1. **Praktikum**
2. Buka Google Colaboratory melalui [tautan ini](https://colab.research.google.com/drive/1Lb3_broszRP4vsWMJbKxFZHdEvmAWh9m?usp=sharing).
3. Tulis kode berikut ke dalam setiap *cell* pada *notebook* tersebut.
   1. Fungsi Step Bipolar

|  |
| --- |
| def bipstep(y, th=0):  return 1 if y >= th else -1 |

* 1. Fungsi Training Adaline

|  |
| --- |
| import sys  def adaline\_fit(x, t, alpha=.1, max\_err=.1, max\_epoch=-1, verbose=False, draw=False):  w = np.random.uniform(0, 1, len(x[0]) + 1)  b = np.ones((len(x), 1))  x = np.hstack((b, x))  stop = False  epoch = 0  while not stop and (max\_epoch == -1 or epoch < max\_epoch):  epoch += 1  max\_ch = -sys.maxsize  if verbose:  print('\nEpoch', epoch)  for r, row in enumerate(x):  y = np.dot(row, w)  for i in range(len(row)):  w\_new = w[i] + alpha \* (t[r] - y) \* row[i]  max\_ch = max(abs(w[i] - w\_new), max\_ch)  w[i] = w\_new  if verbose:  print('Bobot:', w)  if draw:  plot(line(w), x, t)  stop = max\_ch < max\_err  return w |

1. Fungsi Testing Adaline

|  |
| --- |
| def adaline\_predict(X, w):  Y = []  for x in X:  y\_in = w[0] + np.dot(x, w[1:])  y = bipstep(y\_in)  Y.append(y)  return Y |

* 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. Logika AND

|  |
| --- |
| train = (1, 1), (1, -1), (-1, 1), (-1, -1)  target = 1, -1, -1, -1  w, epoch = adaline\_fit(train, target, verbose=True, draw=True)  output = adaline\_predict(train, w)  accuracy = calc\_accuracy(output, target)  print('Output:', output)  print('Epoch:', epoch)  print('Target:', target)  print('Accuracy:', accuracy) |

* 1. Logika OR

|  |
| --- |
| train = (1, 1), (1, -1), (-1, 1), (-1, -1)  target = 1, 1, 1, -1  w, epoch = adaline\_fit(train, target, verbose=True, draw=True)  output = adaline\_predict(train, w)  accuracy = calc\_accuracy(output, target)  print('Output:', output)  print('Epoch:', epoch)  print('Target:', target)  print('Accuracy:', accuracy) |

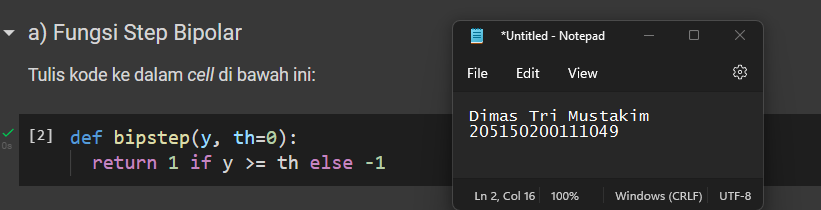
* 1. Logika AND NOT

|  |
| --- |
| train = (1, 1), (1, -1), (-1, 1), (-1, -1)  target = -1, 1, -1, -1  w, epoch = adaline\_fit(train, target, verbose=True, draw=True)  output = adaline\_predict(train, w)  accuracy = calc\_accuracy(output, target)  print('Output:', output)  print('Epoch:', epoch)  print('Target:', target)  print('Accuracy:', accuracy) |

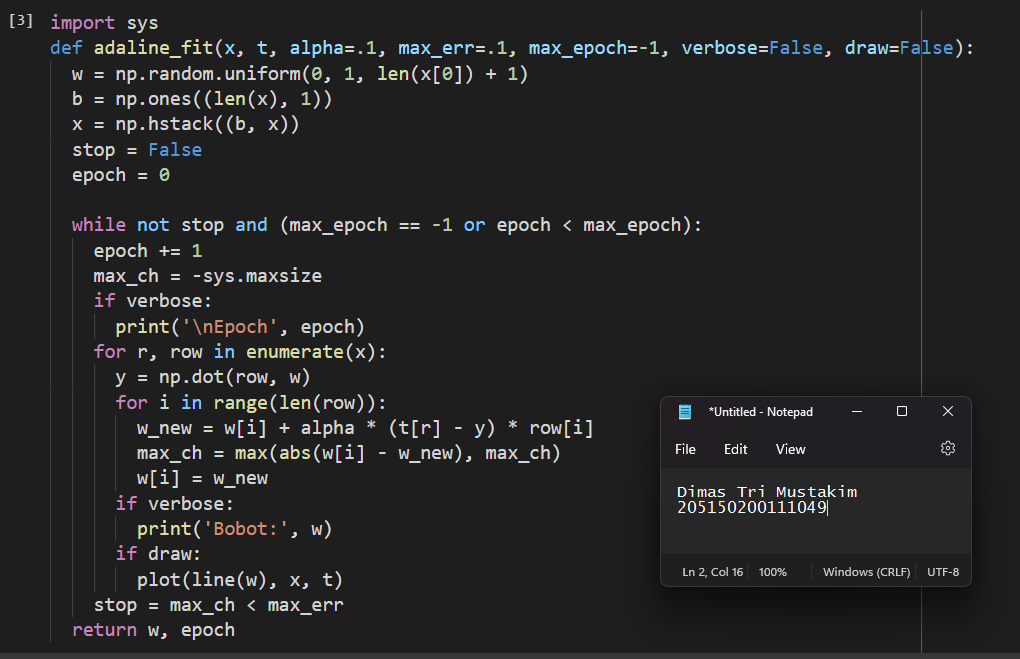
* 1. Logika XOR

|  |
| --- |
| train = (1, 1), (1, -1), (-1, 1), (-1, -1)  target = -1, 1, 1, -1  w, epoch = adaline\_fit(train, target, verbose=True, draw=False)  output = adaline\_predict(train, w)  accuracy = calc\_accuracy(output, target)  print('Output:', output)  print('Epoch:', epoch)  print('Target:', target)  print('Accuracy:', accuracy) |

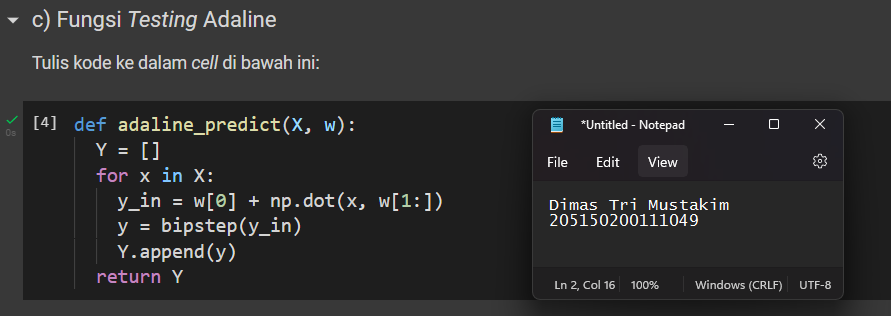
1. **Screenshot**
2. Fungsi Step Bipolar



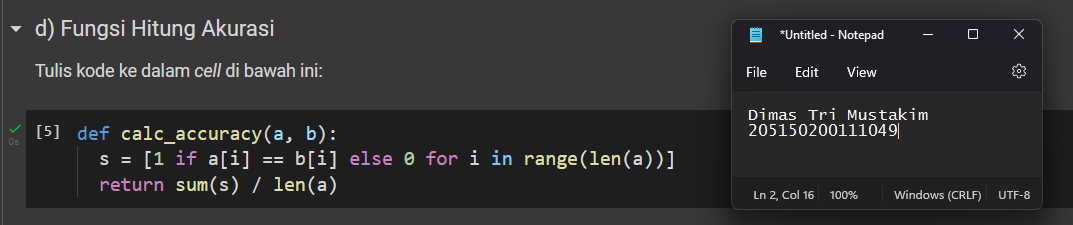
1. Fungsi Training Adaline



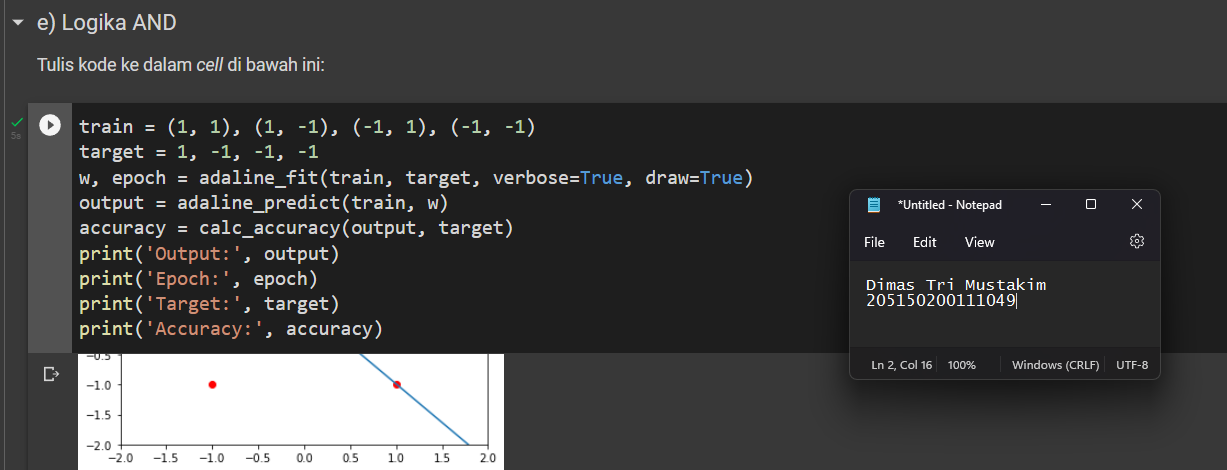
1. Fungsi Testing Adaline

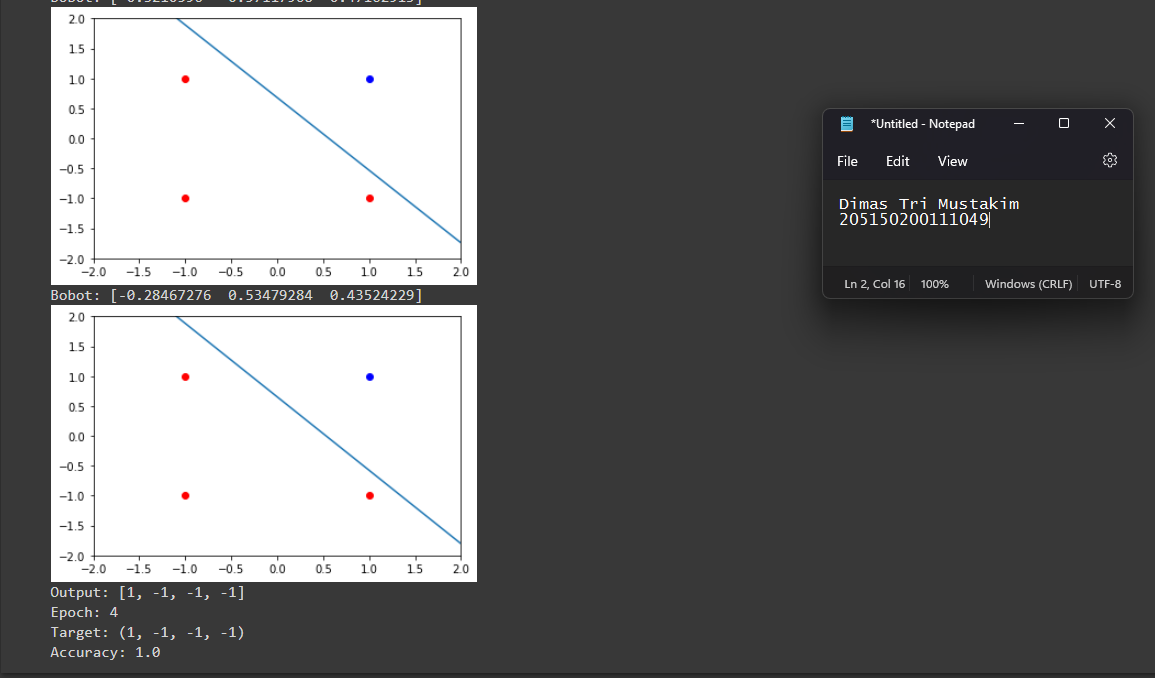


1. Fungsi Hitung Akurasi

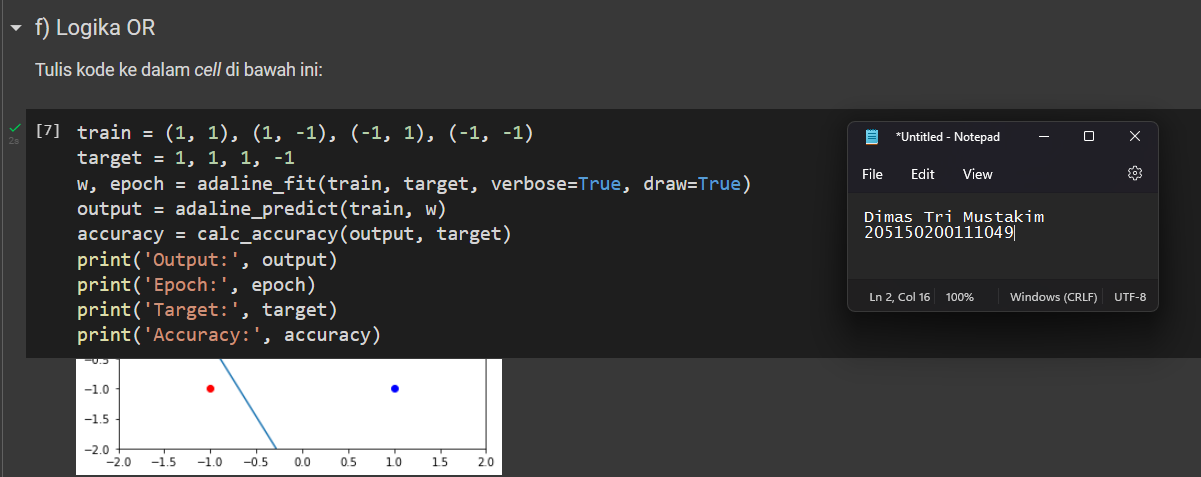


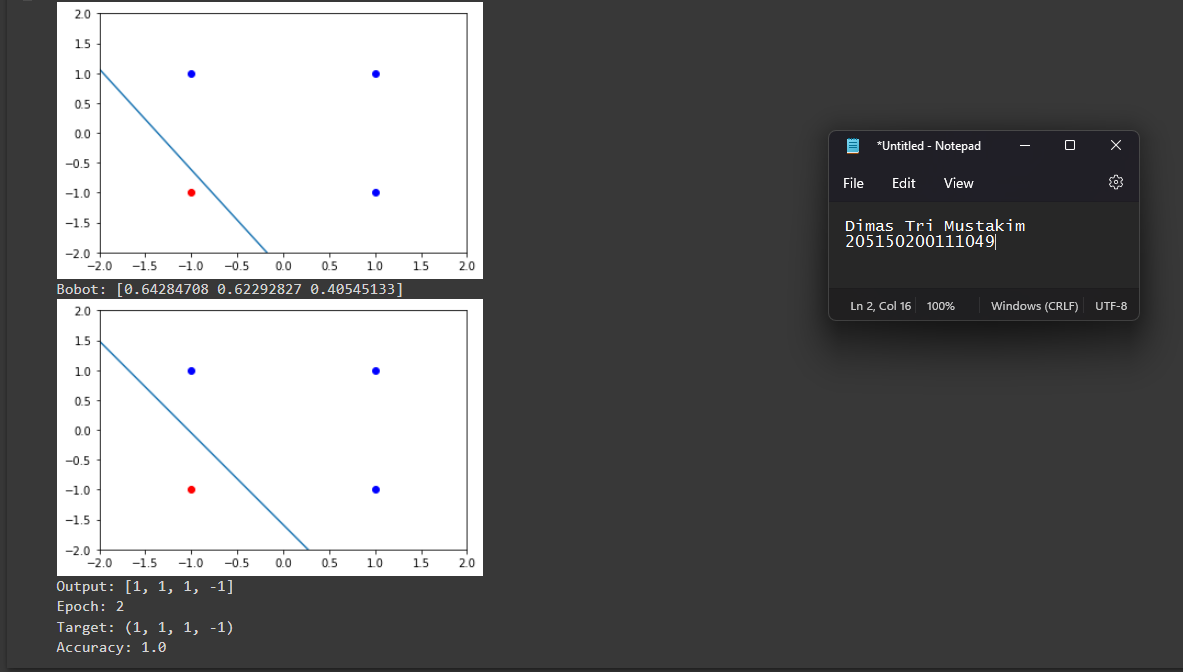
1. Logika AND



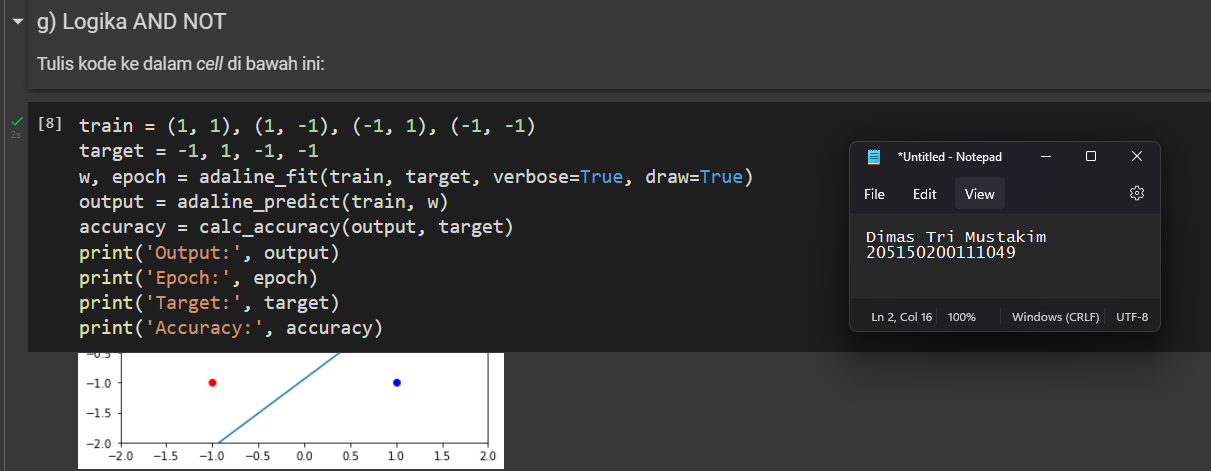


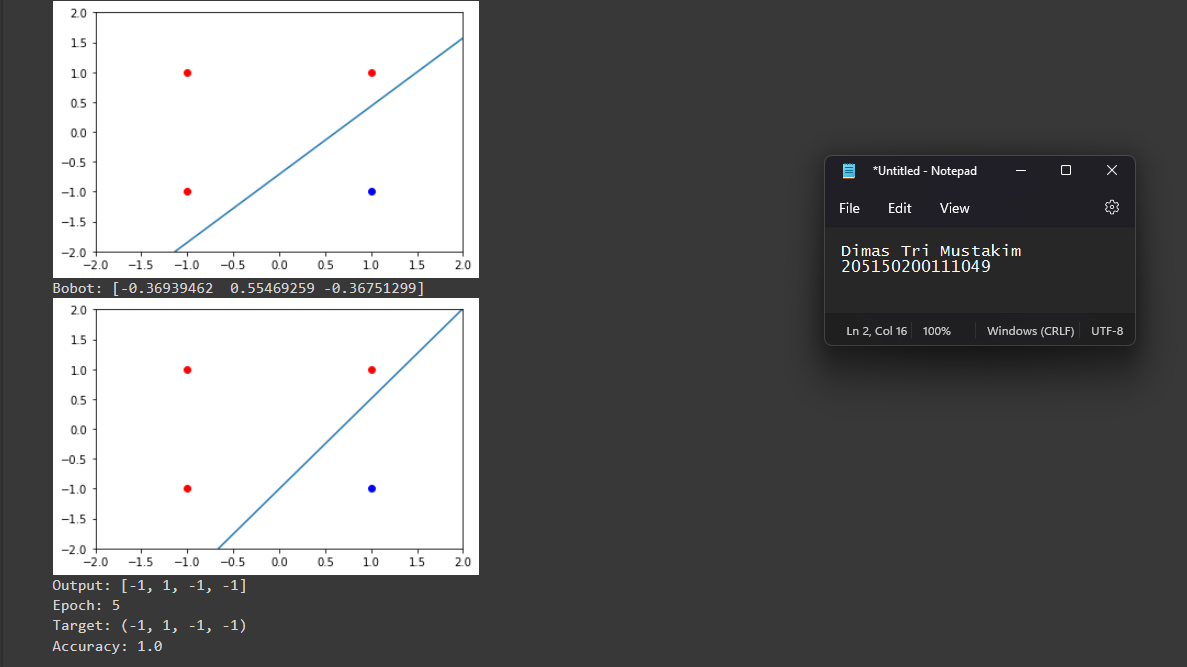
1. Logika OR



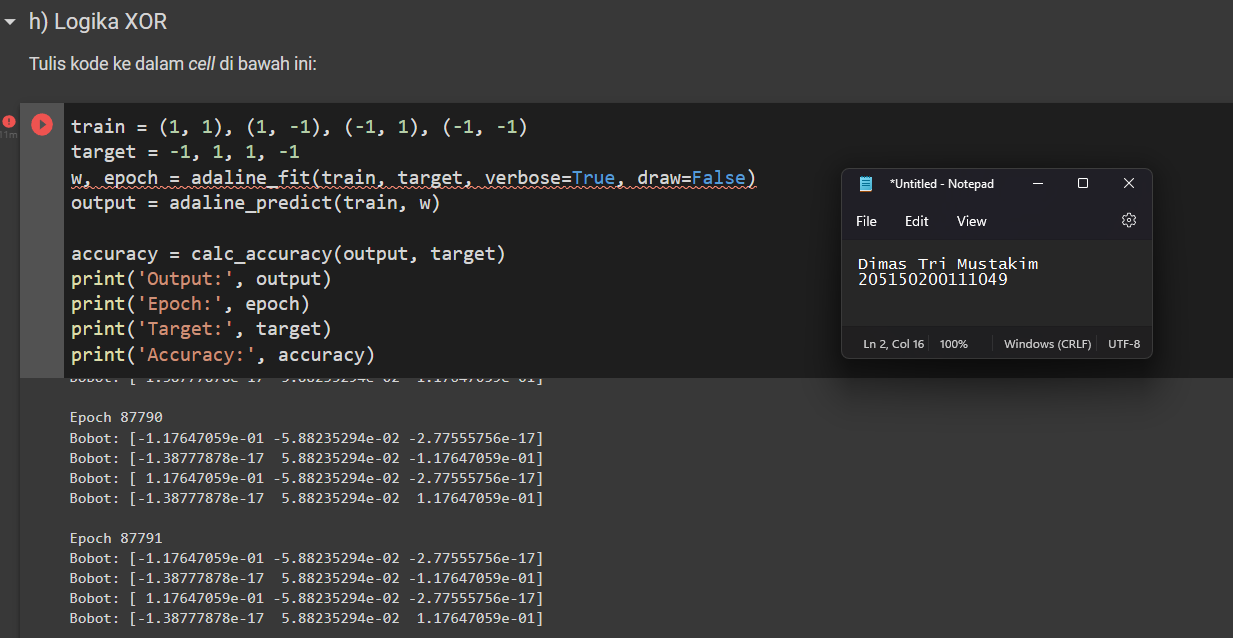


1. Logika AND NOT





1. Logika XOR



1. **Analisis**
2. Jelaskan tujuan dari parameter-parameter alpha, max\_err, dan max\_epoch yang ada pada fungsi adaline\_fit.

**Jawab:**

Parameter alpha berfungsi untuk mendeklarasikan seberapa besar bobot akan diperbarui atau decision boundary akan digeser pada setiap training. Parameter max\_err berfungsi sebagai penentu kapan proses training akan dihentikan. Jika perubahan bobot tertinggi pada epoch lebih kecil dari max\_err, maka proses pelatihan akan berhenti. Sedangkan parameter max\_epoch merupakan parameter yang menentukan batas jumlah epoch yang bisa dilakukan pada proses pelatihan.

1. Pada fungsi adaline\_fit, apakah yang akan dilakukan oleh fungsi tersebut jika parameter max\_epoch diberi nilai -1?

**Jawab:**

Jika max\_epoch diberi nilai -1, berarti proses training tidak dibatasi untuk berapa kali epoch. Dapat dilihat di dalam kode program, terdapat while loop dengan kondisi yang menggunakan nilai max\_epoch. Jika max\_epochnya adalah -1 maka akan akan seperti tidak diatur, dan jika max\_epochnya diatur, maka proses training akan tetap berjalan ketika jumlah epoch lebih kecil dari max\_epoch.

1. Amati jumlah epoch saat melakukan proses pelatihan menggunakan data logika AND, OR, dan AND NOT. Mengapa jumlah epoch pada ketiga proses pelatihan tersebut tidak sama?

**Jawab:**

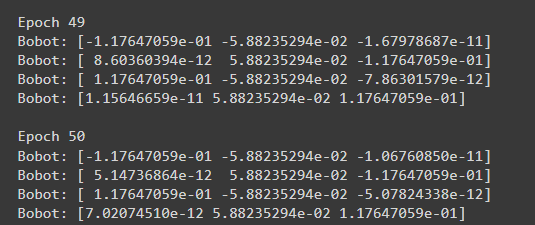
Jumlah epoch pada proses training untuk ketiga permasalahan tersebut tidak sama karena permasalahan yang ingin diklasifikasikan tidak sama dan nilai bobot awal yang digunakan oleh Adaline adalah random, sehingga membutuhkan epoch pelatihan yang berbeda.

1. Apakah yang terjadi saat melakukan proses pelatihan untuk data logika XOR? Mengapa bisa terjadi demikian?

**Jawab:**

Hal yang terjadi adalah yaitu proses pelatihan untuk data logika XOR tidak pernah berhenti meskipun telah mencapai epoch yang sangat besar. Hal tersebut karena Adaline juga merupakan yang hanya bisa mengklasifikasikan permasalahan yang *linearly separable* dan permasalahan XOR merupakan permasalahan yang *non linearly separable*.

Proses training yang tidak pernah selesai terjadi karena kondisi berhenti dari algoritma Adaline di program tersebut yaitu max\_err dan max\_epoch. Max\_epoch tidak dibatasi dan Max\_err tidak tercapai karena bobot selalu berubah setiap kali pelatihan dari minus ke positif dan sebaliknya seperti digambar dibawah yang menyebabkan nilai perubahan bobot cukup besar sehingga tidak pernah lebih kecil nilai max\_err.



1. **Kesimpulan**

Adaline merupakan salah satu jenis algoritma jaringan saraf tiruan *single layer* yang terdiri dari bebearpa input neuron dan satu output neuron (pengklasifikasi biner). Adaline memiliki bias yang memiliki nilai aktivasi konstan sebesar 1. Adaline memiliki algoritma pelatihannya sendiri untuk mengubah nilai-nilai bobot.

Fungsi dari threshold pada pelatihan Adaline adalah untuk menentukan waktu berhenti dari proses training. Jika perubahan bobot terbesar dalam satu epoch lebih kecil dari threshold, maka proses training akan dehentikan. Sedangkan learning rate berfungsi untuk mengontrol seberapa banyak bobot dari jaringan disesuaikan untuk setiap proses training. Pada data yang kompleks, learning rate sangat penting agar pergeseran decision boundary tidak terlalu besar yang menyebabkan posisi decision boundary tersebut tidak baik.

Adaline tidak bisa menyelesaikan permasalahan logika XOR karena merupakan permasalahan yang termasuk not linearly separable. Adaline hanya bisa memisahkan data dengan satu buah garis linear, yang membuatnya tidak bisa melakukan klasifikasi pada permasalahan XOR. Dapat diamati juga pada saat proses training, terjadi kejadian bahwa proses training tersebut tidak akan selesai karena kondisi berhenti yaitu nilai perubahan bobot terbesar lebih kecil dari threshold tidak pernah terjadi.