repository.ub.ac.id

SISTEM PEMBEDA DAGING SAPI DAN DAGING BABI BERDASARKAN WARNA DAN KADAR AMONIA MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN BERBASIS ANDROID

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh :

Muchamad Rafi Dharmawan
NIM: 145150301111064

PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2020

UNIVERSITAS BRAWIJAYA

PENGESAHAN

SISTEM PEMBEDA DAGING SAPI DAN DAGING BABI BERDASARKAN WARNA DAN KADAR AMONIA MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN BERBASIS ANDROID

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :
Muchamad Rafi Dharmawan
NIM: 145150301111064

Sripsi ini telah diuji dan dinyatakan lulus pada
06 Januari 2020
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Dahnial Syauqy, S.T., M.T., M.Sc.	Gembong Edhi Setyawan, S.ST, M.T
NIK: 201607 870423 1 002	NIK. 201208 761201 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, 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, 06 Januari 2020

Muchamad Rafi Dharmawan

NIM: 145150301111064

Universitas Brawijaya

ABSTRAK

Muchamad Rafi Dharmawan, Sistem Pembeda Daging Sapi dan Daging Babi Berdasarkan Warna dan Kadar Amonia Menggunakan Metode Jaringan Syaraf Tiruan Berbasis Android

Pembimbing : Dahnial Syauqy, S.T., M.T., M.Sc. Gembong Edhi Setyawan, S.ST, M.T

Manusia membutuhkan protein untuk dapat melakukan aktivitas sehari-hari. Salah satu sumber protein yang banyak diminati oleh masyarakat yaitu daging sapi. Tingginya minat masyarakat terhadap daging sapi menyebabkan pihak-pihak yang tidak bertanggung jawab memanfaatkan kondisi ini dengan mencampur daging sapi dan daging babi. Hal ini dikarenakan oleh tingginya harga daging sapi dan harga daging babi yang relatif murah serta kurangnya pengetahuan masyarakat dalam membedakan antara daging sapi dan daging babi. Dampak dari pencampuran tersebut tentu sangat merugikan bagi masyarakat Indonesia yang mayoritas muslim. Sehingga dari permasalahan tersebut, diperlukan adanya penelitian terkait dengan sistem pembeda jenis daging sapi dan daging babi yang diharapkan dapat meminimalisir kecurangan dan mengurangi kerugian bagi konsumen. Pada penelitian ini akan menggunakan warna dan bau sebagai parameter untuk membedakan jenis daging. Tujuan dilakukan penelitian ini agar dapat merancang sistem pembeda jenis daging sapi dan daging babi berdasarkan kedua parameter tersebut. Proses klasifikasi oleh sistem berdasarkan data yang diperoleh dari pembacaan sensor warna TCS3200 dan sensor gas MQ135 oleh mikrokontroler Arduino Nano menggunakan metode Jaringan Syaraf Tiruan. Hasil akurasi dari pengujian yang telah dilakukan oleh sistem pembeda daging sapi dan daging babi berdasarkan warna dan kadar amonia menggunakan metode Jaringan Syaraf Tiruan terhadap daging yang diuji sebanyak 60 data sebagai data latih dan 30 sampel daging sebagai data uji adalah sebesar 90%.

Kata kunci: daging sapi, daging babi, warna, kadar amonia, jaringan syaraf tiruan, backpropagation.

repository.ub.ac.id

ABSTRACT

Muchamad Rafi Dharmawan, System of Differentiating Beef and Pork Based on Color and Ammonia Levels Using the Artificial Neural Network Method Based on Android

Supervisors : Dahnial Syauqy, S.T., M.T., M.Sc. Gembong Edhi Setyawan, S.ST, M.T

Humans need protein to be able to do daily activities. One source of protein that is much in demand by the public is beef. The highly public interest in beef causes irresponsible parties to take advantage of this condition by mixing beef and pork. This is due to the high price of beef and the relatively cheap price of pork and the lack of public knowledge in distinguishing between beef and pork. The impact of mixing is certainly very detrimental to the Indonesian majority Muslim community. So that from these problems, research is needed regarding different types of beef and pork which are expected to minimize fraud and reduce losses for consumers. This study uses color and ammonia levels as parameters to distinguish the type of meat. The purpose of this study is to be able to design a system of differentiating types of beef and pork based on these two parameters. The classification processes by the system were based on data obtained from the reading of the TCS3200 color sensor and the MQ135 gas sensor by the Arduino Nano microcontroller using the Artificial Neural Network method. Accuracy results from the tests carried out by the beef and pork differentiator a system based on color and ammonia levels using the Artificial Neural Network method of the meat tested were 60 data as training data and 30 meat samples as a test data at 90%.

Keywords: *beef, pork, color, ammonia levels, artificial neural networks, backpropagation.*

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

SISTEM PEMBEDA DAGING SAPI DAN DAGING BABI BERDASARKAN WARNA DAN KADAR AMONIA MENGGUNAKAN METODE JARINGAN SYARAF TIRUAN BERBASIS ANDROID i
PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
KATA PENGANTAR iv
ABSTRAK vi
ABSTRACT vii
DAFTAR ISI viii
DAFTAR TABEL xii
DAFTAR GAMBAR xiv
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan masalah 2
1.3 Tujuan 3
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 Daging Sapi dan Daging Babi 6
2.2.2 Sensor Warna TCS3200 9
2.2.3 Sensor Gas MQ135 10
2.2.4 Mikrokontroler Arduino Nano (Atmega328P) 12
2.2.5 Modul Bluetooth HC-05 12
2.2.6 Metode Jaringan Syaraf Tiruan 13
2.2.6.1 Struktur Dasar dari Jaringan Syaraf Tiruan 14
2.2.6.2 Transfer Function 14
2.2.6.3 Algoritma Backpropagation 16
2.2.7 Matlab 18

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA
Universitas Brawijaya

2.2.8 Android 18
2.2.9 MIT App Inventor 19
BAB 3 METODOLOGI 20
3.1 Studi Literatur 21
3.2 Analisis Kebutuhan 22
3.3 Perancangan Sistem 22
3.3.1 Perancangan Perangkat Keras 22
3.3.2 Perancangan Perangkat Lunak 23
3.4 Pengumpulan Data 23
3.5 Implementasi Sistem 23
3.6 Pengujian dan Analisis 23
3.7 Kesimpulan 24
BAB 4 ANALISIS KEBUTUHAN 25
4.1 Deskripsi Umum Sistem 25
4.2 Kebutuhan Sistem 25
4.2.1 Kebutuhan Fungsional 25
4.2.2 Kebutuhan Non Fungsional 26
4.2.2.1 Kebutuhan Perangkat Keras 26
4.2.2.2 Kebutuhan Perangkat Lunak 29
4.3 Batasan Desain Sistem 30
BAB 5 PERANCANGAN DAN IMPLEMENTASI 31
5.1 Perancangan Sistem 31
5.1.1 Perancangan Protoype Alat 31
5.1.2 Perancangan Perangkat Keras 32
5.1.3 Perancangan Perangkat Lunak 34
5.1.3.1 Memberikan Input 35
5.1.3.2 Perancangan Pengambilan Data Sensor 35
5.1.3.3 Mengumpulkan Data Latih 37
5.1.3.4 Perancangan Pelatihan Jaringan Syaraf Tiruan 37
5.1.3.5 Perancangan menampilkan hasil klasifikasi pada aplikasi android 46
5.2 Implementasi Sistem 46

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA
Universitas Brawijaya

5.2.1 Implementasi *Prototype* Alat 47
5.2.2 Implementasi Perangkat Keras 47
5.2.3 Implementasi Perangkat Lunak 48
5.2.3.1 Implementasi Pemberian Input 48
5.2.3.2 Implementasi Kode Program Pengambilan Data Sensor 48
5.2.3.3 Implementasi Perangkat Lunak Pelatihan Data 54
5.2.3.4 Implementasi Kode Program Jaringan Syaraf Tiruan. 60
5.2.3.5 Implementasi menampilkan hasil klasifikasi pada aplikasi Android 62
BAB 6 PENGUJIAN DAN ANALISIS 67
6.1 Pengujian Sensor Warna TCS3200 67
6.1.1 Tujuan Pengujian 67
6.1.2 Prosedur Pengujian 67
6.1.3 Hasil dan Analisis Pengujian 68
6.2 Pengujian Sensor Gas MQ135 69
6.2.1 Tujuan Pengujian 69
6.2.2 Prosedur Pengujian 69
6.2.3 Hasil dan Analisis Pengujian 70
6.3 Pengujian modul Bluetooth HC-05 71
6.3.1 Tujuan Pengujian 71
6.3.2 Prosedur Pengujian 71
6.3.3 Hasil dan Analisis Pengujian 72
6.4 Pengujian Aplikasi Android pada *Smartphone* 73
6.4.1 Tujuan Pengujian 73
6.4.2 Prosedur Pengujian 73
6.4.3 Hasil dan Analisis Pengujian 73
6.5 Pengujian Akurasi Hasil Klasifikasi 74
6.5.1 Tujuan Pengujian 74
6.5.2 Prosedur Pengujian 74
6.5.3 Hasil dan Analisis Pengujian 74
6.6 Pengujian Waktu Komputasi Pemrosesan Sistem 76
6.6.1 Durasi Pelatihan Data 76

6.6.1.1 Tujuan Penelitian 76
6.6.1.2 Prosedur Pengujian 76
6.6.1.3 Hasil dan Analisis Pengujian 77
6.6.2 Durasi Prediksi Data 77
6.6.2.1 Tujuan Penelitian 77
6.6.2.2 Prosedur Pengujian 77
6.6.2.3 Hasil dan Analisis Pengujian 78
BAB 7 PENUTUP 80
7.1 Kesimpulan 80
7.2 Saran 81
DAFTAR PUSTAKA 82
LAMPIRAN A DATA LATIH 84
LAMPIRAN B KODE PROGRAM KESELURUHAN PADA ARDUINO 87

DAFTAR TABEL

Tabel 2.1 Nilai RGB pada Skor Standarisasi Warna Daging 7
Tabel 2.2 Hubungan pin S2 dan S3 terhadap Tipe Filter 9
Tabel 2.3 Hubungan pin S0 dan S1 terhadap Skala Frekuensi Keluaran Sensor 9
Tabel 4.1 Spesifikasi Arduino Nano Atmega328P 26
Tabel 4.2 Spesifikasi Sensor Warna TCS3200 27
Tabel 4.3 Spesifikasi Sensor Gas MQ135 28
Tabel 5.1 Keterangan koneksi pin modul Bluetooth HC-05 dengan Arduino Nano 33
Tabel 5.2 Keterangan koneksi pin sensor warna TCS3200 dengan Arduino Nano 33
Tabel 5.3 Keterangan koneksi pin sensor gas MQ135 dengan Arduino Nano 33
Tabel 5.4 Deklarasi bobot layer 1 40
Tabel 5.5 Deklarasi bobot layer 2 40
Tabel 5.6 Kode program deklarasi *library* sistem 49
Tabel 5.7 Kode program deklarasi variable pembacaan sensor 49
Tabel 5.8 Kode program kalibrasi sensor MQ135 50
Tabel 5.9 Kode program pembacaan nilai sensor gas MQ135 51
Tabel 5.10 Kode program pengaturan kalibrasi sensor warna TCS3200 51
Tabel 5.11 Kode pemrograman untuk proses filter warna 52
Tabel 5.12 Kode pemrograman pembacaan nilai sensor warna TCS3200 53
Tabel 5.13 Kode program pengecekan pemberian trigger 53
Tabel 5.14 Kode Program pelatihan Jaringan Syaraf Tiruan Backpropagation 55
Tabel 5.15 Nilai bobot dari proses pelatihan JST 59
Tabel 5.16 Kode program variable nilai bobot Jaringan Syaraf Tiruan 60
Tabel 5.17 Kode program fungsi log-sigmoid 61
Tabel 5.18 Kode program perhitungan *neuron layer* 1 61
Tabel 5.19 Kode program aktivasi fungsi *neuron layer* 1 61
Tabel 5.20 Kode program perhitungan *neuron layer* 2 62
Tabel 5.21 Kode program aktivasi fungsi *hardlimit neuron layer* 2 62
Tabel 6.1 Hasil Pengujian pembacaan sensor warna TCS3200 1
Tabel 6.2 Hasil Pengujian pembacaan sensor gas MQ135 70

Tabel 6.3 Data Uji dan hasil pengujian Jaringan Syaraf Tiruan 75
Tabel 6.4 Pengujian Durasi Pelatihan Data 77
Tabel 6.5 Pengujian Durasi Prediksi 78

DAFTAR GAMBAR

Gambar 2.1 Daging Sapi dan Daging Babi 6
Gambar 2.2 Standar Warna Daging Sapi 7
Gambar 2.3 Sinyal hasil pembacaan electronic nose (a) sapi (b) babi 8
Gambar 2.4 Sensor Warna tipe TCS3200 9
Gambar 2.5 Sensor Gas dengan tipe MQ135 10
Gambar 2.6 Grafik Sensitivitas Gas dari Sensor MQ135 11
Gambar 2.7 Tampilan titik koordinat pada grafik NH3 12
Gambar 2.8 Arduino Nano Atmega328P 12
Gambar 2.9 Modul Bluetooth dengan tipe HC-05 13
Gambar 2.10 Jaringan Layer Jamak 14
Gambar 2.11 Struktur Dasar dari JST 14
Gambar 2.12 Grafik Hardlimit Transfer Function 15
Gambar 2.13 Grafik Linear transfer function 15
Gambar 2.14 Grafik log-sigmoid transfer function 15
Gambar 2.15 Arsitektur Android 19
Gambar 3.1 Diagram Alir Metode Penelitian 20
Gambar 3.2 Blok Diagram Sistem 23
Gambar 4.1 Deskripsi Pin Sensor Warna TCS3200 27
Gambar 4.2 Deskripsi Pin Sensor MQ135 28
Gambar 4.3 Deskripsi Pin Bluetooth HC-05 29
Gambar 5.1 Desain Prototype Alat 31
Gambar 5.2 Diagram Skematik Perancangan Perangkat Keras Sistem 32
Gambar 5.3 Diagram alir perancangan keseluruhan sistem 34
Gambar 5.4 Diagram alir perancangan pengambilan data sensor 35
Gambar 5.5 Arsitektur perancangan JST 38
Gambar 5.6 Diagram alir pelatihan Backpropagatiom 39
Gambar 5.7 *mockup design* untuk aplikasi android 46
Gambar 5.8 Implementasi *prototype* sistem 47
Gambar 5.9 Implementasi rangkaian sensor warna TCS3200, sensor gas MQ135, dan modul Bluetooth HC-05 47

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA
Universitas Brawijaya

Gambar 5.10 Ukuran sampel daging 48
Gambar 5.11 Daging Sapi dan Daging Babi sebagai sampel 48
Gambar 5.12 Nilai R ketika *epoch* 1000 dan *learning rate* 0,1 56
Gambar 5.13 Nilai *error goal* (MSE) ketika *epoch* 1000 dan *learning rate* 0,1 56
Gambar 5.14 Nilai R ketika *epoch* 10000 dan *learning rate* 0,01 57
Gambar 5.15 Nilai *error goal* (MSE) ketika *epoch* 10000 dan *learning rate* 0,01 . 57
Gambar 5.16 Nilai R ketika *epoch* 100000 dan *learning rate* 0,001 58
Gambar 5.17 Nilai *error goal* (MSE) ketika *epoch* 100000 dan *learning rate* 0,001 58
Gambar 5.18 Desain antarmuka keseluruhan sistem 63
Gambar 5.19 Tampilan ketika tombol “Scan Bluetooth” ditekan 63
Gambar 5.20 Tampilan ketika Bluetooth sudah terhubung 64
Gambar 5.21 Tampilan ketika *push button* ditekan 64
Gambar 5.22 Tampilan ketika tombol “Disconnect” ditekan 65
Gambar 5.23 Tampilan ketika menekan “Ya” pada *notifier* 65
Gambar 5.24 Block program *pairing* Bluetooth dengan Aplikasi Android 66
Gambar 5.25 *Block* program pembacaan nilai *serial* Aplikasi Android 66
Gambar 6.1 Analisis Korelasi Kadar Amonia(PPM) dengan Volt 71
Gambar 6.2 Tampilan ketika modul *Bluetooth* HC-05 terdeteksi 72
Gambar 6.3 Tampilan ketika *Bluetooth* membutuhkan *password* untuk *pairing* . 72
Gambar 6.4 Tampilan ketika proses *Pairing* berhasil dilakukan 73
Gambar 6.5 Tampilan hasil pengolahan sistem pada Aplikasi 74
Gambar 6.6 Hasil Pengujian Durasi Prediksi 79

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Protein termasuk zat yang penting bagi tubuh manusia untuk dapat melakukan aktivitas sehari-hari. Karena protein berfungsi sebagi zat pembangun dan pendorong metabolisme pada tubuh. Daging sapi termasuk satu dari berbagai jenis sumber protein yang digemari oleh sebagian besar penduduk (Rahma, 2016). Selain banyak mengandung protein, sebagian besar penduduk lebih menyukai mengkonsumsi daging sapi karena mudah mengolahnya menjadi beragam masakan. Tingginya minat masyarakat pada daging sapi dapat dilihat pada terus meningkatnya nilai pertumbuhan konsumsi daging sapi oleh masyarakat Indonesia dari tahun ke tahun. Konsumsi per kapita daging sapi masyarakat Indonesia tahun 2017 hingga 2021 rata-rata tumbuh 6,30% per tahun (Riniarsi, 2017).

Dengan tingginya minat masyarakat terhadap daging sapi, menyebabkan adanya pedagang nakal yang memanfaatkan situasi ini dengan menjual daging sapi dicampur daging babi. Seperti yang terjadi di daerah Lubuklinggau, Sumatera Selatan Polisi menciduk dua pedagang pasar Inpres karena diketahui mengoplos daging sapi dengan daging babi (Siregar, 2017). Kondisi ini disebabkan karena lebih murahnya harga daging babi dan juga minimnya pengetahuan masyarakat tentang perbedaan antara daging sapi dan daging babi. Dampak dari pencampuran tersebut tentu sangat merugikan bagi konsumen ditambah dengan mayoritas masyarakat Indonesia adalah beragama Islam. Untuk itu dibutuhkan sebuah alat untuk dapat membantu masyarakat dalam membedakan daging sapi dan daging babi serta melindungi agar terhindar dari masalah yang telah disebutkan diatas.

Salah satu cara untuk membedakan jenis daging sapi dan babi yang banyak digunakan saat ini yaitu menggunakan *image processing.* Penelitian menggunakan *image processing* sudah pernah dilakukan oleh (Muhammadiy & Fahmi, 2016). Penelitian serupa juga pernah dilakukan oleh (Hartono, 2012). Hasil dari kedua penelitian tersebut menunjukkan bahwa daging sapi dan babi dapat dibedakan menggunakan proses *image processing*. Namun, pada *image processing* terdapat beberapa kelemahan diantaranya faktor cahaya saat pengambilan data dapat mempengaruhi hasil pembacaan sistem. Selain itu, untuk mendapatkan data yang dibutuhkan oleh sistem harus melalui beberapa tahap diantaranya *image acquisition, cropping and normalization process, grayscale,* dan *histogram equalization.* Sehingga peneliti mencoba menggunakan cara lain untuk dapat membedakan kedua jenis daging tersebut.

Ada beberapa karakteristik yang digunakan untuk membedakan antara kedua daging tersebut seperti warna, serat, tekstur, dan aroma daging (Nafiasari & Handayani, 2018). Pada penelitian ini akan memanfaatkan warna dan aroma sebagai parameter untuk mengetahui perbedaan kedua daging tersebut. Menurut (Hadi, 2011) dari segi warna, daging sapi cenderung lebih cerah. Sedangkan dari segi aroma, daging babi lebih amis. Sementara jika menggunakan serat dan tekstur sebagai parameter, akan membutuhkan kamera sebagai komponen utama dalam

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

sistem dan tentunya akan kembali menggunakan metode *image processing*. Sehingga dengan permasalahan yang telah disebutkan diatas, peneliti mencoba memanfaatkan sensor warna dan sensor gas untuk membedakan antara daging sapi dan babi. Selain untuk mengatasi permasalahan diatas, diharapkan juga penggunaan sensor warna dan sensor gas dapat mengurangi kompleksitas sistem sehingga sistem dapat bekerja lebih cepat.

Pada penelitian yang diusulkan ini akan menggunakan metode Jaringan Syaraf Tiruan (JST) sebagai pengambil keputusan. JST digunakan karena termasuk sebuah paradigma pemrosesan yang bekerja seperti halnya manusia yaitu belajar berdasarkan contoh atau pengalaman (Simamora, 2017). Dalam penelitian ini, peneliti menggunakan Jaringan Syaraf Tiruan *Backpropagation* karena karakteristik dari *backpropragation* tersebut yaitu dapat melatih jaringan agar memperoleh hasil yang seimbang antara kinerja jaringan untuk mengidentifikasi pola selama proses *training* dan mengirim respon yang tepat kepada pola masukan baru yang sejenis tetapi berbeda dengan pola saat *training* (Hartono, 2012). Selain itu metode JST ini digunakan karena memiliki tingkat akurasi yang lebih baik dibandingkan dengan metode lain seperti KNN (*K-Nearest Neighbour*) dan *Naïve Bayes* dalam penerapannya untuk mengklasifikasikan antara daging sapi dan babi. Hal ini ditunjukkan pada hasil penelitian oleh (Muhammadiy & Fahmi, 2016) yang menggunakan metode JST memperoleh tingkat akurasi sebesar 89%. Sementara penerapan metode KNN oleh (Budianita, et al., 2015) memperoleh tingkat akurasi sebesar 73,37% pada pengujian dengan *background.* Sedangkan pada penerapan metode *Naïve Bayes* oleh (Wijaya, et al., 2017) memperoleh tingkat akurasi sebesar 75%.

Sistem ini akan mengaplikasikan sensor warna TCS3200 dan sensor gas MQ135 sebagai pendeteksi jenis daging serta mikrokontroler Arduino Nano sebagai pemroses dan pengolah data. Data hasil klasifikasi akan dikirim oleh modul Bluetooth HC-05 dan menampilkan hasilnya pada aplikasi ponsel android. Dari pembuatan sistem ini peneliti berharap agar sistem dapat menjadi solusi atas berbagai permasalahan yang disebutkan diatas dan diperoleh sistem pendeteksi jenis daging yang lebih efisien.

1.2 Rumusan masalah

Dengan permasalahan yang dihadapi, penulis merumuskan masalah sebagai berikut:

1. Bagaimana penerapan sensor warna TCS3200 dan sensor gas MQ135 untuk mengakuisisi nilai warna dan kadar amonia pada daging sapi dan daging babi?
2. Bagaimana hasil penerapan perangkat sistem dan metode Jaringan Syaraf Tiruan untuk mengklasifikasikan jenis daging sapi dan daging babi?
3. Berapakah hasil akurasi dari sistem dalam membedakan daging sapi dan daging babi?

4. Bagaimanakah kinerja dari Sistem Pembeda Daging Sapi Dan Daging Babi Berdasarkan Warna dan Kadar Amonia Menggunakan Metode Jaringan Syaraf Tiruan Berbasis Android?

1.3 Tujuan

Merujuk pada rumusan masalah diatas, didapatkan tujuan penelitian sebagai berikut:

1. Dapat menggunakan sensor warna TCS3200 dan sensor gas MQ135 untuk mengakuisisi nilai dari warna dan kadar amonia pada daging sapi dan daging babi.
2. Dapat mengimplementasikan metode JST untuk sistem pendeteksi daging sapi dan daging babi.
3. Dapat mengetahui hasil akurasi dari sistem untuk membedakan antara daging sapi dan daging babi.
4. Mengetahui performansi dari sistem dalam membedakan jenis daging sapi dan babi.

1.4 Manfaat

Penelitian ini dilakukan untuk menghasilkan suatu sistem yang memudahkan masyarakat atau konsumen dalam mengetahui dan membedakan antara daging sapi dan babi. Serta meminimalisir adanya kecurangan oleh pedagang nakal berupa pengoplosan kedua jenis daging tersebut saat menjual daging.

1.5 Batasan masalah

Supaya pembahasan tidak melenceng dan lebih terfokus sesuai dengan latar belakang yang dipaparkan, maka dibuat batasan masalah di bawah:

1. Objek yang dideteksi berupa daging sapi dan daging babi yang masih segar.
2. Parameter yang digunakan untuk membedakan daging sapi dan daging babi yaitu warna dan aroma daging.
3. Aplikasi android hanya menampilkan *output* berupa nilai sensor dan hasil klasifikasi yang dilakukan oleh sistem.

1.6 Sistematika pembahasan

Pada penulisan laporan penelitian akan dibagi menjadi beberapa bab yang akan di jelaskan di bawah:

BAB I : Pendahuluan

Akan menjelaskan peristiwa yang melatar belakangi, hipotesis sementara, merumuskan masalah, tujuan, manfaat serta batasan pada penelitian.

BAB II : Landasan Kepustakaan

Akan menjelaskan penelitian atau pustaka sebelumnya yang akan dijadikan refrensi oleh peneliti. Selain penelitian sebelumnya juga dijelaskan dasaran teori yang juga digunakan sebagai refrensi pemahaman secara teori oleh peneliti.

BAB III : Metodologi Penelitian

Akan menjelaskan bagaimana langkah atau proses dilakukannya penelitian, baik proses perancangan sistem sampai tahap penulisan laporan penelitian. Selain itu bab ini akan menjelaskan lebih detail isi setiap bab pada laporan penelitian.

BAB IV : Analisis Kebutuhan

Akan menjelaskan gambaran umum dari sistem, analisis kebutuhan fungsional dan nonfungsional, kebutuhan perangkat keras dan lunak yang diperlukan dalam merancang sistem.

BAB V : Perancangan dan Implementasi

Akan menjelaskan perancangan dan implementasi sistem berupa implementasi perangkat keras dan lunak dalam membangun sistem. Serta penggunaan metode JST untuk melakuan klasifikasi perbedaan jenis daging sapi dan daging babi.

BAB VI : Pengujian dan Analisis

Akan menjelaskan prosedur yang dilaukan dalam tahap pengujian sampai menganalisis hasil dari pengujian yang sudah dilaksanakan.

BAB VII: Penutup

Akan menjelaskan kesimpulan dari dilakukannya penelitian ini secara keseluruhan dan kekurangan pada penelitian yang sudah dilakukan dalam bentuk saran.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian pustaka mengulas mengenai penelitian yang pernah ada dan diusulkan. Kegiatan ini bertujuan untuk mengkaji hasil dari penelitian sebelumnya untuk menjadi acuan selama proses penelitian.

Pada penelitian mengenai klasifikasi kesegaran daging yang memanfaatkan sensor warna dan sensor gas oleh (Simamora, 2017). Penelitian tersebut memanfaatkan sensor warna TCS3200 serta sensor bau (*electronic nose*) sebagai alat untuk mengklasifikasi kesegaran daging. *Electronic nose* pada sistem ini terdiri dari tiga buah sensor yaitu MQ-136, MQ-137, dan TGS 2602. Dari penelitian tersebut didapatkan hasil bahwa sensor warna dapat merespon ketika diberikan masukan berdasarkan parameter tingkat kesegaran daging yang berbeda. Selain itu sensor gas (*electronic nose*) juga dapat membedakan aroma dari daging berdasarkan tingkat kesegarannya.

Penelitian selanjutnya tentang implementasi Jaringan Syaraf Tiruan (JST) Backpropagation oleh (Hartono, 2012). Dari penelitian tersebut dapat dilihat bahwa penggunaan metode Jaringan Syaraf Tiruan sebagai pengambil keputusan adalah langkah yang tepat. Karena Jaringan Syaraf Tiruan (JST) dapat bekerja seperti halnya manusia yaitu bekerja berdasarkan contoh atau pengalaman. Selain itu algoritma *Backpropagation* dapat menjaga keseimbangan antara pola *training* dan juga pola masukan baru. Sehingga dengan kemampuan tersebut diharapkan metode ini dapat meminimalisir kesalahan. Hal itu dapat dibuktikan dengan didapatkan tingkat keberhasilan sebesar 88.3% dari sistem tersebut.

Dari penelitian yang pernah dilakukan, kali ini peneliti mencoba menggunakan sensor warna tipe TCS3200 dan sensor gas tipe MQ135. Kedua sensor tersebut dipilih karena kemampuan yang dimiliki. Sensor TCS3200 sebagai pembaca nilai RGB pada daging. Sedangkan sensor MQ135 digunakan sebagai pengganti *electronic nose* karena diharapkan dapat menekan biaya pembuatan sistem, serta karakteristiknya yang dapat digunakan untuk mendeteksi gas amonia yang ada di udara.

Selain untuk mendeteksi warna dan bau dari daging, namun penggunaan kedua sensor tersebut diharapkan juga dapat membedakan jenis daging. Objek dalam penelitian ini yakni menggunakan daging segar dari sapi dan babi. Dengan memanfaatkan metode jaringan syaraf tiruan sebagai pengambil keputusan, diharapkan dapat mendapatkan hasil pengolahan sistem yang maksimal. Sistem ini nanti akan menampilkan *output* berupa nilai dari pembacaan sensor yaitu R, G, B, dan PPM serta hasil klasifikasi dari pengolahan sistem apakah daging yang dideteksi termasuk daging sapi atau daging babi. Hasil tersebut dapat dilihat pada ponsel android dengan cara menghubungkan ponsel yang sudah terpasang aplikasi dengan sistem melalui konektivitas *Bluetooth.* Pada penelitian ini peneliti memanfaatkan modul Bluetooth HC-05 untuk menghubungkan antara sistem

dengan ponsel android karena diharapkan dapat mempermudah pengguna dalam menggunakan sistem tersebut.

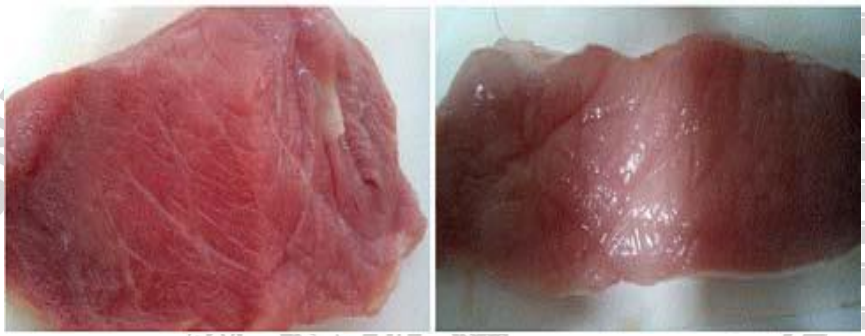
2.2 Dasar Teori

Adapun beberapa teori yang digunakan sebagai dasar dalam menyusun penelitian ini adalah:

2.2.1 Daging Sapi dan Daging Babi

Menurut (Riasari, 2014) Daging yaitu keseluruhan bagian pada hewan yang aman untuk dikomsumsi oleh manusia. Daging termasuk salah satu sumber protein yang amat baik karena kandungan asam amino,kandungan vitamin, dan mineral yang ada didalamnya.

Daging babi dan daging sapi secara karakteristik sebenarnya berbeda. Orang biasanya membedakan berdasarkan warna, bau dan serat daging. Babi biasanya memiliki warna yang lebih pucat dan serat yang kurang padat dengan bau khas spesifik pada babi.



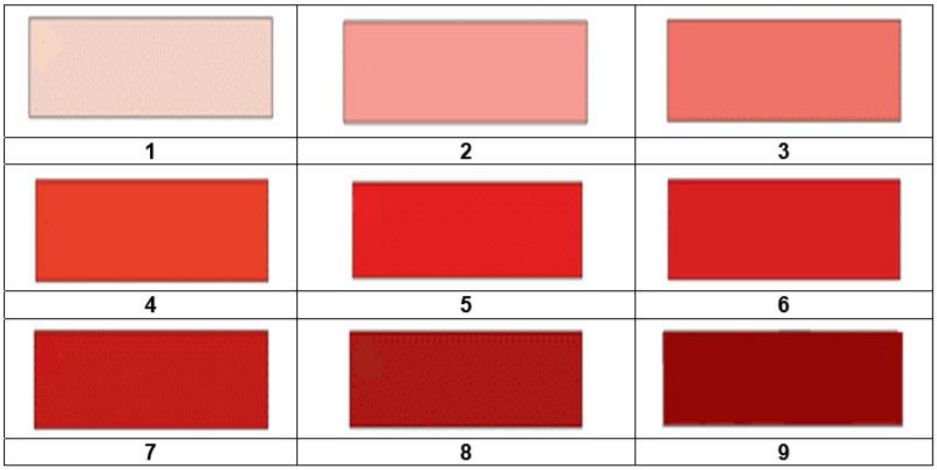
Gambar 2.1 Daging Sapi dan Daging Babi

Sumber : (Muhammadiy & Fahmi, 2016)

a. Warna Daging

Dari segi warna daging, bisa dilihat bahwa daging sapi memiliki warna yang lebih terang daripada daging babi. Selain itu daging babi biasanya cenderung berwarna lebih pucat. Berdasarkan (SNI, 2008) standar warna daging terdiri dari 9 skor mulai dari warna merah muda hingga merah tua, seperti yang terlihat pada Gambar 2.2.

Warna daging dapat diukur menggunakan notasi atau dimensi warna tristimulus. Ketiga notasi warna didefinisikan sebagai *hue*: warna utama (Red, Green, Blue). Setiap warna dapat terbentuk melalui kombinasi dari ketiga warna utama tersebut. Suatu sistem yang direkomendasikan *International Commision on Illuminattion* (CIE) tentang pengukuran warna daging berprinsip pada menambah atau mengurangi nilai setiap hue agar nilai suatu warna didapatkan.



Gambar 2.2 Standar Warna Daging Sapi

Sumber: (SNI, 2008)

Untuk itu diperlukan pengukuran nilai RGB terhadap sembilan skor warna yang terdapat pada standar warna daging sehingga dapat dijadikan sebagai acuan dalam melakukan penelitian ini. Peneliti menggunakan *brush tool* pada aplikasi *Paint 3D* sebagai sarana untuk memperoleh nilai RGB pada masing-masing warna dalam kolom skor standarisasi warna daging. Hasil pembacaan dari aplikasi tersebut dapat dilihat pada Tabel 2.1.

Tabel 2.1 Nilai RGB pada Skor Standarisasi Warna Daging

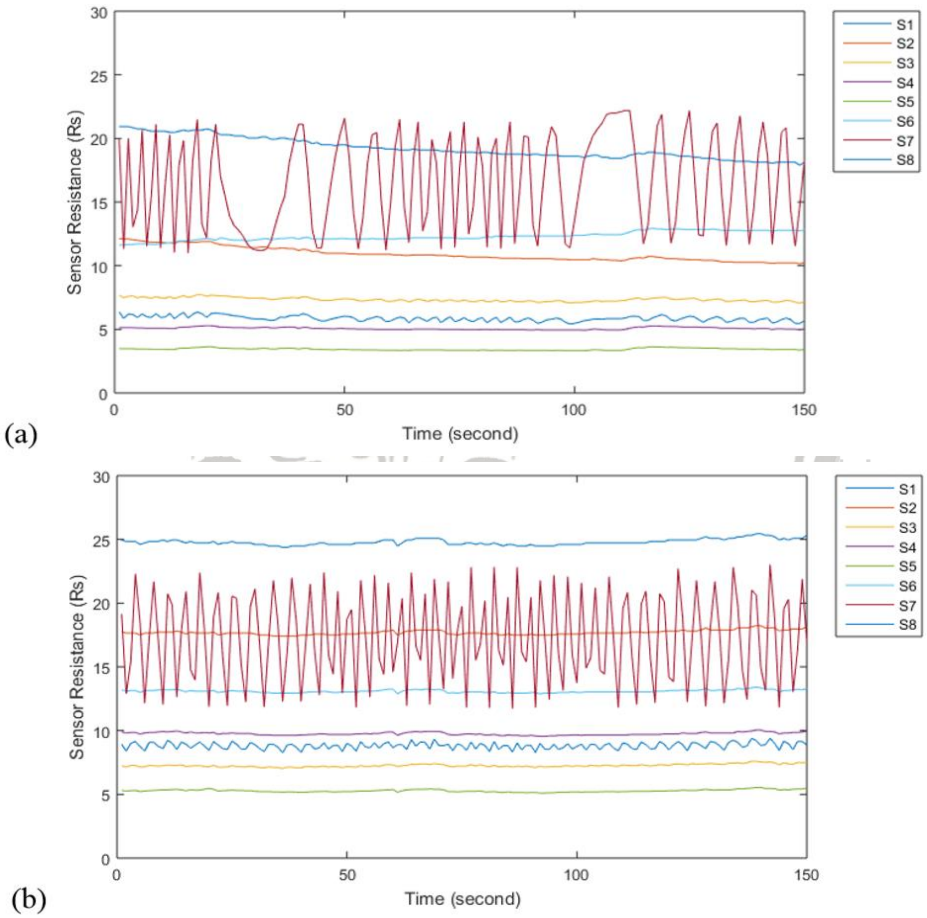
Skor ke-	Nilai RGB melalui aplikasi Paint 3D
1	RGB = (240, 210, 198)
2	RGB = (247, 156, 148)
3	RGB = (239, 115, 104)
4	RGB = (231, 66, 41)
5	RGB = (227, 33, 33)
6	RGB = (214, 33, 33)
7	RGB = (193, 31, 24)
8	RGB = (170, 23, 24)
9	RGB = (148, 8, 8)

b. Aroma Daging

Dari segi aroma/bau, daging sapi cenderung berbau amis tapi segar. Sedangkan daging babi cenderung mengandung bau yang lebih amis. Menurut (Merthayasa, et al., 2015) bau daging merupakan pengaruh campuran dari

aktivitas enzim *lipolitik triasilgliserol*, ketengikan oksidatif asam lemak tak jenuh serta produk degradasi protein yang terakumulasi dalam jaringan lemak. Produk degradasi protein daging dapat diketahui dari pelepasan gas-gas *amonia* (NH_3), dan *hydrogen sulfida* (H_2S) serta *metil merkaptan*.

Sementara, menurut (Wijaya, et al., 2017) dalam penelitiannya menggunakan *electronic nose* untuk membedakan antara daging sapi dan babi menunjukkan bahwa terdapat perbedaan bau/aroma yang cukup signifikan diantara kedua daging tersebut. Misalnya pada sensor S7 (NH_3) menghasilkan respon yang berbeda antara daging sapi dan babi, dimana bau dari daging babi memiliki tingkat frekuensi lebih tinggi. Hasil tersebut menegaskan bahwa daging babi berbau seperti amonia. Hasil pembacaan electronic nose dapat dilihat pada Gambar 2.3, dimana (a) daging sapi dan (b) daging babi.



Gambar 2.3 Sinyal hasil pembacaan electronic nose (a) sapi (b) babi

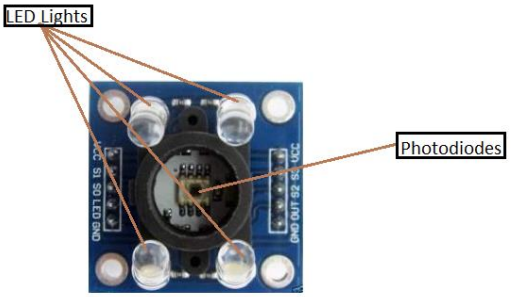
Sumber: (Wijaya, et al., 2017)

c. Serat Daging

Sedangkan dari segi serat, daging sapi mempunyai serat yang lebih padat dibandingkan dengan serat yang ada pada daging babi.

2.2.2 Sensor Warna TCS3200

TCS3200 ialah *Integrated Circuid* yang digunakan untuk merubah warna cahaya menjadi nilai frekuensi. Sensor ini menghasilkan *output* berbentuk gelombang persegi dengan *duty cycle* (panjang pulsa high dalam satu periode sinyal) sebanyak 50% yang frekuensinya berbanding lurus dengan intensitas cahaya (Simamora, 2017).



Gambar 2.4 Sensor Warna tipe TCS3200

Sumber: (circuitstoday.com)

Pada sensor warna dengan tipe TCS3200, pengkonversi dari cahaya menjadi frekuensi membaca 8x8 array fotodioda. 16 fotodioda berfilter merah, 16 fotodioda berfilter hijau, 16 fotodioda berfilter biru, serta 16 fotodioda bening tanpa filter. Setiap 16 fotodioda terhubung secara parallel (Inc, 2009). Jadi jika ingin membaca warna, dapat menggunakan kombinasi dari pin S2 dan S3. Kombinasi pin S2 dan S3 untuk dapat membaca warna RGB dapat dilihat pada tabel 2.2.

Tabel 2.2 Hubungan pin S2 dan S3 terhadap Tipe Filter

S2	S3	Warna
Low	Low	Merah
High	High	Hijau
Low	High	Biru
High	Low	Bening

Sumber : (Inc, 2009)

Selain itu, pada sensor ini juga terdapat dua pin lagi yaitu S0 dan S1. Pin ini digunakan untuk mengatur frekuensi *output* dari sensor. Kombinasi pin untuk mengatur frekuensi melalui pin S0 dan S1 adalah sebagai berikut:

Tabel 2.3 Hubungan pin S0 dan S1 terhadap Skala Frekuensi Keluaran Sensor

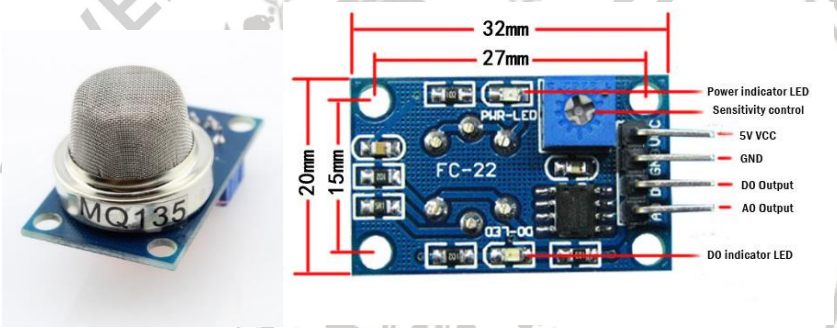
S0	S1	Output Frequency
High	High	100%
High	Low	20%

Low	High	2%
Low	Low	0%

Sumber : (Inc, 2009)

2.2.3 Sensor Gas MQ135

Sensor gas dengan tipe MQ135 ini mampu membaca macam-macam jenis aroma gas diantaranya NH_3, Nox, Benzene, Alkohol, asap, CO_2, dan aroma gas lainnya sehingga sensor gas ini biasanya digunakan sebagai pengontrol kualitas udara. Sensor ini mempunyai prinsip kerja yaitu dengan cara membaca nilai konduktivitas gas yang terdapat di sekitar sensor selaras dengan resistansi pada pin analog. Sensor akan berfungsi apabila mendapat input dengan nilai tegangan 5V. Sensor mampu membaca nilai gas pada rentang jarak 10-1000ppm dengan rentang nilai yang berbeda-beda setiap gasnya, namun untuk kandungan amonia, sensor ini dapat membaca pada rentang 10-300ppm NH_3 (MQ135_datasheet, 2016).



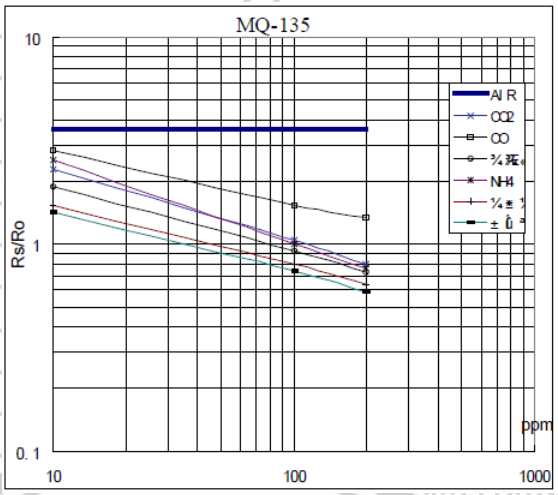
Gambar 2.5 Sensor Gas dengan tipe MQ135

Sumber : (hotmcu.com)

Menurut (Rahmana, 2019) untuk mendapatkan hasil pembacaan sensor MQ135 yang stabil dan akurat, maka diperlukan kalibrasi. Adapun langkah-langkah dalam melakukan kalibrasi terhadap sensor MQ135adalah sebagai berikut:

1. Langkah pertama yaitu mengukur nilai Ro dalam udara bersih. Nilai Ro/Rs pada NH3 dapat diperoleh dengan melihat pada grafik datasheet sensor gas MQ135 dibawah ini.

$$\frac{Ro}{Rs} = 3{,}6 \quad (2.1)$$



Gambar 2.6 Grafik Sensitivitas Gas dari Sensor MQ135

Sumber: (MQ135_datasheet, 2016)

2. Langkah kedua yaitu mengetahui nilai Rs.

$$Rs = \left(\frac{Vc}{VRL} - 1\right) * RL \quad (2.2)$$

3. Langkah ketiga adalah mengetahui nilai Ro.

$$Ro = \frac{Rs}{3{,}6} \quad (2.3)$$

4. Hubungan antara rasio Rs/Ro dengan PPM ditulis ibarat persamaan ini:

$$\log(y) = m * \log(x) + b \quad (2.4)$$

Keterangan:
y = rasio (Rs/Ro)
x = PPM
m = kemiringan garis
b = perpotongan titik

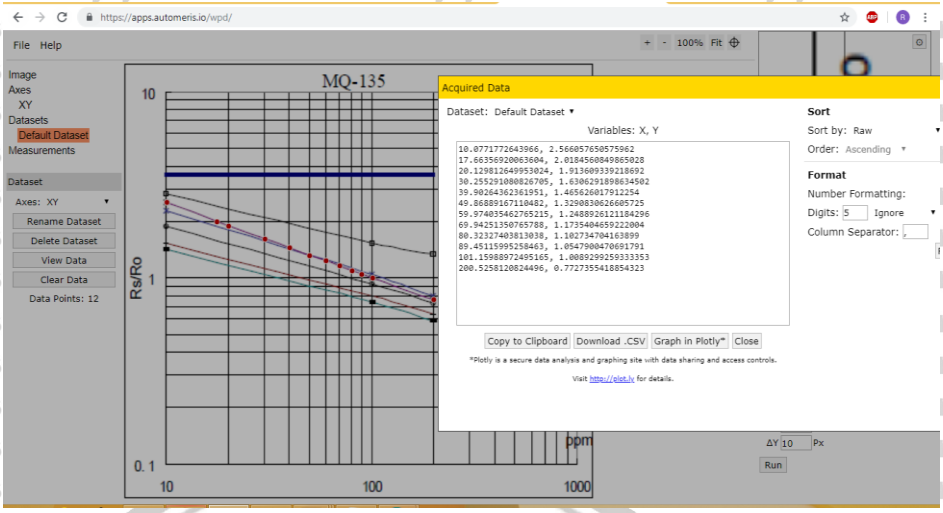
5. Diperlukan setidaknya dua titik koordinat pada persamaan garis untuk mendapatkan nilai m dan b, yakni (x_1 dan y_1) serta (x_2 dan y_2).

$$m = \frac{[\log(y2) - \log(y1)]}{[\log(x2) - \log(x1)]} \quad (2.5)$$

Nilai b didapatkan dengan cara mengambil nilai tengah (x, y) antara dua titik koordinat yang telah ditentukan sebelumnya.

$$b = \log(y) - \mathrm{m} * \log(x) \quad (2.6)$$

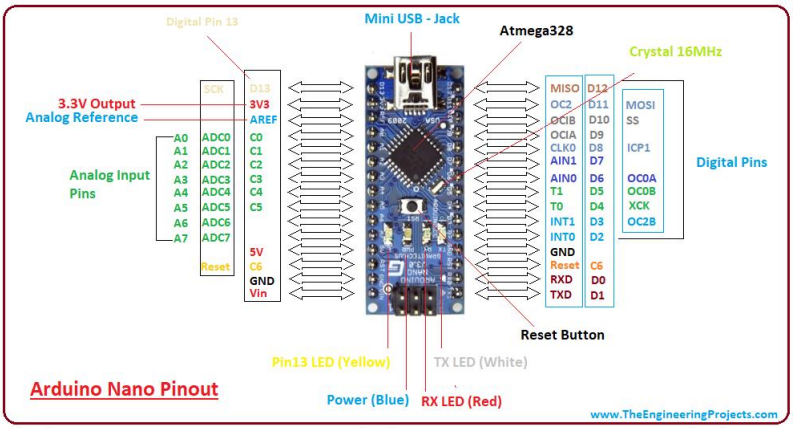
Setelah dilakukan perhitungan menggunakan webplot digitizer pada grafik NH3, titik koordinatnya terlihat pada Gambar 2.7:



Gambar 2.7 Tampilan titik koordinat pada grafik NH3

2.2.4 Mikrokontroler Arduino Nano (Atmega328P)

Arduino Nano yaitu mikrokontroler bersifat *open source,* mikrokontroler ini berupa chip yang bisa diprogram melalui komputer. Melalui Arduino Nano ini pengembang dapat merubah kode program melalui komputer menggunakan aplikasi yang sudah disediakan oleh Arduino. Untuk spesifikasi sebenarnya tidak jauh berbeda dengan Arduino Uno, yang membedakan yaitu pada Arduino Nano memiliki lebih banyak pin analog (8pin) dan juga bentuknya yang lebih kecil (45mm x 18mm). Untuk dapat mengaktifkan Arduino Nano juga memerlukan tipe koneksi USB yang berbeda dengan Arduino Uno, yaitu menggunakan koneksi Mini - B USB (Arduino, 2019).



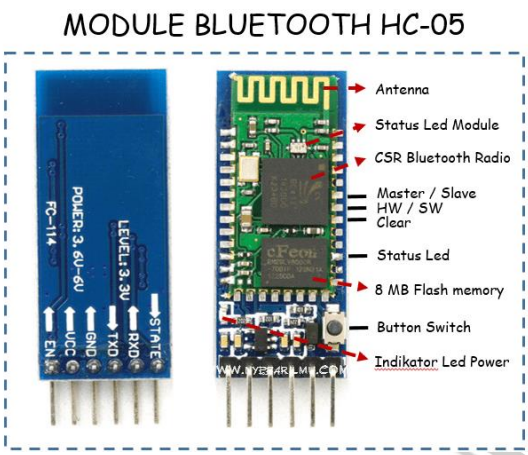
Gambar 2.8 Arduino Nano Atmega328P

Sumber : (theengineeringprojects.com)

2.2.5 Modul Bluetooth HC-05

Modul Bluetooth ialah penerapan komunikasi tanpa kabel yang memiliki kecepatan data sebesar 3Mbps. Modul ini dapat bekerja pada tegangan 1.8V-3.6V

dengan daya pancar RF mencapai +4dBm serta mempunyai sensitifitas -80dBm (Lamidi, 2018).



Gambar 2.9 Modul Bluetooth dengan tipe HC-05

Sumber:(nyebarilmu.com)

2.2.6 Metode Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan (JST) yakni suatu pola dari pemrosesan informasi yang berlandaskan pada cara kerja sistem syaraf biologis layaknya otak ketika melakukan proses terhadap suatu informasi. JST terdiri atas hubungan antar neuron yang membentuk satu kesatuan dalam menyelesaikan sebuah permasalahan. Layaknya manusia, JST belajar berlandaskan contoh atau kejadian sebelumnya. JST dapat digunakan pada aplikasi spesifik layaknya pengenalan pola ataupun klasifikasi data menggunakan proses pembelajaran (Simamora, 2017).

Menurut (Maru'ao, 2010) model jaringan syaraf tiruan berdasarkan arsitekturnya terbagi dalam 3 kelompok, diantaranya:

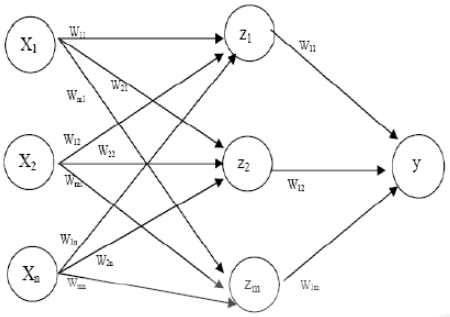
1. Jaringan Layer Tunggal (*Single Layer Network*)

   Pada jaringan ini, sekumpulan neuron *input* dan *output* saling terhubung, sinyal mengalir melalui *layer* (lapisan) *input* menuju ke *layer output*.

2. Jaringan Layar Jamak (*Multiple Layer Network*)

   Jaringan ini termasuk pengembangan dari jaringan *layer* tunggal. Karena terdapat unit-unit lain selain *input* dan *output layer* yakni lapisan tersembunyi (*hidden layer*).

   Algoritma *Backpropagation* termasuk salah satu model jaringan layer jamak.



Gambar 2.10 Jaringan Layer Jamak

Sumber : (Maru'ao, 2010)

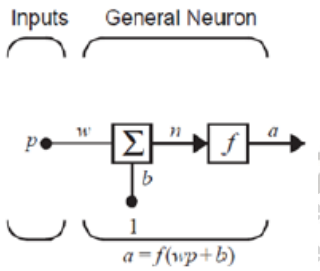
Dalam gambar 2.10 terlihat sebuah jaringan dengan *input* unit sebanyak n (x1, x2, ..., xn), dengan *hidden layer* sebanyak m (z1, z2, … , zm) serta terdapat 1 buah unit sebagai *output layer* (y).

3. Jaringan *Reccurent*

   Pola dari jaringan *reccurent* serupa dengan kedua model jaringan diatas. Namun terdapat sinyal yang dikirim oleh simpul keluaran terhadap unit masukan atau biasa disebut *feedback loop.*

2.2.6.1 Struktur Dasar dari Jaringan Syaraf Tiruan

Gambar 2.11 memperlihatkan contoh bagian yang paling sederhana pada struktur Jaringan Syaraf Tiruan yaitu *neuron* dengan *input* tunggal. Dimana *input* p dikalikan dengan *weight* (w) akan menghasilkan wp yang nantinya akan digunakan sebagai salah satu *input summer*. *Input* lain yang dibutuhkan oleh *summer* yaitu angka 1 untuk dikalikan dengan bias (b). Kemudian *summer* menghitung jumlah seluruh *input* sehingga menghasilkan *output* (n) yang nantinya akan menuju pada sebuah *transfer function* (f) sehingga didapatkan nilai akhir (a) dari *neuron* tersebut (Simamora, 2017).



Gambar 2.11 Struktur Dasar dari JST

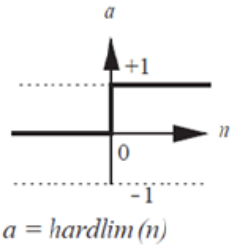
Sumber: (Simamora, 2017)

2.2.6.2 Transfer Function

Menurut (Simamora, 2017) pada *Neural Network* umumnya menggunakan beberapa *transfer function* berikut ini:

a. Hardlimit Transfer Function

Transfer function ini biasanya digunakan untuk menggolongkan *input* menjadi dua kategori, biasanya nilai *output* yang dihasilkan akan bernilai 0 atau 1. *Output* bernilai 0 apabila mendapatkan nilai *input* kurang dari 0. Namun akan bernilai 1 apabila mendapatkan nilai *input* lebih dari atau sama dengan 0.

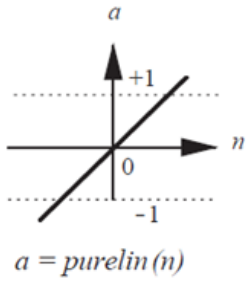


Gambar 2.12 Grafik Hardlimit Transfer Function

Sumber: (Simamora, 2017)

b. Linear Transfer Function

Karakteristik yang dimiliki yakni nilai *input* dan *output* yang diterima dari *transfer function* hasilnya ekuivalen.

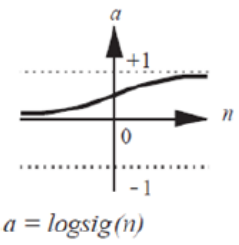


Gambar 2.13 Grafik Linear transfer function

Sumber: (Simamora, 2017)

c. Log-sigmoid Transfer Function

Secara umum dimanfaatkan untuk *multilayer networks* yang dilatih dengan *backpropagation.*



Gambar 2.14 Grafik log-sigmoid transfer function

Sumber: (Simamora, 2017)

Log-sigmoid mempunyai sebuah ciri khas yaitu mampu menerima *input* dengan nilai -∞ sampai-sampai +∞ dan menyusutkan nilai tersebut pada bentang nilai antara 0 dan 1 dalam *output*. Perhitungan matematika dari *log-sigmoid transfer function* yaitu:

$$a = \frac{1}{1+e^{n}} \quad (2.7)$$

2.2.6.3 Algoritma Backpropagation

Jaringan Syaraf Tiruan dikembangkan dengan tujuan untuk mengetahui suatu himpunan optimal berdasarkan bobot dan bias sebagai parameter sehingga fungsi dari JST dapat mewakili atau setidaknya mendekati perilaku dari permasalahan sebenarnya melalui proses *training*. (Simamora, 2017).

Backpropagation melatih kinerja jaringan untuk mengidentifikasi pola saat *training* dan kemampuannya dalam merespon pola masukan sejenis tapi tak sama dengan pola pada saat *training* agar bisa seimbang (Hartono, 2012).

Dalam pelatihan backpropagation fase yang digunakan ada tiga, pertama yakni fase maju (*feed forward*) yang menghitung maju pola masukan mulai dari lapisan *input* hingga *output*. Kedua yaitu fase mundur (*backpropagation*) dimana setiap unit *output* akan memperoleh target pola yang berkaitan dengan pola *input* untuk menghitung nilai kesalahannya. Nilai kesalahan itulah yang akan dipropagasikan mundur. Dan yang terakhir adalah fase modifikasi bobot yang memiliki tujuan untuk meminimalisir terjadinya kesalahan. Semua fase akan melakukan perulangan hingga sudah memenuhi kondisi penghentian (Nurmila, et al., n.d.).

Menurut (Maru'ao, 2010) ada beberapa syarat yang harus terpenuhi untuk melakukan fungsi aktivasi *backpropagation* diantaranya kontinu, terdeferensial dengan mudah, dan termasuk fungsi yang tidak turun. Fungsi sigmoid biner dengan range (0,1) termasuk satu dari sekian fungsi yang mampu memenuhi kebutuhan dari syarat-syarat tersebut.

Fungsi sigmoid biner didefinisikan sebagai berikut.

$$f(x) = \frac{1}{1+e^{-x}}, -\infty \leq x \leq \infty \quad (2.8)$$

$$f'(x) = \frac{(1+f(x))(1-f(x))}{2} \quad (2.9)$$

Algoritma untuk pelatihan jaringan *backpropagation* yang menggunakan satu *hidden layer* dengan fungsi sigmoid biner untuk aktivasi adalah:

a) Tahap 0
   Inialisasi seluruh bobot menggunakan bilangan acak kecil.

b) Tahap 1
   Apabila belum memenuhi kondisi penghentian, lakukanlah tahap 2-8.

c) Tahap 2

Pada tiap pasang data pelatihan, lakukanlah tahap 3-8

d) Tahap 3 (tahap 3-5 termasuk fase 1)

Setiap unit masukan memperoleh sinyal kemudian meneruskan sainyal tersebut menuju unit tersembunyi yang ada diatasnya.

e) Tahap 4

Menghitung seluruh keluaran pada unit tersembunyi zj (j = 1,2,….,p)

$$z_net_j = v_{j0} + \sum_{i=1}^{n} x_i v_{ji} \quad (2.10)$$

$$z_{j=f(z_net_j)} = \frac{1}{1+e^{-y_{netj}}} \quad (2.11)$$

f) Tahap 5

Menghitung seluruh keluaran jaringan pada unit keluaran yk (k= 1,2,…,m).

$$y_net_k = w_{k0} + \sum_{j=i}^{p} z_j w_{kj} \quad (2.12)$$

$$y_k = f(y_{net_k}) = \frac{1}{1+e^{-y_netk}} \quad (2.13)$$

g) Tahap 6 (Tahap 6 dan 7 termasuk fase 2)

Menghitung faktor δ unit keluaran berlandaskan pada kesalahan dari tiap unit keluaran yk (k=1,2,…,m)

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k), \quad t_k = target\ keluaran \quad (2.14)$$

δ_k ialah unit kesalahan yang nantinya digunakan untuk perubahan bobot layar dibawahnya

Menghitung perubahan bobot w_{kj} menggunakan laju pemahaman α.

$$\Delta w_{kj} = \alpha \delta_k z_j, k = 1,2,\dots,m\ ; j = 0,1,\dots,p \quad (2.15)$$

h) Tahap 7

Menghitung faktor δ unit tersembunyi berlandaskan pada kesalahan dalam tiap unit tersembunyi z_j (j= 1,2,…,p)

$$\delta_{net_j} = \sum_{k=1}^{m} \delta_k w_{kj} \quad (2.16)$$

Faktor δ unit tersembunyi.

$$\delta_j = \delta'_{net_j} f'\left(z_{net_j}\right) = \delta_net_j z_j (1 - z_j) \quad (2.17)$$

Menghitung suku perubahan bobot v_{ji}.

$$\Delta v_{ji} = \alpha \delta_j x_i, j = 1,2,\dots,p; i = 1,2,\dots,n \quad (2.18)$$

i) Tahap 8 (fase 3)

Menghitung seluruh perubahan bobot garis yang mengarah ke unit keluaran, yakni:

$$w_{kj}' = w_{kj} + \Delta w_{kj}, (k = 1,2, \dots, m); j = 0,1, \dots, p). \quad (2.19)$$

Perubahan bobot garis yang mengarah ke unit tersembunyi, yaitu:

$$w_{kj}' = w_{kj} + \Delta v_{ji}, (j = 1,2, \dots, p; i = 0,1, \dots, n). \quad (2.20)$$

2.2.7 Matlab

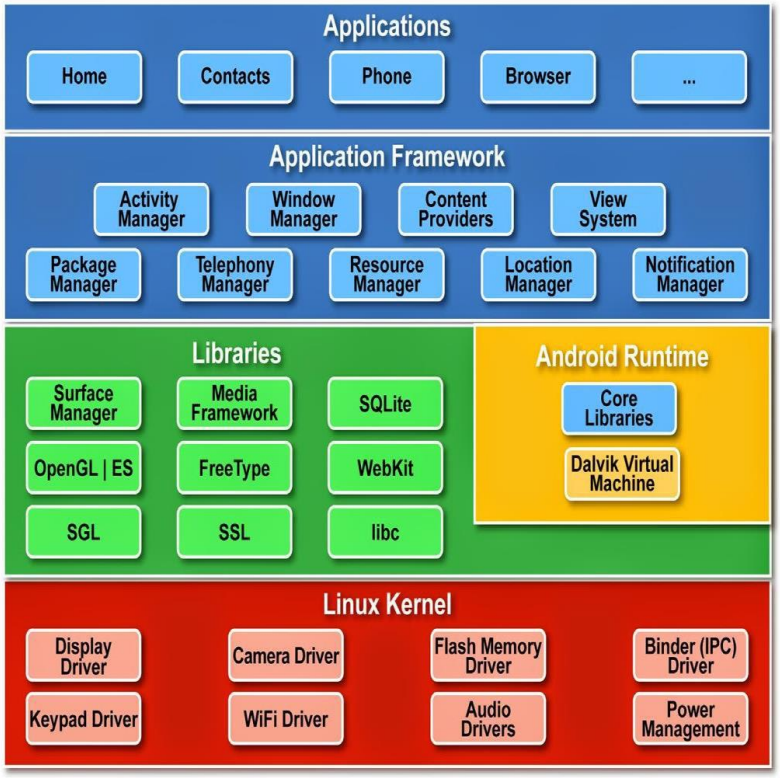
MATLAB (*Matrix Laboratory*) ialah sebuah bahasa pemrograman matematika lanjut yang didesain atas pemikiran dengan mengaplikasikan sifat dan bentuk matriks, matlab biasa digunakan sebagai analisis data dan komputasi numerik (Cahyono, 2013). The Mathworks.Inc mengembangkan sebuah merk *software* bernama MATLAB yang memungkingkan untuk manipulasi matriks, implementasi algoritma, pembagian fungsi dan data berdasarkan plotnya, serta membuat antarmuka menggunakan program dalam bahasa lain. Pada MATLAB juga terdapat *toolbox* yang mampu mengakses kemampuan aljabar komputer. Selain itu, terdapat Simulink yang dapat memberikan tambahan berupa simulasi grafis multiranah dan juga desain berdasarkan model untuk sistem tertanam dan dinamis (Ariyanto & Caesarendra, 2011). MATLAB bersifat *extensible*, artinya *user* dapat menambahkan fungsi baru terhadap *library* apabila fungsi-fungsi built-in yang sudah ada tidak bisa digunakan untuk tugas khusus.

Menurut (Ariyanto & Caesarendra, 2011) pada bukunya yang berjudul Panduan Belajar Mandiri MATLAB, menjelaskan bahwa karakteristik dari MATLAB adalah sebagai berikut:

a. Bahasa pemrograman pada matlab berlandaskan pada baris dan kolom (matriks).
b. Terdapat berbagai *toolbox* yang diperuntukkan pada aplikasi-aplikasi khusus seperti halnya Simulink, *Neural Network*, *State Flow*, *Data* dan *Image Acquisition Toolbox*, *Fuzzy Logic Toolbox*, *Communication* dan *Signal Processing Blockset*, dan lain-lain.
c. *Array* tidak perlu dideklarasikan dahulu saat menulis kode program.
d. Lebih cepat dalam mengembangkan program daripada jika menggunakan pemrograman tradisional layaknya FORTRAN dan C.

2.2.8 Android

Android ialah sistem operasi berbasis linux dan umumnya digunakan pada perangkat *mobile*. Android memfasilitasi para pengembang sehingga mereka dapat membuat sendiri aplikasi menjadi berbagai macam piranti bergerak (Juansyah, 2015). Komponen pada Android di desain layaknya sebuah *stack*. Ulasan tentang arsitektur Android dapat dilihat dalam gambar 2.15 berikut:



Gambar 2.15 Arsitektur Android

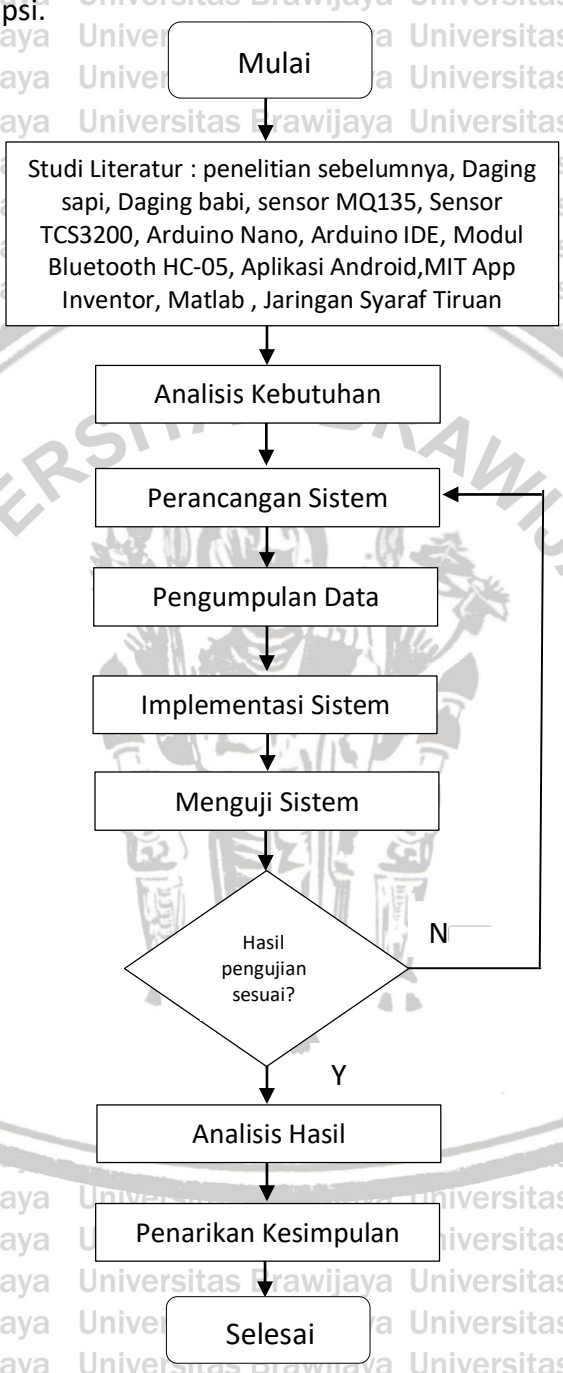
Sumber: (Lamidi, 2018)

2.2.9 MIT App Inventor

MIT app Inventor merupakan sebuah lingkungan pemrograman visual yang intuitif untuk membangun aplikasi smartphone dan tablet yang mungkin saja dapat dibuat oleh semua orang bahkan untuk anak-anak sekalipun. Karena mereka dapat menghasilkan sebuah aplikasi sederhana dan dapat dijalankan dengan waktu yang singkat walaupun mereka baru mengenal MIT App Inventor. Kelebihan lainnya yaitu MIT App Inventor merupakan sebuah pemrograman berbasis blok sehingga dapat mempersingkat waku pembuatan aplikasi dibandingkan dengan lingkungan pemrograman yang lain. Selain itu MIT App Inventor juga memfasilitasi pembuatan aplikasi yang kompleks dan berdampak tinggi karena proyek MIT App Inventor berupaya memberdayakan masyarakat, terutama kaum muda agar mengembangkan perangkat lunak sehingga mereka bisa beralih dari konsumsi teknologi menjadi pencipta teknologi. (Inventor, 2012)

BAB 3 METODOLOGI

Dalam bab Metodologi ini akan dijelaskan perihal bagaimana cara untuk dapat mengimplementasikan sistem serta langkah-langkah yang dilakukan untuk penyusunan skripsi.



Gambar 3.1 Diagram Alir Metode Penelitian

3.1 Studi Literatur

Studi literatur dilakukan untuk mendukung penelitian dan pemahaman terhadap dasar teori sebagai bahan studi. Beberapa studi literatur yang digunakan antara lain:

1. Daging Sapi

   Studi literatur dilaksanakan untuk belajar dan mencari tahu tentang karakteristik serta ciri-ciri dari daging sapi.

2. Daging Babi

   Studi literatur dilakukan untuk mempelajari tentang karakteristik serta ciri-ciri dari daging babi.

3. Sensor Gas MQ135

   Studi literatur dilakukan untuk mempelajari tentang spesifikasi, konfigurasi pin, penggunaan, serta cara kerja dari sensor MQ135 sehingga dapat mendeteksi jenis daging berdasarkan gas atau kadar amonia.

4. Sensor Warna TCS3200

   Studi literatur dilakukan untuk mempelajari tentang spesifikasi, konfigurasi pin, penggunaan, serta cara kerja dari sensor MQ135 sehingga dapat mendeteksi jenis daging berdasarkan warna.

5. Mikrokontroler Arduino Nano Atmega328P

   Studi literatur dilakukan untuk mempelajari tentang spesifikasi dari Arduino Nano, konfigurasi pin, serta bagaimana cara Arduino Nano memperoleh dan mengolah data dari sensor dan meneruskannya sehingga menjadi *output* yang diinginkan.

6. Arduino IDE

   Studi literatur dilakukan untuk mempelajari tentang pengertian, bahasa yang digunakan, serta pengimplementasian Arduino IDE sebagai suatu software untuk menuliskan kode program.

7. Jaringan Syaraf Tiruan

   Studi literatur dilakukan untuk mempelajari tentang pengertian, cara kerja Jaringan Syaraf Tiruan dan cara pengimplementasian dari Jaringan Syaraf Tiruan pada Arduino Nano sebagai metode untuk mendeteksi jenis daging.

8. Matlab

   Studi literatur dilakukan untuk mempelajari tentang cara menghitung data latih sebagai acuan untuk sistem dengan memanfaatkan *library Neural Network Tool* yang terdapat dalam *Matlab* menggunakan perhitungan *weight.*

9. Aplikasi Android

Mempelajari tentang bagaimana cara merancang serta mengimplementasi aplikasi android untuk menampilkan *output* dari sistem agar dapat digunakan sesuai dengan apa yang direncanakan.

10. Modul Bluetooth HC-05

    Mempelajari mengenai spesifikasi dari sensor serta tahapan untuk melakukan konfigurasi pin sensor sehingga sensor dapat digunakan untuk menghubungkan antara pemroses dan *output* sistem. Dalam sistem ini Arduino Nano sebagai pemroses data sedangkan hasilnya ditampilkan pada aplikasi android.

3.2 Analisis Kebutuhan

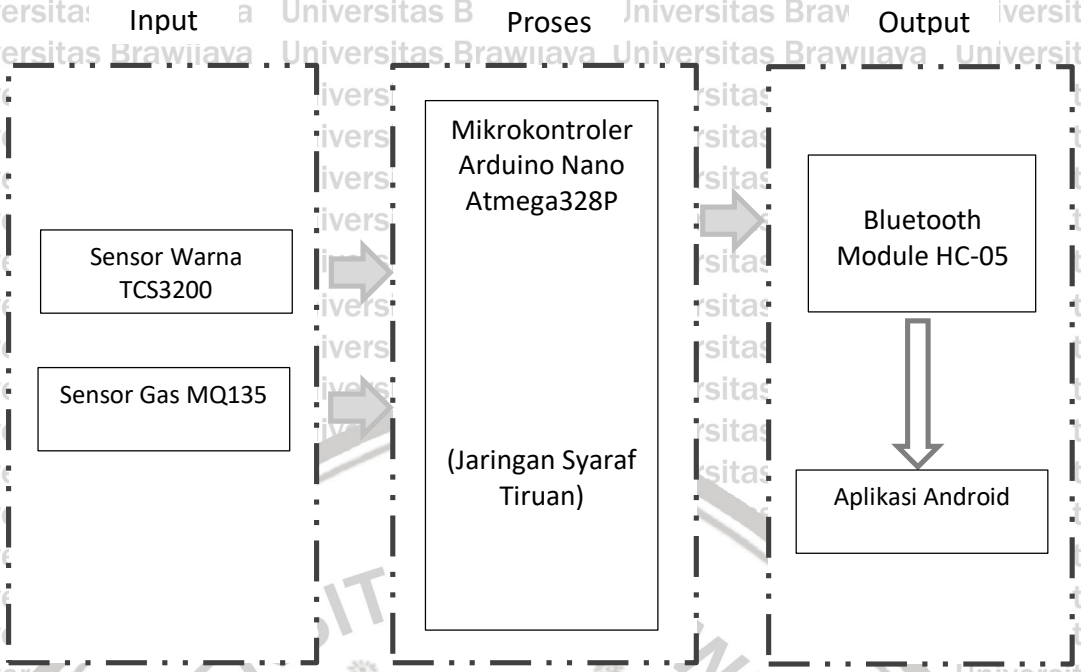
Analisis kebutuhan akan menerangkan semua kebutuhan yang diperlukan oleh sistem serta kenapa sistem membutuhkan itu. Dalam bab ini akan diterangkan juga mengenai kebutuhan fungsional dan non-fungsional dari sistem serta kebutuhan perangkat keras dan kebutuhan perangkat lunak sistem. Adapun perbedaan antar setiap kebutuhan akan diterangkan pada sub-bab 4.

3.3 Perancangan Sistem

Melakukan tahap perancangan dengan tujuan agar sistem dapat dirancang secara terstruktur. Ada dua bagian dalam tahap ini yakni perancangan perangkat keras dan perancangan perangkat lunak.

3.3.1 Perancangan Perangkat Keras

Perancangan Perangkat Keras digambarkan dalam bentuk blok diagram sistem. Terdapat tiga bagian utama dalam blok diagram ini. Bagian pertama tentunya *Input,* karena sistem ini membutuhkan sebuah *input* agar dapat berfungsi sebagaimana mestinya. Pada *Input*, digunakan 2 buah sensor yaitu sensor warna TCS3200 dan sensor gas MQ135. Kedua sensor ini nantinya akan digunakan sebagai pendeteksi daging. Bagian kedua yakni Proses, karena sistem harus memproses data yang diperoleh dari *input*. Untuk memproses data dari sensor, sistem menggunakan mikrokontrolerArduino Nano. Sedangkan bagian terakhir yakni *Output,* dimana akan digunakan untuk menampilkan hasil pemrosesan yang telah dilakukan oleh sistem. *Output* dari sistem ini berupa hasil klasifikasi dari sistem yang ditampilkan dalam Ponsel Android agar memudahkan masyarakat atau konsumen dalam menggunakannya.



Gambar 3.2 Blok Diagram Sistem

3.3.2 Perancangan Perangkat Lunak

Tahap ini berisi tentang bagaimana cara supaya sistem dapat berjalan dan berfungsi sesuai perencanaan. Perancangan perangkat lunak terdiri dari pembuatan program dan logika yang bertujuan agar sistem dapat dijalankan sesuai dengan rencana utama.

3.4 Pengumpulan Data

Tahap pengumpulan data dilakukan agar dijadikan sebagai acuan atau sampel. Pengambilan sampel ini dilakukan pada daging yang masih segar. Sebelum melakukan proses pengambilan sampel maka harus dilakukan proses perancangan sistem terlebih dahulu, karena sampel dapat diperoleh melalui sistem tersebut.

3.5 Implementasi Sistem

Ketika semua proses perancangan sistem sudah dilakukan dan terpenuhi, maka langkah selanjutnya yang harus diambil adalah Implementasi sistem karena kedua proses tersebut saling berkaitan.

3.6 Pengujian dan Analisis

Setelah menyelesaikan semua tahapan di atas, maka dapat dilanjutkan pada tahap pengujian dan analisis. Tahap ini dilakukan dengan cara menguji sistem tersebut menggunakan parameter yang telah ditentukan agar dapat mengetahui seberapa besar presentase tingkat keberhasilan dari sistem tersebut apabila diimplementasikan pada kasus ini. Adapun beberapa parameter pengujian untuk penelitian ini yakni:

1. Pengujian sensor warna tipe TCS3200 apakah mampu berfungsi selaras dengan spesifikasinya.
2. Pengujian sensor gas MQ135 apakah mampu berfungsi selaras dengan spesifikasinya
3. Pengujian modul *Bluetooth* HC-05 apakah mampu berfungsi selaras dengan spesifikasinya
4. Pengujian tampilan pada aplikasi ponsel Android untuk *output* sistem.
5. Pengujian tingkat akurasi dari hasil klasifikasi oleh Jaringan Syaraf Tiruan.
6. Pengujian waktu komputasi dari sistem.

3.7 Kesimpulan

Tahap ini dilakukan ketika semua proses pada bab metodologi mulai dari tahap perancangan hingga tahap pengujian telah diselesaikan. Isi dari semua tahapan yang direncakan pada bab ini disimpulkan menjadi satu dalam tahap kesimpulan. Setelah kesimpulan sudah didapatkan, maka bagian selanjutnya yan dimasukkan adalah saran yang berisi masukan-masukan untuk penelitian selanjutnya dengan kasus yang serupa agar dapat memperbaiki kekurangan dari sistemnya dan menyempurnakan penelitian ini.

BAB 4 ANALISIS KEBUTUHAN

Bab ini menerangkan perihal syarat minimum untuk dapat melakukan perancangan sampai implementasi sistem hingga sistem bisa dijalankan dan berfungsi seperti yang diharapkan.

4.1 Deskripsi Umum Sistem

Secara umum fungsi dari pembuatan sistem ini yaitu sebagai alat pembeda jenis daging antara daging sapi dan babi menggunakan sensor warna tipe TCS3200 dan sensor gas tipe MQ135. Kedua sensor tersebut memiliki fungsi untuk mendapatkan data yang diperlukan oleh sistem kemudian mengirimkan data tersebut pada mikrokontroler untuk diproses. Pada sistem ini memanfaatkan algoritma *backpropagation* pada jaringan syaraf tiruan sebagai pengambil keputusan. Layar ponsel Android akan digunakan sebagai media penampil hasil dari pengolahan oleh sistem melalui sebuah aplikasi.

4.2 Kebutuhan Sistem

Kebutuhan sistem menjelaskan tentang kebutuhan fungsional, kebutuhan non fungsional, kebutuhan perangkat keras, serta kebutuhan perangkat lunak yang akan digunakan pada sistem. Dengan itu diharapkan bisa memudahkan penulis ketika merancang serta mengimplementasi sistem.

4.2.1 Kebutuhan Fungsional

Sistem mempunyai kebutuhan fungsional yang wajib terpenuhi untuk pengimplementasian sistem ini, diantaranya:

1. Sensor warna mampu melakukan pembacaan nilai RGB pada daging.

   Sensor warna tipe TCS3200 dapat melakukan pembacaan nilai frekuensi dari kombinasi fotodioda yang terdapat pada sensor berupa nilai RGB ketika mendeteksi daging. Sehingga dapat memudahkan pemroses dalam melakukan pengolahan data.

2. Sensor gas mampu melakukan pembacaan nilai kadar amonia pada daging.

   Sensor gas tipe MQ135 harus dapat melakukan pembacaan kadar amonia pada daging yang dideteksi dengan satuan PPM.

3. Sistem mengaplikasikan metode Jaringan Syaraf Tiruan sebagai pengambil keputusan.

   Dalam pengambilan keputusan, sistem memanfaatkan metode jaringan syaraf tiruan untuk mengklasifikasikan jenis daging berlandaskan pada *input* yang berasal dari hasil pembacaan kedua sensor serta melalui data latih yang telah ditambahkan dalam pemrograman JST.

4. Hasil pengolahan dari sistem dapat ditampilkan melalui Aplikasi Android.

Sistem harus dapat menampilkan *output* berupa hasil klasifikasi jenis daging pada Aplikasi Android.

4.2.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional akan menerangkan mengenai batasan-batasan dalam melakukan perancangan sistem. Adapun beberapa kebutuhan non-fungsional tersebut ialah:

4.2.2.1 Kebutuhan Perangkat Keras

Penelitian ini memerlukan beberapa perangkat keras sebagai penunjang dan menjadi bagian saat pelaksanaan penelitian, diantaranya:

1. Mikrokontroler Arduino Nano Atmega328P
   Mikrokontroler Arduino Nano selaku kontroler utama yang bertugas sebagai penerima serta pemroses data dari sensor. Arduino Nano digunakan karena kebutuhan dari sistem sudah terpenuhi dan juga ukurannya yang lebih kecil serta kelebihan-kelebihan lain yang dimilikinya, diantaranya:
   a) *Open source*, yaitu tersedia secara gratis sehingga dapat diakses oleh siapa saja dan mudah untuk dimodifikasi.
   b) Adanya *library* gratis, karena pada sistem ini akan melakukan proses perhitungan matematika sehingga memerlukan *library* "math.h" dan *library* "TimerOne.h" yang berguna sebagai *interrupt* pada saat sensor warna melakukan pembacaan.
   c) Jumlah pin, karena pada sistem ini hanya membutuhkan 1 pin analog untuk *output* sensor gas MQ135, 1 pin TX untuk *transfer* data menggunakan modul *Bluetooth* HC-05, serta 6 pin digital lainnya untuk sensor warna TCS3200 dan *push button.*

   Adapun spesifikasi Arduino Nano Atmega328 dan deskripsi dari setiap pin terdapat dalam Tabel 4.1:

Tabel 4.1 Spesifikasi Arduino Nano Atmega328P

Mikrontroler	Atmega328P
Tegangan Operasi	5V
Tegangan *Input*	7-9V (direkomendasikan)
Pin Analog *Input*	8
Pin I/O Digital	22 (terdapat 6 pin untuk *output* PWM)
Arus DC setiap Pin I/O	40 mA(I/O Pin)
Konsumsi Daya	19 mA
Flash Memory	32 KB, 2KB digunakan oleh *boatloader*
SRAM	2 KB

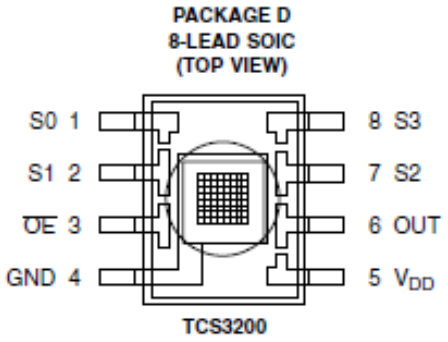
EEPROM	1 KB
Kecepatan *Clock*	16 MHz
Panjang	45 mm
Lebar	18 mm
Berat	7 g
Port USB	Mini – B USB

Sumber: (Arduino, 2019)

2. Sensor warna TCS3200

Pada sistem ini sensor warna difungsikan sebagai pendeteksi warna pada daging untuk parameter mencari perbedaan dari daging sapi dan babi. Selain itu sensor ini digunakan karena output yang dihasilkan berupa data digital sehingga memudahkan mikrokontroler dalam membaca data. Data diperoleh ketika sensor mendeteksi warna daging sehingga menghasilkan nilai RGB.

Adapun spesifikasi dari sensor warna TCS3200 dan deskripsi masing-masing pin terdapat pada Gambar 4.1 dan Tabel 4.2 berikut:



Gambar 4.1 Deskripsi Pin Sensor Warna TCS3200

Sumber: (Inc, 2009)

Tabel 4.2 Spesifikasi Sensor Warna TCS3200

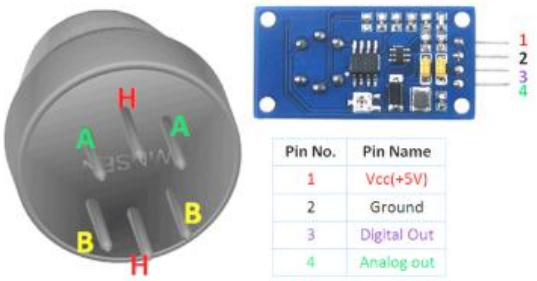
Terminal		I/O	Deskripsi
Nama	No.		
GND	4		Catu daya untuk *ground*.
OE	3	I	Untuk mengaktifkan f_o (*active low*)
OUT	6	O	Frekuensi *Output* (f_o)
S0, S1	1, 2	I	*Input* untuk pemilihan skala frekuensi *output*
S2, S3	7, 8	I	*Input* untuk pemilihan jenis fotodioda
V_{DD}	5		Tegangan masukan

Sumber: (Inc, 2009)

3. Sensor gas MQ135

Sensor ini digunakan sebagai pendeteksi kandungan kadar amonia pada daging, kadar amonia yang terdeteksi akan digunakan sebagai parameter untuk proses klasifikasi sehingga bisa dibedakan antara daging sapi dan babi. Sensor ini digunakan karena memiliki keunggulan sensitif terhadap amonia. Selain itu harga dari sensor ini tergolong rendah sehingga dapat menghemat biaya dalam pembuatan sistem. Data yang dihasilkan berupa nilai kadar amonia dari daging yang dideteksi dalam satuan PPM.

Adapun spesifikasi dari sensor warna TCS3200 dan deskripsi masing-masing pin terdapat pada Gambar 4.2 dan Tabel 4.3.



Gambar 4.2 Deskripsi Pin Sensor MQ135

Sumber: (components101.com)

Tabel 4.3 Spesifikasi Sensor Gas MQ135

No. Pin	Nama Pin	Deskripsi
1	Vcc	Untuk memberikan tegangan pada sensor, umumnya menggunakan +5v.
2	GND	Untuk menghubungkan modul pada *ground* sistem
3	Digital Out	Untuk memperoleh *output* yang bernilai digital, dengan memenentukan terlebih dahulu ambang batasnya menggunakan potensiometer
4	Analog Out	Untuk menghasilkan tegangan analog 0-5v berdasarkan intensitas gas

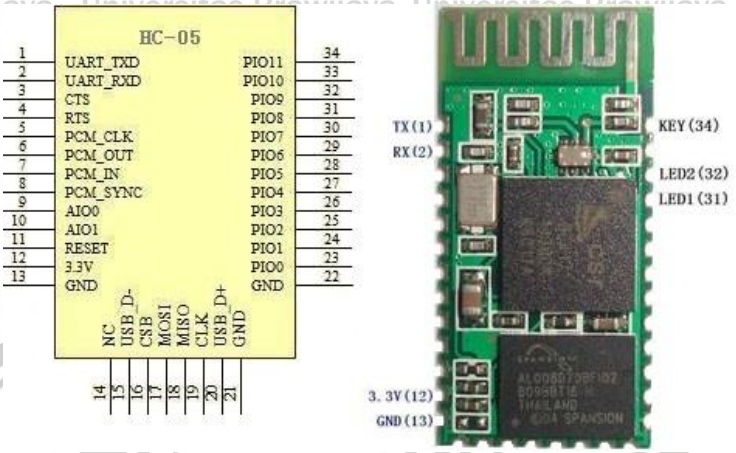
Sumber: (components101.com)

4. Modul Bluetooth HC-05

Modul ini digunakan supaya mikrokontroler Arduino bisa mengirimkan hasil pengolahan yang telah dilakukan kepada ponsel Android. Modul Bluetooth berseri ganjil ini digunakan karena dapat digunakan sebagai

master (pengirim) atau *slave* (penerima). Berbeda dengan modul Bluetooth yang berseri genap yang hanya bisa digunakan sebagai *slave*.

Spesifikasi dan deskripsi masing-masing pin Modul Bluetooth HC-05 terdapat pada Gambar 4.3.



Gambar 4.3 Deskripsi Pin Bluetooth HC-05

Sumber: (os.embed.com)

5. Kabel *Jumper*

   Kabel *Jumper* berperan sebagai penghubung setiap komponen yang terdapat dalam sistem. Keunggulan dari komponen ini yaitu *portable,* mudah untuk dilepas dan dipasang kembali.

6. *Push Button*

   Push Button berguna dalam pemberian *trigger* pada sistem sehingga mikrokontroler melakukan pengambilan data untuk diolah menggunakan Jaringan Syaraf Tiruan.

7. Laptop

   Laptop sebagai media pembuat program untuk mikrokontroler, untuk pembuatan program pelatihan data maka program akan dibuat pada Matlab. Sedangkan untuk program agar sensor dapat membaca data dan untuk proses klasifikasi yang memanfaatkan JST akan dibuat pada Arduino IDE. Laptop juga bertindak sebagai sumber daya yang diperlukan oleh sistem.

8. Ponsel Android

   Ponsel Android sebagai penampil hasil pengolahan yang dilakukan oleh sistem melalui Aplikasi Android. Ponsel Android digunakan karena *compatible* dengan aplikasi android dan banyaknya masyarakat yang menggunakannya saat ini.

4.2.2.2 Kebutuhan Perangkat Lunak

Penelitian ini membutuhkan perangkat lunak untuk menunjang penelitian, diataranya:

1. Arduino IDE
   Merupakan sarana dalam membuat kode program untuk mikrokontroler Arduino Nano. Alasan penggunaan Arduino IDE karena *Open Source,* Bahasa pemrograman yang relatif mudah, dan memudahkan jika ingin dilakukan pengembangan.
2. MIT App Inventor
   Digunakan untuk membuat aplikasi yang berfungsi sebagai sarana menampilkan hasil pengolahan yang dilakukan oleh mikrokontroler.
3. Matlab R2017b
   Merupakan sebuah aplikasi yang berfungsi untuk menghitung data latih yang nantinya akan berguna untuk sistem dengan memanfaatkan JST *Backpropagation*. Penggunaan matlab dalam penelitian ini karena menyediakan *toolbox Neural Network* yang nantinya diterapkan dalam penelitian ini.

4.3 Batasan Desain Sistem

Supaya penerapan dari sistem berjalan sesuai harapan, maka diperlukan beberapa batasan dalam melakukan implementasi desain sistem, diantaranya:

1. Arduino Nano sebagai mikrokontroler dan digunakan untuk pemroses data.
2. Sensor yang dipakai yaitu sensor warna TCS3200 dan sensor gas MQ135.
3. Sistem hanya dapat menampilkan *output* hasil pendeteksian oleh sistem ketika ada *trigger* dari pengguna.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

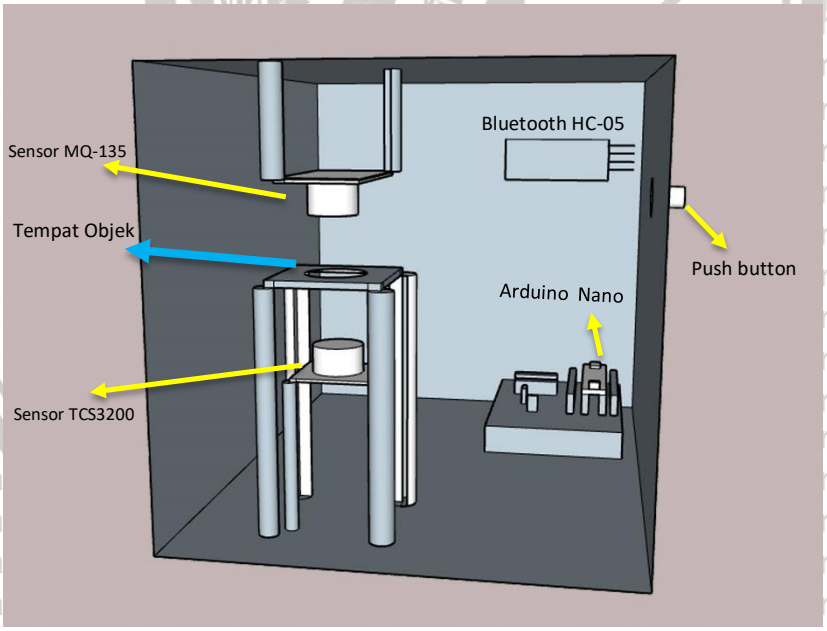
Bab ini menerangkan perihal bagaimana cara merancang sistem dan akan dilanjutkan pada tahapan implementasi sistem, baik implementasi perangkat keras maupun perangkat lunak.

5.1 Perancangan Sistem

Pada tahap ini akan dijabarkan terkait penyusunan sistem diawali dengan merancang *prototype* alat hingga merancang perangkat keras dan juga perangkat lunak sistem.

5.1.1 Perancangan Protoype Alat

Dalam melakukan perancangan *prototype* alat untuk Sistem Pembeda Jenis Daging Sapi dan Daging Babi ini perlu diperhatikan ukuran alat yang akan dibuat dan peletakan masing-masing komponen. Sistem ini akan dirancang dalam *box* dengan ukuran panjang 17cm, lebar 17cm, dan tinggi 17cm. Digunakannya ukuran tersebut dimaksudkan agar sistem dapat bersifat *portable*, sehingga memudahkan dalam pemanfaatan sistem ini nantinya. Selain itu agar tidak terlalu banyak *space* kosong yang terdapat pada alat. Untuk menggambarkan rancangan pembuatan sistem ini, penulis menggunakan aplikasi *SketchUp Pro 2018*. Gambaran mengenai *prototype* alat ditujukan pada gambar 5.1.



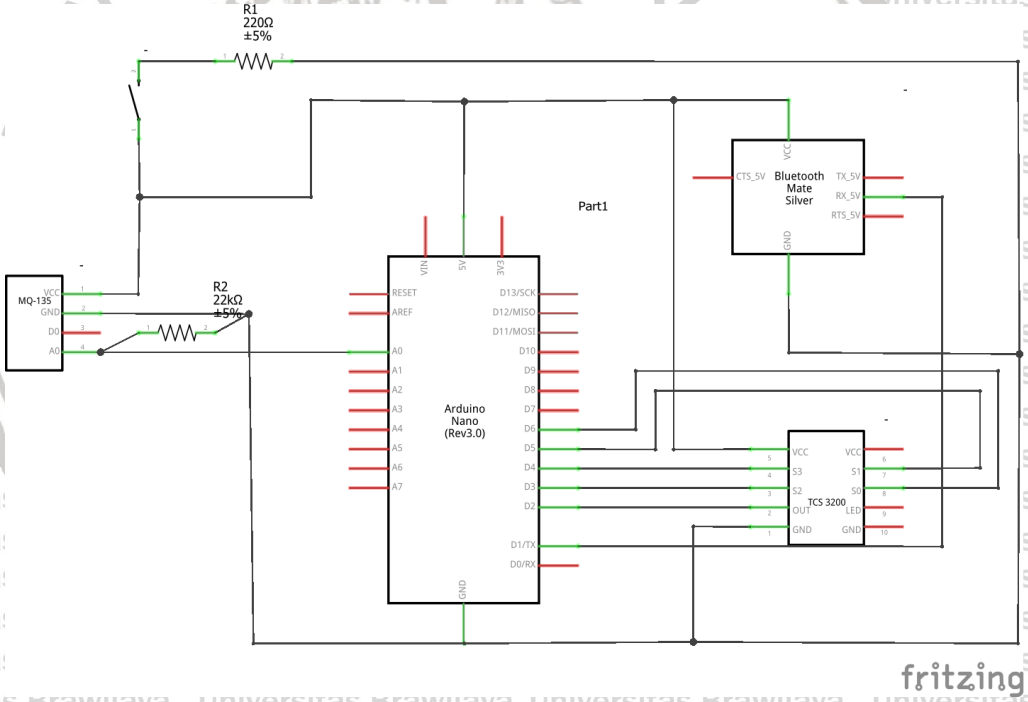
Gambar 5.1 Desain Prototype Alat

Objek (daging sapi dan daging babi) yang akan dideteksi diletakkan tepat diatas sensor warna TCS3200 dan dibawah dari sensor gas MQ-135 agar sensor dapat bekerja secara maksimal dan hasil pembacaan sensor lebih akurat. Bagian atas *box* diberi pintu agar mempermudah saat memasukkan objek yang akan

dideteksi. Sistem ini juga dilengkapi dengan modul Bluetooth HC-05 sebagai pengirim hasil klasifikasi oleh sistem menuju aplikasi Android. Sementara di bagian luar *box* ditambahkan *push button* untuk memberikan *trigger* pada sensor.

5.1.2 Perancangan Perangkat Keras

Perancangan perangkat keras didasarkan pada analisis kebutuhan yang sebelumnya sudah dijelaskan, sehingga diharapkan bisa diciptakan sistem yang sesuai. Merujuk pada blok diagram sistem yang telah dijelaskan pada Gambar 3.2 menunjukkan bahwa sistem ini saling terhubung antara *input*, proses, dan *output.* Pada tahap ini menjelaskan mengenai skematik pin-pin yang menghubungkan komponen satu dengan lainnya. Dimana sensor yang digunakan sebagai *input* (sensor warna TCS3200 dan sensor gas MQ135) disambungkan dengan mikrokontroler Arduino Nano sebagai pemroses dan pengolah data dari sensor. Selain itu terdapat juga hubungan antara mikrokontroler dengan modul Bluetooth HC-05 sebagai pengirim hasil klasifikasi agar dapat ditampilkan melalui aplikasi Android. Skematik diagram perancangan perangkat keras sistem ditunjukkan pada Gambar 5.2.



Gambar 5.2 Diagram Skematik Perancangan Perangkat Keras Sistem

Dari diagram skematik tersebut dapat dilihat bahwa terdapat *push button*. Pada push button terdapat dua kaki, salah satu kakinya terhubung langsung dengan Vcc atau pin 5V pada Arduino. Sementara kaki yang lain disambungkan pada pin digital D8 dari Arduino dan resistor yang dihubungkan langsung pada pin GND pada Arduino.

Kemudian koneksi antar pin dari modul Bluetooth HC-05 dan Arduino diterangkan dalam Tabel 5.1. Pada tabel terdapat hanya 3 pin yang saling berhubungan, yaitu Vcc dan GND yang tersambung dengan pin Vcc dan GND pada Arduino. Sementara pin Rx dirangkai dengan pin Tx pada Arduino, karena dalam sistem ini Bluetooth HC-05 hanya berfungsi untuk *receiver* atau penerima data.

Tabel 5.1 Keterangan koneksi pin modul Bluetooth HC-05 dengan Arduino Nano

Pin modul Bluetooth HC-05	Pin Arduino Nano
GND	GND
Vcc	5v
RX	TX

Tabel 5.2 menjelaskan hubungan antara pin dari sensor TCS3200 dengan Arduino Nano. Sensor TCS3200 mempunyai 2 pin Vcc dan 2 pin GND, salah satu dari pin tersebut disambungkan pada pin 5v dan GND dalam Arduino. Selanjutnya empat pin lainnya yaitu pin S0, S1, S2, dan S3, dihubungkan pada pin digital Arduino yaitu pin D6, pin D5, pin D3, pin D4. Sementara pin OUT yang bertindak sebagai pin *input* dari sensor disambungkan pin D2 pada Arduino.

Tabel 5.2 Keterangan koneksi pin sensor warna TCS3200 dengan Arduino Nano

Pin Sensor Warna TCS3200	Pin Arduino Nano
GND	GND
Vcc	5v
S0	D6
S1	D5
S2	D3
S3	D4
OUT	D2

Tabel 5.3 menjelaskan hubungan antara pin dari sensor MQ135 dengan Arduino Nano. Sistem ini memanfaatkan modul sensor gas MQ135 yang memiliki 4 pin yaitu A0, D0, Vcc, dan GND. Sensor memakai pin A0 untuk *output* karena nilai gas yang terbaca oleh sensor bernilai analog. Sedangkan pin Vcc dan GND disambungkan dengan pin 5v dan pin GND Arduino Nano.

Tabel 5.3 Keterangan koneksi pin sensor gas MQ135 dengan Arduino Nano

Pin Sensor Gas MQ135	Pin Arduino Nano
GND	GND

Vcc	5v
A0	A0

5.1.3 Perancangan Perangkat Lunak

Perancangan dilakukan untuk menerangkan proses-proses yang berlangsung pada perangkat lunak. Dimulai dari memberikan *input,* membaca data sensor, mengumpulkan data latih, merancang JST, hingga menampilkan hasil pada aplikasi Android.

Gambar 5.3 Diagram alir perancangan keseluruhan sistem

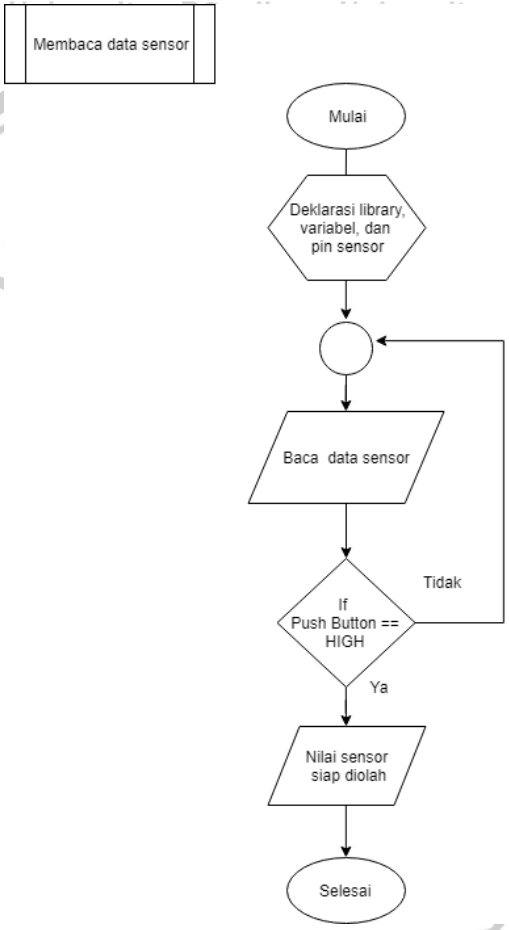
Gambar 5.3 menunjukkan diagram alir keseluruhan sistem. Dari gambar diatas terlihat bahwa agar sistem mampu berjalan seperti yang diharapkan, maka langkah pertama yaitu memberikan *input* pada sistem berupa sampel daging baik daging sapi maupun daging babi. Dengan adanya *input* tersebut maka sistem akan melakukan proses pembacaan data oleh sensor warna TCS3200 dan sensor gas MQ135. Setelah sensor memperoleh data maka nilai-nilai tersebut akan dikumpulkan dan dijadikan data latih untuk dapat merancang pelatihan Jaringan Syaraf Tiruan.Pelatihan dilakukan agar sistem dapat melakukan klasifikasi terhadap perbedaan jenis daging antara daging sapi dan babi memakai algoritma Jaringan Syaraf Tiruan. Langkah terakhir dari sistem yaitu hasil klasifikasi oleh

sistem ditampilkan pada aplikasi Android. Untuk penjelasan lebih detail akan dijelaskan dalam subbab di bawah ini.

5.1.3.1 Memberikan Input

Input yang digunakan pada penelitian ini yaitu sampel daging, baik daging sapi maupun daging babi. Ukuran dari daging untuk sampel pada penelitian ini yaitu berbentuk segi empat dengan ukuran kurang lebih 4cm - 5cm dengan ketebalan kurang lebih 1cm – 1,5cm.

5.1.3.2 Perancangan Pengambilan Data Sensor



Gambar 5.4 Diagram alir perancangan pengambilan data sensor

Gambar 5.4 menunjukkan proses perancangan perangkat lunak yang bertujuan untuk menentukan hasil pembacaan yang dilakukan oleh sensor. Hasil tersebut kemudian diolah sehingga dapat dilanjutkan pada proses klasifikasi memanfaatkan metode JST, hal ini disebabkan karena mikrokontroler terus-menerus membaca nilai sensor.

A. Mulai, sistem mulai mengambil dan mengirim data.
B. Deklarasi *library*, *variable,* dan pin sensor.

a) Deklarasi library, mendeklarasikan library yang dipakai yakni *library* math.h dan *library* TimerOne.h dan untuk penjelasan lebih detail dapat dilihat pada bab selanjutnya.

b) Deklarasi variabel, mendeklarasikan variabel R, G, B untuk pembacaan sensor warna yang masing-masing bernilai awal 0 dan variabel array untuk menyimpan nilai hasil pembacaan sensor. Selanjutnya variabel resistor untuk sensor gas yang digunakan (RL) sebesar 22KΩ, dan variabel lainnya (m, b, Ro) untuk kalibrasi sensor gas. Serta variabel *button* yang digunakan untuk *push button*.

c) Deklarasi pin sensor yang dipakai, sensor pertama yang dipakai ialah sensor warna TCS3200 dengan 4 pinnya (S0 – S3) yang dihubungkan pada pin digital pada mikrokontroler (pin D3-D6). Kemudian pin analog pada sensor gas yang disambungkan pada pin A0 dalam mikrokontroler.

C. Baca data sensor, dalam sistem ini menggunakan sensor gas MQ135 dan sensor warna TCS3200 untuk membaca data.

a) Sensor gas membaca kandungan amonia pada daging yang diuji dan menampilkan hasilnya pada sistem dengan satuan ppm. Untuk mendapatkan hasil pembacaan sensor gas yang akurat harus dilakukan kalibrasi terlebih dahulu terhadap sensor. Untuk melakukan kalibrasi langkah-langkah dapat dilihat pada bab dasar teori.Nilai yang dihasilkan sesuai dengan besarnya kandungan amonia pada daging, semakin besar kandungan amonia pada daging maka nilai ppm juga akan semakin besar.

b) Sensor warna membaca warna pada daging yang diuji dan menampilkan hasilnya dengan nilai R, G, B. Sensor warna juga harus dilakukan kalibrasi agar hasil pembacaannya lebih akurat, selain itu juga dilakukan proses filter oleh sensor agar dapat membaca warna objek dan menghasilkan nilai dalam bentuk R, G, B. nilai minimum RGB (0, 0, 0) sementara nilai maksimumnya yaitu (255, 255, 255). Penjelasan untuk proses tersebut dapat dilihat pada subbab implementasi sistem.

D. *Push button*, saat mikrokontroler sudah mendapatkan nilai dari masing-masing sensor maka dilakukan proses pengecekan *push button* apakah terdapat penekanan atau tidak. Karena pada sistem ini menggunakan *push button* sebagai *trigger* untuk menentukan hasil klasifikasi sistem. Ketika *push button* ditekan, maka sistem akan mengambil nilai yang terakhir kali terbaca oleh masing-masing sensor, dalam hal ini nilai RGB oleh sensor warna dan nilai ppm oleh sensor gas.

E. Nilai sensor siap diolah, setelah mendapatkan nilai terakhir dari pembacaan sensor maka data tersebut digunakan oleh sistem sebagai acuan untuk

menentukan klasifikasi jenis daging yang diuji menggunakan perhitungan jaringan syaraf tiruan.

F. Selesai, sistem telah selesai melakukan proses pembacaan dan pengiriman data.

5.1.3.3 Mengumpulkan Data Latih

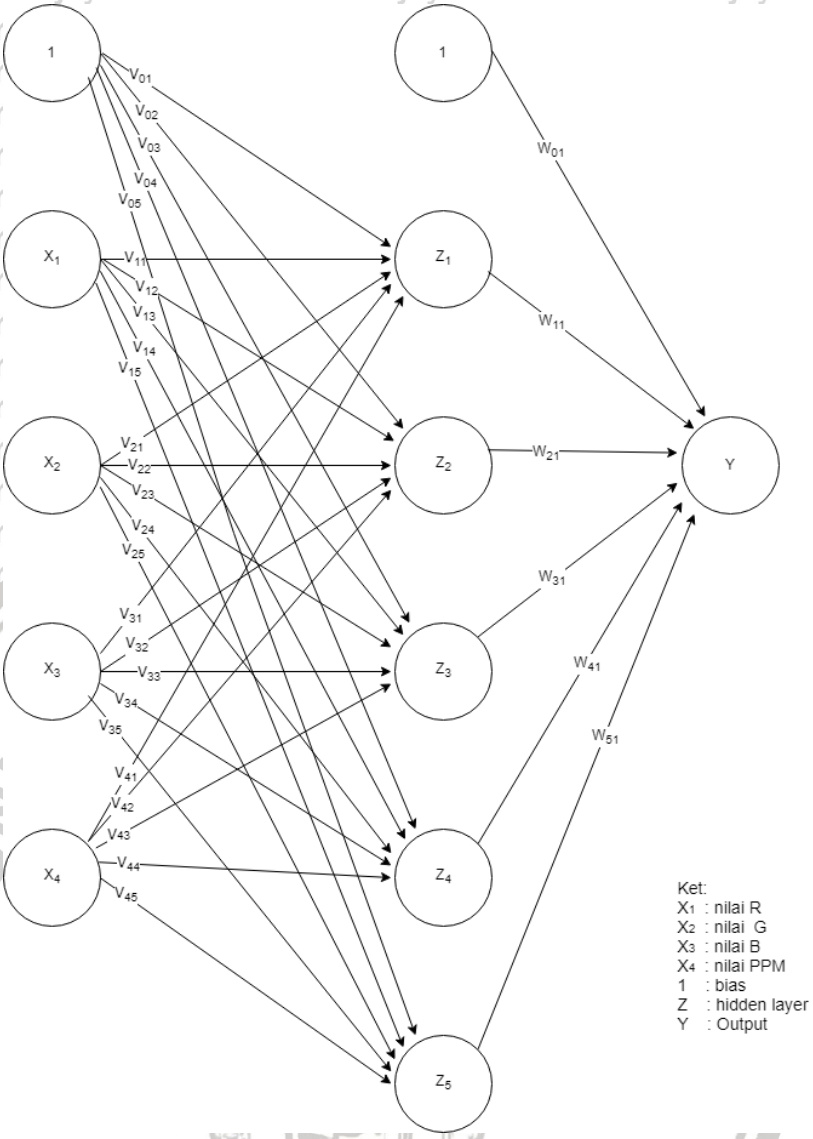
Untuk pengumpulan data latih pada penelitian ini, peneliti membutuhkan percobaan sebanyak 60 sampel daging segar untuk diambil nilai R, G, B, dan nilai PPM gas amonianya. Daging segar yang digunakan oleh peneliti disini adalah daging yang dibeli langsung dari rumah potong hewan pada masing-masing jenis daging, sehingga dapat dipastikan bahwa daging dalam kondisi segar karena berasal dari hewan yang disembelih pada hari itu juga. Data latih yang digunakan terdiri dari 30 sampel daging sapi dan 30 sampel daging babi. Hasil selengkapnya seperti yang terlihat pada lampiran A.

5.1.3.4 Perancangan Pelatihan Jaringan Syaraf Tiruan

Untuk merancang pelatihan JST, maka langkah pertama yakni harus membuat terlebih dahulu desain arsitektur dari Jaringan Syaraf Tiruan. Pembuatan desain arsitektur bertujuan agar dapat mengetahui berapa jumlah *input*, *hidden layer*, bias, serta *output* dari JST. Setelah itu dapat dilanjutkan dengan melakukan pelatihan pada Jaringan Syaraf Tiruan.

1. Arsitektur Jaringan Syaraf Tiruan

Pada Gambar 5.5 dapat dilihat bahwa penelitian menggunakan 4 buah parameter sebagai *input* untuk dapat melakukan klasifikasi terhadap perbedaan jenis daging sapi dan daging babi. 4 parameter tersebut berasal dari *output* yang dihasilkan oleh kedua sensor. *Output* sensor warna terbagi dalam 3 parameter yakni nilai R, G, dan B. Sedangkan *output* sensor gas berupa nilai PPM.

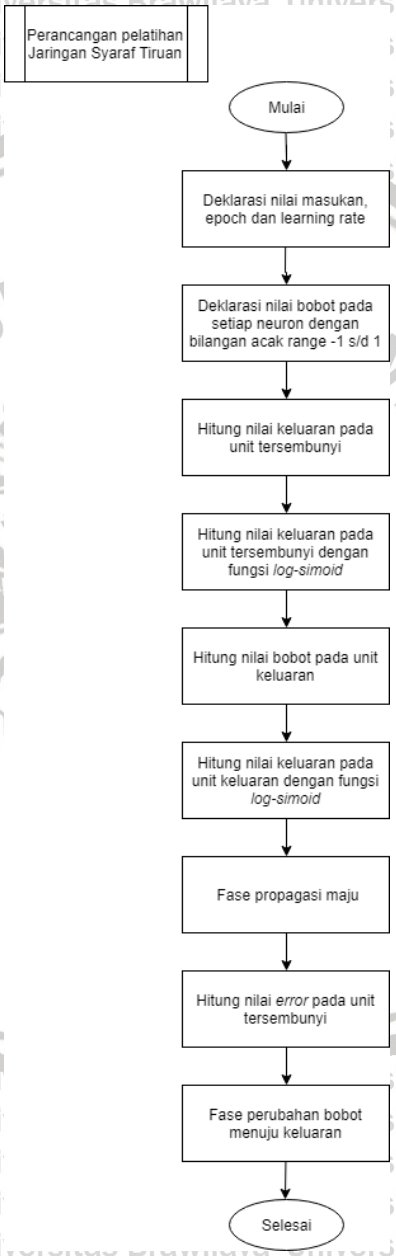


Gambar 5.5 Arsitektur perancangan JST

Sistem ini menggunakan 5 unit *neuron* sebagai *input layer* yang terdiri dari 4 buah parameter *input* (X_1, X_2, X_3, dan X_4) dan 1 neuron tambahan sebagai bias, sehingga diharapkan sistem mampu mengetahui perbedaan antara kedua jenis daging tersebut. Karena dalam sistem ini klasifikasi dibedakan menjadi 2 kelas (daging sapi dan daging babi) maka menggunakan 1 neuron sebagai *output layer* (Y). Sedangkan untuk jumlah *hidden layer* (Z) harus memperhatikan jumlah *input layer* yang digunakan karena akan berpengaruh terhadap performansi serta jumlah memori yang dibutuhkan oleh Arduino dalam melakukan proses prediksi, dimana jika unit *neuron* yang dipakai semakin banyak, maka akan semakin besar pula memori dan jumlah *clock* yang dibutuhkan. Sehingga dalam sistem ini menggunakan 6 unit neuron yang terdiri dari 5 buah parameter *hidden layer*(Z_1, Z_2, Z_3, Z_4, Z_5) serta 1 neuron tambahan sebagai bias. Beberapa hal yang dapat memaksimalkan hasil akurasi saat pelatihan data yakni dengan merubah nilai dari *epoch,* laju pembelajaran (*learning rate)*, dan jumlah *neuron* dalam *hidden layer* agar diperoleh nilai kesalahan yang paling kecil.

2. Pelatihan Jaringan Syaraf Tiruan

Pemberian nilai *epoch* dan *learning rate* saat pelatihan data akan berpengaruh pada hasil akurasi prediksi oleh jaringan syaraf tiruan. Nilai tersebut diperoleh melalui percobaan yang dilakukan beberapa kali sampai didapatkan nilai dengan tingkat kesalahan yang paling kecil dari bobot yang dicari. Gambar 5.6 menjelaskan tentang diagram alir pelatihan *Backpropagation*



Gambar 5.6 Diagram alir pelatihan Backpropagatiom

Diagram alir diatas menunjukkan alur program yang diterapkan oleh sistem, namun tidak pada alat. Karena program ini bekerja secara terpisah pada komputer di dalam Matlab. Fungsi dari program ini yaitu untuk melatih data serta

mendapatkan nilai bobot melalui hasil pelatihan. Untuk mengklasifikasikan antara daging sapi dan daging babi menggunakan JST *Backpropagation,* hal pertama yang perlu dilakukan yaitu melakukan deklarasi nilai yang digunakan sebagai masukan (R, G, B, ppm), *epoch*, dan *learning rate*. Selanjutnya Matlab mendeklarasikan bilangan acak dengan *range* -1 sampai 1 sebagai nilai bobot pada neuron. Kemudian menghitung nilai keluaran pada unit tersembunyi dan menghitungnya memanfaatkan fungsi *log-sigmoid*. Setelah itu menghitung bobot pada unit keluaran dan menghitung nilai keluaran pada unit keluaran menggunakan fungsi *log-sigmoid.* Selanjutnya yaitu fase propagasi maju, setelah itu menghitung nilai *error* dalam unit tersembunyi. Langkah terakhir yaitu fase perubahan bobot yang mengarah ke unit keluaran.

Penjelasan tahapan diatas dapat dilihat pada sebuah contoh perhitungan manual *Backpropagation* untuk mendapatkan bobot pada semua layer JST menggunakan data latih dalam LAMPIRAN A:

a) Melakukan deklarasi nilai masukan (R, G, B, ppm), *epoch*, dan *learning rate*.

$X_1 = 140$, $X_2 = 93$, $X_3 = 84$, $X_4 = 0{,}04$

$t = 0$

$\alpha = 0{,}001$

Nilai masukan didapatkan dari hasil pembacaan sensor yakni X_1 = R, X_2 = G, X_3 = B, X_4 = ppm. Sementara α = *learning rate*, nilai tersebut digunakan karena hasil akurasi yang didapatkan masih kurang dari yang diinginkan ketika menggunakan nilai *default* 0,01 sebagai *learning rate* untuk jaringan *multi layer* seperti ini, sehingga peneliti menggunakan nilai 0,001.

b) Deklarasi bilangan acak dengan *range* -1 sampai 1 sebagai nilai bobot. Dapat dilihat pada Tabel 5-4 dan Tabel 5-5.

Tabel 5.4 Deklarasi bobot layer 1

	Z_1	Z_2	Z_3	Z_4	Z_5
X_1	0,5	0,2	-0,1	-0,3	0,4
X_2	0,1	0,4	0,2	-0,5	0,1
X_3	-0,2	-0,3	0,1	-0,2	0,2
X_4	0,4	0,1	-0,3	0,1	-0,4
1	-0,3	0,2	0,1	0,2	0,3

Tabel 5.5 Deklarasi bobot layer 2

	Y
Z_1	0,2
Z_2	0,3

Z_3	-0,2
Z_4	0,1
Z_5	0,5
1	0,4

c) Sesuai **Persamaan (2.10)** selanjutnya yaitu melakukan perhitungan terhadap semua keluaran pada unit tersembunyi (Zj)

$Z_{netj} = v_{j0} + \sum_{i=1}^{n} xivji$

$Z_{net1} = v_{j0} + \sum_{i=1}^{5} xivji = v_{10} + x_1v_{11} + x_2v_{21} + x_3v_{31} + x_4v_{41}$

= -0,3 + 140.0,5 + 93.0,1 + 84.(0,2) + 0,04.0,4

= 62,216

$Z_{net2} = v_{j0} + \sum_{i=1}^{5} xivji = v_{20} + x_1v_{12} + x_2v_{22} + x_3v_{32} + x_4v_{42}$

= 0,2 + 140.0,2 + 93.0,4 + 84.(-0,3) + 0,04.0,1

= 40,204

$Z_{net3} = v_{j0} + \sum_{i=1}^{5} xivji = v_{30} + x_1v_{13} + x_2v_{23} + x_3v_{33} + x_4v_{43}$

= 0,1 + 140.(-0,1) + 93.0,2 + 84.0,1 + 0,04.(-0,3)

= 13,088

$Z_{net4} = v_{j0} + \sum_{i=1}^{5} xivji = v_{40} + x_1v_{14} + x_2v_{24} + x_3v_{34} + x_4v_{44}$

= 0,2 + 140.(-0,3) + 93.(-0,5) + 84.(-0,2) + 0,04.0,1

= -105,296

$Z_{net5} = v_{j0} + \sum_{i=1}^{5} xivji = v_{50} + x_1v_{15} + x_2v_{25} + x_3v_{35} + x_4v_{45}$

= 0,3 + 140.0,4 + 93.0,1 + 84.0,2 + 0,04 (-0,4)

= 82,384

d) Kemudian sesuai dengan **Persamaan (2.11)** selanjutnya yaitu menghitung semua keluaran pada unit tersembunyi memanfaatkan fungsi *log-sigmoid* (Zj).

$Z_j = f(Z_{netj}) = \frac{1}{1+e^{-znetj}}$

$Z_1 = f(Z_{net1}) = \frac{1}{1+e^{-znet1}} = \frac{1}{1+e^{-62,216}} = 1$

$Z_2 = f(Z_{net2}) = \frac{1}{1+e^{-znet2}} = \frac{1}{1+e^{-40,204}} = 1$

$Z_3 = f(Z_{net3}) = \frac{1}{1+e^{-znet3}} = \frac{1}{1+e^{-13,088}} = 1$

$Z_4 = f(Z_{net4}) = \frac{1}{1+e^{-znet4}} = \frac{1}{1+e^{105,296}} = 5,36$

$Z_5 = f(Z_{net5}) = \frac{1}{1+e^{-znet5}} = \frac{1}{1+e^{-82,384}} = 1$

e) Sesuai dengan **Persamaan (2.12)** langkah selanjutnya yaitu menghitung bobot pada unit keluaran(Y_k)

$Y_{net1} = w_{10} + z_1w_{11} + z_2w_{12} + z_3w_{13} + z_4w_{14} + z_5w_{15}$

$= 0,4 + 1.0,2 + 1.0,3 + 1.(-0,2) + 5,36.0,1 + 1.0,5$

$= 1,736$

f) Berdasarkan **Persamaan (2.13)** selanjutnya yaitu menghitung nilai keluaran dalam unit keluaran memanfaatkan fungsi *log-sigmoid*. (Y_j)

$Y_j = f(Y_{netj}) = \frac{1}{1+e^{-ynetj}}$

$Y_1 = f(Y_{net1}) = \frac{1}{1+e^{-ynet1}} = \frac{1}{1+e^{-1,736}} = 1,176$

g) Langkah selanjutnya yaitu memasuki fase propagasi maju, sesuai dengan **Persamaan (2.14)** dan **(2.15).**

$\delta_1 = (t_1-y_1)\ y_1\ (1-y_1) = (0-1,176)\ 1,176\ (1-1,176) = 0,243$

$\Delta w_{01} = \alpha\delta_1(1) = 0,2.0,243.1 = 0,0486$

$\Delta w_{11} = \alpha\delta_1(Z_1) = 0,2.0,243.1 = 0,0486$

$\Delta w_{21} = \alpha\delta_1(Z_2) = 0,2.0,243.1 = 0,0486$

$\Delta w_{31} = \alpha\delta_1(Z_3) = 0,2.0,243.1 = 0,0486$

$\Delta w_{41} = \alpha\delta_1(Z_4) = 0,2.0,243.5,36 = 0,260$

$\Delta w_{51} = \alpha\delta_1(Z_5) = 0,2.0,243.1 = 0,0486$

h) Sesuai **Persamaan (2.16)** langkah selanjutnya yaitu melakukan perhitungan terhadap nilai *error* pada unit tersembunyi berlandaskan pada kesalahan yang ada dalam tiap unit tersembunyi Zj (j=1,2,3,...p)

$\delta_{net01} = \delta_1 . w_{01} = 0,243.0,0486 = 0,011$

$\delta_{net11} = \delta_1 . w_{11} = 0,243.0,0486 = 0,011$

$\delta_{net21} = \delta_1 . w_{21} = 0,243.0,0486 = 0,011$

$\delta_{net31} = \delta_1 . w_{31} = 0,243.0,0486 = 0,011$

$\delta_{net41} = \delta_1 . w_{41} = 0,243.0,260 = 0,063$

$\delta_{net51} = \delta_1 . w_{51} = 0,243.0,0486 = 0,011$

Sesuai dengan **Persamaan (2.17)** dan **(2.18)** langkah selanjutnya yaitu menghitung faktor kesalahan tersembunyi

$\delta_j = \delta_{netj}\ f'(_{znetj}) = \delta net\ zj\ (1-Zj)$

$\delta_{01} = \delta_{net01}\ 1(1-1) = 0,011.1.(1-1) = 0$

$\delta_{11} = \delta_{net11}\ 1(1-Z1) = 0,011.1.(1-1) = 0$

$\delta_{21} = \delta_{net21}\ 1(1-Z2) = 0,011.1.(1-1) = 0$

$\delta_{31} = \delta_{net31}\ 1(1\text{-}Z3) = 0{,}011.1.(1\text{-}1) = 0$

$\delta_{41} = \delta_{net41}\ 1(1\text{-}Z4) = 0.063.1.(1\text{-}\ 5{,}36) = -0{,}27$

$\delta_{51} = \delta_{net51}\ 1(1\text{-}Z5) = 0{,}011.1.(1\text{-}1) = 0$

$\Delta v_{01} = \alpha\delta_{jxi}$

$\Delta v_{01} = \alpha\delta_{jxi}$

$\Delta v_{01} = \alpha\delta_{11}\ 1 = 0{,}2.0.1 = 0$

$\Delta v_{02} = \alpha\delta_{21}\ 1 = 0{,}2.0.1 = 0$

$\Delta v_{03} = \alpha\delta_{31}\ 1 = 0{,}2.0.1 = 0$

$\Delta v_{04} = \alpha\delta_{41}\ 1 = 0{,}2.(-0{,}27).1 = -0{,}054$

$\Delta v_{05} = \alpha\delta_{51}\ 1 = 0{,}2.0.1 = 0$

$\Delta v_{11} = \alpha\delta_{11}\ X_1 = 0{,}2.0.140 = 0$

$\Delta v_{12} = \alpha\delta_{21}\ X_2 = 0{,}2.0.93 = 0$

$\Delta v_{13} = \alpha\delta_{31}\ X_3 = 0{,}2.0.84 = 0$

$\Delta v_{14} = \alpha\delta_{41}\ X_4 = 0{,}2.(-0{,}27).0{,}04 = -0{,}002$

$\Delta v_{15} = \alpha\delta_{51}\ X_1 = 0{,}2.0.140 = 0$

$\Delta v_{21} = \alpha\delta_{11}\ X_1 = 0{,}2.0.140 = 0$

$\Delta v_{22} = \alpha\delta_{21}\ X_2 = 0{,}2.0.93 = 0$

$\Delta v_{23} = \alpha\delta_{31}\ X_3 = 0{,}2.0.84 = 0$

$\Delta v_{24} = \alpha\delta_{41}\ X_4 = 0{,}2.(-0{,}27).0{,}04 = -0{,}002$

$\Delta v_{25} = \alpha\delta_{51}\ X_2 = 0{,}2.0.93 = 0$

$\Delta v_{31} = \alpha\delta_{11}\ X_1 = 0{,}2.0.140 = 0$

$\Delta v_{32} = \alpha\delta_{21}\ X_2 = 0{,}2.0.93 = 0$

$\Delta v_{33} = \alpha\delta_{31}\ X_3 = 0{,}2.0.84 = 0$

$\Delta v_{34} = \alpha\delta_{41}\ X_4 = 0{,}2.(-0{,}27).0{,}04 = -0{,}002$

$\Delta v_{35} = \alpha\delta_{51}\ X_3 = 0{,}2.0.84 = 0$

$\Delta v_{41} = \alpha\delta_{11}\ X_1 = 0{,}2.0.140 = 0$

$\Delta v_{42} = \alpha\delta_{21}\ X_2 = 0{,}2.0.93 = 0$

$\Delta v_{43} = \alpha\delta_{31}\ X_3 = 0{,}2.0.84 = 0$

$\Delta v_{44} = \alpha\delta_{41}\ X_4 = 0{,}2.(-0{,}27).0{,}04 = -0{,}002$

$\Delta v_{45} = \alpha\delta_{51}\ X_4 = 0{,}2.0.0{,}04 = 0$

i) **Persamaan (2.19)** dan **(2.20)** telah masuk pada Fase 3, yaitu perubahan bobot yang mengarah unit keluaran.

$W_{kj}' = W_{kj} + \Delta W_{kj}$

$W_{10}' = W_{10} + \Delta W_{10} = 0,4 + 0,0486 = 0,4486$

$W_{11}' = W_{11} + \Delta W_{11} = 0,2 + 0,0486 = 0,2486$

$W_{21}' = W_{21} + \Delta W_{21} = 0,3 + 0,0486 = 0,3486$

$W_{31}' = W_{31} + \Delta W_{31} = -0,2 + 0,0486 = -0,1514$

$W_{41}' = W_{41} + \Delta W_{41} = 0,4 + 0,260 = 0,36$

$W_{51}' = W_{51} + \Delta W_{51} = 0,5 + 0,0486 = 0,5486$

$V_{ij}' = V_{ij} + \Delta V_{ij}$

$V_{01}' = V_{01} + \Delta V_{01} = -0,3 + 0 = -0,3$

$V_{02}' = V_{02} + \Delta V_{02} = 0,2 + 0 = 0,2$

$V_{03}' = V_{03} + \Delta V_{03} = 0,1 + 0 = 0,1$

$V_{04}' = V_{04} + \Delta V_{04} = 0,2 + (-0,54) = -0,34$

$V_{05}' = V_{05} + \Delta V_{05} = 0,3 + 0 = 0,3$

$V_{11}' = V_{11} + \Delta V_{11} = 0,5 + 0 = 0,5$

$V_{12}' = V_{12} + \Delta V_{12} = 0,2 + 0 = 0,2$

$V_{13}' = V_{13} + \Delta V_{13} = -0,1 + 0 = -0,1$

$V_{14}' = V_{14} + \Delta V_{14} = -0,3 + -0,002 = -0,302$

$V_{15}' = V_{15} + \Delta V_{15} = 0,4 + 0 = 0,4$

$V_{21}' = V_{21} + \Delta V_{21} = 0,1 + 0 = 0,1$

$V_{22}' = V_{22} + \Delta V_{22} = 0,4 + 0 = 0,4$

$V_{23}' = V_{23} + \Delta V_{23} = 0,2 + 0 = 0,2$

$V_{24}' = V_{24} + \Delta V_{24} = -0,5 + -0,002 = -0,502$

$V_{25}' = V_{25} + \Delta V_{25} = 0,1 + 0 = 0,1$

$V_{31}' = V_{31} + \Delta V_{31} = -0,2 + 0 = -0,2$

$V_{32}' = V_{32} + \Delta V_{32} = -0,3 + 0 = -0,3$

$V_{33}' = V_{33} + \Delta V_{33} = 0,1 + 0 = 0,1$

$V_{34}' = V_{34} + \Delta V_{34} = -0,2 + -0,002 = -0,202$

$V_{35}' = V_{35} + \Delta V_{35} = 0,2 + 0 = 0,2$

$V_{41}' = V_{41} + \Delta V_{41} = 0,4 + 0 = 0,4$

$V_{42}' = V_{42} + \Delta V_{42} = 0,1 + 0 = 0,1$

$V_{43}' = V_{43} + \Delta V_{43} = -0,3 + 0 = -0,3$

$V_{44}' = V_{44} + \Delta V_{44} = -0,1 + -0,002 = -0,102$

$V_{45}' = V_{45} + \Delta V_{45} = -0,4 + 0 = -0,4$

Setelah proses pelatihan data selesai dilakukan dan sudah didapatkan bobot pada semua layer, maka proses selanjutnya yaitu fase pengujian memakai data uji yang terdapat pada LAMPIRAN B:

$X_1 = 136$, $X2 = 89$, $X_3 = 80$, $X_4 = 0,04$

Penjumlahan *Neuron Layer* 1

$N_1 = v_{j0}' + \sum_{i=1}^{5} xivji' = v_{10}' + x_1v_{11}' + x_2v_{21}' + x_3v_{31}' + x_4v_{41}'$
$= -0,3 + 136.0,5 + 89.0,1 + 80.(0,2) + 0,04.0,4$
$= 60,616$

$N_2 = v_{j0}' + \sum_{i=1}^{5} xivji' = v_{20}' + x_1v_{12}' + x_2v_{22}' + x_3v_{32}' + x_4v_{42}'$
$= 0,2 + 136.0,2 + 89.0,4 + 80.(-0,3) + 0,04.0,1$
$= 39,0\ 04$

$N_3 = v_{j0}' + \sum_{i=1}^{5} xivji' = v_{30}' + x_1v_{13}' + x_2v_{23}' + x_3v_{33}' + x_4v_{43}'$
$= 0,1 + 136.(-0,1) + 89.0,2 + 80.0,1 + 0,04.(-0,3)$
$= 12,288$

$N_4 = v_{j0}' + \sum_{i=1}^{5} xivji' = v_{40}' + x_1v_{14}' + x_2v_{24}' + x_3v_{34}' + x_4v_{44}'$
$= 0,34 + 136.(-0,302) + 89.(-0,502) + 80.(-0,202) + 0,04.(-0,102)$
$= -101,574$

$N_5 = v_{j0}' + \sum_{i=1}^{5} xivji' = v_{50}' + x_1v_{15}' + x_2v_{25}' + x_3v_{35}' + x_4v_{45}'$
$= 0,3 + 136.0,4 + 89.0,1 + 80.0,2 + 0,04\ (-0,4)$
$= 79,584$

Nilai aktivasi *Neuron Layer* 1

$N_{netj} = f(N_j) = \frac{1}{1+e^{-znetj}}$

$N_{net1} = f(N_1) = \frac{1}{1+e^{-n1}} = \frac{1}{1+e^{-60,616}} = 1$

$N_{net2} = f(N_2) = \frac{1}{1+e^{-n2}} = \frac{1}{1+e^{-39,004}} = 1$

$N_{net3} = f(N_3) = \frac{1}{1+e^{-n3}} = \frac{1}{1+e^{-12,288}} = 1$

$N_{net4} = f(N_4) = \frac{1}{1+e^{-n4}} = \frac{1}{1+e^{101,574}} = 1,29$

$N_{net5} = f(N_5) = \frac{1}{1+e^{-n5}} = \frac{1}{1+e^{-79,584}} = 1$

Penjumlahan *Neuron Layer* 2

$N_1' = w_{10} + N_{net1}\, w_{11}' + N_{net2} w_{12}' + N_{net3}\, w_{13}' + N_{net4} w_{14}' + N_{net5} w_{15}'$
$= 0,4486 + 1.0,2486 + 1.0,3486 + 1.(-0,1514) + 1.29.0,36 + 1.0,5486$
$= 1,907$

Nilai aktivasi Neuron Layer 2

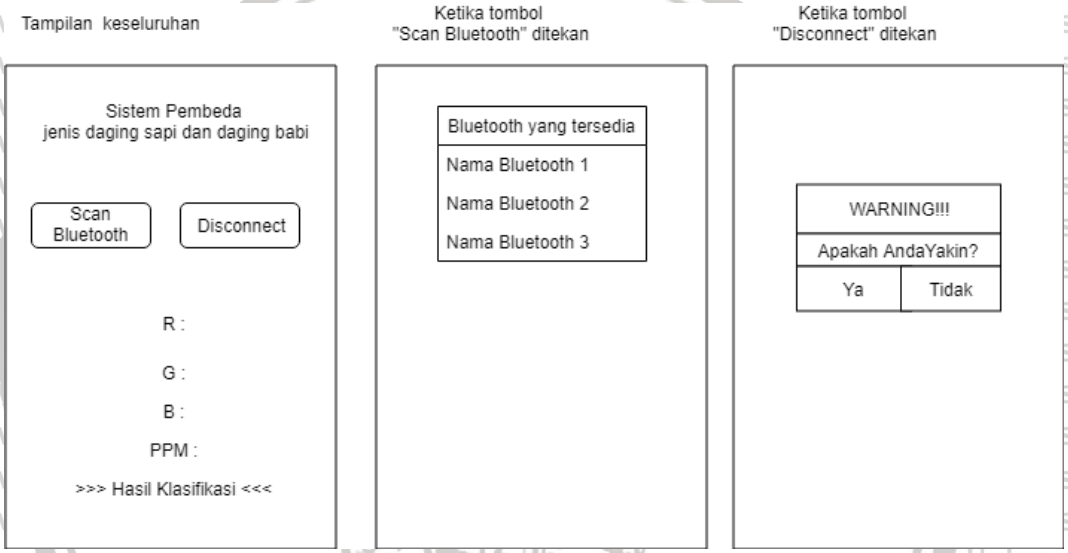
$$a = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

$N_1' = 1$ = Daging Sapi

Perhitungan tersebut menunjukkan bahwa hasil klasifikasi adalah 'Daging Sapi', namun pada kenyataannya data yang diambil sebenarnya adalah daging babi. Sehingga hasil klasifikasi tersebut berbeda dengan data yang sesungguhnya, hal tersebut dapat terjadi karena *epoch* yang dapat dihitung secara manual jumlahnya terbatas.

5.1.3.5 Perancangan menampilkan hasil klasifikasi pada aplikasi android

Untuk membuat sebuah aplikasi android, diperlukan sebuah desain *mockup* agar memudahkan dalam proses pembuatan aplikasi.



Gambar 5.7 *mockup design* untuk aplikasi android

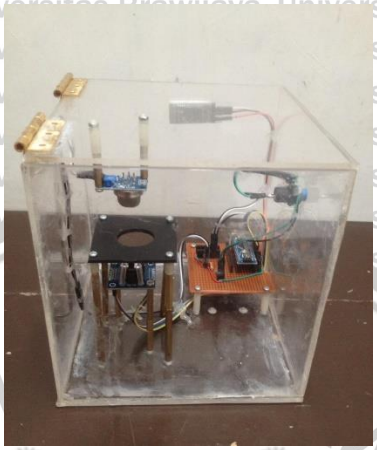
Pada Gambar 5.7 terlihat bahwa aplikasi akan mempunyai tampilan antarmuka secara keseluruhan seperti pada gambar diatas, dimana terdapat 2 tombol yaitu "*Scan Bluetooth*" dan "*Disconnect*" serta terdapat *textbox* sebagai penampil hasil pembacaan oleh sensor dan juga hasil klasifikasi yang dilakukan oleh sistem. Ketika tombol "*Scan Bluetooth*" ditekan maka akan menampilkan sebuah *listview* berisi daftar nama *Bluetooth* yang tersedia. Sedangkan ketika tombol "*Disconnect*" ditekan maka anak muncul sebuah *pop-up* berupa peringatan kepada *user.*

5.2 Implementasi Sistem

Pada tahap ini diterangkan mengenai bagaimana proses sistem dibuat dan merujuk pada keseluruhan perancangan sebelumnya. Dalam subbab ini diterangkan secara rinci mengenai proses implementasi *prototype* alat, implementasi perangkat keras maupun perangkat lunak.

5.2.1 Implementasi *Prototype* Alat

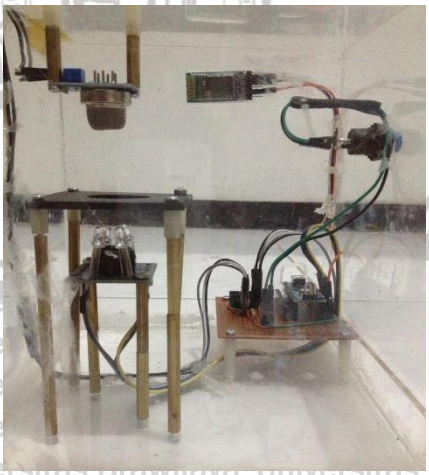
Pada implementasi prototype alat untuk sistem pembeda jenis daging sapi dan babi yang mengacu pada subbab 5.1.1, maka hasilnya seperti pada Gambar 5.8. Bahan menggunakan akrilik yang berwarna bening dengan ketebalan sekitar 0.3mm



Gambar 5.8 Implementasi *prototype* sistem

5.2.2 Implementasi Perangkat Keras

Dalam mengimplementasikan perangkat keras sistem pembeda jenis daging sapi dan babi ini terdapat beberapa komponen elektronik, diantaranya yaitu Arduino Nano, sensor warna TCS3200, sensor gas MQ-135, *module* Bluetooth HC-05, *push button*. Komponen - komponen tersebut dihubungkan sesuai perancangan pada subbab 5.1.2 sehingga menjadi kesatuan dengan memanfaatkan kabel *jumper*. Agar mempermudah saat melakukan pengkabelan antar komponen maka peneliti menambahkan *project board*.



Gambar 5.9 Implementasi rangkaian sensor warna TCS3200, sensor gas MQ135, dan modul Bluetooth HC-05

Hasil implementasi perangkaian sensor dengan Arduino Nano ditunjukkan oleh Gambar 5.9, dimana sensor warna TCS3200, serta modul Bluetooth HC-05 diletakkan didalam *box*. Sedangkan sensor gas MQ135 diletakkan pada bagian atas

agar memudahkan dalam proses membuka dan menutup. Sementara *push button* diletakkan di bagian luar *box* agar memudahkan dalam menggunakannya.

5.2.3 Implementasi Perangkat Lunak

Dalam subbab ini akan dijelaskan perihal proses implementasi program Sistem Pembeda Daging Sapi dan Daging Babi. Dibutuhkan 2 *software* yakni Matlab dan Arduino IDE untuk mengimplementasikan perangkat lunak ini.

5.2.3.1 Implementasi Pemberian *Input*

Input yang digunakan pada penelitian ini berupa sampel daging yang telah di tetapkan ukurannya.

Gambar 5.10 Ukuran sampel daging

Gambar 5.11 Daging Sapi dan Daging Babi sebagai sampel

5.2.3.2 Implementasi Kode Program Pengambilan Data Sensor

Agar sensor dapat membaca suatu nilai, maka diperlukan deklarasi *variable* dan konfigurasi pin sensor sesuai pada perancangan perangkat keras maupun

perangkat lunak. Karena proses pembuatan kode program sepenuhnya pada Arduino IDE 1.8.10 maka perlu melakukan deklarasi *library* terlebih dahulu, selanjutnya dapat dilanjutkan dengan mengimplementasi kode program untuk pengambilan data sensor.

Deklarasi *library* perlu dilakukan dalam membuat sebuah program. Hal ini dimaksudkan agar memudahkan dalam melakukan pemrograman terhadap fungsi-fungsi khusus. Tabel 5.6 menunjukkan proses implementasi *library* pada sistem ini, beberapa *library* yang digunakan yaitu *library* "math.h" untuk perhitungan matematika bersifat kompleks, dan *library* "TimerOne.h" untuk penggunaan interrupt pada pembacaan sensor TCS3200.

Tabel 5.6 Kode program deklarasi *library* sistem

Baris	Kode Program
1	`#include <Math.h>`
2	`#include <TimerOne.h>`

Setelah deklarasi *library* selesai dilakukan, maka langkah selanjutnya yaitu melakukan deklarasi variabel dan konfigurasi pin. Tabel 5.7 ditunjukkan bahwa pada baris ke-1 dan ke-2 dilakukan deklarasi variabel *array* sebagai penyimpan nilai hasil pembacaan sensor. Pada baris ke-3 dilakukan deklarasi *push button,* sedangkan pada baris ke-5 hingga ke-9 dilakukan deklarasi pin untuk sensor warna, pada baris ke-11 hingga ke-19 dilakukan deklarasi variable untuk mengaplikasikan sensor warna. Mulai dari baris ke-21 hingga ke-26 sudah tidak lagi digunakan untuk memprogram sensor warna, melainkan kode program untuk sensor gas. Dimana pada baris ke-21 dilakukan deklarasi pin data untuk sensor gas yakni pada pin analog A0, serta untuk baris ke-22 hingga ke-26 dilakukan deklarasi resistor beban (RL) sebesar 22KΩ, deklarasi kemiringan garis (m), kemudian deklarasi perpotongan titik koordinat (b), serta nilai resistansi dalam kondisi udara bersih (Ro). Nilai-nilai tersebut didapatkan melalui perhitungan secara matematis untuk proses kalibrasi sensor yang sebelumnya telah dijelaskan dalam bab dasar teori.

Tabel 5.7 Kode program deklarasi variable pembacaan sensor

Baris	Kode Program
1	`float bacaa[4];`
2	`float masukk [4];`
3	`int P_button;`
4	
5	`#define S0  6`
6	`#define S1  5`
7	`#define S2  3`
8	`#define S3  4`
9	`#define OUT 2`
10	
11	`bool active = false;`

```
12  int   g_countFreq = 0;     // hitung freq
13  int   g_saveValue[3];      // simpan nilai RGB
14  int   g_Fqueue = 0;        // filtr dari antrian RGB
15  float g_ScaleFactor[3];          // simpan faktor skala RGB
16
17  int   Red = 0;
18  int   Green = 0;
19  int   Blue = 0;
20  //gas sensor
21  #define MQ_sensor A0
22  #define Ro 22000
23  #define RL 22            // nilai resistor 22K
24  #define m -0.395         // kemiringan garis
25  #define b 0.794          // perpotongan titik koordinat
26  #define Ro 82            // nilai resistansi udara bersih
```

Tabel 5.8 termasuk kode program kalibrasi sensor MQ135 yang dilakukan untuk memperoleh nilai Ro. Baris ke-1 merupakan deklarasi resistor (RL) yang dipakai sebesar 22KΩ. Baris ke-3 menunjukkan *baud rate* yang dipakai yaitu 9600. Baris ke-7 sampai ke-10 ialah deklarasi *variable* yang dibutuhkan. Baris ke-15 merupakan sebuah perhitungan matematis untuk merubah nilai analog menjadi nilai tegangan. Pada baris ke-17 adalah penerapan dari **Persamaan (2.2).** Sementara baris ke-18 adalah penerapan dari **Persamaan (2.3)** yaitu untuk memperoleh nilai Ro. Sedangkan untuk baris ke-19 agar nilai Ro ditampilkan.

Tabel 5.8 Kode program kalibrasi sensor MQ135

Baris	Kode Program
1	`#define RL 22`
2	`void setup(){`
3	`  Serial.begin(9600);`
4	`}`
5	
6	`void loop() {`
7	`  float analog_valuee;`
8	`  float VRL;`
9	`  float Rs;`
10	`  float Ro;`
11	`  for(int test_cycle = 1 ; test_cycle <= 500 ; test_cycle++) {`
12	`    analog_valuee = analog_valuee + analogRead(A0); for 200`
13	`  }`
14	`  analog_valuee = analog_valuee/500.0;`
15	`  VRL = analog_valuee*(5.0/1023.0);`
16	
17	`  Rs = ((5.0/VRL)-1) * RL;`
18	`  Ro = Rs/3.6;`
19	`  Serial.print("Ro = ");`
20	`  Serial.println(Ro);`
21	`  delay(500);`

22	}

Tabel 5.9 menunjukkan bahwa terdapat sebuah fungsi (read_sensor()) yang digunakan oleh sensor gas MQ135 agar dapat mendeteksi kandungan kadar amonia dalam daging yang dideteksi. Baris ke-3 sampai ke-4 merupakan proses agar nilai PPM amonia dalam objek dapat diperoleh. Sensor MQ135 membaca nilai secara *analog,* kemudian nilai tersebut dirubah dalam bentuk *digital* menggunakan proses ADC, karena *input* tegangan yang digunakan pada mikrokontroler Arduino adalah menggunakan nilai *digital*. Pada baris ke-5 menunjukkan bahwa terdapat sebuah perhitungan matematis yang dilakukan untuk mendapatkan nilai resistansi sensor. Setelah mendapatkan nilai resistansi sensor, maka dilanjutkan pada baris ke-6 yaitu dengan melakukan proses kalibrasi agar nilai PPM yang diinginkan dapat diperoleh. Selanjutnya untuk baris ke-7 ditunjukkan bahwa nilai yang sudah didapatkan disimpan pada variable *array* agar bisa dipakai untuk perhitungan JST.

Tabel 5.9 Kode program pembacaan nilai sensor gas MQ135

Baris	Kode Program
1	`void read_sensorr(float arraybacaa[4]) {`
2	
3	`  float analogg = analogRead(gasSensor);`
4	`  float voltagee = (analogg / 1024) * 5;`
5	`  float Ra = ((Ro / voltagee * 5) - Ro);`
6	`  float ppm = MQ135_SCALINGFACTORR * pow((Ra / Ro), MQ135_EXPONENTT);`
7	`  arraybacaa[3] = ppm;`

Agar sensor warna tipe TCS3200 dapat melakukan pembacaan terhadap objek yang akan dideteksi, maka harus dilakukan proses kalibrasi dulu agar memperoleh nilai RGB dengan akurasi yang lebih baik. Pada Tabel 5.10, menjelaskan bahwa untuk melakukan proses kalibrasi terhadap sensor warna, ditunjukkan pada baris ke-3 bahwa langkah pertama yang harus dilakukan yaitu mendeklarasikan sensor. Baris ke-4 menunjukkan *baudrate* yang digunakan, selanjutnya dapat dilihat pada baris ke-5 sampai ke-9 digunakan untuk mengatur *interrupt* pembacaan setiap *photodiode* dalam sensor warna memanfaatkan *library* "timer1". Kemudian untuk menentukan skala faktor yang digunakan sebagai kalibrasi pada sensor dapat dilihat pada baris ke-10 sampai ke-12. Sementara nilai pembagi yang digunakan didapatkan melalui percobaan kalibrasi menggunakan kertas dengan warna putih, karena mempunyai nilai RGB maksimal yakni 255.

Tabel 5.10 Kode program pengaturan kalibrasi sensor warna TCS3200

Baris	Kode Program
1	`void setup() {`
2	`  pinMode(8, INPUT);`

```
3    TCS_Init();
4    Serial.begin(9600);
5    Timer1.initialize();
6    Timer1.attachInterrupt(TCS_Callbackk);
7    attachInterrupt(0, TCS_Countt, RISING);
8    delay(500);
9
10   g_ScaleFactor[0] = 255.0 / 579.973491 ;
11   g_ScaleFactor[1] = 255.0 / 598.707950 ;
12   g_ScaleFactor[2] = 255.0 / 819.271348 ;
13
14   delay(1000);
15 }
```

Tabel 5.11 menunjukkan sebuah proses *filter* warna untuk menentukan *photodiode* manakah yang dipakai nantinya. Jika pin S2 dan S3 nilainya masing-masing *LOW*, maka sensor tersebut akan memfilter warna merah. Sebaliknya jika pin S2 dan S3 bernilai masing-masing *HIGH,* maka *filter* pada warna hijau akan dilakukan oleh sensor. Sedangkan apabila pin S2 nilainya *LOW* dan pin S3 nilainya *HIGH,* maka warna biru yang di *filter* oleh sensor. Namun jika kebalikannya, pin S2 bernilai *HIGH* dan pin S3 bernilai *LOW* maka tidak ada proses *filter* yang dilakukan oleh sensor.

Tabel 5.11 Kode pemrograman untuk proses filter warna

Baris	Kode Program
1	`void TCS_Callbackk() {`
2	`  switch (g_Fqueue) {`
3	`    case 0:`
4	`      TCS_WhiteBalance (LOW, LOW);  //tanpa merah`
5	`      break;`
6	`    case 1:`
7	`      g_saveValue[0] = g_countFreq * 10;`
8	`      TCS_WhiteBalance (HIGH, HIGH);  //tanpa hijau`
9	`      break;`
10	`    case 2:`
11	`      g_saveValue[1] = g_countFreq * 10;`
12	`      TCS_WhiteBalance (LOW, HIGH);  //tanpa biru`
13	`      break;`
14	`    case 3:`
15	`      g_saveValue[2] = g_countFreq * 10;`
16	`      TCS_WhiteBalance (HIGH, LOW);  //bening tanpa filtr`
17	`      break;`
18	`    default:`
19	`      g_countFreq = 0;`
20	`      break;`
21	`  }`
22	`}`

Selanjutnya pada Tabel 5.12 dalam baris ke-1 sampai dengan ke-13 menjelaskan mengenai tahapan yang dilakukan oleh sensor warna TCS3200 untuk mendapatkan nilai RGB dari objek berdasarkan nilai frekuensinya. Sedangkan pada baris ke-15 hingga ke-18 ialah tahap menyimpan nilai RGB yang sudah didapatkan pada variabel *array*.

Tabel 5.12 Kode pemrograman pembacaan nilai sensor warna TCS3200

Baris	Kode Program
1	`g_Fqueue = 0;`
2	`  Red = g_saveValue[0] * g_ScaleFactor[0];`
3	`  Green = g_saveValue[1] * g_ScaleFactor[1];`
4	`  Blue = g_saveValue[2] * g_ScaleFactor[2];`
5	
6	`  if (Red > 255) {`
7	`    Red = 255;`
8	`  }`
9	`  if (Green > 255) {`
10	`    Green = 255;`
11	`  }`
12	`  if (Blue > 255) {`
13	`    Blue = 255;`
14	`  }`
15	
16	`  arraybacaa[0] = Red;`
17	`  arraybacaa[1] = Green;`
18	`  arraybacaa[2] = Blue;`
	`  arraybacaa[3] = ppm;`

Pada Tabel 5.13 dalam baris ke-3 terlihat bahwa telah dilakukan deklarasi terhadap *push button* pada pin digital 8. Baris ke-5 sampai ke-17 ialah kondisi dari *push button* ketika ditekan. Pada baris ke-19 sampai dengan ke-25 menjelaskan sebuah proses implementasi saat sensor mengambil data untuk diolah ketika terjadi penekanan pada *push button*. Sedangkan baris ke-26 sampai ke-28 berguna untuk memberikan petunjuk pada *user* agar menekan *push button* jika ingin mengetahui hasil klasifikasi oleh sistem.

Tabel 5.13 Kode program pengecekan pemberian trigger

Baris	Kode Program
1	`void loop() {`
2	
3	`  P_button = digitalRead (8);`
4	
5	`  if (state == 1) {`
6	`    if (P_button == HIGH) {`
7	`      active = true;`
8	`      state = 0;`
9	`    }`

```
10    }
11
12    if (state == 0) {
13      if (P_button == LOW) {
14        active = true;
15        state = 1;
16      }
17    }
18
19    if (active) {
20      read_sensor(bacaa);
21      masukk [0] = bacaa [0];
22      masukk [1] = bacaa [1];
23      masukk [2] = bacaa [2];
24      masukk [3] = bacaa [3];
25      active = false;
26    } else {
27      Serial.println(" Press Button ");
28      delay(1000);
29    }
```

5.2.3.3 Implementasi Perangkat Lunak Pelatihan Data

Tahap ini dilakukan dengan tujuan agar memperoleh nilai bobot yang nantinya akan digunakan dalam melakukan proses prediksi. Program untuk pelatihan ini diterapkan pada perangkat tambahan berupa Matlab yang dijalankan pada komputer. Untuk tahapan dalam melakukan pelatihan Jaringan Syaraf Tiruan *Backpropagation* terlihat pada Gambar 5.6.

Tabel 5.14 terlihat bahwa pada baris ke-1 terdapat *syntax* '*newff*' sebagai perintah untuk membentuk sebuah jaringan, dimana pada sistem ini akan dibuat JST *Backpropagation* umpan maju menggunakan *input* elemen *matrix* dengan nilai minimal dan maksimal pembacaan data yang telah masuk pada *form* Matlab, '*minmax*' digunakan supaya tidak melakukan *input* nilai satu persatu. Selanjutnya terdapat angka 5 dan 1 yang menunjukkan bahwa pada pelatihan ini digunakan 5 unit *neuron* sebagai *hidden layer* dengan *output layer* sebanyak *1 neuron*. Karena sistem ini membutuhkan *output* nilai antara 0 dan 1, maka digunakan fungsi *log-sigmoid* pada *hidden layer*. Sedangkan *output layer* memanfaatkan fungsi pelatihan *trainlm*. Pembuatan *form* untuk menyimpan nilai mse terdapat pada baris ke-2. Untuk mengatur nilai *learning rate* dilakukan pada baris ke-3, baris ke-4 untuk menampilkan frekuensi perubahan pada MSE, sedangkan untuk mengatur nilai *epoch* dilakukan pada baris ke-5.

Sementara untuk mengetahui berapakah waktu yang dibutuhkan saat proses *training,* digunakan kata "*tic*" sebagai awalan dan diakhiri menggunakan kata "*toc*", hal ini dapat dilihat pada baris ke-6 hingga ke-8. Untuk menampilkan hasil yang didapatkan setelah melakukan proses *training* dapat dilihat pada baris ke-9 hingga ke-16. Hasil yang didapat diantaranya mulai dari bobot *hidden*, bobot keluaran, hingga nilai *error* mse.

Sementara baris ke-17 menunjukkan kode program untuk melakukan simulasi pada hasil dari pelatihan menggunakan data sesungguhnya. Baris ke-18 digunakan untuk menampilkan grafik regresi dari target dan hasil pelatihan data, sedangkan grafik *performance* pelatihan data dapat dilihat pada baris ke-19.

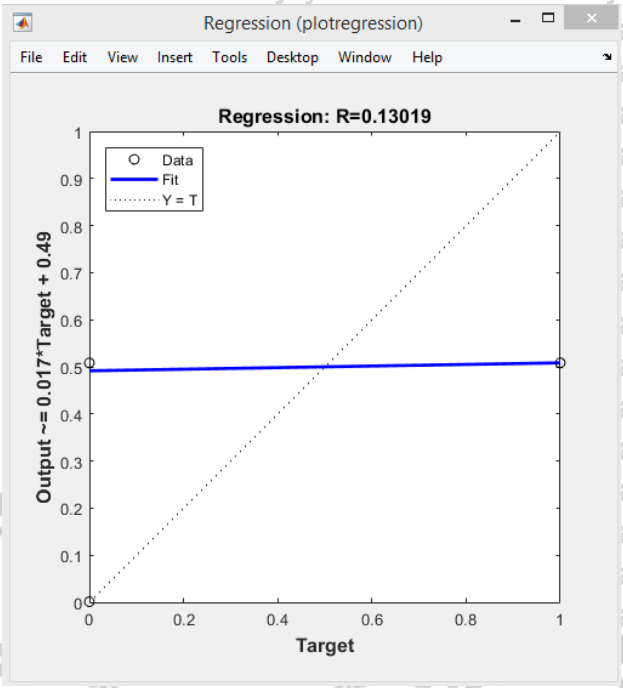
Tabel 5.14 Kode Program pelatihan Jaringan Syaraf Tiruan Backpropagation

Baris	Kode Program
1	`value = newff(minmax(pelatihan),[5 1],{'logsig','logsig'},'trainlm');`
2	`value.performFcn = 'mse';`
3	`value.trainParam.goal = 0.001;`
4	`value.trainParam.show = 25;`
5	`value.trainParam.epochs = 100000;`
6	`tic;`
7	`[value_keluaran,tr,Y,E] = train(value,pelatihan,targett);`
8	`waktu_trainingg = toc;`
9	`bobott_hidden = value_keluaran.IW{1,1};`
10	`bobott_keluaran = value_keluaran.LW{2,1};`
11	`biass_hidden = value_keluaran.b{1,1};`
12	`biass_keluaran = value_keluaran.b{2,1};`
13	`jumlah_iterasii = tr.num_epochs;`
14	`nilai_keluarann = Y;`
15	`nilai_eror = E;`
16	`eror_MSE = (1/60)*sum(nilai_eror.^2);`
17	`hasil_pelatihan = sim(value_keluaran,pelatihan);`
18	`figure, plotregression(targett,hasil_pelatihan,'Regression')`
19	`figure, plotperform(tr)`

Setelah melakukan implementasi terhadap algoritma *Backpropagation*, untuk mendapatkan nilai akurasi yang terbaik maka dibutuhkan penyesuaian nilai *learning rate* dan *epoch* pada program. Pada penelitian ini, peneliti telah mengumpulkan sebanyak 60 data sebagai data latih.

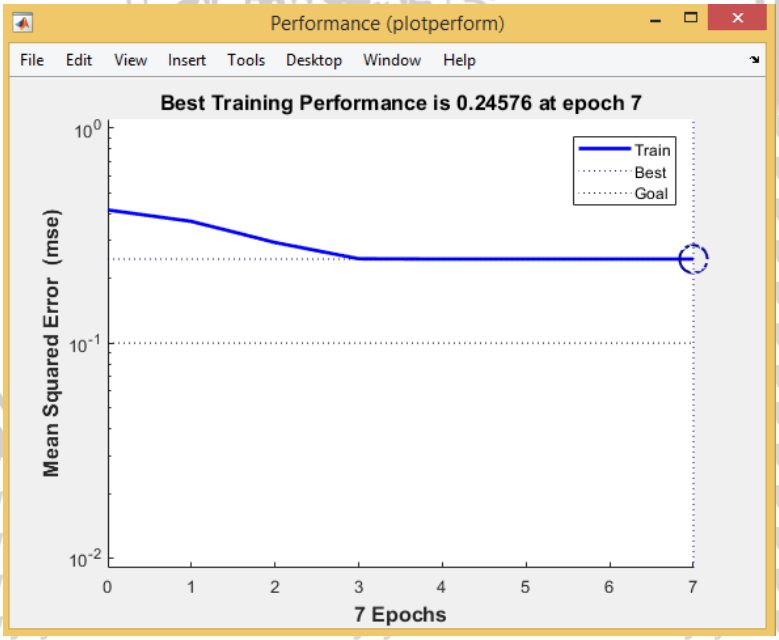
Peneliti mencoba memakai 5 neuron pada *hidden layer* sebagai percobaan dengan memanfaatkan beberapa kombinasi dari nilai *epoch* dan *learning rate*. Nilai *epoch* yang digunakan antara lain: 1000, 10000, dan 100000. Sedangkan nilai *learning rate* yang digunakan diantaranya: 0.1, 0.01, dan 0.001.

Percobaan dilaksanakan untuk mencari kombinasi yang sesuai sehingga diperoleh nilai yang memiliki tingkat kesalahan paling kecil saat pelatihan. Hasilnya terlihat pada Gambar 5.12 hingga Gambar 5.17 berikut:



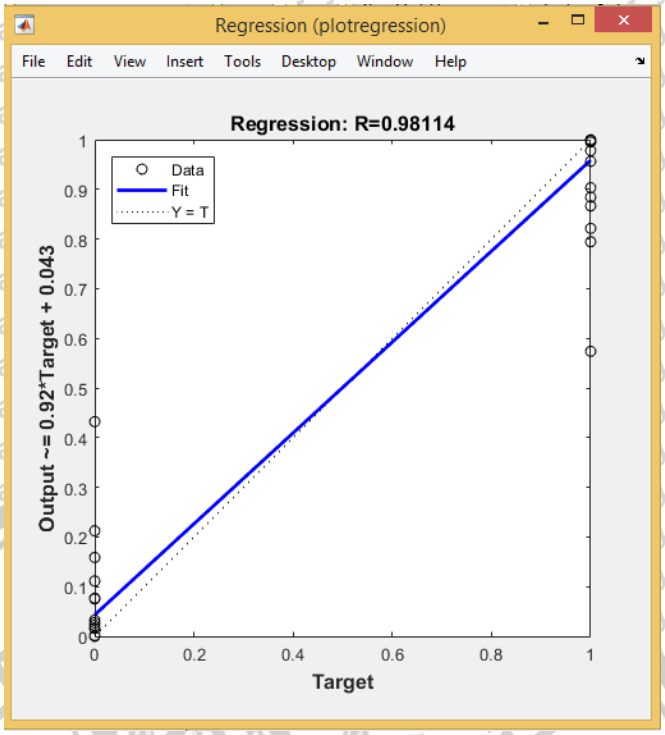
Gambar 5.12 Nilai R ketika *epoch* 1000 dan *learning rate* 0,1

Dari Gambar 5.12 dapat dilihat bahwa setelah dilakukan proses pelatihan data, diperoleh hasil koefisisen korelasi R sebesar 0.13019.



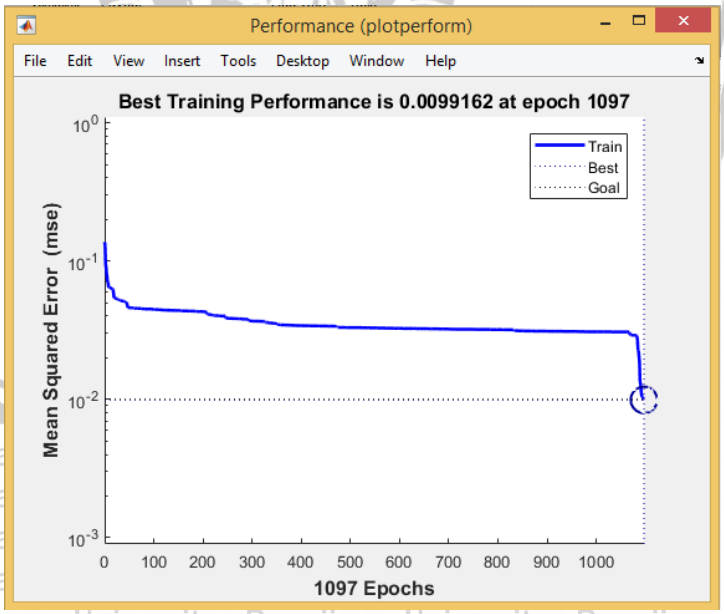
Gambar 5.13 Nilai *error goal* (MSE) ketika *epoch* 1000 dan *learning rate* 0,1

Dari Gambar 5.13 dapat dilihat bahwa setelah dilakukan proses pelatihan data, diperoleh hasil nilai *error goal* (MSE) sebesar 0.24576 pada *epoch* ke-7.



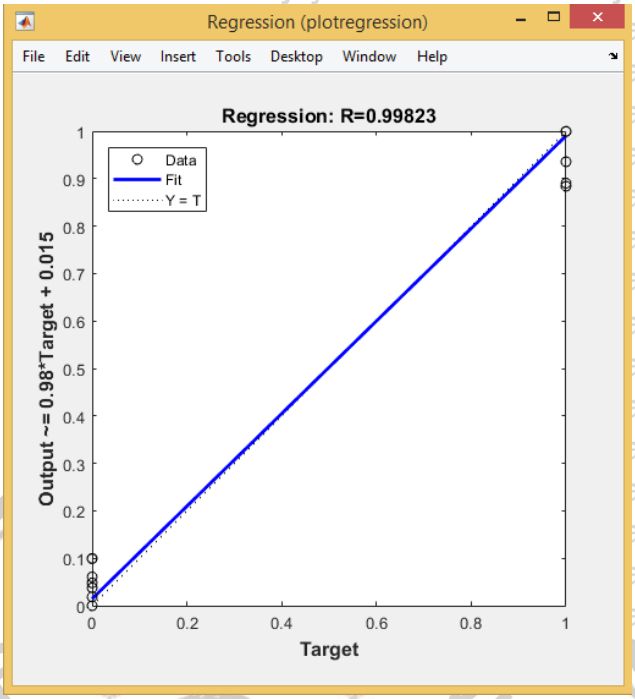
Gambar 5.14 Nilai R ketika *epoch* 10000 dan *learning rate* 0,01

Dari Gambar 5.14 dapat dilihat bahwa setelah dilakukan proses pelatihan data, diperoleh hasil koefisisen korelasi R sebesar 0.98114.



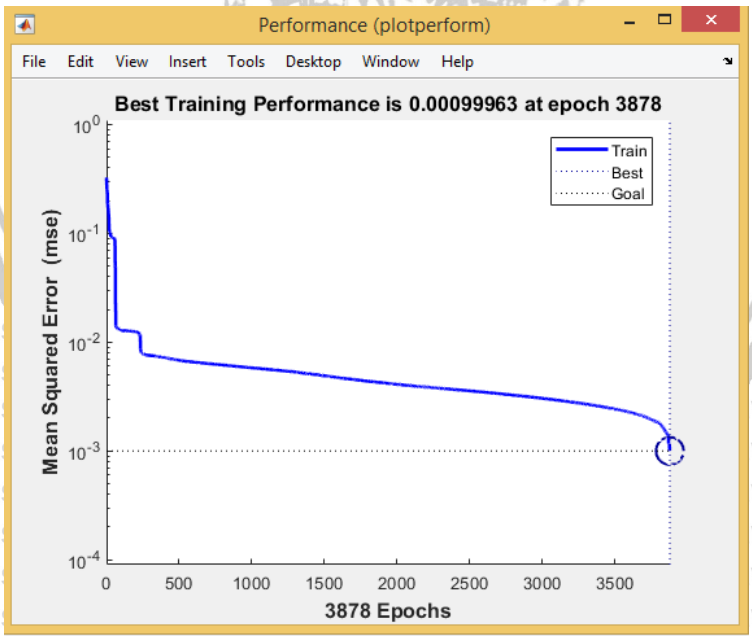
Gambar 5.15 Nilai *error goal* (MSE) ketika *epoch* 10000 dan *learning rate* 0,01

Dari Gambar 5.15 dapat dilihat bahwa setelah dilakukan proses pelatihan data, diperoleh hasil nilai *error goal* (MSE) sebesar 0.0099162 pada *epoch* ke-1097.



Gambar 5.16 Nilai R ketika *epoch* 100000 dan *learning rate* 0,001

Dari Gambar 5.16 dapat dilihat bahwa setelah dilakukan proses pelatihan data, diperoleh hasil koefisisen korelasi R sebesar 0.99823



Gambar 5.17 Nilai *error goal* (MSE) ketika *epoch* 100000 dan *learning rate* 0,001

Dari Gambar 5.17 dapat dilihat bahwa setelah dilakukan proses pelatihan data, diperoleh hasil nilai *error goal* (MSE) sebesar 0.00099963 pada *epoch* ke-3878.

Merujuk pada data hasil dari percobaan diatas, dapat ditarik kesimpulan bahwa kombinasi nilai yang paling bagus yaitu saat menggunakan *epoch* sebesar 100000 dan *learning rate* sebesar 0,001. Hal tersebut selaras dengan apabila nilai *error goal* (MSE) yang dihasilkan semakin kecil, sedangkan nilai koefisien korelasi R semakin mendekati 1, maka pelatihan tersebut akan memperoleh hasil dengan tingkat kesalahan yang semakin kecil.

Dari proses pelatihan JST yang dilakukan pada Matlab, didapatkan hasil yang digunakan sebagai nilai bobot. Dimana nilai-nilai tersebut mewakili nilai dari masing-masing bobot pada garis yang terdapat dalam arsitektur jaringan syaraf tiruan yang telah dirancang sebelumnya (Gambar 5.5).

Tabel 5.15 Nilai bobot dari proses pelatihan JST

	Bobot	
Bias *Hidden*	V_{01}	-5.73943435544124
	V_{02}	-11.2721842056059
	V_{03}	9.26738260010132
	V_{04}	-40.8879685123589
	V_{05}	-27.5913691401162
Bobot *Hidden*	V_{11}	0.0891424363801548
	V_{12}	0.524631480471023
	V_{13}	2.27439785541971
	V_{14}	0.152071481901794
	V_{15}	-0.596967404377651
	V_{21}	-0.198548446784767
	V_{22}	0.423797571615013
	V_{23}	1.35488894224097
	V_{24}	0.232675713121640
	V_{25}	0.462921799455575
	V_{31}	0.427289162522732
	V_{32}	0.143242396878870
	V_{33}	1.65694734501821
	V_{34}	0.0162114894204103
	V_{35}	0.722478211057660
	V_{41}	96.4898924418868
	V_{42}	10.2130041229696

	V_{43}	-101.849779629083
	V_{44}	-70.2804244623574
	V_{45}	-7.48844031236359
Bias Keluaran	W_{01}	2.34701385950815
Bobot Keluaran	W_{11}	7.20968252753860
	W_{12}	3.16237465585140
	W_{13}	6.60960365240546
	W_{14}	-80.8345568045453
	W_{15}	58.2682756364005

5.2.3.4 Implementasi Kode Program Jaringan Syaraf Tiruan.

Tahap ini dilakukan dengan tujuan agar dapat menerapkan kode program pengambilan keputusan oleh sistem sehingga bisa diketahui perbedaan jenis daging sapi dan babi menurut warna dan bau amonia pada daging yang akan dideteksi.

Tabel 5.16 Kode program variable nilai bobot Jaringan Syaraf Tiruan

Baris	Kode Program
1	`float v11 = 0.0891424363801548;  //bobot_hidden`
2	`float v12 = 0.524631480471023;`
3	`float v13 = 2.27439785541971;`
4	`float v14 = 0.152071481901794;`
5	`float v15 = -0.596967404377651;`
6	`float v21 = -0.198548446784767;`
7	`float v22 = 0.423797571615013;`
8	`float v23 = 1.35488894224097;`
9	`float v24 = 0.232675713121640;`
10	`float v25 = 0.462921799455575;`
11	`float v31 = 0.427289162522732;`
12	`float v32 = 0.143242396878870;`
13	`float v33 = 1.65694734501821;`
14	`float v34 = 0.0162114894204103;`
15	`float v35 = 0.722478211057660;`
16	`float v41 = 96.4898924418868;`
17	`float v42 = 10.2130041229696;`
18	`float v43 = -101.849779629083;`
19	`float v44 = -70.2804244623574;`
20	`float v45 = -7.48844031236359;`
21	
22	`float v01 = -5.73943435544124;  //bias_hidden`
23	`float v02 = -11.2721842056059;`
24	`float v03 = 9.26738260010132;`

```
25  float v04 = -40.8879685123589;
26  float v05 = -27.5913691401162;
27
28  float w11 = 7.20968252753860;       //bobot_keluaran
29  float w12 = 3.16237465585140;
30  float w13 = 6.60960365240546;
31  float w14 = -80.8345568045453;
32  float w15 = 58.2682756364005;
33
34  float w01 = 2.34701385950815;       //bias_keluaran
35
```

Tabel 5.16 terlihat bahwa dilakukan inisalisasi *variable* dan tipe data dari bobot yang diperoleh melalui pelatihan data menggunakan Matlab. Nilai tersebut dipakai oleh Arduino untuk memprogram metode JST.

Tabel 5.17 Kode program fungsi log-sigmoid

Baris	Kode Program
1	`float logsigmoid (float x) {`
2	`  return 1 / (1 + exp(-x));`
3	`}`

Tabel 5.17 menunjukkan kode program menghitung fungsi log-sigmoid seperti yang ditunjukkan pada Gambar 2.12.

Tabel 5.18 Kode program perhitungan *neuron layer* 1

Baris	Kode Program
1	`  float Neuron1layer1 = (arraybacaa [0] * v11) + (arraybacaa [1] * v21) + (arraybacaa [2] * v31) + (arraybacaa [3] * v41) + v01 ;`
2	`  float Neuron2layer1 = (arraybacaa [0] * v12) + (arraybacaa [1] * v22) + (arraybacaa [2] * v32) + (arraybacaa [3] * v42) + v02 ;`
3	`  float Neuron3layer1 = (arraybacaa [0] * v13) + (arraybacaa [1] * v23) + (arraybacaa [2] * v33) + (arraybacaa [3] * v43) + v03 ;`
4	`  float Neuron4layer1 = (arraybacaa [0] * v14) + (arraybacaa [1] * v24) + (arraybacaa [2] * v34) + (arraybacaa [3] * v44) + v04 ;`
5	`  float Neuron5layer1 = (arraybacaa [0] * v15) + (arraybacaa [1] * v25) + (arraybacaa [2] * v35) + (arraybacaa [3] * v45) + v05 ;`

Langkah awal untuk melakukan proses perhitungan Jaringan Syaraf Tiruan yaitu menjumlahkan *neuron layer* 1. Pada Tabel 5.18 dapat dilihat proses deklarasi variabel pada *neuron layer* 1 dan perhitungan menggunakan *input* (nilai bobot *hidden* dan bias *hidden*).

Tabel 5.19 Kode program aktivasi fungsi *neuron layer* 1

Baris	Kode Program
1	`  Neuron1layer1 = logsigmoid(Neuron1layer1);`
2	`  Neuron2layer1 = logsigmoid(Neuron2layer1);`
3	`  Neuron3layer1 = logsigmoid(Neuron3layer1);`

```
4     Neuron4layer1 = logsigmoid(Neuron4layer1);
5     Neuron5layer1 = logsigmoid(Neuron5layer1);
```

Proses dilanjutkan dengan menghitung nilai *neuron layer* 1 terhadap fungsi *logsigmoid* seperti dalam Tabel 5.19.

Tabel 5.20 Kode program perhitungan *neuron layer* 2

Baris	Kode Program
1 2	float Neuron1layer2 =(Neuron1layer1 * w11) + (Neuron2layer1 * w12) + (Neuron3layer1 * w13) + (Neuron4layer1 * w14) + (Neuron5layer1 * w15) + w01 ;

Setelah menyelesaikan perhitungan fungsi pada *neuron layer* 1, maka dilanjutkan dengan menghitung fungsi pada *neuron layer* 2 layaknya yang ditunjukkan dalam Tabel 5.20.

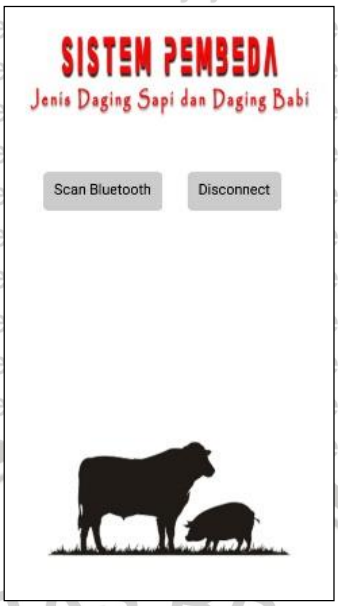
Tabel 5.21 Kode program aktivasi fungsi *hardlimit neuron layer* 2

Baris	Kode Program
1	if (Neuron1layer2 > 0) {
2	Serial.print (">>>>"); //tampilkan (Neuron1layer2);
3	Serial.print (" Daging Sapi ");
4	Serial.println ("<<<<");
5	Serial.println("------------------");
6	} else if (Neuron1layer2 < 0) {
7	Serial.print (">>>>"); //tampilkan (Neuron1layer2);
8	Serial.print (" DAGING BABI ");
9	Serial.println ("<<<<");
10	Serial.println("------------------");
11	}
12	delay (5000);
13	Serial.println ("");
14	}

Pada Tabel 5.21 ini termasuk proses terakhir dalam perhitungan ini. Setelah proses ini selesai dilakukan maka akan didapatkan sebuah kesimpulan dari perhitungan JST yang dilakukan. Kesimpulan didapatkan dari nilai hasil perhitungan pada neuron layer 2 yang ditambahkan ke fungsi hardlimit layaknya yang ditunjukkan dalam Gambar 2.10.

5.2.3.5 Implementasi menampilkan hasil klasifikasi pada aplikasi Android

Pembuatan aplikasi untuk menampilkan hasil klasifikasi dari sistem dilakukan pada MIT App Inventor. Di bawah ini merupakan implementasi dari desain antarmuka dan *block* sistem pada MIT App Inventor. Pada Gambar 5.18 menunjukkan desain antarmuka aplikasi android menggunakan 2 tombol ("*Scan Bluetooth*" dan "*Disconnect*") yang digunakan sebagai penghubung dan pemutus koneksi *Bluetooth* antara *smartphone* android dengan sistem.

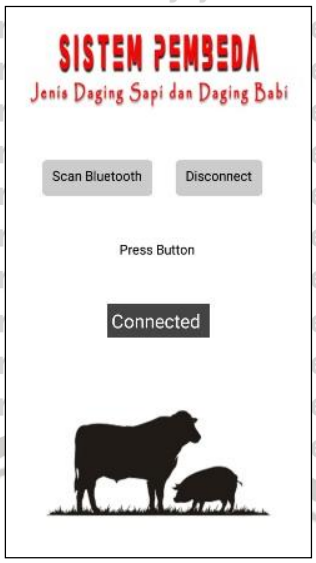


Gambar 5.18 Desain antarmuka keseluruhan sistem

Gambar 5.19 menunjukkan ketika tombol "Scan Bluetooth" ditekan maka akan menampilkan nama-nama *Bluetooth* yang tersedia dalam bentuk *listview*.

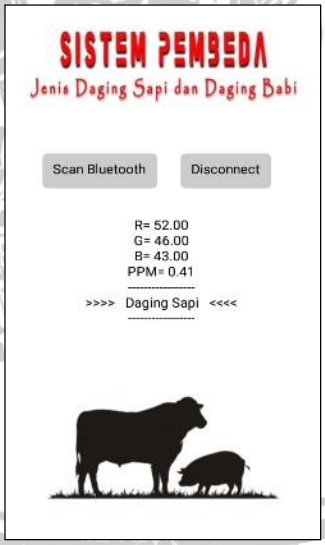
Gambar 5.19 Tampilan ketika tombol "Scan Bluetooth" ditekan

Pada Gambar 5.20 dapat dilihat bahwa ketika *Bluetooth* sudah terkoneksi antara *device* dengan sistem, maka akan memberikan sebuah *pop-up* "*Connected*" yang menunjukkan bahwa *Bluetooth* sudah terkoneksi.



Gambar 5.20 Tampilan ketika Bluetooth sudah terhubung

Pada Gambar 5.21 menunjukkan tampilan ketika terjadi penekanan pada *push button*, maka *trigger* akan diberikan kepada sistem agar membaca nilai dari sensor dan mengolahnya menggunakan Jaringan Syaraf Tiruan sehingga didapatkan hasil klasifikasi jenis daging apa yang sedang dideteksi dan menampilkan hasil tersebut pada *textbox* yang terdapat pada aplikasi.

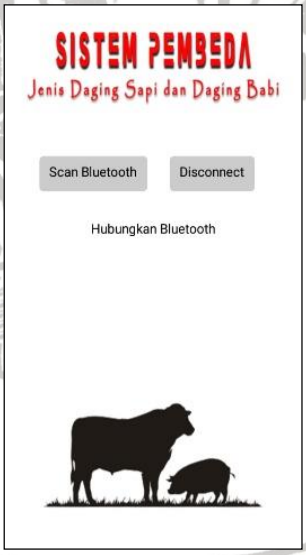


Gambar 5.21 Tampilan ketika *push button* ditekan

Gambar 5.22 menunjukkan bahwa ketika tombol "*Disconnect*" ditekan, maka akan muncul sebuah *notifier* berupa peringatan apakah *user* yakin ingin memutuskan koneksi *Bluetooth* antara aplikasi dengan sistem.

Gambar 5.22 Tampilan ketika tombol "Disconnect" ditekan

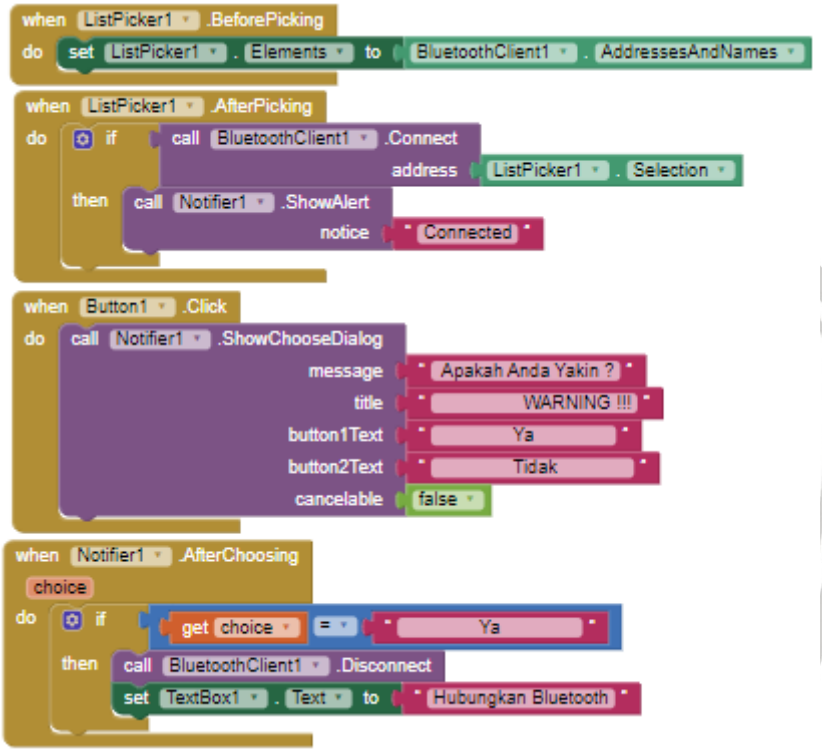
Gambar 5.23 menunjukkan ketika *user* menekan tombol "Ya" pada pesan peringatan yang berarti setuju untuk memutuskan koneksi *Bluetooth* antara aplikasi dan sistem. Sehingga koneksi *Bluetooth* terputus dan pada aplikasi akan memunculkan perintah untuk "Hubungkan *Bluetooth*" yang artinya *user* harus menghubungkan *Bluetooth* terlebih dahulu agar sistem dapat digunakan.



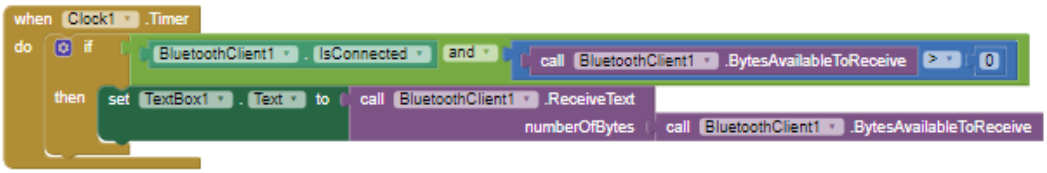
Gambar 5.23 Tampilan ketika menekan "Ya" pada *notifier*

Gambar 5.24 menjelaskan tahapan dalam membuat program *pairing Bluetooth* antara sistem dengan ponsel android memanfaatkan modul Bluetooth HC-05. Pada *block* pertama berguna untuk melakukan pembacaan terhadap alamat dan nama - nama dari koneksi *Bluetooth* yang tersedia untuk dapat dilakukan proses *pairing*. *Block* kedua berfungsi untuk *pairing* terhadap alamat *Bluetooth* yang sudah dipilih, kemudian setelah *Bluetooth* sudah terkoneksi maka akan muncul sebuah *notifier* "*Connected*" yang menunjukkan bahwa *Bluetooth* sudah terkoneksi.

Pada *block* ketiga berfungsi untuk memberikan sebuah pesan peringatan ketika tombol "*Disconnect*" ditekan. Pesan tersebut berisi sebuah pertanyaan "Apakah anda yakin?" dan juga dua tombol antara "ya" dan "tidak". Sedangkan pada *block* keempat merupakan program untuk memunculkan sebuah pesan peringatan pada *user*. Ketika *user* menekan tombol "tidak", maka tidak akan terjadi perubahan pada konektivitas *bluetooth* dan akan diarahkan kembali pada halaman sebelumnya. Namun ketika *user* memilih tombol "ya", maka yang terjadi adalah *Bluetooth* yang sudah terkoneksi sebelumnya akan terputus dan memunculkan perintah "Hubungkan Bluetooth" pada *textbox*.



Gambar 5.24 Block program *pairing* Bluetooth dengan Aplikasi Android



Gambar 5.25 *Block* program pembacaan nilai *serial* Aplikasi Android

Pada Gambar 5.25 menjelaskan tahapan ketika aplikasi Android menerima serta menampilkan data dari Arduino Nano melalui modul *Bluetooth* HC-05.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini akan dijelaskan alur dari pengujian dan analisis hasil dari pengujian yang dilaksanakan. Terdapat tiga jenis pengujian yang dilakukan yaitu secara fungsional dari setiap komponen, pengujian akurasi sistem, dan waktu komputasi pemrosesan sistem.

6.1 Pengujian Sensor Warna TCS3200

Sensor warna dengan tipe TCS3200 mempunyai fungsi untuk mendapatkan nilai warna dasar yaitu RGB dari daging. Sensor ini diuji dengan cara mendekatkan sensor pada beberapa sampel daging menggunakan sensor TCS3200 kemudian membandingkan hasilnya menggunakan *Eyedropper tool* yang ada pada aplikasi *Adobe Photoshop* sebagai alat ukur warna. Warna yang digunakan sebagai parameter pengujian pada *Eyedropper tool* diperoleh dari foto sampel daging yang diambil terlebih dahulu.

6.1.1 Tujuan Pengujian

Pengujian bertujuan untuk memperoleh nilai warna dasar dari objek penelitian yaitu daging. Dimana warna yang didapat dari sensor akan dibandingkan dengan nilai warna dasar dari foto yang sudah diambil sebelumnya. Selanjutnya nilai warna dasar dari keduanya akan dibandingkan untuk memperoleh selisih nilai dari keduanya dan juga dapat mengetahui berapakah nilai *error* serta tingkat akurasi yang dihasilkan oleh sensor.

6.1.2 Prosedur Pengujian

Adapun prosedur dalam melakukan pengujian terhadap sensor warna TCS3200:

a) Menguhubungkan sistem dengan laptop.
b) Mengunggah kode program untuk sensor warna TCS3200 ke mikrokontroler.
c) Mengukur nilai warna dasar RGB pada sepuluh sampel daging yang terdiri dari lima sampel daging babi dan lima sampel daging sapi memakai sensor warna tipe TCS3200.
d) Menyesuaikan warna dasar RGB dari daging yang dijadikan sebagai sampel dengan memfoto permukaan daging tersebut, hasil foto akan diukur nilai RGBnya dengan fitur *Eyedropper tool* pada *Adobe Photoshop*.
e) Meninjau hasil pembacaan nilai warna dasar RGB dari sensor pada aplikasi Android kemudian membandingkannya dengan nilai yang ada pada fitur *eyedropper tool*. Kemudian ditentukan besar nilai error yaitu dengan mengkonversi nilai warna dasar RGB dari sensor dan nilai warna dasar RGB dari gambar daging pada Photoshop menjadi nilai HEX.
f) Dari kedua nilai HEX sebelumnya diambil selisihnya dan dilakukan komputasi untuk mendapatkan presentase eror dalam desimal.

Berikut persamaan untuk mendapatkan presentase error:

$$Presentase\ Error = \frac{Selisih\ nilai\ pembacaan}{Pembacaan\ eyedropper} x\ 100\% \quad (6.1)$$

Untuk memperoleh selisih nilai pembacaan menggunakan persamaan di bawah:

$$Selisih\ pembacaan\ nilai = |pembacaan\ sensor - pembacaan\ eyedropper| \quad (6.2)$$

6.1.3 Hasil dan Analisis Pengujian

Hasil pengujian pada sensor warna tipe TCS3200 dapat dilihat di Tabel 6.1, pada tabel menunjukkan jika sensor tidak hanya bisa membaca warna dasar saja seperti Merah, Hijau, dan Biru. Tetapi juga bisa melakukan pembacaan terhadap beberapa macam warna yang berbeda. Setelah membandingkan nilai antara warna dasar yang dihasilkan oleh sensor dengan *eyedropper tool* yang didapat dari daging yang sama, ternyata bisa memiliki nilai yang berbeda.

Dari persamaan 6.1 yang digunakan dalam setiap pengujian, didapatkan rata-rata eror sebesar 1,73%. Berikut contoh menggunakan persamaan 6.1 pada pengujian pertama:

Nilai HEX warna dari pembacaan sensor = 8C6150 (HEX) = 9199952 (DEC)

Nilai HEX warna dari pembacaan alat ukur = 88514C (HEX) = 8933708 (DEC)

$$Selisih\ pembacaan\ nilai = |pembacaan\ sensor - pembacaan\ alat\ ukur|$$

$$= |9199952 - 8933708|$$

$$= 266244$$

$$Presentase\ Error = \frac{Selisih\ nilai\ pembacaan}{Pembacaan\ alat\ ukur} x\ 100\%$$

$$= \frac{266244}{89337608} x\ 100\%$$

$$= 2{,}89\%$$

Tabel 6.1 Hasil Pengujian pembacaan sensor warna TCS3200

Pengujian ke -	Warna Daging Pengujian	Pembacaan Sensor					Pembacaan Eyedropper tool Photoshop					Selisih Error		Error
		R	G	B	HEX	Warna	R	G	B	HEX	Warna	HEX	Decimal	
1.		140	97	80	#8c6150		139	84	79	#88514c		#41004	266244	2,89%
2.		136	93	84	#885d54		133	86	78	#85564e		#30706	198406	2,22%
3.		162	110	96	#a26e60		164	106	94	#a46a5e		#1fbfe	130046	1,22%
4.		184	127	105	#b87f69		183	134	118	#b78676		#f8f3	63731	0,52%
5.		171	105	99	#ab6963		169	111	97	#a96f61		#1fa02	129538	1,15%
6.		131	93	84	#835d54		135	90	81	#875a51		#3fcfd	261373	3,03%
7.		118	85	71	#765547		117	78	73	#754e49		#106fe	67326	0,86%

8.		114	76	71	#724c47		119	74	68	#774a44		#4fdfd	327165	0,04%
9.		127	85	77	#7f554d		124	85	80	#7c5550		#2fffd	196605	2,35%
10.		131	89	77	#83594d		135	83	76	#87524c		#3f8ff	260351	3,02%
Rata - rata														1,73%

Berikut komputasi rata-rata eror pada seluruh pengujian yang dilakukan:

$$Rata - rata\ error = |\frac{Jumlah\ persentase\ error}{Jumlah\ pengujian}|x\ 100\%$$

$$= |\frac{2{,}89 + 2{,}22 + 1{,}22 + 0{,}52 + 1{,}15 + 3{,}03 + 0{,}86 + 0{,}04 + 2{,}35 + 3{,}02}{10}|$$

$$= 1{,}73\%$$

Akurasi pembacaan sensor dalam membaca warna dapat dikatakan sangat baik, karena nilai *error* yang dihasilkan saat melakukan pembacaan relatif kecil. Meskipun terdapat *error* pada saat melakukan pembacaan, sensor mampu membaca warna sesuai dengan objek yang dideteksi meskipun terdapat sedikit perbedaan warna. Hal itu dapat dilihat pada kepekatan warna yang berbeda pada kolom warna *Eyedropper tool*. Jika warna objek yang terdeteksi semakin gelap, maka nilai warna dasar RGB yang dihasilkan akan semakin rendah (0, 0, 0). Namun jika warna objek yang terdeteksi semakin terang, maka nilai warna dasar RGB yang dihasilkan akan semakin tinggi (255, 255, 255).

6.2 Pengujian Sensor Gas MQ135

Sensor gas tipe MQ135 dalam sistem ini bertugas sebagai pembaca nilai kadar amonia pada daging. Pengujian dilaksanakan dengan cara melakukan pengukuran terhadap nilai gas amonia pada daging dan tegangan keluaran pada sensor ketika mendeteksi daging. Dalam melakukan pengujian ini, daging terlebih dahulu dibiarkan dalam waktu yang lama sampai daging mulai membusuk. Ketika daging mengalami pembusukan, maka nilai amonia yang dikeluarkan oleh daging akan semakin membesar.

6.2.1 Tujuan Pengujian

Pengujian ini bertujuan untuk mengetahui kinerja fungsional sensor gas tipe MQ135 dalam membaca kadar amonia pada daging.

6.2.2 Prosedur Pengujian

Adapun prosedur melaksanakan pengujian pada sensor gas tipe MQ135:

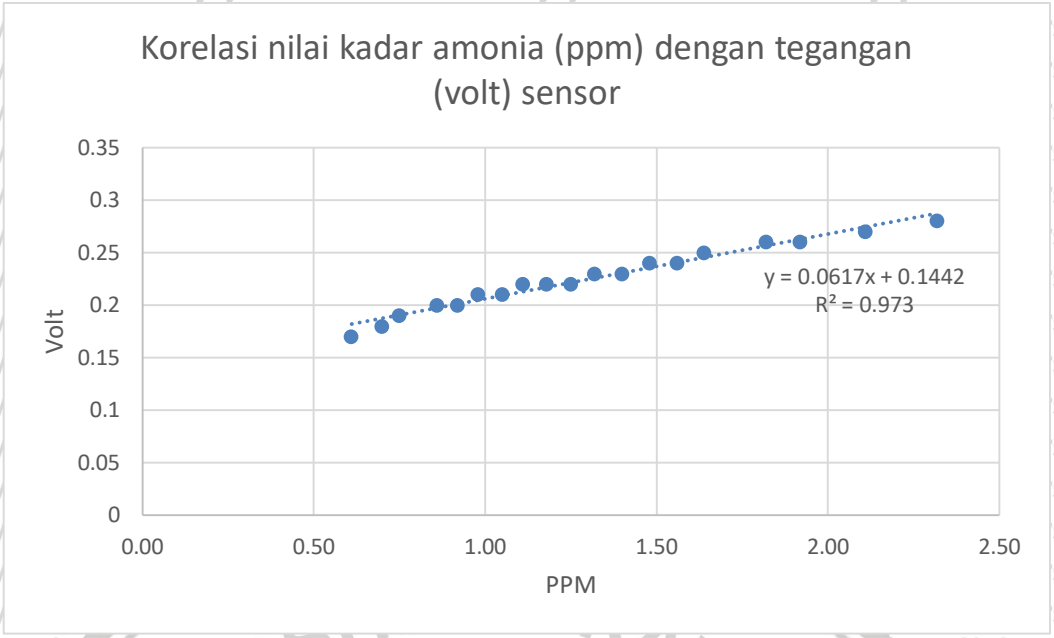
a) Menghubungkan sistem dengan laptop.
b) Mengunggah kode program sensor gas MQ135.
c) Menghitung nilai gas amonia yang terbaca oleh sensor dan tegangan keluaran dari sensor gas tipe MQ135 pada sampel daging yang digunakan sebagai data uji.
d) Mengamati hasilnya dan menganalisis karakteristik sensor tipe MQ135 pada *datasheet*.

6.2.3 Hasil dan Analisis Pengujian

Tabel 6.2 Hasil Pengujian pembacaan sensor gas MQ135

No.	Urutan Pengujian	PPM	V_{out} (V)
1.	Pertama	0,52	0,16
2.	Kedua	0,61	0,17
3.	Ketiga	0,70	0,18
4.	Keempat	0,75	0,19
5.	Kelima	0,86	0,20
6.	Keenam	0,92	0,20
7.	Ketujuh	0,98	0,21
8.	Kedelapan	1,05	0,21
9.	Kesembilan	1,11	0,22
10.	Kesepuluh	1,18	0,22
11.	Kesebelas	1,25	0,22
12.	Kedua belas	1,32	0,23
13.	Ketiga belas	1,40	0,23
14.	Keempat belas	1,48	0,24
15.	Kelima belas	1,56	0,24
16.	Keenam belas	1,64	0,25
17.	Ketujuh belas	1,82	0,26
18.	Kedelapan belas	1,92	0,26
19.	Kesembilan belas	2,11	0,27
20.	Kedua puluh	2,32	0,28
Rata - rata		1,249	0,222

Tabel 6.2 merupakan hasil pengujian nilai ppm dan tegangan keluaran dari sensor gas tipe MQ135. Dari 20 kali pengujian yang telah dilakukan, sensor gas tipe MQ135 sudah selaras dengan karakteristik. Mengacu dari datasheet sensor gas tipe MQ135, sensor ini tergolong sensor yang resistif dikarenakan tingginya nilai resistansi sensor dapat mempengaruhi nilai sensor (ppm) akan turun, selaras dengan rumus tegangan yakni V = I x R.



Gambar 6.1 Analisis Korelasi Kadar Amonia(PPM) dengan Volt

Hubungan antara nilai hasil pembacaan gas amonia pada sampel dengan tegangan keluaran sensor ditunjukkan oleh grafik diatas. Dari Gambar 6.1 menunjukan nilai $y = 0.0617x + 0.1442$ adalah model regresi *power* dari kedua variabel yang digunakan, sementara $R^2 = 0.973$ biasa disebut dengan koefisien determinasi, artinya variabel PPM dipengaruhi oleh variabel tegangan keluaran sebanyak 97,3% dan sisanya dipengaruhi oleh variabel lain selain konsentrasi amonia.

6.3 Pengujian modul Bluetooth HC-05

Modul Bluetooth HC-05 difungsikan untuk mengirim data ke aplikasi Android oleh komponen Arduino. Tampilan pada aplikasi akan selaras dengan yang ditentukan pada Arduino, daam hal ini akan ditampilkan nilai hasil pembacaan kedua sensor serta hasil klasifikasi sistem.

6.3.1 Tujuan Pengujian

Pengujian bertujuan untuk mencari tahu kinerja fungsional dari modul *bluetooth* yaitu sinkronisasi antar perangkat sistem dengan aplikasi android yang sudah diciptakan.

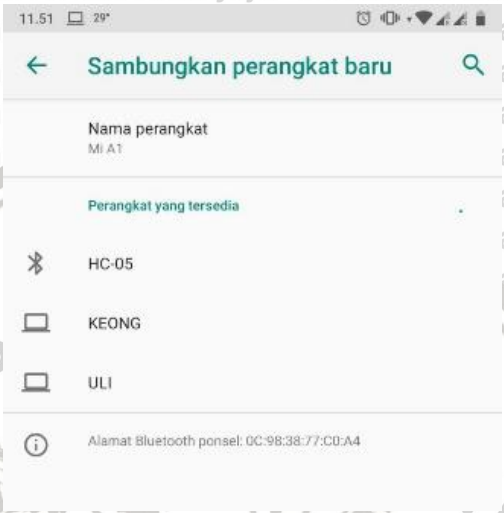
6.3.2 Prosedur Pengujian

a) Menyambungkan modul *bluetooth* pada Arduino Nano sesuai dengan perancangan perangkat keras.
b) Menghubungkan sistem dengan laptop.
c) Menghidupkan *Bluetooth* pada ponsel Android.
d) Membuka menu "*Bluetooth Settings*" pada ponsel Android, kemudian lakukan penyambungan antara ponsel dengan Modul *bluetooth* HC-05.

e) Mengamati hasil dari proses *pairing* antara ponsel Android dan Modul HC-05 apakah sudah tersambung atau belum.

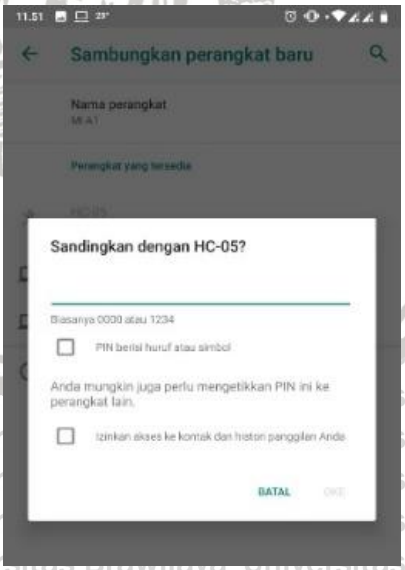
6.3.3 Hasil dan Analisis Pengujian

Berdasarkan hasil pengujian yang sudah dilaksanakan, melalui menu *"Bluetooth Settings"* terlihat bahwa ponsel Android mampu mendeteksi modul *Bluetooth* HC-05.



Gambar 6.2 Tampilan ketika modul *Bluetooth* HC-05 terdeteksi

Untuk dapat melakukan pairing antara *Bluetooth* pada modul dengan ponsel Android biasanya dibutuhkan sebuah *password* untuk dapat melakukan pairing. Dan PIN yang digunakan sebagai *password* yaitu "1234".



Gambar 6.3 Tampilan ketika *Bluetooth* membutuhkan *password* untuk *pairing*

Setelah PIN sudah dimasukkan maka proses *pairing* antara sistem dengan ponsel telah berhasil dilakukan.

Gambar 6.4 Tampilan ketika proses *Pairing* berhasil dilakukan

Setelah proses *pairing* berhasil, untuk mengecek apakah benar-benar sudah terhubung atau belum. Buka aplikasi yang sudah terpasang pada ponsel Android dan tekan tombol *"Scan Bluetooth"*. Kemudian amatilah apakah *Bluetooth* HC-05 sudah terdeteksi atau belum. Jika sudah terdeteksi maka proses *pairing* antara sistem dengan ponsel Android telah berhasil dilakukan dan sistem siap untuk digunakan.

6.4 Pengujian Aplikasi Android pada *Smartphone*

Aplikasi Android pada ponsel merupakan komponen untuk melihat hasil olahan oleh mikrokontroler agar dapat ditampilkan pada ponsel Android.

6.4.1 Tujuan Pengujian

Pengujian terhadap aplikasi ponsel Android ini bertujuan untuk mencari tahu apakah keluaran yang ditampilkan pada ponsel Android sesuai keluaran yang telah ditentukan pada Arduino Nano.

6.4.2 Prosedur Pengujian

a) Menyambungkan sistem pada laptop.
b) Meng-*install* aplikasi yang sudah dirancang ke ponsel Android.
c) Menghidupkan *Bluetooth* pada ponsel Android.
d) Membuka menu "*Bluetooth Settings*" pada ponsel Android, kemudian lakukan penyambungan antara ponsel dengan Modul HC-05.
e) Menjalankan aplikasi "Pembeda Daging" pada ponsel Android, kemudian tekan tombol "Scan Bluetooth". Lakukan pairing dengan memilih ID untuk Bluetooth HC-05.
f) Menyimak hasil pembacaan nilai dari sistem pada tampilan aplikasi ponsel.

6.4.3 Hasil dan Analisis Pengujian

Pengujian yang dilaksanakan menghasilkan sebuah aplikasi ponsel yang dapat menampilkan hasil pengolahan oleh Arduino Nano sesuai harapan dan

dapat berjalan secara real time. Hasil dari pengolahan sistem terlihat dalam Gambar 6.4.

Gambar 6.5 Tampilan hasil pengolahan sistem pada Aplikasi

6.5 Pengujian Akurasi Hasil Klasifikasi

Penelitian ini memanfaatkan metode Jaringan Syaraf Tiruan (JST) untuk melakukan klasifikasi, metode ini digunakan untuk mencari perbedaan antara daging sapi dengan daging babi. Namun diperlukan pengujian terlebih dahulu agar diketahui tingkat akurasi dari penggunaan metode ini untuk penelitian ini.

6.5.1 Tujuan Pengujian

Pengujian ini dilakukan dengan tujuan agar dapat diketahui seberapa besar tingkat akurasi dari sistem ini untuk melakukan klasifikasi terhadap jenis daging antara daging sapi dan babi memanfaatkan metode JST *Backpropagation*.

6.5.2 Prosedur Pengujian

a) Menyambungkan sistem pada peragkat laptop.
b) Meletakkan daging yang akan dideteksi ke tempat objek (di atas sensor warna tipe TCS3200 dan dibawah sensor gas tipe MQ135).
c) Menghubungkan koneksi *Bluetooth* antara sistem dengan *Ponsel*.
d) Menekan *push button* pada sistem untuk memberi *trigger* agar sensor mengambil data untuk dilakukan proses klasifikasi oleh sistem.
e) Mengamati hasil klasifikasi melalui aplikasi pada *Ponsel.*

6.5.3 Hasil dan Analisis Pengujian

Melalui pengujian pada 30 sampel daging yang terdiri dari 15 sampel daging sapi dan 15 sampel daging babi, didapatkan hasil seperti yang ditujukan Tabel 6.3. Agar dapat menentukan nilai akurasi system menggunakan persamaan 6.1.

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

Tabel 6.3 Data Uji dan hasil pengujian Jaringan Syaraf Tiruan

No.	R	G	B	PPM	Klasifikasi Asli	Klasifikasi Sistem	Kesesuaian
1.	127	89	77	0,42	Daging Sapi	Daging Sapi	Sesuai
2.	109	80	71	0,31	Daging Sapi	Daging Sapi	Sesuai
3.	123	85	77	0,20	Daging Sapi	Daging Sapi	Sesuai
4.	131	93	84	0,33	Daging Sapi	Daging Sapi	Sesuai
5.	118	85	71	0,13	Daging Sapi	Daging Sapi	Sesuai
6.	123	85	77	0,23	Daging Sapi	Daging Sapi	Sesuai
7.	114	85	74	0,09	Daging Sapi	Daging Sapi	Sesuai
8.	118	80	74	0,31	Daging Sapi	Daging Sapi	Sesuai
9.	114	76	71	0,16	Daging Sapi	Daging Sapi	Sesuai
10.	140	97	87	0,21	Daging Sapi	Daging Babi	Tidak Sesuai
11.	118	80	74	0,21	Daging Sapi	Daging Sapi	Sesuai
12.	127	85	74	0,14	Daging Sapi	Daging Sapi	Sesuai
13.	131	89	77	0,21	Daging Sapi	Daging Sapi	Sesuai
14.	114	80	71	0,20	Daging Sapi	Daging Sapi	Sesuai
15.	109	76	68	0,22	Daging Sapi	Daging Sapi	Sesuai
16.	136	93	84	0,02	Daging Babi	Daging Babi	Sesuai
17.	140	97	80	0,06	Daging Babi	Daging Babi	Sesuai
18.	145	97	84	0,05	Daging Babi	Daging Babi	Sesuai
19.	145	102	87	0,15	Daging Babi	Daging Sapi	Tidak Sesuai
20.	136	93	84	0,05	Daging Babi	Daging Babi	Sesuai
21.	136	93	80	0,04	Daging Babi	Daging Babi	Sesuai
22.	171	110	99	0,05	Daging Babi	Daging Babi	Sesuai
23.	162	110	96	0,05	Daging Babi	Daging Babi	Sesuai
24.	145	102	87	0,08	Daging Babi	Daging Babi	Sesuai
25.	162	110	93	0,07	Daging Babi	Daging Babi	Sesuai
26.	184	127	105	0,11	Daging Babi	Daging Babi	Sesuai
27.	180	123	105	0,00	Daging Babi	Daging Babi	Sesuai
28.	184	123	105	1,07	Daging Babi	Daging Sapi	Tidak Sesuai

29.	184	127	105	0,06	Daging Babi	Daging Babi	Sesuai
30.	180	123	105	0,10	Daging Babi	Daging Babi	Sesuai

Pada Tabel 6.1 terlihat dari 30 sampel daging, didapatkan hasil klasifikasi yang dilakukan oleh sistem tidak sesuai sebanyak tiga kali. Hasil klasifikasi sistem yang tidak sesuai dapat terjadi karena beberapa faktor, seperti halnya adanya intervensi cahaya dan juga daging yang terlalu lama dibiarkan dalam udara bebas sehingga mempengaruhi warna dan bau dari daging. Dengan 27 data sesuai dari 30 sampel, maka didapatkan hasil akurasi sistem pembeda daging sebesar 90%. Proses perhitungannya dapat dilihat di bawah:

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ Data}\ x\ 100\%$$

$$= \frac{30 - 3}{30}\ x\ 100\%$$

$$= \frac{27}{30}\ x\ 100\%$$

$$= 90\%$$

6.6 Pengujian Waktu Komputasi Pemrosesan Sistem

Pemrosesan akan dilakukan oleh sistem, sehingga diperlukan pengujian agar diketahui performansi dari sistem dengan menggunakan waktu sebagai parameter pengujian. Dimana waktu dari sistem menjalankan proses klasifikasi akan diperhatikan, baik saat melakukan klasifikasi pelatihan data ataupun pada saat melakukan klasifikasi prediksi data.

6.6.1 Durasi Pelatihan Data

Pada pengujian pelatihan data, waktu yang dibutuhkan sistem dalam melaksanakan pelatihan data akan menjadi fokus dalam pengujian ini. Data yang akan dilatih dimasukkan dalam kode program Matlab yang kemudian diterapkan pada sistem. Dalam pengujian ini, peneliti melakukan kombinasi nilai dari *learning rate* dan *epoch*. Untuk pengujian ini, *learning rate* (tingkat pembelajaran) yang digunakan sebesar 0,1, 0,01, dan 0,001. Sementara nilai *epoch* yang akan digunakan sebesar 1000, 10000, dan 100000.

6.6.1.1 Tujuan Penelitian

Tujuan dilakukan penelitian ini adalah untuk mencari tahu waktu yang dibutuhkan oleh sistem untuk menyelesaikan proses pelatihan data memanfaatkan algoritma *Backpropagation.*

6.6.1.2 Prosedur Pengujian

a) Menjalankan Matlab.
b) Menjalankan kode program pelatihan data.

c) Merubah besaran nilai *learning rate* dan *epoch* sesuai dengan yang telah ditentukan.
d) Menyisipkan fungsi *tic* sebagai awalan dan *toc* sebagai akhiran agar mendapat waktu sekarang dalam satuan detik.
e) Menjalankan kode program.
f) Menyimak selisih waktu proses sistem.

6.6.1.3 Hasil dan Analisis Pengujian

Hasil pengujian dapat dilihat dalam tabel 6.4 di bawah:

Tabel 6.4 Pengujian Durasi Pelatihan Data

No.	Learning rate	Epoch	Durasi (s)
1.	0,1	1000	1
2.	0,1	10000	2
3.	0,1	100000	2
4.	0,01	1000	3
5.	0,01	10000	2
6.	0,01	100000	2
7.	0,001	1000	2
8.	0,001	10000	2
9.	0,001	100000	2

Pada tabel 6.4 disimpulkan bahwa besar atau kecilnya nilai *learning rate* dan nilai *epoch* tidak berpengaruh terhadap durasi waktu pelatihan.

6.6.2 Durasi Prediksi Data

Pengujian prediksi data ini akan menguji durasi waktu yang diperlukan oleh sistem ketika proses prediksi data.

6.6.2.1 Tujuan Penelitian

Pengujian ini dilaksanakan untuk mencari tahu durasi waktu yang diperlukan oleh sistem untuk memperoleh hasil klasifikasi antara daging sapi dan daging babi menggunakan JST.

6.6.2.2 Prosedur Pengujian

a) Membuka aplikasi Arduino IDE.
b) Memasukkan *syntax millis*() pada kode program Arduino sebagai penghitung selisih waktu mulai dengan waktu selesai dan menampilkannya dalam *millisecond.*
c) Menyambungkan mikrokontroler Arduino pada perangkat laptop.

d) Mengunggah kode program klasifikasi JST ke Arduino Nano.
e) Memilih serial monitor pada Arduino IDE.
f) Mengamati waktu berdasarkan nilai yang tampil di *serial monitor*.

6.6.2.3 Hasil dan Analisis Pengujian

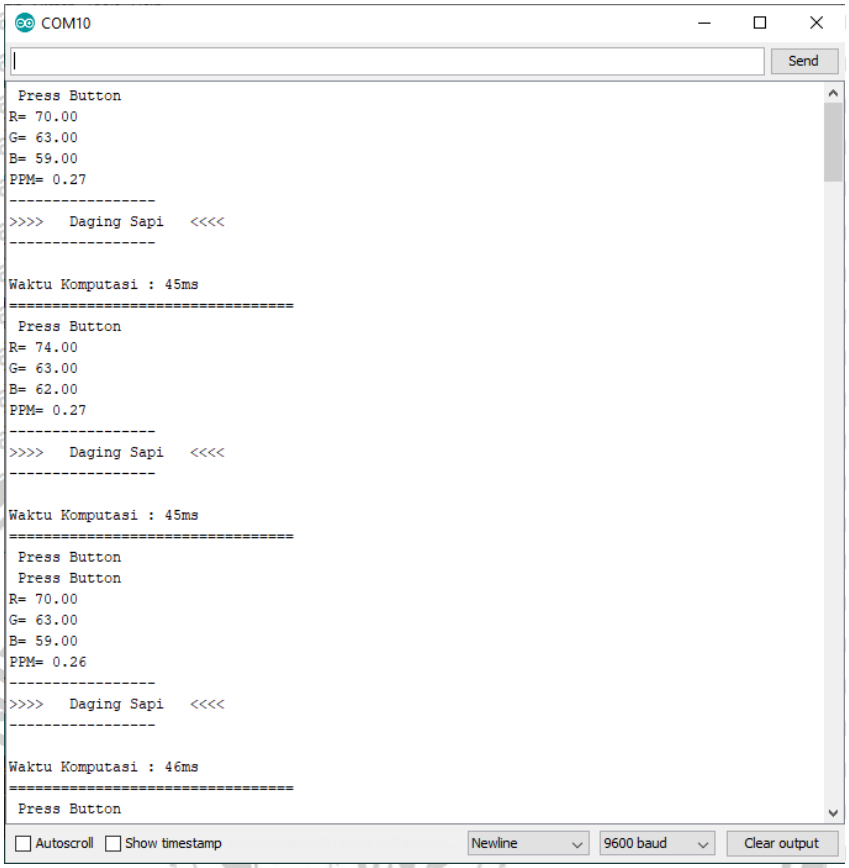
Merujuk pada pengujian terhadap 20 sampel percobaan, hasilnya terlihat pada Tabel 6.5.

Tabel 6.5 Pengujian Durasi Prediksi

No.	Urutan Pengujian	Waktu Komputasi (ms)
1.	Pertama	45
2.	Kedua	45
3.	Ketiga	45
4.	Keempat	45
5.	Kelima	46
6.	Keenam	45
7.	Ketujuh	46
8.	Kedelapan	45
9.	Kesembilan	46
10.	Kesepuluh	46
11.	Kesebelas	45
12.	Kedua belas	45
13.	Ketiga belas	45
14.	Kempat belas	45
15.	Kelima belas	46
16.	Keenam belas	45
17.	Ketujuh belas	45
18.	Kedelapan belas	46
19.	Kesembilan belas	45
20.	Kedua puluh	45
Rata - rata		45,3

Melalui 20 kali pengujian, didapatkan hasil waktu komputasi dalam melakukan klasifikasi pada sistem pembeda jenis daging antara daging sapi dan

babi dengan rata-rata waktu 1048,55ms atau 1,04s. Gambar 6.6 di bawah menunjukkan hasil dari pengujian waktu komputasi sistem.



Gambar 6.6 Hasil Pengujian Durasi Prediksi

BAB 7 PENUTUP

Bab ini berisikan mengenai penarikan kesimpulan terhadap tahapan-tahapan yang sebelumnya telah dilakukan. Serta saran oleh peneliti dengan harapan saran tersebut dapat berguna dalam pengembangan penelitian serupa untuk penelitian-penelitian selanjutnya.

7.1 Kesimpulan

Merujuk pada rumusan masalah yang diajukan saat akan melakukan penelitian serta merujuk pada hasil pengujian yang telah dilaksanakan, kesimpulan yang didapat diantaranya:

1. Sensor warna TCS3200 yang difungsikan sebagai pembaca warna RGB pada daging dapat melakukan pembacaan nilai RGB dengan rata-rata error sebesar 1,73% bila sensor tersebut ditempatkan tepat di bawah objek yang dideteksi(daging). Sehingga dapat disimpulkan bahwa sensor telah bekerja dengan sangat baik karena tetap mampu membaca adanya perubahan warna meskipun perubahan nilainya kecil. Kemudian sensor gas MQ135 yang ditempatkan tepat di atas objek yang dideteksi (daging) dapat membaca dengan baik nilai kandungan amonia yang dihasilkan oleh aroma daging. Hal ini dapat dilihat dari sensor yang berfungsi dan bekerja sesuai karakteriktiknya dalam datasheet sensor gas MQ135 yaitu nilai hasil pembacaan sensor akan berbanding lurus dengan nilai tegangan keluaran yang dihasilkan. Jika nilai ppm yang dihasilkan semakin besar, maka akan semakin besar juga tegangan keluaran dari sensor. Nilai korelasi antar kedua variabel tersebut mencapai 97,3%. Selanjutnya yang terakhir yaitu modul *Bluetooth* HC-05 dapat dilakukan *pairing* dengan ponsel Android serta menampilkan hasil pada aplikasi yang telah dibuat sebelumnya sesuai dengan data yang terdapat dalam *serial monitor* Arduino IDE, sehingga dapat ditarik kesimpulan bahwa modul *Bluetooth* dapat bekerja dengan baik.
2. Pada penelitian ini telah dibuat sistem pembeda daging sapi dan daging babi berdasarkan warna dan kadar amonia menggunakan jaringan syaraf tiruan berbasis android. Keseluruhan komponen yang terdapat dalam sistem ini baik alat maupun metode penelitian yang diterapkan mampu berjalan sesuai harapan. Hal ini dibuktikan dengan berhasilnya sistem mengklasifikasikan jenis daging sapi dan babi.
3. Hasil akurasi yang didapatkan oleh sistem pembeda jenis daging sapi dan daging babi berdasarkan warna dan kadar amonia menggunakan jaringan syaraf tiruan berbasis android yang diuji sebanyak 30 kali pengujian dengan data latih sebanyak 60 sampel yang terdiri dari 30 sampel daging sapi dan 30 sampel daging babi mendapatkan prosentase keberhasilan sebesar 90%. Dengan didapatkannya nilai prosesntase keberhasilan tersebut, maka dapat dikatakan sistem telah bekerja dengan baik dalam membedakan jenis daging sapi dan babi.

4. Sistem ini membutuhkan waktu pemrosesan dengan kecepatan rata-rata sebesar 45,3ms untuk membedakan jenis daging sapi dan babi berdasarkan 20 kali pengujian. Dengan didapatkannya waktu tersebut, maka dapat dikatakan performa dari sistem dalam melakukan klasifikasi tergolong cepat.

7.2 Saran

Peneliti menambahkan saran agar bisa digunakan sebagai dasar dalam mengembangkan penelitian ini atau penelitian sejenis selanjutnya, diantaranya:

1. Memanfaatkan sensor lain yang mempunyai fungsi khusus dalam mendeteksi aroma daging sehingga dimungkinkan akan lebih meningkatkan hasil akurasi dari sistem.
2. Menambahkan baterai sebagai sumber tegangan sehingga sistem dapat bekerja secara *portable.*
3. Membuat sistem pada *box* yang berwarna gelap sehingga meminimalisir intervensi dari cahaya luar dan hasil pembacaan sensor warna lebih maksimal.

DAFTAR PUSTAKA

Arduino, 2019. *Arduino Nano Rev3.* [Online] Available at: https://store.arduino.cc/usa/arduino-uno-rev3 [Accessed 20 April 2019].

Ariyanto, M. & Caesarendra, W., 2011. *Panduan Belajar Mandiri MATLAB.* Jakarta: PT Elex Media Komputindo.

Cahyono, B., 2013. Penggunaan Software Matrix Laboratory (MATLAB) dalam Pembelajaran Aljabar Linier. *PHENOMENON,* Volume 1, pp. 45-62.

Hadi, R. W., 2011. Perancangan Alat Pendeteksi Kualitas Daging Sapi Berdasarkan Warna dan Bau Berbasis Mikrokontroler Atmega32 Menggunakan Logika Fuzzy. *Transmisi,* pp. 21-26.

Hartono, A. F., 2012. Implementasi Jaringan Syaraf Tiruan Backpropagation sebagai Sistem Pengenalan Citra Daging Sapi dan Citra Daging Babi. *UNNES Journal of Mathematics,* pp. 125-130.

Inc, T., 2009. TCS3200,TCS3210 Progammable Color Light to Frequency Converter. *Texas Advance Optoelectric Solutions Inc,* pp. 1-14.

Inventor, M. A., 2012. *Anyone Can Build Apps That Impact the World.* [Online] Available at: https://appinventor.mit.edu/explore/about-us.html [Accessed 10 Juni 2019].

Juansyah, A., 2015. Pembangunan Aplikasi Child Tracker Berbasis Asisted-Global Positioning System (A-GPS) dengan Platform Android. *Jurnal Ilmiah Komputer dan Informatika(KOMPUTA),* Volume 1, pp. 1-8.

Lamidi, 2018. Sistem Pendeteksi Penyakit Diabetes Melitus dan Tingkat Dehidrasi Berdasarkan Kondisi Urin dengan Metode Jaringan Saraf Tiruan Berbasis Aplikasi Android. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* Volume 3, pp. 9255-9264.

Maru'ao, D. O., 2010. Neural Network Implementation in Foreign Exchange Kurs Prediction. *Gunadarma University,* pp. 1-19.

MQ135_datasheet, 2016. *Technical Data MQ-135 Gas Sensor,* s.l.: Hanwei Electronics co.,ltd.

Muhammadiy, W. & Fahmi, F., 2016. Mobile Application to Differentiate Flesh Meat between Beef and Pork. *IEEE.*

Nafiasari, N. A. & Handayani, A. M., 2018. Penganalisis Kesegaran Daging Sapi dan Daging Babi Mentah Berdasarkan Klasifikasi Warna dan Kelembaban. *Jurnal TeknoSains,* Volume 8, pp. 1-88.

Nurmila, N., Sugiharto, A. & Sarwoko, E. A., n.d. Algoritma Backpropagation Neural Network untuk Pengenalan Pola Karakter Huruf Jawa. *Jurnal Masyarakat Informatika,* Volume 1, pp. 1-10.

Rahmana, M. A. F., 2019. Sistem Deteksi Lama Waktu Penyimpanan Daging Ayam berdasarkan Warna dan Kadar Amonia Berbasis Sensor TCS3200 dan MQ135 dengan Metode Jaringan Syaraf Tiruan. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* Volume 3, pp. 1908-1916.

Rahma, N. I., 2016. KLASIFIKASI POLA RASA DAGING SAPI DAN DAGING BABI BERBASIS ELECTRONIC TONGUE DENGAN 17 ARRAY SENSOR MENGGUNAKAN METODE PRINCIPLE COMPONENT ANALYSIS(PCA) DAN CLUSTER ANALYSIS(CA).

Riasari, J. R., 2014. Perbedaan Karakteristik Daging Sapi dan Daging Babi. *Makalah I Higiene Daging,* pp. 1-8.

Riniarsi, D., 2017. OUTLOOK DAGING SAPI. In: Jakarta: Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal Kementrian Pertanian, p. 31.

Simamora, J., 2017. Rancang Bangun Sistem Pendeteksi Kesegaran Daging Berdasarkan Sensor Bau dan Warna.

Siregar, R. A., 2017. *Polisi Ciduk Pengoplos Daging Sapi dan Babi di Lubuklinggau,* Lubuklinggau Sumatera Selatan: detikNews.

LAMPIRAN A DATA LATIH

No.	R	G	B	Amonia (PPM)	Jenis Daging
1.	123	85	74	0.02	Daging Sapi
2.	123	85	77	0.05	Daging Sapi
3.	118	80	71	0.01	Daging Sapi
4.	118	89	77	0.01	Daging Sapi
5.	131	97	87	0.02	Daging Sapi
6.	118	85	74	0.08	Daging Sapi
7.	118	89	80	0.06	Daging Sapi
8.	105	76	71	0.01	Daging Sapi
9.	109	80	74	0.01	Daging Sapi
10.	114	80	74	0.01	Daging Sapi
11.	114	85	77	0.01	Daging Sapi
12.	118	85	74	0.01	Daging Sapi
13.	114	85	74	0.01	Daging Sapi
14.	118	85	74	0.01	Daging Sapi
15.	123	89	80	0.02	Daging Sapi
16.	127	89	80	0.03	Daging Sapi
17.	127	89	80	0.03	Daging Sapi
18.	114	80	74	0.02	Daging Sapi
19.	127	85	71	0.01	Daging Sapi
20.	123	85	77	0.05	Daging Sapi
21.	109	80	71	0.01	Daging Sapi
22.	123	85	74	0.01	Daging Sapi
23.	118	80	71	0.01	Daging Sapi
24.	123	85	77	0.01	Daging Sapi
25.	109	76	74	0.01	Daging Sapi

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA
Universitas Brawijaya

26.	131	83	84	0.04	Daging Sapi
27.	118	89	77	0.04	Daging Sapi
28.	118	85	74	0.03	Daging Sapi
29.	105	80	68	0.03	Daging Sapi
30.	118	85	74	0.03	Daging Sapi
31.	131	93	84	0.01	Daging Babi
32.	127	93	80	0.01	Daging Babi
33.	127	93	84	0.01	Daging Babi
34.	131	93	80	0.01	Daging Babi
35.	123	89	77	0.01	Daging Babi
36.	136	97	87	0.01	Daging Babi
37.	136	93	84	0.01	Daging Babi
38.	131	93	80	0.01	Daging Babi
39.	136	97	84	0.01	Daging Babi
40.	145	97	84	0.01	Daging Babi
41.	149	97	84	0.01	Daging Babi
42.	162	128	108	0.01	Daging Babi
43.	158	119	108	0.01	Daging Babi
44.	153	106	90	0.01	Daging Babi
45.	136	97	87	0.01	Daging Babi
46.	127	89	77	0.01	Daging Babi
47.	145	102	87	0.01	Daging Babi
48.	136	93	84	0.01	Daging Babi
49.	153	97	87	0.01	Daging Babi
50.	131	97	87	0.01	Daging Babi
51.	149	102	87	0.02	Daging Babi
52.	140	102	90	0.02	Daging Babi
53.	145	119	105	0.02	Daging Babi

54.	149	119	112	0.02	Daging Babi
55.	140	110	96	0.02	Daging Babi
56.	131	102	90	0.02	Daging Babi
57.	127	89	80	0.02	Daging Babi
58.	123	93	77	0.02	Daging Babi
59.	131	93	84	0.02	Daging Babi
60.	123	89	90	0.02	Daging Babi

LAMPIRAN B KODE PROGRAM KESELURUHAN PADA ARDUINO

Baris	Kode Program
1	`#include <Math.h>`
2	`#include <TimerOne.h>`
3	
4	`unsigned long time1, time2, waktu;`
5	
6	
7	`//inisalisasi variabel array untuk pengambilan`
8	`data sensor`
9	`float baca[4];`
10	`float masuk [4];`
11	`int button;`
12	`//pin Color Sensor`
13	`#define S0  6`
14	`#define S1  5`
15	`#define S2  3`
16	`#define S3  4`
17	`#define OUT 2`
18	
19	`bool active = false;`
20	`int   g_count = 0;     // count the frequecy`
21	`int   g_array[3];      // store the RGB value`
22	`int   g_flag = 0;      // filter of RGB queue`
23	`float g_SF[3];       // save the RGB Scale factor`
24	
25	`int   R = 0;`
26	`int   G = 0;`
27	`int   B = 0;`
28	
29	`//jst`
30	`//------//`
31	`//bobot_hidden`
32	`float v11 = 0.0891424363801548;`
33	`float v12 = 0.524631480471023;`
34	`float v13 = 2.27439785541971;`
35	`float v14 = 0.152071481901794;`
36	`float v15 = -0.596967404377651;`
37	`float v21 = -0.198548446784767;`
38	`float v22 = 0.423797571615013;`
39	`float v23 = 1.35488894224097;`
40	`float v24 = 0.232675713121640;`
41	`float v25 = 0.462921799455575;`
42	`float v31 = 0.427289162522732;`
43	`float v32 = 0.143242396878870;`
44	`float v33 = 1.65694734501821;`

```
45  float v34 = 0.0162114894204103;
46  float v35 = 0.722478211057660;
47  float v41 = 96.4898924418868;
48  float v42 = 10.2130041229696;
49  float v43 = -101.849779629083;
50  float v44 = -70.2804244623574;
51  float v45 = -7.48844031236359;
52  //bias_hidden
53  float v01 = -5.73943435544124;
54  float v02 = -11.2721842056059;
55  float v03 = 9.26738260010132;
56  float v04 = -40.8879685123589;
57  float v05 = -27.5913691401162;
58  //bobot_keluaran
59  float w11 = 7.20968252753860;
60  float w12 = 3.16237465585140;
61  float w13 = 6.60960365240546;
62  float w14 = -80.8345568045453;
63  float w15 = 58.2682756364005;
64  //bias_keluaran
65  float w01 = 2.347013859508l5;
66
67  //fungsi logsigmoid jst
68  float sigmoid (float x) {
69    return 1 / (1 + exp(-x));
70  }
71
72
73  //gas sensor
74  const int gasSensor = 0;
75  #define Ro 22000
76  float  MQ135_SCALINGFACTOR  =  37.58805473;  //for
77  NH3
78  float MQ135_EXPONENT = -3.235365807; //for NH3
79
80
81  // Init TCS3200 and setting Frequency.
82  void TCS_Init() {
83    pinMode(S0, OUTPUT);
84    pinMode(S1, OUTPUT);
85    pinMode(S2, OUTPUT);
86    pinMode(S3, OUTPUT);
87    pinMode(OUT, INPUT);
88
89    digitalWrite(S0,  LOW);    //  OUTPUT  FREQUENCY
90  SCALING 2% x10
91    digitalWrite(S1, HIGH);
92  }
93
```

```
94  // Select the filter color
95  void TCS_FilterColor(int Level01, int Level02) {
96    if (Level01 != 0)
97      Level01 = HIGH;
98
99    if (Level02 != 0)
100     Level02 = HIGH;
101
102   digitalWrite(S2, Level01);
103   digitalWrite(S3, Level02);
104 }
105
106 void TCS_Count() {
107   g_count ++ ;
108 }
109
110 void TCS_Callback() {
111   switch (g_flag) {
112     case 0:
113       TCS_WB(LOW, LOW);                    //Filter
114 without Red
115       break;
116     case 1:
117       g_array[0] = g_count * 10;
118       TCS_WB(HIGH, HIGH);                  //Filter
119 without Green
120       break;
121     case 2:
122       g_array[1] = g_count * 10;
123       TCS_WB(LOW, HIGH);                   //Filter
124 without Blue
125       break;
126     case 3:
127       g_array[2] = g_count * 10;
128       TCS_WB(HIGH, LOW);                 //Clear(no
129 filter)
130       break;
131     default:
132       g_count = 0;
133       break;
134   }
135 }
136
137 void TCS_WB(int Level0, int Level1) {
138   g_count = 0;
139   g_flag ++;
140   TCS_FilterColor(Level0, Level1);
141   Timer1.setPeriod(100000);
142 }
```

```
143
144  void setup() {
145    pinMode(8, INPUT);
146    TCS_Init();
147    Serial.begin(9600);
148    Timer1.initialize();                    // nilai
149  default 1s
150    Timer1.attachInterrupt(TCS_Callback);
151  //memanggil fungsi TCS_Callback sebagai fungsi
152  interrupt
153    attachInterrupt(0,    TCS_Count,    RISING);
154  //menjalankan fungsi TCS_Count bila pin INT0
155  mengalami perubahan dari LOW ke HIGH
156    delay(500);
157
158    g_SF[0] = 255.0 / 579.973491 ;          //R Scale
159  factor
160    g_SF[1] = 255.0 / 598.707950 ;          //G Scale
161  factor
162    g_SF[2] = 255.0 / 819.271348 ;          //B Scale
163  factor
164
165    delay(500);
166  }
167
168  int state = 1;
169  void loop() {
170    time1 = millis();
171    bool isProcess = true;
172    button = digitalRead (8);
173
174    if (state == 1) {
175      if (button == HIGH) {
176        active = true;
177        state = 0;
178      }
179    }
180
181    if (state == 0) {
182      if (button == LOW) {
183        active = true;
184        state = 1;
185      }
186    }
187
188
189    if (active) {
190      read_sensor(baca);
191      masuk [0] = baca [0];
```

```
192       masuk [1] = baca [1];
193       masuk [2] = baca [2];
194       masuk [3] = baca [3];
195       //active = false;
196       while (isProcess) {
197         button =  digitalRead(8);
198
199         //miliis
200         time2 = millis();
201         waktu = time2 - time1;
202         Serial.print("Waktu Komputasi : ");
203         Serial.print(waktu);
204         Serial.println ("ms");
205         waktu = 0;
206         Serial.println
207   ("==================================");
208
209
210   //      if (button != HIGH) {
211             isProcess = false;
212   //      Serial.println(" Press Button ");
213   //      delay(1000);
214   //            }
215          }
216          active = false;
217       }else{
218           Serial.println(" Press Button ");
219       delay(1000);
220       }
221   }
222
223   void read_sensor(float arraybaca[4]) {
224     //view PPM Amonia
225     float analog = analogRead(gasSensor);
226     float voltage = (analog / 1024) * 5;
227     float Ra = ((Ro / voltage * 5) - Ro);
228     float ppm = MQ135_SCALINGFACTOR * pow((Ra /
229   Ro), MQ135_EXPONENT);
230     arraybaca[3] = ppm;
231
232     //tcs_sensor
233     g_flag = 0;
234     R = g_array[0] * g_SF[0];
235     G = g_array[1] * g_SF[1];
236     B = g_array[2] * g_SF[2];
237
238     if (R > 255) {
239       R = 255;
240     }
```

```
241     if (G > 255) {
242       G = 255;
243     }
244     if (B > 255) {
245       B = 255;
246     }
247
248     arraybaca[0] = R;
249     arraybaca[1] = G;
250     arraybaca[2] = B;
251     arraybaca[3] = ppm;
252
253     //Serial.print("masukan[0]= ");
254     Serial.print("R= ");
255     Serial.println(arraybaca[0]);
256     //Serial.print(" | ");
257     //Serial.print("masukan[1]= ");
258     Serial.print("G= ");
259     Serial.println(arraybaca[1]);
260     //Serial.print(" | ");
261     //Serial.print("masukan[2]= ");
262     Serial.print("B= ");
263     Serial.println(arraybaca[2]);
264     //Serial.println(" | ");
265     //Serial.print("masukan[3]= ");
266     Serial.print("PPM= ");
267     Serial.println(arraybaca[3]);
268   //  Serial.print("volt= ");
269   //  Serial.println(voltage);
270     //Serial.println(" | ");
271     Serial.println("------------------");
272
273     //nilai Penjumlahan Neuron Layer 1
274     float N1layer1 = (arraybaca [0] * v11) +
275   (arraybaca [1] * v21) + (arraybaca [2] * v31) +
276   (arraybaca [3] * v41) + v01 ;
277     float N2layer1 = (arraybaca [0] * v12) +
278   (arraybaca [1] * v22) + (arraybaca [2] * v32) +
279   (arraybaca [3] * v42) + v02 ;
280     float N3layer1 = (arraybaca [0] * v13) +
281   (arraybaca [1] * v23) + (arraybaca [2] * v33) +
282   (arraybaca [3] * v43) + v03 ;
283     float N4layer1 = (arraybaca [0] * v14) +
284   (arraybaca [1] * v24) + (arraybaca [2] * v34) +
285   (arraybaca [3] * v44) + v04 ;
286     float N5layer1 = (arraybaca [0] * v15) +
287   (arraybaca [1] * v25) + (arraybaca [2] * v35) +
288   (arraybaca [3] * v45) + v05 ;
289
```

```
290     //Nilai Aktivasi Neuron Layer 1
291     N1layer1 = sigmoid(N1layer1);
292     N2layer1 = sigmoid(N2layer1);
293     N3layer1 = sigmoid(N3layer1);
294     N4layer1 = sigmoid(N4layer1);
295     N5layer1 = sigmoid(N5layer1);
296
297     //Nilai Penjumlahan Neuron Layer 2
298     float N1layer2 = (N1layer1 * w11) + (N2layer1
299   * w12) + (N3layer1 * w13) + (N4layer1 * w14) +
300   (N5layer1 * w15) + w01 ;
301     //Serial.print(N1layer2);
302     //Serial.print("|");
303
304     //fungsi hardlimit
305     if (N1layer2 > 0) {
306   //    delay (1000);
307   //    N1layer2 = 1;
308   //    Serial.println (N1layer2);
309       Serial.print (">>>>");
310       Serial.print ("   Daging Sapi   ");
311       Serial.println ("<<<<");
312       Serial.println("-----------------");
313     } else if (N1layer2 < 0) {
314   //    delay (1000);
315   //    N1layer2 = 0;
316   //   Serial.println (N1layer2);
317       Serial.print (">>>>");
318       Serial.print ("   DAGING BABI   ");
319       Serial.println ("<<<<");
320       Serial.println("-----------------");
321     }
322   //  delay (1000);
323     Serial.println ("");
324   }
```