

Pengenalan Huruf *Braille* menggunakan Jaringan Syaraf Tiruan Algoritma *Perceptron*

Lany Harianto¹, Kristien Margi Suryaningrum²

^{1,2}Program Studi Teknik Informatika, Fakultas Teknologi dan Desain, Universitas Bunda Mulia
Jalan Lodan Raya No. 2, Ancol, Jakarta Utara

¹*lany_harianto@yahoo.com*

²*ksuryaningrum@bundamulia.ac.id*

ABSTRACT

Braille is a letter that is made especially for the blind people to be able perform reading activities such as normal people. As technology advances, people with visual impairment can also make writing using *braille*. However, their written works need to be translated into Latin letters in order normal people could read and understand it. In this research, *braille* recognition application was made to help translating *braille* letters into Latin letters using pixel population matrix for feature extraction and *Perceptron* algorithm for letters learning. From the experiments, *braille* recognition application has success percentage 100%.

Keywords: Character Recognition, *Braille*, Pixel Population Matrix, Artificial Neural Network, *Perceptron*.

INTISARI

Braille merupakan huruf yang dibuat khusus bagi penyandang tunanetra agar dapat melakukan kegiatan membaca seperti orang bukan penyandang tunanetra lain. Seiring kemajuan teknologi, para penyandang tunanetra juga dapat membuat karya tulis menggunakan huruf *braille*. Namun hasil karya tulis mereka perlu diterjemahkan menjadi huruf latin agar dapat dibaca dan dimengerti masyarakat yang bukan penyandang tunanetra. Pada penelitian ini dibuat aplikasi pengenalan huruf *braille* untuk membantu penerjemahan huruf *braille* menjadi huruf latin menggunakan ekstraksi fitur matriks populasi piksel dan pembelajaran huruf menggunakan algoritma *Perceptron*. Dari percobaan yang dilakukan, aplikasi pengenalan huruf *braille* ini menghasilkan persentase keberhasilan sebesar 100%.

Kata Kunci: Pengenalan Huruf, *Braille*, Matriks Populasi Piksel, Jaringan Syaraf Tiruan, *Perceptron*.

PENDAHULUAN

Braille merupakan huruf yang dibuat khusus bagi para penyandang tunanetra agar dapat melakukan kegiatan membaca seperti orang bukan penyandang tunanetra lain. Seiring kemajuan teknologi, para penyandang tunanetra ini juga dapat menyalurkan imajinasinya menjadi suatu karya tulis menggunakan huruf *braille*. Namun hasil karya tulis ini, perlu diterjemahkan menjadi huruf latin agar dapat dibaca dan dimengerti masyarakat yang bukan penyandang tunanetra.

Oleh karena itu penulis tertarik untuk membuat aplikasi pengenalan huruf *braille* menggunakan algoritma *Perceptron* yang diharapkan dapat membantu para pengembang *software braille* dapat membantu pengguna aplikasi dalam mempelajari huruf *braille*.

Berdasarkan latar belakang permasalahan yang ada, maka rumusan masalah yang didapat yaitu bagaimana cara mengimplementasi algoritma *Perceptron* untuk pengenalan huruf *braille* dan bagaimana tingkat keberhasilan algoritma *Perceptron* dalam melakukan pengenalan huruf *braille*.

Tujuan dari penelitian ini ialah untuk mengetahui cara mengimplementasi algoritma *Perceptron* untuk pengenalan huruf *braille* dan untuk mengetahui tingkat keberhasilan algoritma *Perceptron* dalam melakukan pengenalan huruf *braille*.

Braille

Braille adalah sebuah bentuk komunikasi tertulis yang menggunakan titik-titik timbul di atas kertas atau logam untuk memudahkan orang tunanetra membaca.

Bentuk komunikasi *braille* yang telah diciptakan oleh Louis Braille ini terdiri atas enam posisi titik yang disusun dalam dua kolom. Setiap kolom berisi tiga posisi titik sehingga membentuk persegi panjang. Satu titik atau lebih kemudian ditimbulkan pada salah satu atau beberapa dari keenam posisi titik tersebut untuk mewakili huruf latin, tanda baca, atau bilangan tertentu. Louis Braille menemukan 63 kombinasi susunan titik timbul yang mungkin (Doserinal, 2012).

a	b	c	d	e	f	g	h	i	j

Gambar 1. Huruf Braille

Citra Digital

Citra Digital berarti pengolahan citra/gambar yang dilakukan secara digital menggunakan komputer. Agar citra dapat diolah menggunakan komputer digital, citra tersebut harus direpresentasikan terlebih dahulu ke dalam bentuk numerik dengan nilai-nilai diskrit. Sebuah citra dapat diwakili oleh sebuah matriks dua dimensi $f(x,y)$ yang terdiri dari M kolom dan N baris. Perpotongan dari kolom dan baris ini disebut dengan *pixel* (*Picture Element*) atau elemen terkecil dalam sebuah citra (Kusumanto & Alan, 2011).

Citra RGB

Citra RGB atau yang biasa disebut dengan citra warna memiliki warna tertentu pada masing-masing *pixel*-nya, warna tersebut antara lain merah (*Red*), hijau (*Green*) dan biru (*Blue*). Masing-masing warna tersebut memiliki *range* 0-255. Citra warna ini terdiri dari tiga matriks yang mewakili nilai-nilai merah, hijau, dan biru untuk setiap *pixel*-nya (Kusumanto & Alan, 2011).

Citra Grayscale

Citra *greyscale* atau biasa dikenal juga sebagai citra *black and white* memiliki warna gradasi mulai dari putih sampai hitam pada setiap *pixel*-nya. Setiap *pixel* tersebut diwakili oleh 8 bit. Rentang warna pada citra *greyscale* ini sangat cocok digunakan untuk pengolahan *file* gambar. Citra *greyscale* ini merupakan hasil rata-rata dari citra warna, yaitu $(\text{nilai pixel Red} + \text{nilai pixel Green} + \text{nilai pixel Blue})/3$ (Kusumanto & Alan, 2011).

Citra Biner

Citra biner merupakan citra yang setiap *pixel*-nya hanya terdiri dari warna hitam atau putih. Setiap *pixel*-nya hanya diperlukan 1 bit (0 dan 1) atau jika dalam 8 bit (0 dan 255), hal ini menyebabkan citra biner sangat efisien dalam hal penyimpanan (Sugeng, Rita, & Hilman, 2016). Citra biner didapat dari hasil proses pengambangan (*thresholding*). Jika nilai *pixel* lebih besar daripada nilai *thresholding*, maka nilai *pixel* tersebut akan diubah menjadi 1, dan jika nilai *pixel* lebih kecil daripada nilai

thresholding maka nilai *pixel* akan diubah menjadi 0.

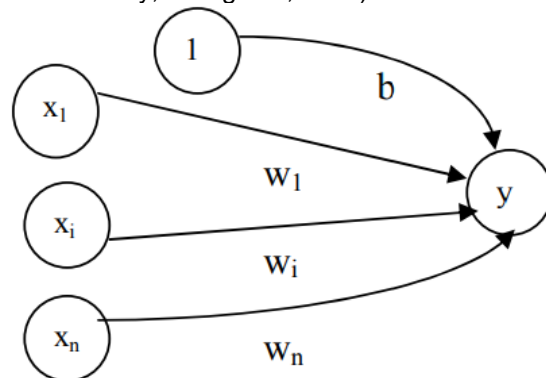
Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan merupakan paradigma pengolahan informasi yang terinspirasi dari sistem syaraf secara biologis seperti proses informasi yang terdapat pada otak manusia. Cara kerja jaringan syaraf tiruan ini sama seperti cara kerja manusia, yaitu dengan bekerja melalui contoh, maka diperlukannya pengenalan pola atau klasifikasi data melalui proses pembelajaran dalam aplikasi jaringan syaraf tiruan ini. Hal ini melibatkan penyesuaian terhadap koneksi yang ada antara *neuron* tersebut.

Aplikasi jaringan syaraf tiruan sudah sangat beragam baik dalam bidang teknologi maupun bidang-bidang lainnya. Salah satu aplikasi jaringan syaraf tiruan pada bidang teknologi yaitu dalam bidang pengenalan pola yang berkaitan dengan bidang visi komputer / *Computer Vision* yang mencoba meniru kapabilitas mata manusia dan sistem otak untuk membentuk dan menginterpretasikan citra (Simbolon, 2013).

Perceptron

Salah satu macam dari jaringan syaraf tiruan yaitu *Perceptron*. Model jaringan *Perceptron* ini ditemukan oleh Rosenblatt pada tahun 1962 dan Minsky-Papert pada tahun 1969, model jaringan ini memiliki aplikasi dan pelatihan yang sangat baik saat itu. *Perceptron* dapat dikatakan sebagai salah satu Teknik jaringan syaraf tiruan yang sederhana. Teknik ini hanya memiliki sebuah lapisan *input* dan sebuah unit *output* dan terdapat bias (b) yaitu unit yang aktifasinya selalu 1 dan berperilaku sebagai layaknya bobot (w) (Hafizah, Sulindawaty, & Tugiono, 2013).



Gambar 2. Arsitektur Jaringan *Perceptron*

Keterangan:

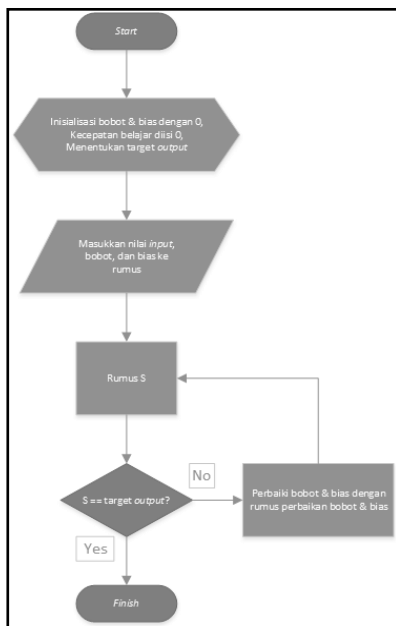
$X_1...X_i...X_n$ = *neuron input*
 y = *neuron output*
 b = bias

W_1, W_i, W_n = bobot

Arsitektur jaringan *Perceptron* ini mirip dengan arsitektur jaringan *Hebb* namun *Perceptron* lebih baik jika dibandingkan dengan *Hebb*, berikut kelebihanannya (Yudhistiro, 2017):

- Tidak setiap pola masukan (iterasi) mengalami perubahan bobot, perubahan bobot hanya dilakukan saat pola dimasukkan menghasilkan keluaran jaringan yang jika dibandingkan dengan target yang sesungguhnya terdapat perbedaan;
- Perubahan bobot tidak hanya ditentukan oleh perkalian antara target dengan masukan, namun juga melibatkan laju pembelajaran (α) yang besarnya bisa diatur, sehingga bisa mempercepat proses pembelajaran jaringan;
- Pelatihan (iterasi) dilakukan terus menerus untuk semua kemungkinan pola yang ada hingga jaringan dapat mengerti polanya ($y=t$);
- Sirkulasi pelatihan yang melibatkan semua pola masukan disebut *epoch*.

Algoritma *Perceptron* ini dapat digunakan untuk mengenali pola karakter yang menyerupai huruf alfabet atau membedakan karakter menyerupai huruf atau pola tertentu atau tidak.



Gambar 3. *Flowchart* pembelajaran algoritma *Perceptron*

Pada gambar 3 digambarkan *flowchart* mengenai cara pembelajaran algoritma *Perceptron*. Pertama, inisialisasi bobot dan bias dengan 0, serta kecepatan belajar dengan 0.1, tentukan pula target *output*, setelah itu,

masukkan nilai *input*, bobot, serta bias ke dalam rumus 1 untuk menghitung respons unit *output*.

$$y_{in} = b + \sum_t x_i w_i \quad (\text{Rumus 1})$$

Jika hasil perhitungan rumus sesuai dengan target *output*, maka proses pembelajaran telah berhasil. Namun jika hasil tidak sesuai, maka perbaiki bobot dan bias dengan rumus perbaikan bobot (rumus 2) dan bias (rumus 3):

$$w_i(\text{baru}) = w_i(\text{lama}) + \alpha * t * x_i \quad (\text{Rumus 2})$$

$$b(\text{baru}) = b(\text{lama}) + \alpha * t \quad (\text{Rumus 3})$$

Keterangan:

y = *output*

b = bias

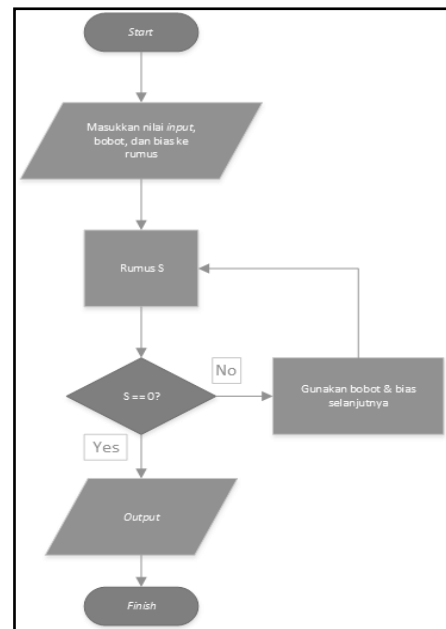
t = target yang nilainya +1 atau -1

x_i = *input* ke- i

w_i = bobot ke- i

α = kecepatan belajar yaitu $0 \leq \alpha \leq 1$

Lakukan hingga hasil rumus sama dengan target *output*.



Gambar 4. *Flowchart* pengenalan algoritma *Perceptron*

Pada gambar 4 digambarkan *flowchart* untuk pengenalan algoritma *Perceptron*. Pertama masukkan nilai *input*, bobot, serta bias ke dalam rumus 1, jika hasil rumus sama dengan 0 maka *output* akan keluar, namun jika hasil rumus tidak sama dengan 0, maka

gunakan bobot dan bias selanjutnya untuk perhitungan rumus 1.

HASIL DAN PEMBAHASAN

Langkah pertama yang perlu dilakukan untuk melakukan proses pembelajaran huruf *braille* menggunakan algoritma *Perceptron* yaitu dengan melakukan binerisasi pada gambar yang di-*input*.

Setelah melakukan proses binerisasi pada gambar yang diinput, langkah selanjutnya ialah melakukan proses segmentasi pada gambar hasil proses binerisasi.

Berikutnya lakukan proses ekstraksi fitur pada gambar hasil segmentasi untuk mendapatkan nilai fitur dari setiap *pixel*-nya.

Setelah penghitungan ekstraksi fitur selesai dilakukan, selanjutnya proses pembelajaran menggunakan algoritma *Perceptron* dilakukan. Dan ditunjukkan pada gambar 5.

```

for i := 0 to 16 do
begin
  w[i] := 0
endfor
bias := 0
c := 0.1
e := 1
max := 0
read huruf
while e>0
  e := 0
  if huruf = alphabet then
    t := 1
  else
    t := 0
  endif
  for i := 0 to 16 do
  begin
    s := s + (f[i]*w[i])
  endfor
  s := s + bias
  if s > 0 then
    s := 1
  else
    s := 0
  endif
  x := t - s
  while x != 0
    s := 0
    e += 1
    nilaiBobot := ""
    for i := 0 to 16 do
    begin
      w[i] := w[i] + (c*x*f[i])
    endfor
    bias := bias + (c*x)
    for l := 0 to 16 do

```

```

begin
  s := s + (f[i]*w[i])
endfor
s := s + bias
if s > 0 then
  s := 1
else
  s := 0
endif
x := t - s
endwhile
endwhile
max := max + 1
if max = 25000 then
  e := 0
endif

```

Gambar 5. Proses pembelajaran menggunakan algoritma *Perceptron*
Tabel 1 merupakan tabel hasil dari proses pembelajaran huruf *braille* menggunakan algoritma *Perceptron*.

Tabel 1. Bobot hasil pembelajaran huruf

huruf	Braille			
	bobot 1	bobot 2	...	bobot 16
a	-0.49	-0.101	...	-0.177
b	-0.328	0.166	...	0.492
c	-0.114	-0.217	...	0.042
d	0.173	-0.085	...	-0.031
e	0.016	-0.003	...	0.035
F	0.16	0.249	...	-0.959
g	6.727	-11.72	...	0.474
h	0.073	-0.67	...	-0.302
i	-0.273	-0.141	...	-0.384
j	-0.253	-0.288	...	0.243
k	-0.164	0.005	...	-0.454
l	-0.074	0.341	...	-0.205
m	0.131	-0.082	...	-0.44
n	0.235	-0.341	...	-0.762
o	-0.044	-0.013	...	-0.365
p	0.446	0.174	...	-0.546
q	0.381	0.07	...	-1.101
r	0.168	0.003	...	-0.45
s	-0.184	-0.094	...	-0.092
t	-0.25	-0.217	...	-0.526
u	-0.032	0.037	...	0.032
v	0	-0.045	...	-0.03
w	-0.312	-0.3	...	0.073
x	0.479	-0.908	...	0.601

y	0.106	-0.66	...	0.281
z	-0.3	-0.174	...	0.334

Proses pengenalan huruf *braille* baru dapat dilakukan jika proses pembelajaran selesai dilakukan. Gambar 6 ialah proses pengenalan huruf *braille*:

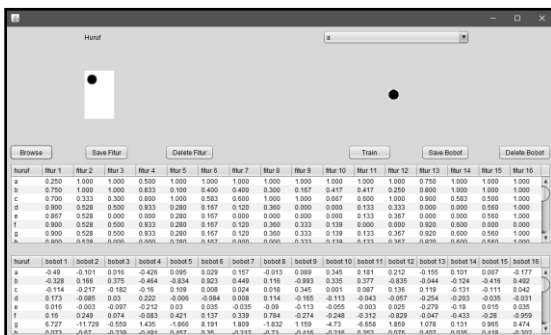
```

for i := 0 to 16 do
  begin
    soutput := soutput + (fitur[i] * b[i])
  endfor
soutput := soutput + bias
if soutput > 0 then
  if soutput > temp then
    temp := soutput
    tempstr := alphabet
  endif
endif
endif

```

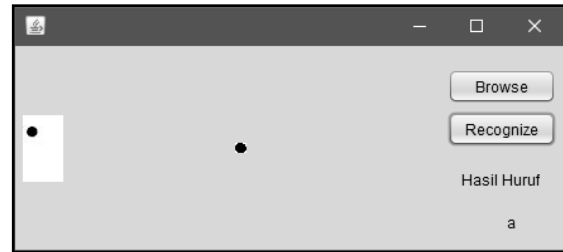
Gambar 6. Proses pengenalan menggunakan algoritma *Perceptron*

Gambar 7 merupakan tampilan antarmuka dari pembelajaran huruf *braille* menggunakan algoritma *Perceptron*. Pada tampilan ini, setiap gambar huruf yang di-*input* perlu dihitung terlebih dahulu nilai fiturnya. Setelah semua gambar huruf dihitung nilai fiturnya, maka tahap pembelajaran dapat dilakukan pada setiap huruf yang ada.



Gambar 7. Tampilan antarmuka pembelajaran huruf *braille* menggunakan algoritma *Perceptron*

Gambar 8 merupakan tampilan antarmuka dari pengenalan huruf *braille* menggunakan algoritma *Perceptron*. Pada tampilan ini, pengguna cukup memasukkan gambar huruf yang ingin dikenali, kemudian tekan tombol "Recognize", maka sistem akan bekerja untuk mengenali huruf tersebut dan menampilkan hasilnya di bagian kanan bawah dari form tampilan.



Gambar 8. Tampilan antarmuka pengenalan huruf *braille* menggunakan algoritma *Perceptron*

Untuk mengetahui keberhasilan metode pada sistem, maka dilakukan pengujian. Tabel 2 merupakan tabel hasil pengujian yang dilakukan pada sampel. Pengujian pengenalan pada aplikasi pengenalan huruf *braille* ini menggunakan 4 model gambar *input*-an yang berbeda sebagai sampel dengan jumlah total gambar sebanyak 104 buah. Pengujian pengenalan yang dilakukan terhadap data sampel memperoleh persentase keberhasilan sebesar 100%.

Tabel 2. Hasil Pengujian sampel

					Total
	26	26	26	26	104
	26	26	26	26	104
	0	0	0	0	0
	100%	100%	100%	100%	100%

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan ini dapat ditarik kesimpulan bahwa aplikasi pengenalan huruf *braille* dengan menggunakan algoritma *Perceptron* mampu mengenali huruf yang di-*input* berupa citra digital dengan persentase keberhasilan sebesar 100%.

Saran yang dapat dilakukan untuk penelitian selanjutnya yaitu penelitian ini dapat dikembangkan sehingga bisa mengenali tidak hanya 1 huruf *braille* saja dalam 1 file citra digital, melainkan dapat mengenali kalimat *braille* yang terdiri dari banyak huruf *braille* dalam 1 file citra digital. Penelitian ini juga dapat dicoba menggunakan algoritma pembelajaran lainnya.

DAFTAR PUSTAKA

Chen, Tao., Takagi, Miko., "Image Binarization By Back Propagation Algorithm", International Society for Photogrammetry

- and Remote Sensing XXIX, 1992, August 2-14, 1992, Washington, D.C., USA. p:345-349
- Dosrinal. 2012. *Yakin Bisa dari Nothing jadi Something*. Jakarta: Raih Asa Sukses.
- Hafizah, Sulindawaty, dan Tugiono. 2015. *Penerapan Jaringan Syaraf Tiruan dengan Algoritma Perceptron untuk Mendeteksi Karakteristik Sidik Jari*. Jurnal Ilmiah Sains dan Komputer, XIV(2), pp.83-92.
- Khardon, Roni., Wachman, Gabriel., "Noise Tolerant Variants of the Perceptron Algorithm", Journal of Machine Learning Research 8 (2007) 227-248 Submitted 11/05; Revised 10/06; Published 2/07
- Kusumanto dan Alan N.T. 2011. *Pengolahan Citra Digital untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB*. Semantik, I(1), pp.1-7.
- Pugh, Justin K., Soltoggio, Andrea,, and Stanley, Kenneth O., "Real-time Hebbian Learning from Autoencoder Features for Control Tasks", Proceeding of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems (ALIFE 14).Cambridge, MA: MIT Press, 2014.
- Sathasivam, Saratha., "Learning Rules Comparison in Neuro-Symbolic Integration", International Journal of Applied Physics and Mathematics, Vol. 1, No. 2, September 2011, Abu Dhabi University, UAE (ISSN: 2010-362X) page 129-132
- Simbolon, R. 2013. *Perangkat Lunak untuk Identifikasi Pengenalan Huruf Braille dengan Algoritma Perceptron*. Jurnal Pelita Informatika Budi Darma, IV(2), pp.155-159.
- Sugeng, O.P.K., Rita M., dan Hilman F. 2016. *Sistem Deteksi Glaukoma dengan Pengukuran Area Optik Disk pada Citra Fundus*. E-Proceeding of Engineering, III(3), pp.4823-4830.
- Yudhistiro, Kuku. 2017. *Pemanfaatan Neural Network Perceptron pada Pengenalan Pola Karakter*. Journal of Information Technology and Computer Science, II(2), pp.83-86.