

---

## Implementasi *Neural Network* Untuk Prediksi Penjualan Produk (Studi Kasus Siomay Mirza)

Agung Firmansyah<sup>1\*</sup>, Mutaqin Akbar<sup>2</sup>

Program Studi Informatika, Universitas Mercu Buana Yogyakarta

Jl. Jembatan Merah No. 84C Gejayan, Yogyakarta, Indonesia

\*Email Corresponding Author: agf.firman@gmail.com

### Abstrak

Ketepatan prediksi produksi sangat diperlukan dalam usaha penjualan produk, khususnya produk makanan yang bahan bakunya memiliki umur simpan yang pendek. Tulisan ini bertujuan menyajikan model aplikasi sistem berbasis algoritma *Neural Network* untuk memprediksi jumlah penjualan *Siomay* di masa datang, sebagai acuan mempersiapkan bahan baku dengan tepat. Prediksi menggunakan data historis sebagai data pelatihan sistem. Uji coba *Neural Network* menggunakan 357 data historis penjualan, 7 data awal digunakan sebagai referensi, 315 data sebagai data latih, dan 35 data terakhir sebagai data uji. Variabel masukan *neural network* berupa rerata penjualan 7 hari sebelumnya, nilai penjualan 1 hingga 3 hari sebelumnya, akhir bulan, identifikasi hari diskon/benefit, dan akhir pekan. Pelatihan *neural network* dilakukan dengan model 7-6-1, 7-7-1 dan 7-8-1 dengan laju pelatihan [0.05, 0.1, 0.5, 1, 2 dan 3]. Dari hasil pengujian menggunakan 35 data uji pada ketiga model didapatkan nilai kesalahan terkecil pada model 7-7-1 dengan *Mean Square Error* (MSE) 1879.8 dan tingkat akurasi prediksi sebesar 90.8%.

**Kata kunci:** Jaringan saraf tiruan; Data latih; Data uji; Tingkat akurasi; Mean Square Error

### Abstract

*The accuracy of production predictions is very necessary in the business of selling products, especially food products whose raw materials have a short shelf life. This paper aims to present a system application model based on the Neural Network algorithm to predict the number of Siomay sales in the future, as a reference for preparing raw materials appropriately. Prediction uses historical data as system training data. The Neural Network trial uses 357 historical sales data, 7 initial data are used as references, 315 data as training data, and the last 35 data as test data. The input variables for the neural network are the average sales of the previous 7 days, the value of sales from the previous 1 to 3 days, the end of the month, identification of discount/benefit days, and weekends. Neural network training was carried out using 7-6-1, 7-7-1 and 7-8-1 models with training rates [0.05, 0.1, 0.5, 1, 2 and 3]. From the test results using 35 test data on the three models, the smallest error value is obtained in the 7-7-1 model with a Mean Square Error (MSE) of 1879.8 and a prediction accuracy rate of 90.8%.*

**Keywords:** Artificial neural network; Training data; Test data; Accuracy level; Mean Square Error

### 1. Pendahuluan

Dalam perdagangan sebuah produk sangat mungkin sekali untuk memaksimalkan keuntungan dan meminimalisir kerugian, tetapi pada kondisi nyata permintaan konsumen tidak mudah diketahui, pada saat waktu tertentu permintaan sangat banyak sampai kekurangan persediaan dilain waktu permintaan berkurang sampai produk sisa terlalu banyak dan berubah lagi pada waktu berikutnya, apabila terjadi kasus permintaan lebih besar dari produksi maka disitu terlihat dengan jelas sebuah penolakan pada keuntungan dimana konsumen yang akan membeli diwaktu tersebut tidak jadi membeli dikarenakan stok produk sudah tidak tersedia, dan apabila dimana permintaan lebih kecil dari produksi terjadi maka produk akan tersisa pada waktu tersebut, pada produk tertentu mungkin tidak berpengaruh besar tapi lain halnya pada produk-produk olahan makanan dimana sisa barang produksi yang gagal terjual pada waktu tertentu akan berpengaruh dengan kualitas produk itu sendiri, hal tersebut akan sangat mempengaruhi kualitas lebih tepatnya berpengaruh pada keuntungan yang didapat.

Salah satu permasalahan utama perusahaan makanan adalah karena karakteristik khas mereka seperti umur simpan pendek produk terkait, kebutuhan memiliki kualitas produk tinggi dan ketidakpastian permintaan konsumen.

Prediksi sangat diperlukan pada kasus penjualan produk siomay di CV. Pangan Berkah terkenal dengan Siomay Mirza yang berada di Yogyakarta, dimana dalam pembuatan siomay memerlukan bahan baku, peralatan dan sumber daya yang harus dipersiapkan. Bahan baku yang digunakan dalam pembuatannya adalah daging, tepung dan lain sebagainya yang didapatkan di pasar dimana harga bahan baku bisa saja naik pada waktu tertentu, hal tersebut akan mempengaruhi biaya produksi dan harga jual namun tidak memungkinkan merubah harga produk yang dijual terlalu sering, dengan mengetahui jumlah produk yang akan dijual di waktu berikutnya akan membantu penentuan pembelian bahan baku bertahap setiap waktu tertentu ataupun pembelian secara banyak untuk menghindari kemungkinan lonjakan harga bahan baku. Tidak tepatnya menentukan jumlah pembuatan produk akan mengalami kekurangan persediaan ataupun sisa pada waktu penjualan, jika terjadi sisa maka akan mempengaruhi kualitas dan ketersediaan peralatan untuk penyimpanan produk sisa tersebut dan akan memakan biaya lagi.

Zudaruddin menyampaikan dalam bukunya bahwa metodologi peramalan atau prediksi telah ada sejak abad kesembilan belas dan perkembangan terkini seperti *neural network* telah secara signifikan memperluas bidang peramalan[1]. *Neural network* dapat diterapkan untuk perkiraan penjualan produk di berbagai industri salah satunya adalah industri makanan[2]. Prediksi penjualan produk menggunakan *neural network* menghasilkan tingkat kesalahan lebih rendah daripada dengan model *simple moving average* dalam penelitian yang berjudul "*Sales Prediction through Neural Networks for a Small Dataset*"[3].

Makalah ini menyajikan sistem prediksi penjualan menggunakan *neural network* diharapkan mendapatkan hasil prediksi penjualan produk yang lebih akurat dan hasil prediksi dapat digunakan untuk mempersiapkan segala sesuatu dalam pembuatan produk sehingga siap untuk penjualan berikutnya.

## 2. Tinjauan Pustaka

Dalam penelitian yang berjudul "*Implementation of Artificial Neural Network in Forecasting Sales Volume in Tokopedia Indonesia*" menerapkan *neural network* dalam melakukan prediksi penjualan barang dagangan *seller* di *marketplace* Tokopedia Indonesia sehingga dapat mempersiapkan atau menyimpan barang yang diprediksi akan meningkat dalam penjualan. Dengan mempertimbangkan variabel data yaitu jenis barang, harga barang, jumlah orang yang melihat item, waktu pengiriman barang ke kurir, pelanggan dari toko barang, jumlah orang yang telah menyukai sebagai parameter masukan *neural network*. Hasil analisis penelitian ini menunjukkan bahwa model jaringan yang diperoleh mencapai tingkat akurasi 95%[4].

Dalam penelitian yang berjudul "*Online Sales Forecasting of E-Products of Online Store Using the Artificial Neural Network Approach and Customer Sentiment Analysis*" bertujuan untuk memperkirakan apakah produk dibeli oleh pelanggan, data yang diperlukan diekstraksi untuk memperkirakan penjualan dengan memisahkan kata-kata dan kalimat, memberi label komponen pidato dan *rooting* kata-kata. Prediksi penjualan produk dilakukan dengan menggunakan jaringan saraf dan menambahkan 4 indikator yang efektif dari iklan, harga, diskon dan pengiriman gratis. Dan akhirnya, keakuratan 83,65% tercapai dengan meningkatkan jaringan saraf menggunakan algoritma *adaboost*[5].

Dalam penelitian yang berjudul "*Implementation of Product Sales Forecast Using Artificial Neural Network Method*" melakukan prediksi menggunakan jaringan saraf tiruan *backpropagation* dengan laju pelatihan 0,1, lapisan tersembunyi 4, nilai kesalahan 0,01, dan atribut yang digunakan pada jaringan adalah harga jual dan jumlah terjual, jaringan berhasil diimplementasikan dengan akurasi perkiraan rata-rata 97% dengan MAPE 3.35%[6].

Penelitian yang berjudul "*Prediksi Penjualan Menggunakan Algoritma Neural Network: Studi Kasus di PT. Balaraja Food Makmur Abadi*" melakukan prediksi penjualan untuk bulan berikutnya dengan metode pengenalan pola yaitu *neural network* dengan algoritma *backpropagation* yang menggunakan data penjualan dari tahun 2013 sampai dengan 2015, penghitungan yang telah dilakukan menggunakan *neural network backpropagation* maka dapat dihasilkan akurasi prediksi mencapai 90,8%[7].

Penelitian yang berjudul "*Penerapan Jaringan Syaraf Tiruan dalam Memprediksi Jumlah Ekspor Minyak Bumi di Indonesia*" melakukan prediksi tingkat ekspor minyak bumi di Indonesia di tahun-tahun berikutnya dengan menggunakan jaringan saraf tiruan *backpropagation* dengan

menggunakan data jumlah ekspor minyak bumi mentah menurut negara tujuan utama. Dari pengujian diperoleh hasil tingkat akurasi sebesar 90% dengan model arsitektur yang terbaik yaitu 4-41-1 serta *epoch* sebanyak 2063 iterasi[8].

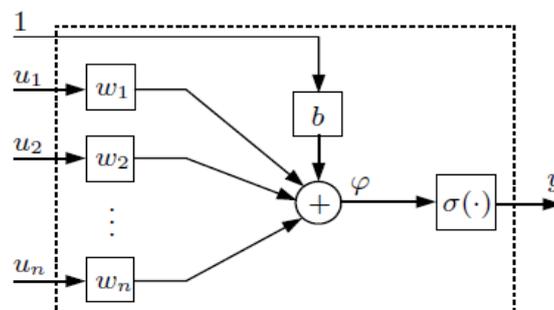
Dari beberapa sumber referensi yang dirujuk diatas dapat ditarik hipotesa bahwa penerapan *neural network* untuk mendapatkan nilai prediksi penjualan dari data yang sudah ada sebelumnya dengan menambahkan variabel dapat dilakukan bahkan dapat menghasilkan nilai prediksi yang akurat. Dari beberapa referensi yang dirujuk, bahwa penggunaan atribut dan parameter *neural network* bisa berbeda disetiap kasusnya. Penelitian ini merujuk pada penelitian yang sudah dilakukan dengan merubah dan menambah variabel atau atribut bahkan model *neural network* sehingga cocok diaplikasikan pada kasus dalam penelitian ini, dengan bertujuan membantu menyelesaikan kasus terkait.

**2.1 Neural Network**

*Neural network* adalah model komputer yang diturunkan dari konsep otak manusia yang disederhanakan dan penelitian *neural network* telah aktif sejak tahun 1940-an[9] serta Suhartono[10] mengemukakan bahwa ide dasar *neural network* dimulai dari otak manusia. *Neural network* terdiri dari banyak elemen (*Artificial Neuron*), dikelompokkan ke dalam lapisan dan sangat saling berhubungan (dengan sinapsis), struktur ini memiliki beberapa *input* dan *output*, yang dilatih untuk bereaksi (atau memberikan nilai) dengan cara yang memberikan nilai *input* rangsangan. Jaringan saraf diperlukan pembelajaran untuk berperilaku (*Learning*) dan seseorang harus memberikan pengajaran atau pelatihan (*Training*), berdasarkan pengetahuan tentang masalah-masalah sebelumnya. Properti terpenting dari jaringan syaraf tiruan adalah kemampuannya untuk belajar dari serangkaian pola pelatihan, yaitu mampu menemukan model yang sesuai dengan data[11]. *Neural network* (jaringan saraf) adalah alat pemrosesan data paralel yang mampu mempelajari dependensi fungsional data. Fitur ini sangat berguna saat memecahkan masalah pengenalan pola yang berbeda. Selain itu, jaringan saraf dicirikan oleh tingkat komputasi yang tinggi, toleransi kesalahan input yang besar, dan kemampuan beradaptasi dalam hal kondisi kerja yang berubah-ubah[12].

Jaringan saraf tiruan dibangun menggunakan sejumlah unit pemrosesan tunggal yang disebut *neuron*. Model McCulloch-Pitts adalah model *neuron* klasik yang mendasar dan dijelaskan oleh Persamaan 1[12].

$$y = \sigma(\phi) = \sigma(\sum_{i=1}^n w_i u_i + b), \dots\dots\dots(1)$$



Gambar 1. Skema *neuron* dengan n masukan dan satu keluaran

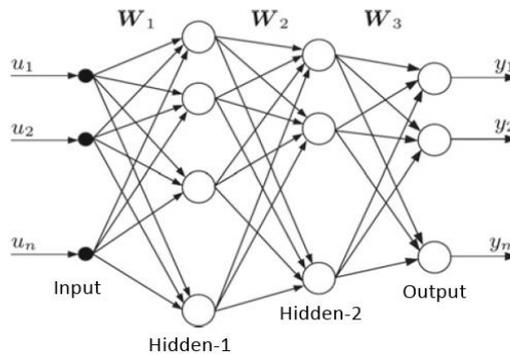
Dimana  $u_i, i=1,2,\dots,n$ , menyimbolkan *input*,  $b$  adalah bias (ambang batas),  $w_i$  menunjukkan bobot,  $\phi$  menyimbolkan sinyal aktivasi,  $\sigma(\cdot)$  merupakan fungsi aktivasi nonlinear. Parameter bias dapat diimplementasikan dalam bentuk tambahan bobot input tereksitasi oleh sinyal konstan bernilai sama dengan 1. Fungsi aktivasi adalah salah satu parameter penting dalam *neural network* [13]. Pilihan fungsi aktivasi yang sesuai bergantung pada aplikasi spesifik dari jaringan saraf[12]. Ada beberapa tipe fungsi aktivasi yang digunakan dalam *neural network*, salah satunya adalah fungsi *sigmoid*[14]. Pada fungsi *sigmoid biner* memiliki batas nilai antara 0 sampai 1, sedangkan pada fungsi *sigmoid bipolar* memiliki batas nilai antara -1 sampai 1[15]. Dari nilai asli dapat dilakukan normalisasi agar nilai data sesuai dengan *range* fungsi aktivasi yang digunakan. Ada beberapa teknik yang dapat diterapkan untuk normalisasi data, salah satunya adalah menggunakan normalisasi *min-max*[16]. Representasi *input-output* dari *neuron* buatan ditunjukkan pada Gambar 1[12].

**2.2 Feed Forward Multilayer Network**

*Multilayer perceptron* adalah sebuah *network*, dimana beberapa *neuron* digabungkan kedalam sebuah lapisan (Gambar 2). Jaringan memiliki lapisan masukan, satu atau lebih lapisan tersembunyi, dan lapisan keluaran. Harus ditekankan bahwa tautan antara *neuron* dirancang sedemikian rupa sehingga setiap elemen dari lapisan sebelumnya terhubung dengan setiap elemen dari lapisan berikutnya. Koneksi ini diberikan koefisien bobot yang sesuai, untuk setiap kasus yang berbeda, tergantung pada tugas yang dirancang untuk dipecahkan oleh jaringan. Bobot jaringan awal dapat diberikan nilai secara acak[17], biasanya dalam kisaran -0,5 sampai dengan 0,5 dan kemudian diperbarui untuk mendapatkan output yang konsisten[18]. Komputasi jaringan saraf pada Gambar 2 dapat dihasilkan Persamaan 2.

$$y = \sigma_3(W_3\sigma_2(W_2\sigma_1(W_1u))) \dots\dots\dots(2)$$

Dari Persamaan 2, dimana  $\sigma_1$ ,  $\sigma_2$  dan  $\sigma_3$  adalah fungsi aktivasi bernilai vektor yang mendefinisikan transformasi sinyal saraf melalui lapisan pertama, kedua dan keluaran. Gambar 2 menunjukkan  $W_1$ ,  $W_2$  dan  $W_3$  adalah matriks koefisien bobot yang menentukan intensitas koneksi di antara neuron di lapisan tetangga, dimana  $u$  adalah vektor masukan dan  $y$  adalah vektor keluaran[12].



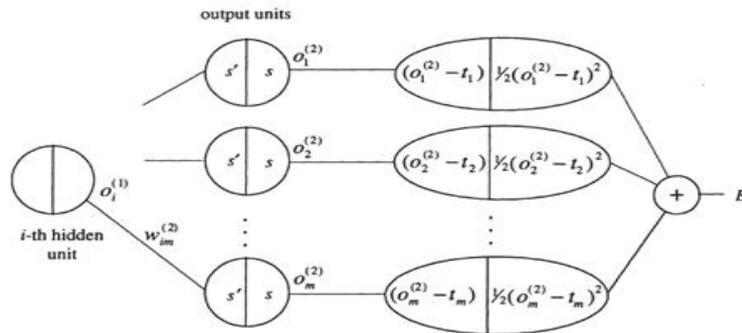
Gambar 2. *Feed forward network* dengan dua lapisan tersembunyi

**2.3 Backpropagation**

Patan[12] mengemukakan bahwa dari sudut pandang teknis, pelatihan jaringan saraf tidak lain adalah penentuan nilai koefisien bobot di antara unit pemrosesan tetangga. Algoritma *backpropagation* adalah algoritma pelatihan fundamental yang digunakan dalam jaringan *multilayer feed-forward*. *Backpropagation* memberikan prosedur yang memungkinkan untuk mengubah nilai bobot yang ditetapkan untuk koneksi di antara unit pemrosesan di lapisan tetangga. Algoritma ini adalah tipe berulang dan didasarkan pada minimalisasi *sumsquared error* yang memanfaatkan *gradient descent* untuk optimasi.

Gambar 3 menunjukkan jaringan yang diperluas untuk perhitungan fungsi kesalahan untuk menyederhanakan penanganan pasangan *input-output* tunggal (o, t) dan menggeneralisasi untuk contoh pelatihan p. Jaringan telah diperpanjang dengan menambahkan unit lapisan. Pada bagian sebelah kanan merupakan perhitungan *quadratic deviation*  $\frac{1}{2}(o_i^{(2)} - t_i)^2$  untuk setiap komponen *output vector* i dan bagian kiri  $(o_i^{(2)} - t_i)$ . Setiap unit output i pada jaringan asli menghitung *sigmoid* s dan menghasilkan keluaran  $o_i^{(2)}$ . Penambahan *quadratic deviations* menghasilkan nilai kesalahan E(Persamaan 3). Fungsi kesalahan untuk contoh p *input-output* dapat dihitung dengan membuat jaringan p seperti yang ditunjukkan, satu untuk setiap pasangan pelatihan, dan menambahkan output dari semuanya untuk menghasilkan total kesalahan pelatihan.

$$E = \frac{1}{2} \sum_{i=1}^p \|o_i - t_i\|^2 \dots\dots\dots(3)$$



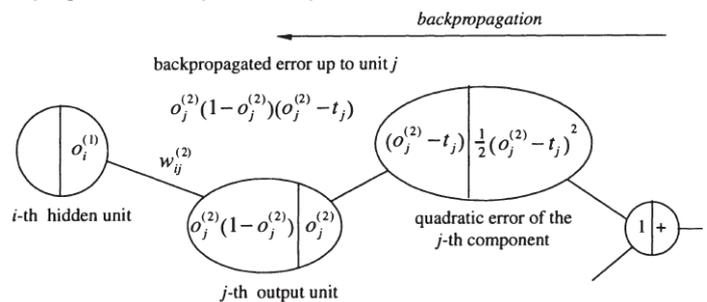
Gambar 3. Perpanjangan jaringan untuk perhitungan total error

Setelah memilih bobot awal jaringan (koneksi di antara unit pemrosesan) secara acak, algoritma *backpropagation* digunakan untuk menghitung koreksi yang diperlukan. Algoritma dapat didekomposisi dalam empat langkah berikut:

- 1) Perhitungan umpan maju (*feed-forward*),
- 2) Perhitungan *backpropagation* pada lapisan keluaran,
- 3) Perhitungan *backpropagation* pada lapisan tersembunyi,
- 4) Pembaharuan bobot.

Algoritma berhenti ketika nilai fungsi kesalahan telah menjadi cukup kecil.

Pada langkah pertama yaitu perhitungan *feed-forward*, vektor  $\mathbf{o}^{(1)}$  dan  $\mathbf{o}^{(2)}$  dihitung dan disimpan. Hasil perhitungan turunan dari fungsi aktivasi juga disimpan di setiap unitnya. Langkah kedua yaitu *backpropagation* ke lapisan *output*.



Gambar 4. *Backpropagation* ke jalur keluaran

Dari jalur ini (Gambar 4) kita dapat menyederhanakan semua istilah multiplikasi yang menentukan dengan *backpropagated error*  $\delta_j^{(2)}$ . Sehingga menghasilkan Persamaan 4,

$$\delta_j^{(2)} = \mathbf{o}_j^{(2)} (1 - \mathbf{o}_j^{(2)}) (\mathbf{o}_j^{(2)} - \mathbf{t}_j), \dots \dots \dots (4)$$

dan *partial derivative* yang dicari adalah Persamaan 5,

$$\frac{\partial E}{\partial w_{ij}^{(2)}} = \left[ \mathbf{o}_j^{(2)} (1 - \mathbf{o}_j^{(2)}) (\mathbf{o}_j^{(2)} - \mathbf{t}_j) \right] = \delta_j^{(2)} \mathbf{o}_i^{(1)} \dots \dots \dots (5)$$

perlu diingat bahwa untuk langkah terakhir ini menganggap bobot  $w_{ij}^{(2)}$  menjadi variabel dan inputnya  $\mathbf{o}_i^{(1)}$  konstan.

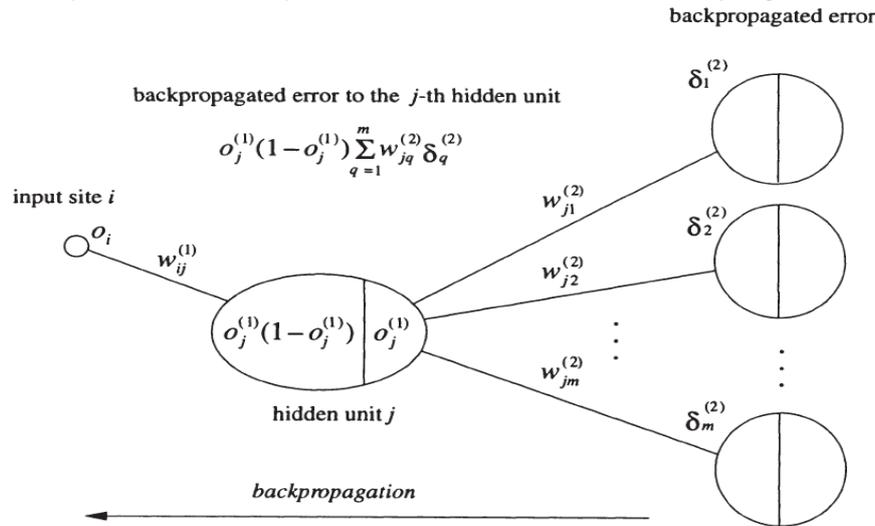
Langkah ketiga yaitu *backpropagation* ke lapisan tersembunyi, yang berarti melakukan perhitungan turunan parsial  $\partial E / \partial w_{ij}^{(1)}$ . Setiap unit  $j$  pada lapisan tersembunyi terhubung ke unit  $q$  pada lapisan keluaran dengan memiliki sebuah bobot  $w_{jq}^{(2)}$ , untuk  $q = 1, \dots, m$ . Propagasi balik kesalahan ke arah  $j$  unit lapisan tersembunyi harus dihitung dengan mempertimbangkan semua jalur yang terhubung (Gambar 5) kemudian mendapatkan Persamaan 6.

$$\delta_j^{(1)} = \mathbf{o}_j^{(1)} (1 - \mathbf{o}_j^{(1)}) \sum_{q=1}^m w_{jq}^{(2)} \delta_q^{(2)}, \dots \dots \dots (6)$$

dan *partial derivative* yang dicari adalah Persamaan 7,

$$\frac{\partial E}{\partial w_{ij}^{(1)}} = \delta_j^{(1)} \mathbf{o}_i \dots \dots \dots (7)$$

*Backpropagated error* dapat dihitung dengan cara yang sama untuk sejumlah lapisan tersembunyi dan persamaan untuk *partial derivatives E* memiliki bentuk yang sama.



Gambar 5. Backpropagation ke jalur lapisan tersembunyi

Langkah keempat yaitu memperbarui bobot, setelah menghitung semua derivatif parsial maka bobot diperbarui di arah gradien negatif. Laju pelatihan  $\gamma$  menentukan langkah panjang koreksi. Persamaan 8 dan Persamaan 9 merupakan persamaan untuk koreksi bobot,

$$\Delta w_{ij}^{(2)} = -\gamma o_i^{(1)} \delta_j^{(2)}, \text{ untuk } i=1, \dots, k+1; j=1, \dots, m; \dots \dots \dots (8)$$

$$\Delta w_{ij}^{(1)} = -\gamma o_i \delta_j^{(1)}, \text{ untuk } i=1, \dots, n+1; j=1, \dots, k; \dots \dots \dots (9)$$

dimana dengan ketentuan  $o_{n+1} = o_{k+1}^{(1)} = 1$ . Ini sangat penting untuk membuat koreksi terhadap bobot hanya setelah *backpropagated error* dihitung untuk semua unit di jaringan[17].

**3. Metodologi**

Penelitian ini bertujuan untuk mengembangkan suatu sistem informasi yang bertujuan untuk memprediksi jumlah penjualan di masa mendatang berdasarkan data-data historis penjualan yang telah terjadi, langkah-langkah penelitian terlihat pada Gambar 6.



Gambar 6. Tahapan Penelitian

**3.1 Identifikasi Kebutuhan**

Bahan penelitian yang digunakan adalah data penjualan siomay yang berisikan tanggal, hari serta jumlah produk yang terjual setiap harinya pada awal bulan tahun 2020 hingga akhir bulan tahun 2020 sebagai kebutuhan masukan. Terdapat sebanyak 357 data aktual yang digunakan pada penelitian ini. Dalam kebutuhan proses sistem memerlukan variabel-variabel masukan yang memperkuat kinerja sebuah *neural network* untuk melakukan prediksi dan dimana *neural network* itu sendiri memerlukan beberapa nilai parameter untuk melakukan proses pelatihannya.

**3.2 Representasi Pengetahuan**

Pelatihan *neural network* memerlukan variabel-variabel masukan dan beberapa parameter agar *neural network* dapat bekerja. Dalam penelitian ini faktor yang mempengaruhi tingkat penjualan ditransformasikan kedalam variabel masukan jaringan, Tabel 1 adalah tabel keterangan masukan dan keluaran (target) yang digunakan pada model *neural network* tahap pelatihan ini. Penulis menentukan variabel masukan tersebut adalah merupakan beberapa faktor yang mempengaruhi jumlah penjualan yang terjadi secara umum.

Table 1. Keterangan variabel masukan dan keluaran

Kode	Keterangan
X1	Rata-rata jumlah penjualan 7 hari sebelumnya.
X2, X3, X4	Nilai penjualan 3 hari sebelumnya. Dimana X1 adalah nilai 1 hari sebelumnya, X2 adalah 2 hari sebelumnya dan X3 adalah 3 hari sebelumnya.
X5	Kategori tanggal terhadap bulan. Dengan menggunakan nilai 0 untuk awal bulan dan nilai 1 untuk akhir bulan. Dimana tanggal 1 hingga 15 termasuk awal bulan.
X6	Kategori hari di minggu spesial/benefit, hari raya, akhir tahun dan adanya diskon, dengan nilai 0 untuk hari biasa dan nilai 1 untuk spesial.
X7	Merupakan akhir pekan, dengan nilai 0 dan 1 untuk nilai variabel akhir pekan. Dimana nilai 1 adalah akhir pekan yaitu hari Sabtu dan Minggu.
Y	Variabel keluaran (target). Nilai penjualan yang terjadi.

Neural network memerlukan beberapa parameter dalam melakukan proses pelatihannya, parameter yang dibutuhkan antara lain adalah laju pelatihan, maksimum *epoch*, bobot awal disetiap tautan antar *neuron* dan fungsi aktivasi. Fungsi aktivasi pada penelitian ini menggunakan fungsi aktivasi *sigmoid biner*, batas nilai pada fungsi aktivasi ini antara 0 sampai 1, sehingga normalisasi data diperlukan. Normalisasi dilakukan untuk menyesuaikan nilai data dengan *range* fungsi aktivasi yang digunakan dalam *neural network*. Tujuan normalisasi adalah agar diperoleh interval data yang sesuai dengan fungsi aktivasi yang digunakan yaitu fungsi aktivasi *sigmoid biner*. Sebagai catatan bahwa nilai dari variabel masukan dan keluaran adalah nilai yang sudah dinormalisasi. Sebelum digunakan sebagai masukan, data dinormalisasi terlebih dahulu.

Proses normalisasi data dilakukan dengan normalisasi *min-max*, diketahui nilai terkecil penjualan adalah 238 dan nilai tertinggi adalah 484, penelitian ini menggunakan interval 0.1 sampai 0.9 sehingga masih dalam lingkup *range* fungsi aktivasi *sigmoid biner* dan persamaan normalisasi menjadi

$$x' = \frac{x-238}{484-238} (0.9 - 0.1) + 0.1, \dots\dots\dots(10)$$

dan nilai asli dapat dihitung kembali (de-normalisasi) dengan persamaan

$$x = \frac{(x' - 0.1)}{(0.9 - 0.1)} (484 - 238) + 238, \dots\dots\dots(11)$$

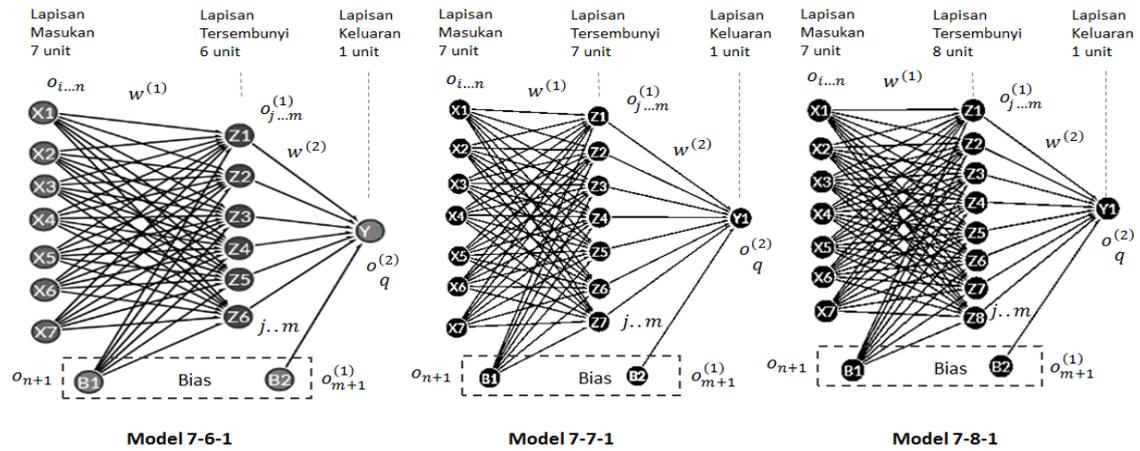
dimana x adalah nilai asli dan x' adalah nilai normalisasi, persamaan ini digunakan sebagai basis perhitungan normalisasi dan de-normalisasi data pada penelitian ini.

Penelitian ini dilakukan percobaan dari 3 model *simple neural network* sehingga dapat digunakan perbandingan nilai hasil pelatihan untuk mencari model yang cocok. Pada penelitian ini menggunakan model 7-6-1, 7-7-1 dan 7-8-1 sebagai perbandingan. Model *neural network* dapat dilihat pada Gambar 7.

Dalam proses pelatihan *neural network* pada penelitian ini menggunakan langkah-langkah sebagai berikut:

- 1) Persiapan data.
- 2) Inisiasi.
- 3) Perhitungan umpan maju (*feed forward*).
- 4) Perhitungan galat yang terjadi.
- 5) Perhitungan *backpropagation*.
- 6) Pembaruan bobot.

Pada tahap pelatihan ini menghasilkan keluaran data yaitu nilai-nilai bobot pada setiap tautan yang terhubung antar *neuron* dan informasi galat yang terjadi pada saat proses pelatihan pada setiap model. Nilai galat dari hasil pelatihan digunakan sebagai ukuran tingkat kecocokan model tersebut.



Gambar 7. Model *neural network*

**3.3 Tahap Pengujian dan Evaluasi**

Pada penelitian ini melatih 3 model *neural network*, dimana model yang dilatih terbatas hanya berbeda jumlah *neuron* pada lapisan tersembunyi dan nilai parameter, setelah model *neural network* yang telah dilatih dengan data latih pada tahap pelatihan dibandingkan diantaranya, model dengan nilai kesalahan terkecil digunakan sebagai pilihan untuk evaluasi. Evaluasi dilakukan dengan membandingkan data hasil prediksi dengan data uji akan mengetahui tingkat akurasi atau kinerja dari model *neural network* pada penelitian ini. Pada penelitian ini menggunakan *Mean Square Error* (MSE) untuk mengevaluasi hasil prediksi.

**4. Hasil dan Pembahasan**

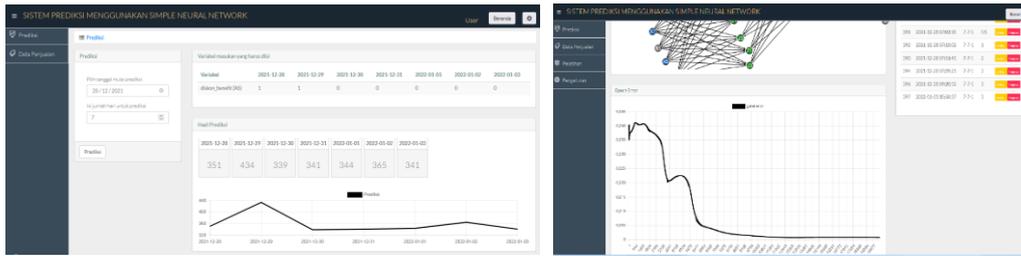
Penelitian ini menggunakan total 357 data aktual penjualan harian siomay, data merupakan data penjualan yang berurutan berdasarkan tanggal, data penjualan tersebut digunakan dalam penelitian ini sebagai data pelatihan dan pengujian. Dimana sebanyak 7 data awal dijadikan sebagai referensi untuk variabel masukan, dimulai dari data ke-8 dan sebanyak 315 data sebagai data pelatihan jaringan dan 35 data terakhir sebagai data pengujian. Tabel 2 adalah tampilan sebagian data aktual penjualan.

Tabel 2. Aktual data penjualan

No	Tanggal	Hari	Terjual	Ada diskon
1	02/01/2020	Kamis	352	Tidak
2	03/01/2020	Jumat	352	Tidak
...	...	...	...	...
8	09/01/2020	Kamis	321	Tidak
9	10/01/2020	Jumat	399	Tidak
10	11/01/2020	Sabtu	292	Tidak
11	12/01/2020	Minggu	310	Tidak
...	...	...	...	...
59	29/02/2020	Sabtu	341	Iya
60	01/03/2020	Minggu	437	Iya
...	...	...	...	...
323	23/11/2020	Senin	400	Tidak
324	24/11/2020	Selasa	372	Tidak
...	...	...	...	...
356	26/12/2020	Sabtu	370	Tidak
357	27/12/2020	Minggu	344	Tidak

**4.1. Pengujian Sistem**

Pengujian program dilakukan setelah selesai tahap pembuatan program untuk mengetahui kebenaran sistem dalam melakukan perhitungan prediksi. Tahap pengujian ini dilakukan dengan cara menguji tingkat kebenaran sistem dalam proses prediksi dengan data aktual. Gambar 8 merupakan tampilan utama sistem yang telah dibangun.



Gambar 8. Tampilan sistem

4.2. Proses Pelatihan

Proses pelatihan dilakukan dengan menggunakan 315 data dari 357 semua data penjualan aktual dimulai dari data ke-8 hingga data ke-322 yang sudah dinormalisasi agar dapat dimasukkan ke dalam sistem, dimana data aktual ke-1 hingga ke-7 hanya dijadikan sebagai referensi untuk mendapatkan nilai variabel masukan X1 hingga X4. Sebagian data hasil normalisasi dapat dilihat pada Tabel 4, pada urutan 1 adalah data ke-8 dari data penjualan aktual.

Tabel 3. Data latih yang dinormalisasi

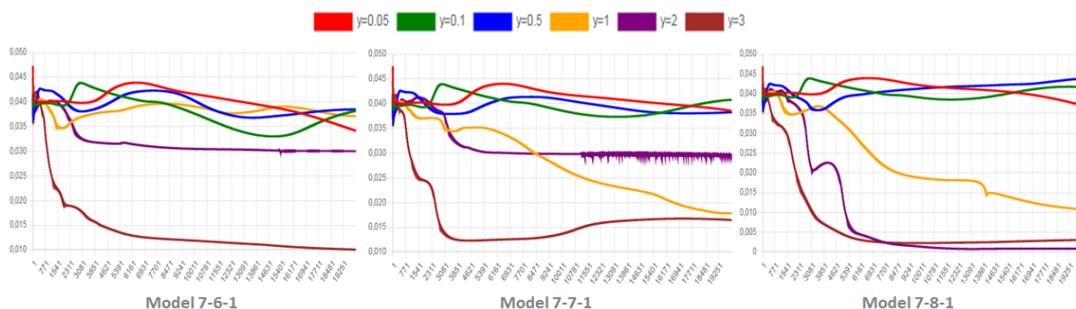
No	X1	X2	X3	X4	X5	X6	X7	Y
1	0.558072	0.412195	0.428455	0.886992	0	0	0	0.369919
2	0.54367	0.369919	0.412195	0.428455	0	0	0	0.623577
3	0.565505	0.623577	0.369919	0.412195	0	0	1	0.275610
...	...	...	...	...	...	...	...	...
314	0.435424	0.460976	0.373171	0.275610	1	0	1	0.428455
315	0.394541	0.428455	0.460976	0.373171	1	0	1	0.704878

Pelatihan menggunakan sistem yang telah dibangun terhadap 3 model 7-6-1, 7-7-1, dan 7-8-1 menghasilkan data galat yang dapat dilihat pada Tabel 5.

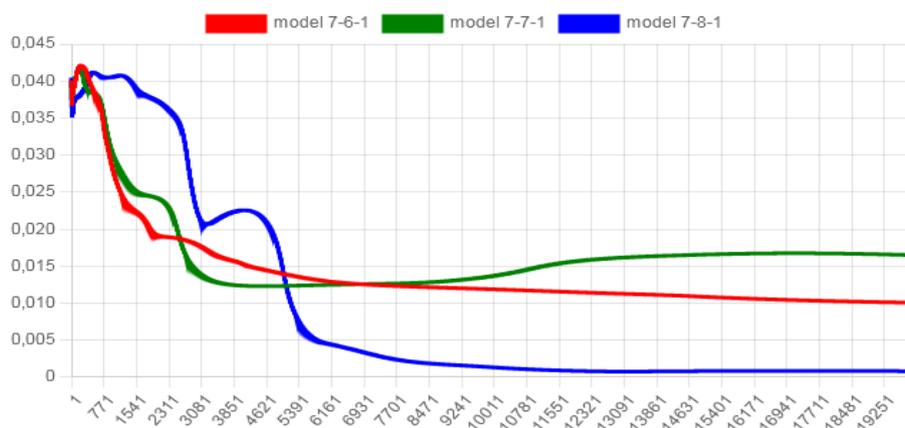
Tabel 4. Hasil galat terhadap jumlah unit tersembunyi dan parameter

Lapisan Tersembunyi	Epoch	Laju Pelatihan (y)					
		0.05	0.1	0.5	1	2	3
6	5000	0.042712	0.041565	0.040263	0.038085	0.031528	0.014021
	10000	0.041595	0.037692	0.040306	0.039083	0.030469	0.011878
	20000	0.0342	0.038166	0.038468	0.037117	0.03006	<b>0.010066</b>
7	5000	0.042814	0.041884	0.038988	0.035164	0.030515	<b>0.012328</b>
	10000	0.04193	0.038589	0.040834	0.027004	0.02982	0.013684
	20000	0.038539	0.04077	0.038231	0.017827	0.029876	0.016485
8	5000	0.042849	0.041909	0.038914	0.032701	0.014911	0.004642
	10000	0.042083	0.038971	0.041313	0.018696	0.001338	0.002256
	20000	0.037154	0.041708	0.04385	0.010779	<b>0.000817</b>	0.00306

Dari Tabel 5 dapat dilihat bahwa ketiga model memiliki galat terkecil terjadi pada model 7-6-1 dengan laju pelatihan 3 dan epoch 20000, model 7-7-1 dengan laju pelatihan 3 dan epoch 5000, model 7-8-1 dengan laju pelatihan 2 dan epoch 20000, perbandingan pergerakan galat terhadap laju pelatihan dapat dilihat pada Gambar 9. Dan dari ketiga model dengan memiliki tingkat galat terkecil dapat dilihat perbandingannya pada Gambar 10.



Gambar 9. Perbandingan galat terhadap laju pelatihan



Gambar 10. Perbandingan galat terkecil pada ketiga model

### 4.3. Proses Prediksi

Pada proses pelatihan yang telah dilakukan pada ketiga model terjadi galat terkecil pada model 7-6-1 dengan laju pelatihan 3 dan *epoch* 20000, model 7-7-1 dengan laju pelatihan 3 dan *epoch* 5000, model 7-8-1 dengan laju pelatihan 2 dan *epoch* 20000 akan dilakukan proses prediksi pada model tersebut. Proses prediksi dilakukan tahap umpan maju dengan hasil bobot pada proses pelatihan pada ketiga model. Proses prediksi dilakukan dengan menggunakan data uji sebanyak 35 data, pada urutan pertama adalah data ke-323 pada aktual penjualan yang dinormalisasi.

Tabel 5. Data uji yang dinormalisasi

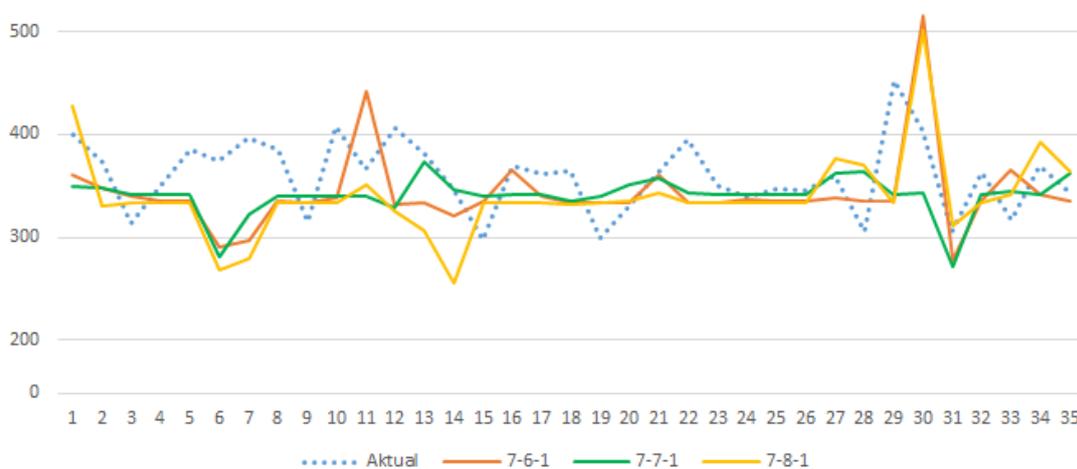
No	X1	X2	X3	X4	X5	X6	X7
1	0.45029	0.704878	0.428455	0.460976	1	0	0
2	0.474913	0.626829	0.704878	0.428455	1	0	0
3	0.486527	0.535772	0.626829	0.704878	1	0	0
4	0.496748	0.347154	0.535772	0.626829	1	0	0
...	...	...	...	...	...	...	...
34	0.492102	0.356911	0.506504	0.321138	1	0	1
35	0.496748	0.529268	0.356911	0.506504	1	0	1

Data uji dihitung dengan sistem dan hasil yang didapatkan dapat dilihat pada Tabel 7 untuk model 7-6-1(A), model 7-7-1(B) dan model 7-8-1(C), kolom prediksi adalah nilai dari proses de-normalisasi dari nilai yang dihasilkan perhitungan *neural network*, kolom selisih adalah nilai selisih dari prediksi dan aktual, kolom persentase kesalahan adalah nilai persentase dari nilai selisih terhadap nilai aktual.

Tabel 6. Perbandingan hasil prediksi dan data uji aktual

No	Aktual (y)	Prediksi (y')			Selisih			Persentase Kesalahan (e) %		
		A	B	C	A	B	C	A	B	C
1	400	360	349	427	40	51	27	10	12.8	6.8
2	372	348	348	330	24	24	42	6.5	6.5	11.3
3	314	340	341	334	26	27	20	8.3	8.6	6.4
4	351	335	341	334	16	10	17	4.6	2.8	4.8
5	386	335	342	334	51	44	52	13.2	11.4	13.5
6	374	290	281	268	84	93	106	22.5	24.9	28.3
7	396	297	323	280	99	73	116	25	18.4	29.3
8	385	335	340	333	50	45	52	13	11.7	13.5
9	315	334	340	333	19	25	18	6	7.9	5.7
10	408	338	340	334	70	68	74	17.2	16.7	18.1
11	366	442	340	352	76	26	14	20.8	7.1	3.8
12	406	332	329	325	74	77	81	18.2	19	20
13	381	333	373	306	48	8	75	12.6	2.1	19.7

No	Aktual (y)	Prediksi (y')			Selisih			Persentase Kesalahan (e) %		
		A	B	C	A	B	C	A	B	C
14	347	321	346	256	26	1	91	7.5	0.3	26.2
15	298	335	340	334	37	42	36	12.4	14.1	12.1
16	370	366	342	334	4	28	36	1.1	7.6	9.7
17	362	340	341	334	22	21	28	6.1	5.8	7.7
18	364	333	336	332	31	28	32	8.5	7.7	8.8
19	299	334	340	334	35	41	35	11.7	13.7	11.7
20	332	334	351	336	2	19	4	0.6	5.7	1.2
21	364	361	357	344	3	7	20	0.8	1.9	5.5
22	395	334	344	334	61	51	61	15.4	12.9	15.4
23	350	334	342	333	16	8	17	4.6	2.3	4.9
24	339	337	341	334	2	2	5	0.6	0.6	1.5
25	347	336	342	334	11	5	13	3.2	1.4	3.7
26	345	335	342	334	10	3	11	2.9	0.9	3.2
27	360	338	363	376	22	3	16	6.1	0.8	4.4
28	306	336	364	370	30	58	64	9.8	19	20.9
29	453	335	341	334	118	112	119	26	24.7	26.3
30	405	515	343	501	110	62	96	27.2	15.3	23.7
31	306	278	271	311	28	35	5	9.2	11.4	1.6
32	363	335	341	334	28	22	29	7.7	6.1	8
33	317	366	345	342	49	28	25	15.5	8.8	7.9
34	370	341	342	393	29	28	23	7.8	7.6	6.2
35	344	335	362	364	9	18	20	2.6	5.2	5.8



Gambar 11. Grafik perbandingan aktual dan prediksi pada ketiga model

Dari prediksi pada semua model dapat dihitung nilai *Mean Square Error* (MSE) dan rerata persentase kesalahan pada pengujian terlihat pada Tabel 8, dimana m adalah banyaknya data uji yang dilakukan.

Tabel 7. Perbandingan hasil prediksi

Model	MSE $\frac{1}{m} \sum_{t=1}^m (y_t - y'_t)^2$	Rerata Persentase Kesalahan $\frac{1}{m} \sum_{t=1}^m e_t$	Akurasi
7-6-1	$\frac{1}{35} * 85128 = 2432.2$	$\frac{1}{35} * 365.2 = 10.4\%$	89.6%
7-7-1	$\frac{1}{35} * 65793 = 1879.8$	$\frac{1}{35} * 323.7 = 9.2\%$	90.8%
7-8-1	$\frac{1}{35} * 100260 = 2864.57$	$\frac{1}{35} * 397.6 = 11.4\%$	88.6%

## 5. Simpulan

Penelitian yang berjudul Implementasi *Neural Network* Untuk Prediksi Penjualan Produk Studi Kasus Siomay Mirza dapat disimpulkan bahwa dari hasil evaluasi didapat nilai kesalahan terkecil pada model 7-7-1 dengan nilai MSE 1879.8 dan didapat tingkat akurasi prediksi 90.8%, dimana model ini yang akan dijadikan basis perhitungan prediksi pada sistem yang dibangun. Rekomendasi masa mendatang berupa ujicoba untuk menambah atau menyesuaikan periode yang berbeda atau variabel masukan, contohnya cuaca yang akan terjadi yang mungkin mempengaruhi tingkat penjualan produk tertentu.

## Daftar Referensi

- [1] R. Zudaruddin, *Forecasting untuk kegiatan ekonomi dan bisnis*. Samarinda: RV Pustaka Horizon, 2019.
- [2] A. Sreeram and S. Carraher, "Sales Forecast In Industries Using Artificial Neural Networks," p. 12, 2018.
- [3] R. M. Cantón Croda, D. E. Gibaja Romero, and S. O. Caballero Morales, "Sales Prediction through Neural Networks for a Small Dataset," *IJIMAI*, vol. 5, no. 4, p. 35, 2019, doi: 10.9781/ijimai.2018.04.003.
- [4] M. - and D. L. Warganegara, "Implementation of Artificial Neural Network in Forecasting Sales Volume in Tokopedia Indonesia," *IJACSA*, vol. 12, no. 5, 2021, doi: 10.14569/IJACSA.2021.0120551.
- [5] M. Alborzi, "Online Sales Forecasting of E-Products of Online Store Using the Artificial Neural Network Approach and Customer Sentiment Analysis," *Science Arena Publications Specialty Journal of Knowledge Management*, vol. 3, no. 1, pp. 1-6, 2018.
- [6] C. Fauzi and A. Dzulfikar, "Implementation of Product Sales Forecast Using Artificial Neural Network Method," *IJISTECH*, vol. 5, no. 2, p. 153, Aug. 2021, doi: 10.30645/ijistech.v5i2.126.
- [7] B. Tiara, "Prediksi Penjualan Menggunakan Algoritma Neural Network: Studi Kasus di PT. Balaraja Food Makmur Abadi," vol. 6, no. 1, p. 7, 2018.
- [8] S. Setti, I. A. R. Simbolon, M. Syafiq, and I. Parlina, "Penerapan Jaringan Syaraf Tiruan dalam Memprediksi Jumlah Ekspor Minyak Bumi di Indonesia," *JITE*, vol. 2, no. 1, p. 31038, Jul. 2018, doi: 10.31289/jite.v2i1.1658.
- [9] J. Devillers, *Neural networks in QSAR and drug design*. London San Diego: Academic Press, 1996.
- [10] D. Suhartono, "Dasar Pemahaman Neural Network," *Dasar Pemahaman Neural Network*, Jul. 26, 2012. <https://socs.binus.ac.id/2012/07/26/konsep-neural-network/> (accessed Oct. 08, 2021).
- [11] F. Gaxiola, P. Melin, and F. Valdez, *New Backpropagation Algorithm with Type-2 Fuzzy Weights for Neural Networks*. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-34087-6.
- [12] K. Patan, *Robust and Fault-Tolerant Control: Neural-Network-Based Solutions*, vol. 197. Cham: Springer International Publishing, 2019. doi: 10.1007/978-3-030-11869-3.
- [13] P. Sibi, S. A. Jones, and P. Siddarth, "Analysis of Different Activation Functions Using Back Propagation Neural Networks", vol. 47, p. 5, 2013.
- [14] S. Sharma, S. Sharma, U. Scholar, and A. Athaiya, "Activation Functions In Neural Networks," vol. 4, no. 12, p. 7, 2020.
- [15] Widodo, *Sistem Neuro Fuzzy untuk Pengolahan Informasi, Pemodelan, dan Kendali*. Yogyakarta: Graha Ilmu, 2005.
- [16] H. Junaedi, H. Budianto, and I. Maryati, "Data Transformation Pada Data Mining," p. 7, 2011.
- [17] R. Rojas, "The Backpropagation Algorithm," in *Neural Networks*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 149–182. doi: 10.1007/978-3-642-61068-4\_7.
- [18] M. Negnevitsky, *Artificial intelligence: a guide to intelligent systems*, 2nd ed. Harlow, England ; New York: Addison-Wesley, 2005.