

KLASIFIKASI BINER MENGGUNAKAN SINGLE LAYER PERCEPTRON PADA DATA KUESIONER PELAMAR ASISTEN LAB KOMPUTER

Diterima Redaksi: 4 Februari 2025; Revisi Akhir: 29 Mei 2025; Diterbitkan Online: 20 Juni 2025

Nur Shafwa Aulia Sitorus¹⁾, Armansyah²⁾

^{1, 2)} Program Studi Ilmu Komputer, Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara
^{1, 2)} Jl. IAIN No. 1 Medan, 20235 Telp. (061) 6615683, 6622925, Fax. (061) 6615683
e-mail: nushalito@gmail.com¹⁾, armansyah@uinsu.ac.id²⁾

Abstrak: Minat mahasiswa untuk menjadi asisten laboratorium komputer perlu dianalisis guna memahami faktor yang memengaruhinya. Penelitian ini menggunakan Jaringan Saraf Tiruan Single Layer Perceptron (SLP) untuk melakukan klasifikasi biner terhadap data kuesioner pelamar asisten lab yang dikumpulkan melalui Google Forms. Model SLP dilatih dengan bobot dan bias awal nol, laju pembelajaran 0,1, serta ambang batas 0,5. Hasil menunjukkan bahwa dalam dua epoch, model mampu mengenali pola dengan akurasi 75%. Model ini memiliki presisi 100%, namun recall hanya 50%, menghasilkan F1 Score sebesar 67%. Temuan ini menunjukkan bahwa SLP dapat mengolah data kuesioner dengan baik dan memiliki potensi untuk diterapkan pada dataset yang lebih besar guna meningkatkan kinerja model.

Kata Kunci— *Klasifikasi Biner, Jaringan Saraf Tiruan, Model Single Layer Perceptron, Pengenalan Pola, Pelatihan Model.*

Abstract: Students' interest in becoming computer lab assistants needs to be analyzed to understand the factors that influence it. This research uses a Single Layer Perceptron (SLP) Neural Network to perform binary classification on the questionnaire data of lab assistant applicants collected through Google Forms. The SLP model was trained with initial weights and biases of zero, a learning rate of 0.1, and a threshold of 0.5. The results show that within two epochs, the model was able to recognize patterns with an accuracy of 75%. This model has a precision of 100%, but a recall of only 50%, resulting in an F1 Score of 67%. These findings indicate that SLP can process questionnaire data well and has the potential to be applied to larger datasets to improve model performance.

Keywords— *Binary Classification, Artificial Neural Networks, Single Layer Perceptron Model, Pattern Recognition, Model Training.*

I. PENDAHULUAN

JARINGAN Syaraf Tiruan (JST) adalah salah satu pendekatan dalam kecerdasan buatan yang terinspirasi oleh cara kerja otak manusia dan telah terbukti efektif untuk menyelesaikan berbagai masalah komputasi, salah satunya klasifikasi [1][2]. Menurut [3], terdapat empat model klasifikasi yang terkenal yaitu, *Decision Tree*, Jaringan Syaraf Tiruan, Regresi Logistik dan *Naïve Bayes*. Klasifikasi adalah pengelompokan data atau objek baru ke dalam kelas atau label berdasarkan atribut-atribut tertentu [4][5], yang dimulai dengan pengumpulan data, kemudian membaginya untuk dilatih dengan data latih dan dievaluasi dengan data uji [6]. Dengan memanfaatkan pola dari data historis, teknik ini digunakan untuk memprediksi atau menentukan label pada data baru. Dalam berbagai aplikasi, klasifikasi telah diterapkan di bidang seperti diagnosis medis [7], deteksi penipuan, dan analisis sentimen dalam teks.

Single Layer Perceptron (SLP) merupakan model matematika dari *Artificial Neural Network* (ANN), yang diperkenalkan oleh Frank Rosenblatt pada tahun 1958. Model ini dianggap sebagai konfigurasi paling sederhana dari JST [8] dan dirancang untuk menyelesaikan masalah klasifikasi biner yang dapat dipisahkan secara linear [9]. SLP bekerja dengan menggabungkan fitur *input* dan bobot melalui fungsi aktivasi untuk menghasilkan keluaran yang sesuai [10]. Pemilihan SLP dalam penelitian

ini didasarkan pada beberapa alasan utama, salah satunya karena SLP memiliki struktur yang sederhana sehingga mudah dipahami dan diterapkan dalam berbagai studi awal mengenai jaringan syaraf tiruan. Model ini cukup efektif dalam menyelesaikan masalah klasifikasi biner yang dapat dipisahkan secara linear, sehingga sesuai untuk skenario di mana data memiliki distribusi yang jelas. Meskipun sederhana, SLP efektif dalam mengidentifikasi pola pada data biner [11] dan memberikan landasan penting bagi pemahaman jaringan syaraf tiruan secara keseluruhan.

Dalam berbagai penerapannya, termasuk analisis data kuisioner, pengambilan keputusan berbasis data biner sering kali menghadapi kendala ketika pola dalam data sulit dikenali secara langsung. Misalnya, pada kuesioner pelamar asisten laboratorium komputer, beberapa responden memilih untuk tidak menjadi asisten karena alasan ketidaksiapan psikis dalam membimbing mahasiswa lain. Tantangan seperti ini menunjukkan perlunya metode analisis yang sistematis dan akurat untuk mengklasifikasikan pola respons secara objektif. Dalam konteks ini, SLP dapat digunakan untuk mengolah data kuesioner guna mengidentifikasi pola dan membantu proses pengambilan keputusan berbasis data biner secara lebih terstruktur.

Penelitian ini bertujuan untuk mengimplementasikan *Single Layer Perceptron* dalam tugas klasifikasi biner pada data kuesioner pelamar asisten laboratorium komputer. Masalah utamanya adalah bagaimana menentukan kelayakan kandidat secara objektif dengan menggunakan data kuesioner yang tersedia. Proses pemilihan asisten laboratorium sering kali dilakukan secara manual, yang dapat menyebabkan subjektivitas dalam pengambilan keputusan. Oleh karena itu, penelitian ini berusaha menawarkan solusi dengan menggunakan SLP sebagai model klasifikasi sederhana yang memiliki kemampuan untuk mengotomatisasi proses seleksi yang didasarkan pada pola yang ditemukan dalam data.

Dengan memanfaatkan model sederhana seperti SLP, penelitian ini diharapkan dapat menjelaskan mekanisme kerja dasar JST serta menunjukkan penerapannya dalam menyelesaikan masalah klasifikasi biner secara efektif. Selain itu, hasil penelitian ini dapat menjadi dasar untuk pengembangan sistem seleksi berbasis kecerdasan buatan yang lebih kompleks dan akurat.

II. TINJAUAN PUSTAKA

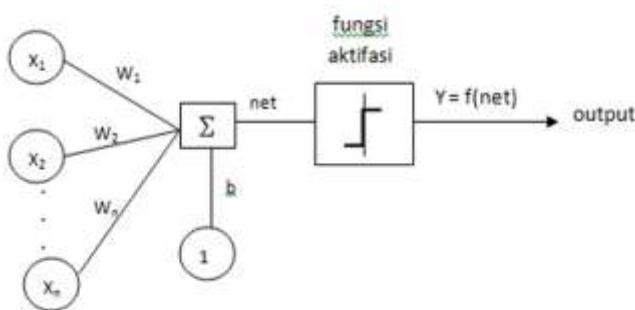
A. Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST: *Neural Network*) merupakan paradigma pemrosesan informasi yang terinspirasi oleh sistem syaraf secara biologis, sama seperti otak yang memproses suatu informasi [12]. Ini merupakan representasi buatan dari otak manusia yang digunakan untuk memecahkan permasalahan komputasi dan pengenalan pola. Istilah buatan disini digunakan karena jaringan syaraf diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran [13]. Dalam beberapa tahun terakhir, JST mengalami perkembangan yang cukup baik walaupun sempat mengalami masa vakum selama bertahun-tahun [14].

B. Single Layer Perceptron (SLP)

Menurut [15], model jaringan Perceptron ditemukan pertama kali oleh Rosenbatt (1962) dan Minsky – Papert (1969). Ini merupakan model yang memiliki aplikasi dan pelatihan yang paling baik pada era tersebut. Perceptron merupakan salah satu bentuk jaringan sederhana, yang biasanya digunakan untuk mengklasifikasikan suatu pola tipe tertentu yang sering dikenal dengan pemisahan secara linear [16]. Pada dasarnya *Single Layer Perceptron* atau SLP memiliki bobot yang dapat diatur, juga dapat digunakan dalam kasus untuk mengenali fungsi logika “AND” dengan masukan dan keluaran bipolar.

Pada proses pembuatan arsitektur jaringan, kita harus mempunyai rancangan data yang akan digunakan untuk membangun sebuah struktur jaringan. Data yang dibutuhkan seperti *learning rate*, bobot, ambang batas (*threshold*), bias, jumlah variabel *input*, nilai variabel *input* dan target (*output*) yang nantinya akan digunakan dalam proses pelatihan dan pengujian sistem.



Gambar 1. Arsitektur JST Model Perceptron

Algoritma Perceptron dapat dituliskan sebagai berikut :

1. Inisialisasi bobot dengan melakukan penyesuaian semua bobot dan *threshold* (θ) untuk bilangan acak terkecil atau sama dengan nilai 0.
2. Lakukan kalkulasi aktivasi dengan langkah berikut:
 - a. Hitung respons untuk *output* dengan rumus:

$$y_{in} = \sum_1^n x_i w_i \quad (1)$$

Dimana W adalah bobot, X_i ($I = 1, 2, 3, \dots, n$ dimana n adalah banyaknya lapisan *input*), dengan *threshold* (θ) sebagai berikut :

$$y = \begin{cases} 1 & \text{jika } y_{in} \geq \theta \\ 0 & \text{jika } -\theta \leq y_{in} < \theta \\ -1 & \text{jika } y_{in} < -\theta \end{cases} \quad (2)$$

- b. Bandingkan nilai *output* dengan target. Jika $y = t$, maka tidak akan terjadi perubahan bobot dan bias. Namun jika $y \neq t$, maka lakukan perubahan bobot dan bias dengan rumus sebagai berikut:

$$w_i (\text{baru}) = w_i (\text{lama}) + \Delta w \quad (3)$$

$$b_i (\text{baru}) = b_i (\text{lama}) + \Delta b \quad (4)$$

Dari penjelasan diatas dapat disimpulkan bahwa JST mendasari adanya proses belajar (latihan). Oleh karena itu, Jaringan Saraf Tiruan yang akan digunakan harus dikenai pelatihan secara berulang-ulang [17].

C. Klasifikasi Biner

Klasifikasi merupakan pengelompokan data, bisa berupa objek baru ke kelas atau label berdasarkan atribut yang dimiliki. Klasifikasi biasanya dilakukan dengan menggunakan algoritma pembelajaran mesin, yang akan mempelajari pola atau hubungan antara fitur-fitur yang dimiliki oleh objek-objek dalam data latih, dan kemudian membuat model atau aturan-aturan untuk memprediksi label atau kelas dari objek-objek yang belum diidentifikasi kelas nya dalam data uji [5].

Klasifikasi biner adalah jenis tugas dalam pembelajaran mesin [18] dan statistik yang melibatkan pengelompokan data ke dalam salah satu dari dua kategori atau kelas yang sering disebut kelas positif (1) dan kelas negatif (0), tentu dengan tujuan untuk mempelajari pola dari data yang diberikan. Klasifikasi sendiri memiliki tantangan, diantaranya ketidakseimbangan kelas yang memiliki jumlah data jauh lebih sedikit dibandingkan kelas lain, dan sering terjadinya *overfitting* [19] dimana model bekerja sangat baik pada data latih tetapi buruk pada data uji.

D. Confusion Matrix

Confusion Matrix adalah tabel yang digunakan untuk menilai kinerja model klasifikasi dalam *machine learning* dimana tabel menyatakan klasifikasi jumlah data uji yang benar dan salah [20]. Tabel ini memberikan representasi prediksi yang dilakukan oleh model yang dibandingkan dengan label sebenarnya [21][22]. Tabel ini berfungsi untuk menunjukkan bagaimana model menangani setiap kelas yang ada di dalam dataset.

Confusion Matrix sendiri umumnya berbentuk tabel 2x2 [23] untuk kasus klasifikasi biner, dimana tabel tersebut terdiri dari empat elemen utama, yaitu:

1. *True Positive* (TP) : Jumlah data yang benar diprediksi sebagai kelas positif.
2. *True Negative* (TN) : Jumlah data yang benar diprediksi sebagai kelas negatif.
3. *False Positive* (FP) : Jumlah data yang salah diprediksi sebagai kelas positif (seharusnya negatif).
4. *False Negative* (FN) : Jumlah data yang salah diprediksi sebagai kelas negatif (seharusnya positif).

Dari *confusion matrix*, ada beberapa matriks evaluasi yang dapat dihitung, yaitu :

1. *Accuracy*, yaitu proporsi prediksi yang benar terhadap keseluruhan data.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

2. *Precision*, yaitu proporsi prediksi positif yang benar terhadap semua prediksi positif.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (6)$$

3. *Recall*, yaitu proporsi data positif yang benar-benar terdeteksi oleh model.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (7)$$

4. *F1-Score*, yaitu rata-rata harmonis antara *Precision* dan *Recall*, yang berguna untuk data yang tidak seimbang.

$$F1 - \text{Score} = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (8)$$

III. METODE PENELITIAN

Penelitian ini merupakan penelitian kuantitatif, dengan menggunakan data berupa data kuesioner terkait dengan pelamar asisten laboratorium. Adapun teknik pengumpulan datanya adalah dengan menyebarluaskan kuesioner kepada mahasiswa yang berinteraksi dengan asisten laboratorium, khususnya mahasiswa semester ganjil 2024/2025. Penyebarluasan kuesioner dilakukan secara *online* melalui *platform* Google Form, sehingga memudahkan responden untuk mengisi kuesioner dengan cepat dan efisien.

Tabel 1 Instrumen Kuisisioner

| Instrumen Ke- | Pertanyaan | Jawaban |
|---------------|---|----------|
| 1 | Apakah kamu senang berbagi ilmu dengan orang lain, baik secara lisan maupun tulisan? | Ya/Tidak |
| 2 | Apakah kamu memiliki minat untuk belajar hal-hal terkait perangkat komputer serta pengembangan perangkat lunak dan aplikasi? | Ya/Tidak |
| 3 | Apakah kamu senang belajar hal-hal baru terkait teknologi komputer, seperti merakit dan memperbaiki komputer, serta membuat program sederhana? | Ya/Tidak |
| 4 | Apakah dengan menjadi asisten laboratorium ilmu komputer dapat membantu kamu untuk meningkatkan pengetahuan dan kompetensi serta membantu pengembangan diri kamu? | Ya/Tidak |
| 5 | Apakah kamu siap untuk membimbing atau mengajar mahasiswa pada saat jam praktikum berlangsung dan dosen tidak dapat berhadir di laboratorium? | Ya/Tidak |
| 6 | Apakah kamu berminat untuk mendaftar sebagai asisten laboratorium ilmu komputer tahun ajaran 2025? | Ya/Tidak |

Data kuesioner yang dikumpulkan dianalisis dengan menggunakan jaringan syaraf tiruan model *single layer Perceptron* dengan menggunakan arsitektur pada Gambar 1.

IV. HASIL DAN PEMBAHASAN

A. Hasil

Tabel di bawah ini merupakan hasil pengumpulan data yang diambil langsung dari data kuisisioner yang telah disebarluaskan, terlihat peneliti memperoleh 78 data mahasiswa dari kuisisioner tersebut. Hasilnya adalah sebagai berikut:

Tabel 2 Data Hasil Kuisioner

| Variabel | x ₁ | x ₂ | x ₃ | x ₄ | x ₅ | Target |
|--------------|----------------|----------------|----------------|----------------|----------------|--------|
| Mahasiswa 1 | Ya | Ya | Ya | Tidak | Ya | Ya |
| Mahasiswa 2 | Ya | Ya | Ya | Ya | Ya | Ya |
| Mahasiswa 3 | Ya | Ya | Ya | Ya | Ya | Tidak |
| Mahasiswa 4 | Ya | Tidak | Ya | Ya | Ya | Tidak |
| Mahasiswa 5 | Ya | Tidak | Ya | Ya | Ya | Ya |
| Mahasiswa 6 | Tidak | Ya | Ya | Tidak | Tidak | Tidak |
| Mahasiswa 7 | Ya | Ya | Tidak | Ya | Ya | Ya |
| Mahasiswa 8 | Ya | Ya | Tidak | Ya | Ya | Tidak |
| Mahasiswa 9 | Ya | Ya | Ya | Tidak | Tidak | Tidak |
| Mahasiswa 10 | Tidak | Tidak | Tidak | Tidak | Ya | Tidak |
| ... | ... | ... | ... | ... | ... | ... |
| Mahasiswa 78 | Tidak | Tidak | Ya | Ya | Tidak | Tidak |

Berdasarkan hasil yang ditampilkan pada Tabel 2, berikut adalah 4 sampel data yang diambil dari hasil kuisioner yang digunakan untuk melakukan analisis komputasi model SLP, di antaranya :

Tabel 3 Data Hasil Kuisioner

| Variabel | Mahasiswa 1 | Mahasiswa 2 | Mahasiswa 3 | Mahasiswa 4 |
|----------------|-------------|-------------|-------------|-------------|
| x ₁ | Ya | Tidak | Ya | Ya |
| x ₂ | Ya | Ya | Ya | Ya |
| x ₃ | Ya | Ya | Ya | Ya |
| x ₄ | Tidak | Tidak | Ya | Tidak |
| x ₅ | Ya | Tidak | Ya | Tidak |
| Target | Ya | Tidak | Ya | Tidak |

Di bawah ini adalah data untuk analisis komputasi pada penelitian, di antaranya sebagai berikut:

Tabel 4 Data untuk Perhitungan Komputasi

| x ₁ | x ₂ | x ₃ | x ₄ | x ₅ | Target |
|----------------|----------------|----------------|----------------|----------------|--------|
| 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |

Hasil penelitian di bawah ini menunjukkan data yang dikumpulkan dapat dikenali dengan SLP. Penelitian ini menghasilkan pengklasifikasian data kuesioner dengan tepat dimana pada Epoch ke-2, pola sudah dapat dikenali dengan bobot dan bias awal 0 serta *learning rate* 0,1 dan ambang batas 0,5. Hasil penelitian dapat dilihat pada Gambar 2 di bawah ini:

| Masukan | | | | | Target | $\sum w_i + b$ | Output | Perubahan Bobot | | | | | | Bobot Baru | | | | | | | | |
|----------------|----------------|----------------|----------------|----------------|--------|----------------|--------|-----------------|------------------------|--------------|--------------|--------------|--------------|--------------|------------|----------------|----------------|----------------|----------------|----------------|-----|--|
| w ₁ | w ₂ | w ₃ | w ₄ | w ₅ | | * | t | o _t | y = f(w _t) | Δw_1 | Δw_2 | Δw_3 | Δw_4 | Δw_5 | Δb | w ₁ | w ₂ | w ₃ | w ₄ | w ₅ | b | |
| 1 | 1 | 1 | 0 | 1 | | 0,1 | 1 | 0 | 0 | 0,1 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | 0 | 0 | 0 | 0 | 0,1 | 0,1 | |
| 0 | 1 | 1 | 0 | 0 | | 0,1 | 0 | 0,5 | 0 | 0,1 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | 0,3 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | |
| 1 | 1 | 1 | 1 | 1 | | 0,1 | 1 | 0,5 | 1 | 0,1 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | 0,3 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | |
| 1 | 1 | 1 | 0 | 0 | | 0,1 | 0 | 0,4 | 0 | 0,1 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | 0,3 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | |
| Epoch ke-1 | | | | | | | | | | | | | | | | | | | | | | |
| Epoch ke-2 | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | 0 | 1 | | 0,1 | 1 | 0,5 | 1 | 0,1 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | 0,3 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | |
| 0 | 1 | 1 | 1 | 0 | | 0,1 | 0 | 0,3 | 0 | 0,1 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | 0,3 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | |
| 1 | 1 | 1 | 1 | 1 | | 0,1 | 1 | 0,5 | 1 | 0,1 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | 0,3 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | |
| 1 | 1 | 1 | 0 | 0 | | 0,1 | 0 | 0,4 | 0 | 0,1 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | 0,3 | 0,1 | 0,1 | 0 | 0,1 | 0,1 | |

Gambar 2. Hasil Penelitian

B. Pembahasan

Berdasarkan data hasil untuk analisis komputasi pada Tabel 4, lakukan perhitungan manual dimana 0 adalah untuk keluaran “Tidak” dan 1 untuk keluaran “Ya”, dengan bobot dan bias awal, *learning rate*, serta *threshold* sebagai berikut:

| | | |
|--------------------------|-------|--|
| Bobot awal (w) | = 0 | dengan |
| Bias awal (b) | = 0 | |
| Threshold (θ) | = 0,5 | $y = \begin{cases} 1 & \text{jika } y_{in} \geq \theta \\ 0 & \text{jika } -\theta \leq y_{in} < \theta \\ -1 & \text{jika } y_{in} < -\theta \end{cases}$ |
| Learning rate (η) | = 0,1 | |

Dengan ketentuan di atas, maka hitung pada epoch ke-1

1. Pada pola (1, 1, 1, 0, 1) $\rightarrow t = 1$

$(1 \times 0) + (1 \times 0) + (1 \times 0) + (0 \times 0) + (1 \times 0) + 0 = 0$, maka $y = 0$ karena $-\theta \leq y_{in} < \theta$

Karena *output* tidak sesuai dengan target, maka lakukan perubahan bobot.

| | |
|--|--|
| $\Delta w = \eta \cdot t \cdot x_i$ | w _i (baru) = w _i (lama) + Δw |
| $\Delta w_1 = 0,1 \times 1 \times 1 = 0,1$ | w ₁ = 0 + 0,1 = 0,1 |
| $\Delta w_2 = 0,1 \times 1 \times 1 = 0,1$ | w ₂ = 0 + 0,1 = 0,1 |
| $\Delta w_3 = 0,1 \times 1 \times 1 = 0,1$ | w ₃ = 0 + 0,1 = 0,1 |
| $\Delta w_4 = 0,1 \times 1 \times 0 = 0$ | w ₄ = 0 + 0 = 0 |
| $\Delta w_5 = 0,1 \times 1 \times 1 = 0,1$ | w ₅ = 0 + 0,1 = 0,1 |
| $\Delta b = 0,1 \times 1 = 0,1$ | b = 0 + 0,1 = 0,1 |

2. Pada pola (0, 1, 1, 0, 0) $\rightarrow t = 0$

$(0 \times 0,1) + (1 \times 0,1) + (1 \times 0,1) + (0 \times 0) + (0 \times 0,1) + 0,1 = 0,3$, maka $y = 0$ karena $-\theta \leq y_{in} < \theta$

Karena *y* sesuai dengan target, maka tidak terjadi perubahan bobot.

3. Pada pola (1, 1, 1, 1, 1) $\rightarrow t = 1$

$(1 \times 0,1) + (1 \times 0,1) + (1 \times 0,1) + (1 \times 0) + (1 \times 0,1) + 0,1 = 0,5$, maka $y = 1$ karena $y_{in} \geq \theta$

Karena *y* sesuai dengan target, maka tidak terjadi perubahan bobot.

4. Pada pola (1, 1, 1, 0, 0) $\rightarrow t = 0$

$(1 \times 0,1) + (1 \times 0,1) + (1 \times 0,1) + (0 \times 0) + (0 \times 0,1) + 0,1 = 0,4$, maka $y = 0$ karena $-\theta \leq y_{in} < \theta$

Karena *y* sesuai dengan target, maka tidak terjadi perubahan bobot.

Tabel 5 Tabel Perbandingan pada Epoch-1

| Target (t) | Prediksi (p) |
|---------------------|--------------|
| 1 (Tidak Terpenuhi) | 0 |
| 0 (Terpenuhi) | 0 |
| 1 (Terpenuhi) | 1 |
| 0 (Terpenuhi) | 0 |

Dari Tabel 5 di atas, ada 1 pola lagi yang masih belum terpenuhi, maka perhitungan akan dilanjutkan ke epoch ke-2, dengan bobot awal dan bias, *learning rate* dan *threshold* masih sama seperti perubahan bobot yang terakhir dilakukan.

1. Pada pola (1, 1, 1, 0, 1) $\rightarrow t = 1$

$(1 \times 0,1) + (1 \times 0,1) + (1 \times 0,1) + (0 \times 0) + (1 \times 0,1) + 0,1 = 0,5$, maka $y = 1$ karena $y_{in} \geq \theta$

Karena *y* sesuai dengan target, maka tidak terjadi perubahan bobot.

2. Pada pola (0, 1, 1, 0, 0) $\rightarrow t = 0$

$(0 \times 0,1) + (1 \times 0,1) + (1 \times 0,1) + (0 \times 0) + (0 \times 0,1) + 0,1 = 0,3$, maka $y = 0$ karena $-\theta \leq y_{in} < \theta$

Karena *y* sesuai dengan target, maka tidak terjadi perubahan bobot.

3. Pada pola (1, 1, 1, 1, 1) $\rightarrow t = 1$

$(1 \times 0,1) + (1 \times 0,1) + (1 \times 0,1) + (1 \times 0) + (1 \times 0,1) + 0,1 = 0,5$, maka $y = 1$ karena $y_{in} \geq \theta$

Karena *y* sesuai dengan target, maka tidak terjadi perubahan bobot.

4. Pada pola (1, 1, 1, 0, 0) $\rightarrow t = 0$

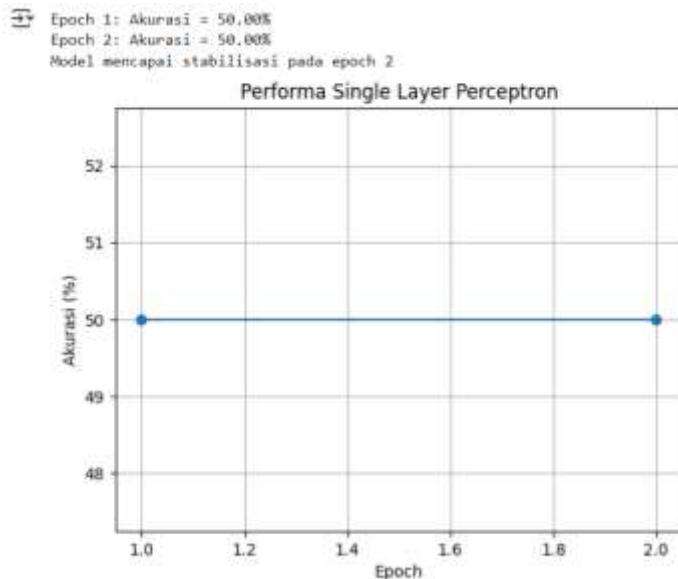
$(1 \times 0,1) + (1 \times 0,1) + (1 \times 0,1) + (0 \times 0) + (0 \times 0,1) + 0,1 = 0,4$, maka $y = 0$ karena $-\theta \leq y_{in} < \theta$

Karena y sesuai dengan target, maka tidak terjadi perubahan bobot.

Tabel 6. Tabel Perbandingan pada Epoch-2

| Target(t) | Prediksi (p) |
|---------------|--------------|
| 1 (Terpenuhi) | 1 |
| 0 (Terpenuhi) | 0 |
| 1 (Terpenuhi) | 1 |
| 0 (Terpenuhi) | 0 |

Dari Tabel 6 di atas, seluruh pola sudah berhasil dikenali semuanya pada epoch ke-2. Pola dapat dikenali dengan menggunakan bobot awal 0 dan bobot baru $w_1 = 0,1$, $w_2 = 0,1$, $w_3 = 0,1$, $w_4 = 0$, $w_5 = 0,1$, dan $b = 0,1$. Serta dengan menggunakan *learning rate* 0,1 dan *threshold* 0,5. Setelah pola berhasil dikenali pada epoch ke-2, maka lakukan pengujian program dengan menggunakan Google Colab.



Gambar 3. Hasil Pengujian Program Perhitungan Manual

Setelah pola berhasil dikenali dengan program, tahap selanjutnya adalah melakukan perhitungan *confusion matrix* secara manual, dengan menggunakan data pelatihan pada Tabel 7.

Tabel 7 Data Pelatihan untuk Confusion Matrix

| x₁ | x₂ | x₃ | x₄ | x₅ | Target | Prediksi |
|----------------------|----------------------|----------------------|----------------------|----------------------|---------------|-----------------|
| 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |

Confusion Matrix disusun dengan cara menghitung jumlah *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN) dari data hasil prediksi.

True Positive (TP) adalah kondisi ketika model memprediksi nilai positif dan sesuai dengan target sebenarnya. *False Positive* (FP) terjadi ketika model memprediksi nilai positif, tetapi target sebenarnya adalah negatif. *False Negative* (FN) merupakan kondisi ketika model memprediksi nilai negatif, tetapi target sebenarnya adalah positif. Sementara itu, *True Negative* (TN) terjadi ketika model memprediksi nilai negatif dan sesuai dengan target sebenarnya. Berikut ini tabel hasil perhitungan *confusion matrix*:

Tabel 8 Hasil *Confusion Matrix*

| Klasifikasi Aktual | Prediksi Positif (1) | Prediksi Negatif (0) |
|---------------------------|-----------------------------|-----------------------------|
| Positif (1) | TP = 1 | FN = 1 |
| Negatif (0) | FP = 0 | TN = 2 |

Dengan Tabel 8, lakukan evaluasi model yang dapat dilakukan dengan menghitung matriks seperti *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Berikut ini perhitungan manual *confusion matrix* [24]:

$$1. \text{ Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{1+2}{1+2+0+1} = \frac{3}{4} = 0,75$$

Dimana $0,75 \times 100\% = 75\%$

$$2. \text{ Precision} = \frac{TP}{TP+FP} = \frac{1}{1+0} = \frac{1}{1} = 1,0$$

Dimana $1,0 \times 100\% = 100\%$

$$3. \text{ Recall} = \frac{TP}{TP+FN} = \frac{1}{1+1} = \frac{1}{2} = 0,5$$

Dimana $0,5 \times 100\% = 50\%$

$$4. \text{ F1 - Score} = 2 \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \frac{1,0 \times 0,5}{1,0 + 0,5} = 2 \frac{0,5}{1,5} = 0,67$$

Dimana $0,67 \times 100\% = 67\%$

Berdasarkan perhitungan di atas, dapat dijelaskan bahwa algoritma *Single Layer Perceptron* (SLP) mampu memprediksi dengan tingkat akurasi sebesar 75%, yang berarti 3 dari 4 data berhasil diprediksi dengan benar.

- Untuk nilai *Precision* mencapai 100%, menunjukkan bahwa semua prediksi positif model benar-benar sesuai dengan target positif. Dengan kata lain, model sangat andal dalam memastikan prediksi positifnya.
- Untuk nilai *Recall* hanya 50%, yang berarti model hanya mampu mendeteksi separuh dari total target positif yang ada.
- Sedangkan untuk nilai *F1-Score* sebesar 67%, memberikan gambaran rata-rata harmonis antara *precision* dan *recall*, mencerminkan keseimbangan kinerja model dalam deteksi target positif.

Secara keseluruhan, model memiliki tingkat presisi yang tinggi, tetapi kemampuan untuk mendeteksi target positif secara menyeluruh masih perlu ditingkatkan.

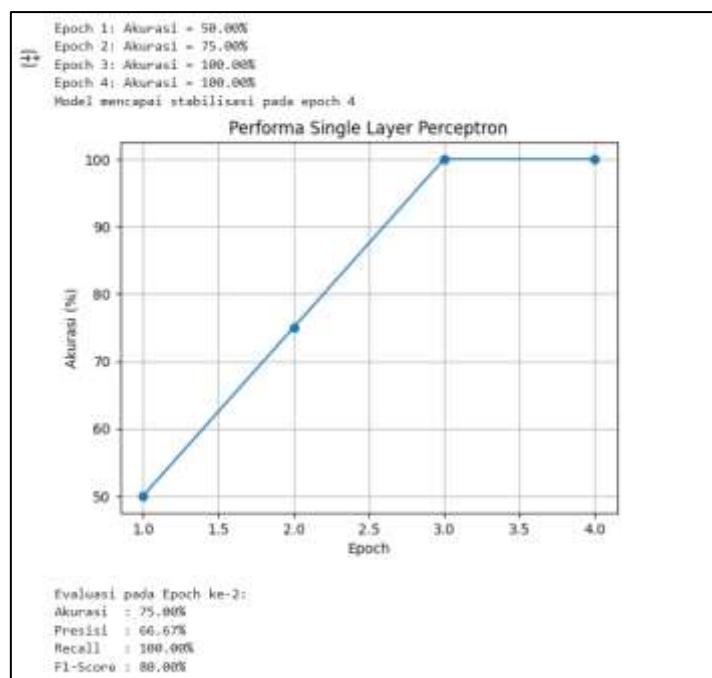
Setelah pola berhasil dikenali, tahap selanjutnya adalah melakukan pengujian algoritma dengan menggunakan Google Colab.

| Confusion Matrix: | | | | | |
|-------------------|-----------|--------|----------|---------|--|
| [[2 0] [1 1]] | | | | | |
| | precision | recall | f1-score | support | |
| 0 | 0.67 | 1.00 | 0.80 | 2 | |
| 1 | 1.00 | 0.50 | 0.67 | 2 | |
| accuracy | | | 0.75 | 4 | |
| macro avg | 0.83 | 0.75 | 0.73 | 4 | |
| weighted avg | 0.83 | 0.75 | 0.73 | 4 | |

Gambar 4. Hasil Pengujian Accuracy, Precision, Recall, dan F1-Score

Sesuai dengan perhitungan *confusion matrix* yang telah dilakukan, didapatkan hasil akurasi sebesar 0,75 atau 75%, presisi sebesar 1,0 atau 100%, *recall* sebesar 0,5 atau 50%, dan *F1-Score*

sebesar 0,67 atau 67%. Dibutuhkan hasil dari evaluasi akhir untuk membuktikan bahwa hasil keseluruhan perhitungan sudah dilakukan sesuai dengan rumus.



Gambar 5. Hasil Evaluasi Akhir

Hasil eksperimen menunjukkan bahwa model *Single Layer Perceptron* (SLP) mampu mencapai akurasi sebesar 75% pada epoch ke-2, dengan nilai presisi sebesar 100%, *recall* sebesar 50%, dan *F1-score* sebesar 66,67%. Meskipun jumlah epoch relatif sedikit, hal ini dapat dijelaskan melalui karakteristik dataset yang sederhana dan jumlah fitur yang terbatas, sehingga memudahkan proses penyesuaian bobot pada tahap awal pelatihan.

Selain itu, penggunaan fungsi aktivasi *step function* yang bersifat deterministik memungkinkan model melakukan klasifikasi secara langsung tanpa proses aktivasi *non-linear* bertingkat. Kecepatan konvergensi ini juga didukung oleh ukuran dataset yang kecil, yang mempercepat proses propagasi dan koreksi bobot. Dengan demikian, stabilisasi bobot dapat tercapai lebih awal dibandingkan model jaringan syaraf tiruan yang lebih kompleks, yang umumnya membutuhkan jumlah epoch lebih banyak untuk mencapai kestabilan.

V. KESIMPULAN DAN SARAN

Penggunaan model *Single Layer Perceptron* (SLP) dalam klasifikasi biner data kuesioner pelamar asisten laboratorium komputer menunjukkan hasil yang menjanjikan dalam mengotomatisasi proses seleksi berdasarkan pola dalam data. Dengan menggunakan parameter *learning rate* 0,1 dan *threshold* 0,5, model mampu mencapai akurasi sebesar 75% hanya dalam dua epoch, dengan presisi sebesar 100%, *recall* sebesar 50%, dan *F1-score* sebesar 66,67%.

Meskipun jumlah epoch yang dibutuhkan tergolong sedikit, hal ini disebabkan oleh karakteristik data yang sederhana, jumlah fitur yang terbatas, serta penggunaan fungsi aktivasi *step* yang memungkinkan proses klasifikasi berjalan cepat dan langsung. Proses penyesuaian bobot terjadi lebih awal karena data yang relatif linear dan ukuran dataset yang kecil, sehingga model cepat mencapai stabilitas. Namun, *recall* yang rendah menunjukkan bahwa SLP masih mengalami keterbatasan dalam mendeteksi seluruh kasus positif. Oleh karena itu, meskipun SLP cukup efektif sebagai metode awal dalam klasifikasi berbasis kecerdasan buatan, peningkatan performa dapat dicapai melalui penggunaan model yang lebih kompleks seperti *Multi-Layer Perceptron* (MLP), yang mampu mengenali pola non-linear dan menangani data dengan kompleksitas lebih tinggi.

Selain itu, optimasi *hyperparameter*, pemilihan fungsi aktivasi yang lebih sesuai, serta teknik prapemrosesan seperti normalisasi dan penyeimbangan kelas, dapat meningkatkan kinerja klasifikasi

secara keseluruhan. Penelitian mendatang juga dapat mengeksplorasi pendekatan berbasis *Deep Learning* atau *ensemble learning* guna membangun sistem seleksi yang lebih akurat dan andal.

DAFTAR PUSTAKA

- [1] Pratama, H. K. (2020). *Analisis Perbandingan Pengenalan Tanda Tangan Dengan Menggunakan Metode Perceptron Dan Backpropagation*. Universitas Islam Negeri Syarif Hidayatullah Jakarta.
- [2] Yuhandri, M. H., & Mayola, L. (2023). Identifikasi Pola Seleksi Penentuan Calon Wali Nagari dengan Menggunakan Artificial Neural Network Algoritma Perceptron. *Jurnal KomtekInfo*, 10(4), 158–165. <https://doi.org/10.35134/komtekinfo.v10i4.485>.
- [3] Kahar, N., & Aritonang, W. (2022). Implementasi Jaringan Syaraf Tiruan Dengan Algoritma Perceptron Dalam Penentuan Program Studi Mahasiswa Baru. *LP2M STMIK Nurdin Hamzah Jambi*, 14(2), 74–80. <https://doi.org/10.53564/akademika.v14i2.864>.
- [4] Kokyay, S., Kilinc, E., Uysal, F., Kurt, H., Celik, E., & Dugenci, M. (2020). A prediction model of artificial neural networks in development of thermoelectric materials with innovative approaches. *Engineering Science and Technology, an International Journal*, 23(6), 1476–1485. <https://doi.org/10.1016/j.jestch.2020.04.007>.
- [5] Al-afghoni, J. M. H. Y., Setiawan, W., Dwi, Y., Negara, P., Informasi, S., Madura, U. T., Matrix, C., Tree, D., & Rusdiana, L. (2025). *Klasifikasi Jenis Benih Kacang Menggunakan Smote Dan Decision Tree C4 . 5*. 9(1), 462–469.
- [6] Bhakti, I. N., Sholikhin, A. Z., Abi Lukman, M., Daniati, E., & Ristyawan, A. (2024). Klasifikasi Kategori Berita Menggunakan Naive Bayes. *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, 8(2), 1155–1164.
- [7] Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., & Caicedo, O. M. (2020). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9(16), 1–99. <https://doi.org/10.1186/s13174-018-0087-2>.
- [8] Medeiros, M. C., Varella, W. A., Viana, G. D., Mendes, A., & Neto, O. (2023). *Implementation Of An Algorithm For Binary Classification Of Handwritten Digits Using Scilab Implementation Of An Algorithm For Binary Classification. October*. <https://doi.org/10.5281/zenodo.10051944>.
- [9] Sari, N. R., & Mar'atullatifah, Y. (2023). Penerapan Multilayer Perceptron Untuk Identifikasi Kanker Payudara. *Jurnal Cakrawala Ilmiah*, 2(8), 3261–3268. <https://doi.org/10.31862/9785426311961>.
- [10] Budiman, Y. U. (2022). Identifikasi Kata Benda Dan Bukan Kata Benda Menggunakan Single Layer Perceptron Network. *BULLET : Jurnal Multidisiplin Ilmu*, 1(5), 759–768.
- [11] Teixeira Zavadzki de Pauli, S., Kleina, M., & Bonat, W. H. (2020). Comparing Artificial Neural Network Architectures for Brazilian Stock Market Prediction. *Annals of Data Science*, 7(4), 613–628. <https://doi.org/10.1007/s40745-020-00305-w>.
- [12] Rahmadani, F., Pardede, A. M. H., & Nurhayati. (2021). Jaringan Syaraf Tiruan Prediksi Jumlah Pengiriman Barang Menggunakan Metode Backpropagation (Studi Kasus: Kantor Pos Binjai). *Jurnal Teknik Informatika Kaputama (JTIK)*, 5(1), 100–106.
- [13] Kusuma, D. T., Siswipraptini, P. C., & Sangadji, I. (2022). *Sistem Syaraf Buatan Mengola Data Dengan Sistem Cerdas*. Universitas Brawijaya Press.
- [14] Rahmadani, F., Pardede, A. M. H., & Nurhayati. (2021). Jaringan Syaraf Tiruan Prediksi Jumlah Pengiriman Barang Menggunakan Metode Backpropagation (Studi Kasus: Kantor Pos Binjai). *Jurnal Teknik Informatika Kaputama (JTIK)*, 5(1), 100–106.
- [15] Afriyanti, L. (2020). *Rancang Bangun Tool Untuk Jaringan Syaraf Tiruan (Jst) Model Perceptron*. Universitas Islam Negeri Jakarta.
- [16] Rahmadani, F., Pardede, A. M. H., & Nurhayati. (2021). Jaringan Syaraf Tiruan Prediksi Jumlah Pengiriman Barang Menggunakan Metode Backpropagation (Studi Kasus: Kantor Pos Binjai). *Jurnal Teknik Informatika Kaputama (JTIK)*, 5(1), 100–106.
- [17] Siswipraptini, P. C., Ruli, R., Siregar, A., Iriansyah, ;, Sangadji, B. M., Annisa, ;, & Wahyulia, S. (2022). Algoritma Perceptron Menggunakan Teknik Machine Learning Untuk Model Smart Distribution Beban Listrik. *Jurnal Ilmiah*, 14(2), 150–159.

- [18] Prasetyo, S. Y., & Nabiilah, G. Z. (2023). Perbandingan Model Machine Learning pada Klasifikasi Tumor Otak Menggunakan Fitur Discrete Cosine Transform. *Jurnal Teknologi Terpadu*, 9(1), 29–34. <https://doi.org/10.54914/jtt.v9i1.605>.
- [19] Purwanto, A., & Nugroho, H. W. (2020). Analisa Perbandingan Kinerja Algoritma C4.5 Dan Algoritma K-Nearest Neighbors Untuk Klasifikasi Penerima Beasiswa. *Jurnal Teknoinfo*, 14(1), 236–243. <https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/article/view/336/329>.
- [20] Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 5(2), 697–711.
- [21] Dwi Fasnuari, H. A., Yuana, H., & Chulkamdi, M. T. (2022). Penerapan Algoritma K-Nearest Neighbor Untuk Klasifikasi Penyakit Diabetes Melitus. *Antivirus : Jurnal Ilmiah Teknik Informatika*, 16(2), 133–142. <https://doi.org/10.35457/antivirus.v16i2.2445>.
- [22] Putri, I. P. (2021). Analisis Performa Metode K-Nearest Neighbor (KNN) dan Crossvalidation pada Data Penyakit Cardiovascular. *Indonesian Journal Of Data And Science*, 2(1), 21–28.
- [23] Muslihati, M., Sahibu, S., & Taufik, I. (2024). Implementasi Algoritma Convolutional Neural Network untuk Klasifikasi Jenis Sampah Organik dan Non Organik. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(3), 840–852. <https://doi.org/10.57152/malcom.v4i3.1346>.
- [24] Wahyudi, R., Orisa, M., & Vendyansyah, N. (2021). Penerapan Algoritma K-Nearest Neighbors Pada Klasifikasi Penentuan Gizi Balita (Studi Kasus Di Posyandu Desa Bluto). *JATI (Jurnal Mahasiswa Teknik Informatika)*, 5(2), 750–757.