



Penjadwalan Matakuliah Dengan Menggunakan Metode Jaringan Neural Network (Studi Kasus IIB Darmajaya)

Aldo Gilar Visitama*¹, Hary Sabita ²

^{1,2} Fakultas Ilmu Komputer, Informatics & Business Institute Darmajaya Jl. 2.A. Pagar Alam No. 93, Bandar Lampung - Indonesia 35142 Telp. (0721) 787214 Fax. (0721) 700261

*Email Penulis Korespondensi: aldo25321@gmail.com

Abstrak

Penjadwalan matakuliah merupakan tugas kompleks yang dihadapi oleh lembaga pendidikan dengan tantangan seperti peningkatan jumlah mahasiswa dan keterbatasan ruang kuliah. Dalam konteks ini, pemanfaatan teknologi seperti Jaringan Syaraf Tiruan (JST) menjadi relevan. Penelitian ini bertujuan untuk mengimplementasikan JST dengan metode Backpropagation untuk mengoptimalkan penjadwalan matakuliah. Penelitian ini dilakukan dengan menggunakan metode machine learning life cycle yang dimulai dari proses preprocessing yang mencakup pengumpulan, pembersihan, dan pengolahan data menjadi format yang sesuai. Selanjutnya, dilakukan proses pembuatan dan pelatihan model JST, dengan arsitektur model yang terdiri dari 11 input data, 2 hidden layer, dan menghasilkan 1 output data. Evaluasi model dilakukan dari berbagai aspek, seperti epochs, waktu pelatihan, performa, gradient, dan mu. Model juga diuji terhadap data baru sebanyak 364 data, dan hasil menunjukkan bahwa model mampu memprediksi data dengan benar. Dari keseluruhan kinerja model, diperoleh hasil akhir berupa MSE sebesar 0.000461 dan RMSE sebesar 0.021470. Hal ini menunjukkan bahwa model ini mampu melakukan prediksi dengan akurat dan memiliki tingkat kesalahan yang rendah.

Kata kunci— Penjadwalan, JNN, Backpropagation, Artificial Intelligence, Deep Learning

Abstract

Course scheduling is a complex task faced by educational institutions with challenges such as increasing student numbers and limited lecture space. In this context, the use of technology such as Artificial Neural Networks (ANN) becomes relevant. This research aims to implement ANN with the Backpropagation method to optimize course scheduling. This research was carried out using the machine learning life cycle method which starts from the preprocessing process which includes collecting, cleaning and processing data into an appropriate format. Next, the process of creating and training an ANN model is carried out, with a model architecture consisting of 11 data inputs, 2 hidden layers, and produces 1 data output. Model evaluation is carried out from various aspects, such as epochs, training time, performance, gradient, and mu. The model was also tested on 364 new data, and the results showed that the model was able to

predict the data correctly. From the overall performance of the model, the final results were obtained in the form of an MSE of 0.000461 and an RMSE of 0.021470. This shows that this model is able to make predictions accurately and has a low error rate.

Keywords— *Scheduling, ANN, Backpropagation, Artificial Intelligence, Deep Learning*

1. PENDAHULUAN

Institut Informatika dan Bisnis Darmajaya adalah perguruan tinggi swasta di Bandar Lampung yang didirikan pada tahun 1995. Terletak di Jl. Zainal Abidin Pagar Alam, No.93B Labuhan Ratu, Bandar Lampung, institut ini dimiliki oleh Yayasan Alfian Husin. Saat ini, institut ini menawarkan 15 program studi, termasuk Akuntansi, Bisnis Digital, Desain Interior, Desain Komunikasi Visual, Hukum Bisnis, Manajemen, Pariwisata, Pendidikan Teknologi Informasi, Sains Data, Sistem Informasi, Sistem Komputer, Teknik Informatika (tingkat S1), serta Manajemen, Manajemen Teknologi, dan Teknik Informatika (tingkat S2). Perguruan tinggi memiliki peran kunci dalam mengembangkan sumber daya manusia berkualitas [1]. Di era globalisasi ini, inovasi dalam layanan pendidikan menjadi sangat penting. Penjadwalan matakuliah yang efisien dan efektif sangat berkontribusi pada peningkatan kualitas pendidikan, optimalisasi sumber daya, dan kenyamanan mahasiswa [2], [3].

Penjadwalan matakuliah di IIB Darmajaya dan perguruan tinggi lainnya merupakan tugas yang rumit. Dalam menghadapi tantangan seperti peningkatan jumlah mahasiswa, ruang kelas yang terbatas, preferensi dosen, dan aturan administrasi, perencanaan jadwal memerlukan perhatian khusus. Penggunaan teknologi seperti Jaringan *Neural Network* (JNN) dan algoritma *Backpropagation* dapat membantu dalam mengatasi masalah ini. JNN adalah teknik kecerdasan buatan yang dapat memecahkan masalah kompleks, sedangkan *Backpropagation* adalah algoritma yang digunakan dalam JNN untuk meningkatkan kinerja jaringan [4], [5]. Metode ini memungkinkan jaringan untuk mempelajari pola-pola data historis, mengidentifikasi tren, dan merespons perubahan dalam kebutuhan mahasiswa dan dosen, sehingga membantu dalam penjadwalan matakuliah dengan lebih baik.

Beberapa penelitian sebelumnya telah menggunakan JNN dengan metode *Backpropagation* dalam berbagai konteks. Sebagai contoh, [6] mengaplikasikan JNN untuk memprediksi hasil panen gabah padi dengan arsitektur yang berbeda, mencapai akurasi hingga 92.90%. [7] menggunakan JNN untuk memprediksi jumlah mahasiswa baru, menemukan arsitektur optimal dengan jumlah input 12, neuron tersembunyi 8, dan output 1, menghasilkan prediksi sebanyak 115 mahasiswa., sementara [8] menerapkan JNN untuk memprediksi beban listrik dengan tingkat kesalahan yang rendah, dengan *MSE* 0.0037377 dan *MAPE* yang tetap dalam toleransi, dan [9] menggunakan JNN untuk memprediksi penjualan keripik, menunjukkan nilai *MSE* sebesar 0.14181 untuk keripik singkong dan 0.22071 untuk keripik pisang. Dalam konteks pariwisata, [10] memprediksi kunjungan wisata Kota Payakumbuh dengan JNN *Backpropagation*, mendapatkan *MSE* sebesar 0.00991 dengan pola 4-27-1. Hasil penelitian menunjukkan bahwa metode ini memiliki potensi dalam berbagai aplikasi, termasuk prediksi hasil panen, jumlah mahasiswa baru, beban listrik, penjualan produk, dan kunjungan wisata.

Metode JNN dengan algoritma *Backpropagation* dipilih untuk penelitian ini karena keunggulan dalam mempelajari pola kompleks dan menangkap hubungan non-linear dalam data, sesuai dengan kompleksitas penjadwalan matakuliah. Dibandingkan dengan metode lain seperti algoritma genetika, JNN menawarkan fleksibilitas dan kemampuan adaptasi yang lebih tinggi, serta mampu menghasilkan solusi yang lebih baik melalui optimisasi dengan *Backpropagation*. Dalam konteks penjadwalan matakuliah yang melibatkan banyak variabel dan kompleksitas, JNN dapat menghasilkan prediksi yang akurat dan solusi penjadwalan yang efisien berdasarkan data historis dan dinamika lingkungan. Oleh karena itu, penggunaan JNN dengan *Backpropagation* memiliki potensi untuk memberikan solusi yang efektif dalam penjadwalan matakuliah.

Penelitian yang dilakukan memiliki perbedaan dengan penelitian-penelitian sebelumnya yang telah menggunakan JNN dengan metode *Backpropagation* dalam berbagai konteks seperti prediksi hasil panen, jumlah mahasiswa baru, beban listrik, penjualan produk, dan kunjungan wisata. Fokus penelitian ini adalah pada pembuatan model penjadwalan mata kuliah di PLPP IIB Darmajaya, dengan menerapkan tahapan-tahapan dalam *Machine Learning Life Cycle* secara khusus untuk membangun model tersebut. Data yang digunakan adalah data historis penjadwalan mata kuliah yang spesifik untuk PLPP IIB Darmajaya dengan variabel *input* dan *output* yang berbeda dengan jenis data yang digunakan dalam penelitian-penelitian sebelumnya. Penggunaan teknik *SMOTE* untuk menangani ketidakseimbangan variabel output Evaluasi kinerja model dilakukan menggunakan *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)* untuk mengevaluasi prediksi model penjadwalan mata kuliah. Dengan demikian, pendekatan, konteks, tujuan, dan evaluasi model dalam penelitian ini mengalami perbedaan atau kebaruan dengan penelitian-penelitian sebelumnya.

Penelitian ini berpotensi untuk meningkatkan sistem penjadwalan mata kuliah di Institut Informatika dan Bisnis Darmajaya. Dengan proses penjadwalan yang lebih efisien dan optimal, diharapkan dapat meningkatkan efektivitas dan menghemat sumber daya. Penelitian tentang "Penjadwalan Mata Kuliah dengan Menggunakan Metode Jaringan *Neural Network* (Studi Kasus IIB Darmajaya)" menjadi penting untuk membantu meningkatkan efisiensi dan kualitas perguruan tinggi di sana. Model Jaringan *Neural Network* dengan Algoritma yang dibangun bertujuan untuk memprediksi penjadwalan mata kuliah di Institut Informatika dan Bisnis Darmajaya.

2. METODE PENELITIAN

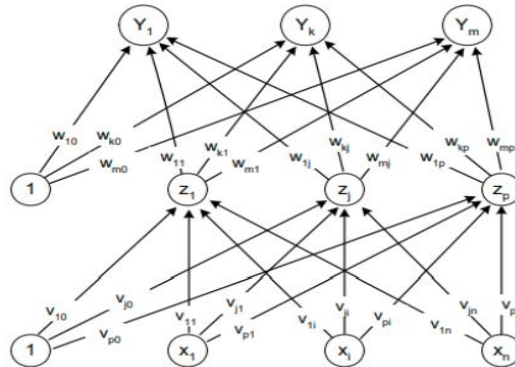
2.1 Artificial Intelligence

Kecerdasan buatan (*AI*) merupakan program yang memungkinkan komputer berperilaku seperti manusia, memberikan masukan atau hasil yang mirip dengan kecerdasan manusia berdasarkan data yang diberikan. *AI* terbagi menjadi dua bagian utama: *Machine Learning*, yang memungkinkan komputer membuat keputusan tanpa perlu program yang kompleks, dan *Deep Learning*, di mana komputer belajar menggunakan jaringan saraf seperti otak manusia, memproses berbagai jenis data seperti gambar, teks, video, dan suara dengan performa optimal yang dimungkinkan oleh *GPU*. Teknologi ini telah menunjukkan kemajuan pesat dalam bidang seperti pengenalan suara, gambar, penerjemahan bahasa, dan sistem cerdas lainnya [11].

Di dalam *Artificial Intelligence* terdapat sebuah metode pelatihan Jaringan *Neural Network* atau *Artificial Neural Network (ANN)* adalah cara untuk membuat komputer beroperasi seperti otak manusia, dengan tujuan utama untuk melakukan fungsi seperti pemecahan masalah dan pembelajaran [12]. *ANN* memiliki beberapa jenis [13], termasuk *Feedforward Neural Network* yang mengalirkan data dari input ke output tanpa memberikan umpan balik, *Radial Basis Function Neural Network* yang menghitung jarak dari titik pusat, *Recurrent Neural Network (RNN)* yang menggunakan output sebelumnya sebagai *input*, *Convolutional Neural Network (CNN)* yang cocok untuk pemrosesan gambar, dan *Modular Neural Network (MNN)* yang terdiri dari beberapa jaringan yang bekerja secara independen. Masing-masing jenis *ANN* memiliki kelebihan dan kegunaan tersendiri dalam pemrosesan informasi dan komputasi.

Selain itu terdapat Algoritma yang digunakan dalam pelatihan JNN yang disebut dengan algoritma *Backpropagation* yaitu metode pembelajaran yang digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot yang terhubung dengan neuron-neuron pada lapisan tersembunyi. Ini adalah metode yang baik untuk menangani masalah pengenalan pola-pola kompleks, seperti yang digunakan dalam berbagai aplikasi seperti pengenalan pola, peramalan, dan optimisasi [14], [15]. Metode ini melibatkan pembelajaran yang terbimbing, di mana pola masukan dan target diberikan sebagai data pasangan. Bobot awal dilatih melalui tahap maju dan kemudian disesuaikan melalui tahap mundur untuk memperkecil galat sehingga target keluaran

yang diinginkan tercapai [16]. Gambar 1 adalah arsitektur yang dimiliki oleh algoritma *backpropagation*.



Gambar 1 Arsitektur *Backpropagation*

Pada Gambar 1 v_{ji} merupakan bobot garis dari unit masukan x_i ke unit layar tersembunyi z_j atau v_{j0} merupakan bobot garis yang menghubungkan padanan lainnya di unit masukan ke unit layar tersembunyi z_j . w_{kj} merupakan bobot dari unit layar tersembunyi z_j ke unit keluaran y_k atau w_{k0} merupakan bobot dari bias di layar tersembunyi ke unit keluaran z_k .

Backpropagation adalah metode yang digunakan untuk meminimalkan kesalahan kuadrat pada keluaran jaringan [17] dengan melatih jaringan syaraf tiruan dengan langkah-langkah utama [18]: perambatan maju, perambatan balik, dan penyesuaian bobot dan bias.

- Inisialisasi bobot dan bias. Bobot dan bias dapat di-inisialisasi dengan sembarang angka (acak) dan biasanya terletak antara 0, 1 ataupun -1.
- Selama kondisi berhenti bernilai salah, kerjakan
- Tahap 1 : Perambatan Maju (*Forward propagation*)
 - Setiap unit masukan ($x_i, i = 1, 2, 3, \dots, n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan tersembunyi
 - Setiap unit tersembunyi ($z_j, j = 1, 2, 3, \dots, n$) menjumlahkan bobot sinyal masukan dengan persamaan berikut:

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (1)$$

Dan menerapkan fungsi aktivasi untuk menghitung sinyal keluarannya:

$$z_j = f(z_in_j) \quad (2)$$

- Setiap unit keluaran ($y_ink = 1, 2, 3, \dots, n$) menjumlahkan bobot sinyal masukan dengan persamaan berikut:

$$y_in_k = w_{0k} + \sum_{i=1}^n z_i w_{jk} \quad (3)$$

Dan menerapkan fungsi aktivasi untuk menghitung sinyal keluarannya:

$$y_k = f(y_in_k) \quad (4)$$

Kemudian mengirimkan sinyal tersebut ke semua unit keluaran

- Tahap 2 : Perambatan Balik (*Backpropagation*)

- Setiap *unit* keluaran ($y_k, k=1,2,3, \dots n$) menerima pola target yang sesuai dengan pola masukan pelatihan, kemudian hitung galat dengan persamaan berikut:

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

F' adalah turunan dari fungsi aktivasi kemudian menghitung koreksi bobot dengan persamaan berikut:

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

Menghitung pula koreksi bias dengan persamaan berikut:

$$\Delta w_{0k} = \alpha \delta_k \quad (7)$$

- Setiap unit tersembunyi ($z_j, j=1,2,3 \dots n$) menjumlahkan delta masukannya:

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \quad (8)$$

Untuk menghitung informasi *error* atau galat, kalikan nilai ini dengan turunan dari fungsi aktivasinya:

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad (9)$$

Kemudian menghitung koreksi bobot dengan persamaan sebagai berikut:

$$\Delta v_{ij} = \alpha \delta_i x_j \quad (10)$$

- Tahap 3 : Perubahan Bobot dan Bias
 - Setiap *unit* keluaran ($y_k, k=1, 2, 3, \dots n$) dilakukan perubahan bobot ($j=0, 1, 2, \dots n$) dengan persamaan sebagai berikut:

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

Setiap *unit* tersembunyi ($z_j, j=1, 2, 3, \dots n$) dilakukan perubahan bobot ($i=0, 1, 2, \dots n$) dengan persamaan sebagai berikut:

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (12)$$

- Tes kondisi berhenti
Terdapat tiga tahap dalam jaringan saraf tiruan: tahap pelatihan jaringan, tahap pengujian jaringan dan tahap prediksi

Selain itu terdapat metode penelitian yang digunakan di dalam penelitian ini yaitu dengan menggunakan metode *machine learning life cycle* [19] adalah metode pengembangan arsitektur model jaringan syaraf tiruan seperti pada pemaparan berikut ini:

2. 2 Metode Machine Learning Life Cycle

Berikut adalah tahapan yang dilakukan di dalam penelitian ini:

2. 2.1 Model Requirements

Pada tahap ini, dilakukan analisis dan observasi kebutuhan model dengan mengumpulkan data yang diperlukan untuk membuat model penjadwalan mata kuliah di PLPP IIB Darmajaya.

2. 2.2 Data Collection

Pada tahap ini, data yang dikumpulkan merupakan data historis penjadwalan mata kuliah yang telah digunakan sebelumnya oleh PLPP IIB Darmajaya.

2. 2.3 Data Cleaning

Pada tahap ini, data yang telah dikumpulkan dibersihkan dengan menghapus data duplikat, data kosong, dan variabel yang tidak relevan untuk meningkatkan kinerja pelatihan model. Variabel baru juga ditambahkan untuk memudahkan proses pembuatan dan pelatihan model sesuai dengan tujuan penelitian.

2. 2.4 Data Labeling

Proses *Labeling* dilakukan dengan memberikan label pada setiap *dataset*. Sehingga pada tahap ini yang dilakukan adalah sesuai dengantahap sebelumnya yaitu dengan menambah variable baru “Tipe Kelas” dimana dalam variable ini memiliki dua atribut yaitu “kelas reguler dan kelas ekstensi” yang kemudian akan diubah menjadi label dalam bentuk angka atau numerik yaitu 0 dan 1.

2. 2.5 Feature Engineering

Proses *Feature Engineering* dimulai dengan membagi data menjadi data *training* (80% dari total) dan data *testing* (20% dari total). Sebelum pembagian, dilakukan standarisasi data menggunakan teknik *MinMaxScaling*, yang mengubah semua data menjadi bentuk numerik antara 0.1 hingga 0.9. Setelah itu, teknik *SMOTE* diterapkan untuk menangani ketidakseimbangan kelas dalam dataset, sehingga kedua kelas menjadi seimbang. Hal ini bertujuan untuk menghindari *overfitting* saat melatih model dan meningkatkan kualitas *dataset*.

2. 2.6 Model Training

Model *training* dilakukan untuk melatih model jaringan syaraf tiruan yang sudah dibuat. *Training* di lakukan agar model dapat bekerja dengan optimal dimana model dilatih berdasarkan arsitektur jaringan syaraf tiruan dengan menggunakan algoritma *backpropagation*.

2. 2.7 Model Evaluation

Pada tahap evaluasi model, dilakukan konfirmasi apakah model dapat memprediksi data baru dengan baik. Data yang digunakan dalam evaluasi ini adalah data *testing*, yang merupakan 20% dari seluruh *dataset*. Setelah model memprediksi data, langkah selanjutnya adalah menggunakan dua jenis metrik evaluasi, yaitu *Mean Squared Error (MSE)* dan *Root Mean Squared Error (RMSE)*, untuk mengevaluasi kinerja model yang telah dibuat.

3. HASIL DAN PEMBAHASAN

3.1 Preprocessing Data

Langkah yang dilakukan dalam tahap *preprocessing* data meliputi data *collecting*, *data cleaning*, *data labeling* dan *Feature Engineering*, membagi data ke dalam data uji dan data latih pada pemaparan berikut ini.

Data dikumpulkan dari jadwal matakuliah sebelumnya yang disusun oleh PLPP IIB Darmajaya. Ini mencakup jadwal untuk semester ganjil dan genap tahun akademik 2021/2022, termasuk variabel seperti hari, waktu, kode mata kuliah, nama mata kuliah, jumlah SKS, program studi, ruang, kelas, tipe, jumlah peserta, dan informasi dosen. Sebanyak 916 jadwal untuk semester ganjil dan 948 jadwal untuk semester genap telah terdokumentasi, dengan total 1.864 jadwal matakuliah.

Setelah mengumpulkan data kemudian dilakukan tahap pembersihan data untuk memastikan kualitas *dataset* penelitian. Pertama, dilakukan penyesuaian atribut untuk memastikan konsistensi data seperti pada variabel "PRODI". Kemudian, variabel "Peserta" yang tidak relevan dihapus, menyisakan variabel yang esensial. Selanjutnya, data yang tidak lengkap atau memiliki nilai *null* pada variabel "Ruang" dan "Dosen Mr/Mrs" dihapus. Variabel "Waktu" dipisahkan menjadi "Waktu Mulai" dan "Waktu Selesai" untuk memudahkan analisis. Dengan ini, *dataset* menjadi lebih terstruktur dan siap untuk pelatihan model jaringan syaraf tiruan. Langkah berikutnya adalah memberikan label pada setiap data.

Proses selanjutnya adalah melakukan pelabelan data yang bertujuan memberikan kelas pada setiap entri dalam *dataset*. Kelas dibagi menjadi dua: "Kelas Reguler" dan "Kelas Ekstensi", berdasarkan waktu mulai dan selesai. Kelas "Reguler" untuk jadwal antara jam 7:10 pagi hingga 16:10 siang, sementara "Kelas Ekstensi" untuk jadwal antara jam 16:30 sore hingga 20:00 malam.

Kemudian pada tahap *feature engineering* dilakukan untuk mempersiapkan *dataset* sebelum dilakukan proses pelatihan dan pengujian model jaringan syaraf tiruan. Tahap ini melibatkan beberapa langkah, termasuk transformasi data menjadi format numerik, normalisasi menggunakan *MinMaxScaler*, pembagian *dataset* menjadi data latih dan data uji, serta implementasi teknik *SMOTE* untuk menangani ketidakseimbangan kelas. Pertama, data yang awalnya dalam format kategorikal diubah menjadi format numerik untuk memudahkan proses analisis. Kemudian, nilai-nilai dalam *dataset* dinormalisasi menggunakan *MinMaxScaler*, memastikan konsistensi skala untuk setiap variabel. Setelah itu, *dataset* dibagi menjadi data latih (80%) dan data uji (20%) untuk melatih dan menguji model. Terakhir, teknik *SMOTE* diterapkan untuk menangani ketidakseimbangan kelas dengan meningkatkan jumlah sampel pada kelas minoritas. Dengan langkah-langkah tersebut, *dataset* siap digunakan untuk pelatihan dan pengujian model jaringan syaraf tiruan.

3. 2 Pembuatan dan Pelatihan Arsitektur Jaringan Syaraf Tiruan

Dalam pembuatan arsitektur model JNN, langkah-langkahnya dapat diuraikan sebagai berikut. Pertama, jaringan dibuat dengan menggunakan fungsi *feedforwardnet* dan memiliki 2 *unit* pada lapisan tersembunyi. Selanjutnya, aktivasi lapisan tersembunyi diatur menggunakan fungsi *sigmoid* ('*logsig*'), yang menentukan respons *neuron* terhadap input dan menghasilkan *output*. Parameter pelatihan ditetapkan, termasuk algoritma pelatihan (*Levenberg-Marquardt backpropagation*), jumlah iterasi (1000 kali), target tingkat kesalahan (0.0001), dan tingkat pembelajaran (0.1). Evaluasi kinerja jaringan menggunakan *Mean Squared Error (MSE)*. Selanjutnya, jaringan diinisialisasi dan dilatih dengan data latih menggunakan fungsi *train*, yang mengubah bobot dan bias jaringan iteratif berdasarkan data latih untuk mengurangi kesalahan prediksi. Arsitektur model ini memiliki 11 *input*, 2 *hidden layer*, dan 1 *output*.

3. 3 Model Evaluation

Setelah model di lakukan ditarining maka selanjutnya akan di evaluasi untuk mengetahui kinerja dari model apakah memiliki kinerja yang abik atau buruk yang akan di jelaskan sebagai berikut:

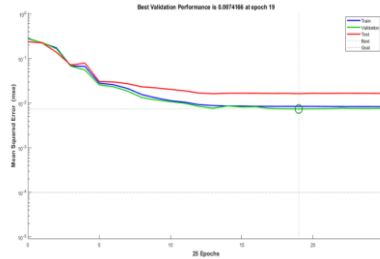
3. 3.1 Epochs and Waktu Pelatihan

Dalam penelitian ini, model dilatih menggunakan algoritma *backpropagation* atau yang disebut dengan *Levenberg-Marquardt*. Tujuannya adalah mencapai kinerja yang diinginkan.

Proses pelatihan dilakukan selama 25 *epoch* dari target total 1000 *epochs*, dengan waktu pelatihan sekitar 2 detik.

3. 3.2 Performance

Performa jaringan dievaluasi menggunakan *Mean Squared Error (MSE)* pada Gambar 2.

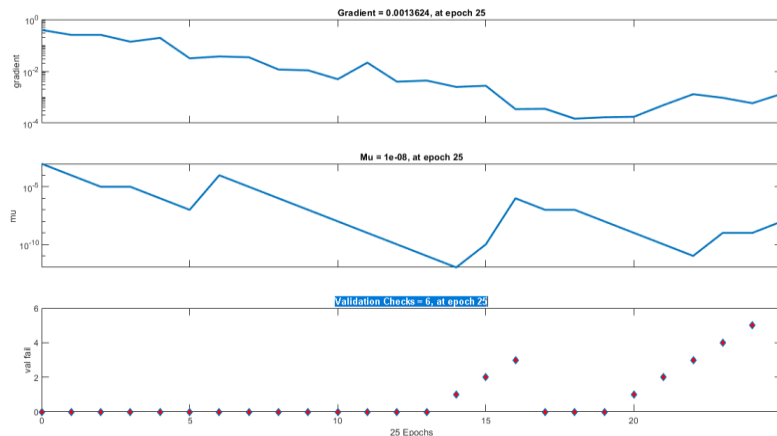


Gambar 2 Evaluasi *Performance*

Awalnya, performa jaringan adalah 0.283, namun turun menjadi 0.00837 selama pelatihan. Target performa yang diinginkan adalah 0.0001. Performa terbaik, dengan nilai 0.0074166, terjadi pada *epoch* ke-19 saat jaringan diuji dengan data validasi yang tidak digunakan selama pelatihan. Ini menunjukkan kemampuan jaringan dalam menghasilkan prediksi akurat pada data baru yang belum pernah dilihat sebelumnya.

3. 3.3 Gradient

Gradient menggambarkan kemiringan kurva kesalahan berikut merupakan grafik *gradient* dari hasil pelatihan model yang dapat dilihat pada Gambar 3.

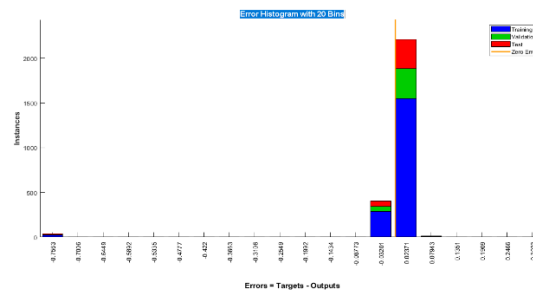


Gambar 3 Evaluasi *Gradient*

Nilai awal *gradient* adalah 0.391, turun menjadi 0.00136 selama pelatihan. Targetnya adalah mencapai nilai sekitar $1e-07$. Pada *epoch* ke-25, *gradient* mencapai 0.0013624, menunjukkan pembelajaran dalam jaringan. μ (μ) pada titik tersebut adalah $1e-08$, mengontrol laju pembelajaran. Total 6 pemeriksaan validasi dilakukan pada *epoch* ke-25 untuk memantau kinerja jaringan pada data baru.

3. 3. 4 $M\mu$

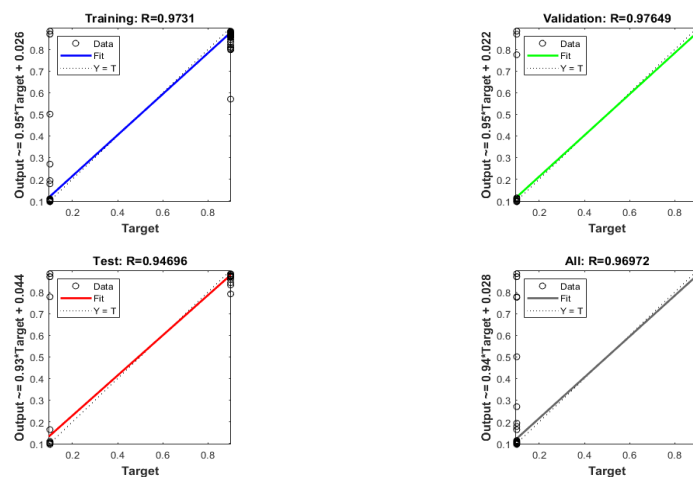
$M\mu$ (μ) adalah faktor yang mengatur laju pembelajaran dalam algoritma pelatihan grafik hasil dari laju pembelajaran atau yang disebut dengan *error history* dapat dilihat pada Gambar 4.

Gambar 4 Evaluasi μ

Nilai awal μ adalah 0.001, turun menjadi $1e-08$ selama pelatihan dengan target mencapai $1e+10$. Pengaturan yang tepat untuk μ penting dalam mengontrol perubahan bobot jaringan. Hasil evaluasi μ ditampilkan dalam *Error Histogram*, yang memperlihatkan distribusi kesalahan selama pelatihan. Rentang *instance* yang diamati adalah 0-2000, dengan kesalahan sebesar 0.02731 pada rentang tersebut. Ditemukan lebih dari 2000 *instances* dengan nilai *error* nol, menunjukkan bahwa jaringan dapat memprediksi secara tepat untuk sebagian besar *instance*.

3. 3.5 Validation

Validation mencatat 6 pemeriksaan validasi selama pelatihan pada Gambar 5.



Gambar 5 Validation

Dalam evaluasi ini, "*Output*" adalah hasil prediksi yang dihasilkan oleh jaringan, sedangkan "*Target*" adalah nilai yang diharapkan atau target sebenarnya. Berikut adalah persamaan linier yang menggambarkan hubungan antara *Output* dan *Target*, serta koefisien korelasi (R) yang menunjukkan seberapa baik *Output* menyesuaikan dengan *Target* pada setiap fase:

- *Training*:
Persamaan *Output*: $Output \text{ prediksi} \approx 0.95 * Target + 0.026$
Korelasi (R) pada data pelatihan: $R = 0.9731$
- *Validation*:
Persamaan *Output*: $Output \text{ prediksi} \approx 0.95 * Target + 0.022$
Korelasi (R) pada data validasi: $R = 0.97649$
- *Test*:
Persamaan *Output*: $Output \text{ prediksi} \approx 0.93 * Target + 0.044$
Korelasi (R) pada data pengujian: $R = 0.94696$
- *All*:

Persamaan *Output*: $Output \text{ prediksi} \approx 0.94 * Target + 0.028$
 Korelasi (R) pada semua data: $R = 0.96972$

Nilai korelasi (R) yang tinggi menunjukkan bahwa jaringan syaraf tiruan memiliki kemampuan yang baik dalam memprediksi nilai yang mendekati nilai sebenarnya (*Target*) pada setiap tahap dan pada semua data yang tersedia. Ini menunjukkan bahwa model jaringan memiliki kemampuan yang baik untuk melakukan generalisasi pada data baru yang belum pernah dilihat sebelumnya.

3. 4 Data Testing

Setelah model dilatih, dilakukan pengujian untuk memeriksa kemampuan prediksi model terhadap data baru pada Gambar 6.

```

Command window
Data ke-352: Prediksi=Kelas Reguler, Aktual=0.90
Prediksi Benar (Green)
Data ke-353: Prediksi=Kelas Reguler, Aktual=0.90
Prediksi Benar (Green)
Data ke-354: Prediksi=Kelas Reguler, Aktual=0.90
Prediksi Benar (Green)
Data ke-355: Prediksi=Kelas Reguler, Aktual=0.90
Prediksi Benar (Green)
Data ke-356: Prediksi=Kelas Reguler, Aktual=0.90
Prediksi Benar (Green)
Data ke-357: Prediksi=Kelas Ekstensi, Aktual=0.10
Prediksi Benar (Green)
Data ke-358: Prediksi=Kelas Reguler, Aktual=0.90
Prediksi Benar (Green)
Data ke-359: Prediksi=Kelas Reguler, Aktual=0.90
Prediksi Benar (Green)
Data ke-360: Prediksi=Kelas Ekstensi, Aktual=0.10
Prediksi Benar (Green)
Data ke-361: Prediksi=Kelas Ekstensi, Aktual=0.10
Prediksi Benar (Green)
Data ke-362: Prediksi=Kelas Ekstensi, Aktual=0.10
Prediksi Benar (Green)
Data ke-363: Prediksi=Kelas Reguler, Aktual=0.90
Prediksi Benar (Green)
Data ke-364: Prediksi=Kelas Reguler, Aktual=0.90
Prediksi Benar (Green)
Jumlah Prediksi Benar : 364
Jumlah Prediksi Salah : 0
Warning: In a future release, UI components will not be ir
  
```

Gambar 6 Data Testing

Hasil pengujian menunjukkan bahwa model mampu memprediksi data baru dengan benar. Dari total 364 data uji, seluruhnya diprediksi dengan tepat oleh model. Evaluasi model menggunakan data uji menghasilkan *MSE* (*Mean Squared Error*) sebesar 0.000461 dan *RMSE* (*Root Mean Squared Error*) sebesar 0.021470. Dalam kasus di mana semua prediksi benar, *MSE* akan menjadi 0, dan *RMSE* akan mendekati 0 karena tidak ada perbedaan antara prediksi dan nilai aktual. *MSE* adalah rata-rata dari kuadrat selisih antara nilai prediksi dan nilai aktual, sedangkan *RMSE* adalah akar kuadrat dari *MSE*. Oleh karena itu, nilai *MSE* dan *RMSE* yang sangat kecil atau mendekati 0 menunjukkan ketepatan model dalam memprediksi data uji.

4. KESIMPULAN

Penelitian ini menggunakan model jaringan syaraf tiruan untuk menganalisis penjadwalan mata kuliah dengan algoritma *backpropagation*. Tujuannya adalah memahami struktur penjadwalan untuk membantu dalam penyusunan jadwal. Tantangan utama adalah ketidakseimbangan data latih dan uji antara kelas reguler dan ekstensi, yang berhasil diatasi dengan teknik *SMOTE*.

Hasil penelitian menunjukkan bahwa model jaringan syaraf tiruan mampu memprediksi dengan akurat, dengan kesalahan yang rendah (*MSE* sebesar 0.000461 dan *RMSE* sebesar 0.021470). Evaluasi melibatkan tahap pelatihan, validasi, dan pengujian, yang semuanya menunjukkan korelasi tinggi antara prediksi dan nilai sebenarnya. Ini menunjukkan kemampuan model untuk melakukan generalisasi pada data baru. Kesimpulannya, model ini efektif dalam memprediksi dan generalisasi pada *dataset* yang belum pernah dilihat sebelumnya.

5. SARAN

Saran yang dapat diberikan dari penelitian ini meliputi: pertama, memperluas *dataset* atau menggunakan *dataset* yang lebih besar dan representatif untuk meningkatkan keakuratan dan generalisasi model. Kedua, melakukan perbandingan dengan beberapa jenis model atau algoritma pembelajaran mesin lainnya untuk memahami keefektifan model jaringan syaraf tiruan. Ketiga, eksplorasi lebih lanjut terhadap parameter pelatihan seperti *learning rate*, jumlah lapisan tersembunyi, jumlah *neuron*, dan jenis fungsi aktivasi untuk meningkatkan performa model. Dan terakhir, disarankan untuk menggunakan *tools* atau *software* pengembangan model seperti *Jupyter Notebook* untuk memudahkan tahap *deployment model*, mengingat beberapa keterbatasan dalam proses *deployment* menggunakan *Matlab*.

UCAPAN TERIMA KASIH

Saya mengucapkan terima kasih yang tulus kepada Allah SWT atas berkah, dan kepada orang tua saya atas dukungan mereka yang luar biasa selama proses pembuatan dan penyusunan penelitian ini. Terima kasih kepada Bapak Hary Sabita, S.Kom., M.T.I, Ibu Dr. Chairani, S.Kom., M.Eng, MTI, dan Bapak Nisar, S.Kom., M.T., atas bimbingannya. Apresiasi khusus kepada fakultas ilmu komputer di IIB Darmajaya, khususnya program studi Teknik Informatika serta terima kasih kepada Tim Redaksi Jurnal Teknik Politeknik Negeri Sriwijaya yang telah memberi memberi kesempatan, sehingga artikel ilmiah ini dapat diterbitkan.

DAFTAR PUSTAKA

- [1] L. Dihe dan Y. Wangdra, "Pendidikan Adalah Faktor Penentu Daya Saing Bangsa," no. September, hal. 84–90, 2023.
- [2] A. Rizquha dan A. Abdullah, "Trilogy and The Five Branding Awareness: Strategies to Build Public Trust in Higher Education," *J. Educ. Manag. Res.*, vol. 1, no. 1, hal. 12–23, 2022, [Daring]. Tersedia pada: <https://serambi.org/index.php/jemr/article/view/61/44>
- [3] M. Suti, M. Z. Syahdi, dan D. Didiharyono, "JEMMA (Jurnal of Economic , Management , and Accounting) Tata Kelola Perguruan Tinggi dalam Era Teknologi Informasi dan Digitalisasi," vol. 3, no. September, hal. 203–214, 2020.
- [4] A. C. Dewanto, "Risiko Dan Mitigasi Penggunaan Kecerdasan Buatan Dalam Bidang Pendidikan," *Pros. Konf. Ilm. Pendidik.*, vol. 4, no. 2018, hal. 1–10, 2023.
- [5] M. Thoriq, "Peramalan Jumlah Permintaan Produksi Menggunakan Jaringan Saraf Tiruan Algoritma Backpropagation," *J. Inf. dan Teknol.*, vol. 4, hal. 27–32, 2022, doi: 10.37034/jidt.v4i1.178.
- [6] R. Maiyuriska, "Jurnal Informatika Ekonomi Bisnis Penerapan Jaringan Syaraf Tiruan dengan Algoritma Backpropagation dalam Memprediksi Hasil Panen Gabah Padi," vol. 4, hal. 28–33, 2022, doi: 10.37034/infeb.v4i1.115.
- [7] A. F. Suahati, A. A. Nurrahman, dan O. Rukmana, "Penggunaan Jaringan Syaraf Tiruan – Backpropagation dalam Memprediksi Jumlah Mahasiswa Baru," *J. Media Tek. dan Sist. Ind.*, vol. 6, no. 1, hal. 21, 2022, doi: 10.35194/jmsti.v6i1.1589.
- [8] F. Rohman, "Prediksi Beban Listrik Dengan Menggunakan Jaringan Syaraf Tiruan Metode Backpropagation," *J. Surya Energy*, vol. 5, no. 2, hal. 55–60, 2022, doi: 10.32502/jse.v5i2.3092.
- [9] F. D. N. Fitri, A. Pandu Kusuma, dan S. Nur Budiman, "Implementasi Algoritma Backpropagation Untuk Memprediksi Penjualan Keripik (Studi Kasus : Rumah Industri Keripik Age Enak)," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 2, hal. 782–789, 2022, doi: 10.36040/jati.v6i2.5737.
- [10] N. Aulya, "Prediksi Kunjungan Wisata Kota Payakumbuh Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation," *J. Inform. Ekon. Bisnis*, vol. 4, hal. 7–9, 2022, doi: 10.37034/infeb.v4i4.157.

- [11] T. Markiewicz dan J. Zheng, *Getting Started with Artificial Intelligence*. 2019.
- [12] M. T. Suwindra, E. Ernawati, dan A. Erlansari, “Analisis Kemiripan Jenis Burung Menggunakan Siamese Neural Network,” *Rekursif J. Inform.*, vol. 9, no. 2, hal. 193–205, 2021.
- [13] A. Merchant, “Neural Networks Explained,” *Medium*, 2019. <https://medium.datadriveninvestor.com/neural-networks-explained-6e21c70d7818>
- [14] A. Zamsuri, “Backpropagation Untuk Prediksi Nilai Akreditasi Program Studi (Studi Kasus : Universitas Islam Indragiri),” *Semaster*, hal. 316–322, 2020.
- [15] M. Masrizal dan A. Hadiansa, “Prediksi Jumlah Lulusan Mahasiswa Stmik Dumai Menggunakan Jaringan Syaraf Tiruan,” *INFORMATIKA*, vol. 10, no. 1, hal. 9, 2019, doi: 10.36723/juri.v9i2.98.
- [16] W. H. Sunandi, “Klasifikasi Penjurusan Sekolah Menengah Atas Menggunakan Metode Backpropagation Berbasis Website (Studi Kasus: SMA Negeri 1 Mlonggo),” hal. 1–13, 2019.
- [17] N. L. Br Sitepu, “Jaringan Saraf Tiruan Memprediksi Nilai Pemelajaran Siswa Dengan Metode Backpropagation (Studi kasus : SMP Negeri 1 Salapian),” *J. Inf. Technol.*, vol. 1, no. 2, hal. 54–58, 2021, doi: 10.32938/jitu.v1i2.1006.
- [18] C. L. Pradipika, S. Handoko, dan B. Winardi, “Peramalan Beban Puncak Harian Pada Sistem PT PLN APB Jawa Tengah Dan DIY Menggunakan Metode Jaringan Saraf Tiruan – Algoritma Genetika,” *Transmisi*, vol. 19, no. 1, hal. 28–34, 2017.
- [19] M. Gärtler, V. Khaydarov, B. Klöpper, dan L. Urbas, “The Machine Learning Life Cycle in Chemical Operations – Status and Open Challenges,” *Chemie-Ingenieur-Technik*, vol. 93, no. 12, hal. 2063–2080, 2021, doi: 10.1002/cite.202100134.