

**TERAPAN JARINGAN SYARAF TIRUAN PERCEPTRON  
UNTUK PENGENALAN KARAKTER PADA GAMBAR**

**SKRIPSI**

**Oleh:  
TAUFIK HIDAYATULLAH  
NIM. 06510014**



**JURUSAN MATEMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2012**

**TERAPAN JARINGAN SYARAF TIRUAN PERCEPTRON  
UNTUK PENGENALAN KARAKTER PADA GAMBAR**

**SKRIPSI**

Diajukan kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Sains (S.Si)

Oleh:  
**TAUFIK HIDAYATULLAH**  
NIM. 06510014

**JURUSAN MATEMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2012**

**TERAPAN JARINGAN SYARAF TIRUAN PERCEPTRON  
UNTUK PENGENALAN KARAKTER PADA GAMBAR**

**SKRIPSI**

Oleh:  
**TAUFIK HIDAYATULLAH**  
NIM. 06510014

Telah Diperiksa dan Disetujui untuk Diuji:  
Tanggal: 13 Januari 2012

Pembimbing I

Pembimbing II,

Mohammad Jamhuri, M.Si  
NIP. 19810502 200501 1 004

Ach. Nashichudin, M.A  
NIP. 19730705 200003 1 001

Mengetahui,  
Ketua Jurusan Matematika

Abdussakir, M.Pd  
NIP. 19751006 200312 1 001

**TERAPAN JARINGAN SYARAF TIRUAN PERCEPTRON  
UNTUK PENGENALAN KARAKTER PADA GAMBAR**

**SKRIPSI**

Oleh:  
**TAUFIK HIDAYATULLAH**  
NIM. 06510014

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan  
Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
untuk Memperoleh Gelar Sarjana Sains (S.Si)

Tanggal : 21 Januari 2012

**Susunan Dewan Penguji**

**Tanda Tangan**

|                           |                                                        |                          |
|---------------------------|--------------------------------------------------------|--------------------------|
| <b>Penguji Utama</b>      | : Muhammad Faizal, MT.<br>NIP. 19740510 200501 1 007   | (                      ) |
| <b>Ketua Penguji</b>      | : Abdussakir, M.Pd<br>NIP. 19751006 200312 1 001       | (                      ) |
| <b>Sekretaris Penguji</b> | : Mohammad Jamhuri, M.Si<br>NIP. 19810502 200501 1 004 | (                      ) |
| <b>Anggota Penguji</b>    | : Ach. Nashichudin, M.A<br>NIP. 19730705 200003 1 001  | (                      ) |

Mengetahui dan Mengesahkan,  
Ketua Jurusan Matematika

Abdussakir, M.Pd  
NIP. 19751006 200312 1 001

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Taufik Hidayatullah  
NIM : 06510014  
Jurusan : Matematika  
Fakultas : Sains dan Teknologi

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 13 Januari 2012  
Yang membuat pernyataan,

Taufik Hidayatullah  
NIM. 06510014

## MOTTO

*Allah dulu, Allah lagi, Allah terus...*

*(Ust. Yusuf Mansur)*

*Apapun yang kita lakukan sekarang (baik atau buruk),*

*suatu saat pasti ada pengaruhnya.*

## HALAMAN PERSEMBAHAN



*Skripsi ini penulis persembahkan teruntuk:  
Kedua orang tua tercinta  
Djoko Suwijono dan Halimatus Sa'diyah  
&  
Adik tercinta  
Masagung Hariyandi  
yang tiada henti selalu memberikan dukungan dan do'a*

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT atas rahmat, taufik serta hidayah-Nya, sehingga penulis mampu menyelesaikan penyusunan skripsi ini sebagai salah satu syarat untuk memperoleh gelar sarjana dalam bidang matematika di Fakultas Sains dan Teknologi, Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.

Dalam proses penyusunan skripsi ini, penulis banyak mendapat bimbingan dan arahan dari berbagai pihak. Untuk itu ucapan terimakasih yang sebesar-besarnya penulis sampaikan kepada semua yang terlibat dan telah membantu terselesaikannya skripsi ini, terutama kepada:

1. Prof. Dr. H. Imam Suprayogo, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Prof. Drs. Sutiman Bambang Sumitro, SU., D.Sc, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Bapak Abdussakir, M.Pd, selaku Ketua Jurusan Matematika, Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Bapak Mohammad Jamhuri, M.Si, selaku dosen pembimbing skripsi yang telah bersedia meluangkan waktu, tenaga, pikiran serta memberikan arahan dan masukan yang sangat berguna dalam menyelesaikan skripsi ini.
5. Bapak Ach. Nashichuddin, M.A, selaku pembimbing agama yang telah bersedia memberikan pengarahan keagamaan dalam penyelesaian skripsi ini.
6. Segenap dosen dan staf pengajar, terima kasih atas semua ilmu yang telah diberikan.
7. Kedua orang tua tercinta dan segenap keluarga yang tiada henti selalu memberikan dukungan dan do'a.

8. Teman-teman matematika angkatan 2006, terima kasih atas segala pengalaman yang berharga dan kenangan indah yang telah terukir.
9. Teman-teman kost Jl. Jetis 105 Dau Malang, terima kasih banyak atas motivasi dan semangat yang telah diberikan.
10. Teman-teman *facebook*, terima kasih banyak atas *sharing* dan dukungan dari kalian.
11. Semua pihak yang telah membantu hingga terselesaikannya skripsi ini, baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari kesempurnaan, oleh karena itu, penulis selalu terbuka untuk menerima saran dan kritik yang konstruktif dari pembaca yang budiman, untuk perbaikan penulisan selanjutnya.

Malang, 13 Januari 2012  
Penulis,

Taufik Hidayatullah

## DAFTAR ISI

|                                   |        |
|-----------------------------------|--------|
| HALAMAN JUDUL .....               | i      |
| HALAMAN PENGAJUAN .....           | 2      |
| HALAMAN PERSETUJUAN .....         | 3      |
| HALAMAN PENGESAHAN .....          | 4      |
| PERNYATAAN KEASLIAN TULISAN ..... | 5      |
| MOTTO .....                       | 6      |
| HALAMAN PERSEMBAHAN .....         | 7      |
| KATA PENGANTAR .....              | 8      |
| DAFTAR ISI .....                  | 10     |
| DAFTAR GAMBAR .....               | xii    |
| DAFTAR TABEL .....                | 13     |
| ABSTRAK.....                      | 14     |
| ABSTRACT .....                    | 15     |
| مستخلص البحث .....                | 16     |
| <br><b>BAB I PENDAHULUAN</b>      |        |
| 1.1 Latar Belakang.....           | Error! |
| <b>Bookmark not defined.</b>      |        |
| 1.2 Rumusan Masalah .....         | Error! |
| <b>Bookmark not defined.</b>      |        |
| 1.3 Tujuan Penelitian .....       | Error! |
| <b>Bookmark not defined.</b>      |        |
| 1.4 Batasan Masalah .....         | Error! |
| <b>Bookmark not defined.</b>      |        |
| 1.5 Manfaat Penelitian.....       | Error! |
| <b>Bookmark not defined.</b>      |        |
| 1.6 Metode Penelitian .....       | Error! |
| <b>Bookmark not defined.</b>      |        |
| 1.7 Sistematika Penulisan.....    | Error! |
| <b>Bookmark not defined.</b>      |        |

**BAB II LANDASAN TEORI**

|                                                                     |               |
|---------------------------------------------------------------------|---------------|
| 2.1 Konsep Dasar Matematika .....                                   | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.1.1 Vektor .....                                                  | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.1.2 Matriks .....                                                 | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.2 Dasar Pengolahan Citra.....                                     | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.2.1 Pengertian Citra <i>Digital</i> .....                         | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.2.2 Koordinat Citra <i>Digital</i> .....                          | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.2.3 Jenis Citra.....                                              | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.2.4 Operasi Pengolahan Citra.....                                 | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.2.5 Analisis Citra .....                                          | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.3 Jaringan Syaraf Tiruan .....                                    | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.3.1 Aturan Pembelajaran .....                                     | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.3.2 Fungsi Aktivasi.....                                          | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.4 <i>Perceptron</i> .....                                         | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.5 Pengenalan Pola.....                                            | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.5.1 Pengenalan Sebuah Pola Karakter.....                          | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.5.2 Pengenalan Beberapa Pola Karakter .....                       | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 2.6 Kajian Agama yang Terkandung dalam Jaringan Syaraf Tiruan ..... | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| <b>BAB III PEMBAHASAN</b>                                           |               |
| 3.1 Deskripsi Sistem .....                                          | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |
| 3.2 Deskripsi Data .....                                            | <b>Error!</b> |
| <b>Bookmark not defined.</b>                                        |               |

|                                                              |        |
|--------------------------------------------------------------|--------|
| 3.2.1 Data <i>Training</i> .....                             | Error! |
| <b>Bookmark not defined.</b>                                 |        |
| 3.2.2 Data Uji Coba .....                                    | Error! |
| <b>Bookmark not defined.</b>                                 |        |
| 3.3 Proses Pengolahan Citra .....                            | Error! |
| <b>Bookmark not defined.</b>                                 |        |
| 3.3.1 <i>Preprocessing</i> .....                             | Error! |
| <b>Bookmark not defined.</b>                                 |        |
| 3.3.2 Proses Segmentasi Citra .....                          | Error! |
| <b>Bookmark not defined.</b>                                 |        |
| 3.4 Proses Jaringan Syaraf Tiruan .....                      | Error! |
| <b>Bookmark not defined.</b>                                 |        |
| 3.4.1 Menentukan Arsitektur Jaringan <i>Perceptron</i> ..... | Error! |
| <b>Bookmark not defined.</b>                                 |        |
| 3.4.2 Proses <i>Training</i> .....                           | Error! |
| <b>Bookmark not defined.</b>                                 |        |
| 3.4.3 Proses <i>Testing</i> (Uji Coba) .....                 | Error! |
| <b>Bookmark not defined.</b>                                 |        |
| 3.5 Jaringan Syaraf Tiruan dalam Pandangan Islam .....       | Error! |
| <b>Bookmark not defined.</b>                                 |        |
| <b>BAB IV PENUTUP</b>                                        |        |
| 4.1 Kesimpulan .....                                         | Error! |
| <b>Bookmark not defined.</b>                                 |        |
| 4.2 Saran .....                                              | Error! |
| <b>Bookmark not defined.</b>                                 |        |
| <b>DAFTAR PUSTAKA</b>                                        |        |
| <b>LAMPIRAN</b>                                              |        |

## DAFTAR GAMBAR

|            |                                                                                  |                                               |
|------------|----------------------------------------------------------------------------------|-----------------------------------------------|
| Gambar 2.1 | Koordinat Citra <i>Digital</i> .....                                             | <b>Error!</b><br><b>Bookmark not defined.</b> |
| Gambar 2.2 | Proses <i>Cropping</i> Citra .....                                               | <b>Error!</b><br><b>Bookmark not defined.</b> |
| Gambar 2.3 | Hubungan Ketetanggaan antar Piksel.....                                          | <b>Error!</b><br><b>Bookmark not defined.</b> |
| Gambar 2.4 | Arsitektur Model <i>Perceptron</i> .....                                         | <b>Error!</b><br><b>Bookmark not defined.</b> |
| Gambar 2.5 | Arsitektur Model <i>Perceptron</i> untuk Pengenalan Beberapa Pola Karakter ..... | <b>Error! Bookmark not defined.</b>           |
| Gambar 3.1 | Diagram Alir Sistem Pelatihan Pola Karakter .....                                | <b>Error!</b><br><b>Bookmark not defined.</b> |
| Gambar 3.2 | Diagram Alir Sistem Pengenalan Pola .....                                        | <b>Error!</b><br><b>Bookmark not defined.</b> |
| Gambar 3.3 | Arsitektur Jaringan <i>Perceptron</i> .....                                      | <b>Error!</b><br><b>Bookmark not defined.</b> |

## DAFTAR TABEL

|                                |        |
|--------------------------------|--------|
| Tabel 3.1 Bahan Uji Coba ..... | Error! |
| <b>Bookmark not defined.</b>   |        |
| Tabel 3.2 Hasil Uji Coba ..... | Error! |
| <b>Bookmark not defined.</b>   |        |





## ABSTRAK

Hidayatullah, Taufik. 2012. **Terapan Jaringan Syaraf Tiruan Perceptron untuk Pengenalan Karakter pada Gambar**. Skripsi, Jurusan Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.  
Pembimbing : (I) Mohammad Jamhuri, M.Si.

(II) Ach. Nashichudin, M.A.

Metode *perceptron* dalam jaringan syaraf tiruan dikenal sebagai algoritma yang hanya digunakan untuk mengklasifikasikan apakah sebuah pola masuk ke suatu kelas atau tidak. Namun dari sifat tersebut nampaknya *perceptron* juga mampu digunakan untuk mengklasifikasikan sebuah pola masuk ke kelas mana, dengan cara membandingkan pola ke dalam setiap kelas yang ada. Karena itu peneliti akan mencoba menerapkannya untuk mengenali beberapa karakter pada gambar.

Pada penelitian ini data masukan yang digunakan berupa gambar dari karakter, data tersebut diproses dan diwujudkan dalam proses pengolahan citra dengan proses *preprocessing* dan segmentasi, kemudian hasil dari pengolahan citra dimasukkan sebagai *input unit* pada jaringan syaraf tiruan *perceptron* yang berfungsi sebagai pengambil keputusan dengan tujuan mengenali karakter pada gambar yang sebelumnya telah dilakukan proses pembelajaran.

Dari hasil uji coba yang telah dilakukan, didapatkan nilai persentase keberhasilan sebesar 96,5%. Dapat disimpulkan bahwa jaringan syaraf tiruan *perceptron* dapat mengenali beberapa karakter pada gambar dengan sangat baik.

**Kata kunci :** Jaringan syaraf tiruan *perceptron*, pengolahan citra digital, pengenalan karakter.

## ABSTRACT

Hidayatullah, Taufik. 2012. **Perceptron Artificial Neural Networks Applied to Character Recognition in Image**. Thesis. Mathematics Programme Faculty of Science and Technology The State of Islamic University Maulana Malik Ibrahim Malang.

Promotor : (I) Mohammad Jamhuri, M. Si.

(II) Ach. Nashichudin, M.A.

Perceptron method in artificial neural network known as algorithms that are only used to classify whether a pattern into a class or not. But of these properties seems perceptron also capable of being used to classify a pattern into which class, by comparing the pattern into any existing class. Therefore researchers will try to apply it to recognize a few characters in the image.

In this study the input data used in the form of images of the characters, the data is processed and embodied in the process of image processing with the preprocessing and segmentation, then the result of image processing to be included as input units perceptron artificial neural network that functions as decision makers with the aim to recognize characters the image that has previously been carried out the learning process.

From the test results that have been made, a score is the percentage of success of 96.5%, it can be concluded that the perceptron artificial neural network can recognize a few characters in the image very well.

**Keyword :** *Perceptron artificial neural network, image processing, character recognition.*

## مستخلص البحث

هداية الله, توفيق. 2012. تطبيق الشبكات العصبية الاصطناعية فرجبطران (Perceptron) لاعتراف الحرف على الصور. رسالة البحث. قسم الرياضيات بكلية العلوم والتكنولوجيا. جامعة مولانا مالك ابراهيم الإسلامية الحكومية مالانج.  
المشرفين: (1). محمد جمهوري, الماجستير  
(2). أحمد نصيح الدين, الماجستير

الأسلوب الفرجبطران (Perceptron) في الشبكة العصبية الاصطناعية المعروف باسم خوارزميات فقط يتم استخداما لتصنيف ما إذا كان نمط إلى فئة أم لا. ولكن يبدو من هذه الخصائص أن فرجبطران (Perceptron) يمكن أيضا استخدامه لتصنيف نمط إلى أي فئة، من خلال مقارنة نمط في أي فئة موجودة. ولذلك فإن الباحث في محاولة لتطبيقه لاعتراف بعض الأحرف في الصورة. البيانات المستخدمة في هذه الدراسة هي شكل صور من الأحرف، يتم معالجة البيانات والمجسدة في عملية معالجة الصور مع تجهيزها وتجزئة، ثم تدرج نتيجة معالجة الصور في شبكة المدخلات العصبية الاصطناعية فرجبطران (Perceptron) الذي يعمل على صناع القرار وذلك بهدف التعرف على الأحرف الصورة التي سبق أن نفذت عملية التعلم. من نتائج الاختبارات التي أجريت، والنتيجة هي النسبة المئوية للنجاح 96.5 %، ويمكن استنتاج أن الشبكة العصبية الاصطناعية فرجبطران (Perceptron) تستطيع الاعتراف على بعض الأحرف في الصورة بشكل جيد للغاية.

الكلمات الرئيسية : الشبكات العصبية الاصطناعية فرجبطران (Perceptron) ومعالجة الصور الرقمية، والتعرف على النمط.

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Manusia sebagai makhluk yang paling sempurna diciptakan Allah mempunyai banyak sekali kelebihan jika dibandingkan dengan makhluk-makhluk ciptaan Allah yang lainnya. Bukti otentik dari kebenaran bahwa manusia merupakan makhluk yang paling sempurna di antara makhluk yang lain adalah ayat Al-Qur'an surat Al-Israa' ayat 70 dan At-Tiin ayat 4 sebagai berikut:

\* وَلَقَدْ كَرَّمْنَا بَنِي آدَمَ وَحَمَلْنَاهُمْ فِي الْبَرِّ وَالْبَحْرِ وَرَزَقْنَاهُمْ مِنَ الطَّيِّبَاتِ  
وَفَضَّلْنَاهُمْ عَلَى كَثِيرٍ مِّمَّنْ خَلَقْنَا تَفْضِيلًا ﴿٧٠﴾

Artinya : “dan Sesungguhnya telah Kami muliakan anak-anak Adam, Kami angkut mereka di daratan dan di lautan, Kami beri mereka rezki dari yang baik-baik dan Kami lebihkan mereka dengan kelebihan yang sempurna atas kebanyakan makhluk yang telah Kami ciptakan.” (Q.S. Al-Israa' [17]: 70)

لَقَدْ خَلَقْنَا الْإِنْسَانَ فِي أَحْسَنِ تَقْوِيمٍ ﴿٤﴾

Artinya : “Sesungguhnya Kami telah menciptakan manusia dalam bentuk yang sebaik-baiknya.” (Q.S. At-Tiin [95]: 4)

Satu hal yang membuat manusia lebih baik dari makhluk yang lain yaitu karena manusia dianugerahi oleh Allah dengan akal sehingga dengannya manusia mampu berfikir, mempertimbangkan, dan menentukan jalan pikirannya sendiri, sebagaimana Firman Allah dalam surat Al-Anfaal ayat 22 :

\* إِنَّ شَرَّ الدَّوَابِّ عِنْدَ اللَّهِ الصُّمُّ الْبُكْمُ الَّذِينَ لَا يَعْقِلُونَ ﴿٢٢﴾

*Artinya : “Sesungguhnya binatang (mahluk) yang seburuk-buruknya pada sisi Allah ialah; orang-orang yang pekak dan tuli (tidak menggunakan akalanya) yang tidak mengerti apa-apapun.” (Q.S. Al-Anfaal [8]: 22)*

Tahun 1943 Waren McCulloch dan Walter Pits membuat jaringan syaraf tiruan yang terinspirasi dari sel syaraf biologi, sama seperti otak (akal) pada manusia yang memproses suatu informasi. Elemen mendasar dari paradigma tersebut adalah struktur yang baru dari sistem pemrosesan informasi. Jaringan syaraf tiruan, seperti manusia, belajar dari suatu contoh. Jaringan syaraf tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran (Yani, 2005).

Dr. M. Isa Irawan (Dosen matematika ITS) dalam sebuah seminar nasional yang diadakan oleh jurusan matematika ITS bertema Matematika Kini dan Esok (4/12), menyatakan kaitan antara jaringan syaraf tiruan (JST) dengan matematika sangat kompleks. Jaringan ini telah dikembangkan sebagai generalisasi model matematika dari jaringan syaraf biologi (JSB). Lengkapnya, JST telah dikembangkan dengan menggunakan model matematis untuk menirukan cara kerja JSB. Melalui kompleksitas dan penyederhanaan JSB, komputasi matematika sederhana cukup mewakili mekanisme kerja JSB tersebut. Algoritma komputasi matematis ini yang dikenal dengan JST (jaringan syaraf tiruan, temuan matematikawan, Anonim, 2005).

Jaringan syaraf tiruan memiliki pendekatan yang berbeda untuk memecahkan masalah bila dibandingkan dengan sebuah komputer. Umumnya komputer menggunakan pendekatan algoritma (komputer menjalankan sekumpulan perintah untuk memecahkan masalah). Jika suatu perintah tidak

diketahui oleh komputer maka komputer tidak dapat memecahkan masalah yang ada. Sangat penting untuk mengetahui cara memecahkan suatu masalah pada komputer, sehingga komputer akan sangat bermanfaat jika dapat melakukan sesuatu ketika pengguna belum mengetahui cara melakukannya (Yani, 2005).

Jaringan syaraf tiruan dan suatu algoritma komputer tidak saling bersaing namun saling melengkapi satu sama lain. Pada suatu kegiatan yang besar, sistem yang diperlukan biasanya menggunakan kombinasi antara keduanya, misalkan di kantor-kantor dengan segala kegiatannya antara lain kegiatan administrasi, *marketing*, publikasi dan lain sebagainya, yang terkadang muncul kesulitan terutama dalam menghadapi data-data yang menumpuk dan tidak dapat mengolahnya dikarenakan suatu hal antara lain data tersebut berupa gambar (misal berekstensi bitmap) dan secara terpaksa harus diolah dengan mode gambar pula (Fauzi, 2009).

Tahun 1950-an, Frank Rosenblatt dan beberapa peneliti lain mengembangkan suatu kelas jaringan saraf yang disebut *Perceptron*. *Neuron* dalam jaringan yang mirip dengan McCulloch dan Pitts. Kunci kontribusi Rosenblatt adalah pengenalan aturan belajar untuk pelatihan jaringan *perceptron* untuk memecahkan masalah pengenalan pola. Ia membuktikan bahwa aturan belajarnya akan selalu bertemu untuk bobot jaringan yang benar (Putra, 2010).

Dari uraian di atas, penulis mencoba menerapkan jaringan syaraf tiruan untuk mengenali karakter pada gambar dengan menggunakan metode pembelajaran *perceptron*. Pengenalan karakter ini menggunakan data berupa gambar sebagai data masukannya, sehingga nanti akan diproses terlebih dahulu

dengan menggunakan pengolahan citra *digital* dan hasilnya akan digunakan sebagai *input unit* pada jaringan syaraf tiruan *perceptron*.

## 1.2 Rumusan Masalah

Dari latar belakang di atas, rumusan masalah penelitian ini adalah sebagai berikut:

1. Bagaimana mengenali karakter pada gambar?
2. Seberapa besar jaringan syaraf tiruan *perceptron* mengenali karakter pada gambar?

## 1.3 Tujuan Penelitian

Dari rumusan masalah di atas, tujuan penelitian ini adalah sebagai berikut:

1. Untuk menjelaskan langkah mengenali karakter pada gambar.
2. Untuk mengetahui seberapa besar jaringan syaraf tiruan *perceptron* mengenali karakter pada gambar.

## 1.4 Batasan Masalah

Agar penelitian ini tetap fokus terhadap masalah yang akan dibahas, maka diperlukan adanya batasan-batasan masalah. Adapun batasan-batasan masalah penelitian ini adalah sistem ini dibuat untuk mengenali karakter huruf cetak alphabet (A sampai Z latin kecil dan kapital), dan angka numerik (0 sampai 9).

## 1.5 Manfaat Penelitian

Hasil penelitian ini nantinya dapat dimanfaatkan, misalkan untuk melihat kode barang, sistem absen elektronik atau KTP elektronik, pengisian formulir *digital*, dan lain-lain.

## 1.6 Metode Penelitian

Metode yang digunakan dalam penelitian ini menggunakan metode kepustakaan (*library research*), yaitu dengan membaca buku, serta artikel yang berhubungan dengan pengolahan citra, jaringan syaraf tiruan *perceptron*, dan pengenalan pola. Adapun langkah-langkahnya sebagai berikut :

### 1. Mengumpulkan data

Data yang digunakan dalam penelitian ini berupa data gambar yang dibuat dalam program *windows paint* dengan latar berwarna putih dan objeknya (karakter) berwarna hitam.

### 2. Mendeskripsikan data

Data yang ada dideskripsikan dan dikelompokkan menurut jenis dan fungsinya, sehingga mempermudah dalam penelitian. Data yang digunakan pada penelitian ini dibagi menjadi 2 macam, yaitu data untuk *training* dan data untuk bahan uji coba

### 3. Pengolahan data dengan teknik pengolahan citra

Data yang digunakan dalam penelitian ini berupa gambar, karena itu terlebih dahulu data akan diolah dengan teknik pengolahan citra agar dapat diproses ke tahap selanjutnya.

### 4. Menentukan arsitektur jaringan syaraf tiruan *perceptron*

Untuk menggunakan jaringan syaraf tiruan, terlebih dahulu yang dilakukan adalah menentukan arsitektur jaringan syaraf tiruan *perceptron*, menentukan arsitekturnya ini berdasarkan permasalahan yang akan dipecahkan dengan menentukan hubungan *input* dengan target.

#### 5. Melakukan *training* pola

Jaringan syaraf tiruan ini memecahkan masalah yang ada dengan melalui tahap pembelajaran, dengan mempelajari data-data yang ada berdasarkan parameter yang diberikan, jaringan syaraf tiruan mempelajarinya sehingga didapatkan bobot yang sesuai. Dalam penelitian ini metode pembelajaran yang digunakan adalah metode pembelajaran *single layer perceptron*.

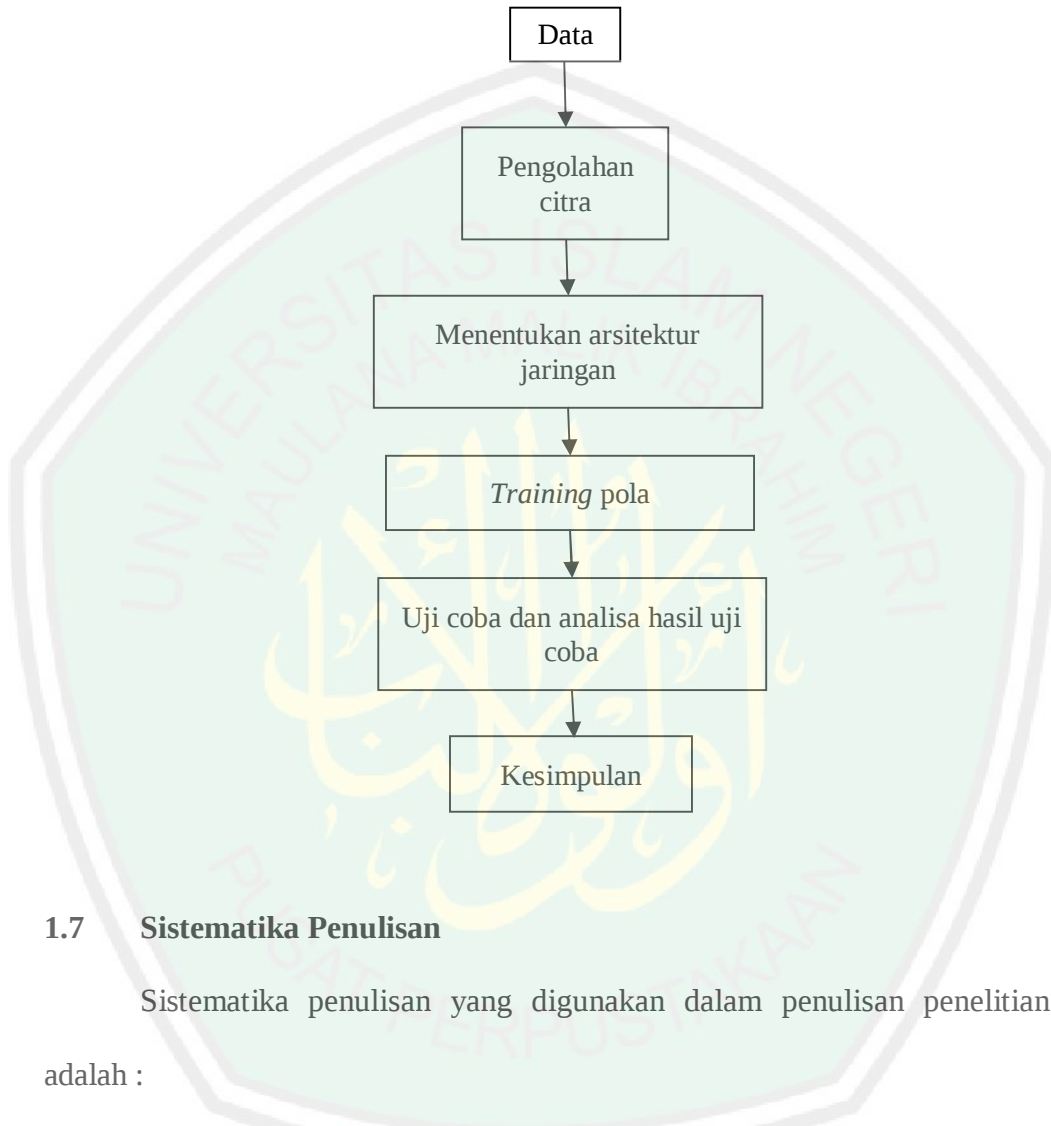
#### 6. Uji coba dan analisa hasil uji coba

Setelah sebelumnya jaringan dilatih dengan data-data pelatihan dan menghasilkan bobot-bobot yang sesuai, selanjutnya jaringan diuji coba dengan memakai data yang lain dan dengan menggunakan bobot yang sama dari hasil pelatihan. Selanjutnya hasil uji coba ini akan dianalisa, kemudian dihitung tingkat keberhasilannya.

#### 7. Kesimpulan

Dari hasil uji coba dan penelitian yang dilakukan, penulis dapat mengambil kesimpulan untuk menjawab permasalahan yang dikemukakan.

Secara garis besar, langkah-langkah penelitian ini sebagai berikut:



### 1.7 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penulisan penelitian ini adalah :

**BAB I PENDAHULUAN** : Dalam bab ini dijelaskan mengenai latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metode penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI : Dalam bab ini dijelaskan beberapa hal yang menjadi dasar dalam penelitian ini yaitu tentang matriks dan vektor, pengolahan citra, jaringan syaraf tiruan *perceptron*, dan pengenalan pola.

BABA III PEMBAHASAN : Dalam bab ini dijelaskan mengenai cara untuk mengenali karakter pada gambar menggunakan jaringan syaraf tiruan *perceptron*.

BAB IV PENUTUP : Dalam bab ini dipaparkan mengenai kesimpulan yang diperoleh dari hasil penelitian dan beberapa saran.

## BAB II

### LANDASAN TEORI

#### 1.1 Konsep Dasar Matematika

Teori-teori yang digunakan dalam penelitian ini tidak lepas dari konsep dasar matematika, karena itu terlebih dahulu akan dijelaskan dasar-dasar matematika yang dipakai pada penelitian ini.

##### 1.1.1 Vektor

Himpunan seluruh *tupel-n* yang terdiri dari bilangan-bilangan real, dinyatakan sebagai  $\mathbf{R}^n$ , disebut ruang-*n*. Suatu *tupel-n* tertentu pada  $\mathbf{R}^n$ , misalnya

$$u = (a_1, a_2, a_3, \dots, a_n)$$

disebut titik atau vektor. Bilangan-bilangan  $a_i$  disebut koordinat, komponen, *entri*, atau elemen dari  $u$ . Kadang-kadang suatu vektor pada ruang-*n*  $\mathbf{R}^n$  ditulis secara vertikal, dan bukannya secara horizontal. Vektor semacam ini disebut vektor kolom (Lipschutz & Lipson, 2006).

##### 2.1.1.1 Penjumlahan dan Perkalian Vektor

Perhatikan dua vektor  $u$  dan  $v$  pada  $\mathbf{R}^n$ , misalnya

$$u = (a_1, a_2, \dots, a_n) \text{ dan } v = (b_1, b_2, \dots, b_n)$$

Jumlah keduanya, ditulis  $u + v$ , adalah vektor yang diperoleh dengan menjumlahkan komponen-komponen yang bersesuaian dari  $u$  dan  $v$  yaitu,

$$u + v = (a_1 + b_1, a_2 + b_2, \dots, a_n + b_n) \quad (2.1)$$

Hasil kali *skalar* atau, sederhananya, hasilkali, dari vektor  $u$  dengan suatu bilangan real  $k$ , ditulis  $ku$ , adalah vektor yang diperoleh dengan mengalikan setiap komponen dari  $u$  dengan  $k$ . Sehingga

$$ku = k (a_1, a_2, \dots, a_n) = (ka_1, ka_2, \dots, ka_n) \quad (2.2)$$

Amati bahwa  $u + v$  pada persamaan (2.1) dan  $ku$  pada persamaan (2.2) juga adalah vektor-vektor pada  $\mathbf{R}^n$ . Penjumlahan vektor-vektor, di mana vektor-vektor tersebut memiliki komponen-komponen yang tidak sama banyak, tidak dapat didefinisikan.

Vektor negatif dan pengurangan pada  $\mathbf{R}^n$  didefinisikan sebagai berikut :

$$-u = (-1)u \quad \text{dan} \quad u - v = u + (-v) \quad (2.3)$$

Vektor  $-u$  disebut sebagai negatif dari  $u$  dan  $u - v$  disebut sebagai selisih antara  $u$  dan  $v$ .

Sifat-sifat dasar vektor dalam operasi penjumlahan vektor dan operasi skalar adalah sebagai berikut (Lipschutz & Lipson, 2006):

Untuk sebarang vektor  $u, v, w$  pada  $\mathbf{R}^n$  dan sebarang skalar  $k, k'$  pada  $\mathbf{R}$ , berlaku

- |                                  |                              |
|----------------------------------|------------------------------|
| (i) $(u + v) + w = u + (v + w),$ | (v) $k(u + v) = ku + kv,$    |
| (ii) $u + 0 = u,$                | (vi) $(k + k')u = ku + k'u,$ |
| (iii) $u + (-u) = 0,$            | (vii) $(kk')u = k(k'u),$     |
| (iv) $u + v = v + u,$            | (viii) $1u = u.$             |

#### 1.1.1.2 Hasilkali Titik (Hasilkali-Dalam)

Perhatikan sebarang vektor-vektor  $u$  dan  $v$  pada  $\mathbf{R}^n$ , misalnya

$$u = (a_1, a_2, \dots, a_n) \quad \text{dan} \quad v = (b_1, b_2, \dots, b_n)$$

Hasilkali titik (*dot product*) atau hasilkali-dalam (*inner product*) atau hasilkali skalar dari  $u$  dan  $v$  dinyatakan dan didefinisikan sebagai

$$u \cdot v = a_1 b_1 + a_2 b_2 + \dots + a_n b_n \quad (2.4)$$

dengan kata lain,  $u \cdot v$  pada persamaan 2.4 diperoleh dengan cara mengalikan komponen-komponen yang bersesuaian dan menjumlahkan hasilkali-hasilkali yang diperoleh. Vektor-vektor  $u$  dan  $v$  dikatakan *orthogonal* (atau saling tegak lurus) jika hasilkali titiknya adalah nol, yaitu jika  $u \cdot v = 0$ .

Sifat-sifat dasar dari hasilkali titik pada  $\mathbf{R}^n$  adalah sebagai berikut (Lipschutz & Lipson, 2006):

Untuk sebarang vektor-vektor  $u, v, w$  pada  $\mathbf{R}^n$  dan sebarang skalar  $k$  pada  $\mathbf{R}$ , berlaku

- (i)  $(u + v) \cdot w = u \cdot w + v \cdot w$ , (iii)  $u \cdot v = v \cdot u$ ,
- (ii)  $(ku) \cdot v = k(u \cdot v)$ , (iv)  $u \cdot u \geq 0$ , dan  $u \cdot u = 0$  jika  $u = 0$

### 2.1.2 Matriks

Matriks  $A$  atas *field*  $K$  atau, cukup disebut matriks  $A$  (jika  $K$  implisit) adalah susunan segiempat dari skalar-skalar yang biasanya dinyatakan dalam bentuk sebagai berikut:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

Baris-baris matriks  $A$  semacam ini adalah  $m$  deretan horizontal yang terdiri dari skalar-skalar:

$$(a_{11}, a_{12}, \dots, a_{1n}), (a_{21}, a_{22}, \dots, a_{2n}), \dots, (a_{m1}, a_{m2}, \dots, a_{mn})$$

dan kolom-kolom dari  $A$  adalah  $n$  deretan vertikal yang terdiri dari skalar-skalar:

$$\begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix}, \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix}, \dots, \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}$$

Perhatikan bahwa elemen  $a_{ij}$ , yang disebut entri  $ij$  atau elemen  $ij$ , muncul pada baris  $i$  dan kolom  $j$ . Seringkali menuliskan matriks tersebut hanya sebagai  $A = [a_{ij}]$ .

Matriks yang hanya memiliki satu baris disebut matriks baris atau vektor baris, matriks yang hanya memiliki satu kolom disebut matriks kolom atau vektor kolom. Matriks yang seluruh entrinya adalah nol disebut matriks nol dan biasanya dilambangkan dengan  $0$  (Lipschutz & Lipson, 2006).

#### 2.1.2.1 Penjumlahan Matriks dan Perkalian Skalar

Misalkan  $A = [a_{ij}]$  dan  $B = [b_{ij}]$ , adalah dua matriks dengan ukuran yang sama, misalnya matriks  $m \times n$ . Jumlah  $A$  dan  $B$ , ditulis  $A + B$ , adalah matriks yang diperoleh dengan menjumlahkan elemen-elemen yang bersesuaian dari  $A$  dan  $B$ , yaitu

$$A + B = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} & \dots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} & \dots & a_{2n} + b_{2n} \\ a_{31} + b_{31} & a_{32} + b_{32} & a_{33} + b_{33} & \dots & a_{3n} + b_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & a_{m3} + b_{m3} & \dots & a_{mn} + b_{mn} \end{bmatrix} \quad (2.5)$$

Hasilkali dari matriks  $A$  dengan suatu skalar  $k$ , ditulis  $kA$ , adalah matriks yang diperoleh dengan cara mengalikan setiap elemen  $A$  dengan  $k$ , yaitu

$$kA = \begin{bmatrix} ka_{11} & ka_{12} & ka_{13} & \dots & ka_{1n} \\ ka_{21} & ka_{22} & ka_{23} & \dots & ka_{2n} \\ ka_{31} & ka_{32} & ka_{33} & \dots & ka_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ ka_{m1} & ka_{m2} & ka_{m3} & \dots & ka_{mn} \end{bmatrix} \quad (2.6)$$

Dapat dilihat bahwa  $A + B$  pada persamaan (2.5) dan  $kA$  pada persamaan (2.6) juga merupakan matriks  $m \times n$ . Jadi

$$-A = (-1)A \quad \text{dan} \quad A - B = A + (-B) \quad (2.7)$$

Matriks  $-A$  disebut sebagai negatif dari matriks  $A$  dan  $A - B$  disebut sebagai selisih matriks  $A$  dan  $B$ . Penjumlahan matriks-matriks dengan ukuran yang berbeda tidak didefinisikan.

Berikut ini sifat-sifat dasar matriks dalam kaitannya dengan operasi penjumlahan matriks dan perkalian skalar (Lipschutz & Lipson, 2006).

Perhatikan sebarang matriks  $A, B, C$  (dengan ukuran yang sama) dan sebarang skalar  $k$  dan  $k'$ . Maka

- |                                   |                               |
|-----------------------------------|-------------------------------|
| (i) $(A + B) + C = A + (B + C)$ , | (v) $k(A + B) = kA + kB$ ,    |
| (ii) $A + 0 = 0 + A = A$ ,        | (vi) $(k + k')A = kA + k'A$ , |
| (iii) $A + (-A) = (-A) + A = 0$   | (vii) $(kk')A = k(k'A)$ ,     |
| (iv) $A + B = B + A$ ,            | (viii) $1 \cdot A = A$ .      |

#### 2.1.2.2 Perkalian Matriks

Hasilkali matriks  $A$  dan  $B$ , ditulis  $AB$ , agak rumit. Karena alasan inilah akan dimulai kasus khusus. Hasil kali  $AB$  dari matriks baris  $A = [a_i]$  dan matriks kolom  $B = [b_i]$  dengan elemen yang sama banyaknya didefinisikan sebagai skalar (atau matriks  $1 \times 1$ ) yang diperoleh dengan mengalikan entri-entri yang bersesuaian dan menjumlahkannya, yaitu

$$AB = [a_1, a_2, \dots, a_n] \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = a_1 b_1 + a_2 b_2 + \dots + a_n b_n = \sum_{k=1}^n a_k b_k$$

Ditekankan bahwa  $AB$  adalah sebuah skalar (atau matriks  $1 \times 1$ ). Hasil kali  $AB$  tidak dapat didefinisikan jika elemen-elemen  $A$  dan  $B$  tidak sama banyaknya.

Perkalian matriks secara umum didefinisikan sebagai berikut (Lipschutz & Lipson, 2006).

**Definisi:** Anggaplah  $A = [a_{ik}]$  dan  $B = [b_{kj}]$  adalah matriks-matriks yang sedemikian rupa sehingga banyaknya kolom dari  $A$  sama dengan banyaknya baris dari  $B$ ; misalnya,  $A$  adalah matriks  $m \times p$  dan  $B$  adalah matriks  $p \times n$ . Maka hasil kali  $AB$  adalah matriks  $m \times n$  yang entri  $ij$ -nya diperoleh dengan cara mengalikan baris ke- $i$  dari  $A$  dengan kolom ke- $j$  dari  $B$ . di mana,

$$c_{ij} = a_{i1} b_{1j} + a_{i2} b_{2j} + \dots + a_{ip} b_{pj} = \sum_{k=1}^p a_{ik} b_{kj}$$

Hasil kali  $AB$  tidak didefinisikan jika  $A$  adalah matriks  $m \times p$  dan  $B$  adalah matriks  $q \times n$ , di mana  $p \neq q$ .

Misalkan  $A, B, C$  adalah matriks-matriks. Maka, perkalian dan penjumlahan matriks-matriks tersebut memenuhi sifat-sifat berikut (Lipschutz & Lipson, 2006).

- (i)  $(AB)C = A(BC)$  (hukum asosiatif),
- (ii)  $A(B + C) = AB + AC$  (hukum distributif kiri),
- (iii)  $(B + C)A = BA + CA$  (hukum distributif kanan),
- (iv)  $k(AB) = (kA)B = A(kB)$ , di mana  $k$  adalah suatu skalar.

Diketahui bahwa  $0A = 0$  dan  $B0 = 0$ , di mana  $0$  adalah matriks nol.

### 2.1.2.3 Transpose Matriks

*Transpose* dari matriks  $A$ , ditulis  $A^T$ , adalah matriks yang diperoleh dengan cara menuliskan kolom-kolom dari  $A$  berurutan, sebagai baris-barisnya.

Dengan kata lain, jika  $A = [a_{ij}]$  adalah matriks  $m \times n$ , maka  $A^T = [b_{ij}]$  adalah matriks  $n \times m$ , di mana  $b_{ij} = a_{ji}$ .

Misalkan  $A$  dan  $B$  adalah matriks-matriks dan  $k$  adalah suatu skalar. Maka, berlaku sifat berikut (Lipschutz & Lipson, 2006).

- |                               |                           |
|-------------------------------|---------------------------|
| (i) $(A + B)^T = A^T + B^T$ , | (iii) $(kA)^T = kA^T$ ,   |
| (ii) $(A^T)^T = A$ ,          | (iv) $(AB)^T = B^T A^T$ . |

## 2.2 Dasar Pengolahan Citra

Secara umum, pengolahan citra digital menunjuk pada pemrosesan gambar 2 dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data 2 dimensi.

### 2.2.1 Pengertian Citra Digital

Suatu citra agar dapat direpresentasikan secara numerik, maka citra harus didigitalisasi. Citra dapat didefinisikan sebagai suatu matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar / piksel) menyatakan tingkat keabuan pada titik tersebut.

Pada umumnya citra berbentuk empat persegi panjang dan dimensi ukurannya dinyatakan sebagai (tinggi x lebar). Citra dengan tinggi  $M$  piksel,

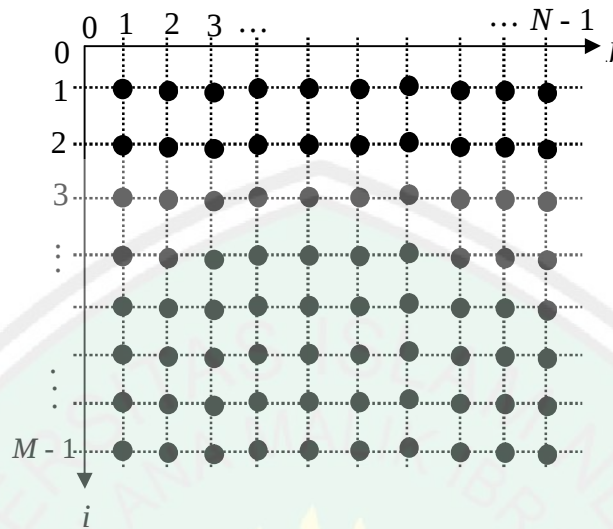
lebarnya  $N$  piksel, dan memiliki intensitas  $A$  dapat direpresentasikan sebagai suatu matriks yang berukuran  $M$  baris dan  $N$  kolom sebagai berikut:

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1N} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2N} \\ A_{31} & A_{32} & A_{33} & \dots & A_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{M1} & A_{M2} & A_{M3} & \dots & A_{MN} \end{bmatrix} \quad (2.8)$$

Pada persamaan (2.8) dapat dijelaskan bahwa indeks baris ( $i$ ) dan indeks kolom ( $j$ ) menyatakan suatu koordinat titik pada citra sedangkan  $A(i,j)$  merupakan intensitas (derajat keabuan) pada titik ( $i,j$ ). Masing-masing titik ( $i,j$ ) di citra tersebut piksel. Jadi citra yang berukuran  $M \times N$  mempunyai  $MN$  buah piksel. Apabila nilai  $i$ ,  $j$ , dan intensitas  $A$  secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra *digital* (Putra, 2010).

### 2.2.2 Koordinat Citra Digital

Secara umum, metode yang digunakan untuk menyatakan lokasi dari suatu citra *digital* adalah menggunakan koordinat piksel. Representasi bentuk koordinat dalam citra digital sedikit berbeda dengan representasi dalam grafik matematika pada umumnya (koordinat *cartesius*). Koordinat pusat dalam citra *digital* berada di sudut kiri atas, seperti yang ditunjukkan pada gambar 2.1 berikut ini.



Gambar 2. 1 Koordinat Citra *Digital*

### 2.2.3 Jenis Citra

Nilai suatu piksel memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0-255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra integer. Berikut adalah jenis-jenis citra berdasarkan nilai pikselnya.

#### 2.2.3.1 Citra Warna

Pada citra warna, setiap titik mempunyai warna spesifik yang merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau dan biru. Format citra ini sering disebut citra RGB. Setiap warna dasar mempunyai intensitas sendiri dengan nilai maksimum 255 (8 bit) (Wulan, 2010).

RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang-nilai. Untuk monitor komputer, nilai

rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan *biner* yang digunakan oleh mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak  $256 \times 256 \times 256 = 16777216$  jenis warna. Sebuah jenis warna, dapat dibayangkan sebagai sebuah vektor di ruang 3 dimensi yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y dan komponen-z. Misalkan sebuah vektor dituliskan sebagai  $r = (x,y,z)$ . Untuk warna, komponen-komponen tersebut digantikan oleh komponen R(ed), G(reen), B(lue). Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = RGB (30, 75, 255). Putih = RGB (255,255,255), sedangkan untuk hitam= RGB (0,0,0).

#### 2.2.3.2 Citra Grayscale

Citra *grayscale* mempunyai nilai kuantitas derajat keabuan sampai tingkatan ke-256, artinya mempunyai skala abu dari 0 sampai 255 atau selang [0, 255] dan citra ini membutuhkan 1 byte (8 bit) untuk representasi setiap pikselnya (Munir, 2004).

#### 2.2.3.3 Citra Biner

Citra *biner* adalah citra yang hanya mempunyai dua nilai derajat keabuan : hitam dan putih. Piksel-piksel objek bernilai nol (0) dan piksel-piksel latar belakang bernilai satu (1).

Pada waktu menampilkan representasi *biner* pada citra, nilai satu (1) adalah putih dan nilai nol (0) adalah hitam.

## 2.2.4 Operasi Pengolahan Citra

### 2.2.4.1 Operasi Skala Keabuan (Grayscale)

Dengan menggunakan representasi nilai-nilai intensitas komponen RGB (*Red, Green, Blue*) pada koordinat piksel, sebuah citra warna dapat diubah menjadi citra yang terdiri dari putih dan hitam yang biasanya disebut citra *grayscale*. Operasi *grayscale* dirumuskan sebagai berikut:

$$\text{grayscale} = \frac{R + G + B}{3} \quad (2.9)$$

atau bisa juga dengan memberi bobot ( $w$ ) pada RGB karena mata manusia lebih sensitif pada warna hijau, kemudian merah, terakhir biru. Secara algoritma dapat dituliskan seperti pada persamaan (2.10) berikut ini.

$$\text{grayscale} = w_r R + w_g G + w_b B \quad (2.10)$$

Berdasarkan NTSC (*National Television System Committee*), diketahui

$$w_r = 0,299 \quad R = \text{Red}$$

$$w_g = 0,587 \quad G = \text{Green}$$

$$w_b = 0,144 \quad B = \text{Blue}$$

dimana  $w_r$  adalah bobot R,  $w_g$  adalah bobot G, dan  $w_b$  adalah bobot B (Budhi, dkk).

### 2.2.4.2 Operasi Pengambangan (Thresholding)

Konversi dari citra hitam-putih menjadi citra *biner* dilakukan dengan operasi pengambangan (*thresholding*). Operasi pengambangan mengelompokkan nilai derajat keabuan setiap piksel ke dalam dua kelas, yaitu hitam dan putih. Dengan setiap piksel di dalam citra dipetakan dengan dua nilai, satu (1) dan nol

(0), dengan fungsi pengambangan. Fungsi pengambangan dapat dirumuskan sebagai berikut:

$$A_b(i, j) = \begin{cases} 0, & \text{jika } A_g(i, j) \leq T \\ 1, & \text{lainnya} \end{cases} \quad (2.11)$$

Dalam hal ini pada persamaan (2.11),  $A_g(i, j)$  adalah citra *grayscale*,  $A_b(i, j)$  adalah citra *biner* dan  $T$  adalah nilai ambang yang dispesifikasikan. Objek dibuat berwarna gelap (0 atau hitam) sedangkan latar belakang terang (1 atau putih) (Munir, 2004).

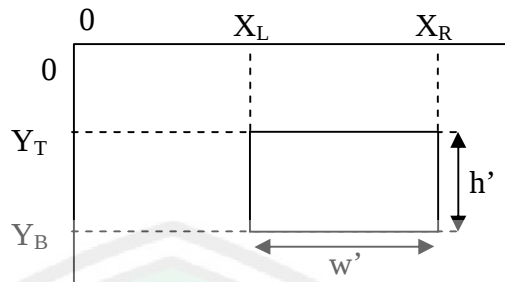
#### 2.2.4.3 Pemotongan (Cropping)

*Cropping* pada pengolahan citra berarti memotong satu bagian dari citra sehingga diperoleh citra yang diharapkan. Ukuran pemotongan citra tersebut berubah sesuai dengan ukuran citra yang diambil. *Cropping* dilakukan pada koordinat  $(x, y)$  sampai pada koordinat  $(m, n)$ . Oleh karena itu, pertama kali yang harus dilakukan adalah menentukan koordinat-koordinat tersebut. Penulis menyebutnya koordinat  $X_L$ ,  $Y_T$ ,  $X_R$  dan  $Y_B$ , dimana  $x$  memiliki koordinat  $X_L$  sampai  $X_R$  ( $X_L < x < X_R$ ) dengan selang  $[X_L, X_R]$  dan  $y$  memiliki koordinat  $Y_T$  sampai  $Y_B$  ( $Y_T < y < Y_B$ ) dengan selang  $[Y_T, Y_B]$ , didapat  $(X_L, Y_T)$  adalah koordinat titik sudut kiri atas dan  $(X_R, Y_B)$  adalah koordinat titik sudut kanan bawah. Maka ukuran *cropping* citra dapat dirumuskan sebagai berikut (Fauzi, 2009) :

$$w' = X_R - X_L \quad (2.12)$$

$$h' = Y_B - Y_T \quad (2.13)$$

Proses *cropping* dapat digambarkan sebagai berikut.



Gambar 2. 2 Proses *Cropping* Citra

#### 2.2.4.4 Normalisasi

Normalisasi pada pengolahan citra berarti mentransformasikan citra ke bentuk citra normal yang sesuai dengan kebutuhan. Penskalaan ini tergantung besar dan kecil ukuran pada citra semula artinya tidak berarti apakah citra membesar atau mengecil tergantung ukuran citra semula. Penskalaan citra dirumuskan sebagai berikut:

$$M' = S_M \cdot M \quad (2.14)$$

$$N' = S_N \cdot N \quad (2.15)$$

Pada persamaan (2.14) dan (2.15), dimana  $(M, N)$  adalah ukuran citra input,  $(M', N')$  adalah ukuran citra output, dan  $(S_M, S_N)$  adalah variabel penskalaan yang diinginkan (Putra, 2010).

### 2.2.5 Analisis Citra

#### 2.2.5.1 Segmentasi Citra

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

##### 2.2.5.1.1 Edge Detection (Deteksi Tepi)

Tepian dari citra mengandung informasi penting dari citra bersangkutan. Tepian citra dapat merepresentasikan objek-objek yang tergantung dalam citra

tersebut, bentuk, dan ukurannya serta terkandung juga informasi tentang teksturnya. Tepian citra adalah posisi di mana intensitas piksel dari citra berubah dari nilai rendah ke nilai tinggi atau sebaliknya. Deteksi tepi umumnya adalah langkah awal melakukan segmentasi citra.

Terdapat berbagai operator deteksi tepi yang telah dikembangkan berdasarkan turunan pertama (*first order derivative*), diantaranya operator *Robert*, operator *Sobel*, operator *Prewitt*, operator *Krisch*, dan operator *Canny*. Konsep dasar dari perhitungan deteksi tepi menggunakan turunan pertama adalah dengan memanfaatkan perbedaan nilai suatu piksel dengan piksel tetangganya, seperti persamaan (2.16) dan (2.17) berikut :

$$\Delta M = A(i, j) - A(i, j + 1) \quad (2.16)$$

$$\Delta N = A(i, j) - A(i - 1, j) \quad (2.17)$$

$\Delta M$  menyatakan perbedaan nilai piksel dalam arah  $M$  , sedangkan  $\Delta N$  menyatakan perbedaan nilai piksel dalam arah  $N$  (Putra, 2010).

Dalam penelitian ini, operator yang digunakan untuk deteksi tepi adalah operator *Sobel*, ini dikarenakan operator *Sobel* lebih sensitif terhadap tepian diagonal. Operator ini terbentuk dari matriks berukuran 3x3 seperti berikut :

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

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

Operator *Sobel* juga dapat terpisah untuk mendapatkan gradient horizontal dan gradient vertikalnya. Untuk mendapatkan gradient gabungan dapat dihitung dengan menggunakan persamaan (2.18) berikut ini (Putra, 2010):

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.18)$$

#### 2.2.5.1.2 Penandaan Komponen Terhubung (*Connected Component Labeling*)

Penandaan 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.

Suatu piksel  $p$  pada koordinat  $(i, j)$  memiliki empat piksel tetangga (2 dalam arah horizontal dan 2 arah vertikal) dengan koordinat sebagai berikut.

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1) \quad (2.19)$$

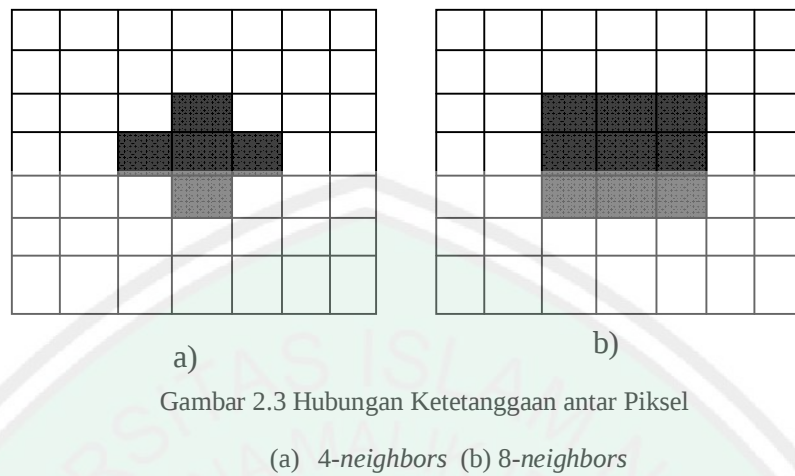
Keempat piksel tersebut sering disebut dengan *4-neighbors* dari  $p$  dan dinyatakan dengan  $N_4(p)$ .

Selain dalam arah horizontal dan vertikal, piksel tetangga dari  $p$  juga berada pada arah diagonal dengan koordinat

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1) \quad (2.20)$$

dan dinyatakan dengan  $N_D(p)$ .

$N_4(p)$  dan  $N_D(p)$  bersama-sama membentuk *8-neighbors* dari  $p$  dan dinyatakan dengan  $N_8(p)$ . Hubungan ketetanggaan antar piksel dapat digambarkan sebagai berikut :



Penandaan komponen terhubung dilakukan dengan memeriksa suatu citra, piksel per piksel (dari kiri ke kanan dan atas ke bawah) untuk mengidentifikasi area piksel terhubung yaitu suatu area dari piksel berbatasan yang memiliki nilai intensitas sama atau nilai intensitas berada dalam satu himpunan  $V$  (pada citra biner  $V = \{1\}$ ) (Putra, 2010).

### 2.3 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) (Bahasa Inggris: *Artificial Neural Network* (ANN), atau juga disebut *Simulated Neural Network* (SNN), atau umumnya hanya disebut *Neural Network* (NN), adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia. JST muncul pertama kali setelah model sederhana dari *neuron* buatan yang diperkenalkan oleh McCulloch dan Pitts pada tahun 1943. Model sederhana tersebut dibuat berdasarkan fungsi *neuron* biologis yang merupakan dasar *unit* persinyalan dari sistem saraf (Putra, 2010).

JST merupakan sistem adaptif yang dapat merubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut. JST dapat digunakan untuk memodelkan hubungan yang kompleks antara *input* dan *output* untuk menemukan pola-pola pada data. Memperhatikan prinsip kerja jaringan syaraf tiruan tersebut terlihat bahwa betapa luasnya pengetahuan Allah SWT. Sebagaimana Firman-Nya dalam sura An-Nisaa' ayat 126 :

وَلِلَّهِ مَا فِي السَّمَوَاتِ وَمَا فِي الْأَرْضِ وَكَانَ اللَّهُ بِكُلِّ شَيْءٍ مُّحِيطًا ﴿١٢٦﴾

Artinya : “Kepunyaan Allah-lah apa yang di langit dan apa yang di bumi, dan adalah (pengetahuan) Allah Maha meliputi segala sesuatu”. (Q.S. An-Nisaa' : 126)

Ayat ini menjelaskan betapa luasnya pengetahuan Allah, Dia menciptakan manusia sebagai makhluk yang paling sempurna dengan menganugerahinya akal sehingga mereka dapat berfikir. Sebagaimana yang dijelaskan dalam surat At-Tiin ayat 4 sebagai berikut:

لَقَدْ خَلَقْنَا الْإِنْسَانَ فِي أَحْسَنِ تَقْوِيمٍ ﴿٤﴾

Artinya : “Sesungguhnya Kami telah menciptakan manusia dalam bentuk yang sebaik-baiknya.” (Q.S. At-Tiin [95]: 4)

dari ayat-ayat di atas hendaknya kita berfikir bahwa betapa kita ini tidak ada apa-apanya dibandingkan dengan kepintaran-Nya Allah yang Maha luas pengetahuan-Nya.

### 2.3.1 Aturan Pembelajaran

Pembelajaran berarti belajar dari kesalahan. Semakin sering jaringan syaraf tiruan digunakan semakin cerdaslah dia, artinya kesalahan semakin kecil. Kecepatan peningkatan kecerdasan ditentukan oleh nilai parameter kecepatan pembelajaran (*learning rate*) disimbolkan dengan  $\alpha$ . Bagaimanapun, jika sampai iterasi ke-N komputasi ulang (istilahnya: N iterasi) kesalahannya tetap besar maka proses pembelajaran dihentikan dan jaringan digunakan apa adanya.

Ada banyak jenis aturan belajar jaringan syaraf tiruan. Mereka dibagi dalam 3 kategori besar, yaitu:

1. Pembelajaran dengan pengawasan (*supervised learning*)

Pada pembelajaran dengan pengawasan, aturan pembelajaran dilengkapi dengan serangkaian contoh (himpunan pelatihan) dari perilaku jaringan yang tepat :

$$\{P_1, t_1\}, \{P_2, t_2\}, \dots, \{P_q, t_q\}$$

dimana  $P_i$  adalah sebuah *input* jaringan dan yang berhubungan dengan (*target*) *output* yang sesuai, dengan  $i = 1$  sampai  $q$ . Seperti *input* yang diterapkan ke jaringan, *output* jaringan dibandingkan dengan target. Aturan pembelajaran kemudian digunakan untuk mengatur bobot dan bias dari jaringan untuk memindahkan *output* jaringan mendekati target. Aturan *perceptron* masuk dalam kategori pembelajaran pengawasan ini.

2. Pembelajaran tanpa pengawasan (*unsupervised learning*)

Pada pembelajaran tanpa pengawasan, bobot dan bias diubah sebagai tanggapan untuk jaringan *input* saja. Tidak ada target *output* yang tersedia.

Kebanyakan dari algoritma ini melakukan semacam operasi *clustering* (pengelompokan). Mereka belajar untuk mengkategorikan pola *input* dalam sejumlah kelas yang terbatas. Hal ini sangat berguna khususnya dalam aplikasi vektor kuantisasi.

### 3. Pembelajaran penguatan tanpa pengawasan atau dinilai (*reinforcement learning*).

Pembelajaran dengan penguatan adalah serupa dengan pembelajaran dengan pengawasan, kecuali bahwa pada *supervised learning* dilengkapi dengan *output* yang benar untuk setiap *input* jaringan, sementara algoritma *reinforcement learning* hanya diberikan satu nilai. Nilai adalah sebuah ukuran dari kinerja jaringan atas beberapa rangkaian *input*. Saat ini, jenis pembelajaran ini jauh lebih sedikit dibandingkan pembelajaran dengan pengawasan. Pembelajaran ini tampaknya paling cocok untuk aplikasi-aplikasi sistem control (Kusumadewi, 2004).

#### 2.3.2 Fungsi Aktivasi

Dalam JST, fungsi aktivasi digunakan untuk menentukan keluaran suatu *neuron*. Argument fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya) (Siang, 2004).

$$\text{Jika net} = \sum x_i w_i$$

Maka fungsi aktivasinya adalah

$$f(\text{net}) = f\left(\sum x_i w_i\right) \quad (2.21)$$

Beberapa fungsi aktivasi yang digunakan adalah (Siang, 2004) :

- 1) Fungsi *threshold* (batas ambang)

$$f(x) = \begin{cases} 1, & \text{jika } x \geq \text{threshold} \\ 0, & \text{jika } x \leq \text{threshold} \end{cases} \quad (2.22)$$

Fungsi *threshold* di atas merupakan fungsi *threshold biner*. Untuk kasus bilangan bipolar, maka 0 diganti dengan angka -1. Sehingga diubah menjadi

$$f(x) = \begin{cases} 1, & \text{jika } x \geq \text{threshold} \\ -1, & \text{jika } x \leq \text{threshold} \end{cases} \quad (2.23)$$

Adakalanya dalam JST ditambahkan suatu unit yang nilainya selalu 1. Unit tersebut dikenal dengan bias. Bias dapat dipandang sebagai sebuah *input* yang nilainya selalu 1. Bias berfungsi untuk mengubah *threshold* menjadi = 0.

## 2) Fungsi sigmoid

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.24)$$

Fungsi ini sering digunakan karena nilai fungsinya sangat mudah untuk dideferensiasikan.

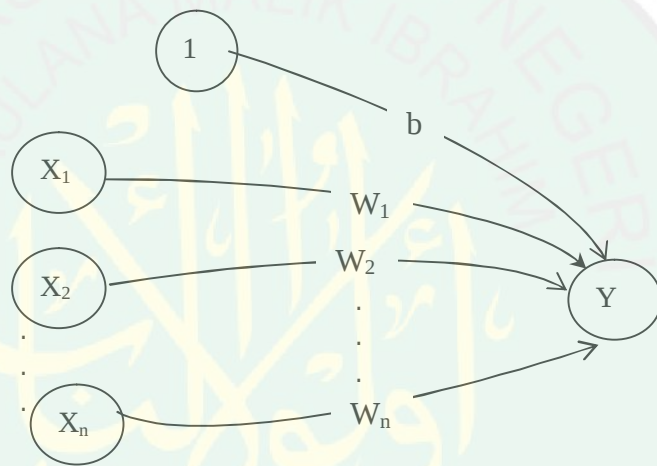
## 3) Fungsi identitas

$$f(x) = x \quad (2.25)$$

digunakan jika keluaran yang dihasilkan oleh JST merupakan sembarang bilangan riil (bukan hanya pada *range* [0, 1] atau [1, -1]).

## 2.4 Perceptron

*Perceptron* termasuk salah satu bentuk jaringan syaraf tiruan yang paling sederhana yang diperkenalkan oleh Rosenblatt sekitar tahun 1957. *Perceptron* biasanya digunakan untuk mengklasifikasikan suatu tipe pola tertentu yang sering dikenal dengan pemisahan secara linier. Arsitektur jaringan *perceptron* dapat dilihat pada gambar 2.4 berikut.



Gambar 2. 4 Arsitektur Model *Perceptron*

Jaringan terdiri dari beberapa *unit* masukan (ditambah sebuah bias), dan memiliki sebuah *unit* keluaran. Hanya saja, fungsi aktivasinya bukan merupakan fungsi *biner (bipolar)*, tetapi memiliki kemungkinan nilai -1, 0 atau 1.

Untuk suatu harga *threshold*  $\theta$  yang ditentukan, fungsi aktivasi dirumuskan pada persamaan (2.26) sebagai berikut.

$$y = f(\text{net}) = \begin{cases} 1, & \text{jika } \text{net} > \theta \\ 0, & \text{jika } -\theta \leq \text{net} \leq \theta \\ -1, & \text{jika } \text{net} < -\theta \end{cases} \quad (2.26)$$

Fungsi aktivasi juga ditentukan secara random atau acak tetapi nilainya tetap. Nilai inilah yang disebut *threshold*. Secara geometris, fungsi aktivasi

membentuk 2 garis sekaligus, masing-masing dengan persamaan (2.27) dan (2.28) berikut (Siang, 2004).

$$X_1W_1 + X_2W_2 + \dots + X_nW_n + b = \theta \quad \text{dan} \quad (2.27)$$

$$X_1W_1 + X_2W_2 + \dots + X_nW_n + b = -\theta \quad (2.28)$$

Algoritma pembelajaran :

Misalkan :

$s$  adalah vektor masukan, dan  $t$  adalah target keluaran.

$\alpha$  adalah laju pemahaman (*learning rate*) yang ditentukan.

$\Theta$  adalah *threshold* yang ditentukan

1. Inisialisasi semua bobot dan bias (umumnya  $W_i = b = 0$ ), tentukan laju pemahaman ( $\alpha$ ). Untuk penyederhanaan, biasanya  $\alpha$  diberi nilai = 1.
2. Selama ada elemen vektor masukan yang respon *unit* keluarannya tidak sama dengan target, lakukan:
  - a. Set aktivasi unit masukan  $x_i = s_i$  ( $i = 1, \dots, n$ ).
  - b. Hitung respon unit keluaran :

$$net = \sum_{i=1}^n X_i W_i + b$$

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

- c. Perbaiki bobot pola yang mengandung kesalahan ( $y \neq t$ ) menurut persamaan :

$$w_i (\text{baru}) = w_i (\text{lama}) + \Delta w, \quad \text{dengan } \Delta w = \alpha.t.x_i$$

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

Keterangan :  $\Delta w$  = perubahan bobot

$\Delta b$  = perubahan bias

$b$  = bias

$w_i$  = bobot yang ke- $i$ ,  $i = 1, 2, \dots, n$

$t$  = target

$x_i$  = unit masukan yang ke- $i$ ,  $i = 1, 2, \dots, n$

$\alpha$  = laju pemahaman

Ada beberapa hal yang perlu diperhatikan dalam algoritma tersebut :

- a. Iterasi dilakukan terus hingga semua pola memiliki keluaran jaringan yang sama dengan targetnya (jaringan sudah memahami pola). Iterasi tidak berhenti setelah semua pola dimasukkan.
- b. Pada langkah 2 (c), perubahan bobot hanya dilakukan pada pola yang mengandung kesalahan (keluaran jaringan  $\neq$  target). Perubahan tersebut merupakan hasil kali unit masukan dengan target dan laju pemahaman. Perubahan bobot hanya akan terjadi jika *unit* masukan  $\neq 0$ .
- c. Kecepatan iterasi ditentukan pula oleh laju pemahaman ( $=\alpha$  dengan  $0 \leq \alpha \leq 1$ ) yang dipakai. Semakin besar harga  $\alpha$ , semakin sedikit iterasi yang diperlukan. Akan tetapi jika  $\alpha$  terlalu besar, maka akan merusak pola yang sudah benar sehingga pemahaman menjadi lambat.

## 2.5 Pengenalan Pola

*Perceptron* dapat dipakai untuk mengenali pola karakter. Dengan berbagai pola masukan yang menyerupai huruf-huruf *alphabeth*, *perceptron* dapat dilatih untuk mengenalinya.

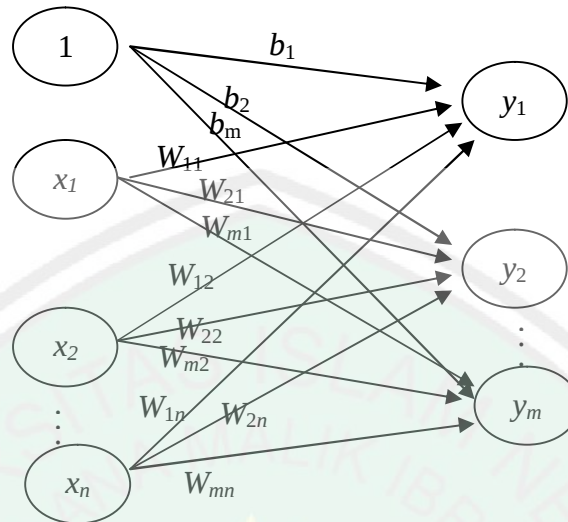
### 2.5.1 Pengenalan Sebuah Pola Karakter

Algoritma untuk mengenali apakah pola masukan yang diberikan menyerupai sebuah karakter tertentu (misal mirip huruf “A”) atau tidak adalah sebagai berikut (Siang, 2004):

- i. Nyatakan tiap pola masukan sebagai vektor *biner* yang elemennya adalah tiap titik dalam pola tersebut.
- ii. Berikan nilai target = +1 jika pola masukan menyerupai huruf yang diinginkan. Jika sebaliknya, berikan nilai target = -1
- iii. Lakukan proses pelatihan *perceptron* seperti yang dijelaskan sebelumnya.

### 2.5.2 Pengenalan Beberapa Pola Karakter

Pengenalan beberapa pola karakter sekaligus (misal “A” atau bukan “A”, “B” atau bukan “B”, dan seterusnya) dilakukan dengan cara menggabungkan beberapa model *perceptron*. Jadi ada beberapa unit keluaran sekaligus, seperti yang tampak pada gambar 2.5 dibawah ini :



Gambar 2. 5 Arsitektur Model *Perceptron* untuk Pengenalan Beberapa Pola Karakter

Setiap *unit* masukan dihubungkan dengan setiap *unit* target. Bobot penghubung dari *unit*  $x_i$  ke  $y_j$  adalah  $W_{ji}$ . Demikian juga bias dihubungkan dengan semua *unit* keluaran dengan bobot masing-masing  $b_1, b_2, \dots, b_m$ .

Dari gambar 2.5 didapatkan sebuah matriks  $W$  sebagai berikut.

$$W = \begin{bmatrix} W_{11} & W_{12} & W_{13} & \dots & W_{1n} \\ W_{21} & W_{22} & W_{23} & \dots & W_{2n} \\ W_{31} & W_{32} & W_{33} & \dots & W_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_{m1} & W_{m2} & W_{m3} & \dots & W_{mn} \end{bmatrix}$$

Algoritma pembelajaran *perceptron* untuk pengenalan beberapa pola sekaligus adalah sebagai berikut :

1. Nyatakan tiap pola masukan sebagai vektor *biner* yang elemennya adalah tiap titik dalam pola tersebut.
2. Berikan nilai target  $t_j = +1$  jika pola masukan menyerupai huruf yang diinginkan. Jika sebaliknya, berikan nilai target  $= t_j = -1$  ( $j = 1, 2, \dots, m$ ).
3. Berikan inisialisasi bobot, bias, laju pemahaman dan *threshold*.

4. Lakukan proses pelatihan *perceptron* untuk tiap *unit* keluaran sebagai berikut :

a. Hitung respon *unit* keluaran ke- $j$  dengan menggunakan persamaan (2.29):

$$net_j = \sum_{i=1}^n x_i W_{ji} + b_j, \quad j = 1, 2, 3, \dots, m \quad (2.29)$$

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

b. Perbaiki bobot pola yang mengandung kesalahan ( $y_j \neq t_j$ ) menurut persamaan (2.30) dan (2.31) berikut ini.

$$W_{ji} (\text{baru}) = W_{ji} (\text{lama}) + \alpha t_j x_i \quad (2.30)$$

$$b_j (\text{baru}) = b_j (\text{lama}) + \alpha t_j \quad (2.31)$$

c. Lakukan langkah 4a-b terus menerus hingga  $t_j = y_j$

Keterangan :  $net_j$  = respon *unit* keluaran ke- $j$ , dimana  $j = 1, 2, \dots, m$ .

$X_i$  = unit masukan ke- $i$ .

$W_{ij}$  = bobot pada unit yang ke- $i$  dan yang ke- $j$ .

$b_j$  = bias yang ke- $j$ , dimana  $j = 1, 2, \dots, m$ .

$y_j$  = unit keluaran ke- $j$ .

$\alpha$  = laju pemahaman.

$t_j$  = target ke- $j$ , dimana  $j = 1, 2, \dots, m$ .

## 2.6 Kajian Agama yang Terkandung dalam Jaringan Syaraf Tiruan

Allah berfirman dalam surat Al-An'am ayat 38 :

وَمَا مِنْ دَابَّةٍ فِي الْأَرْضِ وَلَا طَائِرٍ يَطِيرُ بِجَنَاحَيْهِ إِلَّا أُمَمٌ أَمْثَالُكُمْ ۚ مَا فَرَّطْنَا فِي  
الْكِتَابِ مِنْ شَيْءٍ ثُمَّ إِلَىٰ رَبِّهِمْ يُحْشَرُونَ ﴿٣٨﴾

Artinya: “dan Tiadalah binatang-binatang yang ada di bumi dan burung-burung yang terbang dengan kedua sayapnya, melainkan umat (juga) seperti kamu. Tiadalah Kami alpakan sesuatupun dalam Al-Kitab (Al-Qur'an), kemudian kepada Tuhanlah mereka dihimpunkan.” (Q.S. Al-An'am [6] : 38).

Dalam ayat di atas dijelaskan bahwasanya semua yang ada di alam ini tercantum dalam Al-Qur'an, tidak satu perkarapun yang luput dari Al-Qur'an. Lebih lanjut Achmad Baiquni mengatakan, “sebenarnya segala ilmu yang diperlukan manusia itu tersedia di dalam Al-Qur'an” (Baiquni, 1997).

Merujuk dari penjelasan di atas, jaringan syaraf tiruan juga tidak luput dari Al-Qur'an. Dalam jaringan syaraf tiruan, pertama kali yang dilakukan adalah *training* (pembelajaran), pada proses ini jaringan di-*training* terhadap pola-pola karakter untuk mendapatkan bobot-bobot yang sesuai, proses pembelajaran ini dilakukan berulang-ulang dengan mengubah nilai bobotnya sampai menghasilkan keluaran yang diinginkan.

Islam juga menganjurkan untuk belajar, bahkan begitu pentingnya belajar sehingga Allah menurunkan ayat pertama kali kepada Nabi Muhammad SAW yaitu surat Al-Alaq ayat 1 yang berbunyi :

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ ﴿١﴾

*Artinya: “bacalah dengan (menyebut) nama Tuhanmu yang Menciptakan” (Q.S. Al-Alaq [96] : 1).*

Menurut Quraish shihab dalam ayat di atas mengandung ajakan untuk membaca dan belajar, dan bahwa Tuhan Yang mampu menciptakan manusia dari asal yang lemah akan mampu pula untuk mengajarkannya menulis dan mengajarkannya sesuatu yang belum pernah diketahuinya (Anonim, <http://beta.quran.com/id/96/1-19/#1/>).



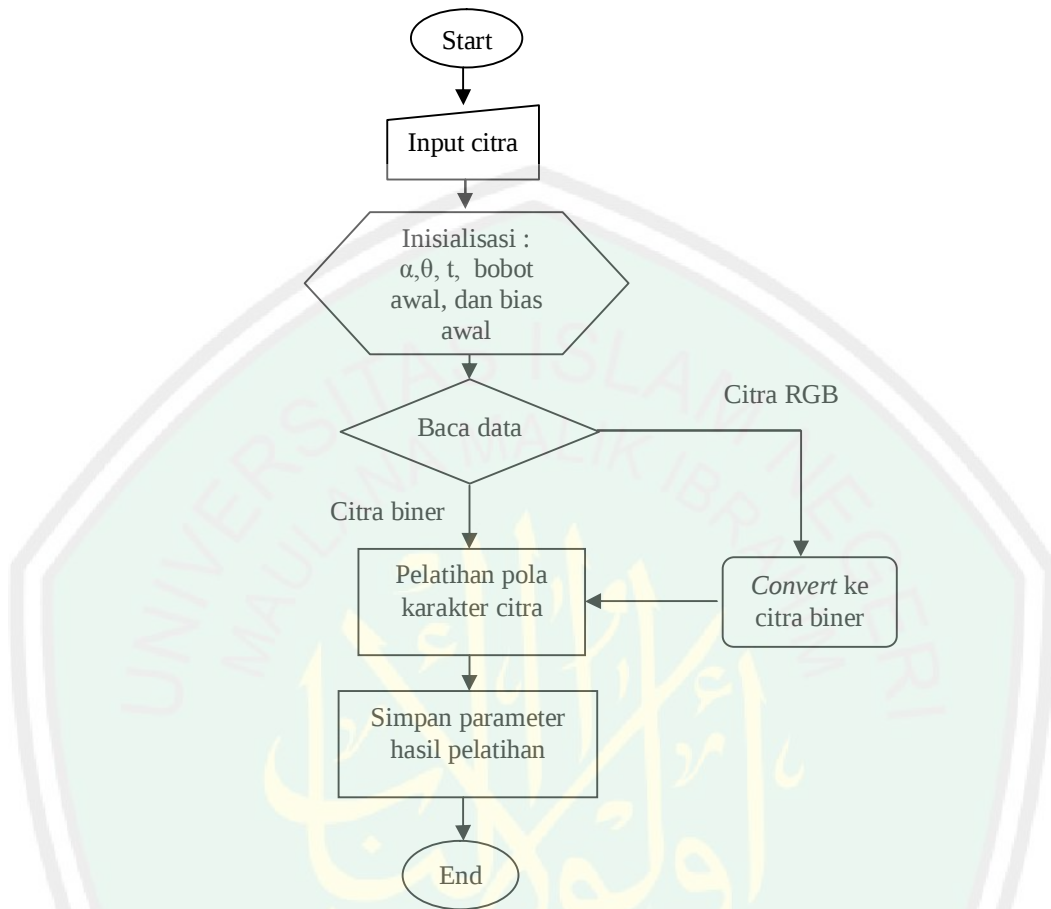
## BAB III

### PEMBAHASAN

#### 3.1 Deskripsi Sistem

Pada subbab ini akan membahas mengenai deskripsi sistem yang dikerjakan pada penelitian ini. Tujuan pembuatan sistem ini adalah untuk mengenali karakter alfanumerik (A-Z, a-z, 0-9). Pada awalnya pengguna memasukkan input data berupa citra (gambar). Citra masukan adalah citra *grayscale* 8 bit atau citra warna (RGB). Kemudian pengguna diminta untuk melakukan *preprocessing* yaitu proses awal agar citra tersebut dapat diproses selanjutnya. Jika semua operasi pada *preprocessing* telah dikerjakan, maka sistem siap melakukan proses selanjutnya.

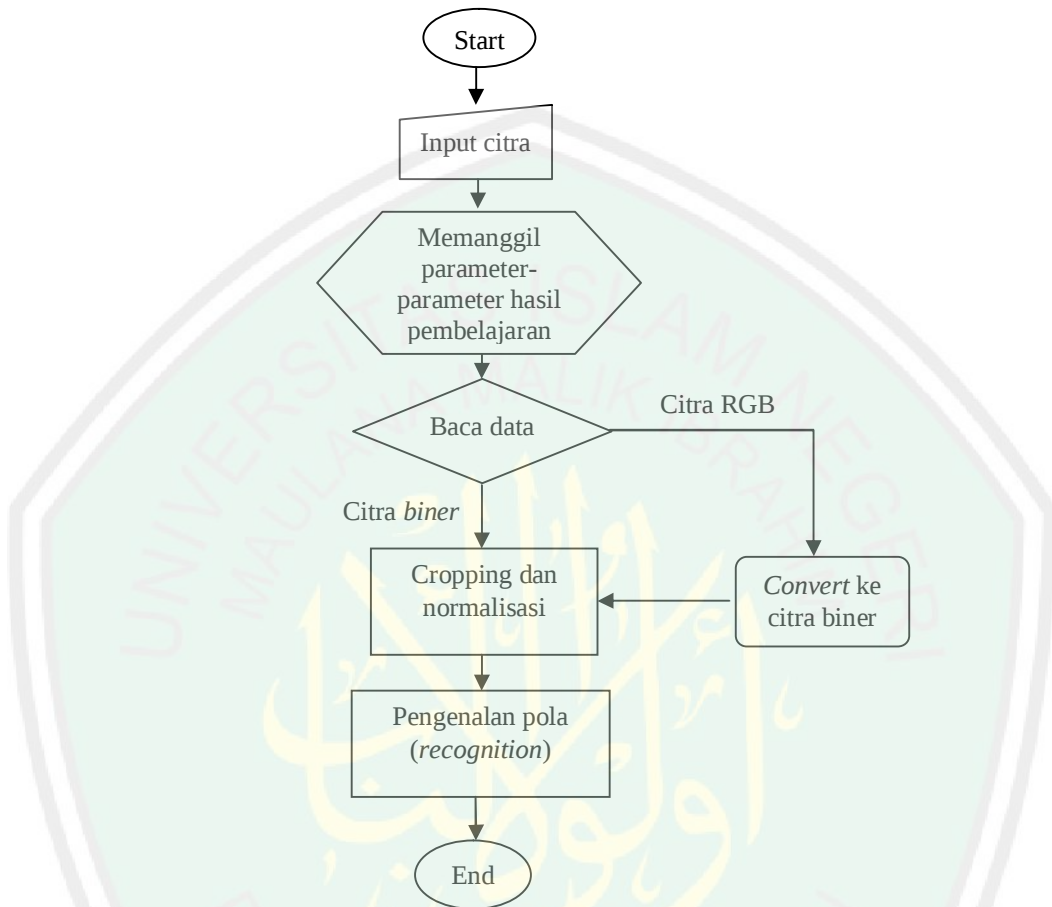
Implementasi metode pengenalan karakter ini terdiri dari 2 tahap utama yaitu tahap proses pelatihan pola (*process training*), dan tahap pengenalan (*recognition*). Pada tahap *training*, citra *input* yang akan di-*training* terlebih dahulu diproses dengan pengolahan citra. Diagram alir dari sistem ini adalah sebagai berikut:



Gambar 3. 1 Diagram Alir Sistem Pelatihan Pola Karakter

Sesuai dengan gambar 3.1, jika pengguna tidak memasukkan citra *biner* (8bit) maka citra akan diubah terlebih dahulu menjadi citra *biner*. Selanjutnya melakukan proses pembelajaran untuk mendapatkan bobot-bobot yang sesuai. Hasil dari proses ini kemudian disimpan untuk dipakai dalam proses selanjutnya yaitu pengenalan pola.

Kemudian pada tahap kedua, sistem dibuat untuk melakukan pengenalan terhadap pola citra masukan, digram alir dari sistem tahap dua sebagai berikut:



Gambar 3. 2 Diagram Alir Sistem Pengenalan Pola

Pada gambar 3.2, setelah data dimasukkan selanjutnya dilakukan *cropping* terhadap objek yang akan dilakukan pengenalan, dan setelah itu dilakukan normalisasi dengan penskalaan yang dibandingkan dengan ukuran citra normal yang diinginkan. Sistem pengenalan pola ini menggunakan program 6.5, program perhitungannya dapat dilihat pada lampiran 2.

### 3.2 Deskripsi Data

Data yang digunakan dalam penelitian ini dibagi menjadi dua yaitu data citra (gambar) pola karakter yang nantinya dipakai jaringan dalam proses pembelajaran dan data citra yang akan digunakan sebagai uji coba sistem.

#### 1.2.1 Data Training

Data yang digunakan sebagai pola *input* untuk proses *training* adalah berupa gambar pola-pola karakter alfanumerik dari A sampai Z, a sampai z, dan 0 sampai 9, dan dari tiap pola karakter diambil 5 jenis tipe *font* (huruf) yaitu *Times New Romans*, *Arial*, *Tahoma*, *Courier New*, dan *Calibri*, sehingga jumlahnya ada 310 pola gambar. Tiap satu *file* gambar terdapat satu pola karakter.

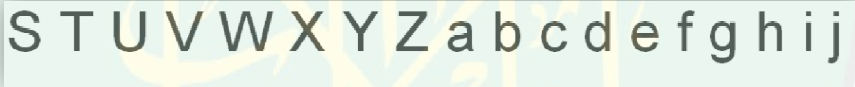
Data-data *training* ini dibuat dengan menggunakan program *windows paint*, dengan latar berwarna putih dan objeknya berwarna hitam, sedang ukuran fontnya 50.

#### 1.2.2 Data Uji Coba

Data yang digunakan sebagai *input* dalam proses uji coba adalah berupa gambar dari beberapa karakter yang dibuat dalam *windows paint*, dengan latar berwarna putih dan objeknya berwarna hitam, ukuran fontnya dibuat 50. Kemudian disimpan dalam format \*.bmp. Pada penelitian ini, data gambar yang digunakan dalam uji coba sebanyak 16 gambar.

Data uji coba dapat dilihat pada tabel 3.1 berikut ini.

Tabel 3.1 Bahan Uji Coba

| No. | Gambar bahan uji coba                                                                |
|-----|--------------------------------------------------------------------------------------|
| 1.  |    |
| 2.  |    |
| 3.  |    |
| 4.  |    |
| 5.  |  |
| 6.  |  |
| 7.  |   |
| 8.  |  |
| 9.  |  |
| 10. |   |

|     |                                     |
|-----|-------------------------------------|
| 11. | ABCDE FGHI JKL MNOPQRS TU V W X Y Z |
| 12. | abcdefghijklmnopqrstuvwxyz          |
| 13. | 0123456789                          |
| 14. | ABCDEFGHIJKLMNOPQRSTUVWXYZ          |
| 15. | abcdefghijklmnopqrstuvwxyz          |
| 16. | 0123456789                          |

### 3.3 Proses Pengolahan Citra

#### 3.3.1 *Preprocessing*

Setelah data didapatkan, selanjutnya data-data tersebut diolah menjadi citra *biner*. Untuk mendapatkan citra *biner*, citra RGB diubah terlebih dahulu menjadi citra *grayscale*, dan kemudian menghitung nilai *threshold*-nya, sehingga didapatkan citra *biner*.

Seperti yang telah dijelaskan pada bab sebelumnya, citra RGB atau warna dapat dikatakan sebuah matriks di ruang 3 dimensi, misalkan sebuah gambar berukuran  $M \times N$  piksel, maka didapatkan 3 buah matriks  $R$ ,  $G$ ,  $B$  sebagai berikut:

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} & \dots & R_{1N} \\ R_{21} & R_{22} & R_{23} & \dots & R_{2N} \\ R_{31} & R_{32} & R_{33} & \dots & R_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{M1} & R_{M2} & R_{M3} & \dots & R_{MN} \end{bmatrix}$$

$$G = \begin{bmatrix} G_{11} & G_{12} & G_{13} & \dots & G_{1N} \\ G_{21} & G_{22} & G_{23} & \dots & G_{2N} \\ G_{31} & G_{32} & G_{33} & \dots & G_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ G_{M1} & G_{M2} & G_{M3} & \dots & G_{MN} \end{bmatrix}$$

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} & \dots & B_{1N} \\ B_{21} & B_{22} & B_{23} & \dots & B_{2N} \\ B_{31} & B_{32} & B_{33} & \dots & B_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ B_{M1} & B_{M2} & B_{M3} & \dots & B_{MN} \end{bmatrix}$$

Tiap piksel dalam gambar adalah vektor di ruang 3 dimensi yang ditentukan oleh 3 koordinat dari matriks  $R$ ,  $G$ , dan  $B$ . Misalkan piksel ke-1 berwarna putih, dapat dituliskan:

$$RGB_{11} (R_{11} \ G_{11} \ B_{11}) = RGB_{11} (255 \ 255 \ 255)$$

Proses *preprocessing* ini akan dijelaskan dengan langkah-langkah berikut.

**Langkah I** merubah citra RGB menjadi citra *grayscale*

Untuk merubah dari citra RGB ke citra *grayscale* menggunakan persamaan (2.9), yaitu

$$grayscale = \frac{R + G + B}{3}$$

Operasi ini dilakukan pada tiap piksel, dengan mengambil elemen dari matriks  $R$ ,  $G$ , dan  $B$ . maka didapatkan:

$$gray_{11} = \frac{255+255+255}{3} = 255$$

Sehingga didapatkan sebuah matriks baru, yaitu

$$gray = \begin{bmatrix} gray_{11} & gray_{12} & gray_{13} & \dots & gray_{1N} \\ gray_{21} & gray_{22} & gray_{23} & \dots & gray_{2N} \\ gray_{31} & gray_{32} & gray_{33} & \dots & gray_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ gray_{M1} & gray_{M2} & gray_{M3} & \dots & gray_{MN} \end{bmatrix}$$

**Langkah II** merubah citra *grayscale* menjadi citra *biner*

Selanjutnya citra *grayscale* diubah menjadi citra *biner* menggunakan pengambangan atau *threshold*, nilai dari *threshold* ditentukan sebelumnya, pada penelitian ini penulis memberikan nilai *threshold* yaitu bernilai 90. Jika nilainya dibawah atau sama dengan *threshold*, maka nilainya menjadi 0, jika sebaliknya maka nilainya 1. Misalkan dari contoh di atas diketahui:

$$gray_{11} = 255$$

jika nilai *threshold* ditentukan 90, sehingga  $255 > 90$ , maka

$$biner_{11} = 1$$

dari proses ini, maka citra *biner* didefinisikan menjadi sebuah matriks sebagai berikut.

$$biner = \begin{bmatrix} biner_{11} & biner_{12} & biner_{13} & \dots & biner_{1N} \\ biner_{21} & biner_{22} & biner_{23} & \dots & biner_{2N} \\ biner_{31} & biner_{32} & biner_{33} & \dots & biner_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ biner_{M1} & biner_{M2} & biner_{M3} & \dots & biner_{MN} \end{bmatrix}$$

### 3.3.2 Proses Segmentasi Citra

Proses ini bertujuan untuk mendapatkan objek yang diperlukan untuk penelitian dan membuang yang tidak diperlukan. Segmentasi citra yang digunakan adalah deteksi tepi, *cropping*, dan normalisasi. Adapun langkah-langkahnya adalah sebagai berikut:

### Langkah I

Proses awal segmentasi diawali dengan mendeteksi tepi dengan cara *edge detection* (deteksi tepi), proses deteksi tepi ini bertujuan untuk mencari titik-titik tepi dari karakter, dengan menggunakan persamaan (2.16) dan (2.17), yaitu :

$$\Delta M = A(i, j) - A(i, j + 1)$$

$$\Delta N = A(i, j) - A(i - 1, j)$$

dengan melihat hubungan antar piksel tetangganya sehingga didapatkan titik  $X_L$  (tepi kiri),  $X_R$  (tepi kanan),  $Y_T$  (tepi atas), dan  $Y_B$  (tepi bawah).

### Langkah II

Selanjutnya, setelah titik-titik tepinya telah diketahui proses selanjutnya adalah *cropping*, proses ini bertujuan untuk memperoleh bagian yang akan digunakan dalam penelitian. Proses *cropping* menggunakan persamaan (2.5) dan (2.6), dari matriks *biner* yang didapatkan dari proses *preprocessing* yang telah dijelaskan sebelumnya, maka hasil *cropping* dari citra *biner* adalah:

$$crop = \begin{bmatrix} biner_{Y_T X_L} & biner_{Y_T X_L+1} & biner_{Y_T X_L+2} & \dots & biner_{Y_T X_R} \\ biner_{Y_T+1 X_L} & biner_{Y_T+1 X_L+1} & biner_{Y_T+1 X_L+2} & \dots & biner_{Y_T+1 X_R} \\ biner_{Y_T+2 X_L} & biner_{Y_T+2 X_L+1} & biner_{Y_T+2 X_L+2} & \dots & biner_{Y_T+2 X_R} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ biner_{Y_B X_L} & biner_{Y_B X_L+1} & biner_{Y_B X_L+2} & \dots & biner_{Y_B X_R} \end{bmatrix}$$

Dari hasil di atas dapat dilihat bahwa citra *biner* yang semula berukuran  $M \times N$  berubah menjadi citra yang berukuran  $Y_B \times X_R$ .

### Langkah III

Untuk merubah ukuran citra hasil *cropping* menjadi ukuran yang diinginkan atau diperlukan untuk penelitian, maka dilakukanlah proses

normalisasi dengan cara membandingkan ukuran citra hasil *cropping* dengan ukuran citra normal (untuk mendapatkan nilai  $S_N$  dan  $S_M$ ), setelah  $S_N$  dan  $S_M$  diketahui, selanjutnya dapat menggunakan persamaan (2.14) dan (2.15). Dalam penelitian ini, ukuran citra normal yang diinginkan adalah 20x20, jadi:

$$S_N = X_R : 20$$

$$S_M = Y_B : 20$$

Jika variabel penskalaan bernilai lebih besar dari 1 maka hasil penskalaannya akan memperbesar ukuran citra, sebaliknya apabila variabel penskalaannya lebih kecil dari 1 maka hasilnya akan memperkecil ukuran citra. Penskalaan ini tergantung besar dan kecil ukuran pada citra semula, artinya tidak berarti apakah citra membesar atau mengecil tergantung ukuran citra semula.

Hasil dari proses ini didapatkan matriks 20x20 sebagai berikut.

$$normal = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1\ 20} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2\ 20} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3\ 20} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{20\ 1} & x_{20\ 2} & x_{20\ 3} & \dots & x_{20\ 20} \end{bmatrix}$$

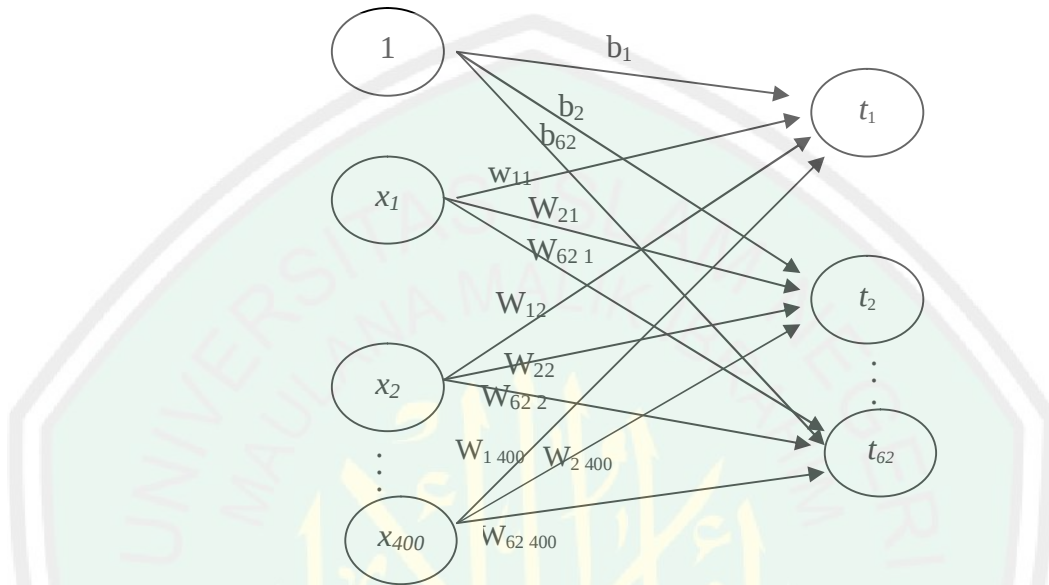
### 3.4 Proses Jaringan Syaraf Tiruan

#### 3.4.1 Menentukan Arsitektur Jaringan *Perceptron*

Proses ini bertujuan untuk menentukan arsitektur jaringan *perceptron* yang akan digunakan, arsitektur jaringan ini akan menentukan keberhasilan penelitian, karena itu arsitektur jaringan harus benar-benar tepat sesuai dengan permasalahan yang akan dijawab.

Arsitektur jaringan yang digunakan pada penelitian ini yaitu terdiri dari 400 elemen *input* sebanyak 310 vektor, dan target jaringan sebanyak 62 vektor,

sedangkan biasanya hanya sebuah vektor dengan 62 elemen. Arsitektur jaringan *perceptron* dapat dilihat pada gambar 3.3 berikut ini.



Gambar 3.3 Arsitektur Jaringan *Perceptron*

Pada gambar 3.3 dapat dilihat bahwa setiap *input* ( $x_i$ ) dihubungkan dengan setiap target ( $t_j$ ) dengan bobot ( $W_{ji}$ ), bobot  $W$  dapat dituliskan sebagai matriks berikut.

$$W = \begin{bmatrix} W_{11} & W_{12} & W_{13} & \dots & W_{1 400} \\ W_{21} & W_{22} & W_{23} & \dots & W_{2 400} \\ W_{31} & W_{32} & W_{33} & \dots & W_{3 400} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_{62 1} & W_{62 2} & W_{62 3} & \dots & W_{62 400} \end{bmatrix}$$

### 3.4.2 Proses *Training*

Proses *training* ini bertujuan untuk mencari bobot dan bias pada setiap pasangan pola *input* ( $X$ ) dengan target ( $t$ ) yang sesuai. Pasangan pola *input* ( $X$ ) dan target ( $t$ ) dapat dilihat pada lampiran 1.

Proses *training* mengikuti langkah-langkah algoritma pembelajaran beberapa pola seperti yang telah dijelaskan pada bab 2. Berikut langkah-langkahnya:

1. Nyatakan tiap pola masukan sebagai vektor *biner* yang elemennya adalah tiap titik pada matriks normal tersebut.

Untuk membuat vektor *biner* dari matriks normal adalah dengan menyusun deretan baris dari elemen matriks normal, disusun dari kiri ke kanan, dan dimulai dari baris pertama, kedua dan seterusnya.

Misal diketahui hasil citra normalisasi di atas adalah

$$normal = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1\ 20} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2\ 20} \\ x_{31} & x_{32} & x_{33} & \dots & x_{3\ 20} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{20\ 1} & x_{20\ 2} & x_{20\ 3} & \dots & x_{20\ 20} \end{bmatrix}, \text{ maka vektor } \textit{biner} \text{ untuk}$$

matriks normal adalah :

$$[x_{11} \ x_{12} \ x_{13} \ x_{14} \ \dots \ x_{1\ 20} \ x_{21} \ x_{22} \ x_{23} \ \dots \ x_{20\ 1} \ x_{20\ 2} \ x_{20\ 3} \ \dots \ x_{20\ 20}]$$

Vektor inilah yang akan dijadikan *input* (*X*) jaringan. Pada penelitian ini, *input* jaringan sebanyak 310 vektor dan tiap vektor terdiri dari 400 elemen.

2. Berikan nilai target  $t_j = +1$  jika pola masukan menyerupai huruf yang diinginkan. Jika sebaliknya, berikan nilai target  $= t_j = -1$ .

Jaringan syaraf tiruan *perceptron* ini termasuk dalam pembelajaran dengan pengawasan, artinya jaringan ini belajar dari contoh, karena itu penulis memberikan contoh keluaran dari jaringan yang disebut dengan target (*t*).

Target (*t*) terdiri dari 62 vektor, maksudnya adalah vektor pertama untuk karakter A, vektor ke-2 untuk karakter B, vektor ke-3 untuk karakter C, dan

seterusnya sampai vektor ke-62 untuk karakter angka 9. Setiap vektor target memiliki 310 elemen dan nilainya adalah bipolar (bernilai 1 atau -1). Bernilai 1 untuk urutan yang mewakili karakter tersebut, dan -1 untuk elemen yang lainnya. Contohnya misalkan pola karakter A berarti vektor targetnya bernilai 1 untuk elemen pertama, dan bernilai -1 untuk elemen kedua, ketiga, dan seterusnya. Untuk lebih jelasnya dapat dilihat pada lampiran 1.

3. Berikan inisialisasi bobot, bobot bias, laju pemahaman dan *threshold*.

Pada langkah ini peneliti memberikan nilai bobot awal, bobot bias, laju pemahaman ( $\alpha$ ) dan *threshold* ( $\theta$ ), biasanya untuk mempermudah penelitian, laju pemahaman = 1. Sedangkan untuk *threshold* dalam penelitian ini penulis memberikan nilai 0,5. Dan iterasi maksimum (*epoch*) adalah 100 kali.

Bobot bias =  $b = [b_1 \quad b_2 \quad b_3 \quad \dots \quad b_{62}]$

$$\text{Bobot} = W = \begin{bmatrix} W_{11} & W_{12} & W_{13} & \dots & W_{1\ 400} \\ W_{21} & W_{22} & W_{23} & \dots & W_{2\ 400} \\ W_{31} & W_{32} & W_{33} & \dots & W_{3\ 400} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_{62\ 1} & W_{62\ 2} & W_{62\ 3} & \dots & W_{62\ 400} \end{bmatrix}$$

4. Menghitung nilai keluaran untuk setiap pasangan *input* ( $X$ ) dan target ( $t$ ) yang sesuai.

Hitung nilai keluaran ke- $j$  menggunakan persamaan (2.29):

$$net_j = \sum_{i=1}^{400} x_i W_{ji} + b_j, \quad j = 1, 2, 3, \dots, 62$$

$$net_j = [x_1 \quad x_2 \quad x_3 \quad \dots \quad x_{400}] \begin{bmatrix} W_{j1} \\ W_{j2} \\ \vdots \\ W_{j400} \end{bmatrix} + b_j,$$

$$net_j = k$$

$$y_j = f(\text{net}_j) = \begin{cases} 1, & \text{jika } \text{net}_j > \theta \\ 0, & \text{jika } -\theta \leq \text{net}_j \leq \theta \\ -1, & \text{jika } \text{net}_j < -\theta \end{cases}$$

Jika  $y_j \neq t_j$ , maka bobot ( $W_{ji}$ ) dan bobot bias ( $b_j$ ) diperbarui dengan menggunakan persamaan (2.30) dan (2.31):

$$W_{ji}(\text{baru}) = W_{ji}(\text{lama}) + \alpha t_j x_i$$

$$b_j(\text{baru}) = b_j(\text{lama}) + \alpha t_j$$

Setelah bobot-bobotnya diperbarui, lakukan perhitungan yang di atas untuk pasangan *input* ( $X$ ) dan target ( $t$ ) berikutnya dengan menggunakan  $W_{ji}$  (baru) dan  $b_j$  (baru). Lakukan terus untuk semua pasangan *input* ( $X$ ) dan target ( $t$ ) sampai tidak ada perubahan bobot lagi, jika sampai *epoch* maksimum bobot tetap mengalami perubahan, maka pelatihan dihentikan dan bobot dipakai apa adanya. Kemudian lakukan lagi perhitungan seperti di atas untuk  $y_j$  selanjutnya.

Bobot-bobot inilah yang nanti akan digunakan untuk pengenalan karakter.

### 3.4.3 Proses *Testing* (Uji Coba)

Setelah dilakukan proses *training*, maka proses selanjutnya adalah proses *testing*. Proses ini dilakukan pada saat proses *training* berakhir dan data dari proses *training* akan diuji dan diterapkan. Menerapkannya dilakukan dengan bobot yang dihasilkan dari algoritma *training*.

Untuk proses *testing*, langkah-langkahnya adalah sebagai berikut:

1. Lakukan proses pengolahan citra pada gambar yang akan diuji seperti yang telah dijelaskan di atas sehingga nanti akan didapatkan matriks hasil normalisasi.
2. Setelah itu, matrik hasil normalisasi dijadikan vektor *biner* seperti pada langkah 1 pada proses training.
3. Hitung nilai keluaran ke-*j* jaringan kemudian bandingkan dengan nilai target ke-*j*.

Hitung nilai keluaran dengan persamaan :

$$net_j = \sum_{i=1}^{400} x_i W_{ji} + b_j, \quad j = 1, 2, 3, \dots, 62$$

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

Misalkan vektor binernya adalah  $X = [x_1 \ x_2 \ x_3 \ \dots \ x_{400}]$ , bobot

$$W = \begin{bmatrix} W_{11} & W_{12} & W_{13} & \dots & W_{1\ 400} \\ W_{21} & W_{22} & W_{23} & \dots & W_{2\ 400} \\ W_{31} & W_{32} & W_{33} & \dots & W_{3\ 400} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_{62\ 1} & W_{62\ 2} & W_{62\ 3} & \dots & W_{62\ 400} \end{bmatrix},$$

dan bobot bias  $b = [b_1 \ b_2 \ b_3 \ \dots \ b_{62}]$ , maka

$$net_j = [x_1 \ x_2 \ x_3 \ \dots \ x_{400}] \begin{bmatrix} W_{j1} \\ W_{j2} \\ \vdots \\ W_{j400} \end{bmatrix} + b_j,$$

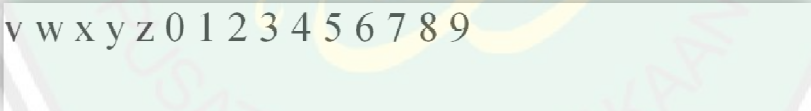
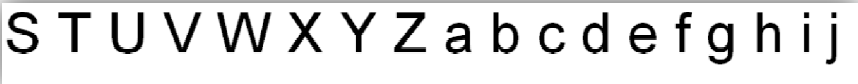
$$net_j = k$$

$$y_j = f (net_j) = \begin{cases} 1, & \text{jika } net_j > 0,5 \\ 0, & \text{jika } -0,5 \leq net_j \leq 0,5 \\ -1, & \text{jika } net_j < -0,5 \end{cases}$$

Jaringan dikatakan berhasil mengenal dengan baik jika nilai *output* jaringan ke-*j* sama dengan nilai target ke-*j* yang sesuai.

Dari uji coba yang sudah dilakukan pada 16 gambar, penulis mendapatkan hasil sebagai berikut :

Tabel 3.2 Hasil Uji Coba

| No. | Gambar-gambar bahan uji coba                                                         | Hasil uji coba                                         |
|-----|--------------------------------------------------------------------------------------|--------------------------------------------------------|
| 1.  |    | Jumlah karakter yang dikenali = 21<br>Jumlah error = 0 |
| 2.  |  | Jumlah karakter yang dikenali = 24<br>Jumlah error = 2 |
| 3.  |  | Jumlah karakter yang dikenali = 15<br>Jumlah error = 0 |
| 4.  |  | Jumlah karakter yang dikenali = 18<br>Jumlah error = 0 |
| 5.  |  | Jumlah karakter yang dikenali = 16<br>Jumlah error = 2 |
| 6.  |  | Jumlah karakter yang dikenali = 16<br>Jumlah error = 0 |

|     |                                       |                                                        |
|-----|---------------------------------------|--------------------------------------------------------|
| 7.  | 1234567890                            | Jumlah karakter yang dikenali = 10<br>Jumlah error = 0 |
| 8.  | ABCDEFGH I J KLMN O PQ RSTU V W X Y Z | Jumlah karakter yang dikenali = 26<br>Jumlah error = 0 |
| 9.  | abcdefghijklmnopqrstuvwxyz            | Jumlah karakter yang dikenali = 24<br>Jumlah error = 2 |
| 10. | 0123456789                            | Jumlah karakter yang dikenali = 10<br>Jumlah error = 0 |
| 11. | ABCDE FGHI JKL MNOPQRS TU V W X Y Z   | Jumlah karakter yang dikenali = 26<br>Jumlah error = 0 |
| 12. | abcdefghijklmnopqrstuvwxyz            | Jumlah karakter yang dikenali = 24<br>Jumlah error = 2 |
| 13. | 0123456789                            | Jumlah karakter yang dikenali = 10<br>Jumlah error = 0 |
| 14. | ABCDEFGH I J KLMNOPQRSTU VWXYZ        | Jumlah karakter yang dikenali = 26<br>Jumlah error = 0 |
| 15. | abcdefghijklmnopqrstuvwxyz            | Jumlah karakter yang dikenali = 24<br>Jumlah error = 2 |

|     |            |                                                        |
|-----|------------|--------------------------------------------------------|
| 16. | 0123456789 | Jumlah karakter yang dikenali = 10<br>Jumlah error = 0 |
|-----|------------|--------------------------------------------------------|

Pada penelitian ini persentase keberhasilan jaringan dalam mengenali karakter didapatkan dari hasil perhitungan menggunakan rumus sebagai berikut:

$$\text{Persentase} = \frac{\text{jumlah karakter yang dikenali}}{\text{jumlah karakter keseluruhan}} \times 100$$

Dari tabel 3.2 di atas didapatkan data sebagai berikut :

Jumlah karakter keseluruhan = 310

Jumlah karakter yang dikenali = 300

Jumlah error = 10

Sehingga persentase keberhasilan dari penelitian ini adalah

$$\text{Persentase} = \frac{300}{310} \times 100$$

$$\text{Persentase} = 96,5 \%$$

### 3.5 Jaringan Syaraf Tiruan dalam Pandangan Islam

Jaringan syaraf tiruan adalah paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem sel syaraf biologi, sama seperti otak yang memproses suatu informasi. Jaringan syaraf tiruan diharapkan dapat menyelesaikan permasalahan-permasalahan dalam skala yang besar dan kompleks.

Dalam islam kita dianjurkan untuk mempelajari dan meniru apa yang telah Allah ciptakan sebelumnya, seperti yang dijelaskan dalam surat Al-Ghasyiah ayat 17 berikut :

أَفَلَا يَنْظُرُونَ إِلَى الْإِبِلِ كَيْفَ خُلِقَتْ ﴿١٧﴾

Artinya : “Maka Apakah mereka tidak memperhatikan unta bagaimana Dia diciptakan?” (Q.S. Al-Ghasyiah[88] : 17).

dengan meneliti dan mempelajari apa yang telah Allah ciptakan sebelumnya kita juga dapat memperoleh pelajaran dan manfaat darinya, dan sesungguhnya apa yang diciptakan Allah semuanya pasti ada artinya. Seperti yang dijelaskan dalam Firman-Nya dalam surat Yunus ayat 101 :

قُلْ أَنْظُرُوا مَاذَا فِي السَّمَوَاتِ وَالْأَرْضِ وَمَا تُغْنِي الْآيَاتُ وَالنُّذُرُ عَنْ قَوْمٍ لَا يُؤْمِنُونَ ﴿١٠١﴾

Artinya: (Katakanlah) "Perhatikanlah apa yang ada di langit dan di bumi. tidaklah bermanfaat tanda kekuasaan Allah dan Rasul-rasul yang memberi peringatan bagi orang-orang yang tidak beriman" (Q.S. Yunus [10] : 101).

Allah telah membolehkan bahkan menyuruh manusia meneliti dengan nalar yang dalam dan jauh, manfaat yang dapat di ambil dari semua yang diciptakan Allah baik di langit maupun di bumi. Sebab, setiap ciptaan Allah bukan tidak ada artinya. Allah menciptakan semua benda yang ada di alam ini semuanya mempunyai arti yang luas bagi kehidupan manusia sekarang dan masa depan.

Perintah untuk menelaah dan meneliti ini juga di tuangkan pada ayat pertama yang diturunkan kepada Nabi Muhammad SAW, yaitu perintah untuk membaca (*iqra'*) yang menurut Quraish shihab, kata ini terambil dari akar kata *qara'a* yang berarti menghimpun. Dari menghimpun lahir aneka makna seperti menyampaikan, menelaah, mendalami, meneliti, mengetahui ciri sesuatu, dan membaca baik teks yang tertulis maupun tidak (Husna, 2011).

Menurut Harun yahya dalam bukunya yang berjudul “menyingkap rahasia alam semesta” : dalam Al-Qur’an, manusia diseru untuk merenungi berbagai kejadian dan benda alam, yang dengan jelas memberikan kesaksian akan keberadaan dan keesaan Allah beserta sifat-sifat-Nya. Dalam Al-Qur’an, segala sesuatu yang memberikan kesaksian ini disebut “tanda-tanda”, yang berarti “bukti yang teruji kebenarannya, pengetahuan mutlak, dan pernyataan kebenaran.” Jadi, tanda-tanda kebesaran Allah terdiri dari atas segala sesuatu di alam semesta ini yang memperlihatkan dan menyampaikan keberadaan dan sifat-sifat Allah. Orang-orang yang dapat mengamati dan senantiasa ingat akan hal ini akan memahami bahwa seluruh jagat raya tersusun hanya dari tanda-tanda kebesaran Allah.

Sungguh, adalah kewajiban bagi manusia untuk dapat melihat tanda-tanda kebesaran Allah. Dengan demikian, orang tersebut akan mengenal Sang Pencipta yang menciptakan dirinya dan segala sesuatu yang lain, menjadi lebih dekat kepada-Nya, menemukan makna keberadaan dan hidupnya, dan menjadi orang yang beruntung dunia akhirat. (Yahya, 2002)

Dari penelitian jaringan syaraf tiruan ini kita bisa mendapatkan pelajaran darinya. Dapat dibayangkan bagaimana repotnya orang di zaman dulu sebelum adanya jaringan syaraf tiruan ini, data yang banyak dan berupa gambar sulit untuk diolah dalam komputer, dan dengan terpaksa harus diolah dalam mode gambar pula. Setelah ada jaringan syaraf tiruan, data tersebut dapat dikenali sebagai teks dan komputer dapat mendeteksinya dengan baik, pekerjaan mereka sekarang jadi lebih mudah dan fleksibel. Ini menunjukkan betapa bermanfaatnya jika kita mau

meneliti dan mempelajari ciptaan Allah, dan manfaat itu hanya diperuntukkan bagi umat-Nya yang mau berfikir.



## BAB IV

### PENUTUP

#### 4.1 Kesimpulan

Dari uraian dan pembahasan pada bab sebelumnya, penulis memperoleh beberapa kesimpulan dari penelitian ini, yaitu :

1. Ada beberapa langkah untuk mengenali karakter pada gambar, yaitu:
  - a. Melakukan proses *binerisasi* dengan pengolahan citra, kemudian merepresentasikannya ke dalam matriks.
  - b. Memasukkan *input* jaringan dari hasil *binerisasi* sebagai *input* pelatihan *perceptron*.
  - c. Mencari bobot-bobot yang sesuai yang menghubungkan target dengan *input*.
  - d. Menghitung keluaran jaringan dengan menggunakan bobot-bobot hasil perhitungan sebelumnya.
2. Dengan menggunakan metode *perceptron* ini, tingkat keberhasilannya untuk mengenal pola sudah sangat baik. Ini dapat dilihat dari hasil uji coba yang penulis lakukan mendapatkan nilai persentase keberhasilan mencapai 96,5%.

#### 4.2 Saran

Berdasarkan hasil penelitian, penulis memberikan saran sebagai berikut:

1. Untuk mendapatkan hasil yang lebih baik lagi diharapkan pada proses pengolahan citranya menggunakan metode yang berbeda, sehingga dapat memperbaiki hasil ganda seperti dalam kasus karakter “i”.

2. Ada baiknya suatu saat dilakukan penelitian untuk membandingkan dengan metode-metode jaringan syaraf tiruan yang lain.



## DAFTAR PUSTAKA

- Anonim. 2012. *Al-Quran dan Terjemahan*. <http://beta.quran.com/id/96/1-19/#1/>. (diakses pada tanggal 16 Januari 2012).
- Anonim. 2005. *Jaringan Syaraf Tiruan, Temuan Matematikawan*. ITS online. <http://www.its.ac.id/berita.php?nomer=1662>. (diakses pada tanggal 16 Januari 2012).
- Balza, Ahmad & Firdausy Kartika. 2005. *Teknik Pengolahan Citra Menggunakan Delphi*. Yogyakarta: Ardi Publishing.
- Baiquni, Achmad. 1997. *Al-Qur'an dan Ilmu Pengetahuan Kealaman*. Yogyakarta: PT. Dana Bhakti Prima Yasa.
- Budhi, G.S. dkk. 2007. *Metode Jaringan Saraf Tiruan Backpropagation untuk Pengenalan Huruf Cetak pada Citra Digital*. [http://fportfolio.petra.ac.id/user\\_files/02-030/TE-12.DOC](http://fportfolio.petra.ac.id/user_files/02-030/TE-12.DOC). (diunduh pada tanggal 2 maret 2011)
- Fauzi, Achmad. 2009. *Perangkat Lunak Pengkonversi Teks Tulisan Tangan Manjadi Teks Digital*. Skripsi Tidak Diterbitkan. Surabaya: Jurusan Matematika FMIPA ITS.
- Husna, Nurul. 2011. *Tafsir 'Ilmi*. <http://nurulhusnayusuf-makalahku.blogspot.com> (diakses pada tanggal 14 Januari 2012)
- Siang, Jong Jek. 2004. *Jaringan Saraf Tiruan & Pemrogramannya Menggunakan Matlab*. Yogyakarta: Penerbit Andi.
- Kusumadewi, Sri. 2004. *Membangun Jaringan Syaraf Tiruan Menggunakan Matlab & Excel Link*. Yogyakarta: Penerbit Graha Ilmu.
- Kusumadewi, Sri. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Penerbit Graha Ilmu.
- Lipschutz, Seymour & Marc Lars Lipson. 2006. *Schaum's Outlines Teori dan Soal ALJABAR LINEAR Edisi Ketiga*. Jakarta: Penerbit Erlangga.
- Muis, Saludin. 2006. *Teknik Jaringan Syaraf Tiruan*. Yogyakarta: Penerbit Graha Ilmu.
- Munir, Rinaldi. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika.

Puspitaningrum, Diyah. 2006. *Pengantar Jaringan Saraf Tiruan*. Yogyakarta: Penerbit Andi.

Putra, Darma. 2010. *Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.

Yahya, Harun. 2002. *Menyingkap Rahasia Alam Semesta*. Bandung: Penerbit Dzikra.

Yani, Eli. 2005. *Artikel Pengantar Jaringan Syaraf Tiruan*. MateriKuliah.com. (diunduh pada tanggal 23 januari 2012).

Wulan, Sari, Zunita. 2010. *Pengenalan Pola Golongan Darah Menggunakan Jaringan Syaraf Tiruan Backpropagation*. Skripsi Tidak Diterbitkan. Malang: Jurusan Teknik Informatika Fakultas Saintek UIN Malang.



## Lampiran 1

Tabel Pasangan Pola dan target ( $t$ )

| No. | Pola | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ | $t_6$ | $t_7$ | $t_8$ | $t_9$ | $t_{10}$ | $t_{11}$ | ... | $t_{60}$ | $t_{61}$ | $t_{62}$ |
|-----|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|-----|----------|----------|----------|
| 1   | A    | 1     | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 2   | B    | -1    | 1     | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 3   | C    | -1    | -1    | 1     | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 4   | D    | -1    | -1    | -1    | 1     | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 5   | E    | -1    | -1    | -1    | -1    | 1     | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 6   | F    | -1    | -1    | -1    | -1    | -1    | 1     | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 7   | G    | -1    | -1    | -1    | -1    | -1    | -1    | 1     | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 8   | H    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | 1     | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 9   | I    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | 1     | -1       | -1       | ... | -1       | -1       | -1       |
| 10  | J    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | 1        | -1       | ... | -1       | -1       | -1       |
| 11  | K    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | 1        | ... | -1       | -1       | -1       |
| 12  | L    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 13  | M    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 14  | N    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 15  | O    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 16  | P    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 17  | Q    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 18  | R    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 19  | S    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 20  | T    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 21  | U    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 22  | V    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 23  | W    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 24  | X    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 25  | Y    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 26  | Z    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 27  | a    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 28  | b    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 29  | c    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 30  | d    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 31  | e    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 32  | f    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 33  | g    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 34  | h    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 35  | i    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 36  | j    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 37  | k    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 38  | l    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |
| 39  | m    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1    | -1       | -1       | ... | -1       | -1       | -1       |

|     |     |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |
|-----|-----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|
| 40  | n   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 41  | o   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 42  | p   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 43  | q   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 44  | r   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 45  | s   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 46  | t   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 47  | u   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 48  | v   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 49  | w   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 50  | x   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 51  | y   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 52  | z   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 53  | "0" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 54  | "1" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 55  | "2" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 56  | "3" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 57  | "4" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 58  | "5" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 59  | "6" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 60  | "7" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | 1  | -1 | -1 |
| 61  | "8" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | 1  | -1 |
| 62  | "9" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | 1  |
| 63  | A   | 1  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 64  | B   | -1 | 1  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 65  | C   | -1 | -1 | 1  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 66  | D   | -1 | -1 | -1 | 1  | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 67  | E   | -1 | -1 | -1 | -1 | 1  | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 68  | F   | -1 | -1 | -1 | -1 | -1 | 1  | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 69  | G   | -1 | -1 | -1 | -1 | -1 | -1 | 1  | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 70  | H   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1  | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 71  | I   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1  | -1 | -1 | ... | -1 | -1 | -1 |
| 72  | J   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1  | -1 | ... | -1 | -1 | -1 |
| 73  | K   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1  | ... | -1 | -1 | -1 |
| 74  | L   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 75  | M   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| 76  | N   | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | -1 |
| .   | .   | .  | .  | .  | .  | .  | .  | .  | .  | .  | .  | .  | .   | .  | .  | .  |
| .   | .   | .  | .  | .  | .  | .  | .  | .  | .  | .  | .  | .  | .   | .  | .  | .  |
| .   | .   | .  | .  | .  | .  | .  | .  | .  | .  | .  | .  | .  | .   | .  | .  | .  |
| 308 | "7" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | 1  | -1 | -1 |
| 309 | "8" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | 1  | -1 |
| 310 | "9" | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | ... | -1 | -1 | 1  |

## Lampiran 2

### PROGRAM MATLAB PENGENALAN KARAKTER PADA GAMBAR

```
[namafile,direktori] =  
uigetfile({'*.bmp'; '*.jpg'; '*.png'; '*.tif'}, 'Buka Gambar');  
if isequal(namafile,0)  
return;  
end  
gambar = imread (namafile);  
imshow(gambar);  
  
tetha = 0.5;  
bias=importdata('Vbias.dat');  
Mbobot=importdata('Mbobot.dat');  
[baris, kolom]=size(Mbobot);  
  
Mtarget1=-(ones(62));  
for en=1:62  
    Mtarget1(en,en)=1;  
end  
Mtarget2=-(ones(62));  
for en=1:62  
    Mtarget2(en,en)=1;  
end  
Mtarget3=-(ones(62));  
for en=1:62  
    Mtarget3(en,en)=1;  
end  
Mtarget4=-(ones(62));  
for en=1:62  
    Mtarget4(en,en)=1;  
end  
Mtarget5=-(ones(62));  
for en=1:62  
    Mtarget5(en,en)=1;  
end  
  
Mtarget=[Mtarget1; Mtarget2; Mtarget3; Mtarget4; Mtarget5];  
[row, coulom]=size(Mtarget);  
  
gray = rgb2gray(gambar); %merubah citra RGB ke citra hitam-  
putih  
biner=gray>90;  
  
d= edge(uint8(biner));  
  
se = strel('square',2);
```

```

e = imdilate(d, se);

f= imfill(e,'holes');

[Ilabel num] = bwlabel(f);

Iprops = regionprops(Ilabel);
Ibox = [Iprops.BoundingBox];
Ibox = reshape(Ibox,[4 num]);
figure(7), imshow(biner), title('segmentasi')

hold on

hold on;
for cnt = 1:num
    rectangle('position',Ibox(:,cnt),'edgecolor','r');
    crop=imcrop(biner, Ibox(:,cnt));
    normalisasi = imresize(crop, [20 20]);
    Vinput =
    [normalisasi(1,:),normalisasi(2,:),normalisasi(3,:),normalis
asi(4,:),normalisasi(5,:),normalisasi(6,:),normalisasi(7,:),
normalisasi(8,:), normalisasi(9,:), normalisasi(10,:),
normalisasi(11,:), normalisasi(12,:), normalisasi(13,:),
normalisasi(14,:), normalisasi(15,:), normalisasi(16,:),
normalisasi(17,:), normalisasi(18,:), normalisasi(19,:),
normalisasi(20,:)]];

    for je = 1:62
        net(je) = bias(je)+(Vinput*(Mbobot(je,:))');
        if net(je) > tetha
            output(je) = 1;
        elseif net(je) >= -tetha & net(je) <= tetha
            output(je) = 0;
        elseif net(je) < -tetha
            output(je) = -1;
        end
    end
    if output == Mtarget(1,:)
        hasil (cnt) =('A')
    elseif output == Mtarget(2,:)
        hasil (cnt) =('B')
    elseif output == Mtarget(3,:)
        hasil (cnt) =('C')
    elseif output == Mtarget(4,:)
        hasil (cnt) =('D')
    elseif output == Mtarget(5,:)
        hasil (cnt) =('E')
    elseif output == Mtarget(6,:)
        hasil (cnt) =('F')
    elseif output == Mtarget(7,:)
        hasil (cnt) =('G')
    end
end

```

```

elseif output == Mtarget(8,:)
    hasil (cnt) =('H')
elseif output == Mtarget(9,:)
    hasil (cnt) =('I')
elseif output == Mtarget(10,:)
    hasil (cnt) =('J')
elseif output == Mtarget(11,:)
    hasil (cnt) =('K')
elseif output == Mtarget(12,:)
    hasil (cnt) =('L')
elseif output == Mtarget(13,:)
    hasil (cnt) =('M')
elseif output == Mtarget(14,:)
    hasil (cnt) =('N')
elseif output == Mtarget(15,:)
    hasil (cnt) =('O')
elseif output == Mtarget(16,:)
    hasil (cnt) =('P')
elseif output == Mtarget(17,:)
    hasil (cnt) =('Q')
elseif output == Mtarget(18,:)
    hasil (cnt) =('R')
elseif output == Mtarget(19,:)
    hasil (cnt) =('S')
elseif output == Mtarget(20,:)
    hasil (cnt) =('T')
elseif output == Mtarget(21,:)
    hasil (cnt) =('U')
elseif output == Mtarget(22,:)
    hasil (cnt) =('V')
elseif output == Mtarget(23,:)
    hasil (cnt) =('W')
elseif output == Mtarget(24,:)
    hasil (cnt) =('X')
elseif output == Mtarget(25,:)
    hasil (cnt) =('Y')
elseif output == Mtarget(26,:)
    hasil (cnt) =('Z')
elseif output == Mtarget(27,:)
    hasil (cnt) =('a')
elseif output == Mtarget(28,:)
    hasil (cnt) =('b')
elseif output == Mtarget(29,:)
    hasil (cnt) =('c')
elseif output == Mtarget(30,:)
    hasil (cnt) =('d')
elseif output == Mtarget(31,:)
    hasil (cnt) =('e')
elseif output == Mtarget(32,:)
    hasil (cnt) =('f')
elseif output == Mtarget(33,:)

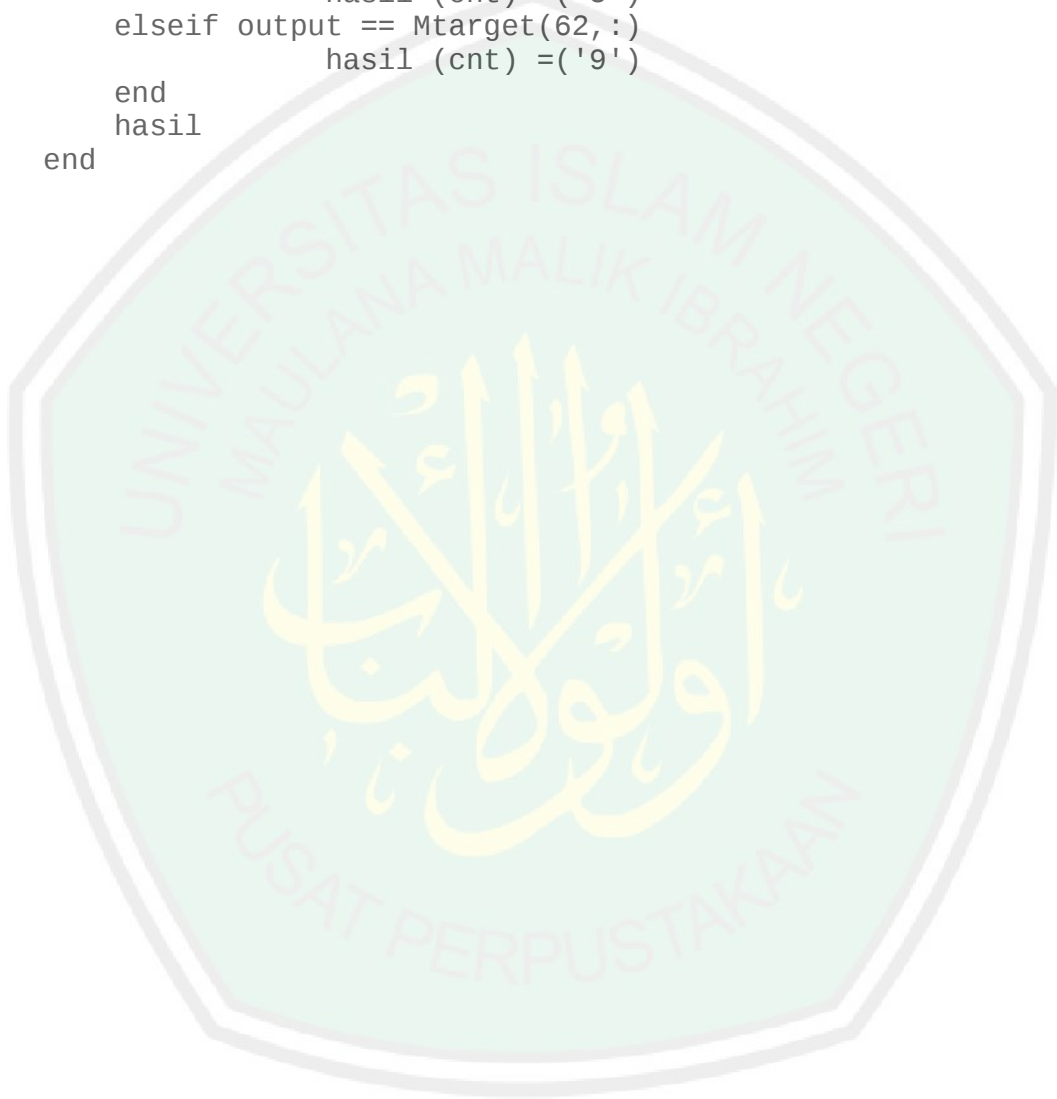
```

```

        hasil (cnt) =('g')
elseif output == Mtarget(34,:)
        hasil (cnt) =('h')
elseif output == Mtarget(35,:)
        hasil (cnt) =('i')
elseif output == Mtarget(36,:)
        hasil (cnt) =('j')
elseif output == Mtarget(37,:)
        hasil (cnt) =('k')
elseif output == Mtarget(38,:)
        hasil (cnt) =('l')
elseif output == Mtarget(39,:)
        hasil (cnt) =('m')
elseif output == Mtarget(40,:)
        hasil (cnt) =('n')
elseif output == Mtarget(41,:)
        hasil (cnt) =('o')
elseif output == Mtarget(42,:)
        hasil (cnt) =('p')
elseif output == Mtarget(43,:)
        hasil (cnt) =('q')
elseif output == Mtarget(44,:)
        hasil (cnt) =('r')
elseif output == Mtarget(45,:)
        hasil (cnt) =('s')
elseif output == Mtarget(46,:)
        hasil (cnt) =('t')
elseif output == Mtarget(47,:)
        hasil (cnt) =('u')
elseif output == Mtarget(48,:)
        hasil (cnt) =('v')
elseif output == Mtarget(49,:)
        hasil (cnt) =('w')
elseif output == Mtarget(50,:)
        hasil (cnt) =('x')
elseif output == Mtarget(51,:)
        hasil (cnt) =('y')
elseif output == Mtarget(52,:)
        hasil (cnt) =('z')
elseif output == Mtarget(53,:)
        hasil (cnt) =('0')
elseif output == Mtarget(54,:)
        hasil (cnt) =('1')
elseif output == Mtarget(55,:)
        hasil (cnt) =('2')
elseif output == Mtarget(56,:)
        hasil (cnt) =('3')
elseif output == Mtarget(57,:)
        hasil (cnt) =('4')
elseif output == Mtarget(58,:)
        hasil (cnt) =('5')

```

```
elseif output == Mtarget(59,:)
    hasil (cnt) =('6')
elseif output == Mtarget(60,:)
    hasil (cnt) =('7')
elseif output == Mtarget(61,:)
    hasil (cnt) =('8')
elseif output == Mtarget(62,:)
    hasil (cnt) =('9')
end
hasil
end
```





**KEMENTERIAN AGAMA RI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK  
IBRAHIM MALANG**

**FAKULTAS SAINS & TEKNOLOGI**

Jln. Gajayana No. 50 Malang Telp. (0341) 551354 Fax. (0341) 572533

**BUKTI KONSULTASI SKRIPSI**

**Nama** : Taufik Hidayatullah  
**NIM** : 06510014  
**Fakultas/ Jurusan** : Sains dan Teknologi/ Matematika  
**Judul Skripsi** : Terapan Jaringan Syaraf Tiruan Perceptron Untuk Pengenalan Karakter Pada Gambar  
**Pembimbing I** : Mohammad Jamhuri, M.Si  
**Pembimbing II** : Ach. Nashichuddin, M.Ag

| No | Tanggal           | Materi Konsultasi           | Tanda Tangan |     |
|----|-------------------|-----------------------------|--------------|-----|
| 1  | 08 Agustus 2011   | Konsultasi BAB I dan BAB II | 1.           |     |
| 2  | 15 September 2011 | Revisi BAB I dan BAB II     |              | 2.  |
| 3  | 22 Oktober 2011   | Revisi BAB II               | 3.           |     |
| 4  | 18 November 2011  | Konsultasi BAB III          |              | 4.  |
| 5  | 20 Desember 2011  | Konsultasi BAB III          | 5.           |     |
| 6  | 01 Januari 2012   | Revisi BAB III              |              | 6.  |
| 7  | 07 Januari 2012   | Konsultasi BAB IV           | 7.           |     |
| 8  | 09 Januari 2012   | Revisi BAB IV dan BAB V     |              | 8.  |
| 9  | 10 Januari 2012   | Konsultasi kajian agama     | 9.           |     |
| 10 | 13 Januari 2012   | Revisi kajian agama         |              | 10. |
| 11 | 13 Januari 2012   | ACC Keseluruhan             | 11.          |     |

Malang, 13 Januari 2012  
Mengetahui  
Ketua Jurusan Matematika

**Abdussakir, M.Pd**  
**NIP. 197510062003121001**