



**PERBANDINGAN ANTARA METODE
BACKPROPAGATION DENGAN METODE *LEARNING
VECTOR QUANTIZATION (LVQ)* PADA
PENGENALAN CITRA BARCODE**

skripsi
disajikan sebagai salah satu syarat
untuk memperoleh gelar Sarjana Sains
Program Studi Matematika

oleh
Muhamad Fithri Qomari Azizi
4111409021

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI SEMARANG**

2013

PERNYATAAN

Saya menyatakan bahwa skripsi ini bebas plagiat, dan apabila di kemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai ketentuan peraturan perundang-undangan.

Semarang, Agustus 2013

Muhamad Fithri Qomari Azizi

4111409021

PENGESAHAN

Skripsi yang berjudul

Perbandingan antara Metode *Backpropagation* dengan Metode *Learning Vector Quantization* (LVQ) pada Pengenalan Citra *Barcode*.

disusun oleh

Muhamad Fithri Qomari Azizi

4111409021

telah dipertahankan di hadapan sidang Panitia Ujian Skripsi FMIPA UNNES pada tanggal 4 September 2013

Panitia:

Ketua

Sekretaris

Prof. Dr. Wiyanto, M.Si.
19631012 198803 1 001

Drs. Arief Agoestanto, M.Si.
19680722 199303 1 005

Ketua Penguji

Riza Arifudin, S.Pd., M.Cs.
19800525 200501 1 001

Anggota Penguji/
Pembimbing Utama

Anggota Penguji/
Pembimbing Pendamping

Endang Sugiharti, S.Si., M.Kom.
19740107 199903 2 001

Much Aziz Muslim, S.Kom., M.Kom.
19740420 200812 1 001

MOTTO DAN PERSEMPAHAN

MOTTO

- ★ *Barangsiapa bersungguh-sungguh, sesungguhnya kesungguhan itu adalah untuk dirinya sendiri. (QS. Al-Ankabut :6)*
- ★ *Jika sore tiba, janganlah tunggu waktu pagi, jika pagi tiba, janganlah tunggu waktu sore. Manfaatkan masa sehatmu sebelum tiba masa sakitmu dan manfaatkan masa hidupmu sebelum tiba ajalmu. (Umar bin Khattab)*
- ★ *Banyak kegagalan dalam hidup ini dikarenakan orang-orang tidak menyadari betapa dekatnya mereka dengan keberhasilan saat mereka menyerah. (Thomas Alva Edison)*

PERSEMPAHAN

Skripsi ini ku persembahkan untuk,

- ♥ *Ibuku Siti Khasiatiah dan Bapakku Damari atas ketulusan doa, kasih sayang, semangat, nasihat, dan segalanya yang selama ini engkau berikan.*
- ♥ *Adikku Siti Nur Anisa Azizah Kamila atas doa dan dukungannya.*
- ♥ *Ratna Zahrotus Sania atas keikhlasan doa, kesetiaannya untuk menemaniku, menyemangati, canda tawanya, serta dukungannya.*
- ♥ *Teman-temanku seangkatan seperjuangan Matematika 2009 atas bantuan dan partisipasinya selama ini.*
- ♥ *Untuk sahabat-sahabatku satu kost yang selalu ada dalam suka dan duka, dalam perjuangan meraih asa, menuju jalan *Lillahi ta'ala*.*

PRAKATA

Puji syukur penulis haturkan ke hadirat Allah SWT atas segala rahmat dan hidayah-Nya yang selalu tercurah sehingga tersusunlah skripsi yang berjudul **“Perbandingan antara Metode *Backpropagation* dengan Metode *Learning Vector Quantization (LVQ)* pada Pengenalan Citra Barcode”**.

Penulis menyadari sepenuhnya bahwa skripsi ini selesai berkat bantuan, petunjuk, saran, bimbingan dan dorongan dari berbagai pihak. Untuk itu penulis menyampaikan terima kasih kepada yang terhormat:

1. Prof. Dr. Fathur Rokhman, M. Hum., Rektor Universitas Negeri Semarang.
2. Prof. Dr. Wiyanto, M.Si., Dekan FMIPA Universitas Negeri Semarang.
3. Drs. Arief Agoestanto, M.Si., Ketua Jurusan Matematika FMIPA Universitas Negeri Semarang.
4. Dra. Kristina Wijayanti, M.S., Ketua Prodi Matematika FMIPA Universitas Negeri Semarang.
5. Endang Sugiharti, S.Si., M.Kom., Dosen Pembimbing Utama yang telah memberikan bimbingan, motivasi, dan pengarahan.
6. Much Aziz Muslim, S.Kom., M.Kom., Dosen Pembimbing Pendamping yang telah memberikan bimbingan, motivasi, dan pengarahan.
7. Pimpinan GIANT SUPERDOME Semarang yang telah memberikan ijin penelitian.
8. Ibu, bapak, dan adik tercinta yang selalu memberikan doa serta dukungan baik secara moral maupun spiritual.

9. Segenap civitas matematika Unnes, khususnya Bapak dan Ibu dosen yang telah memberikan ilmu dan motivasi dengan tulus.
10. Seluruh pihak yang turut membantu dalam penyusunan skripsi ini yang tidak dapat penulis sebutkan satu persatu.

Akhirnya penulis berharap semoga skripsi ini dapat bermanfaat bagi para pembaca.

Semarang, Agustus 2013

Penulis

ABSTRAK

Azizi, Muhamad Fithri Qomari. 2013. *Perbandingan antara Metode Backpropagation dengan Metode Learning Vector Quantization (LVQ) pada Pengenalan Citra Barcode*. Skripsi, Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing Utama Endang Sugiharti, S.Si., M.Kom. dan Pembimbing Pendamping Much Aziz Muslim, S.Kom., M.Kom.

Kata kunci: *Barcode*, Pengolahan Citra Digital, Jaringan Syaraf Tiruan (JST), *Backpropagation*, *Learning Vector Quantization (LVQ)*, dan *Matrix Laboratory*

Salah satu cara yang digunakan manusia dalam menyandikan sejumlah data secara khas adalah dengan *barcode*. Perangkat *input* yang digunakan untuk membaca *barcode* membutuhkan posisi sudut *barcode* yang tepat agar dapat dibaca dan sering juga kita jumpai bahwa kode *barcode* tersebut mengalami goresan atau kerusakan akibat gesekan dengan benda lain yang mengakibatkan alat pembaca *barcode* yang biasa tidak dapat mengenali *barcode* dengan baik. Metode jaringan syaraf tiruan *backpropagation* dan LVQ merupakan metode yang bisa dipakai untuk pengenalan pola secara tipikal. Permasalahan yang diangkat dalam penelitian ini adalah (1) bagaimana hasil pengenalan citra *barcode* dengan metode *Backpropagation* dibandingkan dengan metode LVQ; (2) bagaimana hasil optimal pengenalan citra *barcode* dengan metode *Backpropagation* dan LVQ; (3) manakah yang lebih baik antara metode *Backpropagation* atau metode LVQ. Dalam pengenalan citra *barcode* metode yang digunakan untuk menganalisis masalah adalah dengan menentukan masalah, studi pustaka, mengumpulkan data-data penelitian, merancang sistem, menguji sistem, membandingkan hasil uji dan penarikan kesimpulan.

Dari hasil pengujian sistem diperoleh tingkat akurasi sistem dalam mengenali citra *barcode* sebesar 75,5% untuk *backpropagation* dan 94% untuk LVQ. Arsitektur jaringan yang paling baik digunakan dalam proses pengenalan citra *barcode* dengan metode *backpropagation* adalah dengan variasi jumlah iterasi 1000, *learning rate* 0,05 dan jumlah *neuron hidden layer* 100. Sedangkan dengan metode *learning vector quantization* yaitu dengan variasi jumlah iterasi 10, *learning rate* 1 dan jumlah *neuron hidden layer* 20. Arsitektur jaringan tersebut menghasilkan *Mean Square Error* (MSE) sebesar $6,45 \times 10^{-28}$ dengan waktu *training* 102 detik untuk *backpropagation*. MSE sebesar 0 dengan waktu *training* 0,3 detik untuk LVQ. Dari hasil perbandingan, metode *learning vector quantization* lebih unggul dibandingkan dengan metode *backpropagation* dalam segi akurasi dan waktu pelatihan. Sistem dalam penelitian ini dibuat hanya untuk pengenalan *barcode* saja. Disarankan pada penelitian lebih lanjut dapat dikembangkan menjadi klasifikasi *barcode*. Selain itu peneliti juga menyarankan untuk penilitian selanjutnya dapat dicari metode jaringan syaraf tiruan lain, guna mendapatkan hasil pengenalan *barcode* yang lebih tepat dan akurat.

DAFTAR ISI

| | |
|-----------------------------|------|
| HALAMAN JUDUL | i |
| PERNYATAAN | ii |
| HALAMAN PENGESAHAN | iii |
| MOTTO DAN PERSEMBAHAN | iv |
| PRAKATA | v |
| ABSTRAK | vii |
| DAFTAR ISI..... | viii |
| DAFTAR TABEL | x |
| DAFTAR GAMBAR | xi |
| DAFTAR LAMPIRAN | xii |

BAB

1. PENDAHULUAN

| | |
|---------------------------------|---|
| 1.1 Latar Belakang Masalah..... | 1 |
| 1.2 Perumusan Masalah | 4 |
| 1.3 Batasan Masalah..... | 4 |
| 1.4 Tujuan Penulisan | 5 |
| 1.5 Manfaat Penulisan | 6 |
| 1.6 Sistematika Penulisan | 7 |

2. TINJAUAN PUSTAKA

| | |
|--|----|
| 2.1 Pengenalan <i>Barcode</i> | 9 |
| 2.2 Pengolahan Citra Digital | 15 |
| 2.3 <i>Artificial Intelligence</i> (Kecerdasan Buatan) | 24 |
| 2.4 Jaringan Syaraf Tiruan (JST) | 31 |
| 2.5 <i>Backpropagation</i> | 39 |
| 2.6 <i>Learning Vector Quantization</i> (LVQ) | 47 |
| 2.7 <i>Matrix Laboratory</i> (MATLAB) | 50 |

3. METODE PENELITIAN

| | |
|-----------------------------|----|
| 3.1 Perumusan Masalah | 55 |
|-----------------------------|----|

| | |
|--|-----|
| 3.2 Jenis dan Sumber Data..... | 55 |
| 3.3 Metode Pengumpulan Data | 55 |
| 3.4 Proses Pembuatan Perangkat Lunak | 57 |
| 3.5 Pemecahan Masalah..... | 58 |
| 3.6 Prosedur Penelitian | 60 |
| 3.7 Penarikan Simpulan | 61 |
| 4. HASIL DAN PEMBAHASAN | |
| 4.1 Akuisisi Citra (<i>Image Acquition</i>) | 63 |
| 4.2 Perancanaan Sistem | 64 |
| 4.3 Pengukuran Evaluasi Unjuk Kerja Sistem | 81 |
| 4.4 Analisis Hasil Kerja Sistem | 98 |
| 4.5 Membandingkan Hasil Sistem Pengenalan <i>Barcode</i> | 99 |
| 5. SIMPULAN DAN SARAN | |
| 5.1 Simpulan | 101 |
| 5.2 Saran | 102 |
| DAFTAR PUSTAKA | 103 |
| LAMPIRAN | 105 |

DAFTAR TABEL

| Tabel | Halaman |
|--|---------|
| 4.1 Spesifikasi <i>Hardware</i> dan <i>Software</i> dalam Proses <i>Cropping</i> | 66 |
| 4.2 Variasi <i>Learning Rate</i> dan Jumlah <i>Neuron Hidden Layer</i> | 70 |
| 4.3 Pengaruh <i>Learning Rate</i> Terhadap Akurasi | 70 |
| 4.4 Analisis Hasil Variasi Arsitektur JST <i>Backpropagation</i> | 81 |
| 4.5 Analisis Hasil Variasi Arsitektur JST LVQ | 82 |
| 4.6 Hasil Pengenalan Variasi 1 | 83 |
| 4.7 Hasil Pengenalan Variasi 2 | 86 |
| 4.8 Hasil Pengenalan Variasi 3 | 89 |
| 4.9 Hasil Pengenalan Variasi 4 | 92 |
| 4.10 Hasil Pengenalan Variasi 5 | 95 |

DAFTAR GAMBAR

| Gambar | Halaman |
|--|---------|
| 2.1 Anatomi <i>Barcode</i> Model UPC | 13 |
| 2.2 Proses <i>Edge Detection</i> pada Citra <i>Barcode</i> | 23 |
| 2.3 Penerapan Konsep Kecerdasan Buatan di Komputer | 25 |
| 2.4 Skema Logika Fuzzy..... | 28 |
| 2.5 Susunan Saraf Manusia | 32 |
| 2.6 Sebuah Sel Syaraf Tiruan | 36 |
| 2.7 Jaringan Layar Tunggal | 37 |
| 2.8 Jaringan Layar Jamak | 38 |
| 2.9 Jaringan <i>Recurrent</i> | 38 |
| 2.10 Arsitektur <i>Backpropagation</i> | 40 |
| 2.11 Fungsi Sigmoid Biner dengan <i>Range</i> (0,1) | 41 |
| 2.12 Fungsi Sigmoid Bipolar dengan <i>Range</i> (-1,1) | 42 |
| 2.13 Arsitektur LVQ | 48 |
| 2.14 Arsitektur Jaringan LVQ | 50 |
| 3.1 Model Sekuensial Linier | 57 |
| 4.1 Citra <i>Barcode</i> EAN-13 | 63 |
| 4.2 <i>Flowchart Pre processing</i> | 65 |
| 4.3 Tahap Proses <i>Pre processing</i> | 67 |
| 4.4 Gambar Vektor Hasil Ekstraksi Ciri | 69 |
| 4.5 Arsitektur Jaringan dengan 4 Unit <i>Neuron Hidden Layer</i> | 72 |
| 4.6 <i>Flowchart</i> Proses Pelatihan <i>Backpropagation</i> | 73 |
| 4.7 Arsitektur Jaringan LVQ dengan 100 Vektor <i>Input</i> dan 100 Vektor Bobot. | 74 |
| 4.8 <i>Flowchart</i> Proses Pelatihan LVQ | 76 |
| 4.9 <i>Flowchart</i> Sistem Pengenalan Citra <i>Backpropagation</i> | 78 |
| 4.10 <i>Flowchart</i> Sistem Pengenalan LVQ | 80 |

DAFTAR LAMPIRAN

| Lampiran | Halaman |
|---|---------|
| 1. Desain Tampilan Halaman Depan Program dengan Propertinya | 106 |
| 2. Kode Program untuk Halaman Depan | 107 |
| 3. Desain Tampilan Form_Uji dengan Propertinya | 110 |
| 4. Kode Program untuk Form_Uji | 112 |
| 5. Contoh Program Saat Dijalankan..... | 123 |
| 6. Citra Latih <i>Barcode</i> | 126 |
| 7. Citra Uji (Non-Latih) <i>Barcode</i> | 129 |
| 8. Surat Ketetapan Dosen Pembimbing | 132 |
| 9. Surat Ijin Penelitian dari Fakultas | 133 |
| 10. Surat Balasan Ijin Penelitian dari GIANT <i>SUPERDOME</i> Semarang..... | 134 |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini perkembangan teknologi sangat pesat sekali, seiring dengan kemajuan jaman dan perkembangan ilmu pengetahuan. Teknologi yang canggih memberikan kemudahan bagi pekerjaan manusia yang sifatnya memberatkan, sehingga pekerjaan yang dahulu masih didominasi oleh tenaga manusia, sekarang bisa dialihkan dengan tenaga mesin (komputer). Sekarang ini aplikasi dari komputer sudah banyak diterapkan dalam berbagai bidang kehidupan, misalnya bidang militer, kedokteran, industri, perdagangan, dan lain sebagainya, sehingga komputer menjadi alat bantu yang handal bagi manusia.

Komputer memberikan peran yang cukup besar pada bidang perdagangan, yaitu dalam kalkulasi harga barang secara cepat, *database*, juga dalam pembacaan citra *barcode*. ‘*Barcode*’ adalah salah satu cara yang digunakan manusia dalam menyandikan sejumlah data secara khas meliputi nama barang, jenis barang, jumlah, harga, periode suatu barang, dan lain sebagainya. Semua informasi yang dibutuhkan untuk keperluan tertentu mengenai suatu barang akan dijadikan suatu kode yang mempunyai bentuk batang dengan warna hitam/gelap. Aplikasi dari *barcode* selain untuk penyandian suatu barang juga dapat digunakan untuk penyandian sistem keamanan, seperti *doorlock system*. Aplikasi *barcode* yang biasa dijumpai yaitu pada supermarket, di mana kode *barcode* yang tertera pada barang mengandung kode jenis barang, dan kode produsen (Yuan *et al*, 1994).

Perangkat *input* yang digunakan untuk membaca *barcode* membutuhkan posisi sudut *barcode* yang tepat agar dapat dibaca dan sering juga kita jumpai bahwa kode *barcode* tersebut mengalami goresan atau kerusakan akibat gesekan dengan benda lain yang mengakibatkan alat pembaca *barcode* tidak dapat mengenali dengan baik. Dalam penelitian ini akan mengimplementasikan sebuah program yang dapat mengenali *barcode* sehingga dapat meminimalisir kekurangan yang ada pada perangkat *input* yang sering digunakan selama ini.

Salah satu metode yang dapat digunakan untuk identifikasi pola citra adalah jaringan syaraf tiruan. Banyak aplikasi-aplikasi yang telah dihasilkan seiring berkembangnya teknologi. Penelitian-penelitian tentang aplikasi penggunaan jaringan syaraf tiruan melalui pengenalan pola telah banyak dikaji dalam berbagai bidang ilmu pengetahuan. Penelitian yang dilakukan sebelumnya adalah Perbandingan Metode Jaringan Syaraf Tiruan *Backpropagation* dan *Learning Vector Quantization* pada Pengenalan Wajah (Maharani & Irawan, 2012). Penelitian lain yang berhubungan dengan pengenalan pola adalah yang dilakukan oleh Anindito, dkk tahun 2006 yaitu Perbandingan antara Metode *Kohonen Neural Network* dengan Metode *Learning Vector Quantization* pada Pengenalan Pola Tandatangan. Hal ini dapat dimanfaatkan untuk membantu meyelesaikan masalah yang akan dikaji dalam penelitian ini yaitu membuat suatu sistem yang dapat mengenali citra *barcode*.

Menurut Kusumadewi (2003:221), terdapat 2 jenis metode pembelajaran pada jaringan syaraf tiruan, yaitu pembelajaran terawasi (*supervised learning*) dan pembelajaran tak terawasi (*unsupervised learning*). Metode pembelajaran pada

jaringan syaraf disebut terawasi jika *output* yang diharapkan sudah diketahui sebelumnya. Pada metode pembelajaran yang tak terawasi ini tidak memerlukan target *output*. Untuk melakukan pengenalan *barcode*, pembelajaran terawasi lebih cocok karena menggunakan target keluaran, diantaranya yang termasuk metode pembelajaran terawasi adalah *backpropagation* dan *learning vector quantization*. *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tetapi tidak sama) dengan pola yang dipakai selama pelatihan (Siang, 2005:97). *Learning vector quantization* merupakan metode pembelajaran pada lapisan kompetitif yang akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor masukan (Kusumadewi, 2004:295).

Metode *backpropagation* dan *learning vector quantization* memiliki kelebihan dan kekurangan masing-masing dalam proses pembelajaran untuk pengenalan gambar. *Backpropagation* menggunakan 2 alur dalam perhitungan bobot, yaitu propagasi maju (*forward*) dan propagasi mundur (*backward*). Sedangkan *learning vector quantization* hanya menggunakan 1 alur dalam perhitungannya, yaitu alur maju. Keduanya memiliki perbedaan dalam penentuan hasil pengenalan, maka penilitian ini mengambil judul Perbandingan antara Metode *Backpropagation* dengan Metode *Learning Vector Quantization* (LVQ) pada Pengenalan Citra *Barcode*.

1.2 Perumusan Masalah

Berdasarkan permasalahan yang telah diuraikan pada bagian latar belakang, maka permasalahan yang timbul adalah sebagai berikut.

- a. Bagaimana hasil pengenalan citra *barcode* dengan metode *backpropagation* dibandingkan dengan metode LVQ (*Learning Vector Quantization*)?
- b. Parameter apa saja yang digunakan untuk memperoleh hasil optimal pengenalan citra *barcode* dengan metode *backpropagation* dan LVQ (*Learning Vector Quantization*)?
- c. Manakah yang lebih baik antara metode *backpropagation* atau metode LVQ (*Learning Vector Quantization*) dalam pengenalan citra *barcode*?

1.3 Batasan Masalah

Agar dalam pembahasan skripsi ini tidak terlalu meluas, maka penulis mencantumkan pembatasan masalah sebagai berikut.

- a. Citra *barcode* yang akan dikenali adalah data jenis EAN-13.
- b. Citra *barcode* diambil dari Supermarket GIANT Jalan Siliwangi Semarang.
- c. Metode deteksi tepi yang digunakan adalah metode sobel.
- d. Akuisisi citra dilakukan dengan menggunakan kamera *merk* Fujifilm dengan resolusi 16 *megapixel* dengan pengambilan gambar dilakukan pada jarak 5 cm sampai 10 cm.
- e. Citra yang digunakan adalah citra statik berwarna berukuran 100×60 *pixel*.

- f. Proses yang dilakukan adalah pembelajaran citra *barcode*. Pengenalan citra *barcode* dengan metode *backpropagation* dan *learning vector quantization*, serta perbandingan hasil pengenalan citra *barcode*.
- g. Menciptakan suatu sistem untuk membantu proses perbandingan antara metode *backpropagation* dan *learning vector quantization* pada citra *barcode* dengan menggunakan *software Matlab R2009a*.
- h. Data keluaran berupa hasil perhitungan pengenalan citra *barcode*, persentase akurasi, dan perbandingan antara metode *backpropagation* dan *learning vector quantization* pada citra *barcode*.

1.4 Tujuan Penulisan

Tujuan penulisan skripsi ini adalah sebagai berikut.

- a. Membandingkan hasil pengenalan citra *barcode* dengan metode *backpropagation* dan *learning vector quantization*.
- b. Mengetahui parameter apa saja yang digunakan untuk memperoleh hasil optimal pengenalan citra *barcode* dengan metode *backpropagation* dan LVQ (*Learning Vector Quantization*).
- c. Memberikan rekomendasi metode yang lebih baik antara metode *backpropagation* dan *learning vector quantization* dalam pengenalan citra *barcode*.

1.5 Manfaat Penulisan

Manfaat yang bisa diperoleh dari penyusunan skripsi ini adalah.

1.5.1 Bagi Peneliti

- a) Peneliti dapat membuat sistem pengenalan citra *barcode* dengan jaringan syaraf tiruan *backpropagation* dan *learning vector quantization*.
- b) Peneliti dapat memberikan salah satu modul (referensi) yang dapat membantu perusahaan untuk mengatasi kekurangan dari pembaca *barcode* yang berupa sinar laser, dimana program yang dihasilkan nanti dapat mengenali citra *barcode*.

1.5.2 Bagi Mahasiswa

Semoga penelitian ini dapat memberikan inspirasi dan gagasan baru bagi mahasiswa untuk mengembangkan dan merancang aplikasi menggunakan jaringan syaraf tiruan yang nantinya dapat dimanfaatkan dalam kehidupan sehari-hari.

1.5.3 Bagi Perusahaan Terkait

Harapannya modul (referensi) ini dapat bermanfaat bagi perusahaan dan pebisnis *retail* untuk memberikan bantuan dalam mengenali gambar *barcode* yang lebih akurat, efektif, dan efisien agar produk yang dipasarkan terjamin keamanannya, meningkat produktifitasnya, serta meningkat profitnya.

1.6 Sistematika Penulisan

1.6.1 Bagian Awal

Dalam penulisan skripsi ini, bagian awal berisi halaman judul, halaman pengesahan, pernyataan, motto dan persembahan, kata pengantar, abstrak dan daftar isi.

1.6.2 Bagian Pokok

Bagian pokok dari penulisan skripsi ini adalah isi skripsi yang terdiri dari lima bab.

(1) BAB I PENDAHULUAN

Berisi tentang latar belakang, permasalahan, pembatasan masalah, tujuan, manfaat, dan sistematika penulisan.

(2) BAB II LANDASAN TEORI

Berisi tentang pengenalan *barcode*, pengolahan citra digital, *artificial intelligence*, jaringan syaraf tiruan, *backpropagation*, *Learning Vector Quantization* (LVQ), dan *Matrix Laboratory* (Matlab).

(3) BAB III METODE PENELITIAN

Berisi tentang prosedur atau langkah-langkah yang dilakukan dalam penelitian ini meliputi perumusan masalah, jenis dan sumber data, metode pengumpulan data, proses pembuatan perangkat lunak, pemecahan masalah, prosedur penelitian, dan penarikan simpulan.

(4) BAB IV HASIL DAN PEMBAHASAN

Berisi tentang perancangan sistem pelatihan dan pengenalan citra *barcode* dengan metode *backpropagation* dan *learning vector quantization*, pengolahan data, pengukuran evaluasi unjuk kerja sistem, dan membandingkan antara hasil pegenalan citra *barcode* menggunakan metode *backpropagation* dengan metode *learning vector quantization*.

(5) BAB V PENUTUP

Berisi simpulan dari penulisan skripsi ini dan saran.

1.6.3 Bagian Akhir

Berisi daftar pustaka sebagai acuan penulisan yang mendukung kelengkapan skripsi ini.

BAB II

TINJAUAN PUSTAKA

2.1 Pengenalan *Barcode*

2.1.1 Sejarah dan Definisi *Barcode*

Malik dkk (2010:1) dalam buku Implementasi Teknologi *Barcode* dalam Dunia Bisnis menyatakan bahwa *barcode* pertama kali diperkenalkan oleh dua orang mahasiswa *Drexel Institute of Technology*, yakni Bernard Silver dan Norman Joseph Woodland di tahun 1948. Mereka mendaftarkan hak paten atas inovasi tersebut pada tahun 1949 dan dikabulkan pada tahun 1952. Namun demikian baru pada tahun 1996 penemuan mereka digunakan di dunia komersial. Pada kenyataannya penggunaan *barcode* tidak begitu sukses hingga pasca tahun 1980-an. Namun demikian pada hari ini nyaris semua produk, mulai dari makanan, pakaian, hingga sabun mandi, langsung dilengkapi dengan *barcode* begitu masuk pusat perbelanjaan, minimarket atau toko. Walter Satterthwaite, konsultan dari *master food*, mengatakan bahwa keberadaan *barcode* telah membantu semua pusat perbelanjaan diseluruh dunia.

Menurut Malik dkk (2010), ada banyak definisi *barcode*, tetapi semua definisi tersebut merujuk pada satu hal yang sama, yaitu.

- a. *Barcode* secara harfiah berarti kode berbentuk garis.
- b. Sebagai kumpulan kode yang berbentuk garis, dimana masing-masing ketebalan setiap garis berbeda sesuai dengan isi kodennya.

- c. Informasi terbacakan mesin (*machine readable*) dalam format visual yang tercetak. Umumnya *barcode* berbentuk garis-garis vertikal tipis tebal yang terpisah oleh jarak tertentu.
- d. Sejenis kode yang mewakili data atau informasi tertentu, biasanya jenis dan harga barang, seperti makanan dan buku. Kode berbentuk batangan balok dan berwarna hitam putih ini mengandung satu kumpulan kombinasi batang yang berlainan ukuran yang disusun sedimikian rupa. Kode ini dicetak di atas stiker atau di kotak pembungkus barang.

2.1.2 Manfaat *Barcode*

Ada beberapa manfaat yang dapat diambil dari penggunaan *barcode*, antara lain.

1) Akurasi

Meningkatkan akurasi dengan mengurangi kesalahan manusia dari pemasukan data secara manual atau item yang salah baca atau salah label.

2) Kemudahan Pemakain

Barcode mudah digunakan. Dengan *hardware* dan *software* yang tepat bisa memaksimalkan proses otomatisasi pengumpulan data.

3) Keseragaman Pengumpulan Data

Beragam standar pemenuhan dan simbologi *barcode* yang terstandardisasi menjamin informasi diterima dan disampaikan dengan cara yang benar sehingga bisa diterima dan dipahami secara umum.

4) *Feedback* yang tepat waktu

Barcode menawarkan *feedback* yang tepat waktu. Begitu mencul, data bisa diterima dengan cepat sehingga memungkinkan pengambilan keputusan yang cepat berdasarkan informasi terbaru.

5) Keamanan

Pada bisnis *retail* seperti supermarket, banyak pembeli nakal yang menukar label harga produk dengan label harga yang lebih murah. Kesalahan yang sama bisa terjadi pada saat menempel label maupun pada saat kasir menghitung total belanja.

6) Meningkatkan Produktivitas

Barcode membuat aktivitas operasional dalam bisnis menjadi lebih singkat.

7) Meningkatkan Profit

Peningkatan efisiensi yang diberikan *barcode* memungkinkan perusahaan menghemat biaya sehingga profit bisnisnya jadi meningkat.

2.1.3 Perkembangan dan Teknologi *Barcode*

Barcode kini telah menjadi bahasa global dalam standardisasi bisnis sehingga tidak akan timbul masalah apabila dua perusahaan memiliki produk yang sama dengan harga yang berbeda. Susunan *barcode* terdiri atas angka awalan (*prefiks*) yang mengidentifikasi nama perusahaan dan akhiran (*sufiks*) yang variasi panjangnya tergantung pada jenis produk. Mayoritas penggunaan *barcode* di seluruh dunia mengikuti standar yang ditetapkan *European Article Number Association* (EANA), kecuali Amerika Serikat (AS) dan Kanada. Kedua negara tersebut menggunakan standar tersendiri, yaitu standar yang ditemukan pada

tahun 1973 oleh George J. Laurer. Standar ini dikenal dengan nama *Universal Product Code* (UPC). (Malik dkk, 2010: 3)

Beberapa *barcode* standar telah dikembangkan selama beberapa tahun, yang biasa disebut dengan simbologi. Simbologi *barcode* dibedakan dalam 2 jenis, dalam bentuk linier atau dimensial. Simbologi *barcode* linier berisi garis-garis hitam yang berjajar, dan garis putih dengan ukuran tinggi dan lebar tertentu. Demikian halnya dengan simbologi dimensional, yang bisa diatur dalam format tegak lurus atau matriks, yang berupa sandi segiempat khusus, sehingga bisa lebih banyak menyimpan informasi. Adapun teknologi yang diterapkan pada *barcode* adalah teknologi laser, CCD, dan *linier imager*.

2.1.4 Sistem Kerja *Barcode*

Barcode merupakan instrumen yang bekerja berdasarkan asas digital. Pada konsep digital, hanya ada dua sinyal data yang dikenal dan bersifat *Boolean*, yaitu 0 atau 1. *Barcode* menerapkannya pada batang-batang baris yang terdiri dari warna hitam dan putih. Warna hitam mewakili bilangan 0, dan warna putih mewakili bilangan 1. Warna hitam akan menyerap cahaya yang dipancarkan oleh alat pembaca *barcode*, sedangkan warna putih akan memantul-balikkan cahaya tersebut. Masing-masing *barcode* memiliki ketebalan yang berbeda. Ketebalan inilah yang akan diterjemahkan ke dalam suatu nilai.

Ada dua macam koneksi alat pembaca *barcode*, yaitu *system keyboard wedge* dan *system output RS232*. Sistem ini menerjemahkan hasil pembacaan *barcode* sebagai masukan (*input*) dari *keybord*. Biasanya menggunakan *port serial* pada komputer. Kita memerlukan *software* perantara, umumnya disebut *software*

wedge, yang akan mengalamatkan bacaan dari *barcode* ke *software* pengolah data *barcode* tersebut.

2.1.5 Anatomi *Barcode*

Barcode memiliki garis tipis dan tebal serta bilangan inisial *barcode* yang tertera di bawahnya. Komputer tidak membaca bilangan yang berada di bagian bawah *barcode*. Bilangan itu dicetak agar orang dapat membacanya dengan mudah bilamana memerlukannya.



Gambar 2.1. Anatomi *Barcode* Model UPC

Keterangan gambar *barcode*.

1. *Number system Character*

Angka ini merupakan sebuah sistem bilangan *barcode* UPC yang mengkarakteristikkan jenis-jenis khusus pada *barcode*. Di dalam *barcode* UPC, *Number System Character* ini biasanya terletak di sebelah kiri *barcode*. Kode-kode yang tertera adalah sebagai berikut.

0 – Standard UPC number;

1 – Reserved;

2 – Random weight item like fruits, vegetables, and meats;

- 3 – *Pharmaceuticals*;
- 4 – *In-store code for retailers*;
- 5 – *coupons*;
- 6 – *standard UPC number*;
- 7 – *standard UPC number*;
- 8 – *Reserved*;
- 9 – *Reserved*.

2. *Guard Bars*

Ada tiga *guard bars* yang ditempatkan di awal, di tengah, dan akhir *barcode*. *Guard bars* bagian awal dan akhir di-*encode*-kan sebagai “bar-space-bar” atau “101”. *Guard bars* bagian tengah di-*encode*-kan sebagai “space-bar-space-bar-space” atau “01010”.

3. *Manufacturer Code*

Kode perusahaan ini ada lima digit bilangan yang secara khusus menentukan manufaktur suatu produk. Kode perusahaan atau manufaktur ini dilindungi dan ditetapkan oleh *Uniform Code Council* (UCC).

4. *Product Code*

Kode produk ini ada lima digit bilangan yang ditetapkan oleh perusahaan atau manufaktur untuk setiap produk yang dihasilkan.

5. *Check digit*

Disebut sebagai digit *self-check*. *Check digit* ini terletak dibagian luar sebelah kanan *barcode*. *Check digit* merupakan suatu *out-programer's trick* untuk

memvalidasi digit-digit lainnya (*number system character, manufacturer code, product code*) yang dibaca secara teliti.

2.2 Pengolahan Citra Digital

2.2.1 Pengertian Citra Digital

Citra dapat didefinisikan sebagai fungsi dua dimensi dari $f(a, b)$ dimana a dan b adalah koordinat bidang dan amplitudo dari f pada pasangan koordinat apapun adalah intensitas atau tingkat keabuan dari citra pada titik tersebut. Jika a , b , dan nilai intensitas dari f tersebut berhingga, bernilai diskrit, citra tersebut disebut citra digital (Gonzalez & Woods, 2008).

2.2.2 Pengolahan Citra

Pengolahan gambar atau pengolahan citra yang sering disebut *image processing*, merupakan suatu proses yang mengubah sebuah gambar menjadi gambar lain yang memiliki kualitas lebih baik untuk tujuan tertentu (Ardhianto, dkk, 2011). Sari (2010) menyatakan bahwa pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Menurut Pratiarso, dkk (2009), sesuai dengan perkembangan komputer itu sendiri, pengolahan citra mempunyai dua tujuan utama, yaitu sebagai berikut.

- a. Memperbaiki kualitas citra, dimana citra yang dihasilkan dapat menampilkan informasi secara jelas. Hal ini berarti manusia sebagai pengolah informasi (*human perception*).

- b. Mengekstrasi informasi ciri yang menonjol pada suatu citra, hasilnya adalah informasi citra dimana manusia mendapatkan informasi ciri dari citra secara numerik.

Operasi-operasi pada pengolahan citra diterapkan pada citra bertujuan untuk:

- a. meningkatkan kualitas penampakan citra/menonjolkan beberapa aspek informasi yang terkandung dalam citra (*image enhancement*), contoh : perbaikan kontras gelap/terang, perbaikan tepi objek, penajaman, pemberian warna semu, dll;
- b. menghilangkan atau meminimumkan cacat pada citra (*image restoration*), contoh : penghilangan kesamaran (*deblurring*);
- c. *image segmentation*, yaitu pengelompokan, pencocokan, dan mengukur elemen dalam citra. Operasi ini berkaitan erat dengan pengenalan pola;
- d. mengetahui ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu mengidentifikasi objek (*image analysis*). Proses segmentasi (*Image Segmentation*) kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh : pendekripsi tepi objek;
- e. memperkecil ukuran citra agar mudah diolah dalam komputer (*image compression*), contoh : suatu *file* citra berbentuk JPEG berukuran 258 KB dikompres dengan metode JPEG menjadi berukuran 49 KB atau menjadi ukuran *pixel* yang lebih kecil.

2.2.3 Pengenalan Pola

Pengertian pengenalan pola adalah mengelompokkan data numerik dan simbolik (termasuk citra) secara otomatis oleh mesin (komputer). Tujuan pengelompokan adalah untuk mengenali suatu objek di dalam citra. Manusia bisa mengenali objek yang dilihatnya karena otak manusia telah belajar mengklasifikasi objek-objek di alam sehingga mampu membedakan suatu objek dengan objek lainnya. Kemampuan sistem visual manusia yang dicoba ditiru oleh mesin.

Komputer menerima masukan berupa citra objek yang akan diidentifikasi, memproses citra tersebut dan memberikan keluaran berupa informasi/deskripsi objek di dalam citra. Secara umum, pengolahan citra digital menjadi sebuah informasi deskripsi sebuah objek dibagi menjadi 3 proses utama, yaitu pengambilan data citra, pengolahan citra (*pre processing*), dan pengenalan pola (*classification*).

2.2.3.1 Data Citra

Gambar-gambar yang selama ini dimiliki dan dilihat oleh manusia merupakan hasil pencitraan dari sebuah cahaya yang ditangkap oleh mata melalui retina (Gonzales, 2002).

Menurut Sari (2010), komputer dapat mengolah isyarat-isyarat elektronik digital yang merupakan kumpulan sinyal biner (bernilai 0 dan 1). Untuk itu, citra digital harus mempunyai format tertentu yang sesuai sehingga dapat merepresentasikan objek pencitraan dalam bentuk kombinasi data biner. Data citra

gambar yang dapat diolah oleh komputer adalah berupa format *.JPEG, *.BMP, *.TIF, *.GIF, *.PNG, dan *.FWD.

2.2.3.2 Pengolahan Citra (*Pre Processing*)

Sebelum citra diklasifikasikan sebagai objek tertentu dalam proses pengenalan pola, citra yang diperoleh harus diolah terlebih dahulu agar citra tersebut dapat diolah oleh sistem komputer. Proses-proses tersebut antara lain adalah sebagai berikut.

2.2.3.2.1 *Image Acquisition*

Proses ini berfungsi agar data yang diperoleh menjadi data yang homogen. Proses ini dilakukan dengan *cropping*, yaitu memotong gambar menjadi bagian yang diinginkan dengan ukuran *pixel* yang sama untuk setiap citra.

2.2.3.2.2 *Image Segmentation*

Operasi ini bertujuan untuk mengelompokkan citra menjadi beberapa *region* berdasarkan kriteria tertentu. Segmentasi citra bertujuan untuk menemukan karakteristik yang dimiliki oleh citra sehingga sering juga digunakan untuk pengenalan pola citra.

Proses ini dilakukan apabila data yang diperoleh berupa citra dari gambar warna (RGB) mempunyai nilai *pixel* yang besar sehingga pengolahan citra akan sulit. Untuk itulah perlu konversi citra dari RGB ke *grayscale*.

2.2.3.2.3 *Image Enhancement*

Operasi ini bertujuan untuk meningkatkan kualitas citra di mata manusia dengan mengubah parameter-parameter pada suatu citra. Gambar yang telah dikonversi ke dalam bentuk *grayscale* harus dipertajam kualitas citranya agar mempermudah saat proses pengenalan pola.

Proses yang digunakan untuk mempertajam citra antara lain adalah modifikasi histogram, penggunaan filter, dan analisis frekuensi.

2.2.3.2.4 *Morphology*

Proses ini digunakan untuk mendapatkan komponen dari suatu gambar yang berguna untuk merepresentasikan dan mendeskripsikan suatu bentuk, seperti *boundary*, *skeleton*, dan *convex hull* (Gonzales, 2002). Operasi *morphology* bertujuan untuk memperbaiki bentuk objek agar dapat menghasilkan fitur-fitur yang lebih akurat ketika analisis dilakukan terhadap objek.

Terdapat 4 proses *Morphology* (Gonzales, 2002) yaitu.

(a) Dilasi

Dilasi berarti mengubah ukuran suatu gambar tanpa merubah bentuk dasarnya. Dilasi A dan B dinyatakan dengan $A \oplus B$ dapat dilambangkan dengan :

$$A \oplus B = \{z | [(\hat{B})_z \cap A] \subseteq A\}$$

Persamaan ini didasarkan pada perefleksian B terhadap originalnya, dan penggeseran refleksi oleh z . Dilasi A dan B kemudian adalah himpunan semua *displacement* z , sebagaimana B dan A overlap oleh paling sedikit satu elemen. Himpunan B adalah *strel*, sedangkan A himpunan (objek citra) yang terdilasi.

(b) Erosion

Erosion dari A oleh B adalah ketika *subset* dari A yang tidak sama dengan B , maka *subset* itu dihapus (sama dengan 0). Erosi A dan B dinyatakan dengan $A \ominus B$ dapat dilambangkan dengan:

$$A \ominus B = \{z | [(B)_z \cap A] \subset A\}$$

Persamaan di atas menunjukkan bahwa erosi A oleh B adalah kumpulan semua titik dimana B ditranslasikan oleh z di dalam isi A . B di sini adalah *strel*.

(c) *Opening*

Opening bekerja dengan cara menghilangkan objek-objek yang kecil.

(d) *Closing*

Closing bekerja dengan cara menghilangkan lubang-lubang kecil.

2.2.3.2.5 *Image Analisys*

Proses ini berfungsi untuk mengetahui ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu mengidentifikasi objek. Salah satu metode yang bisa digunakan adalah *edge detection* (deteksi tepi).

Tepi objek adalah pertemuan antara bagian objek dan bagian latar belakang objek (Ahmad, 2005). Pelacakan tepi pada dasarnya merupakan operasi untuk menemukan perubahan intensitas lokal yang berbeda nyata dalam sebuah bidang citra.

Edge detection pada dasarnya merupakan operasi untuk menemukan perubahan intensitas lokal yang berbeda nyata dalam sebuah bidang citra. *Gradient* adalah hasil pengukuran perubahan dalam sebuah fungsi intensitas, dan sebuah citra dapat dipandang sebagai kumpulan beberapa fungsi intensitas kontinyu citra (Ahmad, 2005). *Gradient* merupakan kesamaan dua dimensi dari turunan pertama. *Gradient* fungsi 2-D $f(x, y)$ didefinisikan sebagai vektor seperti diperlihatkan dalam persamaan:

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Persamaan di atas, besaran *gradient* dapat dinyatakan dengan hukum Pythagoras sebagai berikut:

$$G[f(x, y)] = \sqrt{G_x^2 + G_y^2}$$

Akan tetapi pada pengolahan citra digital, besaran *gradient* didekati dengan nilai mutlak persamaan berikut:

$$G[f(x, y)] = |G_x| + |G_y|, \text{ atau } G[f(x, y)] \approx \max|G_x| + |G_y|$$

Dengan menggunakan analisis vektor, arah dari suatu *gradient* dapat dinyatakan dengan persamaan:

$$\alpha(x, y) = \tan^{-1} \left(\frac{G_x}{G_y} \right)$$

Dimana sudut α diukur dari sumbu x sebagai garis acuan.

Cahyono (2006) menyatakan bahwa deteksi tepi (*edge detection*) adalah suatu proses yang menghasilkan tepi-tepi dari objek-objek gambar. Suatu titik (x, y) dikatakan suatu tepi (*edge*) dari suatu citra, bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangga. Macam-macam metode untuk proses deteksi tepi, antara lain sebagai berikut.

(a) Metode Robert

Metode Robert adalah nama lain dari teknik differensial, yaitu differensial pada arah horizontal dan differensial ke arah vertikal, dengan ditambahkan proses konversi biner setelah dilakukan differensial. Bentuk operator Robert dapat dilihat pada rumus (Nurismu, dkk, 2012):

$$G_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \text{ dan } G_y \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

(b) Metode Prewitt

Metode Prewitt merupakan pengembangan metode Robert dengan menggunakan filter *High Pass Filter* (HPF) yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi *Laplacian* yang dikenal sebagai fungsi untuk membangkitkan HPF. Berikut akan ditunjukkan operator Prewitt yang digunakan yaitu (Prasetyo, 2011: 215):

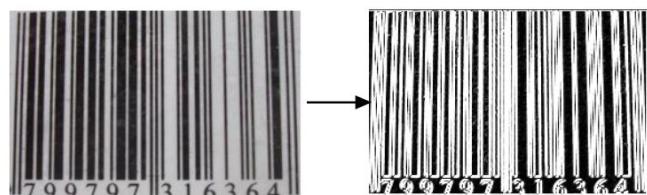
$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ dan } G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

(c) Metode Sobel

Metode Sobel merupakan pengembangan metode Robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi *Laplacian* dan *Gaussian* yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode ini adalah kemampuan untuk mengurangi *noise* sebelum melakukan perhitungan deteksi tepi. Persamaan *gradient* pada operator Sobel adalah sebagai berikut (Pujiyono, dkk, 2009):

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

Proses *Edge Detection* citra barcode seperti tampak pada Gambar 2.2.



Gambar 2.2 Proses *Edge Detection* pada Citra Barcode

2.2.4 Pengolahan Warna

Dalam gambar berwana RGB, pengolahan citra sulit dilakukan. Untuk itu lah perlu dibedakan intensitas dari masing-masing warna. Hal yang perlu dilakukan dalam proses ini adalah bagaimana melakukan pembacaan nilai-nilai R, G, dan B pada suatu *pixel*, menampilkan dan menafsirkan hasil perhitungan sehingga mempunyai arti sesuai yang diinginkan.

Salah satu cara yang mudah untuk menghitung nilai warna dan menafsirkan hasilnya dalam model warna RGB adalah dengan melakukan normalisasi terhadap ketiga komponen warna tersebut (Ahmad, 2005).

Cara melakukan normalisasi adalah sebagai berikut:

$$r = \frac{R}{R + G + B}$$

$$g = \frac{G}{R + G + B}$$

$$b = \frac{B}{R + G + B}$$

Nilai warna hasil normalisasi kemudian ditafsirkan dengan melihat besarnya. Jadi dominasi warna dapat dilihat dari besaran nilai tiap indeks. Prasetyo (2011: 181) menyatakan bahwa semua nilai R, G, dan B diasumsikan dalam *range* [0,1]. Citra yang direpresentasikan dalam model warna RGB terdiri dari tiga komponen citra, masing-masing untuk setiap warna primer (*Red, Green, and Blue*).

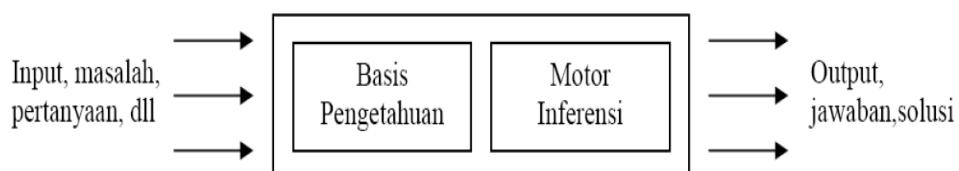
2.3 Artificial Intelligence (Kecerdasan Buatan)

Kusumadewi (2003:1) dalam buku *Artificial Intelligence* (Teknik dan Aplikasinya) menyatakan bahwa kecerdasan buatan (*artificial intelligence*) adalah

bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik dari pada yang dilakukan manusia. Menurut John McCarthy dalam buku Kusumadewi (2003), “untuk mengetahui dan memodelkan proses–proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia”.

Cerdas adalah memiliki pengetahuan dan pengalaman. Penalaran yaitu bagaimana membuat keputusan dan mengambil tindakan, serta moral yang baik. Agar mesin bisa cerdas atau bertindak seperti dan sebaik manusia, maka harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Penerapan kecerdasan buatan digambarkan pada Gambar 2.3. Kusumadewi (2003:3) menyatakan bahwa, dua bagian utama yg dibutuhkan untuk aplikasi kecerdasan buatan adalah.

- a. Basis pengetahuan (*knowledge base*): berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
- b. Motor inferensi (*inference engine*): kemampuan menarik kesimpulan berdasarkan pengetahuan.



Gambar 2.3. Penerapan Konsep Kecerdasan Buatan di Komputer

2.3.1 Perbedaan Kecerdasan Buatan dan Kecerdasan Alami

Berikut ini merupakan perbedaan antara kecerdasan buatan dan kecerdasan alami.

2.3.1.1 Kelebihan Kecerdasan Buatan

- a) Lebih bersifat permanen. Kecerdasan alami bisa berubah karena sifat manusia pelupa. Kecerdasan buatan tidak berubah selama sistem komputer dan program tidak mengubahnya.
- b) Lebih mudah diduplikasi dan disebarluaskan. Memindahkan pengetahuan manusia dari 1 orang ke orang lain membutuhkan proses yang sangat lama dan keahlian tidak akan pernah dapat diduplikasi dengan lengkap. Jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.
- c) Lebih murah. Menyediakan layanan komputer akan lebih mudah dan murah dibandingkan mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
- d) Bersifat konsisten dan teliti karena kecerdasan buatan adalah bagian dari teknologi komputer sedangkan kecerdasan alami senantiasa berubah-ubah.
- e) Dapat didokumentasikan. Keputusan yang dibuat komputer dapat didokumentasikan dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.
- f) Dapat mengerjakan beberapa tugas dengan lebih cepat dan lebih baik dibanding manusia.

2.3.1.2 Kelebihan Kecerdasan Alami

- a) Kreatif, manusia memiliki kemampuan untuk menambah pengetahuan, sedangkan pada kecerdasan buatan untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun.
- b) Memungkinkan orang untuk menggunakan pengalaman atau pembelajaran secara langsung. Sedangkan pada kecerdasan buatan harus mendapat masukan berupa data-data simbolik.
- c) Pemikiran manusia dapat digunakan secara luas, sedangkan kecerdasan buatan sangat terbatas.

2.3.2 Kajian Kecerdasan Buatan

Terdapat banyak irisan dalam ruang lingkup kecerdasan buatan yang membuat klasifikasi kecerdasan buatan menjadi rumit, sehingga untuk memudahkan klasifikasi kecerdasan buatan, pengklasifikasian kecerdasan buatan didasarkan pada keluaran yang diberikan. Secara garis besar, beberapa kajian yang termasuk kecerdasan buatan adalah sebagai berikut.

2.3.2.1 Sistem Pakar (*Expert System*)

Sistem pakar (*expert system*) adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja dari para ahli. Sistem pakar merupakan sistem yang memindahkan keahlian dari seorang pakar atau sumber kepakaran yang lain ke komputer, pengetahuan yang ada disimpan dalam komputer, dan pengguna dapat berkonsultasi pada komputer itu

untuk suatu nasehat, lalu komputer dapat mengambil inferensi (menyimpulkan, mendeduksi, dll.) seperti layaknya seorang pakar, kemudian menjelaskannya ke pengguna tersebut, bila perlu dengan alasan-alasannya.

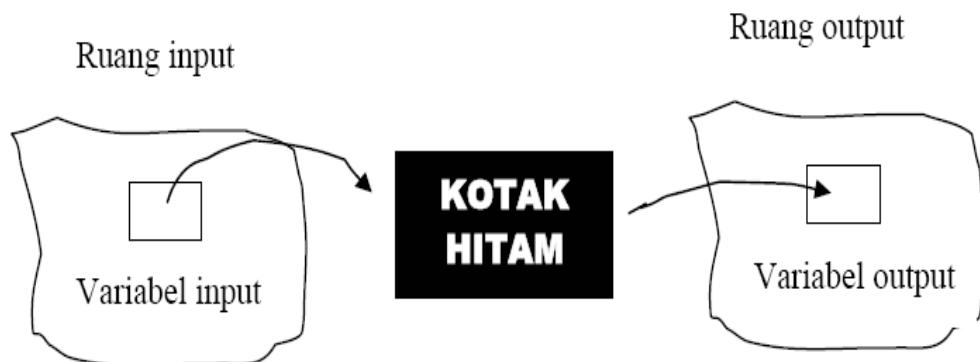
2.3.2.2 *Permainan (Game Playing)*

Games adalah fasilitas yang sangat menarik dalam komputer. Ide *games* pertama kali dimunculkan oleh Claude Shannon (1950) yang menulis karya tulis tentang mekanisme pembuatan program permainan catur. Beberapa alasan mengapa *games* merupakan domain yang baik untuk dieksplorasi yaitu.

- a) Sangat mudah untuk menentukan ukuran kesuksesan dan kegagalannya (menang atau kalah).
- b) Tidak membutuhkan terlalu banyak pengetahuan. Permainan dapat diselesaikan dengan melakukan pencarian dari arah mulai sampai posisi menang. Namun alasan ini tidak cocok untuk *games* yang besar (kompleks).
- c) Ruang keadaannya mudah direpresentasikan.

2.3.2.3 *Logika Fuzzy*

Logika fuzzy adalah suatu cara untuk memetakan suatu ruang masukan ke dalam suatu ruang keluaran. Antara masukan dan keluaran terdapat suatu kotak hitam yang harus memetakan masukan ke keluaran yang sesuai. Skema logika fuzzy dapat dilihat pada Gambar 2.4.



Gambar 2.4. Skema Logika Fuzzy

Beberapa alasan menggunakan logika fuzzy adalah sebagai berikut:

- konsep logika fuzzy mudah dimengerti;
- logika fuzzy sangat fleksibel;
- memiliki toleransi terhadap data-data yang tidak tepat.

2.3.2.4 Jaringan Syaraf Tiruan (*Artificial Neural Network*)

Jaringan syaraf merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. Jaringan Syaraf Tiruan (JST) adalah sistem pemrosesan informasi yang memiliki karakteristik unjuk kerja tertentu yang menyerupai jaringan syaraf biologis (Fausett, 1994). JST telah dikembangkan sebagai generalisasi model matematika dari aspek kognitif manusia atau syaraf biologi, yaitu didasarkan pada asumsi-asumsi bahwa:

- a. pemrosesan informasi terjadi pada elemen-elemen yang disebut *neuron*;
- b. sinyal-sinyal merambat di antara neuron melalui interkoneksi;
- c. setiap interkoneksi memiliki bobot yang bersesuaian pada kebanyakan jaringan syaraf berfungsi untuk mengalikan sinyal yang dikirim;
- d. setiap *neuron* menerapkan fungsi aktivasi pada masukan jaringan untuk menentukan sinyal keluaran.

2.3.2.5 Algoritma Genetika (*Genetic Algorithm*)

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Variasi kromosom ini akan mempengaruhi laju produksi dan tingkat kemampuan organisme untuk tetap hidup. Individu yang lebih kuat akan memiliki tingkat kelangsungan hidup dan tingkat reproduksi yang lebih tinggi.

2.3.2.6 Robotika

Kecerdasan buatan dalam robotika diterapkan sebagai algoritma yang dimasukkan kedalam kontroler robot. Dengan kecerdasan buatan membuat robot tersebut berkemampuan secara sendiri untuk merespon atau bereaksi didalam kondisi yang tidak ditentukan sebelumnya.

2.3.3 Penerapan Kecerdasan Buatan

Kecerdasan buatan banyak diterapkan pada berbagai bidang, misalnya pada bidang ilmu komputer (informatika), teknik elektro, kesehatan, manajemen dan

lain sebagainya. Beberapa contoh penerapan kecerdasan buatan adalah sebagai berikut.

2.3.3.1 *Pengenalan Pola*

Pengenalan pola dilakukan untuk mengenali suatu objek tertentu, misalnya untuk keperluan absensi menggunakan pengenalan sidik jari (*finger recognition*), pengenalan suara (*voice recognition*) atau pengenalan wajah (*face recognition*). Selain itu juga dapat digunakan sebagai mesin pencarian dengan gambar atau suara sebagai kata. Kuncinya (*keyword*) dengan menggunakan pengenalan gambar (*image recognition*) dan pengenalan ucapan (*speech recognition*).

2.3.3.2 *Sistem Pakar*

Sistem pakar digunakan sebagai sarana untuk menyimpan pengetahuan para pakar, sehingga komputer dapat menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar, misalnya analisis penyakit, kerusakan komputer, penasihat keuangan dan lain sebagainya.

2.3.3.3 *Game Playing*

Game merupakan fasilitas menarik yang terdapat dalam komputer. *Game* dalam komputer seperti catur, kartu, dan lainnya menggunakan kecerdasan buatan agar dapat bermain selayaknya manusia, memiliki strategi langkah selanjutnya, pengetahuan peraturan permainan dan dapat mengetahui pemenangnya.

2.3.3.4 *Robotika dan Sistem Sensor*

Salah satu contoh sistem sensor adalah pada mesin cuci yaitu menggunakan sensor optik, mengeluarkan cahaya ke air dan mengukur bagaimana cahaya tersebut sampai ke ujung lainnya. Makin kotor, maka sinar yang sampai makin

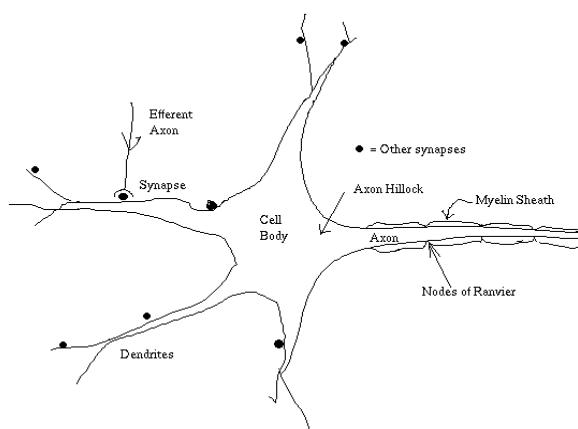
redup. Sistem juga mampu menentukan jenis kotoran tersebut. Sistem juga bisa menentukan putaran yang tepat secara otomatis berdasarkan jenis dan banyaknya kotoran serta jumlah yang akan dicuci.

2.4 Jaringan Syaraf Tiruan (JST)

2.4.1 Inspirasi Biologi

Jaringan syaraf tiruan keluar dari penelitian kecerdasan buatan, terutama percobaan untuk menirukan *fault-tolerance* dan kemampuan untuk belajar dari sistem syaraf biologi dengan model struktur *low-level* dari otak.

Otak terdiri dari sekitar (10.000.000.000) sel syaraf yang saling berhubungan. Sel syaraf mempunyai cabang struktur *input* (*dendrites*), sebuah inti sel dan percabangan struktur *output* (*axon*). *Axon* dari sebuah sel terhubung dengan *dendrites* yang lain melalui sebuah *synapses*. Ketika sebuah sel syaraf aktif, kemudian menimbulkan suatu sinyal *electrochemical* pada *axon*. Sinyal ini melewati *synapses* menuju ke sel syaraf yang lain. Sebuah sel syaraf lain akan mendapatkan sinyal jika memenuhi batasan tertentu yang sering disebut dengan nilai ambang atau *threshold* (Eliani, 2005).



Gambar 2.5. Susunan Saraf Manusia

2.4.2 Sejarah Jaringan Syaraf Tiruan

Menurut Siang (2005), sejarah jaringan syaraf tiruan adalah sebagai berikut.

- a. Sejarah jaringan syaraf tiruan pertama kali secara sederhana diperkenalkan oleh McCulloch dan Pitts pada tahun 1943. McCulloch dan Pitts menyimpulkan bahwa kombinasi beberapa *neuron* sederhana menjadi sebuah sistem *neural* akan meningkatkan kemampuan komputasinya. Bobot dalam jaringan yang diusulkan oleh McCulloch dan Pitts diatur untuk melakukan fungsi logika sederhana. Fungsi aktivasi yang dipakai adalah fungsi *threshold*.
- b. Pada tahun 1958, Rosenblatt memperkenalkan dan mulai mengembangkan model jaringan baru yang terdiri dari beberapa lapisan yang disebut *perceptron*. Metode pelatihan diperkenalkan untuk mengoptimalkan hasil iterasinya.
- c. Widrow dan Hoff (1960) mengembangkan *perceptron* dengan memperkenalkan aturan pelatihan jaringan, yang dikenal sebagai aturan delta (atau sering disebut kuadrat rata-rata terkecil). Aturan ini akan mengubah bobot *perceptron* apabila keluaran yang dihasilkan tidak sesuai dengan target yang diinginkan.
- d. Rumelhart (1986) mengembangkan *perceptron* menjadi *Backpropagation*, yang memungkinkan jaringan diproses melalui beberapa layer karena apa yang dilakukan peneliti terdahulu hanya menggunakan jaringan dengan layer tunggal (*single layer*).
- e. Selain itu, beberapa model jaringan syaraf tiruan lain juga dikembangkan oleh Kohonen (1972), Hopfield (1982), dan lain-lain. Pengembangan yang ramai dibicarakan sejak tahun 1990an adalah aplikasi model-model jaringan syaraf tiruan untuk menyelesaikan berbagai masalah di dunia nyata.

2.4.3 Definisi Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) didefinisikan sebagai suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai jaringan syaraf manusia (Hermawan, 2006). Beberapa istilah dalam jaringan syaraf tiruan yang sering ditemui adalah sebagai berikut.

- a. *Neuron* atau *Node* atau *Unit*: sel syaraf tiruan yang merupakan elemen pengolahan jaringan syaraf tiruan. Setiap *neuron* menerima data *input*, memproses *input* tersebut kemudian mengirimkan hasilnya berupa sebuah *output*.
- b. Jaringan: kumpulan *neuron* yang saling terhubung dan membentuk lapisan.
- c. Lapisan tersembunyi (*hidden layer*): lapisan yang tidak secara langsung berinteraksi dengan dunia luar. Lapisan ini memperluas kemampuan jaringan syaraf tiruan dalam menghadapi masalah-masalah yang kompleks.
- d. *Input*: sebuah nilai *input* yang akan diproses menjadi nilai *output*.
- e. *Output*: solusi dari nilai *input*.
- f. Bobot: nilai matematis dari sebuah koneksi antar *neuron*.
- g. Fungsi aktivasi: fungsi yang digunakan untuk mengupdate nilai-nilai bobot per-iterasi dari semua nilai *input*.
- h. Fungsi aktivasi sederhana adalah mengalikan *input* dengan bobotnya dan kemudian menjumlahkannya (disebut penjumlahan sigma) berbentuk linier atau tidak linier, dan sigmoid.
- i. Paradigma pembelajaran: bentuk pembelajaran, *supervised learning*, atau *unsupervised learning*.

Menurut Siang (2005:2), Jaringan Syaraf Tiruan (JST) adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi. JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa:

- a. pemrosesan informasi terjadi pada banyak elemen sederhana (*neuron*);
- b. sinyal dikirimkan diantara *neuron* melalui penghubung-penghubung;
- c. penghubung antar *neuron* memiliki bobot yang akan memperkuat atau memperlemah sinyal;
- d. untuk menentukan *output*, setiap *neuron* menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlahan *input* yang diterima. Besarnya *output* ini selanjutnya dibandingkan dengan suatu batas ambang.

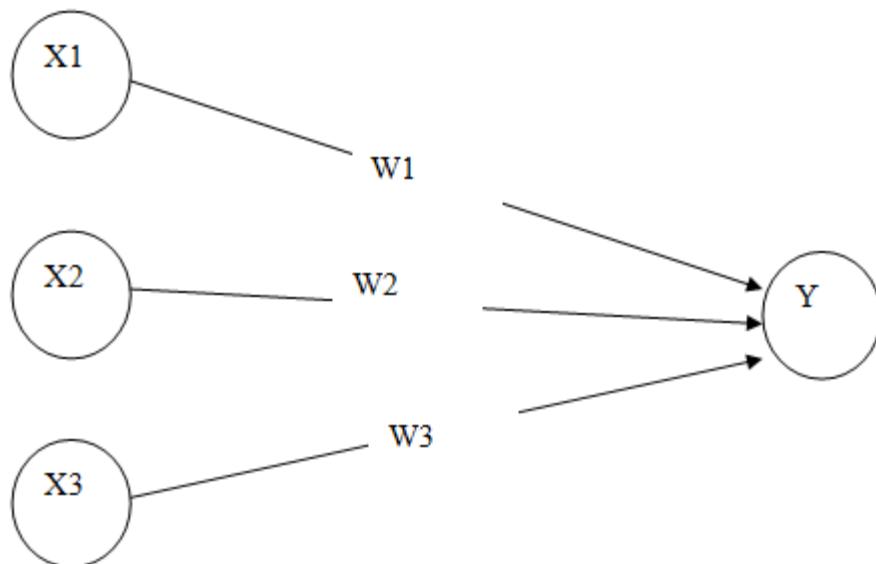
Jaringan syaraf tiruan ditentukan oleh 3 hal:

- a. pola hubungan antar *neuron* (disebut arsitektur jaringan);
- b. metode untuk menentukan bobot penghubung (disebut metode *training/learning/algoritma*);
- c. fungsi aktivasi (fungsi transfer).

Neuron dalam jaringan syaraf tiruan sering diganti dengan istilah simpul. Setiap simpul tersebut berfungsi untuk menerima atau mengirim sinyal ke simpul-simpul lainnya. Pengiriman sinyal disampaikan melalui penghubung. Kekuatan hubungan yang terjadi antara setiap simpul yang saling terhubung dikenal dengan nama bobot.

Arsitektur jaringan dan algoritma pelatihan sangat menentukan model-model jaringan syaraf tiruan. Arsitektur tersebut gunanya untuk menjelaskan arah

perjalanan sinyal atau data di dalam jaringan. Sedangkan algoritma belajar menjelaskan bagaimana bobot koneksi harus diubah agar pasangan masukan-keluaran yang diinginkan dapat tercapai. Dalam setiap perubahan harga bobot koneksi dapat dilakukan dengan berbagai cara, tergantung pada jenis algoritma pelatihan yang digunakan. Dengan mengatur besarnya nilai bobot ini diharapkan bahwa kinerja jaringan dalam mempelajari berbagai macam pola yang dinyatakan oleh setiap pasangan masukan-keluaran akan meningkat. Sebagai contoh, perhatikan *neuron Y* pada Gambar 2.6.



Gambar 2.6. Sebuah Sel Syaraf Tiruan

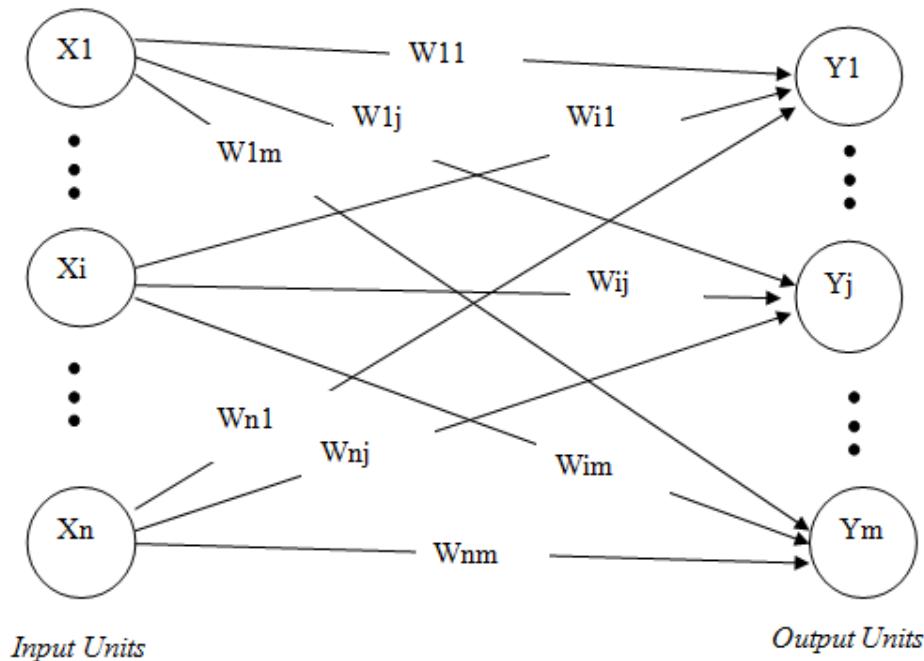
Y menerima *input* dari *neuron* X_1 , X_2 , dan X_3 dengan bobot hubungan masing-masing adalah W_1 , W_2 , dan W_3 . Ketiga *impuls* *neuron* yang ada dijumlahkan $net = X_1W_1 + X_2W_2 + X_3W_3$. Besarnya *impuls* yang diterima oleh Y mengikuti fungsi aktivasi $y = f(net)$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk merubah bobot.

2.4.4 Arsitektur Jaringan

Arsitektur jaringan syaraf tiruan digolongkan menjadi 3 model yaitu sebagai berikut.

2.4.4.1 Jaringan Layar Tunggal

Jaringan layar tunggal merupakan sekumpulan *input neuron* dihubungkan langsung dengan sekumpulan outputnya, seperti Gambar 2.7.



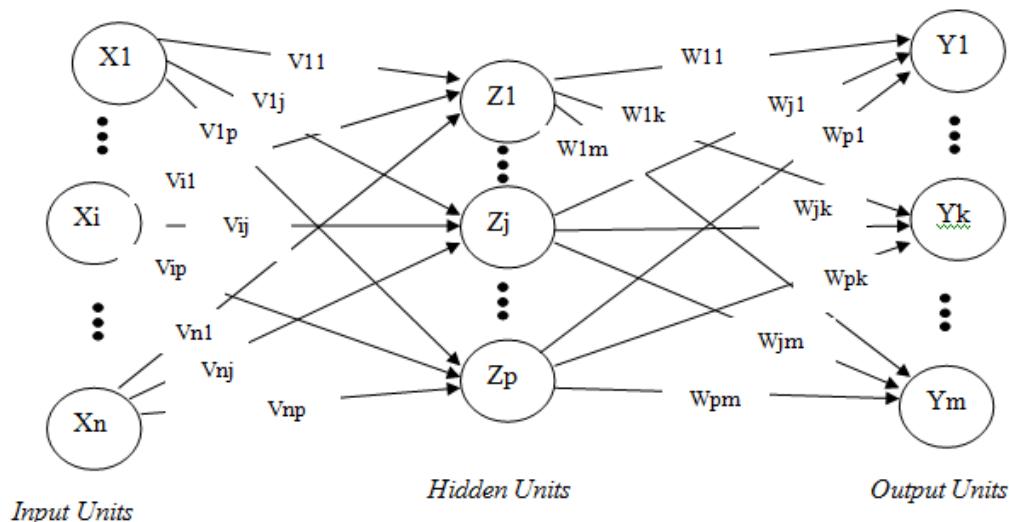
Gambar 2.7. Jaringan Layar Tunggal

Pada Gambar 2.7 diperlihatkan bahwa arsitektur jaringan layar tunggal dengan n buah masukan (X_1, X_2, \dots, X_n) dan m buah keluaran (Y_1, Y_2, \dots, Y_m). Dalam jaringan ini semua unit *input* dihubungkan dengan semua unit *output*.

2.4.4.2 Jaringan Layar Jamak

Jaringan layar jamak merupakan jaringan dengan n buah unit masukan (X_1, X_2, \dots, X_n), sebuah layar tersembunyi yang terdiri dari m buah unit

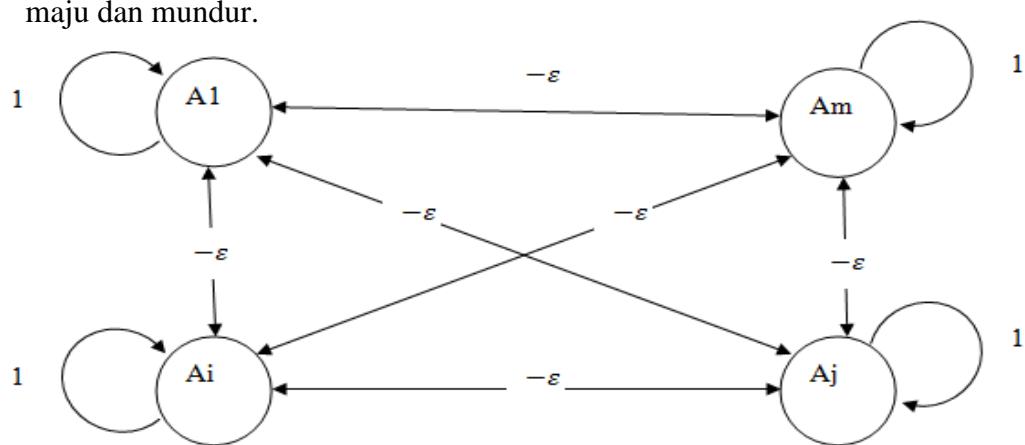
(Z_1, Z_2, \dots, Z_p) dan m buah unit keluaran (Y_1, Y_2, \dots, Y_m) . Jaringan layar jamak dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan layar tunggal, meskipun kadangkala proses pelatihan lebih kompleks dan lama. Arsitektur jaringan layar jamak dapat dilihat pada Gambar 2.8.



Gambar 2.8. Jaringan Layar Jamak

2.4.4.3 Jaringan Recurrent

Model jaringan *recurrent* mirip dengan jaringan layar tunggal ataupun ganda. Hanya saja, ada *neuron output* yang memberikan sinyal pada unit *input* (sering disebut *feedback loop*). Dengan kata lain sinyal mengalir dua arah, yaitu maju dan mundur.



Gambar 2.9. Jaringan Recurrent

2.5 *Backpropagation*

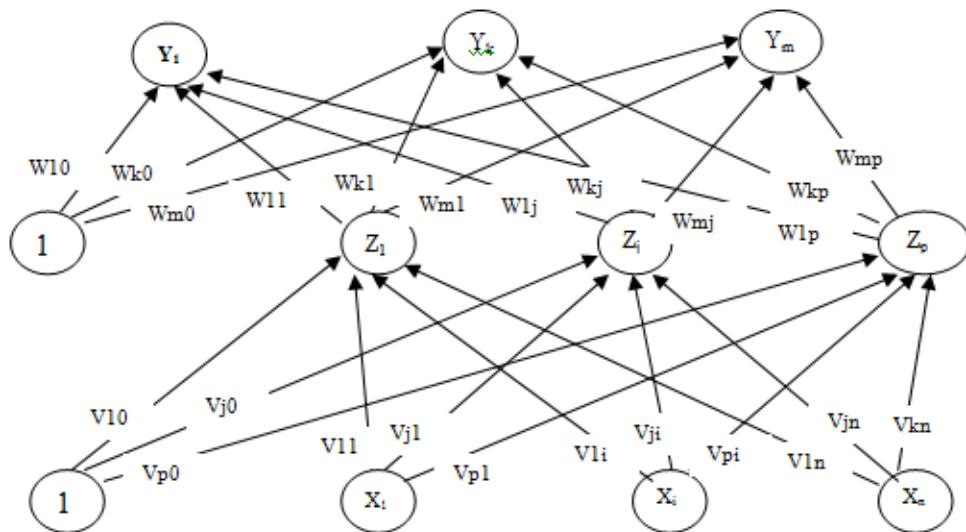
Backpropagation adalah salah satu metode dari jaringan syaraf tiruan yang dapat diaplikasikan dengan baik dalam bidang peramalan (*forecasting*). *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan mengenali pola yang digunakan selama *training* serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa namun tidak sama dengan pola yang dipakai selama pelatihan (Siang, 2005:119).

Menurut Kusumadewi (2003:116), dalam pelatihan dengan *backpropagation* sama halnya seperti pelatihan pada jaringan syaraf yang lain. Pada jaringan *feedforward* (umpan maju), pelatihan dilakukan dalam rangka perhitungan bobot sehingga pada akhir pelatihan akan diperoleh bobot-bobot yang baik. Selama proses pelatihan, bobot-bobot diatur secara iteratif untuk meminimumkan *error* (kesalahan) yang terjadi. Kesalahan dihitung berdasarkan rata-rata kuadrat kesalahan (MSE). Rata-rata kuadrat kesalahan juga dijadikan dasar perhitungan untuk kerja fungsi aktivasi.

Sebagian besar pelatihan untuk jaringan *feedforward* (umpan maju) menggunakan *gradient* dari fungsi aktivasi untuk menentukan bagaimana mengatur bobot-bobot dalam rangka meminimumkan kinerja. *Gradient* ini ditentukan dengan menggunakan suatu teknik yang disebut *backpropagation*. Pada dasarnya, algoritma pelatihan standar *backpropagation* akan menggerakkan bobot dengan arah *gradient* negatif. Prinsip dasar dari algoritma *backpropagation*

adalah memperbaiki bobot-bobot jaringan dengan arah yang membuat fungsi aktivasi menjadi turun dengan cepat.

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih lapisan tersembunyi (Siang, 2005:98). Gambar 2.10 adalah arsitektur *backpropagation* dengan n buah masukan (ditambah sebuah bias), sebuah lapisan tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah keluaran.



Gambar 2.10. Arsitektur *Backpropagation*

V_{ji} merupakan bobot garis dari unit masukan X_i ke unit lapisan tersembunyi Z_j (V_{j0} merupakan bobot garis yang menghubungkan bias di unit masukan ke unit lapisan tersembunyi Z_j). W_{kj} merupakan bobot dari lapisan tersembunyi Z_j ke unit keluaran Y_k (W_{k0} merupakan bobot dari bias dilayar tersembunyi ke unit keluaran Z_k).

2.5.1 Fungsi Aktivasi

Fungsi aktivasi jaringan syaraf tiruan *backpropagation* yang dipakai harus memenuhi beberapa syarat yaitu: kontinyu, terdiferensial dengan mudah, dan merupakan fungsi yang tidak turun (Siang, 2005:99). Fungsi aktivasi diharapkan jenuh (mendekati nilai-nilai maksimum dan minimum secara asimtot).

Beberapa fungsi yang digunakan menurut Puspitaningrum (2006:133), adalah sebagai berikut.

2.5.1.1 Fungsi Sigmoid Biner

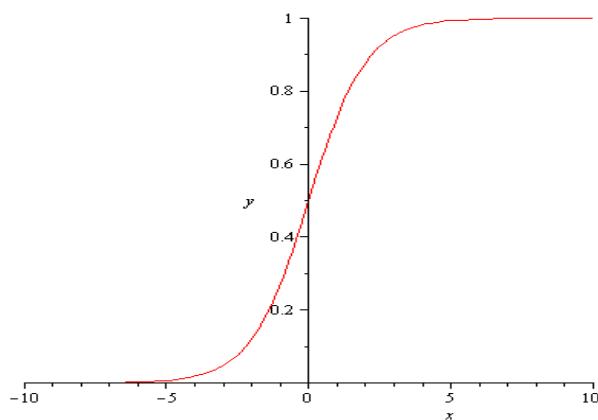
Fungsi ini merupakan fungsi yang umum digunakan. *Range*-nya adalah $(0,1)$ dan didefinisikan sebagai berikut:

$$f_1(x) = \frac{1}{1 + e^{-x}}$$

dengan turunan:

$$f_1'(x) = f_1(x)(1 - f_1(x))$$

Fungsi sigmoid biner diilustrasikan sebagai berikut:



Gambar 2.11. Fungsi Sigmoid Biner dengan *Range* $(0,1)$

2.5.1.2 Fungsi Sigmoid Bipolar

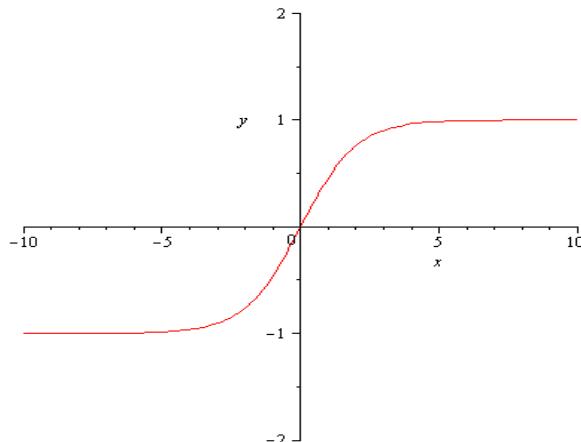
Fungsi sigmoid bipolar merupakan fungsi yang umum digunakan dan memiliki *range* $(-1,1)$ yang didefinisikan sebagai:

$$f_2(x) = 2f_1(x) - 1$$

dengan turunan:

$$f_2'(x) = \frac{1}{2}(1 + f_2(x))(1 - f_2(x))$$

Fungsi sigmoid bipolar diilustrasikan pada Gambar 2.12.



Gambar 2.12. Fungsi Sigmoid Bipolar dengan *Range* $(-1,1)$

2.5.1.3 Fungsi Tangen Hiperbolik

Fungsi tangen hiperbolik didefinisikan sebagai:

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

dengan turunan:

$$\tanh'(x) = (1 + \tanh(x))(1 - \tanh(x))$$

2.5.2 Pelatihan Standar *Backpropagation*

Ada 3 fase pelatihan *backpropagation* menurut Siang (2005:100-104) yaitu.

2.5.2.1 *Fase 1 (Propagasi Maju)*

Setiap sinyal masukan propagasi maju dihitung maju ke layar tersembunyi hingga layar keluaran dengan menggunakan fungsi aktivasi yang ditentukan. Selama tahap ini, setiap unit masukan (X_i) menerima sinyal masukan dan mengirim sinyal ini ke setiap unit tersembunyi Z_1, \dots, Z_p . Setiap unit tersembunyi kemudian menghitung aktivasinya dan mengirim sinyalnya (z_j) ke setiap unit keluaran. Setiap unit keluaran (Y_k) menghitung aktivasinya (y_k) untuk menunjukkan respon jaringan terhadap pola masukan yang diberikan.

2.5.2.2 *Fase 2 (Propagasi Mundur)*

Kesalahan (selisih antara keluaran jaringan dengan target yang diinginkan) yang terjadi dipropagasi mundur mulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Selama pelatihan, setiap unit keluaran membandingkan aktivasi y_k dengan targetnya t_k untuk menentukan galat antara pola masukan dengan unit keluaran tersebut. Setelah didapat galat, faktor δ_k ($k = 1, \dots, m$) dihitung. δ_k digunakan untuk mendistribusikan galat pada unit keluaran Y_k kembali ke seluruh unit pada lapis sebelumnya (unit tersembunyi yang terhubung dengan Y_k).

2.5.2.3 *Fase 3 (Perubahan Bobot)*

Fase ini dilakukan modifikasi bobot untuk menurunkan kesalahan yang terjadi. Galat yang diperoleh pada langkah 2 dipakai untuk mengubah bobot antara keluaran dengan lapisan tersembunyi. Menggunakan cara yang sama, faktor δ_j ($j = 1, \dots, p$) dihitung untuk setiap unit Z_j . Faktor δ_j digunakan untuk mengubah bobot seluruh antara lapisan tersembunyi dengan lapisan masukan.

Setelah faktor δ ditentukan, bobot untuk seluruh lapisan langsung disesuaikan.

Penyesuaian bobot W_{jk} (dari unit tersembunyi Z_j ke unit keluaran Y_k) didasarkan pada faktor δ_k dan aktivasi dari unit Z_j , yaitu z_j . Penyesuaian bobot v_{ij} (dari unit masukan X_i ke unit tersembunyi Z_j) adalah didasarkan pada faktor δ_j dan aktivasi unit masukan X_i . Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi.

2.5.3 Algoritma Pelatihan

Menurut Siang (2005:102-104), algoritma pelatihan untuk jaringan dengan satu *hidden layer* (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut.

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil.

Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-9.

Langkah 2 : Untuk setiap pasang data pelatihan, lakukan langkah 3-8.

Fase I : Propagasi Maju

Langkah 3 : Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi diatasnya.

Langkah 4 : Hitung semua keluaran di unit tersembunyi z_j ($j = 1, 2, \dots, p$):

$$z_{netj} = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$z_j = f(z_{netj}) = \frac{1}{1 + e^{-z_{netj}}}$$

Langkah 5 : Hitung semua keluaran jaringan di unit y_k ($k = 1, 2, \dots, m$):

$$y_{netk} = w_{k0} + \sum_{j=1}^p z_j w_{kj}$$

$$y_k = f(y_{netk}) = \frac{1}{1 + e^{-y_{netk}}}$$

Fase II: Propagasi Mundur

Langkah 6 : Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k = 1, 2, \dots, m$):

$$\delta_k = (t_k - y_k) f'(y_{netk}) = (t_y - y_k) y_k (1 - y_k)$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layar dibawahnya (langkah 7).

Hitung suku perubahan bobot W_{kj} (yang akan dipakai nanti untuk merubah bobot W_{kj}) dengan laju percepatan α :

$$\delta w_{kj} = \alpha \delta_k z_j \quad ; k = 1, 2, \dots, m \quad ; j = 0, 1, \dots, p$$

Langkah 7 : Hitung faktor δ unit tersembunyi berdasarkan kesalahan disetiap unit tersembunyi Z_j ($j = 1, 2, \dots, p$):

$$\delta_{netj} = \sum_{k=1}^m \delta_k w_{kj}$$

Faktor δ unit tersembunyi:

$$\delta_j = \delta_{netj} f'(z_{netj}) = \delta_{netj} z_j (1 - z_j)$$

Hitung suku perubahan bobot V_{ij} (yang dipakai nanti untuk merubah bobot V_{ij}):

$$\delta v_{ij} = \alpha \delta_j x_i \quad ; j = 1, 2, \dots, p \quad ; i = 0, 1, \dots, n$$

Fase III: Perubahan Bobot

Langkah 8 : Hitung semua perubahan bobot.

Perubahan bobot garis yang menuju ke unit keluaran:

$$w_{jk} (\text{baru}) = w_{jk} (\text{Lama}) + \delta w_{jk}$$

Dari unit ke-1 sampai unit ke- p dilapisan tersembunyi juga dilakukan peng-update-an pada bias dan bobotnya ($i = 1, 2, \dots, n; j = 1, \dots, p$):

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \delta v_{ij}$$

Langkah 9 : Test kondisi berhenti.

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Langkah propagasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan. Apabila fungsi aktivasi yang dipakai bukan sigmoid biner, maka langkah 4 dan 5 harus disesuaikan, demikian juga turunannya pada langkah 6 dan 7.

Beberapa kasus pelatihan yang dilakukan memerlukan iterasi yang banyak sehingga membuat proses pelatihan menjadi lama. Iterasi dapat dipercepat dengan parameter α atau laju pemahaman. Nilai α terletak antara 0 dan 1 ($0 \leq \alpha \leq 1$). Jika harga α semakin besar, maka iterasi yang dipakai semakin sedikit. Akan tetapi jika harga α terlalu besar, maka akan merusak pola yang sudah benar sehingga pemahaman menjadi lambat.

Proses pelatihan yang baik dipengaruhi pada pemilihan bobot awal, karena bobot awal sangat mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya. Oleh karena itu dalam standar *backpropagation*, bobot dan bias diisi dengan bilangan acak kecil dan biasanya bobot awal diinisialisasi secara *random* dengan nilai antara -0,5 sampai 0,5 (atau -1 sampai 1 atau interval yang lainnya).

2.5.4 Momentum

Standar perubahan bobot *backpropagation* didasarkan atas *gradient* yang terjadi untuk pola yang dimasukkan saat itu. Modifikasi dilakukan dengan merubah bobot yang didasarkan atas arah *gradient* pola terakhir dan pola sebelumnya (momentum) yang dimasukkan. Jadi perhitungannya tidak hanya pola masukan terakhir saja.

Momentum ditambahkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan data yang lain. Jika beberapa data terakhir yang diberikan ke jaringan memiliki pola serupa (berarti arah *gradient* sudah benar), maka perubahan bobot dilakukan secara cepat. Namun jika data terakhir yang dimasukkan memiliki pola yang berbeda dengan pola sebelumnya, maka perubahan bobot dilakukan secara lambat.

Penambahan momentum, bobot baru pada waktu ke $(t + 1)$ didasarkan atas bobot pada waktu t dan $(t - 1)$. Langkah ini harus ditambahkan dua variabel yang mencatat besarnya momentum untuk dua iterasi terakhir. Jika μ adalah konstanta $(0 \leq \mu \leq 1)$ yang menyatakan parameter momentum maka bobot baru dihitung berdasarkan persamaan berikut:

$$w_{kj}(t + 1) = w_{kj}(t) + \alpha \delta_k z_j + \mu (w_{kj}(t) - w_{kj}(t - 1))$$

dan

$$v_{ji}(t + 1) = v_{ji}(t) + \alpha \delta_j x_i + \mu (v_{ji}(t) - v_{ji}(t - 1))$$

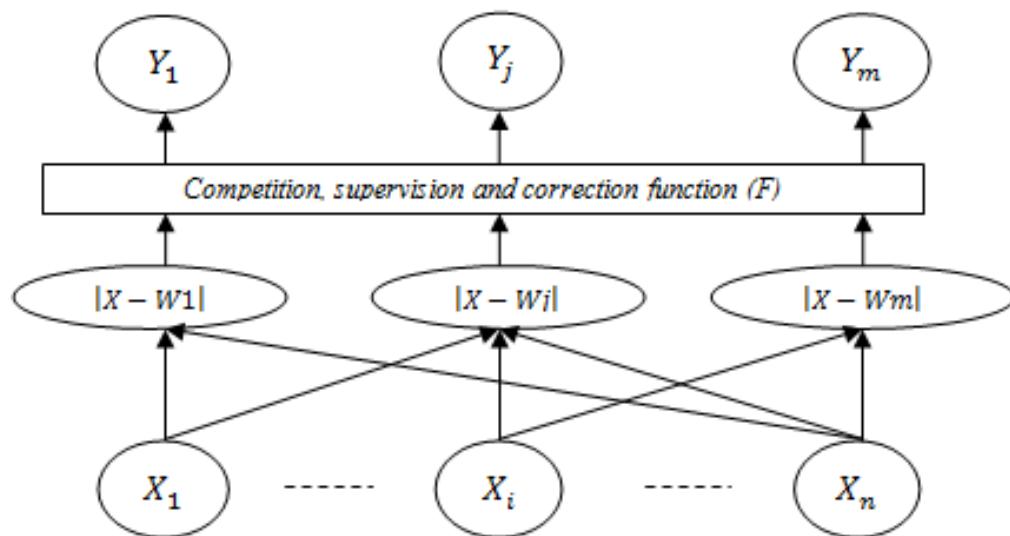
2.6 Learning Vector Quantization (LVQ)

LVQ adalah jaringan *single layer* yang terdiri dari dua lapisan yaitu lapisan *input* dan *output* (Nurkhozin, dkk, 2011). Menurut Kusumadewi (2003:258), *Learning Vector Quantization* (LVQ) adalah suatu metode untuk melakukan

pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor *input*. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor *input*. Jika 2 vektor *input* mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor *input* tersebut kedalam kelas yang sama.

2.6.1 Arsitektur LVQ

Menurut Putro (2011), arsitektur LVQ sama halnya dengan SOM (*Self Organizing Map*), LVQ juga terdiri dari 2 lapisan, *input* (X) dan *output* (Y), dimana antara lapisannya dihubungkan oleh bobot tertentu yang sering disebut sebagai vektor pewakil (W). Informasi yang diberikan ke jaringan pada saat pembelajaran bukan hanya vektor data saja melainkan informasi kelas dari data juga ikut dimasukkan.



Gambar 2.13. Arsitektur LVQ

Keterangan:

X = vektor masukan (X_1, \dots, X_n)
 W = vektor bobot atau vektor pewakil
 $|X-W|$ = selisih nilai jarak *Euclidian* antara vektor *input* dengan vektor bobot
 F = lapisan kompetitif
 Y = keluaran (*output*)

Ketika hasil pemrosesan jaringan memberikan hasil klasifikasi yang sama dengan informasi kelas yang diberikan di awal, maka vektor pewakil akan disesuaikan agar lebih dekat dengan vektor masukan. Sebaliknya ketika hasil klasifikasi tidak sama dengan informasi kelas yang diberikan di awal, maka vektor pewakil akan disesuaikan agar menjauhi vektor masukan.

2.6.2 Algoritma LVQ, yaitu sebagai berikut.

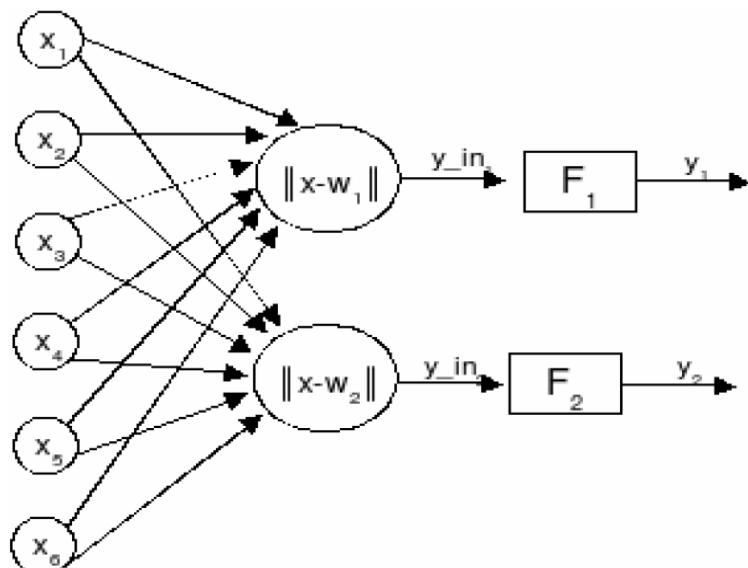
- a. Tetapkan: bobot (W), maksimum *epoch* (*max. epoch*), *error* minimum yang diharapkan (eps), *learning rate* (α).
- b. Masukkan:
 - 1) *input* : $X(m, n)$;
 - 2) target : $T(1, n)$.
- c. Tetapkan kondisi awal:
 - 1) $epoch = 0$;
 - 2) $error = 1$.
- d. Kerjakan jika ($epoch < max. epoch$) atau ($\alpha > eps$):
 - 1) $epoch = epoch + 1$;
 - 2) kerjakan untuk $i = 1$ sampai n .
- e. Tentukan $/$ sedemikian hingga $\|X - W_j\|$ minimum (sebut sebagai C_j).
- f. Perbaiki W_j dengan ketentuan:

1) jika $T = C_j$ maka:

$$W_j(\text{baru}) = W_j(\text{lama}) + \alpha(x - W_j(\text{lama}));$$

2) jika $T \neq C_j$ maka:

$$W_j(\text{baru}) = W_j(\text{lama}) - \alpha(x - W_j(\text{lama})).$$



Gambar 2.14. Arsitektur Jaringan LVQ

2.7 Matrix Laboratory (MATLAB)

Pengenalan citra *barcode* memerlukan suatu *software* yang dapat membantu mengimplementasikan pengenalan citra menjadi sebuah angka-angka yang selanjutnya dapat diolah menjadi suatu informasi. *Software* yang digunakan dalam penilitian ini adalah *software Matrix Laboratory* (MATLAB) untuk mencari penyelesaian masalah pengenalan pola dari suatu data yang telah disiapkan.

Menurut Iqbal (2009), Matlab adalah sebuah bahasa dengan (*high-performance*) kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model

yang sangat mudah untuk dipakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar. Penggunaan Matlab meliputi bidang–bidang:

1. matematika dan komputansi;
2. pembentukan *algorithm*;
3. akusisi data;
4. pemodelan, simulasi, dan pembuatan *prototype*;
5. analisa data, eksplorasi, dan visualisasi;
6. grafik keilmuan dan bidang rekayasa.

2.7.1 *Window-window* pada Matlab

Ada beberapa macam *window* yang tersedia dalam Matlab, yang dapat dijelaskan sebagai berikut.

2.7.1.1 *Matlab Command Window/editor*

Matlab *command window/editor* merupakan *window* yang dibuka pertama kali setiap kali Matlab dijalankan. Pada Matlab *command window* dapat dilakukan akses-akses ke *command-command* Matlab dengan cara mengetikkan barisan-barisan ekspresi Matlab, seperti mengakses *help window* dan lain-lainnya.

Command windows juga digunakan untuk memanggil *tool* Matlab seperti *editor*, *debugger* atau fungsi. Ciri dari *window* ini adalah adanya *prompt* (>>) yang menyatakan Matlab siap menerima perintah. Perintah dapat berupa fungsi-fungsi pengaturan *file* (seperti perintah DOS/UNIX) maupun fungsi-fungsi bawaan/*toolbox* Matlab sendiri. Berikut ini beberapa fungsi pengaturan *file* dalam Matlab:

`dir/ls` : digunakan untuk melihat isi dari sebuah direktori aktif,
`cd` : digunakan untuk melakukan perpindahan dari direktori aktif,
`pwd` : digunakan untuk melihat direktori yang sedang aktif,
`mkdir` : digunakan untuk membuat sebuah direktori,
`what` : digunakan untuk melihat nama *file m* dalam direktori aktif,
`who` : digunakan untuk melihat variabel yang sedang aktif,
`whos` : digunakan untuk menampilkan nama setiap variabel,
`delete` : digunakan untuk menghapus *file*,
`clear` : digunakan untuk menghapus variabel,
`clc` : digunakan untuk membersihkan layar,
`doc` : digunakan untuk melihat dokumentasi *The MathWorks, Inc.* dalam format html secara *online*, dan
`demo` : digunakan untuk mencoba beberapa tampilan demo yang disediakan oleh Matlab.

2.7.1.2 *Matlab Editor/Debugger (Editor M-File/Pencarian Kesalahan)*

Matlab *editor/debugger* merupakan *tool* yang disediakan oleh Matlab 5 ke atas. Berfungsi sebagai *editor script* Matlab (*M-file*). Cara lain untuk *editor script* Matlab (*M-file*) adalah menggunakan *notepad*, *wordpad* bahkan *microsoft word*.

2.7.1.3 *Figure Windows*

Figure windows adalah hasil visualisasi dari *script* Matlab, namun Matlab memberi kemudahan bagi *programmer* untuk mengedit *window* ini sekaligus memberikan program khusus untuk itu. *Window* ini selain berfungsi sebagai visualisasi *output* dapat juga sekaligus menjadi media *input* yang interaktif.

2.7.1.4 Matlab Help Window

Matlab menyediakan sistem *help* yang dapat diakses dengan perintah *help*. Misalnya, untuk memperoleh informasi mengenai fungsi *elfun* yaitu fungsi untuk trigonometri, eksponensial, *kompleks* dan lain-lain.

2.7.2 Kelengkapan pada Sistem Matlab

Sebagai sebuah sistem, Matlab tersusun dari lima bagian utama yaitu sebagai berikut (Iqbal, 2009).

2.7.2.1 Development Environment

Development environment merupakan sekumpulan perangkat dan fasilitas yang membantu untuk menggunakan fungsi-fungsi dan *file-file* Matlab. Beberapa perangkat ini merupakan sebuah *Graphical User Interfaces* (GUI). Termasuk di dalamnya adalah Matlab *desktop* dan *command window*, *command history*, sebuah *editor* dan *debugger*, dan *browsers* untuk melihat *help*, *workspace*, *files*, dan *search path*.

2.7.2.2 Matlab Mathematical Function Library

Matlab *mathematical function library* merupakan sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar, seperti *sum*, *sin*, *cos*, dan *complex arithmetic*, sampai dengan fungsi-fungsi yang lebih kompleks seperti *matrix inverse*, *matrix eigenvalues*, *bessel functions*, dan *fast fourier transforms*.

2.7.2.3 Matlab Language

Matlab *language* merupakan suatu *high-level matrix/array language* dengan *control flow statements*, *functions*, *data structures*, *input/output*, dan fitur-fitur *object-oriented programming*. Fitur ini memungkinkan bagi kita untuk melakukan

kedua hal baik "pemrograman dalam lingkup sederhana" untuk mendapatkan hasil yang cepat, dan "pemrograman dalam lingkup yang lebih besar" untuk memperoleh hasil-hasil dan aplikasi yang kompleks.

2.7.2.4 *Graphics*

Matlab memiliki fasilitas untuk menampilkan vektor dan matriks sebagai suatu grafik. Di dalamnya melibatkan *high-level functions* (fungsi-fungsi tingkat tinggi) untuk visualisasi data dua dimensi dan data tiga dimensi, *image processing, animation, dan presentation graphics*. Selain itu di dalam Matlab juga melibatkan fungsi level rendah yang memungkinkan bagi pengguna membiasakan diri untuk memunculkan grafik mulai dari bentuk yang sederhana sampai dengan tingkatan *graphical user interface* pada aplikasi Matlab.

2.7.2.5 *Matlab Application Program Interface (API)*

API merupakan suatu *library* yang memungkinkan program yang telah ditulis dalam bahasa *C* dan *Fortran* mampu berinteraksi dengan Matlab. Fasilitas ini untuk pemanggilan *routines* dari Matlab (*dynamic linking*), pemanggilan Matlab sebagai contoh *computational engine*, dan untuk membaca dan menuliskan mat-*files*.

BAB III

METODE PENELITIAN

Rencana penelitian harus logis, diikuti oleh unsur-unsur yang urut, konsisten, dan operasional, menyangkut bagaimana penelitian tersebut akan dijalankan (Suharto dkk, 2004:98). Langkah-langkah yang digunakan pada penelitian ini adalah sebagai berikut.

3.1 Perumusan Masalah

Setelah masalah diidentifikasi, dipilih, maka perlu dirumuskan (Fathoni, 2006:16). Langkah ini dimaksudkan untuk memperjelas permasalahan sehingga mempermudah pembahasan selanjutnya.

3.2 Jenis dan Sumber Data

Jenis data yang digunakan dalam penelitian ini adalah data primer. Data primer yaitu data yang langsung dikumpulkan oleh peneliti dari sumber pertama (Fathoni, 2006:38). Data diperoleh dengan cara mengambil foto-foto *barcode* yang tertera pada produk di supermarket GIANT jalan Siliwangi Semarang untuk melakukan pelatihan program dan sebagai data uji sistem.

3.3 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan adalah sebagai berikut.

3.3.1. Studi Pustaka

Menurut Fathoni (2006), penelitian pustaka adalah sesuatu penelitian yang dilakukan di ruang perpustakaan untuk menghimpun dan menganalisis data yang bersumber dari perpustakaan, baik berupa buku, periodikal, seperti majalah ilmiah yang diterbitkan secara berkala, kisah sejarah, dokumen dan materi perpustakaan lainnya, yang dapat dijadikan sumber rujukan untuk menyusun suatu laporan ilmiah.

Langkah studi pustaka ini, digunakan sumber pustaka yang relevan untuk mengumpulkan informasi yang diperlukan dalam penelitian, yaitu dengan mengumpulkan sumber pustaka yang berupa buku, artikel ilmiah, jurnal dan prosiding seminar nasional. Setelah sumber pustaka terkumpul, dilanjutkan dengan penelaahan dari sumber pustaka tersebut. Pada akhirnya sumber pustaka itu dijadikan landasan untuk menganalisis permasalahan.

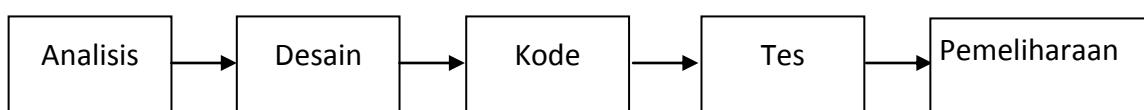
Studi pustaka yang berkaitan dengan permasalahan yang ada pada skripsi ini yaitu tentang pengenalan *barcode*, pengolahan citra digital, *artificial intelligence*, jaringan syaraf tiruan, *backpropagation*, *Learning Vector Quantization* (LVQ), dan *Matrix Laboratory (Matlab)*, yang akan dimanfaatkan untuk membantu menyelesaikan masalah yang ada.

3.3.2 Metode Survei

Pengumpulan data secara langsung dengan cara pengamatan langsung dan analisa dalam jangka waktu tertentu, dengan mengambil semua data yang dibutuhkan. Data yang diperoleh berupa foto-foto *barcode* yang tertera pada produk di supermarket GIANT jalan Siliwangi Semarang.

3.4 Proses Pembuatan Perangkat Lunak

Membuat sistem pengenalan citra *barcode* dengan metode *backpropagation* dan metode *learning vector quantization* bertujuan untuk membantu menyelesaikan masalah yang ada dalam penelitian ini. Proses pembuatan perangkat lunak menggunakan model sekuensial linier, yang dapat dilihat pada Gambar 3.1..



Gambar 3.1 Model Sekuensial Linier

Menurut Pressman (2002:37), sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan. Proses-proses pembuatan perangkat lunak model sekuensial linier yang dilakukan pada penelitian ini adalah sebagai berikut.

a. *Analisis*

Merupakan tahap menganalisis hal-hal yang diperlukan dalam pelaksanaan pembuatan perangkat lunak.

b. *Design*

Tahap penerjemahan dari data yang dianalisis kedalam bentuk yang mudah dimengerti oleh *user*.

c. *Coding*

Tahap penerjemahan data atau pemecahan masalah yang telah dirancang keadalam bahasa pemrograman tertentu.

d. Pengujian

Merupakan tahap pengujian terhadap perangkat lunak yang dibangun.

e. *Maintenance* (Pemeliharaan)

Tahap akhir dimana suatu perangkat lunak yang sudah selesai dapat mengalami perubahan-perubahan atau penambahan sesuai dengan permintaan user.

3.5 Pemecahan Masalah

Berdasarkan permasalahan yang diangkat pada penelitian ini, maka disusun langkah-langkah untuk memecahkan masalah tersebut. Langkah-langkah tersebut diuraikan sebagai berikut.

3.5.1 Mengambil Foto-foto *Barcode* Produk di Supermarket GIANT Jalan Siliwangi Semarang untuk Data Pelatihan Sistem

Pengambilan citra *barcode* dilakukan secara langsung dengan mengambil sampel *barcode* dari supermarket GIANT jalan Siliwangi Semarang. Citra diambil dengan menggunakan kamera digital *merk* Fujifilm tanpa menggunakan bantuan alat lain seperti pencahayaan.

3.5.2 Melakukan *Pre Processing* Citra

Pre processing dilakukan untuk memperoleh citra yang selanjutnya dapat dimanfaatkan untuk sistem yang akan dibuat. Proses ini meliputi *cropping*, normalisasi ukuran citra, mengubah citra menjadi *grayscale*, penajaman kualitas citra, dan deteksi tepi sobel, dilanjutkan dengan proses ekstraksi ciri, untuk mendapatkan data karakteristik ciri dari citra *barcode*.

3.5.3 Membuat Sistem Pengenalan Citra *Barcode* dengan Metode *Backpropagation* dan Metode *Learning Vector Quantization* Menggunakan Software Matlab R2009a

Setelah data citra telah diperoleh, langkah selanjutnya adalah merancang sistem pengenalan untuk mengenali citra *barcode*. Sistem pengenalan ini dirancang dengan menggunakan *software* Matlab R2009a. *Software* ini memiliki *tools-tools* yang dapat memudahkan dalam proses pembuatan program khususnya jaringan syaraf tiruan.

3.5.4 Melakukan Pengujian Sistem dengan Menggunakan Foto *Barcode* yang Beredar di Supermarket Wilayah Kota Semarang

Langkah ini dilakukan untuk menguji sistem yang telah dibangun. Pengujian sistem dilakukan dengan menghitung tingkat keakuratan sistem dalam mengenali citra *barcode*. Langkah yang harus ditempuh yaitu melakukan variasi pada *learning rate* dan jumlah *hidden layer*. Arsitektur jaringan yang paling efektif, dapat dilihat dari *Mean Square Error* (MSE) dan waktu pembelajaran dari proses pelatihan serta melihat hasil *recognition rate*.

3.5.5 Membandingkan Hasil Pengenalan *Barcode* Menggunakan Metode *Backpropagation* dengan Metode *Learning Vector Quantization* dan Memberikan Rekomendasi Metode yang Paling Akurat untuk Melakukan Pengenalan Citra *Barcode*

Setelah didapatkan hasil dari proses pengujian sistem baik itu hasil dari pengenalan *barcode* menggunakan metode *backpropagation* maupun dengan metode *learning vector quantization*, selanjutnya akan dilakukan proses

perbandingan. Dalam tahap ini akan diperoleh persentase hasil pengenalan citra *barcode* yang paling tepat dan akurat. Tujuannya adalah untuk memberikan rekomendasi metode jaringan syaraf tiruan yang paling tepat untuk mengenali citra *barcode*.

3.6 Prosedur Penelitian

Prosedur penelitiannya adalah sebagai berikut.

3.6.1 Mengumpulkan Materi-materi tentang Permasalahan yang Sedang Diteliti

Langkah ini dimulai dengan mengumpulkan bahan-bahan atau materi yang akan digunakan sebagai dasar penelitian. Materi yang digunakan berupa buku, jurnal, artikel ilmiah, dan prosiding seminar nasional.

3.6.2 Mengumpulkan Data-data

Langkah ini merupakan langkah untuk mengumpulkan data-data yang dibutuhkan dalam penelitian. Data-data yang diperlukan dalam penelitian ini adalah citra *barcode* produk yang beredar di supermarket GIANT jalan Siliwangi Semarang. Data-data ini nantinya akan digunakan sebagai data latih dan data uji untuk sistem pengenalan citra *barcode* yang akan dibuat dalam penelitian ini.

3.6.3 Membuat Sistem Pengenalan Menggunakan *Software Matlab R2009a*

Setelah proses pengambilan data-data untuk penelitian selesai, langkah berikutnya adalah merancang sistem untuk mengenali citra *barcode* dengan metode *backpropagation* dan metode *learning vector quantization*. *Software* yang digunakan untuk membuat sistem ini adalah *software Matlab R2009a*.

3.6.4 Menguji Sistem

Langkah ini merupakan langkah untuk mengetahui tingkat akurasi dari sistem yang telah dibuat. Termasuk yang akan diteliti adalah pengaruh variasi *learning rate* dan jumlah *neuron hidden layer* dalam proses pembelajaran sistem. Sehingga akan mempengaruhi MSE, lama pembelajaran, dan hasil pengenalan.

3.6.5 Membandingkan Hasil Sistem Pengenalan Barcode Menggunakan Metode *Backpropagation* dengan Metode *Learning Vector Quantization*

Sistem telah selesai diuji, langkah selanjutnya adalah membandingkan antara hasil pengujian sistem pengenalan *barcode* menggunakan metode *backpropagation* dengan hasil pengujian sistem pengenalan *barcode* menggunakan metode *learning vector quantization*. Langkah ini untuk mendapatkan hasil pengenalan paling baik dan akurat. Hasil perbandingan tersebut digunakan untuk memberikan rekomendasi metode yang paling tepat untuk mengenali citra *barcode*.

3.7 Penarikan Simpulan

Langkah ini merupakan langkah terakhir dari penelitian. Penarikan simpulan didasarkan pada studi pustaka dan pembahasan permasalahan serta merupakan hasil analisis dari penelitian. Simpulan yang diperoleh adalah mengetahui bahwa metode jaringan syaraf tiruan *backpropagation* dan *learning vector quantization* dapat digunakan untuk pengenalan citra *barcode*. Selain itu, dapat mengetahui besar tingkat akurasi sistem yang dihasilkan pada proses pengenalan citra

barcode baik dengan metode *backpropagation* maupun menggunakan metode *Learning Vector Quantization* (LVQ). Langkah ini bertujuan untuk memperoleh metode jaringan syaraf tiruan yang paling akurat dan tepat dalam mengenali citra *barcode*.

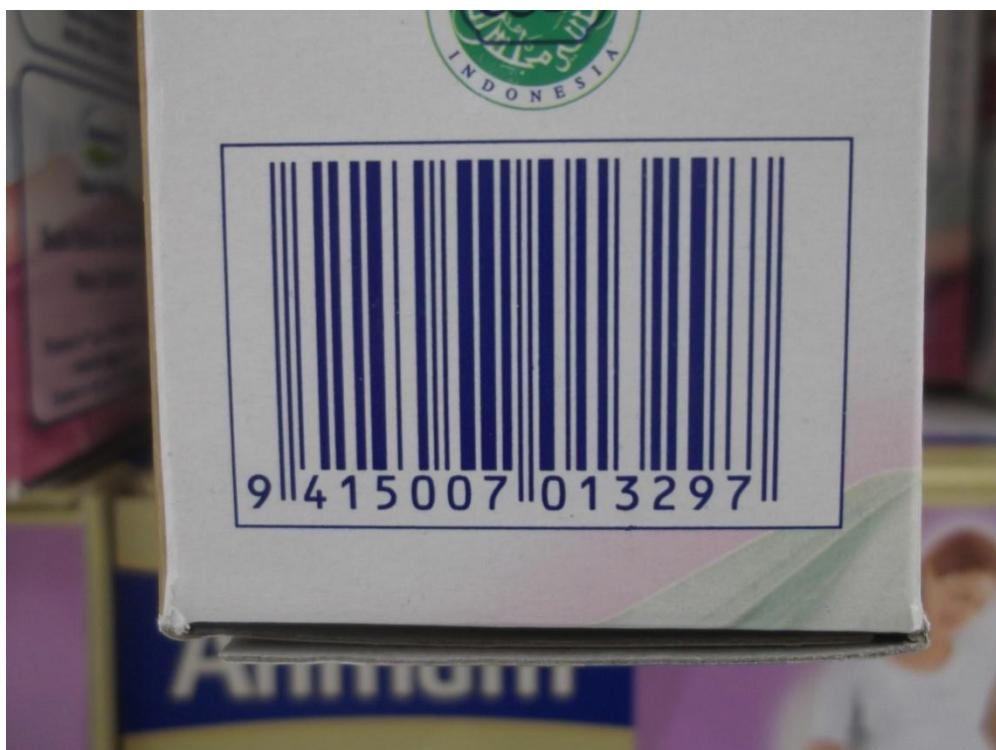
BAB IV

HASIL DAN PEMBAHASAN

Sistem pengenalan citra *barcode* ini dibuat melalui beberapa proses. Berikut adalah beberapa proses yang harus dikerjakan.

4.1 Akuisisi Citra (*Image Acquition*)

Akuisisi citra (*image acquisition*) merupakan proses pengambilan citra dengan menggunakan sebuah alat bantu pengambil gambar. Kamera digital *merk* Fujifilm dengan resolusi 16 *megapixel* digunakan untuk pengambilan gambar. Citra diambil dalam jarak antara 5 cm sampai dengan 10 cm tanpa menggunakan penambahan cahaya. Citra yang diperoleh berupa citra warna (RGB) dengan ukuran 4608×3440 *pixel* seperti contoh pada Gambar 4.1.



Gambar 4.1 Citra Barcode EAN-13

Perlu menggunakan alat khusus yang mampu mengambil citra secara homogen baik dari segi pencahayaan maupun jarak kamera dengan objek yang akan diambil gambarnya. Akan tetapi pada penelitian ini hanya dibatasi tidak sampai membuat alat yang dimaksud.

4.2 Perancangan Sistem

Tahap perancangan sistem bertujuan untuk mempersiapkan segala sesuatu yang dibutuhkan dan berhubungan dengan pembuatan sistem. Tahapan perancangan sistem meliputi serangkaian proses sebagai berikut.

4.2.1 Perancangan Sistem Pelatihan

Sistem pelatihan merupakan bagian awal dari sistem pengenalan citra *barcode*. Sistem ini berfungsi untuk menyiapkan informasi-informasi yang akan digunakan dalam proses pengenalan citra *barcode*. Proses ini memiliki beberapa tahapan penting sebagai berikut.

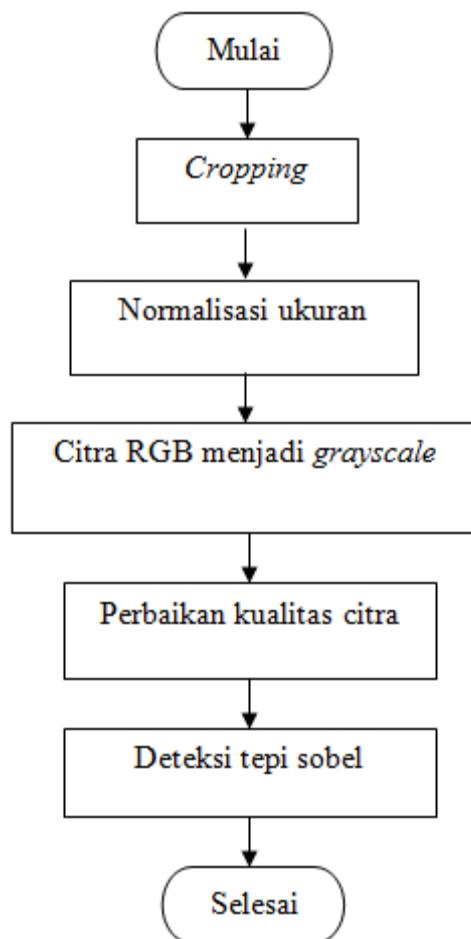
4.2.1.1 Pembagian Citra

Jumlah data yang digunakan dalam penelitian ini adalah 200 citra *barcode* EAN-13 yang semuanya diambil dari supermarket GIANT jalan Siliwangi Semarang. Selanjutnya 200 citra *barcode* tersebut dibagi menjadi dua bagian yaitu 100 citra *barcode* digunakan sebagai citra latih, yakni citra yang akan digunakan untuk pembelajaran JST. Sedangkan 100 citra *barcode* yang lain dimanfaatkan sebagai citra *query* yaitu citra uji yang tidak melalui proses pelatihan.

Proses pengujian nanti akan digunakan keseluruhan citra baik citra latih maupun citra non-latih, sehingga tingkat akurasi sistem secara keseluruhan dapat diukur.

4.2.1.2 Pre Processing

Serangkaian proses *pre processing* yang dilakukan pada penilitian ini meliputi proses *cropping*, normalisasi ukuran, mengubah citra RGB menjadi citra *grayscale*, melakukan perbaikan citra, dan deteksi tepi sobel. *Flowchart pre processing* dapat dilihat pada Gambar 4.2.



Gambar 4.2. Flowchart Pre Processing

4.2.1.2.1 *Cropping*

Cropping merupakan proses memotong citra dan mengambil bagian dari citra yang dibutuhkan. Gambar *barcode* yang diambil merupakan gambar yang umum, artinya tidak semua komponen gambar dibutuhkan dalam sistem yang akan dirancang, sehingga perlu proses *cropping*. Proses ini menggunakan *software Microsoft Paint*.

Proses *cropping* citra menggunakan spesifikasi sebagai berikut.

Tabel 4.1 Spesifikasi *Hardware* dan *Software* dalam Proses *Cropping*

| | | |
|-----------------|----------------------|-------------------------------------|
| Perangkat Keras | Processor | : Intel® Core™ i3-380M @ 500 GB HDD |
| | Memori | : 2 GB |
| | Piranti Masukan | : Mouse, Keyboard |
| Perangkat Lunak | Sistem Operasi | : Microsoft Windows 7 Ultimate |
| | Perangkat Pengembang | : Microsoft paint |

4.2.1.2.2 Normalisasi Ukuran

Normalisasi ukuran merupakan proses untuk menyeragamkan ukuran citra. Hal ini dimaksudkan agar citra yang akan diolah dalam sistem mempunyai ukuran yang sama. Citra masukan yang digunakan adalah citra RGB dengan ukuran 4608×3440 pixel. Pada citra yang seperti ini, matriks yang dihasilkan akan berukuran besar. Proses komputasi dimensi matriks yang besar membutuhkan waktu proses yang lama sehingga diperlukan reduksi dimensi matriks untuk meminimalisir waktu proses. Ukuran citra dalam penelitian ini diubah menjadi 100×60 pixel.

4.2.1.2.3 Mengubah Citra RGB Menjadi Citra *Grayscale*

Data yang diinputkan dalam sistem awalnya adalah gambar tipe RGB. Citra RGB akan membentuk vektor 3 lapis sehingga akan sulit untuk dilakukan proses selanjutnya. Citra perlu diubah tipenya menjadi *grayscale* untuk mempermudah proses pengolahan citra pada proses selanjutnya.

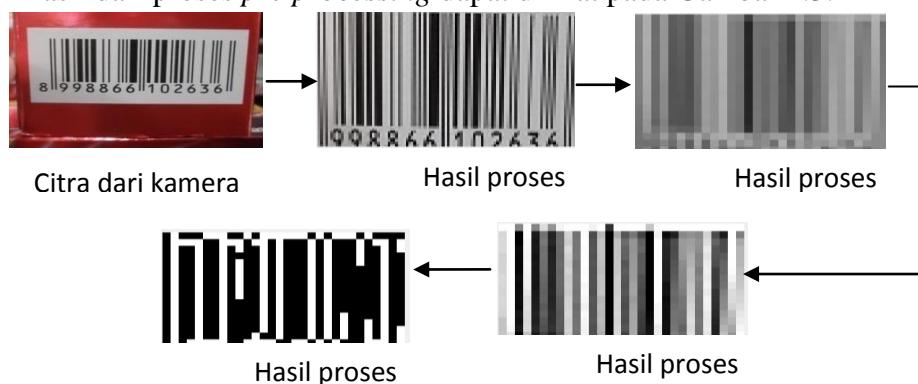
4.2.1.2.4 Perbaikan Citra

Setelah citra diubah menjadi *grayscale*, selanjutnya citra dipertajam untuk memperoleh dan memperjelas citra yang akan digunakan pada proses selanjutnya. Langkah ini dimaksudkan agar pola garis-garis pada *barcode* tampak lebih jelas.

4.2.1.2.5 Deteksi Tepi

Proses deteksi tepi ini, citra *barcode* dari proses *enhancement* diolah dengan deteksi tepi sobel sehingga tepi citra *barcode* tampak lebih jelas serta memberikan efek *smoothing* sehingga citra bersih dari *noise*. Selanjutnya akan diteruskan proses *thresholding* untuk mengubah nilai *pixel* menjadi 2 macam nilai yaitu 0 dan 1.

Hasil dari proses *pre processing* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Tahapan Proses *Pre Processing*

4.2.1.3 Feature Extraction (*Ekstraksi Ciri*)

Feature extraction merupakan salah satu cara untuk mengenali karakter dengan melihat ciri-ciri khusus yang dimiliki oleh setiap *barcode*. Diketahui bahwa *barcode* merupakan kumpulan kode yang berbentuk garis, di mana masing-masing ketebalan setiap garis berbeda sesuai dengan isi kode. Oleh karena itu, setelah melakukan proses deteksi tepi langkah selanjutnya adalah *feature extraction* (*ekstraksi ciri*).

Proses ekstraksi ciri ini, citra masukan dikodekan menurut *pixel*-nya, jika berwarna hitam atau kurang dari nilai *threshold* maka citra tersebut dikodekan 0, sebaliknya jika berwarna putih dan lebih dari nilai *threshold* maka dikodekan 1. Tujuan dari proses ini yaitu untuk memberi kode yang berbeda pada setiap karakter, sehingga karakter yang satu dengan karakter yang lain dapat dipisahkan berdasarkan kode yang dimilikinya. Disamping itu, proses pengkodean karakter ini dimaksudkan untuk mengambil ciri (*feature*) dari sebuah *barcode*.

Menurut Prasetyo dalam buku Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab menyebutkan satu cara yang jelas untuk mengekstrak obyek dari *background* adalah dengan memilih *threshold* T yang membagi mode-mode ini. Sembarang titik (x, y) untuk dimana $f(x, y) \geq T$ disebut *object point*, sedangkan yang lain disebut *background point*. Citra yang di-*threshold* $g(x, y)$ didefinisikan sebagai:

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

Pixel yang diberi nilai 1 berkaitan dengan obyek sedangkan *pixel* yang diberi nilai 0 berkaitan dengan *background*.

Proses ini citra *barcode* yang dimasukkan ke dalam jaringan akan diproses lebih lanjut, yaitu diambil nilai per pikselnya yang selanjutnya akan diubah ke bentuk numerik 0 dan 1 atau disebut juga dengan proses normalisasi. Tidak seluruh *pixel* citra *barcode* diambil nilainya, tetapi hanya nilai *pixel* citra secara horizontal. Letak pengambilan secara horizontal ini bisa dilakukan pada bagian atas, tengah atau bawah. Pengambilan ini tidak dilakukan secara keseluruhan sebab ciri khusus dari *barcode* adalah sama untuk setiap baris. Nilai-nilai *pixel* dari proses *grayscale*, *edge detection*, dan *thresholding* yang masih berukuran besar tidak diambil secara keseluruhan tetapi diambil hanya nilai *pixel* pada bagian tengah-tengah citra secara horizontal. Citra *input* berukuran 60×100 *pixel* dikonversi menjadi 20×30 *pixel* supaya pada saat *training* lebih efisien waktu dan tidak berat. Data citra yang dilatihkan berbentuk vektor dengan ukuran 1×30 kemudian disimpan dalam *database* dengan nama *file code_grid.mat*. Seperti pada Gambar 4.4.

The screenshot shows the MATLAB Variable Editor window with two variables displayed:

- code_grid <1x30 logical>**: A matrix with 30 columns and 1 row. The first column has value 1, the second column has value 0, and the third column has value 1. All other columns have value 0.
- code_grid <1x30 logical>**: A matrix with 30 columns and 1 row. The values are: 19, 0, 1, 0, 21, 0, 22, 0, 23, 0, 24, 0, 25, 0, 26, 0, 27, 1, 28, 0, 29, 0, 30, 0, 31, 0, 32, 0, 33, 0, 34, 0, 35, 0, 36.

Gambar 4.4 Gambar Vektor Hasil Ekstraksi Ciri

Vektor ini diperoleh dari matriks data citra yang kemudian diubah kedalam bentuk vektor. Karena jumlah citra latih sebanyak 100, maka matriks yang terbentuk

dalam *database* sistem adalah matriks berukuran 100×30 yaitu gabungan dari vektor-vektor citra latih sistem yang disimpan dalam satu *database* yang dinamakan *code_grid_all_pro.mat*.

4.2.1.4 Arsitektur Jaringan Syaraf Tiruan

Arsitektur jaringan yang dirancang dalam penelitian ini berjumlah 5 variasi yang dapat dilihat pada Tabel 4.2 yang sesuai dengan variasi arsitektur jaringan yang akan digunakan dalam pelatihan terhadap sistem. Variasi yang dilakukan adalah variasi jumlah *neuron hidden layer* serta nilai α (*learning rate*) untuk mencari hasil pengenalan yang optimal oleh sistem. Variasi yang dilakukan terhadap *learning rate* dan jumlah *neuron hidden layer* dapat dilihat di Tabel 4.2.

Tabel 4.2 Variasi *Learning Rate* dan Jumlah *Neuron Hidden Layer*

| No. | <i>Learning Rate</i> (α) | Jumlah <i>Neuron Hidden Layer</i> |
|-----|-----------------------------------|-----------------------------------|
| 1 | 0,01 | 100 |
| 2 | 0,05 | 100 |
| 3 | 0,1 | 50 |
| 4 | 0,5 | 20 |
| 5 | 1 | 10 |

Variasi pada Tabel 4.2 ini seperti yang dilakukan oleh penelitian Cahyono (2006), dimana variasi *learning rate* dan iterasi dapat dilihat pada Tabel 4.3.

Tabel 4.3 Pengaruh *Learning Rate* terhadap Akurasi

| No. | <i>Learning Rate</i> | Iterasi |
|-----|----------------------|---------|
| 1 | 0,05 | 100 |
| 2 | 0,1 | 100 |
| 3 | 0,5 | 100 |
| 4 | 1 | 100 |

Selain itu peneliti juga melakukan beberapa eksperimen sebelum menentukan variasi *learning rate* dan jumlah *neuron hidden layer*. Eksperimen yang dilakukan

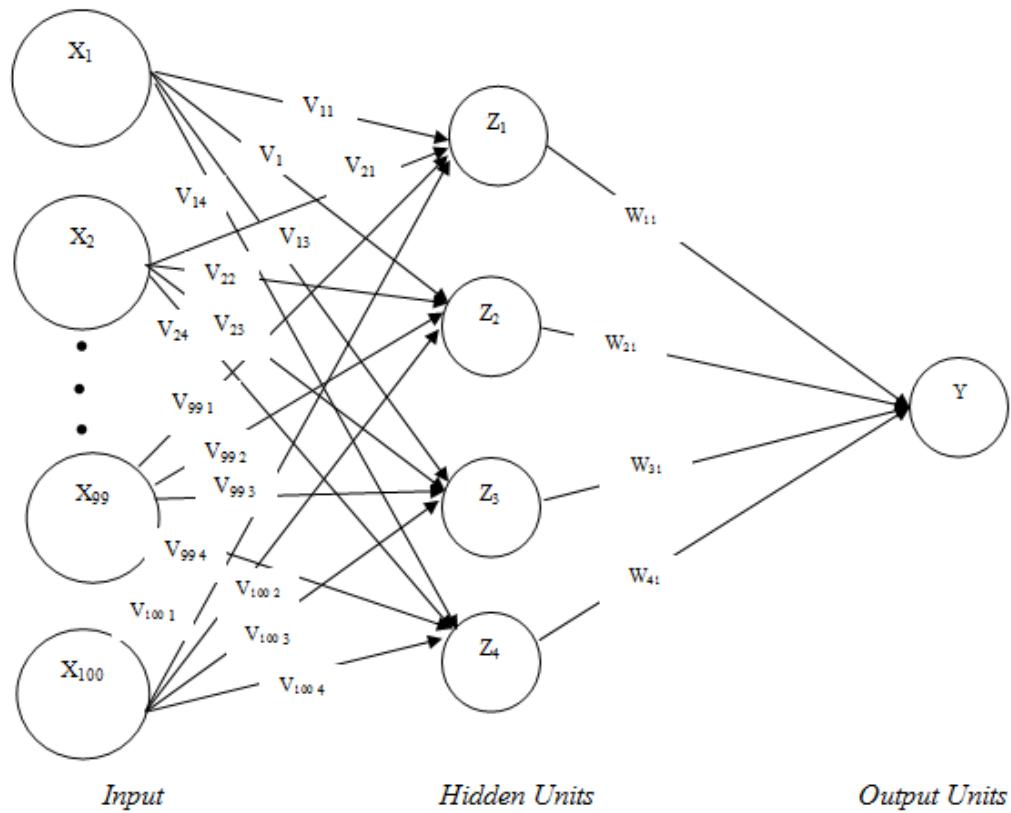
yaitu dengan variasi *learning rate* 0,02, dan 0,2 dan jumlah *neuron hidden layer* 100, dan 30. Hasil pengenalan dari eksperimen ini mirip dengan variasi *learning rate* 0,01, dan 0,1 dan jumlah *neuron hidden layer* 100, dan 50.

Paradigma pembelajaran pada JST yang dirancang adalah *supervised learning* karena dalam sistem ini terdapat sejumlah pasangan data (masukan – target luaran) yang dipakai untuk melatih jaringan. Pada arsitektur jaringan *backpropagation* fungsi aktivasi yang digunakan untuk mengubah bobot awal dari *input layer* ke *hidden layer* maupun dari *hidden layer* ke *output layer* adalah fungsi sigmoid biner. Fungsi sigmoid biner memiliki *range* 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutukan nilai *output* yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan syaraf yang nilai *output*-nya 0 atau 1.

Pada pembobotan awal, dipilih bobot secara acak, yaitu dengan cara membangkitkan bilangan acak dengan *range* antara 0 sampai 1. Arsitektur jaringan yang digunakan dalam skripsi ini adalah.

4.2.1.4.1 Arsitektur Jaringan Layar Jamak dengan 1 *Hidden Layer*

Arsitektur jaringan layar jamak dengan 1 *hidden layer* yang dirancang dalam penelitian ini dapat dilihat pada Gambar 4.5.

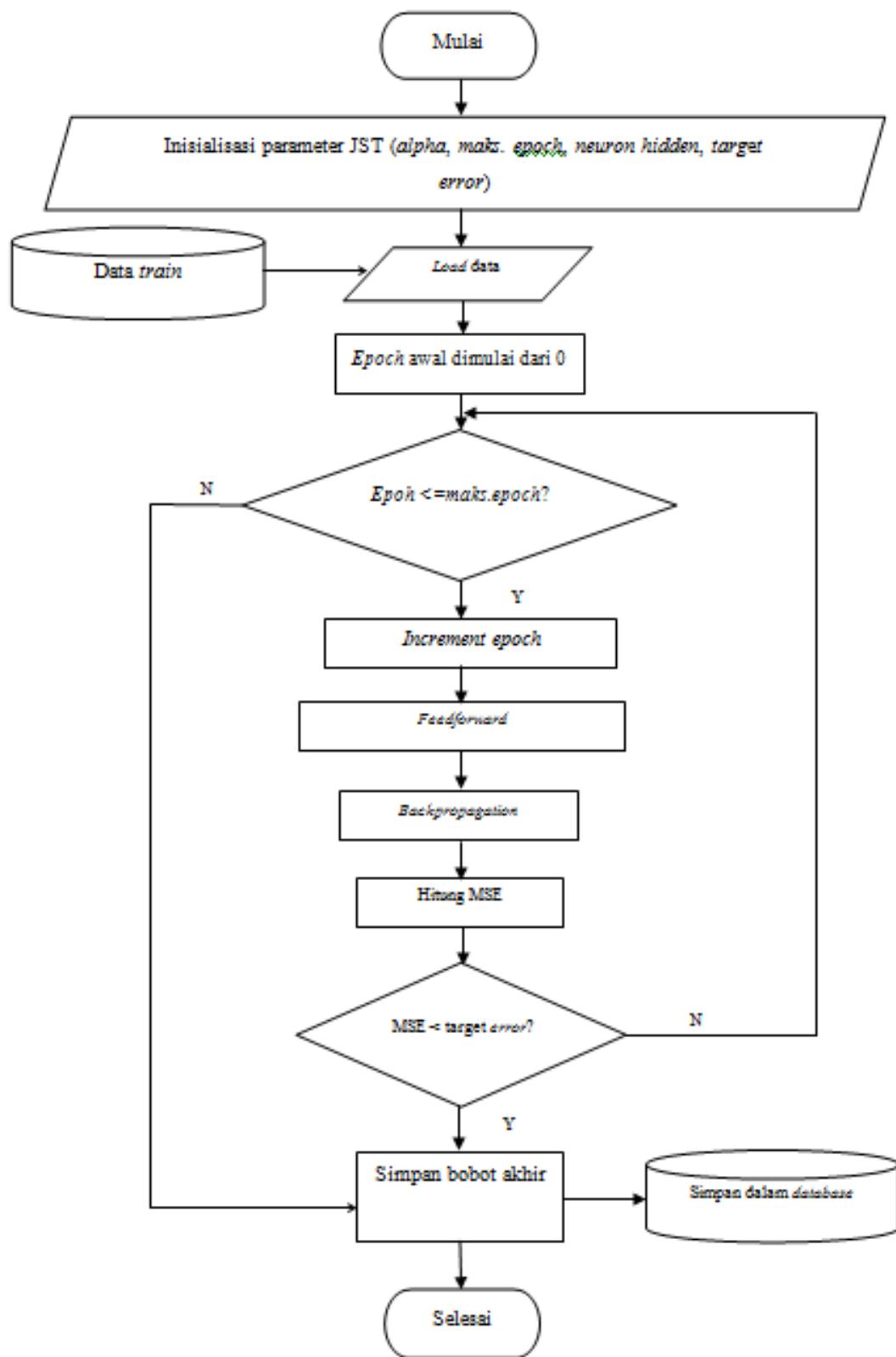


Gambar 4.5 Arsitektur Jaringan dengan 4 Unit *Neuron Hidden Lay*

Keterangan:

- | | |
|----------------------|-------------------------------------|
| X | : nilai hasil proses ekstraksi ciri |
| Z_1, Z_2, Z_3, Z_4 | : <i>neuron hidden layer</i> |
| Y | : nilai target luaran |

Pada Gambar 4.5 diperlihatkan jaringan dengan 100 buah unit masukan X_1, X_2, \dots, X_{100} hasil dari proses ekstraksi ciri yaitu untuk satu citra dikonversi menjadi vektor berukuran 1×30 , sebuah *hidden layer* yang terdiri dari 4 buah unit (Z_1, Z_2, Z_3, Z_4) dan sebuah unit keluaran (Y). *Flowchart* untuk langkah pelatihan ini dapat dilihat pada Gambar 4.6.

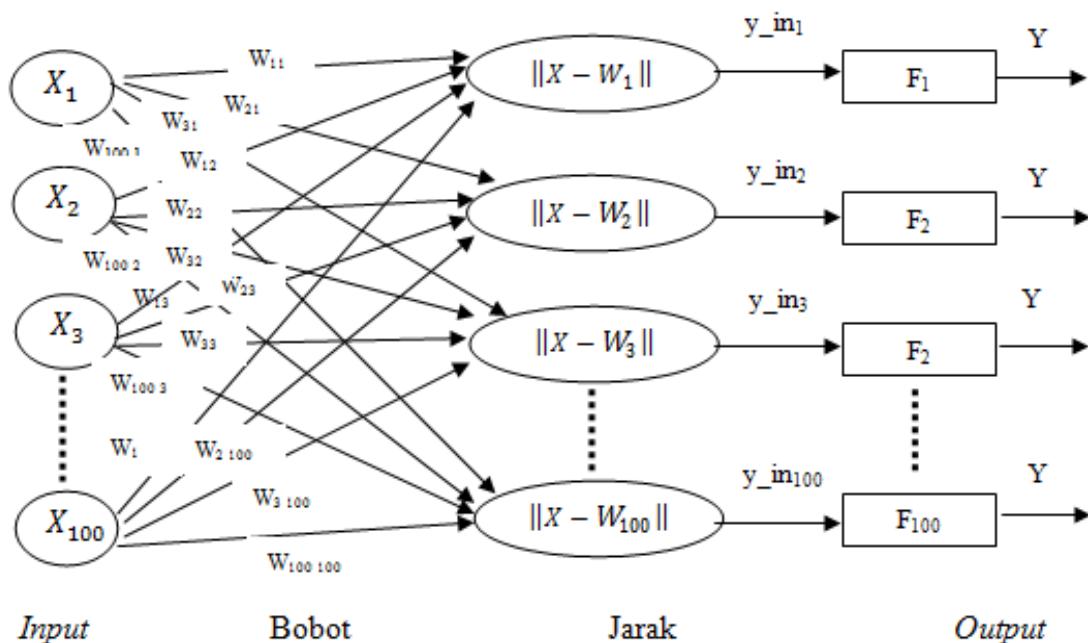


Gambar 4.6 Flowchart Proses Pelatihan Backpropagation

4.2.1.4.2 Arsitektur Jaringan LVQ

Arsitektur jaringan LVQ terdiri dari lapisan *input* (*input layer*), lapisan kompetitif (terjadi kompetisi pada *input* untuk masuk ke dalam suatu kelas berdasarkan kedekatan jaraknya) dan lapisan *output* (*output layer*). Lapisan *input* dihubungkan dengan lapisan kompetitif oleh bobot. Proses pembelajaran dilakukan secara terawasi dalam lapisan kompetitif. *Input* akan bersaing untuk dapat masuk ke dalam suatu kelas. Hasil dari lapisan kompetitif ini berupa kelas, yang kemudian akan dihubungkan dengan lapisan *output* oleh fungsi aktivasi.

Ada 100 variabel dari vektor *input* dalam penelitian ini, yaitu $X = (X_1, X_2, X_3, \dots, X_{100})$ dengan *neuron* keluaran Y_1, Y_2, \dots, Y_{100} serta 100 vektor bobot yaitu $W_{ij} = (W_{1,1}, W_{2,1}, W_{3,1}, \dots, W_{100,100})$, maka arsitektur jaringan LVQ dapat dilihat pada Gambar 4.7.

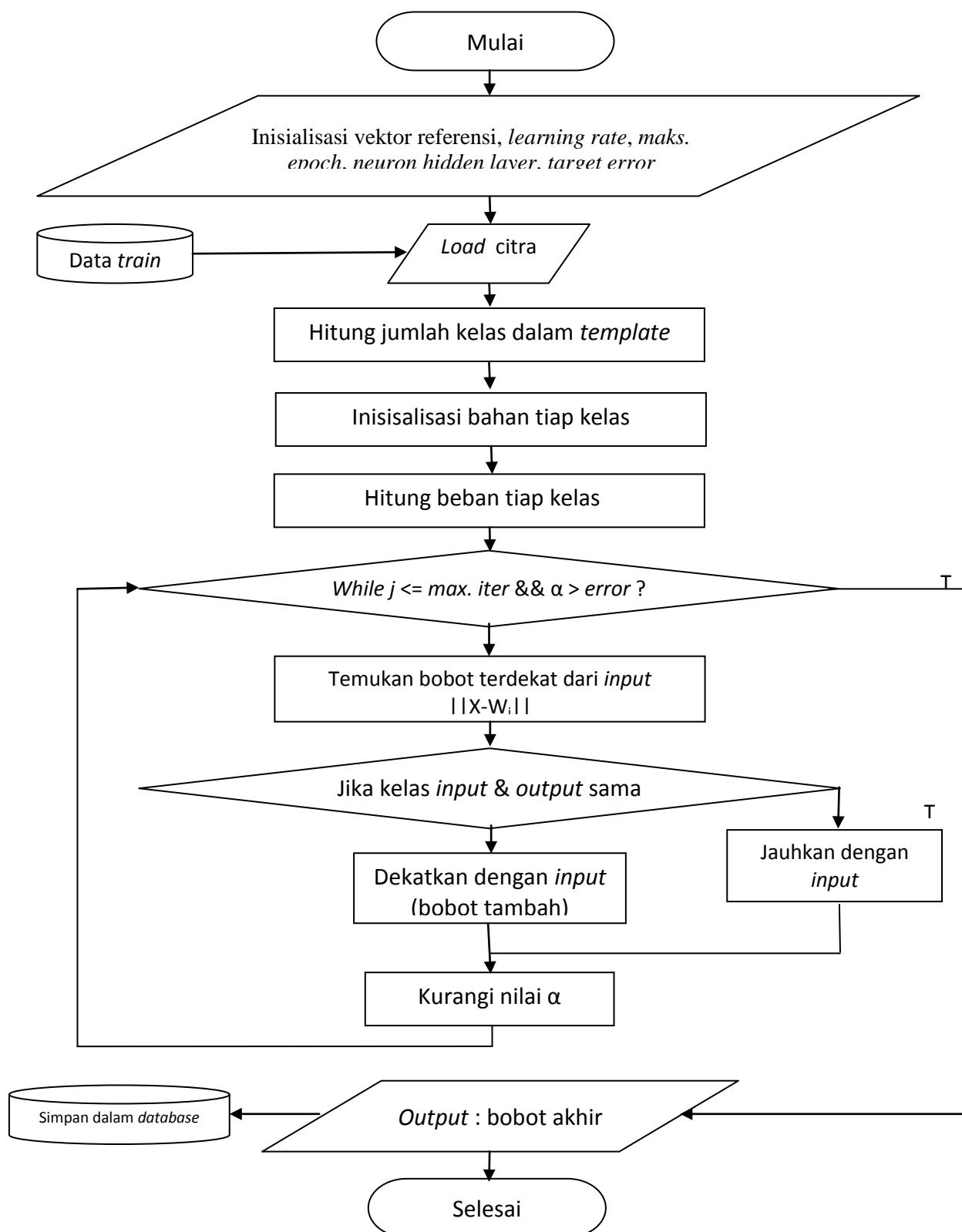


Gambar 4.7 Arsitektur Jaringan LVQ dengan 100 Vektor *Input* dan 100 Vektor

Bobot

Berdasarkan Gambar 4.7, tampak bahwa dalam LVQ terdapat 100 vektor bobot yang menghubungkan setiap *neuron* masukan dengan *neuron* keluaran sehingga dapat dikatakan bahwa setiap *neuron* keluaran pada LVQ berhubungan dengan sebuah vektor bobot. Pada proses pengenalan dan pembelajaran, LVQ menggunakan operasi-operasi vektor. Pola-pola akan disajikan dalam bentuk vektor. Proses yang terjadi pada setiap *neuron* adalah mencari jarak antara suatu vektor *input* ke bobot yang bersangkutan. Fungsi aktivasi (F) yang digunakan pada arsitektur jaringan LVQ adalah fungsi linier, tujuannya adalah agar diperoleh keluaran yang sama dengan masukan, sesuai dengan rumus fungsi linier $y = x$. Fungsi aktifasi F_1 akan memetakan y_{in1} ke $y_1 = 1$ apabila $|X - W_1| < |X - W_2|$, dan $y_1 = 0$ jika sebaliknya. Demikian pula fungsi aktifasi F_2 , akan memetakan y_{in2} ke $y_2 = 1$ apabila $|X - W_2| < |X - W_1|$, dan $y_2 = 0$ jika sebaliknya.

Metode LVQ akan melakukan pengenalan terlebih dahulu terhadap pola masukan yang harus disajikan dalam bentuk vektor agar dapat dicari kelasnya. Setiap *neuron* keluaran menyatakan kelas atau kategori tertentu, maka pola masukan dapat dikenali kelasnya berdasarkan *neuron* keluaran yang diperoleh. Metode LVQ mengenali pola masukan berdasarkan pada kedekatan jarak antara dua vektor yaitu vektor dari unit/*neuron* masukan dengan vektor bobot. *Flowchart* untuk langkah pelatihan LVQ ini dapat dilihat pada Gambar 4.8.



Gambar 4.8 Flowchart Proses Pelatihan LVQ

4.2.2 Perancangan Sistem Pengenalan

Hal yang perlu diperhatikan sebelum merancang sistem adalah spesifikasi *hardware* yang akan digunakan dalam membangun sistem. Berikut ini merupakan spesifikasi *hardware* yang digunakan dalam pembuatan sistem:

1. *processor* : Intel® Core™ i3-380M;
2. *memory* : 2 GB;
3. *hardisk* : 500 MB;
4. *monitor* : 14 inch;
5. *mouse dan keyboard*.

Jikalau data yang diolah dalam jumlah sangat besar, memori pada *hardware* terlalu kecil atau tidak mencukupi akan memicu sistem terjadi *error* atau *out of memory* saat dilakukan *training*, karena *software* Matlab R2009a membutuhkan memori yang cukup besar.

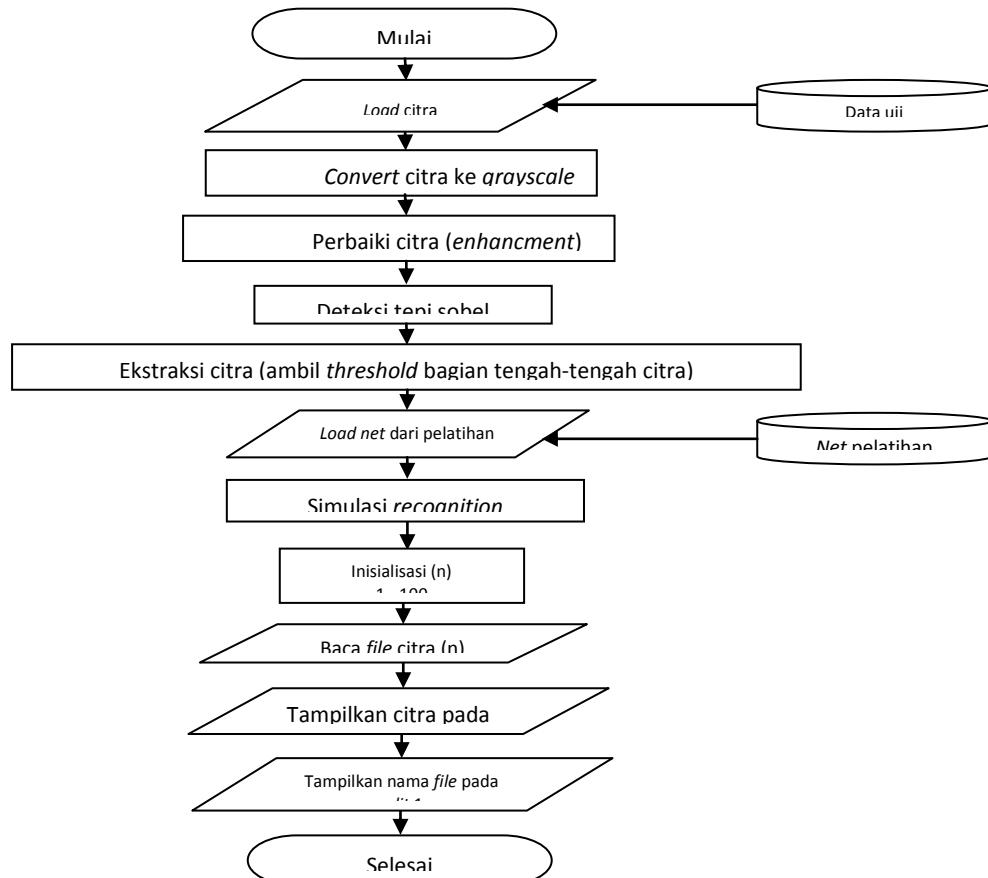
Langkah-langkah yang harus ditempuh pada perancangan sistem pengenalan sama halnya pada perancangan sistem pelatihan, yaitu proses *pre processing* dilanjutkan dengan proses ekstraksi ciri untuk mendapatkan *feature* dari citra. Setelah *feature* diperoleh, langkah selanjutnya adalah melakukan proses pelatihan dilanjutkan proses pengenalan citra. Pada langkah ini akan dilaksanakan 2 perancangan sistem pengenalan yaitu.

4.2.2.1 Perancangan Sistem Pengenalan Backpropagation

Setelah melakukan proses pelatihan *backpropagation* jaringan syaraf tiruan dan telah menghasilkan nilai keluaran yang diinginkan, maka sistem siap untuk melaksanakan proses pengenalan terhadap data-data *barcode* yang sudah di-

training maupun *barcode* baru yang bukan termasuk ke dalam data yang sudah di-*training* atau digolongkan dalam data uji. Data-data yang akan dikenali adalah data *training* dan data uji. Data uji ini merupakan data *barcode* baru yang masih dari kelas yang sama, sehingga nanti akan diketahui data *barcode* baru tersebut masuk ke dalam kelas yang sudah di-*training*.

Proses pengenalan dengan *backpropagation* ini yang digunakan adalah nilai bobot keluaran dari proses *training* yang akan dicocokkan dengan nilai bobot keluaran *barcode* yang baru. Jadi data *barcode* baru yang diujikan juga mengalami proses pencarian nilai bobot keluaran dari data pengujian. *Flowchart* untuk sistem pengenalan citra *backpropagation* dapat dilihat pada Gambar 4.9.

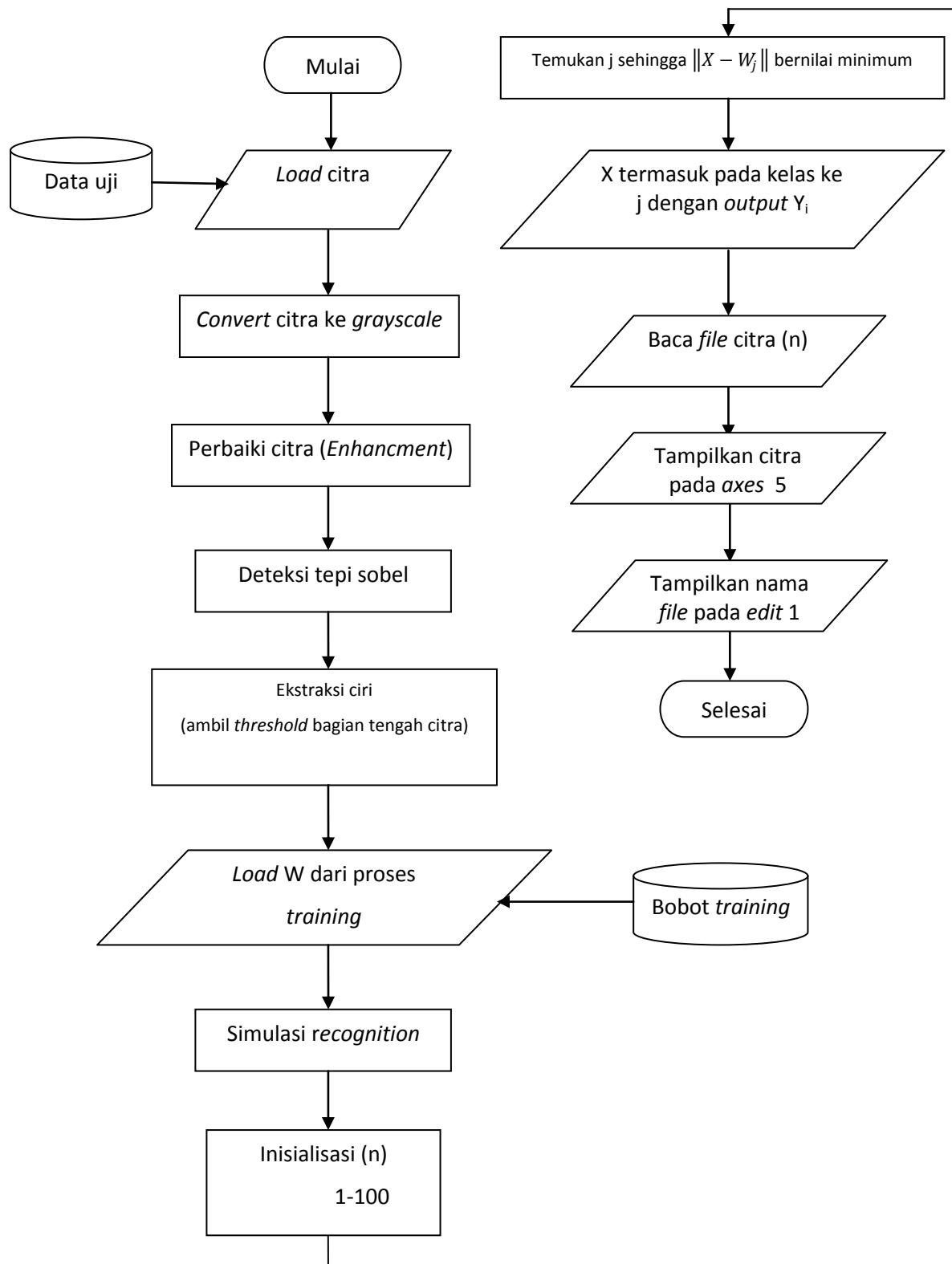


Gambar 4.9 *Flowchart Sistem Pengenalan Citra Backpropagation*

4.2.2.2 Perancangan Sistem Pengenalan Learning Vector Quantization

Sistem pengenalan *learning vector quantization* ini dibuat pada dasarnya sama dengan sistem pengenalan *backpropagation* yaitu melaksanakan proses pengenalan terhadap data-data *barcode* yang sudah di-*training* (dilatih) maupun *barcode* baru yang belum melalui proses *training* (pelatihan) dalam penelitian ini disebut data uji. Data-data yang akan dikenali adalah data *training* dan data uji. Data uji diambil dari data *barcode* baru yang masih dari kelas yang sama, sehingga akan diketahui data *barcode* uji digolongkan ke dalam kelas yang sudah di-*training*.

Sistem pengenalan LVQ ini, bobot akhir dari proses pembelajaran digunakan sebagai penentu hasil karena dalam algoritma LVQ jarak terdekat dari *input* data dengan bobot akhir adalah hasil dari pengenalan. Hasil pengenalan pada LVQ berupa nilai kelas yang dikenali pada saat pengujian. Adapun penjelasan proses pengenalan LVQ dalam penelitian ini akan dijelaskan melalui *flowchart* sistem pengenalan LVQ pada Gambar 4.10.



Gambar 4.10 Flowchart Sistem Pengenalan LVQ

4.3 Pengukuran Evaluasi Unjuk Kerja Sistem

Setelah sistem selesai dirancang, selanjutnya harus diuji tingkat akurasi sistem dalam mengenali citra *barcode*. Hal ini dilakukan untuk mengetahui sejauh mana sistem dapat bekerja dalam mengenali citra *barcode*.

4.3.1 Hasil Arsitektur Jaringan Syaraf Tiruan Backpropagation

Akan dilihat pengaruh perubahan *learning rate* dan jumlah *neuron hidden layer* terhadap MSE, lama waktu pelatihan, kemudian jumlah citra yang dikenali oleh sistem dengan benar dan tepat. Jumlah citra masukan adalah 100 citra latih dan 100 citra uji, diperoleh analisis pada Tabel 4.4.

Tabel 4.4 Analisis Hasil Variasi Arsitektur JST *Backpropagation*

| No. | α | Jumlah <i>neuron hidden layer</i> | <i>Epoch</i> | Hasil | | | | |
|-----|----------|--|--------------|------------------------|------------------|---------------------------------|-------------------------------|------------------------------------|
| | | | | MSE | Waktu (detik) | Citra latih yang dikenali | Citra uji yang dikenali | <i>Recog- nition rate%</i> |
| 1 | 0,01 | 100 | 1000 | $1,65 \times 10^{-28}$ | 42 | 49 | 100 | 74,5% |
| 2 | 0,05 | 100 | 1000 | $6,45 \times 10^{-28}$ | 102 | 100 | 51 | 75,5% |
| 3 | 0,1 | 50 | 100 | $3,14 \times 10^{-28}$ | 29 | 100 | 50 | 75% |
| 4 | 0,5 | 20 | 50 | 11,4 | 6 | 70 | 35 | 52,5% |
| 5 | 1 | 10 | 20 | 616 | 2 | 6 | 2 | 4% |

Berdasarkan Tabel 4.4, didapatkan bahwa arsitektur jaringan yang optimal dalam proses pengenalan citra adalah sebagai berikut:

learning rate : 0,05;

Jumlah *neuron hidden layer* : 100;

Jumlah iterasi (*epoch*) : 1000.

Diperoleh *recognition rate* sebesar 75,5 % dengan MSE sebesar $6,45 \times 10^{-28}$.

4.3.2 Hasil Arsitektur Jaringan Syaraf Tiruan LVQ

Pembahasan hasil arsitektur JST LVQ ini juga akan dilakukan variasi *learning rate* dan jumlah *neuron hidden layer*, sehingga terlihat pengaruh terhadap MSE, lama waktu pelatihan, kemudian jumlah citra yang dikenali oleh sistem dengan benar dan tepat. Jumlah citra masukan adalah 100 citra latih dan 100 citra uji. Analisis hasil variasi dapat dilihat pada Tabel 4.5.

Tabel 4.5 Analisis Hasil Variasi Arsitektur Jaringan Syaraf Tiruan LVQ

| No. | α | Jumlah <i>neuron hidden layer</i> | <i>Epoch</i> | Hasil | | | | <i>Recog- nition rate%</i> |
|-----|----------|--|--------------|-------|------------------|---------------------------------|-------------------------------|------------------------------------|
| | | | | MSE | Waktu (detik) | Citra latih yang dikenali | Citra uji yang dikenali | |
| 1 | 0,01 | 100 | 1000 | 0 | 38 | 100 | 88 | 94% |
| 2 | 0,05 | 100 | 1000 | 0 | 72 | 100 | 88 | 94% |
| 3 | 0,1 | 50 | 100 | 0 | 1 | 100 | 88 | 94% |
| 4 | 0,5 | 20 | 50 | 0 | 0.9 | 100 | 88 | 94% |
| 5 | 1 | 10 | 20 | 0 | 0.3 | 100 | 88 | 94% |

Dari Tabel 4.5, didapatkan bahwa arsitektur jaringan yang optimal dalam proses pengenalan citra adalah sebagai berikut:

learning rate : 1;

Jumlah *neuron hidden layer* : 10;

Jumlah iterasi (*epoch*) : 20.

Diperoleh *recognition rate* sebesar 94 % dengan MSE sebesar 0.

Arsitektur ini dipilih karena memiliki waktu pelatihan yang paling minimum, sehingga akan lebih efisien apabila digunakan dalam proses pengenalan citra *barcode*.

4.3.3 Hasil Pengenalan Sistem

Citra yang diuji dalam sistem merupakan citra yang telah *di-training* dan citra yang belum *di-training*. Disini akan dilakukan 4 variasi pengujian. Berikut hasil dari pengenalan oleh sistem berdasarkan variasi yang dilakukan.

Variasi 1

Tabel 4.6 Hasil Pengenalan Variasi 1

| <i>Learning Rate</i> | Jumlah <i>Neuron Hidden Layer</i> | <i>Epoch</i> | MSE | | <i>Time</i> (Detik) | |
|----------------------|--------------------------------------|--------------|------------------------|-----------|------------------------|-----------|
| | | | <i>Backpropagation</i> | LVQ | <i>Backpropagation</i> | LVQ |
| 0,01 | 100 | 1000 | $1,65 \times 10^{-28}$ | 0 | 42 | 38 |
| Hasil Pengenalan | | | | | | |
| No. | Nama Produk | | <i>Backpropagation</i> | | LVQ | |
| | | | Citra Train | Citra Uji | Citra Train | Citra Uji |
| 1. | ABC Kacang Hijau | | ✓ | ✗ | ✓ | ✓ |
| 2. | Adem Sari Fres | | ✓ | ✗ | ✓ | ✓ |
| 3. | Adidas Ice Dive | | ✓ | ✓ | ✓ | ✓ |
| 4. | Anada Susu Bayi | | ✓ | ✓ | ✓ | ✓ |
| 5. | Anlene Gold | | ✓ | ✗ | ✓ | ✓ |
| 6. | Anlene | | ✓ | ✓ | ✓ | ✓ |
| 7. | Annum | | ✓ | ✗ | ✓ | ✓ |
| 8. | Astor | | ✓ | ✗ | ✓ | ✓ |
| 9. | Baygon Mat Elektrik | | ✓ | ✗ | ✓ | ✓ |
| 10. | Bellagio Home | | ✓ | ✗ | ✓ | ✓ |
| 11. | Casablanca | | ✓ | ✓ | ✓ | ✓ |
| 12. | Chocolatos | | ✓ | ✓ | ✓ | ✓ |
| 13. | Citra Bubuk Mutiara Cina | | ✓ | ✓ | ✓ | ✓ |
| 14. | Clear Man | | ✓ | ✓ | ✓ | ✓ |
| 15. | Close Up | | ✓ | ✓ | ✓ | ✓ |
| 16. | Cocoa | | ✓ | ✗ | ✓ | ✓ |
| 17. | Copiko Brown Coffe | | ✓ | ✗ | ✓ | ✓ |
| 18. | Dairy Milk | | ✓ | ✗ | ✓ | ✓ |
| 19. | Dancow Actigo | | ✓ | ✗ | ✓ | ✓ |
| 20. | Diabetasol Susu Formula 180 g | | ✓ | ✓ | ✓ | ✓ |
| 21. | Diabetasol Susu Formula Coklat | | ✓ | ✓ | ✓ | ✓ |

| | | | | | |
|----|-----------------------------|---|---|---|---|
| 22 | Dove White Beauty Bar | √ | × | √ | × |
| 23 | Emeron | √ | × | √ | √ |
| 24 | Energen Sereal | √ | × | √ | √ |
| 25 | Enzim | √ | √ | √ | √ |
| 26 | Fox | √ | √ | √ | √ |
| 27 | Frisian Baby 1 | √ | √ | √ | √ |
| 28 | Frisian Baby 2 | √ | √ | √ | √ |
| 29 | Frisian Flag Coklat 800 g | √ | × | √ | √ |
| 30 | Funtime | √ | × | √ | √ |
| 31 | Garnier Color Naturals | √ | √ | √ | √ |
| 32 | Garnier Extra UV Protection | √ | × | √ | × |
| 33 | Garnier Face Pouder | √ | × | √ | √ |
| 34 | Garnier Night Restore | √ | √ | √ | √ |
| 35 | Gatsby Cool | √ | √ | √ | √ |
| 36 | Gatsby hair Shampoo | √ | √ | √ | √ |
| 37 | Glade 400 ml | √ | √ | √ | √ |
| 38 | Glade Automatic Spray | √ | × | √ | √ |
| 39 | Goodday Mocafrio | √ | × | √ | √ |
| 40 | Goodtime | √ | × | √ | × |
| 41 | Happy Tos | √ | × | √ | √ |
| 42 | Head&Shoulders | √ | √ | √ | √ |
| 43 | Hello Panda | √ | √ | √ | √ |
| 44 | Hit Non_stop | √ | √ | √ | √ |
| 45 | Hydro Coco | √ | √ | √ | √ |
| 46 | Indomie | √ | × | √ | × |
| 47 | Jess Cool | √ | × | √ | √ |
| 48 | Koko Krunch | √ | √ | √ | √ |
| 49 | Kopiko 78°C | √ | √ | √ | √ |
| 50 | La Fonte | √ | × | √ | √ |
| 51 | Lactamil | √ | × | √ | √ |
| 52 | Lactogen 2 | √ | × | √ | √ |
| 53 | Larutan Penyegar Cap Kaki 3 | √ | × | √ | √ |
| 54 | Lifebuoy Clear Skin | √ | √ | √ | √ |
| 55 | Lux Magic Spell | √ | √ | √ | √ |
| 56 | Max Tea | √ | × | √ | √ |
| 57 | Maxi Class | √ | × | √ | √ |
| 58 | Milkuat | √ | √ | √ | √ |
| 59 | Milna Biskuit Bayi | √ | √ | √ | √ |
| 60 | Milna Bubur Bayi | √ | × | √ | × |
| 61 | Mr Muscle | √ | √ | √ | √ |

| | | | | | |
|--------------------------|------------------------|-----|----|-----|----|
| 62 | Nescafe | √ | √ | √ | √ |
| 63 | Nestle Cerelac | √ | √ | √ | √ |
| 64 | Nestle Coffe Mat | √ | ✗ | √ | ✗ |
| 65 | Nestle Milo Sereal | √ | √ | √ | √ |
| 66 | OBH | √ | ✗ | √ | √ |
| 67 | Olay Total Effect | √ | ✗ | √ | √ |
| 68 | Oreo | √ | ✗ | √ | √ |
| 69 | Ovaltine | √ | √ | √ | √ |
| 70 | Pentene Conditioner | √ | ✗ | √ | √ |
| 71 | Pepsodent Action | √ | √ | √ | √ |
| 72 | Pepsodent Center Fresh | √ | ✗ | √ | ✗ |
| 73 | Pepsodent Herbal | √ | ✗ | √ | √ |
| 74 | Ponds White Beauty | √ | ✗ | √ | ✗ |
| 75 | Rapika | √ | ✗ | √ | ✗ |
| 76 | Richeese | √ | ✗ | √ | √ |
| 77 | Rocky Snack | √ | ✗ | √ | √ |
| 78 | Scott Imulsion | √ | ✗ | √ | √ |
| 79 | Sensodyne | √ | ✗ | √ | √ |
| 80 | Serena | √ | ✗ | √ | √ |
| 81 | SGM 1 | √ | √ | √ | √ |
| 82 | SGM 2 | √ | √ | √ | √ |
| 83 | Shinzui | √ | √ | √ | √ |
| 84 | Snack Big Sheet | √ | ✗ | √ | √ |
| 85 | Soffel Kulit Jeruk | √ | ✗ | √ | √ |
| 86 | Springles | √ | ✗ | √ | ✗ |
| 87 | Stella Car Perfume | √ | √ | √ | √ |
| 88 | Susu Bendera 1000 ml | √ | √ | √ | √ |
| 89 | Tango Cruch Cake | √ | √ | √ | √ |
| 90 | Tango Fushion | √ | √ | √ | √ |
| 91 | Teh Botol Sosro | √ | √ | √ | √ |
| 92 | Teh Celup Dandang | √ | √ | √ | √ |
| 93 | Teh Celup Tong Tji | √ | ✗ | √ | ✗ |
| 94 | Torabika Cappucino | √ | √ | √ | √ |
| 95 | Tresemme | √ | √ | √ | √ |
| 96 | Twister | √ | √ | √ | √ |
| 97 | Vape Liquid | √ | ✗ | √ | ✗ |
| 98 | Wafer Selamat | √ | √ | √ | √ |
| 99 | WRP Nutritious Drink | √ | √ | √ | √ |
| 100 | Zact | √ | ✗ | √ | √ |
| Jumlah berhasil dikenali | | 100 | 49 | 100 | 88 |

Kesimpulan :

| Metode | Citra Train | | Citra Uji | |
|-----------------|-----------------|-----------------------|-----------------|-----------------------|
| | Jumlah Berhasil | Jumlah Tidak Berhasil | Jumlah Berhasil | Jumlah Tidak Berhasil |
| Backpropagation | 100 | 0 | 49 | 51 |
| LVQ | 100 | 0 | 88 | 12 |

Variasi 2

Tabel 4.7 Hasil Pengenalan Variasi 2

| Learning Rate | Jumlah Neuron Hidden Layer | Epoch | MSE | | Time (Detik) | |
|---------------|----------------------------|-------|------------------------|-----|-----------------|-----|
| | | | Backpropagation | LVQ | Backpropagation | LVQ |
| 0,05 | 100 | 1000 | $6,45 \times 10^{-26}$ | 0 | 102 | 72 |

| No. | Nama Produk | Hasil Pengenalan | | | |
|-----|--------------------------------|------------------|-----------|-------------|-----------|
| | | Backpropagation | | LVQ | |
| | | Citra Train | Citra Uji | Citra Train | Citra Uji |
| 1. | ABC Kacang Hijau | ✓ | ✗ | ✓ | ✓ |
| 2. | Adem Sari Fres | ✓ | ✗ | ✓ | ✓ |
| 3. | Adidas Ice Dive | ✓ | ✓ | ✓ | ✓ |
| 4. | Anada Susu Bayi | ✓ | ✓ | ✓ | ✓ |
| 5. | Anlene Gold | ✓ | ✗ | ✓ | ✓ |
| 6. | Anlene | ✓ | ✓ | ✓ | ✓ |
| 7. | Annum | ✓ | ✗ | ✓ | ✓ |
| 8. | Astor | ✓ | ✗ | ✓ | ✓ |
| 9. | Baygon Mat Elektrik | ✓ | ✗ | ✓ | ✓ |
| 10. | Bellagio Home | ✓ | ✗ | ✓ | ✓ |
| 11. | Casablanca | ✓ | ✓ | ✓ | ✓ |
| 12. | Chocolatos | ✓ | ✓ | ✓ | ✓ |
| 13. | Citra Bubuk Mutiara Cina | ✓ | ✓ | ✓ | ✓ |
| 14. | Clear Man | ✓ | ✓ | ✓ | ✓ |
| 15. | Close Up | ✓ | ✓ | ✓ | ✓ |
| 16. | Cocoa | ✓ | ✗ | ✓ | ✓ |
| 17. | Copiko Brown Coffe | ✓ | ✗ | ✓ | ✓ |
| 18. | Dairy Milk | ✓ | ✓ | ✓ | ✓ |
| 19. | Dancow Actigo | ✓ | ✓ | ✓ | ✓ |
| 20. | Diabetasol Susu Formula 180 g | ✓ | ✓ | ✓ | ✓ |
| 21. | Diabetasol Susu Formula Coklat | ✓ | ✓ | ✓ | ✓ |

| | | | | | |
|----|-----------------------------|---|---|---|---|
| 22 | Dove White Beauty Bar | √ | × | √ | × |
| 23 | Emeron | √ | × | √ | √ |
| 24 | Energen Sereal | √ | × | √ | √ |
| 25 | Enzim | √ | √ | √ | √ |
| 26 | Fox | √ | √ | √ | √ |
| 27 | Frisian Baby 1 | √ | √ | √ | √ |
| 28 | Frisian Baby 2 | √ | √ | √ | √ |
| 29 | Frisian Flag Coklat 800 g | √ | × | √ | √ |
| 30 | Funtime | √ | × | √ | √ |
| 31 | Garnier Color Naturals | √ | √ | √ | √ |
| 32 | Garnier Extra UV Protection | √ | × | √ | × |
| 33 | Garnier Face Pouder | √ | × | √ | √ |
| 34 | Garnier Night Restore | √ | √ | √ | √ |
| 35 | Gatsby Cool | √ | √ | √ | √ |
| 36 | Gatsby hair Shampoo | √ | √ | √ | √ |
| 37 | Glade 400 ml | √ | √ | √ | √ |
| 38 | Glade Automatic Spray | √ | × | √ | √ |
| 39 | Goodday Mocafrio | √ | × | √ | √ |
| 40 | Goodtime | √ | × | √ | × |
| 41 | Happy Tos | √ | × | √ | √ |
| 42 | Head&Shoulders | √ | √ | √ | √ |
| 43 | Hello Panda | √ | √ | √ | √ |
| 44 | Hit Non_stop | √ | √ | √ | √ |
| 45 | Hydro Coco | √ | √ | √ | √ |
| 46 | Indomie | √ | × | √ | × |
| 47 | Jess Cool | √ | × | √ | √ |
| 48 | Koko Krunch | √ | √ | √ | √ |
| 49 | Kopiko 78°C | √ | √ | √ | √ |
| 50 | La Fonte | √ | × | √ | √ |
| 51 | Lactamil | √ | × | √ | √ |
| 52 | Lactogen 2 | √ | × | √ | √ |
| 53 | Larutan Penyegar Cap Kaki 3 | √ | × | √ | √ |
| 54 | Lifebuoy Clear Skin | √ | √ | √ | √ |
| 55 | Lux Magic Spell | √ | √ | √ | √ |
| 56 | Max Tea | √ | × | √ | √ |
| 57 | Maxi Class | √ | × | √ | √ |
| 58 | Milkuat | √ | √ | √ | √ |
| 59 | Milna Biskuit Bayi | √ | √ | √ | √ |
| 60 | Milna Bubur Bayi | √ | × | √ | × |
| 61 | Mr Muscle | √ | √ | √ | √ |

| | | | | | |
|--------------------------|------------------------|-----|----|-----|----|
| 62 | Nescafe | √ | √ | √ | √ |
| 63 | Nestle Cerelac | √ | √ | √ | √ |
| 64 | Nestle Coffe Mat | √ | ✗ | √ | ✗ |
| 65 | Nestle Milo Sereal | √ | √ | √ | √ |
| 66 | OBH | √ | ✗ | √ | √ |
| 67 | Olay Total Effect | √ | ✗ | √ | √ |
| 68 | Oreo | √ | ✗ | √ | √ |
| 69 | Ovaltine | √ | √ | √ | √ |
| 70 | Pentene Conditioner | √ | ✗ | √ | √ |
| 71 | Pepsodent Action | √ | √ | √ | √ |
| 72 | Pepsodent Center Fresh | √ | ✗ | √ | ✗ |
| 73 | Pepsodent Herbal | √ | ✗ | √ | √ |
| 74 | Ponds White Beauty | √ | ✗ | √ | ✗ |
| 75 | Rapika | √ | ✗ | √ | ✗ |
| 76 | Richeese | √ | ✗ | √ | √ |
| 77 | Rocky Snack | √ | ✗ | √ | √ |
| 78 | Scott Imulsion | √ | ✗ | √ | √ |
| 79 | Sensodyne | √ | ✗ | √ | √ |
| 80 | Serena | √ | ✗ | √ | √ |
| 81 | SGM 1 | √ | √ | √ | √ |
| 82 | SGM 2 | √ | √ | √ | √ |
| 83 | Shinzui | √ | √ | √ | √ |
| 84 | Snack Big Sheet | √ | ✗ | √ | √ |
| 85 | Soffel Kulit Jeruk | √ | ✗ | √ | √ |
| 86 | Springles | √ | ✗ | √ | ✗ |
| 87 | Stella Car Perfume | √ | √ | √ | √ |
| 88 | Susu Bendera 1000 ml | √ | √ | √ | √ |
| 89 | Tango Cruch Cake | √ | √ | √ | √ |
| 90 | Tango Fushion | √ | √ | √ | √ |
| 91 | Teh Botol Sosro | √ | √ | √ | √ |
| 92 | Teh Celup Dandang | √ | √ | √ | √ |
| 93 | Teh Celup Tong Tji | √ | ✗ | √ | ✗ |
| 94 | Torabika Cappucino | √ | √ | √ | √ |
| 95 | Tresemme | √ | √ | √ | √ |
| 96 | Twister | √ | √ | √ | √ |
| 97 | Vape Liquid | √ | ✗ | √ | ✗ |
| 98 | Wafer Selamat | √ | √ | √ | √ |
| 99 | WRP Nutritious Drink | √ | √ | √ | √ |
| 100 | Zact | √ | ✗ | √ | √ |
| Jumlah berhasil dikenali | | 100 | 51 | 100 | 88 |

Kesimpulan :

| Metode | Citra Train | | Citra Uji | |
|-----------------|-----------------|-----------------------|-----------------|-----------------------|
| | Jumlah Berhasil | Jumlah Tidak Berhasil | Jumlah Berhasil | Jumlah Tidak Berhasil |
| Backpropagation | 100 | 0 | 51 | 49 |
| LVQ | 100 | 0 | 88 | 12 |

Variasi 3

Tabel 4.8 Hasil Pengenalan Variasi 3

| Learning Rate | Jumlah Neuron Hidden Layer | Epoch | MSE | | Time (Detik) | |
|---------------|----------------------------|-------|------------------------|-----|-----------------|-----|
| | | | Backpropagation | LVQ | Backpropagation | LVQ |
| 0,1 | 50 | 100 | $3,17 \times 10^{-28}$ | 0 | 29 | 1 |

| No. | Nama Produk | Hasil Pengenalan | | | |
|-----|--------------------------------|------------------|-----------|-------------|-----------|
| | | Backpropagation | | LVQ | |
| | | Citra Train | Citra Uji | Citra Train | Citra Uji |
| 1. | ABC Kacang Hijau | √ | ✗ | √ | √ |
| 2. | Adem Sari Fres | √ | ✗ | √ | √ |
| 3. | Adidas Ice Dive | √ | √ | √ | √ |
| 4. | Anada Susu Bayi | √ | √ | √ | √ |
| 5. | Anlene Gold | √ | ✗ | √ | √ |
| 6. | Anlene | √ | √ | √ | √ |
| 7. | Annum | √ | ✗ | √ | √ |
| 8. | Astor | √ | ✗ | √ | √ |
| 9. | Baygon Mat Elektrik | √ | ✗ | √ | √ |
| 10. | Bellagio Home | √ | √ | √ | √ |
| 11. | Casablanca | √ | √ | √ | √ |
| 12. | Chocolatos | √ | √ | √ | √ |
| 13. | Citra Bubuk Mutiara Cina | √ | √ | √ | √ |
| 14. | Clear Man | √ | √ | √ | √ |
| 15. | Close Up | √ | √ | √ | √ |
| 16. | Cocoa | √ | ✗ | √ | √ |
| 17. | Copiko Brown Coffe | √ | ✗ | √ | √ |
| 18. | Dairy Milk | √ | ✗ | √ | √ |
| 19. | Dancow Actigo | √ | √ | √ | √ |
| 20. | Diabetasol Susu Formula 180 g | √ | ✗ | √ | √ |
| 21. | Diabetasol Susu Formula Coklat | √ | √ | √ | √ |

| | | | | | |
|----|-----------------------------|---|---|---|---|
| 22 | Dove White Beauty Bar | √ | ✗ | √ | ✗ |
| 23 | Emeron | √ | ✗ | √ | √ |
| 24 | Energen Sereal | √ | ✗ | √ | √ |
| 25 | Enzim | √ | ✓ | ✓ | ✓ |
| 26 | Fox | √ | ✓ | ✓ | ✓ |
| 27 | Frisian Baby 1 | √ | ✓ | ✓ | ✓ |
| 28 | Frisian Baby 2 | √ | ✓ | ✓ | ✓ |
| 29 | Frisian Flag Coklat 800 g | √ | ✗ | ✓ | ✓ |
| 30 | Funtime | √ | ✗ | ✓ | ✓ |
| 31 | Garnier Color Naturals | √ | ✓ | ✓ | ✓ |
| 32 | Garnier Extra UV Protection | √ | ✗ | ✓ | ✗ |
| 33 | Garnier Face Pouder | √ | ✗ | ✓ | ✓ |
| 34 | Garnier Night Restore | √ | ✓ | ✓ | ✓ |
| 35 | Gatsby Cool | √ | ✓ | ✓ | ✓ |
| 36 | Gatsby hair Shampoo | √ | ✓ | ✓ | ✓ |
| 37 | Glade 400 ml | √ | ✓ | ✓ | ✓ |
| 38 | Glade Automatic Spray | √ | ✗ | ✓ | ✓ |
| 39 | Goodday Mocafrio | √ | ✗ | ✓ | ✓ |
| 40 | Goodtime | √ | ✗ | ✓ | ✗ |
| 41 | Happy Tos | √ | ✗ | ✓ | ✓ |
| 42 | Head&Shoulders | √ | ✓ | ✓ | ✓ |
| 43 | Hello Panda | √ | ✓ | ✓ | ✓ |
| 44 | Hit Non_stop | √ | ✓ | ✓ | ✓ |
| 45 | Hydro Coco | √ | ✓ | ✓ | ✓ |
| 46 | Indomie | √ | ✗ | ✓ | ✗ |
| 47 | Jess Cool | √ | ✗ | ✓ | ✓ |
| 48 | Koko Krunch | √ | ✓ | ✓ | ✓ |
| 49 | Kopiko 78°C | √ | ✓ | ✓ | ✓ |
| 50 | La Fonte | √ | ✗ | ✓ | ✓ |
| 51 | Lactamil | √ | ✗ | ✓ | ✓ |
| 52 | Lactogen 2 | √ | ✗ | ✓ | ✓ |
| 53 | Larutan Penyegar Cap Kaki 3 | √ | ✗ | ✓ | ✓ |
| 54 | Lifebuoy Clear Skin | √ | ✓ | ✓ | ✓ |
| 55 | Lux Magic Spell | √ | ✓ | ✓ | ✓ |
| 56 | Max Tea | √ | ✗ | ✓ | ✓ |
| 57 | Maxi Class | √ | ✗ | ✓ | ✓ |
| 58 | Milkuat | √ | ✓ | ✓ | ✓ |
| 59 | Milna Biskuit Bayi | √ | ✓ | ✓ | ✓ |
| 60 | Milna Bubur Bayi | √ | ✗ | ✓ | ✗ |
| 61 | Mr Muscle | √ | ✓ | ✓ | ✓ |

| | | | | | |
|--------------------------|------------------------|-----|----|-----|----|
| 62 | Nescafe | √ | √ | √ | √ |
| 63 | Nestle Cerelac | √ | √ | √ | √ |
| 64 | Nestle Coffe Mat | √ | ✗ | √ | ✗ |
| 65 | Nestle Milo Sereal | √ | √ | √ | √ |
| 66 | OBH | √ | ✗ | √ | √ |
| 67 | Olay Total Effect | √ | ✗ | √ | √ |
| 68 | Oreo | √ | ✗ | √ | √ |
| 69 | Ovaltine | √ | √ | √ | √ |
| 70 | Pentene Conditioner | √ | ✗ | √ | √ |
| 71 | Pepsodent Action | √ | √ | √ | √ |
| 72 | Pepsodent Center Fresh | √ | ✗ | √ | ✗ |
| 73 | Pepsodent Herbal | √ | ✗ | √ | √ |
| 74 | Ponds White Beauty | √ | ✗ | √ | ✗ |
| 75 | Rapika | √ | ✗ | √ | ✗ |
| 76 | Richeese | √ | ✗ | √ | √ |
| 77 | Rocky Snack | √ | ✗ | √ | √ |
| 78 | Scott Imulsion | √ | ✗ | √ | √ |
| 79 | Sensodyne | √ | ✗ | √ | √ |
| 80 | Serena | √ | ✗ | √ | √ |
| 81 | SGM 1 | √ | √ | √ | √ |
| 82 | SGM 2 | √ | √ | √ | √ |
| 83 | Shinzui | √ | √ | √ | √ |
| 84 | Snack Big Sheet | √ | ✗ | √ | √ |
| 85 | Soffel Kulit Jeruk | √ | ✗ | √ | √ |
| 86 | Springles | √ | ✗ | √ | ✗ |
| 87 | Stella Car Parfume | √ | √ | √ | √ |
| 88 | Susu Bendera 1000 ml | √ | √ | √ | √ |
| 89 | Tango Cruch Cake | √ | √ | √ | √ |
| 90 | Tango Fushion | √ | √ | √ | √ |
| 91 | Teh Botol Sosro | √ | √ | √ | √ |
| 92 | Teh Celup Dandang | √ | √ | √ | √ |
| 93 | Teh Celup Tong Tji | √ | ✗ | √ | ✗ |
| 94 | Torabika Cappucino | √ | √ | √ | √ |
| 95 | Treseme | √ | √ | √ | √ |
| 96 | Twister | √ | √ | √ | √ |
| 97 | Vape Liquid | √ | ✗ | √ | ✗ |
| 98 | Wafer Selamat | √ | √ | √ | √ |
| 99 | WRP Nutritious Drink | √ | √ | √ | √ |
| 100 | Zact | √ | ✗ | √ | √ |
| Jumlah berhasil dikenali | | 100 | 50 | 100 | 88 |

Kesimpulan:

| Metode | Citra Train | | Citra Uji | |
|-----------------|-----------------|-----------------------|-----------------|-----------------------|
| | Jumlah Berhasil | Jumlah Tidak Berhasil | Jumlah Berhasil | Jumlah Tidak Berhasil |
| Backpropagation | 100 | 0 | 50 | 50 |
| LVQ | 100 | 0 | 88 | 12 |

Variasi 4

Tabel 4.9 Hasil Pengenalan Variasi 4

| Learning Rate | Jumlah Neuron Hidden Layer | Epoch | MSE | | Time (Detik) | |
|---------------|----------------------------|-------|-----------------|-----|-----------------|-----|
| | | | Backpropagation | LVQ | Backpropagation | LVQ |
| 0,5 | 20 | 50 | 11,4 | 0 | 6 | 0,9 |

| No. | Nama Produk | Hasil Pengenalan | | | |
|-----|--------------------------------|------------------|-----------|-------------|-----------|
| | | Backpropagation | | LVQ | |
| | | Citra Train | Citra Uji | Citra Train | Citra Uji |
| 1. | ABC Kacang Hijau | ✗ | ✗ | ✓ | ✓ |
| 2. | Adem Sari Fres | ✗ | ✗ | ✓ | ✓ |
| 3. | Adidas Ice Dive | ✗ | ✗ | ✓ | ✓ |
| 4. | Anada Susu Bayi | ✗ | ✗ | ✓ | ✓ |
| 5. | Anlene Gold | ✓ | ✗ | ✓ | ✓ |
| 6. | Anlene | ✓ | ✓ | ✓ | ✓ |
| 7. | Annum | ✗ | ✗ | ✓ | ✓ |
| 8. | Astor | ✗ | ✗ | ✓ | ✓ |
| 9. | Baygon Mat Elektrik | ✓ | ✗ | ✓ | ✓ |
| 10. | Bellagio Home | ✓ | ✗ | ✓ | ✓ |
| 11. | Casablanca | ✓ | ✓ | ✓ | ✓ |
| 12. | Chocolatos | ✗ | ✗ | ✓ | ✓ |
| 13. | Citra Bubuk Mutiara Cina | ✗ | ✗ | ✓ | ✓ |
| 14. | Clear Man | ✗ | ✗ | ✓ | ✓ |
| 15. | Close Up | ✗ | ✗ | ✓ | ✓ |
| 16. | Cocoa | ✗ | ✗ | ✓ | ✓ |
| 17. | Copiko Brown Coffe | ✗ | ✗ | ✓ | ✓ |
| 18. | Dairy Milk | ✓ | ✗ | ✓ | ✓ |
| 19. | Dancow Actigo | ✓ | ✗ | ✓ | ✓ |
| 20. | Diabetasol Susu Formula 180 g | ✓ | ✗ | ✓ | ✓ |
| 21. | Diabetasol Susu Formula Coklat | ✓ | ✓ | ✓ | ✓ |

| | | | | | |
|----|-----------------------------|---|---|---|---|
| 22 | Dove White Beauty Bar | √ | ✗ | √ | ✗ |
| 23 | Emeron | ✗ | ✗ | √ | √ |
| 24 | Energen Sereal | ✗ | ✗ | √ | √ |
| 25 | Enzim | √ | √ | √ | √ |
| 26 | Fox | √ | √ | √ | √ |
| 27 | Frisian Baby 1 | √ | √ | √ | √ |
| 28 | Frisian Baby 2 | √ | √ | √ | √ |
| 29 | Frisian Flag Coklat 800 g | √ | ✗ | √ | √ |
| 30 | Funtime | √ | ✗ | √ | √ |
| 31 | Garnier Color Naturals | √ | √ | √ | √ |
| 32 | Garnier Extra UV Protection | √ | ✗ | √ | ✗ |
| 33 | Garnier Face Pouder | √ | ✗ | √ | √ |
| 34 | Garnier Night Restore | √ | √ | √ | √ |
| 35 | Gatsby Cool | √ | √ | √ | √ |
| 36 | Gatsby hair Shampoo | √ | √ | √ | √ |
| 37 | Glade 400 ml | √ | √ | √ | √ |
| 38 | Glade Automatic Spray | √ | ✗ | √ | √ |
| 39 | Goodday Mocafrio | ✗ | ✗ | √ | √ |
| 40 | Goodtime | √ | ✗ | √ | ✗ |
| 41 | Happy Tos | ✗ | ✗ | √ | √ |
| 42 | Head&Shoulders | √ | √ | √ | √ |
| 43 | Hello Panda | √ | √ | √ | √ |
| 44 | Hit Non_stop | √ | √ | √ | √ |
| 45 | Hydro Coco | ✗ | ✗ | √ | √ |
| 46 | Indomie | √ | ✗ | √ | ✗ |
| 47 | Jess Cool | √ | ✗ | √ | √ |
| 48 | Koko Krunch | ✗ | ✗ | √ | √ |
| 49 | Kopiko 78°C | √ | √ | √ | √ |
| 50 | La Fonte | √ | ✗ | √ | √ |
| 51 | Lactamil | √ | ✗ | √ | √ |
| 52 | Lactogen 2 | √ | ✗ | √ | √ |
| 53 | Larutan Penyegar Cap Kaki 3 | √ | ✗ | √ | √ |
| 54 | Lifebuoy Clear Skin | √ | √ | √ | √ |
| 55 | Lux Magic Spell | √ | √ | √ | √ |
| 56 | Max Tea | √ | ✗ | √ | √ |
| 57 | Maxi Class | √ | ✗ | √ | √ |
| 58 | Milkuat | √ | √ | √ | √ |
| 59 | Milna Biskuit Bayi | √ | √ | √ | √ |
| 60 | Milna Bubur Bayi | √ | ✗ | √ | ✗ |
| 61 | Mr Muscle | √ | √ | √ | √ |

| | | | | | |
|--------------------------|------------------------|----|----|-----|----|
| 62 | Nescafe | ✓ | ✓ | ✓ | ✓ |
| 63 | Nestle Cerelac | ✓ | ✓ | ✓ | ✓ |
| 64 | Nestle Coffe Mat | ✗ | ✗ | ✓ | ✗ |
| 65 | Nestle Milo Sereal | ✗ | ✗ | ✓ | ✓ |
| 66 | OBH | ✓ | ✗ | ✓ | ✓ |
| 67 | Olay Total Effect | ✓ | ✗ | ✓ | ✓ |
| 68 | Oreo | ✓ | ✗ | ✓ | ✓ |
| 69 | Ovaltine | ✓ | ✓ | ✓ | ✓ |
| 70 | Pentene Conditioner | ✓ | ✗ | ✓ | ✓ |
| 71 | Pepsodent Action | ✓ | ✓ | ✓ | ✓ |
| 72 | Pepsodent Center Fresh | ✓ | ✗ | ✓ | ✗ |
| 73 | Pepsodent Herbal | ✗ | ✗ | ✓ | ✓ |
| 74 | Ponds White Beauty | ✓ | ✗ | ✓ | ✗ |
| 75 | Rapika | ✓ | ✗ | ✓ | ✗ |
| 76 | Richeese | ✗ | ✗ | ✓ | ✓ |
| 77 | Rocky Snack | ✓ | ✗ | ✓ | ✓ |
| 78 | Scott Imulsion | ✗ | ✗ | ✓ | ✓ |
| 79 | Sensodyne | ✓ | ✗ | ✓ | ✓ |
| 80 | Serena | ✓ | ✗ | ✓ | ✓ |
| 81 | SGM 1 | ✓ | ✓ | ✓ | ✓ |
| 82 | SGM 2 | ✓ | ✓ | ✓ | ✓ |
| 83 | Shinzui | ✓ | ✓ | ✓ | ✓ |
| 84 | Snack Big Sheet | ✗ | ✗ | ✓ | ✓ |
| 85 | Soffel Kulit Jeruk | ✓ | ✗ | ✓ | ✓ |
| 86 | Springles | ✗ | ✗ | ✓ | ✗ |
| 87 | Stella Car Perfume | ✓ | ✓ | ✓ | ✓ |
| 88 | Susu Bendera 1000 ml | ✓ | ✓ | ✓ | ✓ |
| 89 | Tango Cruch Cake | ✓ | ✓ | ✓ | ✓ |
| 90 | Tango Fushion | ✓ | ✓ | ✓ | ✓ |
| 91 | Teh Botol Sosro | ✗ | ✗ | ✓ | ✓ |
| 92 | Teh Celup Dandang | ✓ | ✓ | ✓ | ✓ |
| 93 | Teh Celup Tong Tji | ✗ | ✗ | ✓ | ✗ |
| 94 | Torabika Cappucino | ✓ | ✓ | ✓ | ✓ |
| 95 | Tresemme | ✓ | ✓ | ✓ | ✓ |
| 96 | Twister | ✗ | ✗ | ✓ | ✓ |
| 97 | Vape Liquid | ✓ | ✗ | ✓ | ✗ |
| 98 | Wafer Selamat | ✗ | ✗ | ✓ | ✓ |
| 99 | WRP Nutritious Drink | ✗ | ✗ | ✓ | ✓ |
| 100 | Zact | ✓ | ✗ | ✓ | ✓ |
| Jumlah berhasil dikenali | | 70 | 35 | 100 | 88 |

Kesimpulan:

| Metode | Citra Train | | Citra Uji | |
|-----------------|-----------------|-----------------------|-----------------|-----------------------|
| | Jumlah Berhasil | Jumlah Tidak Berhasil | Jumlah Berhasil | Jumlah Tidak Berhasil |
| Backpropagation | 70 | 30 | 35 | 65 |
| LVQ | 100 | 0 | 88 | 12 |

Variasi 5

Tabel 4.10 Hasil Pengenalan Variasi 5

| Learning Rate | Jumlah Neuron Hidden Layer | Epoch | MSE | | Time (Detik) | |
|---------------|----------------------------|-------|-----------------|-----|-----------------|-----|
| | | | Backpropagation | LVQ | Backpropagation | LVQ |
| 1 | 10 | 20 | 616 | 0 | 2 | 0,3 |

| No. | Nama Produk | Hasil Pengenalan | | | |
|-----|--------------------------------|------------------|-----|-------------|-----------|
| | | Backpropagation | LVQ | Citra Train | Citra Uji |
| 1. | ABC Kacang Hijau | × | × | √ | √ |
| 2. | Adem Sari Fres | × | × | √ | √ |
| 3. | Adidas Ice Dive | × | × | √ | √ |
| 4. | Anada Susu Bayi | × | × | √ | √ |
| 5. | Anlene Gold | × | × | √ | √ |
| 6. | Anlene | × | × | √ | √ |
| 7. | Annum | × | × | √ | √ |
| 8. | Astor | × | × | √ | √ |
| 9. | Baygon Mat Elektrik | × | × | √ | √ |
| 10. | Bellagio Home | × | × | √ | √ |
| 11. | Casablanca | × | × | √ | √ |
| 12. | Chocolatos | × | × | √ | √ |
| 13. | Citra Bubuk Mutiara Cina | × | × | √ | √ |
| 14. | Clear Man | × | × | √ | √ |
| 15. | Close Up | × | × | √ | √ |
| 16. | Cocoa | × | × | √ | √ |
| 17. | Copiko Brown Coffe | × | × | √ | √ |
| 18. | Dairy Milk | × | × | √ | √ |
| 19. | Dancow Actigo | × | × | √ | √ |
| 20. | Diabetasol Susu Formula 180 g | × | × | √ | √ |
| 21. | Diabetasol Susu Formula Coklat | × | × | √ | √ |
| 22. | Dove White Beauty Bar | × | × | √ | ✗ |

| | | | | | |
|----|-----------------------------|---|---|---|---|
| 23 | Emeron | ✗ | ✗ | ✓ | ✓ |
| 24 | Energen Sereal | ✗ | ✗ | ✓ | ✓ |
| 25 | Enzim | ✗ | ✗ | ✓ | ✓ |
| 26 | Fox | ✗ | ✗ | ✓ | ✓ |
| 27 | Frision Baby 1 | ✗ | ✗ | ✓ | ✓ |
| 28 | Frision Baby 2 | ✓ | ✓ | ✓ | ✓ |
| 29 | Frision Flag Coklat 800 g | ✗ | ✗ | ✓ | ✓ |
| 30 | Funtime | ✗ | ✗ | ✓ | ✓ |
| 31 | Garnier Color Naturals | ✗ | ✗ | ✓ | ✓ |
| 32 | Garnier Extra UV Protection | ✗ | ✗ | ✓ | ✗ |
| 33 | Garnier Face Pouder | ✓ | ✗ | ✓ | ✓ |
| 34 | Garnier Night Restore | ✗ | ✗ | ✓ | ✓ |
| 35 | Gatsby Cool | ✗ | ✗ | ✓ | ✓ |
| 36 | Gatsby hair Shampoo | ✗ | ✗ | ✓ | ✓ |
| 37 | Glade 400 ml | ✗ | ✗ | ✓ | ✓ |
| 38 | Glade Automatic Spray | ✗ | ✗ | ✓ | ✓ |
| 39 | Goodday Mocafrio | ✗ | ✗ | ✓ | ✓ |
| 40 | Goodtime | ✗ | ✗ | ✓ | ✗ |
| 41 | Happy Tos | ✗ | ✗ | ✓ | ✓ |
| 42 | Head&Shoulders | ✗ | ✗ | ✓ | ✓ |
| 43 | Hello Panda | ✗ | ✗ | ✓ | ✓ |
| 44 | Hit Non_stop | ✗ | ✗ | ✓ | ✓ |
| 45 | Hydro Coco | ✗ | ✗ | ✓ | ✓ |
| 46 | Indomie | ✗ | ✗ | ✓ | ✗ |
| 47 | Jess Cool | ✗ | ✗ | ✓ | ✓ |
| 48 | Koko Krunch | ✗ | ✗ | ✓ | ✓ |
| 49 | Kopiko 78°C | ✗ | ✗ | ✓ | ✓ |
| 50 | La Fonte | ✗ | ✗ | ✓ | ✓ |
| 51 | Lactamil | ✗ | ✗ | ✓ | ✓ |
| 52 | Lactogen 2 | ✗ | ✗ | ✓ | ✓ |
| 53 | Larutan Penyegar Cap Kaki 3 | ✗ | ✗ | ✓ | ✓ |
| 54 | Lifebuoy Clear Skin | ✗ | ✗ | ✓ | ✓ |
| 55 | Lux Magic Spell | ✗ | ✗ | ✓ | ✓ |
| 56 | Max Tea | ✗ | ✗ | ✓ | ✓ |
| 57 | Maxi Class | ✗ | ✗ | ✓ | ✓ |
| 58 | Milkuat | ✗ | ✗ | ✓ | ✓ |
| 59 | Milna Biskuit Bayi | ✗ | ✗ | ✓ | ✓ |
| 60 | Milna Bubur Bayi | ✗ | ✗ | ✓ | ✗ |
| 61 | Mr Muscle | ✗ | ✗ | ✓ | ✓ |
| 62 | Nescafe | ✗ | ✗ | ✓ | ✓ |

| | | | | | |
|--------------------------|------------------------|---|---|-----|----|
| 63 | Nestle Cerelac | × | × | √ | √ |
| 64 | Nestle Coffe Mat | × | × | √ | × |
| 65 | Nestle Milo Sereal | × | × | √ | √ |
| 66 | OBH | √ | × | √ | √ |
| 67 | Olay Total Effect | × | × | √ | √ |
| 68 | Oreo | × | × | √ | √ |
| 69 | Ovaltine | × | × | √ | √ |
| 70 | Pentene Conditioner | × | × | √ | √ |
| 71 | Pepsodent Action | × | × | √ | √ |
| 72 | Pepsodent Center Fresh | × | × | √ | × |
| 73 | Pepsodent Herbal | × | × | √ | √ |
| 74 | Ponds White Beauty | × | × | √ | × |
| 75 | Rapika | √ | √ | √ | × |
| 76 | Richeese | × | × | √ | √ |
| 77 | Rocky Snack | × | × | √ | √ |
| 78 | Scott Imulsion | × | × | √ | √ |
| 79 | Sensodyne | × | × | √ | √ |
| 80 | Serena | × | × | √ | √ |
| 81 | SGM 1 | × | × | √ | √ |
| 82 | SGM 2 | × | × | √ | √ |
| 83 | Shinzui | × | × | √ | √ |
| 84 | Snack Big Sheet | √ | × | √ | √ |
| 85 | Soffel Kulit Jeruk | × | × | √ | √ |
| 86 | Springles | × | × | √ | × |
| 87 | Stella Car Perfume | √ | × | √ | √ |
| 88 | Susu Bendera 1000 ml | × | × | √ | √ |
| 89 | Tango Cruch Cake | × | × | √ | √ |
| 90 | Tango Fushion | × | × | √ | √ |
| 91 | Teh Botol Sosro | × | × | √ | √ |
| 92 | Teh Celup Dandang | × | × | √ | √ |
| 93 | Teh Celup Tong Tji | × | × | √ | × |
| 94 | Torabika Cappucino | × | × | √ | √ |
| 95 | Treseme | × | × | √ | √ |
| 96 | Twister | × | × | √ | √ |
| 97 | Vape Liquid | × | × | √ | × |
| 98 | Wafer Selamat | × | × | √ | √ |
| 99 | WRP Nutritious Drink | × | × | √ | √ |
| 100 | Zact | × | × | √ | √ |
| Jumlah berhasil dikenali | | 6 | 2 | 100 | 88 |

Kesimpulan:

| Metode | Citra Train | | Citra Uji | |
|------------------------|-----------------|-----------------------|-----------------|-----------------------|
| | Jumlah Berhasil | Jumlah Tidak Berhasil | Jumlah Berhasil | Jumlah Tidak Berhasil |
| <i>Backpropagation</i> | 6 | 94 | 2 | 98 |
| LVQ | 100 | 0 | 88 | 12 |

4.4 Analisis Hasil Kerja Sistem

Dari hasil pengujian sistem diperoleh hasil pengenalan *barcode* metode *backpropagation* sebesar 75,5%. Hasil ini sangatlah berbeda dengan hasil pengenalan yang dicapai oleh sistem pengenalan *barcode* metode *learning vector quantization* yaitu sebesar 94%. Hasil *recognition* pada pengujian sistem pengenalan LVQ cenderung sama, hal ini dikarenakan setiap 1 *feature barcode* akan dikelompokkan dalam 1 kelas yang berbeda, dengan kata lain 100 *feature barcode* dikelaskan ke dalam 100 kelas yang berbeda pula. Hal ini disebabkan setiap *barcode* memiliki karakteristik dan pola yang berbeda-beda, jadi variasi *learning rate* dan jumlah *hidden layer* pada pengujian sistem pengenalan LVQ hanya berpengaruh pada lama waktu pada saat *training* bukan pada hasil pengenalannya.

Akan dijelaskan sebab kecenderungan sistem mengalami kesalahan dalam mengenali citra, dengan kata lain sistem mengenali citra *barcode* tidak sesuai dengan *barcode* yang diujikan. Penulis menganalisis penyebab sistem salah mengenali citra yang diujikan, yaitu proses akuisisi citra.

Pada penelitian ini, akuisisi citra tidak menggunakan alat khusus. Artinya, data citra yang diperoleh dari proses foto dari supermarket GIANT jalan Siliwangi

Semarang menjadi tidak homogen dari segi pencahayaan dan jarak kamera dengan citra. Hal ini menimbulkan kondisi sebagai berikut.

1. Tingkat pencahayaan

Cahaya yang terlalu terang akan membuat citra yang diambil tampak lebih terang, kondisi ini menyebabkan tingkat kontras lebih tinggi. Sedangkan cahaya yang terlalu redup akan membuat citra yang diambil tampak lebih gelap, artinya citra memiliki tingkat kontras lebih rendah, sehingga akan mempengaruhi saat dilakukan ekstraksi ciri.

2. Jarak kamera dengan citra

Pada saat pengambilan citra sebaiknya dilakukan pada jarak yang sama. Artinya perlu dilakukan pembuatan alat bantu sistem yang dapat menunjang pengambilan citra pada jarak yang sama guna mendapatkan data citra yang seragam (homogen). Jika data citra yang diambil tidak seragam, maka akan didapatkan hasil pengenalan yang *random* dan kurang sesuai.

Kondisi-kondisi tersebut akan mempengaruhi sistem dalam mengenali citra *barcode* yang dilatihkan maupun diujikan, sehingga tingkat akurasi sistem menjadi lebih rendah dan menimbulkan kesalahan dalam proses pengenalan.

4.5 Membandingkan Hasil Sistem Pengenalan Barcode Menggunakan Metode Backpropagation dengan Metode Learning Vector Quantization

Setelah diperoleh hasil yang paling optimal dari proses pengujian sistem, langkah selanjutnya akan dilakukan proses perbandingan. Diperoleh 100 citra latih dan 51 citra uji yang berhasil dikenali dengan menggunakan metode *backpropagation* atau setara dengan 75,5%. Sedangkan hasil uji sistem menggunakan metode *learning vector quantization* diperoleh 100 citra latih dan 88 citra uji yang berhasil dikenali atau setara dengan 94%. Persentase hasil pengenalan citra *barcode* dapat dicari dengan rumus:

$$\text{Persentase akurasi} = \frac{\text{Jumlah citra barcode yang berhasil dikenali}}{\text{Jumlah citra yang diuji}} \times 100\%$$

Hasil ini diperoleh dengan menggunakan variasi terbaik *learning rate* = 0,05 dan jumlah *hidden layer* = 100 untuk *backpropagation*, sedangkan untuk *learning vector quantization* menggunakan variasi terbaik *learning rate* = 1 dan jumlah *hidden layer* = 10. Dengan variasi tersebut dapat diketahui waktu pembelajaran dan besarnya MSE. Waktu pembelajaran = 102 detik untuk *backpropagation* dan 0,3 detik pada saat pembelajar *learning vector quantization*. Pada *backpropagation* diperoleh MSE sebesar $6,45 \times 10^{-28}$, sedangkan pada *learning vector quantization* diperoleh MSE sebesar 0.

Dari hasil pengujian dan ilustrasi tersebut, dapat direkomendasikan metode yang paling baik untuk mengenali citra *barcode* dari segi akurasi dan waktu

pembelajaran. Metode *learning vector quantization* memiliki akurasi dan waktu pembelajaran yang lebih baik untuk pengenalan citra *barcode* dibandingkan dengan *backpropagation*, dimana dari hasil pengujian diperoleh tingkat akurasi pengenalan 94 % dan waktu pembelajaran 0,3 detik.

BAB V

SIMPULAN DAN SARAN

5.1 SIMPULAN

Dari hasil penelitian, pengolahan citra, perancangan, pembuatan, pengujian sistem hingga membandingkan hasil sistem pengenalan citra *barcode* menggunakan metode *backpropagation* dengan metode *learning vector quantization* pada penelitian ini, didapatkan simpulan sebagai berikut.

1. Terdapat 151 (75,5%) hasil pengenalan cocok dengan menggunakan metode *backpropagation* dan 188 (94%) hasil pengenalan cocok dengan menggunakan metode *learning vector quantization* dari total pengenalan citra sebanyak 100 citra latih dan 100 citra uji. Diperoleh waktu *training* 102 detik dengan menggunakan metode *backpropagation* dan 0,3 detik dengan menggunakan metode *learning vector quantization*.
2. Tingkat kecocokan dan akurasi hasil pengenalan tergantung pada variasi nilai parameter yang digunakan dalam proses pembelajarannya. Dari hasil pengujian, maka didapatkan variasi parameter terbaik dari *backpropagation* yaitu *learning rate* = 0,05 dan jumlah *hidden layer* = 100. Sedangkan variasi parameter terbaik dari *learning vector quantization* yaitu *learning rate* = 1 dan jumlah *hidden layer* = 10.
3. Dari hasil pengujian, diperoleh metode jaringan syaraf tiruan yang paling tepat untuk pengenalan citra *barcode* dari segi akurasi dan waktu, metode *learning*

vector quantization lebih baik dibandingkan dengan *backpropagation*. Dengan tingkat akurasi pengenalan 94 % dan waktu pembelajaran 0,3 detik.

5.2 SARAN

Dari hasil penelitian, pengolahan citra, perancangan, pembuatan, pengujian sistem, hingga membandingkan hasil sistem pengenalan citra *barcode* menggunakan metode *backpropagation* dengan metode *learning vector quantization*, saran yang dapat ditulis untuk penelitian ini adalah sebagai berikut.

1. Proses akuisisi citra dapat dibuat lebih baik dengan menciptakan sebuah alat khusus untuk mengakuisisi citra agar tingkat pencahayaan dan jarak kamera dengan objek sama (homogen). Sehingga dapat memperkecil tingkat kesalahan pengenalan oleh sistem.
2. Sistem dalam penelitian ini dibuat hanya untuk pengenalan *barcode*. Disarankan pada penelitian lebih lanjut dapat dikembangkan menjadi klasifikasi *barcode*.
3. Untuk penelitian selanjutnya dapat dicari metode jaringan syaraf tiruan yang lain, guna memperoleh hasil yang lebih baik dibandingkan dengan 2 metode yang digunakan pada penelitian ini.

DAFTAR PUSTAKA

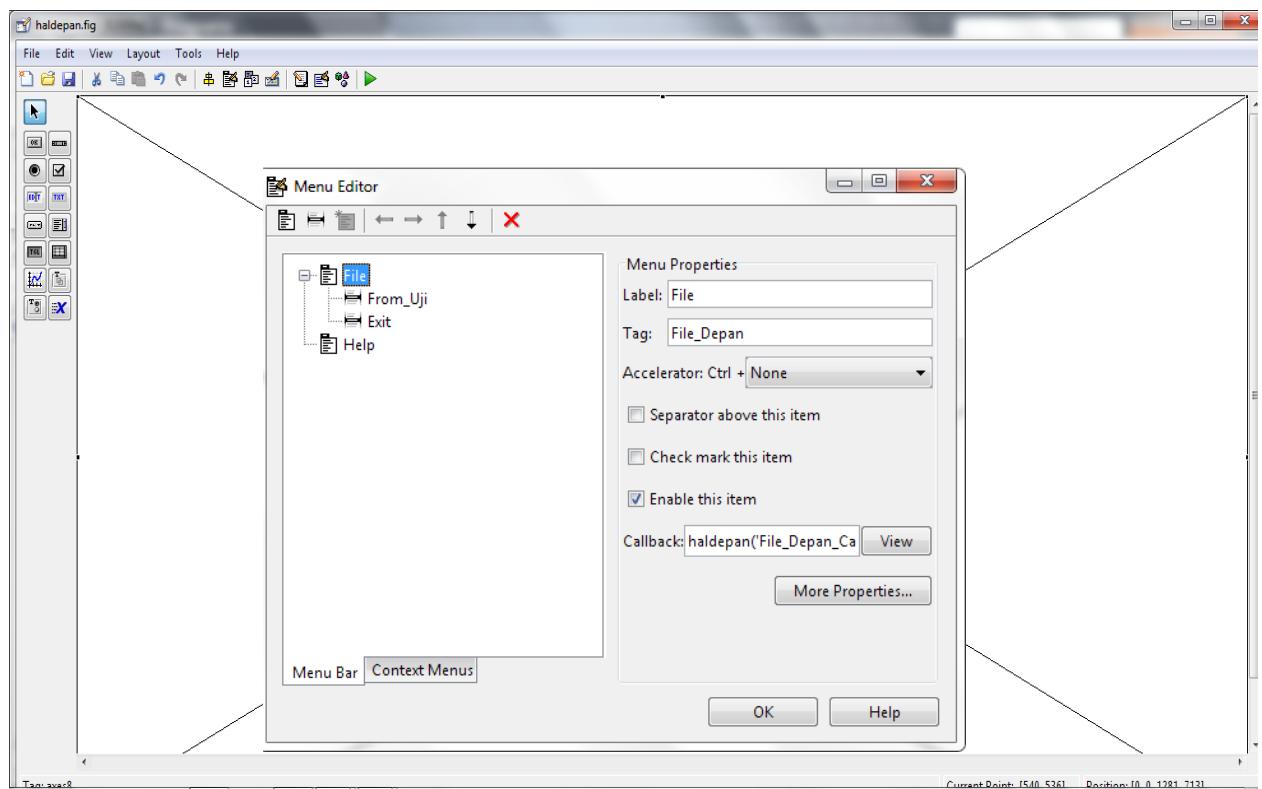
- Ahmad, A. M., Ismail, S., & Samaon, D. F. 2004. *Recurrent Neural Network with Backpropagation through Time for Speech Recognition*. International Symposium on Communications and Information Technologies. Oktober 26-29
- Ardhianto, E., Munawaroh, S. & Prihandono, A. 2011. Pengolahan Citra Digital untuk Identifikasi Ciri Sidik Jari Berbasis Minutiae. *Jurnal Teknologi Informatika DINAMIK*. 16 (1): 3
- Cahyono, G.P. 2006. *Sistem Pengenalan Barcode Menggunakan Jaringan Syaraf Tiruan Learning Vector Quantization*. FMIPA ITS
- Eliyani.2005. *Pengantar Jaringan Syaraf Tiruan*. MateriKuliah.com, hal: 1
- Fathoni, A. 2006. *Metodologi Penelitian dan Teknik Penyusunan Skripsi*. Jakarta: Rineka Cipta
- Fausett, L. 1994. *Fundamentals of Neural Network: Architectures, Algorithms, and Applications*. Prentice-Hall,Inc: New Jersey
- Gonzales, R. C., dkk. 2002. *Digital Image Processing Using Matlab*. Prentice Hall, Upper Saddle River, NJ
- Hermawan, A. 2006. *Jaringan Syaraf Tiruan: Teori dan Aplikasi*. Yogyakarta: ANDI
- Iqbal, M. 2009. *Dasar Pengolahan Citra Menggunakan MATLAB*. Departemen Ilmu dan Teknologi Kelautan ITB
- Kusumadewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu
- _____. 2004. *Membangun Jaringan Syaraf Tiruan (Menggunakan Matlab dan Excel Link)*. Yogyakarta: Graha Ilmu
- Malik,J.J., dkk. 2010. *Implementasi Tecnologi Barcode dalam Dunia Bisnis*. Yogyakarta: ANDI
- Nurismu, M.T., dkk. 2012. Rancangan Bangun Sistem Pengujian Distorsi Menggunakan Concentric Circle Method Pada Kaca Spion Kendaraan Bermotor Kategori L3 Berbasis Sdge Detection. *Jurnak Teknik POMITS*. 1 (1): 5
- Nurkhozin, A., dkk. 2011. Komparasi Hasil Klasifikasi Penyakit Diabetes Mellitus Menggunakan Jaringan Syaraf Tirua Backpropagation dan

- Learning Vector Quantization. *Prosiding Seminar Nasional Penelitian, Pendidikan dan Penerapan FMIPA UNY*. 14 Mei 2011
- Prabowo,A., dkk. 2006. Perbandingan Antara Metode Kohonen Neurel Network dengan Metode Learning Vektor Quantization Pada Pengenalan Pola Tandatangan. *Jurnal Sains dan Matematika (JSM)*. 14 (4): 1
- Prasetyo, E. 2011. *Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab*. Yogyakarta: ANDI
- Pratiarso, A., dkk .2009. *Perbandingan Metode POC, Backpropagation, Coding Pada Pembacaan Plat Nomor Kendaraan Berbasis Image Processing*. Politeknik Elektronika Negeri Surabaya.
- Pressman, R.S. 2002. *Rekayasa Perangkat Lunak*. Yogyakarta: ANDI
- Pujiyono, dkk. 2009. Perbandingan Kinerja Metode Gradient Berdasarkan Operator Sobel dan Prewitt Implementasi Pada Deteksi Sidik Jari. *Jurnal Informatika*. 3 (1): 5
- Puspitaningrum, D. 2006. *Pengantar Jaringan Syaraf Tiruan*. Yogyakarta: ANDI
- Putro, D.S. 2011. *Pengembangan Jaringan Syaraf Tiruan dengan Metode SOM dan LVQ Fuzzy*. Tesis. Depok. Universitas Indonesia
- Raheja, J.L, dan Kumar, U. 2010. Human Facial Expression Detection From Detected In Captured Image Using Propagation Neural Network. *Internasional journal of computer science and information technologi (IJCSIT)*. 2 (1): 7
- Sari, Z. W. 2010. *Pengenalan Pola Golongan Darah Menggunakan Jaringan Saraf Tiruan Backpropagation*. Skripsi. Malang: Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Malang
- Siang, J.J. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta: ANDI
- Suharto, I., dkk. 2004. *Perekayaan Metodologi Penelitian*. Yogyakarta: ANDI
- Wuryandari, M.D. dan Afrianto,I. 2012. Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation Dan Learning Vector Quantization Pada Pengenalan Wajah. *Jurnal Komputer dan Informatika (KOMPUTA)*. 1 (1): 1
- Yuan L, Jen L, and Rong T.1994. *A Bar-Code Recognition System Using Backpropagation Neural Networks*. pp. 81-90, 1995

LAMPIRAN-LAMPIRAN

Lampiran 1

Desain Tampilan Halaman Depan Program dengan Propertinya



| No. | Nama Komponen | Properti | Setting |
|-----|---------------|--------------|--|
| 1 | Panel | Title Tag | Menu Utama uipanel1 |
| 2 | Axes | Tag | axes8 |
| 3 | Menu Editor | Label Tag | File, Form_Uji, Exit, Help File_Depan, Uji_haldepan, Exit_Depan, Menu_Help |

Lampiran 2

Kode Program untuk Halaman Depan

```

function varargout = haldepan(varargin)
% HALDEPAN M-file for haldepan.fig
%     HALDEPAN, by itself, creates a new HALDEPAN or raises
% the existing
%     singleton*.
%
%     H = HALDEPAN returns the handle to a new HALDEPAN or
% the handle to
%     the existing singleton*.
%
%     HALDEPAN('CALLBACK', hObject, eventData, handles,...)
% calls the local
%     function named CALLBACK in HALDEPAN.M with the given
% input arguments.
%
%     HALDEPAN('Property','Value',...) creates a new
% HALDEPAN or raises the
%     existing singleton*. Starting from the left,
% property value pairs are
%     applied to the GUI before haldepan_OpeningFcn gets
% called. An
%     unrecognized property name or invalid value makes
% property application
%     stop. All inputs are passed to haldepan_OpeningFcn
% via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI
% allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
haldepan

% Last Modified by GUIDE v2.5 31-Jul-2013 11:37:10

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',          mfilename, ...
                   'gui_Singleton',    gui_Singleton, ...
                   'gui_OpeningFcn',   @haldepan_OpeningFcn,
...
                   'gui_OutputFcn',    @haldepan_OutputFcn,
...
                   'gui_LayoutFcn',    [], ...
                   'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

```

```

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before haldepan is made visible.
function haldepan_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see
GUIDATA)
% varargin     command line arguments to haldepan (see
VARARGIN)
axes(handles.axes8);
image(imread('Background.jpg'));
axis off;
% Choose default command line output for haldepan
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes haldepan wait for user response (see
UIRESUME)
% uiwait(handles.haldepan);

% --- Outputs from this function are returned to the command
line.
function varargout = haldepan_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see
GUIDATA)

% Get default command line output from handles structure

% -----
function File_Depan_Callback(hObject, eventdata, handles)
% hObject    handle to File_Depan (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see
GUIDATA)

```

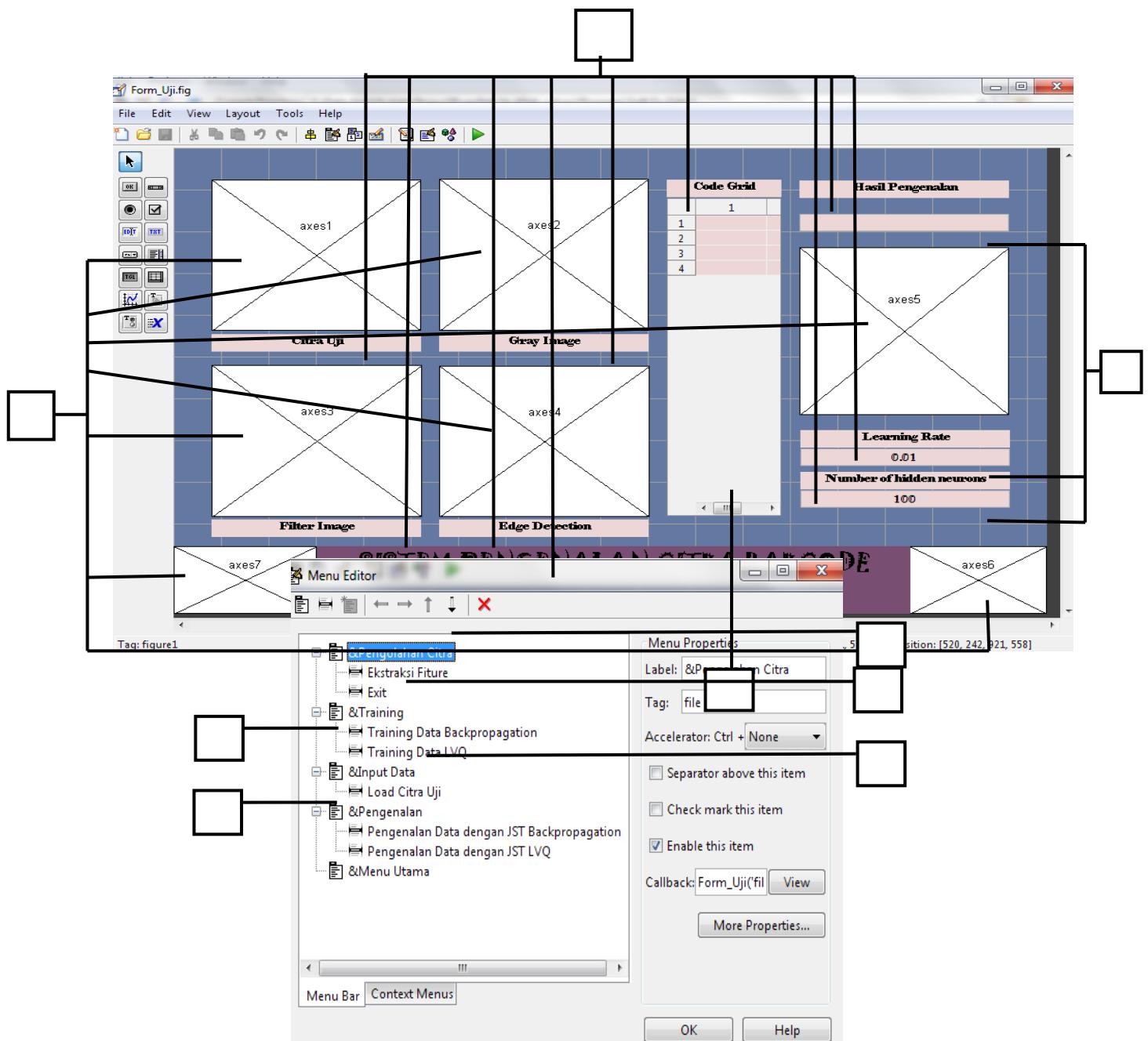
```
% -----
-----
function Uji_haldepan_Callback(hObject, eventdata, handles)
% hObject    handle to Uji_haldepan (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)
delete(handles.haldepan);
Form_Uji

% -----
-----
function Exit_Depan_Callback(hObject, eventdata, handles)
% hObject    handle to Exit_Depan (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)
pos_size=get(handles.haldepan,'position');
user_response=tanya_keluar_utama('Exit','Konfirmasi
Mengakhiri Program');
switch user_response
    case {'No'}
    case 'Yes'
        delete(handles.haldepan);
    close
end

% -----
-----
function Menu_Help_Callback(hObject, eventdata, handles)
% hObject    handle to Menu_Help (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)
%delete(handles.haldepan);
Form_Help
```

Lampiran 3

Desain Tampilan Form_Uji dengan Propertinya



| No | Nama Komponen | Properti | Setting |
|----|--------------------|--------------|--|
| 1 | Axes | Tag | axes 1, axes 2, axes 3, axes 4, axes 5, axes 6, axes 7 |
| 2 | <i>Static text</i> | Title | Citra Uji, <i>Gray Image, Filter Image, Edge Detection, Code Grid</i> , Hasil Pengenalan, <i>Learning Rate, Number of hidden neuron</i> , Sistem pengenalan citra <i>barcode</i> |
| | | Tag | Text1, text2, text3, text4, text7, text6, text8, text9, text5 |
| 3 | <i>Edit text</i> | Tag | Edit1, edit2, edit3 |
| 4 | <i>Table</i> | Tag | uitable1 |
| 5 | <i>Menu Editor</i> | Title | &Pengolahan Citra Digital, Ekstraksi Fiture, Exit |
| | | Tag | File, fiture, Exit |
| 6 | <i>Menu Editor</i> | Title | &Training, Training Data Backpropagation, Training Data LVQ |
| | | Tag | Training_Training_Data_Backpro_Trian_LVQ |
| 7 | <i>Menu Editor</i> | Title | &Input Data, Load Citra Uji |
| | | Tag | Input_Data, Load_Citra_Uji |
| 8 | <i>Menu Editor</i> | Title | &Pengenalan, Pengenalan Data dengan JST Backpropagation, Pengenalan Data dengan JST LVQ |
| | | Tag | Pengenalan, Pengenalan_Backpro, Pengenalan_LVQ |
| 9 | <i>Menu Editor</i> | Title Tag | &Menu Utama Menu_Utama |

Lampiran 4

Kode Program untuk Form_Uji

```

function varargout = Form_Uji(varargin)
% FORM_UJI M-file for Form_Uji.fig
%     FORM_UJI, by itself, creates a new FORM_UJI or raises
the existing
%     singleton*.
%
%     H = FORM_UJI returns the handle to a new FORM_UJI or
the handle to
%     the existing singleton*.
%
%     FORM_UJI('CALLBACK', hObject, eventData, handles,...)
calls the local
%         function named CALLBACK in FORM_UJI.M with the given
input arguments.
%
%         FORM_UJI('Property','Value',...) creates a new
FORM_UJI or raises the
%         existing singleton*. Starting from the left,
property value pairs are
%         applied to the GUI before Form_Uji_OpeningFcn gets
called. An
%         unrecognized property name or invalid value makes
property application
%         stop. All inputs are passed to Form_Uji_OpeningFcn
via varargin.
%
%         *See GUI Options on GUIDE's Tools menu. Choose "GUI
allows only one
%         instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
Form_Uji

% Last Modified by GUIDE v2.5 30-Jul-2013 06:14:35

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',          mfilename, ...
                   'gui_Singleton',    gui_Singleton, ...
                   'gui_OpeningFcn',   @Form_Uji_OpeningFcn,
...
                   'gui_OutputFcn',    @Form_Uji_OutputFcn,
...
                   'gui_LayoutFcn',    [], ...
                   'gui_Callback',     []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

```

```

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Form_Uji is made visible.
function Form_Uji_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see
GUIDATA)
% varargin command line arguments to Form_Uji (see
VARARGIN)

% Choose default command line output for Form_Uji
handles.output = hObject;
handles.axesgui=
[handles.axes1;handles.axes2;handles.axes3;handles.axes4;handles.axes5];
set(handles.axesgui,'XTickLabel',[],'XTick',[],...
'YTickLabel',[],'YTick',[],'box','on');

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Form_Uji wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command
line.
function varargout = Form_Uji_OutputFcn(hObject, eventdata,
handles)
% varargout cell array for returning output args (see
VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see
GUIDATA)
axes(handles.axes6);
image(imread('Bar1.jpg'));
axis off;
axes(handles.axes7);
image(imread('Bar1.jpg'));
axis off;

```

```
% Get default command line output from handles structure
varargout{1} = handles.output;

% -----
function file_Callback(hObject, eventdata, handles)
% hObject    handle to file (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% -----
function      Load_Citra_Uji_Callback(hObject,      eventdata,
handles)
% hObject    handle to Load_Citra_Uji (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)
axes(handles.axes1);
global files
global patch
[filename,           pathname] =
uigetfile({'*.jpg';'*.bmp';'*.gif';'*.*'}, 'Load Image');
if isequal(filename,0)
    return;
end
I = imread([pathname,filename]);
imshow (I);
b=20;
c=30;
RE=imresize (I,[b c]);
%save ('RE');

axes(handles.axes2);
J = rgb2gray(RE); % Mengkonversi gambar RGB ke Grayscale
imshow(J);

axes (handles.axes3);
%J=histeq(J,64)
h=fspecial('unsharp'); % Image Filtering penajaman citra
K=imfilter(J,h);
imshow (K);

z=double(reshape(K,[],1));

axes(handles.axes4);
%g_sobel_best=im2bw(K,0.6);
g_sobel_best = edge(K,'sobel',0.08);
imshow(g_sobel_best);
code_grid=g_sobel_best(b/2,:);
```

```

t = uitable(handles.uitable1);
set(t,'Data',double(code_grid'));
save code_grid.mat code_grid
%imshow(g)
%glcm= graycomatrix(g_sobel_best);

%
% -----
function fiture_Callback(hObject, eventdata, handles)
% hObject    handle to fiture (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)
code_grid_all_pro=[];
[nama_file1,nama_path1]=uigetfile({'*.jpg'},'Load File');
if isequal(nama_file1,0)
    return;
end

%=====
%lokasi file input
patch = fullfile(nama_path1);
%scan(patch)
files = dir([patch '*.jpg']);
code_grid_all=[];
global files
global patch

cc=length(files)
whos cc
for n = 1 : length(files)
    files = dir([patch '*.jpg']);
    filename = files(n).name;
    file = imread([patch filename]);
    axes(handles.axes1);
    imshow(file)
    b=20;
    c=30;
    RE=imresize (file,[b c]);
    axes(handles.axes2);
    J = rgb2gray(RE);      % Mengkonversi gambar RGB ke
Grayscale
    imshow(J);

    axes (handles.axes3);
    %J=histeq(J,64)
    h=fspecial('unsharp');      % Image Filtering penajaman
citra
    K=imfilter(J,h);
    imshow (K);

```

```

axes(handles.axes4);
%g_sobel_best=im2bw(K,0.6);
g_sobel_best = edge(K, 'sobel',0.08);
imshow(g_sobel_best);
%z=double(reshape(K,[],1))/255;
%code_grid=z';
code_grid=g_sobel_best(b/2,:);
whos code_grid
code_grid_all=[code_grid_all;code_grid];
t = uitable(handles.uitable1);
set(t,'Data',double(code_grid'));

end
%whos code_grid_all
code_grid_all_pro=[code_grid_all_pro; code_grid_all];
whos code_grid_all_pro
save code_grid_all_pro.mat code_grid_all_pro
save files files
save patch patch

% -----
-----
function Training_Data_Backpro_Callback(hObject, eventdata,
handles)
% hObject    handle to Training_Data_Backpro (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)
%Mengambil data feature
Train=open('code_grid_all_pro.mat');
Train = struct2cell(Train);
Train =cell2mat(Train);
whos Train
u=Train;
%n=100; %jumlah clas
Train = u;%data feature
%ukuran feature
[a,b] = size(Train);
%% prepare the input/output vectors (1-of-N output encoding)
input = Train'; %'matrix of size numFeatures-
by-numImages
whos input
%membuat inisial
target=1:a;
target=target;
whos target % matrix target
LearningRate = str2num(get(handles.edit2,'string'));
Hiddenlayer = str2num(get(handles.edit3,'string'));
% membentuk jaringan backpro
net=newff(minmax(input),[Hiddenlayer           1],{'logsig'
'purelin'});
net=init(net);
net.trainParam.epochs = 1000;
net.trainParam.lr=LearningRate;

```

```

%% training
net=train(net,input,target);
output=sim(net,input)
save net net;
%nntraintool('close')
figure(1)
plot(input,target,'bo',input,output,'r*');
xlabel('input')
ylabel('Target atau output')
grid on

% -----
%
function Exit_Callback(hObject, eventdata, handles)
% hObject    handle to Exit (see GCBO)
% eventdata   reserved - to be defined in a future version of
% MATLAB
% handles      structure with handles and user data (see
% GUIDATA)
close;

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata   reserved - to be defined in a future version of
% MATLAB
% handles      structure with handles and user data (see
% GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as
text
%         str2double(get(hObject,'String')) returns contents
of edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata   reserved - to be defined in a future version of
% MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata   reserved - to be defined in a future version of
% MATLAB
% handles      structure with handles and user data (see
% GUIDATA)

%
% -----
%
function Training_Callback(hObject, eventdata, handles)
% hObject    handle to Training (see GCBO)
% eventdata   reserved - to be defined in a future version of
% MATLAB
% handles      structure with handles and user data (see
% GUIDATA)

%
% -----
%
function Pengenalan_Callback(hObject, eventdata, handles)
% hObject    handle to Pengenalan (see GCBO)
% eventdata   reserved - to be defined in a future version of
% MATLAB
% handles      structure with handles and user data (see
% GUIDATA)

%
% -----
%
function Uji_Callback(hObject, eventdata, handles)
% hObject    handle to Uji (see GCBO)
% eventdata   reserved - to be defined in a future version of
% MATLAB
% handles      structure with handles and user data (see
% GUIDATA)

%
% -----
%
function Trian_LVQ_Callback(hObject, eventdata, handles)
% hObject    handle to Trian_LVQ (see GCBO)
% eventdata   reserved - to be defined in a future version of
% MATLAB
% handles      structure with handles and user data (see
% GUIDATA)
%mengambil data feature
tic
global w
Train=open('code_grid_all_pro.mat');
Train = struct2cell(Train);
Train =cell2mat(Train);
whos Train
```

```

u=Train;
%n=100; %jumlah clas
Train = u;%data feature
%ukuran feature
[a,b] = size(Train);
%% prepare the input/output vectors (1-of-N output encoding)
input = Train; %'matrix of size numFeatures-
by-numImages
whos input
%membuat inisial
target=1:a;
target=target;

P=input;
whos P
T=target;
whos Tc

%bobot awal
w=P;
[m n]=size(P)
[m1 n1]=size(w)
LearningRate = str2num(get(handles.edit2,'string'));

%set parameter
alfa = LearningRate;
DecAlfa=0.01;
MinAlfa =0.001;
MaxEpohs=1000;

%Mulai iterasi
epoh =0;
while (epoh<=MaxEpohs) & (alfa>=MinAlfa),
    epoh=epoh+1;
    for k=1:m,
        for c=1:m1,
            D(c)=0;
            for i=1:n,
                D(c)=D(c)+(w(c,i)-P(k,i))^2;
            end
            D(c)=sqrt(D(c));
        end;
        [Jmin idx]=min(D);
        if idx==T(k),
            w(idx,:)=w(idx,:)+alfa*(P(k,:)-w(idx,:));
        else
            w(idx,:)=w(idx,:)-alfa*(P(k,:)-w(idx,:));
        end;
    end
    disp(strcat('Epoh
',int2str(epoh),';alfa=',num2str(alfa)));
    alfa=alfa-DecAlfa*alfa;
end;

%Display Bobot Akhir

```

```

BobotAkhir = w
whos BobotAkhir

%pengujian dengan data pelatihan
X=P;
H=[];
for k=1:size(X,1),
    for c=1:m1,
        D(c)=0;
        for i=1:n,
            D(c)=D(c)+(w(c,i)-P(k,i))^2;
        end;
        D(c)=sqrt(D(c));
    end;
    [Jmin idx]=min(D);
    H=[H; idx];
end
disp('Data ke- Target Hasil')
disp([(1:m)' T' H])
save w w

%hitung mse
D = abs(T'-H).^2;
mse = sum(D(:))/numel(T')

waktu=toc

% -----
-----
function Pengenalan_LVQ_Callback(hObject, eventdata,
handles)
% hObject handle to Pengenalan_LVQ (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see
GUIDATA)
%mengambil data grid
load w
Uji = open('code_grid.mat');
Uji = struct2cell(Uji);
Uji = cell2mat(Uji);
Test= double(Uji)
whos Test
cc=[];
for b=1:size(w,1)
    s=abs(Test-w(b,:));
    s=mean(s)
    cc=[cc s];
end
[v dd]=min(cc)

load files
load patch
%% Testing (on Test data)

```

```

filename = files(dd).name
file = imread(fullfile('Foto Produk\',filename));

% membaca nama depan file/user
k = strfind(filename, '.');
j=num2str(k-1);
s=['%',j,'c'];
[nama, pos] = textscan(filename,s, 1);
namauser=cell2mat(nama);

axes(handles.axes5);
imshow(file)
set(handles.edit1,'string',namauser)

%
-----
function Pengenalan_Backpro_Callback(hObject, eventdata,
handles)
% hObject    handle to Pengenalan_Backpro (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)
Uji = open('code_grid.mat');
Uji = struct2cell(Uji);
Uji = cell2mat(Uji);
Test= double(Uji)';
whos Test
load net;
load files
load patch
% Testing (on Test data)
y = sim(net, Test)
n=int8(y);
whos n
filename = files(n).name
whos filename
file = imread(fullfile('Foto Produk\',filename));
whos file
% membaca nama depan file/user
k = strfind(filename, '.');
j=num2str(k-1);
s=['%',j,'c'];
[nama, pos] = textscan(filename,s, 1);
namauser=cell2mat(nama);
axes(handles.axes5);
imshow(file)
set(handles.edit1,'string',namauser)

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB

```

```
% handles      structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as
text
%           str2double(get(hObject,'String')) returns contents
of edit2 as a double

% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if     ispc    &&    isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as
text
%           str2double(get(hObject,'String')) returns contents
of edit3 as a double

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if     ispc    &&    isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');

end

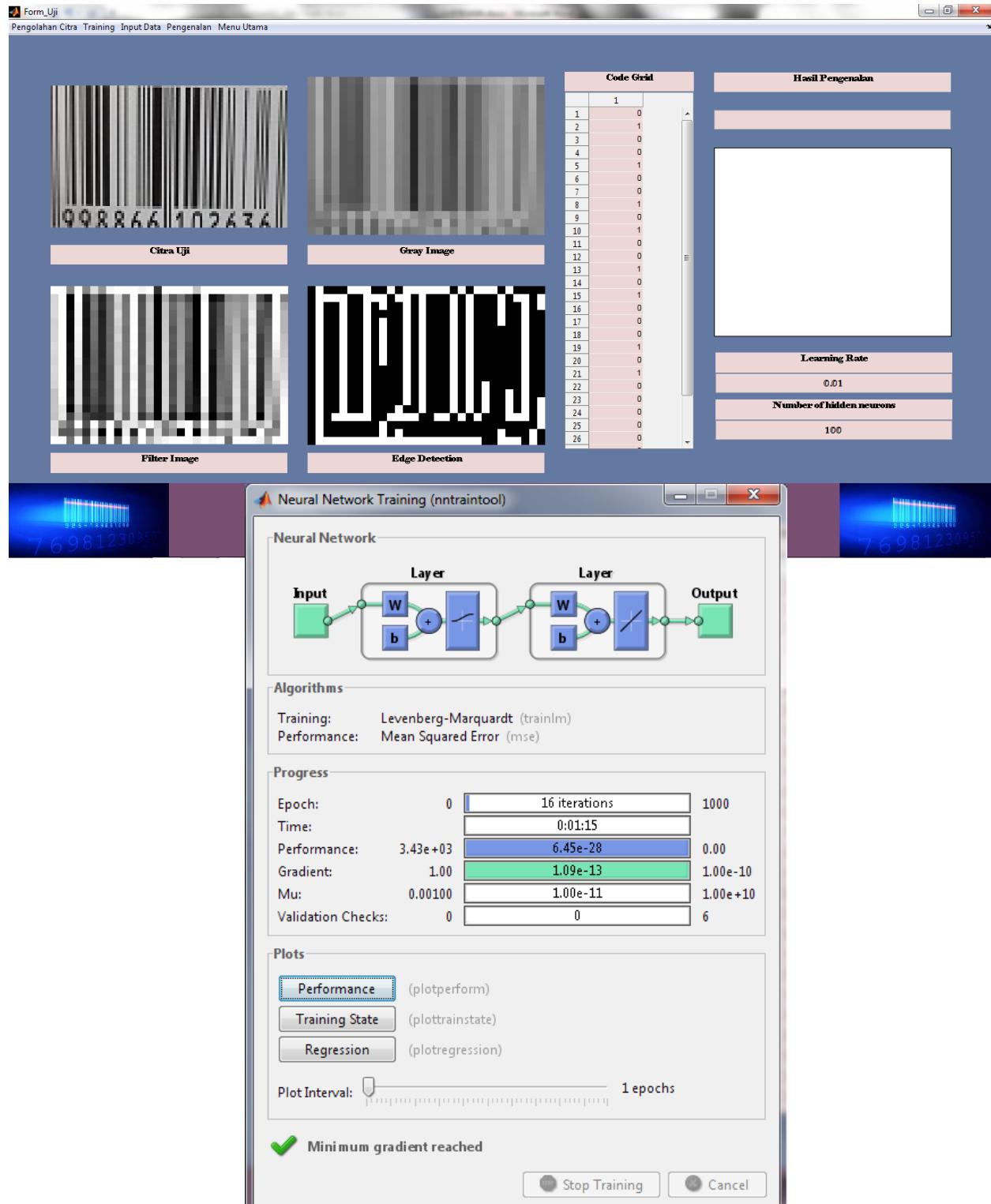
% -----
function Input_Data_Callback(hObject, eventdata, handles)
% hObject    handle to Input_Data (see GCBO)
% eventdata   reserved - to be defined in a future version of
% MATLAB
% handles      structure with handles and user data (see
% GUIDATA)

% -----
function Menu_Utama_Callback(hObject, eventdata, handles)
% hObject    handle to Menu_Utama (see GCBO)
% eventdata   reserved - to be defined in a future version of
% MATLAB
% handles      structure with handles and user data (see
% GUIDATA)
pos_size=get(handles.figure1,'position');
user_response=tanya_kembali_utama('Menu Utama','Konfirmasi
Kembali ke Menu Utama');
switch user_response
    case {'No'}
    case 'Yes'
        delete(handles.figure1);
        haldepan
end
```

Lampiran 5

Contoh Program Saat Dijalankan





MATLAB 7.8.0 (R2009a)

File Edit Debug Parallel Desktop Window Help

Current Directory: E:\Data Azizi\Kuliah\Skripsi\Bismillah Ya Allah sukses\Program\Soft Fix Edit 2

Shortcuts How to Add What's New

Command Window

New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```

80 80 80
81 81 81
82 82 82
83 83 83
84 84 84
85 85 85
86 86 86
87 87 87
88 88 88
89 89 89
90 90 90
91 91 91
92 92 92
93 93 93
94 94 94
95 95 95
96 96 96
97 97 97
98 98 98
99 99 99
100 100 100

mse =
0

waktu =
4.5325

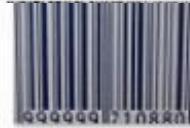
f2 >>

```

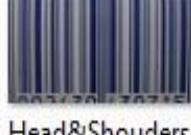
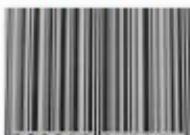
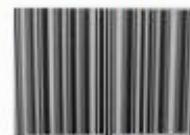
[Close](#)

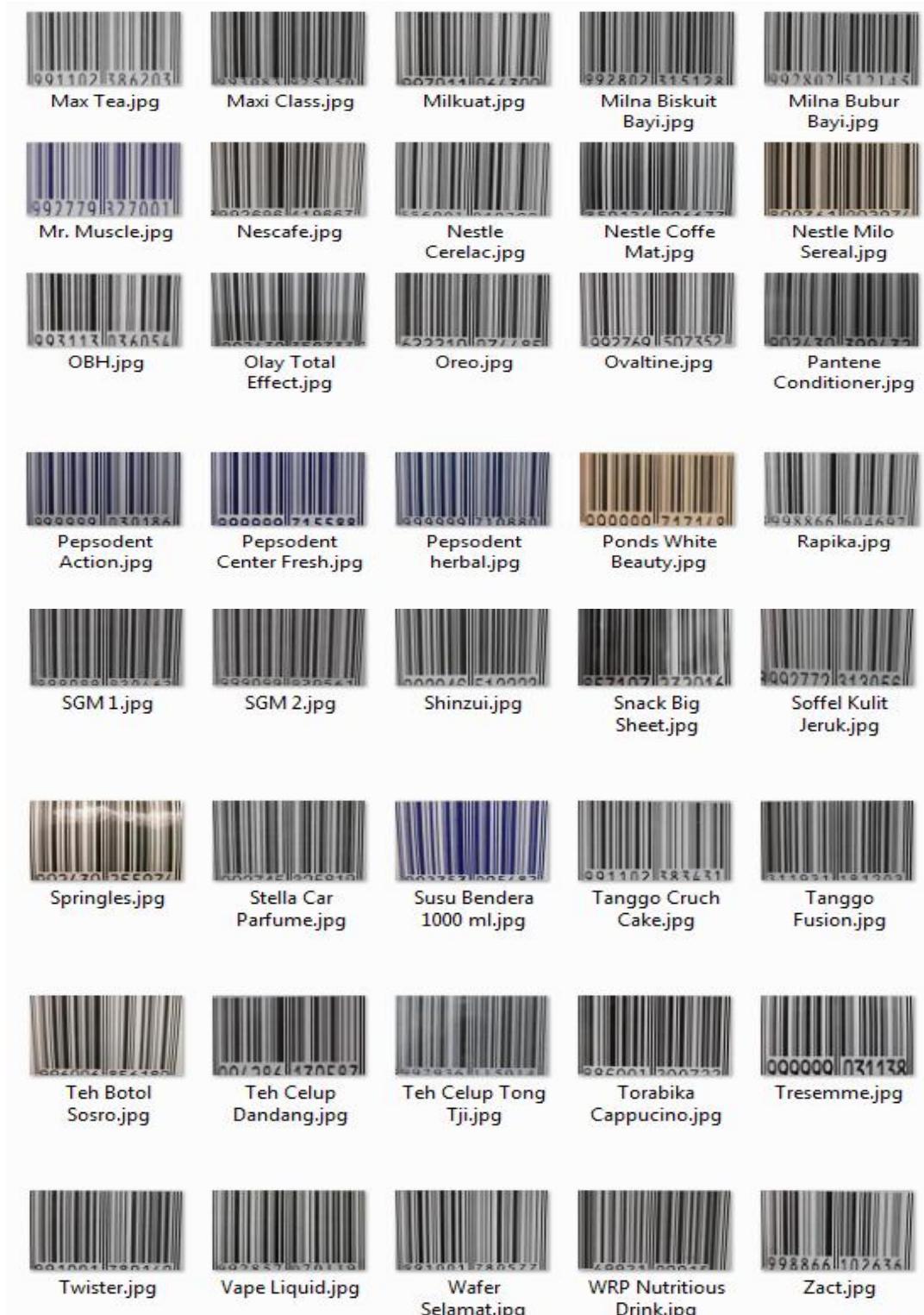


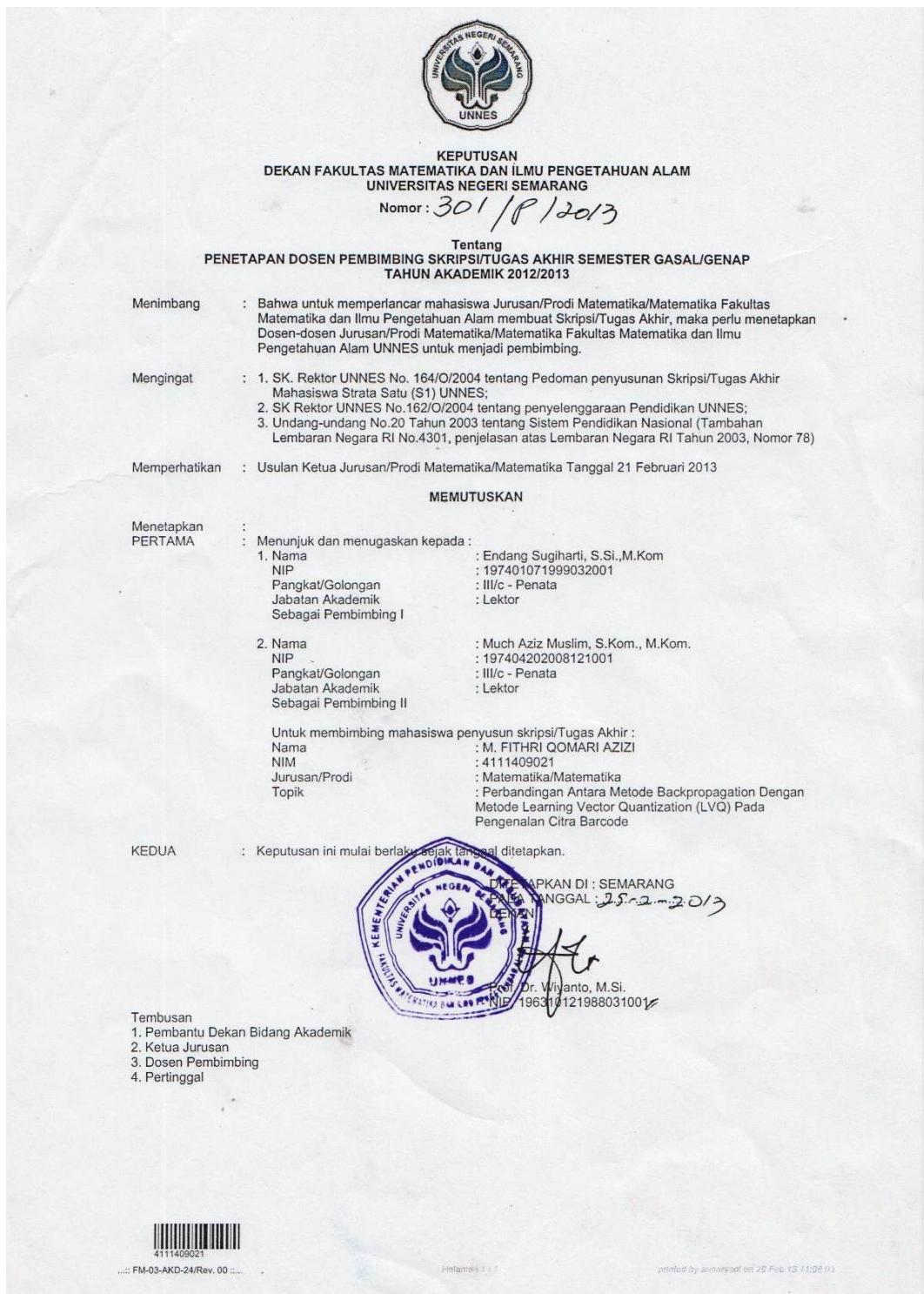
*Lampiran 6***Citra Latih Barcode**

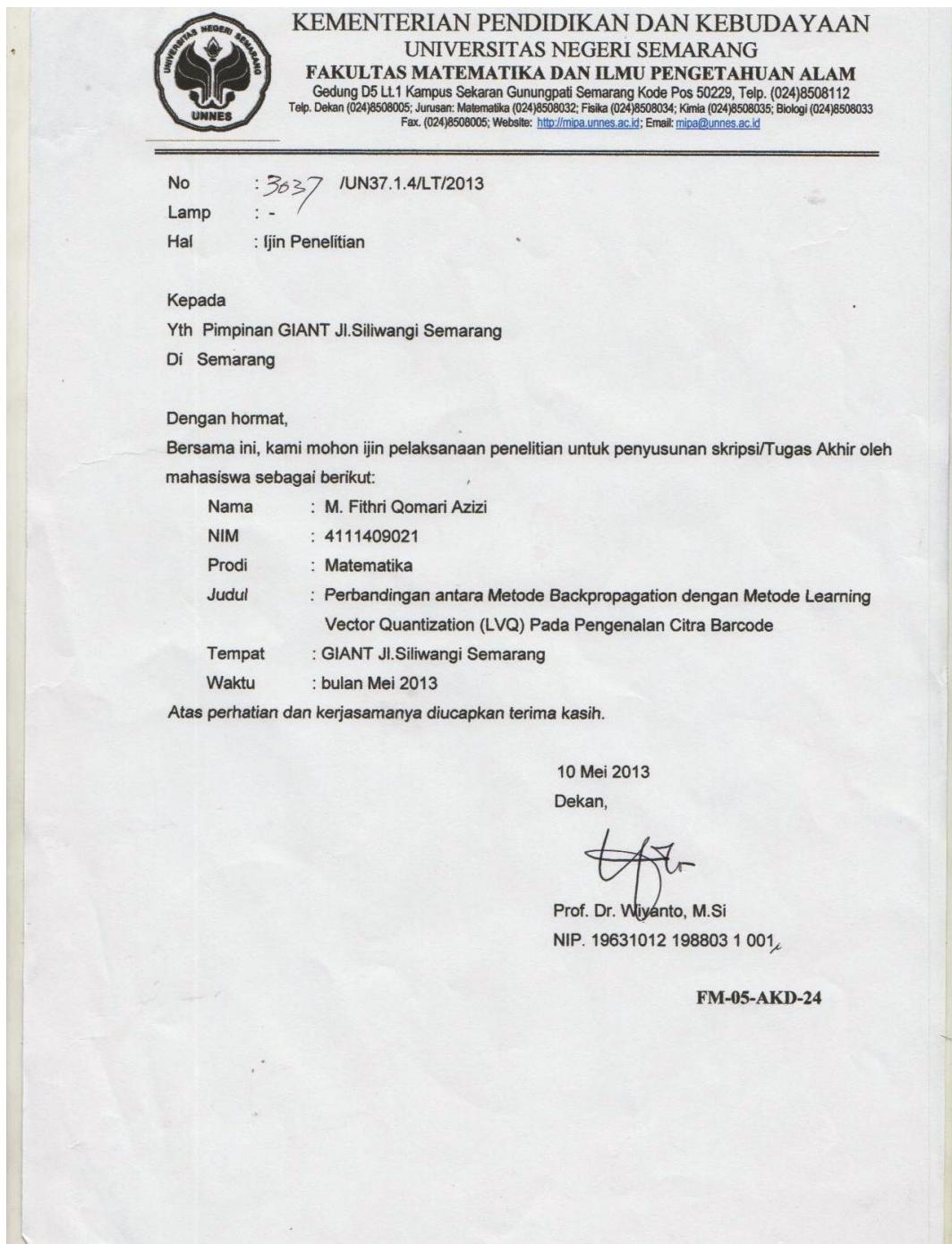
| | | | | |
|---|--|--|--|--|
|  Pepsodent Action.jpg |  Pepsodent Center Fresh.jpg |  Pepsodent herbal.jpg |  Ponds White Beauty.jpg |  Rapika.jpg |
|  Richeese.jpg |  Rocky Snack.jpg |  Scotts Imulsion.jpg |  Sensodyne.jpg |  Serena.jpg |
|  SGM 1.jpg |  SGM 2.jpg |  Shinzui.jpg |  Snack Big Sheet.jpg |  Soffel Kulit Jeruk.jpg |
|  Springles.jpg |  Stella Car Perfume.jpg |  Susu Bendera 1000 ml.jpg |  Tango Cruch Cake.jpg |  Tango Fushion.jpg |
|  Teh Botol Sosro.jpg |  Teh Celup Dandang.jpg |  Teh Celup Tong Tji.jpg |  Torabika Cappuccino.jpg |  Tresemme.jpg |
|  Twister.jpg |  Vape Liquid.jpg |  Wafer |  WRP Nutritious |  Zact.jpg |

*Lampiran 7***Citra Uji (Non-Latih) Barcode**

| | | | | |
|---|---|---|--|---|
|  |  |  |  |  |
| Fox.jpg | Frisian Baby 1.jpg | Frisian Baby 2.jpg | Frisian Flag Coklat 800 g.jpg | Funtime.jpg |
|  |  |  |  |  |
| Garnier Color Naturals.jpg | Garnier Extra UV Protection.jpg | Garnier Face Pouder.jpg | Garnier Night Restore.jpg | Gatsby Cool.jpg |
|  |  |  |  |  |
| Gatsby Hair Shampoo.jpg | Glade 400ml.jpg | Glade Automatic Spray.jpg | Good day Mocafrio.jpg | Goodtime.jpg |
|  |  |  |  |  |
| Happy Tos.jpg | Head&Shoulders.jpg | Hello Panda.jpg | Hit Non-stop.jpg | Hydro Coco.jpg |
|  |  |  |  |  |
| Indomie.jpg | Jess Cool.jpg | Koko Krunch.jpg | Kopiko 78C.jpg | La Fonte.jpg |
|  |  |  |  |  |
| Lactamil.jpg | Lactogen 2.jpg | Larutan Penyegar Cap Kaki 3.jpg | Lifebuoy Clear Skin.jpg | Lux Magic Spell.jpg |



*Lampiran 8***Surat Ketetapan Dosen Pembimbing**

*Lampiran 9***Surat Ijin Penelitian dari Fakultas**

*Lampiran 10***Surat Balasan Ijin Penelitian dari GIANT SUPERDOME Semarang**