

Pengenalan Karakter Menggunakan Metode *Bidirectional Associative Memory* (BAM) Kontinu

Swingly Purba

Staf Pengajar Institut Sains Dan Teknologi TD. Pardede Medan

e-mail : swingly.purba61@gmail.com

ABSTRAK

Pengenalan karakter dapat dilakukan dengan menggunakan kecerdasan buatan (*Artificial Intelligence*), yaitu dengan menggunakan ilmu Jaringan Syaraf Tiruan (JST). Caranya adalah halaman buku yang mau diketik ke komputer, di-scan terlebih dahulu. Hasil gambar *scanner* kemudian diidentifikasi karakter per karakter dengan JST dan diubah secara otomatis menjadi karakter tulisan (*string*).

Salah satu metode JST yang dapat digunakan adalah *Bidirectional Associative Memory* (BAM) Kontinu. BAM adalah metode jaringan syaraf yang memiliki 2 lapisan, yaitu lapisan *input* dan lapisan *output* yang mempunyai hubungan timbal balik antara keduanya. Hubungan ini bersifat *bidirectional*, artinya jika bobot matriks dari sinyal yang dikirim dari lapisan *input* X ke lapisan output Y adalah W , maka bobot matriks dari sinyal yang dikirim dari lapisan *output* Y ke lapisan input X adalah W^T . Metode BAM Kontinu akan mengubah *input* ke *output* secara lebih halus dengan nilai yang terletak pada range $[0,1]$. Fungsi aktivasi yang digunakan adalah fungsi sigmoid.

Aplikasi hasil rancangan dapat melakukan segmentasi terhadap citra karakter, dan melakukan proses pelatihan serta proses pengenalan terhadap pola karakter dengan metode BAM.

Kata Kunci : Jaringan saraf tiruan, *Bidirectional Associative Memory*, fungsi sigmoid

ABSTRACT

Character recognition can be done by using artificial intelligence (Artificial Intelligence), that is by using the science of Artificial Neural Networks (ANN). The trick is a book page that you want to type on a computer, scan it first. The results of the scanner image are then identified as characters per character with ANN and are automatically converted to writing characters (strings).

One of the ANN methods that can be used is Continuous Bidirectional Associative Memory (BAM). BAM is a neural network method that has 2 layers, namely the input layer and the output layer which has a reciprocal relationship between the two. This relationship is bidirectional, meaning that if the matrix weight of the signal sent from the input X layer to the output layer Y is W , then the matrix weight of the signal sent from the output layer Y to the input layer X is W^T . The Continuous BAM method will change input to output more subtly with a value located in the range $[0,1]$. The activation function used is the sigmoid function.

Design applications can segment character images, and carry out the training process and the process of recognizing character patterns using the BAM method.

Keywords: Artificial neural networks, *Bidirectional Associative Memory*, sigmoid function

1. Latar Belakang

Pengetikan dokumen adalah suatu pekerjaan yang sering sekali dilakukan di komputer seperti sebuah buku atau artikel. Bila dokumen yang diketik, tidak terlalu panjang, maka tidak akan menjadi masalah. Namun, bila dokumen yang akan diketik, cukup banyak, maka akan membutuhkan waktu yang relatif lama untuk menyelesaikan pekerjaan tersebut. Pekerjaan ini dapat diefisienkan dengan menggunakan kecerdasan buatan (*Artificial Intelligence*). Caranya adalah halaman buku yang

mau diketik ke komputer, di-scan terlebih dahulu. Hasil gambar *scanner* kemudian diidentifikasi huruf per huruf dengan menggunakan jaringan syaraf tiruan dan diubah secara otomatis menjadi karakter tulisan (*string*).

Untuk melakukan pengenalan pola terhadap karakter tulisan, *input* gambar hasil *scan* dari halaman buku akan dilakukan proses segmentasi, yaitu dengan menggunakan proses *thresholding*, untuk memisahkan antara tulisan dan latar (piksel tulisan berwarna hitam dan piksel latar halaman berwarna

putih), dan melakukan segmentasi untuk memisahkan gambar antar huruf / karakter. Selanjutnya, gambar hasil segmentasi akan dikenali dengan menggunakan metode *Bidirectional Associative Memory* (BAM) Kontinu. *Bidirectional Associative Memory* (BAM) adalah metode jaringan syaraf yang memiliki 2 lapisan, yaitu lapisan *input* dan lapisan *output* yang mempunyai hubungan timbal balik antara keduanya. Hubungan ini bersifat *bidirectional*, artinya jika bobot matriks dari sinyal yang dikirim dari lapisan *input* X ke lapisan *output* Y adalah W , maka bobot matriks dari sinyal yang dikirim dari lapisan *output* Y ke lapisan *input* X adalah W^T . Metode BAM Kontinu akan mengubah *input* ke *output* secara lebih halus dengan nilai yang terletak pada range $[0,1]$. Fungsi aktivasi yang digunakan adalah fungsi sigmoid.

1.2. Perumusan Masalah

Bagaimana membuat aplikasi yang dapat mengenali karakter di dalam gambar dengan menggunakan segmentasi citra dan metode *Bidirectional Associative Memory* (BAM) Kontinu.

1.1.2. Tujuan

2. Mempelajari cara kerja proses *thresholding*, algoritma segmentasi dan metode *Bidirectional Associative Memory* (BAM) Kontinu dalam melatih dan mengenali pola karakter.
3. Merancang perangkat lunak yang dapat mengenal / membaca karakter dalam format citra.

2. URAIAN TEORITIS

2.1. Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) adalah paradigma pengolahan informasi yang terinspirasi oleh sistem syaraf secara biologis, seperti proses informasi pada otak manusia. Elemen kunci dari paradigma ini adalah struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (neuron), bekerja serentak untuk menyelesaikan masalah tertentu. Cara kerja JST seperti cara kerja manusia, yaitu belajar melalui contoh. Sebuah JST dikonfigurasi untuk aplikasi tertentu, seperti pengenalan pola atau klasifikasi data, melalui proses pembelajaran. Belajar dalam sistem biologis melibatkan penyesuaian terhadap koneksi synaptic yang ada antara neuron. Hal ini berlaku juga untuk JST.

JST mempunyai kemampuan yang luar biasa untuk mendapatkan informasi dari data yang rumit atau tidak tepat, mampu menyelesaikan permasalahan yang tidak terstruktur dan sulit didefinisikan, dapat belajar dari pengalaman, mampu mengakuisisi pengetahuan walaupun tidak ada kepastian, mampu

melakukan generalisasi dan ekstraksi dari suatu pola data tertentu, dapat menciptakan suatu pola pengetahuan melalui suatu pengaturan diri atau kemampuan belajar (*self organizing*), mampu memilih suatu *input* data ke dalam kategori tertentu yang sudah ditetapkan (klasifikasi), mampu menggambarkan suatu objek secara keseluruhan walaupun hanya diberikan sebagian data dari objek tersebut (asosiasi), mempunyai kemampuan mengolah data-data *input* tanpa harus mempunyai target (*self organizing*), dan mampu menemukan jawaban terbaik sehingga mampu meminimalisasi fungsi biaya (optimasi).

2.2. Paradigma Pembelajaran

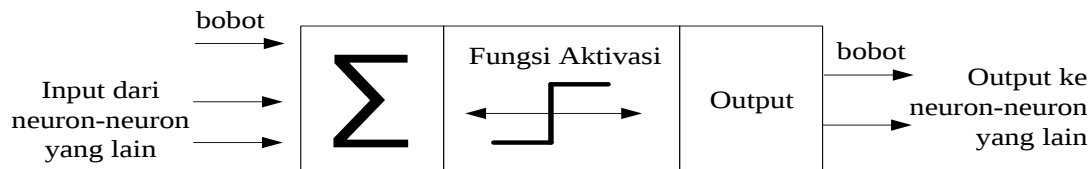
Bila dilihat dari cara memodifikasi bobotnya, pelatihan JST dibagi menjadi dua, yaitu pelatihan dengan supervisi (pembimbing), dan pelatihan tanpa supervisi. Pada proses pelatihan, suatu *input* dimasukkan ke jaringan, kemudian jaringan akan memproses dan mengeluarkan suatu keluaran. Keluaran yang dihasilkan oleh jaringan dibandingkan dengan target. Jika keluaran jaringan tidak sama dengan target, maka perlu dilakukan modifikasi bobot. Tujuan dari pelatihan ini adalah memodifikasi bobot hingga diperoleh bobot yang bisa membuat keluaran jaringan sama dengan target yang diinginkan.

Dalam pelatihan dengan supervisi, jaringan dipandu oleh sejumlah pasangan data (masukan dan target) yang berfungsi sebagai pembimbing untuk melatih jaringan hingga diperoleh bobot yang terbaik. Pelatihan dilakukan dengan memberikan pasangan pola-pola masukan dan keluaran. Sebaliknya, dalam pelatihan tanpa supervisi, tidak ada pembimbing yang digunakan untuk memandu proses pelatihan. Artinya, jaringan hanya diberi *input*, tetapi tidak mendapatkan target yang diinginkan, sehingga modifikasi bobot pada jaringan dilakukan menurut parameter tertentu. Sebagai contoh, pola-pola masukan yang tersedia diklasifikasikan ke dalam kelompok-kelompok yang berbeda.

2.3. Komponen Jaringan Syaraf Tiruan

Ada beberapa tipe jaringan syaraf, namun demikian, hampir semuanya memiliki komponen-komponen yang sama. Seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa neuron dan terdapat hubungan antar neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju ke neuron-neuron yang lain.

Pada jaringan syaraf, hubungan ini dikenal dengan bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Gambar 1 menunjukkan struktur neuron pada jaringan syaraf.



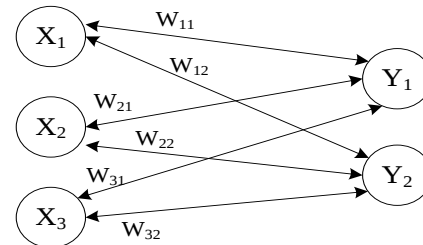
Gambar 1 Susunan Neuron Jaringan Syaraf

Jika dilihat, neuron buatan ini sebenarnya mirip dengan sel neuron biologis. Neuron-neuron buatan tersebut bekerja dengan cara yang sama pula dengan neuron-neuron biologis. Informasi (disebut dengan: input) akan dikirim ke neuron dengan bobot kedatangan tertentu. Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (threshold) tertentu melalui fungsi aktivasi setiap neuron. Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, tapi kalau tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya.

Pada jaringan syaraf, neuron-neuron akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (*neuron layers*). Biasanya neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya, kecuali lapisan input dan lapisan output. Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan input sampai ke lapisan output melalui lapisan yang lainnya, yang sering dikenal dengan nama lapisan tersembunyi (*hidden layer*). Tergantung pada algoritma pembelajarannya, bisa jadi informasi tersebut akan dirambatkan secara mundur.

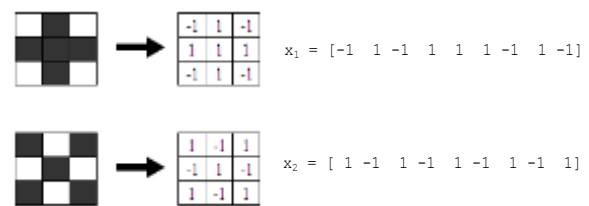
2.4. Bidirectional Associative Memory (BAM)

Bidirectional Associative Memory (BAM) adalah model jaringan syaraf yang memiliki 2 lapisan, yaitu lapisan *input* dan lapisan *output* yang mempunyai hubungan timbal balik antara keduanya. Hubungan ini bersifat *bidirectional*, artinya jika bobot matriks dari sinyal yang dikirim dari lapisan *input* X ke lapisan *output* Y adalah W , maka bobot matriks dari sinyal yang dikirim dari lapisan *output* Y ke lapisan *input* X adalah W^T . Arsitektur jaringan untuk 3 neuron pada lapisan *input* dan 2 neuron pada lapisan *output*, dapat dilihat pada gambar 2.



Gambar 2. Jaringan Bidirectional Associative Memory

BAM kontinu akan mengubah *input* ke *output* secara lebih halus dengan nilai yang terletak pada range $[0,1]$. Fungsi aktivasi yang digunakan adalah fungsi aktivasi sigmoid. Sebagai contoh, buatlah jaringan BAM untuk memetakan dua pola tanda “+” dan tanda “x” yang direpresentasikan dengan kode bipolar, seperti terlihat pada gambar 3 berikut.



Gambar 3. Pola Tanda “+” dan “x”

Pola tanda “+” disimpan dalam pola target $[1, -1]$, sedangkan pola tanda “x” disimpan dalam pola target $[1, 1]$. Dengan demikian, maka perhitungan matriks bobot untuk menyimpan tanda “+” dan tanda “x” adalah:

$$W_{(\text{tanda } +)} = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 1 \\ -1 \\ -1 \end{bmatrix} * [1 \ -1] = \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$W_{(\text{tanda } x)} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} * [1 \ 1] = \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \end{bmatrix}$$

$$W = W_{(tanda +)} + W_{(tanda x)}$$

$$W = \begin{bmatrix} -1 & 1 \\ 1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & -2 \\ 2 & 0 \\ 0 & -2 \\ 0 & 2 \\ 0 & -2 \end{bmatrix}$$

Matriks bobot W adalah matriks bobot yang menghubungkan antara neuron-neuron lapisan *input* ke lapisan *output*. Sedangkan matriks yang

menghubungkan neuron-neuron lapisan *output* ke lapisan *input* adalah W^T .

$$W^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & -2 & 2 & -2 & 0 & -2 & 2 & -2 & 2 \end{bmatrix}$$

Berikut dilakukan pengujian proses pengenalan terhadap masing-masing *input* vektor:

1. Pengujian jaringan jika diberi input vektor tanda “+”

$$y_{in_1} = x_1 * W = \begin{bmatrix} -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 \end{bmatrix} * \begin{bmatrix} 0 & 2 \\ 0 & -2 \\ 0 & 2 \\ 0 & -2 \\ 2 & 0 \\ 0 & -2 \\ 0 & 2 \\ 0 & -2 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 2 & -16 \end{bmatrix}$$

$$\text{Fungsi aktivasi : } y_j = \begin{cases} 1; & \text{jika } y_{in_j} > 0 \\ y_j; & \text{jika } y_{in_j} = 0 \\ -1; & \text{jika } y_{in_j} < 0 \end{cases}$$

Maka, $y_{in_1} = [1, -1]$ atau sama dengan target yang diharapkan, yaitu target dari vektor tanda “+”.

2. Pengujian jaringan jika diberi input vektor tanda “x”

$$y_{in_2} = x_2 * W = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 2 \\ 0 & -2 \\ 0 & 2 \\ 0 & -2 \\ 2 & 0 \\ 0 & -2 \\ 0 & 2 \\ 0 & -2 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 16 \end{bmatrix}$$

$$\text{Fungsi aktivasi : } y_j = \begin{cases} 1; & \text{jika } y_{in_j} > 0 \\ y_j; & \text{jika } y_{in_j} = 0 \\ -1; & \text{jika } y_{in_j} < 0 \end{cases}$$

Maka, $y_{in_2} = [1, 1]$ atau sama dengan target yang diharapkan, yaitu target dari vektor tanda “x”.

2.5.Segmentasi Citra

Segmentasi merupakan teknik untuk membagi suatu citra menjadi beberapa daerah (region) dimana setiap daerah memiliki kemiripan atribut.

2.6.Proses Pengambangan

Proses pengambangan (thresholding) akan menghasilkan citra biner, yaitu citra yang memiliki dua nilai tingkat keabuan, yaitu hitam dan putih. Secara umum, proses pengambangan citra *grayscale* untuk menghasilkan citra biner adalah sebagai berikut:

$$g(x,y) = \begin{cases} 1, & \text{if } f(x,y) \geq T \\ 0, & \text{if } f(x,y) < T \end{cases}$$

dengan $g(x,y)$ adalah citra biner dari citra *grayscale* $f(x,y)$, dan T menyatakan nilai ambang (*threshold*) yang bernilai di antara 1-255. Nilai T memegang peran sangat penting dalam proses pengambangan. Kualitas hasil citra biner sangat tergantung pada nilai T yang digunakan.

2.7.Penandaan Komponen Terhubung

Pemeriksaan komponen terhubung memeriksa suatu citra dan mengelompokkan setiap piksel ke dalam suatu komponen terhubung menurut aturan keterhubungan (4, 8 atau *m-connectivity*). Setiap komponen terhubung yang saling tidak terhubung (*disjoin*) pada suatu citra akan diberi tanda (label) berbeda. Memisahkan dan memberikan label pada setiap komponen terhubung maupun tidak terhubung pada suatu citra memegang peranan sentral pada beberapa aplikasi analisis citra secara otomatis.

Penandaan komponen terhubung untuk 8-connected hampir sama dengan 4-connected, hanya saja piksel tetangga yang diperiksa selain piksel atas dan kiri, juga kedua piksel diagonal atas dari piksel p, sehingga ada 4 piksel tetangga p yang diperiksa yaitu atas, kiri, diagonal atas kiri dan diagonal atas kanan dengan ketentuan berikut:

- Bila keempat piksel tetangga bernilai 0, maka berilah tanda baru pada p.
- Bila hanya salah satu piksel tetangga bernilai 1, maka berilah tanda dari piksel tetangga tersebut pada p.
- Bila dua atau lebih piksel tetangga bernilai 1, maka berilah salah satu tanda dari piksel tetangga tersebut pada p, kemudian buat catatan bahwa semua tanda dari piksel tetangga bernilai 1 tersebut adalah ekuivalen.

Proses berikutnya adalah membuat kelas-kelas ekuivalen dan memberi setiap kelas ekuivalen tanda yang berbeda. Langkah terakhir dari proses penandaan adalah melakukan proses *scanning* kembali pada citra dan ganti setiap tanda dengan tanda dari kelas ekuivalen.

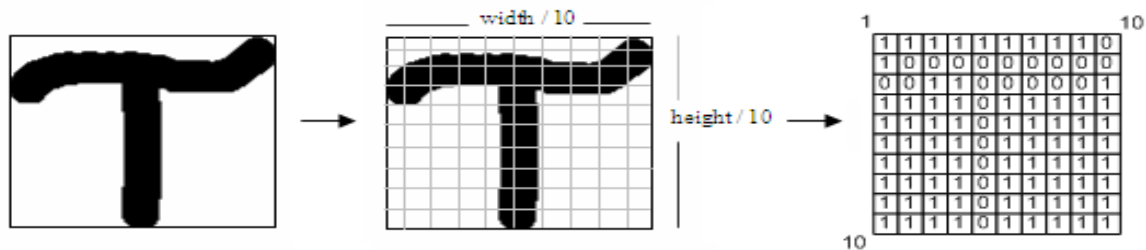
Ekstraksi Ciri

Proses ekstraksi ciri berfungsi untuk mengubah suatu pola menjadi bit-bit digital sehingga dapat dimengerti oleh komputer. Misalkan dilakukan proses ekstraksi ciri terhadap citra karakter sebagai berikut,



Gambar 4. Karakter ‘T’

Citra pola karakter dibagi menjadi matriks berukuran $m \times n$ (m baris dan n kolom). Pada contoh ini, matriks yang digunakan adalah 10×10 . Pada setiap kotak (elemen), dilakukan penelusuran. Jika ditemukan nilai 0 (piksel hitam) dalam jumlah tertentu, maka diberikan nilai 0 (warna hitam) pada kotak tersebut. Namun, jika tidak demikian, maka diberikan nilai 1 (warna putih) untuk kotak tersebut. Dari sini, diperoleh hasil ekstraksi ciri berupa nilai barisan bit 0 dan 1 sebanyak 100 bit.



Gambar 5. Proses Ekstraksi Ciri

3. PEMBAHASAN

Pembahasan mencakup pembahasan tahapan proses untuk mengenali karakter dari gambar, seperti:

1. Proses Pengambangan (Thresholding)
2. Proses Segmentasi dengan 8-connected
3. Proses Ekstraksi Ciri
4. Proses *Bidirectional Associative Memory*

3.1. Proses Pengambangan (Thresholding)

Fungsi dari proses pengambangan (*thresholding*) adalah mengubah citra berwarna menjadi citra biner yang hanya memiliki dua tingkat keabuan, yaitu hitam dan putih. Tujuannya adalah untuk memisahkan antara tulisan (yang akan diberi warna hitam) dan latar (yang akan diberi warna putih).

Proses pengambangan untuk menghasilkan citra hitam putih dapat dituliskan sebagai berikut:

$$g(x, y) = \begin{cases} 0 \text{ (piksel putih), if } f(x, y) \geq T \\ 1 \text{ (piksel hitam), if } f(x, y) < T \end{cases}$$

dengan:

$g(x, y)$ = citra biner sebagai *output* proses pengambangan

$f(x, y)$ = fungsi *grayscale* dari piksel citra pada posisi (x, y)

$f(x, y)$ = $(\text{Red} + \text{Green} + \text{Blue}) / 3$

T = nilai ambang (*threshold*) antara 1-255

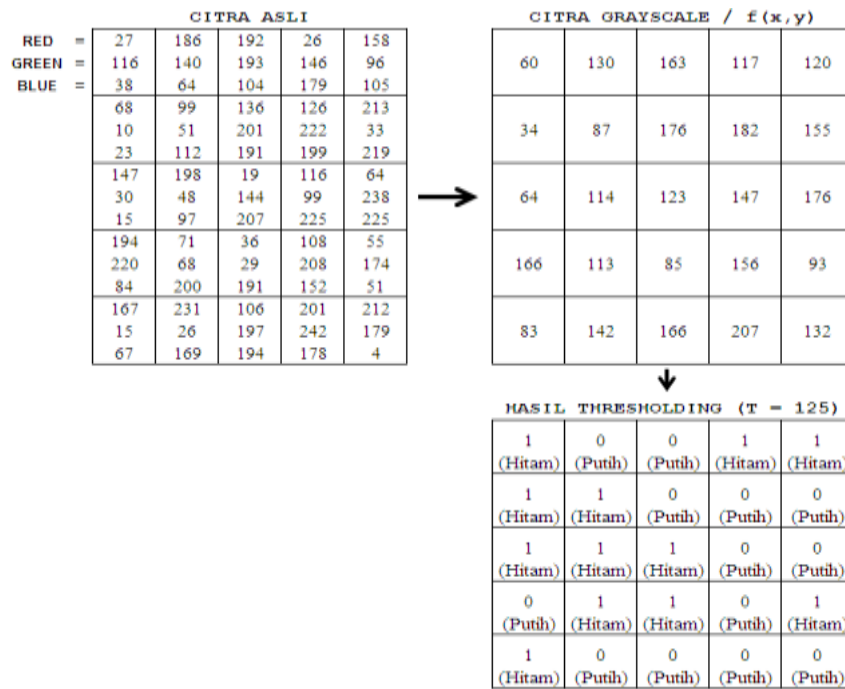
Agar lebih jelas, misalkan citra yang akan diubah menjadi citra biner dengan menggunakan proses pengambangan, seperti terlihat pada gambar 6.

CITRA ASLI					
RED =	27	186	192	26	158
GREEN =	116	140	193	146	96
BLUE =	38	64	104	179	105
	68	99	136	126	213
	10	51	201	222	33
	23	112	191	199	219
	147	198	19	116	64
	30	48	144	99	238
	15	97	207	225	225
	194	71	36	108	55
	220	68	29	208	174
	84	200	191	152	51
	167	231	106	201	212
	15	26	197	242	179
	67	169	194	178	4

Gambar 6. Citra Asli

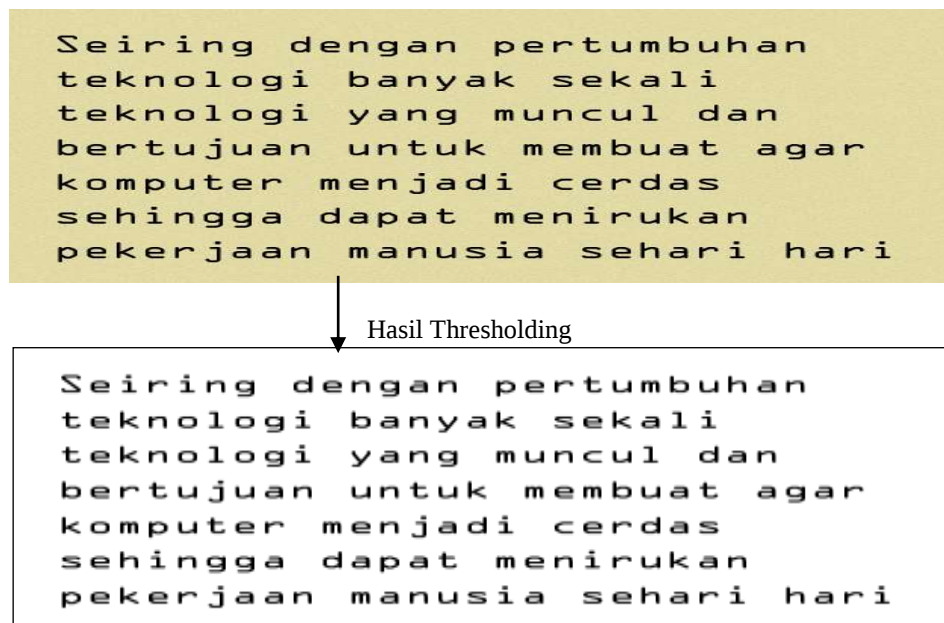
Misalkan nilai *Threshold* (T) = 125, maka proses pengambangan terhadap citra pada gambar III.1 adalah sebagai berikut:

1. Piksel-1 ($x = 1, y = 1$)
 $RGB = (27, 116, 38)$
 $f(x,y) = (27 + 116 + 38) / 3 = 60$
 $f(x,y) < T$, maka set warna piksel = hitam
2. Piksel-2 ($x = 2, y = 1$)
 $RGB = (186, 140, 64)$
 $f(x,y) = (186 + 140 + 64) / 3 = 130$
 $f(x,y) \geq T$, maka set warna piksel = putih
3. Piksel-3 ($x = 3, y = 1$)
 $RGB = (192, 193, 104)$
 $f(x,y) = (192 + 193 + 104) / 3 = 163$
 $f(x,y) \geq T$, maka set warna piksel = putih
4. Piksel-4 ($x = 4, y = 1$)
 $RGB = (26, 146, 179)$
 $f(x,y) = (26 + 146 + 179) / 3 = 117$
 $f(x,y) < T$, maka set warna piksel = hitam
5. Piksel-5 ($x = 5, y = 1$)
 $RGB = (158, 96, 105)$
 $f(x,y) = (158 + 96 + 105) / 3 = 120$
 $f(x,y) < T$, maka set warna piksel = hitam
6. dan seterusnya, hingga semua piksel di dalam citra diubah menjadi warna hitam / putih.
 Hasil proses pengambangan dapat dilihat pada gambar 7 berikut.



Gambar 7. Proses Pengambangan

Contoh proses *thresholding* terhadap gambar hasil *scan* tulisan dapat dilihat pada gambar 8 berikut.



Gambar 8. Proses *Thresholding* Terhadap Gambar Hasil Scan Tulisan

3.2. Proses Segmentasi dengan 8-connected

Algoritma 8-connected akan mengelompokkan setiap piksel yang saling terhubung ke dalam suatu *cluster*. Setiap komponen terhubung yang saling tidak terhubung (*disjoin*) pada suatu citra akan diberi tanda (label) berbeda. Memberikan label yang sama pada setiap piksel terhubung dan memberikan label berbeda pada piksel yang tidak

terhubung, akan dapat mensegmentasi setiap karakter di dalam gambar.

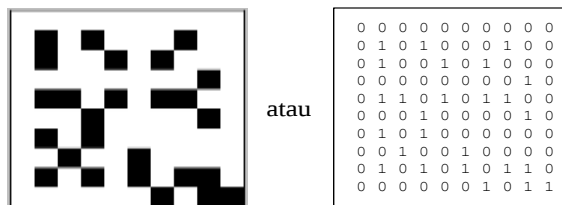
Penandaan komponen terhubung untuk 8-connected memiliki 4 piksel tetangga p yang diperiksa yaitu atas, kiri, diagonal atas kiri dan diagonal atas kanan. Bila piksel yang diperiksa saat ini bernilai 1 (piksel hitam), maka ikuti ketentuan berikut:

1. Bila keempat piksel tetangga bernilai 0 (putih), maka berilah tanda baru pada p.

2. Bila hanya salah satu piksel tetangga bernilai 1 (hitam), maka berilah tanda dari piksel tetangga tersebut pada p.
3. Bila dua atau lebih piksel tetangga bernilai 1, maka berilah salah satu tanda dari piksel tetangga tersebut pada p, kemudian buat catatan bahwa semua tanda dari piksel tetangga bernilai 1 tersebut adalah ekuivalen.

Proses terakhir adalah membuat kelas-kelas ekuivalen dan memberi setiap kelas ekuivalen tanda yang berbeda. Langkah terakhir dari proses penandaan adalah melakukan proses *scanning* kembali pada citra dan ganti setiap tanda dengan tanda dari kelas ekuivalen.

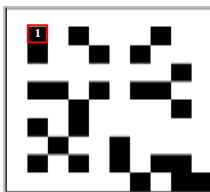
Misalkan, diberikan contoh gambar 9 untuk dikelompokkan, antara piksel-piksel yang terhubung dan yang tidak terhubung.



Gambar 9. Contoh Piksel-Piksel untuk Disegmentasi

Maka, proses pelabelan dengan *8-connected* adalah sebagai berikut:

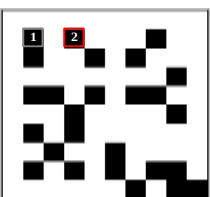
1. Piksel Hitam-1



Piksel Hitam-1

Tidak terdapat piksel tetangga (kiri, atas, diagonal atas kiri, diagonal atas kanan) yang berwarna hitam, berikan label baru untuk piksel hitam-1, yaitu label “1”

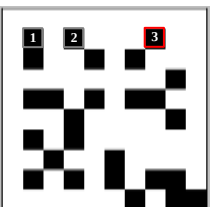
2. Piksel Hitam-2



Piksel Hitam-2

Tidak terdapat piksel tetangga (kiri, atas, diagonal atas kiri, diagonal atas kanan) yang berwarna hitam, berikan label baru, yaitu label “2”

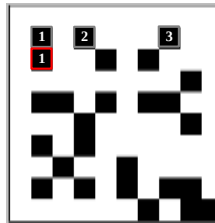
3. Piksel Hitam-3



Piksel Hitam-3

Tidak terdapat piksel tetangga (kiri, atas, diagonal atas kiri, diagonal atas kanan) yang berwarna hitam, berikan label baru, yaitu label “3”

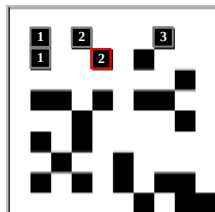
4. Piksel Hitam-4



Piksel Hitam-4

Terdapat 1 piksel tetangga (sebelah atas) yang memiliki piksel berwarna hitam, maka berikan tanda / label “1” terhadap piksel hitam-4

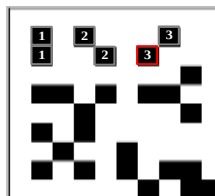
5. Piksel Hitam-5



Piksel Hitam-5

Terdapat 1 piksel tetangga (diagonal kiri atas) yang memiliki piksel berwarna hitam, maka berikan tanda / label “2” terhadap piksel hitam-5

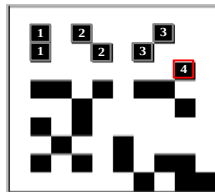
6. Piksel Hitam-6



Piksel Hitam-6

Terdapat 1 piksel tetangga (diagonal kanan atas) yang memiliki piksel berwarna hitam, maka berikan tanda / label “3” terhadap piksel hitam-6

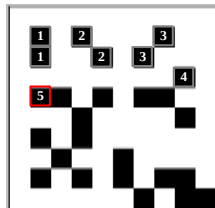
7. Piksel Hitam-7



Piksel Hitam-7

Tidak terdapat piksel tetangga (kiri, atas, diagonal atas kiri, diagonal atas kanan) yang berwarna hitam, berikan label baru, yaitu label “4”

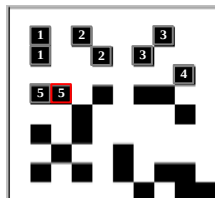
8. Piksel Hitam-8



Piksel Hitam-8

Tidak terdapat piksel tetangga (kiri, atas, diagonal atas kiri, diagonal atas kanan) yang berwarna hitam, berikan label baru, yaitu label “5”

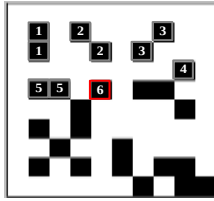
9. Piksel Hitam-9



Piksel Hitam-9

Terdapat 1 piksel tetangga (sebelah kiri) yang memiliki piksel berwarna hitam, maka berikan tanda / label “5” terhadap piksel hitam-9

10. Pikel Hitam-10



Pikel Hitam-10

Tidak terdapat piksel tetangga (kiri, atas, diagonal atas kiri, diagonal atas kanan) yang berwarna hitam, berikan label baru, yaitu label “6”

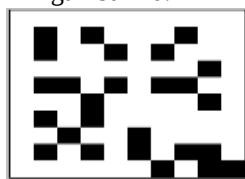
11. Dan seterusnya, hingga hasil akhir pelabelan piksel adalah sebagai berikut:

0	0	0	0	0	0	0	0	0	0
0	1	0	2	0	0	0	3	0	0
0	1	0	0	2	0	3	0	0	0
0	0	0	0	0	0	0	0	4	0
0	5	5	0	6	0	7	4	0	0
0	0	0	6	0	0	0	0	4	0
0	8	0	6	0	0	0	0	0	0
0	0	6	0	0	9	0	0	0	0
0	6	0	6	0	9	0	10	10	0
0	0	0	0	0	0	10	0	10	10

dengan catatan tanda yang ekuivalen, sebagai berikut:

- Tanda 4 = 7
- Tanda 5 = 6
- Tanda 6 = 8
- Tanda 9 = 10

12. Berikutnya, lakukan penyederhanaan berdasarkan pada catatan ekuivalen yang ada, sehingga hasil pelabelan seperti terlihat pada gambar 10.



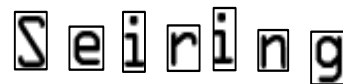
atau

0	0	0	0	0	0	0	0	0	0
0	1	0	2	0	0	0	3	0	0
0	1	0	0	2	0	3	0	0	0
0	0	0	0	0	0	0	0	4	0
0	5	5	0	5	0	4	4	0	0
0	0	0	5	0	0	0	0	4	0
0	5	0	5	0	0	0	0	0	0
0	0	5	0	0	9	0	0	9	0
0	5	0	5	0	9	0	9	9	0
0	0	0	0	0	9	0	9	9	9

Gambar 10. Hasil Pelabelan

Dapat dilihat bahwa, algoritma 8-connected berhasil memberikan tanda / label yang sama untuk piksel yang berhubungan, dan tanda / label yang berbeda untuk piksel yang tidak berhubungan. Dengan demikian, maka piksel-piksel yang terhubung dapat dipisahkan / disegmentasi dengan mudah, berdasarkan label yang dimilikinya.

Contoh hasil segmentasi terhadap citra biner hasil proses *thresholding* pada gambar 8 dengan menggunakan pelabelan 8-connected dapat dilihat pada gambar 11.

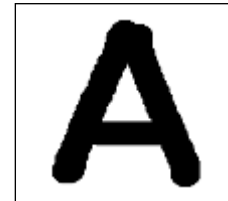


Gambar 11 Hasil Segmentasi Per Karakter

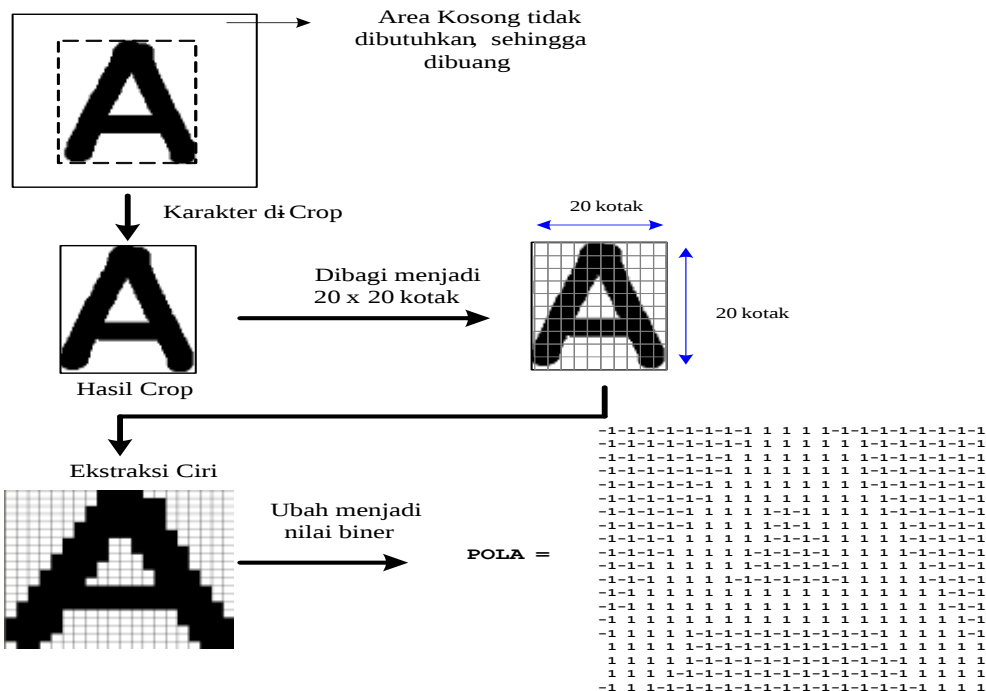
Proses Ekstraksi Ciri

Setelah di-segmentasi, maka proses berikutnya adalah setiap gambar karakter akan didigitalisasi menjadi bit-bit biner melalui proses ekstraksi ciri.

Sebagai contoh, misalkan huruf / karakter yang akan diekstraksi ciri seperti terlihat pada gambar III.7. Proses ekstraksi ciri terhadap gambar 11 ke dalam pola 20 x 20 dapat dilihat pada gambar 12.



Gambar 12. Huruf “A”



Gambar 12. Proses Ekstraksi Ciri Terhadap Huruf “A”

Algoritma proses ekstraksi ciri dapat dituliskan sebagai berikut:

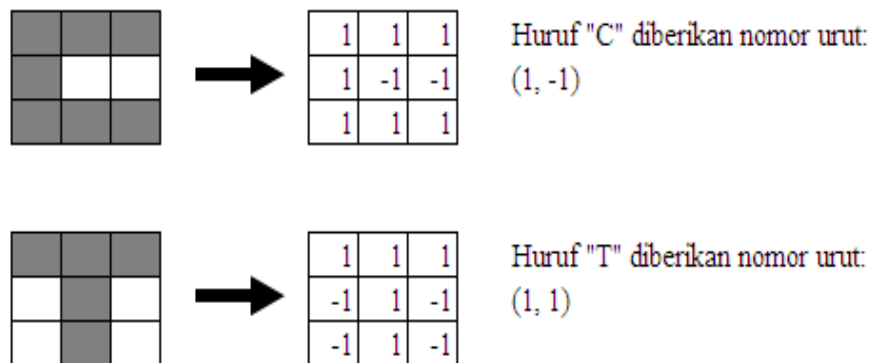
1. Pada gambar coretan, periksa dari kiri ke kanan. Apabila suatu kolom mempunyai piksel hitam, maka set $X1 = \text{nomor / posisi kolom}$.
2. Periksa dari kanan ke kiri. Apabila suatu kolom mempunyai piksel hitam, maka set $X2 = \text{nomor / posisi kolom}$.
3. Periksa dari atas ke bawah. Apabila suatu baris mempunyai piksel hitam, maka set $Y1 = \text{nomor / posisi baris}$.
4. Periksa dari bawah ke atas. Apabila suatu baris mempunyai piksel hitam, maka set $Y2 = \text{nomor / posisi baris}$.
5. *Crop* gambar dari posisi $(X1, Y1)$ sampai posisi $(X2, Y2)$. Hasil *crop* adalah coretan huruf / karakter.
6. Hasil *crop* kemudian dibagi menjadi 20×20 kotak.

7. Periksa setiap kotak, apabila terdapat lebih banyak piksel hitam daripada piksel putih di dalam area kotak, maka set warna kotak ke hitam. Bila tidak, maka set warna ke putih.
8. Lakukan proses nomor-7 pada semua kotak yang berjumlah 400 buah (20×20), sehingga semua kotak mempunyai warna hitam / putih.
9. Lakukan digitalisasi bit biner untuk setiap kotak. Kotak yang berwarna hitam diwakili oleh nilai 1, sedangkan kotak yang berwarna putih diwakili oleh nilai -1.
10. Hasil ekstraksi ciri adalah berupa barisan bit biner 0 atau 1 sebanyak 400 digit.

Selanjutnya, bit-bit hasil ekstraksi ciri merupakan representasi digital dari karakter, dan digunakan dalam fase pelatihan atau pengenalan metode *Bidirectional Associative Memory* (BAM).

Proses Bidirectional Associative Memory (BAM)

Algoritma BAM digunakan untuk mengenali hasil ekstraksi pola dari karakter / huruf. Proses ini terdiri dari fase pelatihan dan pengenalan. Sebagai contoh sederhana, misalkan terdapat 2 buah karakter “C” dan “T” dengan ekstraksi ciri 3×3 sebagai berikut:



Gambar 13. Pola Karakter “C” dan “T”

Proses pelatihan dengan menggunakan metode BAM dari kedua karakter di atas adalah sebagai berikut:

1. Perhitungan bobot untuk huruf “C”

$$W_{(\text{huruf C})} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} * [1 \ -1] = \begin{bmatrix} 1 & -1 \\ 1 & -1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ 1 & -1 \end{bmatrix}$$

2. Perhitungan bobot untuk huruf “T”

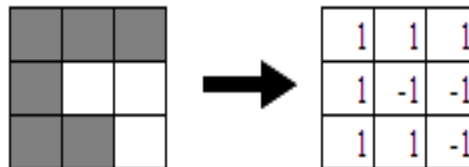
$$W_{(\text{huruf T})} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{bmatrix} * [1 \ 1] = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \end{bmatrix}$$

3. Perhitungan bobot akhir (W):

$$W = W_{(huruf C)} + W_{(huruf T)}$$

$$W = \begin{bmatrix} 1 & -1 \\ 1 & -1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ 1 & -1 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 2 & 0 \\ 2 & 0 \\ 0 & -2 \\ 0 & 2 \\ -2 & 0 \\ 0 & -2 \\ 2 & 0 \\ 0 & -2 \end{bmatrix}$$

Hasil bobot w disimpan dan akan digunakan pada saat proses pengenalan. Berikut adalah contoh proses pengenalan. Sebagai contoh, misalkan diberikan hasil ekstraksi ciri 3 x 3 sebagai berikut untuk dikenali:



Gambar 14. Pola Karakter untuk Dikenali

Proses perhitungan untuk pengenalan karakter pada gambar 15 adalah sebagai berikut

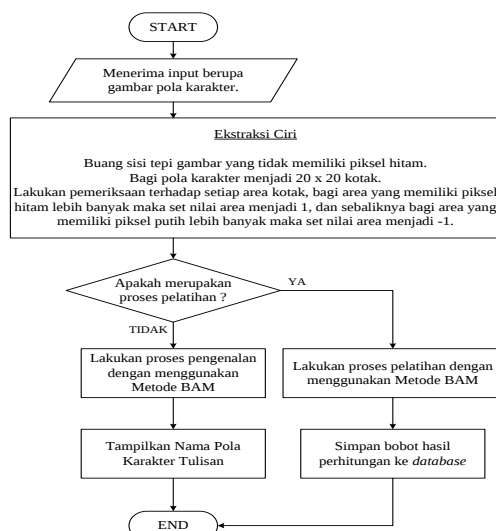
$$y_{in_1} = x_1 * W = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \end{bmatrix} * \begin{bmatrix} 2 & 0 \\ 2 & 0 \\ 2 & 0 \\ 0 & -2 \\ 0 & 2 \\ -2 & 0 \\ 0 & -2 \\ 2 & 0 \\ 0 & -2 \end{bmatrix} = [10, -4]$$

Masukkan y_{in} ke fungsi aktivasi untuk menghasilkan y .

$$\text{Fungsi aktivasi : } y_j = \begin{cases} 1; & \text{jika } y_{in_j} > 0 \\ y_j; & \text{jika } y_{in_j} = 0 \\ -1; & \text{jika } y_{in_j} < 0 \end{cases}$$

Maka, $y = [1, -1]$ atau sama dengan nomor urut yang diharapkan, yaitu nomor urut dari vektor huruf “C”.

Secara garis besar, proses pengenalan karakter ini dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar 15 berikut.



Gambar 15. Gambaran Sistem Pengenalan Karakter / Huruf

Algoritma

Algoritma yang digunakan dibagi menjadi beberapa bagian yaitu:

1. Algoritma Pengambangan (Thresholding)
2. Algoritma Segmentasi
3. Algoritma Ekstraksi Ciri
4. Algoritma *Bidirectional Associative Memory*

Algoritma Pengambangan (Thresholding)

Algoritma ini berfungsi sebagai *pre-processing* terhadap gambar karakter. Proses *thresholding* (pengambangan) akan mengubah dan menyederhanakan gambar menjadi gambar hitam putih, sehingga dapat diproses lebih lanjut oleh algoritma proses ekstraksi ciri.

Algoritma proses *thresholding* adalah sebagai berikut:

1. Untuk $B = 0$ sampai (tinggi gambar – 1), lakukan prosedur berikut.

2. Untuk $A = 0$ sampai (lebar gambar – 1), lakukan prosedur berikut.
3. Set Red = nilai intensitas warna merah pada piksel (A, B).
4. Set Green = nilai intensitas warna hijau pada piksel (A, B).
5. Set Blue = nilai intensitas warna biru pada piksel (A, B).
6. Set Grayscale = (Red + Green + Blue) / 3.
7. Jika Grayscale $\geq$ Threshold, maka set piksel putih pada posisi (A, B). Jika tidak, maka set piksel hitam pada posisi (A, B).
8. Next A
9. Next B

Algoritma Segmentasi

Proses segmentasi menggunakan algoritma *8-Connected*. Fungsi dari proses ini adalah untuk melakukan pemisahan terhadap karakter yang tidak terhubung satu sama lain.

Algoritmanya adalah untuk nilai $Y = 0$ sampai (tinggi_gambar) – 1 dan untuk nilai $X = 0$ sampai (lebar_gambar) – 1, maka lakukan algoritma berikut:

1. Untuk nilai $Y = 0$ sampai (tinggi_gambar) – 1, maka lakukan poin 2 sampai poin 4.
2. Untuk nilai $X = 0$ sampai (lebar_gambar) – 1, maka lakukan poin 3 sampai poin 4.
3. p = nilai piksel pada posisi (X, Y)
4. Jika warna piksel adalah hitam ($p = 0$), maka:
 - a. Cek piksel tetangga,
 - b. $PT(1)$ = nilai piksel sebelah kiri
 - c. $PT(2)$ = nilai piksel sebelah atas
 - d. $PT(3)$ = nilai piksel sebelah kiri atas
 - e. $PT(4)$ = nilai piksel sebelah kanan atas
 - f. Jika $PT(1) + PT(2) + PT(3) + PT(4) = (255 * 4)$, maka beri tanda baru ke piksel saat ini:
 - i. $nTanda = nTanda + 1$
 - ii. $ATanda(X, Y) = nTanda$
 - g. Jika $PT(1) + PT(2) + PT(3) + PT(4) = (255 * 3)$, maka beri tanda dari piksel tetangga ke piksel saat ini:
 - i. $tmpStr1 = ""$
 - ii. Jika $PT(1) = 0$, maka $tmpStr1 = ATanda(X - 1, Y)$
 - iii. Jika $PT(2) = 0$, maka $tmpStr1 = ATanda(X, Y - 1)$
 - iv. Jika $PT(3) = 0$, maka $tmpStr1 = ATanda(X - 1, Y - 1)$
 - v. Jika $PT(4) = 0$, maka $tmpStr1 = ATanda(X + 1, Y - 1)$
 - vi. $ATanda(X, Y) = tmpStr1$
 - h. Jika tidak, maka beri salah satu tanda terkecil dari piksel tetangga ke piksel saat ini, dan berikan catatan bahwa semua tanda itu adalah ekuivalen.
5. Periksa semua piksel kembali, dan ubah tanda-tanda yang ekuivalen menjadi tanda yang sama.

6. Potong semua karakter dengan tanda ekuivalen yang sama.

Algoritma Ekstraksi Ciri

Algoritma ekstraksi ciri akan membagi pola karakter menjadi pola 20×20 . Hasil ekstraksi ciri adalah 400 bit bipolar (1/-1), yang akan menjadi ciri dari suatu pola tanda tangan di dalam pengetahuan JST.

Algoritma ekstraksi ciri adalah sebagai berikut:

1. Set LebarKotak = (lebar gambar hasil *cropping*) / 20
2. Set TinggiKotak = (tinggi gambar hasil *cropping*) / 20
3. Untuk $B = 1$ sampai 20, lakukan prosedur berikut.
4. Untuk $A = 1$ sampai 20, lakukan prosedur berikut.
5. Set $X2 = (\text{LebarKotak} * A) - 1$
6. Set $X1 = (X2 - \text{LebarKotak}) + 1$
7. Set $Y2 = (\text{TinggiKotak} * B) - 1$
8. Set $Y1 = (Y2 - \text{TinggiKotak}) + 1$
9. Set $nHitam = 0$ dan $nPutih = 0$.
10. Untuk $K2 = Y1$ sampai $Y2$, lakukan prosedur berikut.
11. Untuk $K1 = X1$ sampai $X2$, lakukan prosedur berikut.
12. Jika warna piksel pada posisi ($K1, K2$) = hitam, maka set $nHitam = nHitam + 1$. Jika tidak, maka set $nPutih = nPutih + 1$.
13. Next $K1$
14. Next $K2$
15. Set $K1 = (B - 1) * nY + A$.
16. Jika total warna piksel hitam lebih dari total piksel putih di dalam kotak, maka set warna kotak ke- $K1$ menjadi hitam (atau bit 1). Jika tidak, maka set warna kotak ke- $K1$ menjadi putih (atau bit -1).

Algoritma Bi-directional Associative Memory (BAM)

Metode BAM terdiri dari 2 fase, yaitu fase pelatihan dan fase pengenalan. Berikut adalah algoritma fase pelatihan metode BAM:

1. X = ekstraksi ciri dari karakter di dalam database
2. T = nomor urut karakter di dalam database.
3. Untuk setiap karakter di dalam database, Untuk $i=1$ sampai 200 (banyak bit ekstraksi ciri), Untuk $j=1$ sampai 8 (banyak bit nomor urut), Hitung $W(i,j) = X(i) * T(j)$
Next j
Next i
4. Jumlahkan semua bobot ke W .
5. Simpan bobot W ke database.

Berikut adalah algoritma fase pengenalan karakter dengan menggunakan metode BAM:

1. X = hasil ekstraksi ciri karakter yang ingin dikenali
2. Load bobot W dari database.
3. Y_{in_j} = perkalian antara X dan W .
4. Lakukan fungsi aktivasi,
Jika $Y_{in_j} > 0$, maka $Y_j = 1$
Jika $Y_{in_j} < 0$, maka $Y_j = -1$
5. Urutkan bit Y_j dan hasil pengenalan adalah karakter yang mempunyai nomor urut / target sama dengan urutan bit bipolar Y_j .

Kesimpulan

1. Aplikasi dapat melakukan segmentasi terhadap citra karakter, dan melakukan proses pelatihan serta proses pengenalan terhadap pola atau bentuk karakter, yang pada akhirnya dapat mengenali setiap karakter di dalam kalimat.
2. Proses pelatihan dapat dilakukan beberapa kali untuk satu pola karakter, untuk meningkatkan akurasi pengenalan terhadap pola tersebut.

REFERENSI

- Kusumadewi, S., **Artificial Intelligence (Teknik dan Aplikasinya)**, Penerbit Graha Ilmu, Yogyakarta, 2003
- Munir, R., **Pengolahan Citra Digital**, Penerbit Informatika, Bandung, 2004
- Putra, D., **Pengolahan Citra Digital**, Penerbit Andi, Yogyakarta, 2010.
- Puspitaningrum, D., **Pengantar Jaringan Syaraf Tiruan**, Penerbit Andi, Yogyakarta, 2006
- Sutojo, T., Mulyanto, E. dan Suhartono, V., **Kecerdasan Buatan**, Penerbit Andi, Yogyakarta, 2011