
Penerapan Metode McCulloch-Pitts Menggunakan Python Untuk Pengujian Pengenalan Pola Operator AND, Operator OR, Operator XOR Pada Fungsi Logika

Rana Sulthanah¹, Nazaruddin Ahmad^{2*}

^{1,2}Universitas Islam Negeri Ar-Raniry Banda Aceh, Fakultas Sains dan Teknologi, Teknologi Informasi, Banda Aceh, Indonseia

Informasi Artikel

Diterima: 28-11-2023

Direvisi: 15-12-2023

Diterbitkan: 2512-2023

Kata Kunci

Jaringan Saraf; McCulloch-Pitts; logic; Python

***Email Korespondensi:**

nazar.ahmad@ar-raniry.ac.id

Abstrak

Jaringan saraf tiruan (JST) adalah bagian dari kecerdasan buatan yang menerapkan konsep cara kerja jaringan saraf biologis pada manusia. Keunggulan utama JST terletak pada kemampuannya dalam mengenali pola tertentu, seperti pola notasi fungsi logika yang diuji dalam penelitian ini menggunakan cara kerja McCulloch-Pitts. Pengujian pengenalan pola fungsi logika, yang menggunakan operator AND, operator OR, dan operator XOR, melibatkan variabel input: nilai X_1 , X_2 , dan nilai input lainnya (X_n), serta nilai target (t) yang akan disesuaikan dengan bentuk tabel kebenaran dari masing-masing fungsi logika. Hasil pengujian menunjukkan adanya kesesuaian proses pengujian menggunakan konsep jaringan saraf tiruan dengan pola fungsi logika. Implementasi pengujian menggunakan bahasa pemrograman Python yang dapat menghasilkan kesesuaian bentuk pola dari fungsi logika. Proses pengujian dilakukan dengan menentukan nilai input untuk setiap variabel yang digunakan pada fungsi logika, memasukkan nilai bobot (W) untuk setiap nilai input, menentukan nilai *threshold* (ambang batas) yang ditetapkan secara langsung sehingga mendapatkan hasil yang sesuai antara output dengan target.

Abstract

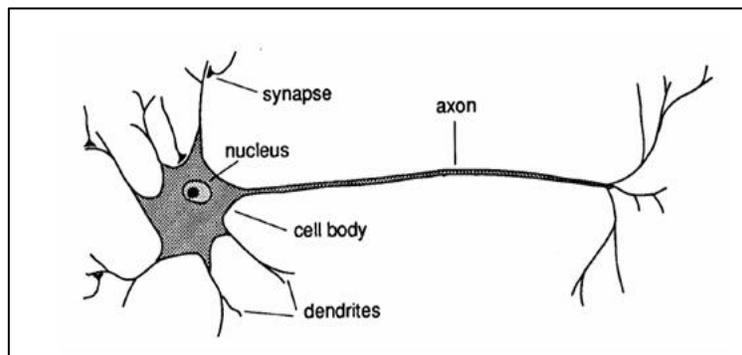
Artificial neural networks (ANNs) are a part of artificial intelligence that applies the concept of how the biological neural network in humans operates. The primary advantage of ANNs lies in their ability to recognize specific patterns, such as the pattern notation of logic functions tested in this study using the McCulloch-Pitts model. Testing the pattern recognition of logic functions, involving AND, OR, and XOR operators, entails input variables: values X_1 , X_2 , and other input values (X_n), as well as target values (t) that match the truth table forms of each logic function. The test results indicate the alignment of the testing process using the concept of artificial neural networks with the patterns of logic functions. The implementation of the test is carried out using the Python programming language, which can produce conformity in the pattern forms of logic functions. The testing process is conducted by determining input values for each variable used in the logic functions, inputting weights (W) for each input value, and setting a threshold value directly to obtain results that align the output with the target.

1. Pendahuluan

Penerapan teknologi informasi sudah sangat merata, pemanfaatan teknologi informasi sudah digunakan untuk kegiatan yang kompleks pada segala bidang dengan pemanfaatan metode dan konsep teknologi informasi yang mampu memberikan manfaat yang besar (Roza et al., 2023). Salah satunya adalah pemanfaatan konsep Jaringan saraf tiruan yang merupakan bagian dari kecerdasan buatan. Jaringan saraf tiruan adalah suatu metode komputasi yang digunakan untuk melakukan pemrosesan informasi dengan mengadopsi jaringan syaraf biologis sehingga menghasilkan pengetahuan yang cerdas seperti otak manusia (Ahmad et al., 2023). Jaringan saraf tiruan juga dapat diartikan sebagai suatu sistem pemrosesan informasi yang meniru cara kerja jaringan saraf manusia. Jaringan saraf tiruan digunakan sebagai penerapana model matematis secara umum yang berasal dari pemahaman manusia (*human cognition*) dalam memodelkan suatu permasalahan (Sudarsono, 2016).

Jaringan saraf tiruan tersusun dari sejumlah besar elemen yang melakukan kegiatan yang analog dengan fungsi-fungsi biologis neuron yang paling elementer. Elemen-elemen ini terorganisasi sebagaimana layaknya anatomi otak sehingga mampu menyelesaikan proses perhitungan dengan model pembelajaran (Ayu, 2019).

Jaringan saraf tiruan dapat belajar dari pengalaman, melakukan generalisasi atas contoh-contoh yang diperolehnya dan mengabstraksi karakteristik esensial input bahkan untuk data yang tidak relevan. Struktur jaringan saraf tiruan pada manusia dapat dilihat pada gambar 1 berikut ini:



Gambar 1. Struktur jaringan saraf biologis manusia

Jaringan saraf pada manusia terdiri atas jutaan sel saraf yang saling terhubung. Manusia dapat bergerak dan memerintahkan anggota tubuh karena adanya pemberian informasi kepada setiap jaringan sel saraf pada tubuh manusia. Proses kegiatan inilah yang menjadi referensi jaringan saraf tiruan dapat bekerja layaknya jaringan saraf manusia. Pada jaringan saraf ada yang namanya *dendrit*, *cell body*, dan *axon*. *Dendrit* dapat diartikan sebagai bagian yang menerima informasi dari luar tubuh, maka *dendrit* ini dapat diartikan sebagai bagian input dari jaringan saraf tiruan. *Cell body* merupakan bagian yang mengolah informasi atau melakukan pemrosesan setelah dilakukan input. *Axon* merupakan bagian yang mengirimkan sinyal kepada sel saraf lain, pada jaringan saraf tiruan *axon* ini adalah bagian output yang dapat menghasilkan informasi.

Jaringan saraf tiruan sebagai salah satu alat dalam pembelajaran mesin, memiliki peran penting dalam pengenalan pola dengan kemampuannya mengklasifikasikan data, mengenali pola yang kompleks, dan mengekstraksi informasi yang bermanfaat dari data yang rumit. Keterkaitan antara pengenalan pola dan JST menjadi bagian krusial dalam pengembangan kecerdasan buatan dan aplikasi-aplikasi praktisnya (Bishop, 2006). Konsep jaringan saraf tiruan ini sudah digunakan dalam berbagai proses pengujian seperti pengenalan pola nomor kendaraan bermotor berdasarkan wilayah menggunakan jaringan saraf tiruan (Maryana et al., 2018). Pembuatan aplikasi untuk identifikasi nomor kendaraan bermotor menggunakan konsep pengolahan citra dan jaringan saraf tiruan dengan metode backpropagation (Noprizal & Candra,

2019). Melakukan analisis prakiraan cuaca menggunakan jaringan saraf tiruan dengan metode backpropagation (Intan et al., 2021).

Jaringan Syaraf Tiruan (JST) merupakan cabang dari ilmu *artificial intelligence* (kecerdasan buatan). Model yang menjadi dasar penggunaan pengujian pengenalan pola pada jaringan saraf tiruan adalah cara kerja McCulloch-Pitts. Pendekatan McCulloch-Pitts melibatkan proses *learning* (pembelajaran) yang diawasi dalam konsep jaringan saraf. Saat mendesain jaringan neuron, hal utama yang diperhatikan adalah jumlah kebutuhan yang akan diinventarisir (Aziz et al., 2022). Salah satu penerapan konsep jaringan saraf tiruan adalah mengimplementasikan pada pengenalan pola fungsi logika dengan menerapkan gerbang logika secara praktis menggunakan model neuron McCulloch-Pitts.

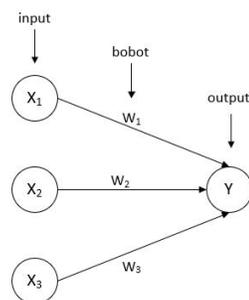
Dalam artikel ini akan dilakukan implementasi pengujian pengenalan pola fungsi logika menggunakan bahasa pemrograman Python untuk melihat kesesuaian nilai hasil uji dengan bentuk tabel kebenaran pada fungsi logika yang menggunakan operator AND, operator OR, dan operator XOR. Python merupakan bahasa pemrograman yang muncul pada tahun 1995, dan dapat dikatakan bahasa pemrograman yang baru hadir dibandingkan dengan bahasa pemrograman C, C++, Java. Mengalami perkembangan versi dari Python hingga tahun 2017 dan sampai sekarang. Salah satu keunggulannya adalah Python merupakan bahasa pemrograman yang *open source* dengan sebagian besar versinya menggunakan lisensi *GFL-compatible* (Wahyono, 2018). MATLAB juga merupakan bahasa pemrograman yang dapat digunakan untuk menerapkan konsep jaringan saraf tiruan dengan berbagai fasilitas yang disediakan. Tetapi Python memiliki beberapa keunggulan seperti *readability* yaitu kemudahan untuk dibaca dan dipahami, *efisien* yaitu memiliki *library* yang lengkap dan mudah didapatkan, *multifungsi* yaitu dapat digunakan untuk berbagai keperluan seperti membuat website, membuat aplikasi jaringan, membuat aplikais robotika, dan membuat aplikasi kecerdasan buatan, *interoperabilitas* yaitu mampu berinteraksi dengan bahasa pemrograman lainnya, dan *dukungan komunitas* yaitu adanya proses diskusi secara langsung dalam pengembangan bahasa pemrograman Python.

2. Metode Penelitian

Metode penelitian yang digunakan adalah metode eksperimental, dengan langkah-langkah yang telah dirancang untuk mengamati input dan output neuron dalam lingkungan kontrol yang dibuat secara khusus. Adapun tahapan yang dilakukan adalah sebagai berikut:

2.1 Arsitektur Jaringan Saraf Tiruan

Di dalam jaringan saraf tiruan terdapat pola hubungan antar neuron di dalam sistem yang disebut dengan arsitektur jaringan. Bentuk arsitektur jaringan inilah yang menggambarkan pola data yang akan diproses menggunakan model jaringan saraf tiruan. Terdapat 3 hal yang menentukan proses di dalam *jaringan* saraf tiruan yaitu bentuk keterhubungan antar *neuron* (arsitektur jaringan), cara kerja yang digunakan dalam menentukan nilai bobot untuk menghubungkan (metode *pelatihan/pembelajaran/algorithm*) dan fungsi aktivasi (Arifin et al., 2018). Dapat dilihat pada gambar 2 berikut ini:



Gambar 2. Arsitektur jaringan saraf tiruan

Dari gambar 2 di atas, data di input melalui neuron dilambangkan dengan X_1, X_2, X_3 dengan setiap neuron memiliki bobot masing-masing yaitu W_1, W_2, W_3 yang akan menghasilkan output neuron Y . Proses penghitungan neuron yang diinput akan dikalikan dengan bobot dan dijumlahkan untuk setiap neuron input yang akan menghasilkan nilai *net*. Bentuk persamaannya adalah sebagai berikut:

$$Y = net = X_1.W_1 + X_2.W_2 + X_3.W_3 \quad (1)$$

Nilai input neuron bisa saja lebih dari 3 nilai input sehingga dapat menjadi neuron n (X_n) dan juga memiliki bobot n (bobot_n), sehingga bentuk persamaannya sebagai berikut:

$$Y = net = X_1.W_1 + X_2.W_2 + X_3.W_3 + \dots + X_n.W_n \quad (2)$$

2.2 Fungsi Aktivasi

Pada jaringan saraf tiruan terdapat sebuah fungsi yang disebut dengan fungsi aktivasi. Fungsi aktivasi adalah fungsi yang digunakan untuk mengaktifkan neuron atau tidak mengaktifkan neuron (Julpan et al., 2015). Terdapat 2 jenis fungsi aktivasi yaitu fungsi aktivasi biner dan fungsi aktivasi bipolar. Fungsi aktivasi biner mempunyai nilai range yang akan menjadi output yaitu 0 sampai 1. Bentuk fungsi aktivasi biner dapat dilihat dengan bentuk definisi sebagai berikut:

$$Y = f(net) = \begin{cases} 1, & \text{jika } net \geq 0 \\ 0, & \text{jika } net < 0 \end{cases} \quad (3)$$

Sedangkan fungsi aktivasi bipolar mempunyai nilai range yang akan menjadi output yaitu 1 sampai -1. Bentuk fungsi aktivasi bipolar dapat dilihat dengan bentuk definisi sebagai berikut:

$$Y = f(net) = \begin{cases} 1, & \text{jika } net > 0 \\ 0, & \text{jika } net = 0 \\ -1, & \text{jika } net < 0 \end{cases} \quad (4)$$

Dengan menggunakan fungsi aktivasi maka proses penghitungan menggunakan model arsitektur jaringan dapat dituliskan ke dalam bentuk persamaan sebagai berikut :

$$Y = f(net) = X_1.W_1 + X_2.W_2 + X_3.W_3 + \dots + X_n.W_n \quad (5)$$

Pada proses pengenalan pola yang dilakukan, jika tidak terdapat kesamaan pola maka dapat digunakan nilai *threshold* (ambang batas) yang dilakukan secara analitik. Artinya nilai *threshold* dapat ditentukan berapa saja sampai mendapatkan kesamaan pola atau mendapatkan nilai $f(net)$ yang sama dengan nilai yang diuji pada sebuah pola.

2.3 Penggunaan Microsoft excel menyusun tabel kebenaran untuk melihat pola

Operasi fungsi logika digunakan didalam pengujian pengenalan pola ini adalah menggunakan operator logika AND, operator logika OR, dan operator logika XOR. Proses pengujian akan melihat kepada bentuk tabel kebenaran dari ketiga fungsi logika tersebut. Bentuk tabel kebenarannya dapat dilihat pada tabel 1, tabel 2, dan tabel 3 di bawah ini.

Tabel 1. Tabel kebenaran logika AND dengan 2 variabel

X1	X2	X1 AND X2
1	1	1
1	0	0
0	1	0
0	0	0

Tabel 2. Tabel kebenaran logika OR dengan 2 variabel

X1	X2	X1 OR X2
1	1	1
1	0	1
0	1	1
0	0	0

Tabel 3. Tabel kebenaran logika XOR dengan 2 variabel

X1	X2	X1 XOR X2
1	1	0
1	0	1
0	1	1
0	0	0

3. Hasil dan Pembahasan

Penerapan metode McCulloch-Pitts pada jaringan saraf tiruan untuk pengenalan pola dapat dilihat pada beberapa percobaan yang dilakukan berikut ini:

3.1 Fungsi Logika AND dengan 2 variabel masukan

Tabel 5. Tabel pengujian logika AND dengan 2 variabel

X1	X2	Target
1	1	1
1	0	0
0	1	0
0	0	0

Dalam melakukan pengujian pengenalan pola pada tabel kebenaran logika AND dengan 2 variabel seperti pada tabel 1, maka disusun tabel pengujian logika AND seperti pada tabel 5 dimana hasil X1 AND X2 dijadikan sebagai target pengujian pengenalan pola yang akan dicocokkan dengan tabel 1. Untuk setiap bobot (W) masing-masing neuron ($X1$ dan $X2$) diberikan nilai 1.

Untuk mendapatkan hasil perhitungan untuk tabel 5 maka gunakan rumus pada persamaan 1 dengan bentuknya sebagai berikut:

$$\text{baris1} = (X1 * W1) + (X2 * W2)$$

$$\text{baris2} = (X1 * W1) + (X2 * W2)$$

$$\text{baris3} = (X1 * W1) + (X2 * W2)$$

$$\text{baris4} = (X1 * W1) + (X2 * W2)$$

Hasil dari bentuk pembelajaran pola diatas untuk mendapatkan nilai *net* adalah sebagai berikut:

$$\text{Net1} = (1*1)+(1*1) = 2$$

$$\text{Net2} = (1*1)+(0*1) = 1$$

$$\text{Net3} = (0*1)+(1*1) = 1$$

$$\text{Net4} = (0*1)+(0*1) = 0$$

Hasil perhitungannya dapat dilihat pada tabel 6 berikut ini:

Tabel 6. Hasil perhitungan nilai net untuk fungsi logika AND dengan 2 variabel

X1	X2	Target	net
1	1	1	2
1	0	0	1
0	1	0	1
0	0	0	0

Selanjutnya dilakukan proses aktivasi menggunakan fungsi aktivasi untuk melihat kesamaan pola antara target dengan output (Y) seperti persamaan 3.

Hasil perhitungan menggunakan metode McCulloch-Pitts dapat dilihat pada tabel 7 berikut ini:

Tabel 7. Hasil perhitungan nilai net untuk fungsi logika AND dengan 2 variabel

X1	X2	Target	$net = \sum_{i=1} X_{ij} \cdot W_{ij}$	$Y = f(net) = \begin{cases} 1, & \text{jika } net \geq 0 \\ 0, & \text{jika } net < 0 \end{cases}$	Hasil
1	1	1	$(1*1)+(1*1) = 2$	1	Sesuai
1	0	0	$(1*1)+(0*1) = 1$	1	Tidak Sesuai
0	1	0	$(0*1)+(1*1) = 1$	1	Tidak Sesuai
0	0	0	$(0*1)+(0*1) = 0$	1	Tidak Sesuai

Karena hasil dengan fungsi aktivasi yang digunakan tidak sesuai hasilnya dengan target, maka untuk fungsi aktivasi dapat menggunakan nilai *threshold* (ambang batas) yang dapat ditentukan langsung. Bentuk fungsi aktivasinya adalah sebagai berikut:

$$Y = f(net) = \begin{cases} 1, & \text{jika } net \geq \alpha \\ 0, & \text{jika } net < \alpha \end{cases} \quad (6)$$

dimana α adalah *threshold* (ambang batas).

Dilakukan pengujian pengenalan pola logika AND menggunakan nilai *threshold* pada fungsi aktivasi dengan nilai $\alpha=1$.

Tabel 8. Hasil perhitungan nilai net untuk fungsi logika AND dengan $\alpha = 1$

X1	X2	Target	$net = \sum_{i=1} X_{ij} \cdot W_{ij}$	$Y = f(net) = \begin{cases} 1, & \text{jika } net \geq 1 \\ 0, & \text{jika } net < 1 \end{cases}$	Hasil
1	1	1	$(1*1)+(1*1) = 2$	1	Sesuai
1	0	0	$(1*1)+(0*1) = 1$	1	Tidak Sesuai
0	1	0	$(0*1)+(1*1) = 1$	1	Tidak Sesuai
0	0	0	$(0*1)+(0*1) = 0$	0	Sesuai

Dari hasil pengujian pengenalan pola dengan menggunakan nilai *threshold*=1 pada tabel 8, didapatkan hasil yang tidak sesuai antara nilai target dengan nilai $f(net)$ sehingga hasilnya adalah pengenalan pola fungsi logika AND tidak dikenali.

Dilakukan pengujian pengenalan pola logika AND menggunakan nilai *threshold* pada fungsi aktivasi dengan nilai $\alpha=2$.

Tabel 9. Hasil perhitungan nilai net untuk fungsi logika AND dengan $\alpha = 2$

X1	X2	Target	$net = \sum_{i=1}^2 X_{ij} \cdot W_{ij}$	$Y = f(net) = \begin{cases} 1, & \text{jika } net \geq 2 \\ 0, & \text{jika } net < 2 \end{cases}$	Hasil
1	1	1	$(1*1)+(1*1) = 2$	1	Sesuai
1	0	0	$(1*1)+(0*1) = 1$	0	Sesuai
0	1	0	$(0*1)+(1*1) = 1$	0	Sesuai
0	0	0	$(0*1)+(0*1) = 0$	0	Sesuai

Dari hasil pengujian pengenalan pola dengan menggunakan nilai *threshold*=2 pada tabel 9, didapatkan hasil yang sesuai antara nilai target dengan nilai $f(net)$ sehingga hasilnya adalah pengenalan pola fungsi logika AND dikenali.

3.2 Fungsi Logika OR dengan 2 variabel masukan

Tabel 10. Tabel pengujian logika OR dengan 2 variabel

X1	X2	Target
1	1	1
1	0	1
0	1	1
0	0	0

Dalam melakukan pengujian pengenalan pola pada tabel kebenaran logika OR dengan 2 variabel seperti pada tabel 2, maka disusun tabel pengujian logika OR seperti pada tabel 10 dimana hasil X1 OR X2 dijadikan sebagai target pengujian pengenalan pola yang akan dicocokkan dengan tabel 2. Untuk setiap bobot (W) masing-masing neuron (X1 dan X2) diberikan nilai 1.

Untuk mendapatkan hasil perhitungan untuk tabel 10 maka gunakan rumus pada persamaan 1 dengan bentuknya sebagai berikut:

$$\text{baris1} = (X1*W1) + (X2*W2)$$

$$\text{baris2} = (X1*W1) + (X2*W2)$$

$$\text{baris3} = (X1*W1) + (X2*W2)$$

$$\text{baris4} = (X1*W1) + (X2*W2)$$

Hasil dari bentuk pembelajaran pola diatas untuk mendapatkan nilai *net* adalah sebagai berikut:

$$\text{Net1} = (1*1)+(1*1) = 2$$

$$\text{Net2} = (1*1)+(0*1) = 1$$

$$\text{Net3} = (0*1)+(1*1) = 1$$

$$\text{Net4} = (0*1)+(0*1) = 0$$

Hasil perhitungannya dapat dilihat pada tabel 11 berikut ini:

Tabel 11. Hasil perhitungan nilai net untuk fungsi logika OR dengan 2 variabel

X1	X2	Target	net
1	1	1	2
1	0	1	1
0	1	1	1
0	0	0	0

Selanjutnya dilakukan proses aktivasi menggunakan fungsi aktivasi untuk melihat kesamaan pola antara target dengan output (Y) seperti persamaan 3.

Hasil perhitungan menggunakan metode McCulloch-Pitts adalah sebagai berikut:

Tabel 12. Hasil perhitungan nilai net untuk fungsi logika OR dengan 2 variabel

X1	X2	Target	$net = \sum_{i=1}^1 X_{ij} \cdot W_{ij}$	$Y = f(net) = \begin{cases} 1, & \text{jika } net \geq 0 \\ 0, & \text{jika } net < 0 \end{cases}$	Hasil
1	1	1	$(1*1)+(1*1) = 2$	1	Sesuai
1	0	1	$(1*1)+(0*1) = 1$	1	Sesuai
0	1	1	$(0*1)+(1*1) = 1$	1	Sesuai
0	0	0	$(0*1)+(0*1) = 0$	1	Tidak Sesuai

Dari hasil pengujian pengenalan pola dengan menggunakan fungsi aktivasi persamaan 3 pada tabel 12, didapatkan hasil yang tidak sesuai antara nilai target dengan nilai $f(net)$ sehingga hasilnya adalah pengenalan pola fungsi logika OR tidak dikenali.

Dilakukan pengujian pengenalan pola logika OR menggunakan nilai *threshold* pada fungsi aktivasi dengan nilai $\alpha=1$.

Tabel 13. Hasil perhitungan nilai net untuk fungsi logika OR dengan $\alpha = 1$

X1	X2	Target	$net = \sum_{i=1}^1 X_{ij} \cdot W_{ij}$	$Y = f(net) = \begin{cases} 1, & \text{jika } net \geq 1 \\ 0, & \text{jika } net < 1 \end{cases}$	Hasil
1	1	1	$(1*1)+(1*1) = 2$	1	Sesuai
1	0	1	$(1*1)+(0*1) = 1$	1	Sesuai
0	1	1	$(0*1)+(1*1) = 1$	1	Sesuai
0	0	0	$(0*1)+(0*1) = 0$	0	Sesuai

Dari hasil pengujian pengenalan pola dengan menggunakan nilai *threshold*=1 pada tabel 13, didapatkan hasil yang sesuai antara nilai target dengan nilai $f(net)$ sehingga hasilnya adalah pengenalan pola fungsi logika OR dikenali.

3.3 Fungsi Logika XOR dengan 2 variabel masukan

Tabel 14. Tabel pengujian logika OR dengan 2 variabel

X1	X2	Target
1	1	0
1	0	1
0	1	1
0	0	0

Dalam melakukan pengujian pengenalan pola pada tabel kebenaran logika XOR dengan 2 variabel seperti pada tabel 3, maka disusun tabel pengujian logika OR seperti pada tabel 14 dimana hasil X1 XOR X2 dijadikan sebagai target pengujian pengenalan pola yang akan dicocokkan dengan tabel 3. Untuk setiap bobot (W) masing-masing neuron (X1 dan X2) diberikan nilai 1.

Untuk mendapatkan hasil perhitungan untuk tabel 14 maka gunakan rumus pada persamaan 1 dengan bentuknya sebagai berikut:

$$\text{baris1} = (X1 * W1) + (X2 * W2)$$

$$\text{baris2} = (X1 * W1) + (X2 * W2)$$

$$\text{baris3} = (X1 * W1) + (X2 * W2)$$

$$\text{baris4} = (X1 * W1) + (X2 * W2)$$

Hasil dari bentuk pembelajaran pola diatas untuk mendapatkan nilai *net* adalah sebagai berikut:

$$\text{Net1} = (1 * 1) + (1 * 1) = 2$$

$$\text{Net2} = (1 * 1) + (0 * 1) = 1$$

$$\text{Net3} = (0 * 1) + (1 * 1) = 1$$

$$\text{Net4} = (0 * 1) + (0 * 1) = 0$$

Hasil perhitungannya dapat dilihat pada tabel 15 berikut ini:

Tabel 15. Hasil perhitungan nilai net untuk fungsi logika XOR dengan 2 variabel

X1	X2	Target	net
1	1	0	2
1	0	1	1
0	1	1	1
0	0	0	0

Selanjutnya dilakukan proses aktivasi menggunakan fungsi aktivasi untuk melihat kesamaan pola antara target dengan output (Y) seperti persamaan 3.

Hasil perhitungan menggunakan metode McCulloch-Pitts adalah sebagai berikut:

Tabel 16. Hasil perhitungan nilai net untuk fungsi logika XOR dengan 2 variabel

X1	X2	Target	$net = \sum_{i=1} X_{ij} \cdot W_{ij}$	$Y = f(net) = \begin{cases} 1, & \text{jika } net \geq 0 \\ 0, & \text{jika } net < 0 \end{cases}$	Hasil
1	1	0	$(1 * 1) + (1 * 1) = 2$	1	Tidak Sesuai
1	0	1	$(1 * 1) + (0 * 1) = 1$	1	Sesuai
0	1	1	$(0 * 1) + (1 * 1) = 1$	1	Sesuai
0	0	0	$(0 * 1) + (0 * 1) = 0$	1	Tidak Sesuai

Dari hasil pengujian pengenalan pola dengan menggunakan fungsi aktivasi persamaan 3 pada tabel 16, didapatkan hasil yang tidak sesuai antara nilai target dengan nilai $f(net)$ sehingga hasilnya adalah pengenalan pola fungsi logika XOR tidak dikenali.

Dilakukan pengujian pengenalan pola logika XOR menggunakan nilai *threshold* pada fungsi aktivasi dengan nilai $\alpha=1$.

Tabel 17. Hasil perhitungan nilai net untuk fungsi logika XOR dengan $\alpha = 1$

X1	X2	Target	$net = \sum_{i=1}^1 X_{ij} \cdot W_{ij}$	$Y = f(net) = \begin{cases} 1, & \text{jika } net \geq 1 \\ 0, & \text{jika } net < 1 \end{cases}$	Hasil
1	1	0	$(1*1)+(1*1) = 2$	1	Tidak Sesuai
1	0	1	$(1*1)+(0*1) = 1$	1	Sesuai
0	1	1	$(0*1)+(1*1) = 1$	1	Sesuai
0	0	0	$(0*1)+(0*1) = 0$	0	Sesuai

Dari hasil pengujian pengenalan pola dengan menggunakan nilai $threshold=1$ pada tabel 17, didapatkan hasil yang tidak sesuai antara nilai target dengan nilai $f(net)$ sehingga hasilnya adalah pengenalan pola fungsi logika XOR dikenali.

Dilakukan pengujian pengenalan pola logika XOR menggunakan nilai $threshold$ pada fungsi aktivasi dengan nilai $\alpha=2$.

Tabel 18. Hasil perhitungan nilai net untuk fungsi logika XOR dengan $\alpha = 2$

X1	X2	Target	$net = \sum_{i=1}^2 X_{ij} \cdot W_{ij}$	$Y = f(net) = \begin{cases} 1, & \text{jika } net \geq 2 \\ 0, & \text{jika } net < 2 \end{cases}$	Hasil
1	1	0	$(1*1)+(1*1) = 2$	1	Tidak Sesuai
1	0	1	$(1*1)+(0*1) = 1$	0	Tidak Sesuai
0	1	1	$(0*1)+(1*1) = 1$	0	Tidak Sesuai
0	0	0	$(0*1)+(0*1) = 0$	0	Sesuai

Dari hasil pengujian pengenalan pola dengan menggunakan nilai $threshold=2$ pada tabel 18, didapatkan hasil yang tidak sesuai antara nilai target dengan nilai $f(net)$ sehingga hasilnya adalah pengenalan pola fungsi logika XOR tidak dikenali.

Maka gunakan cara berikut untuk menghasilkan tabel kebenaran X_1 XOR X_2 :

- Langkah 1 : negasikan nilai X_1
- Langkah 2 : negasikan nilai X_2
- Langkah 3 : konjungsikan X_1 dengan $\neg X_2$
- Langkah 4 : konjungsikan $\neg X_1$ dengan X_2
- Langkah 5 : Disjungsikan nilai yang didapatkan dari langkah 3 dan langkah 4.
- Langkah 6 : maka akan didapatkan nilai X_1 XOR X_2

Dari langkah-langkah diatas maka dapat dilihat hasilnya pada tabel 21 berikut ini:

Tabel 19. Tabel kebenaran fungsi logika XOR dengan 2 variabel

X_1	X_2	$\neg X_1$	$\neg X_2$	$X_1 \wedge \neg X_2$	$\neg X_1 \wedge X_2$	$(X_1 \wedge \neg X_2) \vee (\neg X_1 \wedge X_2)$	X_1 XOR X_2
1	1	0	0	0	0	0	0
1	0	0	1	1	0	1	1
0	1	1	0	0	1	1	1
0	0	1	1	0	0	0	0

Misalkan :

$$q_1 = X_1 \wedge \neg X_2$$

$$q_2 = \neg X_1 \wedge X_2$$

Nilai bobot (w_1) $X_1 = 1$

Nilai bobot (w_2) $X_2 = 1$

Threshold = 1

Maka yang digunakan adalah fungsi aktivasi q_1 dan q_2 :

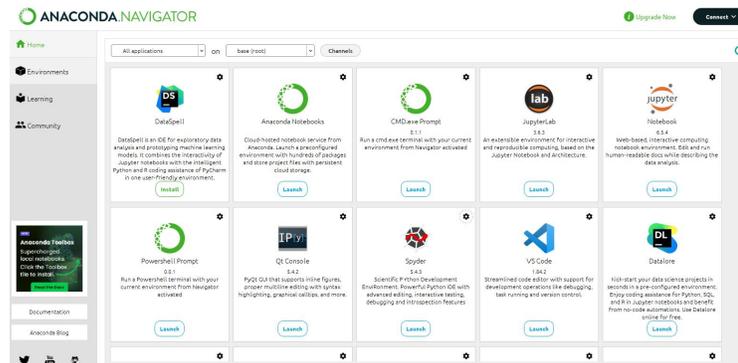
Tabel 20. Tabel pengujian fungsi logika XOR dengan 2 variabel

q_1	q_2	Target	$net = \sum_{i=1}^1 x_i \cdot w$	$y = f(net) = \begin{cases} 1, & \text{jika } net \geq 1 \\ 0, & \text{jika } net < 1 \end{cases}$	Hasil
0	0	0	$0.1 + 0.1 = 0$	0	Sesuai
1	0	1	$1.1 + 0.1 = 1$	1	Sesuai
0	1	1	$0.1 + 1.1 = 1$	1	Sesuai
0	0	0	$0.1 + 0.1 = 0$	0	Sesuai

Dari hasil pengujian pengenalan pola dengan menggunakan nilai $threshold=1$ pada tabel 20, didapatkan hasil yang sesuai antara nilai target dengan nilai $f(net)$ sehingga hasilnya adalah pengenalan pola fungsi logika XOR dikenali.

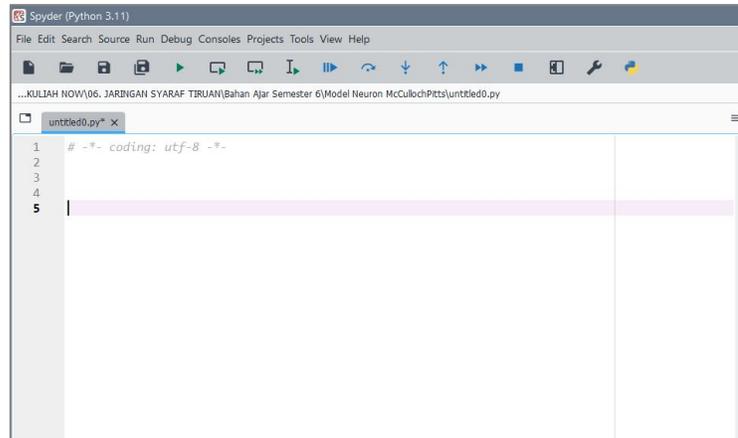
3.4 Implementasi dengan Python

Metode McCulloch-Pitts untuk pengenalan pola menggunakan konsep jaringan saraf tiruan ini dapat diimplementasikan menggunakan bahasa pemrograman Python. Langkah awal yang dilakukan adalah melakukan proses instalasi Anaconda Navigator pada komputer atau laptop anda. Setelah dilakukan proses instalasi maka akan muncul tampilan seperti gambar 3 berikut ini:



Gambar 3. Anaconda Navigator

Untuk mengetikkan sintaks Python maka digunakan teks editor Spyder yang memberikan kemudahan dalam pengetikan kode program Python. Dapat dilihat seperti gambar 4 berikut ini:



Gambar 4. Teks editor Spyder

Implementasi fungsi logika AND dengan Python

Mendeklarasikan nilai X1, X2 dan Output fungsi logika AND

```

import numpy as np
x1=np.array([1,1,0,0])
x2=np.array([1,0,1,0])
y_output=np.array([1,0,0,0])
  
```

Menentukan nilai bobot untuk masing-masing input (neuron) dan nilai *threshold*

```

w1=int(input('Masukkan Bobot 1 (w1)='))
w2=int(input('Masukkan Bobot 2 (w2)='))
theta=int(input('Masukkan Nilai Threshold (theta)='))
  
```

Menghitung nilai $f(net)$

```
fnet=x1*w1+x2*w2
```

Melakukan pengujian untuk nilai *threshold* (alpha)

```

T=alpha
def nilai_alpha(fnet:int,T:int)->int:
    if fnet >= T:
        return 1
    else:
        return 0
  
```

Menampilkan hasil pengujian pengenalan pola

```

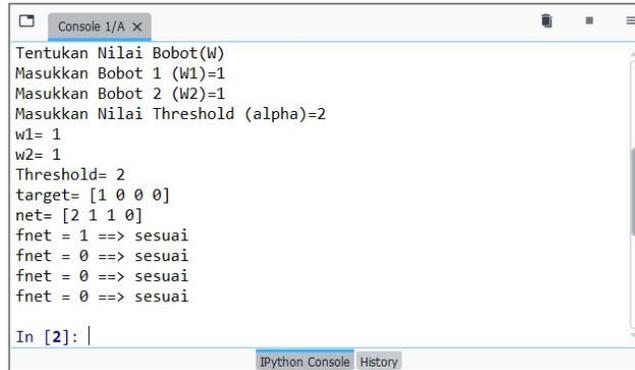
for i in range(0,4):
    aktivasi=nilai_alpha(fnet[i],T)
    if nilai_alpha(fnet[i],T) == y_output[i]:
  
```

```

print(f'fnet = {aktivasi}', '==>', "sesuai")
else:
print(f'fnet = {aktivasi}', '==>', "tidak sesuai")

```

Hasilnya adalah sebagai berikut:



Gambar 4. Tampilan console spyder untuk pengujian pola fungsi logika AND

Implementasi diatas dapat juga dilakukan untuk fungsi logika OR dan fungsi logika XOR sehingga didapatkan hasilnya adalah:

Tabel 21. Tabulasi data pengujian pengenalan pola fungsi logika

Pola Fungsi Logika	Nilai Pengenalan Pola	Tabel Kebenaran	Python
AND	Masukan Nilai Bobot (W) = 1 Nilai Alpha = 2	Sesuai	Sesuai
OR	Masukan Nilai Bobot (W) = 1 Nilai Alpha = 1	Sesuai	Sesuai
XOR	Masukan Nilai Bobot (W) = 1 Nilai Alpha = 1	Sesuai	Sesuai

Dari hasil pada tabel 21 diatas dapat disimpulkan bahwa pengenalan pola fungsi logika menggunakan operator AND, operator OR, dan operator XOR dapat dilakukan menggunakan bahasa pemrograman Python untuk konsep jaringan saraf tiruan menggunakan cara kerja McCulloch-Pitts. Sehingga bahasa pemrograman Python dapat digunakan untuk proses pengujian pengenalan pola pada jaringan saraf tiruan. Pengenalan pola dapat diterapkan pada objek berupa gambar (*image*) dimana yang menjadi proses pengenalan polanya adalah nilai piksel dari gambar yang menjadi nilai input harus dikonversi ke dalam bentuk biner (0 dan 1) yang akan diterjemahkan untuk membedakan mana yang menjadi latar belakang gambar, dan mana yang menjadi objek dari gambar (Widodo, 2019). Hal ini sejalan dengan konsep penerapan fungsi logika biner dalam mengenali pola dengan konsep jaringan saraf tiruan.

4. Kesimpulan

Proses pengenalan pola menggunakan jaringan saraf tiruan dapat diterapkan untuk pengenalan pola tabel kebenaran pada fungsi logika yang menggunakan operator AND, operator OR, dan operator XOR. Pengenalan pola pada jaringan saraf tiruan ini menggunakan metode yang sangat dasar yaitu metode McCulloch-Pitts. Hasilnya adalah metode McCulloch-Pitts dapat digunakan untuk melakukan pengujian pengenalan pola dengan memberikan nilai bobot (W) untuk setiap neuron (masukan) dan memberikan nilai *threshold*

(ambang batas) dengan metode analitik yaitu menentukan secara langsung nilai *threshold* sampai mendapatkan hasil yang sesuai. Pada fungsi logika AND dan OR dapat dilakukan secara langsung, tetapi pada fungsi logika XOR dibutuhkan hidden layer untuk mendapatkan fungsi logika yang tepat sehingga hasil pengujian pengenalan pola sesuai dengan target. Implementasi pengujian pengenalan pola fungsi logika ini dapat dilakukan menggunakan bahasa pemrograman Python sehingga akan memberikan hasil yang akurat untuk setiap pengujian pengenalan pola. Tentunya implementasi menggunakan bahasa pemrograman Python dapat juga diterapkan untuk metode lainnya yang ada pada jaringan saraf tiruan.

5. Referensi

- Ahmad, N., Efriyanti, L., Efendi, Y., Yahya, S. R., Ningsih, S. R., S, W., Sena, B., Adriyendi, A., Umar, N., & Setyawati, O. (2023). *Intelligent System* (B. Rianto (ed.)). Penerbit Nuta Media.
- Arifin, M., Asfani, K., & Handayani, A. N. (2018). Aplikasi Jaringan Saraf Tiruan Metode Perceptron Pada Pengenalan Pola Notasi. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 9(1), 77-86. <https://doi.org/10.24176/simet.v9i1.1737>
- Ayu, F. (2019). *Implementasi Jaringan Saraf Tiruan Untuk Menentukan Kelayakan Proposal Tugas Akhir*. 3(2), 44-53.
- Aziz, A. S., RF, B. R., & Kristianti, T. (2022). Model Neuron Mc Culloch-Pitts dalam Pengenalan Pola Logika Dasar. *Jurnal JEECOM*, 4(2), 51-56.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. In *Springer*.
- Intan, I., Rismayani, Ghani, S. A. D., Nurdin, & Koswara, A. T. . (2021). Analisis Performansi Prakiraan Cuaca Menggunakan Algoritma Machine Learning. *Jurnal Pekommas*, 6(2), 1-8. <https://doi.org/10.30818/jpkm.2021.2060221>
- Julpan, Nababan, E. B., & Zarlis, M. (2015). Analisis Fungsi Aktivasi Sigmoid Biner Dan Sigmoid Bipolar Dalam Algoritma Backpropagation Pada Prediksi Kemampuan Siswa. *Jurnal Teknovasi*, 02(1), 103-116.
- Maryana, S., Qur'ania, A., & Putra, A. P. (2018). Identifikasi Pengenalan Karakter Plat Nomor Kendaraan Menggunakan Jaringan Syaraf Tiruan Berbasis Citra Digital. *Komputasi: Jurnal Ilmiah Ilmu Komputer Dan Matematika*, 15(1), 111-117. <https://doi.org/10.33751/komputasi.v15i1.1266>
- Noprizal, & Candra, F. (2019). Aplikasi Pengenalan Plat Nomor Kendaraan Di Universitas Riau. *Jurnal Fasilkom*, 9(3), 47-52. <https://doi.org/10.37859/jf.v9i3.1670>
- Roza, Y., Fernando, Y., Saragih, R. E., Verdian, I., & Kunci, K. (2023). Perancangan Aplikasi Manajemen Proyek Pada PT . Sintech Berkah Abadi Berbasis Web. *J-INTECH (Journal of Information and Technology)*, 11(1), 167-176.
- Sudarsono, A. (2016). Jaringan Syaraf Tiruan Untuk Memprediksi Laju Pertumbuhan Penduduk Menggunakan Metode Backpropagation. *Jurnal Media Infotama*, 12(1), 61-69.
- Wahyono, T. (2018). *Fundamental Of Python For Machine Learning (Dasar-Dasar Pemrograman Python Untuk Machine Learning dan Kecerdasan Buatan)*. Penerbit Gava Media.
- Widodo, Y. B. (2019). Pengenalan Angka Tulisan Tangan Menggunakan Jaringan Syaraf Buatan. *Jurnal Teknologi Informatika Dan Komputer*, 5(1), 51-54. <https://doi.org/10.37012/jtik.v5i1.221>