

Metode Ekstraksi Data Untuk Pengenalan Huruf Dan Angka Tulisan Tangan Dengan Menggunakan Jaringan Syaraf Buatan Propagasi Balik

A.P. Nugraha dan A.B. Mutiara

Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma,
Jl. Margonda Raya No 100, Depok
Tel. 78881112 ext.309 email: amutiara@staff.gunadarma.ac.id

Abstract– Pada penulisan ini disajikan suatu metode untuk mengenali pola huruf dan angka tulisan tangan dengan menggunakan konsep jaringan syaraf buatan propagasi balik. Jaringan syaraf buatan yang digunakan menggunakan fungsi aktivasi sigmoid pada semua la-pisannya, juga struktur neuronnya bersifat kaku (tidak fleksibel). Adapun metode pengenalan pola huruf dan angka tulisan tangan tersebut difokuskan pada cara pengekstrakan data dari sampel yang telah ada. Langkah-langkah pengekstrakan data tersebut adalah pertama, membagi sampel menjadi beberapa area pengamatan (region). Kedua, dari setiap region tersebut diambil nilai pixel yang aktifnya, sehingga didapat data numerik sebanyak jumlah region yang ada. Ketiga, data numerik yang sudah dihasilkan, dinormali-sasi dengan cara perbandingan tetap, dimana setiap data numerik dari masing-masing region tadi dibagi dengan nilai terbesar dari semua data numerik dari sampel yang sama. Untuk membuktikan metode tersebut penulis membuat suatu aplikasi pendukung yaitu Char-Cog-nitron dengan menggunakan bahasa pemrograman Delphi 6.0 Enterprise. Adapun langkah-langkah peng-analisaan yang pertama kali dilakukan adalah menentu-kan karakteristik jaringan syaraf buatan terbaik untuk metode ini. Kemudian, dianalisa komposisi region ter-baik untuk sampel yang akan digunakan sehingga dapat memberikan hasil akhir yang terbaik. Hasil akhir dari penganalisaan tersebut menunjukkan bahwa pola yang dihasilkan dengan menggunakan metode tersebut dapat dikenali dengan baik oleh jaringan syaraf buatan.

Keywords– Jaringan saraf buatan, propagasi balik, neuron, Learn rate

I. PENDAHULUAN

Secara awam, aspek yang cukup penting yang men-dasari berbagai teori dalam AI adalah sistem pengenalan pola (*Pattern Recognizing*) yang merupakan bagian dari pengimplementasian *Artificial Neural Network (ANN)* atau Jaringan Syaraf Buatan secara praktis. Sistem pengenalan pola merupakan komponen penting dalam proses peniruan kemampuan inderawi manusia terutama penglihatan dan pendengaran. Sebagai contoh, untuk meniru indera penglihatan manusia, komputer harus mempunyai suatu mekanisme standar dan logis dalam mengenali pola yang ada pada suatu citra yang sedang diproses. Dari sinilah diperoleh motivasi untuk mencoba suatu konsep sederhana untuk mengenali pola dari suatu citra sehingga dapat diidentifikasi dengan baik oleh komputer.

Permasalahan utama yang terjadi apabila hendak mengenali suatu pola tertentu dan mengumpulkannya ke dalam suatu proses jaringan syaraf buatan adalah bagaimana proses akuisisi data dilakukan sehingga menghasilkan sejumlah data numerik yang representatif dan konsisten terhadap sampel yang diberikan.

Dalam penulisan ini, kita mencoba mengaplikasikan suatu metoda sederhana untuk mengenali suatu citra sehingga dapat diidentifikasi dengan baik oleh komputer dengan memanfaatkan berbagai teori seperti *Image Processing*, *Artificial Neural Networks*. Citra

yang akan diproses akan dibatasi pada citra yang merepresentasikan huruf-huruf alpabet berikut angka-angka arab secara singular. Program bantu yang penulis buat untuk mensimulasikan metoda ini dibuat dengan menggunakan Bahasa Pemrograman Borland Delphi 6.0 Enterprise pada sistem operasi yang mendukung.

Tujuan utama dari penulisan ini adalah untuk menganalisis dan membuktikan bahwa metoda sederhana untuk mengekstraksi data dari sampel yang berupa huruf dan angka tulisan tangan yang penulis buat dapat diimplementasikan dengan baik sesuai dengan tujuannya sehingga komputer dapat mengidentifikasi citra huruf dan angka tulisan tangan secara baik dan konsisten. Hal ini dapat diukur dengan melihat bahwa dengan menggunakan metode yang dimaksud, suatu jaringan syaraf buatan tertentu dapat ‘mempelajarinya’ serta mengenali pola-pola yang diberikan. Selain itu diharapkan dapat diperoleh konsep sederhana yang dapat mendasari dan dapat diimplementasikan pada suatu proses otomatisasi, misalnya OCR (Object Character Reader).

II. TEORI

A. Unsur Pengolahan Citra

Secara umum dan sederhana, citra dapat didefinisikan sebagai representasi visual dari suatu objek. Lebih jauh citra juga dapat diartikan sebagai gambaran yang representatif mengenai suatu objek sedemikian sehingga citra tersebut dapat memberikan

kesan yang mendalam mengenai objek yang dimaksud. Jika ingin mendefinisikannya lebih bebas lagi, citra dapat didefinisikan sebagai bentuk visual yang dapat diterima secara baik oleh indera penglihatan, apapun bentuknya. Dalam bidang komputer, citra atau disebut juga *image* merupakan representasi visual dari suatu objek setelah mengalami berbagai transformasi data dari berbagai bentuk rangkaian numerik.

Komputer memiliki cara pandang tersendiri terhadap suatu citra. Berbeda dengan citra konvensional yang misalnya dengan melalui proses fotografis seperti pada foto dapat dihasilkan suatu citra nyata yang langsung dapat dinikmati oleh indera penglihatan, citra pada komputer harus melalui beberapa tahapan yang cukup rumit. Tahapan-tahapan tersebut dapat digambarkan sebagai suatu rangkaian proses dari proses akuisisi data, manipulasi data, visualisasi data, serta proses penyimpanan data. Penjelasan lebih lanjut dari proses-proses tersebut dapat dilihat pada [8].

Diantara proses tersebut diatas, proses penyimpanan data adalah bagian paling pelik dari pengolahan citra pada komputer. Saat ini sudah ter-dapat banyak metode yang dapat digunakan untuk menyimpan suatu citra digital ke dalam suatu file. File adalah media penyimpanan standar pada sistem komputer. Metode-metode penyimpanan tadi tentunya memiliki perbedaan yang cukup berarti antara satu dengan yang lainnya. Hal yang paling membedakan biasanya adalah tingkat kompresi data yang ditawarkan oleh masing-masing metode. Beberapa metode yang sudah cukup dikenal adalah BMP, JPEG, PCX, WMF, TIFF, dll.

B. Jaringan Syaraf Propagasai Balik

Propagasi balik merupakan salah satu proses belajar jaringan syaraf buatan dimana dalam proses belajar tersebut pengubahan nilai berlangsung pada arah mundur, yaitu mulai dari lapisan output dan berakhir di lapisan input.

Algoritma belajar propagasi balik adalah sebagai berikut [5] :

1. Perhitungan nilai keluaran neuron pada lapisan tersembunyi dan output.

$$net_i = \sum_{j=1}^N (w_{ij}s_j) + q_i$$

$$s_i = f(net_i)$$

dimana,

- i = nomor neuron yang sedang dihitung sinyal aktivasinya.
 - j = nomor neuron yang outputnya dikontribusikan pada neuron i
 - s_j = nilai output neuron j
 - w_{ij} = nilai bobot hubungan antara neuron ke i dan ke j
 - q_i = nilai bias neuron ke i
- fungsi net_i atau f(net_i) disebut juga sebagai fungsi aktivasi yang bentuknya dapat bermacam-macam.

2. Perhitungan kesalahan dalam proses belajar disebut dengan fungsi energi.

$$E = \sum_{(x,y)} \sum_i (y_i^x - s_i^x)^2$$

dimana,

- y_i = output target neuron output i
- s_i^x = output aktual neuron output I pada saat jaringan terhubung dengan sample x

3. Perhitungan sensitivitas (ä_i) neuron-neuron dalam lapisan tersembunyi dan lapisan output. Dimana persamaan yang digunakan untuk perhitungan sensitivitas baik untuk lapisan tersembunyi maupun lapisan output, tergantung dari fungsi aktivasi yang digunakan.
4. Perhitungan nilai perubahan bobot dan bias.

perubahan bobot : $\Delta w_{ij}(x, y) = \Delta d_i s_j^x$

perubahan bias : $\Delta q_i(x, y) = \Delta d_i$

5. Perhitungan nilai bobot dan bias baru.

bobot baru :

$$w_{ij}^{t+1} = w_{ij}^t + \Delta w_{ij}^t + momentum \cdot \Delta w_{ij}^{t-1}$$

bias baru :

$$q_i^{t+1} = q_i^t + \Delta q_i^t$$

6. Langkah-langkah tersebut diulang sampai deviasi keluaran kecil sehingga mencapai stopping kriteria error yang diharapkan.

Fungsi aktivasi dapat diibaratkan sebagai pendefinisian penguatan non linier dalam sistem analog (*continue*). Penguatan (*gain*) ini dihitung dengan mencari rasio perubahan pada fungsi output neuron pada lapisan tersembunyi dan lapisan keluaran, dimana penguatan merupakan kemiringan kurva pada suatu tingkat eksistensi tertentu, dan nilainya berubah dari harga yang kecil pada eksitasi negatif yang besar (kurvanya hampir mendatar) menjadi harga yang besar pada eksitasi nol, dan nilainya kembali mengecil seiring dengan eksitasi yang semakin besar dan positif.

Fungsi aktivasi tidak hanya digunakan pada saat perhitungan nilai keluaran neuron saja (neuron pada lapisan tersembunyi dan lapisan output), tetapi turunan pertamanya digunakan juga untuk menghitung perubahan bobot dan bias pada proses belajar. Dalam penulisan ini digunakan fungsi aktivasi Sigmoid

Parameter penting lainnya dari JSB adalah bagaimana output dari JSB tersebut direpresentasikan, secara terlokalisasi atau terdistribusi. Dalam penulisan ini akan digunakan metode output yang direpresentasikan secara terdistribusi. Hal ini dikarenakan banyaknya jumlah variasi output yang harus direpresentasikan, sehingga tidaklah efektif bila direpresentasikan dengan jumlah digit sebanyak variasi output yang berbeda. Perlu diketahui, seperti yang telah disebutkan pada bagian batasan masalah, bahwa citra yang akan

dicoba untuk dikenali adalah citra huruf-huruf alpabet (huruf kecil maupun huruf kapital) berikut angka-angka arab satuan, yang berarti ada sejumlah 62 (13 huruf kecil+13 huruf besar+10 angka arab satuan) objek yang harus direpresentasikan sebagai target output. Tentunya sangatlah tidak efektif bila 62 objek tersebut direpresentasikan dengan sejumlah 62 digit biner output.

Pada metode output terdistribusi, setiap huruf alpabet dan angka arab akan dikonversi ke dalam bentuk pengkodean 6 bit. Berarti hanya diperlukan 6 neuron yang akan digunakan sebagai output pada lapisan output JSB. Tetapi dengan pengkodean 6 bit berarti tersedia $2^6 = 64$ variasi output yang berbeda, sedangkan simbol atau objek yang akan direpresentasikan hanya ada 62 objek, untuk itu penulis menambahkan dua objek tambahan untuk melengkapi ketersediaan pengkodean, kedua objek tersebut adalah '>' dan '?'.
'>' dan '?'.

III. IMPLEMENTASI KONSEP

A. Analisi Masalah

Suatu citra huruf atau angka (selanjutnya akan disebut sampel) yang akan diidentifikasi menggunakan JSB, haruslah melalui tahapan-tahapan tertentu terlebih dahulu sehingga dapat menjadi input yang baik bagi JSB. Adapun input yang dapat diterima oleh dengan baik oleh JSB adalah berupa kumpulan data numerik. Dengan demikian maka permasalahan pertama adalah bagaimana mengkonversi suatu citra digital menjadi kumpulan data numerik yang representatif dan konsisten. Tentunya walaupun JSB yang digunakan memiliki karakteristik, struktur ataupun konfigurasi yang sudah cukup spesifik, masih ada parameter-parameter penting lain dari JSB tersebut yang harus diatur untuk mendapatkan hasil terbaik. Parameter-parameter tersebut antara lain adalah pengaruh bias dan bobot awal, nilai momentum, nilai *LearnRates*, nilai *stopping criteria error*, *epoch*, dan lain sebagainya. Penyesuaian terhadap nilai-nilai tersebut adalah masalah berikutnya yang akan dianalisa.

B. Metode Akuisisi Data

Setiap sampel yang akan diamati dan dianalisa oleh JSB harus direpresentasikan secara baik ke dalam bentuk data numerik. Untuk itu diperlukan suatu metode yang dapat mengekstraksi data ciri dari setiap sampel tadi secara konsisten. Tentunya data numerik yang dihasilkan haruslah benar-benar dapat mewakili karakteristik atau ciri-ciri dari sampel yang diamati, sehingga diharapkan dari sekumpulan data dengan target yang sama akan dihasilkan suatu generalisasi atau pencirian secara umum terhadap suatu target yang sejenis. Proses akuisisi data tersebut harus benar-benar akurat dengan mempertimbangkan semua karakteristik dari setiap sampel yang tidak lain adalah suatu citra

digital, dimana setiap citra digital memiliki karakteristik tertentu.

B.1. Karakteristik Sampel

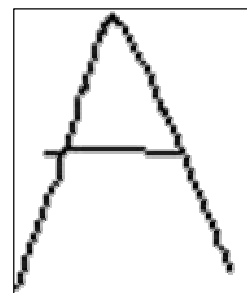
Sampel-sampel yang akan diamati tentunya harus dibatasi dengan suatu struktur dimensi dan homogenisasi pixel yang sederhana, sehingga diharapkan dapat mempermudah proses analisa terhadap konsep ini. Setiap sampel adalah satu citra digital yang nilai warna dari pixel-pixelnya terhomogenisasi menjadi dua representasi warna, yaitu warna aktif (hitam) dan warna nonaktif (selain hitam). Selain itu dimensi dari setiap sampel dibatasi sebesar area yang disediakan pada program aplikasi, tetapi tidak akan mengurangi fleksibilitas dan skalabilitas yang diharapkan.

B.2. Ekstraksi Data

Untuk mendapatkan data yang akurat dan konsisten dari setiap sampel, digunakan suatu metode sederhana yaitu dengan cara menghitung jumlah pixel aktif yang terdapat pada bagian-bagian dari sampel. Adapun algoritma umum dari pengekstrakan data numerik dari setiap sampel adalah sebagai berikut :

1. Setiap sampel yang diamati, dibagi menjadi beberapa area, misalnya 4 kolom dan 5 baris, sehingga akan terdapat 20 area pengamatan;
2. jumlah pixel yang aktif dari setiap area yang ada dihitung secara akurat;
3. dihasilkan sejumlah 20 data numerik dengan atribut kolom dan baris yang diharapkan dapat mewakili data ciri dari sampel yang diamati.

Setelah melalui tahapan normalisasi, data-data numerik tadi akan menjadi data input pada JSB. Dengan demikian jumlah area yang ada pada setiap sampel akan bersesuaian dengan jumlah neuron input JSB yang akan digunakan. Agar dapat dihasilkan kumpulan data yang seragam, maka setiap sampel yang akan diamati haruslah memiliki jumlah area pembagian



Gambar 3.1. Contoh sampel yang merepresentasikan huruf 'A' yang sama. Berikut contoh pengekstrakan data ciri dari sampel pada Gambar 3.1.

0	22	3	0
0	13	13	0
19	14	23	0
13	0	3	10
13	0	0	9

Gambar 3.2. Pembagian region pada sample berikut nilai pixel aktifnya

B.3. Normalisasi Data

Tentunya sampel yang dibuat tidak mungkin akan memiliki dimensi yang identik antara yang satu dengan yang lainnya. Hal ini merupakan faktor skalabilitas. Selain itu setiap sampel juga akan memiliki bentuk yang unik, dapatlah dikatakan tidak mungkin ada dua sampel yang bentuknya identik. Hal tersebut merupakan faktor fleksibilitas. Namun, untuk menjaga faktor skalabilitas dan fleksibilitas tersebut, data numerik yang dihasilkan pada proses sebelumnya akan menjadi tidak konsisten lagi. Untuk itu diperlukan satu proses tambahan untuk mempertahankan konsistensi data terhadap sampel pengamatan, yaitu bahwa setiap data numerik harus melalui suatu proses normalisasi data.

Metode normalisasi yang akan digunakan adalah juga suatu metoda normalisasi yang paling sederhana yaitu normalisasi perbandingan tetap. Dalam kasus ini adalah perbandingan antara jumlah pixel aktif pada masing-masing area dalam suatu sampel akan dibandingkan (dibagi) dengan jumlah pixel aktif terbanyak dalam satu area yang ada pada sampel yang sama. Hasil dari proses normalisasi ini adalah sekumpulan data numerik yang bernilai antara 0 (nol) dan 1 (satu).

Pada Gambar 3.2., nilai pixel tertinggi adalah 23, maka untuk menormalisasi sampel tersebut setiap nilai pixel aktif pada setiap region akan dibagi dengan 23. Berikut adalah contoh dari tabel normalisasi terhadap data numerik yang telah dihasilkan pada contoh sebelumnya.

Tabel 1 Normalisasi dari contoh

Region	Nilai pixel	
	Asli	Ternormalisasi
1	0	0
2	22	0.9565
3	3	0.1304
4	0	0
5	0	0
6	13	0.5652

7	13	0.5652
8	0	0
9	19	0.8261
10	14	0.6087
11	23	1
12	0	0
13	13	0.5652
14	0	0
15	3	0.1304
16	10	0.4348
17	13	0.5652
18	0	0
19	0	0
20	9	0.3913

B.4. Akumulasi Data Numerik

Setelah setiap data numerik pada masing-masing sampel dinormalisasi, maka kumpulan data tersebut akan diakumulasi ke dalam bentuk tabular. Sehingga akhirnya dapat dengan mudah diimplementasikan ke dalam JSB. Data akhir yang terakumulasi haruslah mencakup semua parameter input yang diperlukan oleh JSB dari setiap sampel pengamatan. Adapun data yang harus ada adalah target output dari setiap sampel serta kumpulan data numerik untuk tiap-tiap area pada masing-masing sampel.

C. Struktur Jaringan Syaraf Buatan

JSB hanyalah digunakan sebagai alat untuk menganalisa serta membuktikan konsep ekstraksi yang dibuat. Oleh karenanya bagian ini tentunya bukanlah materi pembahasan utama. Adapun jaringan syaraf yang digunakan memiliki batasan-batasan serta karakteristik tertentu yang disesuaikan secara empiris terhadap kebutuhan. Hal ini menyebabkan sifat dari jaringan syaraf yang digunakan cenderung bersifat *rigid* atau kaku. Tetapi walaupun demikian pemilihan karakteristik maupun pemberian parameter-parameter yang digunakan merupakan hasil beberapa kali percobaan yang dilakukan secara empiris.

Struktur JSB merupakan kumpulan neuron-neuron yang teratur sedemikian rupa sehingga membentuk susunan yang memiliki arti. Struktur JSB sebagaimana dapat dilihat [8]. Adapun ketentuan jumlah neuron pada tiap-tiap lapisan dapat dilihat pada tabel berikut :

Tabel 2. Distribusi Neuron

Lapisan	Jumlah neuron
Input	Tergantung jumlah region
Hidden	10
Output	6

JSB memiliki berbagai jenis algoritma yang dapat diimplementasikan. Disini digunakan algoritma *propagasi balik*. Fungsi aktivasi yang digunakan adalah fungsi aktivasi *sigmoid* ini. fungsi sigmoid memiliki karakteristik serta performa yang paling unggul disbandingkan dengan fungsi aktivasi lainnya.

Selain struktur, algoritma, serta fungsi aktivasi yang digunakan, masih ada parameter-parameter signifikan lainnya yang sangat mempengaruhi karakteris-

tik JSB secara keseluruhan. Nilai dari parameter-parameter tersebut akan sangat mempengaruhi hasil akhir dari JSB. Adapun parameter-parameter penentu lain tersebut antara lain : *Nilai awal, LearnRate (Laju Belajar), Momentum, Stopping Criteria Error, HardTrapping MatchLevel*

Selanjutnya, implementasi detail konsep pengenalan Citra Huruf dan Angka pada Program Aplikasi CharCognitron dapat dilihat di [8]

IV. ANALISIS DAN SIMULASI

A. Preparasi Sampel

Pentunya pembuatan sampel perlu diperhatikan dengan cermat guna memperoleh data yang konsisten terhadap nilai acuan target. Karakteristik sampel yang baik harus memenuhi kriteria yaitu bahwa antara sampel-sampel yang sejenis (sama nilai target acuannya; misal 'A') harus memiliki bentuk keterbacaan yang sama (*Human Readable Character*). Bentuk dari sampel-sampel tersebut harus konsisten dengan nilai acuannya sehingga diharapkan akan menghasilkan kumpulan sampel yang konsisten dan seragam. Jumlah sampel juga akan sangat mempengaruhi performansi dari JSB jika dilihat dari jumlah iterasi yang harus dilakukan pada setiap epoch-nya yang berbanding lurus dengan jumlah sampels. Untuk itu diperlukan optimalisasi jumlah sampel yang akan diuji sehingga bisa didapatkan komposisi yang baik dengan jumlah sampel seminimal mungkin, tetapi tidak mengurangi reliabilitas serta konsistensi kumpulan sampel tersebut.

Dikarenakan masalah-masalah tersebut bersifat empiris, maka dibuat suatu komposisi percobaan seperti terlihat pada Tabel 3., bahwa untuk membuktikan bahwa konsep pengenalan huruf dan angka yang digunakan dengan memberikan hasil yang baik, tidak perlu harus semua target acuan yang mungkin ('a'..'z'; 'A'..'Z'; '0'..'1') diujikan pada percobaan ini.

B. Simulasi Aktifitas Belajar

Percobaan-percobaan pertama yang dilakukan adalah analisa pengaruh nilai-nilai *LearnRates*, *Momentum*, dan terakhir adalah pengaruh jangkauan nilai awal. Untuk dua percobaan pertama digunakan jangkauan nilai awal -2 s/d 2.

Percobaan yang akan dilakukan akan diberi beberapa batasan sehingga dapat diperoleh hasil yang efektif. Adapun batasan-batasan tersebut antara lain adalah :

1. Sampel yang akan digunakan pada pembahasan dan analisa adalah *Sample1* dengan *Region = 5*4*;
2. Jumlah epoch maksimum adalah 1000;
3. Proses belajar dihentikan apabila sudah tercapai salah satu dari dua kriteria, yaitu *HardTrapping MatchLevel* atau *Stopping Criteria Error*;
4. Untuk setiap kasus dilakukan beberapa kali percobaan dan diambil satu yang terbaik untuk

ditampilkan dalam pembahasan, dan untuk dijadikan acuan sementara bagi proses belajar berikutnya;

5. Hasil terbaik dari percobaan ditentukan dari nilai *HardTrapping MatchLevel* terbesar dan atau nilai *Stopping Criteria Error* terkecil;
6. Percobaan dinyatakan berhasil apabila tercapai kondisi *MatchLevel=100%*, yang artinya bahwa JSB tersebut 'sudah dapat mengenali dirinya sendiri';
7. Tujuan utama dari percobaan ini adalah untuk mendapatkan karakteristik JSB yang terbaik sehingga JSB tersebut dapat 'mempelajari' pola yang diberikan dengan baik.

Tabel 3 Komposisi Sampel

Group Sample	Target Acuan	Jumlah Sampel	ASCII	Kode Target	
				Dec	Bin
Sample1 [100]	A	20	65	10	001010
	I	20	73	18	100010
	U	20	85	30	011110
	E	20	69	14	001110
	O	20	79	24	011000
Sample2 [100]	a	20	97	36	100100
	i	20	105	44	101100
	u	20	117	56	111000
	e	20	101	40	101000
	o	20	111	50	110010
Sample3 [60]	0	10	48	0	000000
	1	10	49	1	000001
	2	10	50	2	000010
	3	10	51	3	000011
	4	10	52	4	000100
	5	10	53	5	000101

Berikut akan ditampilkan hasil percobaan untuk sample 1 (sample lain dapat dilihat di [8])

B.1. Pengaruh Learnrate

Untuk melakukan analisa pengaruh *LearnRates* terhadap JSB, pertama kali harus ditetapkan nilai *Momentum = 0*, agar percobaan ini tidak terpengaruh oleh nilai *Momentum* tersebut. Percobaan ini dilakukan pada nilai-nilai *LearnRates* yang kritis dan cukup ekstrim yaitu 0,1; 0,2; 0,5; 0,9; 1. Walaupun nilai *LearnRates* sendiri adalah bilangan real yang lebih besar dari nol, tetapi nilai di atas satu sangatlah tidak dianjurkan, mengingat bahwa dengan semakin tingginya nilai *LearnRates*, maka karakteristik JSB menjadi semakin 'liar' (sangat berfluktuasi) dan tidak dapat diduga (*unpredictable*). Pada simulasi-simulasi berikut, hanya ditampilkan grafik *ErrorLevel* dan hasil yang dicapainya yaitu berupa nilai *ErrorLevel*, *MatchLevel*, dan nilai epoch terakhir.

Tabel 4. Hasil percobaan pengaruh *LearnRates*

Percobaan	LearnRates	Epoch	ErrorLevel	MatchLevel
Simulasi 1	0.1	1000	29.19698	43%
Simulasi 2	0.2	1000	45.42733	55%
Simulasi 3	0.5	1000	40.53675	54%
Simulasi 4	0.9	1000	79.49562	20%
Simulasi 5	1	1000	119.99389	0%

Dari ke lima simulasi tersebut dapat dibuat kesimpulan bahwa *LearnRates* terbaik dengan *ErrorLevel* paling rendah adalah 0,1 dan *LearnRates* terbaik berdasarkan *MatchLevel* tertinggi adalah 0,2. Tetapi dikarenakan tingkat keberhasilan dari percobaan ini tidak diukur berdasarkan rendahnya nilai *ErrorLevel*, melainkan dari tingginya nilai *MatchLevel*, maka Simulasi 2 dengan *LearnRates* = 0,2 yang akan digunakan pada percobaan berikutnya

B.2. Pengaruh Momentum

Percobaan berikut akan menggunakan nilai *LearnRates* terbaik yang didapat dari percobaan sebelumnya dan menggunakan nilai *Momentum* yang berbeda (0,1;0,2;0,5;0,8;0,9;1) untuk dianalisa hasil terbaik berdasarkan nilai *Momentum* tersebut. Adapun parameter lainnya seperti jangkauan nilai awal, masih sama dengan percobaan sebelumnya. Berikut hasil beberapa simulasi yang dilakukan terhadap perubahan nilai *Momentum*.

Dari simulasi-simulasi di atas dapat diperoleh kesimpulan yaitu bahwa JSB memiliki kecenderungan belajar yang lebih baik dengan pemilihan jangkauan simetris dan sebaliknya untuk asimetris. Dari Simulasi 7 dan Simulasi 8 terlihat bahwa dengan magnitasi lebih kecil (-2 s/d +2), JSB juga memiliki kecenderungan belajar yang lebih baik. Untuk itu pada percobaan berikutnya akan dipergunakan jangkauan nilai tersebut.

Tabel 5. Hasil percobaan pengaruh *Momentum*

Percobaan	<i>Momentum</i>	Epoch	<i>ErrorLevel</i>	<i>MatchLevel</i>
Simulasi 1	0.1	1000	25.48247	80%
Simulasi 2	0.2	1000	47.16077	45%
Simulasi 3	0.5	215	12.87525	100%
Simulasi 4	0.8	1000	30.6719	65%
Simulasi 5	0.9	1000	27.94824	67%
Simulasi 6	1	1000	50.2894	52%

Dari enam simulasi di atas dapat disimpulkan bahwa nilai *Momentum* terbaik diperoleh pada Simulasi 3 dengan nilai *Momentum* = 0,5. Pada Simulasi 3 tersebut diperoleh nilai keberhasilan *MatchLevel* = 100% pada epoch 215 yang sangat menunjukkan bahwa JSB dapat belajar dengan baik pada kondisi ini. Oleh karenanya kombinasi nilai *LearnRates* dan *Momentum* inilah yang untuk sementara dapat dijadikan acuan terbaik untuk analisa berikutnya.

B.3. Pengaruh Jangkauan Nilai Awal

Berikutnya adalah percobaan untuk menganalisa pengaruh jangkauan nilai awal pada JSB. Pada percobaan ini akan digunakan JSB dengan karakteristik terbaik yang didapatkan dari percobaan sebelumnya yaitu nilai *LearnRates* = 0,2 dan nilai *Momentum* = 0,5. Berikut hasil beberapa simulasi terhadap perbedaan jangkauan nilai awal.

Tabel 4.4. Hasil percobaan pengaruh Jangkauan

Percobaan	Jangkauan	Epoch	<i>ErrorLevel</i>	<i>MatchLevel</i>
Sim1(A)	0 s/d +2	1000	119,99886	0 %
Sim2(A)	0 s/d +5	1000	239,99459	0 %

Sim3(A)	-5 s/d 0	1000	30,69999	74 %
Sim4(A)	-2 s/d 0	1000	42,82592	41 %
Sim5(A)	-5 s/d 2	1000	35,93632	59 %
Sim6(A)	-2 s/d +5	1000	139,87458	20 %
Sim7(S)	-2 s/d +2	215	12,87525	100 %
Sim8(S)	-5 s/d +5	100	13,38231	91 %

B.4. Pengaruh Komposisi Region pada Sampel JSB

Dari semua percobaan di atas telah didapat suatu karakteristik JSB yang baik dengan parameter-parameter nilai *LearnRates* = 0,2; *Momentum* = 0,5 dan jangkauan nilai awal = -2 s/d +2. Percobaan-percobaan tersebut sudah dapat cukup membuktikan bahwa metode pengenalan pola huruf dan angka tulisan tangan yang dibuat oleh penulis dapat 'dipelajari' dengan baik oleh JSB. Hal tersebut ditandai dengan tercapainya nilai *MatchLevel* = 100% dengan karakteristik yang dimaksud. Namun untuk lebih melengkapi percobaan dan analisa terhadap metode ini, maka penulis melakukan satu lagi percobaan yaitu pengaruh komposisi *Region* terhadap karakteristik JSB dan hasil akhir yang akan didapat. Hasil dari beberapa simulasi yang dilakukan dapat dilihat pada tabel 6.

Dari ke enam simulasi tersebut dapat dilihat bahwa Simulasi 1 memberikan hasil belajar paling baik. Oleh karena itu dapat disimpulkan sementara bahwa komposisi *Region* 5*4 adalah yang terbaik.

Tabel 6. Hasil percobaan pengaruh komposisi *Region*

Percobaan	<i>Region</i>	Epoch	<i>ErrorLevel</i>	<i>MatchLevel</i>
Simulasi 1	5*4	215	12.87525	100%
Simulasi 2	4*5	1000	40.38083	58%
Simulasi 3	4*4	1000	38.39191	45%
Simulasi 4	3*3	1000	40.23196	59%
Simulasi 5	4*3	1000	36.55019	53%
Simulasi 6	3*4	1000	26.20377	78%

V. KESIMPULAN

Kesimpulan utama yang dapat diambil yaitu bahwa metode pengenalan citra huruf dan angka tulisan tangan yang dibuat dapat diimplementasikan dengan baik terhadap suatu jaringan syaraf buatan tertentu dan telah dianalisa dengan menggunakan program aplikasi CharCognitron yang juga telah dibuat sebelumnya.

Secara teknis, ada beberapa hasil analisa yang dapat disimpulkan, diantaranya bahwa

- (i) nilai *LearnRates* terbaik untuk metode ini adalah 0,2;
- (ii) nilai *Momentum* terbaik untuk metode ini adalah 0,5;
- (iii) jangkauan nilai terbaik adalah -0,2 s/d +0,2;
- (iv) komposisi *Region* terbaik adalah 5*4;

Secara umum hal lain yang dapat disimpulkan dari analisa yang dilakukan, antara lain :

- (i) Pemilihan sampel yang baik mutlak harus dilakukan karena akan sangat mempengaruhi hasil akhir;
- (ii) Untuk mendapatkan hasil yang terbaik dengan menggunakan JSB, harus dilakukan *performance tuning* terlebih dahulu terhadap JSB tersebut, sehingga didapatkan karakteristik khas untuk kasus ini;
- (iii) Metode yang dibuat oleh penulis telah terbukti dapat

‘dipelajari’ (*learnable*) dan dikenali pola-polanya dengan baik oleh JSB;

Daftar Pustaka

- [1] Alex Berson and Stephen J., **Data Warehousing, Data Mining and Olap**, McGraw Hill, USA, 1997
- [2] Irwan Arifin, **Perancangan Modul Program dan Praktikum Jaringan Neural Buatan**, Gunadarma, Jakarta, 1999
- [3] Simon Haykin, **Neural Networks : a Comprehensif Foundation**, Prentice Hall International Inc., New Jersey, 1999
- [4] Kemal Ade Sekarwati, **Jaringan Syaraf Tiruan**, <http://www.gunadarma.ac.id>, 1998
- [5] Braun Feulner Malaka, **Praktikum Neural Netze**, Spinger-Verlag, Berlin, 1996
- [6] Igor Aleksander and Helen Morton, **An Introduction to Neural Computing**, Chapman and Hall, London, 1990
- [7] Schneider, G. Michael and Bruell, Steven C., **Advanced Programming and Problem Solving with Pascal**, John Wiley and Sons, New York, 1987
- [8] A.P. Nugraha, Skripsi, Gunadarma, 2002