

BAB III METODE PENELITIAN

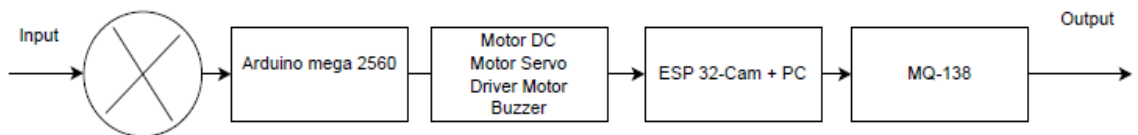
Pada bab ini akan dibahas mengenai proses perancangan dan fabrikasi Alat pendeteksi daging segar dan tidak segar/berformalin yang bertujuan membangun sistem yang bekerja sebagaimana yang diharapkan. Perencanaan sistem ini digambarkan kedalam *flowchart* dan diagram blok yang menggambarkan bagaimana setiap komponen akan terhubung,

3.1 Diagram Blok dan Flowchart

Dalam sub bab ini berisikan diagram blok dan flowchart dari sistem yang akan dibuat untuk memudahkan alur kerja dari sistem.

3.1.1 Blok diagram sistem

Untuk blok diagram sistem bisa dilihat pada **Gambar 3.1** yang menjelaskan secara singkat alur kerja sistem utama.

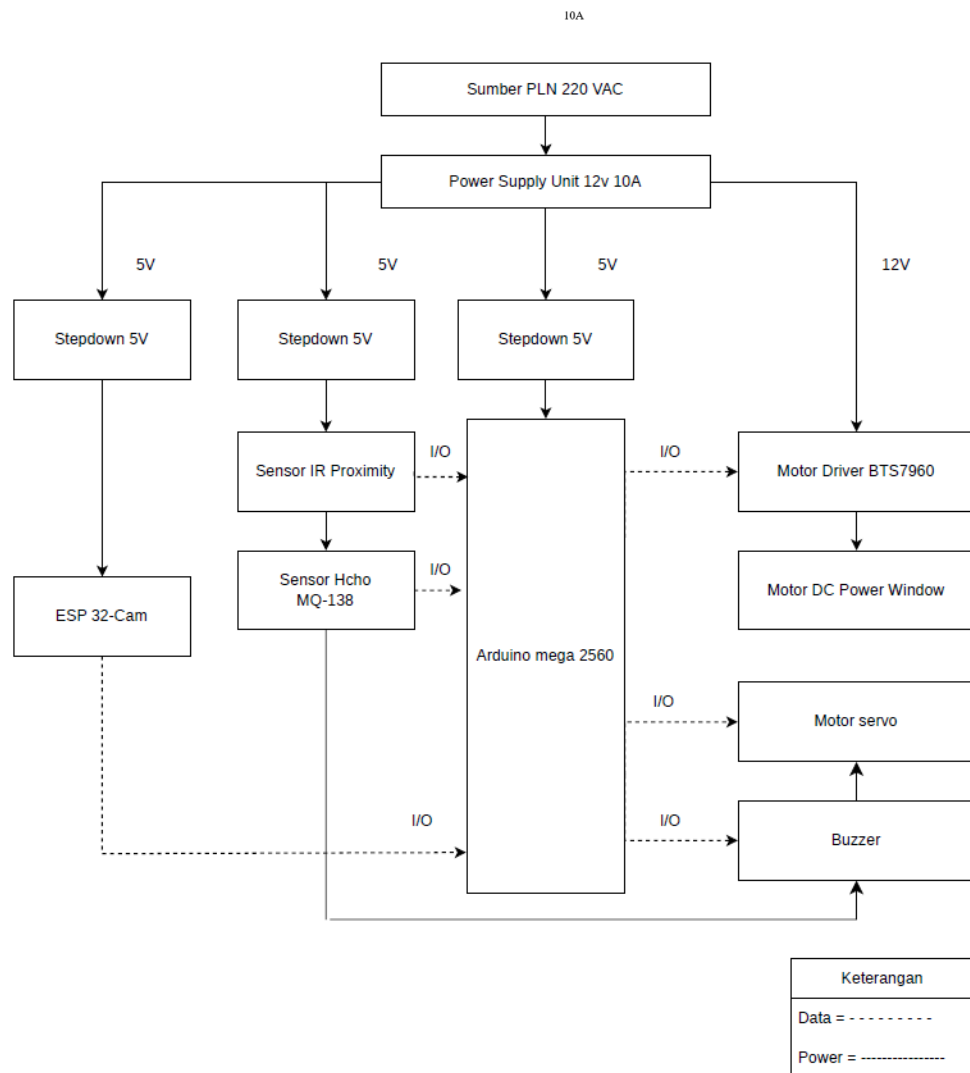


Gambar 3. 1 Diagram blok sistem open loop.

Open loop sistem karena adanya ketiadaan Umpan Balik Korektif (Corrective Feedback) pada aktuator utama. Tetapi sistem ini juga bisa disebut close loop karena umpan balik dari klasifikasi yolo dan sensor MQ-138. Tetapi disimpulkan bahwa sistem ini yaitu sistem control sekuensial open loop, dengan justifikasi bahwa kelemahan control motor open loop untuk penyerderhanaan dan memindahkan kompleksitas ke tahap klasifikasi umpan balik yang merupakan focus utama penelitian.

3.1.2 Blok diagram perangkat keras

Pada Sub bab ini akan dijelaskan tentang blok diagram dari sistem keseluruhan untuk memudahkan alur sistem kerja dari alat seperti pada **Gambar 3.2**.



Gambar 3. 2 Diagram blok sistem keseluruhan.

Berikut merupakan penjelasan dari diagram blok perangkat keras pada alat pendeteksi daging sapi segar atau tidak segar/berformalin:

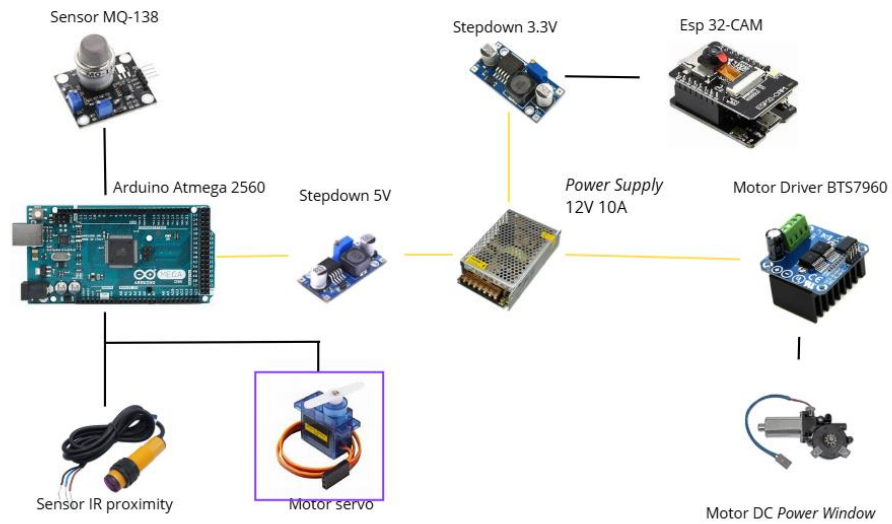
1. 220 VAC merupakan sumber daya utama sistem, yaitu Listrik AC dengan tegangan 220 volt. Digunakan. untuk menyalakan seluruh komponen pada sistem ini dengan di sambungkan ke PSU 12V 10/30A.
2. PSU 12V 10/30A merupakan pengubah tegangan 220 VAC menjadi 12 VDC 10/30A. Dengan tegangan 12V ini digunakan untuk memberi daya pada komponen yang membutuhkan catu daya

3. Step Down 3.3V merupakan modul step-down yang menurunkan tegangan dari 12V menjadi 3.3V. Tegangan ini digunakan untuk ESP 32 – Cam.
4. Sensor Proximity atau IR ini akan digunakan untuk mendeteksi keberadaan daging yang akan membuat motor conveyer berhenti terlebih dahulu.
5. Sensor gas MQ-138 ini digunakan khusus mendeteksi kadar formaldehida (HCHO) di udara. Sensor ini mengirimkan data hasil deteksi ke Arduino atmega 2560 untuk diproses lebih lanjut yaitu image processing dengan yolo.
6. ESP 32-Cam digunakan untuk menangkap gambar maupun video yang nantinya gambar tersebut akan di proses lebih lanjut pada image processing yang hasil outputnya menjadi segar dan tidak segar/berformalin.
7. Arduino Atmega 2560 ini merupakan otak dari sistem ini mendapatkan input berupa data dari sensor MQ- 138, Sensor IR Proximity dan ESP 32-Cam kemudian memproses dan mengontrol outputnya seperti motor conveyer atau actuator (motor servo) menentukan apa daging tersebut segar atau tidak segar maupun berformalin
8. Motor Conveyer untuk motor yang dipakai ini yaitu motor DC power window dikarenakan torsiya cocok untuk menggerakkan conveyer belt dan bisa diatur dengan Arduino ATmega 2560 untuk memindahkan daging sebelum menuju tempat sortir segar dan tidak segar/berformalin.
9. Aktuator atau motor servo digunakan untuk menyortir daging di akhir dengan gerakan yang presisi untuk memindahkan daging ke tempat yang benar.

3.1.3 Rangkaian elektrikal keseluruhan

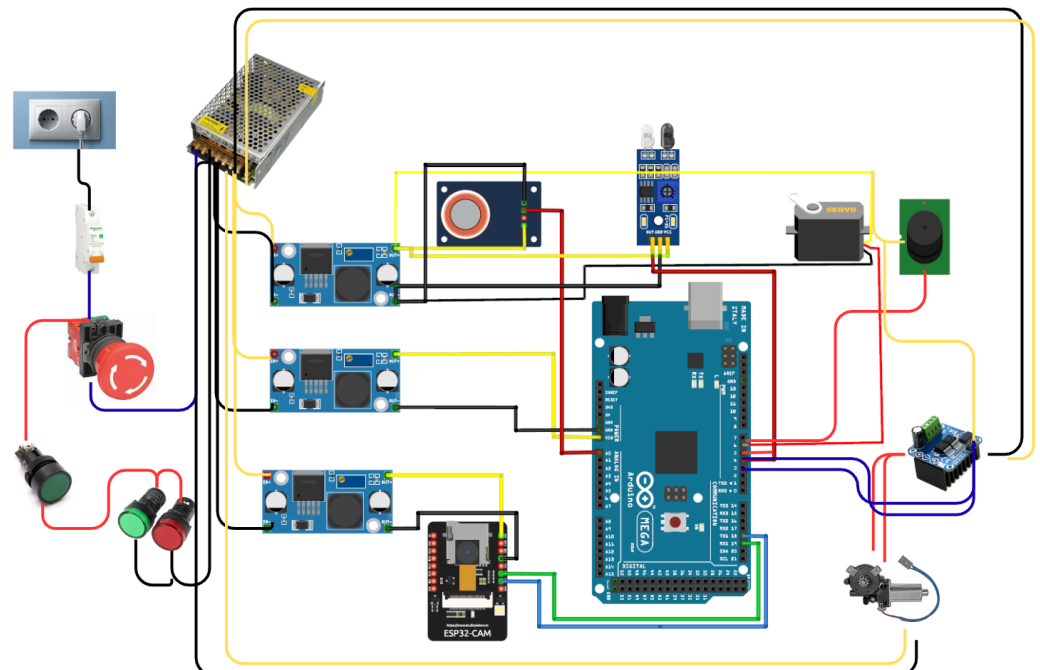
Untuk rangkaian elektrikal berupa skematik yang akan digunakan pada penelitian “Deteksi kualitas daging sapi segar dan tidak segar/berformalin” yang ditunjukkan pada Gambar 3.3 dibawah ini. Dalam skematik ini terdapa: Arduino ATmega 2560, ESP32-cam, Sensor MQ-

138, Power supply 12V 30A, Motor DC 12V, Driver motor, dan sensor IR proximity. Bisa dilihat pada **Gambar 3.3** dibawah.



Gambar 3. 3 Diagram komponen alat utama sistem.

Untuk wiring lengkapnya bisa dilihat di halaman selanjutnya pada **Gambar 3.4**.



Gambar 3. 4 Diagram wiring alat keseluruhan.

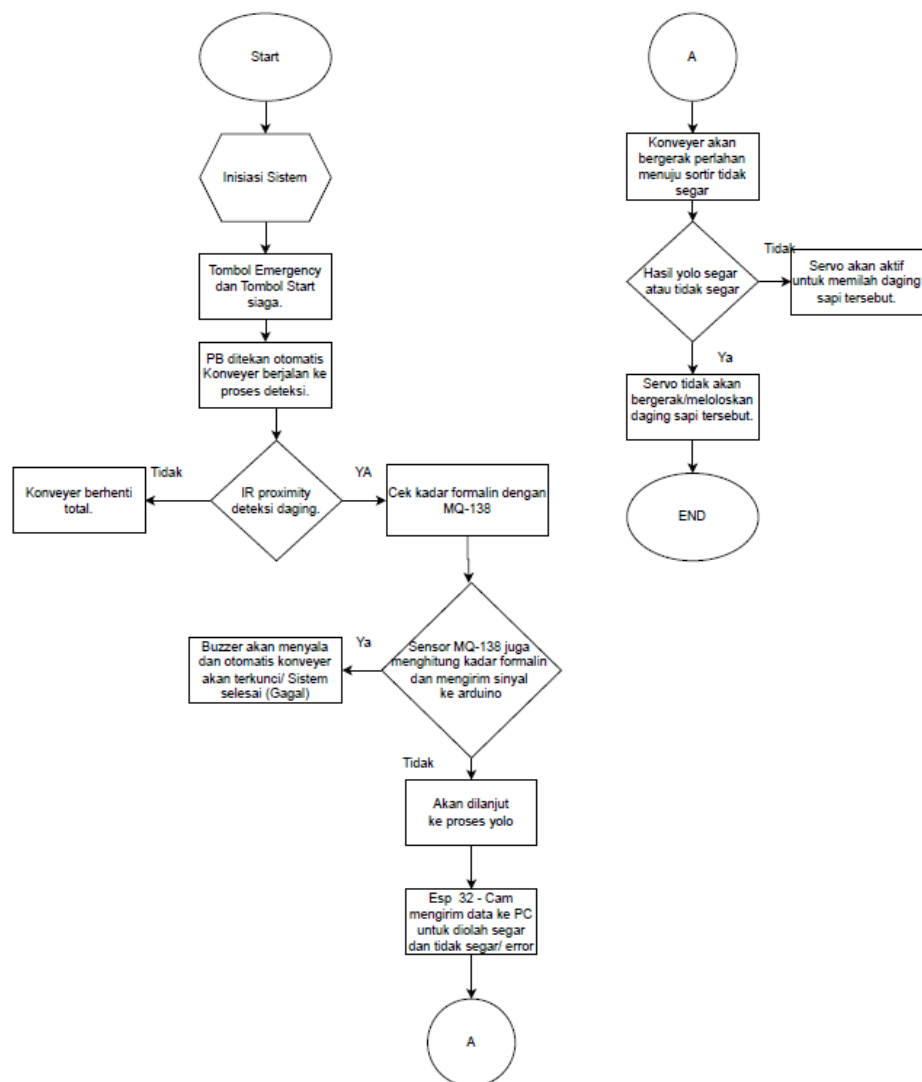
Gambar 3.3 dan 3.4 merupakan perancangan elektrik, fungsi dari setiap komponen yaitu sebagai berikut:

- a. Arduino ATmega 2560 sebagai pusat kendali utama sistem, Mengatur alur kerja motor DC, motor servo, dan menerima data dari sensor MQ-138 dan sensor IR proximity, Berkomunikasi dengan ESP32-CAM untuk mengambil keputusan sortir daging, Buzzer digunakan untuk mengetahui kadar formalin yang tinggi.
- b. ESP32-CAM bertugas mengambil gambar daging sapi yang lewat di conveyor, Mengirimkan gambar ke sistem YOLO (misal di PC/Colab) untuk diproses dan menentukan kualitas daging (segar/tidak segar), Dapat mengirim hasil deteksi kembali ke Arduino via komunikasi serial.
- c. Sensor MQ-138 digunakan untuk mendeteksi keberadaan gas seperti formalin dan senyawa volatil lainnya, Memberi sinyal analog ke Arduino sebagai tambahan informasi tentang kualitas daging.
- d. Sensor IR Proximity mendeteksi keberadaan objek (daging) pada conveyor, Berfungsi sebagai trigger untuk menghentikan motor dan memulai proses pengambilan gambar oleh ESP32-CAM.
- e. Motor Driver BTS7960 mengendalikan motor DC power window 12V (mengatur arah dan kecepatan), Menerima sinyal kontrol dari Arduino dan daya dari PSU 12V.
- f. Motor DC Power Window 12V menggerakkan conveyor belt untuk membawa daging dari titik deteksi ke titik sortir. bekerja sesuai kendali dari BTS7960 yang dikontrol oleh Arduino.
- g. Motor Servo (1 buah) berfungsi untuk menyortir daging ke dalam kotak segar atau tidak segar, Bergerak berdasarkan sinyal dari Arduino setelah hasil deteksi YOLO diterima.
- h. Power Supply 12V 20A/30A menyediakan sumber daya utama untuk seluruh sistem, Memberikan daya ke semua komponen yang digunakan.

- i. Step-down 12V ke 3.3V menurunkan tegangan dari PSU 12V menjadi 3.3V, Digunakan untuk menyuplai daya ke ESP32-CAM yang hanya bekerja pada tegangan 3.3V.
- j. Step-down 12V ke 5V menurunkan tegangan dari PSU 12V menjadi 5V, Digunakan untuk menyuplai daya ke Arduino dan komponen lainnya yang berkapasitas 5V.

3.1.4 Flowchart sistem utama

Pada Sub bab ini akan dijelaskan tentang *flowchart* dari sistem keseluruhan untuk memudahkan alur sistem kerja dari alat.sebagai berikut. Dimana ditunjukkan pada **Gambar 3.5** dibawah ini,



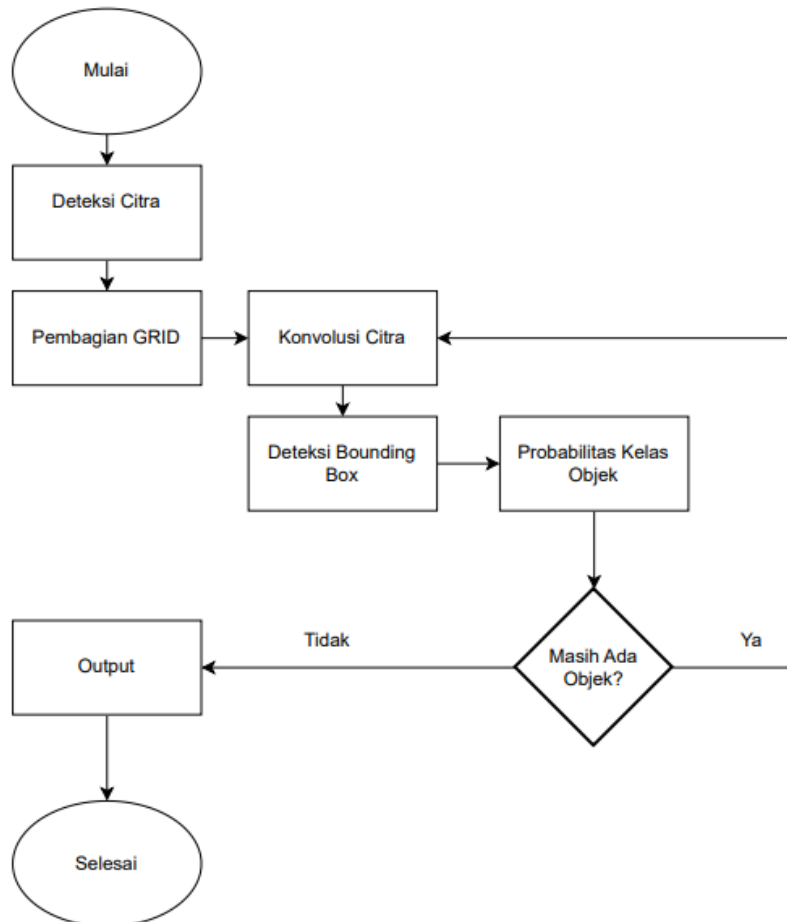
Gambar 3. 5 Flowchart sistem keseluruhan.

Berikut penjelasan dari **Gambar 3.5** flowchart diatas:

1. Memulai Proses (Start) proses dimulai dengan inialisasi sistem. Ini mencakup persiapan semua komponen perangkat keras (hardware) dan perangkat lunak (software) yang diperlukan,
2. Tombol siaga dimana sistem mendeteksi bahwa tombol emergency dan tombol start siap digunakan.
3. Saat PB ditekan maka akan menjalankan konveyer untuk mencapai tempat proses deteksi.
4. Saat daging sapi terdeteksi (IR proximity) sistem memeriksa daging sapi
 - Jika TIDAK terdeteksi = Konveyer akan berhenti total (Gagal dan mengulang loop dari awal lagi).
 - Jika YA terdeteksi = Proses akan dilanjutkan kedalam cek kadar formalin.
5. Cek kadar formalin dengan menggunakan MQ-138 dimulai.
6. Jika sistem memeriksa apakah Sensor MQ-138 mendeteksi kadar formalin melebihi ambang batas maka.
 - Jika YA = Buzzer akan otomatis menyala dan mengunci sistem (Sistem selesai/gagal).
 - Jika TIDAK = Proses akan dilanjutkan ke proses YOLO.
7. ESP32-Cam mengirim data gambar ke PC untuk diolah, dan Arduino menunggu hasil status (Segar/ Tidak Segar/error).
8. Setelah keluar hasil YOLO sistem akan dilanjutkan ke dalam fase penyortiran daging sapi dengan motor servo sebagai pemilahnya.
9. Keputusan klasifikasi yolo diterima (Segar/Tidak segar/error).
 - Jika TIDAK (Tidak segar/error) = Servo akan aktif untuk memilah daging sapi tersebut.
 - Jika YA servo tidak akan bergerak, meloloskan daging tersebut.

10. END setelah aksi penyotiran selesai, proses untuk daging tersebut berakhir dan akan loop ulang.

3.1.1 Flowchart YOLOv5s



Gambar 3. 6 Flowchart sistem YOLOv5s.

Berikut penjelasan dari **Gambar 3.6** flowchart diatas:

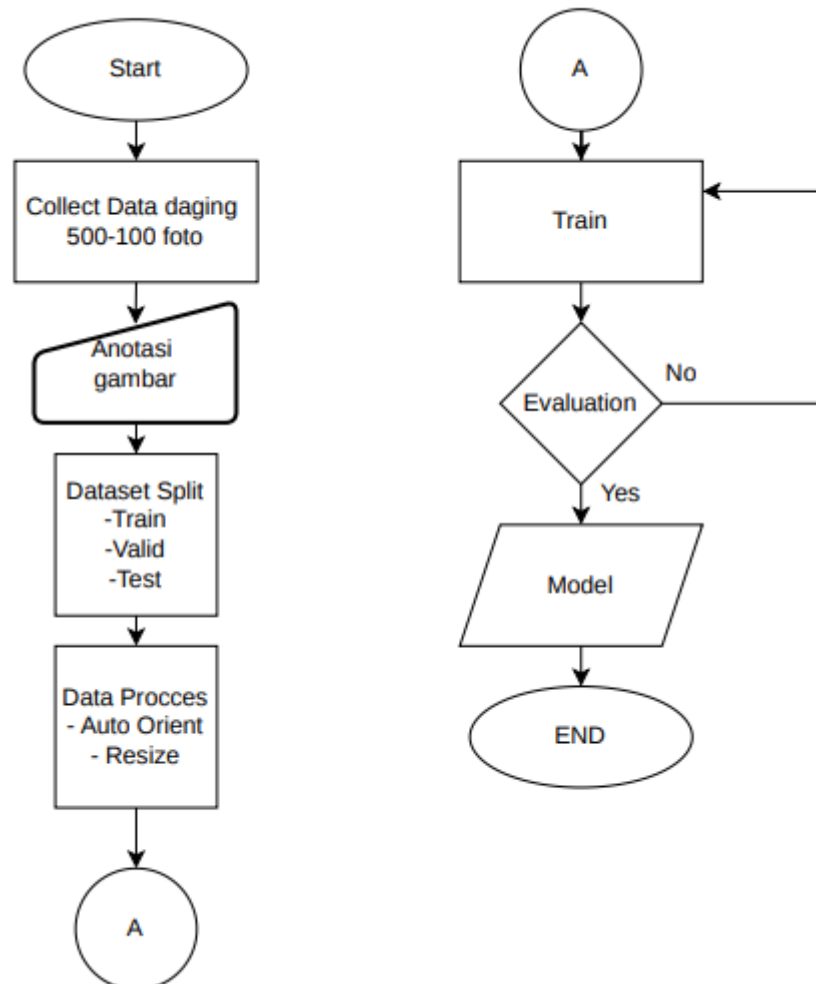
1. Memulai Proses (Mulai) proses diawali dengan inialisasi sistem, di mana model deteksi objek dimuat dan siap menerima input.
2. Pra-pemrosesan Citra Deteksi Citra (Image Detection) sistem menerima input berupa satu citra atau *frame* dari video. Pembagian GRID (Grid Division) citra yang masuk dibagi menjadi beberapa sel grid. Setiap sel grid akan bertanggung jawab untuk mendeteksi keberadaan objek yang

berpusat di dalamnya. Pendekatan ini adalah karakteristik utama dari algoritma deteksi objek *real-time* seperti YOLO (You Only Look Once).

3. Analisis dan Prediksi Objek Konvolusi Citra (Image Convolution) setiap sel grid diproses oleh Jaringan Saraf Tiruan (Convolutional Neural Network - CNN). Jaringan ini akan mengekstrak fitur-fitur penting dari citra (seperti tepi, tekstur, dan bentuk) yang diperlukan untuk mengenali objek. Deteksi Bounding Box (Bounding Box Detection) Berdasarkan fitur yang diekstrak, model memprediksi kotak pembatas (bounding box) di sekitar setiap objek yang terdeteksi. Kotak ini menentukan lokasi dan ukuran objek di dalam citra. Probabilitas Kelas Objek (Object Class Probability) Bersamaan dengan deteksi *bounding box*, model juga menghitung probabilitas bahwa objek di dalam kotak tersebut termasuk dalam kelas tertentu (misalnya, mobil, pejalan kaki, atau lampu lalu lintas).
4. Proses Iterasi sistem akan mengecek apakah masih ada objek yang perlu diproses atau jika proses prediksi perlu diulang untuk *frame* berikutnya. Jika masih ada atau diperlukan, proses akan berulang kembali ke langkah Konvolusi Citra untuk melakukan prediksi yang lebih akurat atau memproses objek lain. Jika semua objek telah teridentifikasi, proses berlanjut.
5. Output setelah semua objek terdeteksi dan diklasifikasikan, sistem menampilkan hasil akhir berupa citra yang telah diberi label dengan kotak pembatas dan nama kelas objek beserta tingkat probabilitasnya. Selesai (Selesai) Proses deteksi untuk satu citra atau *frame* video dianggap selesai. Sistem kemudian siap untuk menerima input citra baru dan mengulang seluruh proses.

3.2 Perancangan Model Deteksi

Perancangan model dilakukan untuk membedakan daging segar dan tidak segar serta berformalin. Alur proses perancangan model deteksi dapat dilihat pada **Gambar 3.7** Alur proses model deteksi dibawah ini (dilanjutkan di page selanjutnya).



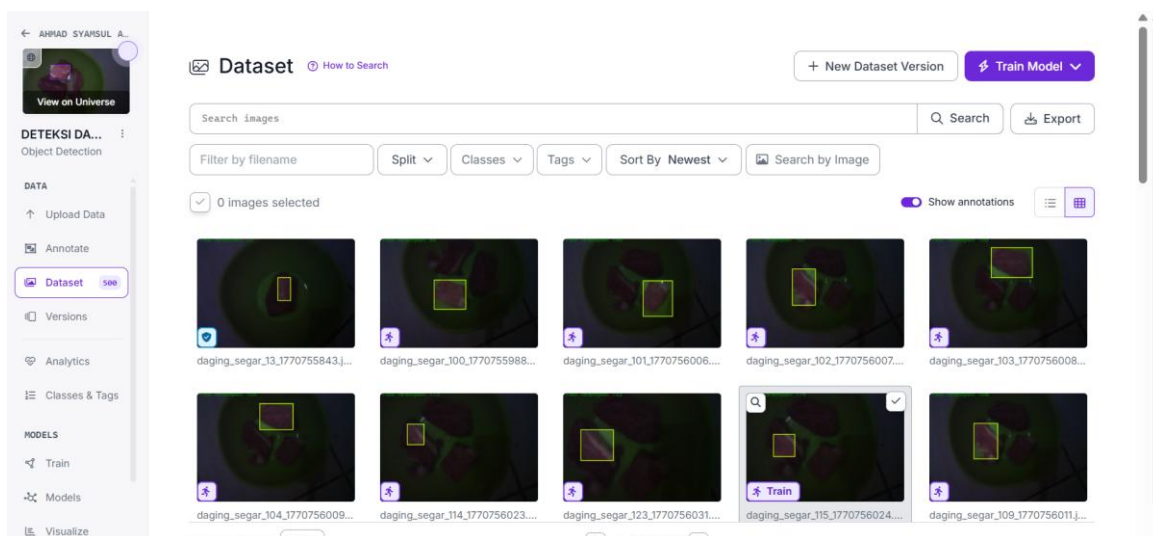
Gambar 3. 7 Alur proses perancangan model deteksi

Dalam **Gambar 3.7**, Diatas menggambarkan metodologi sistematis dalam pengembangan model deteksi objek, Proses ini akan membuat model yang dihasilkan memiliki akurasi tinggi dalam membedakan kualitas daging sapi. Bisa dijelaskan dari alur diatas sebagai berikut:

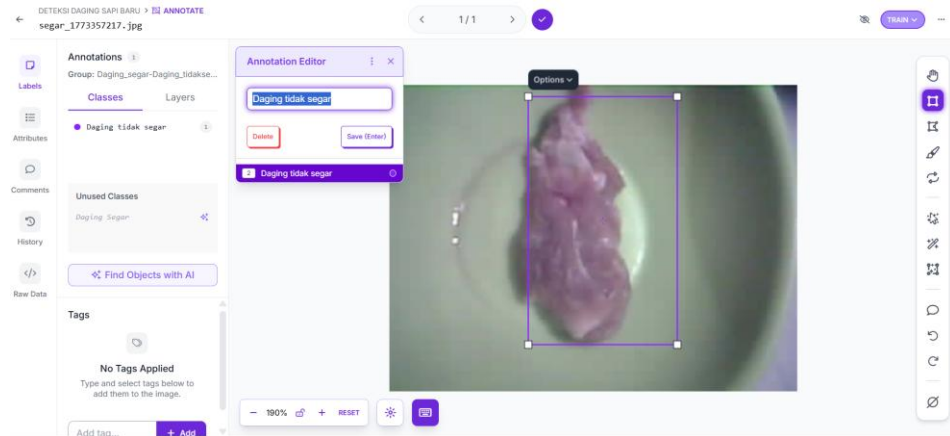
3.2.1 Fase Pra-Pemrosesan Data

Siklus ini dilakukan di ESP32-Cam dimulai dengan Dataset Acquisition atau mengumpulkan data 500 citra yang diambil langsung dari kamera ESP32-Cam dari berbagai sudut dan kondisi pencahayaan. Angka 500 - 1000 dipilih sebagai minimum Dengan variasi sudut dan cahaya, model tidak akan "bingung" saat alat bekerja di lingkungan yang berbeda (misalnya cahaya redup atau terang).

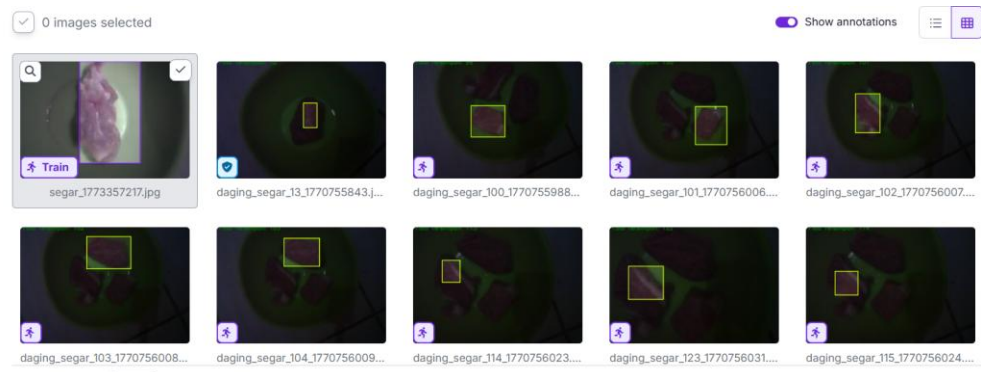
Kemudian masuk ke tahap Data Annotation di roboflow menggunakan format YOLO (koordinat normalisasi x, y, w, h). Setiap *bounding box* akan menghasilkan file .txt yang berisi indeks kelas (0 untuk segar, 1 untuk tidak segar) dan posisi objek. Dataset kemudian akan melalui tahap Dataset *Split*, dimana data akan dibagi menjadi 3 bagian yaitu: *Train* (data utama untuk pembelajaran), *Valid* (untuk evaluasi saat *training* agar tidak *overfitting*), dan *Test* (untuk uji coba) dengan rasio 80% (400 gambar) untuk *Train*, 10% (50 gambar) untuk *Valid*, dan 10% (50 gambar) untuk *Test*. Sebelum masuk ke mesin pelatihan, dilakukan dulu *Augmentasi & Resizing etc.* Proses *resize* ke 640x480 (atau standar YOLOv5s di 640x640) dilakukan untuk menyeimbangkan antara kecepatan komputasi dan ketajaman fitur tekstur daging.



(a) Fase dataset sekitar 500 gambar



(b) Fase Anotasi manual menggunakan bounding box



(c) Fase Train dan Valid dan Test

Gambar 3. 8 Fase pre-processing (a) Pengambilan datasheet; (b) Fase Anotasi gambar; (c) Fase *Train*, *Valid* dan *Test*.

3.2.2 Fase Pelatihan dan Validasi

Setelah fase pra-pemrosesan data sistem akan memasuki fase *Train* yang dilakukan di Google Collab. Di sinilah "otak" YOLOv5s dibentuk. Proses ini terjadi di dalam kartu grafis (GPU) karena membutuhkan ribuan perhitungan matriks per detik.

- Arsitektur YOLOv5s (Small) dipilih tipe "s" agar model ringan dan *real-time* saat dijalankan di VS Code, namun tetap tajam mengenali tekstur daging.

- Hyperparameters (Setingan Mesin):
 1. Epochs dimana menjalankan sekitar 100-300 putaran. Di setiap epoch, model mencoba mengenali fitur (lemak, warna merah, tekstur serat).
 2. Batch Size diatur ke 16 atau 32. Artinya, model membaca 16 gambar sekaligus sebelum melakukan koreksi bobot.
 3. Optimizer (Adam/SGD) algoritma yang bertugas mencari "jalan pintas" tercepat agar tingkat kesalahan (*loss*) mencapai titik terendah.
 4. Learning Rate (10^{-3}) Kecepatan model dalam menyerap informasi baru.
- Proses Evaluasi (Monitoring)
 1. Setiap akhir satu *epoch*, sistem akan mengecek data Validation.
 2. Jika Loss (kesalahan) terus turun dan mAP (akurasi) naik, maka pelatihan sehat.
 3. Jika mAP berhenti naik tapi Loss Train turun, terjadi Overfitting (model hanya menghafal, bukan memahami). Pelatihan harus dihentikan (*Early Stopping*).

Untuk gambarnya bisa dilihat pada sub bab 3.63 perancangan YOLO dibawah.

3.2.3 Fase *Deployment* model akhir

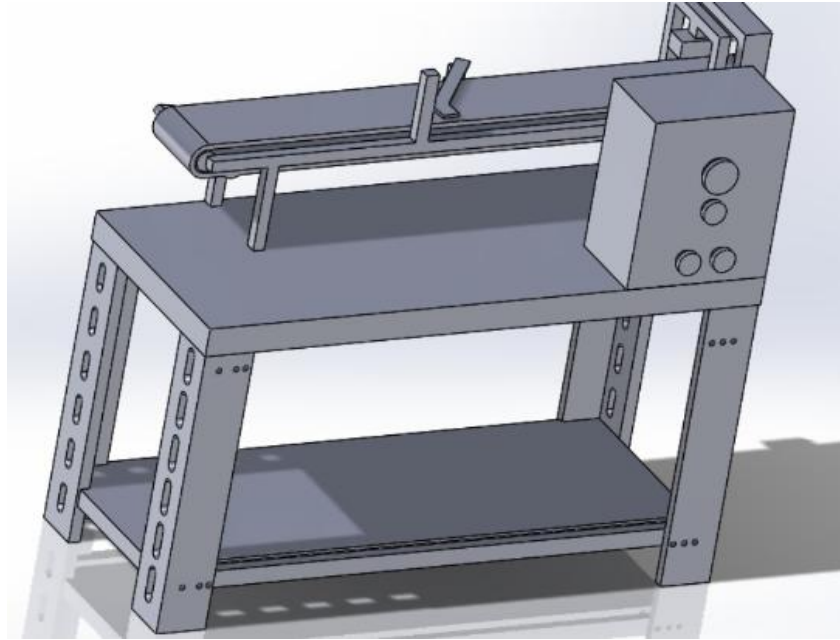
Pada fase ini merupakan output akhir yang dijalankan di VScode dari seluruh rangkaian proses yang berupa file model (.pt atau .onnx) file inilah yang akan menjadi komponen inti dalam skrip python di Visual Studio Code.

- Inference (VS Code)
 1. File best.pt (bobot terbaik hasil training) dimuat ke dalam skrip Python.
 2. Skrip menangkap *stream* video dari IP ESP32-Cam secara nirkabel.

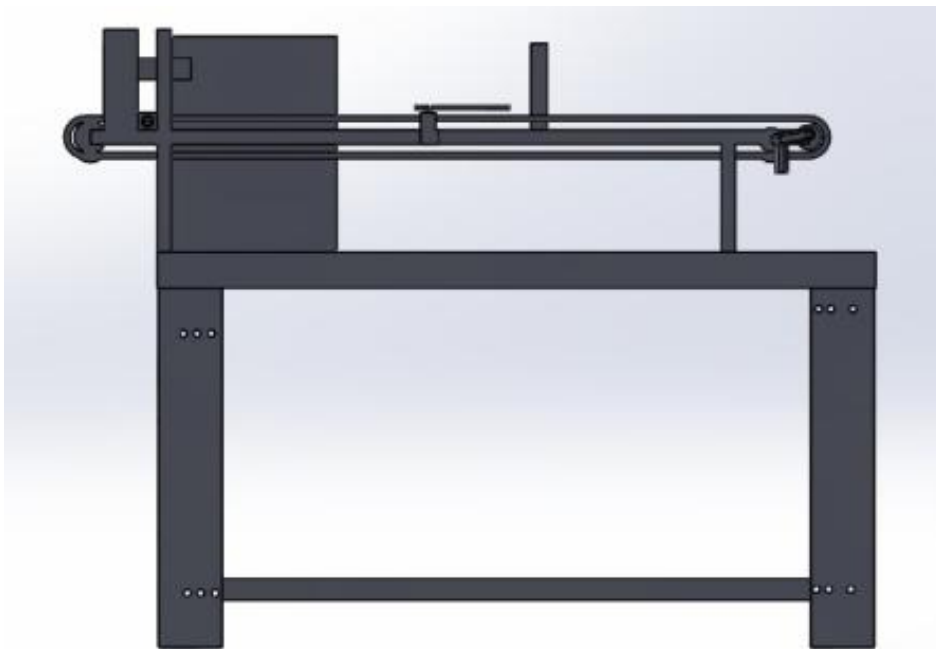
3. Parameter: *Confidence Threshold* (biasanya 0.5 atau 50%). Jika model yakin di atas 50% bahwa itu daging segar, maka label akan muncul.
- Hybrid Logic (Decision Making):
 - a. Terjadi di dalam skrip Python. Sistem membaca dua input:
 1. Visual: Output dari YOLOv5 (Segar/Tidak).
 2. Kimia: Data analog dari sensor MQ-138 (Formalin) yang dikirim Arduino ke Python via Serial.
 - b. Logic: Jika YOLO = Segar DAN MQ-138 < Ambang Batas → Daging Layak. Selain itu → Daging Ditolak.
 - Execution (Action)
 - a. Python mengirim perintah (karakter '1' atau '0') ke Arduino Mega melalui kabel USB (Serial Communication).
 - b. Arduino Mega menerima perintah:
 1. Menggerakkan Servo ke sudut 90° untuk membelokkan daging buruk ke wadah sortir.
 2. Mengontrol BTS7960 untuk mengatur kecepatan konveyor agar pas dengan durasi deteksi.

3.3 Proyeksi 3D perangkat keras sistem

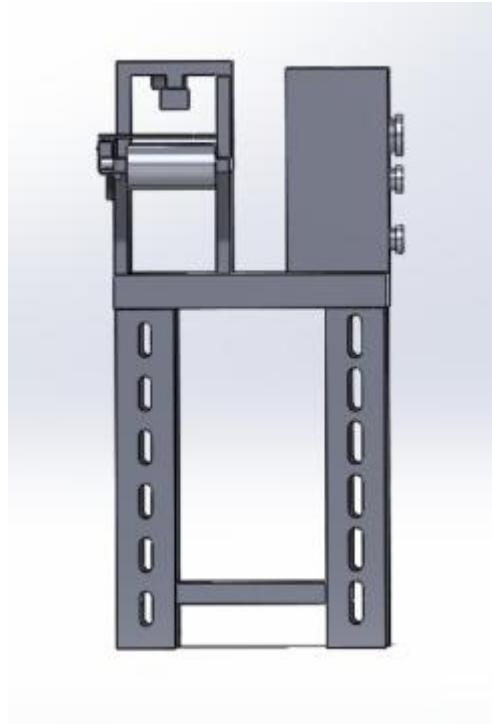
Untuk mendapatkan gambaran pada bagian perancangan dan juga sebagai patokan sistem rangka serta komponen sistem dibuatlah ilustrasi 3D alat. Pada 3D ilustrasi alat ini meliputi rangka alat yang digunakan esp 32-cam, arduino atmega 2560, sensor ir proximity, sensor gas mq-138, Serta aktuator motor DC, dan motor servo. **Gambar 3.8** merupakan ilustrasi 3D alat yang digunakan.



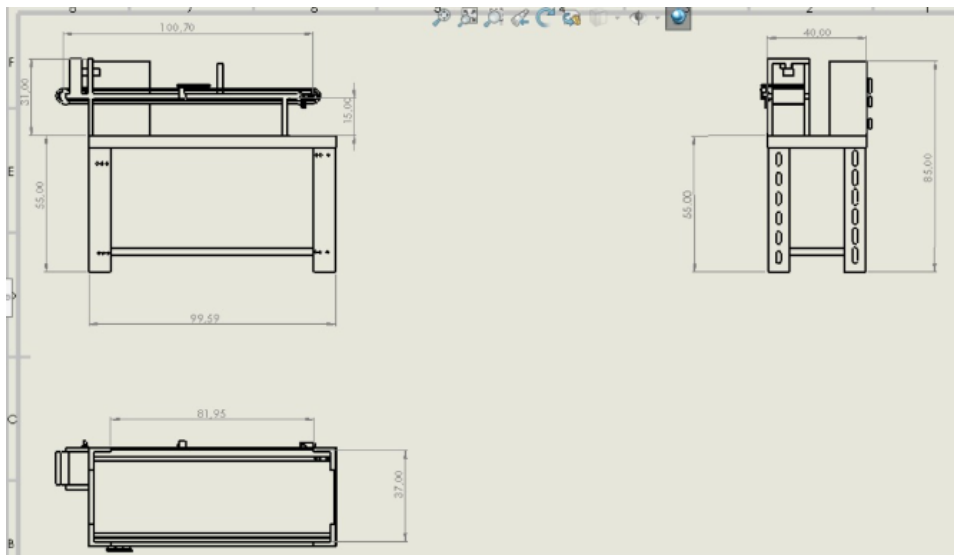
Gambar 3. 9 Model desain 3D alat



(a) Tampak samping alat Panjang 100 cm dan Tinggi 85 cm



(b) Tampak belakang dengan tinggi keseluruhan 85cm



(c) Ukuran dimensi alat keseluruhan

Gambar 3. 10 (a) Tampak samping sekitar 100cm dan tinggi 85 cm; (b) Tampak belakang dengan tinggi keseluruhan 85 cm; (c). Ukuran dimensi alat keseluruhan

3.4 Spesifikasi dan fitur

Spesifikasi dari alat yang dibuat rancang bangun sistem penguji kualitas daging sapi otomatis menggunakan esp 32-cam berbasis you only look once (yolo) yaitu sebagai berikut;

1. Menggunakan catu daya utama 220 VAC Menyediakan sumber listrik utama untuk seluruh sistem, memungkinkan sistem bekerja dengan stabil.
2. Menggunakan Power supply 12V 30A/30A Mengonversi tegangan AC menjadi DC 12V untuk menyalakan motor, mikrokontroler, dan modul lainnya dengan arus yang cukup besar.
3. Menggunakan Arduino Atmega 2560 Berfungsi sebagai pusat kendali utama (mikrokontroler), Memiliki banyak pin digital dan analog, cocok untuk sistem dengan banyak sensor dan aktuator.
4. Menggunakan ESP 32-CAM Modul kamera berbasis WiFi yang dapat menangkap gambar objek secara otomatis dan mengirimkannya untuk diproses oleh sistem deteksi objek.
5. Menggunakan Sensor gas MQ-138 Digunakan untuk mendeteksi keberadaan senyawa gas organik volatile (VOCs), seperti amonia, formaldehida.
6. Menggunakan Sensor IR Proximity Mendeteksi keberadaan objek (misalnya daging) di jalur konveyor.
7. Menggunakan Motor DC Power Window 12V dengan kemampuan RPM mencapai 50-90. Digunakan sebagai penggerak konveyor atau bagian mekanis sistem untuk memindahkan objek secara perlahan dan terkontrol.
8. Menggunakan motor servo Berfungsi untuk mengarahkan atau menyortir objek (misalnya, memindahkan daging ke tempat sesuai klasifikasi: segar/busuk).
9. Menggunakan Box
10. Menggunakan alat elektronika tambahan lainnya yang akan digunakan guna menyempurnakan sistemnya.
11. Menggunakan Stepdown LM2495 12V to 5V yang digunakan untuk Arduino, dan komponen lainnya.

Bisa dilihat pada **Tabel 3.1** untuk mengetahui konsumsi daya sistemnya

Tabel 3. 1 Konsumsi Daya Sistem

Komponen	Tegangan (V)	Arus (A)	Daya (W)
Arduino ATmega 2560	5V	0.05A	0.25
ESP32-Cam	5V	0.31A	1.55
Sensor IR Proximity	5V	0.02A	0.10
Sensor MQ-138	5V	0.18A	0.90
Driver BTS7960	5V	0.04A	0.20
Motor Servo MG995	5V	0.50A	2.50
Motor DC Power Window	12V	8.00A	96.00
Total Daya Operasional			101.5 W

Total konsumsi daya saat kondisi beban puncak (peak load) dihitung dengan persamaan daya DC ($P = V \times I$)

$$P_{total} = P_{logic} + P_{actuator}$$

$$P_{total} = (0,25 + 1,55 + 0,10 + 0,90 + 0,20 + 2,50 + 96,00)$$

$$P_{total} = 101,5 \text{ Watt}$$

Kapasitas PSU (*Power supply unit*)

$$P_{max} = V \times I = 12V \times 30A = 360 \text{ Watt}$$

Berdasarkan perhitungan beban total sebesar 101.5 Watt, maka sistem memiliki cadangan daya (*Safety Factor*) sebesar:

$$Safety\ Factor = \left(\frac{Kapasitas\ Daya\ PSU}{Total\ Beban\ Daya} \right)$$

$$Safety\ Factor = \left(\frac{360}{101.5} \right) = 3,55$$

Untuk Cadangan daya sebesar 3,55 kali lebih besar dari kebutuhan daya maka bisa dikatakan sangat aman untuk mengantisipasi lonjakan arus (*inrush current*) saat motor DC mulai berputar.

Total arus yang digunakan sistem keseluruhan $I_{total} = 9,1A$, dihitung dengan menambah semua arus komponen yang digunakan dan PSU yang digunakan juga memiliki spesifikasi yang jauh lebih besar dari arus keseluruhan jadi aman. Dengan menggunakan PSU 12V 30A ini karena:

1. Cadangan arus yang cukup dimana sistem hanya membutuhkan sekitar 9,1A sehingga penggunaan PSU 30A memberikan Cadangan arus yang sangat besar untuk menjaga sistem stabil.
2. Menghindari PSU untuk kerja pada beban maksimum bisa menyebabkan panas berlebih (*overheating*), Penurunan tegangan output (*Voltage drop*).
3. Antisipasi pengembangan sistem bisa menambah komponen tambahan selama pengerjaan.
4. Antisipasi lonjakan arus motor dc power window biasa memiliki arus sekitar 8A saat operasi normal. Namun biasanya saat start awal arus awal (*inrush current*) yang dapat mencapai 2-3 kali dari arus normalnya untuk perkiraan lonjakan arusnya dapat dihitung sebagai berikut:

$$I_{start} = 3 \times I_{nominal} = 3 \times 8$$

$$I_{start} = 24A$$

Dimana pada saat awal motor arus memungkinkan mencapai 24A.

3.5 Prinsip kerja

Sistem bekerja secara otomatis menggunakan metode validasi ganda (*hybrid*) yang menggabungkan parameter kimia dan visi komputer untuk menjamin akurasi hasil. Proses dimulai ketika motor *power window* menggerakkan sabuk konveyor sepanjang 100 cm dengan kecepatan linear 10,4 cm/s. Saat daging sapi melewati jalur dan memicu Sensor IR Proximity, Arduino Mega 2560 akan menginstruksikan *driver* BTS7960 untuk menghentikan putaran motor secara instan agar daging berhenti tepat di depan area pemindaian.

Pada posisi diam tersebut, sistem melakukan dua tahap deteksi secara berurutan:

1. Analisis Kimia (Sensor MQ-138) sensor gas melakukan pemindaian selama 7 detik untuk mengukur kadar senyawa volatil. Jika perbandingan nilai ADC udara bersih (215-228) terhadap sampel daging menunjukkan $\text{Ratio} \leq 2.5$, maka daging terdeteksi mengandung formalin.
2. Analisis Visual (YOLOv5ss) secara bersamaan, ESP32-CAM mengambil citra daging untuk diolah menggunakan algoritma YOLOv5ss selama 15 detik. Algoritma ini memiliki mAP 53,8% yang difokuskan untuk mengenali klasifikasi kesegaran berdasarkan tekstur dan warna.

Keputusan akhir merupakan integrasi dari kedua parameter tersebut. Jika salah satu atau kedua parameter menyatakan daging "Tidak Layak" (berformalin atau busuk), Arduino akan memerintahkan motor servo bergerak ke sudut 165° (Posisi Sortir) untuk menghadang dan membelokkan daging keluar dari jalur utama. Sebaliknya, jika daging dinilai segar, servo tetap berada pada sudut 125° (Posisi Standby) agar daging dapat melaju ke wadah utama. Setelah proses selesai, konveyor akan kembali aktif secara otomatis untuk memproses sampel berikutnya dengan tingkat akurasi operasional sistem mencapai 80%.

3.6 Teknik Fabrikasi

3.6.1 Alat dan Bahan

Sistem pengujian terhadap kualitas daging sapi secara otomatis membutuhkan beberapa alat dan bahan yang digunakan dalam merancang sistem yang baik dan insyaallah stabil. Berikut adalah alat dan bahan yang digunakan dilihat pada **Tabel 3.1**.

Tabel 3. 1 Alat dan bahan.

Alat dan bahan	Fungsi
Arduino Atmega 2560	Mikrokontroler yang digunakan atau otak sistem.
Esp32-cam	Mengambil gambar daging untuk diproses dengan Yolo.
Sensor MQ-138	Mendeteksi gas formalin pada daging sapi.
Sensor IR Proximity	Mendeteksi keberadaan daging untuk mengulang sistem konveyer.
Motor DC 12V Power Window	Menggerakkan konveyer agar daging berjalan ke tempat sortir.
Motor Servo	Menyortir daging segar dan tidak segar/berformalin.
Motor Driver BTS7960	Sebagai pengontrol kecepatan dan arah putaran motor DC.
Stepdown LM2596 (12V- 5V-3.3V)	Penurun tegangan ke 5V-3.3V dari 12V
Pushbutton	Memulai sistem dan menghentikan

3.6.2 Perancangan perangkat keras

Pada tahapan perancangan perangkat keras untuk membuat sistem pengujian terhadap kualitas daging sapi secara otomatis, Dilakukan tiga tahapan proses:

a. Perancangan dan Perakitan Sistem Mekanik

Pada proses ini mencakup penyusunan dan penyambungan seluruh komponen elektronika sesuai dengan rancangan sistem. Komponen seperti Arduino ATmega2560, ESP32-CAM, sensor MQ-138, sensor IR proximity, motor servo, motor DC konveyor, dan driver motor BTS7960 dirangkai berdasarkan wiring diagram yang telah dibuat sebelumnya. Power supply 12V digunakan sebagai sumber utama, kemudian diturunkan melalui step-down regulator menjadi 5V dan 3.3V sesuai kebutuhan masing-masing komponen. Penempatan setiap komponen juga akan disesuaikan dengan desain mekanik, agar terintegrasi secara efisien dalam panel box.

3.6.2.1 Perancangan Transmisi Pulley and Belt

Sistem transmisi pada conveyor ini menggunakan mekanisme *pulley* dan *belt* untuk mentransmisikan daya dari motor DC (BTS7960 driver) ke *shaft roller*. Pemilihan sistem ini didasarkan pada kelebihanannya yang mampu meredam getaran dan tidak memerlukan pelumasan intensif.

- Komponen yang digunakan *timing belt* berbahan karet dengan penguat serat baja dan dua buah *timing pulley* berbahan aluminium.
- Fungsinya untuk menghubungkan poros motor dengan poros *roller* penggerak. Rasio *pulley* dirancang untuk mendapatkan torsi yang cukup guna menarik beban daging di atas sabuk conveyor tanpa slip.

Bisa dilihat pada **Gambar3.10** dibawah.



Gambar 3. 11 Pemasangan Pulley dan belt

b. Perhitungan Pulley dan Belt (Sistem Transmisi)

Dengan menggunakan Standar GT2 (*Pitch* $P=2\text{mm}$) dengan jumlah gigi *pulley* motor (Z_1) sekitar 20 dan *pulley roller* (Z_2) = 40 (Rasionya 1:2 untuk peningkatan torsi).

- Diameter Pitch Pulley Motor (D_1):

$$D_1 = \frac{Z_1 \cdot P}{\pi} = \frac{20 \cdot 2}{3,14} \\ \approx 12,73 \text{ mm}$$

- Diameter Pitch Pulley Roller (D_2):

$$D_2 = \frac{Z_2 \cdot P}{\pi} = \frac{40 \cdot 2}{3,14} \\ \approx 25,46 \text{ mm}$$

D_1 = Diameter pitch pulley motor saat sabuk melingkar

D_2 = Diameter pitch pulley motor saat sabuk melingkar

P = Jarak antar pusat gigi pada sabuk (belt)

- Rasio Kecepatan (i):

$$i = \frac{z_2}{z_1} = \frac{40}{20} = 2, \text{Dimana setiap 2 putaran motor,}$$

Roller berputar 1x, Torsi motor dilipatgandakan 2x

i = Rasio Transmisi (*Gear Ratio*) = Perbandingan

Kecepatan antara motor dan roller

z_1 = Jumlah gigi pulley motor yang menempel langsung pada as motor DC Power Window.

z_2 = Jumlah gigi pulley yang menempel pada poros (*Shaft*) pemutar konveyer.

Artinya setiap 2 putaran motor akan menghasilkan 1 putaran roller, yang meningkatkan torsi output sebanyak 2x lipat

- Panjang Keliling Sabuk (*Belt Length – L*), Jika jarak antar poros motor ke poros *roller* (C) Adalah 100 mm:

$$L = 2C + 1,57(D_1 + D_2) + \frac{(D_2 - D_1)^2}{4C}$$

$$L = 2(100) + 1,57(12,73 + 25,46) + \frac{(25,46 - 12,73)^2}{400}$$

$$L = 200 + 59,94 + 0,40 = 260,34 \text{ mm}$$

- Daya Tarik Konveyer (F)

Dengan asumsi efisiensi motor dan transmisi (η) sebesar 80% atau 0,8. Daya pada PWM 180 sebesar 51,66 Watt,

$$\text{maka } T_{roller} = \frac{Px\eta}{\omega} = \frac{51,66 \times 0,8}{6,28} \approx 6,28 \text{ Nm}$$

Jika diameter roller 50 mm ($r = 0.025 \text{ m}$)

$$F = \frac{T_{roller}}{r} = \frac{6,28 \text{ Nm}}{0,025 \text{ m}} = 263,2 \text{ N}$$

- Estimasi Kapasitas beban *MAX*

Meenggunakan koefisien gesek (μ) antara belt dan alas sebesar 0,4 maka:

$$F = \mu \times m \times g = 263,2 = 0,4 \times m \times 9,8$$

$$m = \frac{263,2}{3,92} \approx 67 \text{ kg}$$

Jadi, dapat secara teori, motor mampu menarik beban hingga 67 kg, Namun untuk menjaga keawetan motor, beban dibatasi pada sampel untuk tidak melebihi 100 g saja karena masih rancang bangun alatnya.

c. Perhitungan Daya Elektrik (Input)

Dengan menggunakan tabel dari pengujian di bab IV titik operasional paling stabil pada PWM (150) dan titik torsi tertinggi PWM (180):

- Daya pada PWM 150 (kondisi stabil):

$$P = V \times I = 7,06 \text{ V} \times 4,8 \text{ A} = 33,88 \text{ Watt}$$

- Daya pada PWM 180 (Torsi Tinggi):

$$P = V \times I = 8,47 \text{ V} \times 6,1 \text{ A} = 51,66 \text{ Watt}$$

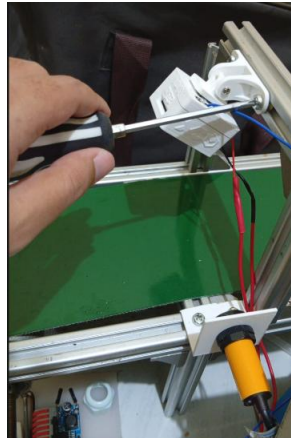
3.6.2.2 Perancangan Kamera ESP32-Cam pada Conveyor

Perancangan peletakan kamera merupakan hal krusial untuk memastikan akurasi deteksi YOLOv5s. Kamera ESP32-CAM ditempatkan pada sebuah *stand* (penyangga) permanen di atas jalur conveyor.

- Posisi kamera (ESP32-Cam) diletakkan tepat di atas titik berhentinya objek (*top-down view*) dengan ketinggian sekitar 15–20 cm dari permukaan sabuk.
- Sudut pandang (POV) dari posisi tegak lurus dipilih untuk meminimalisir distorsi gambar dan memastikan fitur tekstur serta warna daging tertangkap secara optimal untuk diolah oleh sistem AI.

- Pencahayaan pada penyangga kamera dilengkapi dengan ruang untuk modul pencahayaan eksternal guna menjaga konsistensi intensitas cahaya saat proses pengambilan data gambar (Opsional).

Berikut **Gambar 3.11** dari perancangan kamera ESP32-Cam pada *conveyer*.



Gambar 3. 12 Pemasangan ESP32-Cam di conveyor

3.6.2.3 Perancangan Shaft Roller pada *Conveyor*

Shaft roller dirancang sebagai tumpuan utama sabuk conveyor. Terdiri dari dua bagian utama: *drive roller* (penggerak) dan *idle roller* (pengikut).

- Untuk bahannya *Shaft* menggunakan batang besi/stainless steel untuk kekuatan mekanik, sedangkan *roller* menggunakan aluminium yang dilapisi bahan *friction* (karet) agar sabuk tidak mudah bergeser.
- Untuk bearing Setiap ujung *shaft* dilengkapi dengan bantalan gelinding (*ball bearing*) tipe UCFL atau sejenisnya untuk memperkecil gaya gesek saat poros berputar, sehingga kerja motor driver BTS7960 menjadi lebih ringan.

3.6.2.4 Perancangan Sortir pada *Conveyor*

Untuk mekanisme sortir dirancang untuk memisahkan daging secara fisik berdasarkan hasil klasifikasi kualitas (Segar atau Tidak Segar). Berbeda dengan sistem *pusher* (pendorong), alat ini menggunakan metode *diverter* (pengarah) dengan lengan berbentuk L yang digerakkan oleh motor servo secara interupsi.

- Lengan sortir dirancang menyerupai huruf "L" menggunakan material 3D Printing. Bentuk L dipilih agar lengan memiliki area hadang yang lebih luas dan stabil saat menahan laju daging yang sedang dibawa oleh sabuk *conveyor*.
- Mekanisme Kerja (Diverter Method) Lengan ini terhubung langsung ke *horn* motor servo yang diletakkan di sisi pinggir jalur *conveyor*. Ketika sistem memberikan sinyal 'N' (Tidak Segar), servo akan berputar sehingga lengan L yang tadinya berada di pinggir jalur akan bergerak menutup lintasan secara diagonal. Lengan tidak memukul daging, melainkan menghadang dan mengarahkan daging agar tergelincir keluar dari sabuk menuju wadah penampungan (afkir).
- Logika Gerak dan Sudut:
 1. Posisi Standby (125°) lengan berada sejajar dengan pembatas jalur *conveyor* sehingga daging segar dapat lewat dengan bebas di jalur utama.
 2. Posisi Sortir (165°) lengan berputar menutup jalur secara diagonal. Posisi sudut 165° dipilih agar membentuk sudut kemiringan tertentu yang memudahkan daging "tergelincir" keluar tanpa merusak tekstur daging.
- Unit sortir diposisikan setelah area kamera dan sensor MQ-138. Jarak peletakan diperhitungkan berdasarkan kecepatan sabuk *conveyor* agar saat hasil deteksi keluar, lengan L sudah berada pada posisi menutup sebelum daging mencapai titik sortir.

a. Torsi pada Roller (Troller)

Bisa diasumsikan jika efisiensi kemampuan konveyor dalam menarik beban jika beban daging (1-5 kg) pada kondisi PWM 180 (Torsi tinggi)

- Torsi pada roller (T roller)

Diasumsika efisiesni motor dan transmisi sekitar 80% ($\eta = 0.8$) dan kecepatan putar roller saat beban berat adalah sekitar 60 RPM ($\omega = 6,28$ rad/s): Maka,

$$T_{roller} = \frac{Px\eta}{\omega} = \frac{51,66 \times 0,8}{6,28} \approx 6,58 Nm$$

- Gaya Tarik konveyer (F)

Jika diameter roller 50 mm ($\tau = 0,025m$) maka,

$$F = \frac{T_{roller}}{\tau} = \frac{6,58Nm}{0,025m} = 263,2 N$$

- Kapasitas beban maksimal (m)

Gaya Tarik ini harus melawan gaya gesek beban daging

$$F = \mu \cdot m \cdot g$$

Jika koefisien gesek (μ) antara belt dan alas sekitar 0,4:

$$263,2 = 0,4 \cdot m \cdot 9,8$$

$$m = \frac{263,2}{3,92} \approx 67 kg$$

b. Pengujian

Setelah perakitan alat selesai, dilakukan pengujian awal untuk memastikan setiap komponen bekerja dengan baik. Pengujian ini meliputi:

1. Pengecekan koneksi kabel dan kestabilan daya dari PSU ke semua modul.
2. Pemeriksaan sinyal input dari sensor proximity dan sensor MQ-138 ke Arduino.
3. Pengujian komunikasi antara ESP32-CAM dengan server (cloud/Colab),

4. Uji respons motor servo dan motor DC saat perintah sortir dan conveyor diaktifkan.

Hasil ini akan menentukan apakah sistem perlu diperbaiki sebelum masuk ke tahap kalibrasi.

c. Kalibrasi

Tahap ini bertujuan untuk menyelaraskan akurasi dan sensitivitas tiap perangkat sensorik serta respons aktuator secara real-time.

1. Sensor IR proximity dikalibrasi agar dapat mendeteksi keberadaan daging secara akurat tanpa terpicu oleh objek selain daging.
2. Sensor MQ-138 dikalibrasi berdasarkan ambang batas kadar formalin.
3. Kalibrasi motor servo dilakukan untuk mengatur sudut dorong ke arah kotak sortir secara presisi.
4. Sistem deteksi gambar dengan ESP32-CAM juga diuji ulang agar posisi pengambilan gambar optimal, serta sinkronisasi waktu capture dengan proses deteksi YOLOv5s berjalan lancar.

Kalibrasi dilakukan secara bertahap hingga sistem dapat bekerja stabil dan memberikan klasifikasi hasil deteksi dengan akurasi yang dapat diterima.

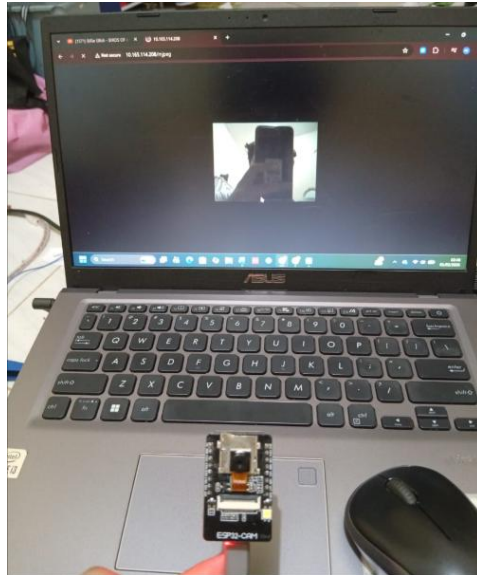
3.6.3 Perancangan sistem YOLO

Tahapan perancangan sistem YOLO (*You Only Look Once*) merupakan bagian penting dalam proses deteksi kualitas daging sapi secara otomatis. Sistem ini dirancang untuk bekerja secara efisien dan real-time dalam mengidentifikasi kondisi daging menggunakan pendekatan *object detection*. Dalam tahapan perancangan YOLOv5s, terdapat beberapa komponen utama yang saling terintegrasi, yaitu:

3.6.3.1 ESP32-Cam

ESP32-CAM merupakan modul kamera berbasis mikrokontroler ESP32 yang digunakan untuk mengambil gambar

daging secara langsung dari jalur conveyor. Kamera ini terintegrasi dengan sistem untuk mengambil citra saat sensor *IR proximity* mendeteksi keberadaan objek. Gambar yang diambil dikompresi dalam format JPEG dan dikirim melalui koneksi WiFi ke server atau layanan cloud untuk dilakukan pemrosesan lebih lanjut.



(a) Pengujian Esp32-Cam



(b) Pengujian Esp32-Cam

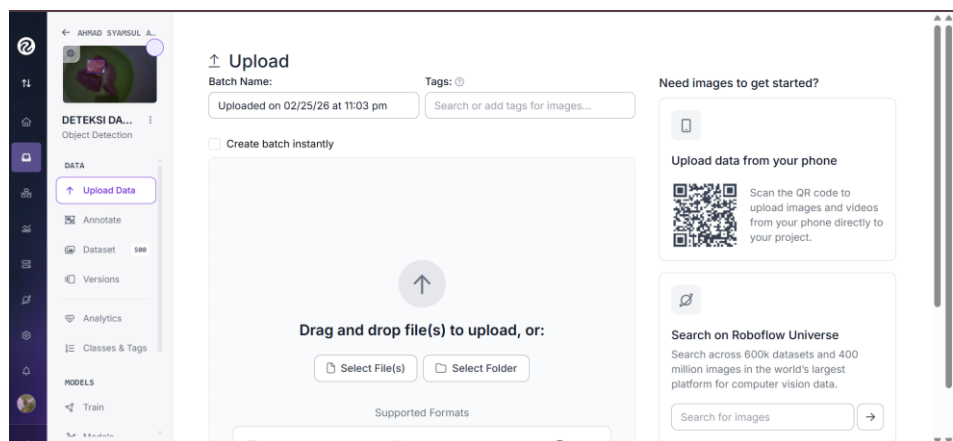


(c) Pengujian Esp32-Cam

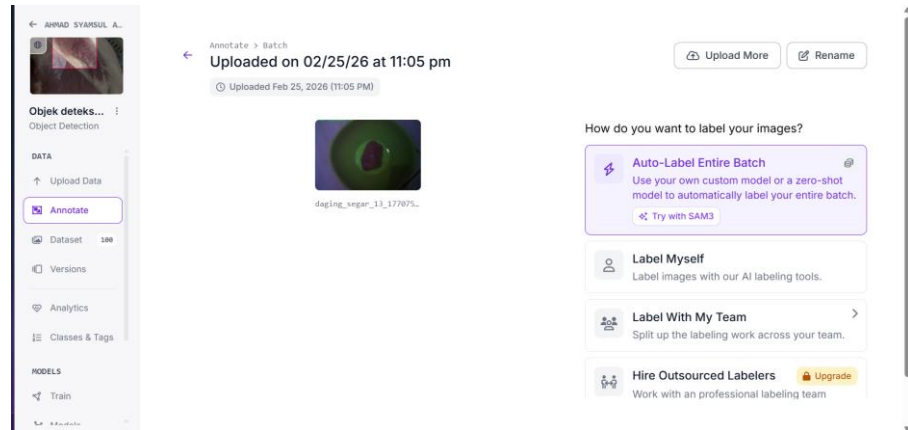
Gambar 3. 13) Pengambilan gambar dan tampilan dengan menggunakan ESP32-Cam

3.6.3.2 Roboflow

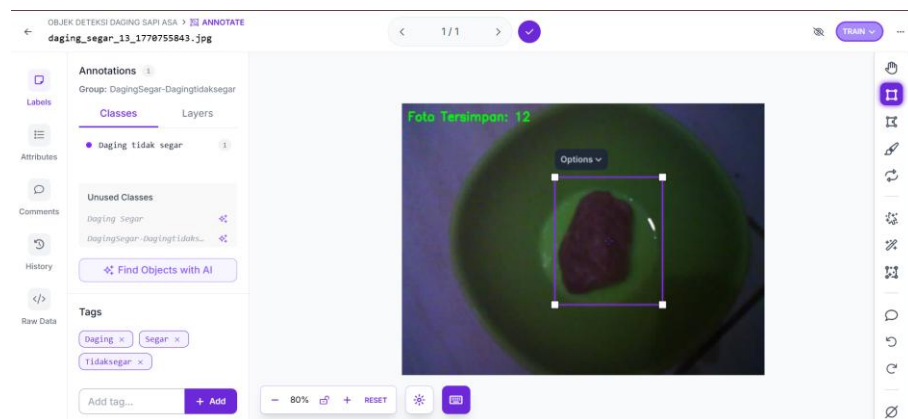
Roboflow digunakan sebagai platform untuk pelabelan data, augmentasi, dan ekspor dataset dalam format yang kompatibel dengan model YOLO. Pada tahap ini, mengumpulkan gambar daging segar dan tidak segar, kemudian dilakukan proses labeling menggunakan bounding box. Setelah itu, dataset diproses melalui fitur preprocessing (seperti *resize*, *flip*, *rotate*) untuk memperkaya variasi data latih, lalu diekspor ke format YOLOv5s dan diunduh untuk pelatihan model.



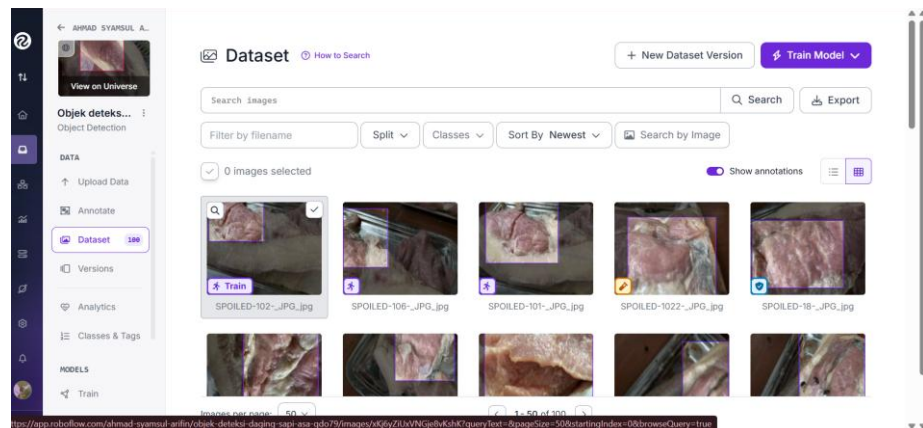
(a) Tampilan pembuatan Model YOLOv5s



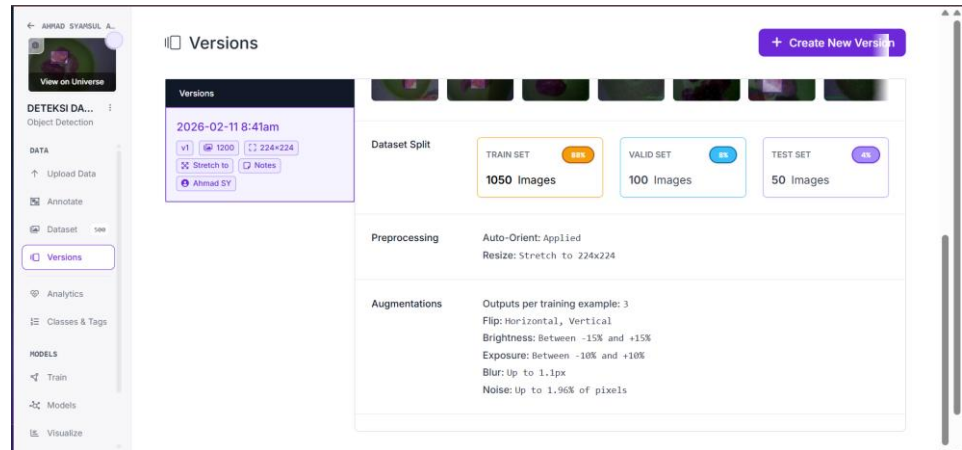
(b) Tampilan pembuatan Model YOLOv5s



(c) Tampilan pembuatan Model YOLOv5s



(d) Tampilan pembuatan Model YOLOv5s



(e) Tampilan pembuatan Model YOLOv5s

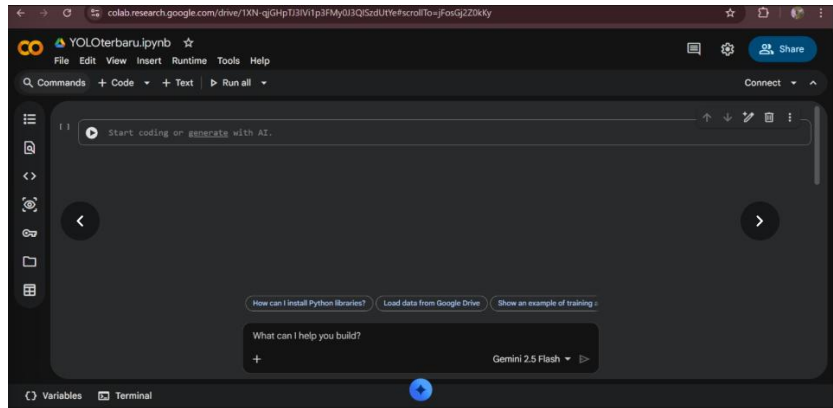
Gambar 3. 14 Merupakan tahap pembuatan model pada roboflow

3.6.3.3 *Firestore (opsional)*

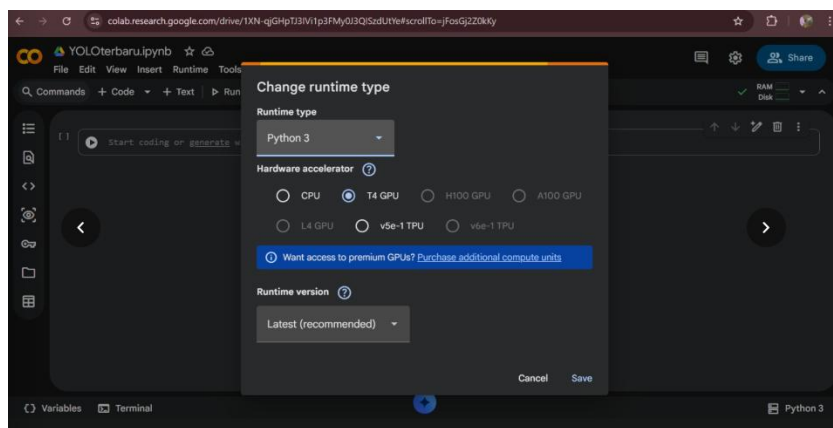
Firestore digunakan sebagai media perantara penyimpanan gambar dari ESP32-CAM ke sistem *cloud*. Gambar yang diambil oleh ESP32-CAM akan di-*upload* ke Firestore Storage secara otomatis. Kemudian, URL gambar ini digunakan oleh Google Colab untuk mengambil citra dan menjalankan proses deteksi menggunakan model YOLO. Firestore juga dapat difungsikan sebagai penyimpan sementara dan integrasi sistem komunikasi antar perangkat.

3.6.3.4 *Google Colab*

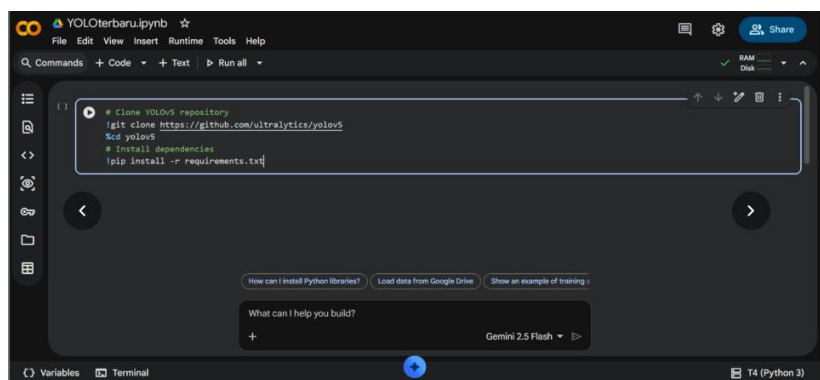
Google Colab berfungsi sebagai platform komputasi berbasis Python untuk menjalankan YOLOv5s. Di tahap ini, file YOLO (meliputi model .pt, file konfigurasi, dan label.yaml) dimuat ke dalam notebook Colab. Gambar dari Firestore diunduh dan diproses dengan OpenCV, kemudian dianalisis menggunakan model YOLO. Hasilnya berupa prediksi kelas (segardaging atau tidaksegardaging) yang dikirim kembali ke Arduino sebagai dasar pengambilan keputusan sortir.



(a) Tampilan pembuatan Model YOLOv5s



(b) Tampilan pembuatan Model YOLOv5s



(c) Tampilan pembuatan Model YOLOv5s

```

Downloading ultralytics-thop-2.0.18-py3-none-any.whl (28 kB)
Installing collected packages: urllib3, ultralytics-thop, thop, ultralytics
Attempting uninstall: urllib3
Found existing installation: urllib3 2.5.0
Uninstalling urllib3-2.5.0:
Successfully uninstalled urllib3-2.5.0
Successfully installed thop-0.1.1.post2209072238 ultralytics-8.4.14 ultralytics-thop-2.0.18 urllib3-2.6.3

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="VF855QIV8S-iB8N2n")
project = rf.workspace("rahad-systemul-arifin").project("deteksi-daging-sapi-baru")
version = project.version(1)
dataset = version.download("yolov5")

```

(d) Tampilan pembuatan Model YOLOv5s

```

28/99 0.8810 0.05161 0.01279 0.001127 30 224: 6X 4/66 [00:00:00:05, 11.13it/s]/cont
with torch.cuda.amp.autocast(amp):
28/99 0.8810 0.05129 0.01273 0.00126 21 224: 9% 6/66 [00:00:00:05, 10.57it/s]/content/yolov5/train.py:4
with torch.cuda.amp.autocast(amp):
28/99 0.8810 0.05226 0.01347 0.001241 34 224: 9% 6/66 [00:00:00:05, 10.57it/s]/content/yolov5/train.py:4
with torch.cuda.amp.autocast(amp):
28/99 0.8810 0.05217 0.01358 0.001435 32 224: 12% 8/66 [00:00:00:05, 10.53it/s]/content/yolov5/train.py:4
with torch.cuda.amp.autocast(amp):
28/99 0.8810 0.05271 0.01378 0.001359 32 224: 12% 8/66 [00:00:00:05, 10.53it/s]/content/yolov5/train.py:4
with torch.cuda.amp.autocast(amp):
28/99 0.8810 0.05231 0.01375 0.001519 23 224: 15% 10/66 [00:00:00:05, 10.14it/s]/content/yolov5/train.py:4
with torch.cuda.amp.autocast(amp):
28/99 0.8810 0.05204 0.01373 0.00143 26 224: 15% 10/66 [00:01:00:05, 10.14it/s]/content/yolov5/train.py:4
with torch.cuda.amp.autocast(amp):
28/99 0.8810 0.05217 0.014 0.001523 34 224: 18% 12/66 [00:01:00:05, 10.58it/s]/content/yolov5/train.py:4
with torch.cuda.amp.autocast(amp):
28/99 0.8810 0.05214 0.01394 0.001442 22 224: 18% 12/66 [00:01:00:05, 10.58it/s]/content/yolov5/train.py:4
with torch.cuda.amp.autocast(amp):
28/99 0.8810 0.05191 0.01376 0.001383 28 224: 21% 14/66 [00:01:00:05, 10.29it/s]/content/yolov5/train.py:4
with torch.cuda.amp.autocast(amp):
28/99 0.8810 0.05163 0.01368 0.001328 23 224: 21% 14/66 [00:01:00:05, 10.29it/s]/content/yolov5/train.py:4
with torch.cuda.amp.autocast(amp):
28/99 0.8810 0.05208 0.0136 0.001306 23 224: 24% 16/66 [00:01:00:04, 10.40it/s]/content/yolov5/train.py:4

```

(e) Tampilan pembuatan Model YOLOv5s

Gambar 3. 15 Tahap lanjutan dari roboflow agar model yolo sempurna dibuat

3.6.3.5 Visual Code Studio

Visual Studio Code berfungsi sebagai VS Code sangat penting karena Arduino Mega memiliki keterbatasan memori untuk menjalankan algoritma AI yang berat. VS Code memungkinkan PC mengambil alih tugas pengolahan citra (dari ESP32-CAM) dan mengirimkan perintah instruksi balik ke Arduino melalui komunikasi Serial. Tanpa perangkat lunak ini, pemrosesan visual berbasis AI tidak dapat dilakukan.

```

1 import torch
2 import cv2
3 import numpy as np
4 import requests
5 import time
6
7 # =====
8 # 1. KONFIGURASI UTAMA
9 # =====
10 # Ganti dengan IP yang muncul di Serial Monitor Arduino IDE
11 url = "http://10.247.34.288:81/"
12 model_path = "best.pt"
13
14 # --- MENGENAL MODEL AI ---
15 # Reload model YOLOv5 menggunakan file best.pt kamu
16 model = torch.hub.load('ultralytics/yolov5', 'custom', path=model_path)
17
18 # =====
19 # =====
20 # =====
21 # =====
22 # =====
23 # =====
24 # =====
25 # =====
26 # =====
27 # =====
28 # =====
29 # =====
30 # =====
31 # =====
32 # =====
33 # =====
34 # =====
35 # =====
36 # =====
37 # =====
38 # =====
39 # =====
40 # =====
41 # =====
42 # =====
43 # =====
44 # =====
45 # =====
46 # =====
47 # =====
48 # =====
49 # =====
50 # =====
51 # =====
52 # =====
53 # =====
54 # =====
55 # =====
56 # =====
57 # =====
58 # =====
59 # =====
60 # =====
61 # =====
62 # =====
63 # =====
64 # =====
65 # =====
66 # =====
67 # =====
68 # =====
69 # =====
70 # =====
71 # =====
72 # =====
73 # =====
74 # =====
75 # =====
76 # =====
77 # =====
78 # =====
79 # =====
80 # =====
81 # =====
82 # =====
83 # =====
84 # =====
85 # =====
86 # =====
87 # =====
88 # =====
89 # =====
90 # =====
91 # =====
92 # =====
93 # =====
94 # =====
95 # =====
96 # =====
97 # =====
98 # =====
99 # =====
100 # =====

```

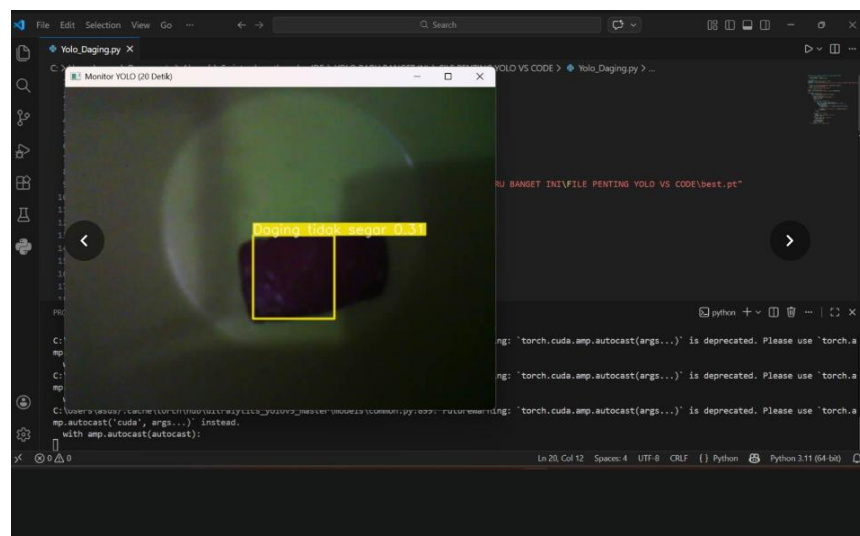
Gambar 3. 16 Tampilan Vscode setelah model berhasil dibuat

3.6.3.6 Python

Bahasa pemrograman Python digunakan untuk menulis seluruh skrip dalam Google Colab. Python juga digunakan untuk:

1. Membaca gambar dari Firebase,
2. Melakukan preprocessing menggunakan OpenCV,
3. Memuat dan menjalankan model YOLOv5s,
4. Mengekstrak bounding box dan confidence score,
5. Melakukan post-processing dengan metode *Non-Maximum Suppression*.

Suppression.



(a) Tampilan Model YOLOv5s

