Penerapan Metode Hebb Rule Menggunakan Python Untuk Pengujian Pengenalan Pola Huruf Hijaiyyah

Nazaruddin Ahmad[1], Fajar[2], Zidan Mubarak[3], M. Bilal Akbar[4]
[1,2,3,4] Teknologi Informasi, Universitas Islam Negeri Ar-Raniry Banda Aceh, Indonesia
Email: [1]nazar.ahmad@ar-raniry.ac.id, [2]faajar12052021@gmail.com, [3]zidanmubarak00@gmail.com, [4]bilalakbar1506@gmail.com

Abstrak – Pengenalan pola memegang peranan penting dalam pengembangan jaringan saraf tiruan terutama untuk aksara non-latin seperti huruf Hijaiyyah yang memiliki bentuk visual kompleks. Penelitian ini bertujuan mengkaji efektivitas algoritma pembelajaran Hebbian dalam mengenali huruf waw (و) dan ba (ب) melalui representasi numerik. Metode yang dilakukan melibatkan Konversi bentuk huruf ke matriks 5x5 bernilai bipolar, pembaruan bobot menggunakan aturan Hebb, serta penerapan fungsi aktivasi *threshold*. Simulasi dilakukan menggunakan bahasa pemrograman Python dan diverifikasi secara manual. Hasil menunjukkan bahwa algoritma Hebbian mampu mengenali kedua huruf dengan akurasi tinggi, dimana representasi bipolar memberikan keluaran yang stabil. Dengan demikian, pendekatan Hebbian efektif untuk pengenalan pola huruf Hijaiyyah dan berpotensi dikembangkan untuk aksara non-latin lainnya.

Kata Kunci – ***Hebbian, Huruf Hijaiyah, Jaringan Saraf Tiruan, Pengenalan Pola, Python***

Pattern recognition plays an important role in the development of artificial neural networks, especially for non-Latin script such as Hijaiyyah letters, which have complex visual forms. This study aims to examine the effectiveness of the Hebbian learning algorithm in recognizing the letters waw (و) and ba (ب) through numerical representation. The method involves converting letter shapes into 5x5 bipolar matrices, updating weigthts using the Hebb rule, and applying a threshold activation function. The simulation was carried out using the Python programming language and manually verified. The results show that the Hebbian algorithm is capable of recognizing both letters with high accuracy, where bipolar representation contributes to stable outputs. Therefore, the Hebbian approach is effective for recognizing Hijayyah letters patterns and has the potential to be developed for other non-Latin scripts.

Keywords – Hebbian, Hijaiyah Letters, Artificial Neural Networks, Pattern recognition, Python.

I. PENDAHULUAN

Karakteristik jaringan saraf tiruan menyerupai cara kerja jaringan saraf biologi untuk melakukan pemrosesan informasi [1]. Jaringan saraf tiruan memiliki lapisan-lapisan yang terdiri dari lapisan input, lapisan tersembunyi, dan lapisan output [2]. Pemrosesan di dalam konsep jaringan saraf tiruan dibagi menjadi dua bagian yaitu pelatihan (*traning*) dan pengujian (*testing*) yang sebelumnya data dibagi menjadi dua yaitu data latih dan data uji [3]. Jaringan saraf tiruan dapat mempelajari dan mengembangkan sesuatu dengan metode pelatihan. Hal ini dapat dilakukan untuk melakukan pengenalan pola, melatih dari data-data yang telah diinputkan dan dapat untuk memprediksi di masa yang akan datang berdasarkan proses pelatihan yang sudah dilakukan [4].

Algoritma yang digunakan dalam jaringan saraf tiruan mengandalkan elemen-elemen dasar pemrosesan ini untuk membantu mempercepat dan meningkatkan akurasi pemrosesan informasi [5]. Pemodelan ini terinspirasi dari kemampuan otak manusia dalam mengatur sel-sel penyusunnya, yaitu neuron, sehingga dapat menjalankan berbagai tugas, terutama dalam mengenali pola dengan tingkat efektivitas yang sangat tinggi [6]. Jaringan saraf tiruan memiliki 2 metode yaitu metode pembelajaran terbimbing (*Supervised learning*) dan metode pembelajaran tidak terbimbing (*Unsupervised learning*) [7].

Jaringan saraf tiruan terdiri dari kumpulan elemen pemrosesan yang saling terhubung, yaitu neuron-neuron yang bekerja secara kolaboratif untuk memecahkan berbagai masalah kompleks. Sebagai model matematika yang canggih, jaringan ini dibangun melalui tiga lapisan utama, yaitu lapisan input (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*), yang setiap lapisan memiliki peran penting dalam pengolahan informasi [8].

JST sering digunakan untuk mempelajari pola dan hubungan dalam data yang kompleks seperti pengenalan pola, prediksi, dan klasifikasi. Dalam proses pembelajaran, JST dilatih menggunakan data berlabel atau keluaran yang diinginkan, memungkinkan JST untuk mempelajari pola dan hubungan dalam data yang kompleks dan digunakan untuk memprediksi hasil atau hasil berdasarkan masukan [9].

Pengenalan pola merupakan komponen yang krusial dalam pengembangan jaringan saraf tiruan, terutama dalam analisis data visual yang melibatkan aksara non-Latin. Meskipun telah terjadi kemajuan dalam bidang pembelajaran mesin dan penglihatan komputer (*computer vision*), sebagian besar sistem pengenalan yang ada saat ini lebih dioptimalkan untuk karakter berbasis Latin, sehingga terdapat kesenjangan signifikan dalam kemampuan mengenali dan memproses aksara seperti huruf Arab. Huruf-huruf Hijaiyyah, sebagai bagian integral dari aksara Arab, memiliki bentuk visual yang kompleks dan khusus, termasuk variasi halus yang sering kali sulit dibedakan oleh

algoritma pengenalan konvensional. Tentunya tantangan ini menegaskan perlunya pendekatan khusus yang disesuaikan dengan karakteristik struktural aksara non-Latin.

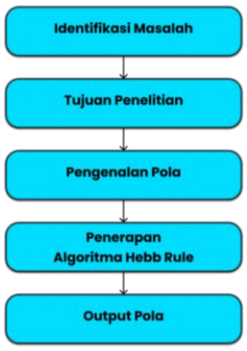
Salah satu metode di dalam konsep jaringan saraf tiruan adalah metode Hebb Rule. Metode Hebb Rule ini adalah metode yang proses pembelajarannya menyerupai proses pembelajaran otak manusia [10]. Hebb rule di dalam konsep jaringan saraf tiruan dikategorikan ke dalam metode pembelajaran yang paling sederhana [1]. Dimana pada model hebb-rule ini memiliki beberapa unit masukan yang dihubungkan langsung dengan sebuah unit keluaran yang digunakan untuk menghitung bobot dan bias secara berkelanjutan [11].

Konsep jaringan saraf tiruan dapat digunakan untuk melakukan proses pengenalan pola untuk melakukan klasifikasi dari sebuah objek [12]. Meskipun berbagai metode pemebelajaran terbimbing (*supervised learning*) seperti SVM, CNN, dan KNN telah banyak diterapkan dalam pengenalan huruf hijaiyyah, pendekatan alternatif yang bersifat biologis seperti Hebb Rule masih jarang dieksplorasi. Padahal, Hebb Rule menawarkan mekanisme asosiasi yang menyerupai proses pembelajaran alami di otak manusia. Hingga saat ini, penerapan Hebb Rule dalam konteks pengenalan huruf hijaiyyah masih belum banyak diteliti, baik dari sisi efektivitas, akurasi, maupun potensinya sebagai metode yang ringan dan efisien. Sehingga, penelitian ini diharapkan mampu memberikan solusi yang efisien dalam pengenalan pola huruf hijaiyyah.

II. METODE PENELITIAN

A. Tahapan Penelitian

Hebbian rule merupakan salah satu teknik yang memiliki potensi besar dalam mengubah sinapsis berdasarkan pola aktivitas yang terjadi antar neuron [13]. Menurut [14] algoritma Hebb rule adalah metode pembelajaran yang sederhana namun efektif. Prinsip kerja dari algoritma ini adalah jika dua neuron yang terhubung melalui sinapsis aktif bersamaan dengan nilai yang serupa, baik positif maupun negatif, kekuatan sinapsis di antara keduanya akan meningkat [15]. Sebaliknya, jika kedua neuron aktif secara tidak bersamaan dengan nilai yang berbeda, salah satu positif dan lainnya negatif, kekuatan sinapsisnya akan melemah. Cara kerja dari metode Hebb Rule ini adalah memperbaiki nilai bobot terhadap neuron-neuron yang saling terhubung [16].



Gambar 1. Tahapan Penelitian

1. Identifikasi Masalah

   Pengenalan pola huruf hijaiyah, khususnya ba dan waw, memiliki tantangan tersendiri. Penelitian ini bertujuan menguji apakah algoritma Hebb dapat mengenali pola kedua huruf tersebut dalam representasi matriks 5x5.

2. Tujuan Penelitian

   Penelitian ini bertujuan membuktikan efektivitas algoritma Hebb dalam mengenali pola huruf hijaiyah berbasis citra, sehingga dapat berkontribusi pada pengembangan sistem pengenalan pola aksara non-latin.

3. Pengenalan Pola

   Pola huruf ba dan waw direpresentasikan dalam bentuk matriks biner 5x5. Representasi ini diolah untuk menghasilkan pola numerik yang sesuai dengan karakteristik algoritma Hebb. Kemudian didapatkan pola untuk huruf ba dan huruf waw, tampak seperti gambar berikut ini:

Gambar 2. (a) Huruf waw dan (b) huruf ba

4. Penerapan Algoritma Hebb Rule

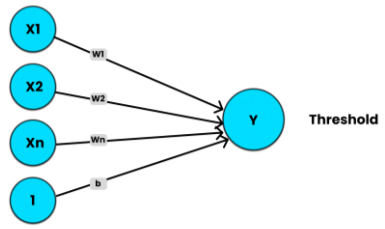
   Algoritma Hebb bekerja dengan memperbarui bobot sinaptik berdasarkan masukan dan keluaran yang diharapkan. Tahapan meliputi:

   1. Representasi pola huruf dalam matriks biner 5x5.
   2. Pembaruan bobt sinaptik berdasarkan formula Hebb.
   3. Validasi hasil keluaran terhadap pola target.

5. Output Pola

   Penelitian ini bertujuan memastikan algoritma Hebb dapat mengenali pola huruf ba dan waw secara akurat, dengan hasil yang divalidasi terhadap pola target.

Dalam jaringan Hebb, bobot dan bias diperbarui melalui iterasi selama proses pelatihan. Proses ini menggunakan fungsi aktivasi berbasis ambang batas (threshold) sebesar 0. Arsitektur jaringan saraf tiruan model Hebb ditujukan pada gambar 3.



Gambar 3. Arsitektur Jaringan Hebb

1. Perbaikan bobot dilakukan dengan metode berikut:

   Bobot baru (wi yang diperbarui) dihitung dengan cara: wi (baru) = wi (lama) + xi * y. b. Nilai bias baru (b yang diperbarui) dihitung dengan cara: b (baru) = b (lama) + y.

2. Dalam rumus-rumus berikut:

   a. W_i adalah bobot yang terkait dengan data input ke-i.

   b. X_i adalah neuron yang menjadi nilai input dari data ke-i.

   c. Y adalah output data.

   d. B adalah nilai bias[17].

Hipotesis Hebb adalah apabila dua buah neuron bipolar yang saling terhubung mempunyai aktivasi (output) yang sama, maka bobot sambungan antara kedua neuron tersebut akan menguat. Sebaliknya, apabila aktivasi keduanya berlawanan, maka bobot sambungan akan melemah[18].

III. HASIL DAN PEMBAHASAN

A. Hasil Perhitungan Manual

Proses pengenalan pola huruf *waw* (و) dan *ba* (ب) dilakukan menggunakan pendekatan penyajian data bipolar. Dalam metode ini, pola huruf direpresentasikan dalam bentuk matriks berukuran 5x5, dimana nilai 1 menunjukkan keberadaan pola (objek), sedangkan nilai -1 untuk mewakili ketidakberadaan pola (background).

Penggunaan algoritma Hebb Rule menjadi inti dari proses ini. Setiap elemen dalam matriks, yang berjumlah 25, bertindak sebagai input bagi algoritma. Algoritma ini dipilih karena kemampuannya untuk mempelajari pola melalui penyesuaian bobot secara iteratif berdasarkan hubungan linear antara input dan target output.

Adapun matriks 5x5 yang digunakan menggambarkan pola karakter dengan jelas dan mendetail, sehingga memungkinkan algoritma untuk pola huruf dengan akurasi yang baik. Ilustrasi dari matriks input 25 elemen tersebut dapat dilihat pada gambar 4.

x1	x2	x3	x4	x5
x6	x7	x8	x9	x10
x11	x12	x13	x14	x15
x16	x17	x18	x19	x20
x21	x22	x23	x24	x25

x1	x2	x3	x4	x5
x6	x7	x8	x9	x10
x11	x12	x13	x14	x15
x16	x17	x18	x19	x20
x21	x22	x23	x24	x25

Gambar 3. Pola huruf waw dan ba

Kemudian, dilakukan perubahan menjadi representasi nilai bipolar, dimana yang menjadi objek dari huruf hijaiyah diberi nilai 1, dan yang bukan objek huruf hijaiyah diberi nilai -1. Hal ini dapat dilihat seperti gambar 5.

-1	-1	1	1	-1
-1	1	-1	-1	1
-1	-1	1	1	1
1	-1	-1	-1	1
-1	1	1	1	-1

1	-1	-1	-1	1
1	-1	-1	-1	1
-1	1	1	1	-1
-1	-1	-1	-1	-1
-1	-1	1	-1	-1

Gambar 4. Pola huruf waw dan ba

Dibawah ini adalah hasil perpaduan pola huruf waw (و) dan ba (ب) yang dimasukkan ke dalam tabel. Hal ini dapat dilihat pada tabel 1.

Tabel 1. Hasil Perpaduan Pola Huruf Waw dan Ba

Pola	و	ب	*Pola*	و	ب
x_1	-1	1	x_{14}	1	1
x_2	-1	-1	x_{15}	1	-1
x_3	1	-1	x_{16}	1	-1
x_4	1	-1	x_{17}	-1	-1
x_5	-1	1	x_{18}	-1	-1
x_6	-1	1	x_{19}	-1	-1
x_7	1	-1	x_{20}	1	-1
x_8	-1	-1	x_{21}	-1	-1
x_9	-1	-1	x_{22}	1	-1
x_{10}	1	1	x_{23}	1	1
x_{11}	-1	-1	x_{24}	1	1
x_{12}	1	1	x_{25}	-1	-1
x_{13}	1	1	*Target*	1	-1

Setelah melakukan konversi ke bipolar, maka dilakukan inisialisasi terlebih dahulu untuk:

bobot(w) = 0

bias(b) = 0

Setelah diinisialisasikan keduanya, selanjutnya mencari nilai perubahan bobot dan biasnya. Rumus yang digunakan adalah:

$$wi\ (baru) = wi(lama) + (xi.y) \quad (1)$$

Perhitungan huruf *waw* (و):

$\Delta w_1 = w_1(lama) + (x_1 * y)$
$= 0 + ((-1) * 1)$
$= -1$

$\Delta w_2 = w_2(lama) + (x_2 * y)$
$= 0 + ((-1) * 1)$
$= -1$

$\Delta w_3 = w_3(lama) + (x_3 * y)$
$= 0 + (1 * 1)$
$= 1$

$\Delta w_4 = w_4(lama) + (x_4 * y)$
$= 0 + (1 * 1)$
$= 1$

$\Delta w_5 = w_5(lama) + (x_5 * y)$
$= 0 + ((-1) * 1)$
$= -1$

$\Delta w_6 = w_6(lama) + (x_6 * y)$
$= 0 + ((-1) * 1)$
$= -1$

$\Delta w_7 = w_7(lama) + (x_7 * y)$
$= 0 + (1 * 1)$
$= 1$

$\Delta w_8 = w_8(lama) + (x_8 * y)$
$= 0 + ((-1) * 1)$
$= -1$

$\Delta w_9 = w_9(lama) + (x_9 * y)$
$= 0 + ((-1) * 1)$
$= -1$

$\Delta w_{10} = w_{10}(lama) + (x_{10} * y)$
$= 0 + (1 * 1)$
$= 1$

$\Delta w_{11} = w_{11}(lama) + (x_{11} * y)$
$= 0 + ((-1) * 1)$
$= -1$

$\Delta w_{12} = w_{12}(lama) + (x_{12} * y)$
$= 0 + ((-1) * 1)$
$= -1$

$\Delta w_{13} = w_{13}(lama) + (x_{13} * y)$
$= 0 + (1 * 1)$
$= 1$

$\Delta w_{14} = w_{14}(lama) + (x_{14} * y)$
$= 0 + (1 * 1)$
$= 1$

$\Delta w_{15} = w_{15}(lama) + (x_{15} * y)$
$= 0 + (1 * 1)$
$= 1$

$\Delta w_{16} = w_{16}(lama) + (x_{16} * y)$
$= 0 + (1 * 1)$
$= 1$

$\Delta w_{17} = w_{17}(lama) + (x_{17} * y)$
$= 0 + ((-1) * 1)$
$= -1$

$\Delta w_{18} = w_{18}(lama) + (x_{18} * y)$
$= 0 + ((-1) * 1)$
$= -1$

$\Delta w_{19} = w_{19}(lama) + (x_{19} * y)$
$= 0 + ((-1) * 1)$
$= -1$

$\Delta w_{20} = w_{20}(lama) + (x_{20} * y)$
$= 0 + (1 * 1)$
$= 1$

$\Delta w_{21} = w_{21}(lama) + (x_{21} * y)$
$= 0 + ((-1) * 1)$
$= -1$

$\Delta w_{22} = w_{22}(lama) + (x_{22} * y)$
$= 0 + (1 * 1)$
$= 1$

$\Delta w_{23} = w_{23}(lama) + (x_{23} * y)$
$= 0 + (1 * 1)$
$= 1$

$\Delta w_{24} = w_{24}(lama) + (x_{24} * y)$
$= 0 + (1 * 1)$
$= 1$

$\Delta w_{25} = w_{25}(lama) + (x_{25} * y)$
$= 0 + ((-1) * 1)$
$= -1$

Proses perhitungan diatas dilakukan untuk mencari perubahan bobot (Δw) pada huruf *waw* (و), selanjutnya mencari bobot baru untuk huruf *ba* (ب) dengan menggunakan rumus:

$$wi\ (baru) = wi(lama) + (xi.y) \qquad (2)$$

Perhitungan huruf *ba* (ب):

$\Delta w_1 = w_1(lama) + (x_1 * y)$
$= (-1) + (1 * (-1))$
$= -2$

$\Delta w_2 = w_2(lama) + (x_2 * y)$
$= (-1) + ((-1) * (-1))$
$= 0$

$\Delta w_3 = w_3(lama) + (x_3 * y)$
$= 1 + ((-1) * (-1))$
$= 2$

$\Delta w_4 = w_4(lama) + (x_4 * y)$
$= 1 + ((-1) * (-1))$
$= 2$

$\Delta w_5 = w_5(lama) + (x_5 * y)$
$= (-1) + (1 * (-1))$
$= -2$

$\Delta w_6 = w_6(lama) + (x_6 * y)$
$= (-1) + (1 * (-1))$
$= -2$

$\Delta w_7 = w_7(lama) + (x_7 * y)$
$= 1 + ((-1) * (-1))$
$= 2$

$\Delta w_8 = w_8(lama) + (x_8 * y)$
$= (-1) + ((-1) * (-1))$
$= 0$

$\Delta w_9 = w_9(lama) + (x_9 * y)$
$= (-1) + ((-1) * (-1))$
$= 0$

$\Delta w_{10} = w_{10}(lama) + (x_{10} * y)$
$= 1 + (1 * (-1))$
$= 0$

$\Delta w_{11} = w_{11}(lama) + (x_{11} * y)$
$= (-1) + ((-1) * (-1))$
$= 0$

$\Delta w_{12} = w_{12}(lama) + (x_{12} * y)$
$= (-1) + (1 * (-1))$
$= -2$

$\Delta w_{13} = w_{13}(lama) + (x_{13} * y)$
$= 1 + (1 * (-1))$
$= 0$

$\Delta w_{14} = w_{14}(lama) + (x_{14} * y)$
$= 1 + (1 * (-1))$
$= 0$

$\Delta w_{15} = w_{15}(lama) + (x_{15} * y)$
$= 1 + ((-1) * (-1))$
$= 2$

$\Delta w_{16} = w_{16}(lama) + (x_{16} * y)$
$= 1 + ((-1) * (-1))$
$= 2$

$\Delta w_{17} = w_{17}(lama) + (x_{17} * y)$
$= (-1) + ((-1) * (-1))$
$= 0$

$\Delta w_{18} = w_{18}(lama) + (x_{18} * y)$
$= (-1) + ((-1) * (-1))$
$= 0$

$\Delta w_{19} = w_{19}(lama) + (x_{19} * y)$
$= (-1) + ((-1) * (-1))$
$= 0$

$\Delta w_{20} = w_{20}(lama) + (x_{20} * y)$
$= 1 + ((-1) * (-1))$
$= 2$

$\Delta w_{21} = w_{21}(lama) + (x_{21} * y)$
$= 1 + ((-1) * (-1))$
$= 2$

$\Delta w_{22} = w_{22}(lama) + (x_{22} * y)$
$= 1 + ((-1) * (-1))$
$= 2$

$\Delta w_{23} = w_{23}(lama) + (x_{23} * y)$
$= 1 + (1 * (-1))$
$= 0$

$\Delta w_{24} = w_{24}(lama) + (x_{24} * y)$
$= 1 + ((-1) * (-1))$
$= 2$

$\Delta w_{25} = w_{25}(lama) + (x_{25} * y)$
$= (-1) + ((-1) * (-1))$
$= 0$

Setelah didapat bobot baru pada huruf *ba* (ب), langkah selanjutnya mencari nilai bias baru dengan rumus:

$$b\ (baru) = b(lama) + y \qquad (3)$$

$b_1(baru) = b_0(lama) + y_1$
$= 0 + 1$
$= 1$

$b_2(baru) = b_1(lama) + y_2$
$= 1 + (-1)$
$= 0$

Pada bagian diatas telah dilakukan perhitungan untuk mencari bobot baru dan bias baru pada huruf *waw* (و), dilakukan dengan cara yang sama untuk pencarian bobot baru pada huruf *ba* (ب). Setelah pencarian bobot baru dan bias baru dari keduanya maka didapatkan bobot dan bias baru. Hal ini dapat dilihat dari table 2.

Tabel 2. Hasil Pencarian Bobot dan Bias Baru

Pola	*w(lama)*	و	ب	*Pola*	*w(lama)*	و	ب
Δw_1	0	-1	-2	Δw_{14}	0	1	0
Δw_2	0	-1	0	Δw_{15}	0	1	2
Δw_3	0	1	2	Δw_{16}	0	1	2
Δw_4	0	1	2	Δw_{17}	0	-1	0
Δw_5	0	-1	-2	Δw_{18}	0	-1	0
Δw_6	0	-1	-2	Δw_{19}	0	-1	0
Δw_7	0	1	2	Δw_{20}	0	1	2
Δw_8	0	-1	0	Δw_{21}	0	-1	0
Δw_9	0	-1	0	Δw_{22}	0	1	2
Δw_{10}	0	1	0	Δw_{23}	0	1	0
Δw_{11}	0	-1	0	Δw_{24}	0	1	2
Δw_{12}	0	1	0	Δw_{25}	0	-1	0
Δw_{13}	0	1	0	*Bias*	0	1	0

Setelah didapat bobot baru dan bias baru, selanjutnya mencari untuk mendapatkan nilai n dengan rumus sebagai berikut:

$$n = xi * wi\ (baru) + b \quad (3)$$

Perhitungan huruf waw:

n = ((-1).2) + ((-1).0) + (1.(-2)) + (1.(-2)) + ((-1).2) + ((-1).2) + (1.(-2)) + ((-1).0) + ((-1).0) + (1.0) + ((-1).0) + ((-1).2) + (1.0) + (1.0) + (1.(-2)) + (1.(-2)) + ((-1).0) + ((-1).0) + ((-1).0) + (1.(-2)) + ((-1).0) + (1.(-2)) + (1.0) + (1.(-2)) + ((-1).0) + 0

= 2 + 0 + 2 + 2 + 2 + 2 + 2 + 0 + 0 + 0 + 0 + 2 + 0 + 0 + 2 + 2 + 0 + 0 + 0 + 2 + 0 + 2 + 0 + 2 + 0 + 0

= 24

Perhitungan huruf waw:

n = (1.2) + ((-1).0) + ((-1).(-2)) + ((-1).(-2)) + (1.2) + (1.2) + ((-1).(-2)) + ((-1).0) + ((-1).0) + (1.0) + ((-1).0) + (1.2) + (1.0) + (1.0) + ((-1).(-2)) + ((-1).(-2)) + ((-1).0) + ((-1).0) + ((-1).0) + ((-1).(-2)) + ((-1).0) + ((-1).(-2)) + (1.0) + ((-1).(-2)) + ((-1).0) + 0

= (-2) + 0 + (-2) + (-2) + (-2) + (-2) + (-2) + 0 + 0 + 0 + 0 + (-2) + 0 + 0 + (-2) + (-2) + 0 + 0 + 0 + (2) + 0 + (-2) + 0 + (-2) + 0

= -24

Dari perhitungan diatas, didapatkan berdasarkan kriteria f(n) yaitu:

1. Jika nilai n lebih besar atau sama dengan 0, maka f(n) bernilai 1.
2. Jika nilai n lebih kecil dari 0, maka f(n) bernilai -1.

berdasarkan hasil perhitungan n yang telah didapatkan pada pola pertama, n bernilai lebih besar dari 0 dan pada pola kedua n bernilai lebih kecil dari 0. Hal ini dapat dilihat dari tabel 3.

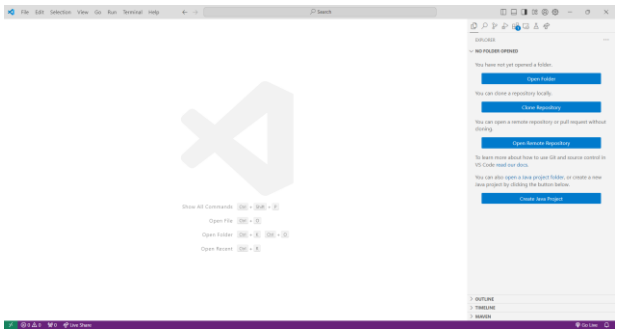
Tabel 3. Hasil Akhir

Pola	و	ب
Bias	0	0
Target	1	-1
n	24	-24
a = f(n)	-1	-1
Hasil	Pola Sesuai	Pola Sesuai

Pemeriksaan terhadap pola huruf waw (و) dan ba (ب) menegaskan bahwa keduanya menghasilkan keluaran yang persis selaras dengan target jaringan. Selarasnya keluaran target inilah yang memungkinkan jaringan metode Hebb mengenali dan memahami pola secara efektif. Performa algoritma metode Hebb memang sangat bergantung pada harmoni antara representasi input dan target. Dalam eksperimen ini, target ditetapkan waw (و) = 1 dan ba (ب) = -1, dan nilai-nilai tersebut terbukti identic atau sama dengan yang ditemukan selama proses identifikasi pola huruf.

B. Hasil Perhitungan Manual

Metode Hebb untuk pengenalan pola menggunakan konsep jaringan saraf tiruan, ini dapat diimplementasikan menggunakan bahasa pemrograman Python. Langkah awal yang dilakukan adalah melakukan proses instalasi Visual Studio Code pada komputer atau laptop anda.



Gambar 5. Aplikasi Visual Studio Code

Inisialisasi parameter-parameter utama yang diperlukan dalam proses pembelajaran menggunakan metode Hebb..

```
import numpy as np
bias = 0
xn = 25
jumlah_data = 2
```

Gambar 5. Inisialisasi awal

Dua pola huruf Hijaiyyah, *Waw* dan *Ba*, didefinisikan dalam bentuk array satu dimensi berisi nilai bipolar (1 dan -1) yang mempresentasikan pola dalam matriks 5x5.

```
pola = {
    "Waw": np.array([
        -1, -1,  1,  1, -1,
        -1,  1, -1, -1,  1,
        -1, -1,  1,  1,  1,
         1, -1, -1, -1,  1,
        -1,  1,  1,  1, -1
    ]),
    "Ba": np.array([
         1, -1, -1, -1,  1,
         1, -1, -1, -1,  1,
        -1,  1,  1,  1, -1,
        -1, -1, -1, -1, -1,
        -1, -1,  1, -1, -1
    ])
}
```

Gambar 6. Matrik bipolar huruf waw dan huruf ba

Setiap pola diberikan target output yang berupa nilai 1 untuk Waw dan -1 untuk Ba.

```
target = {
    "Waw": 1,
    "Ba": -1
}
```

Gambar 7. Target keluaran huruf waw dan huruf ba

Bobot awal diinisialisasikan sebagai array nol sepanjang jumlah fitur input (x_n = 25).

```
W_baru = np.zeros(xn)

x_list = [pola["Waw"], pola["Ba"]]
t_list = np.array([target["Waw"], target["Ba"]])
```

Gambar 8. Menentukan pola dan array huruf waw dan huruf ba

Menampilkan pola huruf dalam format visual 5x5. Simbol * merepresentasikan nilai -1, sedangkan simbol # merepresentasikan nilai 1.

```
def tampilan_pola(pola):
    for i in range(0, len(pola), 5):
        row = pola[i:i + 5]
        print("  ".join(["#" if val == 1 else "*" for val in row]))
    print()

print("Pola Huruf Hijaiyyah:\n")
print("Huruf Waw:")
tampilan_pola(pola["Waw"])
print("Huruf Ba:")
tampilan_pola(pola["Ba"])
```

Gambar 9. Menampilkan pola huruf waw dan huruf ba

Menghitung bobot baru(w) dan bias berdasarkan data masukan(x) dan target output(t).

```
for i in range(jumlah_data):
    W_baru += x_list[i] * t_list[i]
    bias += t_list[i]
```

Gambar 10. Menghitung bobot baru dan bias

Setelah bobot dan bias diperbaharui, nilai net dihitung untuk setiap data masukan, kemudian fungsi aktivasi f(net) digunakan untuk menentukan hasil pengenalan.

```
net_values = np.dot(x_list, W_baru) + bias
f_net = np.where(net_values >= 0, 1, -1)

print("\nHasil Pengenalan:")
for i, net in enumerate(net_values):
    huruf = "Waw" if f_net[i] == 1 else "Ba"
    print(f"Data ke-{i + 1} dikenali sebagai huruf: {huruf}")

print("\nTesting")
print("Bias:", bias)
for i, w in enumerate(W_baru):
    print(f"W{i + 1} = {int(w)}")
```

Gambar 10. Menghitung fungsi aktivasi

Kemudian memasukkan pola baru sebagai data uji, kemudian diklasifikasikan berdasarkan fungsi aktivasi f(net).

```
test_data = np.array([int(input(f"Masukkan uji data x{j + 1} = "))
                      for j in range(xn)])

net_test = np.dot(test_data, W_baru) + bias
fnet_test = 1 if net_test >= 0 else -1

huruf_test = "Waw" if fnet_test == 1 else "Ba"
print("\nNet (uji) =", int(net_test))
print("FNet (uji) =", fnet_test)
print(f"Huruf hasil pengujian: {huruf_test}")

aktivasi = "Sesuai" if fnet_test == target[huruf_test] else "Tidak Sesuai"
print("Hasil pengujian: fnet =", fnet_test, "=>", aktivasi)
```

Gambar 11. Menghitung data uji

Hasilnya adalah sebagai berikut:

```
Huruf Waw:
*  *  #  #  *
*  #  *  *  #
*  *  #  #  #
#  *  *  *  #
*  #  #  #  *

Huruf Ba:
#  *  *  *  #
#  *  *  *  #
*  #  #  #  *
*  *  *  *  *
*  *  #  *  *
```

```
Hasil Pengenalan:
Data ke-1 dikenali sebagai huruf: Waw
Data ke-2 dikenali sebagai huruf: Ba

 Testing
 Bias: 0
 W1 = -2
 W2 = 0
 W3 = 2
 W4 = 2
 W5 = -2
 W6 = -2
 W7 = 2
 W8 = 0
 W9 = 0
 W10 = 0
 W11 = 0
 W12 = -2
 W13 = 0
 W14 = 0
 W15 = 2
        W16 = 2
        W17 = 0
        W18 = 0
        W19 = 0
        W20 = 2
        W21 = 0
        W22 = 2
        W23 = 0
        W24 = 2
        W25 = 0
```

Kemudian dimasukkan nilai uji untuk setiap variabel input yang sudah ditentukan.

```
Masukkan uji data x2 = -1
Masukkan uji data x3 = -1
Masukkan uji data x4 = -1
Masukkan uji data x5 = 1
Masukkan uji data x6 = 1
Masukkan uji data x7 = -1
Masukkan uji data x8 = -1
Masukkan uji data x9 = -1
Masukkan uji data x10 = 1
Masukkan uji data x11 = -1
Masukkan uji data x12 = 1
Masukkan uji data x13 = 1
Masukkan uji data x14 = 1
Masukkan uji data x15 = -1
Masukkan uji data x16 = -1
Masukkan uji data x17 = -1
Masukkan uji data x18 = -1
Masukkan uji data x19 = -1
Masukkan uji data x20 = -1
Masukkan uji data x21 = -1
Masukkan uji data x22 = -1
Masukkan uji data x23 = 1
Masukkan uji data x24 = -1
Masukkan uji data x25 = -1

Net (uji) = -24
FNet (uji) = -1
Huruf hasil pengujian: Ba
Hasil pengujian: fnet = -1 => Sesuai
```

Dari hasil perhitungan menggunakan metode Hebbian menunjukkan bahwa metode Hebbian memiliki potensi sebagai pendekatan alternatif yang ringan dan efisien dalam pengenalan pola huruf hijaiyyah. Efektivitas metode ini dalam memproses data visual sederhana membuka peluang penerapannya pada perangkat dengan keterbatasan sumber daya, seperti aplikasi pembelajran mobile atau alat bantu digital untuk anak-anak dan penyandang disabilitas.

Implementasinya menggunakan bahasa pemrograman Python semakin memperkuat nilai aplikatif dari metode ini, mengingat Python merupakan bahasa yang sedang popular saat ini, fleksibel, dan didukung oleh ekosistem pustaka *open-source* yang luas. Efektivitas metode Hebbian dalam memproses data visual sederhana, disertai dengan kemudahan integrasi ke dalam platform berbasis Python seperti aplikasi pembelajaran mobile berbasis Kivy atau alat bantu visual interaktif berbasis Raspberry Pi, membuka peluang yang luas untuk penerapannya di lingkungan dengan keterbatasan sumber daya, khususnya untuk kebutuhan edukasi anak-anak dan penyandang disabilitas.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang dilakukan, dapat disimpulkan bahwa metode Hebbian menunjukkan kinerja yang potensial sebagai pendekatan alternatif yang ringan, efisien, dan sederhana dalam pengenalan pola huruf hijaiyyah. Metode ini mampu mengenali pola visual secara efektif dengan kompleksitas komputasi yang rendah, menjadikannya cocok diterapkan pada sistem keterbatasan sumber daya.

Dari hasil penelitian yang didapatkan, masih terbuka peluang pemanfaatan metode Hebbian dalam pengembangan aplikasi pembelajaran interaktif berbasis Python, seperti Kivy untuk perangkat mobile atau Raspberry Pi sebagai media bantu visual edukatif yang inklusif, terutama untuk anak-anak, pelajar pemula, dan penyandang disabilitas.

Untuk penelitian selanjutnya, disarankan untuk mengeksplorasi integrasi metode Hebbian dengan teknik *preprocessing* citra atau *feature extraction* yang lebih kompleks. Dapat juga dengan dilakukan pembandingan performanya dengan modela pembelajaran terbimbing lainnya seperti CNN atau SVM serta menguji penerapannya pada dataset huruf hijaiyyah dalam berbagai model citra, atau berbagai gaya tulisan tangan untuk meningkatkan generalisasi dan akurasi model.

DAFTAR PUSTAKA

[1] E. D. Manurung, B. Nadeak, and E. Ndruru, “Implementasi Algoritma Hebb Rule Pada Diagnosa Penyakit Kolik Abdomen Pada Orang Dewasa,” *JURIKOM (Jurnal Ris. Komputer)*, vol. 7, no. 2, p. 250, 2020, doi: 10.30865/jurikom.v7i2.2086.

[2] N. P. Sari, V. A. Akkili, and M. Muryeti, “Penerapan jaringan syaraf tiruan untuk menentukan elemen desain kemasan Numany rempeyek berbasis kansei engineering,” *Agrointek*, vol. 18, no. 3, pp. 742–752, 2024, doi: 10.21107/agrointek.v18i3.21790.

[3] A. Ridho, Suryadi, M. Ria Andini, and Kasmawati, “Hebb’S Rule Dalam Pengenalan Aksara Tulak-Tulak Mandailing,” *J. Tek. Inform. Kaputama*, vol. 5, no. 2, pp. 285–290, 2021.

[4] N. Ahmad *et al.*, *Intelligent System*. Yogyakarta: Penerbit Nuta Media, 2023.

[5] S. Silvilestari, “Pemanfaatan Algoritma Pembelajaran Pola Karakter Menggunakan Metode Hebb Rule,” *MEANS (Media Inf. Anal. dan Sist.*, vol. 8, no. 2, pp. 196–199, 2023, doi:

10.54367/means.v8i2.3061.

[6] B. Etikasari and Trismayanti Dwi Puspitasari, "Jurnal MNEMONIC MENGGUNAKAN ALGORITMA ADALINE Bety | Trismayanti," vol. 2, no. 1, pp. 12–16, 2019.

[7] A. W. Aranski and S. N. Rizki, "Pemanfaatan Algoritma Hebb Rule Mendiagnosis Kerusakan Elektroda Pada Proses Welding Frame Thermostat," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 6, p. 2205, 2022, doi: 10.30865/jurikom.v9i6.5335.

[8] Y. Yendrizal, "Jaringan Saraf Tiruan Pengenalan Pola Huruf Sistem Matriks dengan Algoritma Hebb Rule," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 5, p. 1466, 2022, doi: 10.30865/jurikom.v9i5.5015.

[9] Z. Arif, N. A. Santoso, and A. Muttaqin, "Penerapan Jaringan Syaraf Tiruan dan Certainty Factor untuk Peramalan Penjualan Sepeda Motor," *Remik Ris. dan E-Jurnal Manaj. Inform. Komput.*, vol. 7, no. 3, pp. 1523–1533, 2023.

[10] R. Meri and R. W. Perdana, "Jaringan Syaraf Tiruan Menggunakan Algoritma Hebb Rule Untuk Diagnosa Penyakit Kulit Manusia," *JOISIE J. Inf. Syst. Informatics Eng.*, vol. 6, no. Desember, pp. 78–87, 2022.

[11] D. Setiawan and A. Z. Falani, "Pemanfaatan Artificial Neural Network Dengan Metode Hebb Rule Untuk Pengenalan Bahasa Isyarat Indonesia Statis," *Spirit*, vol. 12, no. 1, pp. 9–15, 2020.

[12] W. Hamidah and L. S. Harahap, "Penerapan Jaringan Syaraf Tiruan untuk Pengenalan Pola Huruf Menggunakan Metode Bidirectional Associative Memory (BAM)," *NeptunusJurnal Ilmu Komput. dan Teknol. Inf.*, vol. 2, no. 4, pp. 168–178, 2024.

[13] A. T. Hsb, M. Tsaqofah, and L. S. Harahap, "Perancangan Model Jaringan Syaraf Tiruan untuk Memprediksi Penyakit Demam Berdarah Menggunakan Algoritma Hebb Rule," *MarsJurnal Tek. Mesin, Ind. Elektro dan Ilmu Komput.*, vol. 2, no. 6, pp. 07–22, 2024.

[14] S. A. Pasaribu and S. P. Sipayung, "Penerapan Aturan Hebb dalam Identifikasi dan Pengobatan Kolik Abdomen pada Pasien Dewasa : Pendekatan Algoritma yang Efektif," in *SNISTIK: Seminar Nasional Inovasi Sains Teknologi Informasi Komputer*, 2024, pp. 465–472.

[15] S. N. Rizki and Y. Mardiansyah, "The Vocal Patterns Recognition In Artificial Neural Network By Using The Hebb Rule Algorithm," *Int. J. Inf. Syst. Technol.*, vol. 5, no. 158, pp. 767–774, 2022.

[16] J. Sinurat, "Jaringan Saraf Tiruan Diagnosa Penyakit Kanker Paru-Paru Menggunakan Metode Hebb Rule," *Bull. Inf. Technol.*, vol. 2, no. 1, pp. 20–27, 2021.

[17] N. Kristianti and W. Widiatry, "Penggunaan Algoritma Hebb Dalam Pola Pengenalan Huruf," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 18, no. 1, pp. 52–60, 2024, doi: 10.47111/jti.v18i1.12561.

[18] M. Nasir, Amri, and I. Maulina, "Pengenalan Aksara Isyarat Menggunakan Metode Hebb Rule," *J. Infomedia*, vol. 4, no. 1, pp. 28–32, 2019.