

**IMPLEMENTASI SISTEM PENENTUAN KESEGERAN DAGING
SAPI LOKAL BERDASARKAN WARNA DAN KADAR AMONIA
DENGAN METODE JARINGAN SARAF TIRUAN BERBASIS
EMBEDDED SYSTEM**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Hamdan Bagus Firmansyah
NIM: 145150301111045



**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019**

PENGESAHAN

IMPLEMENTASI SISTEM PENENTUAN KESEGERAN DAGING SAPI LOKAL
BERDASARKAN WARNA DAN KADAR AMONIA DENGAN METODE JARINGAN SARAF
TIRUAN BERBASIS EMBEDDED SYSTEM

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :
Hamdan Bagus Firmansyah
NIIM: 145150301111045

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Januari 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I


Dahnial Syauqy, S.T., M.T., M.Sc.
NIK: 201607 870423 1 002

Dosen Pembimbing II


Mochammad Hanrats Hanafi Ichsan, S.ST, M.T
NIK. 201405 881229 1 001

Mengetahui
Ketua Jurusan Teknik Informatika



Tri Astoto Kurnidwan, S.T., M.T., Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 23 Januari 2019



Hamdan Bagus Firmansyah

NIM: 145150301111045

KATA PENGANTAR

Puja dan puji syukur penulis panjatkan kepada Allah SWT karena berkat rahmat dan hidayah-Nya, penulis dapat menyelesaikan penelitian dan Laporan Skripsi untuk memperoleh gelar Sarjana Komputer yang berjudul **Implementasi Sistem Penentuan Kesegaran Daging Sapi Lokal Berdasarkan Warna dan Kadar Amonia Dengan Metode Jaringan Saraf Tiruan Berbasis Embedded System**

Dalam pelaksanaan dan penyusunan laporan skripsi ini, tidak sedikit hambatan yang penulis hadapi. Namun penulis menyadari bahwa kelancaran dalam penyusunan laporan ini tidak lain berkat bantuan, dorongan dan bimbingan dari berbagai pihak, sehingga kendala-kendala yang penulis hadapi dapat teratasi. Penghargaan dan terima kasih yang sebesar-besarnya penulis sampaikan kepada:

1. Bapak Juarianto dan Ibu Nisfuanah selaku kedua orangtua penulis yang tiada henti memberikan semangat dan dorongan baik berupa do'a maupun materi selama penulis melakukan penelitian.
2. Bapak Dahnia Syauqy, S.T., M.T., M.Sc. selaku pembimbing I yang telah banyak membantu memberikan arahan dan bimbingan selama pengerjaan skripsi.
3. Bapak Mochammad Hannats Hanafi Ichsan, S.ST, M.T. selaku pembimbing II yang telah banyak membantu memberikan arahan dan bimbingan selama pengerjaan skripsi.
4. Dhuita Puspita yang selalu mengisi hari-hari penulis untuk menemani dan menyemangati dalam menyelesaikan penulisan skripsi ini.
5. Teman-teman seangkatan yang selalu memotivasi penulis untuk segera menyelesaikan penulisan skripsi ini.
6. Seluruh pihak yang tidak dapat disebutkan satu persatu yang telah berperan dalam penyelesaian skripsi ini.

Akhir kata penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan yang perlu disempurnakan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat penulis harapkan. Semoga skripsi ini bermanfaat bagi semua pihak yang membacanya.

Malang, 23 Januari 2019

Penulis

Hamdan Bagus Firmansyah

ABSTRAK

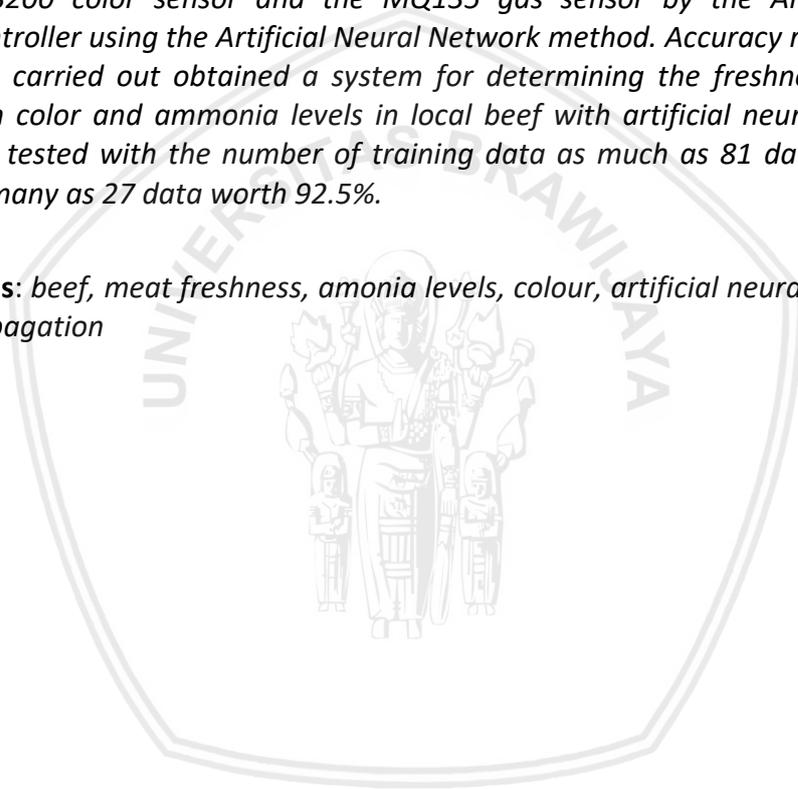
Jenis daging yang banyak dikonsumsi oleh manusia adalah daging sapi. Tingginya angka konsumsi daging sapi masyarakat disertai mahalnnya harga daging sapi mengakibatkan banyak pedagang yang mencampur daging sapi segar dengan daging sapi yang sudah busuk. Dampak mengonsumsi daging yang sudah busuk dapat berupa sakit diare dan keracunan. Dari permasalahan tersebut, diperlukan adanya penelitian yang terkait dengan sistem pendeteksi kesegaran daging guna mengurangi kerugian di kalangan masyarakat. Pada penelitian ini parameter yang digunakan adalah warna dan kadar amonia. Penelitian ini bertujuan untuk merancang sistem penentuan kesegaran daging sapi lokal berdasarkan warna dan kadar amonia. Proses penentuan kesegaran daging diperoleh dari nilai hasil pembacaan sensor warna TCS3200 dan sensor gas MQ135 oleh mikrokontroler Arduino Uno dengan menggunakan metode Jaringan Syaraf Tiruan. Hasil Akurasi dari pengujian yang dilakukan diperoleh sistem penentuan kesegaran daging sapi berdasarkan warna dan kadar amonia pada daging sapi lokal dengan metode jaringan syaraf tiruan yang diuji dengan jumlah data latih sebanyak 81 data dan data uji sebanyak 27 data adalah senilai 92,5 %.

Kata kunci: daging sapi, kesegaran daging, kadar amonia, warna, jaringan syaraf tiruan, *backpropagation*

ABSTRACT

Beef is the most meat that consumed by humans. High consumption of beef along with the high price of beef has caused many butcher mix fresh beef with rotten beef. The impact of consuming rotten meat can be diarrhea and poisoning. From these problems we need related research to make meat freshness detection systems to reduce disadvantages in the society. In this study the parameters used were color and ammonia levels. This study aims to design a system for determining local beef freshness based on color and ammonia levels. The process of determining the freshness of the meat is obtained from the reading results of the TCS3200 color sensor and the MQ135 gas sensor by the Arduino Uno microcontroller using the Artificial Neural Network method. Accuracy results from the tests carried out obtained a system for determining the freshness of beef based on color and ammonia levels in local beef with artificial neural network methods tested with the number of training data as much as 81 data and test data as many as 27 data worth 92.5%.

Keywords: *beef, meat freshness, ammonia levels, colour, artificial neural networks, backpropagation*



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i>	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Dasar Teori.....	6
2.2.1 Kesegaran Daging Sapi	6
2.2.2 Sensor Gas MQ135.....	8
2.2.3 Sensor Warna TCS3200	9
2.2.4 Mikrokontroler Arduino Uno (Atmega328P)	10
2.2.5 Metode Jaringan saraf tiruan.....	13
2.2.5.1 Algoritma Backpropagation	13
BAB 3 METODOLOGI	15
3.1 Studi literatur	16
3.2 Analisis Kebutuhan	16
3.2.1 Kebutuhan Perangkat Keras.....	16
3.3 Pengumpulan Data	17
3.4 Perancangan Sistem.....	17
3.5 Implementasi Sistem	17



3.6 Pengujian dan Analisis	17
3.7 Kesimpulan.....	17
BAB 4 REKAYASA KEBUTUHAN.....	18
4.1 Gambaran Umum Sistem.....	18
4.2 Kebutuhan Sistem	18
4.2.1 Kebutuhan Fungsional.....	18
4.2.2 Kebutuhan Non-Fungsional	18
4.2.3 Kebutuhan Perangkat Keras.....	19
4.2.4 Kebutuhan Perangkat Lunak	19
4.3 Batasan Perancangan Sistem.....	20
4.4 Asumsi dan Ketergantungan.....	20
BAB 5 perancangan dan implementasi	21
5.1 Perancangan Sistem.....	21
5.1.1 Perancangan Perangkat Keras	22
5.1.1.1 Perancangan Prototype Alat Pendeteksi Kesegaran Daging	22
5.1.1.2 Perancangan Subsistem Sensor TCS3200	23
5.1.1.3 Perancangan Subsistem Sensor MQ135	24
5.1.1.4 Perancangan Subsistem Output.....	25
5.1.1.5 Perancangan Push-Button	25
5.1.2 Perancangan Perangkat Lunak.....	26
5.1.2.1 Perancangan Pengambilan Data Sensor	26
5.1.2.2 Perancangan Model Jaringan Syaraf Tiruan	27
5.2 Implementasi Sistem	38
5.2.1 Implementasi Perangkat Keras	38
5.2.2 Implementasi Subsistem Sensor TCS3200	38
5.2.3 Implementasi Subsistem Sensor MQ135	39
5.2.4 Implementasi Subsistem Output	39
5.3 Implementasi Perangkat Lunak	40
5.3.1 Implementasi Python	40
5.3.2 Implementasi Arduino Uno.....	43
BAB 6 PENGUJIAN DAN ANALISIS.....	47



6.1 Pengujian Sensor Warna TCS3200.....	47
6.1.1 Tujuan Pengujian.....	47
6.1.2 Prosedur Pengujian	47
6.1.3 Hasil dan Analisis Pengujian.....	48
6.2 Pengujian Sensor Gas MQ135.....	51
6.2.1 Tujuan Pengujian.....	52
6.2.2 Prosedur Pengujian	52
6.2.3 Hasil dan Analisis Pengujian.....	52
6.3 Pengujian Akurasi Hasil Klasifikasi Jaringan Syaraf Tiruan	54
6.3.1 Tujuan Pengujian.....	54
6.3.2 Prosedur Pengujian	54
6.3.3 Hasil dan Analisis Pengujian.....	54
BAB 7 PENUTUP	56
7.1 Kesimpulan.....	56
7.2 Saran	56
Daftar Pustaka.....	58
LAMPIRAN A DATA PENELITIAN	59
LAMPIRAN B KODE PROGRAM.....	61



DAFTAR TABEL

Tabel 1.2.1 Jumlah Bakteri Daging Sapi Pada Suhu Ruang	6
Tabel 2.2 Pengaturan Skala Frekuensi Keluaran Sensor TCS3200	9
Tabel 2.3 Pengaturan pemilihan tipe Fotodiode Sensor TCS3200.....	9
Tabel 2.4 PinOut Arduino Uno Atmega328.....	10
Tabel 5.1 Keterangan koneksi pin Sensor TCS3200 dengan Arduino Uno.....	24
Tabel 5.2 Kelas Data Latih	30
Tabel 5.3 Program <i>Loader</i>	40
Tabel 5.4 Program Normalisasi Input dan Data	41
Tabel 5.5 Program <i>Generate Layer</i>	41
Tabel 5.6 Program <i>Generate Bias</i>	42
Tabel 5.7 Program Proses Penjumlahan	42
Tabel 5.8 Program Proses <i>Forward</i>	42
Tabel 5.9 Program Proses Update Bobot dan Bias	42
Tabel 5.10 Program Proses Inisialisasi Pin	43
Tabel 5.11 Program Proses Pembacaan Sensor	44
Tabel 5.12 Program Proses Testing dan Output	45
Tabel 6.1 Hasil pengujian pembacaan sensor warna TCS3200	49
Tabel 6.2 Data uji dan hasil pengujian Jaringan Syaraf Tiruan.....	54

DAFTAR GAMBAR

Gambar 2.1 Standar Warna Karkas Sapi	7
Gambar 2.2 Sensor Gas MQ135	8
Gambar 2.3 Grafik karakteristik Sensitivitas Gas dari MQ135	8
Gambar 2.4 Sensor Warna TCS3200	9
Gambar 2.5 Arduino Uno	10
Gambar 3.1 Diagram Alir Metode Penelitian	15
Gambar 5.1 Blok Diagram Perancangan Sistem	21
Gambar 5.2 Blok Diagram Keseluruhan Sistem	22
Gambar 5.3 Desain <i>Prototype</i> Alat Pendeteksi Kesegaran Daging Sapi	23
Gambar 5.4 Skematik Sensor TCS3200	23
Gambar 5.5 Skematik Sensor MQ135	24
Gambar 5.6 Skematik Sistem Output	25
Gambar 5.7 Diagram Alir Perancangan Perangkat Lunak	26
Gambar 5.8 Diagram Alir Algoritma Pelatihan Jaringan Backpropagation	33
Gambar 5.9 Diagram Alir Proses Umpan Maju (<i>Feedforward</i>)	35
Gambar 5.10 Diagram Alir Proses Umpan Mundur (<i>Backward Propagation</i>)	36
Gambar 5.11 Diagram Alir Proses Pengupdatetan Bobot Dan Bias	37
Gambar 5.12 Penempatan Komponen Pada Alat Pendeteksi	38
Gambar 5.13 Penempatan Sensor TCS3200	39
Gambar 5.14 Penempatan Sensor TCS3200	39
Gambar 5.15 Penempatan Sistem Output	40
Gambar 5.16 Contoh Data Training	43
Gambar 6.1 Analisis korelasi kadar Amonia dengan <i>Volt</i>	53

BAB 1 PENDAHULUAN

1.1 Latar belakang

Protein hewani merupakan zat gizi yang dibutuhkan oleh masyarakat dalam meningkatkan kualitas sumberdaya manusia. Salah satu sumber protein hewani adalah daging. Menurut SNI (1995) Daging merupakan urat yang melekat pada kerangka kecuali urat daging di bibir, hidung dan telinga yang berasal dari hewan sehat pada saat dipotong. Selain daging memiliki protein hewani tinggi, pada daging terdapat asam amino essensial yang lengkap serta beberapa mineral dan vitamin. Jenis daging yang banyak dikonsumsi oleh manusia adalah daging sapi (Soeparno, 2005).

Selama tahun 2011-2015 populasi sapi potong meningkat 2 kali dari pertumbuhan tahun sebelumnya yaitu sebesar 3,53% (Suryani, 2015). Meningkatnya populasi sapi juga berdampak pada peningkatan produksi daging sapi di Indonesia yakni sebesar 2,68%. Daging sapi memiliki ciri-ciri warna merah segar, seratnya halus dan lemaknya berwarna kuning (Buege, 2001). Penurunan kualitas daging dapat diketahui dari perubahan warna, rasa, aroma bahkan kebusukan. Sebagian besar kerusakan daging disebabkan oleh penanganan yang kurang baik sehingga memberikan peluang hidup bagi pertumbuhan dan perkembangan mikroba perusak yang berdampak pada berkurangnya daya simpan dan hilangnya nilai gizi. (Kasmadiharja, 2008) menyatakan bahwa kadar air yang meningkat dipengaruhi oleh jumlah air bebas yang terbentuk sebagai hasil samping dari aktivitas bakteri.

Berdasarkan data Survei Sosial Ekonomi Nasional (2014), konsumsi daging sapi Indonesia sebesar 2,08 kg/kapita/tahun. Tingginya angka konsumsi daging sapi masyarakat disertai mahalnya harga daging sapi mengakibatkan banyak pedagang daging sapi nakal yang mencampur antara daging sapi segar dengan daging sapi yang sudah busuk. Pedagang melakukan hal tersebut untuk memperoleh untung yang lebih besar. Perilaku pedagang yang curang dapat merugikan masyarakat serta mengancam kesehatan masyarakat Indonesia. Dampak mengonsumsi daging yang sudah busuk dapat berupa sakit diare dan keracunan daging. Munculnya kasus penjualan daging sapi busuk yang ada di kalangan masyarakat membuat mereka kesulitan memilih daging sapi segar yang layak dikonsumsi. Untuk melakukan identifikasi kesegaran daging diperlukan ilmu pengetahuan yang cukup, dimana tidak semua masyarakat menguasainya.

Kadar air dan kelembapan sebagai identifikasi daging busuk, bau atau kadar amonia juga bisa digunakan sebagai identifikasi daging sapi. Penelitian terkait pendeteksian amonia telah dilakukan dengan judul Deteksi Daging Sapi Menggunakan *Electronic Nose* Berbasis *Bidirectional Associative Memory* dimana dalam penelitian tersebut meneliti kadar amonia didalam daging sapi dan babi menggunakan beberapa sensor yaitu dari TGS 2610, TGS 2602, TGS 2620, TGS 2611, TGS 2600 itu merupakan beberapa sensor untuk deteksi kadar amonia dalam daging tersebut (Eviyan F, 2017).

Kesegaran daging Sapi dapat diidentifikasi melalui warnanya. Ada beberapa klasifikasi warna daging sapi, mulai dari warna terang hingga gelap. Daging sapi yang segar tentunya memiliki warna yang terang dan yang sudah tidak layak makan berwarna gelap. Tingkat kesegaran suatu daging akan menentukan apakah daging tersebut masih layak untuk dikonsumsi. Menurut SNI (2008) standart warna daging terdiri atas sembilan skor, dan ada tiga tingkat mutu. Tiga mutu itu yaitu daging segar, daging setengah segar dan daging busuk.

Dalam penentuan kesegaran daging sensor yang digunakan untuk mengidentifikasi yaitu sensor gas MQ135 dan sensor warna TCS3200. Sensor gas MQ135 yaitu sensor yang berfungsi untuk mendeteksi gas amonia. Sementara itu sensor TCS3200 yaitu sensor warna. Pada prinsipnya pembacaan warna pada TCS3200 dilakukan secara bertahap, selain itu sensor warna TCS3200 dapat membaca beberapa macam warna.

Dari kedua sensor tersebut ada mikrokontroler yang bertugas sebagai pengatur dan pengolah data yang di dapat pada kedua sensor tersebut. Mikrokontroler tersebut menggunakan Arduino Uno Atmega328. Keunggulan dari mikokontroler ini yaitu *open source*, mudah dalam pengoperasiannya, ukuran kecil dan mudah di bawa, dan banyak referensi dari komunitas-komunitas dari mikrokontroler tersebut.

Dalam penelitian ini mencoba meneliti kadar amonia dengan menggunakan sensor gas MQ135 dan menggunakan sensor warna TCS3200 sebagai idenfikasi daging sapi tersebut, pada penelitian ini menganalisa daging sapi busuk atau bukan dengan indikator yang didapat dari kedua sensor tersebut menggunakan metode Jaringan saraf tiruan.

1.2 Rumusan masalah

Berdasarkan latar belakang yang sudah dipaparkan maka dapat dirumuskan beberapa rumusan masalah sebagai berikut:

1. Bagaimana merancang sistem penentuan kesegaran daging sapi lokal berdasarkan warna dan kadar amonia dengan metode jaringan saraf tiruan
2. Bagaimana mengimplementasikan sistem penentuan kesegaran daging sapi lokal berdasarkan warna dan kadar amonia dengan metode jaringan saraf tiruan
3. Bagaimana hasil tingkat keakuratan sistem pendeteksi kesegaran daging berdasarkan warna dan kadar amonia dengan metode jaringan saraf tiruan

1.3 Tujuan

Penelitian pada tugas akhir ini memiliki tujuan sebagai berikut :

1. Mampu merancang sistem penentuan kesegaran daging sapi lokal berdasarkan warna dan kadar amonia
2. Mampu mengimplementasikan sistem penentuan kesegaran daging sapi lokal berdasarkan warna dan kadar amonia

3. Mampu mengetahui tingkat keakuratan sistem pendeteksi kesegaran daging sapi

1.4 Manfaat

Manfaat dalam penelitian ini adalah menghasilkan suatu sistem untuk memberi informasi mengenai identifikasi daging yang tidak segar sehingga meningkatkan kualitas kesehatan masyarakat dan membuat masyarakat lebih sadar akan kualitas daging yang akan diolahnya.

1.5 Batasan masalah

Agar penelitian terfokus, maka ada beberapa hal yang dibatasi yaitu:

1. Parameter untuk yang digunakan untuk menentukan tingkat kesegaran daging adalah warna dan gas yang dihasilkan oleh daging.
2. Objek yang diuji ialah daging sapi lokal khususnya pada otot daging kelapa.
3. Proses akuisisi data yaitu menggunakan mikrokontroler arduino.

1.6 Sistematika pembahasan

Sistematika pembahasan ditujukan untuk memberikan gambaran dan uraian mengenai langkah-langkah dalam proses penyusunan proposal penelitian adalah sebagai berikut:

BAB 1 : Pendahuluan

Menjelaskan tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan dari sistem penentuan kesegaran daging sapi lokal berdasarkan warna dan kadar amonia dengan metode jaringan saraf tiruan

BAB 2 : Landasan Kepustakaan

Pada bab ini akan memuat landasan teori dan pustaka yang terkait dengan penelitian yang dilakukan. Teori dan pustaka diambil dari buku literatur dan teori yang dibahas tentang penelitian kesegaran daging dan metode jaringan saraf tiruan.

BAB 3 : Metodologi

Membahas tentang langkah kerja yang dilakukan dalam penulisan diantaranya studi literatur, analisis kebutuhan sistem dan implementasi sistem penentuan kesegaran daging sapi lokal berdasarkan warna dan kadar amonia dengan metode jaringan saraf tiruan.

BAB 4 : Rekayasa Kebutuhan

Menguraikan secara rinci terkait gambaran umum sistem, analisis kebutuhan fungsional, kebutuhan perangkat keras, kebutuhan perangkat lunak dan batasan desain sistem dalam mengimplementasi

sistem penentuan kesegaran daging sapi lokal berdasarkan warna dan kadar amonia dengan metode jaringan saraf tiruan.

BAB 5 : Perancangan dan Implementasi

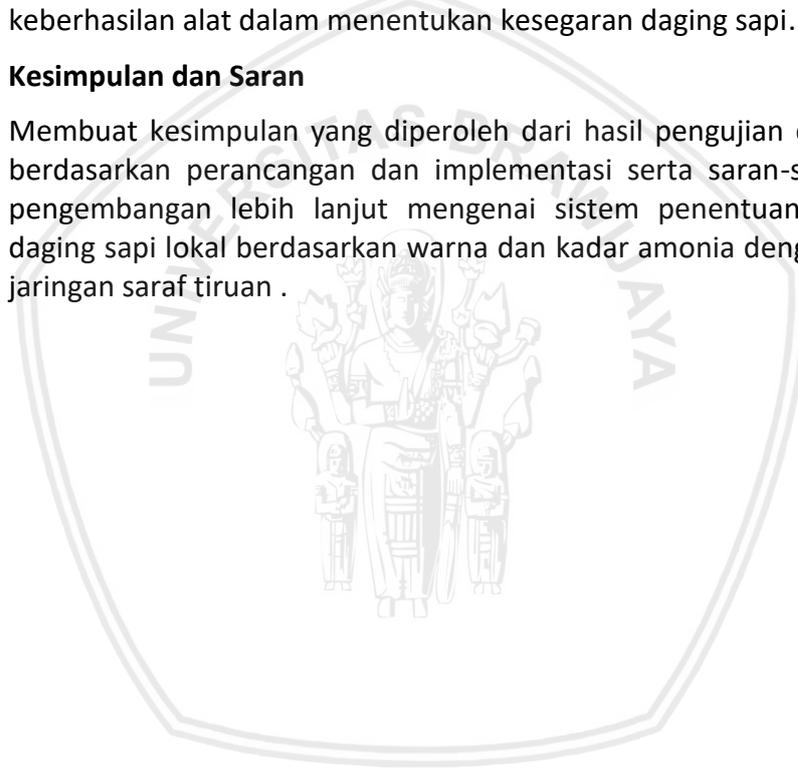
Membahas mengenai proses perancangan alat penentuan kesegaran daging sapi lokal berdasarkan warna dan kadar amonia dengan metode jaringan saraf tiruan, implementasi perangkat keras dan perangkat lunak serta metode Jaringan Syaraf Tiruan dalam melakukan klasifikasi kesegaran daging sapi.

BAB 6 : Pengujian dan Analisis

Membahas pengujian dan analisis berdasarkan perancangan dan implementasi. Pengujian ini dilakukan dengan menggunakan parameter keberhasilan alat dalam menentukan kesegaran daging sapi.

BAB 7 : Kesimpulan dan Saran

Membuat kesimpulan yang diperoleh dari hasil pengujian dan analisis berdasarkan perancangan dan implementasi serta saran-saran untuk pengembangan lebih lanjut mengenai sistem penentuan kesegaran daging sapi lokal berdasarkan warna dan kadar amonia dengan metode jaringan saraf tiruan .



BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepastakaan berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian. Dalam landasan kepastakaan terdapat landasan teori dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian. Jika dibutuhkan sesuai dengan karakteristik penelitiannya dan syarat kecukupan khusus keminatan tertentu, bisa juga terdapat kajian pustaka yang menjelaskan secara umum penelitian-penelitian terdahulu yang berhubungan dengan topik skripsi dan menunjukkan persamaan dan perbedaan skripsi tersebut terhadap penelitian terdahulu yang dituliskan.

2.1 Kajian Pustaka

Pada penelitian pertama dengan judul “Identifikasi Daging Segar dan Busuk Menggunakan Sensor Warna Rgb dan Ph Meter Digital”. Pada penelitian ini penulis mengidentifikasi daging segar dan busuk menggunakan sensor pengukuran warna RGB dan sensor pH digital. Sistem yang digunakan yaitu dengan cara daging dicairkan terlebih dahulu kemudian untuk menentukan kadar pH dari beda potensial pada elektroda gelas. Sementara pada pengukuran warna menggunakan sensor RGB dengan cara menggunakan sempel standart nasional dan yang bukan standart nasional sebagai data acuan. proses identifikasi pada daging segar, alat telah mampu mengidentifikasi dengan baik, sehingga rentang nilai yang digunakan untuk acuan identifikasi telah sesuai. Hal ini ditunjukkan dengan hasil pengukuran tidak mempunyai rentang nilai yang lebar, pada daging sapi non SNI nilai R berkisar antara 71-79, nilai G berkisar antara 31,5-35,5, dan nilai B berkisar antara 37,5-42,5; sedangkan kadar pH berkisar antara 4,72-4,89 sehingga merupakan asam dan termasuk dalam kadar asam lemah (Prasetyo E, 2014)

Penelitian kedua dengan judul “Identifikasi Kesegaran Daging Sapi Berdasarkan Citranya Dengan Ekstraksi Fitur Warna Dan Teksturnya Menggunakan Metode Gray Level Cooccurrence Matrix”. Pada penelitian ini memakai pemrosesan gambar dimana dari nilai RGB yang sudah didapat dikonversi ke *grayscale* setelah mendapatkan nilai peneliti langsung menggunakan metode *gray level cooccurrence matrix* sebagai penentuan tekstur daging. Pada penelitian ini daging yang diteliti yaitu antara daging sapi dan daging babi. Pada penelitian ini mencoba membandingkan antara metode *Jaringan Syaraf Tiruan* dan metode *Decision Tree J48*, Pada masing-masing metode terlihat bahwa parameter uji warna memiliki nilai akurasi tertinggi. Parameter uji warna pada metode Naïve Bayes memiliki nilai akurasi sebesar 95.83%, pada metode *Decission Tree J48* memiliki nilai akurasi sebesar 83.33% (Rosa A, 2017)

2.2 Dasar Teori

Pada dasar teori ini membahas tentang teori-teori yang mendukung atas penelitian yang akan disusun.

2.2.1 Kesegaran Daging Sapi

Daging sapi mengandung beberapa unsur yang membentuknya yaitu daging sapi mengandung air, protein, lemak. Dari beberapa kandungan tersebut, kandungan air yang paling besar persentasenya. Kandungan pada daging sapi yaitu protein 18,8 gram, air 66,0 gram, dan lemak 14,0. Sehingga sebagian besar pada daging kandungan air sangatlah besar, sehingga sifat kimia pada daging segar adalah basa.

Pembusukan daging bisa disebabkan oleh kontaminasi atau aktivitas mikroorganisme pada daging sapi tersebut. Parameter dari kebusukan daging sapi dapat diketahui dari aroma, tekstur, terbentuknya lendir dan terbentuknya gas. Penurunan kualitas daging sapi dapat dilihat dari warna permukaan daging sapi atau dari jumlah koloni mikroorganisme. Semakin banyak koloni mikroorganisme yang tumbuh pada daging tersebut maka semakin rendah kualitas daging sapi.

Menurut (Priharsanti, 2009) daging sapi yang disimpan pada suhu kamar mudah mengalami pembusukan. Pada suhu kamar, jumlah koloni bakteri mulai tampak meningkat pada penyimpanan 6 jam dan pada selang waktu 6 jam sampai 9 jam merupakan pertumbuhan paling cepat. Daging yang telah disimpan selama 9 jam pada suhu kamar memiliki jumlah koloni bakteri $124,3 \times 10^4$ /gram.

Tabel 1.2.1 Jumlah Bakteri Daging Sapi Pada Suhu Ruang

Lama Penyimpanan (Jam)	Jumlah Bakteri (10^4 /gram)
0	40,00
6	74,00
12	150,30
18	128,30

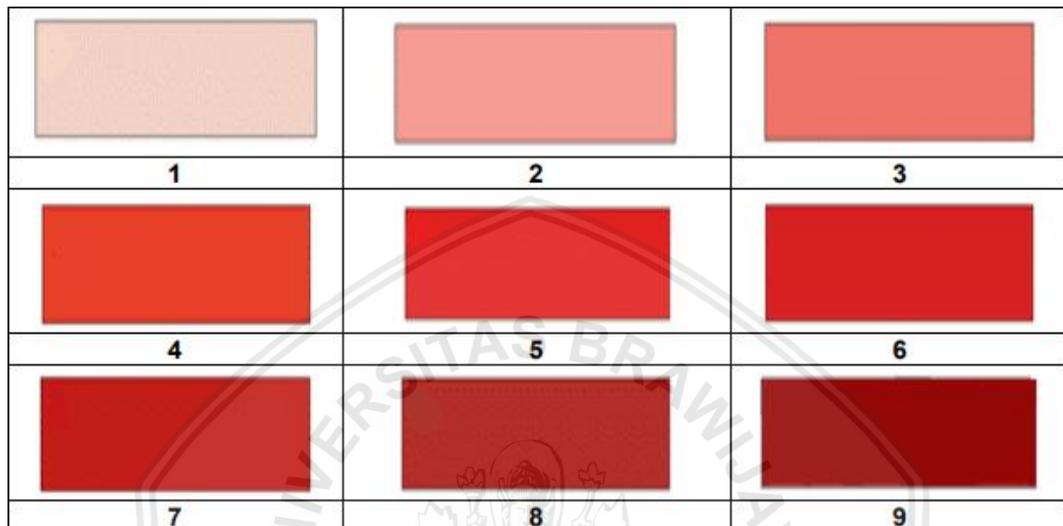
Sumber : (Priharsanti, 2009)

Pada Tabel 1.1 diatas dapat diklasifikasikan menjadi 3 tingkat kesegaran daging sapi, yaitu pada jam ke 0-6 merupakan daging dengan kualitas segar, pada jam ke 6-12 merupakan daging dengan kualitas sedang dan pada jam ke 12-18 merupakan daging dengan kualitas tidak segar atau sudah tidak layak untuk di konsumsi.

a. Warna

Daging yang segar memiliki warna merah yang cerah dan mengkilat, semakin menurunnya tingkat kesegaran maka warna semakin menjadi coklat kehijauan, kuning dan hampir tidak berwarna.

Menurut SNI (2008) standart warna daging terdiri atas sembilan skor, dan ada tiga tingkat mutu, yaitu tingkat mutu segar, tingkat mutu sedang dan tingkat mutu busuk. Tingkat mutu I yaitu skor warna 1-5, tingkat mutu II yaitu skor 6-7, dan tingkat mutu III yaitu skor 8-9. Sementara pada warna terdiri dari merah muda hingga merah tua seperti terlihat dalam gambar 2.1.



Gambar 2.1 Standar Warna Karkas Sapi
Sumber: (Badan Standardisasi Nasional, 2008)

b. Bau

Daging segar tidak begitu berbau masam/busuk, tetapi berbau khas daging yang segar, semakin busuk maka daging tersebut semakin berbau masam/busuk, sehingga kadar amoniaknya tinggi.

c. Tekstur

Daging segar memiliki tekstur kenyal, padat dan tidak kaku, bila ditekan dengan tangan makan bekas tekanan tersebut akan segera kembali ke posisi semula. Daging yang tidak baik ditandai dengan tekstur yang lunak dan bila ditekan sangat mudah hancur.

d. Kenampakan

Daging sapi yang segar tidak berlendir, jika dipegang tidak lengket, dan sebaliknya jika daging busuk akan berlendir, terasa lengket jika di pegang selain itu warna daging juga mulai kusam, kotor dan putih kehijauan dikarenakan kegiatan mikroba pada daging tersebut.

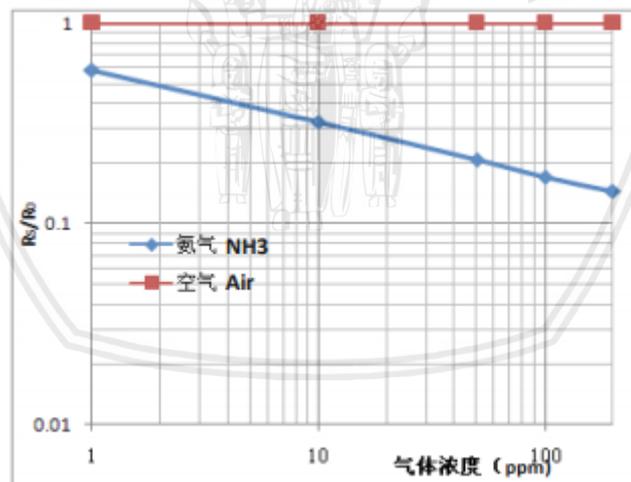
2.2.2 Sensor Gas MQ135



Gambar 2.2 Sensor Gas MQ135

Sumber (Sainsmart, 2018)

Sensor ini dapat mendeteksi beberapa jenis gas, namun pada sensor MQ135 ini memiliki fungsi khusus yaitu untuk mendeteksi sensor gas amonia. (MQ135_datasheet,2016). Sensor ini memiliki keunggulan yaitu sensitif terhadap amonia, kehandalan dengan biaya rendah. Sensor gas dapat bekerja jika diberi tegangan input sebesar 5V. Tingkat kandungan amonia yang dapat di ukur oleh sensor ini pada rentang 5-200ppm NH_3 .



Gambar 2.3 Grafik karakteristik Sensitivitas Gas dari MQ135

Sumber : (MQ135_Datasheet, 2016)

Pada grafik yang terlihat pada Gambar 2.2 terlihat bahwa nilai sensitivitas untuk beberapa gas tersebut yaitu sebesar 100ppm di udaranya yang bersih. Namun pembeda dari grafik ketiganya adalah nilai resistansi yang digunakan, yaitu perbandingan antara R_s/R_o

2.2.3 Sensor Warna TCS3200



Gambar 2.4 Sensor Warna TCS3200

Sumber: (Alexnld, 2018)

TCS3200 ialah IC pengkonversi warna cahaya ke nilai frekuensi. Ada dua komponen utama pembentuk IC ini, yaitu photodiode dan pengkonversi arus ke frekuensi. Keluaran pada IC ini yaitu berupa digital berbentuk pulsa pulsa hasil dari pembacaan dari warna RGB. Berikut karakteristik dari sensor yang ditunjukkan pada Tabel 2.1 dan Tabel 2.2.

Tabel 2.2 Pengaturan Skala Frekuensi Keluaran Sensor TCS3200

S0	S1	Skala Frekuensi Keluaran
L	L	Daya Mati
L	H	2%
H	L	20%
H	H	100%

Sumber : (TAOS, 2009)

Tabel 2.3 Pengaturan pemilihan tipe Fotodiode Sensor TCS3200

S2	S3	Tipe Fotodiode
L	L	Red
L	H	Blue
H	L	No Filter
H	H	Green

Sumber : (TAOS, 2009)

2.2.4 Mikrokontroler Arduino Uno (Atmega328P)



Gambar 2.5 Arduino Uno

Sumber : (Arduino, 2016)

Arduino Uno yaitu mikrokontroler yang sifatnya *open source*, di dalam mikrokontroler ini berjenis chip atau IC yang bisa diprogram menggunakan komputer. Pada Arduino Uno ini pengembang bisa mengubah kode program melalui komputer dengan aplikasi yang sudah disediakan oleh arduino. Arduino uno ini mempunyai 14 pin digital input/output, 6 pin analog, sebuah kristal osilator, port USB, power jack dan sebuah tombol reset (arduino,2016). Spesifikasi dari Arduino Uno Atmega328 dan deskripsi masing-masing pin ditunjukkan pada Tabel 2.3 berikut:

Tabel 2.4 PinOut Arduino Uno Atmega328

Mikrontroler	Atmega328P
Tegangan Operasi	5V
Tegangan Input (direkomendasikan)	7-12V
Tegangan Input (batas)	6-20V
Pin I/O Digital	14 (6 diantaranya menunjang output PWM)
Pin I/O PMW Digital	6
Pin Analog Input	6
Arus DC tiap Pin I/O	20 mA
Arus DC untuk Pin 3.3V	50 mA
Memori Flash	32 KB (Atmega328P)
	Dimana 0.5 KB digunakan oleh boatloader
SRAM	2 KB (Atmega328P)
EEPROM	1 KB (Atmega328P)

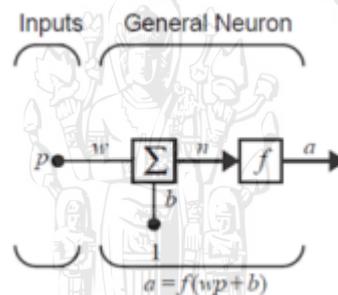
Kecepatan Clock	16 MHz
Panjang	68.6 mm
Lebar	53.4 mm
Berat	25 g

Sumber: (Arduino, 2016)

2.2.5 Neural Network

Neural network merupakan sebuah paradigma pemrosesan informasi yang terinspirasi oleh cara sistem saraf biologis seperti otak dalam memproses informasi. Bagian penting dari paradigma ini adalah struktur baru dari sistem pemrosesan informasi. *Neural network* terdiri atas sejumlah besar neuron yang saling terhubung yang bekerja sebagai satu kesatuan untuk memecahkan suatu masalah. Sama seperti manusia, *neural network* belajar berdasarkan contoh atau pengalaman. *Neural network* dapat diatur untuk sebuah aplikasi yang spesifik seperti *pattern recognition* atau data *classification* melalui proses *learning*.

2.2.5.1 Struktur dasar Neural Network



Gambar 2.6 Struktur dasar dari *neural network*.

Gambar 2.6 menunjukkan sebuah neuron dengan input tunggal dimana struktur ini merupakan bagian paling sederhana dari sebuah *neural network*. Input p akan dikalikan dengan sebuah weight yang dilambangkan dengan w yang akan menghasilkan wp , yang merupakan salah satu input yang akan dimasukkan menuju ke sebuah summer. Input lainnya dari summer adalah angka 1 yang dikalikan dengan sebuah bias yang dilambangkan dengan b dan diinputkan menuju ke summer. Summer akan menjumlahkan seluruh input yang masuk kepadanya dan menghasilkan output yang dilambangkan sebagai n . Selanjutnya output dari summer yaitu n akan masuk menuju ke sebuah transfer function yang dilambangkan dengan f , yang akan menghasilkan output akhir dari neuron yang dilambangkan sebagai a . Diagram diatas dapat ditulis menjadi persamaan matematis sebagai

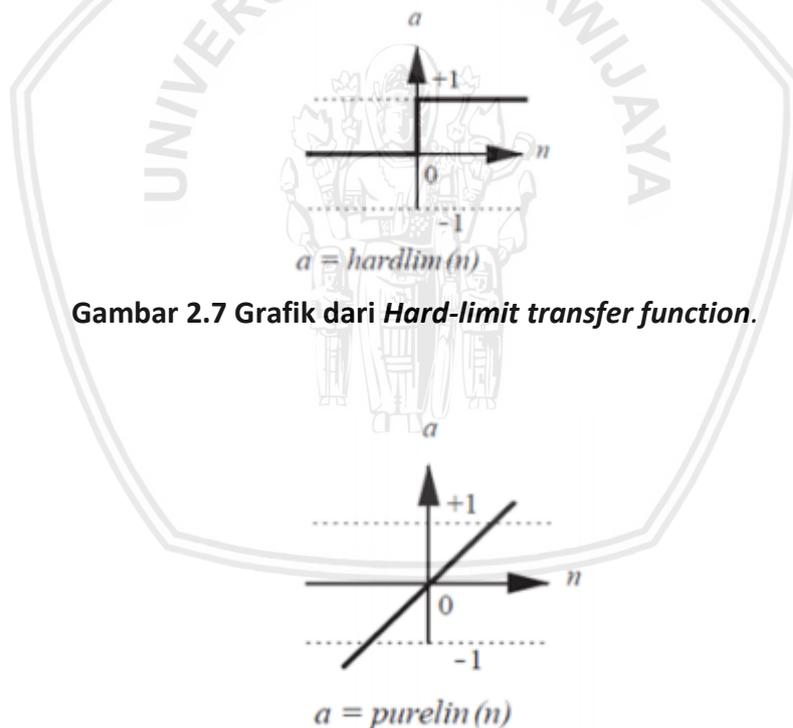
$$a = f(wp + b) \quad (2.1)$$

dimana a merupakan output dari neuron, w adalah *weight*, p adalah input dari neuron, b adalah bias dan f adalah *transfer function*. Sebuah *weight* adalah bobot yang akan mempengaruhi nilai input yang akan masuk menuju ke neuron.

Sedangkan bias hampir sama seperti *weight*, hanya saja ia selalu mempunyai input konstan yang bernilai 1. Dalam beberapa kasus, bias dapat dihilangkan, namun sebuah *neural network* dengan bias terbukti jauh lebih baik dengan *neural network* tanpa bias.

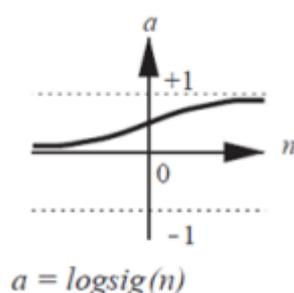
2.2.5.2 Transfer Function

Terdapat beberapa *transfer function* yang umum digunakan dalam *neural network* diantaranya adalah *hardlimit* TF, linear TF, Log-sigmoid TF dan Tan-sigmoid TF. Pemilihan jenis transfer function bergantung pada jenis permasalahan yang akan neuron coba untuk selesaikan. *Hard-limit transfer function* mempunyai karakteristik yaitu hanya mempunyai dua buah nilai output yaitu 0 dan 1. Jika input yang masuk ke transfer function bernilai kurang dari 0 maka output akan bernilai 0 sedangkan jika input bernilai lebih dari atau sama dengan nol maka output akan bernilai 1. Transfer function ini digunakan untuk membuat sebuah neuron yang akan mengklasifikasikan input ke dalam dua kategori yang berbeda.



Gambar 2.7 Grafik dari *Hard-limit transfer function*.

Gambar 2.8 Grafik dari Linear transfer function.



Gambar 2.9 Grafik dari *Log-sigmoid transfer function*.

Linear transfer function memiliki karakteristik yaitu input yang masuk ke transfer function adalah sama dengan output dari transfer function.

Log-sigmoid transfer function memiliki karakteristik yaitu dapat menerima input yang dapat bernilai dari $-\infty$ hingga $+\infty$ dan menyusutkan nilai tersebut sebagai output dengan rentang nilai antara 0 dan 1. Rumus matematika dari Logsigmoid transfer function adalah sebagai berikut :

$$a = \frac{1}{1+e^n} \quad (2.2)$$

Log-sigmoid transfer function umumnya digunakan dalam *multilayer networks* yang dilatih dengan menggunakan algoritma backpropagation, dikarenakan fungsi mempunyai turunan yang unik.

2.2.6 Metode Jaringan saraf tiruan

Jaringan Syaraf Tiruan merupakan sebuah paradigma pemrosesan informasi yang terinspirasi oleh cara sistem saraf biologis seperti otak dalam memproses informasi. Bagian penting dari paradigma ini adalah struktur baru dari sistem pemrosesan informasi. Jaringan Syaraf Tiruan terdiri atas sejumlah besar neuron yang saling terhubung yang bekerja sebagai satu kesatuan untuk memecahkan suatu masalah. Sama seperti manusia, Jaringan Syaraf Tiruan belajar berdasarkan contoh atau pengalaman. Jaringan Syaraf Tiruan dapat diatur untuk sebuah aplikasi yang spesifik seperti pattern recognition atau data classification melalui proses learning (Simamora, 2017).

2.2.6.1 Algoritma Backpropagation

Tujuan utama dalam pengembangan model Jaringan Syaraf Tiruan adalah untuk mencari sebuah himpunan optimal dari parameter weight dan bias sehingga fungsi dari Jaringan Syaraf Tiruan dapat mendekati atau mewakili perilaku dari permasalahan yang aslinya. Hal ini dapat dilakukan melalui proses yang disebut sebagai training. Pada algoritma backpropagation sekumpulan data training disediakan untuk Jaringan Syaraf Tiruan (Simamora, 2017).

Data training adalah berupa pasangan dari

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\}$$

dimana pQ adalah input dari Jaringan Syaraf Tiruan dan tQ merupakan target atau output yang diinginkan dari Jaringan Syaraf Tiruan. Selama training, performa dari Jaringan Syaraf Tiruan dievaluasi dengan menghitung selisih antara output yang dihasilkan oleh Jaringan Syaraf Tiruan dengan output yang diinginkan untuk semua sampel dari data training. Nilai selisih ini dinyatakan sebagai error dan dapat dirumuskan sebagai **Persamaan (2.2)** berikut:

$$J = \frac{1}{2} \sum_{j=1}^i (t_j - \alpha_j)^2 \quad (2.2)$$

atau yang sering dikenal sebagai Mean Squared Error (MSE) (Simamora, 2017).

Tahap-tahap dalam algoritma backpropagation adalah sebagai berikut :

1. Forward Propagation

Forward propagation adalah tahap dimana sinyal yang terdapat pada layer input diteruskan sampai menuju ke layer output. Proses ini diawali dengan menginisialisasi nilai weight dan bias pada setiap neuron (Simamora, 2017).

Umumnya pemilihan nilai weight dan bias ini adalah berupa nilai acak antara 0 sampai 1. Proses yang terjadi pada setiap neuron adalah sebagai **Persamaan (2.3)** berikut:

$$n_i = \sum_{j=1}^k w_{ij} p_j + b_j \quad (2.3)$$

Kemudian nilai n akan dimasukkan menuju ke transfer function yang menghasilkan output dari neuron sebagai **Persamaan (2.4)** berikut:

$$a = \frac{1}{1 + e^{-ni}} \quad (2.4)$$

Setelah forward propagation selesai maka didapatkan output dari Jaringan Syaraf Tiruan. Langkah selanjutnya adalah menghitung local gradient pada masing-masing neuron.

Log-sigmoid transfer function mempunyai turunan sebagai **Persamaan (2.5)** berikut

$$f(n) = f(n)[1 - f(n)] \quad (2.5)$$

sehingga nilai gradient dari neuron dapat dihitung dengan rumus **Persamaan (2.6)** berikut:

$$\delta_j = \delta_{inj} * f(n) \quad (2.6)$$

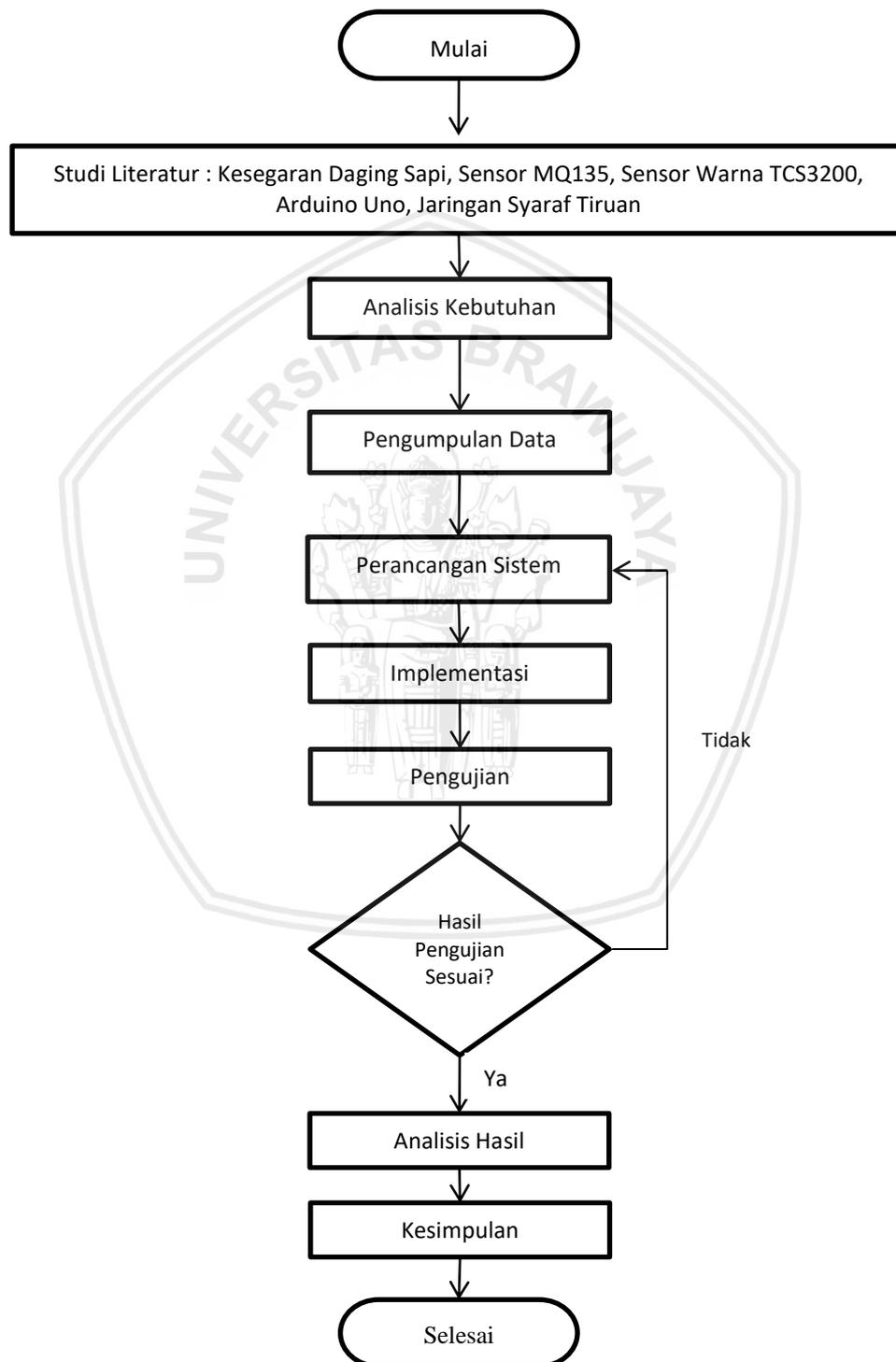
Tahap terakhir dalam algoritma backpropagation adalah melakukan update nilai weight dan bias seperti **Persamaan (2.7)** dan **Persamaan (2.8)** berikut:

$$w(n+1) = w(n) + \varphi * \delta(n) * y \quad (2.7)$$

$$b(n+1) = b(n) + \varphi * \delta(n) * 1 \quad (2.8)$$

BAB 3 METODOLOGI

Pada bab ini menjelaskan langkah – langkah yang dilakukan dalam penyusunan skripsi. Metode penelitian yang digunakan pada skripsi ini ditunjukkan pada diagram alir pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Studi literatur

Pada bagian ini dibahas mengenai dasar teori yang mendukung penelitian. Teori-teori yang digunakan dalam tahapan studi literatur ini berasal dari buku, jurnal, website resmi, artikel, serta e-book. Berikut merupakan dasar teori yang digunakan sebagai bahan studi:

1. Kesegaran Daging Sapi
2. Sensor Gas MQ135
3. Sensor Warna TCS3200
4. Mikrokontroler Arduino Uno Atmega328P
5. Jaringan Syaraf Tiruan

3.2 Analisis Kebutuhan

Analisis kebutuhan diperlukan untuk menganalisis apa saja yang dibutuhkan oleh sistem pada penelitian yang dilakukan, sehingga sistem dapat melakukan sesuai dengan yang diharapkan. Berikut beberapa kebutuhan fungsional yang dibutuhkan oleh sistem dalam penelitian ini :

1. Sistem dapat mendeteksi warna. Warna dapat dideteksi oleh system dengan menggunakan sensor TCS3200 guna sebagai parameter untuk pemilihan kualitas daging.
2. Sistem dapat mendeteksi gas amonia. Gas amonia yang terdapat pada daging sapi local akan dapat dibaca dengan menggunakan sensor MQ135 sebagai pendeteksi tingkat kesegaran daging sapi lokal.
3. Sistem dapat menentukan kesegaran daging sapi dari data-data yang telah di sensing oleh kedua sensor tersebut, yaitu sensor TCS3200 dan MQ135 dengan menggunakan metode jaringan syaraf tiruan.
4. Sistem dapat menampilkan hasil penentuan kesegaran daging pada LCD Display yang sudah disediakan, sehingga pengguna alat bisa mengetahui.

3.2.2 Kebutuhan Perangkat Keras

Berikut beberapa kebutuhan perangkat keras yang digunakan untuk membuat sisem pada penelitian ini :

1. Mikrokontroler Arduino Uno Atmega328
2. Sensor Warna TCS3200
3. Sensor Gas MQ135
4. Kabel Jumper
5. LCD display

6. Laptop

3.3 Pengumpulan Data

Proses pengumpulan data yang dilakukan yaitu untuk mengumpulkan data sebagai data latih, pengambilan data latih ini yaitu hanya dibutuhkan tiga sampai empat hari saja, karena daging yang sifatnya mudah busuk. Sementara pada proses pengumpulan data, maka secara tidak langsung akan mengikuti juga pada proses perancangan sistem, karena pengambilan data latih menggunakan alat yang akan dibuat.

3.4 Perancangan Sistem

Pada perancangan ini, hanya membutuhkan perancangan perangkat keras dan perangkat lunak dengan menggunakan logika sistem yang akan dibuat. Pada perancangan perangkat lunak, algoritma yang digunakan yaitu algoritma jaringan syaraf tiruan, dengan menggunakan metode *backpropagation*.

3.5 Implementasi Sistem

Pada implementasi ini dilakukan dengan mengacu pada perancangan yang sudah dibuat sebelumnya dengan memodifikasi agar sesuai dengan sistem yang dibutuhkan. Pada implementasi juga mencantumkan gambar sistem yang dibuat.

3.6 Pengujian dan Analisis

Pada tahap ini ada beberapa tahapan yang akan dilakukan untuk pengujian sistem, yaitu :

1. Pengujian sensor warna TCS3200 apakah dapat bekerja sesuai dengan spesifikasinya.
2. Pengujian sensor gas MQ135 apakah dapat bekerja sesuai dengan spesifikasinya.
3. Pengujian akurasi hasil identifikasi jaringan syaraf tiruan.

3.7 Kesimpulan

Tahap ini merupakan tahap akhir setelah semua tahap sebelumnya pada metodologi penelitian ini dilaksanakan. Pada tahap ini dilakukan penarikan kesimpulan berdasarkan analisis hasil dari pengujian. Dalam tahap ini pula peneliti menyampaikan saran yang untuk pengembangan penelitian yang serupa selanjutnya baik dari segi penulisan, metode, maupun perangkat-perangkat yang digunakan.

BAB 4 REKAYASA KEBUTUHAN

Pada bab ini akan menjelaskan tentang berbagai kebutuhan yang digunakan oleh sistem, mulai dari gambaran umum sistem, kebutuhan sistem, batasan perancangan, serta asumsi dan ketergantungan pada sistem.

4.1 Gambaran Umum Sistem

Pada sistem yang akan dibuat secara umum memiliki fungsi yaitu sebagai alat pendeteksi kesegaran daging sapi dengan menggunakan sensor warna dan sensor gas. Pada penentuan daging sapi ini, menggunakan metode jaringan syaraf tiruan dengan algoritman *backpropagation*. Pada sistem ini, daging akan dideteksi pada box yang tertutup yang didalamnya dipasangkan 2 sensor sehingga pada sensor gas bisa mendeteksi daging tanpa adanya kontaminasi udara dari luar sistem. Setelah sensor mendapatkan data maka akan diteruskan ke mikrokontroler untuk diproses dan dapat menentukan tingkat kesegaran daging sapi tersebut.

4.2 Kebutuhan Sistem

Pada kebutuhan sistem ini akan dijelaskan tentang apa saja yang dibutuhkan sistem mulai dari kebutuhan fungsional, kebutuhan perangkat lunak dan kebutuhan perangkat keras.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional ialah kebutuhan untuk mengetahui fungsi-fungsi yang harus dicapai dari penelitian ini. Adapun kebutuhan fungsional yang terdapat pada penelitian ini adalah sebagai berikut :

1. Sistem dapat mendeteksi warna. Warna dapat dideteksi oleh sistem dengan menggunakan sensor TCS3200 guna sebagai parameter untuk pemilihan kualitas daging.
2. Sistem dapat mendeteksi gas amonia. Gas amonia yang terdapat pada daging sapi local akan dapat dibaca dengan menggunakan sensor MQ135 sebagai pendeteksi tingkat kesegaran daging sapi lokal.
3. Sistem dapat menentukan kesegaran daging sapi dari data-data yang telah di sensing oleh kedua sensor tersebut, yaitu sensor TCS3200 dan MQ135 dengan menggunakan metode jaringan syaraf tiruan.
4. Sistem dapat menampilkan hasil penentuan kesegaran daging pada LCD Display yang sudah disediakan, sehingga pengguna alat bisa mengetahui.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan fungsional ialah kebutuhan yang harus dicapai dari penelitian ini. Adapun kebutuhan non fungsional yang terdapat pada penelitian ini adalah sebagai berikut :

1. Menghitung kadar amonia yang ada di dalam kandungan daging sapi.
2. Menghitung kadar warna yang ada pada potongan daging sapi yang diuji.
3. Ketepatan dalam penghitungan menggunakan metode *backpropagation*.

4.2.3 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras ialah kebutuhan untuk mengetahui apa saja yang akan digunakan pada sistem, Adapun perangkat keras yang dibutuhkan adalah sebagai berikut :

1. Mikrokontroller Arduino Uno

Mikrokontroller Arduino Uno digunakan sebagai modul pemrosesan data-data yang didapat dari sensor TCS3200 dan MQ135.

2. Sensor TCS3200

Sensor TCS3200 digunakan sebagai pendeteksi warna, pada TCS ini mengeluarkan data-data berupa nilai dari RGB daging sapi tersebut.

3. Sensor MQ135

Sensor MQ135 digunakan sebagai pendeteksi gas amonia, di mana pada daging akan mengeluarkan gas amonia, dari aktivitas bakteri yang ada di dalam daging sapi.

4. LCD DISPLAY

LCD Display ini digunakan untuk menampilkan data yang sudah diproses oleh mikrokontroller agar pengguna bisa mengetahui hasil dari pemrosesan tersebut.

4.2.4 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak ialah kebutuhan perangkat lunak apa saja yang digunakan pada sistem. Pada penelitian ini ada beberapa perangkat lunak yang digunakan yaitu :

1. Arduino IDE

Arduino IDE digunakan untuk memprogram mikrokontroller Arduino uno agar bekerja sebagaimana semestinya.

2. Visual Studio Code

Visual Studio Code atau VCS merupakan *text cross platform* yang gunanya bisa mendukung banyak bahasa pemrograman, pada penelitian ini VSC digunakan sebagai *platform* untuk menggunakan bahasa python.

3. *Library LiquidCrystal_I2C.h*

Library ini digunakan untuk menampilkan pada layar LCD yang sudah dipasangkan, kegunaan *library* ini digunakan untuk memperingkas *wiring* pada arduino agar pin-pin pada arduino tidak sampai kekurangan.

4. *Library TimerOne.h*

Library ini digunakan untuk mengatur interupsi pada sensor TCS3200, *library* ini menjalankan fungsi *TCS_Callback* sebagai fungsi interupsi.

5. *library math.h*

Library math.h ini untuk melakukan perhitungan matematika yang cukup rumit.

4.3 Batasan Perancangan Sistem

Pada tahap ini membahas tentang batasan yang digunakan agar penelitian lebih terarah sesuai tujuan. Adapun batasan –batasan perancangan sistem adalah sebagai berikut :

1. Mikokontroller yang digunakan ialah Arduino Uno.
2. Sensor yang digunakan adalah Sensor TCS3200 dan MQ135.
3. Sistem ditekankan pada keakuratan penentuan kesegaran daging sapi

4.4 Asumsi dan Ketergantungan

Asumsi dan ketergantungan adalah dugaan sistem yang akan beroperasi nantinya, adapun asumsi dan ketergantungan adalah sebagai berikut :

1. Objek yang digunakan hanyalah daging sapi lokal.
2. Objek yang diuji berada pada suhu ruangan.
3. Sistem akan mendeteksi pada ruang tertutup, yaitu pada suatu box yang terdapat dua sensor tersebut.
4. Daging sapi yang digunakan hanya potongan kecil yaitu minimal berukuran 7x7 cm dengan ketebalan minimal 1,5 cm.

4. *Library TimerOne.h*

Library ini digunakan untuk mengatur interupsi pada sensor TCS3200, *library* ini menjalankan fungsi *TCS_Callback* sebagai fungsi interupsi.

5. *library math.h*

Library math.h ini untuk melakukan perhitungan matematika yang cukup rumit.

4.3 Batasan Perancangan Sistem

Pada tahap ini membahas tentang batasan yang digunakan agar penelitian lebih terarah sesuai tujuan. Adapun batasan –batasan perancangan sistem adalah sebagai berikut :

1. Mikokontroller yang digunakan ialah Arduino Uno.
2. Sensor yang digunakan adalah Sensor TCS3200 dan MQ135.
3. Sistem ditekankan pada keakuratan penentuan kesegaran daging sapi

4.4 Asumsi dan Ketergantungan

Asumsi dan ketergantungan adalah dugaan sistem yang akan beroperasi nantinya, adapun asumsi dan ketergantungan adalah sebagai berikut :

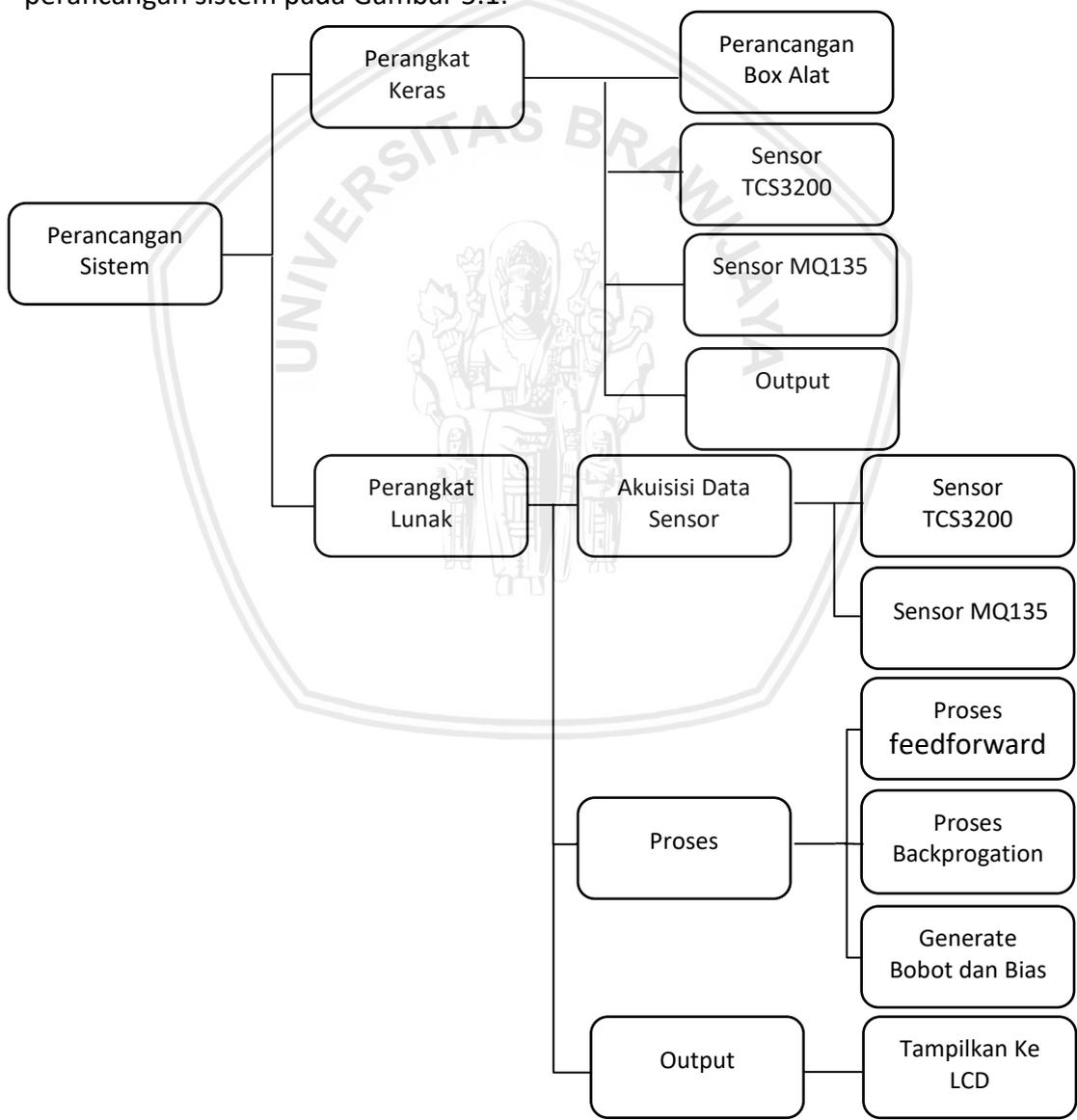
1. Objek yang digunakan hanyalah daging sapi lokal.
2. Objek yang diuji berada pada suhu ruangan.
3. Sistem akan mendeteksi pada ruang tertutup, yaitu pada suatu box yang terdapat dua sensor tersebut.
4. Daging sapi yang digunakan hanya potongan kecil yaitu minimal berukuran 7x7 cm dengan ketebalan minimal 1,5 cm.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini menguraikan proses implementasi dari dasar teori yang telah dipelajari sesuai analisis dan perancangan sistem.

5.1 Perancangan Sistem

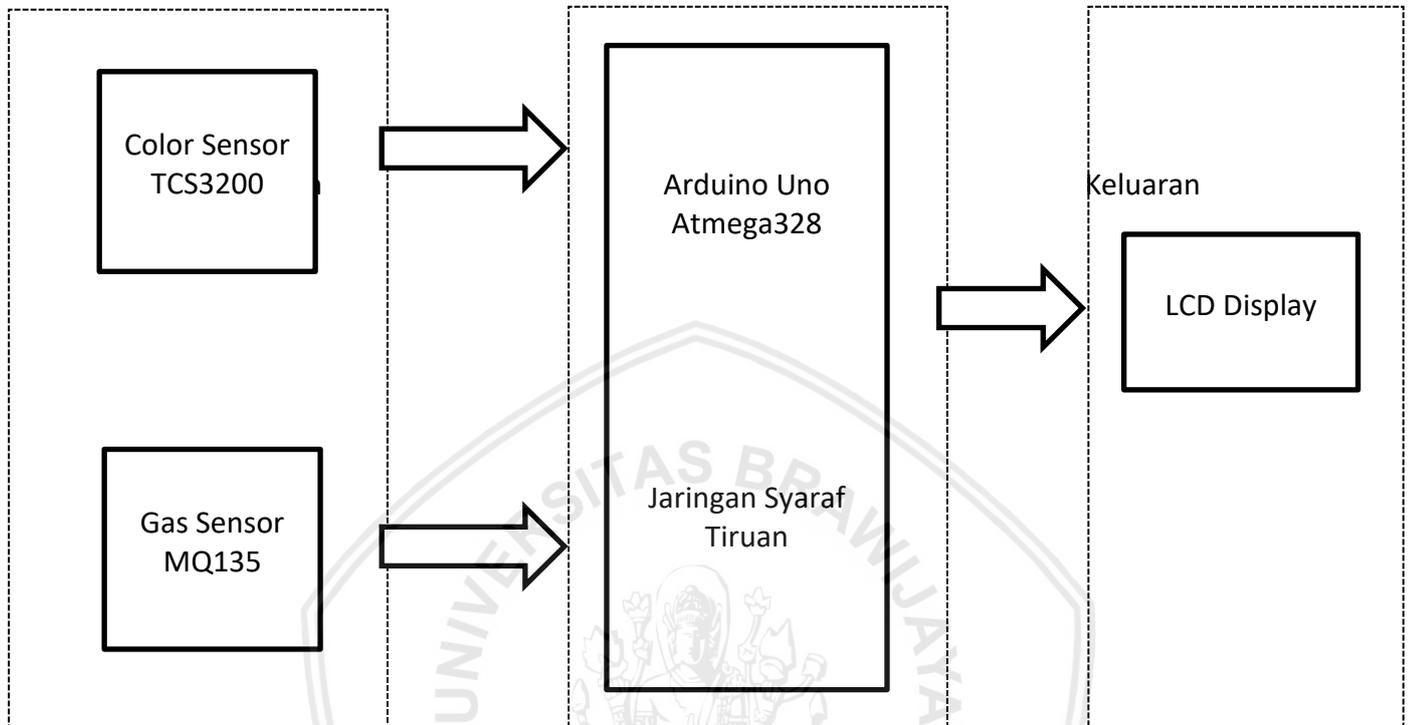
Berdasarkan metode penelitian diatas, perancangan sistem dilakukan setelah melakukan analisis kebutuhan sistem. Perancangan penelitian ini dilakukan untuk merencanakan tahapan-tahapan yang harus dilakukan dalam proses melakukan penelitian. Tahapan penelitian yang harus dilakukan antara lain perancangan perangkat lunak dan perangkat keras. Gambaran secara umum perancangan sistem pada Gambar 5.1.



Gambar 5.1 Blok Diagram Perancangan Sistem

5.1.1 Perancangan Perangkat Keras

Perancangan pada perangkat keras secara keseluruhan dapat dilihat pada diagram blok pada Gambar 5.2.

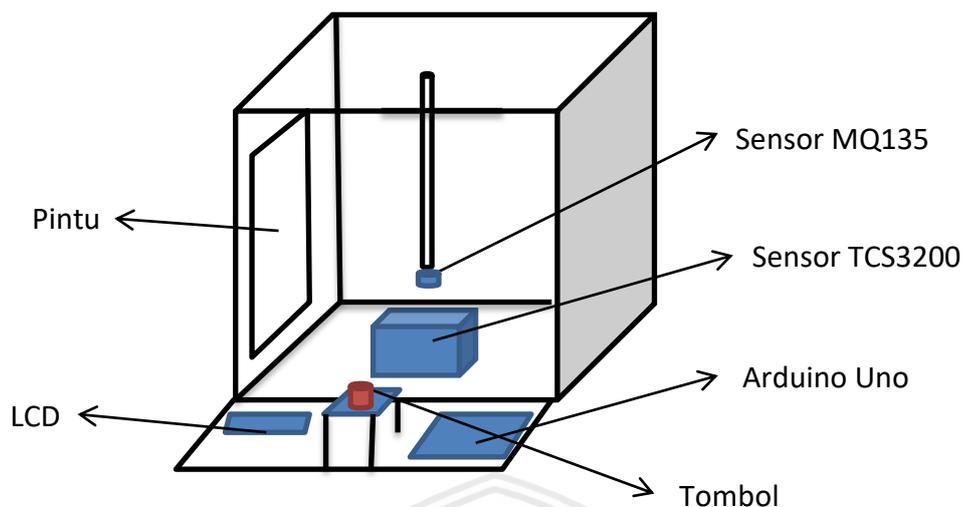


Gambar 5.2 Blok Diagram Keseluruhan Sistem

Pada Gambar 5.2 di jelaskan sistem menggunakan dua sensor, yaitu sensor MQ135 untuk mendeteksi gas amonia yang terdapat pada daging sapi, yang kedua sensor TCS3200 yaitu sensor warna yang memiliki fungsi untuk mendeteksi warna daging sapi. Kemudian data sensor di baca oleh arduino dan diolah oleh arduino menggunakan algoritma jaringan syaraf tiruan metode backpropagation, setelah itu hasil pengolahan dan hasil baca sensor akan di tampilkan pada layar LCD.

5.1.1.1 Perancangan *Prototype* Alat Pendeteksi Kesegaran Daging

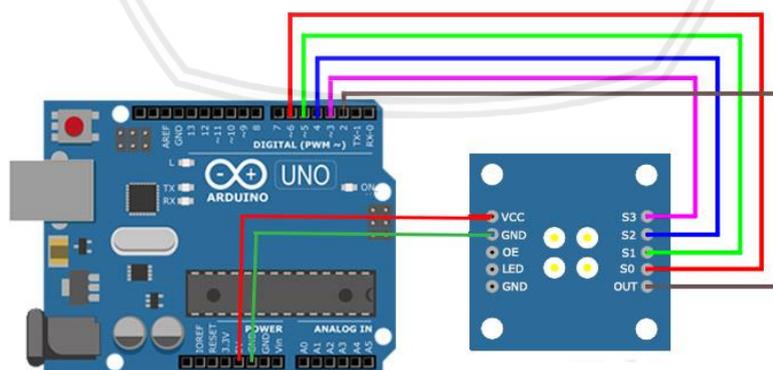
Dalam melakukan desain *prototype* dari Sistem Pendeteksi Daging ini perlu diperhatikan peletakkan tiap-tiap komponen serta ukuran alat yang akan dikembangkan. Pembuatan desain sistem yakni berupa kubus berwarna putih berukuran panjang 20cm, lebar 20 cm dan tinggi 20cm dan terdapat tambahan untuk penempatan komponen lainnya berukuran lebar 7cm dan panjang 20cm. Bentuk *prototype* alat ditunjukkan pada Gambar 5.3 berikut.



Gambar 5.3 Desain *Prototype* Alat Pendeteksi Kesehatan Daging Sapi

Pada Gambar 5.3 diatas bisa dilihat posisi sensor warna TCS3200 diletakkan di dalam kotak untuk menghindari interferensi cahaya lain selain warna dari objek yang akan dibaca dimana dalam hal ini adalah daging. Daging di letakkan tepat diatas sensor warna TCS3200 dan tepat dibawah sensor gas MQ135. Peletakkan sensor gas MQ135 dibuat demikian agar dapat membaca nilai gas amonia dari daging secara langsung. LCD 16x2 ditempatkan tidak jauh dari peletakkan sensor untuk memudahkan pengguna melihat langsung hasil olahan sistem. Selain itu di luar dari kotak juga diletakkan *push button* yang digunakan untuk men-*trigger* data sensor yang akan diolah dan arduino juga diletakkan tidak jauh agar *wiring* tidak terlalu panjang.

5.1.1.2 Perancangan Subsystem Sensor TCS3200



Gambar 5.4 Skematik Sensor TCS3200

Untuk mendapatkan nilai RGB, sensor warna harus dihubungkan menuju ke mikrokontroller Arduino. Hal ini membutuhkan *wiring* antara sensor warna menuju ke Arduino. Hubungan *wiring* ditunjukkan pada Gambar 5.4 Hubungan

pin antara sensor warna dengan Arduino dilakukan dengan mengaturnya pada program yang di tunjukkan pada Tabel 5.1.

Tabel 5.1 Keterangan koneksi pin Sensor TCS3200 dengan Arduino Uno

Pin Sensor Warna TCS3200	Pin Arduino Uno
V _{CC}	5V
GND	GND
S0	D6
S1	D5
S2	D4
S3	D3
OUT	D2

Pada Tabel 5.1 diatas merupakan *wiring* sensor warna dengan arduino, agar data yang didapat dari sensor bisa diteruskan ke arduino. Sensor TCS3200 memiliki masing-masing pin Vcc dan GND yang dihubungkan langsung ke pin Vcc dan GND pada arduino. Selanjutnya untuk keempat pin keluaran dari sensor ini dihubungkan dengan pin digital pada arduino yaitu pin S0 terhubung dengan pin D6, pin S1 terhubung dengan pin D5, pin S2 dihubungkan dengan pin D4 dan pin S3 dihubungkan dengan pin D3 pada arduino. Selain itu pin OUT sebagai masukan dari sensor dihubungkan dengan pin D2.

5.1.1.3 Perancangan Subsystem Sensor MQ135

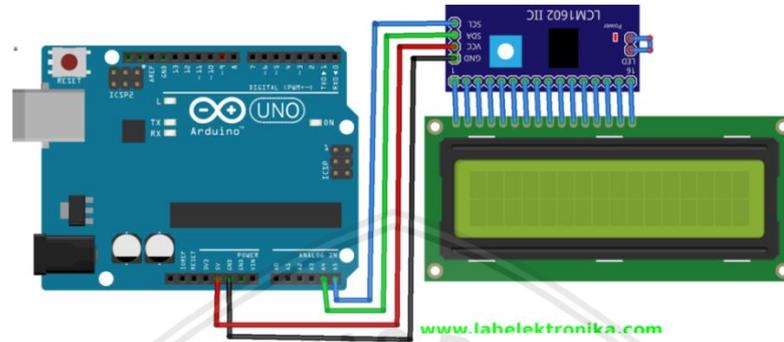


Gambar 5.5 Skematik Sensor MQ135

Pada Sensor MQ135 ini digunakan untuk mendeteksi gas amonia pada daging sapi, pada Gambar 5.5 tersebut terlihat ada tiga *wiring*, yaitu pin A0 arduino yang dihubungkan ke pin A0 pada sensor gas MQ135, yang kedua yaitu pin GND (*ground*) arduino yang dihubungkan pin GND sensor MQ135 dan yang terakhir yaitu pin 5V arduino yang di hubungkan ke pin VCC sensor MQ135. Pada

sensor MQ135 ini data sensor yang didapat adalah berupa data sensor analog, dapat kita lihat pada Gambar 5.5 diatas, bahwa pin yang tersambung ialah pin A0, di mana pin tersebut merupakan pin analog yang sudah tersambung dengan *analog to digital converter* pada arduino. Pada Arduino converter ini memiliki resolusi 10 bit, artinya nilai hasil konversi berkisar dari 0 hingga 1023.

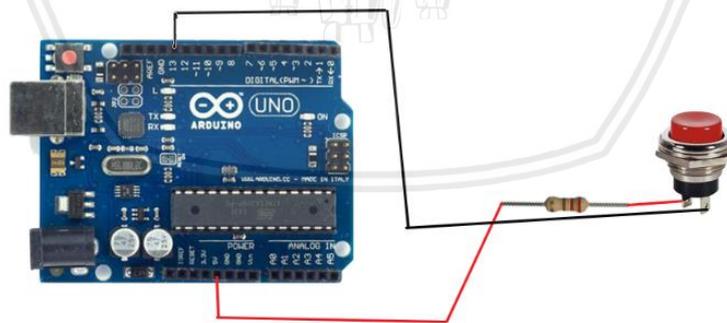
5.1.1.4 Perancangan Subsystem Output



Gambar 5.6 Skematik Sistem

Pada Gambar 5.4 di atas LCD langsung disambungkan menggunakan modul I2C yang memiliki fungsi untuk menghemat pin yang akan disambungkan ke arduino, pada modul I2C tersebut ada empat *wiring*, yaitu SLC pada modul di sambungkan pada pin A5 arduino, kemudian pin SDA modul di hubungkan dengan pin A4, untuk VCC pada modul langsung dihubungkan pada pin 5V arduino dan yan terakhir yaitu pin GND (*ground*) yang di hubungkan pada pin GND arduino.

5.1.1.5 Perancangan Push-Button



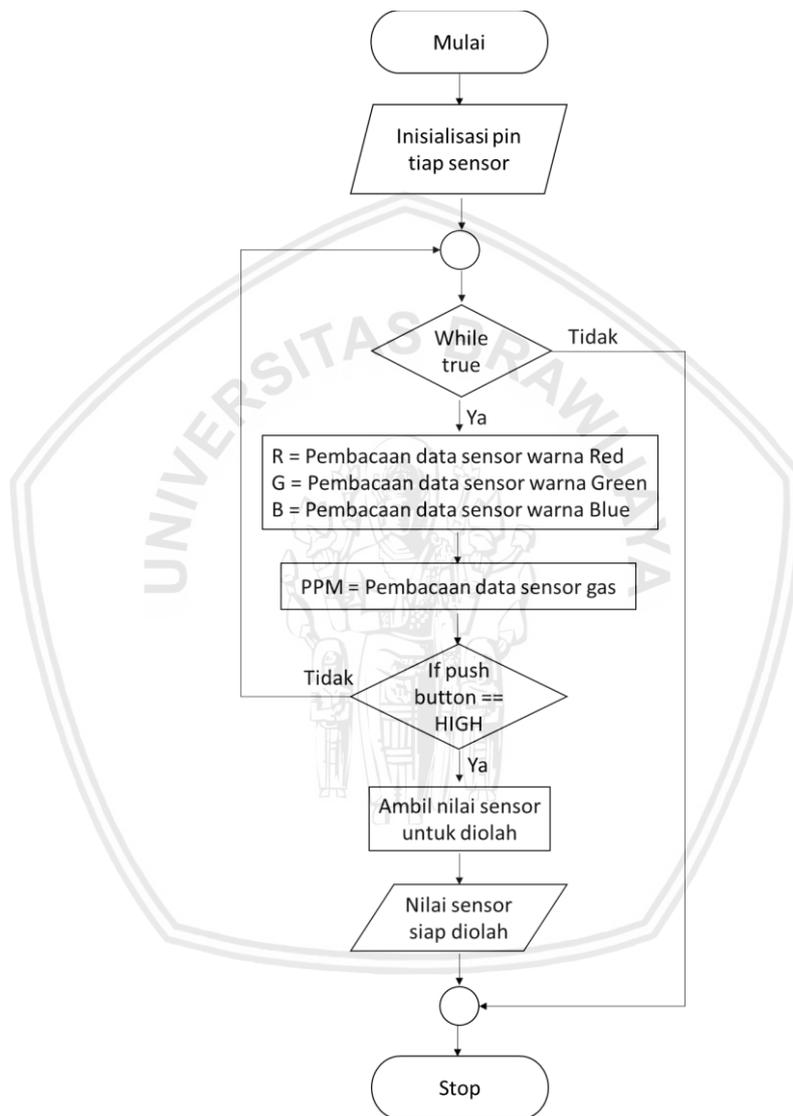
Gambar 5.7 Skematik Sistem Push Button

Pada Gambar 5.7 di atas merupakan skematik dari sistem *push button* yang digunakan, dapat dilihat pada gambar tersebut ada dua pin yang tersambung, yaitu pin pada 5v di arduino disambung dengan resistor 220 ohm. Sementara pada pin yang satu sisanya tersambung dengan pin 13 pada arduino.

5.1.2 Perancangan Perangkat Lunak

Pada subbab perancangan perangkat lunak dibagi menjadi 2 tahapan, yakni perancangan perangkat lunak pada mikrokontroler untuk mengambil data sensor yang akan diolah serta perancangan perangkat lunak untuk melakukan proses klasifikasi dengan metode *Jaringan Syaraf Tiruan*.

5.1.2.1 Perancangan Pengambilan Data Sensor

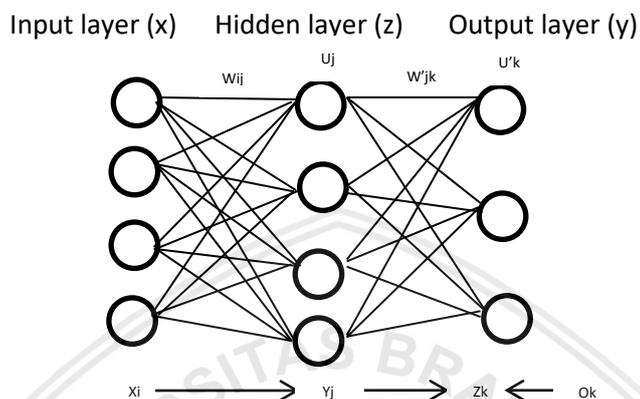


Gambar 5.8 Diagram Alir Perancangan Perangkat Lunak Pengambilan Data Sensor

Proses perancangan perangkat lunak untuk pengambilan data sensor ditunjukkan pada Gambar 5.6 diatas untuk menentukan hasil pembacaan data sensor mana yang akan diolah untuk dilakukan klasifikasi dengan menggunakan jaringan syaraf tiruan, karena pada dasarnya mikrokontroler akan melakukan pembacaan nilai sensor secara terus menerus. Sistem dimulai dengan melakukan inisialisasi pin dari masing-masing sensor pada arduino IDE yang

bertujuan untuk membedakan input dan output yang akan dibaca oleh masing-masing sensor. Selanjutnya saat nilai masing-masing sensor sudah dapat dibaca oleh mikrokontroler maka dilakukan proses pengecekan penekanan *push button*. Apabila terjadi penekanan *push button*, nilai sensor terakhir yang terbaca akan digunakan untuk perhitungan dalam menentukan jenis kesegaran daging sapi.

5.1.2.2 Perancangan Model Jaringan Syaraf Tiruan



Gambar 5.8 Jaringan saraf propagasi balik dengan satu lapis tersembunyi.

Pada Gambar 5.8 dapat dijelaskan dalam beberapa tahapan yang pertama yaitu menentukan jumlah *input*, *hidden layer*, dan *output*. Selanjutnya memberi nilai awal secara acak bagi seluruh bobot antara *input-hidden layer* dan *hidden layer-output*. Melakukan langkah-langkah secara berulang hingga diperoleh nilai error minimal yang memungkinkan bagi JST untuk belajar dengan baik, proses *feed forward propagation* Tiap unit input (X_i) menerima sinyal input dan sinyal tersebut dikirimkan pada seluruh unit *hidden layer*. Tiap unit hidden layer (Z_{in_j}) ditambah dengan input (X_i) yang dikali dengan weight (V_{ij}) dan dijumlah dengan bias bagian input.

$$\begin{aligned} &\{\text{Unit Input*Weight(Input->Hidden)}\} \\ &Z_{in_j} := Z_{in_j} + X_i[i] * V_{ij}[i,j]; \\ &\{\text{Ditambah Bias}\} Z_{in_j} := Z_{in_j} + V_{ij}[0][j]; \\ &\{\text{Dihitung dalam Fungsi Pengaktif}\} \\ &Z_j[j] := f(Z_{in_j}[j]); \end{aligned}$$

Fungsi pengaktif neuron yang digunakan pada seluruh bagian pelatihan harus sama. Tiap unit *output* (Y_{ink}) ditambah dengan nilai keluaran *hidden layer* (Z_j) yang dikali *weight* (W_{jk}) dan dijumlah dengan bias bagian *hidden layer*. Untuk mendapatkan keluaran JST, maka Y_{ink} dihitung dalam fungsi pengaktif menjadi Y_k .

$$\begin{aligned} &\{\text{Unit Keluaran*Weight(Hidden->Keluaran)}\} \\ &Y_{ink}[k] := Y_{ink}[k] + Z_j[j] * W_{jk}[j,k]; \\ &\{\text{Ditambah Bias}\} \\ &Y_{ink}[k] := Y_{ink}[k] + W_{jk}[0,k]; \\ &\{\text{Dihitung dalam Fungsi Pengaktif}\} \end{aligned}$$

$$Y_k[k]:=f(Y_{in}[k]);$$

Selanjutnya yaitu proses feed backward propagation, Tiap *output* dibandingkan dengan target yang diinginkan, untuk memperoleh *error global* digunakan metode *Mean Squared Error* (MSE).

$$\text{Error}:=\text{Error}+((O_{\text{target}}[k]-Y_k[k])*(O_{\text{target}}[k]-Y_k[k]))*0.5);$$

Tiap unit *output* menerima pola target sesuai dengan pola masukan saat pelatihan dan dihitung nilai *error*-nya dan diperbaiki nilai *weight*-nya. Perhitungan Error dalam turunan Fungsi Pengaktif.

$$\text{delta}_k[k]:=(O_{\text{target}}[k]-Y_k[k])*f'(Y_{in}[k]);$$

Perbaikan *weight output-hidden layer* dilakukan dengan memperhitungkan laju pelatihan dan momentum, laju pelatihan dijaga pada nilai kecil antara 0-1 dan momentum pada nilai 0.5-0.9. Perbaikan *weight* antara *hidden layer-output*.

$$\text{update_Wjk}[j,k]:=e\text{Laju}*\text{delta}_k[k]*Z_j[j]+(\text{update_Wjk}[j,k]*e\text{Momentum});$$

{Perbaikan weight bias antara hidden layer-output}

$$\text{update_Wjk}[0,k]:=e\text{Laju}*\text{delta}_k[k];$$

Tiap *weight* yang menghubungkan unit *output* dengan unit *hidden layer* dikali selisih error

(delta_k) dan dijumlahkan sebagai masukan unit berikutnya.

{Perhitungan Error*Bobot Keluaran}

$$\text{delta}_{in_j}[j]:=\text{delta}_{in_j}[j]+\text{delta}_k[k]*\text{Wjk}[j,k];$$

{Perhitungan Error dalam turunan Fungsi Pengaktif}

$$\text{delta}_j[j]:=\text{delta}_{in_j}[j]*f'(Z_{in_j}[j]);$$

Perbaikan *weight hidden layer-input* dilakukan dengan memperhitungkan laju pelatihan dan momentum, laju pelatihan dijaga pada nilai kecil antara 0-1 dan momentum pada nilai 0.5-0.9. Perbaikan *weight* antara masukan dan *hidden layer*.

$$\text{update_Vij}[i,j]:=e\text{Laju}*\text{delta}_j[j]*X_i[i]+(\text{Vij}[i,j]*e\text{Momentum});$$

{ Perbaikan weight bias antara masukan}

$$\text{update_Vij}[0,j]:=e\text{Laju}*\text{delta}_j[j];$$

Tiap *weight* dan bias yang ada pada JST diperbaiki.

{Penambahan Nilai Perbaikan Bobot Hidden layer-Keluaran}

$$\text{Wjk}[j,k]:=\text{Wjk}[j,k]+\text{update_Wjk}[j,k];$$

{Penambahan Nilai Perbaikan Bobot Masukan-Hidden layer}

$$\text{Vij}[i,j]:=\text{Vij}[i,j]+\text{update_Vij}[i,j];$$

uji kondisi pemberhentian pelatihan

Pada kondisi dimana JST telah selesai dilatih, maka JST tersebut dapat diujicoba sebelum pada akhirnya JST tersebut digunakan untuk menyelesaikan suatu masalah. Maka untuk menggunakan hasil pelatihan tersebut digunakan *weight* yang telah diperoleh dari proses pelatihan untuk memperoleh hasil target yang telah dilatihkan. Pada algoritma ujicoba JST yang dapat digunakan adalah sebagai berikut, Tiap unit *input* (X_i) menerima sinyal *input* dan sinyal tersebut dikirimkan pada seluruh unit *hidden layer*. Tiap unit *hidden layer* (Z_{in_j}) ditambah dengan *input* (X_i) yang dikali dengan *weight* (V_{ij}) yang diperoleh dari proses pelatihan dan dijumlah dengan bias bagian input.

```
{Unit Input*Weight(Input->Hidden)}
  Zin_j[j]:=Zin_j[j]+Xi[i]*Vij[i,j];
  {Ditambah Bias}
  Zin_j[j]:=Zin_j[j]+Vij[0][j];
  {Dihitung dalam Fungsi Pengaktif}
  Zj[j]:=f(Zin_j[j]);
```

Fungsi pengaktif neuron yang digunakan pada seluruh bagian ujicoba harus sama. Tiap unit *output* (Y_{ink}) ditambah dengan nilai keluaran *hidden layer* (Z_j) yang dikali *weight* (W_{jk}) yang diperoleh dari proses pelatihan dan dijumlah dengan bias bagian *hidden layer*. Untuk mendapatkan keluaran JST, maka Y_{ink} dihitung dalam fungsi pengaktif menjadi Y_k .

```
{Unit Keluaran*Weight(Hidden->Output)}
  Yink[k]:=Yink[k]+Zj[j]*Wjk[j,k];
  {Ditambah Bias}
  Yink[k]:=Yink[k]+Wjk[0,k];
  {Dihitung dalam Fungsi Pengaktif}
  Yk[k]:=f(Yink[k]);
```

Pada metode jaringan syaraf tiruan ini fungsi aktivasinya yaitu menggunakan sigmoid biner, fungsi aktivasi ini digunakan pada jaringan saraf untuk mengaktifkan atau tidak mengaktifkan neuron. Selain itu pada metode ini menggunakan *learning rate* sebesar 0.2 dan menggunakan epoch sebesar 100000 sebagai kondisi untuk menghentikan pelatihan pada metode ini.

Pada perancangan klasifikasi ini, tahapan pertama kali untuk mengklasifikasi jenis daging yaitu membuat data latih terlebih dahulu, pada tahapan pembuatan data latih, jumlah data latih yang digunakan sebanyak 81 dari jumlah tersebut dibagi menjadi 3 kelas, yaitu kelas segar, kelas sedang dan kelas busuk. Pada pengambilan data latih ini, daging yang sama akan di potong menjadi beberapa potongan kecil sesuai dengan ketentuan yang sudah di tetapkan untuk pengambilan data latih. Untuk pengambilan nilai data latih tersebut terbagi menjadi tiga rentang waktu yaitu rentang waktu 0-6 jam, pada waktu itu daging akan di ambil secara acak dan pada waktu yang acak untuk mengukur nilai yang terbaca oleh sensor. Rentang waktu tersebut akan

mengambil data latih sebanyak 26 kali percobaan untuk rentang 0-6 jam. Sementara itu pada rentang 6-12 dan 12-18 jam juga sama seperti percobaan pada rentang waktu 0-6 jam. Pada bab sebelumnya yaitu pada dasar teori sudah dijelaskan tingkat-tingkat kesegaran daging sapi, yaitu kelas segar yang merupakan daging dengan rentang 0-6 jam, kelas sedang yaitu pada rentang 6-12 jam dan kelas busuk busuk yaitu rentang 12-18 jam. Dari kelas-kelas tersebut Kita bisa lihat data latih yang terdapat pada Tabel 5.2

Tabel 5.2 Kelas Data Latih

NO	R	G	B	PPM	KELAS
1	127	72	71	25.64	SEGAR
2	131	68	62	24.98	SEGAR
3	127	72	68	22.19	SEGAR
4	118	59	56	20.28	SEGAR
5	117	58	56	21.33	SEGAR
6	114	59	55	19.42	SEGAR
7	118	71	67	20.54	SEGAR
8	122	71	67	21.75	SEGAR
9	112	60	56	20.43	SEGAR
10	121	68	62	22.33	SEGAR
11	115	69	55	19.56	SEGAR
12	110	66	58	21.65	SEGAR
13	111	68	62	23.67	SEGAR
14	125	70	66	21.77	SEGAR
15	115	55	57	20.54	SEGAR
16	110	60	56	18.32	SEGAR
17	113	65	66	19.67	SEGAR
18	112	64	56	21.54	SEGAR
19	123	68	63	18.55	SEGAR
20	109	66	54	16.67	SEGAR
21	117	58	57	22.65	SEGAR
22	118	59	56	20.43	SEGAR
23	114	59	55	19.21	SEGAR
24	125	71	67	19.11	SEGAR
25	124	69	60	21.66	SEGAR
26	118	59	55	21.54	SEGAR
27	114	59	55	18.65	SEDANG
28	109	63	65	13.22	SEDANG
29	109	55	62	10.04	SEDANG
30	101	55	52	9.53	SEDANG
31	105	59	55	7.29	SEDANG
32	104	51	53	7.39	SEDANG
33	108	62	64	13.79	SEDANG

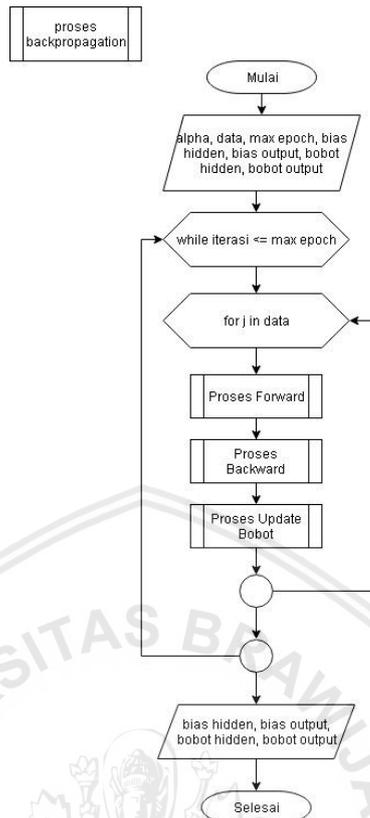
34	107	65	65	13.65	SEDANG
35	106	62	64	13.7	SEDANG
36	101	55	61	10.37	SEDANG
37	96	42	26	7.68	SEDANG
38	99	46	46	7.8	SEDANG
39	101	55	52	10.8	SEDANG
40	109	63	65	13.55	SEDANG
41	98	47	46	9.76	SEDANG
42	109	62	64	13.11	SEDANG
43	109	53	61	10.76	SEDANG
44	106	51	51	9.45	SEDANG
45	105	55	52	10.76	SEDANG
46	109	55	62	10.33	SEDANG
47	105	57	54	9.6	SEDANG
48	100	60	60	10.12	SEDANG
49	109	63	65	12.29	SEDANG
50	101	57	53	13.54	SEDANG
51	105	59	55	9.67	SEDANG
52	96	54	62	10.89	SEDANG
53	105	59	55	9.76	SEDANG
54	80	62	64	13.43	BUSUK
55	87	46	49	11.37	BUSUK
56	74	42	43	9.78	BUSUK
57	74	42	43	8.68	BUSUK
58	61	34	37	7.8	BUSUK
59	65	34	37	7.8	BUSUK
60	87	42	46	8.02	BUSUK
61	79	42	43	8.68	BUSUK
62	85	42	49	7.01	BUSUK
63	79	46	43	7.01	BUSUK
64	83	46	49	6.91	BUSUK
65	61	34	37	6.73	BUSUK
66	61	34	46	6.46	BUSUK
67	71	34	46	7.2	BUSUK
68	73	41	42	9.77	BUSUK
69	71	41	45	9.78	BUSUK
70	77	41	43	8.68	BUSUK
71	65	34	38	6.91	BUSUK
72	70	42	44	9.78	BUSUK
73	65	34	38	7.8	BUSUK
74	80	42	46	7.02	BUSUK
75	61	32	37	7.6	BUSUK
76	72	42	41	8.58	BUSUK

77	63	34	35	6.73	BUSUK
78	62	35	37	6.5	BUSUK
79	70	41	42	8.77	BUSUK
80	60	34	46	5.46	BUSUK
81	68	38	46	6.46	BUSUK

Setelah didapat data latih tersebut, maka data latih tersebut di olah dengan menggunakan bahasa pemrograman python dan proses tersebut terdapat pada komputer, karena memory yang dimiliki oleh arduino sangat kecil, sehingga apabila di jalankan pada arduino akan memberatkan mikrokontroler. Pada pemrosesan di komputer, spesifikasi yang digunakan antara lain :

- Model perangkat : Lenovo Z410
- Prosesor : Intel Core i5-4200 CPU 2,50GHz
- Sistem Operasi : Window 10 Pro 64-bit
- Memory : 8192MB

Pada dasarnya aturan pelatihan jaringan backpropagation terdiri dari 2 tahapan yaitu, *feedforward* dan *backward propagation*. Pada jaringan diberikan sekumpulan contoh pelatihan yang disebut set pelatihan. Set pelatihan ini digambarkan dengan sebuah vektor feature yang disebut dengan vektor input yang diasosiasikan dengan sebuah output yang menjadi target pelatihannya. Dengan kata lain set pelatihan terdiri dari vektor input dan juga vektor output target. Keluaran dari jaringan berupa sebuah vektor output aktual. Selanjutnya dilakukan perbandingan antara output aktual yang dihasilkan dengan output target dengan cara melakukan pengurangan diantara kedua output tersebut. Hasil dari pengurangan berupa error. Error dijadikan sebagai dasar dalam melakukan perubahan dari setiap bobot dengan *mempropagation*kannya kembali. Setiap perubahan bobot yang terjadi dapat mengurangi error. Siklus perubahan bobot (epoch) dilakukan pada setiap set pelatihan sehingga kondisi berhenti dicapai, yaitu bila mencapai jumlah epoch yang diinginkan atau hingga sebuah nilai ambang yang ditetapkan terlampaui.



Gambar 5.9 Diagram Alir Algoritma Pelatihan Jaringan Backpropagation

Algoritma pelatihan jaringan backpropagation terdiri dari 3 tahapan yaitu:

1. Tahap umpan maju (*feedforward*).
2. Tahap umpan mundur (*backpropagation*).
3. Tahap pengupdatean bobot dan bias.

Secara rinci algoritma pelatihan jaringan backpropagation dapat diuraikan sebagai berikut :

Langkah 0 : Inisialisasi bobot-bobot, konstanta laju pelatihan (α), toleransi error atau nilai bobot (bila menggunakan nilai bobot sebagai kondisi berhenti) atau set maksimal epoch (jika menggunakan banyaknya epoch sebagai kondisi berhenti).

Langkah 1 : Selama kondisi berhenti belum dicapai, maka lakukan langkah ke2 hingga langkah ke-9.

Langkah 2 : Untuk setiap pasangan pola pelatihan, lakukan langkah ke-3 sampai langkah ke-8.

a. Umpan maju (*feedforward*)

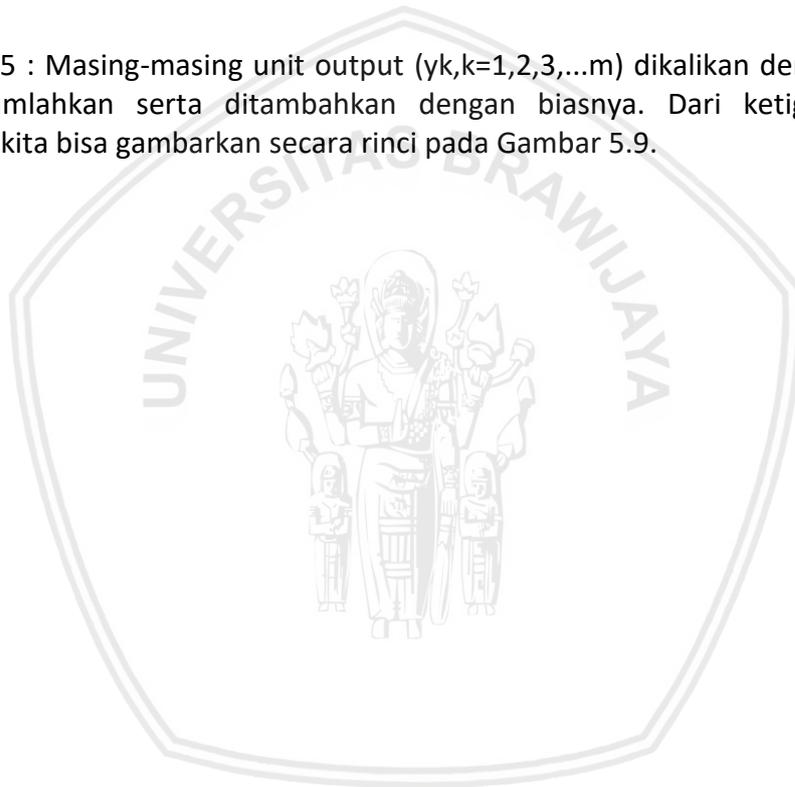
Umpan maju atau *feedforward* adalah tahap dimana sinyal yang terdapat pada layer input diteruskan sampai menuju ke layer output. Proses ini diawali

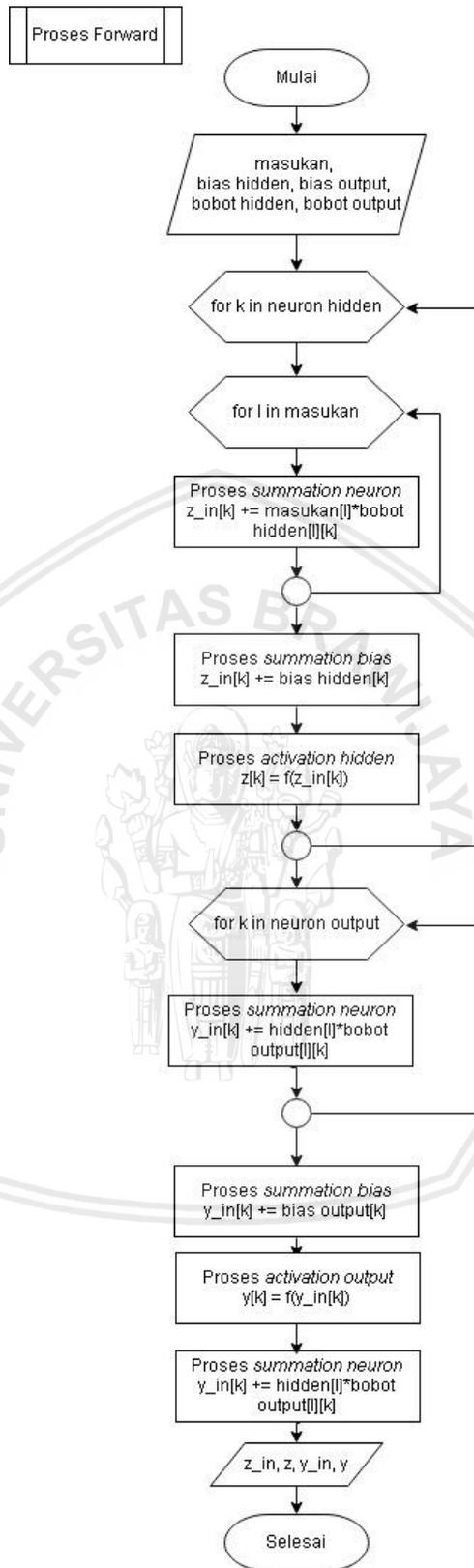
dengan menginisialisasi nilai weight dan bias pada setiap neuron. Umumnya pemilihan nilai weight dan bias ini adalah berupa nilai acak antara 0 sampai 1. Algoritma pada proses ini dapat dijelaskan dalam 3 tahapan utama yaitu,

Langkah 3 : Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya. Masing-masing unit input menerima sinyal masukan. Sinyal masukan dikirim ke seluruh unit *hidden*. Masukan yang di pakai adalah *input* training data yang sudah melalui penyekalaan. Nilai tertinggi dan terendah dari *input* yang dipakai dalam sistem kemudin dicari. Skala yang digunakan disesuaikan dengan fungsi aktivasinya.

Langkah 4 : Masing-masing unit di lapisan tersembunyi (dari unit ke-1 hingga unit ke-p) dikalikan dengan bobotnya dan dijumlahkan serta ditambahkan dengan biasnya.

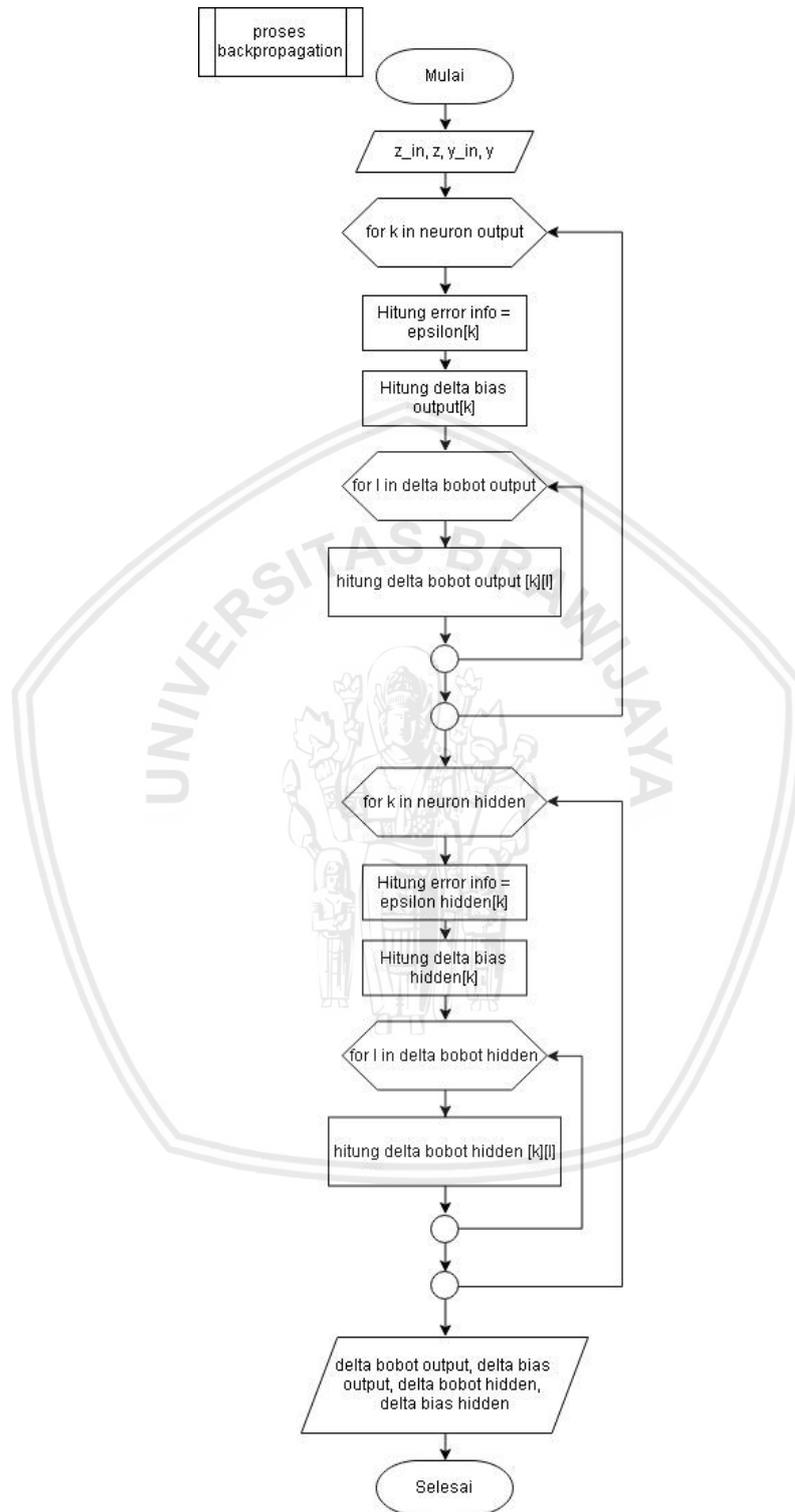
Langkah 5 : Masing-masing unit output ($y_k, k=1,2,3,\dots,m$) dikalikan dengan bobot dan dijumlahkan serta ditambahkan dengan biasnya. Dari ketiga langkah tersebut kita bisa gambarkan secara rinci pada Gambar 5.9.





Gambar 5.10 Diagram Alir Proses Umpan Maju (*Feedforward*)

b. Umpan mundur (*backward propagation*)



Gambar 5.11 Diagram Alir Proses Umpan Mundur (*Backward Propagation*)

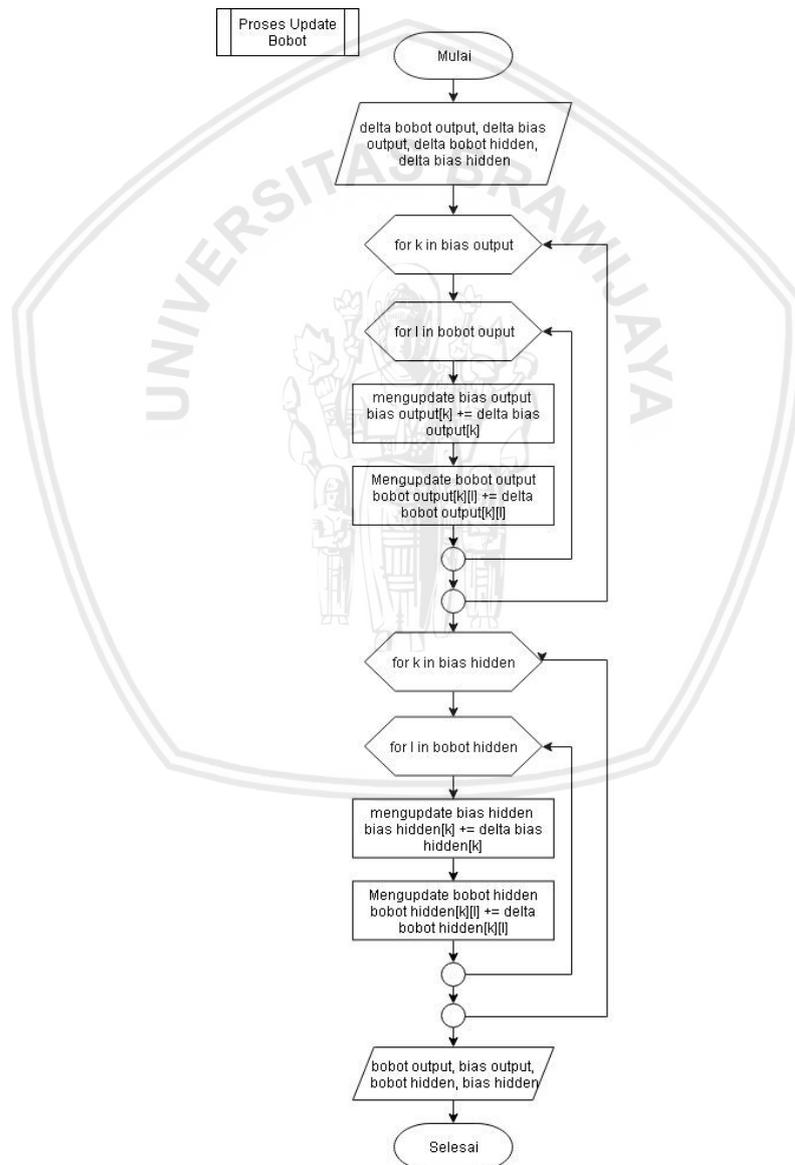
Pada Gambar 5.10 tersebut merupakan diagram alir proses umpan mundur. Dalam diagram alir tersebut dapat di bagi menjadi dua tahapan yang penting yaitu:



Langkah 6 : {Tahap II :)}. Masing-masing unit output ($y_k, k=1,2,3,\dots,m$) menerima pola target t_k sesuai dengan pola masukan/input saat pelatihan dan kemudian informasi kesalahan/error lapisan output (δ_k) dihitung. δ_k dikirim ke lapisan dibawahnya dan digunakan untuk menghitung besarnya koreksi bobot dan bias (ΔW_{jk} dan ΔW_{ok}) antara lapisan tersembunyi dengan lapisan output.

Langkah 7 : Pada setiap unit dilapisan tersembunyi (dari unit ke-1 hingga ke-p; $i=1\dots n; k=1\dots m$) dilakukan perhitungan informasi kesalahan lapisan tersembunyi (δ_j). δ_j kemudian digunakan untuk menghitung besar koreksi bobot dan bias (ΔV_{ji} dan ΔV_{jo}) antara lapisan input dan lapisan tersembunyi.

c. Update Bobot dan Bias



Gambar 5.12 Diagram Alir Proses Pengupdatetan Bobot Dan Bias

Pada Gambar 5.11 diatas merupakan proses pengupdatetan bobot dan bias yang dapat diuraikan dalam 2 langkah hal yang penting yaitu :



Langkah 8 : {Tahap III : }. Masing-masing unit output/keluaran (y_k , $k=1,2,3,\dots,m$) dilakukan pengupdatean bias dan bobotnya ($j=0,1,2,\dots,p$) sehingga menghasilkan bobot dan bias baru. Demikian juga untuk setiap unit tersembunyi mulai dari unit ke-1 sampai dengan unit ke-p dilakukan pengupdatean bobot dan bias.

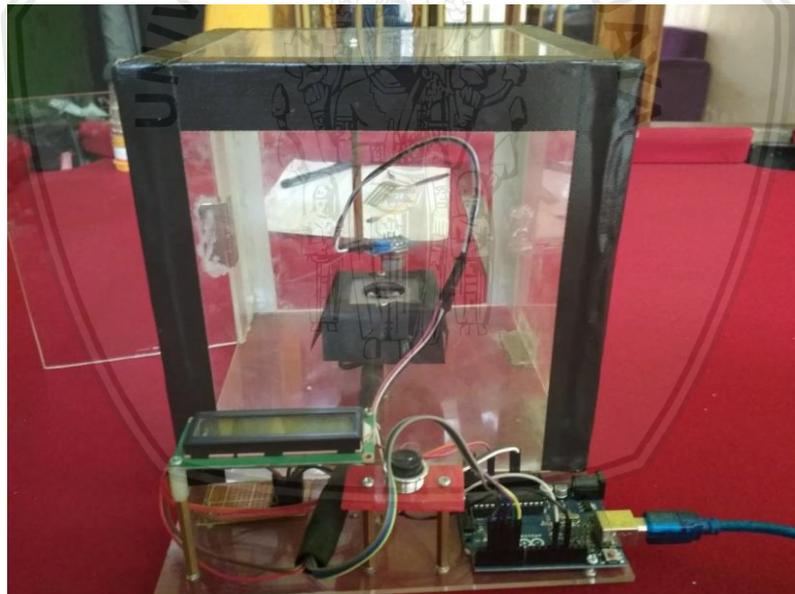
Langkah 9 : Uji kondisi berhenti (akhir iterasi).

5.2 Implementasi Sistem

Setelah melalui tahap perancangan sistem, maka selanjutnya adalah implementasi sistem sesuai hasil perancangan. Implementasi sistem yang dilakukan adalah implementasi perangkat keras dan implementasi perangkat lunak.

5.2.1 Implementasi Perangkat Keras

Pada implementasi perangkat keras dilakukan pemasangan komponen-komponen perangkat keras pada alat pendeteksi kebusukan daging sapi. Pada Gambar 5.11 tersebut merupakan alat yang digunakan untuk penelitian, yang terdiri dari dua sensor, yaitu sensor MQ135 dan sensor TCS3200 yang dapat dilihat pada Gambar 5.12.



Gambar 5.13 Penempatan Komponen Pada Alat Pendeteksi

5.2.2 Implementasi Subsistem Sensor TCS3200

Pada implementasi subsistem sensor TCS3200 dilakukan implementasi sesuai dengan perancangan subsistem sensor TCS3200. Pada sensor tersebut pemasangan di letakkan diatas sensor gas tidak begitu jauh, agar pembacaan bisa cepat. Hasil implementasi dapat dilihat pada Gambar 5.13



Gambar 5.14 Penempatan Sensor TCS3200

5.2.3 Implementasi Subsistem Sensor MQ135

Pada implementasi subsistem sensor MQ135 dilakukan implementasi sesuai dengan perancangan subsistem sensor MQ135. Pada Gambar 5.13 sensor ditutup dengan kertas yang dapat menghalau sinar dari luar agar noise yang terdapat dalam pembacaan bisa kecil Hasil implementasi dapat dilihat pada Gambar 5.14

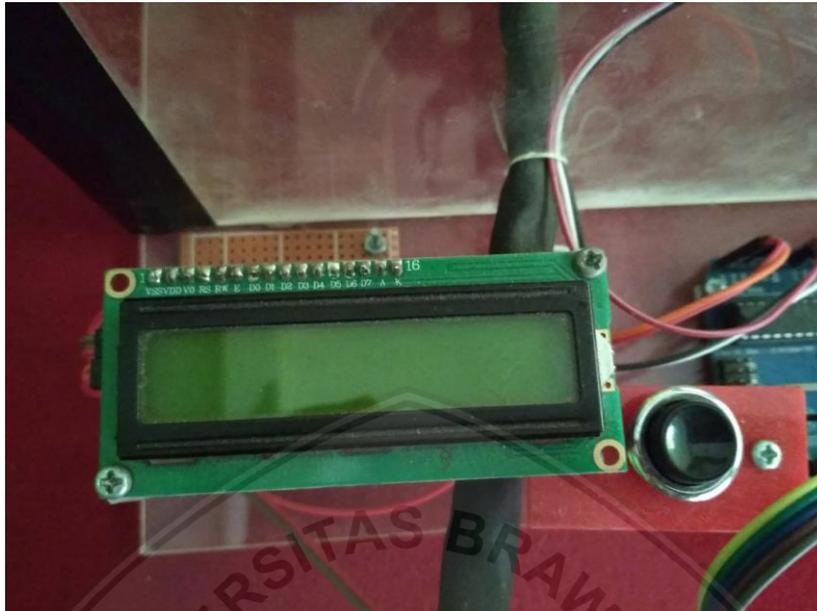


Gambar 5.15 Penempatan Sensor TCS3200

5.2.4 Implementasi Subsistem Output

Pada implementasi subsistem output. dilakukan implementasi sesuai dengan perancangan subsistem output yang menggunakan LCD *display* ini digunakan untuk menampilkan data yang sudah diproses oleh mikokontroler agar

pengguna bisa mengetahui hasil dari pemrosesan tersebut. Hasil implementasi dapat dilihat pada Gambar 5.15



Gambar 5.16 Penempatan Sistem Output

5.3 Implementasi Perangkat Lunak

Pada implementasi perangkat lunak dilakukan beberapa implementasi yaitu dua bagian implementasi, yang pertama implementasi untuk jaringan syaraf tiruan menggunakan python dan untuk implementasi sensor menggunakan arduino.

5.3.1 Implementasi Python

Pada implementasi python di jaringan syaraf tiruan yang pertama yaitu mempersiapkan data latih yang di tulis dengan format csv. Setelah itu data yang terdapat pada file tersebut di *load* untuk mengubah ke array, proses *loader* tersebut bisa kita lihat di dalam Tabel 5.3

Tabel 5.3 Program Loader

```
1 data = []
2     with open(filename) as f:
3         for line in f:
4             line = line.split(';')
5             atribut = []
6             for atr in line:
7                 # print(atr)
8                 atr = atr.split('i»¿')
9                 if (len(atr) > 1):
10                    atribut.append(convertData(isFloat,
11 atr[1].strip()))
12                    else:
13                        atribut.append(convertData(isFloat,
14 atr[0].strip()))
15                    data.append(atribut)
16                f.closed
```

17	return data
18	

Pada Tabel 5.3 diatas merupakan kode program yang digunakan untuk mengambil data latih yang sudah di buat dalam bentuk file dengan ekstensi *csv*. Dari Tabel 5.3 diatas, setelah proses *loader* selesai maka akan diteruskan ke proses *process*. Di dalam bagian *process* terbagi menjadi beberapa proses, yaitu mulai dari normalisasi data dan input, yang bisa di lihat pada Tabel 5.4

Tabel 5.4 Program Normalisasi Input dan Data

1	def normalizeInput(inputData, minVal, maxVal):
2	
3	for idx_atr in range(len(inputData)):
4	inputData[idx_atr] = (inputData[idx_atr] -
5	minVal[idx_atr]) / (
6	maxVal[idx_atr] - minVal[idx_atr])
7	return inputData
8	
9	
10	def normalizeData(data):
11	
12	minVal = [0.0,0.0,0.0,0.0]
13	maxVal = [255.0,255.0,255.0,100.0]
14	for idx_data in range(len(data)):
15	for idx_atr in range(len(data[idx_data])):
16	data[idx_data][
17	idx_atr] = (data[idx_data][idx_atr] -
18	minVal[idx_atr]) / (
19	maxVal[idx_atr] - minVal[idx_atr])
20	
21	return data, minVal, maxVal

Pada Tabel 5.4 merupakan kode program untuk proses normalisasi input dan data. Pada kode program diatas maka selanjutnya yaitu melakukan *generate layer* untuk jaringan saraf tiruan yang terdapat pada Tabel 5.5.

Tabel 5.5 Program Generate Layer

1	def generateLayer(number):
2	temp = []
3	for i in range(number):
4	temp.append(0)
5	return temp

Setelah proses *generate layer* selesai maka selanjutnya yaitu proses *generate bias*, Bias ialah salah satu node input yang sifatnya khusus. Penggunaan Bias dapat mempercepat pelatihan, hal ini dapat terjadi karena keberadaan bias berguna sebagai faktor koreksi terhadap kecukupan variabel-variabel input yang telah kita tetapkan. Misalkan telah menetapkan bahwa output Y dipengaruhi oleh variabel input X1 dan X2, apakah tidak ada variabel input yang lain selain X1 dan X2, mungkin saja masih ada, inilah fungsi dari bias. Dapat kita lihat pada Tabel 5.6 untuk kode program *generate bias*.

Tabel 5.6 Program Generate Bias

```

1 def generateBias(length, val):
2     bias = []
3     for i in range(length):
4         bias.append(val)
5     return bias

```

Pada Tabel 5.6 diatas merupakan kode program untuk *generate* bias, setelah itu di lanjutkan proses penjumlahan, pada proses penjumlahan kita bisa lihat kode program pada Tabel 5.7

Tabel 5.7 Program Proses Penjumlahan

```

1 def process_sum(bias, weight, inputData):
2
3     out_in = []
4     out = []
5     for i in range(len(bias)):
6         temp_in = 0
7         for j in range(len(weight[i])):
8             temp_in += inputData[j] * weight[i][j]
9         temp_in += bias[i]
10        out_in.append(temp_in)
11        temp = activationFunctionBinary(temp_in)
12        out.append(temp)
13    return out_in, out

```

Pada kode program diatas yaitu proses penjumlahan data. Proses berikutnya yaitu proses *forward* atau umpan maju. Pada proses tersebut kita bisa lihat tabel yang terdapat pada Tabel 5.8.

Tabel 5.8 Program Proses Forward

```

1 def processForward(dataTestPerLine, bias_Hidden, bias_Output,
2 weight_Hidden, weight_Output):
3     '''
4         return:
5         z_in, z, y_in, y
6         '''
7     z_in, z = process_sum(bias_Hidden, weight_Hidden,
8 dataTestPerLine)
9     y_in, y = process_sum(bias_Output, weight_Output, z)
10    return z_in, z, y_in, y

```

Setelah proses *forward* selesai maka akan di lanjutkan untuk mengupdate bobot bias. Pada proses tersebut kita bisa lihat kode program tersebut pada Tabel 5.9.

Tabel 5.9 Program Proses Update Bobot dan Bias

```

1 def updateWeight(weight, delta_weight):
2
3     new_weight = []
4     for i in range(len(weight)):
5         temp_weight = []
6         for j in range(len(weight[i])):
7             temp = weight[i][j] + delta_weight[i][j]
8         temp_weight.append(temp)

```

```

9         new_weight.append(temp_weight)
10        return new_weight
11
12    def updateBias(bias, delta_bias):
13
14        new_bias = []
15        for i in range(len(bias)):
16            temp = bias[i] + delta_bias[i]
17            new_bias.append(temp)
18        return new_bias

```

Setelah Proses tersebut maka didapat berupa data training yang akan di masukkan ke dalam kode program arduino, karena proses untuk training data menggunakan bahasa phyton. Hasil data training kita bisa lihat pada Gambar 5.15.

```

float maxVal [4] = {
| 255.0, 255.0, 255.0, 100.0
};

float bias_Hidden [4] = {
| 3.7495019369330795,
|-3.656327540126608,
|-9.984680452117006,
|-11.152360414417029
};

float weight_Hidden [4][4]= {{
| -9.53752943615178,
| 3.5824102760236065,
| -1.0525836867996385,
| 2.4722760114555387
}, {
| 7.196365805742117, -1.0641718207814663, -1.52601546240707, 6.447708677171792
}, {
| 30.91914126872395, -8.233612348898147, 1.203229686445866, 8.966403595091826
}, {
| 14.745179221824337, 8.684840792953562, -9.407265646062209, 27.76462245804956
}}};

float bias_Output [3] = {
| -8.335209788224898, -2.859117305837435, 5.227893654004517
};

```

Gambar 5.17 Contoh Data Training

5.3.2 Implementasi Arduino Uno

Pada Implementasi ini yang pertama yaitu proses yang harus dilakukan pada arduino, mulai dari inialisasi pin-pin yang terhubung ke sistem sensor maupun sistem output yang berupa kode program yang bisa kita lihat pada Tabel 5.10

Tabel 5.10 Program Proses Inialisasi Pin

```

1 //pin LCD Display
2 LiquidCrystal_I2C lcd(0x27, 20, 4);
3
4 //pin Color Sensor
5 #define S0 6
6 #define S1 5
7 #define S2 4
8 #define S3 3
9 #define OUT 2
10
11 //pin TCS Sensor

```



```

12 void TCS_Init(){
13     pinMode(S0, OUTPUT);
14     pinMode(S1, OUTPUT);
15     pinMode(S2, OUTPUT);
16     pinMode(S3, OUTPUT);
17     pinMode(OUT, INPUT);
18
19     digitalWrite(S0, LOW); // OUTPUT FREQUENCY SCALING 2%
20     x10
21     digitalWrite(S1, HIGH);
22 }

```

Setelah itu proses selanjutnya yaitu proses untuk pembacaan sensor, pada proses pembacaan sensor tersebut ada 2 sensor yang akan membaca kondisi daging sapi yaitu sensor MQ135 dan sensor TCS3200, pada proses pembacaan tersebut kita bisa lihat kode program pembacaan sensor pada Tabel 5.11.

Tabel 5.11 Program Proses Pembacaan Sensor

```

1 void read_sensor(float arraybaca[4]){
2     //view PPM Amonia
3     float analog = analogRead(gasSensor);
4     float voltage = (analog/1024)*5;
5     float Ra = ((Ro / voltage * 5) - Ro);
6     float ppm = MQ135_SCALINGFACTOR * pow((Ra / Ro),
7     MQ135_EXPONENT);
8     arraybaca[3] = ppm;
9
10    Serial.print("PPM : ");
11    Serial.println(arraybaca[3]);
12    Serial.print("Volt : ");
13    Serial.println(voltage);
14    lcd.setCursor(0, 0);
15    lcd.print("PPM : ");
16    lcd.print(arraybaca[3]);
17    lcd.setCursor(0, 1);
18    lcd.print("Volt : ");
19    lcd.print(voltage);
20
21    delay(1000);
22    lcd.clear();
23
24    g_flag = 0;
25    R = g_array[0] * g_SF[0];
26    G = g_array[1] * g_SF[1];
27    B = g_array[2] * g_SF[2];
28
29    if (R > 255) {
30        R = 255;
31    }
32    if (G > 255) {
33        G = 255;
34    }
35    if (B > 255) {
36        B = 255;
37    }
38    arraybaca[0] = R;
39    arraybaca[1] = G;
40    arraybaca[2] = B;

```

```

41 Serial.print("R : ");
42 Serial.print(round(arraybaca[0]));
43 Serial.print(" ");
44 Serial.print("G : ");
45 Serial.print(round(arraybaca[1]));
46 Serial.print(" ");
47 Serial.print("B : ");
48 Serial.println(round(arraybaca[2]));
49 Serial.println(" ");
50 lcd.setCursor(1, 0);
51 lcd.print("R = ");
52 lcd.setCursor(0, 1);
53 lcd.print(round(arraybaca[0]));
54 lcd.setCursor(7, 0);
55 lcd.print("G = ");
56 lcd.setCursor(6, 1);
57 lcd.print(round(arraybaca[1]));
58 lcd.setCursor(13, 0);
59 lcd.print("B = ");
60 lcd.setCursor(12, 1);
61 lcd.print(round(arraybaca[2]));
62 delay(500);
63 lcd.clear();
64 }

```

Dari tabel diatas hasil dari pembacaan sensor akan di teruskan ke proses testing apabila terdapat *triger* dari luar sistem, atau jika tombol untuk pengecekan di tekan akan melakukan testing dengan data yang didapat dari kedua sensor tersebut. Proses untuk melakukan tersting kita bisa lihat pada kode program yang terdapat pada tabel 5.12

Tabel 5.12 Program Proses Testing dan Output

```

1 float activationFunctionBinary(float d){
2     return 1.0 / (1.0 + exp(-d));
3 }
4
5 void activation (float y[]){
6     float maxVal = -99999999;
7     int maxIndex = -1;
8     for(int i =0 ; i<3;i++){
9         if(y[i]>maxVal){
10            maxVal=y[i];
11            maxIndex=i;
12        }
13    }
14    String outputString="";
15    if(maxIndex == 0){
16        outputString = "segar";
17    }
18    else if(maxIndex == 1){
19        outputString = "sedang";
20    }
21    else if(maxIndex == 2){
22        outputString = "busuk";
23    }
24    else{
25        outputString = "unknown";

```

```
26     }
27     lcd.print(outputString);
28     Serial.print("Hasil = ");
29     Serial.println(outputString);
30 }
31
32 void testInput(){
33     float normalized_masukan [4] = {};
34     for(int i=0; i<4;i++){
35         normalized_masukan[i] = (baca[i]-minVal[i])/(maxVal[i]-
36 minVal[i]);
37     };
38     float activatedZ [4]={0.0,0.0,0.0,0.0};
39     for(int i = 0; i<4; i++){
40         float temp_inZ = 0;
41         for(int j = 0; j<4; j++){
42             // Serial.println(normalized_masukan[j]);
43             temp_inZ += normalized_masukan[j] *
44 weight_Hidden[i][j];
45         }
46         temp_inZ += bias_Hidden[i];
47         activatedZ[i] = activationFunctionBinary(temp_inZ);
48         // Serial.println("activated z");
49         // Serial.println(activatedZ[i]);
50     }
51
52     float activatedY [3]={0.0,0.0,0.0};
53     for(int i = 0; i<3; i++){
54         float temp_inY = 0;
55         for(int j = 0; j<4; j++){
56             // Serial.println(activatedZ[j]);
57             temp_inY += activatedZ[j] * weight_Output[i][j];
58         }
59         temp_inY += bias_Output[i];
60         activatedY[i] = activationFunctionBinary(temp_inY);
61         // Serial.println("activated y");
62         // Serial.println(activatedY[i]);
63     }
64     activation(activatedY);
65 }
```

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas proses pengujian serta menganalisis hasil dari pengujian yang dilakukan berdasarkan sistem yang telah dibuat. Adapun tujuan dilakukannya pengujian ada untuk mengetahui apakah semua kebutuhan yang diharapkan telah terpenuhi oleh sistem. Proses pengujian yang dilakukan yakni berupa pengujian fungsional, pengujian akurasi sistem, dimana pengujian fungsional yakni menguji fungsi dari perangkat keras dalam hal ini berupa sensor-sensor yang digunakan apakah dapat bekerja sesuai spesifikasinya, pengujian akurasi yakni menguji seberapa akurat sistem yang telah dirancang dan diimplementasikan dibandingkan nilai atau hasil yang sebenarnya. Berikut dijelaskan beberapa skenario pengujian yang dilakukan untuk menguji sistem.

6.1 Pengujian Sensor Warna TCS3200

Sensor warna TC3200 adalah sensor utama dalam sistem ini yang berfungsi untuk membaca nilai RGB warna dari daging yang dideteksi. Pada pengujian sensor warna ini akan dilakukan dengan melakukan pembacaan dari beberapa daging yang berbeda dengan menggunakan sensor warna TC3200 kemudian nilainya akan dibandingkan dengan alat ukur warna berupa *Eyedropper tool* pada *Adobe Photoshop*, dimana warna yang diuji menggunakan fitur *Eyedropper tool* tersebut didapatkan dari pengambilan gambar daging yang diujikan.

6.1.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui tingkat keakuratan sensor TCS3200 dalam membaca warna suatu objek. Dengan melakukan perbandingan nilai antara yang dibaca oleh sensor dengan nilai warna yang terbaca di *color Eyedropper tool* pada *Adobe Photoshop*, maka akan diperoleh selisih nilai dari pembacaan sensor yang kemudian dapat diketahui nilai error dari sensor warna yang diterapkan.

6.1.2 Prosedur Pengujian

Berikut prosedur yang dilakukan untuk menguji sensor warna TCS3200 :

1. Menghubungkan mikrokontroler Arduino Uno dengan laptop.
2. Meng-*upload* kode program dari sensor TCS3200.
3. Mengukur nilai RGB pada jenis objek daging yang berbeda menggunakan sensor warna TCS3200.
4. Menyesuaikan warna daging dengan mengambil gambar daging yang diujikan, kemudian gambar tersebut diukur nilai RGBnya menggunakan fitur *Eyedropper tool* pada *Adobe Photoshop*.
5. Mengamati hasil pembacaan nilai RGB dari sensor warna dari LCD 16x2 dan membandingkan nilai yang terbaca tersebut dengan nilai RGB dari gambar daging yang diukur dengan fitur *Eyedropper tool Photoshop*, kemudian

menentukan besarnya *error* pembacaan sensor dengan cara mengkonversi nilai RGB warna pembacaan sensor dan RGB warna dari gambar daging di *Photoshop* menjadi nilai HEX terlebih dahulu.

6. Nilai HEX dari warna masing-masing pembacaan warna kemudian diambil selisihnya untuk dilakukan perhitungan persentase *error* dalam bentuk nilai desimal

Adapun cara untuk mengukur persentase *error* yaitu dengan menggunakan **Persamaan (6.1)** berikut :

$$\text{Persentase error} = \frac{\text{Selisih nilai pembacaan}}{\text{Pembacaan Tools Photoshop}} \times 100\% \quad (6.1)$$

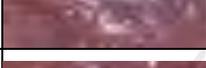
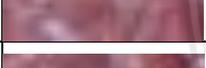
Untuk menghitung nilai selisih pembacaan nilai sensor dan alat ukur yaitu dengan menggunakan **Persamaan (6.2)** berikut:

$$\text{Selisih nilai pembacaan} = |\text{pembacaan alat ukur} - \text{pembacaan sensor}| \quad (6.2)$$

6.1.3 Hasil dan Analisis Pengujian

Hasil pengujian sensor warna TCS3200 ditunjukkan pada Gambar 6.1 di bawah, dimana pada pengujian sensor TCS3200 ini menguji sebanyak 27 data uji. Berdasarkan tabel tersebut terlihat bahwa sensor dapat membaca beberapa macam warna yang berbeda, tidak hanya warna primer (merah, hijau dan biru) melainkan dapat membaca warna sekunder maupun tersier. Hasil perbandingan antara nilai pembacaan sensor dan nilai pembacaan warna dari *Adobe Photoshop* pada objek daging yang sama menunjukkan perbedaan nilai.

Tabel 6.1 Hasil pengujian pembacaan sensor warna TCS3200

Pengujian ke-	Warna Daging Pengujian	Pembacaan Sensor					Photoshop					Selisih Error		Error
		R	G	B	HEX	Warna	R	G	B	HEX	Warna	HEX	DEC	Persentase
1		127	73	69	#7f4945		126	92	96	#7e5c60		#ECE5	60645	0.73
2		153	72	74	#99484a		141	77	77	#8d4d4d		#BFAFD	785149	7.82
3		180	85	84	#b45554		185	80	84	#b95055		#4FB01	326401	2.76
4		167	89	87	#a75957		167	101	107	#a7656b		#C14	3092	0.03
5		171	85	84	#ab5554		171	118	134	#ab7686		#2132	8498	0.08
6		197	97	96	#c56160		197	93	110	#c55d6e		#3F2	1010	0.01
7		193	102	99	#c16663		188	133	152	#bc8598		#4E0CB	319691	2.52
8		189	106	102	#bd6a66		181	77	172	#b595ac		#7D4BA	513210	4.13
9		197	97	102	#c55c66		195	152	166	#c498a7		#C3BF	50111	0.39
10		136	80	84	#885054		137	76	83	#894c53		#FBFF	64511	0.72
11		145	93	93	#915d5d		143	99	112	#8f6370		#1F9ED	129517	1.36

12		145	85	84	#915554		156	87	92	#9c575c		#B0208	721416	7.57
13		140	89	90	#8c595a		137	79	88	#894b58		#30E02	200194	2.18
14		140	85	84	#8c5554		142	76	86	#8e4c56		#1F702	128770	1.40
15		131	80	80	#835050		134	75	80	#864b51		#2FB01	195329	2.27
16		114	72	74	#72484a		113	83	83	#715353		#F4F7	62711	0.84
17		126	80	80	#7e5050		122	83	88	#7a5358		#3FCF8	261368	3.16
18		114	72	71	#724847		114	82	87	#725257		#A10	2576	0.03
19		41	23	31	#29171f		37	20	30	#25151e		4#0201	26265	1.08
20		56	33	46	#38212e		64	40	54	#402836		#80708	52608	1.25
21		68	55	70	#443746		72	57	78	#48394e		#40208	26264	0.55
22		56	30	39	#381e27		63	37	48	#3f2530		#70709	46055	1.11
23		70	52	66	#463442		76	47	61	#4c2f3d		#5FAFB	39193	0.78
24		62	50	61	#3e323d		68	53	67	#443543		#60306	39399	0.88
26		49	39	48	#312730		60	46	59	#3c2e3b		#B070B	72269	1.83
27		60	51	69	#3c3345		67	51	69	#433345		#70000	45875	1.04
Rata-rata													1,73	

Berdasarkan persamaan Tabel 6.1 untuk menentukan persentase *error* pada setiap pengujian, diperoleh rata-rata *error* sebesar 2,11%. Contoh perhitungan persentase *error* pada pengujian ke-1 sebagai berikut :

$$\text{Nilai HEX warna pembacaan sensor} = 7F4945_{(HEX)} = 8341829_{(DEC)}$$

$$\text{Nilai HEX warna pembacaan alat ukur} = 7E5C60_{(HEX)} = 8281184_{(DEC)}$$

$$\begin{aligned} \text{Selisih nilai pembacaan} &= |\text{pembacaan alat ukur} - \text{pembacaan sensor}| \\ &= |8341829 - 8281184| \\ &= 60645 \end{aligned}$$

$$\begin{aligned} \text{Persentase error} &= \frac{\text{Selisih nilai pembacaan}}{\text{Pembacaan alat ukur}} \times 100\% \\ &= \frac{60645}{8281184} \times 100\% \\ &= 0,73\% \end{aligned}$$

Adapun untuk menghitung nilai rata-rata *error* keseluruhan pengujian adalah sebagai berikut:

$$\text{Rata - rata error} = \frac{\text{Jumlah persentase error}}{\text{Jumlah pengujian}}$$

Rata-rata *error* yang dihasilkan sensor sangat kecil, sehingga dapat dikatakan bahwa akurasi pembacaan sensor warna sangat baik. Selain itu, walaupun terdapat *error* antara pembacaan sensor dengan alat ukur, namun sensor tetap dapat membaca warna meskipun perubahan nilainya cukup kecil, terlihat dari adanya perbedaan kepekatan warna yang ditunjukkan pada kolom warna *Eyedropper tool* pada *Photoshop*. Semakin gelap warna objek yang dideteksi maka nilai RGB semakin rendah (minimum 0,0,0), dan semakin terang warna yang dideteksi maka nilai RGB semakin tinggi (maksimum 255,255,255).

6.2 Pengujian Sensor Gas MQ135

Sensor gas MQ135 merupakan sensor yang mempunyai tugas utama dalam sistem ini untuk melakukan pembacaan nilai terhadap gas amonia yang ada pada daging yang dideteksi. Pada pengujian ini dilakukan dengan mengukur nilai gas amonia dari daging serta nilai tegangan keluaran dari sensor saat mendeteksi daging.

6.2.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui fungsionalitas pembacaan gas amonia dari sensor gas MQ135 yang diterapkan apakah sesuai dengan karakteristik sensor MQ135 pada *datasheet*.

6.2.2 Prosedur Pengujian

Berikut prosedur yang dilakukan untuk menguji sensor gas MQ135 :

1. Menghubungkan mikrokontroler Arduino Uno dengan laptop.
2. Meng-*upload* kode program dari sensor MQ135.
3. Mengukur nilai dari sensor MQ135 untuk pembacaan gas amonia pada daging dan nilai tegangan keluaran sensor dari seluruh data uji yang berjumlah 23 data.
4. Mengamati hasil pembacaan nilai dari sensor MQ135 dan nilai tegangan keluaran kemudian menganalisis hasilnya terkait dengan karakteristik sensor MQ135.

6.2.3 Hasil dan Analisis Pengujian

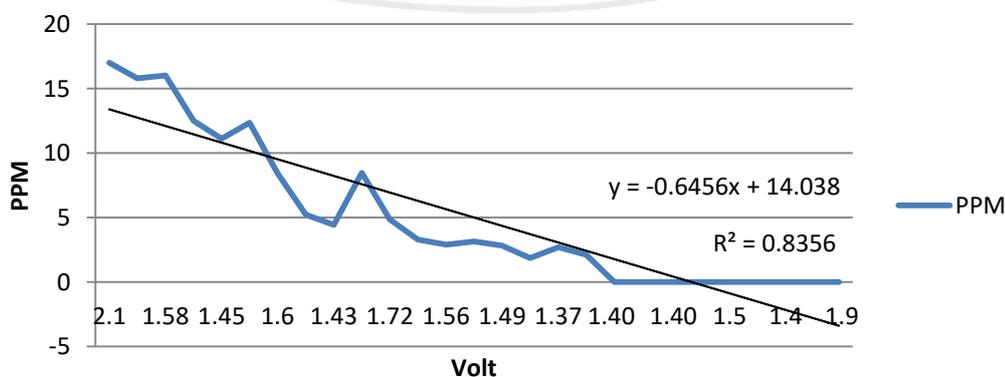
Tabel 6.2 Hasil pengujian pembacaan sensor MQ135

Nama Pengujian	Volt (V)	PPM
Pengujian ke-1	2.1	17
Pengujian ke-2	1.44	15.8
Pengujian ke-3	1.58	16
Pengujian ke-4	1.49	12.49
Pengujian ke-5	1.45	11.11
Pengujian ke-6	1.49	12.34
Pengujian ke-7	1.6	8.43
Pengujian ke-8	1.54	5.23
Pengujian ke-9	1.43	4.44
Pengujian ke-10	1.92	8.46
Pengujian ke-11	1.72	4.84
Pengujian ke-12	1.6	3.3
Pengujian ke-13	1.56	2.89
Pengujian ke-14	1.62	3.15
Pengujian ke-15	1.49	2.81

Pengujian ke-16	1.4	1.85
Pengujian ke-17	1.37	2.71
Pengujian ke-18	1.46	2.11
Pengujian ke-19	1.40	1.75
Pengujian ke-20	1.30	1.5
Pengujian ke-21	1.40	2.10
Pengujian ke-22	1.2	1.3
Pengujian ke-23	1.5	1.6
Pengujian ke-24	1.2	1.8
Pengujian ke-25	1.4	1.5
Pengujian ke-26	1.6	1.2
Pengujian ke-27	1.9	2.4

Pada Tabel 6.2 menunjukkan nilai pembacaan sensor MQ135 berupa nilai PPM dan *voltage* keluaran sensor pada mikrokontroler sebanyak 27 kali pengujian terlihat bahwa semakin tinggi nilai nilai PPM semakin tinggi pula nilai tegangan keluaran sensor. Berdasarkan *datasheet* sensor MQ135 dimana sensor MQ135 merupakan sensor resistif, dimana semakin tinggi nilai resistansi sensor maka nilai PPM akan semakin rendah, dengan begitu sesuai dengan rumus tegangan yaitu $V = I \times R$ maka tegangan yang dihasilkan sensor akan semakin tinggi. Sehingga dapat ditarik kesimpulan bahwa hasil pengujian sensor MQ135 sesuai dengan karakteristik sensor pada *datasheet*.

Korelasi Kadar Amonia dengan Volt Sensor



Gambar 6.1 Analisis korelasi kadar Amonia dengan Volt



Pada Gambar 6.1 di atas menunjukkan grafik hubungan antara nilai pembacaan PPM gas amonia dari daging sapi dan nilai tegangan keluaran sensor, terlihat pada gambar tersebut nilai $y = -0.6456x + 14.038$ merupakan model regresi *power* antara kedua variabel tersebut, sedangkan nilai $R^2 = 0.835$ disebut sebagai koefisien determinasi yang berarti variabel PPM amonia dipengaruhi oleh variabel V_{out} sebesar 83,56%. Adapun 16,44% sisanya untuk konsentrasi amonia dipengaruhi variabel lain.

6.3 Pengujian Akurasi Hasil Klasifikasi Jaringan Syaraf Tiruan

Sistem Pendeteksi Kesegaran Daging Berdasarkan Warna dan Kadar Amonia pada Daging Berbasis Sensor TCS3200 dan MQ135 dengan Metode Jaringan Syaraf Tiruan ini mempunyai tujuan utama untuk dapat mengklasifikasikan jenis daging, oleh karena itu perlu diketahui tingkat keakuratan sistem dalam melakukan klasifikasi.

6.3.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk menentukan nilai akurasi penggunaan metode Jaringan Syaraf Tiruan pada sistem pendeteksi kesegaran daging sapi yang telah dibuat.

6.3.2 Prosedur Pengujian

Prosedur pengujian akurasi Jaringan Syaraf Tiruan dilakukan adanya masukan 4 fitur berdasarkan pembacaan data sensor ini menggunakan rasio perbandingan antara data latih dan data uji yaitu 3:1. Penentuan proporsi data latih dan data uji dengan rasio 3:1 dikarenakan untuk mendapatkan akurasi semakin tinggi maka jumlah data latih harus lebih banyak minimal 2 kali lipat dari data uji (Adhieputra, 2010). Sebanyak data digunakan sebagai data uji. rumus yang digunakan untuk menghitung akurasi ditunjukkan pada **Persamaan (6.3)**.

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ data} \times 100\% \quad (6.3)$$

6.3.3 Hasil dan Analisis Pengujian

Tabel 6.2 Data uji dan hasil pengujian Jaringan Syaraf Tiruan

NO	R	G	B	PPM	KELAS SEBENARNYA	HASIL SISTEM	KESESUAIAN
1	127	73	69	17	SEGAR	SEGAR	SESUAI
2	153	72	74	15.8	SEGAR	SEGAR	SESUAI
3	180	85	84	16	SEGAR	SEGAR	SESUAI
4	167	89	87	12.49	SEGAR	SEDANG	TIDAK SESUAI
5	171	85	84	11.11	SEGAR	SEDANG	TIDAK SESUAI
6	197	97	96	12.34	SEGAR	SEGAR	SESUAI

7	193	102	99	8.43	SEGAR	SEGAR	SESUAI
8	189	106	102	5.23	SEGAR	SEGAR	SESUAI
9	197	97	102	4.44	SEGAR	SEGAR	SESUAI
10	136	80	84	8.46	SEDANG	SEDANG	SESUAI
11	145	93	93	4.84	SEDANG	SEDANG	SESUAI
12	145	85	84	3.3	SEDANG	SEDANG	SESUAI
13	140	89	90	2.89	SEDANG	SEDANG	SESUAI
14	140	85	84	3.15	SEDANG	SEDANG	SESUAI
15	131	80	80	2.81	SEDANG	SEDANG	SESUAI
16	114	72	74	1.85	SEDANG	SEDANG	SESUAI
17	126	80	80	2.71	SEDANG	SEDANG	SESUAI
18	114	72	71	2.11	SEDANG	SEDANG	SESUAI
19	41	23	31	1.75	BUSUK	BUSUK	SESUAI
20	56	33	46	1.5	BUSUK	BUSUK	SESUAI
21	68	55	70	2.10	BUSUK	BUSUK	SESUAI
22	56	30	39	1.3	BUSUK	BUSUK	SESUAI
23	70	52	66	1.6	BUSUK	BUSUK	SESUAI
24	62	50	61	1.8	BUSUK	BUSUK	SESUAI
25	79	55	68	1.5	BUSUK	BUSUK	SESUAI
26	49	39	48	1.2	BUSUK	BUSUK	SESUAI
27	60	51	69	2.4	BUSUK	BUSUK	SESUAI

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 6.3 terlihat bahwa dari jumlah 27 data terdapat 2 hasil dari sistem yang tidak sesuai dengan kelas sebenarnya. Sehingga akurasi yang diperoleh Sistem Pendeteksi Kesegaran Daging dengan Metode Jaringan Syaraf Tiruan ini berdasarkan **Persamaan (6.3)** adalah sebesar 95,65%. Proses perhitungan akurasinya sebagai berikut :

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Total data} - \text{Data tidak sesuai}}{\text{Total data}} \times 100\% \\
 &= \frac{27 - 2}{27} \times 100\% \\
 &= \frac{25}{27} \times 100\% \\
 &= 92,59\%
 \end{aligned}$$

BAB 7 PENUTUP

Bab ini memuat penarikan kesimpulan berdasarkan tahap-tahap yang telah dikerjakan sebelumnya. Selain itu pada bab ini pula peneliti menyampaikan saran yang diharapkan dapat digunakan untuk pengembangan penelitian yang serupa dengan penelitian ini selanjutnya.

7.1 Kesimpulan

Sesuai dengan rumusan masalah yang diajukan diawal penelitian serta berdasarkan hasil analisis dari pengujian yang dilakukan maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Sensor warna TCS3200 yang ditempatkan tepat dibawah daging sapi dapat membaca nilai RGB warna dari daging dengan rata-rata *error* yang kecil yakni sebesar 1,73 % sehingga dapat dikatakan sensor bekerja dengan baik karena dapat membaca perubahan warna yang berbeda dengan perubahan nilai yang kecil sekalipun. Selanjutnya sensor gas MQ135 yang diletakkan tepat diatas daging sapi dapat membaca kadar amonia dari daging dengan baik karena fungsionalitas sensor yang diterapkan sesuai dengan karakteristik pada *datasheet* sensor MQ135, dimana nilai pembacaan sensor berbanding lurus dengan tegangan keluaran yang dihasilkan yaitu semakin tinggi nilai tegangan keluaran sensor maka semakin tinggi nilai pembacaan sensor.
2. Pada penelitian ini telah dibuat sistem penentuan kesegaran daging sapi lokal berdasarkan warna dan kadar amonia pada daging sapi dengan menggunakan metode jaringan syaraf tiruan. Pada metode jaringan syaraf tiruan tersebut menggunakan empat neuron pada *input layer* dan menggunakan empat neuron pada *hidden layer* dan menggunakan tiga neuron pada *output layer*. Semua komponen alat yang digunakan maupun metode yang diterapkan dapat berjalan sesuai dengan yang diinginkan, terbukti dengan sistem dapat mengklasifikasikan jenis daging sapi menjadi daging sapi segar, sedang dan busuk.
3. Akurasi yang diperoleh sistem penentuan kesegaran daging sapi berdasarkan warna dan kadar amonia pada daging sapi lokal dengan metode jaringan syaraf tiruan yang diuji dengan jumlah data latih sebanyak 81 data dan data uji sebanyak 27 data adalah senilai 92,5 %

7.2 Saran

Adapun beberapa saran yang dapat dijadikan acuan untuk pengembangan penelitian ini ataupun penelitian yang serupa kedepannya adalah sebagai berikut:

1. Menggunakan metode atau algoritma klasifikasi lain untuk membandingkan metode atau algoritma manakah yang mempunyai tingkat keakuratan lebih tinggi.

2. Menggunakan sensor yang lebih khusus untuk mendeteksi gas amonia sehingga akan lebih meningkatkan keakuratan pembacaan konsentrasi gas amonia.
3. Menerapkan sistem serupa dengan tampilan antarmuka melalui *mobile apps* sehingga lebih memudahkan mobilitas penggunaan.



DAFTAR PUSTAKA

- Adhieputra, D. A. (2010). *Pemilihan Data Training untuk Meningkatkan Kinerja Voting Feature Interval 5 (VFI 5).*, Institut Pertanian Bogor, Ilmu Komputer, Bogor.
- Arduino. (2016). Arduino Uno Atmega328P. Dipetik Agustus 28, 2016, dari <https://www.arduino.cc/en/Main/ArduinoBoardUno>
- Buege. (2001). D.R. Dalam *Varition in pork lean quality*. District of Colombia: National Pork Producers Council.
- Chairunnas, A., & Sugianto, H. (2013). *Robot Pendeteksi Warna Berbasis Mikrokontroler*. SKRIPSI, Universitas Pakuan, Ilmu Komputer, Bogor.
- Eviyan F, T. W. (2017). Deteksi Daging Sapi menggunakan Electronic Nose Berbasis Bidirectional Associative Memory.
- Hasanudin, F. (2015). *Pemanfaatan Sensor Gas MQ135 untuk Pendeteksi Gas Amonia (NH3) pada Kotoran Ayam Berbasis Arduino Uno*. Tugas Akhir, Universitas Gadjah Mada, D3 Teknik Elektro, Yogyakarta.
- Kasmadiharja. (2008). *Kajian Penyimpanan Sosis, Naget Ayam dan Daging Ayam Berbumbu dalam Kemasan Polipropilen Rigid*. Bogor: Fakultas Teknologi Pertanian.
- Prasetyo E, E. U. (2014). Identifikasi Daging Segar dan Busuk Menggunakan Sensor Warna RGB dan PH Meter Digital.
- Priharsanti, A. H. (2009). Populasi Bakteri dan Jamur pada Daging Sapi dengan Penyimpanan. *Sains Peternakan*.
- Rosa A, D. P. (2017). Identifikasi Kesegaran Daging Sapi Berdasarkan Waktu
- Soeparno. (2005). Dalam *Ilmu dan teknologi daging cetakan keempat*. Yogyakarta: Gadjah Mada University Press.
- Suryani. (2015). DAGING SAPI. Dalam *OUTLOOK KOMODITAS PERTANIA SUBSEKTOR PETERNAKAN* . Jakarta: Pusat data dan sistem Informasi Sekretarian Jenderal Kementerian Pertanian.
- TAOS. (2009, July). *TCS3200, TCS3210 Programmable Color Light-To-Frequency Converter*. Texas: The Lumenology Company.
- Vishay. (2016, Oktober 27). *LCD-016M002B*. Diambil kembali dari Vishay Website: www.vishay.com
- Alexnld, 2018. [image online] Tersedia di: <https://alexnld.com/product/gy-31-tcs3200-color-sensor-recognition-module-for-arduino/> [Diakses 20 Agustus 2018]
- Sainsmart, 2018. [image online] Tersedia di: <https://www.sainsmart.commq-135-ammonia-detection-sensor-nh3-gas-sensor> [Diakses 22 Agustus 2018]