

PENGEMBANGAN APLIKASI TEXT RECOGNITION DENGAN KLASIFIKASI NEURAL NETWORK PADA HURUF HIJAIYAH GUNDUL

M. Anif¹, Safitri Juanita², Ika Disja Afriyani³

¹⁾³⁾Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur

²⁾Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Budi Luhur

Jl. Raya Ciledug, Petungkang Utara, Kebayoran Lama, Jakarta Selatan 12260

Telp. (021) 5853753, Fax. (021) 5866369

e-mail : muhammad.anif@budiluhur.ac.id¹

ABSTRACT

Technological character recognition or OCR (Optical Character Recognition) have had fairly rapid growth and many have been used. Although the text recognition programs are widely available, but the majority of these programs can only recognize the Latin alphabet. How to recognize Arabic characters are converted to text alphabet is still very rare. Though the existence of Arabic text recognition has many advantages, for example, for people who do not recognize the Arabic script, it can be used as an alternative way to learn. This text recognition application using Artificial Neural Network learning algorithms (ANN) which is backpropagation algorithm an effective character classification which described in this study. In this study, the application will be developed to recognize the character recognition of Arabic characters Hijaiyah bare particular letters and Arabic numerals. Algorithms presented in the program as a tangible example of OCR (Optical Character Recognition) will be implemented in the Java programming language (Java Programming Language). With a certain level of accuracy in operation, can be obtained results (output) character appropriate to the class learning formed through the "OCR Train". Once fully developed, the application will then be tested by establishing Hijaiyah characters and Arabic numerals which are stored as image / image. Then open the file with the appropriate type of artificial neural expected to recognize the text in images of text characters / images are tested.

Keywords: *Optical Character Recognition, Text Recognition, Java Language Programming, Artificial Neural Network, Backpropagation, Arab.*

1. PENDAHULUAN

1.1. Latar Belakang

Pengolahan citra digital sekarang ini sudah meliputi teknik pengenalan karakter seperti karakter alfanumerik, karakter tulisan tangan, karakter huruf kanji, dan lain-lain. Teknik pengenalan karakter ini secara umum dikenal dengan teknologi OCR (*Optical Character Recognition*). Namun perkembangan bidang pengenalan karakter huruf arab belum sepesat bidang pengenalan karakter yang lain. Hal ini dikarenakan pengenalan karakter Arab memiliki tingkat kesulitan yang lebih tinggi

dibandingkan dengan pengenalan karakter lainnya. Pada penelitian ini penulis akan mengembangkan sebuah aplikasi komputer *desktop* yang bisa mengenali karakter huruf dan angka dalam karakter Arab menggunakan jaringan saraf tiruan. Dengan menambahkan beberapa *learning class* tentang karakter huruf dan angka Arab didalamnya, yang diharapkan bisa dijadikan salah satu alternatif pembelajaran karakter Arab dengan media komputer.

1.2. Masalah

Saat ini masih sedikit sekali penyediaan aplikasi pembelajaran

karakter Arab dikarenakan adanya masalah pada:

- 1) Sulitnya cara membaca huruf Hijaiyah gundul dan angka Arab.
- 2) Masih sedikit aplikasi *text recognition* khususnya karakter Arab.
- 3) Berapa banyak karakter yang dapat dibaca dalam satu *image*.
- 4) Persentase kegagalan dan kesuksesan program.

1.3. Tujuan

Tujuan penelitian ini adalah untuk mempelajari dan mengembangkan sebuah aplikasi perangkat lunak yang mampu membaca huruf Hijaiyah dan angka Arab dengan dengan klasifikasi *neural network* berbasis *desktop*. Aplikasi tersebut dikembangkan dengan bahasa pemrograman Java yang pada dasarnya *multiplatform*, jadi bisa diaplikasikan di berbagai *platform* komputer.

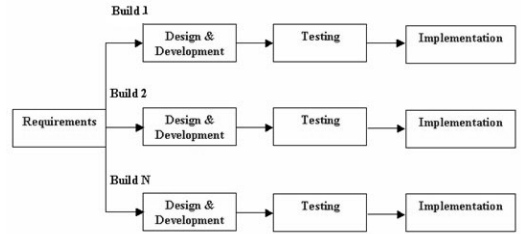
1.4. Ruang Lingkup dan Batasan Masalah

Untuk menyederhanakan pembahasan pada penelitian ini, masalah dibatasi sebagai berikut:

- 1) Melakukan penelitian terhadap huruf Hijaiyah dan angka Arab.
- 2) Hanya dapat mengenali huruf Hijaiyah tunggal gundul dan angka Arab.
- 3) Menghitung jumlah karakter dalam satu *image* maksimal satu karakter.

1.5. Metoda Perancangan

Dalam penyelesaian penelitian ini tahapan-tahapan yang dilakukan dalam penyelesaian aplikasi adalah dengan mengadopsi metode *incremental*.



Gambar 1 : Siklus Model *Incremental*

Berikut adalah penjelasan tahapannya:

- 1) Mengkombinasikan elemen-elemen dari *waterfall* dengan sifat iterasi/perulangan.
- 2) Elemen-elemen dalam *waterfall* dikerjakan dengan hasil berupa produk dengan spesifikasi tertentu, kemudian proses dimulai dari fase pertama hingga akhir dan menghasilkan produk dengan spesifikasi yang lebih lengkap dari yang sebelumnya. Demikian seterusnya hingga semua spesifikasi memenuhi kebutuhan yang ditetapkan oleh pengguna.
- 3) Produk hasil *increment* pertama biasanya produk inti (*core product*), yaitu produk yang memenuhi kebutuhan dasar. Produk tersebut digunakan oleh pengguna atau menjalani *review*/pengecekan detil. Hasil *review* tersebut menjadi bekal untuk pembangunan pada *increment* berikutnya. Hal ini terus dikerjakan sampai produk yang komplit dihasilkan.
- 4) Model *increment* ini cocok untuk tim pengembang/pembangun perangkat lunak yang memiliki anggota sedikit/tidak terlalu banyak.
- 5) Model ini juga mampu mengakomodasi perubahan secara fleksibel.
- 6) Produk yang dihasilkan pada *increment* pertama bukanlah *prototype*, tapi produk yang sudah bisa berfungsi dengan spesifikasi dasar.

2. LANDASAN TEORI

a. OCR

Suatu kerja OCR terdiri dari modul terpisah yang digunakan dalam proses pengenalan, modul-modul tersebut adalah:

1) Pra Pengolahan (*Preprocessing*)

Merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak diperlukan pada gambar *input* untuk proses selanjutnya. Contoh *preprocessing* adalah *noise filtering*.

2) Klasterisasi (*Clusterization*)

Modul klasterisasi bertugas untuk menemukan karakter individu dalam gambar dan menghitung urutan karakter yang benar dalam teks.

3) Ekstraksi Fitur (*Feature extraction*)

Tujuan dari modul ini adalah untuk mengekstrak fitur unik dari karakter individu sehingga dapat diakui oleh modul klasifikasi. Jika kita membandingkan sistem OCR dengan manusia, kita dapat mengatakan bahwa modul ini memainkan peran mata manusia.

4) Klasifikasi (*Classification*)

Pada tahap terakhir sistem OCR mencoba untuk mengenali karakter menggunakan informasi tentang fitur karakter yang telah diekstraksi pada tahap sebelumnya. Meskipun dapat dilakukan secara statistik dengan melibatkan perbandingan konsisten karakter dari *database*, hal ini dipilih untuk mencoba bagaimana jaringan saraf melakukan pengenalan pada tahap ini.

b. Algoritma *Backpropagation*

Menurut F. Suhandi (2009), *Backpropagation* merupakan sebuah metode sistematis pada jaringan saraf tiruan dengan menggunakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak layer untuk mengubah bobot-bobot yang ada pada layer tersembunyinya. *Backpropagation* adalah pelatihan jenis terkontrol dimana

menggunakan pola penyesuaian bobot untuk mencapai nilai kesalahan yang minimum antara keluaran hasil prediksi dengan keluaran yang nyata.

Untuk mendapatkan *error* jaringan ini, fase *forwardpropagation* harus dilakukan sebelumnya. Saat propagasi dalam arah maju, neuron diaktivasi dengan menggunakan fungsi aktivasi sigmoid.

Rumus aktivasi sigmoid adalah:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Algoritma bekerja sebagai berikut:

- 1) Lakukan fase *forwardpropagation* untuk pola *input* dan hitung *error output*.
- 2) Mengaktivasi neuron dari layer tersembunyi.
- 3) Mengaktivasi neuron dilayer.
- 4) Menghitung nilai *error*.
- 5) Ubah semua nilai bobot dari matriks bobot dengan menggunakan rumus: $\text{bobot}_{lama} + \text{learning rate} * \text{error output} * \text{output}(\text{neuron}_i) * \text{output}(\text{neuron}_{i+1}) * (1 - \text{output}(\text{neuron}_{i+1}))$
- 6) Kembali ke langkah 1 hingga nilai *error* semakin lama semakin kecil.
- 7) Algoritma berakhir, jika semua pola *output* cocok dengan pola tujuan dan jika *error* jaringan nol (sempurna) atau mendekati nol.

c. Jaringan Saraf Tiruan

Menurut Haykin, S. (1994) mendefinisikan Jaringan Saraf Tiruan sebagai berikut: "Sebuah jaringan saraf adalah sebuah prosesor yang terdistribusi *parallel* dan mempunyai kecenderungan untuk menyimpan pengetahuan yang didapatkannya dari pengalaman dan membuatnya tetap tersedia untuk digunakan. Hal ini menyerupai kerja otak dalam dua hal yaitu:

- 1) Pengetahuan diperoleh oleh jaringan melalui suatu proses belajar.

- 2) Kekuatan hubungan antar sel saraf yang dikenal dengan bobot sinapsis digunakan untuk menyimpan pengetahuan”.

Menurut Zurada, J.M. (1992) mendefinisikan sebagai berikut: “Sistem saraf tiruan atau Jaringan saraf tiruan adalah sistem selular fisik yang dapat memperoleh, menyimpan dan menggunakan pengetahuan yang didapatkan dari pengalaman”.

Encyclopedia Britannica menulis bahwa: “Jaringan saraf tiruan adalah paradigma pemrosesan suatu informasi yang terinsiprasi oleh sistem sel saraf biologi, sama seperti otak yang memproses suatu informasi. Elemen mendasar dari paradigma tersebut adalah struktur yang baru dari sistem pemrosesan informasi. Jaringan saraf tiruan, seperti manusia, belajar dari suatu contoh. Jaringan saraf tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses

3. ANALISA MASALAH DAN RANCANGAN PROGRAM

a. Identifikasi Masalah

Meskipun sudah banyak aplikasi OCR yang berkembang, namun kebanyakan hanya bisa membaca karakter huruf latin/*alphabet*, karakter alfanumerik, karakter tulisan tangan, karakter huruf kanji, dan lain-lain. Masih jarang aplikasi penunjang penulisan bahasa asing seperti karakter Arab, Jepang, bahkan Aksara Jawa sekalipun. Padahal bahasa asing tersebut sudah mulai banyak digunakan atau sudah sering dipakai sebagai bahasa internasional.

Setelah aplikasi *text recognition* ini dikembangkan, aplikasi ini harus bisa membaca karakter Arab baik berbentuk huruf Hijaiyah maupun angka Arab. Sehingga *user* dapat melakukan *input* gambar yang berisi karakter Arab dengan mudah.

Karena pengembangan aplikasi *text recognition* ini bersifat penelitian, maka dibutuhkan juga pengujian terhadap aplikasi ini yang bertujuan untuk menentukan kemampuan maksimal dan kekurangan apa saja yang berpotensi untuk dikembangkan.

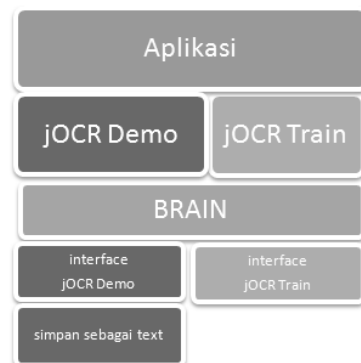
b. Penyelesaian Masalah

Penulis memilih mengembangkan aplikasi ini dengan menggunakan metode *incremental* agar dapat dilakukan dengan cepat, mudah dan akan lebih efisien dalam menambah atau mengurangi fitur dari aplikasi inti. Selain itu aplikasi *text recognition* ini bersifat *open source* dengan menggunakan bahasa pemrograman Java berbasis *desktop*, jadi dapat diimplementasikan pada berbagai jenis *platform operating system*.

c. Program Aplikasi

1) Analisa Aplikasi Usulan

Program aplikasi yang diusulkan adalah sebuah aplikasi *client* berbasis *desktop* yang dengan menggunakan Java Programming Language. Aplikasi *desktop* ini merupakan aplikasi utama *client* yang dikerjakan oleh NetBeans. Setelah aplikasi ini *dicompile* dan dijalankan, maka antar muka aplikasi *text recognition* ini dapat ditampilkan dan digunakan. Berikut adalah bagan dari aplikasi *text recognition* ini:



Gambar 2 : Bagan Aplikasi jOCR Demo dan jOCR Train

2) Metoda Kerja Usulan

a) Konsep Latih OCR

Aplikasi *text recognition* ini merupakan aplikasi yang membutuhkan proses *training* karakter Arab sebelum dapat digunakan pada menu Demo OCR untuk proses pengenalan karakter. Pada menu Latih OCR, pertama kali kita diminta untuk melakukan *training*/pelatihan terhadap brain yang berekstensi .brn agar bisa mengenali karakter. Oleh karena itu penting untuk memberikan pelatihan terhadap karakter gambar dengan huruf ا-ي dan angka Arab 0-9 berekstensi .bmp agar memperoleh hasil maksimal, lalu disimpan pada saraf buatan sebagai *file*.

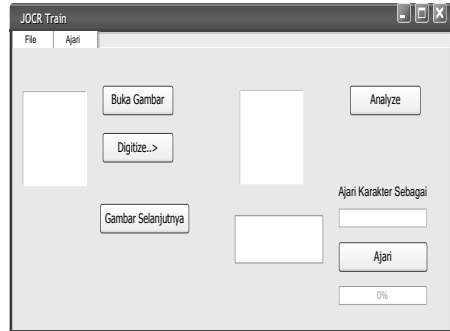
b) Konsep Demo OCR

Aplikasi *text recognition* ini memiliki menu Demo OCR untuk mengimplementasikan hasil saraf buatan dari menu Latih OCR. Pada menu ini, saraf buatan yang sudah disimpan pada direktori folder *brain* harus dipanggil. Lalu *file* karakter teks yang berupa format gambar .gif dipanggil, dan selanjutnya memilih menu “Kenali” untuk menguji aplikasi *text recognition* ini dalam mengenali karakter teks yang sudah dilatih.

c) Rancangan Layar

1) Form JOCR Train

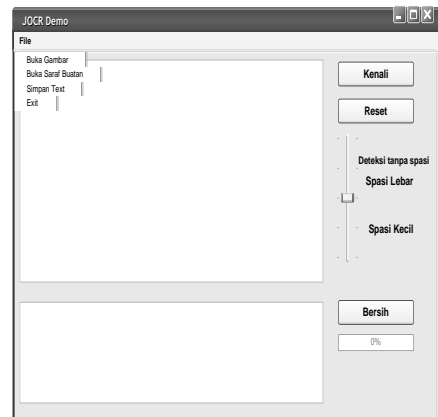
Form JOCR Train merupakan *Form* yang digunakan untuk membuat saraf buatan yang nantinya saraf buatan tersebut akan digunakan dalam proses pengenalan/demo. Dan di *Form* ini pula *user* dapat melatih/*training* data agar dapat dikenali pada saat proses demo berlangsung. Berikut adalah rancangan layarnya:



Gambar 3 : Rancangan Layar OCR Train

2) Form JOCR Demo

Form JOCR Demo merupakan *Form* yang digunakan untuk memproses/mengeksekusi *image* menjadi *text* dengan menggunakan saraf buatan yang telah dibuat dan disimpan di *Form JOCR Train*. Terdapat 4 *Sub Menu*, yaitu Buka Gambar, Buka Saraf Buatan, Simpan Text, dan Exit. Gambar 4 dibawah ini adalah rancangan layarnya:



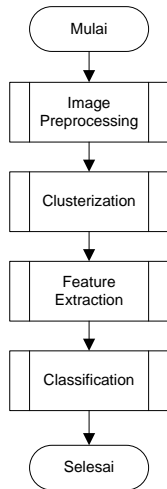
Gambar 4 : Rancangan Layar Menu File pada Form JOCR Demo

d) Alur Proses (Flowchart)

1) Alur Proses Keseluruhan

Alur proses berikut ini merupakan alur proses program dari pertama kali aplikasi dijalankan sampai mendapatkan hasil

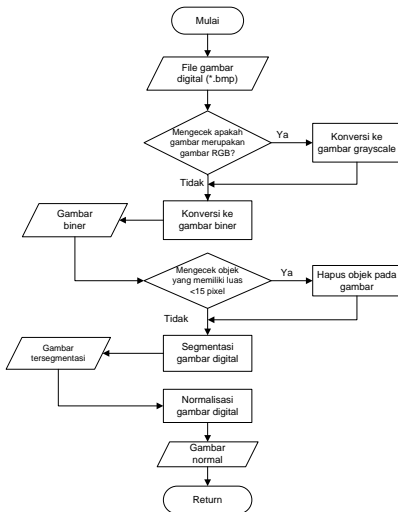
keluaran/output. Gambar 5 adalah alur proses program keseluruhan:



Gambar 5 : Alur Proses Program Keseluruhan

2) Alur Proses *Image Preprocessing*

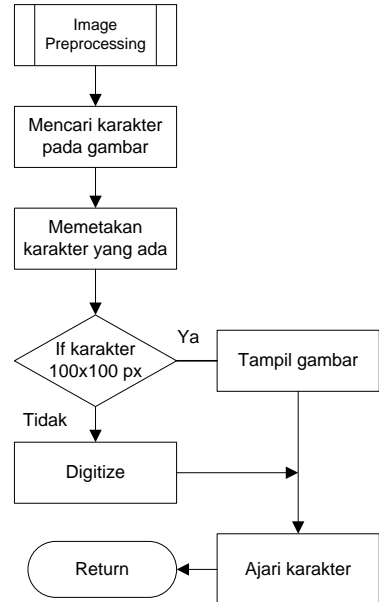
Modul ini merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak diperlukan pada gambar *input* untuk melakukan proses selanjutnya. Gambar 6 adalah alur proses *image preprocessing*:



Gambar 6 : Alur Proses *Image Preprocessing*

3) Alur Proses *Clusterization*

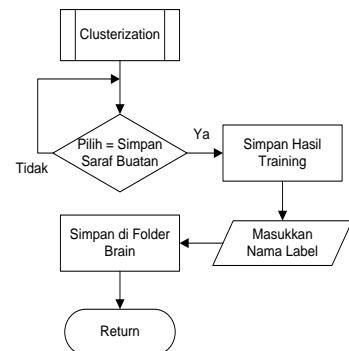
Modul ini bertugas untuk menemukan karakter individu dalam gambar dan menghitung urutan karakter yang benar dalam teks. Gambar 7 adalah alur proses *clusterization*:



Gambar 7 : Alur Proses *Clusterization*

4) Alur Proses *Feature Extraction*

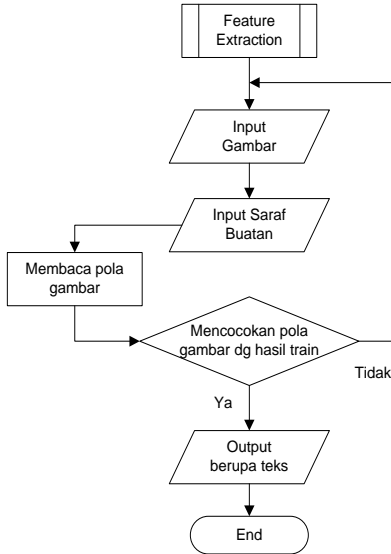
Modul ini bertujuan untuk mengekstrak fitur unik dari karakter individu sehingga dapat diakui oleh modul klasifikasi. Gambar 8 adalah alur proses *feature extraction*:



Gambar 8 : Alur Proses *Feature Extraction*

5) Alur Proses Classification

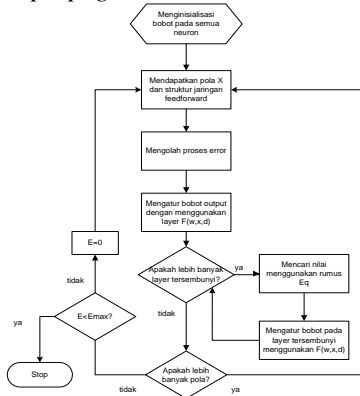
Ini merupakan modul terakhir pada sistem OCR yang mencoba mengenali karakter menggunakan informasi tentang fitur karakter yang telah diekstraksi pada tahap sebelumnya. Gambar 9 adalah alur proses *classification*:



Gambar 9 : Alur Proses Classification

6) Alur Proses Algoritma Backpropagation

Aplikasi *text recognition* ini menggunakan algoritma *backpropagation*. Gambar 10 adalah alur proses algoritma *backpropagation*:



Gambar 10 : Alur Proses Algoritma Backpropagation

d. Algoritma

1) Algoritma Proses Keseluruhan

Berikut adalah algoritma proses program keseluruhan saat aplikasi dijalankan:

1. Tampilkan Layar Awal Aplikasi
2. Lakukan Image Preprocessing Then
3. Lakukan Clusterization Then
4. Lakukan Feature Extraction Then
5. Lakukan Classification
6. End

2) Algoritma Image Preprocessing

Berikut adalah algoritma proses *image preprocessing*:

1. Tampilkan file gambar (*.bmp)
2. If gambar merupakan RGB Then
3. Konversi ke Grayscale
4. Else
5. Konversi ke Biner
6. Tampilkan Gambar Biner
7. If Objek < 15 pixel Then
8. Hapus Objek
9. Else
10. Segmentasi Gambar
11. Tampilkan Gambar Tersegmentasi
12. Lakukan Normalisasi Then
13. Tampilkan Gambar Normal
14. EndIf
15. EndIf
16. Return

3) Algoritma Clusterization

Berikut adalah algoritma proses *clusterization*:

1. Lakukan *Image Preprocessing* Then
2. Mencari Karakter pada Gambar Then
3. Memetakan Karakter yang ada
4. Input Pilih
5. If Pilih = Karakter 100x100 px Then
6. Tampilkan Gambar
7. Ajari Karakter
8. Return
9. Else
10. Digitize Then
11. Ajari Karakter
12. Return
13. EndIf

4) **Algoritma Feature Extraction**

Berikut adalah algoritma proses *feature extraction*:

1. Lakukan *Clusterization* Then
2. Input Pilih
3. If Pilih = Simpan Saraf Buatan Then
4. Simpan Hasil Training Then
5. Masukkan Nama Label Then
6. Simpan di Folder Brain Then
7. Return
8. Else
9. Kembali ke Baris 2
10. End

5) **Algoritma Classification**

Berikut adalah algoritma proses *classification*:

1. Lakukan *Feature Extraction*
2. Input Gambar Then
3. Input Saraf Buatan
4. Membaca Pola Gambar Then
5. If Pola Gambar = Hasil Training Then
6. Output Berupa Teks
7. Else
8. Kembali ke Baris 2
9. End

6) **Algoritma Backpropagation**

Berikut adalah keadaan saat algoritma *backpropagation* berjalan:

1. Menginisialisasi Bobot Pada Semua Neuron
2. Mendapatkan Pola X dan Struktur Jaringan Feedforward Then
3. Mengolah Proses Error Then
4. Mengatur Bobot Output
5. If Banyak Layer yang Tersembunyi Then
6. Mencari Nilai Menggunakan Rumus Then
7. Mengatur Bobot pada Layer Tersembunyi Then
8. Kembali ke Baris 5
9. ElseIf Pola > 1 Then
10. If Ya = Kembali ke Baris 2
11. Else
12. Cek $E < E_{max}$? Then
13. If Ya = Stop
14. Else $E=0$ Then Kembali ke Baris 2
15. EndIf
16. EndIf
17. EndIf

4. KESIMPULAN

Sesuai dengan pembahasan mengenai pengembangan aplikasi *text recognition* dengan klasifikasi *neural network* pada huruf Hijaiyah gundul, maka kesimpulan yang dapat diambil adalah sebagai berikut:

- 1) Hasil *training* dari aplikasi ini belum sempurna, karena tidak dapat mengenali huruf Hijaiyah secara baik.
- 2) Maksimal karakter yang mampu dibaca oleh aplikasi adalah 128 karakter (karakter latin), berdasarkan *code pages ASCII*.
- 3) Aplikasi *text recognition* ini dapat mengenali angka Arab, yaitu 0-9 dengan persentase kesuksesan hingga 90%.
- 4) Inputan karakter untuk ajari di *form train* hanya dapat mengajari 1 karakter saja.

Beberapa saran yang dapat diberikan berkaitan dengan pembuatan sistem ini untuk menambah, memperbaiki serta meningkatkan kualitas dari aplikasi *text recognition* ini antara lain yaitu :

- 1) Aplikasi ini perlu pengembangan lebih lanjut agar bisa membaca karakter Arab, baik huruf Hijaiyah maupun angka Arab.
- 2) Antar muka pemakai (*user interface*) lebih disempurnakan lagi agar tampak lebih menarik dan mempermudah dalam pemakaiannya.
- 3) Dengan pengembangan lebih lanjut, aplikasi *text recognition* ini akan bisa membaca lebih dari 128 karakter.
- 4) Dengan pengembangan lebih lanjut, dalam proses training diharapkan dapat mengajari lebih dari 1 karakter.

DAFTAR PUSTAKA

- [1]. Ahmad Fuad, Effendi., 2005, *Metodologi Pengajaran Bahasa Arab*, Malang: Misykat.
- [2]. Darmawan, Arik., 2009, *jOCR (online)*, Oktober 5, 2012. Blogspot. <http://sagitarius-arik.blogspot.com/2009/12/membuat-deteksi-karakter-dengan-gambar.html>
- [3]. Eyal Reinhold, Johnathan Nightingale, 1999, *Artificial Intelligence Tutorial Review (online)*, Oktober 18, 2012. University Of Toronto.
- [4]. <http://www.psych.utoronto.ca/~reingold/courses/ai/>
- [5]. F. Suhandi Krisna, 2009, *Prediksi Harga Saham dengan Pendekatan Artificial Neural Network menggunakan Algoritma Backpropagation*.
- [6]. Haykin, S., 1994, *Neural Networks: A Comprehensive Foundation*, NY, Macmillan.
- [7]. Haykin, S., 2009, *Neural Networks and Learning Machines*, Upper Saddle River, NJ: Pearson Education, Inc.
- [8]. Hecht-Nielsen Robert, 1989. *Theory of the Backpropagation Neural Network (online)*, November 20, 2012. HNC, Inc. - University of California at San Diego. <http://s112088960.onlinehome.us/anProjects/Research%20Paper%20Library/backPropTheory.pdf>
- [9]. *IEEE, 1998. Optical Font Recognition Using Typographical Features (online)*, November 14, 2012. *IEEE*.
- [10]. <http://www.cs.drexel.edu/classes/CS/cs680-502/proj4/fontrecog.pdf>
- [11]. Kirillov, Andrew, 2005, *Neural Network OCR (online)*, Oktober 18, 2012. Codeproject.
- [12]. http://www.codeproject.com/csharp/neural_network_ocr.asp
- [13]. Kusumadewi, Sri , 2004, *Membangun Jaringan Syaraf Tiruan*, Yogyakarta: Graha Ilmu.
- [14]. Setia, Venuu, 2007, *Neural Network (online)*, Oktober 18, 2012. Britannica, Encyclopædia.
- [15]. <http://www.britannica.com/EBchecked/topic/410549/neural-network>
- [16]. *SoftDevTeam, Incremental Lifecycle Model (online)*, Oktober 16, 2012. *SoftDevTeam*. <http://www.softdevteam.com/Incremental-lifecycle.asp>
- [17]. SourceForge open-source software development portal <http://www.sourceforge.net>
- [18]. Zurada, J.M., 1992, *Introduction To Artificial Neural Systems*, Boston: PWS Publishing Company.