1. **Praktikum**
2. Buka Google Colaboratory melalui [tautan ini](https://colab.research.google.com/drive/1CKVEdyX-oIoxec_rRZnNIhTweZKsYif4).
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* Hebb

|  |
| --- |
| def hebb\_fit(train, target, verbose=False, draw=False, draw\_padding=1):  w = np.zeros(len(train[0]) + 1)  bias = np.ones((len(train), 1))  train = np.hstack((bias, train))    for r, row in enumerate(train):  w = [w[i] + row[i] \* target[r] for i in range(len(row))]  if verbose:  print('Bobot:', w)  if draw:  plot(line(w, 0), train, target, draw\_padding)  return w |

1. Fungsi *Testing* Hebb

|  |
| --- |
| def hebb\_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

|  |
| --- |
| from sklearn.metrics import accuracy\_score  train = (1, 1), (1, -1), (-1, 1), (-1, -1)  target = 1, -1, -1, -1  model = hebb\_fit(train, target, verbose=False, draw=False)  output = hebb\_predict(train, model)  accuracy = accuracy\_score(output, target)  print('Output:', output)  print('Target:', target)  print('Accuracy:', accuracy) |

* 1. Logika OR

|  |
| --- |
| train = (1, 1), (1, -1), (-1, 1), (-1, -1)  target = 1, 1, 1, -1  model = hebb\_fit(train, target, verbose=True, draw=True)  output = hebb\_predict(train, model)  accuracy = accuracy\_score(output, target)  print('Output:', output)  print('Target:', target)  print('Accuracy:', accuracy) |

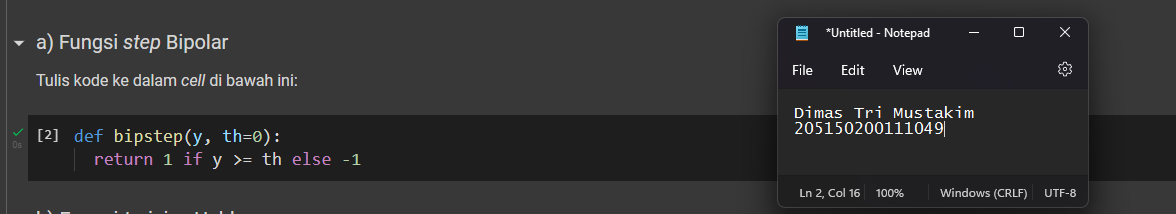
* 1. Logika AND NOT

|  |
| --- |
| train = (1, 1), (1, -1), (-1, 1), (-1, -1)  target = -1, 1, -1, -1  model = hebb\_fit(train, target, verbose=True, draw=True)  output = hebb\_predict(train, model)  accuracy = accuracy\_score(output, target)  print('Output:', output)  print('Target:', target)  print('Accuracy:', accuracy) |

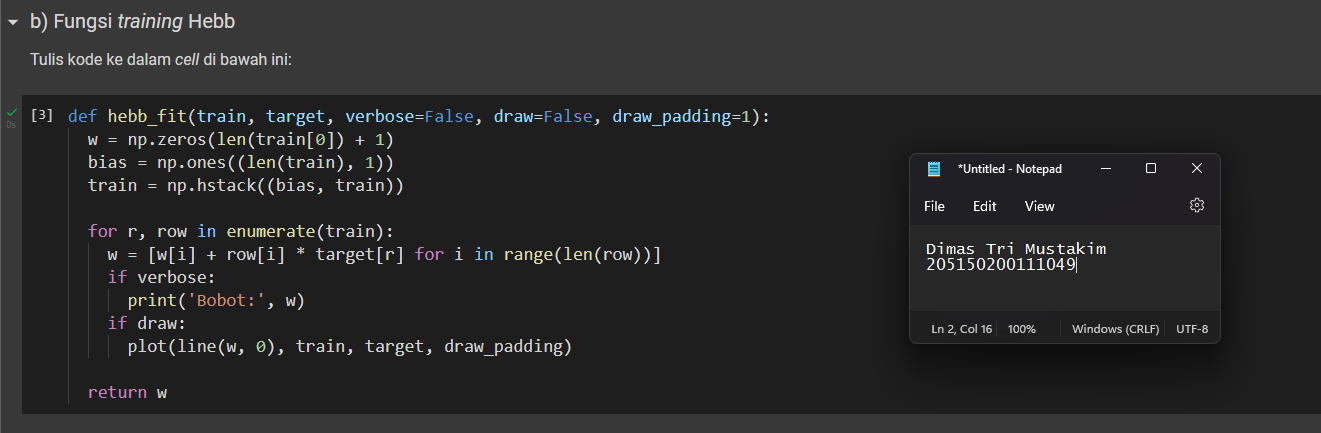
* 1. Logika XOR

|  |
| --- |
| train = (1, 1), (1, -1), (-1, 1), (-1, -1)  target = -1, 1, 1, -1  model = hebb\_fit(train, target, verbose=True, draw=True)  output = hebb\_predict(train, model)  accuracy = accuracy\_score(output, target)  print('Output:', output)  print('Target:', target)  print('Accuracy:', accuracy) |

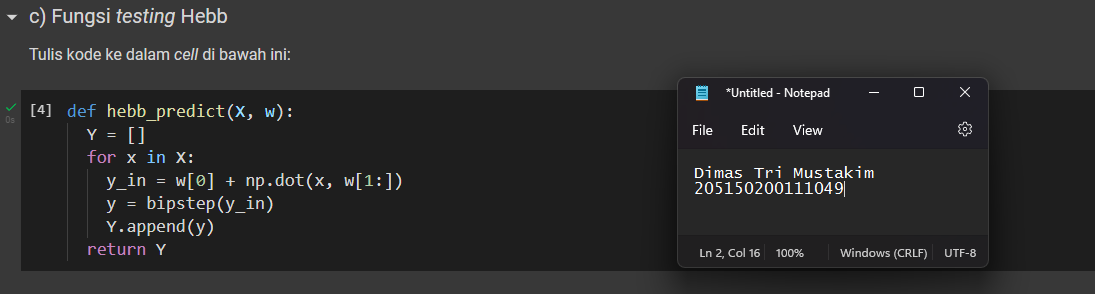
1. **Screenshot**
2. Fungsi *Step* Bipolar



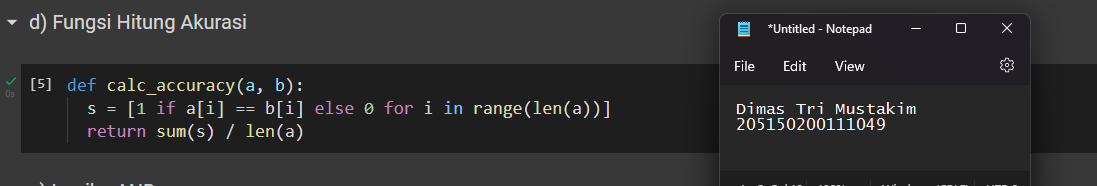
1. Fungsi *training* Hebb



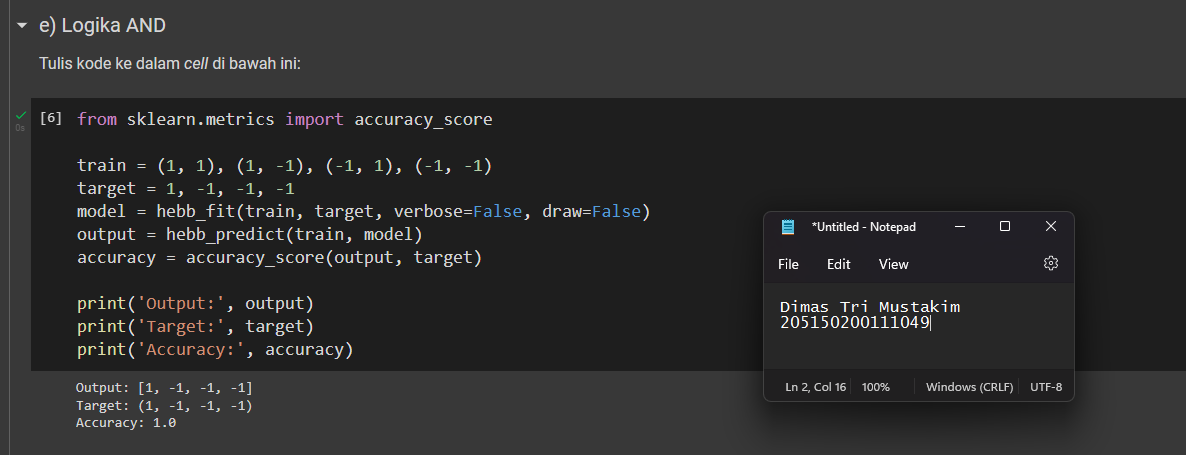
1. Fungsi Testing Hebb



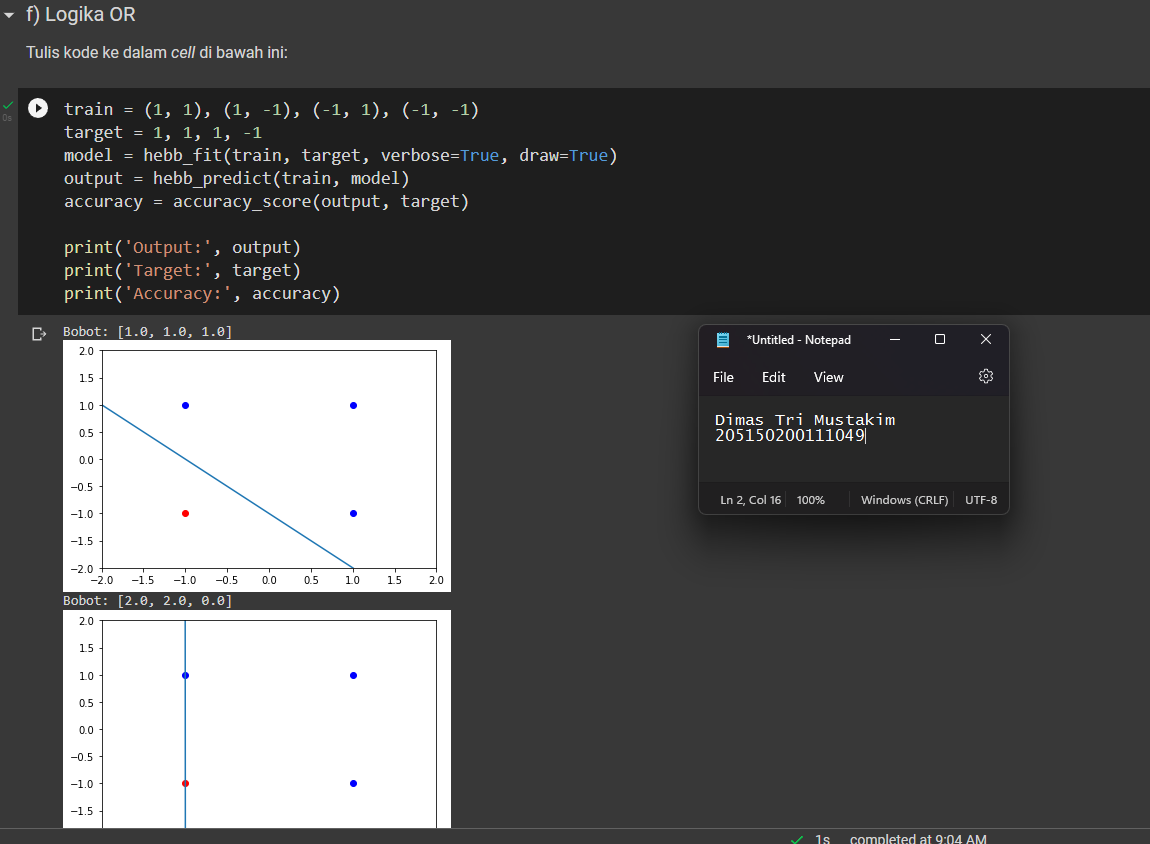
1. Fungsi Hitung Akurasi

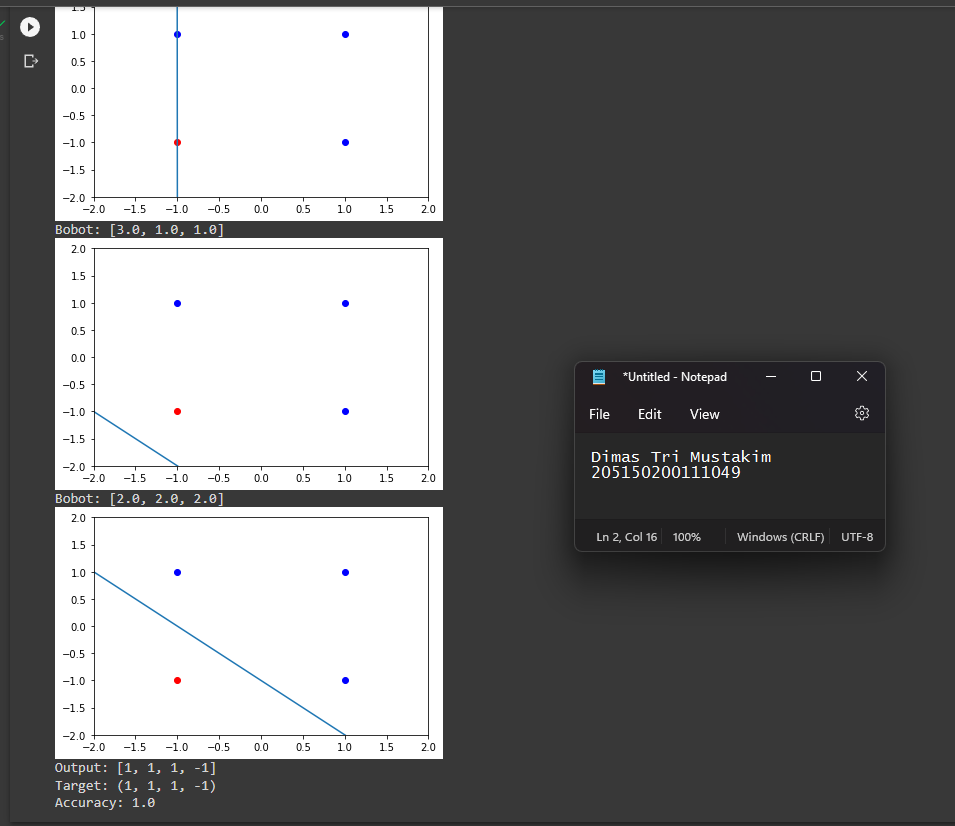


1. Logika AND

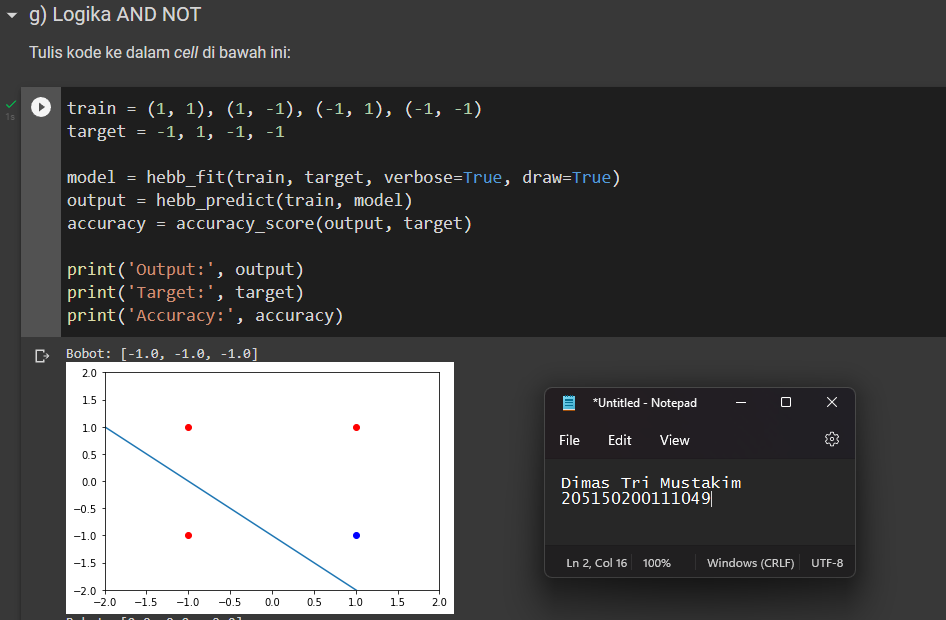


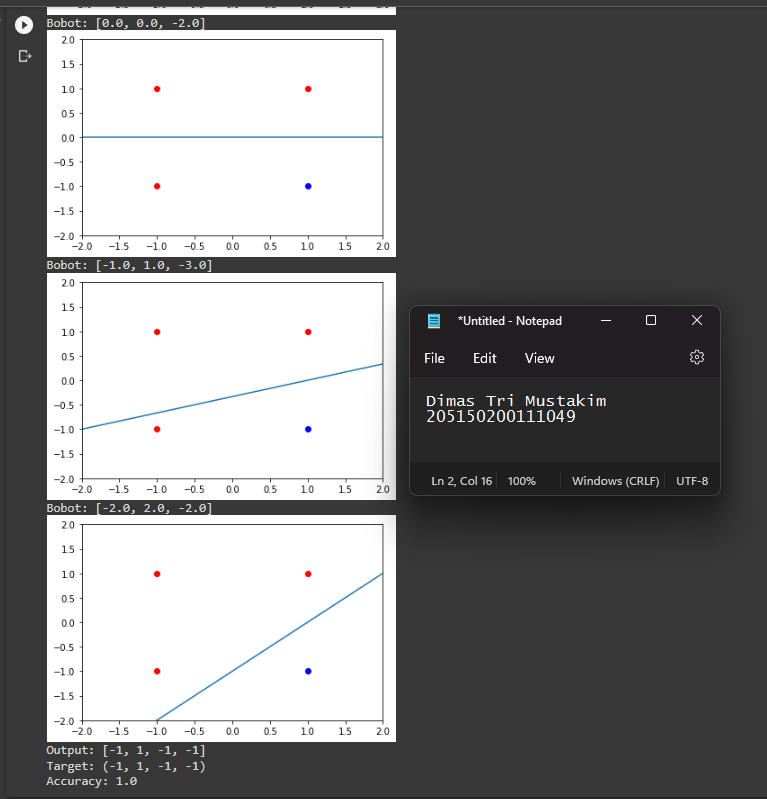
1. Logika OR





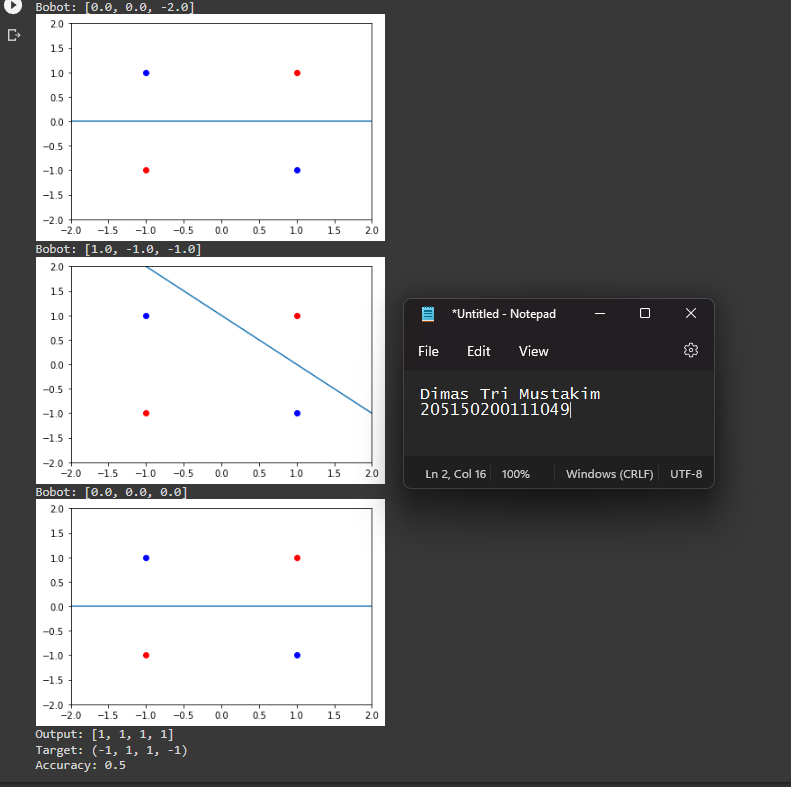
1. Logika AND NOT





1. Logika XOR





1. **Analisis**
2. Pada klasifikasi menggunakan logika XOR, mengapa akurasi yang didapatkan tidak mencapai 1 (100%)?.

**Jawab:**

Karena logika XOR merupakan permasalahan yang termasuk yang *not linearly-separable.* Dapat dilihat dari hasil plotingan, bahwa hebb net berusaha untuk menemukan garis yang dapat memisahkan klasifikasi XOR menggunakan garis linear, tetapi tidak akan bisa karena termasuk non-linear separable.

1. Lakukan proses training dan testing menggunakan data berikut.

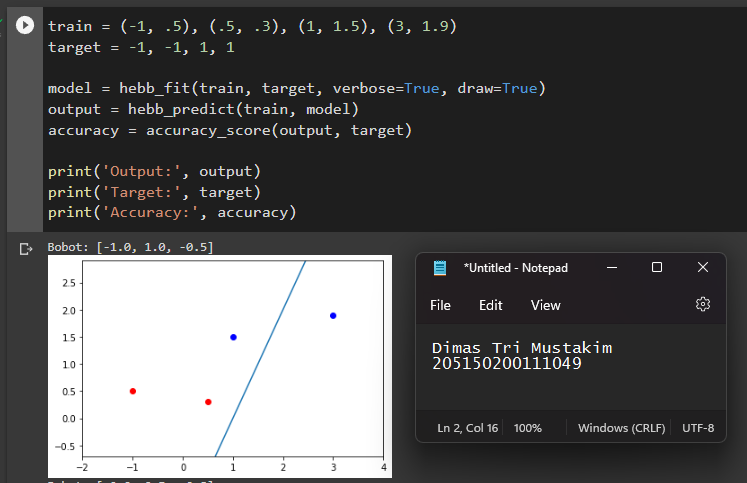
Training: (-1, .5), (.5, .3), (1, 1.5), (3, 1.9)

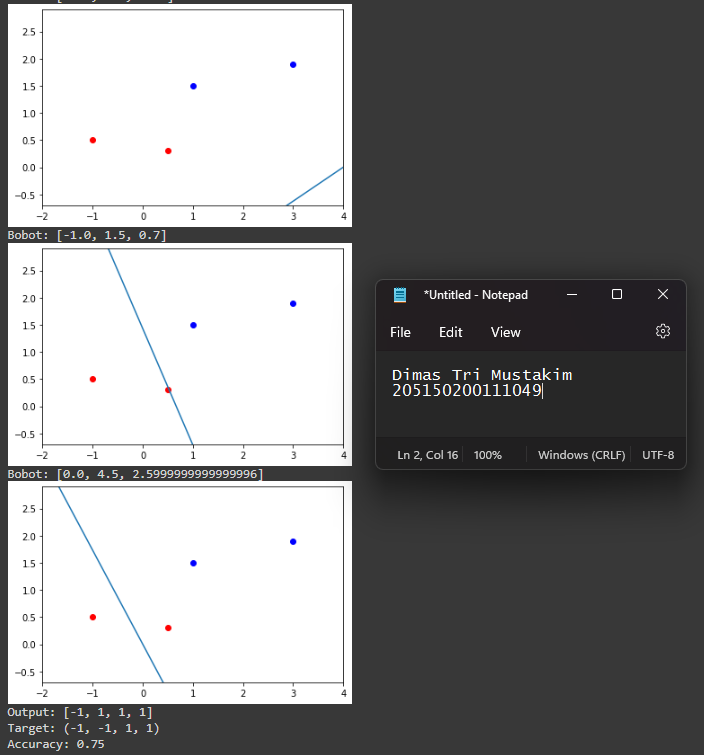
Target: (-1, -1, 1, 1)

Berapa akurasi yang didapatkan? Mengapa tidak dapat mencapai akurasi 1 (100%)?

**Jawab:**

Screenshot:





Merupakan Dari hasil percobaan tersebut, didapatkan bahwa hebbnet mendapatkan akurasi sebesar 75%. Hal tersebut karena bentuk input dari data yang ingin diklasifikasikan tidak berbentuk bipolar. Hebb net paling bagus digunakan untuk data input dan output bipolar.

1. **Kesimpulan**

Jaringan Hebb merupakan salah satu ANN yang menggunakan Hebb rule sebagai algoritma pelatihannya. Algoritma tersebut hanya akan menggunakan setiap data latih sebanyak satu kali dalam proses pelatihan (1 epoch). Jika dibandingkan dengan Mcculloch-pitts, salah satu kelebihan dari Hebb Net adalah adanya proses *training*, berbeda dengan Mcculloch-pitts yang membutuhkan kita untuk memasukkan bobot sendiri. Perbedaan lain adalah tipe data yang baik digunakan, hebb net paling bagus menggunakan data input dan output bipolar, sedangkan Mcculloch pitts menggunakan data binary. Pada Hebb Net terdapat bias, sedangkan Mcculloch-pitts tidak.

Bias adalah bobot tambahan pada suatu jaringan saraf tiruan diluar neuron yang telah ditetapkan. Fungsi dari bias dapat disamakan dengan fungsi konstan pada fungsi linear, yaitu untuk menggeser garis atau hasil agar dapat memberikan *coverage* yang lebih baik. Sama seperti bobot, bias dapat dicari pada saat proses training.

Epoch merupakan istilah dalam machine learing untuk menyebut fase dimana semua data sudah diproses sebanyak satu kali pada proses *training*. Contohnya untuk 10 data latih, ketika semuanya sudah dimasukkan pada proses training, disebut satu epoch. Jika setiap data sudah dimasukkan sebanyak 2 kali (20 pemrosesan data) disebut dua epoch dan seterusnya.