1. **Praktikum**
2. Buka Google Colaboratory melalui [tautan ini](https://colab.research.google.com/drive/1mIk_MBA_LzC2TP9ziU7yeBS-4omokJ1s).
3. Tulis kode berikut ke dalam setiap *cell* pada *notebook* tersebut.
   1. Fungsi *Step* Biner

|  |
| --- |
| def binstep(x, th=0):      return 1 if x >= th else 0 |

* 1. Fungsi McCulloch-Pitts (MCP) Neuron

|  |
| --- |
| import numpy as np  def MCP(x, w, th):      y\_in = np.dot(x, w)      y\_out = binstep(y\_in, th)      return y\_out |

1. Fungsi Hitung Akurasi

|  |
| --- |
| def calc\_accuracy(a, b):      s = [1 if a[i] == b[i] else 0 for i in range(len(a))]      return sum(s) / len(a) |

* 1. Fungsi Logika AND

|  |
| --- |
| def AND(X):      w = 1, 1      th = 2      y = [MCP(i, w, th) for i in X]      return y |

* 1. Eksekusi Logika AND

|  |
| --- |
| data = (0, 0), (0, 1), (1, 0), (1, 1)  output = AND(data)  true = 0, 0, 0, 1  accuracy = calc\_accuracy(output, true)  print('Output:', output)  print('True:', true)  print('Accuracy:', accuracy) |

* 1. Fungsi Logika OR

Buat fungsi baru untuk logika OR berdasarkan fungsi logika AND dengan mengubah nilai bobot menjadi [2, 2].

|  |
| --- |
| def OR(X):  w = 2, 2  th = 2  y = [MCP(i, w, th) for i in X]  return y |

* 1. Eksekusi Logika OR

|  |
| --- |
| data = (0, 0), (0, 1), (1, 0), (1, 1)  output = OR(data)  true = 0, 1, 1, 1  accuracy = calc\_accuracy(output, true)  print('Output:', output)  print('True:', true)  print('Accuracy:', accuracy) |

* 1. Logika AND NOT

Buat fungsi baru untuk logika OR berdasarkan fungsi logika AND dengan mengubah nilai bobot menjadi [2, -1].

|  |
| --- |
| def ANDNOT(X):  w = 2, -1  th = 2  y = [MCP(i, w, th) for i in X]  return y |

* 1. Eksekusi Logika AND NOT

|  |
| --- |
| data = (0, 0), (0, 1), (1, 0), (1, 1)  output = ANDNOT(data)  true = 0, 0, 1, 0  accuracy = calc\_accuracy(output, true)  print('Output:', output)  print('True:', true)  print('Accuracy:', accuracy) |

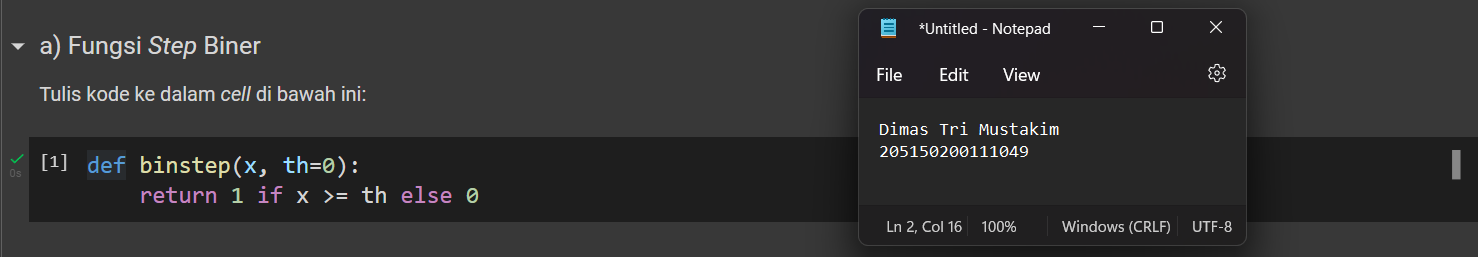
* 1. Logika XOR

|  |
| --- |
| def XOR(X):      X\_flip = [(i[1], i[0]) for i in X]      y1 = ANDNOT(X)      y2 = ANDNOT(X\_flip)      y = zip(y1, y2)      z = OR(y)      return z |

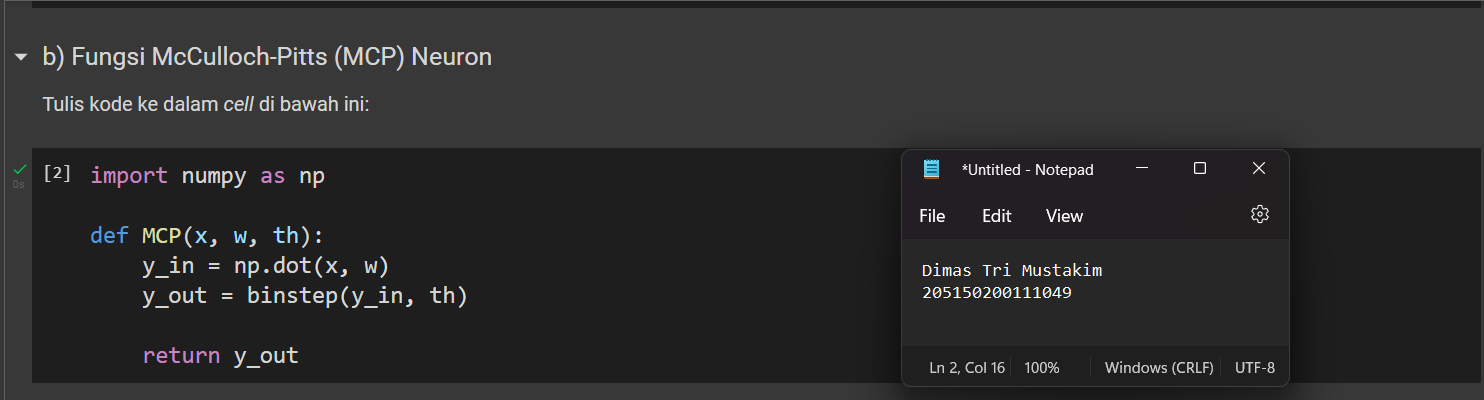
* 1. Eksekusi Logika XOR

|  |
| --- |
| data = (0, 0), (0, 1), (1, 0), (1, 1)  output = XOR(data)  true = 0, 1, 1, 0  accuracy = calc\_accuracy(output, true)  print('Output:', output)  print('True:', true)  print('Accuracy:', accuracy) |

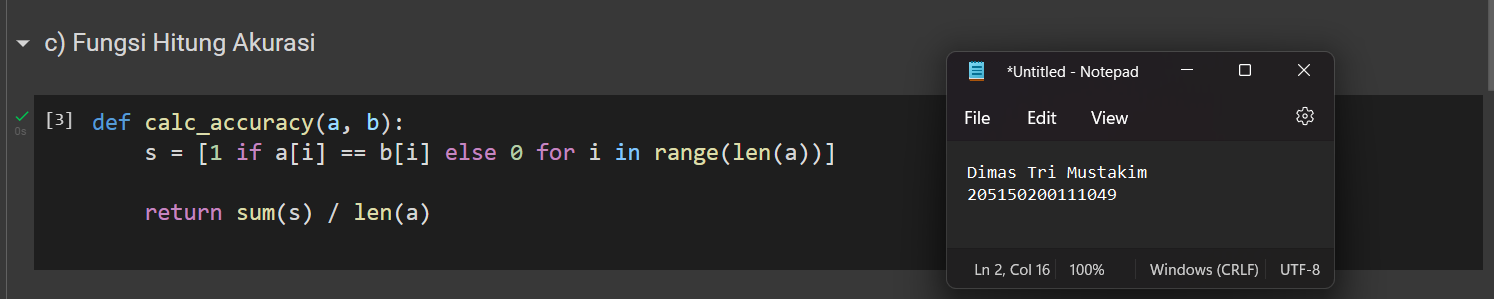
1. **Screenshot**
2. Fungsi *Step* Biner



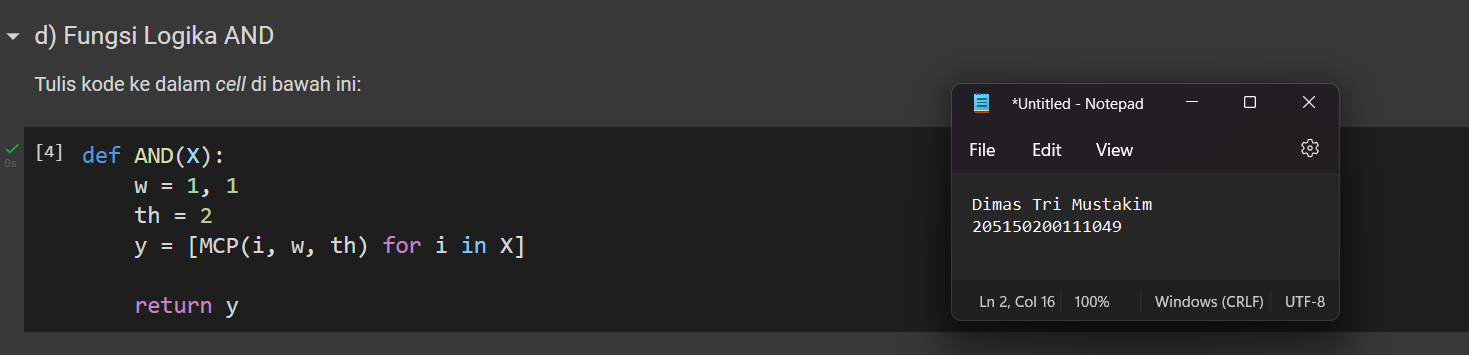
1. Fungsi McCulloch-Pitts (MCP) Neuron



1. Fungsi Hitung Akurasi



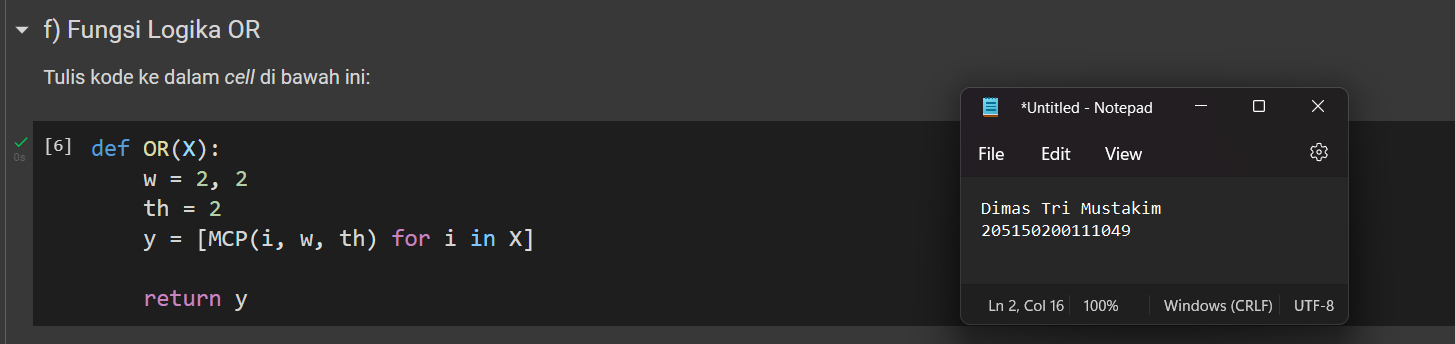
1. Fungsi Logika AND



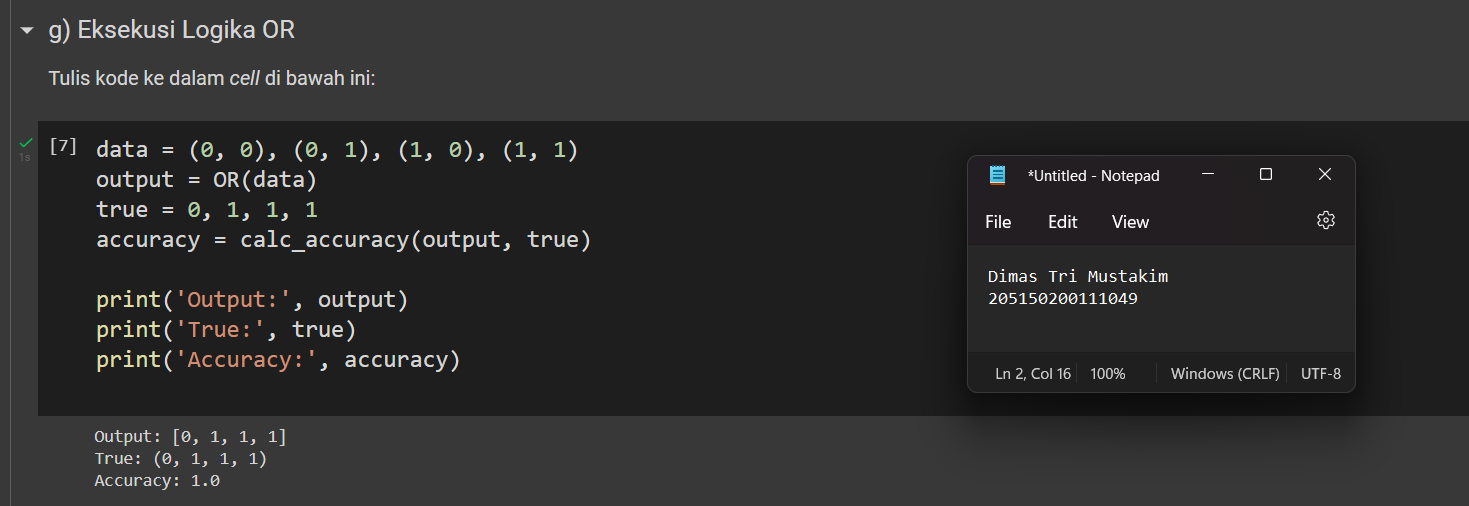
1. Eksekusi Logika AND



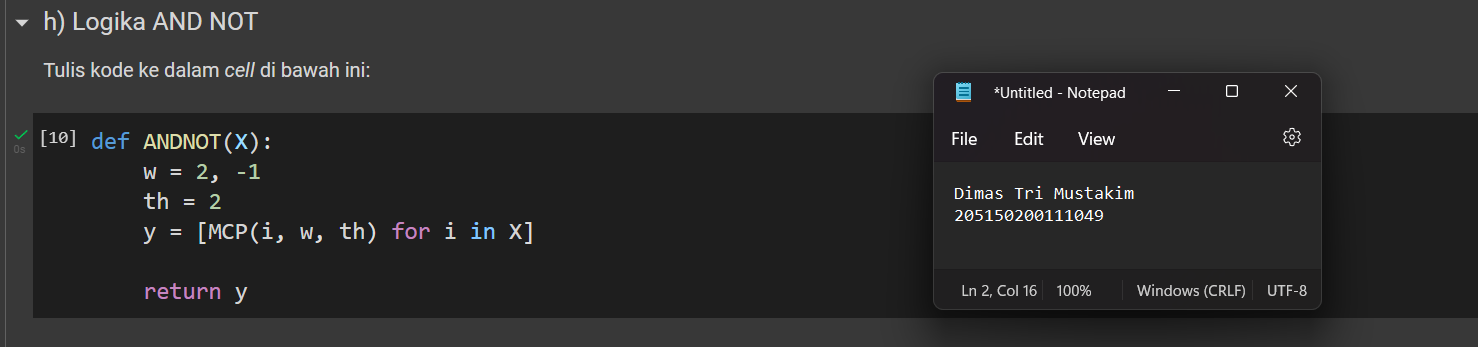
1. Fungsi Logika OR



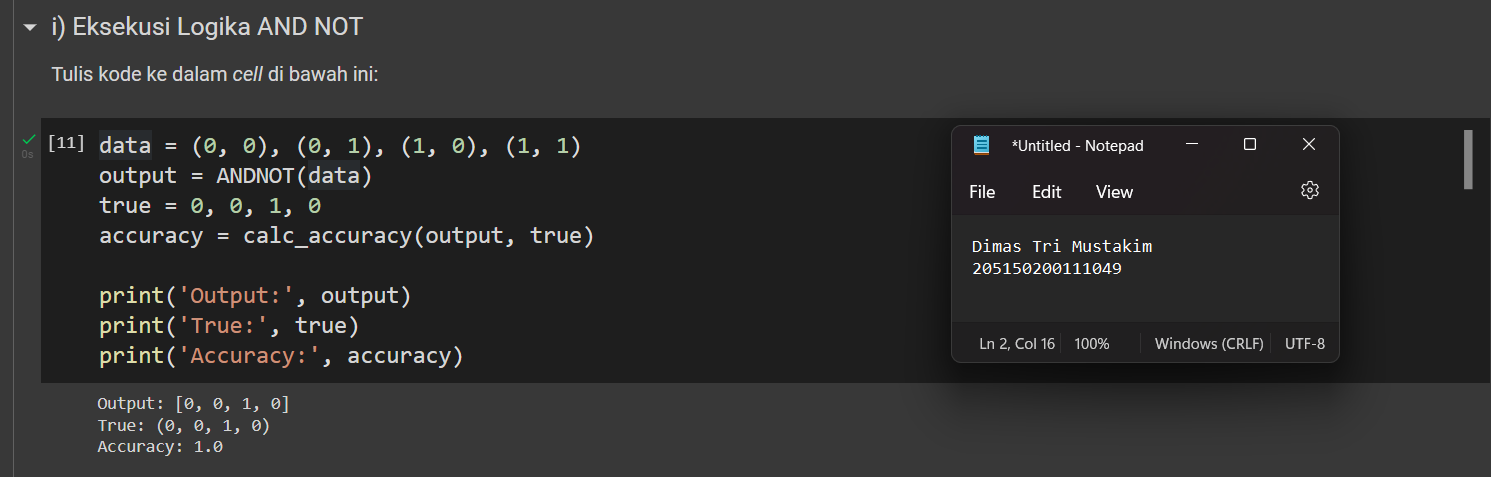
1. Eksekusi Logika OR



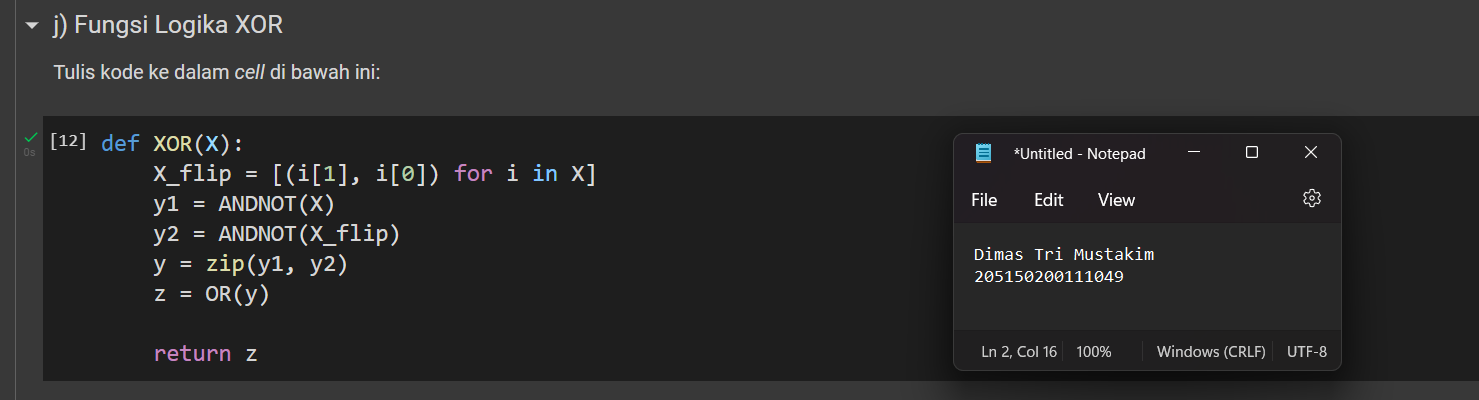
1. Logika AND NOT



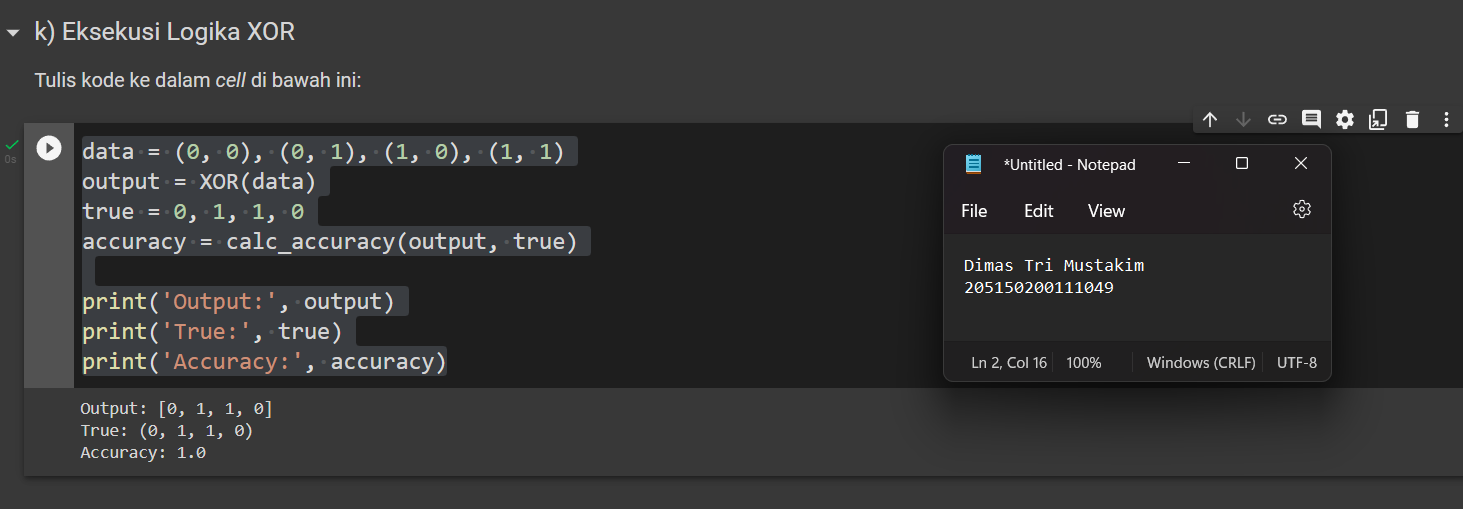
1. Eksekusi Logika AND NOT



1. Logika XOR



1. Eksekusi Logika XOR



1. **Analisis**
2. Jalankan semua *cell* (Ctrl+F9). Amati *output* dan akurasi yang dihasilkan.

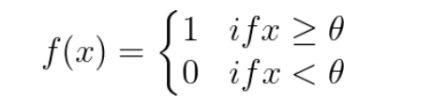
**Jawab:**

Ketika semua *cell* dijalankan, tidak ada yang *error* yang terjadi. Untuk *cell* yang mengeksekusi fungsi logika diatasnya, akan keluar *output* bahwa dengan nilai yang dihasilkan oleh fungsi dan nilai yang di ekspektasikan adalah sama, dan nilai akurasi bernilai 1.0 (sempurnya).

1. Pada kode a, apa maksud dari *statement* return 1 if x >= th else 0?

**Jawab:**

Merupakan kode yang berperan sebagai fungsi aktivasi yaitu fungsi step biner. *Statement* tersebut akan mengembalikan nilai 1 jika nilai x (input) bernilai lebih dari sama dengan *threshold*, dan jika tidak maka akan mengembalikan nilai 0. Fungsi yang digunakan yaitu step biner dengan notasi matematika seperti berikut.



1. Pada kode b:
2. Apakah perbedaan antara variabel y\_in dan y\_out?

**Jawab:**

Variabel y\_in merupakan hasil dari dot product dari input x dan *weight*, sedangkan y\_out merupkan hasil y\_in yang telah melewati fungsi aktivasi yaitu fungsi step biner.

1. Apakah yang dilakukan oleh fungsi np.dot()?

**Jawab:**

Fungsi tersebut akan menghasilkan hasil perkalian (*dot product*) dari *array* yang masuk. Fungsi tersebut digunakan untuk perkalian skalar antara *array input* dan *array* *weight* (bobot).

1. Pada kode c, apakah kegunaan dari variabel-variabel w dan th?

**Jawab:**

Variable w merupakan *weight* (bobot) dan th merupakan threshold. Variable w digunakan dalam perhitungan weighted sum, dimana w dikalikan dengan input x. Weight sendiri merepresentasikan kekuatan koneksi antar unit. Variable th merupakan *threshold*, yang akan digunakan di dalam fungsi aktivasi. Fungsi aktivasi yang digunakan adalah fungsi step biner.

1. Pada kode j, apakah tujuan dari variabel X\_flip?

**Jawab:**

Variabel x\_flip akan terisi nilai variabel X (input) yang telah dibalik posisi masing-masing pasang isinya.

1. Pada hal apa saja terdapat kesamaan antara neuron biologis dan neuron tiruan?

**Jawab:**

Kesamaannya yaitu bahwa neuron di keduanya merupakan unit paling sederhana di dalam sistem jaringan saraf. Keduanya juga merupakan unit yang digunakan sebagai pemrosesan informasi, dan dapat meneruskan informasi ke neuron lain.

1. Jelaskan kelebihan dan kekurangan yang dimiliki McCulloch-Pitts neuron.

**Jawab:**

Kelebihan dari McCulloch-Pitts neuron adalah bahwa modelnya yang sederhana, sehingga kebutuhan komputasinya rendah. Sedangkan kekurangannya yaitu seperti:

* McCulloch-Pitts neuron hanya dapat memproses input dan output dalam bentuk Boolean. Hal tersebut dikarenakan model ini hanya menggunakan fungsi aktivasi step biner yang hanya menghasilkan nilai 0 atau 1, dan fungsi pembobotannya juga hanya mendukung input bertipe Boolean.
* Nilai *threshold* yang ditentukan dan di-*hard code* sehingga harus mengetahui *threshold* yang dibutuhkan terlebih dahulu.
* Tidak bisa menyelesaikan permasalahan yang tidak bisa dipisahkan secara linier seperti fungsi XOR.

1. Mengapa logika XOR lebih rumit sehingga membutuhkan kombinasi beberapa logika yang lain?

**Jawab:**

Karena logika XOR merupakan permasalahan yang tidak bisa dipisahkan secara linier, dan model McCulloch-Pitts hanya bisa menyelesaikan permasalahan yang hanya bisa dipisahkan secara linier.

1. **Kesimpulan**

Jaringan Saraf Tiruan merupakan sistem pemrosesan informasi yang menggunakan jaringan saraf biologis sebagai model. Jaringan saraf tiruan dikembangkan menggunakan model matematis dengan komponen yaitu:

* Neuron, sebagai pemroses informasi.
* Penghubung, yang menghubungkan output satu neuron ke input neuron yang lain.
* Weight atau bobot, yang mempengaruhi seberapa kuatnya informasi diteruskan.
* Fungsi aktivasi, yang memproses sinyal input menjadi sinyal output.

Pada jaringan saraf tiruan terdapat beberapa variabel dan fungsi yang diperlukan, yaitu variable x, w, y, dan fungsi aktivasi. Variabel x melambangkan nilai input. Nilai tersebut akan masuk ke dalam neuron input yang berada pada layer pertama yang juga disebut *input layer*. Jumlah neuron pada layer tersebut disesuaikan dengan panjang vektor pada data input yang digunakan. Variabel w merupakan weight (bobot) yang merepresentasikan kekuatan koneksi antar unit. Variabel y merupakan *output* dari jaringan saraf tiruan. Dan fungsi aktivasi merupakan fungsi yang melakukan perhitungan terhadap sinyal input yang masuk ke dalam neuron dan menghasilkan aktivasi dari neuron tersebut.

Contoh jaringan saraf tiruan yang cukup sederhana adalah McCulloch-Pitts Neuron, yang diperkirakan merupakan jaringan saraf tiruan yang pertama kali dibuat. Model ini dirancang pada tahun 1943 oleh McCulloch dan Pitts. Jaringan saraf tiruan ini menggunakan aktivasi *step biner* dan bobot serta *threshold*-nya harus ditentukan secara manual. Contoh penggunaan model ini adalah pada kasus klasifikasi biner.