumpulan Data

Teknik pengumpulan data ini dilakukan dengan mengandalkan dokumen sebagai salah satu sumber data yang digunakan untuk melengkapi penelitian. Adapun dokumen yang di *copy* merupakan dokumen yang berisi data penjualan kue dari bulan Januari-Desember 2020.

2.2. Analisis Data

Dalam melakukan sebuah penelitian, komponen yang paling penting adalah diperlukannya data penelitian dimana data tersebut akan diolah sehingga menghasilkan suatu tujuan yang bermanfaat. Berikut ini data yang diperoleh dari UD.Mak Kembar pada tabel 1. berikut :

Tabel 1. Data Penjualan

| No. | Nama Kue | Bulan | | | |
|-----------------|--------------|-----------|---------|----------|----------|
| | | September | Oktober | November | Desember |
| 1. | Kue caramel | 67 | 72 | 61 | 59 |
| 2. | Kue pandan | 75 | 88 | 70 | 83 |
| 3. | Kue brownies | 68 | 70 | 56 | 50 |
| 4. | Nastar | 47 | 58 | 62 | 65 |
| 5. | Sempit | 40 | 36 | 39 | 45 |
| 6. | Kue Mangkok | 38 | 39 | 40 | 55 |
| 7. | Kue moca | 35 | 55 | 49 | 54 |
| 8. | Copy Oil | 60 | 57 | 54 | 66 |
| Total Penjualan | | 430 | 475 | 431 | 477 |

2.3. Backpropagation

Backpropagation merupakan sebuah metode sistematis untuk pelatihan multiplayer jaringan saraf tiruan. Metode ini memiliki dasar matematis yang kuat, obyektif dan algoritma ini mendapatkan bentuk persamaan dan nilai koefisien dalam formula dengan meminimalkan jumlah kuadrat galat error melalui model yang dikembangkan [4]. Jaringan saraf lapis jamak (*multilayer network*) dengan pelatihan terbimbing (*supervised*) antara lain adalah jaringan perambatan-balik (*backpropagation*). Metode pelatihan *backpropagation* melibatkan feedforward dari pola pelatihan input, perhitungan dan *backpropagation* dari kesalahan, dan penyesuaian bobot pada sinapsis [2]. Ciri khas *backpropagation* melibatkan tiga lapisan : lapisan *input*, dimana data diperkenalkan ke jaringan; *hidden layer*, dimana data diproses; dan lapisan *output*, di mana hasil dari masukan yang diberikan oleh lapisan input [5]. Secara rinci algoritma pelatihan jaringan *Backpropagation* dalam [6] dapat diuraikan sebagai berikut :

Langkah 1 : Inisialisasikan bobot (set ke nilai kecil secara acak)

Langkah 2 : Selama kondisi berhenti bernilai salah, kerjakan:

1) Untuk masing-masing pasangan pelatihan, lakukan *Feedforward*, yaitu :

a) Masing-masing unit input (X_i , $i=1, \dots, n$) menerima sinyal *input* X_i dan menyebarkannya ke semua unit lapisan atas (unit tersembunyi).

b) Masing-masing unit tersembunyi (Z_j , $j=1, \dots, p$) menjumlahkan bobot sinyal *input* ,

$$Z_{inj} = V_0j + \sum_{i=1}^n X_i V_{ij} \quad (1)$$

dan mengaplikasikan fungsi aktivasi untuk menghitung sinyal *output*

$$z_j = f(z_{inj}) \quad (2)$$

dan mengirimkan sinyal ke semua unit pada lapisan atas (unit *output*)

c) Masing-masing unit *output* (Y_k , $k=1, \dots, m$) menjumlahkan jumlah sinyal *input*,

$$y_{ink} = W_{ok} + \sum_j z_j w_{jk} \quad (3)$$

dan mengaplikasikan fungsi aktivasi untuk menghitung sinyal *output*

$$y_k = f(y_{ink}) \quad (4)$$

2) Untuk masing-masing pasangan pelatihan, lakukan *Backpropagation*, yaitu:

a) Masing-masing unit *output* (Y_k , $k=1, \dots, m$) menerima sebuah pola target yang bersesuaian dengan pola *input* pelatihan, menghitung informasi kesalahan

$$\delta_k = (t_k - y_k) f'(y_{ink}) \quad (5)$$

kemudian menghitung koreksi bobot (digunakan memperbaiki w_{jk}),

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

(digunakan untuk memperbaiki w_{ok})

$$\Delta w_{ok} = \alpha \delta_k \quad (7)$$

setelah itu mengirimkan δ_k ke unit dalam lapisan paling atas.

- b) Masing-masing unit yang tersembunyi ($Z_j, j=1, \dots, p$) menjumlahkan *input* delta (dari unit lapisan atas)

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk} \tag{8}$$
 Kalikan nilai ini dengan turunan fungsi aktivasi untuk menghitung informasi kesalahan.

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \tag{9}$$
 kemudian hitung koreksi bobot (digunakan untuk memperbaiki v_{ij}),

$$\Delta V_{ij} = \alpha \delta_j x_i \tag{10}$$
 setelah itu hitung koreksi bias (digunakan untuk memperbaiki V_{oj})

$$\Delta V_{oj} = \alpha \delta_j \tag{11}$$
- 3) Perbaiki bobot dan bias, yaitu :
 - a) Masing-masing unit *output* ($Y_k, k=1, \dots, m$) memperbaiki bobot dan bias ($Z_j, j=1, \dots, p$)

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta v_{ij} \tag{12}$$
 - b) Masing-masing unit tersembunyi ($Z_j, j=1, \dots, p$) memperbaiki bobot dari bias ($X_i, i=1, \dots, n$)

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \tag{13}$$
- 4. Tes kondisi; berhenti.

3. HASIL DAN PEMBAHASAN

Tahap Normalisasi dilakukan Setelah data *Input* dan real dimasukkan dalam *tools* oleh user ke *Ms.Excel*, maka *tools* akan melakukan proses normalisasi sesuai perintah user, tampilan hasil normalisasi sebagai berikut :

Tabel 2. Tabel Normalisasi Data Training

| No | X1 | X2 | X3 | X4 | Target |
|----|--------|--------|--------|--------|--------|
| 1 | 0,5830 | 0,6585 | 0,4925 | 0,4623 | 0,4170 |
| 2 | 0,7038 | 0,9000 | 0,6283 | 0,8245 | 0,7868 |
| 3 | 0,5981 | 0,6283 | 0,4170 | 0,3264 | 0,2358 |
| 4 | 0,2811 | 0,4472 | 0,5075 | 0,5528 | 0,6660 |
| 5 | 0,1755 | 0,1151 | 0,1604 | 0,2509 | 0,2434 |
| 6 | 0,1453 | 0,1604 | 0,1755 | 0,4019 | 0,4170 |
| 7 | 0,1000 | 0,4019 | 0,3113 | 0,3868 | 0,4925 |
| 8 | 0,4774 | 0,4321 | 0,3868 | 0,5679 | 0,5226 |

Tabel 3. Tabel Normalisasi Data Testing

| No | X1 | X2 | X3 | X4 | Target |
|----|--------|--------|--------|--------|--------|
| 1 | 0,6630 | 0,5000 | 0,4704 | 0,4259 | 0,3296 |
| 2 | 0,9000 | 0,6333 | 0,8259 | 0,7889 | 0,7519 |
| 3 | 0,6333 | 0,4259 | 0,3370 | 0,2481 | 0,1000 |
| 4 | 0,4556 | 0,5148 | 0,5593 | 0,6704 | 0,7222 |
| 5 | 0,1296 | 0,1741 | 0,2630 | 0,2556 | 0,3222 |
| 6 | 0,1741 | 0,1889 | 0,4111 | 0,4259 | 0,5444 |
| 7 | 0,4111 | 0,3222 | 0,3963 | 0,5000 | 0,4926 |
| 8 | 0,4407 | 0,3963 | 0,5741 | 0,5296 | 0,5963 |

Hasil percobaan menunjukkan bahwa Jaringan saraf tiruan dengan Algoritma *Backpropagation* yang sudah dilatih dan diuji dengan baik akan memberikan keluaran yang masuk akal jika diberi masukan yang serupa dengan pola yang dipakai untuk pelatihan dan pengujian. Sifat generalisasi ini membuat pelatihan dan pengujian lebih efisien karena tidak perlu dilakukan pada semua data. Jaringan saraf tiruan dengan Algoritma *Backpropagation* dibentuk dengan membuat generalisasi aturan pelatihan dan pengujian dalam model *Windrow-Hooff* dengan cara menambahkan lapisan tersembunyi (*hidden layer*). Standar Algoritma *Backpropagation* menggunakan algoritma penurunan gradien (*gradien descent*). variasi terhadap model standar dilakukan dengan mengganti algoritmanya dengan algoritma lain.

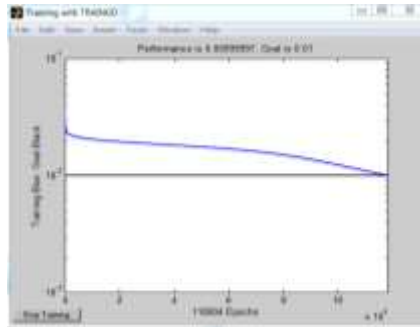
3.1. Pelatihan dan Pengujian Arsitektur 4-2-1

Berikut hasil pelatihan dan pengujian dengan 8 data untuk arsitektur 4-2-1. Adapun parameter yang digunakan adalah:

| Koding Pelatihan | Koding Pengujian |
|---|---|
| <pre>>> net=newff(minmax(P),[2,1],{'logsig','purelin'},'traingd'); >> net.IW{1,1}; >> net.b{1}; >> net.LW{2,1}; >> net.b{2}; >> net.trainParam.epochs= 200000; >> net.trainParam.goal = 0.01;</pre> | <pre>>> PP= >> TT= >> [a,Pf,Af,e,Perf]=sim(net,PP,[],[],TT)</pre> |

| Koding Pelatihan | Koding Pengujian |
|--|------------------|
| <pre>>> net.trainParam.Lr = 0.01; >> net.trainParam.show = 1000; >> net=train(net,P,T) [a,Pf,Af,e,Perf]=sim(net,P,[],[],T)</pre> | |

Setelah dilakukan perulangan maka ditemukan *error minimum* pada *epoch* 118004 seperti pada Gambar 1.



Gambar 1. Pelatihan Arsitektur 4-2-1 Mencapai Goal

Untuk hasil yang lebih rinci dan mengetahui *output* dan *error* dapat dilihat pada tabel berikut.

Tabel 4. Hasil Pelatihan dengan Model 4-2-1

| No | Target | Output JST | Error | SSE | Hasil |
|----|--------|------------|----------|------------|-------|
| 1 | 0,4170 | 0,4897 | -0,07272 | 0,00529 | 1 |
| 2 | 0,7868 | 0,7667 | 0,02009 | 0,00040 | 0 |
| 3 | 0,2358 | 0,3310 | -0,09515 | 0,00905 | 1 |
| 4 | 0,6660 | 0,4893 | 0,17674 | 0,03124 | 0 |
| 5 | 0,2434 | 0,2844 | -0,04100 | 0,00168 | 1 |
| 6 | 0,4170 | 0,4977 | -0,08072 | 0,00652 | 1 |
| 7 | 0,4925 | 0,3423 | 0,15015 | 0,02255 | 0 |
| 8 | 0,5226 | 0,5797 | -0,05706 | 0,00326 | 1 |
| | | | | 0,07998 | 63 |
| | | | MSE | 0,00666500 | |

Tabel 5. Hasil Pengujian dengan Model 4-2-1

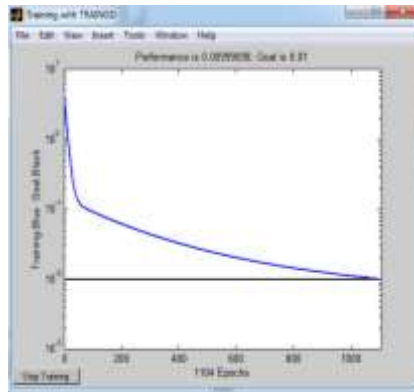
| No | Target | Output JST | Error | SSE | Hasil |
|----|--------|------------|----------|------------|-------|
| 1 | 0,3296 | 0,4935 | -0,16387 | 0,02685 | 1 |
| 2 | 0,7519 | 1,0261 | -0,27425 | 0,07521 | 1 |
| 3 | 0,1000 | 0,2793 | -0,17930 | 0,03215 | 1 |
| 4 | 0,7222 | 0,5908 | 0,13142 | 0,01727 | 0 |
| 5 | 0,3222 | 0,1488 | 0,17342 | 0,03008 | 0 |
| 6 | 0,5444 | 0,3505 | 0,19394 | 0,03761 | 0 |
| 7 | 0,4926 | 0,5281 | -0,03551 | 0,00126 | 1 |
| 8 | 0,5963 | 0,4721 | 0,12420 | 0,01542 | 0 |
| | | | | 0,23586 | 50 |
| | | | MSE | 0,01965509 | |

3.2. Pelatihan dan Pengujian Arsitektur 4-5-1

Berikut hasil pelatihan dan pengujian dengan 8 data untuk arsitektur 4-5-1. Adapun parameter yang digunakan adalah:

| Koding Pelatihan | Koding Pengujian |
|--|---|
| <pre>>> net=newff(minmax(P),[2,1],{'logsig','purelin'},'traingd'); >> net.IW{1,1}; >> net.b{1}; >> net.LW{2,1}; >> net.b{2}; >> net.trainParam.epochs = 200000; >> net.trainParam.goal = 0.01; >> net.trainParam.Lr = 0.01; >> net.trainParam.show = 1000; >> net=train(net,P,T) [a,Pf,Af,e,Perf]=sim(net,P,[],[],T)</pre> | <pre>>> PP= >> TT= >> [a,Pf,Af,e,Perf]=sim(net,PP,[],[],TT)</pre> |

Setelah dilakukan perulangan maka ditemukan *error minimum* pada *epoch* 1104 seperti pada Gambar 2.



Gambar 2. Pelatihan Arsitektur 4-5-1 Mencapai Goal

Untuk hasil yang lebih rinci dan mengetahui *output* dan *error* dapat dilihat pada tabel berikut.

Tabel 6. Hasil Pelatihan dengan Model 4-5-1

| No | Target | Output JST | Error | SSE | Hasil |
|-----|--------|------------|----------|------------|-------|
| 1 | 0,4170 | 0,4407 | -0,02372 | 0,00056 | 1 |
| 2 | 0,7868 | 0,6979 | 0,08889 | 0,00790 | 0 |
| 3 | 0,2358 | 0,2268 | 0,00905 | 0,00008 | 1 |
| 4 | 0,6660 | 0,7634 | -0,09736 | 0,00948 | 1 |
| 5 | 0,2434 | 0,2576 | -0,01420 | 0,00020 | 1 |
| 6 | 0,4170 | 0,3558 | 0,06118 | 0,00374 | 0 |
| 7 | 0,4925 | 0,3229 | 0,16955 | 0,02875 | 0 |
| 8 | 0,5226 | 0,6935 | -0,17086 | 0,02919 | 1 |
| | | | | 0,07991 | 63 |
| MSE | | | | 0,00665928 | |

Tabel 7. Hasil Pengujian dengan Model 4-5-1

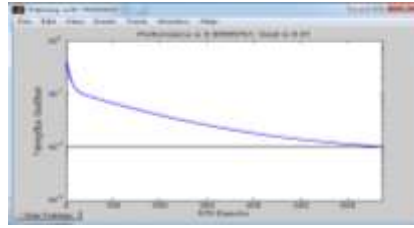
| No | Target | Output JST | Error | SSE | Hasil |
|-----|--------|------------|----------|------------|-------|
| 1 | 0,3296 | 0,5694 | -0,23977 | 0,05749 | 1 |
| 2 | 0,7519 | 0,7247 | 0,02715 | 0,00074 | 0 |
| 3 | 0,1000 | 0,2598 | -0,15980 | 0,02554 | 1 |
| 4 | 0,7222 | 0,8416 | -0,11938 | 0,01425 | 1 |
| 5 | 0,3222 | 0,3105 | 0,01172 | 0,00014 | 0 |
| 6 | 0,5444 | 0,7236 | -0,17916 | 0,03210 | 1 |
| 7 | 0,4926 | 0,7172 | -0,22461 | 0,05045 | 1 |
| 8 | 0,5963 | 0,8388 | -0,24250 | 0,05881 | 1 |
| | | | | 0,23950 | 75 |
| MSE | | | | 0,01995873 | |

3.3. Pelatihan dan Pengujian Arsitektur 4-7-1

Berikut hasil pelatihan dan pengujian dengan 8 data untuk arsitektur 4-7-1. Adapun parameter yang adalah:

| Koding Pelatihan | Koding Pengujian |
|---|---|
| <pre>>> net=newff(minmax(P),[7,1],{'logsig','purelin'},'traingd'); >> net.IW{1,1}; >> net.b{1}; >> net.LW{2,1}; >> net.b{2}; >> net.trainParam.epochs= 200000; >> net.trainParam.goal = 0.01; >> net.trainParam.Lr = 0.01; >> net.trainParam.show = 1000; >> net=train(net,P,T) [a,Pf,Af,e,Perf]=sim(net,P,[],[],T)</pre> | <pre>>> PP= >> TT= >> [a,Pf,Af,e,Perf]=sim(net,PP,[],[],TT)</pre> |

Setelah dilakukan perulangan maka ditemukan *error minimum* pada *epoch* 99540 Seperti pada Gambar 3.



Gambar 3. Pelatihan Arsitektur 4-7-1 Mencapai Goal

Untuk hasil yang lebih rinci dan mengetahui *output* dan *error* dapat dilihat pada tabel berikut

Tabel 8. Hasil Pelatihan dengan Model 4-7-1

| No | Target | Output JST | Error | SSE | Hasil |
|-----|--------|------------|----------|------------|-------|
| 1 | 0,4170 | 0,3314 | 0,08558 | 0,00732 | 0 |
| 2 | 0,7868 | 0,7919 | -0,00511 | 0,00003 | 1 |
| 3 | 0,2358 | 0,2573 | -0,02145 | 0,00046 | 1 |
| 4 | 0,6660 | 0,6659 | 0,00014 | 0,00000 | 1 |
| 5 | 0,2434 | 0,1530 | 0,09040 | 0,00817 | 0 |
| 6 | 0,4170 | 0,2365 | 0,18048 | 0,03257 | 0 |
| 7 | 0,4925 | 0,6105 | -0,11805 | 0,01394 | 1 |
| 8 | 0,5226 | 0,6548 | -0,13216 | 0,01747 | 1 |
| | | | | 0,07996 | 63 |
| MSE | | | | 0,00666302 | |

Tabel 9. Hasil Pengujian dengan Model 4-7-1

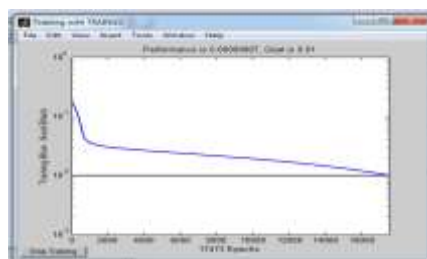
| No | Target | Output JST | Error | SSE | Hasil |
|-----|--------|------------|----------|------------|-------|
| 1 | 0,3296 | 0,2926 | 0,03703 | 0,00137 | 0 |
| 2 | 0,7519 | 0,1928 | 0,55905 | 0,31254 | 0 |
| 3 | 0,1000 | 0,1964 | -0,09640 | 0,00929 | 1 |
| 4 | 0,7222 | 0,7017 | 0,02052 | 0,00042 | 0 |
| 5 | 0,3222 | 0,2604 | 0,06182 | 0,00382 | 0 |
| 6 | 0,5444 | 0,4311 | 0,11334 | 0,01285 | 0 |
| 7 | 0,4926 | 0,5765 | -0,08391 | 0,00704 | 1 |
| 8 | 0,5963 | 0,3491 | 0,24720 | 0,06111 | 0 |
| | | | | 0,40844 | 25 |
| MSE | | | | 0,03403664 | |

3.4. Pelatihan dan Pengujian Arsitektur 4-3-5-1

Berikut hasil pelatihan dan pengujian dengan 8 data untuk arsitektur 4-3-5-1. Adapun parameter yang adalah:

| Koding Pelatihan | Koding Pengujian |
|---|---|
| <pre>net=newff(minmax(P),[3,5,1],{'logsig','tansig','logsig','traingd'); >> net.IW{1,1}; >> net.b{1}; >> net.LW{2,1}; >> net.b{2}; >> net.trainParam.epochs= 200000; >> net.trainParam.goal = 0.01; >> net.trainParam.Lr = 0.01; >> net.trainParam.show = 1000; >> net=train(net,P,T) [a,Pf,Af,e,Perf]=sim(net,P,[],[],T)</pre> | <pre>>> PP= >> TT= >> [a,Pf,Af,e,Perf]=sim(net,PP,[],[],TT)</pre> |

Setelah dilakukan perulangan maka ditemukan *error minimum* pada *epoch* 99540 Seperti pada Gambar 4.



Gambar 4. Pelatihan Arsitektur 4-3-5-1 Mencapai Goal

Untuk hasil yang lebih rinci dan mengetahui *output* dan *error* dapat dilihat pada tabel berikut.

Tabel 10. Hasil Pelatihan dengan Model 4-3-5-1

| No | Target | Output JST | Error | SSE | Hasil |
|-----|--------|------------|----------|------------|-------|
| 1 | 0,4170 | 0,4473 | -0,03032 | 0,00092 | 1 |
| 2 | 0,7868 | 0,6225 | 0,16429 | 0,02699 | 0 |
| 3 | 0,2358 | 0,4165 | -0,18065 | 0,03263 | 1 |
| 4 | 0,6660 | 0,5465 | 0,11954 | 0,01429 | 0 |
| 5 | 0,2434 | 0,2032 | 0,04020 | 0,00162 | 0 |
| 6 | 0,4170 | 0,4190 | -0,00202 | 0,00000 | 1 |
| 7 | 0,4925 | 0,5271 | -0,03465 | 0,00120 | 1 |
| 8 | 0,5226 | 0,5712 | -0,04856 | 0,00236 | 1 |
| | | | | 0,08001 | 63 |
| MSE | | | | 0,00666779 | |

Tabel 11. Hasil Pengujian dengan Model 4-3-5-1

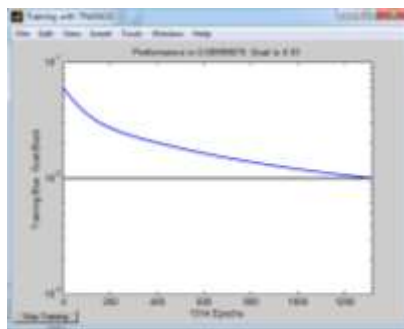
| No | Target | Output JST | Error | SSE | Hasil |
|-----|--------|------------|----------|------------|-------|
| 1 | 0,3296 | 0,2225 | 0,10713 | 0,01148 | 0 |
| 2 | 0,7519 | 0,3184 | 0,43345 | 0,18788 | 0 |
| 3 | 0,1000 | 0,1671 | -0,06710 | 0,00450 | 1 |
| 4 | 0,7222 | 0,3760 | 0,34622 | 0,11987 | 0 |
| 5 | 0,3222 | 0,1273 | 0,19492 | 0,03799 | 0 |
| 6 | 0,5444 | 0,0907 | 0,45374 | 0,20588 | 0 |
| 7 | 0,4926 | 0,2729 | 0,21969 | 0,04826 | 0 |
| 8 | 0,5963 | 0,1750 | 0,42130 | 0,17749 | 0 |
| | | | | 0,79336 | 13 |
| MSE | | | | 0,06611363 | |

3.5. Pelatihan dan Pengujian Arsitektur 4-5-5-1

Berikut hasil pelatihan dan pengujian dengan 8 data untuk arsitektur 4-5-5-1. Adapun parameter yang digunakan adalah:

| Koding Pelatihan | Koding Pengujian |
|---|---|
| <pre>>> net=newff(minmax(P),[5,5,1],{'logsig','tansig','logsig'},'traingd'); >> net.IW{1,1}; >> net.b{1}; >> net.LW{2,1}; >> net.b{2}; >> net.trainParam.epochs = 200000; >> net.trainParam.goal = 0.01; >> net.trainParam.Lr = 0.01; >> net.trainParam.show = 1000; >> net=train(net,P,T) [a,Pf,Af,e,Perf]=sim(net,P,[],T)</pre> | <pre>>> PP= >> TT= >> [a,Pf,Af,e,Perf]=sim(net,PP,[],T)</pre> |

Setelah dilakukan perulangan maka ditemukan *error minimum* pada *epoch 99540* Seperti pada gambar 4.7.



Gambar 5. Pelatihan Arsitektur 4-5-5-1 Mencapai Goal

Untuk hasil yang lebih rinci dan mengetahui *output* dan *error* dapat dilihat pada tabel berikut.

Tabel 12. Hasil Pelatihan dengan Model 4-5-5-1

| No | Target | Output JST | Error | SSE | Hasil |
|----|--------|------------|----------|---------|-------|
| 1 | 0,4170 | 0,4943 | -0,07732 | 0,00598 | 1 |
| 2 | 0,7868 | 0,8065 | -0,01971 | 0,00039 | 1 |

| No | Target | Output JST | Error | SSE | Hasil |
|----|--------|------------|----------|------------|-------|
| 3 | 0,2358 | 0,2466 | -0,01075 | 0,00012 | 1 |
| 4 | 0,6660 | 0,5577 | 0,10834 | 0,01174 | 0 |
| 5 | 0,2434 | 0,1626 | 0,08080 | 0,00653 | 0 |
| 6 | 0,4170 | 0,3943 | 0,02268 | 0,00051 | 0 |
| 7 | 0,4925 | 0,7117 | -0,21925 | 0,04807 | 1 |
| 8 | 0,5226 | 0,4410 | 0,08164 | 0,00667 | 0 |
| | | | | 0,08000 | 50 |
| | | | MSE | 0,00666636 | |

Tabel 13. Hasil Pengujian dengan Model 4-5-5-1

| No | Target | Output JST | Error | SSE | Hasil |
|----|--------|------------|----------|------------|-------|
| 1 | 0,3296 | 0,4422 | -0,11257 | 0,01267 | 1 |
| 2 | 0,7519 | 0,8142 | -0,06235 | 0,00389 | 1 |
| 3 | 0,1000 | 0,1431 | -0,04310 | 0,00186 | 1 |
| 4 | 0,7222 | 0,5130 | 0,20922 | 0,04377 | 0 |
| 5 | 0,3222 | 0,5703 | -0,24808 | 0,06154 | 1 |
| 6 | 0,5444 | 0,5912 | -0,04676 | 0,00219 | 1 |
| 7 | 0,4926 | 0,4289 | 0,06369 | 0,00406 | 0 |
| 8 | 0,5963 | 0,5280 | 0,06830 | 0,00466 | 0 |
| | | | | 0,13464 | 63 |
| | | | MSE | 0,01122006 | |

Pemilihan Arsitektur Terbaik Jaringan saraf tiruan. Hasil *software* aplikasi *Matlab 6.1* yang digunakan untuk model arsitektur 4-2-1, arsitektur 4-5-1, arsitektur 4-7-1, arsitektur 4-3-5-1 dan arsitektur 4-5-5-1 adalah 4-5-1. memperoleh pola arsitektur terbaik. Dari pola ini nanti akan digunakan untuk memprediksi penjualan KUE. Penilaian model arsitektur terbaik dilihat dari beberapa aspek seperti *epoch*, *error minimum* dan akurasi kebenaran. Untuk lebih jelas dapat dilihat pada berikut :

Tabel 14. Rekapitulasi Model Arsitektur

| Arsitektur | Training | | | Testing | |
|------------|----------|------------|---------|------------|---------|
| | Epoch | MSE | Akurasi | MSE | Akurasi |
| 4-2-1 | 118004 | 0,00666500 | 63 | 0,01965509 | 50 |
| 4-5-1 | 1104 | 0,00665928 | 63 | 0,01995873 | 75 |
| 4-7-1 | 676 | 0,00666302 | 63 | 0,03403664 | 25 |
| 4-3-5-1 | 17473 | 0,00666779 | 63 | 0,06611363 | 13 |
| 4-5-5-1 | 1314 | 0,00666636 | 50 | 0,01122006 | 63 |

Dari hasil pengujian data penjualan KUE diatas dapat kita lihat pada arsitektur 4-5-1 yang menunjukkan dari target dikurang dengan *output jst* bahwa SSE 0,23950 yang menunjukkan bahwa adanya kenaikan penjualan sebagai target. Hanya saja kenaikan yang tidak terlalu signifikan. Dari data yang didapat, bahwa *performance* perhitungan jaringan saraf tiruan dengan Algoritma *Backpropagation* adalah 75%. Dapat dilihat dengan perbandingan target yang diinginkan dengan target prediksi. Jumlah peningkatan penjualan KUE terletak pada skala minim. Dan jaringan saraf tiruan dengan menggunakan algoritma *backpropagation* dapat diterapkan dalam menganalisa prediksi penjualan KUE dengan menentukan model arsitektur terbaik dari serangkain proses training dan testing yang dilakukan

4. KESIMPULAN

Kesimpulan yang dapat diambil setelah melakukan pelatihan dan pengujian data pada jaringan syaraf tiruan dengan menggunakan algoritma *backpropagation* adalah sebagai berikut, Jaringan Syaraf Tiruan dapat digunakan untuk memprediksi penjualan kue berdasarkan kriteria sebagai perbandingan target yang diinginkan dengan target prediksi. Jumlah peningkatan penjualan KUE terletak pada skala minim. Akuratnya sebuah hasil pelatihan ataupun pengujian terhadap nilai output dalam menentukan prediksi penjualan kue pada pola arsitektur jaringan yang digunakan. Arsitektur yang terbaik untuk penelitian metode Jaringan Syaraf Tiruan dalam memprediksi penjualan kue pada UD Mak Kembar menggunakan algoritma *backpropagation* adalah model arsitektur 4-5-1 dengan proses perulangan (*epoch*) pada saat pelatihan dengan nilai *epoch* = 75% dan pencapaian SSE pada saat pengujian dengan SSE = 0,23950.

REFERENCES

- [1] A. Ahmad, P. M. Putri, W. Alifah, and I. Gunawan, "Analisis jaringan syaraf tiruan metode *backpropagation* dalam memprediksi ketersediaan komoditas beras berdasarkan provinsi di indonesia," *RESISTOR*, vol. 2, no. 1, pp. 4–16, 2019.
- [2] S. P. Siregar, A. Wanto, and Z. M. Nasution, "Analisis Akurasi Arsitektur JST Berdasarkan Jumlah Penduduk Pada Kabupaten / Kota di Sumatera Utara," pp. 526–536, 2018.
- [3] A. Wanto and A. P. Windarto, "Analisis Prediksi Indeks Harga Konsumen Berdasarkan Kelompok Kesehatan Dengan Menggunakan Metode *Backpropagation*," vol. 2, pp. 37–44, 2019.
- [4] N. Z. Purba and D. Sitompul, "Analisis tingkat akurasi algoritma *backpropagation* dalam prediksi produksi ubi kayu di

- provinsi indonesia,” no. 3, pp. 87–97, 2018.
- [5] I. S. Purba and A. Wanto, “Prediksi Jumlah Nilai Impor Sumatera Utara Menurut Negara Asal Menggunakan Algoritma Backpropagation,” vol. 17, no. 3, pp. 302–311, 2018.
- [6] R. H. Pranata and L. Hakim, “Penerapan Algoritma Jaringan Syaraf Tiruan Untuk Memprediksi Harga Tiket Pesawat,” vol. 3, no. 2, pp. 134–140, 2018.
- [7] A. MS and M. H. Baysha, “Pengaruh Media Pembelajaran Matrix Laboratory (Matlab) Terhadap Hasil Belajar Siswa,” vol. 3, pp. 10–20, 2018.
- [8] Wibawanto, “Bab Ii Landasan Teori,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 8–24, 2018.
- [9] A. Mubarak, “Rancang Bangun Aplikasi Web Sekolah Menggunakan Uml (Unified Modeling Language) Dan Bahasa Pemrograman Php (Php Hypertext Preprocessor) Berorientasi Objek,” vol. 2, no. 1, pp. 19–25, 2019.
- [10] A. Hendini, “Pemodelan UML Sistem Informasi Monitoring Penjualan Dan Stok Barang (Studi Kasus : Distro Zhezha Pontianak),” *J. Inform.*, vol. 10, no. 2, pp. 1–10, 2016.
- [11] R. Nuraini, “Desain algoritma operasi perkalian matriks menggunakan metode flowchart,” vol. 1, no. 1, pp. 144–151, 2015.
- [12] N. F. Hasan, K. Kusriani, and H. Al Fatta, “Analisis Arsitektur Jaringan Syaraf Tiruan Untuk Peramalan Penjualan Air Minum Dalam Kemasan,” *J. Rekayasa Teknol. Inf.*, vol. 3, no. 1, p. 1, 2019.
- [13] S. Solikhun, A. Revi, S. Ramadan, and R. Novita Sari, “Model Jaringan Syaraf Tiruan Dalam Memprediksi Pendapatan Perkapita Masyarakat Perkotaan Pada Garis Kemiskinan Berdasarkan Propinsi,” *Klik - Kumpul. J. Ilmu Komput.*, vol. 5, no. 2, p. 122, 2018.
- [14] A. P. Windarto, M. R. Lubis, and Solikhun, “Model Arsitektur Neural Network Dengan Backpropogation Pada Prediksi Total Laba,” *Kumpul. J. Ilmu Komput.*, vol. 5, no. 2, pp. 147–158, 2018.
- [15] M. D. Nasution, E. Nasution, and F. Haryati, “Pengembangan Bahan Ajar Metode Numeri,” vol. 6, pp. 69–80, 2017.