



Implementasi *Backpropagation* Untuk Rekomendasi Jurusan Peminatan Mahasiswa Program Studi Teknik Informatika Di Universitas Islam Balitar

Muhammad Iqbal Raihan¹, Saiful Nur Budiman², Udkhiati Mawaddah³

Program Studi Teknik Informatika, Universitas Islam Balitar¹²³

Jl. Majapahit No. 2-4, Sananwetan, 66137, Kota Blitar, Jawa Timur

miqbalraihan7567@gmail.com¹, sync.saifulnb@gmail.com², udkhiati.mawaddah@gmail.com³

Kata Kunci :

Backpropagation;
Confusion Matrix; Jaringan Syaraf Tiruan; Peminatan Mahasiswa

ABSTRAK

Jaringan syaraf tiruan khususnya menggunakan metode *backpropagation*, telah menunjukkan efektivitasnya dalam berbagai aplikasi prediksi dan klasifikasi, termasuk dalam bidang pendidikan. Tujuan penelitian ini adalah untuk menganalisis metode *backpropagation* dalam memprediksi jurusan peminatan mahasiswa Program Studi Teknik Informatika Universitas Islam Balitar Blitar. Penggunaan metode penelitian yang dipakai adalah teknik penelitian kuantitatif. Dataset yang digunakan dalam penelitian ini diambil dari dokumen nilai mahasiswa Program Studi Teknik Informatika angkatan 2020 – 2021 sebanyak 47 data nilai mahasiswa. Data tersebut melalui proses normalisasi untuk memastikan konsistensi dan skala yang seragam. Setelah normalisasi, data dibagi menjadi dua jenis, yakni 38 data untuk pelatihan dan 9 data untuk pengujian. Hasil penelitian menunjukkan bahwa metode *backpropagation* mampu memberikan hasil akurasi terbaik sebesar 88%. Hasil tersebut akan menjadi lebih baik dengan menambahkan jumlah dataset yang digunakan.

Keywords

Artificial Neural Network;
Backpropagation;
Confusion Matrix; *Student Specialization*.

ABSTRACT

Artificial neural networks, especially using the backpropagation method, have shown their effectiveness in various prediction and classification applications, including in the field of education. The aim of this research is to analyze the backpropagation method in predicting students' majors in the Informatics Engineering Study Program, Balitar Islamic University, Blitar. The research method used is quantitative research techniques. The dataset used in this research was taken from the 2020 - 2021 Informatics Engineering Study Program student score documents, totaling 47 student score data. The data goes through a normalization process to ensure consistency and uniform scale. After normalization, the data was divided into two types, namely 38 data for training and 9 data for testing. The research results show that the backpropagation method is able to provide the best accuracy results of 88%. These results will get better by increasing the number of datasets used.

---Jurnal JISTI @2024---

PENDAHULUAN

Di era digital dan teknologi informasi saat ini, jurusan seperti Teknik Informatika menawarkan berbagai pilihan peminatan yang beragam, seperti sistem cerdas dan rekayasa perangkat lunak. Menurut Amala & Rosyidah (2015) Jurusan Rekayasa Perangkat Lunak (RPL) adalah salah satu peminatan di bidang Teknik Informatika yang fokus pada pengembangan perangkat lunak atau *software*. Menurut Putu dkk. (2015) Sistem Cerdas merupakan bidang ilmu yang menggunakan



teknologi untuk mengembangkan sistem yang mampu meniru atau melakukan tugas-tugas yang membutuhkan kecerdasan manusia.

Menurut hasil wawancara dengan mahasiswa dan kaprodi, didapatkan suatu kondisi dimana mahasiswa yang mengalami salah jurusan cenderung kurang aktif selama proses pembelajaran, mahasiswa cenderung terlibat dalam kegiatan yang tidak relevan dengan jurusan yang dipilih, dan mungkin mengabaikan tugas yang diberikan oleh dosen. Bahkan dalam kondisi terburuk, mahasiswa tersebut mungkin tidak melaksanakan tugas-tugas tersebut sama sekali. Kondisi semacam ini dapat mengakibatkan keterlibatan pasif dalam proses pembelajaran, yang pada akhirnya dapat berdampak negatif pada hasil akademik mahasiswa.

Dalam beberapa tahun terakhir, metode *Artificial Neural Network* banyak diaplikasikan oleh beberapa peneliti dalam kasus yang berbeda. Haryanto dkk., (2019) menerapkan *Artificial Neural Network* untuk keputusan penentuan penerima beasiswa, Nugraha & Retnowati, (2015) menerapkan *Artificial Neural Network* untuk keputusan penjurusan di SMA Islam Kepanjen Malang, (Pujiyanto dkk., 2018).

Berdasarkan permasalahan di atas, Universitas Islam Balitar Blitar memerlukan sistem pendukung keputusan pada sistem penjurusan mahasiswa Program Studi Teknik Informatika. Sistem pendukung keputusan tersebut digunakan untuk mencegah terjadinya kejadian “salah jurusan” pada teknik informatika dan juga mempermudah mahasiswa dalam menentukan jurusan yang akan diambil pada semester 6.

Dalam penelitian ini, nilai mata kuliah yang menjadi aspek variabel meliputi nilai dari mata kuliah berikut: sistem basis data, praktikum basis data, pemrograman berorientasi objek, praktikum pemrograman berorientasi objek, analisa perancangan sistem, rekayasa perangkat lunak, kecerdasan buatan, sistem pendukung keputusan, pemrograman web lanjut, dan pengantar e-bisnis. Evaluasi prestasi akademik mahasiswa dalam masing-masing mata kuliah tersebut akan menjadi fokus penelitian untuk mengembangkan sistem rekomendasi jurusan yang sesuai dengan minat dan kemampuan mereka dalam Program Studi Teknik Informatika.

Oleh karena itu, penelitian ini bertujuan untuk memanfaatkan algoritma *backpropagation* dalam mengembangkan sistem rekomendasi jurusan untuk mahasiswa Teknik Informatika. Dengan demikian, mahasiswa dapat menerima rekomendasi yang sesuai dengan minat dan kemampuan mereka, yang pada akhirnya dapat membantu mereka dalam memilih jurusan peminatan yang paling cocok.

Dalam pengembangan sistem rekomendasi ini, algoritma *backpropagation* dipilih karena kemampuannya untuk melatih jaringan yang kompleks, penyesuaian bobot dan bias untuk meningkatkan kinerja model, optimisasi melalui perubahan gradien, fleksibilitas dalam penggunaan, serta kemampuan untuk menangani masalah non-linear. Dengan memanfaatkan algoritma *backpropagation*, sistem rekomendasi jurusan untuk mahasiswa Teknik Informatika dapat dikembangkan dengan kemampuan adaptasi yang baik terhadap pola-pola yang kompleks dalam dataset, sehingga menghasilkan rekomendasi yang lebih akurat dan sesuai dengan minat serta kemampuan mahasiswa.

KAJIAN PUSTAKA

1. Pengertian Sistem Pendukung Keputusan

Menurut Aldo, (2019) alat berbasis komputer yang disebut "sistem pendukung keputusan" dapat membantu dalam pengambilan keputusan dengan memanfaatkan model dan data tertentu. Sistem Pendukung Keputusan pada dasarnya adalah kemajuan sistem informasi. Tujuan dari manajemen komputerisasi adalah untuk berkomunikasi dengan penggunanya. Tujuan dari interaksi ini adalah untuk membuatnya lebih mudah untuk mengintegrasikan berbagai aspek dari proses pengambilan keputusan sehingga keputusan yang lebih baik dapat dibuat. Sistem pendukung keputusan (SPK) adalah alat berbasis komputer yang dirancang untuk membantu pengambil keputusan dalam menyelesaikan masalah spesifik dengan memanfaatkan data dan model unik.



2. Pengertian Mahasiswa

Menurut Fauziah, (2015), Mahasiswa memiliki peran sebagai agen perubahan yang dapat memberikan solusi untuk tantangan yang dihadapi oleh masyarakat, bangsa, dan negaranya. Penting bagi mahasiswa untuk memiliki keterampilan berpikir kritis dan keberanian dalam menyampaikan fakta serta realitas yang ada. Selain itu, mahasiswa diharapkan memiliki pengetahuan yang luas untuk mengatasi berbagai masalah atau peristiwa yang berdampak signifikan pada kemajuan dan perkembangan bangsa dan negara.

3. Pengertian Penjurusan

Menurut Mafakhir & Solichin, (2020) penjurusan adalah tahapan penempatan siswa ke dalam suatu jurusan tertentu sesuai dengan keinginan dan kemampuan siswa, sehingga memungkinkan siswa menyerap materi pelajaran secara optimal dan sesuai dengan kapasitas mereka. Proses penjurusan dilakukan karena adanya perbedaan individu yang menyebabkan tingkat pemahaman setiap siswa terhadap pemahaman informasi berbeda-beda.

4. Pengertian *Artificial Neural Network* (ANN)

Menurut Shukla, A., Tiwari, R., dan Kala, R. (dalam Sopian dkk., 2019) Tujuan dari *Artificial Neural Network* adalah untuk mensimulasikan bagaimana otak manusia bekerja. Hal ini diyakini bahwa jutaan unit pengolahan kecil yang disebut neuron, yang beroperasi secara bersamaan, membentuk otak manusia. Melalui koneksi, neuron dihubungkan bersama; Setiap neuron menghasilkan *output* yang ditransmisikan ke neuron lain setelah menerima *input* dari sekelompok neuron. *Output neuron* yang berbeda dikumpulkan untuk diproses lebih lanjut. Koneksi di dalam jaringan rumit yaitu otak manusia selalu bergeser dan berubah.

5. Konsep Dasar *Artificial Neural Network*

Ide dasar di balik jaringan saraf berasal dari sekitar 10^{11} neuron yang membentuk otak manusia. Setiap informasi yang diterima diproses oleh neuron-neuron ini. Ada satu akson dan setidaknya satu dendrit di setiap neuron. Ada sekitar 10^4 sinapsis yang dibentuk oleh koneksi antara neuron ini. Kapasitas tertentu dalam operasi otak manusia dihasilkan oleh koneksi antara sel-sel saraf ini. Ide dasar dari jaringan saraf tiruan adalah untuk meniru cara berpikir sistem atau aplikasi nyata, termasuk bagaimana mereka menganalisis berbagai jenis input, dapat mentolerir kesalahan, dan berfungsi secara paralel (Suhartono, 2012).

6. Arsitektur *Artificial Neural Network*

Menurut Hermawan A. (dalam Hadi Wijaya, 2019), Jaringan saraf tiruan terdiri dari beberapa desain. Yaitu, Jaringan Lapisan Tunggal (*Single Layer Network*), Jaringan Banyak Lapisan (*Multilayer Network*), Jaringan Lapisan Kompetitif (*Competitive Layer*).

Jaringan Banyak Lapisan (*Multilayer Network*). Lapisan *input*, lapisan tersembunyi, dan lapisan *output* adalah tiga bagian utama dari jaringan multilayer. Jaringan yang memiliki beberapa lapisan dapat menangani masalah yang lebih rumit daripada jaringan lapisan tunggal. Namun, butuh beberapa saat bagi jaringan ini untuk dilatih menggunakan data yang diuji.

7. Pengertian *Adam Optimizer*

Adam didefinisikan sebagai sebuah metode yang mampu melakukan optimasi pada stokastik dengan meminimumkan bobot nilai yang hanya membutuhkan gradien orde pertama dalam memori yang kecil. Secara struktural, Adam merupakan kombinasi dari RMSprop dan Stochastic Gradient Descent (SGD) dengan momentum. Adam menggunakan gradien kuadrat dalam melakukan skala tingkat pembelajaran seperti RMSprop dan memanfaatkan momentum dari rata-rata pergerakan gradien seperti SGD dengan momentum. Adam merupakan metode learning rate adaptif dimana penghitungan learning rate untuk individu memiliki parameter yang berbeda (Arkadia dkk., 2022).



8. *Backpropagation Neural Network*

Menurut Heaton (dalam Nurmila dkk., 2010) Jaringan Saraf *Backpropagation* berlapis adalah algoritma pelatihan terpandu. Jaringan saraf *backpropagation* memodifikasi nilai bobotnya ke belakang dengan menggunakan data *output* kesalahan. Tahap propagasi ke depan harus diselesaikan sebelum mendapatkan kesalahan ini.

9. *Confusion Matrix*

Menurut Han dan Kamber (dalam Fluorida Fibrianda & Bhawiyuga, 2018) dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah *classifier* tersebut baik dalam mengenali *tuple* dari kelas yang berbeda. Nilai dari *True Positive* dan *True-Negative* memberikan informasi ketika *classifier* dalam melakukan klasifikasi data bernilai benar, sedangkan *False Positive* dan *False-Negative* memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data.

METODE PENELITIAN

1. Teknik Pengumpulan Data

Teknik pengumpulan data dalam penelitian ini ialah:

a. Observasi

Pada tahap ini peneliti akan meneliti langsung ke lapangan untuk mendapatkan data dan sumber informasi yang akurat dan melakukan interaksi langsung terhadap subjek. Dari hasil observasi disimpulkan bahwa beberapa mahasiswa merasa bahwa belum mendapatkan ilmu yang maksimal, sehingga mereka belum memiliki pandangan atau rencana setelah lulus nanti. Hasil nilai mereka rata-rata mendapatkan nilai yang beragam, karena mereka memiliki kendala pada proses belajar, salah satunya karena tidak memiliki laptop yang memadai dan beberapa dari mahasiswa tersebut terbagi waktu dengan pekerjaan mereka. Dengan demikian masih sedikit dari mereka yang mengimplementasikan ilmu yang didapat dari meja perkuliahan.

b. Wawancara

Pada penelitian ini metode pengumpulan data dengan wawancara dilakukan menggunakan instrumen wawancara kepada Bapak Saiful Nur Budiman, S.Kom., M.Kom., selaku kaprodi teknik informatika. Berikut beberapa pertanyaan yang digunakan pada proses wawancara tersebut.

Tabel 1 Daftar Pertanyaan

No	Pertanyaan
1	Bagaimana cara menentukan jurusan mahasiswa pada prodi teknik informatika yang dilakukan selama ini?
2	Apakah terdapat masalah pada saat menentukan jurusan mahasiswa teknik informatika?
3	Nilai mata kuliah apa saja yang menjadi acuan untuk melakukan penjurusan pada prodi teknik informatika?
4	Apakah ada metode rekomendasi jurusan yang saat ini digunakan oleh prodi teknik informatika?

c. Dokumentasi

Dokumentasi yang dikumpulkan dari fakultas terkait, seperti laporan nilai mata kuliah angkatan tahun 2020 – 2021. Data yang diambil dari dokumen ini akan digunakan untuk melengkapi data yang telah diperoleh melalui wawancara.

2. Metode *Backpropagation*

Beberapa langkah dapat dijelaskan mulai dari memasukkan data latih dan data uji. Selanjutnya, dilakukan normalisasi menggunakan *standard scaler* untuk mengubah nilai data. Setelah



itu, bobot dan bias diinisialisasi dalam suatu rentang tertentu. Setelah selesainya langkah-langkah tersebut, dilakukan metode *backpropagation* yang dimulai dengan tahap *feed forward*, dilanjutkan dengan tahap *backpropagation* dan tahap pembaruan bobot. Setelah proses tersebut selesai, dilakukan pengecekan terhadap nilai *error* dan akurasi, sehingga hasil dapat diklasifikasikan.

3. Perancangan Model

Pada tahap ini dilakukan perancangan model *backpropagation* dengan menentukan:

- a. Menentukan struktur model *backpropagation* seperti *input layer*, *hidden layer*, dan *output layer*. Pada penelitian ini penulis menggunakan pola arsitektur 10-10-1, 10-15-1, dan 10-20-1.
- b. Menentukan bobot dan bias pada model *backpropagation*.
- c. Menentukan *learning rate* pada model *backpropagation*. Pada penelitian ini penulis menggunakan *optimizer* adam dengan *learning rate* 0.0001.
- d. Menentukan *epoch*. Pada penelitian ini penulis menggunakan *epoch* sebanyak 500.
- e. Menentukan fungsi aktivasi sebagai fungsi pemroses *input* menjadi *output*. Pada penelitian ini penulis menggunakan fungsi aktivasi relu pada *hidden layer* dan fungsi aktivasi sigmoid pada *output layer*.
- f. Melakukan validasi dan verifikasi rancangan model

4. Pengujian Backpropagation

Setelah dilakukan proses prediksi oleh model *backpropagation* dilakukan perhitungan *confusion matrix* untuk mendapatkan *accuracy*, *specificity*, *precision* dan *recall*. Akurasi adalah nilai yang mengukur seberapa banyak keberhasilan model dalam membuat prediksi yang benar dari semua prediksi yang dilakukan terhadap *data test*. Untuk menghitung nilai akurasi dapat digunakan rumus (Luthfi Bangun Permadi & Gumilang, 2024):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \dots\dots\dots(13)$$

Sedangkan *precision* adalah nilai yang digunakan untuk mengevaluasi suatu model, seberapa baik dalam membuat prediksi benar untuk kelas positif dari keseluruhan prediksi positif yang dilakukan. Untuk menghitung nilai presisi dapat digunakan rumus:

$$Precision = \frac{TP}{TP+FP} \times 100\% \dots\dots\dots(14)$$

Selanjutnya *recall* merupakan metrik yang mengevaluasi seberapa baiknya model dalam mengidentifikasi kelas positif dengan benar. Untuk menghitung nilai *recall* dapat digunakan rumus:

$$Recall = \frac{TP}{TP+FN} \times 100\% \dots\dots\dots(15)$$

Kemudian *specificity* merupakan metrik yang menunjukkan seberapa baik model mengklasifikasikan kelas negatif. Untuk menghitung nilai *specificity* dapat digunakan rumus:

$$Specificity = \frac{TN}{TN+FP} \times 100\% \dots\dots\dots(16)$$

HASIL DAN PEMBAHASAN

1. Proses Algoritma dengan Google Colab

a. Persiapan Data

Sebelum dilakukan pelatihan data, dilakukan persiapan data yang bertujuan untuk menyiapkan data yang akan digunakan dalam pemodelan *backpropagation*.

Tabel 4 Baris Kode Persiapan Data

No.	Baris Kode
1	<code>from sklearn.preprocessing import StandardScaler</code>
2	<code>train_features = train_data.drop(['NIM', 'Nama Mahasiswa', 'Target'], axis=1)</code>
3	<code>train_targets = train_data['Target']</code>



```

4     test_features = test_data.drop(['NIM', 'Nama Mahasiswa', 'Target'],
5         axis=1)
6     test_targets = test_data['Target']
7     scaler = StandardScaler()
8     train_features_scaled = scaler.fit_transform(train_features)
9     test_features_scaled = scaler.transform(test_features)
10    pd.DataFrame(train_features_scaled,
11        columns=train_features.columns).head(),
12    pd.DataFrame(test_features_scaled,
13        columns=test_features.columns).head()
    
```

Pada tabel 4 digunakan untuk persiapan pelatihan data, yang diawali dengan menghapus kolom “NIM”, “Nama Mahasiswa”. “Target” dari *data train* dan *data test*, juga menyimpan kolom “Target” dari *data train* dan *data test* yang mana akan diperlukan untuk melihat nilai yang ingin diprediksi oleh model. Lalu dilakukan normalisasi data menggunakan *standardscaler* yang berasal dari library *sklearn.preprocessing*, dengan menggunakan *standardscaler*, setiap fitur dalam dataset akan memiliki *mean* nol dan *varians* satu yang membuat dataset akan lebih mudah untuk diolah oleh algoritma *backpropagation*.

Digunakan metode “*fit_transform*” untuk menghitung *mean* dan *standard deviation* dari setiap fitur dalam *data train* untuk kemudian digunakan dalam transformasi. Selanjutnya, *scaler* mentransformasi data pelatihan (*train_features*) sehingga setiap fitur di-skalikan dan dinormalisasi berdasarkan *mean* dan *standard deviation* yang telah dihitung. Untuk *data test*, digunakan metode “*transform*” dengan menggunakan parameter yang sama (*mean* dan *standard deviation*) yang dihitung dari data pelatihan untuk mentransformasi *data test*. Setelah dilakukan normalisasi data, selanjutnya adalah menampilkan beberapa baris pertama dari *data train* dan *data test* menggunakan *Dataframe* dari library *pandas*.

b. Pelatihan Data

Pada proses pelatihan data akan ada 3 pola yang digunakan, yaitu pola 10-10-1, 10-15-1 dan 10-20-1, penulis menggunakan pola berikut didasarkan pada pendekatan eksperimen berulang (*trial and error*) untuk menemukan konfigurasi yang memberikan hasil terbaik. Dengan mencoba berbagai jumlah neuron di lapisan tersembunyi, penulis dapat mengevaluasi bagaimana model menangkap kompleksitas dan pola dalam data. Pola yang digunakan sebagai berikut :

1) Pola 1 dengan 10-10-1

Pada proses pelatihan ini menggunakan jumlah *neuron* pada *input layer* sebanyak 10 *neuron*, jumlah *neuron* pada *hidden layer* sebanyak 10 *neuron*, *learning rate* sebesar 0,0001, *max epoch* sebanyak 500

Tabel 5 Baris Kode Pelatihan dengan Pola 10-10-1

No.	Baris Kode
1	<pre> model = Sequential([Dense(10, input_dim=train_features_scaled.shape[1], activation='relu'), BatchNormalization(), Dropout(0.5), Dense(10, activation='relu'), BatchNormalization(), Dropout(0.5), Dense(1, activation='sigmoid')]) </pre>
2	<pre> optimizer = Adam(learning_rate=0.0001) </pre>
3	<pre> model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy']) </pre>
4	<pre> reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss', </pre>



```
factor=0.2, patience=5, min_lr=0.00001)
5 history = model.fit( train_features_scaled, train_targets, epochs=500,
batch_size=16, verbose=1, validation_split=0.2,
callbacks=[EarlyStopping(monitor='val_loss', patience=20,
restore_best_weights=True), reduce_lr])
6 loss, accuracy = model.evaluate(test_features_scaled, test_targets)
7 print(f"Test Loss: {loss}, Test Accuracy: {accuracy}")
```

Pada tabel 5 digunakan model dibangun menggunakan model “sequential” yang berarti model akan memproses *input* melalui lapisan dalam urutan linier. Model “sequential” terdiri dari beberapa lapisan :

- Lapisan *Dense* : lapisan ini adalah lapisan yang sepenuhnya terhubung di mana setiap *neuron* terhubung ke semua *neuron* di lapisan sebelumnya. Lapisan pertama memiliki 10 *neuron* dan menggunakan fungsi aktivasi ReLU (relu), yang efektif untuk menangani masalah non-linearitas dan membantu mengurangi masalah *vanishing gradient*. Lapisan kedua memiliki 10 neuron dengan aktivasi yang sama.
- Batch Normalization* : Lapisan ini dinormalisasi aktivasi relu sebelumnya, membantu mempercepat pelatihan dengan mengurangi jumlah epoch yang diperlukan untuk melatih model.
- Dropout* : Lapisan ini secara acak mematikan *neuron* selama pelatihan dengan probabilitas tertentu (50% di sini), yang membantu mencegah *overfitting* dengan membuat model tidak terlalu bergantung pada *neuron* tertentu.

Pada model ini menggunakan Adam sebagai algoritma optimasi dengan *learning rate* yang sangat kecil (0.0001), yang membantu dalam konvergensi yang stabil terutama pada masalah yang lebih kompleks atau dengan *dataset* yang lebih kecil. Untuk fungsi *loss* digunakan “binary_crossentropy” karena ini adalah kasus klasifikasi biner. Untuk mengukur kinerja model digunakan metrik “accuracy”.

Model dilatih dengan menggunakan metode fit(), yang mengambil data fitur dan target yang telah dinormalisasi. Pelatihan dilakukan selama 500 *epochs* dengan *batch size* 16. Keras membagi data pelatihan lebih lanjut untuk validasi menggunakan 20% dari data. Dua *callbacks* digunakan:

- ReduceLRonPlateau : *Callback* ini mengurangi *learning rate* ketika metrik yang diamati (val_loss) berhenti membaik, membantu model untuk keluar dari plateau selama pelatihan.
- EarlyStopping : *Callback* ini menghentikan pelatihan lebih awal jika val_loss tidak membaik setelah 20 *epochs* (*patience*) dan mengembalikan bobot terbaik, menghindari pemborosan waktu pelatihan dan *overfitting*.

Setelah pelatihan selesai, model dievaluasi menggunakan set *data test* untuk menentukan *loss* dan akurasi. Ini bertujuan untuk memberikan gambaran tentang seberapa baik model dapat melakukan generalisasi pada data yang tidak terlihat selama pelatihan.

```
Epoch 16/500
2/2 [=====] - 0s 59ms/step - loss: 0.7920 - accuracy: 0.5667 - val_loss: 0.7422 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 17/500
2/2 [=====] - 0s 55ms/step - loss: 0.9351 - accuracy: 0.5333 - val_loss: 0.7463 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 18/500
2/2 [=====] - 0s 37ms/step - loss: 0.6157 - accuracy: 0.6667 - val_loss: 0.7511 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 19/500
2/2 [=====] - 0s 36ms/step - loss: 1.0778 - accuracy: 0.4000 - val_loss: 0.7566 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 20/500
2/2 [=====] - 0s 35ms/step - loss: 0.7786 - accuracy: 0.6667 - val_loss: 0.7603 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 21/500
2/2 [=====] - 0s 38ms/step - loss: 0.8908 - accuracy: 0.6000 - val_loss: 0.7651 - val_accuracy: 0.5000 - lr: 1.0000e-05
1/1 [=====] - 0s 30ms/step - loss: 0.7388 - accuracy: 0.5556
Test Loss: 0.7388083338737488, Test Accuracy: 0.555555820465088
```

Gambar 2 Hasil Pelatihan 10-10-1

Pada gambar 2 di atas adalah hasil dari proses pelatihan data dengan pola 10-10-1, *test loss* yang dicapai adalah 0.7388 dengan mencapai epoch 21 dari 500 epoch, karena pada saat di suatu epoch nilai *val loss* tidak semakin menurun sehingga fungsi *earlystopping* digunakan untuk mencegah terjadinya *overfitting* yang lebih parah.



2) Pola 2 dengan 10-15-1

Pada proses pelatihan ini menggunakan jumlah *neuron* pada *input layer* sebanyak 10 *neuron*, jumlah *neuron* pada *hidden layer* sebanyak 15 *neuron*, *learning rate* sebesar 0,0001, *max epoch* sebanyak 500

Tabel 6 Baris Kode Pelatihan dengan Pola 10-15-1

No.	Baris Kode
1	model = Sequential([Dense(10, input_dim=train_features_scaled.shape[1], activation='relu'), BatchNormalization(), Dropout(0.5), Dense(15, activation='relu'), BatchNormalization(), Dropout(0.5), Dense(1, activation='sigmoid')])
2	optimizer = Adam(learning_rate=0.0001)
3	model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])
4	reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=0.00001)
5	history = model.fit(train_features_scaled, train_targets, epochs=500, batch_size=16, verbose=1, validation_split=0.2, callbacks=[EarlyStopping(monitor='val_loss', patience=20, restore_best_weights=True), reduce_lr])
6	loss, accuracy = model.evaluate(test_features_scaled, test_targets)
7	print(f'Test Loss: {loss}, Test Accuracy: {accuracy}')

Pada tabel 6 digunakan model dibangun menggunakan model “sequential” yang berarti model akan memproses *input* melalui lapisan dalam urutan linier. Model “sequential” terdiri dari beberapa lapisan :

- Lapisan *Dense* : lapisan ini adalah lapisan yang sepenuhnya terhubung di mana setiap *neuron* terhubung ke semua *neuron* di lapisan sebelumnya. Lapisan pertama memiliki 10 *neuron* dan menggunakan fungsi aktivasi ReLU (relu), yang efektif untuk menangani masalah non-linearitas dan membantu mengurangi masalah *vanishing gradient*. Lapisan kedua memiliki 15 *neuron* dengan aktivasi yang sama.
- Batch Normalization* : Lapisan ini dinormalisasi aktivasi relu sebelumnya, membantu mempercepat pelatihan dengan mengurangi jumlah *epoch* yang diperlukan untuk melatih model.
- Dropout* : Lapisan ini secara acak mematikan *neuron* selama pelatihan dengan probabilitas tertentu (50% di sini), yang membantu mencegah *overfitting* dengan membuat model tidak terlalu bergantung pada *neuron* tertentu.

Pada model ini menggunakan Adam sebagai algoritma optimasi dengan *learning rate* yang sangat kecil (0.0001), yang membantu dalam konvergensi yang stabil terutama pada masalah yang lebih kompleks atau dengan *dataset* yang lebih kecil. Untuk fungsi *loss* digunakan “binary_crossentropy” karena ini adalah kasus klasifikasi biner. Untuk mengukur kinerja model digunakan metrik “accuracy”.

Model dilatih dengan menggunakan metode fit(), yang mengambil data fitur dan target yang telah dinormalisasi. Pelatihan dilakukan selama 500 *epochs* dengan batch size 16. Keras membagi data pelatihan lebih lanjut untuk validasi menggunakan 20% dari data. Dua *callbacks* digunakan:



- a. ReduceLROnPlateau : *Callback* ini mengurangi *learning rate* ketika metrik yang diamati (val_loss) berhenti membaik, membantu model untuk keluar dari plateau selama pelatihan.
- b. EarlyStopping : *Callback* ini menghentikan pelatihan lebih awal jika val_loss tidak membaik setelah 20 *epochs (patience)* dan mengembalikan bobot terbaik, menghindari pemborosan waktu pelatihan dan *overfitting*.

Setelah pelatihan selesai, model dievaluasi menggunakan *set data test* untuk menentukan *loss* dan akurasi. Ini bertujuan untuk memberikan gambaran tentang seberapa baik model dapat melakukan generalisasi pada data yang tidak terlihat selama pelatihan.

```
Epoch 17/500
2/2 [=====] - 0s 31ms/step - loss: 0.7450 - accuracy: 0.6000 - val_loss: 0.7387 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 18/500
2/2 [=====] - 0s 31ms/step - loss: 0.9934 - accuracy: 0.5000 - val_loss: 0.7391 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 19/500
2/2 [=====] - 0s 34ms/step - loss: 1.0787 - accuracy: 0.4000 - val_loss: 0.7386 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 20/500
2/2 [=====] - 0s 30ms/step - loss: 0.7998 - accuracy: 0.6000 - val_loss: 0.7407 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 21/500
2/2 [=====] - 0s 31ms/step - loss: 0.7025 - accuracy: 0.6667 - val_loss: 0.7417 - val_accuracy: 0.5000 - lr: 1.0000e-05
Epoch 22/500
2/2 [=====] - 0s 34ms/step - loss: 1.2247 - accuracy: 0.4000 - val_loss: 0.7406 - val_accuracy: 0.5000 - lr: 1.0000e-05
1/1 [=====] - 0s 31ms/step - loss: 0.6662 - accuracy: 0.7778
Test Loss: 0.666151594734192, Test Accuracy: 0.777777910232544
```

Gambar 3 Hasil Pelatihan 10-15-1

Pada gambar 3 di atas adalah hasil dari proses pelatihan data dengan pola 10-15-1, *test loss* yang dicapai adalah 0.5807 dengan mencapai *epoch 22* dari 500 *epoch*, karena pada saat di suatu *epoch* nilai *val loss* tidak semakin menurun sehingga fungsi *earlystopping* digunakan untuk mencegah terjadinya *overfitting* yang lebih parah.

3) Pola 3 dengan 10-20-1

Pada proses pelatihan ini menggunakan jumlah *neuron* pada *input layer* sebanyak 10 *neuron*, jumlah *neuron* pada *hidden layer* sebanyak 20 *neuron*, *learning rate* sebesar 0,0001, *max epoch* sebanyak 500.

Tabel 7. Baris Kode Pelatihan dengan Pola 10-20-1

No.	Baris Kode
1	model = Sequential([Dense(10, input_dim=train_features_scaled.shape[1], activation='relu'), BatchNormalization(), Dropout(0.5), Dense(20, activation='relu'), BatchNormalization(), Dropout(0.5), Dense(1, activation='sigmoid')])
2	optimizer = Adam(learning_rate=0.0001)
3	model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])
4	reduce_lr = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=0.00001)
5	history = model.fit(train_features_scaled, train_targets, epochs=500, batch_size=16, verbose=1, validation_split=0.2, callbacks=[EarlyStopping(monitor='val_loss', patience=20, restore_best_weights=True), reduce_lr])
6	loss, accuracy = model.evaluate(test_features_scaled, test_targets)
7	print(f'Test Loss: {loss}, Test Accuracy: {accuracy}')

Pada tabel 7 digunakan model dibangun menggunakan model “sequential” yang berarti model akan memproses *input* melalui lapisan dalam urutan linier. Model “sequential” terdiri dari beberapa lapisan :



- a) Lapisan *Dense* : lapisan ini adalah lapisan yang sepenuhnya terhubung di mana setiap *neuron* terhubung ke semua *neuron* di lapisan sebelumnya. Lapisan pertama memiliki 10 *neuron* dan menggunakan fungsi aktivasi ReLU (relu), yang efektif untuk menangani masalah non-linearitas dan membantu mengurangi masalah *vanishing gradient*. Lapisan kedua memiliki 20 *neuron* dengan aktivasi yang sama.
- b) *Batch Normalization* : Lapisan ini dinormalisasi aktivasi relu sebelumnya, membantu mempercepat pelatihan dengan mengurangi jumlah *epoch* yang diperlukan untuk melatih model.
- c) *Dropout* : Lapisan ini secara acak mematikan *neuron* selama pelatihan dengan probabilitas tertentu (50% di sini), yang membantu mencegah *overfitting* dengan membuat model tidak terlalu bergantung pada *neuron* tertentu.

c. Pengujian Data

Dari beberapa pola pada pelatihan data, didapatkan pola pengujian data sebagai berikut:

1) Pengujian data dengan pola 10-10-1

```
1/1 [=====] - 0s 45ms/step - loss: 0.7388 - accuracy: 0.5556
Test Loss: 0.738808338737488, Test Accuracy: 0.555555820465088
1/1 [=====] - 1s 653ms/step
      NIM          Nama Mahasiswa Predicted Class Target asli
0 20104410002      El Risiko Noval          1           0
1 20104410004      Indra Kurniawan          0           0
2 20104410006      Bagus Kurniawan Pratama 0           1
3 20104410010      Muhammad Fahmi Ahsan    0           0
4 20104410011      Nana Aidia Fitra Sania  1           1
5 20104410012      Firsta Rahmania Suahyo  0           1
6 20104410013      Bahrur Rochim           1           1
7 20104410014      Nurul Hidayah           0           1
8 20104410015      Muayyin Wafi Khoiriyah  1           1
```

Gambar 4 Hasil Pengujian 10-10-1

Pada gambar 5 merupakan hasil dari pengujian dengan pola arsitektur 10-10-1 menghasilkan akurasi sebesar 55%, pada pola 10-10-1 model bisa memprediksi 5 target dengan benar dari 9 data test yang diberikan.

2) Pengujian data dengan pola 10-15-1

```
1/1 [=====] - 0s 26ms/step - loss: 0.6662 - accuracy: 0.7778
Test Loss: 0.6661551594734192, Test Accuracy: 0.777777910232544
1/1 [=====] - 0s 84ms/step
      NIM          Nama Mahasiswa Predicted Class Target asli
0 20104410002      El Risiko Noval          1           0
1 20104410004      Indra Kurniawan          0           0
2 20104410006      Bagus Kurniawan Pratama 1           1
3 20104410010      Muhammad Fahmi Ahsan    0           0
4 20104410011      Nana Aidia Fitra Sania  1           1
5 20104410012      Firsta Rahmania Suahyo  1           1
6 20104410013      Bahrur Rochim           1           1
7 20104410014      Nurul Hidayah           0           1
8 20104410015      Muayyin Wafi Khoiriyah  1           1
```

Gambar 5 Hasil Pengujian 10-15-1

Pada gambar 6 merupakan hasil dari pengujian dengan pola arsitektur 10-15-1 menghasilkan akurasi sebesar 77%, pada pola 10-15-1 model bisa memprediksi 7 target dengan benar dari 9 data test yang diberikan.

2. Hasil Akurasi Menggunakan Metode *Backpropagation*

Berdasarkan hasil dari pengujian yang telah dilakukan, didapatkan perhitungan *accuracy*, *precision*, *recall* dan *specificity* seperti berikut :

a. Pola 10-10-1

1) Persamaan *accuracy*



$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \\ &= \frac{3+2}{3+2+1+3} \times 100\% \\ &= \frac{5}{9} \times 100\% \\ &= 55\% \end{aligned}$$

2) Persamaan *precision*

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \times 100\% \\ &= \frac{3}{3+1} \times 100\% \\ &= \frac{3}{4} \times 100\% \\ &= 75\% \end{aligned}$$

3) Persamaan *recall*

$$\begin{aligned} Recall &= \frac{TP}{TP+FN} \times 100\% \\ &= \frac{3}{3+3} \times 100\% \\ &= \frac{3}{6} \times 100\% \\ &= 50\% \end{aligned}$$

4) Persamaan *specificity*

$$\begin{aligned} Specificity &= \frac{TN}{TN+FP} \times 100\% \\ &= \frac{2}{2+1} \times 100\% \\ &= \frac{2}{3} \times 100\% \\ &= 66\% \end{aligned}$$

Berdasarkan pengujian pada pola 10-10-1 yang telah dilakukan, didapatkan nilai *accuracy*, *precision*, *recall* dan *specificity*. Pada nilai *accuracy* mendapatkan 55% yang artinya akurasi pada model tersebut pada saat *testing* memiliki tingkat akurasi sebesar 55%, kemudian untuk *precision* didapatkan nilai sebesar 75% yang artinya model dapat memprediksi benar kelas positif dari keseluruhan prediksi positif pada saat *testing* sebesar 75%, kemudian untuk *recall* mendapat 50% yang artinya model dapat mengidentifikasi kelas positif dengan benar sebanyak 50%, dan untuk nilai *specificity* mendapat 66% yang artinya model dapat mengklasifikasikan kelas negatif pada saat *testing* sebesar 66%.

b. Pola 10-20-1

1) Persamaan *accuracy*

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \\ &= \frac{5+3}{5+3+0+1} \times 100\% \\ &= \frac{8}{9} \times 100\% \\ &= 88\% \end{aligned}$$

2) Persamaan *precision*

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \times 100\% \\ &= \frac{5}{5+0} \times 100\% \\ &= \frac{5}{5} \times 100\% \\ &= 100\% \end{aligned}$$

3) Persamaan *recall*

$$Recall = \frac{TP}{TP+FN} \times 100\%$$



$$\begin{aligned} &= \frac{5}{5+1} \times 100\% \\ &= \frac{5}{6} \times 100\% \\ &= 83\% \end{aligned}$$

4) Persamaan *specificity*

$$\begin{aligned} \text{Specificity} &= \frac{TN}{TN+FP} \times 100\% \\ &= \frac{3}{3+0} \times 100\% \\ &= \frac{3}{3} \times 100\% \\ &= 100\% \end{aligned}$$

Berdasarkan pengujian pada pola 10-20-1 yang telah dilakukan, didapatkan nilai *accuracy*, *precision*, *recall* dan *specificity*. Pada nilai *accuracy* mendapatkan 88% yang artinya akurasi pada model tersebut pada saat *testing* memiliki tingkat akurasi sebesar 88%, kemudian untuk *precision* didapatkan nilai sebesar 100% yang artinya model dapat memprediksi benar kelas positif dari keseluruhan prediksi positif pada saat *testing* sebesar 100%, kemudian untuk *recall* mendapat 83% yang artinya model dapat mengidentifikasi kelas positif dengan benar sebanyak 83%, dan untuk nilai *specificity* mendapat 100% yang artinya model dapat mengklasifikasikan kelas negatif pada saat *testing* sebesar 100%.

3. Pembahasan

Hasil dari pengujian *backpropagation* menggunakan 9 *datatest* dari 47 *dataset* mahasiswa dengan 3 pola arsitektur yaitu pola 10-10-1, 10-15-1, dan 10-20-1 menghasilkan nilai akurasi yang cukup baik, pada pola arsitektur 10-10-1 mendapatkan akurasi sebesar 55%, pola arsitektur 10-15-1 mendapatkan akurasi sebesar 77%, dan pola arsitektur 10-20-1 mendapatkan akurasi sebesar 88% dari perhitungan menggunakan confusion matrix.

SIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan penelitian tentang penerapan metode *backpropagation* dalam rekomendasi peminatan jurusan ini, diperoleh beberapa kesimpulan yang meliputi :

1. Analisis metode *backpropagation* dalam rekomendasi jurusan dilakukan dengan dua tahap, yakni tahap pelatihan dan tahap pengujian, dengan 38 dari 47 data. Pada tahap pelatihan data, digunakan 3 model arsitektur yakni pola 10-10-1, 10-15-1 dan 10-20-1 karena dalam setiap jumlah *neuron* yang berbeda akan mempengaruhi hasil pelatihan data tersebut, Hasil pelatihan menunjukkan bahwa model dengan pola arsitektur 10-20-1 memberikan hasil prediksi terbaik dengan akurat.
2. Pada pelatihan ini data yang digunakan sejumlah 9 dari 47 data, dengan menggunakan nilai *learning rate* 0.0001, dengan jumlah *epoch* sebesar 500 *epoch*. Dari data pelatihan dan pengujian diperoleh nilai akurasi yang berbeda. Hasil tersebut diperoleh dari 3 skenario uji coba yang telah dilakukan yakni skenario perbedaan nilai *neuron* pada *hidden layer*, dan *hidden layer* yang memiliki nilai 20 terbukti memiliki pengaruh yang baik sehingga tingkat akurasi semakin baik yakni 88%.

Saran

Dalam pengembangan penerapan metode *backpropagation* dalam merekomendasikan jurusan ini diperlukan beberapa perbaikan untuk mencapai nilai maksimal, yakni :

1. Perlu memperbanyak *dataset* yang digunakan dalam proses pelatihan data agar didapatkan hasil yang akurat.
2. Memperbaiki metode *backpropagation* dengan mengkombinasikan 3 pola model arsitektur selain pola 10-10-1, 10-15-1 dan 10-20-1 dengan jumlah *hidden layer* yang berbeda untuk mencapai hasil yang lebih baik.



DAFTAR PUSTAKA

- Ahajjam, T., Moutaib, M., Aissa, H., Azrour, M., Farhaoui, Y., & Fattah, M. (2022). Predicting Students' Final Performance Using Artificial Neural Networks. *Big Data Mining And Analytics*, 5(4), 294–301. <https://doi.org/10.26599/bdma.2021.9020030>
- Aldo, D. (2019). Pemilihan Bibit Lele Unggul Dengan Menggunakan Metode Weighted Product. *Jurnal Teknologi Dan Open Source*, 2(1), 15–23.
- Amala, E., & Rosyidah, U. (2015). *Aplikasi Prioritas Penanganan Peralatan Kantor Pada Bpjs Kesehatan Regional Vi Semarang*.
- Arkadia, A., Hananto, B., & Prasvita, D. S. (2022). Optimasi Long Short Term Memory Dengan Adam Menggunakan Data Udara Kota Dki Jakarta. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (Senamika)*, 92–101.
- Cynthia, E. P., & Ismanto, E. (2017). Jaringan Syaraf Tiruan Algoritma Backpropagation Dalam Memprediksi Ketersediaan Komoditi Pangan Provinsi Riau. *Rabit (Jurnal Teknologi Dan Sistem Informasi Univrab)*, 2(2), 196–209.
- Fauziah, H. H. (2015). Faktor-Faktor Yang Mempengaruhi Prokrastinasi Akademik Pada Mahasiswa Fakultas Psikologi Uin Sgd Bandung. *Ejournal Of Sunan Gunung Djati State Islamic University (Uin)*, 2(2), 123–132.
- Fluorida Fibrianda, M., & Bhawiyuga, A. (2018). Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (Svm). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(9), 3112–3123. <http://j-ptiik.ub.ac.id>
- Hadi Wijaya, A. (2019). Artificial Neural Network Untuk Memprediksi Beban Listrik Dengan Menggunakan Metode Backpropagation (Studi Kasus Pt. Pln Regional Sumatera Barat). *Jurnal Coreit*, 5(2).
- Haryanto, D., Ramdani, C., Solihatin Wahidah, W., Gita Dinia, A., & Oktaviani, S. (2019). *Perancangan Sistem Pendukung Keputusan Penentuan Penerimaan Beasiswa Menggunakan Metode Artificial Neural Network (Ann) Di Institut Teknologi Telkom Purwokerto*.
- Indrawan, A. M., & Kusuma, A. P. (2022). Analisis Algoritma Jaringan Syaraf Tiruan Dengan Metode Backpropagation Dalam Mendeteksi Keahlian Mahasiswa Program Studi Teknik Informatika Universitas Islam Balitar. *Jurnal Mnemonic*, 5(1).
- Jufani, M. N., Zahro', H. Z., & Achmadi, S. (2022). Pengembangan Penentuan Sistem Pendukung Keputusan Penjurusan Siswa Di Sman 1 Sanggar Menggunakan Metode Analytical Hierarchy Process (Ahp) Dan Technique For Order Preference By Similarity To Ideal Solution (Topsis). *Jurnal Mahasiswa Teknik Informatika*, 6(2).
- Kuniasari, R., & Fatmawati, A. (2021). Penerapan Data Mining Dengan Algoritma C4.5 Untuk Penentuan Jurusan Sekolah Menengah Atas Implementation Of Data Mining With C4.5 Algorithm For Determining Senior High School. *Jurnal Insypro (Information System And Processing)*, 6(2), 1–7.
- Kurniawansyah, A. S. (2018). Implementasi Metode Artificial Neural Network Dalam Memprediksi Hasil Ujian Kompetensi Kebidanan (Studi Kasus: Akademi Kebidanan Dehasen Bengkulu). *Jurnal Pseudocode*, 5(1).
- Kusumadewi, S. (2003). *Artificial Intelligence (Teknik Dan Aplikasinya)*. Graha Ilmu.
- Laengge, I., Wowor, H. F., & Putro, M. D. (2016). Sistem Pendukung Keputusan Dalam Menentukan Dosen Pembimbing Skripsi. *E-Journal Teknik Informatika*, 9(1).
- Lesinski, G., Corns, S., & Dagli, C. (2016). Application Of An Artificial Neural Network To Predict Graduation Success At The United States Military Academy. *Procedia Computer Science*, 95, 375–382. <https://doi.org/10.1016/j.procs.2016.09.348>
- Lubis, D. (2013). Sistem Informasi Penjurusan Siswa Sma Negeri 9 Bandar Lampung Berdasarkan Nilai Yang Berkompetensi. *Mikrotik: Jurnal Manajemen Informatika*, 1(1).
- Lupat, R., Perera, R., Loi, S., & Li, J. (2023). Moanna: Multi-Omics Autoencoder-Based Neural Network Algorithm For Predicting Breast Cancer Subtypes. *Ieee Access*, 11, 10912–10924. <https://doi.org/10.1109/Access.2023.3240515>
- Luthfi Bangun Permadi, M., & Gumilang, R. (2024). Penerapan Algoritma Cnn (Convolutional Neural Network) Untuk Deteksi Dan Klasifikasi Target Militer Berdasarkan Citra Satelit. *Jurnal Sosial Dan Teknologi (Sostech)*, 4(2), 134–143.



- Mafakhir, A. Z., & Solichin, A. (2020). Penerapan Metode Naïve Bayes Classifier Untuk Penjurusan Siswa Pada Madrasah Aliyah Al-Falah Jakarta. *Fountain Of Informatics Journal*, 5(1), 21. <https://doi.org/10.21111/Fij.V5i1.4007>
- Nugraha, D. A., & Retnowati, W. (2015). Sistem Pendukung Keputusan Penjurusan Di Sma Menggunakan Metode Neural Network Backpropagation (Studi Kasus Sma Islam Kepanjen Malang). *Bimasakti : Jurnal Riset Mahasiswa Bidang Teknologi Informasi*, 3(2).
- Nurmila, N., Sugiharto, A., & Adi Sarwoko, E. (2010). Algoritma Back Propagation Neural Network Untuk Pengenalan Pola Karakter Huruf Jawa. *Jurnal Masyarakat Informatika*, 1(1).
- Pujianto, A., Kusriani, & Sunyoto, A. (2018). Perancangan Sistem Pendukung Keputusan Untuk Prediksi Penerima Beasiswa Menggunakan Metode Neural Network Backpropagation. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 5(2), 157–162. <https://doi.org/10.25126/Jtiik.201852631>
- Putu, I. A., Sinthiya, A., Rizal, M., Stmik, S., & Lampung, P. (2015). Rancangan Aplikasi Sistem Cerdas Pembelajaran Ilmu Bangun Datar Sd Negeri 01 Candiretno. Dalam *Technology Acceptance Model* (Vol. 4). www.stmikpringsewu.ac.id
- Santoso, A., Purnamasari, A. I., & Ali, I. (2024). Prediksi Harga Beras Menggunakan Metode Recurrent Neural Network Dan Long Short-Term Memory. *Jurnal Prosisko*, 11(1), 128–136.
- Saraswati, E., Umaidah, Y., & Voutama, A. (2021). Penerapan Algoritma Artificial Neural Network Untuk Klasifikasi Opini Publik Terhadap Covid-19. *Generation Journal*, 5(2), 109–118.
- Sawitri, M. N. D., Sumarjaya, I. W., & Tastrawati, N. K. T. (2018). Peramalan Menggunakan Metode Backpropagation Neural Network. *E-Jurnal Matematika*, 7(3), 264. <https://doi.org/10.24843/Mtk.2018.V07.I03.P213>
- Sopian, A., Wiyatno, A., & Riyandi, A. (2019). Komparasi Algoritma Support Vector Machines Dengan Algoritma Artificial Neural Network Untuk Memprediksi Nilai Persetujuan Kredit Modal Kerja Yang Diberikan Bank Umum. *Jurnal Teknologi Informatika & Komputer*, 5(1), 88–95.
- Suhartono, D. (2012). *Dasar Pemahaman Neural Network*. <https://socs.binus.ac.id/2012/07/26/konsep-neural-network/>
- Yuhandri, M. H., & Mayola, L. (2023). Identifikasi Pola Seleksi Penentuan Calon Wali Nagari Dengan Menggunakan Artificial Neural Network Algoritma Perceptron. *Jurnal Komtekinfo*, 158–165. <https://doi.org/10.35134/komtekinfo.V10i4.485>