Wama = Muham			0.0.41.0.1 and memoral	1
	10142			-
301 · A	71		Contract raises	-
	FUEUTA	Netw		-/
1. diketahul:			TAR! (11) P 191	
W. = 0			dt de Not your on	
W2 = 0	*		1- 5 - + 1. Ci C - alore	6.1
		7.110.70	hopping pradice of ac	
learning rates	a:\			
epoch 23			this has been skipt	
			· Lely Ber X. Cor Mark	
data trainin		4	1. Ald Co without Weally	
X	Xa		11+1- 111+4 4	
	0.000	7-	W leine mi seise	
0		_1_		
	0	-1	4 do 25 T	
	B	-11-	HICKET (COS) 1 HOUR	
			1 1 + Walle How or	
		0	+ D+ D. 1 + D. 1 = see and 1	
E Poch \	1- day not b	outsi de	18 DEAN ENGE - MY	
	10) Target -1		total act total althou	
			projetice marketing . W.	
			Firth But ntall sol	
			0) Hisak sexual Longon Forge	<u></u>
	4/		D) (1211 121)	<u> </u>
		= 1 1	Will to the med rate	
	1.x1 =0+1.(-1).			
60 2 60 4 0 c h	12 - 04 14 (4) VA	* O	- 10 m (10) 5 degg	7 :
	20+1·-1 = -1		dtax-all to youll me	
		1 4-16	Dr. 1-+1-1+C-1 - 100 mg	
120 DOL 12010	VO(30) W220	1.707	tours one of Or o	1.7
100m . (()				
1,408 7 60	1) Targer	1 2	2 pick acts deduct adol get	1
			D. K. J. + J. = 1X1 / 12 - 1/1 = 1	
	The state of the s		1+1+1= and other	
			- ada. Perubatan karena sowal for	
bobot tetap	W1=0 W2=0	1 6	Deal Level may work	and .
Input 3 BB CA	(10) target -1			R.
	1.X, + Wasket	03	A	
(foil)			/m	

Persamaan you 20.1+0.0+	1.1	1001/01/2	London	with the production
Carpet y = -1 Charcong y ki	wary dar	O) 1 serval	target	2 1 11 21
bobot klap: Wi=0 W	laso b	17 ~		4 . 150
	toler des			
Input 4 ([11] Target				saledidik 1
You : Wiexe + We + Xo + b				0 - 114
Personnan = 0.1+0.1+-1 =-1				100 - 100
OUTUR Y = -1 Charana y kurang dar	10). Ha	t Sesual targ	et kanna	target of
			1. 1 16	1 1010 711
Update ballot dan blas				10073
W1:W+a+1.X1 20+1+1.	121			
Was Water tixe = Otl. Til :			1 Marien	nde E
b = 6+a.t = -1+1.1 =	0	L'A	1	/
bobo baru = W1=1. Wa		0200		
	1-		1	3
5 Poch 2	1-	<u> </u>		
1. lingut 1 (O.O) Target -	1 3	0		1
Ym = WiXI +Wa. Xx + b =				
personage = 1.0+1.0+0 = 0				
14th = Output Yin = O Hadk	sesual)	arget -1		/ della
update bottot dan blas		topic 1 (1 1000 1
WI = WITA. NXI = OTHER		14-10-100	الماد عا العاد	160,56,51
Waz Wata. F. X = (+1/-1.	0 =1=	0 + 0.0+	j•⊕. "	ann no my
b. b. a+1 0 b +a.1				
			1-	
bobot para Mi =1. Mi=1	b=-1	wy lack	toded	246693
		-3.110-1	y . 4 . 0 + 1	ul s ha
2 Ippor 2 (OI) Target -1	0000	1-1-1-0:	v. 1.0 +	المناط و العالم
Ym= Wixi+ Wa-Xz+b	1-	1-11-	0-1-0+	did
personaan = 1.0+1.1+-1 =01	red c	LEWY DE	W bus	1 100000
um = O Kidak Serval			•	
		beach	1100	1 10901
W1=W1+a.+.x1=1+11.0	21 1	t War X2 A	IX WY	V south
W2 : Watar.x= +1.7.1 =				mune 227
b = b + cutor = m/+ (od c = to				V = 10.00
bobok baru Will W2.01.				200,004
			the first	4 44 144 7
34		- bear (C	11/2 18	2 4007
		Viel I + V		
(Flat)			************	7 6
(Str.))				

3-1 n Put 3 (1:0) Targer -1
You : Wixi + Wexe +6
Karawau . 11 +0.0+(-7) = 1-9=1.
output you the tate and ferthalian farms of sesson target
LETE seein dendond along
4. (pruh 4 Clis) Target 1
Yin: Wixi + Wax +b
puramean, 1-1+ 20-1+-2 = 1-2=-1 purubahan karong Hotal secuci larget
gasto 1 per send law sail and sound
Elesate popot dan plas
W1 = W1+a. h.x1 = (+ 1. () = (+1 = 2
W== W=+a. f.x= = 0+1.11 =1
b: btait = -2+1:1=-1
bobok baru Wizz Wazi ba-1
Topoch
Epoch 3
1. Input 1 (010) Target -1
YIO = WIXI+WXX+b
persamaan = 2-0+1-0+ - = -1 1- 11575 (110)-1 171'.
Output the 27-1 Head and Perubation Language target.
2. Input 2 (OII) Target -1
hitung wen = Wiskit Waskerthold Helper
Piman: 2.0+1:1+-1:0
output par : O tidak sesual target -1. Update botok dar blas
W. W. + a. h. x 2+ 1 1 2+ 1 1 2 2 7 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Wa = 1 Ala + a, +, x = 1 + 1 + -1 + 1 . = 0
b 2 b ta. h = - + 1. + = 8-2 - 49707 (010) + 4071 +
20bot bar W1 2 W2 60 6-2+27 - 1- 1- +0.
3- I ppv + 3 (110) Tars ex = -1
hiring you a Wickit Weeksto
persamoun: 2-1+0.0+-2=0
Outpop = - O Mitadio se mais fargetion 1 Lot 1800 200 5
UP-late bobot dan bigs
2 W1 = W1 ta. +. X1 = 2 + 1: -1 = 1
Wa= Wa+a.t.x=0+1-4.0 =0
b = b + a + = -2+ 1 - 1 = -3
bobot baru Will Ward bas
the state of the s

9. Input 9 (1) Target 1	to be self the self of the self of
maria. W. V. + W. R. + h	14 O. 11 4 O. 11 . 11 -
	- 1 +21 14-9 - Q - C + 1 - 1 18 - MESE
Outros was + 4 (karen vin 40) .	" fixat swal target -
Uptate bobot dan blas	
W1 = W1+a.t.x1 = 1+1.1.1	1 400 1 Still & - 501 A
Wa= Wata++1x= O+ 1.1-1 = 1	The state of the s
1 1 I delegately and do	The state of the s
babah bass Wind Weel	2
- CO BOY CAPINI DVI32 DOS	111011111111111111111111111111111111111
2. data Uji	1-1-1-1-5-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1
X1 X2 Fag	of 1-10/+1-0 +0+0+0
6 1 1	1-10 1:21/ 51 NO 0200 -00000
1 0 1-1	15-61-1
	1 1009 3
0 0 -1	1- Spant (0.0) 1 20 year -1
	3 + 140 NV + 140 NV = 218
1. Inoch 1 & Oct) Target -	(- = 1-+0.1+0.6 - 1100102109
11 N. X. 1 11. 1X + 14 + 2 · O + 1	· Lt -2 = A Change Change
W(!X(+wx //2 (S	
2- Inpur 2 (10) Target -1	1- 40p 407 (110) 1 4097/ A
2.1+1.0+-2.50 (50	Mah Scharesoras -1) 14.11 a control
• • • • • • • • • • • • • • • • • • • •	1 - 1111 + 1 a 100011
3. Input & G (1) Target 190.	- typing point John! O . man to 9 7 10
2.1 + 1.1+ -2 = + (borger)	- Francis IV W
Di (C ()	00 11-11/2 X.1.0+W. W.
4- Input & (010) Target -1	5- 6. Hilt a firt deal
2.04 @ 1.04 -2 = -2 . 001	194 - (Chanar) . W wood 1947
2010	1- 4. not (Od) : Ival.
prediksi bonar = 3.	1+ xx +1x +1x +1 + 1x + +1
-11 1/2 1	Car be + 0.0 + la C monerly
3 100 = 7506	adi aturasi pado dalla USI 70%
4 100 = 13.90	loted not noted stated
	1 . 1. F. 1 + A . 1x . f. o. t. W W
	0 = 0. HU + 0 = 13 + 10 + 10 V = 18 V
	8-10- + 1-1+5- 1.0+d= V
	6000 Low Will Was 1000
	CV CINO INV
(SIDU)	

-	
Prediksi Pad	a data latinan
1. I nout 1 C	OIO) target -1 (benar)
20+1.0	(henor)
2. Input 1 (0	11) Target [[benar]
2.0+1.1+	-1 = -1 (banar)
3 Indat 3 Cl	10) Targer -1
7.141.0+	-1 = 0 (salah seharumia -1)
4. INMY 4 C	1,1) Target 1
2.1+1.1+	2. ((bornar)
Variation 1	2
Pruises 1 Total 2	
3 4/4	0 : 70 0/4
4	0 = 75 %
3 - a. A kurası	Pada data training = 75 %.
le. Akurau t	Pada daha Uil = 75 %
c berhaul	different for Lotationing don tota (1) model marginasium
c berhaul	different for Lotationing don tota (1) model marginasium
c berhaul	jang sama daga antara data Uji dengan data training
c berhaul nilal y	different for Lotationing don tota (1) model marginasium
c berhaul nilal y	different for Lotationing don tota (1) model marginasium
c berhaul nila y	different for Lotationing don tota (1) model marginasium
c berhaul nilae y	different for Lotationing don tota (1) model marginasium
c berhaul nilai y	different for Lotationing don tota (1) model marginasium
c berhaul	different for Lotationing don tota (1) model marginasium
c berhaul nilai y	different for Lotationing don tota (1) model marginasium
c berhaul nilai y	different for Lotationing don tota (1) model marginasium
c borhaul nilai y	different for Lotationing don tota (1) model marginasium
c borhaul nilai y	different for Lotationing don tota (1) model marginasium
c berhaul nilai y	different for Lotationing don tota (1) model marginasium
c borhaul nilai y	different for Lotationing don tota (1) model marginasium
c borhaul nau y	different for Lotationing don tota (1) model marginasium
c berhaul naa y	different for Lotationing don tota (1) model marginasium
c borhaul nilai y	different for Lotationing don tota (1) model marginasium
C berhaul nilai y	different for Lotationing don tota (1) model marginasium

```
Code
import matplotlib.pyplot as plt
def activation function(y in):
  """Fungsi Activasi tiga nilai."""
  if y_in > 0:
     return 1
  elif 0 <= y_in <= 1:
     return 0
  else:
     return -1
def train_perceptron(data, learning_rate, epochs):
  """Melatih perceptron dengan data yang diberikan"""
  weights = [0, 0] # w1, w2
  bias = 0
  for epoch in range(epochs):
     print(f"Epoch {epoch + 1}")
     for x1, x2, target in data:
       # Hitung Input net
       y_in = bias + (weights[0] * x1 + weights[1] * x2)
       # Hitung Output
       output = activation_function(y_in)
       # Update bobot dan bias jika terjadi error
       if output != target:
          weights[0] += learning_rate * target * x1
          weights[1] += learning rate * target * x2
         bias += learning_rate * target
          print(f"Updated weights: {weights}, bias: {bias}")
       else:
         print("No updates")
  return weights, bias
def calculate_accuracy(data, weights, bias):
  """Menghitung akurasi model."""
  correct\_prediction = 0
  total\_predictions = len(data)
```

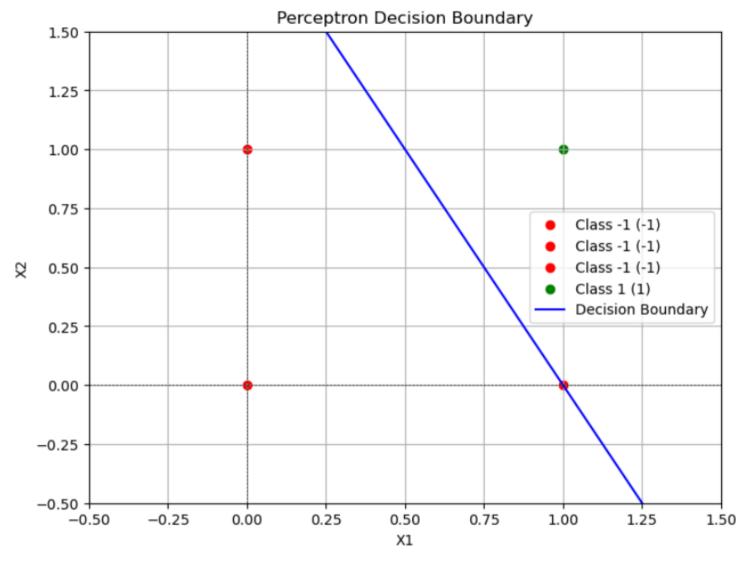
```
for x1, x2, target in data:
     y_{in} = bias + (weights[0] * x1 + weights[1] * x2)
     output = activation function(y in)
     if output == target:
       correct prediction += 1
  accuracy = (correct_prediction / total_predictions) * 100
  return accuracy
def predict(weights, bias, test_data):
  """Melakukan prediksi data uji"""
  predictions = []
  for x1, x2 in test data:
     y in = bias + (weights[0] * x1 + weights[1] * x2)
     output = activation_function(y_in)
     predictions.append(output)
  return predictions
def plot_decision_boundary(data, weights, bias):
  """Memvisualisasi data dan batas keputusan"""
  plt.figure(figsize=(8, 6))
  for x1, x2, target in data:
    if target == 1:
       plt.scatter(x1, x2, color='green', label='Class 1 (1)' if 'Class 1' not in plt.gca().get_legend_handles_labels()[1] else "")
     elif target == -1:
       plt.scatter(x1, x2, color='red', label='Class -1 (-1)' if 'Class -1' not in plt.gca().get legend handles labels()[1] else "")
  # Garis pemisah
  x = [-0.5, 1.5]
  y = [-(weights[0] * xi + bias) / weights[1] if weights[1] != 0 else None for xi in x]
  plt.plot(x, y, label='Decision Boundary', color='blue')
  # Setting plot
  plt.xlim(-0.5, 1.5)
  plt.ylim(-0.5, 1.5)
  plt.xlabel('X1')
  plt.ylabel('X2')
  plt.title('Perceptron Decision Boundary')
  plt.axhline(0, color='black', linewidth=0.5, ls='--')
  plt.axvline(0, color='black', linewidth=0.5, ls='--')
  plt.grid()
```

```
plt.legend()
  plt.show()
# Data input
data = [
  [0, 0, -1],
  [0, 1, -1],
  [1, 0, -1],
  [1, 1, 1]
# Parameter pelatihan
learning\_rate = 1
epochs = 3
# Melatih perceptron
final_weights, final_bias = train_perceptron(data, learning_rate, epochs)
# Menampilkan bobot dan bias akhir
print(f"Final weights: {final weights}, Final bias: {final bias}")
# Menghitung akurasi
accuracy = calculate_accuracy(data, final_weights, final_bias)
print(f"Akurasi: {accuracy:.2f}%")
# Memvisualisasikan hasil
plot_decision_boundary(data, final_weights, final_bias)
# Data Uji
test_data = [
  [0, 1],
  [1, 0],
  [1, 1],
  [0, 0],
  [0.5, 0.5] # Contoh data baru
# Melakukan prediksi
predictions = predict(final_weights, final_bias, test_data)
for i, (x1, x2) in enumerate(test_data):
  print(f"Prediksi untuk input ({x1}, {x2}): {predictions[i]}")
```

Hasil

```
Epoch 1
Updated weights: [0, 0], bias: -1
No updates
No updates
Updated weights: [1, 1], bias: 0
Epoch 2
Updated weights: [1, 1], bias: -1
Updated weights: [1, 0], bias: -2
No updates
Updated weights: [2, 1], bias: -1
Epoch 3
No updates
Updated weights: [2, 0], bias: -2
Updated weights: [1, 0], bias: -3
Updated weights: [2, 1], bias: -2
Final weights: [2, 1], Final bias: -2
Akurasi: 75.00%
```

Akurasi: 75.00%



Prediksi untuk input (0, 1): -1 Prediksi untuk input (1, 0): 0 Prediksi untuk input (1, 1): 1 Prediksi untuk input (0, 0): -1 Prediksi untuk input (0.5, 0.5): -1

```
Code
import numpy as np
# Fungsi aktivasi (step function)
def step function(x):
  return np.where(x \ge 0, 1, 0)
# Perceptron Training Function
def perceptron_train(X, y, lr, epochs):
  # Inisialisasi bobot dengan nilai random kecil
  weights = np.random.rand(X.shape[1])
  bias = np.random.rand()
  # Training loop
  for epoch in range(epochs):
     print(f"Epoch {epoch + 1}/{epochs}")
    for i in range(len(X)):
       # Hitung output
       linear\_output = np.dot(X[i], weights) + bias
       prediction = step_function(linear_output)
       # Update bobot jika terjadi kesalahan
       error = y[i] - prediction
       weights += lr * error * X[i]
       bias += lr * error
     # Cetak bobot dan bias di setiap epoch
     print(f"Weights: {weights}, Bias: {bias}\n")
  return weights, bias
# Perceptron Prediction Function
def perceptron predict(X, weights, bias):
  linear\_output = np.dot(X, weights) + bias
  return step_function(linear_output)
# Input dari keyboard
lr = float(input("Masukkan nilai learning rate: "))
epochs = int(input("Masukkan jumlah epoch: "))
# Contoh data (X1, X2, X3)
X = np.array([
```

```
[0, 0, 0],
  [0, 0, 1],
  [0, 1, 0],
  [0, 1, 1],
  [1, 0, 0],
  [1, 0, 1],
  [1, 1, 1]
1)
# Target output (label dua kelas)
y = np.array([0, 0, 0, 1, 0, 1, 1])
# Training Perceptron
weights, bias = perceptron_train(X, y, lr, epochs)
# Prediksi menggunakan bobot yang sudah dilatih
print("\nPrediksi setelah training:")
for i in range(len(X)):
  prediction = perceptron\_predict(X[i], weights, bias)
  print(f"Input: {X[i]} -> Kelas Data: {y[i]} -> Prediksi: {prediction}")
# Input untuk Data test baru dari keyboard
X_test1 = int(input("Masukkan nilai Xtest1: "))
X_test2 = int(input("Masukkan nilai Xtest2: "))
X_test3 = int(input("Masukkan nilai Xtest3: "))
# Gabungkan input test ke dalam satu array
X \text{ test} = \text{np.array}([X_{\text{test}1}, X_{\text{test}2}, X_{\text{test}3}])
# Prediksi untuk data tes
test_prediction = perceptron_predict(X_test, weights, bias)
print(f"\nPrediksi untuk input tes [{X_test1}, {X_test2}, {X_test3}] adalah: {test_prediction}")
```

Output

```
Masukkan nilai learning rate: 1
Masukkan jumlah epoch: 3
Epoch 1/3
Weights: [-0.55316793 1.2085659 1.56477491], Bias: -0.9998231154189561
Epoch 2/3
Weights: [-0.55316793 2.2085659
                                 1.56477491], Bias: -0.9998231154189561
Epoch 3/3
Weights: [0.44683207 2.2085659 2.56477491], Bias: -0.9998231154189563
Prediksi setelah training:
Input: [0 0 0] -> Kelas Data: 0 -> Prediksi: 0
Input: [0 0 1] -> Kelas Data: 0 -> Prediksi: 1
Input: [0 1 0] -> Kelas Data: 0 -> Prediksi: 1
Input: [0 1 1] -> Kelas Data: 1 -> Prediksi: 1
Input: [1 0 0] -> Kelas Data: 0 -> Prediksi: 0
Input: [1 0 1] -> Kelas Data: 1 -> Prediksi: 1
Input: [1 1 1] -> Kelas Data: 1 -> Prediksi: 1
Masukkan nilai Xtest1: 1
Masukkan nilai Xtest2: 2
Masukkan nilai Xtest3: 3
Prediksi untuk input tes [1, 2, 3] adalah: 1
```