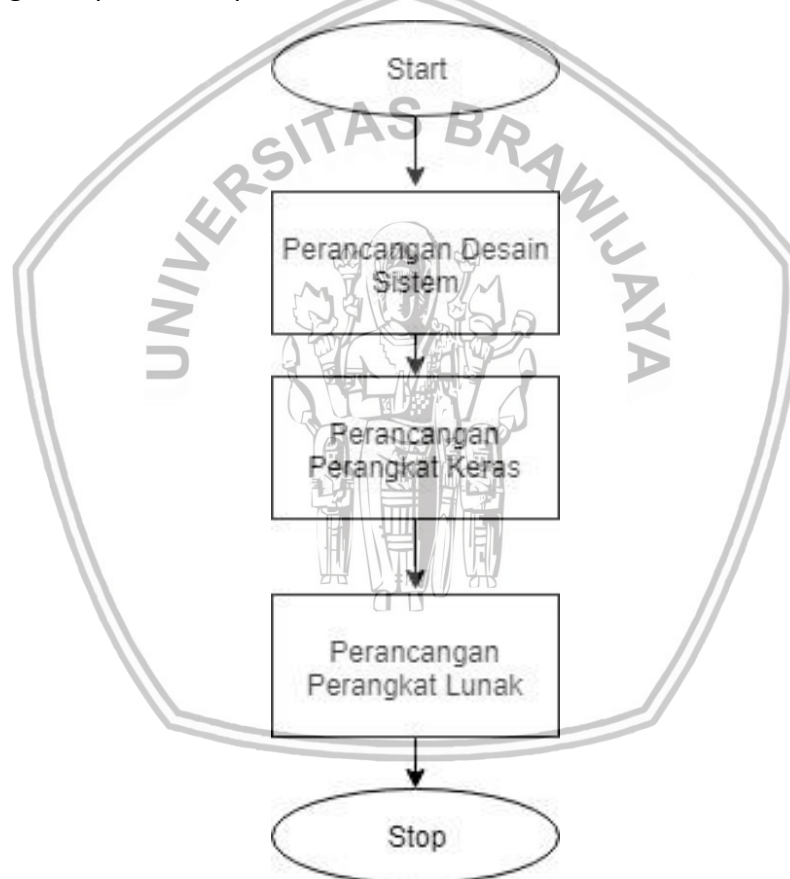


BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan menjelaskan mengenai perancangan dan implementasi dari sistem klasifikasi telur ayam kampung dan ayam negeri menggunakan metode naive bayes. Dalam bab ini perancangan terbagi atas perancangan perangkat keras dan perancangan perangkat lunak dari sistem dan implementasi perangkat keras dan perangkat lunak dari sistem yang telah dirancang.

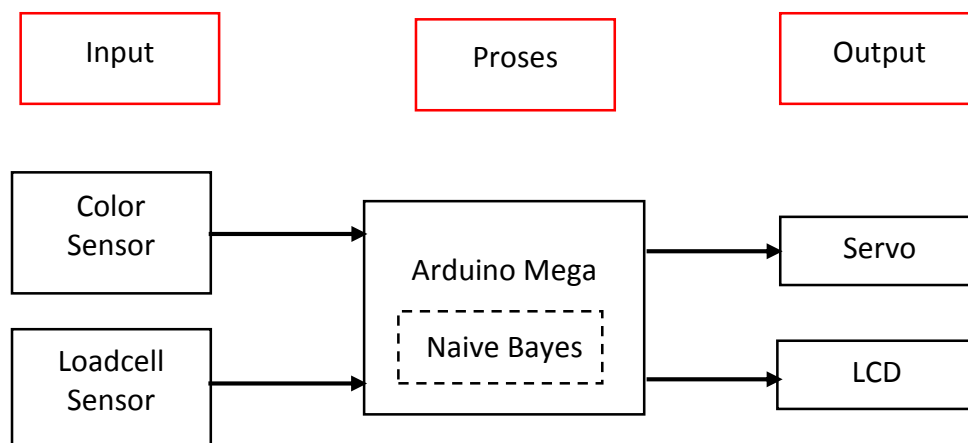
5.1 Perancangan sistem

Perancangan sistem dibagi menjadi tiga bagian, yaitu perancangan desain sistem, perancangan perangkat keras, dan perancangan perangkat lunak. Alur perancangan dapat dilihat pada Gambar 5.1.



Gambar 5.1 Alur Perancangan Sistem Klasifikasi Telur ayam

Pada Gambar 5.1 menjelaskan alur perancangan sistem, penelitian diawali dengan perancangan desain keseluruhan sistem, dan cara kerja setiap komponen yang digunakan. Selanjutnya merancang perangkat keras yang digunakan pada sistem klasifikasi telur ayam ini. Setelah itu melakukan perancangan perangkat lunak dengan menggunakan bahasa C++, pada tahap ini program dibuat untuk kontrol keseluruhan sistem. Diagram perancangan sistem dapat dilihat pada Gambar 5.2.



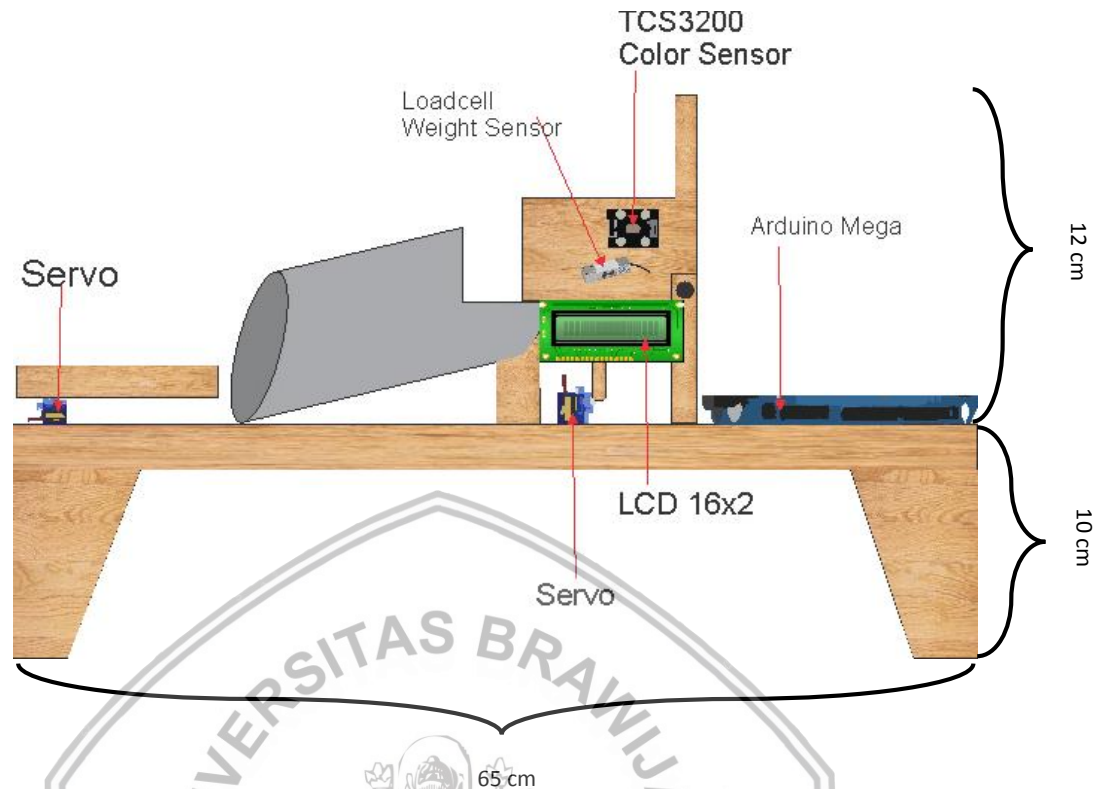
Gambar 5.2 Diagram Perancangan Sistem Klasifikasi Telur Ayam

Pada Gambar 5.2 diagram perancangan sistem terbagi menjadi 3 bagian, yaitu *Input*, *Proses*, dan *Output*. Pada bagian input, terdapat 2 sensor yang digunakan yaitu, color sensor yang berfungsi untuk mengetahui nilai warna dengan mengukur nilai R, G, dan B, serta Loadcell sensor yang berguna untuk mengukur nilai berat dari telur ayam. Pada bagian proses, terdapat mikrokontroler Arduino Mega sebagai otak utama sistem, selain itu Arduino mega dimanfaatkan untuk melakukan proses klasifikasi Naive bayes. Pada bagian output sistem berupa hasil klasifikasi telur ayam yang ditampilkan pada LCD, dan gerakan servo untuk nantinya dikelompokkan antara telur ayam kampung dan telur ayam negeri.

5.1.1 Perancangan Fisik Alat Pada Sistem Klasifikasi Telur Ayam

Dalam melakukan desain dari Sistem klasifikasi telur ayam ini peletakan tiap-tiap komponen merupakan hal penting. Pembuatan desain diawali dengan membuat gambaran dengan menggunakan program corel draw. Desain peletakan sensor, servo, LCD dan mikrokontroler dapat dilihat pada

Color sensor TCS3200 diletakkan pada wadah bagian atas untuk mempermudah mendeteksi warna telur ayam yang ditaruh oleh user, sedangkan Loadcell sensor diletakkan didalam wadah, sehingga mempermudah user untuk mengoperasikan sistem yang telah dibuat. Untuk peletakan LCD berada pada bagian samping alat untuk mempermudah melihat hasil klasifikasi Telur Ayam. Lokasi peletakan servo terdapat pada ujung dan bagian tengah. Pada bagian ujung servo berfungsi sebagai jalan untuk mengelompokkan jenis ayam sesuai dengan hasil klasifikasi, sedangkan servo yang berada pada tengah berfungsi untuk menahan wadah, dan menggerakkan wadah.

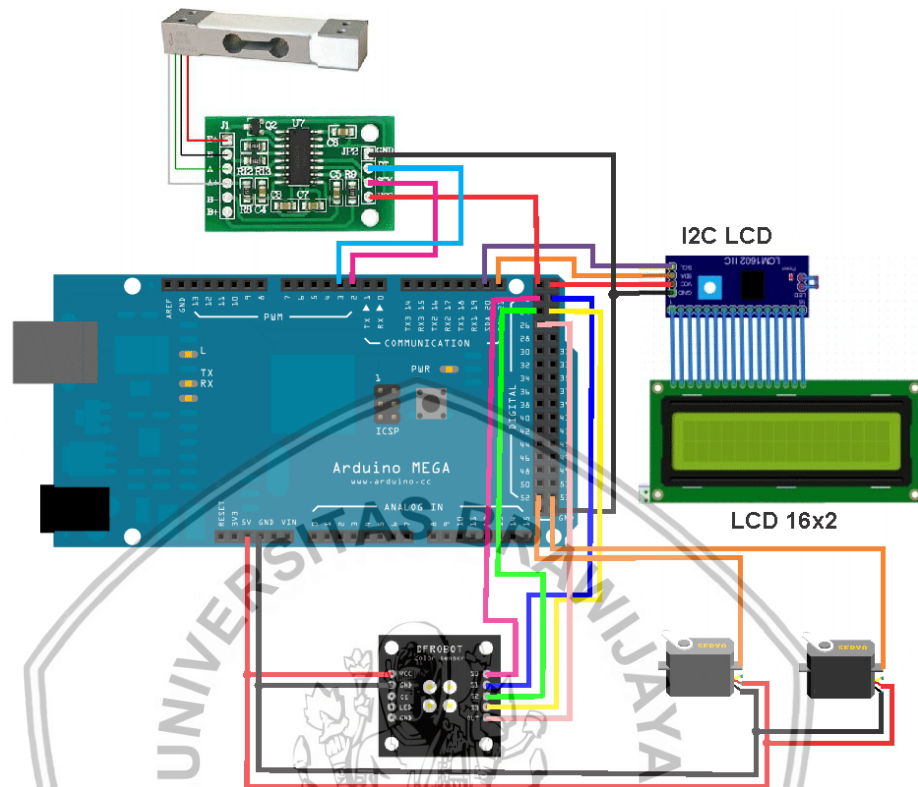


Gambar 5.3 Perancangan Keseluruhan Sistem Klasifikasi Ayam

5.1.2 Perancangan Perangkat Keras

Perancangan perangkat keras dilakukan berdasarkan analisis kebutuhan perangkat keras serta spesifikasi dari masing-masing dari perangkat keras agar dapat membangun sistem sesuai dengan yang diharapkan. Berdasarkan Gambar 5.4 yang merupakan blok diagram yang menjelaskan secara rinci hubungan skematik pin-pin yang digunakan antara tiap komponen perangkat keras, Color sensor TCS3200 dan Loadcell sensor merupakan bagian input yang dihubungkan dengan Mikrokontroler Arduinno Mega, nantinya Arduino Mega berguna untuk pengolah data untuk ditampilkan hasil olahan tersebut melalui LCD 16x2 yang sudah terhubung dengan I2C module. Servo pada sistem ini digunakan untuk menahan dan mengklasifikasikan telur ayam yang telah diletakkan sebelumnya. Skematik diagram keseluruhan perancangan perangkat keras sistem klasifikasi telur ayam ditunjukkan pada Gambar 5.4.

Skematik Perancangan pada Gambar 5.4 Semua komponen diantaranya, Color sensor TCS3200, Loadcell sensor, LCD 16x2 yang terhubung dengan I2C LCD, dan Servo SG-90, terhubung pada sebuah mikrokontroller Arduino Mega dikarenakan mampu mencangkup keseluruhan sistem pada penelitian ini.



Gambar 5.4 Skematik Perancangan Keseluruhan Sistem Klasifikasi Telur Ayam

5.1.2.1 Perancangan Rangkaian Color Sensor TCS3200

Terdapat tujuh Pin yang digunakan pada color sensor TCS3200 yaitu Pin VCC, Ground, S0, S1, S2, S3, S4, dan pin OUT. Komunikasi Color sensor dan Arduino Mega dapat dilihat pada Tabel 5.1.

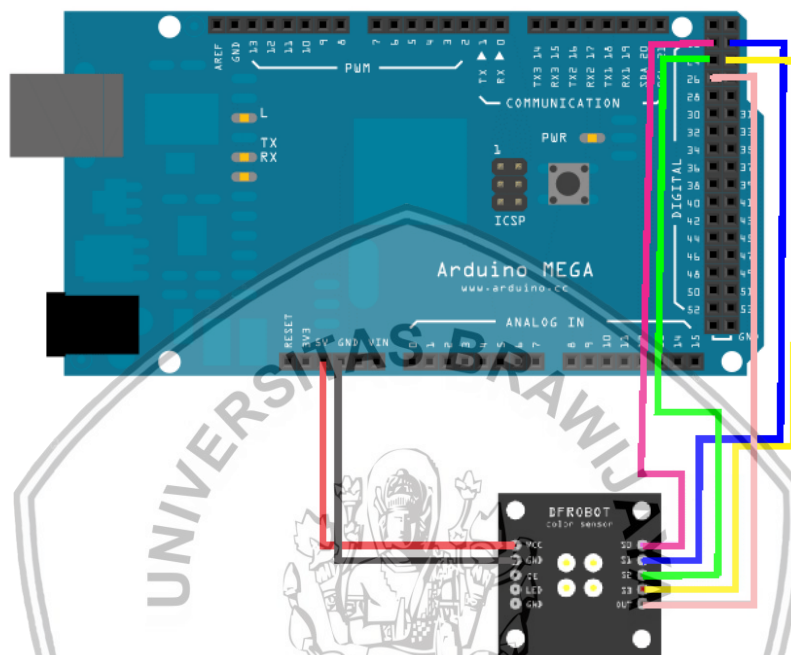
Tabel 5.1 Pinout Color Sensor TCS3200 dengan Arduino

Color Sensor TCS3200	ARDUINO Mega
VCC	5 V
GND	GND
S0	Pin Digital 22
S1	Pin Digital 23
S2	Pin Digital 24
S3	Pin Digital 25
Out	Pin Digital 26

Pin VCC pada sensor dihubungkan dengan 5 Voltz pada Arduino, Pin Ground dihubungkan dengan Ground pada Arduino, 2 pin ini berfungsi sebagai power untuk sensor. Pin S0 dihubungkan dengan Pin Digital 2. Pin S1 dihubungkan



dengan Pin Digital 23. Pin S2 dihubungkan dengan Pin Digital 24. Pin S3 dihubungkan dengan Pin Digital 25. Keempat pin tersebut merupakan pin yang digunakan untuk melakukan sensing warna Merah, Biru, dan Hijau dengan memberikan konfigurasi masing-masing pin yang dapat dilihat pada table. Perancangan Color sensor dengan Arduino dapat dilihat pada Gambar 5.5.



Gambar 5.5 Pinout Perancangan Color Sensor dengan Arduino

5.1.2.2 Perancangan Rangkaian Loadcell Sensor dan HX711 Driver Module

Terdapat empat kabel yang digunakan pada Loadcell sensor, yaitu kabel merah, hitam, hijau, dan putih. Sedangkan pada HX711 driver digunakan Pin VCC, Ground, Data, dan SCK. Komunikasi Loadcell sensor dengan HX711 Driver dapat dilihat pada Tabel 5.3.

Tabel 5.2 Pinout Loadcell Sensor dengan HX711 Driver

Loadcell Sensor	HX711 Driver
Merah	E+
Hitam	E-
Hijau	A-
Putih	A+

Kabel merah pada loadcell sensor merupakan kabel VCC, dihubungkan dengan E+ pada modul HX711, sedangkan kabel hitam merupakan kabel ground, dihubungkan dengan E- pada modul HX711. Kedua kabel tersebut merupakan kabel power pada loadcell sensor. Kabel hijau dihubungkan dengan A- dan kabel putih dihubungkan dengan A+, kedua kabel pada loadcell sensor merupakan kabel

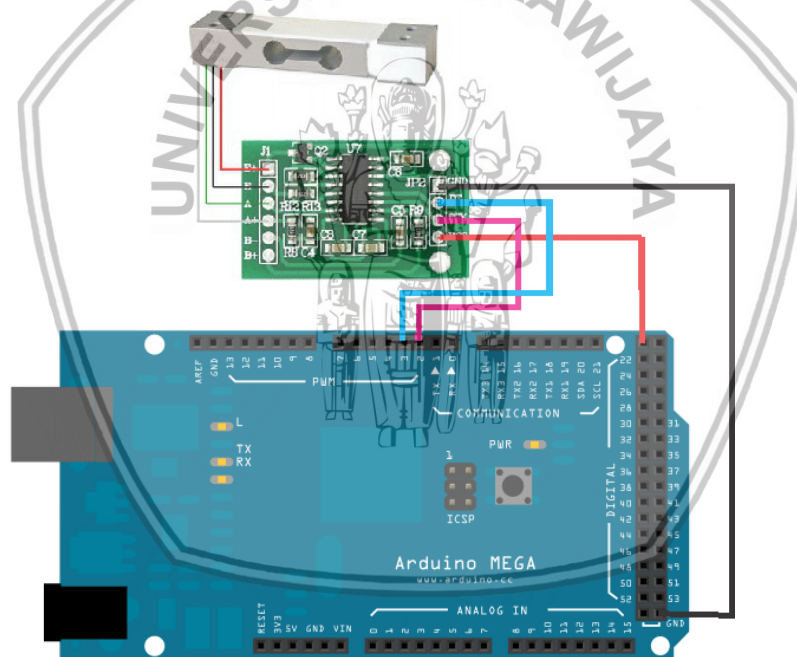


untuk pengiriman data voltase. Komunikasi Loadcell driver HX711 Driver dengan Arduino mega dapat dilihat pada Tabel 5.3.

Tabel 5.3 Pinout HX711 Driver dengan Arduino

Vibration Sensor	ARDUINO Mega
VCC	5 V
GND	GND
Data	Pin Digital 2
SCK	Pin Digital 3

Pin VCC pada sensor dihubungkan dengan 5 Voltz pada Arduino, Pin Ground dihubungkan dengan Ground pada Arduino, 2 pin ini berfungsi sebagai power untuk sensor. Pin data pada driver HX711 dihubungkan dengan digital pin 2 pada Arduino Mega. Pin SCK pada driver HX711 dihubungkan dengan pin digital 3. Kedua pin ini berfungsi untuk mengirimkan data voltase ke Arduino Mega. Rangkaian Loadcell sensor dengan Arduino mega dapat dilihat pada Gambar 5.6.



Gambar 5.6 Pinout Loadcell Sensor dan Modul HX711 dengan Arduino

Pada Gambar 5.6, pin Data, dan SCK akan mengirimkan data voltase berupa data digital melalui interrupt ke Arduino Mega, Tegangan yang masuk pada Arduino akan bertambah seiring dengan bertambahnya beban yang diberikan kepada loadcell sensor.

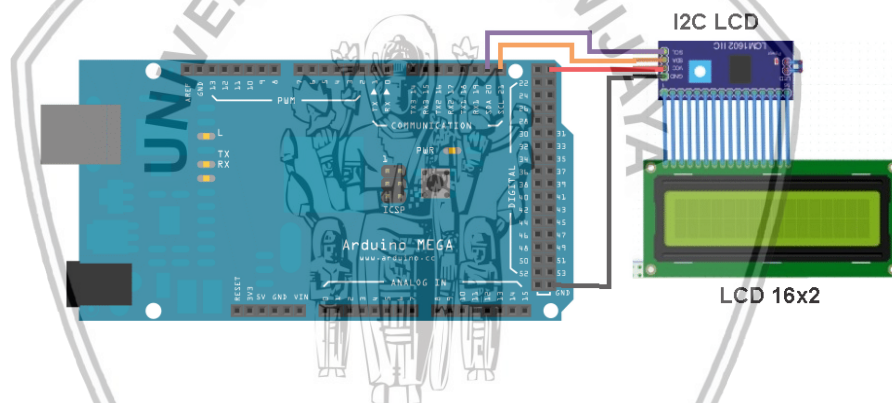
5.1.2.3 Perancangan Sistem LCD dan I2C Module

LCD dan I2C Module dihubungkan, sehingga mengurangi kabel jumper yang dibutuhkan. Terdapat 4 pin pada I2C module, yaitu VCC, Ground, SDA dan SCL. Komunikasi I2C module dan Arduino dapat dilihat pada Tabel 5.4.

Tabel 5.4 Pinout I2C Module dengan Arduino

I2C Module	ARDUINO Mega
VCC	5 V
GND	GND
SDA	Pin Digital 20(SDA)
SCL	Pin Digital 21(SCL)

Pin VCC pada Rangkaian I2C module dihubungkan dengan pin 5V, Pin Ground dihubungkan dengan Ground, 2 Pin ini berfungsi sebagai power. Pin SDA dan SCL dihubungkan dengan Pin digital 20(SDA), Pin Digital 21(SCL). Rangkaian I2C module dan LCD dapat dilihat pada Gambar 5.7.



Gambar 5.7 Pinout Perancangan LCD dan I2C Module dengan Arduino

Terdapat 16 pin pada LCD, setiap Pin dihubungkan dengan I2C module. I2C module berkomunikasi dengan Arduino melalui Pin 20 (SDA) dan Pin 21 (SCL).

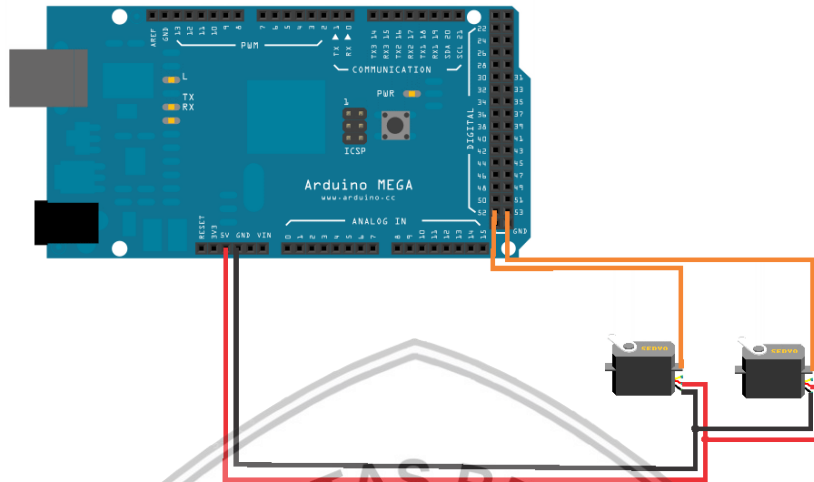
5.1.2.4 Perancangan Sistem Servo SG-90

Terdapat Tiga Pin yang digunakan pada Servo Sg-90, pin VCC, Ground, dan Signal. Ketiga pin ini dihubungkan dengan Arduino mega, sistem komunikasinya dapat dilihat pada table.

Servo Sg-90	ARDUINO Mega
VCC	5 V
GND	GND
Signal	Pin 52 dan 53

Pin VCC pada Rangkaian servo dihubungkan dengan pin 5V, Pin Ground dihubungkan dengan Ground, 2 Pin ini berfungsi sebagai power. Pin signal pada servo dihubungkan dengan Pin digital 52 dan 53, pin signal ini nantinya digunakan

untuk mengatur posisi dari servo SG-90. Rangkaian Servo Sg-90 dapat dilihat pada Gambar 5.8.



Gambar 5.8 Pinout Perancangan Servo dengan Arduino

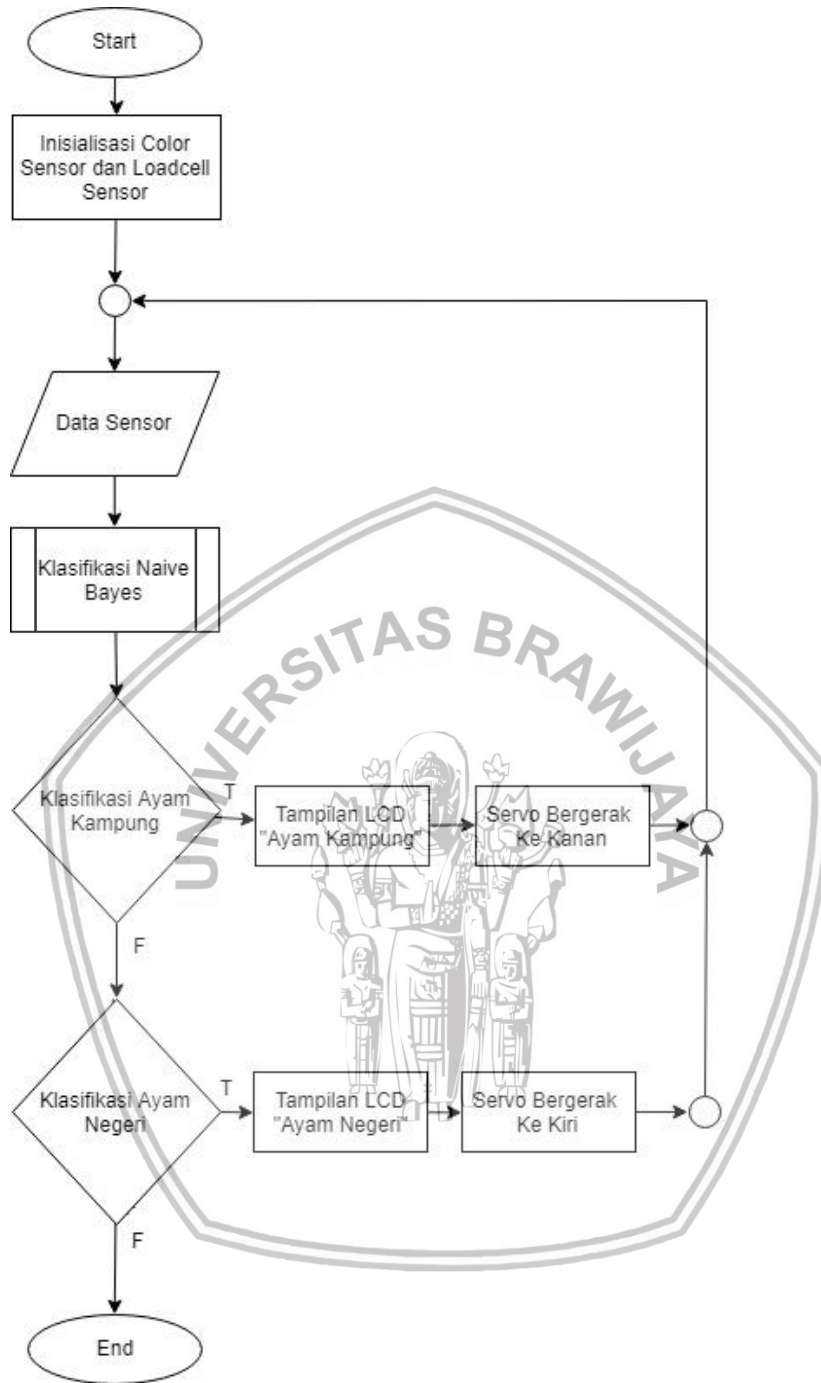
Karena terdapat 2 servo, maka pin signal dihubungkan dengan Pin digital 52, dan pin digital 53.

5.1.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak dibagi menjadi 2 pembahasan, yakni perancangan perangkat lunak pada mikrokontroler untuk mengambil data sensor yang akan diolah, serta perancangan perangkat lunak untuk melakukan proses klasifikasi dengan metode Naive Bayes.

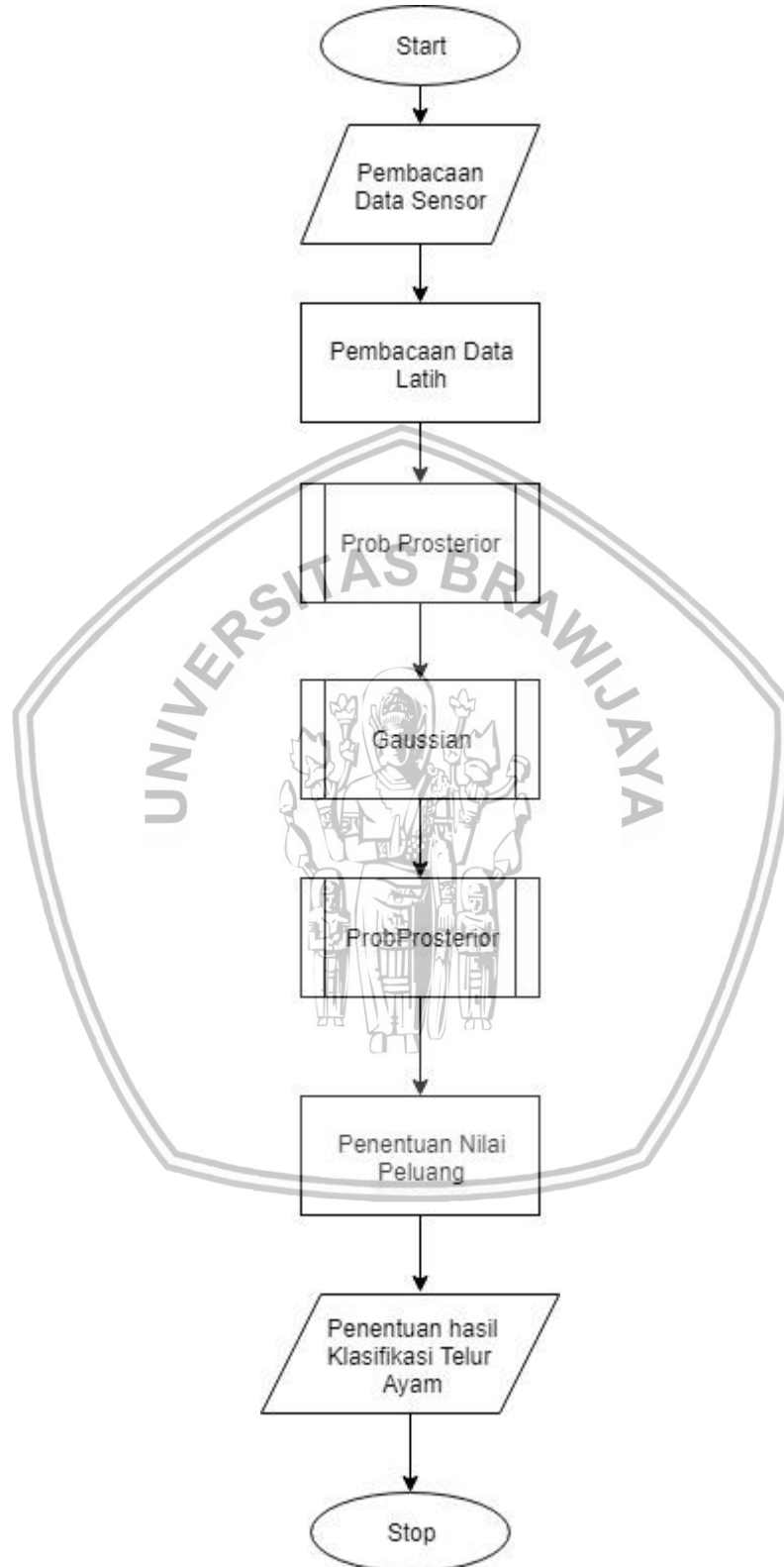
5.1.3.1 Perancangan pembacaan Sistem Sensor

Pada Gambar 5.9 sistem diawali dengan inialisasi color sensor dan loadcell sensor. Input sistem berupa data sensor yaitu nilai warna R,G,B dan Berat, nilai tersebut diolah dan dilakukan klasifikasi menggunakan metode Naive Bayes. Hasil dari penghitungan berupa klasifikasi jenis telur ayam, yaitu telur ayam kampung dan telur ayam negeri. Jika hasil klasifikasi berupa telur ayam kampung, LCD akan menampilkan “Ayam Kampung”, dan servo akan bergerak ke Arah kanan. Namun jika hasil klasifikasi berupa telur ayam Negeri, LCD akan menampilkan “Ayam Negeri”, dan servo akan bergerak ke Arah kiri. Program akan terus berjalan sampai sistem dimatikan.



Gambar 5.9 Flowchart Alur Perancangan Perangkat Lunak Sistem

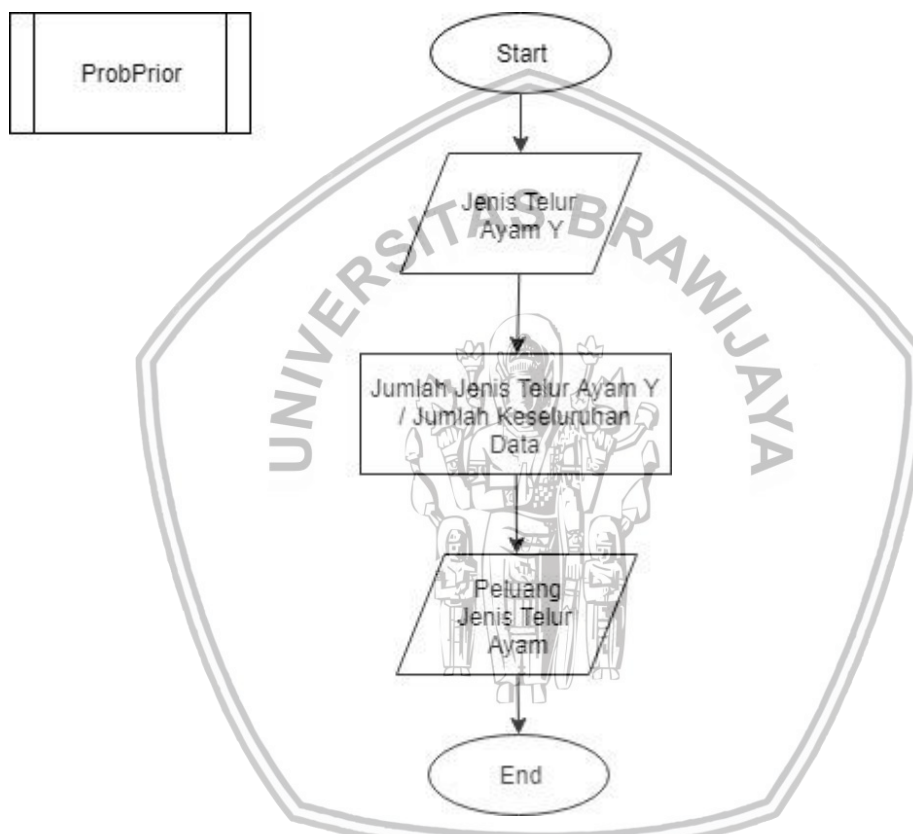
5.1.3.2 Perancangan Klasifikasi Naive bayes



Gambar 5.10 Flowchart Alur Perancangan Klasifikasi Naive Bayes

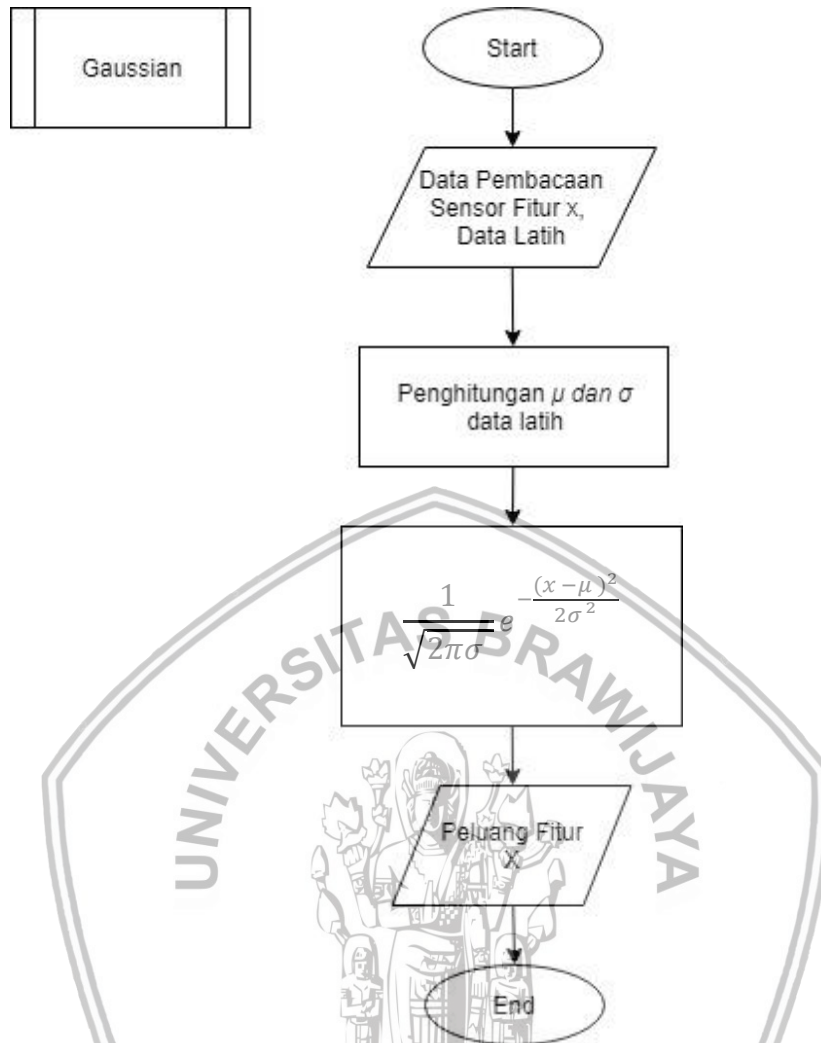
Dalam Melakukan proses klasifikasi terdapat beberapa langkah yang dapat dilihat pada Gambar 5.10. Sistem dimulai ketika sensor membaca nilai berat dan

warna dengan fitur R, G, B. Empat nilai tersebut menjadi masukan awal sistem. Nilai sensor menjadi fitur-fitur yang mempengaruhi penentuan klasifikasi jenis telur ayam. Selain data masukan hasil klasifikasi juga dipengaruhi oleh nilai data latih yang sudah di cari sebelumnya. Proses dimulai dari menentukan hasil dari fungsi ProbPrior, langkah kedua menentukan hasil dari dari fungsi Gaussian, selanjutnya menentukan hasil dari fungsi ProbPosterior, hasil dari penghitungan ketiga proses tersebut untuk menentukan hasil peluang tertinggi hingga didapatkan hasil klasifikasi jenis telur ayam. Penjelasan dari fungsi probprior dapat dilihat pada Gambar 5.11.



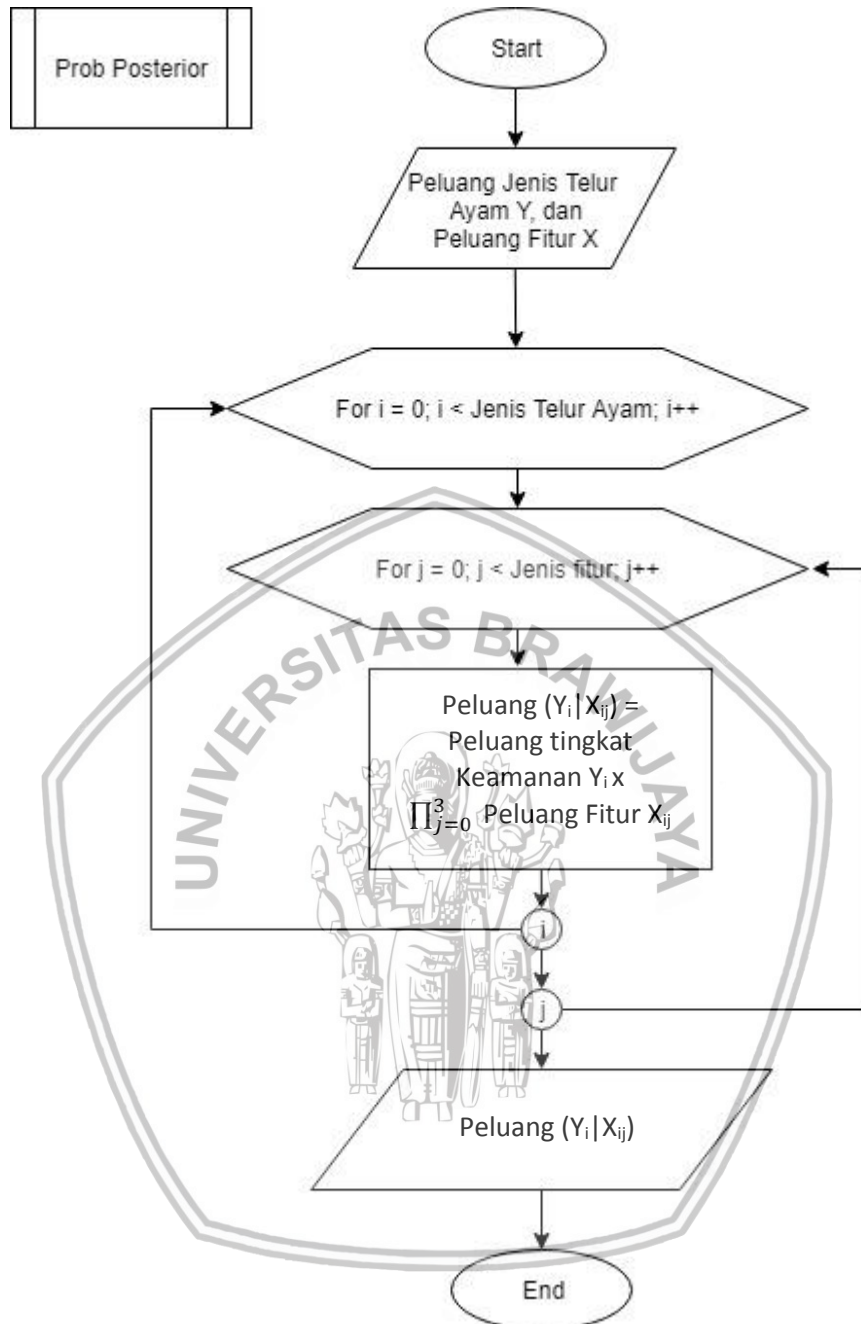
Gambar 5.11 Flowchart Alur Perancangan Fungsi ProbPrior

Langkah pertama yang dilakukan dalam mengkasifikasikan tingkat klasifikasi telur ayam dengan metode Naive Bayes adalah menghitung nilai prior dari masing kelas jenis telur ayam. Nilai prior merupakan nilai peluang terjadinya suatu kelas dengan cara membagi banyaknya data dalam suatu kelas/output (dalam sistem ini terdapat 2 jenis output yaitu jenis Ayam Kampung, dan Ayam negeri) dengan jumlah keseluruhan data yang ada. Data yang dilakukan perhitungan nilai prior adalah data latih. Langkah selanjutnya berupa mencari nilai Gaussian yang dapat dilihat pada Gambar 5.12.



Gambar 5.12 Flowchart Alur Perancangan Fungsi Gaussian

Tahap kedua yaitu untuk menentukan nilai peluang dari masing-masing fitur. Terdapat 2 fitur yang digunakan dalam sistem ini yaitu yaitu nilai berat dari pengolahan loadcell Sensor, dan data warna dengan nilai R, G, B, dari pengolahan color sensor TCS3200. Namun sebelum dapat menentukan nilai peluang dari masing-masing fitur, terlebih dahulu harus dilakukan perhitungan mean dan standar deviasi dari data latih menggunakan Persamaan (2.4) dan Persamaan (2.5). Pada sistem ini, data latih yang berupa nilai mean dan standar deviasi disimpan pada Arduino Mega untuk mempermudah dalam mengakses nilai dari data latih saat sistem dijalankan. Selanjut perhitungan Gaussian dilakukan dengan menggunakan Persamaan (2.3) seperti yang ditunjukkan pada Gambar 5.13 dimana nilai x adalah nilai fitur dari pembacaan sensor.



Gambar 5.13 Flowchart Alur Perancangan Fungsi ProbPosterior

Tahap selanjutnya adalah menentukan nilai dari peluang prob posterior, peluang posterior yaitu peluang untuk menentukan besarnya peluang masing-masing kelas akan terjadi ketika adanya masukan dari tiap fitur. Pada sistem ini yaitu untuk menentukan besarnya peluang masing-masing jenis klasifikasi telur ayam akan terjadi ketika adanya pembacaan nilai fitur berat dan warna. Prosesnya yaitu dengan melakukan perkalian antara hasil dari fungsi ProbPrior dengan fungsi Gaussian.

Tahap akhir dalam pengklasifikasian dengan Naive Bayes ini adalah menentukan nilai peluang posterior yang tertinggi atau nilai maximal dengan cara membandingkan satu sama lain antar peluang posterior. Jenis klasifikasi telur ayam dengan nilai peluang posterior paling tinggi merupakan hasil yang klasifikasi jenis telur ayam yang dideteksi oleh sistem.

Penghitungan manual metode naive bayes untuk mengklasifikasikan jenis telur ayam dapat dilihat pada contoh dibawah ini dengan Input data R=20, G=25, B=22, dan berat=36.14 gram berdasarkan data latih pada lampiran.

1. Menghitung peluang prior dari masing-masing jenis klasifikasi telur ayam.

$$P_{\text{AyamKampung}} = \frac{\text{Jumlah Data Latih Ayam Kampung}}{\text{Jumlah Keseluruhan Data Latih}} = \frac{20}{40} = 0.5$$

$$P_{\text{AyamNegeri}} = \frac{\text{Jumlah Data Latih Ayam Negeri}}{\text{Jumlah Keseluruhan Data Latih}} = \frac{20}{40} = 0.5$$

2. Menghitung nilai mean dan standar deviasi dari masing-masing jenis Klasifikasi telur ayam.

- Mean

$$\begin{aligned} \bar{X}_{R(\text{AyamKampung})} &= \frac{\sum_{i=1}^n x_i}{\text{Jumlah jenis Nilai R}} \\ &= \frac{22+22+36 \dots +26}{20} = \frac{535}{20} = 26.75 \end{aligned}$$

$$\begin{aligned} \bar{X}_{G(\text{AyamKampung})} &= \frac{\sum_{i=1}^n x_i}{\text{Jumlah jenis Nilai G}} \\ &= \frac{32+31+52 \dots +33}{20} = \frac{719}{20} = 35.95 \end{aligned}$$

$$\begin{aligned} \bar{X}_{B(\text{AyamKampung})} &= \frac{\sum_{i=1}^n x_i}{\text{Jumlah jenis Nilai B}} \\ &= \frac{29+29+46 \dots +30}{20} = \frac{646}{20} = 32.3 \end{aligned}$$

$$\begin{aligned} \bar{X}_{\text{Berat}(\text{AyamKampung})} &= \frac{\sum_{i=1}^n x_i}{\text{Jumlah jenis Nilai Berat}} \\ &= \frac{33.53+31.96+31.57 \dots +41.62}{20} = \frac{700.001}{20} = 35.0005 \end{aligned}$$

Tabel 5.5 Hasil Penghitungan mean Keseluruhan Jenis telur ayam

Mean	R	G	B	Berat
Telur Ayam Kampung	26.75	35.95	32.3	35.0005
Telur Ayam Negeri	31.65	55.2	53.1	73.8345

• Standart Deviasi

$$\begin{aligned} \sigma_R (\text{AyamKampung}) &= \sqrt{\frac{\sum_{i=1}^n (\text{AyamKampung}_i - \bar{x}_R \text{ AyamKampung})^2}{\text{Jumlah jenis Nilai R AyamKampung} - 1}} \\ &= \sqrt{\frac{(22-26.75)^2 + (22-26.75)^2 + (36-26.75)^2 \dots + (26-26.75)^2}{20 - 1}} = 5.919414978 \end{aligned}$$

$$\begin{aligned} \sigma_G (\text{AyamKampung}) &= \sqrt{\frac{\sum_{i=1}^n (\text{AyamKampung}_i - \bar{x}_G \text{ AyamKampung})^2}{\text{Jumlah jenis Nilai G AyamKampung} - 1}} \\ &= \sqrt{\frac{(32-35.95)^2 + (31-35.95)^2 + (52-35.95)^2 \dots + (33-35.95)^2}{20 - 1}} = 8.822668173 \end{aligned}$$

$$\begin{aligned} \sigma_B (\text{AyamKampung}) &= \sqrt{\frac{\sum_{i=1}^n (\text{AyamKampung}_i - \bar{x}_B \text{ AyamKampung})^2}{\text{Jumlah jenis Nilai B AyamKampung} - 1}} \\ &= \sqrt{\frac{(29-32.3)^2 + (29-32.3)^2 + (46-32.3)^2 \dots + (30-32.3)^2}{20 - 1}} = 8.176345857 \end{aligned}$$

$$\begin{aligned} \sigma_{\text{Berat}} (\text{AyamKampung}) &= \sqrt{\frac{\sum_{i=1}^n (\text{AyamKampung}_i - \bar{x}_{\text{Berat}} \text{ AyamKampung})^2}{\text{Jumlah jenis Nilai Berat AyamKampung} - 1}} \\ &= \sqrt{\frac{(33.53-35)^2 + (31.96-35)^2 + (31.57-35)^2 \dots + (41.62-35)^2}{20 - 1}} = 3.779365688 \end{aligned}$$

Tabel 5.6 Penghitungan Standart Deviasi Jenis Klasifikasi Telur Ayam

Standart Deviasi	R	G	B	Berat
Telur Ayam Kampung	5.919	8.822	8.176	3.779
Telur Ayam Negeri	2.207	3.874	3.837	9.837

3. Menghitung nilai gaussian dari masing-masing fitur (dengan input data R=20, G=25, B=22, dan berat=36.14 Gram).

$$\begin{aligned} P(R=20 | \text{AyamKampung}) &= \frac{1}{\sqrt{2\pi\sigma_R(\text{AyamKampung})^2}} e^{-\frac{(20-\mu_R(\text{AyamKampung}))^2}{2\pi\sigma_R(\text{AyamKampung})^2}} \\ &= \frac{1}{\sqrt{2 \times 3.14 \times 5.919414978^2}} e^{-\frac{(20-26.75)^2}{2 \times 5.919414978^2}} = 0.066873716 \end{aligned}$$



$$P(R=25 | AyamNegeri) = \frac{1}{\sqrt{2\pi\sigma_{R(AyamNegeri)}^2}} e^{-\frac{(23-\mu_{R(AyamNegeri)})^2}{2\pi\sigma_{R(AyamNegeri)}^2}}$$

$$= \frac{1}{\sqrt{2 \times 3.14 \times 2.207046133^2}} e^{-\frac{(23-31.65)^2}{2 \times 2.207046133^2}} = 0.006825457$$

Tabel 5.7 Penghitungan Gaussian Jenis Klasifikasi Telur Ayam

Gaussian	R	G	B	Berat
Telur Ayam Kampung	0.066873716	0.042770373	0.046911381	0.022775041
Telur Ayam Negeri	0.006825457	7.6425000	1.40804000	0.000190286

4. Menghitung nilai peluang posterior masing-masing jenis Klasifikasi Telur Ayam.

- $P(\text{AyamKampung} | R=20, G=25, B=22, \text{Berat}=36.14 \text{ gram})$
 $= P_{\text{AyamKampung}} \times (R = 23 | \text{AyamKampung}) \times (G = 67 | \text{AyamKampung}) \times$
 $(B = 67 | \text{AyamKampung}) \times (\text{Berat} = 67 | \text{AyamKampung})$
 $= 0.035186871 \times 0.020937777 \times 0.022073179 \times 0.100893074$
 $= 8.20366 \times 10^{-7}$
- $P(\text{AyamNegeri} | R=20, G=25, B=22, \text{Berat}=36.14 \text{ gram})$
 $= P_{\text{AyamNegeri}} \times (R = 23 | \text{AyamNegeri}) \times (G = 67 | \text{AyamNegeri}) \times$
 $(B = 67 | \text{AyamNegeri}) \times (\text{Berat} = 67 | \text{AyamNegeri})$
 $= 1.60997 \times 6.59096 \times 5.68793 \times 2.62866$
 $= 7.93279 \times 10^{-42}$

Pada hasil perhitungan peluang posterior diatas, jenis klasifikasi telur ayam yang mempunyai nilai peluang tertinggi adalah kelar ayam kampung, sehingga dapat dikatakan tingkat keamanan dengan fitur input R=20, G=25, B=22, dan berat=36.14 Gram termasuk ke dalam Kelas **Ayam Kampung**.

5.2 Implementasi sistem

Implementasi sistem merupakan tahap untuk merealisasikan pembuatan sistem berdasarkan semua perancangan yang telah dilakukan sebelumnya. Pada subbab ini menjelaskan satu per satu secara rinci terkait implementasi sistem yang dibuat baik dari segi implementasi perangkat keras maupun implementasi perangkat lunak.

5.2.1 Implementasi Desain Sistem

Implementasi desain sistem klasifikasi telur ayam mengacu pada bab perancangan 5.1.1 yang telah dibuat sebelumnya. Sistem kontrol utama yang dibuat ditempatkan pada bagian tengah sistem dengan tujuan mempermudah menghubungkan kabel. Hasil implementasi Sistem Klasifikasi telur ayam beserta peletakan sistem sensor dapat dilihat pada Gambar 5.14.



Gambar 5.14 Implementasi Desain Sistem

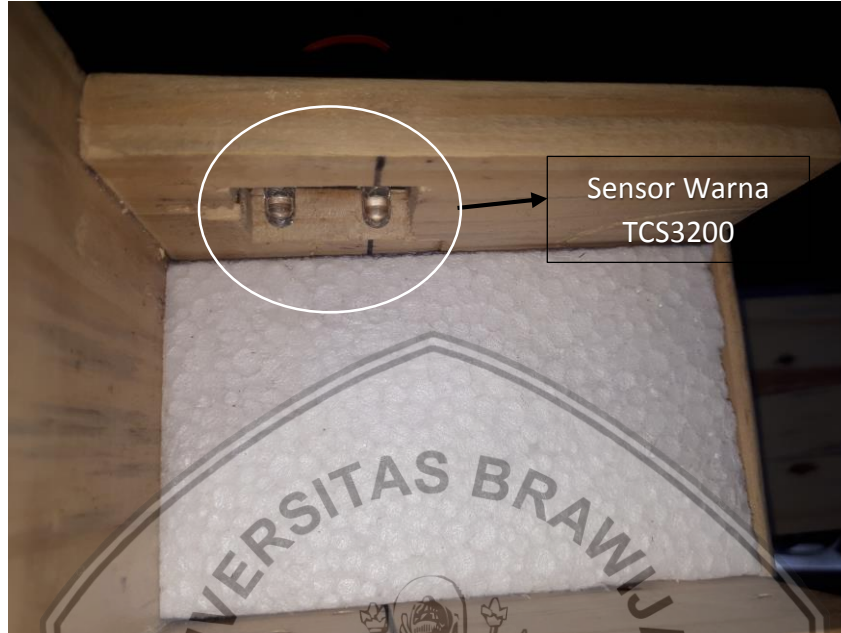
Hasil dari implementasi desain sistem perangkat keras dapat dilihat pada Gambar 5.14, peletakan color dan loadcell sensor terdapat pada wadah yang sama dengan tujuan untuk mempermudah dalam menjalankan sistem, karena user hanya perlu menaruh telur diatas wadah. Peletakan LCD dibagian depan agar mudah melihat hasil status klasifikasi saat ini, sedangkan peletakan servo berada pada bagian belakang agar gerakan telur dapat diarahkan ke wadah yang telah disediakan.

5.2.2 Implementasi Perangkat Keras

Implementasi perangkat keras sistem klasifikasi telur ayam mengacu pada bab perancangan 5.1.2 terbagi menjadi implementasi rangkaian Color sensor, loadcell sensor, LCD, dan servo.

5.2.2.1 Implementasi Color Sensor TCS3200

Untuk mengukur tingkat warna R, G, dan B, Implementasi color sensor TCS3200 dapat dilihat pada Gambar 5.15.

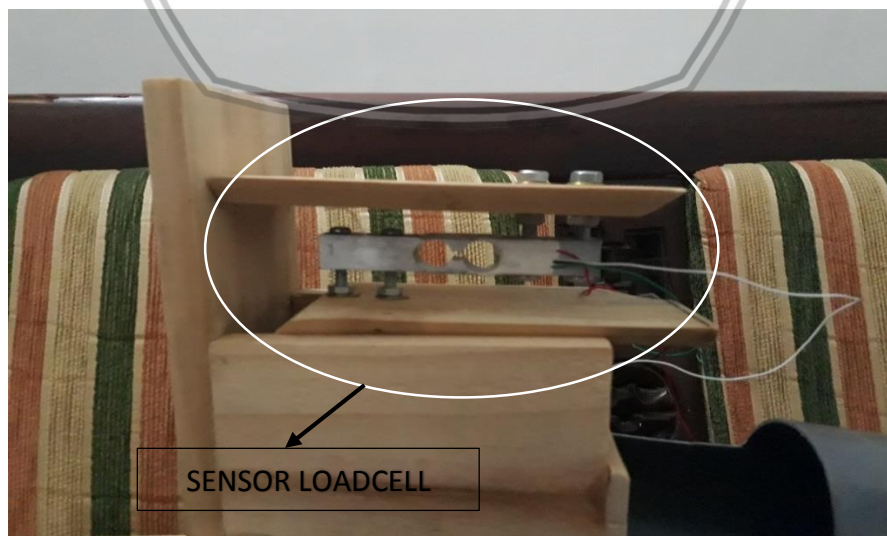


Gambar 5.15 Implementasi Color Sensor TCS3200

Pada Gambar 5.15, Color sensor ditempatkan didalam wadah untuk peletakan telur ayam, sensor akan mendeteksi nilai warna dengan fitur R, G, dan B ketika telur dimasukkan.

5.2.2.2 Implementasi Loadcell Sensor

Untuk mendeteksi berat dari telur ayam, Implementasi loadcell sensor dapat dilihat pada Gambar 5.16.



Gambar 5.16 Implementasi Loadcell Sensor

Loadcell Sensor mendeteksi berat dari telur ayam, sensor ditempatkan didalam wadah telur agar mempermudah pengguna dalam mengoperasikan sistem yang dibuat dalam penelitian ini. Loadcell sensor dihubungkan dengan driver HX711 yang dapat dilihat pada Gambar 5.17.

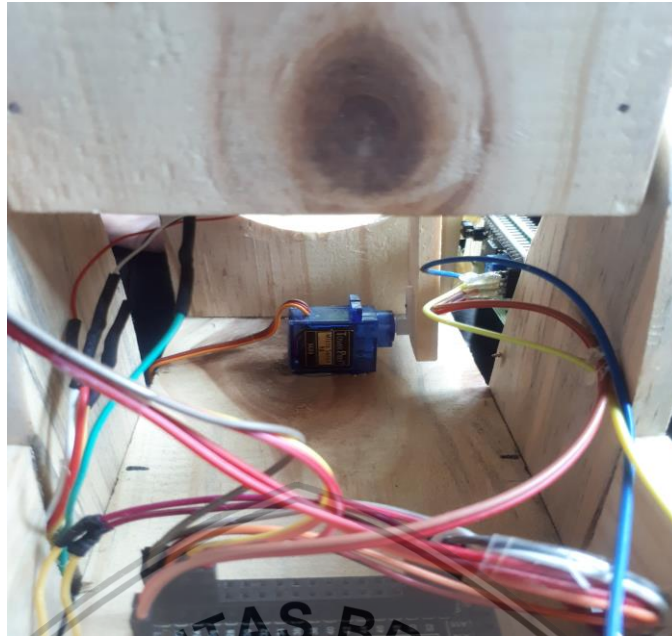


Gambar 5.17 Peletakan Driver HX711

Driver HX711 ditempatkan dibagian bawah dan didekat Arduino untuk mempermudah menghubungkan tiap-tiap kabel.

5.2.2.3 Implementasi Servo

Sebagai output untuk pengelompokan hasil dari klasifikasi sistem pemilah telur ayam, implementasi Servo dapat dilihat pada Gambar 5.18.

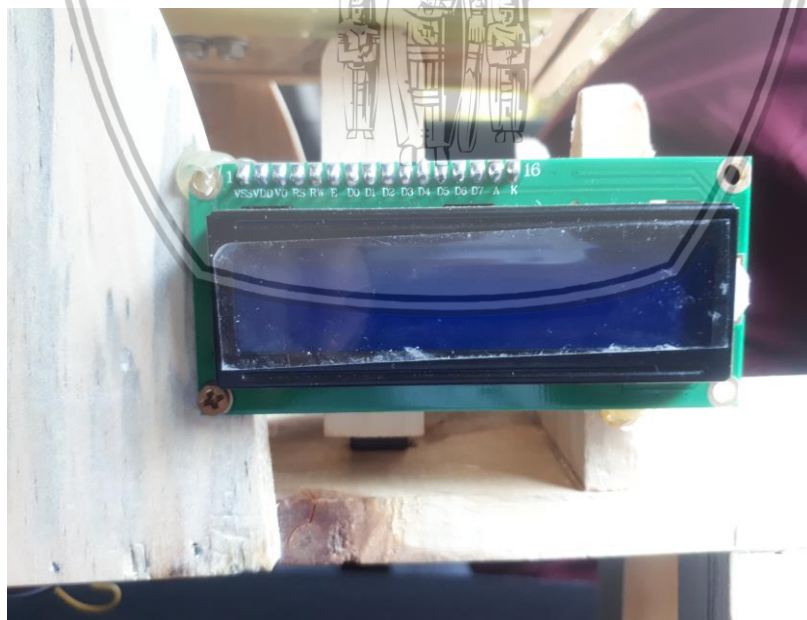


Gambar 5.18 Implementasi Peletakan Servo

Servo berjalan kearah kanan dan kearah kiri ketika telah didapatkan hasil klasifikasi telur ayam.

5.2.2.4 Implementasi Sistem LCD dan I2C

Untuk menampilkan nilai sensor dan pengolahan data, Implementasi dan peletakan LCD serta I2C dapat dilihat pada Gambar 5.19.



Gambar 5.19 Implementasi Peletakan LCD

LCD ditempatkan pada bagian samping alat agar terlihat dengan jelas nilai warna R, G, B dan nilai berat pada baris pertama, Serta menampilkan hasil klasifikasi telur ayam pada baris kedua.

5.2.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak menjelaskan proses pengaplikasian program untuk sistem klasifikasi telur ayam berdasarkan perancangan yang telah dilakukan pada subbab 5.1.3. Dalam melakukan implementasi perangkat lunak ini sepenuhnya proses pengkodean program dilakukan pada Arduino IDE 1.8.5 dimana diawal program dilakukan inialisasi library yang digunakan untuk mempermudah pemrograman beberapa fungsi tertentu.

Tabel 5.8 Implementasi Library Yang Digunakan

Baris	Kode Program
1	<code>#include <LiquidCrystal_I2C.h></code>
2	<code>#include <math.h></code>
3	<code>#include <wire.h></code>
4	<code>#include <HX711.h></code>
5	<code>#include <Servo.h></code>
6	

Pengimplementasian library pada sistem ini dapat dilihat pada Tabel 5.7, diantaranya adalah library “LiquidCrystal_I2C.h” untuk penggunaan LCD 16x2, serta “wire.h” untuk komunikasi I2C dan library, “math.h” untuk melakukan perhitungan matematika yang cukup kompleks, “HX711.h” untuk memanggil pemrograman driver Loadcell, “Servo.h” untuk memanggil pemrograman untuk menggerakkan servo.

5.2.3.1 Implementasi Pembacaan data sensor

Implementasi pembacaan data Loadcell sensor dan driver HX711 untuk mengetahui nilai berat diawali dengan inialisasi pin yang digunakan, yaitu pin 2 dan 3. Lalu inialisasi variable untuk nantinya digunakan untuk pengolahan data sensor menjadi nilai berat.

Tabel 5.9 Kode Program Loadcell Sensor dan Driver HX711

Baris	Kode Program
1	<code>#define DOUT 2</code>
2	<code>#define CLK 3</code>
3	
4	<code>HX711 scale(DOUT, CLK);</code>
5	
6	<code>float calibration_factor = -2456650.00;</code>
7	
8	<code>void setup() {</code>
9	<code> Serial.begin(9600);</code>
10	<code> scale.set_scale();</code>

```
11 scale.tare();
12
13 long zero_factor = scale.read_average();
14 Serial.print("Zero factor: ");
15 Serial.println(zero_factor);
16 }
17
18 void loop() {
19
20     scale.set_scale(calibration_factor);
21
22     Serial.print("Reading: ");
23     Serial.print(scale.get_units()*1000*-1);
24     Serial.print(" grams");
25     Serial.println();
26 }
```

Pada bagian awal program, pada baris ke-4 merupakan pembuatan object yang dipanggil dari library HX711.h lalu dibuat dengan nama scale(), pada baris ini juga sebagai inisialisasi awal untuk driver sensor berat. Pada baris ke-6 merupakan nilai yang digunakan untuk kalibrasi sensor berat dengan nilai -2456650.00, nilai ini didapat melalui kalibrasi yang telah dilakukan sebelumnya. Lalu pada baris ke-10 dan 11, merupakan pemanggilan fungsi yang digunakan untuk kalibrasi didalam software. Pada baris ke 13 sampai 15, berfungsi untuk menampilkan hasil kalibrasi yang telah dilakukan. Pada void loop, sistem akan menampilkan nilai berat yang dihasilkan dari loadcell Sensor. Program akan terus berjalan sampai sistem dimatikan. Kode program untuk pembacaan deteksi getaran dapat dilihat pada Tabel 5.10 berikut.

Tabel 5.10 Kode Program Untuk Mendeteksi Warna Dengan Nilai R,G,B

Baris	Kode Program
1	<code>#define S0 22</code>
2	<code>#define S1 23</code>
3	<code>#define S2 24</code>
4	<code>#define S3 25</code>
5	<code>#define sensorOut 26</code>
6	
7	<code>int frequency = 0;</code>
8	<code>void setup() {</code>
9	<code> pinMode(S0, OUTPUT);</code>
10	<code> pinMode(S1, OUTPUT);</code>
11	<code> pinMode(S2, OUTPUT);</code>
12	<code> pinMode(S3, OUTPUT);</code>
13	<code> pinMode(sensorOut, INPUT);</code>
14	
15	<code> // Setting frequency-scaling to 20%</code>
16	<code> digitalWrite(S0,HIGH);</code>
17	<code> digitalWrite(S1,LOW);</code>
18	
19	<code> Serial.begin(9600);</code>
20	<code>}</code>
21	<code>void loop() {</code>
22	
23	
24	<code> digitalWrite(S2,LOW);</code>
25	<code> digitalWrite(S3,LOW);</code>
26	<code> frequency = pulseIn(sensorOut, LOW);</code>
27	<code> Serial.print("R= ");</code>
28	<code> Serial.print(frequency);</code>
29	<code> Serial.print(" ");</code>
30	<code> delay(100);</code>
31	
32	
33	<code> digitalWrite(S2,HIGH);</code>
34	<code> digitalWrite(S3,HIGH);</code>
35	<code> frequency = pulseIn(sensorOut, LOW);</code>
36	<code> Serial.print("G= ");</code>
37	<code> Serial.print(frequency);</code>
38	<code> Serial.print(" ");</code>
39	<code> delay(100);</code>
40	
41	
42	<code> digitalWrite(S2,LOW);</code>
43	<code> digitalWrite(S3,HIGH);</code>
44	<code> frequency = pulseIn(sensorOut, LOW);</code>
45	<code> Serial.print("B= ");</code>
46	<code> Serial.print(frequency);</code>
47	<code> Serial.println(" ");</code>
48	<code> delay(100);</code>
49	<code>}</code>

Program diawali dengan inialisasi pin untuk S1, S2, S3, S4, OUT dan inialisasi variabel untuk pembacaan nilai warna yang dapat dilihat pada baris 1 sampai 7. Pada baris 9 sampai 13, pin 21-25 disetting sebagai output, sedangkan



pin 26 disetting sebagai input untuk membaca nilai warna. Pada baris 16 – 17 frekuensi scaling sensor warna disetting menjadi 20% jika dihubungkan dengan arduino, hal ini sesuai dengan datasheet sensor warna. Pada baris 24 - 30 merupakan kode program untuk membaca nilai merah. Pada baris 33 - 39 merupakan kode program untuk membaca nilai hijau. Pada baris 42 - 48 merupakan kode program untuk membaca nilai biru. Kode program untuk Servo dapat dilihat pada Tabel 5.11 berikut.

Tabel 5.11 Kode Program Servo

Baris	Kode Program
1	#include <Servo.h>
2	
3	Servo myservo;
4	
5	void setup()
6	{
7	myservo.attach(51);
8	myservo.write(95); //mid point +- 40 gan
9	}
10	
11	void loop() {}

Inisialisasi pin untuk Servo dihubungkan dengan pin digital 51 pada Arduino. Pada baris 3, program diawali dengan membuat object dari library servo.h dengan nama myservo. Gerakan servo dapat diatur sedemikian rupa dengan memanggil fungsi myservo.write() yang dapat dilihat pada baris 8. Program untuk LCD dapat dilihat pada Tabel 5.12 berikut.

Tabel 5.12 Kode Program LCD

Baris	Kode Program
1	#include <Wire.h>
2	#include <LiquidCrystal I2C.h>
3	LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3,
4	POSITIVE);
5	
6	void setup()
7	{
8	
9	lcd.begin(16,2);
10	lcd.backlight();
11	
12	}
13	
14	void loop()
15	{
16	lcd.setCursor(0,0);
17	lcd.print("R");
18	lcd.print(Red);
19	lcd.print("G:");
20	lcd.print(Green);

21	lcd.print("B");
22	lcd.print(Blue);
23	lcd.print("W:");
24	lcd.print(Weight);
25	
26	
27	lcd.setCursor(0,1);
28	lcd.print("Hasil:");
28	lcd.print(Hasil);
30	delay(1000);
31	lcd.clear();
32	delay(15);}

LCD dihubungkan dengan I2C module diawali dengan inialisasi alamat I2C 0x27, alamat ini didapat dari datasheet LCD yang digunakan. Lalu dilanjutkan dengan inialisasi pin yang menghubungkan I2C module dengan LCD display. Pada fungsi setup() LCD inialisasi LCD dengan mode 16 x 2. Pada baris ke 16 sampai 24 LCD menampilkan nilai pembacaan sensor, dan hasil pengolahan status klasifikasi telur ayam dengan metode naive bayes yang dapat dilihat pada baris 27 sampai 32.

5.2.3.2 Implementasi Klasifikasi Naive Bayes

Tahap implementasi kode program *Naive Bayes* dimaksudkan untuk merealisasikan kode program saat melakukan pengambilan keputusan jenis klasifikasi telur ayam dengan warna dan berat yang dideteksi oleh sensor.

Tabel 5.13 Kode Program Inialisasi Variabel

Baris	Kode Program
1	float gaussian[3][4];
2	float hasil[2];
3	float tertinggi = -1.000;
4	int index = 0;
5	int gauske = 0;

Berdasarkan Tabel 5.13 yang menunjukkan proses inialisasi variabel dan tipe data yang digunakan untuk penghitungan metode Naive Bayes. Kode diatas dilakukan inialisasi sebagai variabel global agar dapat diakses oleh keseluruhan sistem.

Tabel 5.14 Kode Program Nilai Peluang Prior

Baris	Kode Program
1	float pAyamKampung = 0.5; //Peluang ayamKampung = 20 / 40
2	
3	float pAyamNegeri = 0.5; //Peluang AyamNegeri = 20 / 40
4	
5	float pAyamKampung [2][4] = {{26.75, 35.95, 32.3,
6	35.0005}, {5.919414978, 8.822668173, 8.176345857,
7	3.779365688}};
8	
9	



10	float pAyamNegeri [2][4] = {{31.65, 55.2, 53.1,
11	73.8345}, {2.207046133, 3.874342049, 3.837488214, 9.837109372}};

Metode Naive Bayes diawali dengan menentukan nilai peluang prior masing-masing jenis klasifikasi yang dapat dilihat pada Tabel 5.14. Pada kode program diatas baris ke-1 sampai ke-3 menerangkan nilai peluang untuk masing-masing jenis klasifikasi telur ayam, yaitu telur ayam kampung dan negeri. Nilai ini mengacu pada penghitungan dan jumlah data latih sebanyak 40 data. Sedangkan pada baris ke-5 sampai ke-11 menerangkan suatu variabel array berdimensi 2x2 dimana tiap dimensi merepresentasikan nilai mean dan nilai standar deviasi dari masing-masing jenis telur ayam yang nilai ini digunakan untuk melakukan perhitungan fungsi gaussian().

Tabel 5.15 Kode Program Penghitungan Gaussian

Baris	Kode Program
1	void gaussian(float data uji[4], float data_latih[2][4])
2	{
3	double d, e, f, g;
4	//mencari nilai gaussian setiap fitur
5	for (int i = 0; i < 4; i++) {
6	d = 2 * 3.14 * (pow(data_latih[1][i], 2));
7	e = -((pow((data uji[i] - data_latih[0][i]), 2)) / (2
8	* pow(data_latih[1][i], 2)));
9	f = pow(2.718282, e);
10	g = 1 / sqrt(d);
11	gaussian[gauske][i] = f * g;
12	//cek nilai gaussian yang masuk
13	Serial.print("nilai gaussian ke-");
14	Serial.print(i);
15	Serial.print(" = ");
16	Serial.println(gaussian[gauske][i] * 1000);
17	}
18	gauske++;
19	}

Langkah pengimplementasian kode program selanjutnya yaitu untuk perhitungan gaussian. Berdasarkan Tabel 5.15 diatas Kode pemrograman fungsi gaussian di atas dijelaskan bahwa pada baris ke-1 merupakan inialisasi fungsi gaussian dengan parameter perhitungannya ditentukan oleh suatu variabel array berukuran 1x4 yang merepresentasikan nilai pembacaan sensor warna dengan nilai R pada index pertama, G pada index kedua, B pada index ketiga, dan berat pada index keempat. Parameter lainnya adalah variabel array berukuran 2x4 yang merepresentasikan nilai hasil olahan data latih. Pada baris ke-2 menunjukkan inialisasi variabel yang akan digunakan untuk menentukan rumus perhitungan. Lalu pada baris ke-4 hingga baris ke-11 menunjukkan proses perhitungan gaussian berdasarkan Persamaan (2.3) secara berulang-ulang hingga keseluruhan fitur dari pembacaan sensor didapatkan peluang gaussiannya masing-masing. Pada baris ke 13 sampai 16 kode untuk menampilkan hasil penghitungan pada serial monitor. Pada baris ke-18 melakukan iterasi sebanyak satu kali untuk variabel gauske.

Tabel 5.16 Kode Program Penghitungan Hasil Peluang Jenis Telur Ayam

Baris	Kode Program
1	<code>void probPosterior(float prior, int i) {</code>
2	<code> for (int j = 0; j < 4; j++) {</code>
3	<code> if (j == 0) {</code>
4	<code> hasil[i] = (gaussian[i][j] * 1000);</code>
5	<code> }</code>
6	<code> else</code>
7	<code> {</code>
8	<code> hasil[i] = hasil[i] * (gaussian[i][j] * 1000);</code>
9	<code> }</code>
10	<code> }</code>
11	<code> Serial.print("hasil peluang likelihood ke-");</code>
12	<code> Serial.print(i);</code>
13	<code> Serial.print(" = ");</code>
14	<code> Serial.println(hasil[i]);</code>

Pada Tabel 5.16 di atas menunjukkan implementasi kode pemrograman untuk mendapatkan nilai hasil peluang tingkat keamanan dari data yang diujikan. Pada baris ke-1 menunjukan inialisasi fungsi ProbPosterior dengan parameter perhitungannya adalah peluang dari masing-masing jenis klasifikasi telur ayam dan nilai gaussian dari masing-masing fitur(Telur ayam kampung, dan negeri). Selanjutnya pada baris ke-2 hingga baris ke-10 menunjukkan perulangan untuk melakukan perkalian antar nilai gaussian keseluruhan fitur. Selanjutnya pada baris ke-8, hasil perkalian antar nilai gaussian keseluruhan fitur di kalikan lagi dengan peluang prior sehingga didapatkan nilai peluang posterior.

Tabel 5.17 Kode Program Penghitungan Klasifikasi Naive Bayes

Baris	Kode Program
1	<code>void kesimpulan() {</code>
2	<code> for (int i = 0; i < 2; i++) {</code>
3	<code> if (i == 0) {</code>
4	<code> tertinggi = hasil[i];</code>
5	<code> index = i + 1;</code>
6	<code> } else if (tertinggi < hasil[i]) {</code>
7	<code> tertinggi = hasil[i];</code>
8	<code> index = i + 1;</code>
9	<code> }</code>
10	<code> }</code>
11	<code> Serial.print("Tertinggi = ");</code>
12	<code> Serial.println(tertinggi);</code>
13	<code> Serial.print("Index = ");</code>
14	<code> Serial.println(index);</code>
15	<code> if (index == 1) {</code>
16	<code> lcd.setCursor(0, 1);</code>
17	<code> lcd.print("Ayam Kampung-");</code>
18	<code> Serial.println("Ayam Kampung");</code>
19	<code> }</code>
20	<code> else if (index == 2) {</code>
21	<code> lcd.setCursor(0, 1);</code>
22	<code> lcd.print("Ayam Negeri-");</code>
23	<code> Serial.println("Ayam Negeri");</code>
24	<code> }</code>

25	delay(300);
26	}

Hasil akhir penghitungan klasifikasi naive bayes jenis telur ayam yaitu dengan membandingkan antar nilai peluang posterior mana yang mempunyai nilai tertinggi seperti ditunjukkan pada Tabel 5.17. Terlihat pada baris ke-2 hingga baris ke-10 setiap nilai peluang posterior dari tingkat keamanan dibandingkan, kemudian nilai yang tertinggi menandakan data uji tersebut termasuk kedalam jenis telur ayam dari peluang posterior yang di maksud. Selanjut pada baris ke-11 hingga baris ke-20 menampilkan hasil klasifikasi telur ayam yang sesuai dengan hasil perbandingan.

