

**KLASIFIKASI DAGING BABI DAN SAPI MENGGUNAKAN
EKSTRAKSI FITUR WARNA DAN TEKSTUR DENGAN
METODE *GRAY LEVEL CO-OCCURRENCE MATRIX***

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV
Politeknik Negeri Malang

Oleh:

**ROBERTUS ROMARIO HARVEY KURNIAWAN
NIM. 1341180117**



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
JULI 2017**

**KLASIFIKASI DAGING BABI DAN SAPI MENGGUNAKAN
EKSTRAKSI FITUR WARNA DAN TEKSTUR DENGAN
METODE *GRAY LEVEL CO-OCCURRENCE MATRIX***

SKRIPSI

Digunakan Sebagai Syarat Maju Ujian Diploma IV
Politeknik Negeri Malang

Oleh:

**ROBERTUS ROMARIO HARVEY KURNIAWAN
NIM. 1341180117**



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
JULI 2017**

HALAMAN PENGESAHAN

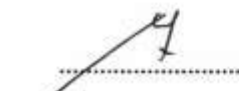

KLASIFIKASI DAGING BABI DAN SAPI MENGGUNAKAN EKSTRAKSI FITUR WARNA DAN TEKSTUR DENGAN METODE *GRAY LEVEL CO-OCCURRENCE MATRIX*

Disusun oleh:

ROBERTUS ROMARIO HARVEY KURNIAWAN
NIM. 1341180117

Laporan Skripsi ini telah diuji pada tanggal 17 Juli 2017

Disetujui oleh:

- | | | | |
|------------------|---|---|---|
| 1. Penguji I | : | Ekojono, S.T., M.KOM
NIP. 19591208 198503 1 004 |  |
| 2. Penguji II | : | Mungki Astiningrum, S.T., M.KOM
NIP. 19771030 200501 2 001 | |
| 3. Pembimbing I | : | Dr. Eng. Rosa Andrie Asmara, S.T., M.T.
NIP. 19801010 200501 1 001 |  |
| 4. Pembimbing II | : | Kadek Suarjuna Batubulan, S.Kom., M.T.
NIP. | |

Mengetahui,

Ketua Jurusan
Teknologi Informasi

Ketua Program Studi
Teknik Informatika



Rudy Ariyanto, S.T., M.Cs.
NIP. 19711110 199903 1 002

Ir. Deddy Kusbianto P., M.MKom.
NIP. 19621128 198811 1 001

PERNYATAAN

Dengan ini saya menyatakan bahwa Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Malang, 17 Juli 2017

Robertus Romario Harvey K.

ABSTRAK

Kurniawan, Robertus Romario Harvey “Klasifikasi Daging Babi dan Sapi Menggunakan Ekstraksi Fitur Warna dan Tekstur Dengan Metode *Gray Level Co-Occurrence Matrix*” **Pembimbing: (1) Dr. Eng. Rosa Andrie Asmara, S.T., M.T., (2) Kadek Suarjuna Batubulan, S.Kom., M.T**

Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2017.

Di Indonesia harga daging sapi cukup mahal. Hal ini terjadi karena mata rantai distribusi yang sangat panjang dari peternak hingga ke tangan konsumen. Akibatnya diperlukan biaya produksi yang sangat tinggi. Dengan mahalnya daging sapi tersebut ada beberapa oknum yang berusaha untuk mencampur daging sapi dengan beberapa daging lainnya seperti daging babi. Kondisi ini sangat merugikan kalangan konsumen yang ingin menikmati daging sapi. Saat ini identifikasi daging dilakukan secara manual dengan menggunakan indra penglihatan. Cara ini memiliki banyak kelemahan bila para konsumen tidak jeli untuk membedakan daging sapi dan daging babi.

Perkembangan ilmu pengetahuan dan teknologi pengolahan citra digital memungkinkan untuk memilih daging sapi atau daging babi tersebut secara otomatis dengan bantuan aplikasi pengolahan citra. Penelitian klasifikasi daging babi dan sapi ini menggunakan metode *Backpropagation*. Fitur yang digunakan pada statistik orde pertama menggunakan *mean* dari warna RGB dan analisis tekstur menggunakan *Gray Level Co-Occurrence Matrix*. Tingkat keberhasilan klasifikasi daging babi dan sapi yang didapatkan menggunakan metode *Backpropagation* dengan tingkat akurasi 89,57%.

Kata Kunci: Daging, *Gray Level Co-Occurrence Matrix*, *Backpropagation*.

ABSTRACT

Kurniawan, Robertus Romario Harvey “Classification of Pork and Beef Using Extraction of Color and Texture Feature by Gray Level Co-Occurrence Matrix Method” **Counseling Lecturer: (1) Dr. Eng. Rosa Andrie Asmara, S.T., M.T., (2) Kadek Suarjuna Batubulan, S.Kom., M.T**

Thesis, Informatics Engineering Study Program, Department of Information Technology, State Polytechnic of Malang, 2017.

In Indonesia, beef meat sell with expensive price. It because of long distribution from the cattle rancher to the end customer. This causing a high-cost on the production and distribution. This condition made some people trying to cheat by mixing beef meat with another meat, such as pork. It occurs the financial loss for customers. Currently, the meat identification done manually using visual identification of human vision. This method has a lot of weaknesses to differ beef and pork due to fatigueness and need experienced persons.

The improvement of science and technology in image processing helps the customer to identify beefmeat and pork automatically by image processing application help. This research classification of beef meat and pork implements backpropagation method. color feature use first orde statistic mean RGB and texture analysis using Gray Level Co-Occurrence Matrix. Successfull result of classification of beef and pork obtained using Backpropagation method with 89,57% of accuracy result.

Keyword: Meat, Gray Level Co-Occurrence Matrix, Backpropagation.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan YME atas segala berkat dan anugerah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Klasifikasi Daging Babi dan Sapi Menggunakan Ekstraksi Fitur Warna dan Tekstur Dengan Metode *Gray Level Co-Occurrence Matrix*”. Tujuan dari penyusunan Skripsi ini sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Penulis menyadari tanpa adanya dukungan dan kerja sama dari berbagai pihak, penulisan skripsi ini tidak akan dapat berjalan baik. Untuk itu, penulis ingin menyampaikan rasa terima kasih kepada:

1. Keluarga tercinta Papa Rudi Hartono K., Mama Sri Susanti Indri P. dan adik tersayang Ivellin Karina Kurniawan serta seluruh keluarga yang selalu mendukung dan saling mendoakan.
2. Bapak Ir. Tundung Subali Padma, MT., selaku Direktur Politeknik Negeri Malang dimana tempat menuntut ilmu.
3. Bapak Rudi Ariyanto, ST., M.Cs., selaku Ketua Jurusan Teknologi Informasi.
4. Bapak Ir. Deddy Kusbianto P.A, MMKOM., selaku Ketua Program Studi Teknik Informatika.
5. Bapak Dr.Eng. Rosa Andrie Asmara, ST, MT., selaku Dosen Pembimbing I Jurusan Teknologi Informasi Prodi Teknik Informatika di Politeknik Negeri Malang.
6. Bapak Kadek Suarjuna Batubulan, S.Kom., M.T., selaku Dosen Pembimbing II Jurusan Teknologi Informasi Prodi Teknik Informatika di Politeknik Negeri Malang.
7. Bapak Farid Ferdiansyah dan Bapak Hari Santoso, selaku petugas dan pembimbing PD. Rumah Potong Hewan (RPH).
8. Seluruh jagal PD. Rumah Potong Hewan (RPH).
9. Dosen-dosen Politeknik Negeri Malang, khususnya di Jurusan Teknologi Informasi yang telah memberikan ilmu dan konsep.
10. Saudara-saudariku Dea Pradana Anindisa Widiyantoro, Naufal Ziyad Luthfiansyah, Ersah Rahmawati A., Vivid Ichtarosa Arinda, Aldio Putra C., dan

Ahmad Wali Satria yang telah memberikan semangat dalam proses penyelesaian skripsi ini.

11. Serta teman-teman Teknik Informatika angkatan 2013 dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan laporan skripsi dari awal hingga akhir.

Penulis menyadari bahwa dalam penyusunan skripsi ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, 17 Juli 2017

Penulis

DAFTAR ISI

SAMPUL DEPAN	i
HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	Error! Bookmark not defined.
PERNYATAAN.....	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Batasan Masalah	3
1.5 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 Daging.....	5
2.2 Tekstur	5
2.3 Perlemakan (<i>Marbling</i>).....	5
2.4 Pengolahan Citra.....	5
2.5 Citra Digital	6
2.6 Model Warna RGB	6
2.7 Proses Normalisasi RGB.....	7
2.8 <i>Grayscale Image</i>	7
2.9 <i>Gray Level Co-occurrence Matrix (GLCM)</i>	8
2.10 <i>Backpropagation</i>	10
2.11 <i>Microsoft Visual Studio</i>	13
BAB III METODOLOGI PENELITIAN.....	14
3.1 Rancangan Penelitian.....	14
3.2 Data.....	15
3.3 Metode Pengolahan Data	15
3.4 Proses <i>Training</i>	16
3.5 Proses <i>Testing</i>	16
3.6 Metode Pengujian	17
BAB IV ANALISIS DAN PERANCANGAN	18
4.1 Analisis Sistem	18
4.1.1 Kebutuhan Perangkat Lunak	18

4.1.2	Kebutuhan Perangkat Lunak	18
4.1.3	Gambaran Umum Aplikasi	18
4.2	Perancangan Sistem	19
4.2.1	Alur Sistem Klasifikasi Daging	19
4.2.2	Perhitungan Sistem	33
4.3	Perancangan <i>Interface</i>	39
4.3.1	Rancangan Tampilan Menu Utama.....	39
4.3.2	Rancangan Tampilan <i>Training</i>	39
4.3.3	Tampilan <i>View Bobot Final V</i> Pada <i>Training</i>	41
4.3.4	Tampilan <i>View Bobot Final W</i> Pada <i>Training</i>	41
4.3.5	Rancangan Tampilan <i>Testing</i>	41
4.3.6	Tampilan <i>Help</i> Pada Form <i>Training</i>	42
4.3.7	Tampilan <i>Help</i> Pada Form <i>Testing</i>	43
BAB V	IMPLEMENTASI.....	44
5.1	Implementasi <i>Home</i>	44
5.2	Implementasi <i>Training</i>	44
5.2.1	Proses <i>Input Citra</i>	45
5.2.2	<i>Pre-Processing</i>	46
5.2.3	Proses Ekstraksi Fitur (GLCM)	47
5.2.4	Proses Penyimpanan Data Set	48
5.2.5	Proses <i>Training Backpropagation</i>	49
5.3	Implementasi <i>Testing</i>	57
5.3.1	Proses <i>Input Citra</i>	57
5.3.2	Proses Ekstraksi Tekstur	59
5.3.3	Proses Memasukan Data Bobot Hasil <i>Training</i>	59
5.3.4	Proses Klasifikasi Daging	60
5.4	Tampilan <i>Help</i>	60
5.4.1	Tampilan <i>Help Training</i>	61
5.4.2	Tampilan <i>Help Testing</i>	61
BAB VI	PENGUJIAN DAN PEMBAHASAN	62
6.1	Pengujian	62
6.1.1	Pengujian Sistem.....	62
6.1.2	Pengujian Akurasi	71
6.2	Pembahasan	77
BAB VII	PENUTUP	80
7.1	Kesimpulan	80
7.2	Saran	80
DAFTAR PUSTAKA	81
LAMPIRAN	84

DAFTAR GAMBAR

Gambar 2.1 Warna RGB	7
Gambar 2.2 Hasil Konversi RGB - <i>Grayscale Image</i>	8
Gambar 2.3 Piksel Bertetangga Dalam Empat Arah	8
Gambar 2.4 Arsitektur Jaringan Syaraf Tiruan <i>Backpropagation</i>	11
Gambar 3.1 Diagram <i>Waterfall</i>	14
Gambar 3.2 Diagram <i>Block</i> Proses Klasifikasi Daging Babi dan Sapi	16
Gambar 4.1 Alur Proses <i>Image Input</i>	19
Gambar 4.2 Alur Proses Ekstraksi Ciri Warna	20
Gambar 4.3 Alur Proses Ekstraksi Tekstur	21
Gambar 4.4 Citra Yang Diambil Dengan <i>Pixel 4x4</i>	21
Gambar 4.5 Gambaran Umum Proses <i>Training</i>	26
Gambar 4.6 Gambaran Umum Proses <i>Testing</i>	26
Gambar 4.6 Rancangan Tampilan <i>Home</i>	39
Gambar 4.7 Rancangan Tampilan <i>Training</i>	40
Gambar 4.8 Rancangan Tampilan <i>View Bobot Final V</i>	41
Gambar 4.9 Rancangan Tampilan <i>View Bobot Final W</i>	41
Gambar 4.10 Rancangan Tampilan <i>Testing</i>	42
Gambar 4.11 Rancangan Tampilan <i>Help Form Training</i>	42
Gambar 4.12 Rancangan Tampilan <i>Help Form Testing</i>	43
Gambar 5.1 Tampilan Awal	44
Gambar 5.2 Tampilan <i>Training</i>	44
Gambar 5.3 Tampilan Proses <i>Input Citra</i>	45
Gambar 5.4 Tampilan <i>File Open Dialog</i>	45
Gambar 5.5 Tampilan <i>Training</i> Saat <i>Input Citra</i>	46
Gambar 5.6 Tampilan Hasil <i>Cropping</i>	46
Gambar 5.7 Tampilan Ekstraksi Warna RGB	47
Gambar 5.8 Tampilan Hasil Konversi Warna	47
Gambar 5.9 Tampilan Hasil Dari Proses GLCM	48
Gambar 5.10 Tampilan <i>Save Data</i>	48
Gambar 5.11 Tampilan Kotak Peringatan	49
Gambar 5.12 Tampilan Data Dalam DGV	49
Gambar 5.13 Data Daging yang Dimasukan ke Dalam XML	50
Gambar 5.14 Tampilan Bobot <i>Final V</i>	56
Gambar 5.15 Tampilan Bobot <i>Final W</i>	56
Gambar 5.16 Tampilan <i>Save Training</i>	57
Gambar 5.17 Tampilan <i>Testing</i>	57
Gambar 5.18 Tampilan Proses <i>Input Citra</i>	58
Gambar 5.19 Tampilan <i>File Open Dialog</i>	58
Gambar 5.20 Tampilan <i>Input Citra</i>	58
Gambar 5.21 Tampilan Konversi Warna	59
Gambar 5.22 Tampilan Ekstraksi Tekstur GLCM	59
Gambar 5.23 Tampilan Proses <i>Input Data</i>	59
Gambar 5.24 Tampilan <i>File Open Dialog</i>	60
Gambar 5.25 Tampilan <i>Input Data</i>	60
Gambar 5.26 Tampilan Hasil <i>Testing</i>	60
Gambar 5.27 Window <i>Help Training</i>	61

Gambar 5.28 Window <i>Help Testing</i>	61
Gambar 6.1 Diagram Batang Berdasarkan <i>Learning Rate</i>	78
Gambar 6.2 Diagram Batang Berdasarkan Kecepatan Waktu	78
Gambar 6.3 Diagram Batang Berdasarkan <i>Epoch</i>	79

DAFTAR TABEL

Tabel 4.1 Citra Asli	21
Tabel 4.2 Komposisi Piksel	21
Tabel 4.3 <i>Framework Matrix</i>	22
Tabel 4.4 <i>Transpose Matrix</i>	22
Tabel 4.5 Hasil Penjumlahan (<i>Simetris Matrix</i>).....	22
Tabel 4.6 <i>Normalization Matrix</i>	22
Tabel 4.7 <i>GLCM Matrix</i>	22
Tabel 4.8 Perhitungan Matrik Energi	22
Tabel 4.9 <i>Energy Matrix</i>	22
Tabel 4.10 Perhitungan Matrik Entropi	23
Tabel 4.11 <i>Entropy Matrix</i>	23
Tabel 4.12 Perhitungan Matrik Kontras	23
Tabel 4.13 <i>Contrast Matrix</i>	23
Tabel 4.14 Perhitungan Matrik Homogenitas	24
Tabel 4.15 <i>Homogeneity Matrix</i>	24
Tabel 4.16 Perhitungan Matrik Korelasi	24
Tabel 4.17 <i>Correlation Matrix</i>	25
Tabel 4.18 Perhitungan Matrik Momentum Selisih Invers	25
Tabel 4.19 <i>Inverse Difference Momentum Matrix</i>	25
Tabel 4.20 Sampel Data Ekstraksi Fitur	33
Tabel 4.21 Nilai Bobot	34
Tabel 5.1 Data <i>Training</i>	50
Tabel 6.1 Pengujian <i>Home</i>	62
Tabel 6.2 Pengujian Sistem <i>Training</i>	63
Tabel 6.3 Pengujian Sistem <i>Testing</i>	68
Tabel 6.4 Pengujian Daging Dengan <i>Learning Rate</i> 0,1 dan <i>Epoch</i> 500	71
Tabel 6.5 Pengujian Daging Dengan <i>Learning Rate</i> 0,1 dan <i>Epoch</i> 1000	73
Tabel 6.6 Pengujian Daging Dengan <i>Learning Rate</i> 0,1 dan <i>Epoch</i> 3000	74
Tabel 6.7 Pengujian Daging Dengan <i>Learning Rate</i> 0,1 dan <i>Epoch</i> 5000	74
Tabel 6.8 Pengujian Daging Dengan <i>Learning Rate</i> 0,2 dan <i>Epoch</i> 500	75
Tabel 6.9 Pengujian Daging Dengan <i>Learning Rate</i> 0,2 dan <i>Epoch</i> 1000	75
Tabel 6.10 Pengujian Daging Dengan <i>Learning Rate</i> 0,2 dan <i>Epoch</i> 3000	76
Tabel 6.11 Pengujian Daging Dengan <i>Learning Rate</i> 0,2 dan <i>Epoch</i> 5000	77

DAFTAR LAMPIRAN

- Lampiran 1: Listing Program
- Lampiran 2: Hasil *Training*
- Lampiran 3: Surat Penelitian
- Lampiran 4 : Lembar Validasi
- Lampiran 5: Lembar Bimbingan Skripsi
- Lampiran 6: Lembar Persetujuan Maju Sidang Skripsi
- Lampiran 7: Lembar Revisi Penguji
- Lampiran 8: Lembar Verifikasi Tata Tulis dan *Abstract*
- Lampiran 9: Biodata Penulis

BAB I

PENDAHULUAN

1.1 Latar Belakang

Daging merupakan bagian lunak pada hewan yang terbungkus oleh kulit dan melekat pada tulang yang menjadi bahan makanan. Daging semacam ini disebut “daging merah”, dan diperdagangkan dalam bentuk potongan-potongan^[1]. Salah satu contoh daging merah adalah daging sapi. Daging sapi merupakan salah satu bahan pangan yang digemari oleh masyarakat kalangan menengah ke atas maupun kebawah, karena kandungan gizi dari daging sapi yang cukup lengkap^[2], membuatnya digemari oleh berbagai kalangan.

Di Indonesia harga daging sapi cukup mahal, hal ini terjadi karena mata rantai distribusi yang sangat panjang dari peternak hingga ke tangan konsumen^[3], sehingga diperlukan biaya yang sangat tinggi untuk membeli daging sapi. Dengan mahalnya daging sapi tersebut ada beberapa oknum yang berusaha untuk mencampur daging sapi dengan beberapa daging lainnya seperti daging babi, kondisi ini sangat merugikan kalangan konsumen yang ingin menikmati daging sapi. Saat ini identifikasi daging dilakukan secara manual dengan menggunakan indra penglihatan. Cara ini memiliki banyak kelemahan bila para konsumen tidak jeli untuk membedakan daging sapi dan daging babi. Selain kalangan konsumen ada juga sebuah instansi atau supermarket yang melakukan penyortiran daging secara manual dari pengepul daging. Hal ini juga dapat merugikan supermarket karena kualitas daging tidak terjamin

Berdasarkan penelitian terdahulu oleh Kiswanto (2012, Universitas Diponegoro Semarang) mengenai identifikasi jenis daging sapi yang menggunakan *Transformasi Wavelet Haar* menghasilkan akurasi sebesar 54% pada *multi-query* dan 33,32% pada *tunggal-query*^[4]. Selain itu juga berdasarkan penelitian Elvia Budianita, dkk (2015, UIN Sultan Syarif Kasim Riau) mengenai pembeda daging sapi dan babi menggunakan metode HSV GLCM dan klasifikasi *K-Nearest Neighbour* (K-NN) menghasilkan akurasi sebesar 88,75% untuk pengujian tanpa *background* dan 73,375% untuk pengujian menggunakan *background*^[5]. Klasifikasi kelas daging menggunakan yang dilakukan oleh Laksono Tri Wibowo (2009, Universitas Diponegoro) menggunakan metode GLCM dan klasifikasi

K-Nearest Neighbour (K-NN) menghasilkan akurasi 100% untuk citra daging kerbau dan 20% untuk citra daging kambing^[6]. Skripsi mengenai Identifikasi Citra Daging Sapi dan Daging Babi oleh Mas'ud (2011, Universitas Dian Nuswantoro) menggunakan metode *Principal Component Analysis* (PCA) dan jarak *Euclidean*. Menghasilkan akurasi tertinggi mencapai 100%^[7]. Metode klasifikasi *K-Nearest Neighbour* untuk citra daging sapi dan babi oleh Winda Rizky Astuti (2016, Universitas Islam Negeri Sunan Kalijaga Yogyakarta) menghasilkan akurasi sebesar 71,8%^[8]. Untuk pengaplikasian di dalam *smartphone* menggunakan jaringan syaraf tiruan model *Multi Layer Perceptron* sebagai metode pengenalnya oleh Wasfi Muhammadiy (2016, Universitas Sumatra Utara) menghasilkan akurasi sebesar 73,34%^[9]. Penelitian serta skripsi di atas adalah beberapa contoh perhitungan suatu objek dengan menggunakan *image processing*.

Dalam penelitian ini penulis mengangkat sebuah judul yang dapat membedakan daging sapi dan daging babi, salah satunya dengan menggunakan *image processing*. *Image processing* dapat dijadikan alternatif karena metode ini tidak merusak objek (*nondestructive*), dengan cara menganalisa warna dan tekstur menggunakan metode *Gray Level Co-Occurrence Matrix* (GLCM) dan klasifikasi *Backpropagation*. Sehingga dapat memudahkan para konsumen dan sebuah institusi atau *supermarket* agar mudah melakukan penyortiran daging.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang diuraikan, maka dapat dirumuskan masalah sebagai berikut:

Bagaimana cara membedakan daging babi dan sapi berdasarkan warna dan tekstur tanpa menyentuh objek.

1.3 Tujuan

Tujuan pembuatan sistem ini adalah:

Membuat suatu sistem yang dapat mengidentifikasi daging babi dan sapi dalam membantu masyarakat umum serta proses penyortiran dengan memanfaatkan teknologi pengelolaan citra digital.

1.4 Batasan Masalah

Agar pembahasan lebih terarah dan tidak meluas, maka terdapat batasan-batasan pembahasan masalah, yaitu:

- Daging sapi dan daging babi yang digunakan merupakan daging yang segar.
- Pengambilan sampel daging menggunakan daging babi dan daging sapi.
- Citra daging berada pada bagian tengah.
- Jarak untuk pengambilan citra berjarak 5-7cm
- Pencahayaan yang digunakan merupakan pencahayaan di luar ruangan dan pencahayaan di dalam ruangan yang terang
- Parameter yang digunakan berdasarkan warna dan tekstur / perlemakan (*marbling*).
- Membuat sistem aplikasi yang dapat mendeteksi warna dan tekstur / perlemakan (*marbling*) daging menggunakan metode *Gray Level Co-Occurrence Matrix* (GLCM) dan klasifikasi *Backpropagation*.

1.5 Sistematika Penulisan

BAB I. PENDAHULUAN

Pada bab ini dijelaskan mengenai hal-hal yang bersifat umum seperti latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, dan sistematika penulisan, dari penulisan skripsi.

BAB II. LANDASAN TEORI

Pada bab ini berisi teori-teori yang mendasari dan berkaitan dengan masalah perencanaan dan pembuatan aplikasi yang digunakan untuk memudahkan pemahaman dan pemecahan terhadap masalah yang ada.

BAB III. METODOLOGI PENELITIAN

Pada bab ini menjelaskan langkah-langkah perencanaan penerapan pengolahan citra pada klasifikasi daging berdasarkan warna dan tekstur menggunakan metode *Gray Level Co-Occurrence Matrix* (GLCM) dengan klasifikasi *Backpropagation*.

BAB IV. ANALISIS DAN PERANCANGAN

Pada bab ini menjelaskan langkah-langkah perencanaan penerapan pengolahan citra pada klasifikasi daging berdasarkan warna dan tekstur

menggunakan metode *Gray Level Co-Occurrence Matrix* (GLCM) dengan klasifikasi *Backpropagation*.

BAB V. IMPLEMENTASI

Pada bab ini membahas mengenai pembuatan aplikasi yang telah didesain pada tahap sebelumnya ke dalam bahasa pemrograman. Disertai dengan gambaran desain, dan *interface* aplikasi.

BAB VI. PENGUJIAN DAN PEMBAHASAN

Pada bab ini membahas mengenai hasil uji coba dari aplikasi yang telah dibuat secara keseluruhan. Mulai dari pengujian sistem sampai penerimaan pengguna.

BAB VII. PENUTUP

Pada bab ini membahas mengenai kesimpulan yang diperoleh dengan melihat hasil pengujian aplikasi yang telah dibuat dan saran yang dapat digunakan untuk kepentingan pengembangan selanjutnya.

BAB II

LANDASAN TEORI

2.1 Daging

Daging adalah hasil ternak yang dikonsumsi sebagian besar oleh masyarakat Indonesia. Daging mengandung banyak manfaat bagi kesehatan tubuh, antara lain berupa kalori, protein, zat besi, vitamin B kompleks, omega 3, dll. Daging sapi yang lezat dikonsumsi yaitu bagian *tenderloin* atau bagian has dalam.

Daging bagian ini adalah potongan daging yang berasal dari sekitar pinggang dan bahu sapi. Daging ini tergolong daging yang lunak karena bagian ini jarang sekali digunakan untuk bergerak^[10]. Komposisi dari daging itu sendiri meliputi 75% air, 19% protein, 3.5% substansi *non* protein yang larut, dan 2.5% lemak.

2.2 Tekstur

Tekstur adalah karakteristik yang dimiliki suatu permukaan karena adanya struktur 3 dimensi dan merupakan unsur rupa sebuah permukaan. Salah satu macam tekstur yang ada yaitu tekstur daging. Tekstur daging merupakan suatu fungsi ukuran dari berkas-berkas serat ke dalam mana septa perimisium dari tenunan pengikat membagi-bagi urat daging secara *longitudinal*^[11]. Tekstur daging yang baik akan terlihat sangat kenyal dan bila di tekan akan kembali seperti bentuk semula.

2.3 Perlemakan (*Marbling*)

Marbling merupakan garis-garis tipis atau bintik-bintik lemak putih yang ada pada potongan daging. *Marbling* dapat dipengaruhi oleh berbagai faktor, antara lain dari pola makan, kondisi lingkungan ternak, serta genetika dari ternak tersebut. Untuk mendapatkan hasil *marbling* yang terbaik, hewan ternak dapat diberi makanan yang kaya akan nutrisi, selain itu hewan ternak yang dibesarkan dalam kondisi ideal sejak lahir juga akan mempengaruhi *marbling* yang diperoleh. Lemak daging yang berasal dari sapi muda akan berwarna putih kekuningan, sedangkan lemak yang berasal dari sapi tua akan berwarna kekuningan.

2.4 Pengolahan Citra

Pengolahan citra atau *image processing* adalah suatu bentuk pengolahan atau pemrosesan sinyal dengan *input* berupa gambar (*image*) dan ditransformasikan

menjadi gambar lain sebagai keluarannya dengan teknik tertentu. Citra keluaran memiliki kualitas lebih baik daripada citra masukan^[12]. Misal citra warnanya kurang tajam, kabur (*blurring*), mengandung *noise* (bintik-bintik putih), dll sehingga perlu ada pemrosesan untuk memperbaiki citra karena citra tersebut menjadi sulit diinterpretasikan karena informasi yang disampaikan menjadi berkurang.

Pengolahan citra dilakukan untuk memperbaiki kesalahan data sinyal gambar yang terjadi akibat transmisi dan selama akuisisi sinyal, serta untuk meningkatkan kualitas penampakan gambar agar lebih mudah diinterpretasi oleh sistem penglihatan manusia baik dengan melakukan manipulasi dan juga penganalisisan terhadap gambar^[13].

2.5 Citra Digital

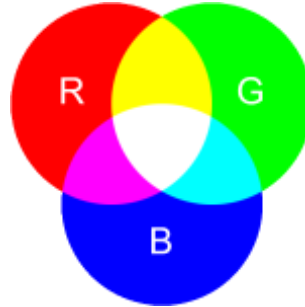
Citra digital adalah representasi dari sebuah citra dua dimensi sebagai sebuah kumpulan nilai digital yang disebut elemen gambar atau piksel. Piksel adalah elemen terkecil yang menyusun citra dan mengandung nilai yang mewakili kecerahan dari sebuah warna pada sebuah titik tertentu.

Umumnya citra digital berbentuk persegi panjang atau bujur sangkar (pada beberapa sistem pencitraan ada pula yang berbentuk segienam) yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya piksel sehingga ukuran citra selalu bernilai bulat. Setiap piksel memiliki koordinat sesuai posisinya dalam citra. Koordinat ini biasanya dinyatakan dalam bilangan bulat positif, yang dapat dimulai dari 0 atau 1 tergantung pada sistem yang digunakan. Setiap piksel juga memiliki nilai berupa angka digital yang merepresentasikan informasi yang diwakili oleh piksel tersebut. Format data citra digital berhubungan erat dengan warna. Pada kebanyakan kasus, terutama untuk keperluan penampakan secara visual, nilai data digital merepresentasikan warna dari citra yang diolah. Format citra digital yang banyak dipakai adalah citra biner (*monokrom*), citra skala keabuan (*grayscale*), Citra warna (*true color*), dan citra warna berindeks^[14].

2.6 Model Warna RGB

Warna merupakan unsur yang sangat penting dalam desain grafis^[15]. Warna merupakan alat pembeda yang dapat dipahami oleh siapaun untuk menentukan sebuah objek. Model Warna RGB (*Red, Green, Blue*) adalah sebuah model warna

aditif dimana pancaran warna *red* (merah), *green* (hijau), dan *blue* (biru) ditambahkan bersama dengan cara yang bervariasi untuk mereproduksi susunan warna yang lebar. RGB memiliki 8-bit *per channel*, maka memiliki 16,777,216 warna (256x256x256). Penggabungan dua warna utama RGB akan menghasilkan warna CMY. Biru-Hijau = *Cyan*, Biru-Merah = *Magenta*, Merah-Hijau = *Yellow*



Gambar 2.1 Warna RGB

2.7 Proses Normalisasi RGB

Normalisasi RGB dilakukan untuk menyetarakan nilai RGB (*Red*, *Green*, *Blue*) yaitu 0-1. Untuk itu diperlukan rumus sebagai berikut^[16]:

$$R = \frac{R}{R+G+B} \quad (2.1)$$

$$G = \frac{G}{R+G+B} \quad (2.2)$$

$$B = \frac{B}{R+G+B} \quad (2.3)$$

2.8 Grayscale Image

Grayscale Image menggunakan 256 skala abu-abu yang berbeda. Pada setiap piksel yang menyusun memiliki nilai kecerahan cahaya dalam selang 0 (hitam) sampai dengan 255 (putih). Banyak kemungkinan nilai minimum dan maksimum tergantung pada jumlah bit yang digunakan yang pada umumnya menggunakan 8-bit^[17].

Terdapat 3 metode untuk merubah RGB menjadi *Grayscale Image*:

- *Lightness*

$$Lightness = \frac{(\max[R,G,B] + \min[R,G,B])}{2} \quad (2.4)$$

- *Average*

$$Average = \frac{(R+G+B)}{3} \quad (2.5)$$

- *Luminosity*

$$Luminosity = (0.21 \times R) + (0.72 \times G) + (0.07 \times B) \quad (2.6)$$

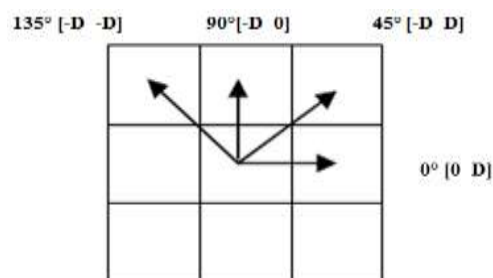


Gambar 2.2 Hasil Konversi RGB - *Grayscale Image*

2.9 *Gray Level Co-occurrence Matrix (GLCM)*

Matriks ko-okurensi merupakan salah satu metode statistik yang dapat digunakan untuk menganalisis tekstur^[18]. Matriks ko-okurensi dibentuk dari suatu citra dengan melihat pada piksel-piksel yang berpasangan yang memiliki intensitas tertentu. Penggunaan metode ini berdasarkan pada hipotesis bahwa dalam suatu tekstur akan terjadi perulangan konfigurasi atau pasangan aras keabuan. Misal, d didefinisikan sebagai jarak antara dua posisi piksel, yaitu (x_1, y_1) dan (x_2, y_2) ; dan θ didefinisikan sebagai sudut diantara keduanya. Orientasi sudut θ terdiri atas empat arah sudut dengan interval sudut 45° , yaitu 0° , 45° , 90° , dan 135°

Maka matriks ko-okurensi didefinisikan sebagai matriks yang menyatakan distribusi spasial antara dua piksel yang bertetangga yang memiliki intensitas i dan j , yang memiliki jarak d diantara keduanya, dan sudut θ diantara keduanya. Matriks ko-okurensi dinyatakan dengan $P_{d,\theta}(i,j)$. Suatu piksel yang bertetangga yang memiliki jarak d diantara keduanya, dapat terletak di empat arah yang berlainan, hal ini ditunjukkan di bawah ini.



Gambar 2.3 Piksel Bertetangga Dalam Empat Arah

Dalam matriks ko-okurensi, terdapat ciri tekstur yang dapat diperoleh dari suatu citra yang digunakan sebagai pembeda antara citra dengan kelas tertentu, dengan kelas lainnya. Ciri-ciri tersebut adalah:

- Energi (*Energy*)

Energi merupakan salah satu fitur GLCM yang digunakan untuk mengukur konsentrasi pasangan intensitas pada matriks GLCM^[19], atau sering disebut *angular second moment*^[20]. Untuk mendapatkan nilai energi didefinisikan pada *equation 2.7*.

$$f_1 = \sum_i \sum_j P_d^2(i, j) \quad (2.7)$$

- Entropi (*Entropy*)

Menunjukkan ukuran ketidakaturan bentuk^[19] Mengukur kompleksitas (keacakan) citra. Entropi akan bernilai tinggi ketika citra tidak seragam^[20]. Untuk mendapatkan nilai entropi didefinisikan pada *equation 2.8*.

$$f_2 = - \sum_i \sum_j P_d(i, j) \log(P_d(i, j)) \quad (2.8)$$

- Kontras (*Contrast*)

Kontras digunakan untuk mengukur frekuensi spasial dari citra dan perbedaan momen GLCM. Perbedaan yang dimaksudkan adalah perbedaan tinggi dan rendahnya piksel^[19]. Kontras akan bernilai 0 jika piksel ketetanggaan mempunyai nilai yang sama^[20]. Jika terletak jauh dari diagonal utama maka nilai kekontrasan besar. Untuk mendapatkan nilai kontras didefinisikan pada *equation 2.9*.

$$f_3 = \sum_i \sum_j (i - j)^2 P_d(i, j) \quad (2.9)$$

- Homogenitas (*Homogeneity*)

Menunjukkan kehomogenan variasi intensitas dalam citra^[19] berderajat kabuan sejenis. Citra homogen akan memiliki *homogeneity* yang besar. Mengukur homogenitas. Nilai ini sangat sensitif terhadap nilai disekitar diagonal utama. Bernilai tinggi ketika semua piksel mempunyai nilai yang sama / seragam. Kebalikan dari kontras yaitu akan bernilai besar jika mempunyai nilai piksel yang sama pada saat energi bernilai tetap^[20]. Untuk mendapatkan nilai homogenitas didefinisikan pada *equation 2.10*.

$$f_4 = \sum_i \sum_j \frac{P_d(i, j)}{1 + |i - j|} \quad (2.10)$$

- Korelasi (*Correlation*)

Mengukur linearitas (*the joint probability*) dari sejumlah pasangan piksel (*pairs*)^[20]. Untuk mendapatkan nilai korelasi didefinisikan pada *equation 2.11*.

$$f_5 = \sum_i \sum_j \frac{(i - \mu_x)(j - \mu_y)P_d(i, j)}{\sigma_x \sigma_y} \quad (2.11)$$

dimana:

$$\begin{aligned} \mu_x &= \sum_i i \sum_j P_d(i, j) & \sigma_y^2 &= \sum_i (i - \mu_y)^2 \sum_j P_d(i, j) \\ \mu_y &= \sum_i j \sum_j P_d(i, j) \\ \sigma_x^2 &= \sum_i (i - \mu_x)^2 \sum_j P_d(i, j) \end{aligned}$$

- Momentum Selisih Invers (*Inverse Difference Momentum*)

Merupakan homogenitas lokal. Jumlahnya akan semakin tinggi jika level keabuan lokalnya seragam dan invers GLCM-nya tinggi^[21]. Untuk mendapatkan nilai momentum selisih didefinisikan pada *equation 2.12*.

$$f_6 = \sum_i \sum_j \frac{1}{1 + (i - j)^2} P_d(i, j) \quad (2.12)$$

2.10 Backpropagation

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan *neuron-neuron* yang ada pada lapisan tersembunyi.

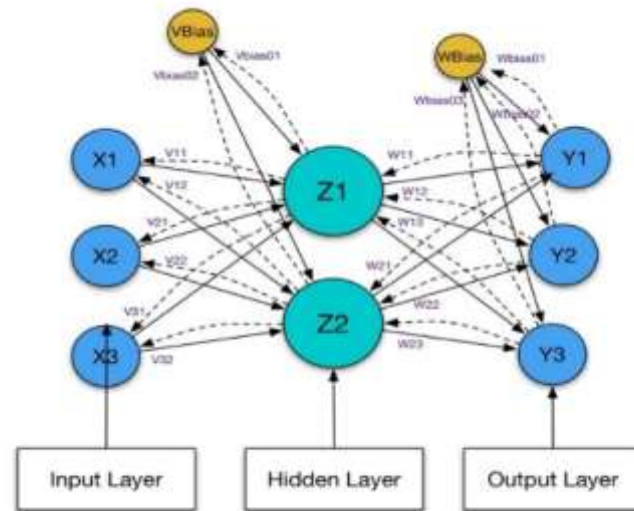
Terdapat tiga fase di dalam algoritma *backpropagation*, yaitu fase propagasi maju yang merupakan fase dimana sinyal-sinyal masukan akan dipropagasikan kepada *layer* di atasnya (*hidden layer* lalu *output layer*) menggunakan fungsi aktivasi yang telah ditentukan. Setelah itu fase yang kedua yaitu propagasi mundur untuk mengetahui kesalahan dari jaringan tersebut. Fase yang ketiga adalah fase modifikasi bobot yang digunakan untuk menghitung perubahan bobot yang terjadi di dalam jaringan^[22]. Saat tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu untuk mendapatkan nilai *error* tersebut. Saat perambatan maju *neuron-neuron* diaktifkan dengan menggunakan fungsi aktivasi *sigmoid biner*

Algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai-nilai bobotnya dalam arah mundur (*backward*). Tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu untuk mendapatkan nilai

error tersebut. Saat perambatan maju *neuron-neuron* diaktifkan dengan menggunakan fungsi aktivasi *sigmoid biner* seperti *equation 2.13*

$$F(x) = \frac{1}{1+e^{-x}} \quad (2.13)$$

Arsitektur jaringan syaraf *backpropagation* seperti terlihat pada gambar dibawah ini:



Gambar 2.4 Arsitektur Jaringan Syaraf Tiruan *Backpropagation*

Keterangan:

x1 s/d xn: *input layer*

y1 s/d ym: *output layer*

z1 s/d zp: *hidden layer*

Dari diagram diatas dapat diketahui X merupakan *input*-an dari jaringan, Z merupakan *hidden layer*, Y merupakan *output layer*, V merupakan bobot dari *neuron* penghubung antara masukan dengan *hidden layer*, VBias merupakan bobot bias untuk *neuron* yang menuju pada *hidden layer*, W merupakan bobot dari *neuron* penghubung antara *hidden layer* dengan *output layer*, serta WBias merupakan bobot bias untuk *neuron* yang menuju *output layer*. Berikut merupakan rumus dasar dari algoritma *backpropagation*^[23]:

- *Feedforward* (Fase Maju)

- Setiap unit *input* (X_i , $i=1,2,3,\dots,n$) menerima sinyal X_i dan meneruskan sinyal tersebut pada lapisan yang ada diatasnya (*hidden layer*).
- Setiap unit tersembunyi (Z_i , $i=1,2,3,\dots,p$) menjumlahkan sinyal-sinyal *input* terbobot:

$$Z_{in_j} = V0_j + \sum_{i=1}^n X_i V_{ij} \quad (2.14)$$

selanjutnya digunakan fungsi aktivasi untuk menghitung sinyal *output*-nya:

$$Z_j = f(Z_{in_j}) \quad (2.15)$$

lalu dikirim sinyal tersebut ke semua unit dilapisan atasnya (unit-unit *output*).

- c) Setiap unit *output* ($Y_k, k=1,2,3,\dots,m$) menjumlahkan sinyal-sinyal *input* terbobot

$$Y_{in_k} = W0_k + \sum_{i=1}^p Z_i W_{jk} \quad (2.16)$$

menggunakan fungsi aktivasi untuk menghitung sinyal *output*-nya.

$$Y_k = f(Y_{in_k}) \quad (2.17)$$

lalu dikirim sinyal tersebut ke semua unit dilapisan atasnya (unit-unit *output*).

- *Backpropagation* (Fase Mundur)

- a) Setiap unit *output* ($Y_k, k=1,2,3,\dots,m$) menerima target pola yang berhubungan dengan pola *input* pembelajaran. Lalu akan dihitung informasi *error*-nya.

$$\delta_k = (t_k - y_k) f'(Y_{in_k}) \quad (2.18)$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki W_{jk})

$$\Delta W_{jk} = \alpha \delta_k Z_j \quad (2.19)$$

lalu dihitung koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai W_{0k}) :

$$\Delta W_{0k} = \alpha \delta_k \quad (2.20)$$

kirikan δ_k ini ke unit-unit yang ada dilapisan bawahnya.

- b) Setiap unit tersembunyi ($Z_j, j=1,2,3,\dots,p$) menjumlahkan *delta input*-nya (dari unit-unit yang berada pada lapisan diatasnya):

$$\delta_{in_j} = \sum_{k=1}^m \delta_k W_{jk} \quad (2.21)$$

lalu kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*:

$$\delta_j = \delta_{in_j} f'(Z_{in_j}) \quad (2.23)$$

kemudian dihitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai V_{ij}):

$$\Delta V_{jk} = \alpha \delta_j X_i \quad (2.24)$$

lalu hitung koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai V_{0j}):

$$\Delta V_{0j} = \alpha \delta_j. \quad (2.25)$$

setiap unit *output* (Y_k , $k=1,2,3,\dots,m$) memperbaiki bias dan bobotnya ($j=0,1,2,3,\dots,p$).

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (2.26)$$

setiap unit tersembunyi (z_j , $j=1,2,3,\dots,p$) memperbaiki bias dan bobotnya ($i=0,1,2,3,\dots,n$)

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \quad (2.27)$$

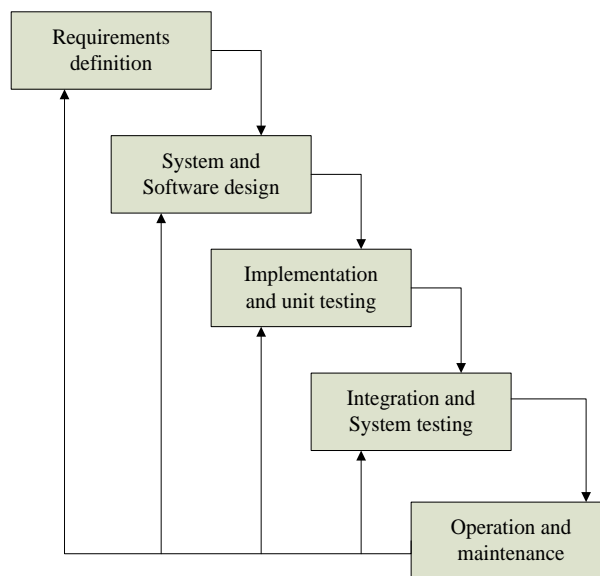
2.11 Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak yang dapat digunakan untuk melakukan pengembangan aplikasi *multi-platform* baik untuk sistem operasi *windows*, *mobile (android, iOS)*. *Visual Studio* mencakup *compiler*, *Integrated Development Environment (IDE)* dan dokumentasi. *Compiler* yang dimasukkan di dalam *visual studio* antara lain *Visual C++*, *Visual C#*, *Visual Basic*, *.NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*.

BAB III METODOLOGI PENELITIAN

3.1 Rancangan Penelitian

Metode yang digunakan dalam pembuatan perangkat lunak ini adalah metode *Waterfall*. Menurut Sommerville (2011:29) *Waterfall* model mengambil kegiatan proses dasar spesifikasi, pengembangan, validasi, dan evolusi dan mewakili kegiatan tersebut sebagai fase terpisah seperti spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian dan sebagainya. Adapun tahap dalam metode *waterfall* ini adalah sebagai berikut:



Gambar 3.1 Diagram *Waterfall*

Keterangan:

- *Requirements Definition*

Tahap ini merupakan tahap awal dalam pembentukan aplikasi untuk mengimplementasikan klasifikasi daging babi dan sapi berdasarkan warna dan tekstur dengan menggunakan *Backpropagation*. dengan menetapkan berbagai kebutuhan yang diperlukan seperti konsultasi dengan pengguna aplikasi dan menetapkan kebutuhan tersebut secara rinci yang berfungsi sebagai spesifikasi aplikasi.

- *Sistem and Software Design*

Merupakan tahap proses desain aplikasi dalam mengimplementasikan klasifikasi daging babi dan sapi berdasarkan warna dan tekstur dengan menggunakan *Backpropagation Implementation and Unit Testing*

Pada tahap ini merupakan tahapan dimana desain sistem akan diterjemahkan kedalam kode program dengan menggunakan bahasa pemrograman yang sudah ditentukan. Pada tahap ini dilakukan pengujian terhadap aplikasi klasifikasi daging babi dan sapi berdasarkan warna dan tekstur dengan menggunakan *Backpropagation*.

- *Integration and Sistem Testing*

Pada tahap ini merupakan tahap penyatuan unit-unit program aplikasi klasifikasi daging babi dan sapi berdasarkan warna dan tekstur dengan menggunakan *Backpropagation* dan kemudian akan dilakukan pengujian secara keseluruhan pada aplikasi tersebut.

- *Operation and Maintenance*

Tahap terakhir ini dilakukan jika seluruh proses tahapan di atas telah dilakukan, mengoperasikan aplikasi dan melakukan pemeliharaan seperti penyesuaian dan perubahan yang dibutuhkan.

3.2 Data

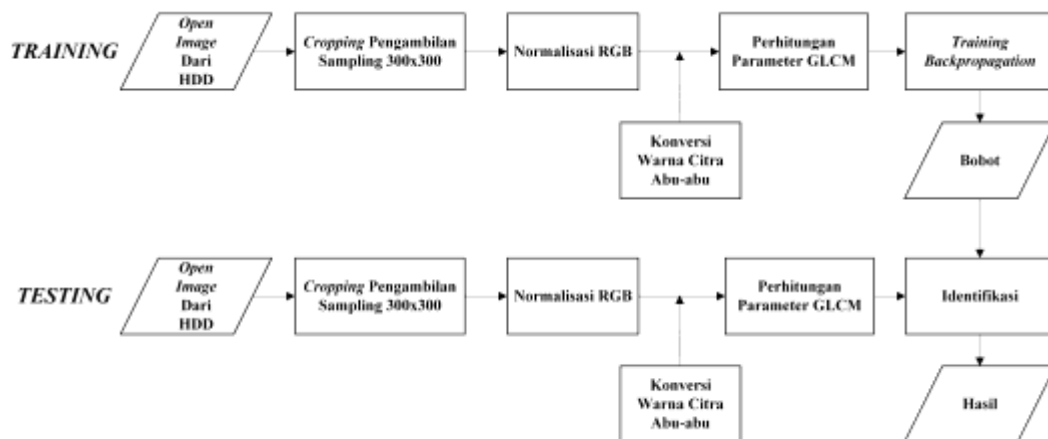
Komponen yang diperlukan untuk melakukan penelitian selanjutnya yaitu adalah data. Data yang diperlukan sebagai objek pengujian klasifikasi daging yaitu adalah daging babi dan daging sapi yang memiliki komposisi warna dan tekstur berbeda. Serta *image* atau gambar yang didapat dari data sampel daging.

3.3 Metode Pengolahan Data

Metode pengolahan data merupakan kerangka penelitian yang telah dikemukakan oleh penulis pada pendahuluan maka penelitian ini menggunakan kerangka sebagai berikut:

Pada gambar 3.2 di bawah merupakan proses *training* dan *testing* yang terdapat pada sistem. Dikarenakan pada sistem klasifikasi daging sapi dan babi berdasarkan perbedaan warna dan tekstur dengan menggunakan jaringan saraf

tiruan metode *backpropagation* terdapat dua buah proses yaitu proses *training* dan proses *testing*.



Gambar 3.2 Diagram *Block* Proses Klasifikasi Daging Babi dan Sapi

3.4 Proses *Training*

Proses *training* dilakukan untuk mendapatkan nilai bobot unggul yang nantinya akan diuji pada proses *testing*. Langkah awal yang dilakukan yaitu mengambil citra pada *local disk*, lalu citra tersebut akan dilakukan proses *cropping*, yang setenjutnya akan dilakukan proses pengambilan nilai *Red, Green, Blue* (RGB) dan dinormalisasi untuk meyetarakan nilai RGB yang bernilai 0–1. Setelah didapatkan nilai normalisasi RGB, citra akan dikonversi menjadi warna abu-abu untuk dapat diekstraksi ke dalam parameter GLCM. Setelah diekstraksi sistem akan melakukan proses *training* hingga menemukan bobot yang optimal.

3.5 Proses *Testing*

Proses *testing* dilakukan untuk menguji sistem dengan nilai bobot unggul yang paling unggul dari proses *testing*. Langkah-langkah yang dilakukan untuk melakukan proses *testing* yaitu mengambil citra pada *local disk*, lalu citra tersebut akan dilakukan proses *cropping*, yang selanjutnya akan dilakukan proses pengambilan nilai *Red, Green, Blue* (RGB) dan dinormalisasi untuk meyetarakan nilai RGB yang bernilai 0–1. Setelah didapatkan nilai normalisasi RGB, citra akan dikonversi menjadi warna abu-abu untuk dapat diekstraksi ke dalam parameter GLCM. Kemudian sistem akan melakukan proses perhitungan dengan menggunakan *backpropagation* dan mengidentifikasinya sesuai dengan bobot optimal yang telah didapat pada saat proses *training*. Hal ini dilakukan untuk memperoleh hasil klasifikasi babi atau sapi.

3.6 Metode Pengujian

- Pengujian Unit

Pengujian unit ditujukan untuk memastikan bahwa setiap fungsi berjalan dengan benar dimana strategi pengujiannya menggunakan *blackbox*.

- Pengujian Akurasi

Pengujian akurasi perhitungan diperlukan untuk menguji tujuan utama dari penelitian ini bahwa dengan menerapkan salah satu metode dari Jaringan Syaraf Tiruan yaitu metode *Backpropagation* sebagai proses klasifikasi daging babi dan sapi berdasarkan warna dan tekstur.

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis Sistem

Analisis sistem merupakan sebuah analisa penjabaran tentang komponen yang dibutuhkan oleh sistem, baik itu perangkat lunak maupun perangkat keras. Serta membahas tentang gambaran umum dari sistem yang akan dibuat.

Terdapat dua proses di dalam aplikasi ini, yaitu proses *training* dan proses *testing*. Dimana pada saat proses *training* dilakukan bertujuan untuk mendapatkan bobot terbaik. Sedangkan pada saat proses *testing* dilakukan bertujuan untuk mengklasifikasi daging babi dan daging sapi dengan menggunakan bobot yang telah didapatkan dari proses *training*.

4.1.1 Kebutuhan Perangkat Lunak

Agar dapat melakukan perancangan dan menjalankan aplikasi klasifikasi daging babi dan sapi menggunakan *Backpropagation* perlu diperhatikan kebutuhan perangkat lunak sebagai berikut:

- a) Sistem Operasi: *Windows 7*
- b) *Software* untuk membuat aplikasi *desktop*: *Microsoft Visual Studio 2012*
- c) *Microsoft Excel 2007* untuk *database* dan pengolahan data *training*.

4.1.2 Kebutuhan Perangkat Lunak

Spesifikasi *minimum* yang diperlukan agar perangkat lunak di atas dapat dijalankan yaitu:

- a) CPU 2.0GHz
- b) HDD 100GB
- c) RAM 4GB
- d) Kamera *Smartphone* dengan resolusi minimum 5MP

4.1.3 Gambaran Umum Aplikasi

Aplikasi klasifikasi daging babi dan sapi menggunakan Jaringan Syaraf Tiruan (JST). Aplikasi ini dapat membuka citra daging yang sebelumnya telah diambil menggunakan kamera *smartphone*. Setelah citra berhasil dibuka oleh sistem, maka akan dilakukan proses *cropping*. Setelah proses *cropping* akan dilakukan proses pengambilan nilai *Red, Green, Blue* (RGB) dan dinormalisasi.

Setelah mendapatkan nilai normalisasi RGB citra akan dikonversi menjadi warna abu-abu lalu diekstraksi ke dalam parameter GLCM.

4.2 Perancangan Sistem

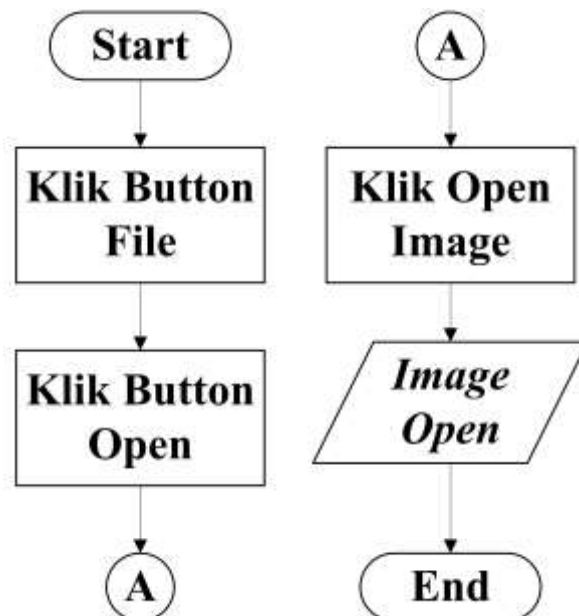
Perancangan sistem merupakan suatu proses desain sistem dalam penggambaran dan pembuatan sketsa *interface* aplikasi hingga perhitungan dari metode itu sendiri yaitu *Backpropagation*. Rancangan ini sendiri akan terbagi menjadi tiga yaitu perancangan perhitungan metode, perancangan proses dalam bentuk *flowchart*, dan perancangan user *interface* atau *mockup* dari aplikasi ini.

4.2.1 Alur Sistem Klasifikasi Daging

Sistem klasifikasi ini terbagi menjadi beberapa proses diantaranya yaitu *Image Input*, *Feature Extraction*, dan *Classification*.

- *Image Input*

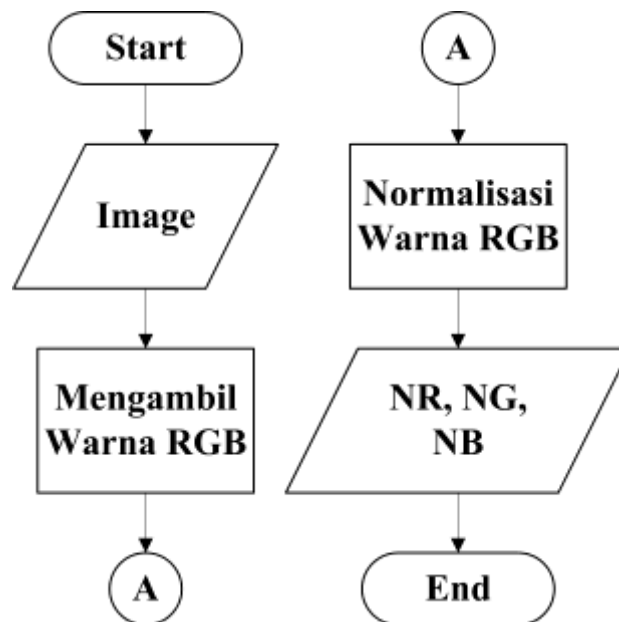
Sistem ini digunakan untuk memperoleh ciri daging babi dan daging sapi dengan mengambil warna dan tekstur pada permukaan daging. Oleh karena itu untuk mendapatkan data perlu dilakukan *image input*. Citra daging akan didapatkan dari HDD yang sebelumnya telah dipotret menggunakan kamera *smartphone*. Jarak antar kamera dan objek sekitar 10cm dan mendapatkan pencahayaan yang cukup. Alur *input image* dapat dilihat pada gambar 4.1



Gambar 4.1 Alur Proses *Image Input*

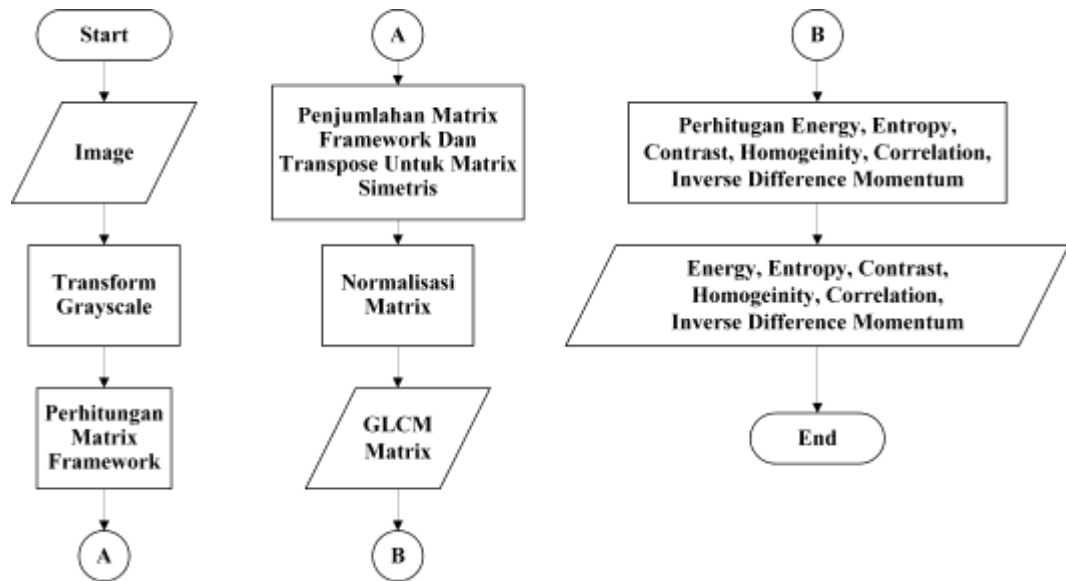
- *Feature Extraction*

Feature Extraction merupakan sebuah proses ekstraksi warna dan tekstur, guna memproses klasifikasi daging. Proses ekstraksi warna dan tekstur ini menggunakan pengambilan nilai citra warna *Red, Green, Blue*, yang kemudian nilai tersebut akan di normalisasi untuk mempermudah proses *training* data. Alur ekstraksi warna digambarkan ada gambar 4.2



Gambar 4.2 Alur Proses Ekstraksi Ciri Warna

Setelah dilakukan ekstraksi ciri warna sistem akan melakukan ekstraksi fitur tekstur. Pada sistem ini ekstraksi fitur akan menggunakan *Gray Level Co-Occurrence Matrix*, citra yang digunakan merupakan citra *grayscale*, yang didapatkan dari *average*. Kemudian dari *image* tersebut akan diolah dan diperoleh *matrix framework* berdasarkan sudut dan jarak yang sudah ditentukan, *matrix* tersebut kemudian dirubah menjadi *matrix simetris* dengan menjumlahkan matrik tersebut dengan *matrix transform* dan *matrix framework*, setelah itu hasil penjumlahan *matrix* tersebut di normalisasi untuk menghilangkan ketergantungan antar *matrix*, hasil *matrix* tersebut disebut dengan *GLCM matrix*, dari *matrix* ini dapat dihitung fitur tekstur *energy*, *entropy*, *contrast*, *homogeinity*, *correlation*, *inverse difference momentum*. Alur ekstraksi tekstur dapat dideskripsikan seperti pada gambar 4.3



Gambar 4.3 Alur Proses Ekstraksi Tekstur

Untuk gambar dibawah, sebagai contoh ketetanggaan piksel dipilih ke arah timur laut (pojok kanan atas). Dengan hubungan tetangga (1, 45) 1 untuk jarak spasial piksel, sedangkan 45 untuk sudut tetangga yang digunakan. Gambar dibawah merupakan gambar contoh yang diambil piksel 4x4 yang telah ditandai. Apabila citra diambil menggunakan sudut 45 derajat dan jarak 1 piksel, maka akan menghasilkan *framework* matriks seperti pada tabel 4.3. Setelah mendapatkan *framework* matriks citra akan ditambahkan dengan transpos matriks seperti pada tabel 4.4 dan akan menghasilkan jumlah matriks seperti tabel 4.5.



Gambar 4.4 Citra Yang Diambil Dengan Pixel 4x4

Tabel 4.1 Citra Asli

3	2	3	1
1	0	3	2
0	2	1	3
0	1	0	1

Tabel 4.2 Komposisi Piksel

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Tabel 4.3 *Framework Matrix*

1	0	1	2
0	1	2	0
0	0	0	1
0	1	0	0

Tabel 4.4 *Transpose Matrix*

1	0	0	0
0	1	0	1
1	2	0	0
2	0	1	0

Tabel 4.5 Hasil Penjumlahan (*Simetris Matrix*)

2	0	1	2
0	2	2	1
1	2	0	1
2	1	1	0

Setelah mendapatkan hasil penjumlahan akan dilakukan normalisasi untuk menghilangkan ketergantungan pada ukuran citra, sehingga jumlahnya menjadi 1, seperti pada tabel 4.7.

Tabel 4.6 *Normalization Matrix*

$\frac{2}{18}$	$\frac{0}{18}$	$\frac{1}{18}$	$\frac{2}{18}$
$\frac{0}{18}$	$\frac{2}{18}$	$\frac{2}{18}$	$\frac{1}{24}$
$\frac{1}{18}$	$\frac{2}{18}$	$\frac{0}{18}$	$\frac{1}{18}$
$\frac{2}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{0}{18}$

Tabel 4.7 GLCM *Matrix*

0,111	0	0,056	0,111
0	0,111	0,111	0,056
0,056	0,111	0	0,056
0,111	0,056	0,056	0

Ciri tekstur yang digunakan ada 6 ciri diantaranya energi, entropi, kontras, homogenitas, korelasi, dan Momentum Selisih Invers. Berikut contoh perhitungan dari berbagai ciri *Gray Level Co-Occurrence Matrix*

✓ Energi (*Energy*)

Untuk mendapatkan nilai energi didefinisikan pada *equation* 4.1. Sehingga dapat menghasilkan matriks energi (*Energy Matrix*) seperti tabel 4.9

$$f_1 = \sum_i \sum_j P_d^2(i, j) \quad (4.1)$$

Tabel 4.8 Perhitungan Matrik Energi

$0,111^2$	0^2	$0,56^2$	$0,111^2$
0^2	$0,111^2$	$0,111^2$	$0,56^2$
$0,56^2$	$0,111^2$	0^2	$0,56^2$
$0,111^2$	$0,56^2$	$0,56^2$	0^2

Tabel 4.9 *Energy Matrix*

0,012	0	0,003	0,012
0	0,012	0,012	0,003
0,003	0,012	0	0,003
0,012	0,003	0,003	0

✓ Entropi (*Entropy*)

Untuk mendapatkan nilai entropi didefinisikan pada *equation* 4.2. Sehingga dapat menghasilkan matriks entropi (*Entropy Matrix*) seperti tabel 4.11

$$f_2 = - \sum_i \sum_j P_d(i, j) \log(P_d(i, j)) \quad (4.2)$$

Tabel 4.10 Perhitungan Matrik Entropi

0,111*Log(0,111)	0*Log(0)	0,056*Log(0,056)	0,111*Log(0,111)
0*Log(0)	0,111*Log(0,111)	0,111*Log(0,111)	0,056*Log(0,056)
0,056*Log(0,056)	0,111*Log(0,111)	0*Log(0)	0,056*Log(0,056)
0,111*Log(0,111)	0,056*Log(0,056)	0,056*Log(0,056)	0*Log(0)

Tabel 4.11 *Entropy Matrix*

0.10603	0	0.0697	0.10603
0	0.106	0.106	0.06974
0.06974	0.106	0	0.06974
0.10603	0.0697	0.0697	0

✓ Kontras (*Contras*)

Untuk mendapatkan nilai kontras didefinisikan pada *equation* 4.3. Sehingga dapat menghasilkan kontras matriks seperti tabel 4.13

$$f_3 = \sum_i \sum_j (i - j)^2 P_d(i, j) \quad (4.3)$$

Tabel 4.12 Perhitungan Matrik Kontras

(0-0)2*0,111	(0-1)2*0	(0-2)2*0,056	(0-3)2*0,111
(1-0)2*0	(1-1)2*0,111	(1-2)2*0,111	(1-3)2*0,056
(2-0)2*0,056	(2-1)2*0,111	(2-2)2*0	(2-3)2*0,056
(3-0)2*0,111	(3-1)2*0,056	(3-2)2*0,056	(3-3)2*0

Tabel 4.13 *Contrast Matrix*

0	0	0.222	1
0	0	0.111	0.222
0.222	0.111	0	0.056
1	0.222	0.056	0

✓ Homogenitas (*Homogeneity*)

Untuk mendapatkan nilai homogenitas didefinisikan pada *equation* 4.4. Sehingga dapat menghasilkan homogenitas matriks seperti tabel 4.15

$$f_4 = \sum_i \sum_j \frac{P_d(i, j)}{1 + |i - j|} \quad (4.4)$$

Tabel 4.14 Perhitungan Matrik Homogenitas

0,111/(1+(0-0))	0/(1+(0-1))	0,056/(1+(0-2))	0,111/(1+(0-3))
0/(1+(1-0))	0,111/(1+(1-1))	0,111/(1+(1-2))	0,056/(1+(1-3))
0,056/(1+(2-0))	0,111/(1+(2-1))	0/(1+(2-2))	0,056/(1+(2-3))
0,111/(1+(3-0))	0,056/(1+(3-1))	0,056/(1+(3-2))	0/(1+(3-3))

Tabel 4.15 Homogeneity Matrix

0.111	0	0.056	0.056
0	0.111	0.056	-0.06
0.056	0.056	0	0.028
0.028	0.019	0.028	0

✓ Korelasi (*Correlation*)

Untuk mendapatkan nilai korelasi didefinisikan pada *equation* 4.5. Sehingga dapat menghasilkan korelasi matriks seperti tabel 4.17

$$f_5 = \sum_i \sum_j \frac{(i - \mu_x)(j - \mu_y)P_d(i, j)}{\sigma_x \sigma_y} \quad (4.5)$$

dimana:

$$\mu_x = \sum_i i \sum_j P_d(i, j)$$

$$\mu_y = \sum_i j \sum_j P_d(i, j)$$

$$\sigma_x^2 = \sum_i (i - \mu_x)^2 \sum_j P_d(i, j)$$

$$\sigma_y^2 = \sum_i (i - \mu_y)^2 \sum_j P_d(i, j)$$

Tabel 4.16 Perhitungan Matrik Korelasi

((0-0,111)*(0-0,111)*0,111)/((0-0,111)2*0,111)*((0-0,111)2*0,111)	((0-0)*(1-0)*0)/((0-0)2*0)*((1-0)2*0)	((0-0,056)*(2-0,056)*0,056)/((0-0,056)2*0,056)*((2-0,056)2*0,056)	((0-0,111)*(3-0,111)*0,111)/((0-0,111)2*0,111)*((3-0,111)2*0,111)
((0-0)*(1-0)*0)/((0-0)2*0)*((1-0)2*0)	((1-0,111)*(1-0,111)*0,111)/((1-0,111)2*0,111)*((1-0,111)2*0,111)	((1-0,111)*(2-0,111)*0,111)/((1-0,111)2*0,111)*((2-0,111)2*0,111)	((1-0,056)*(3-0,056)*0,056)/((1-0,056)2*0,056)*((3-0,056)2*0,056)
((2-0,056)*(0-0,056)*0,056)/((2-0,056)2*0,056)*((0-0,056)2*0,056)	((2-0,111)*(1-0,111)*0,111)/((2-0,111)2*0,111)*((1-0,111)2*0,111)	((2-0)*(2-0)*0)/((2-0)2*0)*((2-0)2*0)	((2-0,056)*(3-0,056)*0,056)/((2-0,056)2*0,056)*((3-0,056)2*0,056)
((3-0,111)*(0-0,111)*0,111)/((3-0,111)2*0,111)*((0-0,111)2*0,111)	((3-0,056)*(1-0,056)*0,056)/((3-0,056)2*0,056)*((1-0,056)2*0,056)	((3-0,056)*(2-0,056)*0,056)/((3-0,056)2*0,056)*((2-0,056)2*0,056)	((3-0)*(3-0)*0)/((3-0)2*0)*((3-0)2*0)

Tabel 4.17 *Correlation Matrix*

0.00137	0	-7.352	-24.11
0	0.0878	0.8424	1.50162
-4.9E-06	0.0413	0	0.72936
-5.3E-05	0.0159	0.1387	0

✓ Momentum Selisih Invers (*Inverse Difference Momentum*)

Untuk mendapatkan nilai momentum selisih didefinisikan pada *equation* 4.6.

Sehingga dapat menghasilkan matriks momentum selisih invers seperti tabel 4.19

$$f_6 = \sum_i \sum_j \frac{1}{1+(i-j)^2} P_d(i, j) \quad (4.6)$$

Tabel 4.18 Perhitungan Matrik Momentum Selisih Invers

1/((+((0-0)2)*0,111	1/((+((0-1)2)*0	1/((+((0-2)2)*0,056	1/((+((0-3)2)*0,111
1/((+((1-0)2)*0	1/((+((1-1)2)*0,111	1/((+((1-2)2)*0,111	1/((+((1-3)2)*0,056
1/((+((2-0)2)*0,056	1/((+((2-1)2)*0,111	1/((+((2-2)2)*0	1/((+((2-3)2)*0,056
1/((+((3-0)2)*0,111	1/((+((3-1)2)*0,056	1/((+((3-2)2)*0,056	1/((+((3-3)2)*0

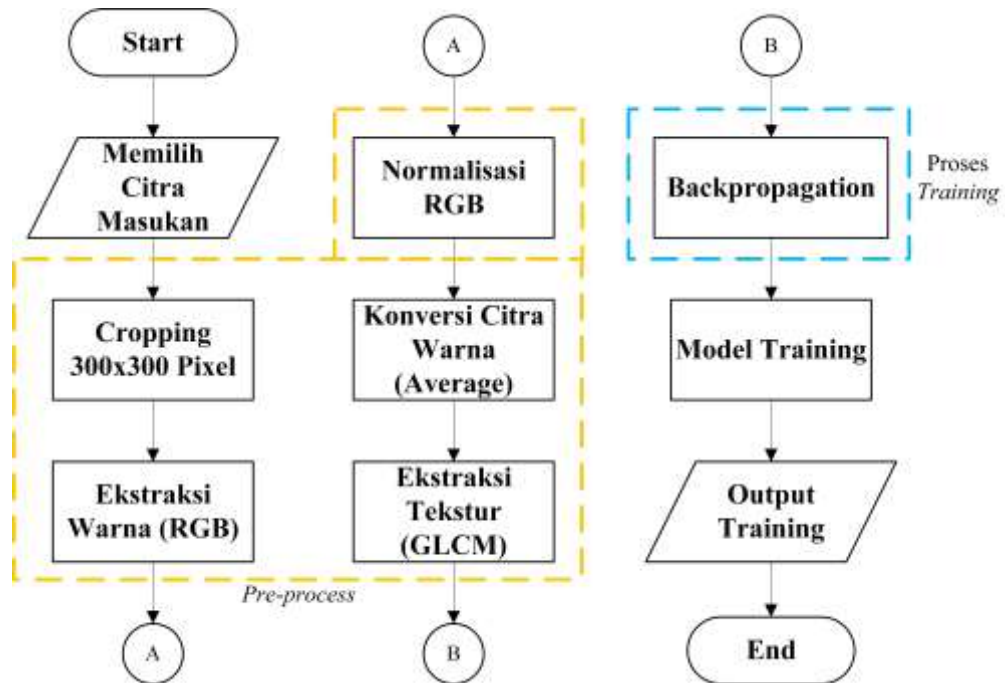
Tabel 4.19 *Inverse Difference Momentum Matrix*

0.111	0	0.011	0.011
0	0.111	0.056	0.011
0.011	0.056	0	0.028
0.011	0.011	0.028	0

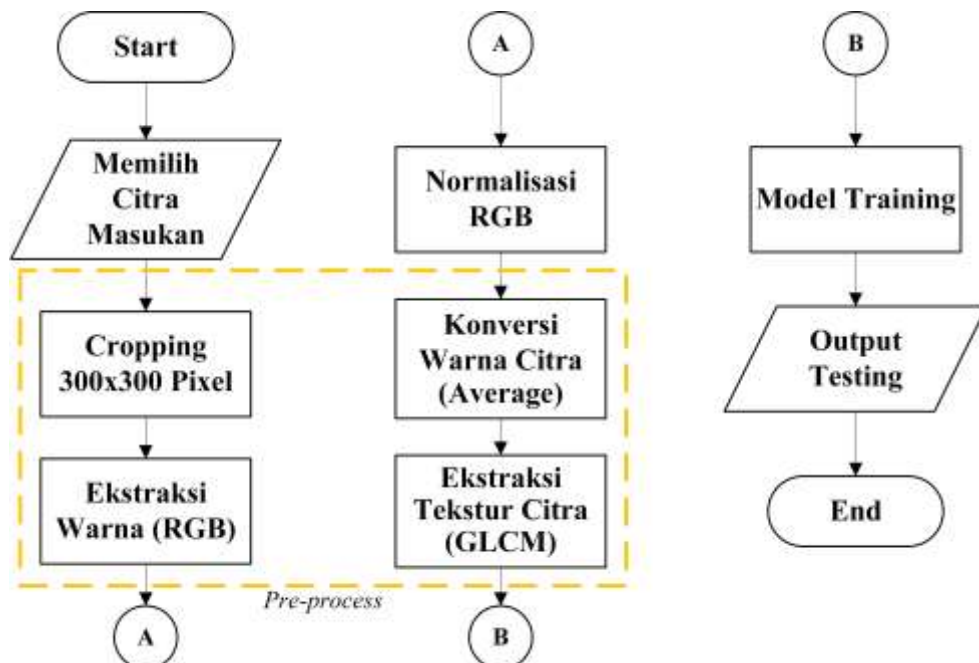
• Klasifikasi

Klasifikasi daging terbagi menjadi 2 proses, yaitu *training* dan *testing* yang akan menggunakan Jaringan Syaraf Tiruan (JST) dengan metode *Backpropagation*. Didalam *training* terdapat *cropping image* untuk memilih daging pada bagian tengah. Setelah itu akan dilakukan ekstraksi warna dan kemudian di normalisasi yang selanjutnya akan dikonversi menjadi warna abu-abu untuk dilakukan proses ekstraksi tekstur. Setelah mendapatkan nilai ekstraksi tekstur maka dilakukan proses *training* yang akan menghasilkan nilai bobot terbaik. Sedangkan untuk proses *testing* gambaran umum yang dilakukan sama seperti proses *training*, hanya saja data akan diambil dari proses *training* yang telah disimpan terlebih dahulu. Gambar 4.5 merupakan alur proses *training*

untuk mendapatkan bobot optimal guna proses *testing*. Sedangkan gambar 4.6 merupakan alur proses *testing*. Untuk menentukan daging menggunakan metode *backpropagation* berdasarkan bobot optimal hasil *training*.



Gambar 4.5 Gambaran Umum Proses *Training*



Gambar 4.6 Gambaran Umum Proses *Testing*

- Perancangan Algoritma *Backpropagation*

Di dalam metode *Backpropagation* terdapat tiga fase yang akan menghasilkan nilai akhir dari metode ini. Berikut penjelasan dari masing-masing tahapannya:

- ✓ Fase Propagasi Maju

Selama propagasi maju ini sinyal-sinyal masukan akan dipropagasikan kepada layer berikutnya yaitu *hidden layer* menggunakan fungsi aktivasi yang telah ditentukan. Kemudian keluaran dari unit-unit tersembunyi ini akan dipropagasikan kepada layer berikutnya yaitu *output layer* dengan menggunakan fungsi aktivasi yang telah ditentukan pula. Berikutnya, nilai *output* yang berada pada *output layer* akan dibandingkan dengan target yang harus dicapai. Selisih diantara keduanya merupakan kesalahan yang terjadi. Jika kesalahan tersebut nilainya lebih kecil dari batas toleransi yang ditentukan maka iterasi akan dihentikan, namun jika nilai kesalahan masih lebih besar dari batas toleransi maka bobot setiap *neuron* dari jaringan akan dimodifikasi untuk mengurangi kesalahan.

- ✓ Fase Propagasi Mundur

Langkah berikutnya dengan berdasarkan kesalahan yang telah didapat, maka dihitunglah faktor kesalahan (δ_z) yang terjadi. Dan akan dipakai untuk mendistribusikan kesalahan ke unit sebelumnya (*hidden layer*). Nilai δ_z juga dipakai untuk mengubah bobot *neuron* yang berhubungan langsung dengan *output*.

Lalu dengan cara yang sama, dihitung faktor δ_y di setiap *hidden layer* sebagai dasar perubahan bobot *neuron* yang berasal dari *hidden layer* kepada layer sebelumnya yaitu *input layer*.

- ✓ Fase Modifikasi Bobot

Setelah semua faktor δ dihitung, maka semua bobot *neuron* dimodifikasi bersamaan. Perubahan bobot suatu *neuron* didasarkan atas faktor δ pada *neuron* di layer atasnya.

- Perancangan Perhitungan *Backpropagation*

Pada implementasi aplikasi klasifikasi daging babi dan sapi berdasarkan warna dan tekstur menggunakan perancangan perhitungan *Backpropagation*

dengan fungsi aktivasi *sigmoid biner*. Di bawah ini merupakan perhitungan dari metode yang digunakan oleh penulis:

1. *Open Image* di dalam *harddisk* ataupun media penyimpanan lainnya
2. Memproses citra tersebut dengan menggunakan cara *cropping* dengan posisi *cropping* tepat pada bagian tengah daging.
3. Mendapatkan nilai RGB dari setiap *pixel* dari hasil *cropping*.
4. Melakukan normalisasi nilai RGB

$$\text{Normalisasi Red (NR)} = \frac{R}{R+G+B} \quad (4.7)$$

$$\text{Normalisasi Green (NG)} = \frac{G}{R+G+B} \quad (4.8)$$

$$\text{Normalisasi Blue (NB)} = \frac{B}{R+G+B} \quad (4.9)$$

5. Melakukan konversi warna ke abu-abu

$$\text{Average} = \frac{(R+G+B)}{3} \quad (4.10)$$

6. Melakukan perhitungan statistik orde 2 *Gray Level Co-Occurrence Matrix* dengan mengambil nilai tekstur *energy* (enerjum), *entropy* (entrojum), *contrast* (konjum), *homogeneity* (homojum), *correlation* (korjum), dan *inverse difference momentum* (msjum) dari 4 sudut yaitu 0, 45, 90 dan 135 dan dengan jarak 1 piksel. Sehingga masing-masing nilai tekstur yaitu *energy*, *entropy*, *contrast*, *homogeneity*, *correlation*, dan *inverse difference momentum* mempunyai 4 nilai dari sudut yang berbeda.
7. Selanjutnya nilai tekstur dari masing-masing sudut di ratakan sehingga nilai tekstur dari empat sudut berbeda dibandingkan satu sama lain, nilai tekstur paling besar dari ke empat sudut adalah nilai yang mewakili nilai tekstur tersebut, sehingga didapatkan nilai tekstur yaitu rata-rata *energy* (renerjum), rata-rata *entropy* (rentrojum), rata-rata *contrast* (rkonjum), rata-rata *homogeneity* (rhomojum), rata-rata *correlation* (rkorjum), dan rata-rata *inverse difference momentum* (rmsjum).

$$\text{renerjum} = \frac{(\text{enerjum0} + \text{enerjum45} + \text{enerjum90} + \text{enerjum135})}{4} \quad (4.11)$$

$$\text{rentrojum} = \frac{(\text{entrojum0} + \text{entrojum45} + \text{entrojum90} + \text{entrojum135})}{4} \quad (4.12)$$

$$\text{rkonjum} = \frac{(\text{konjum0} + \text{konjum45} + \text{konjum90} + \text{konjum135})}{4} \quad (4.13)$$

$$\text{rhomojum} = \frac{(\text{homojum0} + \text{homojum45} + \text{homojum90} + \text{homojum135})}{4} \quad (4.14)$$

$$rkorjum = \frac{(korjum0 + korjum45 + korjum90 + korjum135)}{4} \quad (4.15)$$

$$rmsjum = \frac{(msjum0 + msjum45 + msjum90 + msjum135)}{4} \quad (4.16)$$

8. Memberi Nilai LR (*Learning Rate*), Bobot *Neuron*, serta Bias.

LR (α)	= 0.2	V26	= 0.5
Bias V00	= -0.3	V27	= 0.3
Bias V01	= 0.6	V28	= 0.8
Bias V02	= 0.2	V29	= 0.4
V11	= 0.9	V31	= 0.9
V12	= 0.1	V32	= 0.8
V13	= 0.3	V33	= 0.7
V14	= 0.4	V34	= 0.5
V15	= 0.2	V35	= 0.6
V16	= 0.8	V36	= 0.2
V17	= 0.5	V37	= 0.4
V18	= 0.7	V38	= 0.3
V19	= 0.6	V39	= 0.1
V21	= 0.9	Bias W0	= -0.1
V22	= 0.6	W0	= -0.5
V23	= 0.9	W1	= -0.3
V24	= 0.1	W2	= 0.9
V25	= 0.2		

9. Setelah melakukan inisiasi semua bobot dengan bilangan acak kecil antara 0 sampai dengan 1, penulis melakukan optimalisasi bilangan acak kecil tersebut dengan menggunakan algoritma perhitungan dari *Nguyen-Widrow* dengan tujuan agar perhitungan yang dilakukan oleh sistem optimal.

10. Menentukan target (T)

Daging Babi = 1

Daging Sapi = 0

11. Menghitung semua keluaran di unit tersembunyi Y_j ($j = 1, \dots, n$)

$$Y_{net0} = BiasV00 + (NR * V11) + (NG * V12) + (NB * V13) + (renerjum * V14) + (rentrojum * V15) + (rkonjum * V16) + (rhomojum * V17) + (rkorjum * V18) + (rmsjum * V19) \quad (4.17)$$

$$Y_0 = \frac{1}{(1+EXP^{-Y_{net0}})} \quad (4.18)$$

$$Y_{net1} = BiasV01 + (NR * V21) + (NG * V22) + (NB * V23) + (renerjum * V24) + (rentrojum * V25) + (rkonjum * V26) + (rhomojum * V27) + (rkorjum * V28) + (rmsjum * V29) \quad (4.19)$$

$$Y_1 = \frac{1}{(1+EXP^{-Y_{net1}})} \quad (4.20)$$

$$Y_{net2} = BiasV02 + (NR * V31) + (NG * V32) + (NB * V33) + (renerjum * V34) + (rentrojum * V35) + (rkonjum * V36) + (rhomojum * V37) + (rkorjum * V38) + (rmsjum * V39) \quad (4.21)$$

$$Y_2 = \frac{1}{(1+EXP^{-Y_{net2}})} \quad (4.22)$$

12. Menghitung semua keluaran di unit *output* Z

$$Z_{net} = BiasW0 + (Y1 * W0) + (Y2 * W1) + (Y3 * W2) \quad (4.23)$$

$$Z = \frac{1}{(1+EXP^{-Z_{net}})} \quad (4.24)$$

13. Jika nilai *output* (Z) tidak sesuai dengan target yang ditentukan maka dilakukan pembobotan ulang (memasuki fase propagasi mundur). Maka dihitunglah faktor kesalahan δ_Z (Faktor kesalahan dari Z) dengan rumus:

$$\delta_Z = Z * (1 - Z) * (T - Z) \quad (4.25)$$

Setelah itu dilakukan perhitungan suku perubahan bobotnya (yang akan digunakan untuk merubah bobot dari W) dengan laju pembelajaran α

$$\Delta BiasW0 = \alpha * \delta_Z \quad (4.26)$$

$$\Delta W0 = \alpha * \delta_Z * Y0 \quad (4.27)$$

$$\Delta W1 = \alpha * \delta_Z * Y1 \quad (4.28)$$

$$\Delta W2 = \alpha * \delta_Z * Y2 \quad (4.29)$$

14. Lalu langkah selanjutnya menghitung faktor kesalahan *hidden layer* (faktor kesalahan dari Y) dengan rumus :

$$\delta_{netY0} = \delta_Z * W0 \quad (4.30)$$

$$\delta_{netY1} = \delta_Z * W1 \quad (4.31)$$

$$\delta_{netY2} = \delta_Z * W2 \quad (4.32)$$

$$\delta_{Y0} = \delta_{netY0} * Y0 * (1 - Y0) \quad (4.33)$$

$$\delta_{Y1} = \delta_{netY1} * Y1 * (1 - Y1) \quad (4.34)$$

$$\delta_{Y2} = \delta_{netY2} * Y2 * (1 - Y2) \quad (4.35)$$

Setelah itu dilakukan perhitungan suku perubahan bobotnya (yang akan digunakan untuk merubah bobot dari V) dengan laju pembelajaran α

$$\Delta \text{BiasV00} = \alpha * \delta_{Y0} \quad (4.36)$$

$$\Delta \text{BiasV01} = \alpha * \delta_{Y1} \quad (4.37)$$

$$\Delta \text{BiasV02} = \alpha * \delta_{Y2} \quad (4.38)$$

$$\Delta V11 = \alpha * \delta_{Y0} * NR \quad (4.39)$$

$$\Delta V12 = \alpha * \delta_{Y0} * NG \quad (4.40)$$

$$\Delta V13 = \alpha * \delta_{Y0} * NB \quad (4.41)$$

$$\Delta V14 = \alpha * \delta_{Y0} * \text{renerjum} \quad (4.42)$$

$$\Delta V15 = \alpha * \delta_{Y0} * \text{rentrojum} \quad (4.43)$$

$$\Delta V16 = \alpha * \delta_{Y0} * rkonjum \quad (4.44)$$

$$\Delta V17 = \alpha * \delta_{Y0} * rhomojum \quad (4.45)$$

$$\Delta V18 = \alpha * \delta_{Y0} * rkorjum \quad (4.46)$$

$$\Delta V19 = \alpha * \delta_{Y0} * rmsjum \quad (4.47)$$

$$\Delta V21 = \alpha * \delta_{Y1} * NR \quad (4.48)$$

$$\Delta V22 = \alpha * \delta_{Y1} * NG \quad (4.49)$$

$$\Delta V23 = \alpha * \delta_{Y1} * NB \quad (4.50)$$

$$\Delta V24 = \alpha * \delta_{Y1} * \text{renerjum} \quad (4.51)$$

$$\Delta V25 = \alpha * \delta_{Y1} * \text{rentrojum} \quad (4.52)$$

$$\Delta V26 = \alpha * \delta_{Y1} * rkonjum \quad (4.53)$$

$$\Delta V27 = \alpha * \delta_{Y1} * rhomojum \quad (4.54)$$

$$\Delta V28 = \alpha * \delta_{Y1} * rkorjum \quad (4.55)$$

$$\Delta V29 = \alpha * \delta_{Y1} * rmsjum \quad (4.56)$$

$$\Delta V31 = \alpha * \delta_{Y2} * NR \quad (4.57)$$

$$\Delta V32 = \alpha * \delta_{Y2} * NG \quad (4.58)$$

$$\Delta V33 = \alpha * \delta_{Y2} * NB \quad (4.59)$$

$$\Delta V34 = \alpha * \delta_{Y2} * \text{renerjum} \quad (4.60)$$

$$\Delta V35 = \alpha * \delta_{Y2} * \text{rentrojum} \quad (4.61)$$

$$\Delta V36 = \alpha * \delta_{Y2} * rkonjum \quad (4.62)$$

$$\Delta V37 = \alpha * \delta_{Y2} * rhomojum \quad (4.63)$$

$$\Delta V38 = \alpha * \delta_{Y2} * rkorjum \quad (4.64)$$

$$\Delta V39 = \alpha * \delta_{Y2} * rmsjum \quad (4.65)$$

15. Langkah terakhir hitung semua perubahan bobotnya yang akan digunakan sebagai bobot awal *looping* selanjutnya.

Perubahan bobot *neuron* yang menuju ke *output*:

$$\text{BiasW0 new} = \text{BiasW0} + \Delta\text{BiasW0} \quad (4.66)$$

$$\text{W0 new} = \text{W0} + \Delta\text{W0} \quad (4.67)$$

$$\text{W1 new} = \text{W1} + \Delta\text{W1} \quad (4.68)$$

$$\text{W2 new} = \text{W2} + \Delta\text{W2} \quad (4.69)$$

Perubahan bobot *neuron* yang menuju ke *hidden layer*:

$$\text{BiasV0 new} = \text{BiasV0} + \Delta\text{BiasV0} \quad (4.70)$$

$$\text{BiasV1 new} = \text{BiasV1} + \Delta\text{BiasV1} \quad (4.71)$$

$$\text{BiasV2 new} = \text{BiasV2} + \Delta\text{BiasV2} \quad (4.72)$$

$$\text{V11 new} = \text{V110} + \Delta\text{V11} \quad (4.73)$$

$$\text{V12 new} = \text{V120} + \Delta\text{V12} \quad (4.74)$$

$$\text{V13 new} = \text{V130} + \Delta\text{V13} \quad (4.75)$$

$$\text{V14 new} = \text{V140} + \Delta\text{V14} \quad (4.76)$$

$$\text{V15 new} = \text{V150} + \Delta\text{V15} \quad (4.77)$$

$$\text{V16 new} = \text{V160} + \Delta\text{V16} \quad (4.78)$$

$$\text{V17 new} = \text{V170} + \Delta\text{V17} \quad (4.79)$$

$$\text{V18 new} = \text{V180} + \Delta\text{V18} \quad (4.80)$$

$$\text{V19 new} = \text{V190} + \Delta\text{V19} \quad (4.81)$$

$$\text{V21 new} = \text{V21} + \Delta\text{V21} \quad (4.82)$$

$$\text{V22 new} = \text{V22} + \Delta\text{V22} \quad (4.83)$$

$$\text{V23 new} = \text{V23} + \Delta\text{V23} \quad (4.84)$$

$$\text{V24 new} = \text{V24} + \Delta\text{V24} \quad (4.85)$$

$$\text{V25 new} = \text{V25} + \Delta\text{V25} \quad (4.86)$$

$$\text{V26 new} = \text{V26} + \Delta\text{V26} \quad (4.87)$$

$$\text{V27 new} = \text{V27} + \Delta\text{V27} \quad (4.88)$$

$$\text{V28 new} = \text{V28} + \Delta\text{V28} \quad (4.89)$$

$$\text{V29 new} = \text{V29} + \Delta\text{V29} \quad (4.90)$$

$$\text{V31 new} = \text{V31} + \Delta\text{V31} \quad (4.91)$$

$$\text{V32 new} = \text{V32} + \Delta\text{V32} \quad (4.92)$$

$$\text{V33 new} = \text{V33} + \Delta\text{V33} \quad (4.93)$$

$$V34_{new} = V34 + \Delta V34 \quad (4.94)$$

$$V35_{new} = V35 + \Delta V35 \quad (4.95)$$

$$V36_{new} = V36 + \Delta V36 \quad (4.96)$$

$$V37_{new} = V37 + \Delta V37 \quad (4.97)$$

$$V38_{new} = V38 + \Delta V38 \quad (4.98)$$

$$V39_{new} = V39 + \Delta V39 \quad (4.99)$$

16. Kembali pada langkah ke sembilan (12)

4.2.2 Perhitungan Sistem

Klasifikasi daging pada sistem ini menggunakan metode *backpropagation*, dimana data yang dimasukan merupakan ekstraksi warna (*Red, Green, Blue*) dan tekstur (*energy, entropy, kontras, homogeinity, correlation, dan inverse difference momentum*). Tabel 4.20 merupakan 2 sample data dari ekstraksi tekstur dan ekstraksi warna pada sistem yang mewakili daging yaitu daging sapi dan daging babi.

Tabel 4.20 Sampel Data Ekstraksi Fitur

A	B	C	D	E	F	G	H	I	J
0.44	0.25	0.29	0.00	0.02	0.09	0.40	0.96	0.32	1
6531	5152	8317	2674	6017	102	6224	2555	9849	
0.56	0.20	0.23	0.00	0.02	0.34	0.34	0.84	0.26	0
655	2182	1268	2051	7735	4868	0955	6633	3616	

Keterangan:

A: *Red*

F: *Contras*

B: *Green*

G: *Homogeinity*

C: *Blue*

H: *Correlation*

D: *Energy*

I: *Inverse Difference Momentum*

E: *Entropy*

J: *Target*

Untuk proses klasifikasi daging menggunakan *backpropagation*, dua data sampel akan dilakukan *training* sesuai perhitungan *backpropagation*:

1. Pertama inisialisasi *Learning Rate*, bobot *neuron* penghubung serta bobot bias yang telah ditentukan. Berikut merupakan bobot-bobot yang digunakan untuk perhitungan *backpropagation*:

Tabel 4.21 Nilai Bobot

NO	VARIABEL	BOBOT
1	<i>Learning Rate</i>	0.2
2	Bias V00	-0.3
3	Bias V01	0.6
4	Bias v02	0.2
5	V11	0.9
6	V12	0.1
7	V13	0.3
8	V14	0.4
9	V15	0.2
10	V16	0.8
11	V17	0.5
12	V18	0.7
13	V19	0.6
14	V21	0.9
15	V22	0.6
16	V23	0.7
17	V24	0.1
18	V25	0.2
19	V26	0.5
20	V27	0.3
21	V28	0.8
22	V29	0.4
23	V31	0.9
24	V32	0.8
25	V33	0.7
26	V34	0.5
27	V35	0.6
28	V36	0.2
29	V37	0.4

NO	VARIABEL	BOBOT
30	V38	0.3
31	V39	0.1
32	Bias W	-0.5
33	W0	-0.3
34	W1	0.9
35	W2	-0.1

2. Setelah semua data terkumpul maka langkah selanjutnya menghitung *output* jaringan yang berada pada *hidden layer*, berikut implementasi perhitungannya.

$$Y_{net0} = -0.29313 + (0.446531 * 0.35726) + (0.255152 * 0.35727) + (0.298317 * 0.61448) + (0.002674 * 0.97503) + (0.026017 * 0.89) + (0.09102 * 0.89) + (0.406224 * 0.12455) + (0.962555 * 0.7213) + (0.329849 * 1.00056)$$

$$Y_{net0} = 1.322557$$

$$Y_0 = \frac{1}{(1 + EXP^{-1.322557})} = 0.789607$$

$$Y_{net1} = 0.60359 + (0.446531 * 0.00942) + (0.255152 * 0.12015) + (0.298317 * 0.59238) + (0.002674 * 0.7064) + (0.026017 * 0.13111) + (0.09102 * 0.84643) + (0.406224 * 0.91031) + (0.962555 * 0.71245) + (0.329849 * 0.55885)$$

$$Y_{net1} = 2.137407$$

$$Y_1 = \frac{1}{(1 + EXP^{-2.137407})} = 0.894486$$

$$Y_{net2} = 0.21962 + (0.446531 * 0.54899) + (0.255152 * 0.93535) + (0.298317 * 0.67423) + (0.002674 * 0.70195) + (0.026017 * 0.97203) + (0.09102 * 0.92228) + (0.406224 * -0.2931) + (0.962555 * 0.60359) + (0.329849 * 0.21962)$$

$$Y_{net2} = 1.550012$$

$$Y_2 = \frac{1}{(1 + EXP^{-1.550012})} = 0.824916$$

3. Selanjutnya nilai *output* dihitung, berikut implementasi perhitungannya.

$$Z_{net} = -0.27089 + (0.789607 * 0.251913) + (0.894486 * 0.582666) + (0.824916 * 1.84292)$$

$$Z_{net} = 1.969471$$

$$Z = \frac{1}{(1 + EXP^{-1.969471})} = 0.877554$$

4. Jika nilai *output* Z tidak sesuai dengan target yang ditemukan (dalam hal ini target untuk sampel pertama yaitu 0), maka akan dilakukan fase propagasi mundur. Langkah pertama yang dilakukan untuk pembobotan ulang yaitu mencari faktor kesalahan dari Z (δ_Z)

$$\delta_Z = 0.877554 * (1 - 0.877554) * (1 - 0.877554) = 0.013157122$$

Setelah itu dilakukan perhitungan suku perubahan bobotnya yang akan digunakan untuk merubah bobot dari W.

$$\Delta BiasW0 = 0.05 * 0.013157122 = 0.0026314$$

$$\Delta W00 = 0.05 * 0.013157122 * 0.789607 = 0.0020778$$

$$\Delta W10 = 0.05 * 0.013157122 * 0.894486 = 0.0023538$$

$$\Delta W20 = 0.05 * 0.013157122 * 0.824916 = 0.002170$$

5. Setelah suku perubahan bobot *neuron* W ditemukan, maka selanjutnya menghitung faktor kesalahan dari *neuron* V.

$$\delta_{netY0} = 0.013157122 * 0.251913 = 0.003342$$

$$\delta_{netY1} = 0.013157122 * 0.582666 = 3.09688$$

$$\delta_{netY2} = 0.013157122 * 1.84292 = 2.85602$$

$$\delta_{Y0} = 0.003342 * 0.789607 * (1 - 0.789607) = 0.000555$$

$$\delta_{Y1} = 3.09688 * 0.894486 * (1 - 0.894486) = 2.92286$$

$$\delta_{Y2} = 2.85602 * 0.824916 * (1 - 0.824916) = 4.12494$$

Selanjutnya dilakukan penghitungan suku perubahan bobot *neuron* V dengan menggunakan nilai dari faktor kesalahan yang telah ditemukan.

$$\Delta BiasV00 = 0.05 * 0.000555 = 0.000111$$

$$\Delta BiasV01 = 0.05 * 2.92286 = 5.8457$$

$$\Delta BiasV02 = 0.05 * 4.12494 = 8.2499$$

$$\Delta V10 = 0.05 * 0.000555 * 0.446531 = 4.9579$$

$$\Delta V20 = 0.05 * 0.000555 * 0.255152 = 2.8330$$

$$\Delta V_{30} = 0.05 * 0.000555 * 0.298317 = 3.3123$$

$$\Delta V_{40} = 0.05 * 0.000555 * 0.002674 = 2.9687$$

$$\Delta V_{50} = 0.05 * 0.000555 * 0.026017 = 2.8887$$

$$\Delta V_{60} = 0.05 * 0.000555 * 0.09102 = 1.0106$$

$$\Delta V_{70} = 0.05 * 0.000555 * 0.406224 = 4.5104$$

$$\Delta V_{80} = 0.05 * 0.000555 * 0.962555 = 1.0687$$

$$\Delta V_{90} = 0.05 * 0.000555 * 0.329849 = 3.6624$$

$$\Delta V_{11} = 0.05 * 2.92286 * 0.446531 = 2.6103$$

$$\Delta V_{21} = 0.05 * 2.92286 * 0.255152 = 1.4915$$

$$\Delta V_{31} = 0.05 * 2.92286 * 0.298317 = 1.7438$$

$$\Delta V_{41} = 0.05 * 2.92286 * 0.002674 = 1.5630$$

$$\Delta V_{51} = 0.05 * 2.92286 * 0.026017 = 1.5209$$

$$\Delta V_{61} = 0.05 * 2.92286 * 0.09102 = 5.3207$$

$$\Delta V_{71} = 0.05 * 2.92286 * 0.406224 = 2.3746$$

$$\Delta V_{81} = 0.05 * 2.92286 * 0.962555 = 5.6268$$

$$\Delta V_{91} = 0.05 * 2.92286 * 0.329849 = 6.9564$$

$$\Delta V_{12} = 0.05 * 4.12494 * 0.446531 = 2.6103$$

$$\Delta V_{22} = 0.05 * 4.12494 * 0.255152 = 1.4915$$

$$\Delta V_{32} = 0.05 * 4.12494 * 0.298317 = 1.7438$$

$$\Delta V_{42} = 0.05 * 4.12494 * 0.002674 = 1.5630$$

$$\Delta V_{52} = 0.05 * 4.12494 * 0.026017 = 1.5208$$

$$\Delta V_{62} = 0.05 * 4.12494 * 0.09102 = 5.3207$$

$$\Delta V_{72} = 0.05 * 4.12494 * 0.406224 = 2.3746$$

$$\Delta V_{82} = 0.05 * 4.12494 * 0.962555 = 5.6268$$

$$\Delta V_{92} = 0.05 * 4.12494 * 0.329849 = 1.9282$$

6. Lalu hitung perubahan bobot baru yang akan digunakan sebagai bobot awal perhitungan dari *looping* selanjutnya. Bobot baru *neuron W* yang menuju ke *output*:

$$\text{Bias}W_{0 \text{ new}} = -0.27089 + 0.0026314 = -0.26826$$

$$W_{00 \text{ new}} = 0.251913 + 0.0020778 = 0.25354$$

$$W_{10 \text{ new}} = 0.582666 + 0.0023538 = 0.002354$$

$$W_{20 \text{ new}} = 1.84292 + 0.002170 = 0.002117$$

Perubahan bobot *neuron* yang menuju ke *hidden layer*:

$$\text{BiasV00 new} = -0.29313 + 0.000111 = -0.2931$$

$$\text{BiasV01 new} = 0.60359 + 5.8457 = 0.603585$$

$$\text{BiasV02 new} = 0.21962 + 8.2499 = 0.219621$$

$$\text{V10 new} = 0.35726 + 4.9579 = 0.357265$$

$$\text{V20 new} = 0.357265 + 2.8330 = 0.35731$$

$$\text{V30 new} = 0.614483 + 3.3123 = 0.614483$$

$$\text{V40 new} = 0.97503 + 2.9687 = 0.97503$$

$$\text{V50 new} = 0.89 + 2.8887 = 0.89$$

$$\text{V60 new} = 0.89 + 1.0106 = 0.89$$

$$\text{V70 new} = 0.12455 + 4.5104 = 0.124553$$

$$\text{V80 new} = 0.7213 + 1.0687 = 0.721304$$

$$\text{V90 new} = 1.00056 + 3.6624 = 0.73492$$

$$\text{V11 new} = 0.00942 + 2.6103 = 0.00942$$

$$\text{V21 new} = 0.12015 + 1.4915 = 120259$$

$$\text{V31 new} = 0.59238 + 1.7438 = 0.592378$$

$$\text{V41 new} = 0.7064 + 1.5630 = 0.7064$$

$$\text{V51 new} = 0.13111 + 1.5209 = 0.131107$$

$$\text{V61 new} = 0.84643 + 5.3207 = 0.8464$$

$$\text{V71 new} = 0.91031 + 2.3746 = 0.910309$$

$$\text{V81 new} = 0.71245 + 5.6268 = 0.71245$$

$$\text{V91 new} = 0.55885 + 6.9564 = 0.814922$$

$$\text{V12 new} = 0.54899 + 2.6103 = 0.549$$

$$\text{V22 new} = 0.93535 + 1.4915 = 0.9353$$

$$\text{V32 new} = 0.67423 + 1.7438 = 0.67423$$

$$\text{V42 new} = 0.70195 + 1.5630 = 0.701953$$

$$\text{V52 new} = 0.97203 + 1.5208 = 0.9721$$

$$\text{V62 new} = 0.92228 + 5.3207 = 0.84643$$

$$\text{V72 new} = -0.2931 + 2.3746 = -0.29313$$

$$\text{V82 new} = 0.60359 + 5.6268 = 0.603585$$

$$\text{V92 new} = 0.21962 + 1.9282 = 0.22432$$

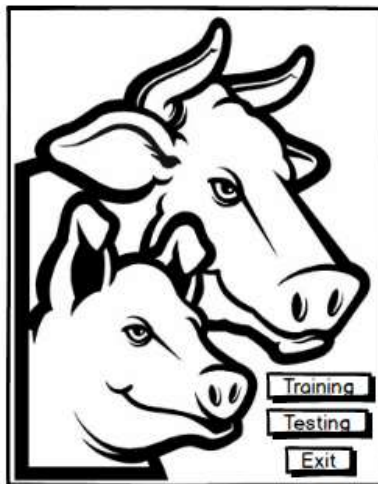
7. Setelah bobot baru ditemukan, maka akan digunakan sebagai bobot awal untuk perhitungan dari data sampel yang kedua yaitu $red = 0.56655$, $green = 0.202182$, $blue = 0.231268$, $energy = 0.002051$, $entropy = 0.027735$, $contrast = 0.344868$, $homogeneity = 0.340955$, $correlation = 0.846633$, dan $inverse\ difference\ momentum = 0.263616$, dan $target = 0$.

4.3 Perancangan *Interface*

Berikut ini merupakan rancangan *user-interface* (berupa *mockup*) dari aplikasi klasifikasi daging babi dan sapi menggunakan ekstraksi warna dan tekstur menggunakan GLCM. Pada sistem ini terbagi menjadi dua proses utama yaitu *training* dan *testing*.

4.3.1 Rancangan Tampilan Menu Utama

Tampilan awal sistem ketika sistem pertama kali dijalankan ditunjukkan pada gambar 4.6, pada tampilan awal terdapat tiga *button* yang dapat digunakan, *button* pertama merupakan *button training* untuk menuju pada *form training* sedangkan *button* kedua merupakan *button testing* untuk menu pada *form testing*. Dan yang terakhir *button exit* untuk keluar dari aplikasi. Dalam form ini terdapat dua proses utama dalam klasifikasi daging menggunakan metode *backpropagation*, yaitu *training* dan *testing*.



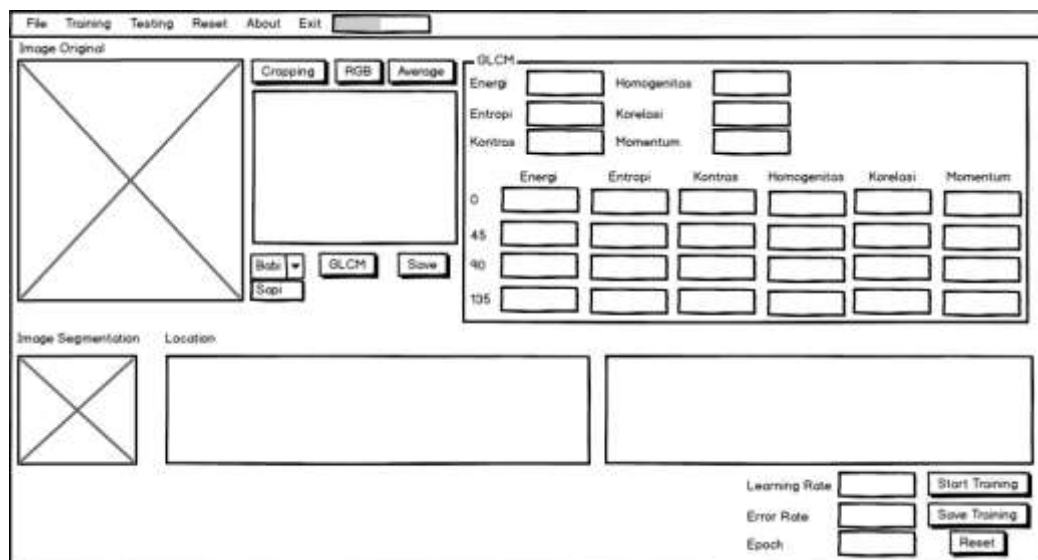
Gambar 4.6 Rancangan Tampilan *Home*

4.3.2 Rancangan Tampilan *Training*

Training data digunakan untuk mengenali pola dari objek menggunakan fitur-fitur yang sudah diambil melalui ekstraksi ciri. Pada rancangan tampilan *training*, data seluruh ekstraksi ciri ditampilkan pada *datagrid view*, kemudian untuk proses *training* data menggunakan metode *backpropagation* memerlukan

beberapa komponen antara lain *textbox* untuk *hidden layer*, *learning rate*, *error rate* dan *epoch*. Selain itu untuk memulai *training* dan menyimpan hasil diperlukan komponen *button*. Hasil *training* data akan ditampilkan pada *datagrid view*.

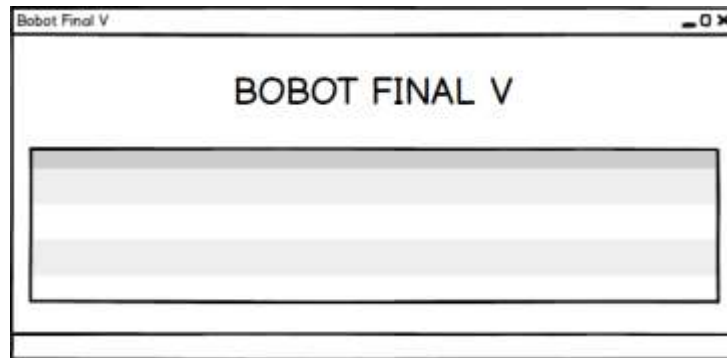
Terdapat *pre-processing* sebelum melakukan *training*. Pertama mempersiapkan proses ekstraksi warna dan ekstraksi tekstur dengan merubah ukuran citra serta menghilangkan *noise* pada citra. Pada *pre-processing* terdapat beberapa komponen yaitu menu *bar*, *button*, *picturebox*, dan *textbox*. *Button cropping* yang di gunakan saat sebelum pengambilan nilai citra RGB. Selanjutnya terdapat *button* RGB yang digunakan untuk menormalisasi nilai RGB yang akan dimasukkan ke dalam *listbox* yang menampilkan nilai RGB. Untuk ekstraksi tekstur akan menggunakan *button average* dan GLCM. *Button average* akan mengonversi citra menjadi abu-abu dan *button* GLCM akan mengambil nilai *energy*, *entropy*, *contrast*, *homogeneity*, *correlation*, dan *inverse difference momentum* yang akan dimasukkan ke dalam *datagrid view*. *Button start training* akan menjalankan metode *backpropagation* yang nilainya diambil dari *textbox* GLCM dan akan ditampilkan pada *datagrid view*. Rancang bangun tampilan training ditunjukkan pada gambar 4.7.



Gambar 4.7 Rancangan Tampilan Training

4.3.3 Tampilan View Bobot Final V Pada Training

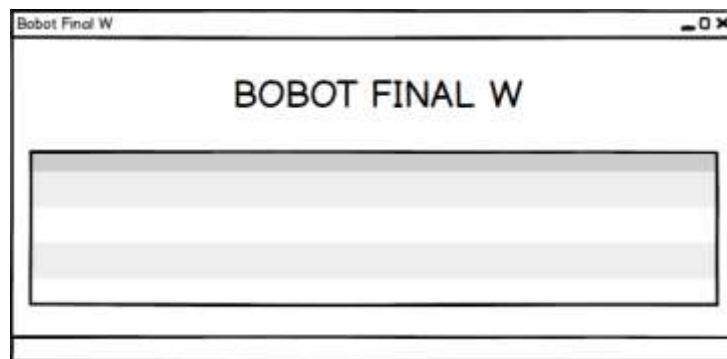
Gambar di bawah ini akan muncul jika user memilih *button start training*. Pada *window* ini akan menampilkan nilai akhir dari bobot *neuron* penghubung antara *input* (dalam hal ini X) dengan *hidden layer* (dalam hal ini Y) serta bias setelah proses *training* menggunakan metode *backpropagation* dilakukan oleh *user*.



Gambar 4.8 Rancangan Tampilan View Bobot Final V

4.3.4 Tampilan View Bobot Final W Pada Training

Gambar di bawah ini akan muncul jika user memilih *button start training*. Pada *window* ini akan menampilkan nilai akhir dari bobot *neuron* penghubung antara *hidden layer* (dalam hal ini X) dengan *output* (dalam hal ini Y) serta bias setelah proses *training* menggunakan metode *Backpropagation* dilakukan oleh *user*.



Gambar 4.9 Rancangan Tampilan View Bobot Final W

4.3.5 Rancangan Tampilan Testing

Proses *testing* dilakukan untuk mengenali daging berdasarkan fitur dan bobot pada *training* menggunakan *backpropagation*. Pada proses ini terdapat *pre-processing* juga sama seperti proses *training*, hanya saja untuk *testing* saat membuka *file open image*, nilai RGB dan GLCM akan secara otomatis muncul ke

4.3.7 Tampilan *Help* Pada Form *Testing*

Window Help Testing akan muncul jika *user* memilih menu *Help* yang ada di dalam *Menu Bar Help*. *Window* ini berisikan penjelasan fungsi-fungsi yang terdapat di dalam sistem Klasifikasi Daging Babi dan Sapi Menggunakan Ekstraksi Fitur Warna dan Tekstur Menggunakan *Gray Level Co-Occurrence Matrix*. Sehingga dapat membantu *user* dalam lebih memahami fungsi-fungsi dari menu yang ada di dalam sistem jika mengalami kesulitan.

Aplikasi ini Digunakan Untuk Mengidentifikasi Daging Babi atau Daging Sapi	
Penjelasan Menu	
File	
Open Image	Digunakan untuk memilih citra daging yang akan digunakan sebagai objek.
Open Data	Digunakan untuk memilih source nilai RGB yang sebelumnya telah disimpan.
Save Image	Digunakan untuk menyimpan citra hasil cropping.
Training	Digunakan untuk melihat form Training.
Testing	Digunakan untuk melihat form Testing
Reset	Digunakan untuk mereset semua form
About	Digunakan untuk melihat penjelasan tentang aplikasi.
Exit	Digunakan untuk keluar dari sistem
Average	Digunakan untuk konversi warna ke abu-abu
GLCM	Digunakan untuk mengambil nilai ekstraksi tekstur citra
Identifikasi	Digunakan untuk identifikasi citra

Gambar 4.12 Rancangan Tampilan *Help Form Testing*

BAB V IMPLEMENTASI

5.1 Implementasi *Home*

Pada gambar 5.1 merupakan tampilan halaman utama saat aplikasi ini di jalankan, kita dapat memilih *training*, *testing*, dan *exit*.



Gambar 5.1 Tampilan Awal

Pada tampilan ini kita dapat memilih menu *training* atau *testing* untuk menjalankan aplikasi dan bila kita tidak berkenan kita dapat memilih menu *exit*.

5.2 Implementasi *Training*

Pada implementasi *training* ini akan dijelaskan bagaimana proses *training* akan mengidentifikasi daging berdasarkan warna dan tekstur dengan menggunakan nilai RGB dan *Gray Level Co-Occurrence Matrix* menggunakan bahasa pemrograman Visual Basic.



Gambar 5.2 Tampilan *Training*

5.2.1 Proses *Input* Citra

Langkah awal yang dilakukan di dalam proses *training* yaitu *user* memasukan citra daging untuk mendeteksi daging tersebut. Langkah yang dilakukan untuk melakukan input citra sebagai berikut dan dapat dilihat pada gambar dibawah ini

- Klik *toolbar open* pada *menu bar file*. Lalu pilih *open image* untuk meng-*input*-kan citra yang akan diidentifikasi



Gambar 5.3 Tampilan Proses *Input* Citra

- Setelah *toolbar open image* di klik akan muncul sebuah *file dialog* yang dimana *user* dapat memilih citra yang akan diidentifikasi seperti pada gambar 5.4



Gambar 5.4 Tampilan *File Open Dialog*

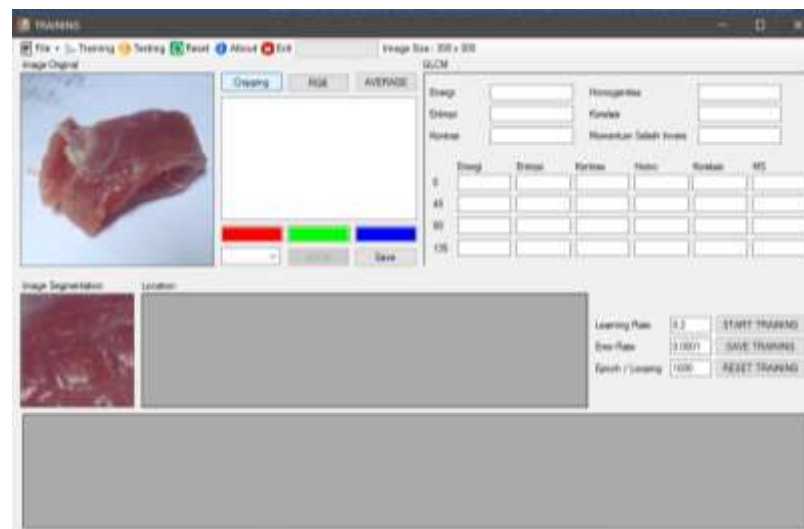
5.2.2 Pre-Processing

Langkah selanjutnya setelah citra di-input-kan ke dalam sistem yaitu melakukan proses *pre-processing* yang digunakan untuk mengambil nilai RGB yang. Langkah-langkah untuk mengambil nilai warna RGB dapat diuraikan sebagai berikut:

- Klik *button cropping* yang bertujuan untuk memilih bagian citra daging seperti pada gambar 5.6

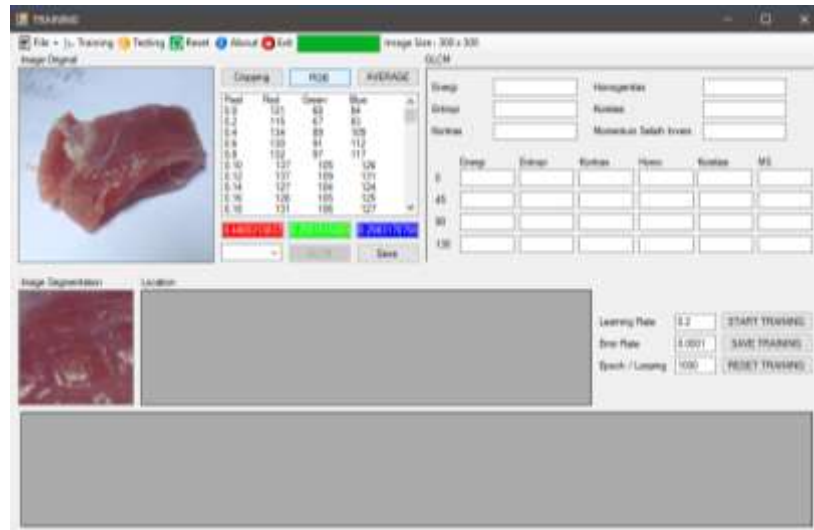


Gambar 5.5 Tampilan *Training* Saat *Input* Citra



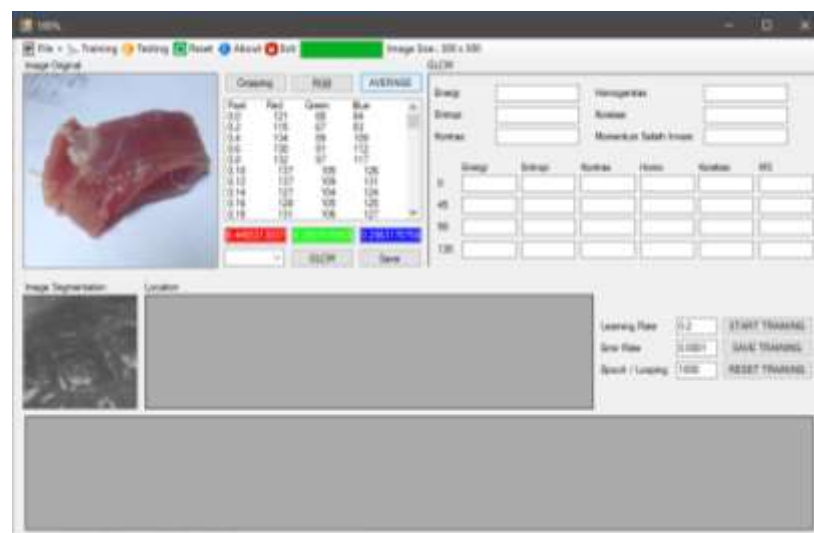
Gambar 5.6 Tampilan Hasil *Cropping*

- Selanjutnya klik *button RGB* yang bertujuan untuk mendapatkan nilai citra RGB yang akan ditampilkan pada *listbox* dan akan di normalisasi yang nilainya akan dimasukan ke dalam *textbox* seperti gambar pada 5.7



Gambar 5.7 Tampilan Ekstraksi Warna RGB

- Setelah nilai normalisasi RGB muncul pada *textbox* klik *button average* untuk mengonversi citra warna RGB menjadi warna abu-abu seperti gambar 5.8

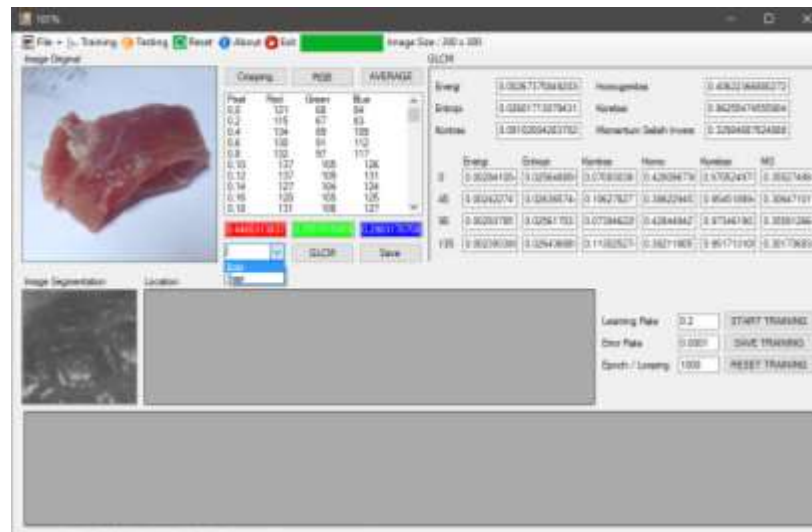


Gambar 5.8 Tampilan Hasil Konversi Warna

5.2.3 Proses Ekstraksi Fitur (GLCM)

Setelah melakukan *pre-processing* maka akan dilakukan proses ekstraksi fitur yang digunakan untuk mengidentifikasi citra daging menggunakan tekstur yang ada di permukaan daging. Langkah yang dilakukan untuk melakukan proses ekstraksi fitur dapat diuraikan sebagai berikut:

- Klik *button GLCM* yang berfungsi untuk mengekstraksi tekstur dan akan ditampilkan pada *textbox* yang berada pada *menu field*. Selanjutnya pilih klasifikasi yang akan digunakan pada *combobox*. Untuk melakukan ekstraksi tekstur citra sebagai berikut dan dapat dilihat pada gambar 5.9

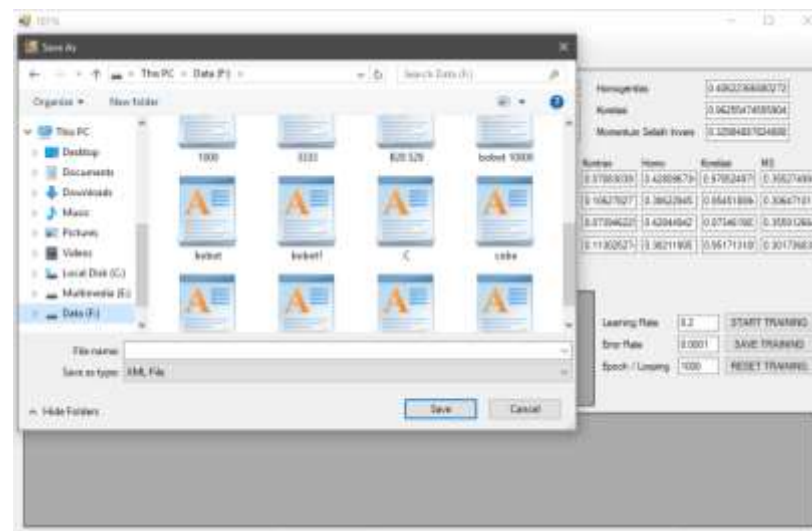


Gambar 5.9 Tampilan Hasil Dari Proses GLCM

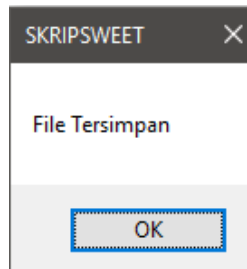
5.2.4 Proses Penyimpanan Data Set

Setelah mendapatkan nilai normalisasi RGB dan GLCM, maka lakukan *save data* dengan cara:

- Klik *button save*, lalu akan muncul *form* dimana kita akan menyimpan *dataset* yang telah dibuat seperti gambar 5.10. dan akan muncul kotak peringatan bahwa data telah tersimpan seperti pada gambar 5.11.



Gambar 5.10 Tampilan Save Data



Gambar 5.11 Tampilan Kotak Peringatan

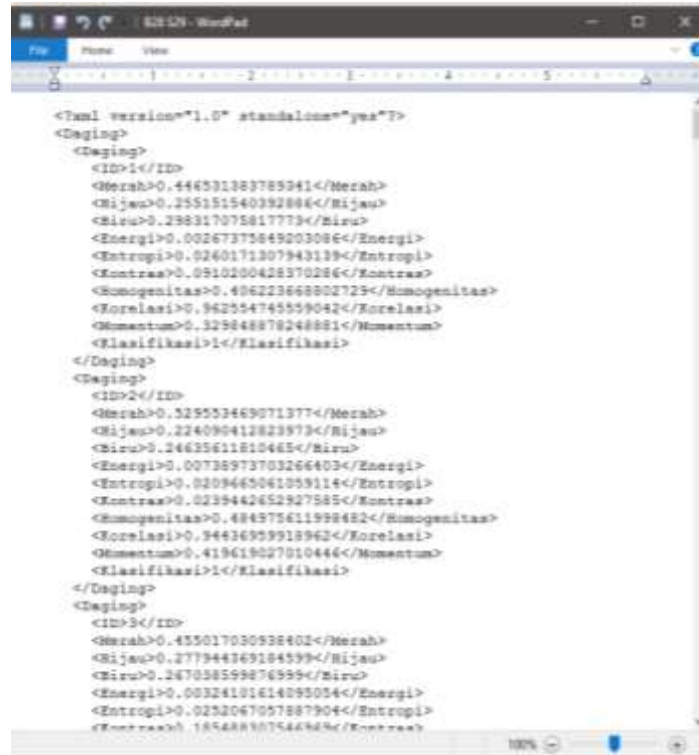
- Setelah data set disimpan di dalam HDD maka data set akan ditampilkan di dalam DGV. Di dalam DGV terdapat seluruh nilai yang telah diproses oleh ekstraksi warna dan tekstur, serta nilai data set yang telah di proses pada seelumnya, seperti gambar 5.12



Gambar 5.12 Tampilan Data Dalam DGV

5.2.5 Proses Training Backpropagation

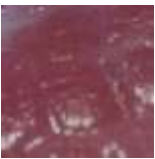


Pada proses ini *database* yang digunakan merupakan *file XML* dan pemrograman di dalam VB sebagai pemroses data. Hal ini dilakukan karena data yang diolah tidak berukuran besar dan dapat terorganisir dengan hanya menggunakan XML atau bisa disebut *flatfile database*. Setelah didapat nilai normalisasi RGB dan GLMC dari setiap sampel maka akan dimasukkan ke dalam XML. Berikut merupakan tahap dari setiap prosesnya:



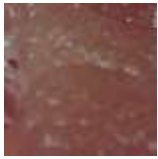










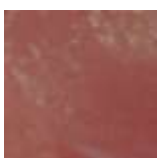
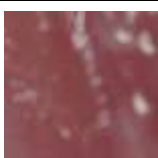



Gambar 5.13 Data Daging yang Dimasukan ke Dalam XML




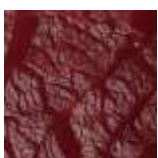
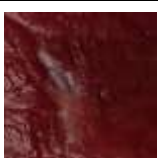



- Selanjutnya data akan di-load kedalam *datagrid* view untuk digunakan sebagai *input* dari proses klasifikasi daging. Dalam hal ini nilai *input* RGB dan GLCM serta target klasifikasi terdiri dari 40 data daging dengan rincian 20 babi dan 20 sapi.


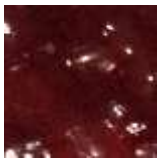

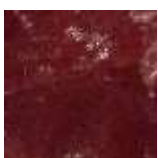

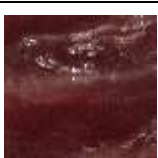

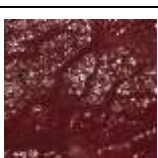
Tabel 5.1 Data *Training*






No	Citra Daging	NR	NG	NB	ener jum	entro jum	konj um	homo jum	korj um	msj um	Tar get
1		0.44 653	0.25 515	0.29 831	0.00 267	0.026 01	0.09 102	0.406 22	0.96 255	0.32 984	1
2		0.52 955	0.22 409	0.24 635	0.00 738	0.020 96	0.02 394	0.484 97	0.94 436	0.41 961	1
3		0.45 501	0.27 794	0.26 703	0.00 324	0.025 20	0.18 548	0.398 10	0.86 768	0.32 194	1

No	Citra Daging	NR	NG	NB	ener jum	entro jum	konj um	homo jum	korj um	msj um	Tar get
4		0.46 771	0.23 499	0.29 729	0.00 186	0.028 16	0.18 236	0.404 07	0.97 307	0.33 364	1
5		0.46 721	0.25 839	0.27 439	0.00 186	0.028 16	0.18 236	0.404 07	0.97 307	0.33 364	1
6		0.44 173	0.29 325	0.26 501	0.00 325	0.024 69	0.05 158	0.455 96	0.95 578	0.38 719	1
7		0.48 987	0.23 442	0.27 570	0.00 774	0.020 84	0.02 107	0.508 98	0.93 419	0.44 995	1
8		0.49 354	0.22 131	0.28 513	0.00 794	0.020 88	0.01 722	0.514 95	0.94 555	0.45 683	1
9		0.48 923	0.25 925	0.25 150	0.00 218	0.026 11	0.07 622	0.412 70	0.95 056	0.33 617	1
10		0.45 703	0.28 263	0.26 033	0.00 332	0.024 36	0.02 018	0.524 37	0.99 123	0.46 946	1
11		0.45 648	0.27 582	0.26 768	0.00 197	0.026 37	0.04 374	0.459 98	0.98 778	0.39 145	1

No	Citra Daging	NR	NG	NB	ener jum	entro jum	konj um	homo jum	korj um	msj um	Tar get
12		0.42 107	0.28 136	0.29 756	0.00 268	0.026 31	0.09 615	0.434 07	0.99 173	0.36 246	1
13		0.53 452	0.22 236	0.24 310	0.00 349	0.025 48	0.37 776	0.419 34	0.87 988	0.34 955	1
14		0.50 025	0.25 325	0.24 649	0.00 422	0.023 48	0.02 737	0.493 22	0.97 822	0.43 106	1
15		0.46 980	0.27 656	0.25 363	0.00 505	0.022 35	0.02 595	0.484 75	0.94 521	0.41 984	1
16		0.44 953	0.26 895	0.28 151	0.00 376	0.024 30	0.04 174	0.458 83	0.97 646	0.38 973	1
17		0.49 705	0.24 006	0.26 287	0.00 248	0.026 46	0.14 145	0.390 91	0.92 017	0.31 495	1
18		0.51 004	0.25 348	0.23 646	0.00 624	0.022 26	0.03 670	0.469 38	0.92 533	0.40 221	1
19		0.47 343	0.27 888	0.24 767	0.00 303	0.024 65	0.03 334	0.463 99	0.96 927	0.39 483	1

No	Citra Daging	NR	NG	NB	ener jum	entro jum	konj um	homo jum	korj um	msj um	Tar get
20		0.43 943	0.29 019	0.27 036	0.00 303	0.024 65	0.03 334	0.463 99	0.96 927	0.39 483	1
21		0.56 654	0.20 218	0.23 126	0.00 193	0.028 09	0.37 301	0.341 97	0.88 035	0.26 575	0
22		0.61 765	0.17 162	0.21 071	0.00 193	0.028 09	0.37 301	0.341 97	0.88 035	0.26 575	0
23		0.53 639	0.21 798	0.24 561	0.00 129	0.030 11	0.39 743	0.314 42	0.91 949	0.24 081	0
24		0.61 582	0.18 404	0.20 013	0.00 339	0.026 19	0.12 633	0.433 49	0.96 114	0.36 755	0
25		0.59 985	0.18 804	0.21 209	0.00 450	0.025 09	0.05 445	0.498 39	0.98 520	0.44 186	0
26		0.54 715	0.20 076	0.25 208	0.00 230	0.025 92	0.19 542	0.406 63	0.91 546	0.32 941	0
27		0.52 081	0.22 538	0.25 380	0.00 177	0.027 18	0.28 209	0.359 22	0.88 601	0.27 568	0

No	Citra Daging	NR	NG	NB	ener jum	entro jum	konj um	homo jum	korj um	msj um	Tar get
28		0.49 586	0.23 864	0.26 548	0.00 191	0.027 52	0.35 771	0.359 66	0.93 156	0.27 694	0
29		0.61 449	0.18 435	0.20 115	0.00 354	0.025 25	0.24 877	0.426 39	0.96 257	0.35 487	0
30		0.52 063	0.23 309	0.24 626	0.00 147	0.027 67	0.22 431	0.351 88	0.91 151	0.26 643	0
31		0.59 641	0.18 603	0.21 755	0.00 257	0.026 09	0.21 865	0.382 55	0.91 783	0.30 230	0
32		0.52 063	0.23 309	0.24 626	0.00 147	0.027 67	0.22 431	0.351 88	0.91 151	0.26 643	0
33		0.51 405	0.23 939	0.24 654	0.00 153	0.028 21	0.29 343	0.350 66	0.92 940	0.26 782	0
34		0.55 229	0.20 024	0.24 746	0.00 319	0.025 39	0.57 463	0.380 64	0.86 796	0.30 258	0
35		0.52 335	0.23 316	0.24 348	0.00 142	0.029 82	0.97 771	0.294 91	0.86 938	0.21 531	0

No	Citra Daging	NR	NG	NB	ener jum	entro jum	konj um	homo jum	korj um	msj um	Tar get
36		0.64 544	0.16 828	0.18 627	0.00 364	0.024 94	0.11 632	0.420 55	0.89 046	0.34 984	0
37		0.65 388	0.16 217	0.18 394	0.00 364	0.024 94	0.11 632	0.420 55	0.89 046	0.34 984	0
38		0.66 748	0.15 251	0.18 000	0.00 478	0.024 64	0.14 762	0.458 27	0.91 666	0.39 745	0
39		0.63 751	0.16 589	0.19 658	0.00 450	0.025 59	0.30 432	0.435 57	0.88 743	0.37 602	0
40		0.63 037	0.17 944	0.19 017	0.00 450	0.025 59	0.30 432	0.435 57	0.88 743	0.37 602	0

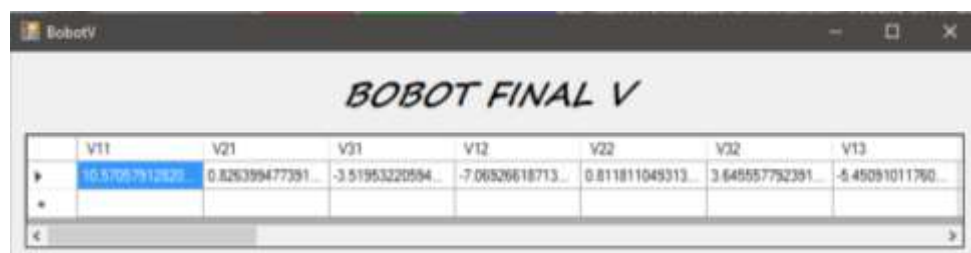
Pada tabel di atas, target diperlukan sebagai acuan sistem dalam menentukan bobot terbaik. Karena bobot terbaik yang akan didapat sistem akan bergantung pada selisih nilai keluaran dengan target.

- Selanjutnya dalam tahap ini akan di inisiasi input dari bobot-bobot *neuron* serta bias di dalam metode *backpropagation* yang digunakan.

<i>Learning Rate</i>	= 0.2	V14	= 0.4
Bias V00	= -0.3	V15	= 0.2
Bias V01	= 0.6	V16	= 0.8
Bias v02	= 0.2	V17	= 0.5
V11	= 0.9	V18	= 0.7
V12	= 0.1	V19	= 0.6
V13	= 0.3	V21	= 0.9

V22	= 0.6	V34	= 0.5
V23	= 0.7	V35	= 0.6
V24	= 0.1	V36	= 0.2
V25	= 0.2	V37	= 0.4
V26	= 0.5	V38	= 0.3
V27	= 0.3	V39	= 0.1
V28	= 0.8	Bias W	= -0.5
V29	= 0.4	W0	= -0.3
V31	= 0.9	W1	= 0.9
V32	= 0.8	W2	= -0.1
V33	= 0.7		

- Setelah bobot-bobot *neuron* di inisiasi, maka selanjutnya memulai perhitungan metode *backpropagation*. Terdapat tiga fase dalam metode tersebut, yang pertama fase maju kemudian fase propagasi mundur dan yang terakhir pembobotan baru.
- Dalam perhitungan akan dilakukan maksimal sebanyak jumlah *epoch* yang sudah ditentukan. Dan akan berhenti jika *error rate* (*error minimal*) sama dengan RMSE dari data tersebut. Jika telah memenuhi salah satu dari kondisi diatas maka bobot terbaik akan ditampilkan. Seperti pada gambar 5.14 dan 5.15



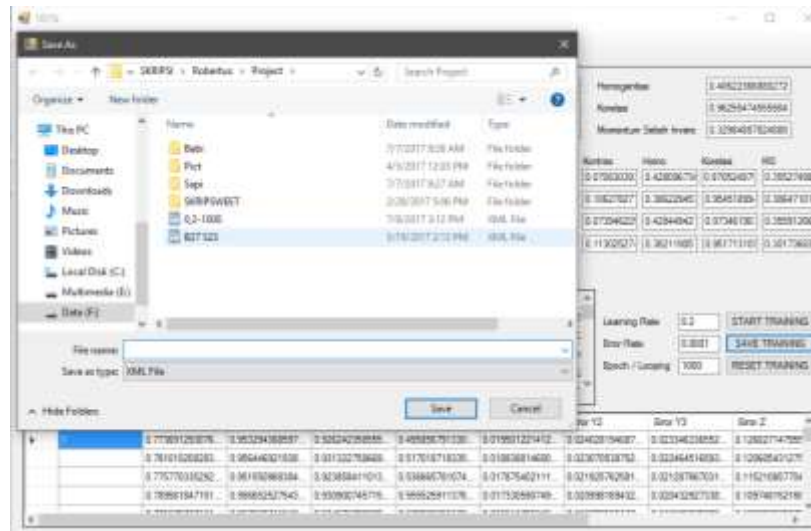
	V11	V21	V31	V12	V22	V32	V13
	10.57057912835	0.826399477391	-3.51953220594	-7.06526618713	0.811811049313	3.648557752391	-5.45091011760

Gambar 5.14 Tampilan Bobot *Final V*


	W1	W2	W3	Wbias
	18.7798712407	1.8398804910589	7.433578368302	2.344333023750

Gambar 5.15 Tampilan Bobot *Final W*

- Setelah proses perhitungan *training* berhenti dan bobot terbaik telah ditampilkan, maka data bobot terbaik tersebut harus disimpan.
- Klik *button save training*, lalu akan muncul *form* dimana kita akan menyimpan data set yang telah dibuat seperti gambar 5.16.



Gambar 5.16 Tampilan *Save Training*

5.3 Implementasi *Testing*

Pada implementasi *testing* akan dijelaskan bagaimana proses *testing* akan mengidentifikasi daging berdasarkan warna dan tekstur dengan menggunakan nilai RGB dan *Gray Level Co-Occurrence Matrix* menggunakan bahasa pemrograman *Visual Basic*.



Gambar 5.17 Tampilan *Testing*

5.3.1 Proses *Input Citra*

Proses *input* citra merupakan langkah pertama dalam memulai klasifikasi daging pada *testing* dengan komponen *Red, Green, Blue* (RGB). Klik *toolbar open* pada *menu bar file*. Lalu pilih *open image* untuk meng-*input*-kan citra yang akan diidentifikasi. Setelah *toolbar open image* di klik akan muncul sebuah *file dialog*

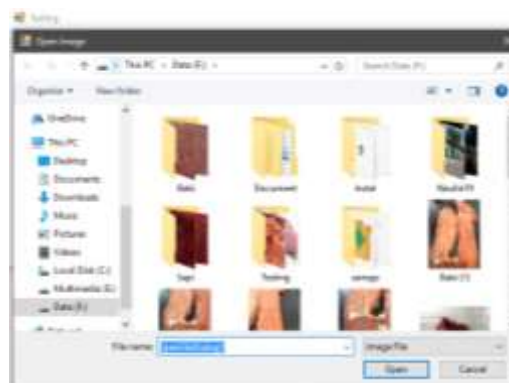
yang dimana *user* dapat memilih citra yang akan diidentifikasi Langkah yang dilakukan untuk melakukan *input* citra sebagai berikut dan dapat dilihat pada gambar dibawah ini:

- Klik *toolbar* open pada menu bar file. Lalu pilih *open image* untuk meng-*input*-kan citra yang akan diidentifikasi



Gambar 5.18 Tampilan Proses *Input* Citra

- Setelah *toolbar open image* di klik akan muncul sebuah *file dialog* yang dimana *user* dapat memilih citra yang akan diidentifikasi seperti pada gambar 5.20.



Gambar 5.19 Tampilan *File Open Dialog*



Gambar 5.20 Tampilan *Input* Citra

5.3.2 Proses Ekstraksi Tekstur

Setelah citra di-input-kan dan mengidentifikasi warna, citra akan dikonversi menjadi warna abu-abu dan selanjutnya akan diidentifikasi teksturnya menggunakan *button* GLCM seperti gambar 5.22



Gambar 5.21 Tampilan Konversi Warna



Gambar 5.22 Tampilan Ekstraksi Tekstur GLCM

5.3.3 Proses Memasukan Data Bobot Hasil *Training*

Bobot pada implementasi *testing* dimasukkan dengan cara Klik *toolbar open* pada *menu bar file*. Lalu pilih *open data* untuk meng-input-kan data yang akan digunakan. Setelah *toolbar open data* di pilih akan muncul sebuah file dialog yang dimana *user* dapat memilih citra yang akan diidentifikasi Langkah yang dilakukan untuk melakukan input citra sebagai berikut dan dapat dilihat pada gambar dibawah ini:

- Klik *toolbar open* pada *menu bar file*. Lalu pilih *open data* untuk meng-input-kan data yang akan diidentifikasi



Gambar 5.23 Tampilan Proses *Input* Data

- Setelah *toolbar open data* di klik akan muncul sebuah *file dialog* yang dimana *user* dapat memilih citra yang akan diidentifikasi seperti pada gambar 5.24

Gambar 5.24 Tampilan *File Open Dialog*Gambar 5.25 Tampilan *Input Data*

5.3.4 Proses Klasifikasi Daging

Pada tahap ini merupakan tahap terakhir dalam implementasi *testing*. Yaitu proses perhitungan menggunakan *backpropagation* berdasarkan bobot yang telah didapat pada proses *training* dan telah di-*input*-kan pada form *testing* seperti gambar 5.23.

Gambar 5.26 Tampilan Hasil *Testing*

5.4 Tampilan Help

Window help akan tampil seperti gambar di bawah ini jika memilih menu tersebut yang terdapat pada *menu bar Help*. Pada *window help testing* informasi

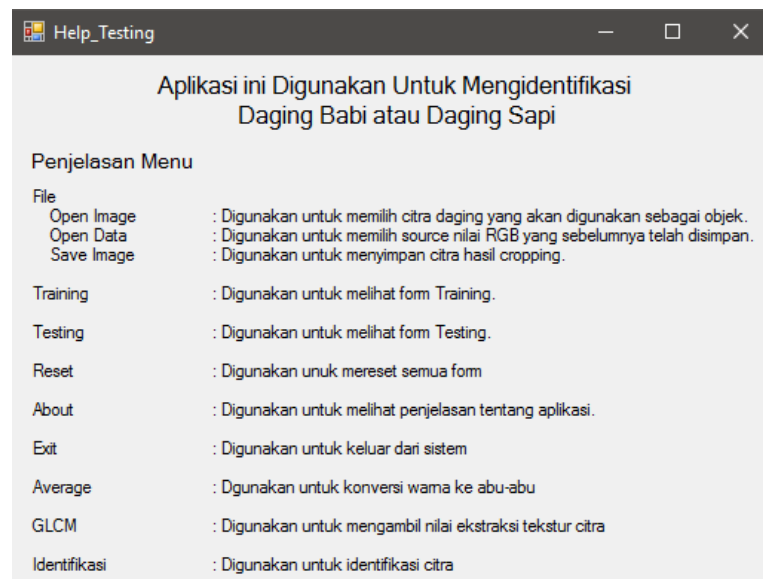
yang ditampilkan berbeda dengan yang terdapat pada sistem *training* dan *testing*, dikarenakan menyesuaikan dengan menu-menu yang terdapat di dalamnya.

5.4.1 Tampilan *Help Training*



Gambar 5.27 Window *Help Training*

5.4.2 Tampilan *Help Testing*



Gambar 5.28 Window *Help Testing*

BAB VI

PENGUJIAN DAN PEMBAHASAN

6.1 Pengujian

Pada bab ini dilakukan pengujian setelah dilakukan proses implementasi sistem. Terdapat dua macam pengujian yang akan dilakukan yaitu pengujian sistem dan pengujian akurasi. Pengujian sistem digunakan untuk mengetahui apakah sistem sudah berjalan dengan fungsinya. Untuk pengujian akurasi bertujuan untuk mengetahui seberapa besar tingkat akurasi aplikasi klasifikasi daging sapi dan babi berdasarkan warna dan tekstur menggunakan Jaringan Syaraf Tiruan dengan metode *Backpropagation*

6.1.1 Pengujian Sistem

Pengujian sistem ini dilakukan dengan cara menjalankan aplikasi secara detail pada setiap menu yang ada, dengan tujuan mengetahui fitur mana yang sudah berfungsi dengan baik maupun tidak berfungsi, yang tujuan akhirnya adalah berjalan sesuai dengan fungsinya. Terdapat tiga pengujian sistem yaitu pengujian *home*, pengujian *training* dan pengujian *testing*. Pengujian sistem ini menggunakan metode *blackbox*. Metode ini memungkinkan adanya pengembangan untuk melatih seluruh fungsi pada sistem. Metode ini digunakan untuk mendemonstrasikan jalannya aplikasi dan menemukan kesalahan saat aplikasi dijalankan. Dengan menggunakan metode ini dapat dinilai apakah input yang diterima dan *output* yang dihasilkan sudah tepat atau belum.

- Pengujian Sistem *Home*

Tabel 6.1 Pengujian *Home*

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Klik <i>Button Training</i>	Muncul <i>form Training</i> ,	Sesuai harapan	Berhasil
2.	Klik <i>Button Testing</i>	Muncul <i>form Testing</i>	Sesuai harapan	Berhasil

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
3.	Klik <i>Button Exit</i>	<i>User</i> akan keluar dari sistem	Sesuai harapan	Berhasil

- Pengujian Sistem *Training*

Tabel 6.2 Pengujian Sistem *Training*

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Tab menu <i>file</i> . Klik toolbar <i>Open</i> , lalu klik subbar <i>Open Image</i>	Akan muncul <i>file dialog</i> yang dimana <i>user</i> dapat memilih citra yang akan diidentifikasi	Sesuai harapan	Berhasil
2.	Tab menu <i>file</i> . Klik toolbar <i>open</i> , lalu klik subbar <i>Open data</i>	Akan muncul <i>file dialog</i> yang dimana <i>user</i> dapat memilih data yang akan digunakan	Sesuai harapan	Berhasil
3.	Tab menu <i>file</i> . Klik toolbar <i>save</i> , lalu klik subbar <i>Save Image</i>	Akan muncul <i>file dialog</i> yang dimana <i>user</i> dapat menyimpan <i>Image</i> tersebut	Sesuai harapan	Berhasil

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
		Jika <i>error</i> akan muncul <i>messagebox</i> “Gambar Tidak Boleh Kosong”	Sesuai harapan	Berhasil
4.	Tab menu <i>file</i> . Klik <i>toolbar save</i> , lalu klik <i>subbar Save Data</i>	Akan muncul <i>file dialog</i> yang dimana <i>user</i> dapat menyimpan <i>data</i> tersebut Jika <i>error</i> akan muncul <i>messagebox</i> “Data Tidak Boleh Kosong”	Sesuai harapan Sesuai harapan	Berhasil Berhasil
5.	Klik <i>tab menu Testing</i>	Memunculkan <i>form testing</i> dan <i>form training</i> tersembunyi	Sesuai harapan	Berhasil
6.	Klik <i>tab menu reset</i>	Me-reset semua data, gambar yang ada pada <i>form training</i>	Sesuai harapan	Berhasil

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
7.	Klik <i>tab menu About</i>	Memunculkan <i>form help</i> yang berisi tentang kegunaan <i>button</i>	Sesuai harapan	Berhasil
8.	Klik <i>tab menu exit</i>	Mengeluarkan <i>user</i> dari sistem	Sesuai harapan	Berhasil
9.	Klik <i>button cropping</i>	Memilih bagian citra yang akan diidentifikasi Jika <i>error</i> akan muncul <i>messagebox</i> “Gambar Tidak Boleh Kosong”	Sesuai harapan Sesuai harapan	Berhasil Berhasil
10.	Klik <i>button RGB</i>	Mengekstraksi warna dengan mengambil nilai <i>red, green, blue</i> pada citra yang telah di <i>cropping</i>	Sesuai harapan	Berhasil

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
		Jika <i>error</i> akan muncul <i>messagebox</i> “Gambar Tidak Boleh Kosong”	Sesuai harapan	Berhasil
11.	Klik <i>button Average</i>	Mengkonversi citra RGB menjadi abu-abu	Sesuai harapan	Berhasil
		Jika <i>error</i> akan muncul <i>messagebox</i> “Gambar Tidak Boleh Kosong”	Sesuai harapan	Berhasil
12.	Klik <i>button GLCM</i>	Mengekstraksi fitur tekstur untuk mendapatkan nilai parameter GLCM	Sesuai harapan	Berhasil
		Jika <i>error</i> akan muncul <i>messagebox</i> “Gambar	Sesuai harapan	Berhasil

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
		Tidak Boleh Kosong”		
13.	Klik <i>button Save</i>	Menyimpan data ke dalam <i>datagrid view</i> Jika <i>error</i> akan muncul <i>messagebox</i> “Data Tidak Boleh Kosong”	Sesuai harapan Sesuai harapan	Berhasil Berhasil
14.	Klik <i>Start Training</i>	Melakukan <i>training data</i> untuk mendapatkan bobot terbaik Jika <i>error</i> akan muncul <i>messagebox</i> “Data Tidak Boleh Kosong”	Sesuai harapan Sesuai harapan	Berhasil Berhasil
15.	Klik <i>Save Training</i>	Menyimpan data ke dalam <i>HDD</i> Jika <i>error</i> akan muncul	Sesuai harapan Sesuai harapan	Berhasil Berhasil

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
		<i>messagebox</i> “Data Tidak Boleh Kosong”		
16.	Klik <i>Reset Training</i>	Me- <i>reset</i> data, yang ada pada <i>datagrid view</i>	Sesuai harapan	Berhasil

- Pengujian Sistem *Testing*

Tabel 6.3 Pengujian Sistem *Testing*

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	<i>Tab menu file.</i> Klik <i>toolbar Open</i> , lalu klik <i>subbar Open Image</i>	Akan muncul <i>file dialog</i> yang dimana <i>user</i> dapat memilih citra yang akan diidentifikasi	Sesuai harapan	Berhasil
2.	<i>Tab menu file.</i> Klik <i>toolbar open</i> , lalu klik <i>subbar Open data</i>	Akan muncul <i>file dialog</i> yang dimana <i>user</i> dapat memilih data yang akan digunakan	Sesuai harapan	Berhasil
3.	<i>Tab menu file.</i> Klik <i>toolbar save</i> , lalu klik <i>subbar Save Image</i>	Akan muncul <i>file dialog</i> yang dimana <i>user</i> dapat menyimpan <i>Image</i> tersebut	Sesuai harapan	Berhasil

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
		Jika <i>error</i> akan muncul <i>messagebox</i> “Gambar Tidak Boleh Kosong”	Sesuai harapan	Berhasil
4.	<i>Tab menu file.</i> Klik <i>toolbar save</i> , lalu klik <i>subbar Save Data</i>	Akan muncul <i>file dialog</i> yang dimana <i>user</i> dapat menyimpan <i>data</i> tersebut Jika <i>error</i> akan muncul <i>messagebox</i> “Data Tidak Boleh Kosong”	Sesuai harapan Sesuai harapan	Berhasil Berhasil
5.	Klik <i>tab menu Testing</i>	Memunculkan <i>form testing</i> dan <i>form training</i> tersembunyi	Sesuai harapan	Berhasil
6.	Klik <i>tab menu reset</i>	Me-reset semua data, gambar yang ada pada <i>form training</i>	Sesuai harapan	Berhasil
7.	Klik <i>tab menu About</i>	Memunculkan <i>form help</i> yang berisi tentang kegunaan <i>button</i>	Sesuai harapan	Berhasil

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
8.	Klik <i>tab menu exit</i>	Mengeluarkan <i>user</i> dari sistem	Sesuai harapan	Berhasil
9.	Klik <i>button Average</i>	Mengkonversi citra RGB menjadi abu-abu Jika <i>error</i> akan muncul <i>messagebox</i> “Gambar Tidak Boleh Kosong”	Sesuai harapan Sesuai harapan	Berhasil Berhasil
10.	Klik <i>button GLCM</i>	Mengekstraksi fitur tekstur untuk mendapatkan nilai parameter GLCM Jika <i>error</i> akan muncul <i>messagebox</i> “Gambar Tidak Boleh Kosong”	Sesuai harapan Sesuai harapan	Berhasil Berhasil
11.	Identifikasi	Mengidentifikasi daging sapi atau daging babi Jika <i>error</i> akan muncul <i>messagebox</i>	Sesuai harapan Sesuai harapan	Berhasil Berhasil

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
		“Data Tidak Boleh Kosong”		






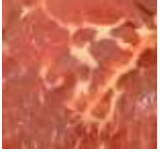
6.1.2 Pengujian Akurasi















Pengujian akurasi ini digunakan untuk menguji seberapa besar sistem klasifikasi daging babi dan sapi dengan menerapkan jaringan syaraf tiruan metode *Backpropagation*.





Pada pengujian ini menggunakan sebanyak 12 sampel daging yang terdiri dari 6 daging babi dan 6 daging sapi. Berikut merupakan data sampel testing beserta hasil pengujian daging menggunakan metode *Backpropagation*.

Tabel dibawah ini menunjukkan hasil identifikasi daging dengan menggunakan *learning rate* 0,1 dan *epoch* sebanyak 500 dengan citra berukuran 300x300 dan rata-rata akurasi dari proses tersebut adalah **83,3%**.

Tabel 6.4 Pengujian Daging Dengan *Learning Rate* 0,1 dan *Epoch* 500

No.	Citra Daging	Citra Cropping	Identifikasi Visual	Identifikasi Sistem	Hasil	Nilai Output
1.			Daging Sapi (Foto Dari Internet)	Sapi	Benar	0.00498 2925031 37839
2.			Daging Sapi (Potongan Melintang)	Sapi	Benar	0.03517 2059933 1675
3.			Daging Sapi (Degan Lemak Dominan)	Sapi	Benar	0.04443 8602695 8313

No.	Citra Daging	Citra Cropping	Identifikasi Visual	Identifikasi Sistem	Hasil	Nilai Output
4.			Daging Sapi (Cahaya Redup)	Sapi	Benar	0.25300 5721688 253
5.			Daging Sapi (Cahaya Maximal)	Sapi	Benar	0.05300 4874785 7912
6.			Daging Sapi (Bagian Paha)	Sapi	Benar	0.02760 0107829 0195
7.			Daging Babi (Foto Dari Internet)	Babi	Benar	0.96848 5842220 648
8.			Daging Babi (Potongan Melintang)	Babi	Benar	0.87802 9468944 563
9.			Daging Babi (Dengan Lemak Dominan)	Babi	Benar	0.99160 7085294 398
10.			Daging Babi (Cahaya Redup)	Babi	Benar	0.86086 5236637 565

No.	Citra Daging	Citra Cropping	Identifikasi Visual	Identifikasi Sistem	Hasil	Nilai Output
11.			Daging Babi (Cahaya Maximal)	Sapi	Salah	0.27402 5929885 813
12.			Daging Babi (Bagian Paha)	Sapi	Salah	0.23900 5014679 098

Tabel dibawah ini menunjukkan hasil identifikasi daging dengan menggunakan *learning rate* 0,1 dan *epoch* sebanyak 1000 dengan citra berukuran 300x300 dan rata-rata akurasi dari proses tersebut adalah **91,67%**.

Tabel 6.5 Pengujian Daging Dengan *Learning Rate* 0,1 dan *Epoch* 1000

No.	Citra Daging	Identifikasi Visual	Identifikasi Sistem	Keterangan	Nilai Output
1.	Gambar 1	Daging Sapi	Sapi	Benar	0.00028859
2.	Gambar 2	Daging Sapi	Sapi	Benar	0.00658375
3.	Gambar 3	Daging Sapi	Sapi	Benar	0.00385820
4.	Gambar 4	Daging Sapi	Sapi	Benar	0.12600274
5.	Gambar 5	Daging Sapi	Sapi	Benar	0.00580625
6.	Gambar 6	Daging Sapi	Sapi	Benar	0.00689634
7.	Gambar 7	Daging Babi	Babi	Benar	0.99879271
8.	Gambar 8	Daging Babi	Babi	Benar	0.98589005
9.	Gambar 9	Daging Babi	Babi	Benar	0.99985862
10.	Gambar 10	Daging Babi	Babi	Benar	0.97770759
11.	Gambar 11	Daging Babi	Sapi	Salah	0.10088771
12.	Gambar 12	Daging Babi	Babi	Benar	0.56589000

Tabel dibawah ini menunjukkan hasil identifikasi daging dengan menggunakan *learning rate* 0,1 dan *epoch* sebanyak 3000 dengan citra berukuran 300x300 dan rata-rata akurasi dari proses tersebut adalah **91,67%**.

Tabel 6.6 Pengujian Daging Dengan *Learning Rate* 0,1 dan *Epoch* 3000

No.	Citra Daging	Identifikasi Visual	Identifikasi Sistem	Keterangan	Nilai Output
1.	Gambar 1	Daging Sapi	Sapi	Benar	1.2175E-06
2.	Gambar 2	Daging Sapi	Sapi	Benar	8.5939E-05
3.	Gambar 3	Daging Sapi	Sapi	Benar	5.5217E-05
4.	Gambar 4	Daging Sapi	Sapi	Benar	0.03911523
5.	Gambar 5	Daging Sapi	Sapi	Benar	0.00017908
6.	Gambar 6	Daging Sapi	Sapi	Benar	0.00082882
7.	Gambar 7	Daging Babi	Babi	Benar	0.99999248
8.	Gambar 8	Daging Babi	Babi	Benar	0.9997135
9.	Gambar 9	Daging Babi	Babi	Benar	0.99999936
10.	Gambar 10	Daging Babi	Babi	Benar	0.99948840
11.	Gambar 11	Daging Babi	Sapi	Salah	0.02023179
12.	Gambar 12	Daging Babi	Babi	Benar	0.94575767

Tabel dibawah ini menunjukkan hasil identifikasi daging dengan menggunakan *learning rate* 0,1 dan *epoch* sebanyak 5000 dengan citra berukuran 300x300 dan rata-rata akurasi dari proses tersebut adalah **91,67%**.

Tabel 6.7 Pengujian Daging Dengan *Learning Rate* 0,1 dan *Epoch* 5000

No.	Citra Daging	Identifikasi Visual	Identifikasi Sistem	Keterangan	Nilai Output
1.	Gambar 1	Daging Sapi	Sapi	Benar	1.1851E-07
2.	Gambar 2	Daging Sapi	Sapi	Benar	1.1167E-05
3.	Gambar 3	Daging Sapi	Sapi	Benar	8.0293E-06
4.	Gambar 4	Daging Sapi	Sapi	Benar	0.01969044
5.	Gambar 5	Daging Sapi	Sapi	Benar	3.4802E-05
6.	Gambar 6	Daging Sapi	Sapi	Benar	0.00024717
7.	Gambar 7	Daging Babi	Babi	Benar	0.99999905
8.	Gambar 8	Daging Babi	Babi	Benar	0.99995373
9.	Gambar 9	Daging Babi	Babi	Benar	0.99999992
10.	Gambar 10	Daging Babi	Babi	Benar	0.99991450

No.	Citra Daging	Identifikasi Visual	Identifikasi Sistem	Keterangan	Nilai Output
11.	Gambar 11	Daging Babi	Sapi	Salah	0.00918475
12.	Gambar 12	Daging Babi	Babi	Benar	0.98160769

Tabel dibawah ini menunjukkan hasil identifikasi daging dengan menggunakan *learning rate* 0,2 dan *epoch* sebanyak 500 dengan citra berukuran 300x300 dan rata-rata akurasi dari proses tersebut adalah **91,67%**.

Tabel 6.8 Pengujian Daging Dengan *Learning Rate* 0,2 dan *Epoch* 500

No.	Citra Daging	Identifikasi Visual	Identifikasi Sistem	Keterangan	Nilai Output
1.	Gambar 1	Daging Sapi	Sapi	Benar	1.0201E-07
2.	Gambar 2	Daging Sapi	Sapi	Benar	9.6093E-06
3.	Gambar 3	Daging Sapi	Sapi	Benar	6.9270E-06
4.	Gambar 4	Daging Sapi	Sapi	Benar	0.01773697
5.	Gambar 5	Daging Sapi	Sapi	Benar	3.0235E-05
6.	Gambar 6	Daging Sapi	Sapi	Benar	0.00021586
7.	Gambar 7	Daging Babi	Babi	Benar	0.99999915
8.	Gambar 8	Daging Babi	Babi	Benar	0.99995816
9.	Gambar 9	Daging Babi	Babi	Benar	0.99999993
10.	Gambar 10	Daging Babi	Babi	Benar	0.99992214
11.	Gambar 11	Daging Babi	Sapi	Salah	0.00830028
12.	Gambar 12	Daging Babi	Babi	Benar	0.98200501

Tabel dibawah ini menunjukkan hasil identifikasi daging dengan menggunakan *learning rate* 0,2 dan *epoch* sebanyak 1000 dengan citra berukuran 300x300 dan rata-rata akurasi dari proses tersebut adalah **83,3%**.

Tabel 6.9 Pengujian Daging Dengan *Learning Rate* 0,2 dan *Epoch* 1000

No.	Citra Daging	Identifikasi Visual	Identifikasi Sistem	Keterangan	Nilai Output
1.	Gambar 1	Daging Sapi	Sapi	Benar	0.00011707
2.	Gambar 2	Daging Sapi	Sapi	Benar	0.00316129
3.	Gambar 3	Daging Sapi	Sapi	Benar	0.00164993

No.	Citra Daging	Identifikasi Visual	Identifikasi Sistem	Keterangan	Nilai Output
4.	Gambar 4	Daging Sapi	Sapi	Benar	0.07277590
5.	Gambar 5	Daging Sapi	Sapi	Benar	0.00229681
6.	Gambar 6	Daging Sapi	Sapi	Benar	0.00211125
7.	Gambar 7	Daging Babi	Babi	Benar	0.99945187
8.	Gambar 8	Daging Babi	Babi	Benar	0.99288303
9.	Gambar 9	Daging Babi	Babi	Benar	0.99994524
10.	Gambar 10	Daging Babi	Babi	Benar	0.98726181
11.	Gambar 11	Daging Babi	Sapi	Salah	0.06338475
12.	Gambar 12	Daging Babi	Sapi	Salah	0.47033447

Tabel dibawah ini menunjukkan hasil identifikasi daging dengan menggunakan *learning rate* 0,2 dan *epoch* sebanyak 3000 dengan citra berukuran 300x300 dan rata-rata akurasi dari proses tersebut adalah **91,67%**.

Tabel 6.10 Pengujian Daging Dengan *Learning Rate* 0,2 dan *Epoch* 3000

No.	Citra Daging	Identifikasi Visual	Identifikasi Sistem	Keterangan	Nilai Output
1.	Gambar 1	Daging Sapi	Sapi	Benar	3.4183E-08
2.	Gambar 2	Daging Sapi	Sapi	Benar	3.5850E-06
3.	Gambar 3	Daging Sapi	Sapi	Benar	2.7200E-06
4.	Gambar 4	Daging Sapi	Sapi	Benar	0.01251086
5.	Gambar 5	Daging Sapi	Sapi	Benar	1.3486E-05
6.	Gambar 6	Daging Sapi	Sapi	Benar	0.00011693
7.	Gambar 7	Daging Babi	Babi	Benar	0.99999968
8.	Gambar 8	Daging Babi	Babi	Benar	0.99998293
9.	Gambar 9	Daging Babi	Babi	Benar	0.99999997
10.	Gambar 10	Daging Babi	Babi	Benar	0.99996770
11.	Gambar 11	Daging Babi	Sapi	Salah	0.00561736
12.	Gambar 12	Daging Babi	Babi	Benar	0.98960707

Tabel dibawah ini menunjukkan hasil identifikasi daging dengan menggunakan *learning rate* 0,2 dan *epoch* sebanyak 5000 dengan citra berukuran 300x300 dan rata-rata akurasi dari proses tersebut adalah **91,67%**.

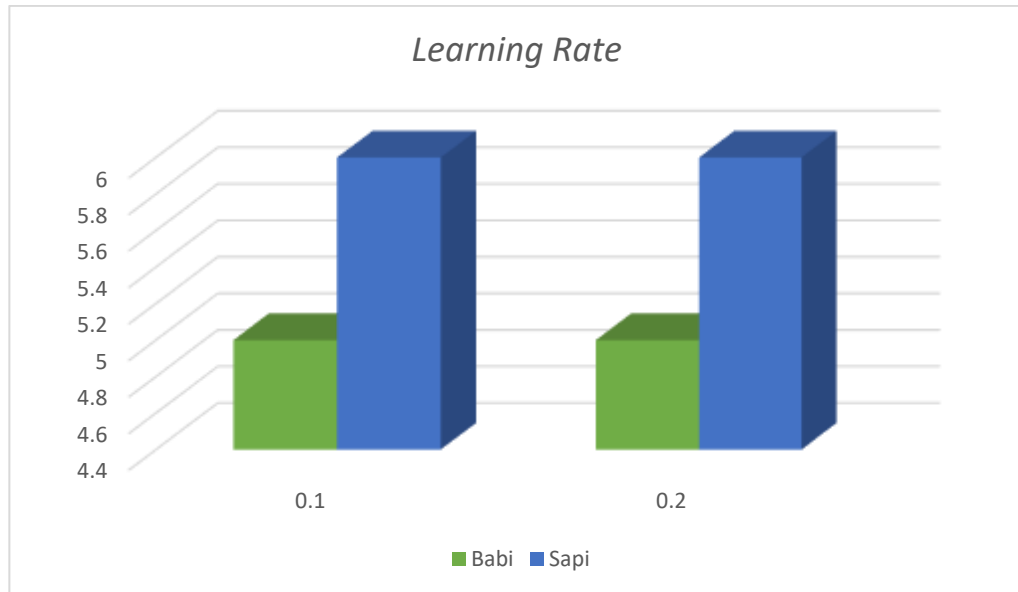
Tabel 6.11 Pengujian Daging Dengan *Learning Rate* 0,2 dan *Epoch* 5000

No.	Citra Daging	Identifikasi Visual	Identifikasi Sistem	Keterangan	Nilai Output
1.	Gambar 1	Daging Sapi	Sapi	Benar	1.1401E-08
2.	Gambar 2	Daging Sapi	Sapi	Benar	1.3074E-06
3.	Gambar 3	Daging Sapi	Sapi	Benar	1.0398E-06
4.	Gambar 4	Daging Sapi	Sapi	Benar	0.00854951
5.	Gambar 5	Daging Sapi	Sapi	Benar	5.8049E-06
6.	Gambar 6	Daging Sapi	Sapi	Benar	6.0620E-05
7.	Gambar 7	Daging Babi	Babi	Benar	0.99999988
8.	Gambar 8	Daging Babi	Babi	Benar	0.99999311
9.	Gambar 9	Daging Babi	Babi	Benar	0.99999999
10.	Gambar 10	Daging Babi	Babi	Benar	0.99998672
11.	Gambar 11	Daging Babi	Sapi	Salah	0.00369903
12.	Gambar 12	Daging Babi	Babi	Benar	0.99400894

6.2 Pembahasan

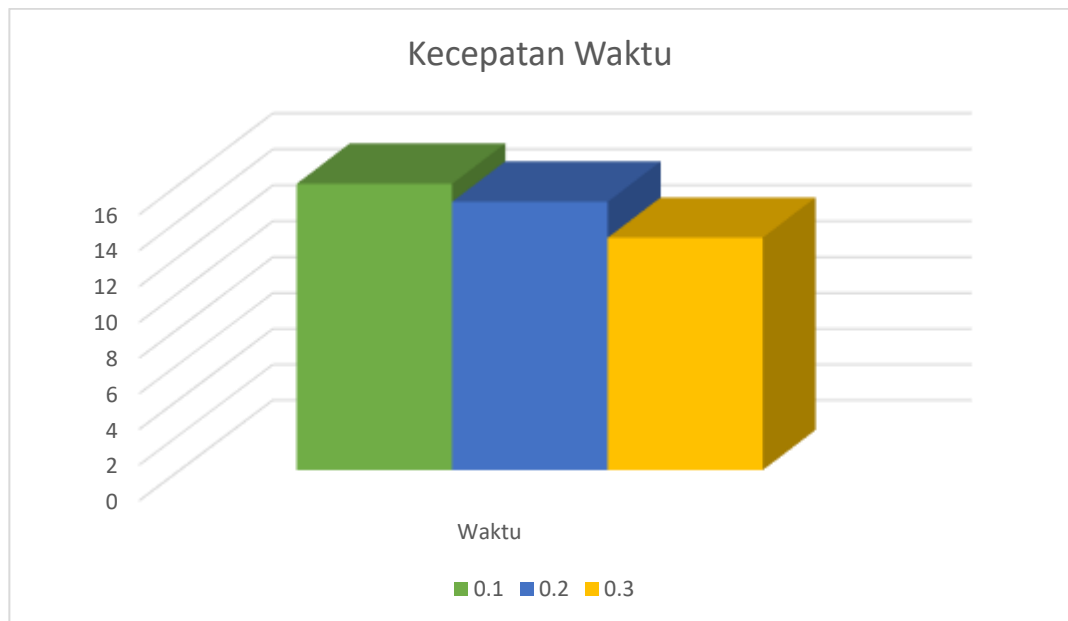
Dari hasil identifikasi pada tabel 6.4, sampai 6.11 didapatkan tingkat keberhasilan identifikasi daging menggunakan metode *backpropagation* untuk identifikasi daging babi dan sapi akan semakin baik jika jumlah *epoch* terus ditambah. Dalam kasus ini jaringan mulai memberikan bobot stabil ketika berada pada *epoch* ke 5.000. berikut merupakan hasil diagram dari *epoch* yang telah diuji

- Dalam Pengujian ini data yang diuji adalah data berdasarkan *Learning Rate*, dan menunjukan jumlah akurasi yang diperoleh. sedangkan untuk ukuran citra masukan menggunakan ukuran citra 300x300 piksel, dan *epoch* random sebesar 5000



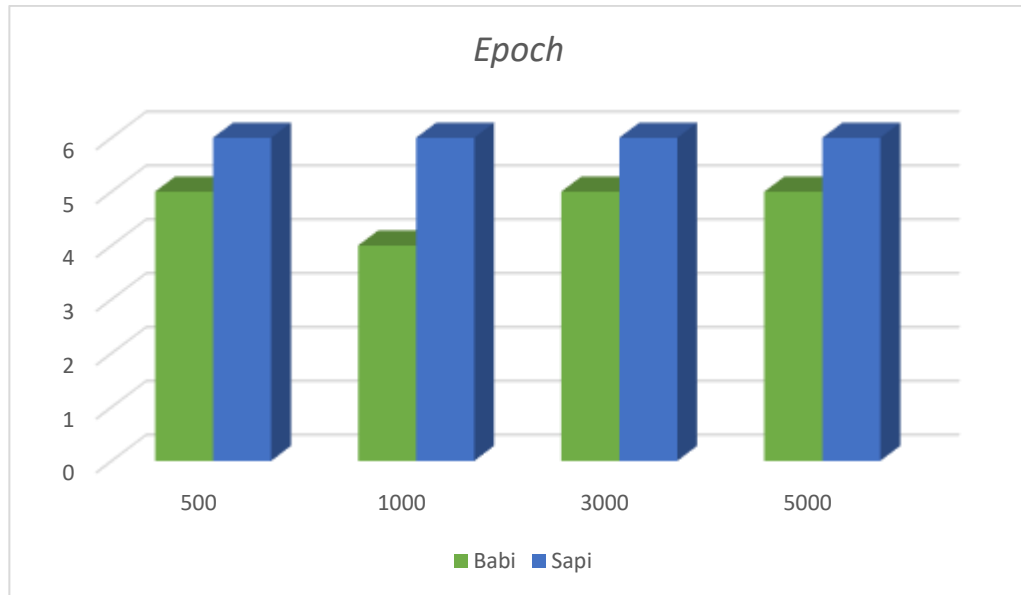
Gambar 6.1 Diagram Batang Berdasarkan *Learning Rate*

- Dalam Pengujian ini data yang diuji adalah data berdasarkan waktu (detik) sedangkan untuk ukuran citra masukan menggunakan ukuran citra 300x300 piksel, dan *epoch* random sebesar 5000



Gambar 6.2 Diagram Batang Berdasarkan Kecepatan Waktu

- Dalam Pengujian ini data yang diuji adalah data berdasarkan *Epoch*, dan menunjukkan jumlah akurasi yang diperoleh. Sedangkan untuk ukuran citra masukan menggunakan ukuran citra 300x300 piksel, dan *Learning Rate* sebesar 0.2



Gambar 6.3 Diagram Batang Berdasarkan *Epoch*

Dengan presentase keakuratan tiap jumlah *epoch*-nya secara keseluruhan dapat dihitung menggunakan rumus berikut:

1. Akurasi berdasarkan jumlah data terklasifikasi

$$\frac{\text{Jumlah data terklasifikasi}}{\text{Jumlah pengambilan seluruh data}} \times 100\%$$

Maka didapatkan tingkat akurasi sebagai berikut :

$$\text{Data training 500 epoch} = \frac{11}{12} \times 100\% = 83,3\%$$

$$\text{Data training 1000 epoch} = \frac{10}{12} \times 100\% = 91,67\%$$

$$\text{Data training 3000 epoch} = \frac{11}{12} \times 100\% = 91,67\%$$

$$\text{Data training 5000 epoch} = \frac{11}{12} \times 100\% = 91,67\%$$

2. Akurasi berdasarkan hasil persentase

$$\frac{\text{Jumlah Hasil Persentase}}{\text{Banyak data}}$$

$$\text{Data training 500 epoch} = \frac{100+83,3}{2} = 91,65\%$$

$$\text{Data training 1000 epoch} = \frac{100+66,67}{2} = 83,33\%$$

$$\text{Data training 3000 epoch} = \frac{100+83,3}{2} = 91,65\%$$

$$\text{Data training 5000 epoch} = \frac{100+83,3}{2} = 91,65\%$$

Tingkat keberhasilan identifikasi daging babi dan sapi secara keseluruhan adalah 89,57%.

BAB VII PENUTUP

7.1 Kesimpulan

Berdasarkan hasil dari penelitian yang dilakukan dapat diambil beberapa kesimpulan sebagai berikut:

1. Implementasi Sistem Klasifikasi Daging Babi dan Sapi Menggunakan Ekstraksi Fitur Warna dan Tekstur Dengan Metode GLCM yang dibangun sesuai dengan perancangan yang telah dibuat
2. Aplikasi ini dapat digunakan sebagai acuan pada penelitian lebih lanjut.
3. Metode klasifikasi *Backpropagation* dapat diterapkan untuk mengenali citra daging sapi dan babi berdasarkan ekstraksi ciri warna RGB dan tekstur karena memperoleh hasil rata-rata akurasi seluruh pengujian sebesar 89,57%.
4. Aplikasi ini minimal menggunakan kamera minimal 5MP dan jarak pengambilan citra 5-7cm. Aplikasi ini dapat mendeteksi berbagai bagian dari daging.
5. Dari hasil klasifikasi tersebut didapat presentase keberhasilan berdasarkan dua output yaitu daging babi sebesar 79,14%, dan daging sapi sebesar 100%.

7.2 Saran

Dari hasil penelitian yang telah dilakukan masih terdapat kekurangan. Adapun saran untuk penelitian selanjutnya sebagai berikut:

1. Menambahkan jumlah data set agar persentase akurasi lebih maksimal
2. Diharapkan pada pengembangan aplikasi selanjutnya dapat menggunakan sistem *realtime*.
3. Aplikasi dapat dikembangkan ke dalam *platform mobile*.
4. Menambahkan objek lain seperti daging kambing, daging ayam, daging kerbau, dan lainnya.

DAFTAR PUSTAKA

- [1] _____. (2014, Apr.29). Macam-macam Daging [online] Available: <https://tester.co.id/blog/macam-macam-daging/>.
- [2] Muhammad, Yudha. (2015, Sep.16). Manfaat Daging Sapi Berdasarkan Kandungan Gizi Lengkap [online] Available: <http://www.cantikitu.com/2015/09/manfaat-daging-sapi-dan-kandungan-gizinya.html>.
- [3] Fiki, Ariyanti. (2016, Agu.07). Ternyata Ini yang Bikin Harga Daging Sapi Mahal [online] Available : <http://bisnis.liputan6.com/read/2571327/ternyata-ini-yang-bikin-harga-daging-sapi-mahal>.
- [4] Kisanto, “Identifikasi Citra Untuk Mengidentifikasi Jenis Daging Sapi Dengan Menggunakan Transformasi Wavelet Haar” pp. i, Mar. 2012
- [5] Budianita, Elvina, dkk, “Implementasi Pengolahan Citra dan Klasifikasi K-Nearest Neighbour Untuk Membangun Aplikasi Pembeda Daging Sapi dan Babi” pp.242, Jun 2015
- [6] Wibowo, Laksono Tri, “Klasifikasi Kelas Daging Menggunakan Pencirian Matriks Ko-Okurensi Aras Keabuan” pp.1, 2010
- [7] Mas’ud, dan Catur Supriyanto, “Implementasi Principal Component Analysis (PCA) dan Euclidean Distance untuk Identifikasi Citra Daging Sapi dan Daging Babi” 2015
- [8] Astuti, Winda Rizky. “Implementasi Principal Component Analysis (PCA) dan Euclidean Distance untuk Identifikasi Citra Daging Sapi dan Daging Babi” pp.80, Agust 2016
- [9] Muhammadiy, Wasfi. “Aplikasi Pendeteksi Daging Sapi Dan Babi Mentah Pada Smartphone Android” pp. i, Agust 2016
- [10] Yuli, Anah. (-). Jenis-jenis Bagian Pemotongan Daging Sapi [online] Available : <http://resepcaramemasak.org/jenis-jenis-bagian-daging-sapi>.
- [11] Andi, Wawang. (2011). Tekstur Daging [online] Available: <http://peternakan-id.blogspot.co.id/2011/04/tekstur-daging.html>.
- [12] Ade. (2009, Mei.2) Image Processing [online] Available: <http://ndoware.com/image-processing.html>.

- [13] Hestiningsih, Idhawati. (-). Pengolahan Citra [online] Available: <http://27afiril-file.weebly.com/uploads/1/3/0/7/13077226/pengolahan-citra.pdf>.
- [14] Sapermana, Romli. (-). Pengertian Citra dan Pengolahan Citra [online] Available: <http://www.romlisapermana.com/2015/07/pengertian-citra-dan-pengolahan-citra.html>.
- [15] Johan. (2012, Jun.1). Penfertian dan Arti Warna. [online] Available: <http://www.ilmugrafis.com/artikel.php?page=pengertian-arti-warna>.
- [16] Putra, Darma. “Dasar Pengolahan Citra” Pengolahan Citra, Ed.1. Yogyakarta: ANDI, 2010, BAB 3, 2.8.4, pp.57
- [17] Tyas., (2013, Jan.2). *Grayscale* [online] Available: <http://informatika.web.id/grayscale-2.htm>.
- [18] Mulkan, Syarif. (-). Analisis Tekstur – GLCM. [online] Available: <https://id.scribd.com/doc/202665932/Analisis-Tekstur-Citra-Dengan-Teknik-GLCM>.
- [19] Purnomo, Adi. (-). “Aplikasi Pemrograman C# Untuk Analisis Tekstur Kayu Parquet Dengan Menggunakan Metode Grey Level Co-occurrence Matrix (GLCM)” [online] Available: http://www.gunadarma.ac.id/library/articles/graduate/industrial-technology/2009/Artikel_50405013.pdf
- [20] _____. (2015, Mei.18). GLCM - Analisis Tekstur Citra Digital [online] Available: <http://www.softscients.web.id/2015/01/glcm-analisis-tekstur-citra-digital.html>.
- [21] Purnamasari, Intan. (2016). *Pengenalan Ciri Garis Telapak Tangan Menggunakan Ekstraksi Fitur Gray Level Co-Occurence Matrix (GLCM) Dan Metode K-Nearest Neighbors (K-NN)* [online] Available : Http://Eprints.Dinus.Ac.Id/18247/9/Bab2_17797.Pdf.
- [22] Romadhon, Fajar. 2016 “Rancang Bangun Sistem Klasifikasi Tingkat Kematangan Buah Mangga Podang Berdasarkan Perbedaan Warna Menggunakan Jaringan Syaraf Tiruan Metode *Backpropagation*”. Malang: Politeknik Negeri
- [23] Kusumadewi, Sri. (2003). “Artificial Intelligence (Teknik dan Aplikasinya)”. Yogyakarta; Graha Ilmu.

- [24] Kusumaningtyas, Sella. 2015. “Identifikasi Kematangan Buah Tomat Berdasarkan Warna Menggunakan Metode Jaringan Syaraf Tiruan (JST)”. Malang: Politeknik Negeri.

LAMPIRAN 1
Listing Program

➤ Testing

```
Public Class Testingg
    Dim selection As Rectangle
    Dim V(9, 2) As Double
    'Angka menandakan array
    Dim W(2) As Double
    Dim vbias(2) As Double
    Dim wbias2 As Double
    Dim dt As DataTable

    'OPEN IMAGE
    Private Sub
OpenImageToolStripMenuItem1_Click(sender As Object, e As
EventArgs) Handles
OpenImageToolStripMenuItem1.Click
        OpenFileDialog1.Title =
"Open Image"
        OpenFileDialog1.Filter = "Image
File|*.bmp;*.jpg;*.png;*.jpeg"
        If OpenFileDialog1.ShowDialog()
= Windows.Forms.DialogResult.OK
Then
            PBI.Image =
Image.FromFile(OpenFileDialog1.
FileName)
            PBI.SizeMode =
PictureBoxSizeMode.StretchImage
        End If
        Cropping(PBI.Image)
        MeanNormalisasi()
    End Sub

    'Cropping
    Function Cropping(ByRef bp
As Bitmap)
        Try
            Dim imagel As New
Bitmap(PBI.Image)
            Me.Text =
"TRAINING"
            'Citra inputan akan dilakukan
cropping
            Dim selectionWidth As Integer =
300
            Dim selectionHeight As Integer
= 300

            If imagel.Width <
selectionWidth Or imagel.Height
< selectionHeight Then
                PBO.Image = PBI.Image
            Else
                selection = New Rectangle(New
Point((imagel.Width -
selectionWidth) / 2,
(imagel.Height -
selectionHeight) / 2), New
```

```
Size(selectionWidth,
selectionHeight))
                PBO.Image =
CropImage(PBI.Image)
            End If
            ToolStripProgressBar1.Value =
Int(100)
            PBO.SizeMode =
PictureBoxSizeMode.StretchImage
            ToolStripLabel1.Text = "Image
Size : " &
PBO.Image.Width.ToString & " x
" & PBO.Image.Height.ToString
        Catch ex As Exception
        End Try
    End Function

    Function CropImage(ByRef
Image As Bitmap) As Bitmap
        Try
            Dim cropped As Bitmap =
Image.Clone(selection,
Image.PixelFormat)
            Return cropped
        Catch ex As Exception
        End Try
    End Function

    'RGB
    Function MeanNormalisasi()
        Dim imagel As New
Bitmap(PBO.Image)
        Dim baris, kolom,
merah, hijau, biru, temp As
Integer
        temp = 0
        ProgressBar1.Show()
        Dim total As Integer =
imagel.Height * imagel.Width
        Dim mer, hij, bir As
Double
        For baris = 0 To
imagel.Height - 1
            For kolom = 0 To
imagel.Width - 1
                merah = imagel.GetPixel(kolom,
baris).R
                hijau = imagel.GetPixel(kolom,
baris).G
                biru = imagel.GetPixel(kolom,
baris).B
                mer = mer + merah
                hij = hij + hijau
                bir = bir + biru
                kolom = (kolom + 1)
            Next
            baris = (baris + 1)
        Me.Text = Int(100 * baris /
imagel.Width).ToString & "%"
```

```

ProgressBar1.Value = Int(100 *
baris / image1.Width)
    Next
end_of_for:
    Me.Text = "TRAINING"
    Red.Text = Math.Round(mer /
total).ToString
    Green.Text = Math.Round(hij
/ total).ToString
    Blue.Text = Math.Round(bir
/ total).ToString
    Red.Text = ((mer / total) /
(mer / total + hij / total +
bir / total)).ToString
    Green.Text = ((hij / total)
/ (mer / total + hij / total +
bir / total)).ToString
    Blue.Text = ((bir / total)
/ (mer / total + hij / total +
bir / total)).ToString
    ProgressBar1.Hide()
    End Function

'OPEN BOBOT
Private Sub
OToolStripMenuItem_Click(sender
As Object, e As EventArgs)
Handles
OToolStripMenuItem.Click
    Using open As New
OpenFileDialog
        With open
            .Filter = "XML File|*.xml"
            If .ShowDialog =
Windows.Forms.DialogResult.Canc
el Then Exit Sub
            Dim ds_xml As New DataSet
            ds_xml.ReadXml(.FileName)
            dt = ds_xml.Tables(0)
            If dt.Columns.Count > 30
Then
                W0.Text = dt.Rows(0).Item(1)
                W1.Text = dt.Rows(0).Item(2)
                W2.Text = dt.Rows(0).Item(3)
                V00.Text = dt.Rows(0).Item(5)
                V01.Text = dt.Rows(0).Item(6)
                V02.Text = dt.Rows(0).Item(7)
                V10.Text = dt.Rows(0).Item(8)
                V11.Text = dt.Rows(0).Item(9)
                V12.Text = dt.Rows(0).Item(10)
                V20.Text = dt.Rows(0).Item(11)
                V21.Text = dt.Rows(0).Item(12)
                V22.Text = dt.Rows(0).Item(13)
                V30.Text = dt.Rows(0).Item(14)
                V31.Text = dt.Rows(0).Item(15)
                V32.Text = dt.Rows(0).Item(16)
                V40.Text = dt.Rows(0).Item(17)
                V41.Text = dt.Rows(0).Item(18)
                V42.Text = dt.Rows(0).Item(19)
                V50.Text = dt.Rows(0).Item(20)
                V51.Text = dt.Rows(0).Item(21)
                V52.Text = dt.Rows(0).Item(22)
                V60.Text = dt.Rows(0).Item(23)
                V61.Text = dt.Rows(0).Item(24)
                V62.Text = dt.Rows(0).Item(25)
                V70.Text = dt.Rows(0).Item(26)
                V71.Text = dt.Rows(0).Item(27)
                V72.Text = dt.Rows(0).Item(28)
                V80.Text = dt.Rows(0).Item(29)
                V81.Text = dt.Rows(0).Item(30)
                V82.Text = dt.Rows(0).Item(31)
                VBias0.Text =
dt.Rows(0).Item(32)
                VBias1.Text =
dt.Rows(0).Item(33)
                VBias2.Text =
dt.Rows(0).Item(34)
                WBias.Text = dt.Rows(0).Item(4)
            Else
                MsgBox("salah!")
            End If
        End With
    End Using
End Sub

'SAVE IMAGE
Private Sub
SaveImageToolStripMenuItem_Clic
k(sender As Object, e As
EventArgs) Handles
SaveImageToolStripMenuItem.Clic
k
    If IsNothing(PBO.Image) Then
        MessageBox.Show("Gambar Tidak
Boleh Kosong!!")
    Else
        SaveFileDialog1.FileName =
"Image"
        SaveFileDialog1.Filter =
"Image|*.bmp;*.jpg;*.png;*.jpeg"
        If SaveFileDialog1.ShowDialog()
= Windows.Forms.DialogResult.OK
Then
            PBO.Image.Save(SaveFileDialog1.
FileName)
            PBO.SizeMode =
PictureBoxSizeMode.StretchImage
        End If
    End If
End Sub
Private Sub
Open_Click(sender As Object, e
As EventArgs)
    OpenFileDialog1.Title =
"Open Image"
    OpenFileDialog1.Filter
=

```

```

"Image|*.bmp;*.jpg;*.png;*.jpeg
"

    If
OpenFileDialog1.ShowDialog =
Windows.Forms.DialogResult.OK
Then
        PBI.Image =
Image.FromFile(OpenFileDialog1.
FileName)
        PBI.SizeMode =
PictureBoxSizeMode.Zoom
ToolStripLabel1.Text = "Image
Size : " &
PBI.Image.Width.ToString & " x
" & PBI.Image.Height.ToString
        Exit Sub
    End If
End Sub

'TRAINING SHOW
    Private Sub
ToolStripDropDownButton1_Click(
sender As Object, e As
EventArgs) Handles
ToolStripDropDownButton1.Click
Trainingg.Show()
        Me.Hide()
    End Sub

'RESET
    Private Sub
Open_Click_1(sender As Object,
e As EventArgs) Handles
Open.Click
        Button2.Enabled = False
identifikasi.Enabled = False
LabelKlasifikasi.Text = Nothing
    End Sub
    Private Sub
Reset_Click_1(sender As Object,
e As EventArgs) Handles
Reset.Click
LabelKlasifikasi.Text = Nothing
        Button2.Enabled = False
identifikasi.Enabled = False
        PBI.Image = Nothing
        PBO.Image = Nothing
ToolStripProgressBar1.Value = 0
        Red.Text = Nothing
        Green.Text = Nothing
        Blue.Text = Nothing
        Energi.Text = Nothing
        Entropi.Text = Nothing
        Kontras.Text = Nothing
        Homogenitas.Text = Nothing
        Korelasi.Text = Nothing
        MSI.Text = Nothing
        TextBox11.Text = Nothing
        TextBox12.Text = Nothing

```

```

TextBox13.Text = Nothing
TextBox14.Text = Nothing
TextBox15.Text = Nothing
TextBox16.Text = Nothing
TextBox17.Text = Nothing
TextBox18.Text = Nothing
TextBox19.Text = Nothing
TextBox20.Text = Nothing
TextBox21.Text = Nothing
TextBox22.Text = Nothing
TextBox23.Text = Nothing
TextBox24.Text = Nothing
TextBox25.Text = Nothing
TextBox26.Text = Nothing
TextBox27.Text = Nothing
TextBox28.Text = Nothing
TextBox29.Text = Nothing
TextBox30.Text = Nothing
TextBox31.Text = Nothing
TextBox32.Text = Nothing
TextBox33.Text = Nothing
TextBox34.Text = Nothing
VBias0.Text = Nothing
VBias1.Text = Nothing
VBias2.Text = Nothing
WBias.Text = Nothing
W0.Text = Nothing
W1.Text = Nothing
W2.Text = Nothing
V00.Text = Nothing
V10.Text = Nothing
V20.Text = Nothing
V30.Text = Nothing
V40.Text = Nothing
V50.Text = Nothing
V60.Text = Nothing
V70.Text = Nothing
V80.Text = Nothing
V01.Text = Nothing
V11.Text = Nothing
V21.Text = Nothing
V31.Text = Nothing
V41.Text = Nothing
V51.Text = Nothing
V61.Text = Nothing
V71.Text = Nothing
V81.Text = Nothing
V02.Text = Nothing
V12.Text = Nothing
V22.Text = Nothing
V32.Text = Nothing
V42.Text = Nothing
V52.Text = Nothing
V62.Text = Nothing
V72.Text = Nothing
V82.Text = Nothing
output.Text = Nothing
End Sub

```

```

'ABOUT
    Private Sub
ToolStripSplitButton1_Click(sender As Object, e As EventArgs)
Handles
ToolStripSplitButton1.Click
        Help_Testing.Show()
    End Sub

'EXIT
    Private Sub
Out_Click_1(sender As Object, e
As EventArgs) Handles Out.Click
        Home.Show()
        Me.Hide()
    End Sub

'AVERAGE
    Private Sub
Button1_Click(sender As Object,
e As EventArgs) Handles
Button1.Click
        Button2.Enabled = True
If IsNothing(PBO.Image) Then
    MessageBox.Show("Gambar Tidak
Boleh Kosong!!")
Else
    Dim image1 As New
    Bitmap(PBO.Image)
    'Jumlah
        Dim brs, klm, toljum0,
toljum45, toljum90, toljum135,
        norjum0, norjum45,
        norjum90, norjum135,
        enerjum0, enerjum45,
        enerjum90, enerjum135,
        entrojum0, entrojum45,
        entrojum90, entrojum135,
        konjum0, konjum45,
        konjum90, konjum135,
        homojum0, homojum45,
        homojum90, homojum135,
        korjum0, korjum45,
        korjum90, korjum135,
        msjum0, msjum45, msjum90,
        msjum135 As Double
    'ambil warna
        Dim red As Double
    'Sudut
        Dim r0, r45, r90, r135 As
Double
    'Framework & Transpos
    Dim fw0(255, 255), t0(255,
255),
    fw45(255, 255), t45(255, 255),
    fw90(255, 255), t90(255, 255),
    fw135(255, 255), t135(255, 255)
    As Integer
    'Perhitungan
        Dim jum0(255, 255),
jum45(255, 255), jum90(255,
255), jum135(255, 255),
        nor0(255, 255), nor45(255,
255), nor90(255, 255),
        nor135(255, 255),
        energi0(255, 255),
        energi45(255, 255),
        energi90(255, 255),
        energi135(255, 255),
        entropi0(255, 255),
        entropi45(255, 255),
        entropi90(255, 255),
        entropi135(255, 255),
        End If
    End Sub

    Dim image1 As New
    Bitmap(PBO.Image)
    PBO.Image = image1
    PBO.SizeMode =
PictureBoxSizeMode.StretchImage
    Dim brs, klm As Integer
    Dim mrh, hj, br, ab As Integer
    For brs = 0 To image1.Width - 1
    For klm = 0 To image1.Height -
1
        mrh = image1.GetPixel(brs,
klm).R
        hj = image1.GetPixel(brs,
klm).G
        br = image1.GetPixel(brs,
klm).B
        ab = CInt((mrh + hj + br) / 3)
        image1.SetPixel(brs, klm,
Color.FromArgb(ab, ab, ab))
    Next
    Me.Text = Int((100 * brs /
image1.Width) + 1).ToString &
"%"
    ToolStripProgressBar1.Value =
Int((100 * brs / image1.Width)
+ 1)
    ToolStripLabel1.Text = "Image
Size : " &
    image1.Width.ToString & " x " &
    image1.Height.ToString
    Next
    PBO.Refresh()

```

```

        kontras0(255, 255),
        kontras45(255, 255),
        kontras90(255, 255),
        kontras135(255, 255),
        homogenitas0(255, 255),
        homogenitas45(255, 255),
        homogenitas90(255, 255),
        homogenitas135(255, 255),
        korelasi0(255, 255),
        korelasi45(255, 255),
        korelasi90(255, 255),
        korelasi135(255, 255),
        momsel0(255, 255),
        momsel45(255, 255),
        momsel90(255, 255),
        momsel135(255, 255),
        rrj0(255, 255), rrj45(255,
255), rrj90(255, 255),
        rrj135(255, 255),
        mean_i0, mean_i45,
        mean_i90, mean_i135, mean_j0,
        mean_j45, mean_j90, mean_j135,
        varians_i0, varians_i45,
        varians_i90, varians_i135,
        varians_j0, varians_j45,
        varians_j90, varians_j135 As
Double
'Reset
        Energi.Clear()
        Entropi.Clear()
        Kontras.Clear()
        Homogenitas.Clear()
        Korelasi.Clear()
        MSI.Clear()
        TextBox11.Clear()
        TextBox12.Clear()
        TextBox13.Clear()
        TextBox14.Clear()
        TextBox15.Clear()
        TextBox16.Clear()
        TextBox17.Clear()
        TextBox18.Clear()
        TextBox19.Clear()
        TextBox20.Clear()
        TextBox21.Clear()
        TextBox22.Clear()
        TextBox23.Clear()
        TextBox24.Clear()
        TextBox25.Clear()
        TextBox26.Clear()
        TextBox27.Clear()
        TextBox28.Clear()
        TextBox29.Clear()
        TextBox30.Clear()
        TextBox31.Clear()
        TextBox32.Clear()
        TextBox33.Clear()
        TextBox34.Clear()

```

```

'Perulangan Baris
        For brs = 0 To
        imagel.Width - 1
            For klm = 0 To
            imagel.Height - 1
                red = imagel.GetPixel(brs,
klm).R
                If (brs + 1 <=
            imagel.Width - 1) Then
                    r0 = imagel.GetPixel(brs +
1, klm).R
                    fw0(red, r0) = fw0(red, r0) + 1
                    t0(r0, red) = t0(r0, red) + 1
                End If
            If (brs + 1 <= imagel.Width - 1
            AndAlso klm - 1 >= 0) Then
                    r45 = imagel.GetPixel(brs +
1, klm - 1).R
                    fw45(red, r45) = fw45(red,
r45) + 1
                    t45(r45, red) = t45(r45, red)
+ 1
                End If
                If (klm - 1 >= 0) Then
                    r90 = imagel.GetPixel(brs,
klm - 1).R
                    fw90(red, r90) = fw90(red, r90)
+ 1
                    t0(r90, red) = t0(r90, red) + 1
                End If
                If (brs - 1 >= 0 AndAlso
            klm - 1 >= 0) Then
                    r135 = imagel.GetPixel(brs
- 1, klm - 1).R
                    fw135(red, r135) = fw135(red,
r135) + 1
                    t135(r135, red) = t135(r135,
red) + 1
                End If
            Next
        Next
'Hasil Penjumlahan
        For i = 0 To 254
            For j = 0 To 254
                jum0(i, j) = fw0(i, j) +
            t0(i, j)
                toljum0 += jum0(i, j)
                jum45(i, j) = fw45(i, j) +
            t45(i, j)
                toljum45 += jum45(i, j)
                jum90(i, j) = fw90(i, j) +
            t90(i, j)
                toljum90 += jum90(i, j)
                jum135(i, j) = fw135(i, j) +
            t135(i, j)
                toljum135 += jum135(i, j)
            Next
        Next

```

```

'Normalisasi
    For i = 0 To 254
        For j = 0 To 254
            nor0(i, j) = jum0(i, j) /
toljum0
            norjum0 += nor0(i, j)
            nor45(i, j) = jum45(i, j)
/ toljum45
            norjum45 += nor45(i, j)
            nor90(i, j) = jum90(i, j)
/ toljum90
            norjum90 += nor90(i, j)
            nor135(i, j) = jum135(i,
j) / toljum135
            norjum135 += nor135(i, j)
        Next
    Next
'Energi
    For i = 0 To 254
        For j = 0 To 254
            energi0(i, j) = nor0(i, j) ^ 2
            enerjum0 += energi0(i, j)
            energi45(i, j) = nor45(i, j) ^
2
            enerjum45 += energi45(i, j)
            energi90(i, j) = nor90(i, j) ^
2
            enerjum90 += energi90(i, j)
            energi135(i, j) = nor135(i, j)
^ 2
            enerjum135 += energi135(i, j)
        Next
    Next
'Entropi
    For i = 0 To 254
        For j = 0 To 254
            If nor0(i, j) = 0 Then
                entropi0(i, j) =
(Math.Abs(nor0(i, j) *
Math.Log(1))) / 255
            Else
                entropi0(i, j) =
(Math.Abs(nor0(i, j) *
Math.Log(nor0(i, j)))) / 255
            End If
            entrojum0 += entropi0(i, j)
            If nor45(i, j) = 0 Then
                entropi45(i, j) =
(Math.Abs(nor45(i, j) *
Math.Log(1))) / 255
            Else
                entropi45(i, j) =
(Math.Abs(nor45(i, j) *
Math.Log(nor45(i, j)))) / 255
            End If
            entrojum45 += entropi45(i, j)
            If nor90(i, j) = 0 Then
                entropi90(i, j) =
(Math.Abs(nor90(i, j) *
Math.Log(1))) / 255
            Else
                entropi90(i, j) =
(Math.Abs(nor90(i, j) *
Math.Log(nor90(i, j)))) / 255
            End If
            entrojum90 += entropi90(i, j)
            If nor135(i, j) = 0 Then
                ntropi135(i, j) =
(Math.Abs(nor135(i, j) *
Math.Log(1))) / 255
            Else
                entropi135(i, j) =
(Math.Abs(nor135(i, j) *
Math.Log(nor135(i, j)))) / 255
            End If
            entrojum135 += entropi135(i, j)
        Next
    Next
'Kontras
    For i = 0 To 254
        For j = 0 To 254
            kontras0(i,
j) = (((i - j) ^ 2) * nor0(i,
j)) / 255
            konjum0 += kontras0(i, j)
            kontras45(i, j) = (((i - j)
^ 2) * nor45(i, j)) / 255
            konjum45 += kontras45(i, j)
            kontras90(i, j) = (((i - j)
^ 2) * nor90(i, j)) / 255
            konjum90 += kontras90(i, j)
            kontras135(i, j) = (((i -
j) ^ 2) * nor135(i, j)) / 255
            konjum135 += kontras135(i,
j)
        Next
    Next
'Homogenitas
    For i = 0 To 254
        For j = 0 To 254
            homogenitas0(i, j) = nor0(i, j)
/ (1 + Math.Abs(i - j))
            .homojum0 += homogenitas0(i, j)
            homogenitas45(i, j) = nor45(i,
j) / (1 + Math.Abs(i - j))
            .homojum45 += homogenitas45(i,
j)
            homogenitas90(i, j) = nor90(i,
j) / (1 + Math.Abs(i - j))
            .homojum90 += homogenitas90(i,
j)
            homogenitas135(i, j) =
nor135(i, j) / (1 + Math.Abs(i
- j))
            .homojum135 += homogenitas135
(i, j)

```



```

        Next
    Next
    'Kolerasi
    For i As Integer = 0 To 254
    For j As Integer = 0 To 254
    mean_i0 += i * nor0(i, j)
    mean_i45 += i * nor45(i, j)
    mean_i90 += i * nor90(i, j)
    mean_il35 += i * nor135(i, j)
    mean_j0 += j * nor0(i, j)
    mean_j45 += j * nor45(i, j)
    mean_j90 += j * nor90(i, j)
    mean_jl35 += j * nor135(i, j)
    Next
    Next
    For i As Integer = 0 To 254
    For j As Integer = 0 To 254
        varians_i0 += nor0(i, j) *
        ((i - mean_i0) * (i - mean_i0))
        varians_j0 += nor0(i, j) *
        ((j - mean_j0) * (j - mean_j0))
        varians_i45 += nor0(i, j)
        * ((i - mean_i45) * (i -
        mean_i45))
        varians_j45 += nor0(i, j)
        * ((j - mean_j45) * (j -
        mean_j45))
        varians_i90 += nor0(i, j)
        * ((i - mean_i90) * (i -
        mean_i90))
        varians_j90 += nor0(i, j)
        * ((j - mean_j90) * (j -
        mean_j90))
        varians_il35 += nor0(i, j)
        * ((i - mean_il35) * (i -
        mean_il35))
        varians_jl35 += nor0(i, j)
        * ((j - mean_jl35) * (j -
        mean_jl35))
    Next
    Next
    For i = 0 To 254
    For j = 0 To 254
    korelasi0(i, j) = nor0(i, j) *
    ((i - mean_i0) * (j - mean_j0)
    / Math.Sqrt(varians_i0 *
    varians_j0))
    korjum0 += korelasi0(i, j)
    korelasi45(i, j) = nor45(i, j)
    * ((i - mean_i45) * (j -
    mean_j45) /
    Math.Sqrt(varians_i45 *
    varians_j45))
    korjum45 += korelasi45(i, j)
    korelasi90(i, j) = nor90(i, j)
    * ((i - mean_i90) * (j -
    mean_j90) /
    Math.Sqrt(varians_i90 *
    varians_j90))
    korjum90 += korelasi90(i, j)
    korelasi135(i, j) = nor135(i,
    j) * ((i - mean_il35) * (j -
    mean_jl35) /
    Math.Sqrt(varians_il35 *
    varians_jl35))
    korjum135 += korelasi135(i, j)
    Next
    Next
    'Momentum selisih
    For i = 0 To 254
    For j = 0 To 254
        momsel0(i, j) = 1 / (1 + (i -
        j) ^ 2) * nor0(i, j)
        msjum0 += momsel0(i, j)
        momsel45(i, j) = 1 / (1 + (i
        - j) ^ 2) * nor45(i, j)
        msjum45 += momsel45(i, j)
        momsel90(i, j) = 1 / (1 + (i
        - j) ^ 2) * nor90(i, j)
        msjum90 += momsel90(i, j)
        momsel135(i, j) = 1 / (1 + (i
        - j) ^ 2) * nor135(i, j)
        msjum135 += momsel135(i, j)
    Next
    Next
    'Penampilan
    Energi.Text = (enerjum0 +
    enerjum45 + enerjum90 +
    enerjum135) / 4
    Entropi.Text = (entrojum0
    + entrojum45 + entrojum90 +
    entrojum135) / 4
    Kontras.Text = (konjum0 +
    konjum45 + konjum90 +
    konjum135) / 4
    Homogenitas.Text =
    (homojum0 + homojum45 +
    homojum90 + homojum135) / 4
    Korelasi.Text = (korjum0 +
    korjum45 + korjum90 +
    korjum135) / 4
    MSI.Text = (msjum0 +
    msjum45 + msjum90 + msjum135) /
    4
    TextBox11.Text = enerjum0
    TextBox12.Text = enerjum45
    TextBox13.Text = enerjum90
    TextBox14.Text = enerjum135
    TextBox15.Text = entrojum0
    TextBox16.Text = entrojum45
    TextBox17.Text = entrojum90
    TextBox18.Text = entrojum135
    TextBox19.Text = konjum0
    TextBox20.Text = konjum45
    TextBox21.Text = konjum90
    TextBox22.Text = konjum135
    TextBox23.Text = homojum0
    TextBox24.Text = homojum45

```

```

        TextBox25.Text = homojum90
        TextBox26.Text = homojum135
        TextBox27.Text = korjum0
        TextBox28.Text = korjum45
        TextBox29.Text = korjum90
        TextBox30.Text = korjum135
        TextBox31.Text = msjum0
        TextBox32.Text = msjum45
        TextBox33.Text = msjum90
        TextBox34.Text = msjum135
        ToolStripProgressBar1.Value = 0
        Me.Text = Int((100 * brs /
        image1.Width)).ToString & "%"
        ToolStripProgressBar1.Value =
        Int(100 * brs / image1.Width)
        ToolStripLabel1.Text = "Image
        Size : " &
        image1.Width.ToString & " x " &
        image1.Height.ToString
            End If
        End Sub

'IDENTIFIKASI
    Private Sub
    identifikasi_Click(sender As
    Object, e As EventArgs) Handles
    identifikasi.Click
        If (V00.Text) = Nothing
        Then
            MsgBox("Masukan bobot
            terlebih dahulu!")
        Else
            V(1, 0) = V00.Text
            V(2, 0) = V10.Text
            V(3, 0) = V20.Text
            V(4, 0) = V30.Text
            V(5, 0) = V40.Text
            V(6, 0) = V50.Text
            V(7, 0) = V60.Text
            V(8, 0) = V70.Text
            V(9, 0) = V80.Text
            V(1, 1) = V01.Text
            V(2, 1) = V11.Text
            V(3, 1) = V21.Text

            V(4, 1) = V31.Text
            V(5, 1) = V41.Text
            V(6, 1) = V51.Text
            V(7, 1) = V61.Text
            V(8, 1) = V71.Text
            V(9, 1) = V81.Text
            V(1, 2) = V02.Text
            V(2, 2) = V12.Text
            V(3, 2) = V22.Text
            V(4, 2) = V32.Text
            V(5, 2) = V42.Text
            V(6, 2) = V52.Text
            V(7, 2) = V62.Text
            V(8, 2) = V72.Text
            V(9, 2) = V82.Text
            W(0) = W0.Text
            W(1) = W1.Text
            W(2) = W2.Text
            vbias(0) = VBias0.Text
            vbias(1) = VBias1.Text
            vbias(2) = VBias2.Text
            wbias2 = WBias.Text
            Dim z_hasil As Double
            z_hasil =
            Backpro.hitungTesting(Red.Text,
            Green.Text, Blue.Text,
            Energi.Text, Entropi.Text,
            Kontras.Text, Homogenitas.Text,
            Korelasi.Text, MSI.Text, vbias,
            V, wbias2, W)
            output.Text = z_hasil
            If (output.Text <= 0.49
            And output.Text > 0) Then
                LabelKlasifikasi.Text = "Sapi"
            '0
            ElseIf (output.Text <=
            1 And output.Text > 0.5) Then
                LabelKlasifikasi.Text = "Babi"
            '1
            End If
            End If
        End Sub
    End Class

```

LAMPIRAN 2
Hasil *Training*

Data *Training*

[illegible][illegible][illegible]

LAMPIRAN 3
Surat Penelitian



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
Jl. Soekarno Hatta No.9 Malang 65141
Telp (0341) 404424 – 404425 Fax (0341) 404420
Laman://www.polinema.ac.id



Nomor : 126 /PL2.1/PM/2017
Perihal : Observasi Data

Yth. Kepala
Badan Kesatuan Bangsa Dan Politik Kota Malang
di tempat

Dengan hormat,

Bersama ini kami mohon bantuan Bapak/Ibu, agar dapat memberi kesempatan kepada mahasiswa kami Jurusan Teknologi Informasi Program Studi Teknik Informatika untuk dapat melakukan observasi di perusahaan/instansi yang Bapak/Ibu pimpin untuk kepentingan pengambilan data untuk tugas akhir.

Adapun nama mahasiswa tersebut adalah sebagai berikut :

NO	NAMA	NIM
1.	Robertus Romario H.K.	1341180117

Observasi tersebut menurut rencana akan dilaksanakan pada tanggal tanggal 01-28 Pebruari 2017,
Demikian, atas perhatian dan kerjasamanya disampaikan terima kasih.



a.n. Direktur
Pembantu Direktur I,

Dr. Drs. Lutfi Djaiano, MBA
NIP. 19620421198803 1 003

Tembusan Yth. :

1. Ketua Jurusan Teknologi Informasi
2. Ketua Program Studi Teknik Informatika

FRM.RIF.01.20.00



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
Jl. Soekarno Hatta No.9 Malang 65141
Telp (0341) 404424 – 404425 Fax (0341) 404420
Laman://www.polinema.ac.id



Nomor : 1285 /PL2.1/PM/2017
Perihal : Observasi Data

10 FEB 2017

Yth. **Kepala
Dinas Pertanian dan Ketahanan Pangan
di tempat**

Dengan hormat,

Bersama ini kami mohon bantuan Bapak/Ibu, agar dapat memberi kesempatan kepada mahasiswa kami Jurusan Teknologi Informasi Program Studi Teknik Informatika untuk dapat melakukan* observasi di perusahaan/instansi yang Bapak/Ibu pimpin untuk kepentingan pengambilan data untuk tugas akhir.

Adapun nama mahasiswa tersebut adalah sebagai berikut :

NO	NAMA	NIM
1.	Robertus Romario H.K.	1341180117

Observasi tersebut menurut rencana akan dilaksanakan pada tanggal tanggal 01-28 Pebruari 2017.
Demikian, atas perhatian dan kerjasamanya disampaikan terima kasih.

a.n.Direktur
Pembantu Direktur I,

Dr. Drs. Eudfi Djaiano, MBA
NIP. 19620421198803 1 003

Tembusan Yth. :

1. Ketua Jurusan Teknologi Informasi
2. Ketua Program Studi Teknik Informatika

FRM.RIF.01.20.00



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
Jl. Sockarno Hatta No.9 Malang 65141
Telp (0341) 404424 – 404425 Fax (0341) 404420
Laman://www.polinema.ac.id



Nomor : 1185 /PL2.1/PM/2017
Perihal : Observasi Data

08 FEB 2017

Yth. Kepala
Rumah Potong Hewan Malang
di tempat

Dengan hormat,

Bersama ini kami mohon bantuan Bapak/Ibu, agar dapat memberi kesempatan kepada mahasiswa kami Jurusan Teknologi Informasi Program Studi Teknik Informatika untuk dapat melakukan observasi di perusahaan/instansi yang Bapak/Ibu pimpin untuk kepentingan pengambilan data untuk tugas akhir.

Adapun nama mahasiswa tersebut adalah sebagai berikut :

NO	NAMA	NIM
1.	Robertus Romario H.K	1341180117

Observasi tersebut menurut rencana akan dilaksanakan pada tanggal tanggal 01 Pebruari-28 Pebruari 2017. Demikian, atas perhatian dan kerjasamanya disampaikan terima kasih.

a.n. Direktur
Pembantu Direktur I,

Dr. Drs. Ludfi Djaizanto, MBA
NIP. 19620421198803 1 003

Tembusan Yth. :

1. Ketua Jurusan Teknologi Informasi
2. Ketua Program Studi Teknik Informatika

FRM.RIF.01.20.00



PEMERINTAH KOTA MALANG
BADAN KESATUAN BANGSA DAN POLITIK
Jl. A. Yani No. 98 Telp. (0341) 491180 Fax. 474254
M A L A N G

Kode Pos 65125

REKOMENDASI PELAKSANAAN PENELITIAN
NOMOR : 072/181.02.P/35.73.406/2017

Berdasarkan pemenuhan ketentuan persyaratan sebagaimana ditetapkan dalam Peraturan Walikota Malang Nomor 24 Tahun 2011 Tentang Pelayanan Pemberian Rekomendasi Pelaksanaan Penelitian dan Praktek Kerja Lapangan di Lingkungan Pemerintah Kota Malang Oleh Badan Kesatuan Bangsa dan Politik Kota Malang serta menunjuk surat Pembantu Direktur I Politeknik Negeri Malang 1286/PL2.I/PM/2017 tanggal 10 Pebruari 2017 perihal : Observasi Data, kepada pihak sebagaimana disebut di bawah ini :

- a. Nama : ROBERTUS ROMARIO H.K., (peserta : - orang peserta).
- b. Nomor Identitas : 1341180117.
- c. Judul Penelitian : Klasifikasi Daging Babi dan Sapi Menggunakan Ekstraksi Fitur Warna dan Tekstur dengan Metode Gray Level Co-Occurrence Matrix.

dinyatakan memenuhi persyaratan untuk melaksanakan penelitian tugas skripsi yang berlokasi di :

- Dinas Pertanian dan Ketahanan Pangan Kota Malang.
- PD. Rumah Pemotongan Hewan Kota Malang.

Sepanjang yang bersangkutan memenuhi ketentuan sebagai berikut :

- a. Tidak melakukan penelitian yang tidak sesuai atau tidak ada kaitannya dengan judul, maksud dan tujuan penelitian;
- b. Menjaga perilaku dan mentaati tata tertib yang berlaku pada Lokasi tersebut di atas;
- c. Mentaati ketentuan peraturan perundang-undangan.

Demikian rekomendasi ini dibuat untuk dipergunakan sebagaimana mestinya, dan masa berlaku rekomendasi ini adalah sejak tanggal *ditetapkan s/d 10 Maret 2017*.

Malang, 10 Pebruari 2017
An. KEPALA BAKESBANGPOL
M A L A N G

[Signature]
Drs. KUNTJORO TRIKTMADJI
Pembina Tk. I
NIP. 19600212199111 1 001







Tembusan :







Yth. Sdr. - Pembantu Direktur I Politeknik
Negeri Malang;







▲ Vana beresanebisan







LAMPIRAN 4
Lembar Validasi







Tabel Uji Validasi Data *Training* Aplikasi Klasifikasi Daging Babi dan Sapi Menggunakan Ekstraksi Warna dan Tekstur Dengan Metode *Gray Level Co-Occurrence Matrix*







No	Citra Daging	Data Aplikasi	Hasil Validasi
1		Babi	Sesuai / Tidak Sesuai
2		Babi	Sesuai / Tidak Sesuai
3		Babi	Sesuai / Tidak Sesuai
4		Babi	Sesuai / Tidak Sesuai
5		Babi	Sesuai / Tidak Sesuai
6		Babi	Sesuai / Tidak Sesuai





No	Citra Daging	Data Aplikasi	Hasil Validasi
7		Babi	Sesuai / Tidak Sesuai
8		Babi	Sesuai / Tidak Sesuai
9		Babi	Sesuai / Tidak Sesuai
10		Babi	Sesuai / Tidak Sesuai
11		Babi	Sesuai / Tidak Sesuai
12		Babi	Sesuai / Tidak Sesuai

No	Citra Daging	Data Aplikasi	Hasil Validasi
13		Babi	Sesuai / Tidak Sesuai
14		Babi	Sesuai / Tidak Sesuai
15		Babi	Sesuai / Tidak Sesuai
16		Babi	Sesuai / Tidak Sesuai
17		Babi	Sesuai / Tidak Sesuai
18		Babi	Sesuai / Tidak Sesuai

No	Citra Daging	Data Aplikasi	Hasil Validasi
19		Babi	Sesuai / Tidak Sesuai
20		Babi	Sesuai / Tidak Sesuai
21		Sapi	Sesuai / Tidak Sesuai
22		Sapi	Sesuai / Tidak Sesuai
23		Sapi	Sesuai / Tidak Sesuai
24		Sapi	Sesuai / Tidak Sesuai

No	Citra Daging	Data Aplikasi	Hasil Validasi
25		Sapi	Sesuai / Tidak Sesuai
26		Sapi	Sesuai / Tidak Sesuai
27		Sapi	Sesuai / Tidak Sesuai
28		Sapi	Sesuai / Tidak Sesuai
29		Sapi	Sesuai / Tidak Sesuai
30		Sapi	Sesuai / Tidak Sesuai

No	Citra Daging	Data Aplikasi	Hasil Validasi
31		Sapi	Sesuai / Tidak Sesuai
32		Sapi	Sesuai / Tidak Sesuai
33		Sapi	Sesuai / Tidak Sesuai
34		Sapi	Sesuai / Tidak Sesuai
35		Sapi	Sesuai / Tidak Sesuai
36		Sapi	Sesuai / Tidak Sesuai







No	Citra Daging	Data Aplikasi	Hasil Validasi
37		Sapi	Sesuai / Tidak Sesuai
38		Sapi	Sesuai / Tidak Sesuai
39		Sapi	Sesuai / Tidak Sesuai
40		Sapi	Sesuai / Tidak Sesuai







Malang, Juli 2017







Penguji







6000
FARD FERDIAN-SYAH
 0852 3464 7049







Tabel Uji Validasi Data *Training* Aplikasi Klasifikasi Daging Babi dan Sapi Menggunakan Ekstraksi Warna dan Tekstur Dengan Metode *Gray Level Co-Occurrence Matrix*







No	Citra Daging	Data Aplikasi	Hasil Validasi
1		Babi	Sesuai / Tidak Sesuai
2		Babi	Sesuai / Tidak Sesuai
3		Babi	Sesuai / Tidak Sesuai
4		Babi	Sesuai / Tidak Sesuai
5		Babi	Sesuai / Tidak Sesuai
6		Babi	Sesuai / Tidak Sesuai





No	Citra Daging	Data Aplikasi	Hasil Validasi
7		Babi	Sesuai / Tidak Sesuai
8		Babi	Sesuai / Tidak Sesuai
9		Babi	Sesuai / Tidak Sesuai
10		Babi	Sesuai / Tidak Sesuai
11		Babi	Sesuai / Tidak Sesuai
12		Babi	Sesuai / Tidak Sesuai

No	Citra Daging	Data Aplikasi	Hasil Validasi
13		Babi	Sesuai / Tidak Sesuai
14		Babi	Sesuai / Tidak Sesuai
15		Babi	Sesuai / Tidak Sesuai
16		Babi	Sesuai / Tidak Sesuai
17		Babi	Sesuai / Tidak Sesuai
18		Babi	Sesuai / Tidak Sesuai

No	Citra Daging	Data Aplikasi	Hasil Validasi
19		Babi	Sesuai / Tidak Sesuai
20		Babi	Sesuai / Tidak Sesuai
21		Sapi	Sesuai / Tidak Sesuai
22		Sapi	Sesuai / Tidak Sesuai
23		Sapi	Sesuai / Tidak Sesuai
24		Sapi	Sesuai / Tidak Sesuai

No	Citra Daging	Data Aplikasi	Hasil Validasi
25		Sapi	Sesuai / Tidak Sesuai
26		Sapi	Sesuai / Tidak Sesuai
27		Sapi	Sesuai / Tidak Sesuai
28		Sapi	Sesuai / Tidak Sesuai
29		Sapi	Sesuai / Tidak Sesuai
30		Sapi	Sesuai / Tidak Sesuai

No	Citra Daging	Data Aplikasi	Hasil Validasi
31		Sapi	Sesuai / Tidak Sesuai
32		Sapi	Sesuai / Tidak Sesuai
33		Sapi	Sesuai / Tidak Sesuai
34		Sapi	Sesuai / Tidak Sesuai
35		Sapi	Sesuai / Tidak Sesuai
36		Sapi	Sesuai / Tidak Sesuai

No	Citra Daging	Data Aplikasi	Hasil Validasi
37		Sapi	Sesuai / Tidak Sesuai
38		Sapi	Sesuai / Tidak Sesuai
39		Sapi	Sesuai / Tidak Sesuai
40		Sapi	Sesuai / Tidak Sesuai



Malang, Juli 2017

Penguji



HARI SANTOSO

LAMPIRAN 5

Lembar Bimbingan Skripsi



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
Jl. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



NO SKRIPSI: 02

LEMBAR BIMBINGAN SKRIPSI 2016/2017

JUDUL : Klasifikasi Daging Babi dan Sapi Menggunakan Ekstraksi Fitur Warna dan Tekstur
Dengan Menggunakan Metode Gray Level Co-occurrence Matrix

Nama : Robertus Romario Harvey Kurniawan

NIM : 1341180117

No.	Tanggal	Materi Bimbingan	Tanda Tangan	
			Mahasiswa	Dosen
1.	01-03-2017	Pemantapan Konsep		
2.	08-03-2017	Penerapan Ekstraksi Warna HSV		
3.	10-03-2017	Segmentasi Image		
4.	16-03-2017	Konversi warna		
5.	22-03-2017	Cropping Image		
6.	05-04-2017	Data Set		
7.	12-04-2017	Penerapan GLCM		
8.	19-04-2017	Penerapan GLCM		
9.	26-04-2017	Percobaan Sistem		
10.	03-05-2017	Pembahasan About		
11.	22-05-2017	Testing Backpropagation		
12.	02-06-2017	Revisi Program		
13.	20-06-2017	Revisi Laporan		
14.	08-07-2017	Revisi Laporan		
15.	10-07-2017	Penilaian		
16.				
17.				
18.				
19.				

Malang, 10-07-2017
Dosen Pembimbing Skripsi,

Dr. Eng. Rosa Andrie Asmara, S.T., M.T
NIP. 198010102005011001



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
Jl. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



NO SKRIPSI: 02

LEMBAR BIMBINGAN SKRIPSI 2016/2017

JUDUL : Klasifikasi Daging Babi dan Sapi Menggunakan Ekstraksi Fitur Warna dan Tekstur
Dengan Menggunakan Metode Gray Level Co-occurrence Matrix

Nama : Robertus Romario Harvey Kurniawan

NIM : 1341180117

21/06/2017

No.	Tanggal	Materi Bimbingan	Tanda Tangan	
			Mahasiswa	Dosen
1.	08-04-2017	Pembahasan		DRUS
2.	29-04-2017	Revisi 1		DRUS
3.	10-05-2017	Aplikasi dan perhitungan GLCM		DRUS
4.	09-05-2017	Aplikasi dan perhitungan Entropi		DRUS
5.	20-05-2017	Aplikasi dan Proses BatchPro		DRUS
6.	05-05-2017	Pengujian Alat		DRUS
7.	15-05-2017	Laporan & PPT		DRUS
8.	26-05-2017	Laporan & PPT		DRUS
9.	29-05-2017	Pengujian Alat Akurasi		DRUS
10.	02-06-2017	Pengujian Akurasi		DRUS
11.	05-06-2017	Pembahasan Konten About		DRUS
12.	09-06-2017	Laporan & PPT		DRUS
13.	12-06-2017	Laporan & Jurnal		DRUS
14.	15-06-2017	Laporan & Jurnal		DRUS
15.	21-06-2017	Laporan PPT Jurnal		DRUS
16.				
17.				
18.				
19.				

Malang, 21-06-2017

Dosen Pembimbing Skripsi,

Kadec Sarjuna Batubulan S.Kom.,MT
NIP.

LAMPIRAN 6

Lembar Persetujuan Maju Sidang Skripsi



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
Jl. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



NO SKRIPSI : 02

LEMBAR PERSETUJUAN
MENGIKUTI UJIAN SKRIPSI 2016/2017
PROGRAM STUDI TEKNIK INFORMATIKA

N A M A : Robertus Romario Harvey Kurniawan N I M / K E L A S : 1341180117 / TI-4E
JUDUL SKRIPSI : KLASIFIKASI DAGING BABI DAN SAPI MENGGUNAKAN EKSTRAKSI FITUR WARNA DAN
TEKSTUR DENGAN METODE GRAY LEVEL CO-OCCURRENCE MATRIX
PEMBIMBING : 1. Dr. Eng. Rosa Andrie A., S.T., M.T. NIP : 19801010 200501 1 001
2. Kadek Suarjuna Batubulan, S.Kom., M.T. NIP :

No.	Uraian / Bab	Diselesaikan	Tanda Tangan	
			Pembimbing 1	Pembimbing 2
1.	PENDAHULUAN	✓		
2.	LANDASAN TEORI	✓		
3.	METODOLOGI PENELITIAN	✓		
4.	ANALISIS DAN PERANCANGAN	✓		
5.	IMPLEMENTASI	✓		
6.	PENGUJIAN DAN PEMBAHASAN	✓		
7.	KESIMPULAN DAN SARAN	✓		
8.	BAGIAN AKHIR - Daftar Pustaka - Lampiran (Isi lampiran disesuaikan dengan judul laporan akhir) - Profile Penulis (Riwayat Penulis)	✓		
9.	Hardware/Software - Didemonkan di depan pembimbing	✓		
10.	Draft Makalah	✓		

Malang,
Ketua Pelaksana LA & SKRIPSI 2016/2017
Program Studi Teknik Informatika

Arief Prasetyo, S.Kom., M.Kom.
NIP. 19790313 200812 1 002

DISETUJUI UNTUK DAPAT MAJU UJIAN SETELAH HASIL KARYA
DINILAI LAYAK SERTA HASIL UJI SESUAI DENGAN SPESIFIKASI
YANG DIRENCANAKAN

Pembimbing I

Dr. Eng. Rosa Andrie A., S.T., M.T.
NIP. 19801010 200501 1 001

Pembimbing II

Kadek Suarjuna B., S.Kom., M.T.

LAMPIRAN 7
Lembar Revisi Penguji



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
Jl. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



FORM REVISI SKRIPSI

No. Skripsi : 02

Nama Mahasiswa : Robertus Romario Harvey K. NIM : 1341180117
Tanggal Ujian : 17 Juli 2017
Judul : Klasifikasi Daging Babi Dan Sapi Menggunakan Ekstraksi
Fitur Warna Dan Tekstur Dengan Metode Gray Level Co-
Occurrence Matrix

NO	SARAN PERBAIKAN	PARAF
	<ul style="list-style-type: none">- Lakukan uji pbs daging dan potongan- Mel'in'ny dan. membujur- lakukan uji tng variabel $\frac{1}{2}$<ul style="list-style-type: none">- jarak- Cahay- Bobot daging- Lakukan perbaikan layout/klampops- sistem	

Malang, 17 Juli 2017
Dosen Penguji,

(.....)

FORM VERIFIKASI:

Skripsi telah diperbaiki sesuai dengan saran perbaikan dari dosen penguji.

PENGUJI/PEMBIMBING	NAMA	TTD	TANGGAL
Penguji			25-07-2017
Pembimbing 1	Dr. Eng. Rosa Andrie A., S.T., M.T.		25-07-2017
Pembimbing 2	Kadek Suarjuna Batubulan, S.Kom., M.T.		25-7/2017



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
Jl. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



FORM REVISI SKRIPSI

No. Skripsi : 02

Nama Mahasiswa : Robertus Romario Harvey K. NIM : 1341180117
Tanggal Ujian : Juli 2017
Judul : Klasifikasi Daging Babi Dan Sapi Menggunakan Ekstraksi
Fitur Warna Dan Tekstur Dengan Metode *Gray Level Co-Occurrence Matrix*

NO	SARAN PERBAIKAN	PARAF
1.	Buku laporan : - Sumber - alur perancangan lebih detail font lebih lagi Dr. Rosa Andrie keahli → kemas → contoh - tulisan di diagram kurang jelas	
2.	Aplikasi - uji coba dgn berbagai bentuk daging (bisa di bagian daging, potongan) - pengalihan keanekaragaman (susu & daging)	

Malang, 17 Juli 2017

Dosen/Penguji,

(Mangki A.)

FORM VERIFIKASI:

Skripsi telah diperbaiki sesuai dengan saran perbaikan dari dosen penguji.

PENGUJI/PEMBIMBING	NAMA	TTD	TANGGAL
Penguji	Mangki A.		24/07/2017
Pembimbing 1	Dr. Eng. Rosa Andrie A., S.T., M.T.		25/07/2017
Pembimbing 2	Kadek Suarjuna Batubulan, S.Kom., M.T.		25/07/2017



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
Jl. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



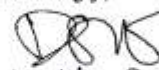
FORM REVISI SKRIPSI

No. Skripsi : 02

Nama Mahasiswa : Robertus Romario Harvey K. NIM : 1341180117
Tanggal Ujian : 17 Juli 2017
Judul : Klasifikasi Daging Babi Dan Sapi Menggunakan Ekstraksi
Fitur Warna Dan Tekstur Dengan Metode Gray Level Co-
Occurrence Matrix

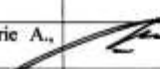
NO	SARAN PERBAIKAN	PARAF
1	Perubahan Laporan ✓	DRUS
2	Dataset diperbanyak ✓	DRUS

Malang, 17 Juli 2017
Dosen Penguji,


(...padelc S...)

FORM VERIFIKASI:

Skripsi telah diperbaiki sesuai dengan saran perbaikan dari dosen penguji.

PENGUJI/PEMBIMBING	NAMA	TTD	TANGGAL
Penguji			
Pembimbing 1	Dr. Eng. Rosa Andrie A., S.T., M.T.		25-7-2017
Pembimbing 2	Kadek Suarjuna Batubulan, S.Kom., M.T.	DRUS	21-7-2017

LAMPIRAN 8

Lembar Verifikasi Tata Tulis dan *Abstract*



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK INFORMATIKA
Jl. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



No. Skripsi : 02

FORM VERIFIKASI

ABSTRAK BAHASA INGGRIS DAN TATA TULIS BUKU SKRIPSI

Nama Mahasiswa : ROBERTUS ROMARIO HARVEY K. NIM : 1341180117
Tanggal Ujian : 17 Juli 2017
Judul : KLASIFIKASI DAGING BABI DAN SAPI MENGGUNAKAN
EKSTRAKSI FITUR WARNA DAN TEKSTUR DENGAN METODE *GRAY LEVEL CO-
OCCURRENCE MATRIX*

NO	BAGIAN YANG DIVERIFIKASI	NAMA VERIFIKATOR	TANGGAL VERIFIKASI	TTD
1	Abstrak Berbahasa Inggris	Rosa Andrie A	28-07-2017	
2	Tata Tulis Buku Skripsi	Rosa Andrie A	28-07-2017	

LAMPIRAN 9

Biodata Penulis

BIODATA PENULIS



Nama Lengkap	: Robertus Romario Harvey Kurniawan
Nomor Induk Mahasiswa	: 1341180117
Jurusan	: Teknologi Informasi
Program Studi	: Teknik Informatika
Jenis Kelamin	: Laki-laki
Tempat, tanggal lahir	: Blitar, 01 Mei 1995
Alamat asal	: Jl. Raya Timur 10 Talun, Blitar
No. Telepon	: 082143159599
Email	: rhk929@gmail.com
Agama	: Katolik
Nama Ayah	: Rudi Hartono Kurniawan
Nama Ibu	: Sri Susanti Indri Purwati

Riwayat Pendidikan

1999 – 2001	: TKK Katolik Santa Maria Blitar
2001 – 2007	: SD Katolik Santa Maria Blitar
2007 – 2010	: SMP Negeri 1 Blitar
2010 – 2013	: SMA Katolik Diponegoro Blitar
2013 – 2017	: D4 Teknik Informatika Politeknik Negeri Malang