

# SISTEM PENGENALAN KARAKTER DENGAN JARINGAN SYARAF TIRUAN ALGORITMA PERCEPTRON

Anindita Septiarini

Program Studi Ilmu Komputer FMIPA, Universitas Mulawarman  
Jalan Barong Tongkok No. 4 Samarinda  
Email: deeta82@yahoo.com

## ABSTRAK

Jaringan syaraf tiruan (JST) adalah model pembelajaran yang dapat digunakan untuk pengenalan objek. Perceptron merupakan salah satu algoritma pembelajaran dari JST yang dapat digunakan untuk mengenali objek berupa karakter. JST terdiri dari dua proses utama, yaitu proses *training* (pembelajaran) dan proses *testing* (pengujian). Pada sistem ini proses *training* membutuhkan data masukan berupa citra karakter, dimana setiap karakter memiliki ciri yang berbeda. Data ciri setiap karakter berupa matriks berukuran  $10 \times 10$  dan akan menjadi data masukan pada struktur JST. Keluaran sistem ini berupa data teks hasil pengenalan citra karakter. Sebagai uji coba sistem ini akan menghasilkan keluaran sebanyak sepuluh kategori karakter, yaitu karakter A – J.

*Keywords* : Neural Network, Perceptron, Pengenalan karakter

## PENDAHULUAN

Pada saat ini pengenalan suatu karakter yang terdapat pada citra dapat dilakukan oleh komputer. Metode yang digunakan untuk sistem pengenalan karakter pada penelitian ini menggunakan metode jaringan syaraf tiruan (JST) dengan algoritma perceptron.

JST merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. JST terdiri dari beberapa algoritma diantaranya adalah perceptron.

Perceptron merupakan algoritma pembelajaran dengan pengawasan dalam sistem JST, sehingga jaringan yang dihasilkan harus mempunyai parameter yang dapat diatur dengan cara mengubah melalui aturan pembelajaran dengan pengawasan. JST terdiri dari sejumlah neuron dan sejumlah masukan.

Data masukan pada JST merupakan ciri-ciri dari setiap karakter. Ciri-ciri setiap karakter berupa matrik dengan ukuran  $10 \times 10$ , sehingga jumlah neuron masukan sebanyak 100 neuron. Hasil dari sistem ini adalah berupa karakter huruf A sampai dengan J.

## METODE PENELITIAN

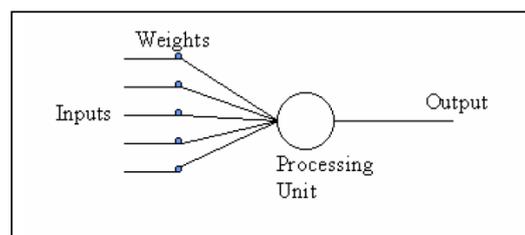
Sistem ini terdiri dari dua proses utama, yaitu proses pembelajaran (*training*) dan proses

pengujian (*testing*) dimana algoritma JST yang digunakan adalah perceptron.

## Jaringan Syaraf Tiruan

Jaringan syaraf tiruan adalah suatu prosesor yang melakukan pendistribusian secara besar-besaran, yang memiliki kecenderungan alami untuk menyimpan suatu pengenalan yang pernah dialaminya, dengan kata lain JST ini memiliki kemampuan untuk dapat melakukan pembelajaran dan pendeteksian terhadap sesuatu objek.

Secara mendasar, sistem pembelajaran merupakan proses penambahan pengetahuan pada JST yang sifatnya kontinuitas sehingga pada saat digunakan pengetahuan tersebut akan dieksploitasikan secara maksimal dalam mengenali suatu objek. Neuron adalah bagian dasar dari pemrosesan suatu JST. Bentuk dasar dari suatu neuron dapat dilihat pada gambar 1.



Gambar 1. Bentuk dasar neuron

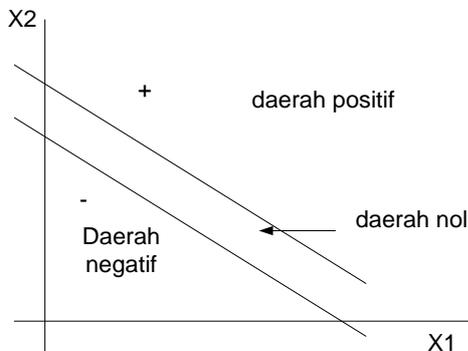
Keterangan :

1. Input merupakan masukan yang digunakan baik saat pembelajaran maupun dalam mengenali suatu objek.
2. Weight, beban yang selalu berubah setiap kali diberikan input sebagai proses pembelajaran.
3. Processing Unit merupakan tempat berlangsungnya proses pengenalan suatu objek berdasarkan pembebanan yang diberikan.
4. Output, keluaran dari hasil pengenalan suatu objek.

**Perceptron**

Perceptron merupakan salah satu bentuk JST yang sederhana. Perceptron biasanya digunakan untuk mengklasifikasikan suatu pola tertentu yang sering dikenal dengan pemisahan secara linier. Perceptron memiliki kecenderungan yang sama dengan jenis JST lainnya, namun setiap jenis memiliki karakteristik masing-masing.

Perceptron pada JST dengan satu lapisan memiliki bobot yang bisa diatur dan suatu nilai ambang (threshold). Algoritma yang digunakan oleh aturan ini akan mengatur parameter-parameter bebasnya melalui proses pembelajaran. Nilai threshold ( $\Theta$ ) pada fungsi aktivasi adalah non negatif. Fungsi aktivasi ini dibuat sedemikian rupa sehingga terjadi pembatasan antara daerah positif dan daerah negatif dapat dilihat pada gambar 2.



**Gambar 2.** Pembatasan linear dengan perceptron

Garis pemisah antara daerah positif dan daerah nol memiliki pertidaksamaan :

$$w_1x_1 + w_2x_2 + b > \Theta$$

Sedangkan garis pemisah antara daerah negatif dengan daerah nol memiliki pertidaksamaan :

$$w_1x_1 + w_2x_2 + b < -\Theta$$

Misalkan kita gunakan pasangan vektor input  $s$  dan vektor output sebagai pasangan vektor yang akan dilatih.

**Algoritma Perceptron**

Perceptron memberikan pengaruh yang kuat di awal pengembangan JST. Proses belajar pada perceptron menggunakan pengaturan bobot secara iterative (berulang).

Salah satu model sederhana dari perceptron menggunakan aktivasi biner untuk unit sensor dan unit associator, serta aktivasi +1, 0 atau  $\pm 1$  untuk unit response. Unit sensor dihubungkan ke unit associator oleh jalur dengan bobot yang tetap dan bernilai +1, 0, -1 yang ditentukan secara random.

Fungsi aktivasi dari tiap unit associator adalah fungsi undak biner dengan nilai ambang yang dapat diubah-ubah. Sehingga signal yang dikirim dari unit associator ke unit output adalah biner (1 atau 1). Output dari perceptron adalah  $y = f(y\_in)$ , dimana fungsi aktivasinya adalah:

$$f(y\_in) = \begin{cases} 1 & \text{jika } y\_in > \theta \\ 0 & \text{jika } -\theta \leq y\_in \leq \theta \\ -1 & \text{jika } y\_in < -\theta \end{cases}$$

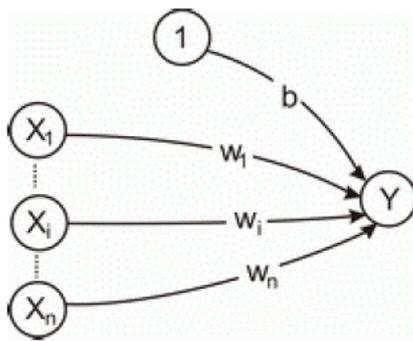
Bobot dari unit associator ke unit response (output) di atur melalui aturan pelatihan perceptron. Untuk setiap input pelatihan, jaringan akan menghitung respon dari unit output. Kemudian jaringan akan menentukan apakah error terjadi pada pola input tersebut dengan membandingkan output hasil perhitungan dengan nilai target.

Jaringan tidak dapat membedakan error yang terjadi (apakah antara 0 dan  $\pm 1$  atau +1 dan  $\pm 1$  dll), tetapi tanda (+/-) dari error akan mengarahkan perubahan bobot untuk mendekati nilai target. Namun, hanya bobot dengan signal tidak sama dengan nol saja yang diatur oleh perceptron. Jika terjadi error pada pola pelatihan tertentu, maka bobot harus diubah menurut rumusan:

$$W_i(\text{baru}) = W_i(\text{lama}) + \alpha t x_i$$

dimana  $t$  = nilai target (1 atau  $\pm 1$ , bipolar); dan  $a$  adalah laju pelatihan. Pelatihan terus dilakukan hingga error tidak terjadi. Berikut arsitektur dari sebuah jaringan perceptron sederhana untuk klasifikasi pola (pattern classification). Output dari unit associator adalah vektor biner dan vektor ini selanjutnya digunakan sebagai signal input pada unit output.

Karena hanya bobot antara unit associator dan unit output yang bisa diatur maka kita dapat membatasi pengamatan kita pada layer tunggal bagian dari jaringan, yang ditunjukkan oleh gambar 3.



Gambar 3. Proses antar unit pada perceptron

Algoritma ini dapat digunakan baik untuk input bipolar maupun biner, dengan bipolar target, nilai ambang ( $\Theta$ ) tetap, dan bias yang bisa diatur. Nilai ambang ( $\Theta$ ) disini tidak memainkan peranan seperti dalam fungsi undak. Sehingga keduanya (bias dan nilai ambang) di perlukan. Algoritma ini tidak terlalu sensitif terhadap nilai inisialisasi bobot (nilai awal bobot) maupun terhadap laju pelatihan.

**Algoritma Perceptron :**

- **Langkah 0.** Inisialisasi Bobot dan Bias  
Untuk lebih mudah set bias dan bobot ke nol. Lalu tentukan nilai laju pelatihan.
- **Langkah 1** Selama kondisi henti tidak terpenuhi jalankan langkah 2-6
- **Langkah 2** Untuk setiap pasangan pola s:t lakukan langkah 3-5
- **Langkah 3** Set nilai aktivasi untuk unit input  
 $X_i = S_i$
- **Langkah 4** hitung respon dari unit output  
$$y\_in = b + \sum_i x_i w_i$$
  
$$y = \begin{cases} 1 & \text{jika } y\_in > \theta \\ 0 & \text{jika } -\theta \leq y\_in \leq \theta \\ -1 & \text{jika } y\_in < -\theta \end{cases}$$
- **Langkah 5** Update nilai bobot dan bias jika error terjadi pada pola

Jika  $y \neq t$ ,

$$W_i(\text{baru}) = W_i(\text{lama}) + \alpha t x_i$$

$$b(\text{baru}) = b(\text{lama}) + \alpha t$$

atau

$$W_i(\text{baru}) = W_i(\text{lama})$$

$$b(\text{baru}) = b(\text{lama})$$

➤ **Langkah 6** Test kondisi henti

Jika tidak ada perubahan bobot di langkah 2, berhenti; jika tidak, lanjut. Dari nilai bobot hasil pelatihan dapat diperoleh garis pemisah (separating line)

$$w_1 x_1 + w_2 x_2 + b > \theta \quad \text{untuk respon positif, dan}$$

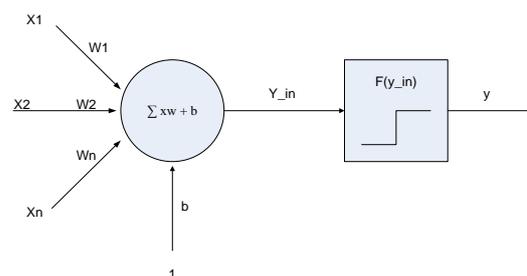
$$w_1 x_1 + w_2 x_2 + b < -\theta \quad \text{untuk respon negatif.}$$

Nb: jika input pelatihan hanya 2 pasang

Algoritma tersebut bisa digunakan baik untuk input biner maupun bipolar, dengan  $\Theta$  tertentu, dan bias yang dapat diatur. Pada algoritma tersebut bobot - bobot yang diperbaiki hanyalah bobot - bobot yang berhubungan dengan input yang aktif ( $x_i \neq 0$ ) dan bobot - bobot yang tidak menghasilkan nilai  $y$  yang benar.

**HASIL DAN PEMBAHASAN**

Arsitektur jaringan yang digunakan pada sistem pengenalan karakter ini dapat dilihat pada gambar 4.



Gambar 4. Arsitektur jaringan yang digunakan

Set nilai awal adalah sebagai berikut :

Bobot awal	: $w = 0$
Bobot bias	: $b = 0$
Learning rate ( alfa )	: 1
Treshold	: 0,3

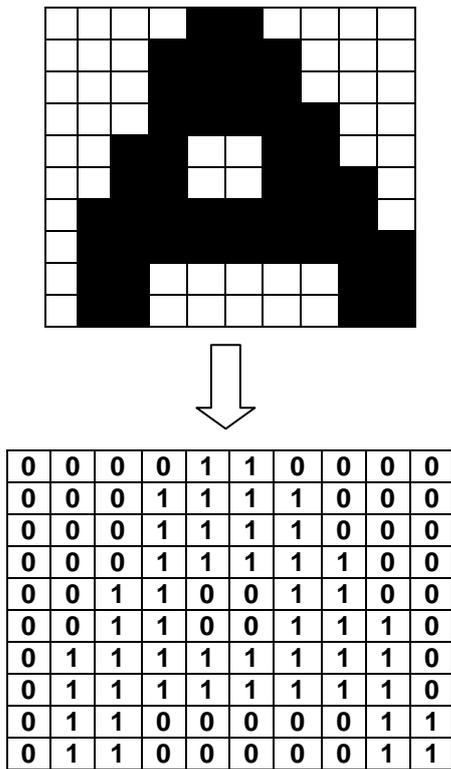
**Proses Input Data Training**

Pada proses input data training data yang menjadi inputan berupa gambar karakter huruf.

Karakter huruf yang diinputkan tersebut akan ditampung pada area buffer yang terdiri dari area yang bernilai 0 dan area yang bernilai 1, untuk lebih jelas dapat dilihat pada gambar 5.

Area tersebut yang digunakan sebagai input untuk menentukan bobot sesuai dengan algoritma yang digunakan. Gambar karakter huruf yang diinputkan oleh user akan disimpan dalam bentuk gambar berekstensi bmp.

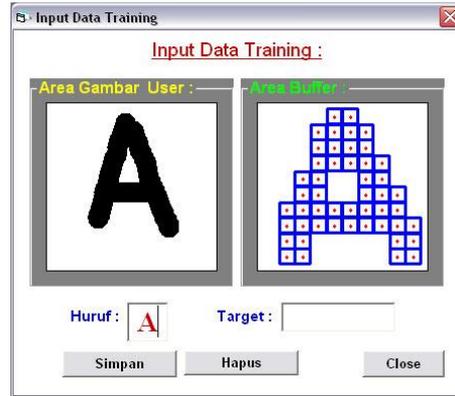
Target akan diisi secara otomatis oleh sistem. Sebagai contoh, misalkan ketika user menggambar huruf A dan menginput "A" pada huruf. Pada saat user menekan tombol Simpan maka pada textbox target otomatis akan mengeluarkan "-1-1-1-1-1". Untuk huruf B target yang akan dikeluarkan adalah "-1-1-1-11". Proses input data training untuk lebih jelasnya dapat dilihat pada gambar 6.



Gambar 5. Huruf A sebagai matrik 10 x 10

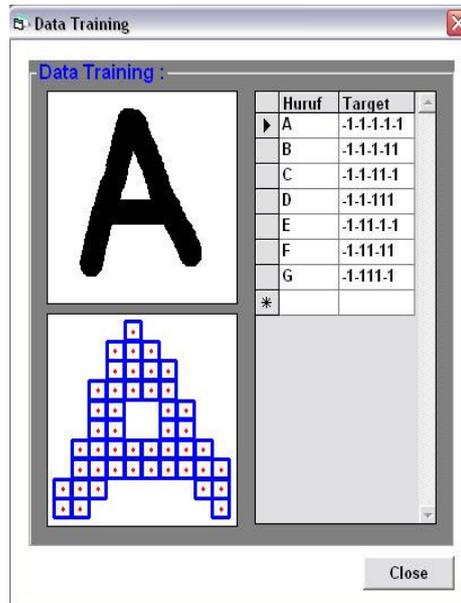
Komponen vektor bernilai 1, menandakan bahwa kotak yang diwakilinya berwarna hitam, sedangkan vektor bernilai 0, menandakan bahwa kotak yang diwakilinya berwarna putih. Sehingga vektor (Input Vektor) yang bersesuaian yang terbentuk adalah :

```
00001100000001111000000111100000011111000
0110011000011001110011111110111111110011
00000110110000011
```



Gambar 6. Proses Training program dengan input huruf "A"

Karakter huruf yang diinputkan oleh user tersebut selanjutnya akan disimpan pada database. Untuk melihat daftar karakter huruf apa saja yang telah diinputkan oleh user lebih jelasnya dapat dilihat pada gambar 7.



Gambar 7. Daftar karakter huruf pada database

**Proses Training**

Pada proses training input program merupakan vektor dengan 100 komponen yang merepresentasikan pola karakter 2 dimensi (10x10). Selain itu data yang harus diinputkan berupa nilai parameter alpha, treshold dan error.

Proses ini nantinya akan menghasilkan sejumlah nilai bobot dan bias yang sesuai dengan pola data training yang tersimpan pada database. Adapun contoh proses training untuk data ke-1 adalah sebagai berikut :

Training untuk data ke – 1

- Set data vektor sebagai data input.
- Hitung respon untuk nilai input

$$y_{in} = b + \sum_i x_i w_i \quad (i = 1 \text{ sampai } 100)$$

$$= 0 + ([\text{input vektor}] * 0)$$

$$= 0$$

- Hasil aktivasi = 0  
( 0 jika  $-0,1 \leq y_{in} \leq 0,1$  )
- Target = 1
- Perbaiki bobot dan bias jika terjadi eror  
Jika  $y \neq t$  maka :  
Bobot baru  
 $w_i(\text{baru}) = w_i(\text{lama}) + \alpha * t * x_i$   
 $w_i(\text{baru}) = 0 + 1 * 1 * (\text{input vektor})_i$   
Bobot bias baru :  
 $b_i(\text{baru}) = b_i(\text{lama}) + \alpha * t$   
 $b_i(\text{baru}) = 0 + 1 * 1$   
 $= 1$

Jika tidak, maka :

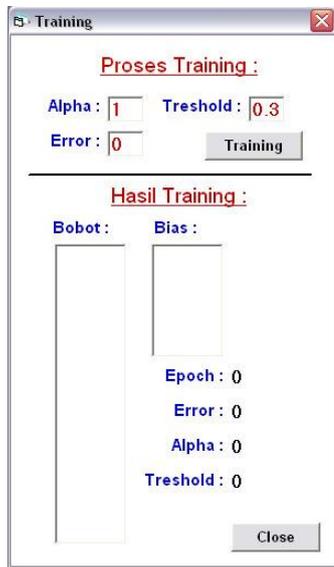
$$w_i(\text{baru}) = w_i(\text{lama})$$

$$b_i(\text{baru}) = b_i(\text{lama})$$

(berulang untuk i dari 1 sampai 100)

- Tes kondisi berhenti jika tidak terjadi perubahan bobot pada (i) maka kondisi berhenti TRUE, namun jika masih terjadi perubahan maka kondisi berhenti FALSE.  
Dan seterusnya untuk data – data selanjutnya.

Sistem ini juga menampilkan nilai epoch atau berapa jumlah iterasi yang dilakukan, lihat gambar 8.

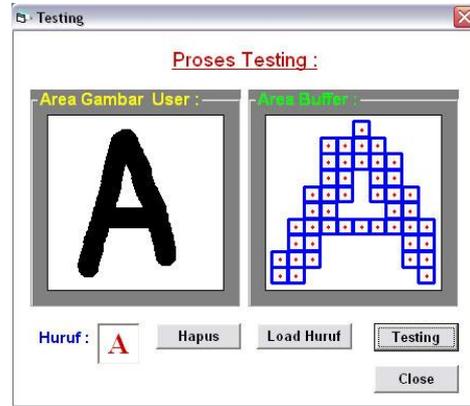


Gambar 8. Proses Training program

**Proses Testing**

Jaringan diuji dengan data baru setelah beberapa data di training untuk dikenali, hal ini bermaksud untuk menguji seberapa besar JST mengenali data yang baru. Dalam proses testing, user memasukkan

input karakter. Selanjutnya sistem akan menampilkan karakter apa yang sesuai dengan inputan user tersebut. Untuk lebih jelas lihat gambar 9 dan hasil uji coba dapat dilihat pada tabel 1.



Gambar 9. Proses Testing program

**KESIMPULAN**

Dari hasil penelitian dan pembahasan yang telah dilakukan menunjukkan bahwa proses binerisasi baik digunakan untuk segmentasi, dimana keberhasilannya mencapai hingga 90%. keberhasilan tersebut dipengaruhi oleh faktor ketepatan pemilihan nilai ambang, karena jika citra teks mengalami perubahan warna dan background maka nilai ambang yang tepat juga kemungkinan akan berubah.

Untuk penelitian lebih lanjut dapat digunakan metode segmentasi yang lain, atau bisa juga menggunakan metode binerisasi tetapi proses binerisasinya dibuat secara otomatis.

**DAFTAR PUSTAKA**

- [1] Fausett, L. 1994. "Fundamentals of Neural Networks: Architectures, Algorithms, and Applications", Prentice-Hall, New Jersey.
- [2] Kusumadewi, S. 2003, Artificial Intelligence Teknik dan Aplikasi, Graha Ilmu Yogyakarta
- [3] Munir, R. 2004, Pengolahan Citra Digital dengan Pendekatan Algoritmik, Informatika Bandung.
- [4] Siang, J.J. 2005. Jaringan Syaraf Tiruan dan Pemrogramannya Dengan Matlab. Yogyakarta: Penerbit Andi.