

**RANCANG BANGUN APLIKASI *MOBILE* CERDAS
PENDETEKSI BERAT BADAN SAPI
MENGUNAKAN YOLOV5**

SKRIPSI



**OLEH
MOHD. ABDUL GHANI
H071181004**

**PROGRAM STUDI SISTEM INFORMASI DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN**

MAKASSAR

2024



**RANCANG BANGUN APLIKASI *MOBILE* CERDAS
PENDETEKSI BERAT BADAN SAPI
MENGUNAKAN YOLOV5**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer pada Program Studi Sistem Informasi Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin**

MOHD. ABDUL GHANI

H071181004

**PROGRAM STUDI SISTEM INFORMASI DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN**

MAKASSAR

JUNI 2024



PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama : Mohd. Abdul Ghani

NIM : H071181004

Program Studi : Sistem Informasi

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul :

Rancang Bangun Aplikasi *Mobile* Cerdas Pendeteksi Berat Badan Sapi Menggunakan YOLOV5

Adalah karya tulisan saya sendiri, bukan merupakan pengambil alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, Juni 2024



Mohd. Abdul Ghani
H071181004



**RANCANG BANGUN APLIKASI *MOBILE* CERDAS PENDETEKSI
BERAT BADAN SAPI MENGGUNAKAN YOLOV5**

Disusun dan diajukan oleh :

Mohd. Abdul Ghani

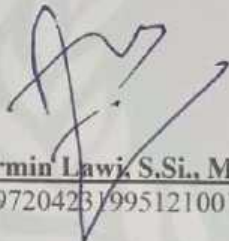
H071181004

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar *Sarjana Komputer* pada Program studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Menyetujui,

Pembimbing Utama,

Pembimbing Pertama,


Dr. Eng. Armin Lawi, S.Si., M.Eng.
NIP. 197204231995121001


Edy Saputra Rusdi, S.Si., M.Si.
NIP. 199104102020053001

Ketua Program Studi,


Dr. Khaeruddin, M.Sc.
NIP. 196509141991031003



HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Mohd. Abdul Ghani

NIM : H071181004

Program Studi : Sistem Informasi

Judul Skripsi : Rancang Bangun Aplikasi *Mobile* Cerdas Pendeteksi Berat Badan Sapi Menggunakan YOLOV5

Telah berhasil mempertahankan di hadapan dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

Ketua : Dr. Eng. Armin Lawi, S.Si., M.Eng.

(.....)

Sekretaris : Edy Saputra Rusdi, S.Si., M.Si.

(.....)

Anggota : Dr. Hendra, S.Si., M.Kom.

(.....)

Anggota : Muhammad Sadno, S.Si., M.Si.

(.....)

Ditetapkan di : Makassar

: Juni 2024



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat SWT, atas segala limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul **“Rancang Bangun Aplikasi *Mobile* Cerdas Pendeteksi Berat Badan Sapi Menggunakan YOLOV5”**. Pembuatan skripsi ini merupakan salah satu syarat untuk penyelesaian studi penulis pada jenjang pendidikan Strata Satu Program Studi Sistem Informasi, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin.

Penulis mengucapkan terima kasih setulus tulusnya kepada Ibunda tercinta **Tati**, serta kepada kakak tercinta **Nurdin** terima kasih atas segala pengorbanan, doa, dan dukungan yang tak ternilai hingga penulis dapat menyelesaikan studi.

Selama proses penyelesaian tugas akhir ini penulis menerima begitu banyak bantuan dari berbagai pihak baik berupa materi maupun non materi, untuk itu pada kesempatan kali ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Rektor Universitas Hasanuddin Makassar **Prof. Dr. Ir. Jamaluddin Jompa, M.Sc.**, dan seluruh Wakil Rektor dalam Lingkungan Universitas Hasanuddin.
2. Bapak Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam **Dr. Eng Amiruddin** dan para Wakil Dekan serta seluruh staf yang telah memberikan bantuan selama penulis mengikuti pendidikan di FMIPA Universitas Hasanuddin.
3. Bapak **Dr. Firman, S.Si., M.Si.** selaku Ketua Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.



4. Bapak **Dr. Khaeruddin, M.Sc.** selaku Ketua Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.

5. Bapak **Dr. Eng. Armin Lawi, S.Si., M.Eng.** selaku dosen pembimbing utama dan Bapak **Edy Saputra Rusdi, S.Si, M.Si.** selaku dosen pembimbing pertama atas segala ilmu, nasehat, dan motivasi yang telah diberikan serta kesabaran dalam membimbing penulis dalam proses penyusunan skripsi ini.
6. Bapak **Dr. Hendra, S.Si., M.Kom.** dan bapak **Muhammad Sadno, S.Si., M.Si.** sebagai tim penguji atas saran dan masukan pada penelitian yang telah dilakukan oleh penulis.
7. Terkhusus **Musfika Ilyas** terima kasih atas dukungan dan bantuannya serta kebersamaan yang diberikan kepada penulis hingga dapat menyelesaikan skripsi ini.
8. Semua pihak yang telah memberikan bantuan kepada penulis baik berupa materi dan non materi yang tidak dapat penulis sebutkan satu per satu, terima kasih untuk bantuan dan dukungannya.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan karena keterbatasan penulis. Oleh karena itu, saran dan kritik demi penyempurnaan skripsi ini sangat penulis harapkan. Akhir kata, semoga skripsi ini membawa manfaat dan semoga Allah SWT membalas semua kebaikan semua pihak yang telah membantu.

Makassar, Juni 2024

Penulis



PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : Mohd. Abdul Ghani
NIM : H071181004
Program Studi : Sistem Informasi
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin Hak Bebas Royalti Non Eksklusif (Non-exclusive Royalty Free Right) atas karya ilmiah saya yang berjudul:

“Rancang Bangun Aplikasi *Mobile* Cerdas Pendeteksi Berat Badan Sapi Menggunakan YOLOV5”

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal Juni 2024

Yang Menyatakan



Abdul Ghani

ABSTRAK

Salah satu parameter penting bagi peternak sapi adalah berat badan karena mencerminkan kondisi tubuh sapi, pembibitan dan nutrisi. Penggunaan alat timbang konvensional menjadi kurang efisien dan dapat menyebabkan stres pada sapi. Oleh karena itu, merancang aplikasi *mobile* pendeteksi berat badan sapi menjadi sangat penting. Penelitian ini menganalisis kinerja YOLOv5 dalam mendeteksi berat badan sapi, merancang aplikasi *mobile*, dan mengintegrasikan model tersebut ke dalam aplikasi. Metode yang digunakan meliputi pelatihan model YOLOv5 dengan dataset gambar sapi dari Kaggle, yang dibagi menjadi data latih dan uji dengan rasio 70:30, selama 200 *epochs* dengan *batch size* 32, menghasilkan mAP50 sebesar 0.995, mAP50-90 sebesar 0.88, *precision* sebesar 0.999, dan *recall* sebesar 1. Kemudian model dikonversi menjadi *TensorFlow Lite* untuk diintegrasikan ke aplikasi *mobile*. Evaluasi menunjukkan bahwa model YOLOv5 memiliki performa baik dalam mendeteksi bagian tubuh sapi. Namun, terdapat perbedaan signifikan antara berat badan aktual dan prediksi, akibat variasi jarak sapi dengan kamera. Evaluasi hasil deteksi berat badan sapi menunjukkan MAPE sebesar 14.30% dan MAE sebesar 24.87 kg. YOLOv5 mampu mendeteksi bagian tubuh sapi dengan akurat, dan integrasi model ke aplikasi *mobile* berhasil dilakukan. Namun deteksi berat badan sapi masih dipengaruhi oleh jarak dan kondisi pengambilan gambar, sehingga diperlukan pengembangan lebih lanjut.

Kata Kunci: YOLOv5, deteksi berat sapi, aplikasi *mobile*



ABSTRACT

One important parameter for cattle farmers is body weight as it represents the cow's body condition, breeding and nutrition. The use of conventional weighing tools is less efficient and can cause stress to the cattle. Therefore, designing a cattle weight detection mobile application is very important. This study analyzes the performance of YOLOv5 in detecting cattle weight, designs a mobile application, and integrates the model into the application. The method used includes training the YOLOv5 model with a dataset of cow images from Kaggle, which is divided into training and test data with a ratio of 70:30, for 200 epochs with a batch size of 32, resulting in mAP50 of 0.995, mAP50-90 of 0.88, precision of 0.999, and recall of 1. Then the model is converted into TensorFlow Lite to be integrated into the mobile application. The evaluation showed that the YOLOv5 model performed well in detecting cattle body parts. However, there is a significant difference between actual and predicted body weight, due to variations in the distance between the cow and the camera. Evaluation of cow weight detection results showed a MAPE of 14.30% and MAE of 24.87 kg. YOLOv5 is able to accurately detect cattle body parts, and the integration of the model into mobile applications is successful. However, cow weight detection is still affected by distance and image capture conditions, so further development is needed.

Keywords: *YOLOv5, cow weight detection, mobile application*



DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	ii
LEMBAR PERNYATAAN KEASLIAN	iii
LEMBAR PERSETUJUAN PEMBIMBING	iv
HALAMAN PENGESAHAN.....	v
KATA PENGANTAR	vi
PERSETUJUAN PUBLIKASI ILMIAH	viii
ABSTRAK	ix
ABSTRACT.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1. Penelitian Relevan.....	4
2.2. Berat Badan Sapi	6
2.3. <i>You Only Look Once</i> v5 (YOLOv5).....	7
2.3.1. <i>Backbone</i>	10
2.3.2. <i>Neck</i>	11
2.3.3. <i>Head</i>	11
2.3.4. <i>Activation Function</i>	13
2.3.5. <i>Loss Function</i>	14
Transfer Learning	15
Non-maximum Suppresion (NMS).....	16
Evaluasi Kinerja Model.....	16



2.6.1.	<i>Precision</i>	17
2.6.2.	<i>Recall</i>	17
2.6.3.	<i>Intersection Over Union (IoU)</i>	17
2.6.4.	<i>Mean Average Precision (mAP)</i>	19
2.7.	Evaluasi Prediksi Berat Sapi	19
2.7.1.	<i>Mean Absolute Error (MAE)</i>	20
2.7.2.	<i>Mean Absolute Percentage Error (MAPE)</i>	20
2.8.	Rancang Bangun Aplikasi <i>Mobile</i> (Android).....	21
2.8.1.	Android	21
2.8.2.	<i>TensorFlow Lite</i>	21
BAB III METODE PENELITIAN.....		23
3.1.	Waktu dan Lokasi Penelitian.....	23
3.2.	Perangkat Penelitian	23
3.2.1.	Perangkat Keras	23
3.2.2.	Perangkat Lunak.....	23
3.3.	Sumber Data	23
3.4.	Tahapan Penelitian	24
3.4.1.	<i>Preprocessing</i>	24
3.4.2.	Pembagian Data	25
3.4.3.	Augmentasi Data.....	25
3.4.4.	Pelatihan Model	25
3.4.5.	Evaluasi Kinerja Model.....	25
3.4.6.	Konversi Model.....	25
3.4.7.	Perancangan Aplikasi.....	25
3.5.	Integrasi Model Pada Rancangan Aplikasi <i>Mobile</i>	27
BAB IV PEMBAHASAN HASIL PENELITIAN.....		28
4.1.	Hasil Penelitian	28
4.1.1.	Tampilan Awal Aplikasi	35
4.1.2.	Tampilan Dari Tombol <i>Open Camera</i>	36
3.	Tampilan Tombol <i>Input Gambar</i>	37
4.	Tampilan Tombol <i>Detect Weight</i>	37
Pembahasan Penelitian		38



BAB V KESIMPULAN DAN SARAN.....	44
5.1. Kesimpulan.....	44
5.2. Saran.....	45
DAFTAR PUSTAKA	46
LAMPIRAN.....	51



DAFTAR TABEL

Tabel 2.1 Perbandingan Akurasi Model	8
Tabel 4.1 Hasil evaluasi kinerja model YOLOv5.....	33
Tabel 4.2 Hasil validasi model setelah dikonversi menjadi <i>Tensorflow Lite</i>	35
Tabel 4.3 Perhitungan statistik.....	40
Tabel 4.4 Outlier	41
Tabel 4.5 Penghapusan outlier berdasarkan error (%) > 40%	42



DAFTAR GAMBAR

Gambar 2.1 Bagian tubuh sapi yang sering digunakan untuk mengukur berat badan sapi (a) Panjang badan (b) Lingkar dada (c) Tinggi pundak.....	6
Gambar 2.2 (a) Lingkar Dada (b) Tinggi Pundak	7
Gambar 2.3 Grafik tingkat akurasi dan kecepatan varian YOLOv5.....	9
Gambar 2.4 Arsitektur YOLOv5	9
Gambar 2.5 Arsitektur <i>Backbone</i> Model	10
Gambar 2.6 Struktur Blok SPPF	11
Gambar 2.7 Kotak pembatas dengan dimensi sebelumnya dan prediksi lokasi ...	12
Gambar 2.8 Fungsi aktivasi ReLU dan SiLU	14
Gambar 2.9 Contoh penerapan NMS	16
Gambar 2.10 Contoh <i>intersection over union</i>	18
Gambar 2.11 Ilustrasi <i>TensorFlow Lite</i>	22
Gambar 3.1 Diagram alur penelitian	24
Gambar 3.2 Rancangan aplikasi <i>mobile</i>	26
Gambar 3.3 <i>Use case</i> diagram	26
Gambar 3.4 Alur integrasi model.....	27
Gambar 4.1 Data citra RGB	28
Gambar 4.2 Gambar sapi yang telah diberi anotasi	29
Gambar 4.3 Hasil resize citra ukuran 640x640 pixel	29
Gambar 4.4 Struktur <i>folder</i> dataset	30
Gambar 4.5 Isi <i>data.yaml</i>	30
Gambar 4.6 Ilustrasi proses augmentasi	31
Gambar 4.7 Perintah untuk menjalankan pelatihan model YOLOv5	31
Gambar 4.8 Kurva hasil training YOLOv5.....	32
Gambar 4.9 Kurva loss validation.....	32
Gambar 4.10 Hasil train dan validation model YOLOv5	33
Gambar 4.11 Confusion matrix model YOLOv5.....	33
Gambar 4.12 Plot hasil <i>training</i> model YOLOv5.....	34
Gambar 4.13 Perintah untuk mengonversi model YOLOv5.....	34
Gambar 4.14 Perintah untuk melakukan validasi model	35
Gambar 4.15 Struktur direktori proyek.....	35
Gambar 4.16 Tampilan awal aplikasi.....	36
Gambar 4.17 Fungsi tombol <i>Open Camera</i> dan fungsinya (a) Tampilan kamera (b) Tampilan dialog konfirmasi (c) Tampilan setelah tombol simpan ditekan.....	37
Gambar 4.18 Fungsi tombol <i>Input Gambar</i> dan prosesnya (a) Tampilan awal (b) Tampilan hasil dari <i>input</i> gambar	37
Gambar 4.19 Tampilan saat tidak ada gambar yang dideteksi.....	38
Gambar 4.20 Tampilan hasil deteksi.....	38
Gambar 4.21 Perbandingan awal berat badan sapi aktual dan hasil prediksi	40
4.22 Distribusi penyimpangan persentase.....	41
4.23 Setelah membersihkan outlier	42
4.24 Pengaruh jarak objek dengan kamera (a) Berat aktual 71 kg (b) Berat aktual 481 kg	43



BAB I

PENDAHULUAN

1.1. Latar Belakang

Produk hewani merupakan salah satu sumber makanan yang paling penting bagi populasi dunia. Permintaan akan protein hewani mengalami peningkatan, khususnya di negara-negara yang berkembang (Pretto dkk., 2022). Selain itu, adanya keberadaan produk hewani secara langsung akan meningkatkan status gizi masyarakat, terutama dalam hal pemenuhan kebutuhan kalori dan protein hewani. Hal ini akan berdampak positif pada kualitas sumber daya manusia (BPS, 2022).

Pertumbuhan populasi sapi di Indonesia mengalami peningkatan dalam 5 tahun terakhir. Rata-rata pertumbuhan populasi sapi di Indonesia setiap tahunnya sejak 2015 hingga 2021 mencapai 2.68% atau sebesar 440 ribu ekor. Pada tahun 2015 populasi sapi potong di Indonesia mencapai 15.42 juta ekor dan terus meningkat secara positif hingga mencapai 17.44 juta ekor pada tahun 2020, dan pada tahun 2021 mencapai 18.05 juta ekor (BPS, 2022).

Berat badan sapi merupakan parameter yang sangat penting dalam manajemen ternak. Hal ini dikarenakan berat badan sapi dapat membantu peternak dalam mengevaluasi berbagai aspek seperti konsumsi pakan, potensi perkembangbiakan, perilaku sosial, dan juga dapat mencerminkan kondisi tubuh sapi (Z. Wang dkk., 2021). Selain itu berat badan sapi juga memiliki kaitan yang erat dengan aspek ekonomi dalam jual beli sapi.

Umumnya, untuk mengetahui berat badan sapi dapat dilakukan pengukuran secara langsung menggunakan alat timbang konvensional. Proses ini melibatkan alat timbang yang cukup berat yang telah ditempatkan pada lokasi yang telah ditentukan, seperti lorong khusus penimbangan, atau diletakkan di samping tempat makan dan minum ternak. Hewan ternak ini kemudian digiring ke lokasi penimbangan untuk mengukur berat badan ternak (Z. Wang dkk., 2021). Pengukuran dengan cara ini mampu memberikan hasil yang akurat, namun proses tersebut kurang praktis, memakan waktu yang lama, dan cukup menyusahkan dalam prosesnya. Selain itu, proses pengukuran ini bisa menyebabkan cedera hewan ternak, dan juga bisa saja melukai peternak (Thapar dkk., 2023).



Deep learning, salah satu cabang dari *machine learning* telah menjadi salah satu isu yang hangat untuk dibahas karena pencapaiannya yang mengesankan dalam menangani berbagai kasus yang kompleks seperti mengidentifikasi wajah dan mendeteksi objek (Dohmen dkk., 2021). *Convolutional Neural Network* yang merupakan salah satu dari metode *Deep learning* menjadi metode yang dominan untuk hampir semua kasus deteksi dan identifikasi, serta mendekati kinerja manusia dalam beberapa kasus (LeCun dkk., 2015).

YOLOv5 merupakan model deteksi objek seri kelima dari model *You Only Look Once* (YOLO) yang dikembangkan oleh *Ultralytics* dengan menggunakan *PyTorch* sebagai *framework deep learning* yang dikembangkan oleh *Facebook*. YOLOv5 memiliki kecepatan dan akurasi dalam mendeteksi objek yang tinggi, serta ukuran yang relatif ringan dengan spesifik *weight files* sebesar 27mb (Solawetz, 2020). Ini sesuai dengan kebutuhan perangkat *mobile* yang sering memiliki kapasitas penyimpanan dan daya komputasi yang terbatas.

YOLOv5 telah digunakan untuk berbagai tugas deteksi objek. Oliveira dkk. menggunakan YOLOv5 untuk mendeteksi dan menghitung hama pada buah zaitun, dalam hal ini lalat buah zaitun (*Bactrocera oleae*) (Oliveira dkk., 2022). Hu dkk. menggunakan YOLOv5 untuk mengenali perilaku atau kebiasaan domba dengan skenario di padang rumput (Hu dkk., 2023). Penelitian lainnya juga dilakukan oleh H. Li, dengan basis YOLOv5, mendeteksi penyakit daun apel secara *real-time* (H. Li dkk., 2023).

T. Li dkk. melakukan deteksi pengenalan dan lokasi berbasis YOLOv5s dengan menggunakan struktur modul CARAFE untuk meningkatkan kemampuan jaringan *receptive field* dalam memahami konteks lokal dalam gambar, namun penggunaan sumber daya komputasi tetap ringan, *Quality Focal Loss* dan *EIoU* juga digunakan untuk meningkatkan kinerja jaringan dengan memperbaiki masalah dalam pelatihan dan pengukuran performa (T. Li dkk., 2023).

Berdasarkan penelitian-penelitian tersebut, peneliti tertarik untuk melakukan penelitian dengan menggunakan model *You Only Look Once v5*

5) untuk mendeteksi berat badan sapi. Sehingga peneliti memutuskan mbuat penelitian yang berjudul “Rancang Bangun Aplikasi *Mobile* cerdas si berat badan sapi menggunakan YOLOv5”.



1.2. Rumusan Masalah

Berdasarkan uraian pada latar belakang masalah di atas, dapat dirumuskan masalah:

1. Bagaimana melatih dan mengevaluasi kinerja model YOLOv5 dalam mendeteksi berat badan sapi?
2. Bagaimana merancang aplikasi *mobile* untuk mendeteksi berat badan sapi?
3. Bagaimana mengintegrasikan model YOLOv5 untuk mendeteksi berat badan sapi ke dalam aplikasi *mobile* berbasis *Android*?

1.3. Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Data yang digunakan yaitu dataset berupa gambar sapi yang diperoleh dari *kaggle* (<https://www.kaggle.com/datasets/sadhliroomyprime/cattle-weight-detection-model-dataset-12k/>).
2. Dataset tidak memiliki informasi mengenai panjang tubuh, lingkar dada sapi dan jarak antara sapi dan kamera, hanya memiliki informasi mengenai berat badan sapi yang tertera pada nama fail, dengan tidak memperhitungkan lebih dalam seperti ras, usia, jumlah gigi, dan lain-lain.
3. Citra yang digunakan hanya tampak dari samping, dan diambil berdasarkan di berbagai kondisi waktu dan cuaca.
4. Aplikasi *mobile* yang digunakan berupa *android*.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan dari penelitian ini adalah:

1. Menganalisis kinerja model YOLOv5 dalam mendeteksi berat badan sapi.
2. Merancang aplikasi *mobile* untuk mendeteksi berat badan sapi.
3. Mengintegrasikan model yang telah dibuat menggunakan model YOLOv5 ke dalam aplikasi *mobile* berbasis *Android*.

1.5. Manfaat Penelitian

Berdasarkan tujuan penelitian, maka manfaat dari penelitian ini adalah :

1. Menjadi sumber informasi mengenai kinerja model YOLOv5 dalam mendeteksi berat badan sapi.
2. Menyediakan aplikasi *mobile* berbasis *android* yang dapat mendeteksi berat badan sapi.



BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Relevan

Penelitian ini mengacu pada beberapa penelitian sebelumnya. Berikut adalah penelitian terdahulu yang berkaitan dengan penelitian ini, antara lain :

Thapar dkk. dalam jurnal yang berjudul “*Accurate Estimation of Body Weight of Pigs Through Smartphone Image Measurement App*” bertujuan untuk mengembangkan metode pengukuran dimensi tubuh dan prediksi berat badan pada babi jenis Ghongroo (usia 30 hingga 180 hari) dengan menggunakan aplikasi pengukuran objek pada *smartphone*, yaitu “*On 3D CameraMeasure*”. Kemudian mengevaluasi pengukuran berdasarkan citra dan membandingkannya dengan metode pengukuran manual. Penelitian ini menyimpulkan bahwa pengukuran berdasarkan citra memiliki korelasi yang tinggi dan positif dengan pengukuran manual untuk sebagian besar dimensi tubuh yang diukur. Metode yang diusulkan mampu memprediksi berat badan babi dengan akurasi yang tinggi, dan hubungan antara berat badan babi yang sebenarnya dengan berat badan babi yang diprediksi sangat kuat (korelasi 0.99) (Thapar dkk., 2023).

Penelitian yang dilakukan oleh Gjergji dkk., dengan judul “*Deep Learning Techniques for Beef Cattle Body Weight Prediction*” pada tahun 2020, bertujuan untuk menganalisis beberapa model, seperti *Convolutional Neural Networks*, *Recurrent Neural Networks*, *Recurrent Attention Models*, dan *Recurrent Attention Models with Convolutional Neural Networks*, dengan menggunakan gambar yang dikumpulkan dari sapi potong *Embrapa* di Campo Grande MS, Brazil, dengan masing-masing 10 ekor sapi jantan jenis *Nellore* dan *Angus* yang tersebar dalam dua kandang. Berdasarkan hasil penelitian yang dilakukan oleh Gjergji dkk., ditemukan bahwa performa tertinggi dihasilkan oleh model *Convolutional Neural Networks*, dengan *Mean Average Error* (MAE) 23.19 kg (Gjergji dkk., 2020).

Penelitian yang lainnya dilakukan oleh Mustafid dan ‘Uyun, dengan judul “Sistem Pengolahan Citra Digital Untuk Menentukan Bobot Sapi Menggunakan Teknik Berat” pada tahun 2018, bertujuan untuk membandingkan dua rumus yang digunakan, yaitu rumus *Schoorl* dan rumus Modifikasi/*Lambourne*, serta lingkaran beberapa algoritma yang diusulkan untuk menentukan panjang



badan dan lingkaran dada dalam penelitian tersebut. Berdasarkan hasil penelitian, ditemukan bahwa penyimpangan rumus Modifikasi/Lambourne lebih kecil dibandingkan dengan rumus Schoorl, dengan nilai deviasi bobot badan untuk rumus Schoorl sebesar 16.87% dan untuk rumus Modifikasi/Lambourne sebesar 10.58%. Hasil analisis perbandingan beberapa algoritma yang diusulkan tidak berbeda secara signifikan, dengan MAE (*Mean Absolute Error*) sebesar 8.15% untuk panjang badan dan 4.10% untuk lingkaran dada. Kemudian aplikasi pengolahan citra yang telah dibangun mampu mengetahui berat sapi dengan MAE (*Mean Absolute Error*) sebesar 8.97% dengan menggunakan rumus Modifikasi/Lambourne (Mustafid & 'Uyun, 2018).

Penelitian lainnya dilakukan oleh Weber dkk. (2020), dengan judul “*Cattle Weight estimation using active contour models and Regression Trees Bagging*” bertujuan untuk secara otomatis mengekstraksi pengukuran pada gambar 2D punggung sapi jenis *Nellore* untuk memprediksi berat sapi menggunakan algoritma regresi. Penelitian ini memanfaatkan jarak *euclidean* dari titik-titik yang dihasilkan oleh model *Active contour* serta fitur yang diperoleh dari *dorsal convex*. Penelitian tersebut membandingkan tiga algoritma regresi, yaitu algoritma *Bagging*, *regression by discretization* dan algoritma *Random forest*. Penelitian ini membuktikan, algoritma *Bagging* memiliki hasil yang terbaik, dengan *Mean Absolute Error* (MAE) sebesar 13.44 kg (± 2.76), *Square Root of The Mean Error* (RMSE) sebesar 15.88 kg (± 2.86), *Mean Absolute Percentage Error* (MAPE) sebesar 2.27% dan koefisien korelasi sebesar 0.75 (Weber dkk., 2020).

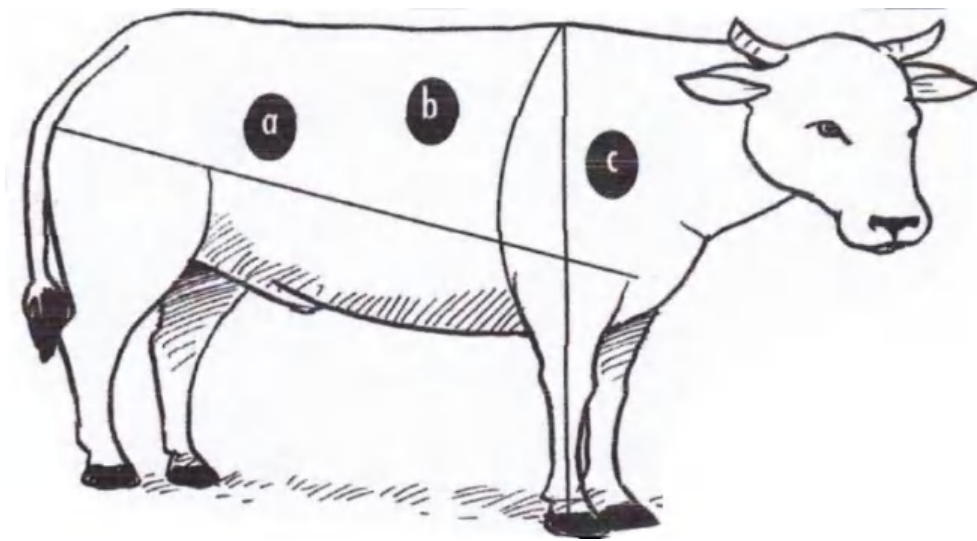
Penggunaan YOLOv5 pada bidang peternakan pernah diteliti oleh Qiao dkk. dengan judul “*Cattle Body Detection Based On YOLOv5-ASFF for Precision Livestock Farming*”. Penelitian ini menggunakan model YOLOv5-ASFF untuk mendeteksi bagian-bagian tubuh sapi (seperti kepala, tubuh dan kaki) dengan berbagai kondisi seperti masalah pencahayaan, *depth of field* yang berbeda, dan beberapa sapi dalam satu *frame*, serta gambar sapi yang tampak kecil. Penelitian tersebut menyimpulkan bahwa model YOLOv5-ASFF mampu mendeteksi objek sapi dengan akurasi mencapai 96.2%, *recall* 92%, *F1-score* 94.1% dan *mean Precision* (mAP) sebesar 94.57% (Qiao dkk., 2023).



2.2. Berat Badan Sapi

Berat sapi merupakan salah satu indikator penting bagi peternak. Pasalnya dengan mengetahui berat badan sapi, peternak dapat mengetahui kondisi tubuh sapi, dapat mengambil keputusan dalam pemberian jumlah pakan, harga penjualan dan pembelian sapi, dan juga dalam pengambilan keputusan untuk pemilihan bibit sapi.

Selain pengukuran berat badan sapi yang dilakukan secara langsung, berat badan sapi juga dapat dilakukan dengan cara tidak langsung, dalam hal ini penggunaan pita ukur untuk mengukur lingkar dada sapi, dan juga panjang badan sapi.



Gambar 2.1 Bagian tubuh sapi yang sering digunakan untuk mengukur berat badan sapi (a) Panjang badan (b) Lingkar dada (c) Tinggi pundak

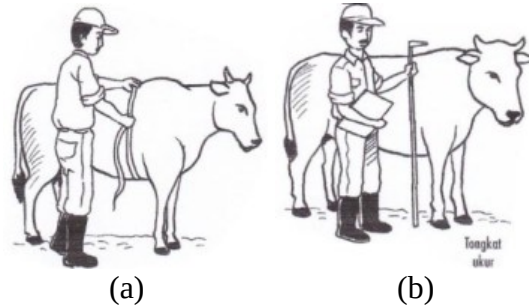
Sumber: Kementerian Pertanian

Proses dalam mengukur lingkar dada dengan melingkarkan bagian tubuh sapi dimulai dari pundak sapi hingga bagian belakang kaki depan sapi. Kemudian untuk mengukur panjang tubuh sapi, dimulai dari belakang kaki depan sapi, hingga ke pinggang sapi. Perhitungan dari tinggi lingkar dada dan panjang tubuh ini diidentifikasi sebagai prediktor terbaik untuk berat badan sapi (Thapar dkk., 2023). Yang kemudian nilai dari lingkar dada dan panjang tubuh sapi dapat dihitung dengan menggunakan beberapa rumus yang telah ada, seperti penggunaan rumus rumus *winter*.



pendugaan bobot badan tidak selalu sama dengan nilai aktualnya secara berkali-kali terjadi kesalahan atau penyimpangan. Namun, yang lebih disukai

adalah pendugaan dengan penyimpangan yang paling kecil. Dalam kasus perbandingan antara bobot badan aktual (hasil penimbangan) dan bobot yang dihitung dengan menggunakan rumus *winter* menghasilkan penyimpangan yang lebih kecil daripada rumus *schoorl* (Meidina dkk., 2021).



Gambar 2.2 (a) Lingkar Dada (b) Tinggi Pundak

Sumber: Kementerian Pertanian

Persamaan 2.1 menunjukkan rumus *Winter* untuk menghitung berat badan sapi:

$$BB = \frac{LD \times LD \times PB}{300} \quad (2.1)$$

Keterangan :

BB : Berat Badan (*Pound*)

LD : Lingkar dada (*Inchi*)

PB : Panjang Badan (*Inchi*)

$$1 \text{ inch} = 2,54 \text{ cm} , 1 \text{ cm} = \frac{\text{cm}}{\text{inches}} = \frac{1}{2,54} = 0,393701 \text{ inch}$$

$$1 \text{ kg} = 2,205 \text{ pound} , 1 \text{ pound} = \frac{\text{pound}}{\text{kg}} = \frac{1}{2,205} = 0,453592 \text{ kg}$$

Soenarjo (1998) mendeskripsikan bahwa penyimpangan pendugaan bobot badan yang bisa diterima adalah yang penyimpangannya berkisar 2% - 6% (Meidina dkk., 2021).

2.3. *You Only Look Once* v5 (YOLOv5)

YOLOv5 merupakan salah satu model deteksi objek yang sangat populer dalam bidang visi komputer. Model ini dikenal karena kemampuannya untuk mendeteksi objek secara *real-time*, yang berarti mampu mengenali dan memetakan

ek dalam gambar atau video secara cepat dan efisien.

Salah satu fitur utama dari YOLOv5 adalah kecepatannya yang tinggi dalam deteksi objek dengan tingkat akurasi yang baik. Sehingga sangat cocok untuk



berbagai aplikasi, termasuk pengawasan keamanan, mobil otonom, analisis lalu lintas, pemantauan dan pengelolaan di bidang peternakan dan pertanian, dan masih banyak lagi.

YOLOv5 merupakan model deteksi objek yang merupakan iterasi ke lima dari seri model YOLO. Model ini pertama kali diperkenalkan oleh Redmon dkk pada tahun 2015 (Redmon dkk., 2015). Kemudian pada tahun 2016, Redmon dkk merilis YOLOv2 atau YOLO9000. Versi ini merupakan perbaikan dari versi sebelumnya, termasuk peningkatan dalam akurasi dan kemampuan mendeteksi lebih banyak kelas objek. Iterasi berikutnya pada tahun 2018, YOLOv3 dirilis. Versi ini menambahkan lebih banyak lapisan ke jaringan untuk meningkatkan akurasi. YOLOv4 dianggap sebagai salah satu yang paling canggih, dikarenakan versi ini membawa peningkatan signifikan dalam kecepatan dan akurasi, serta menambah fitur-fitur seperti deteksi objek dengan tingkat kepercayaan yang tinggi.

YOLOv5 memiliki beberapa varian, seperti *nano* (n), *small* (s), *medium* (m), *large* (l) dan *extra large* (x) (Solawetz, 2020), ini memungkinkan untuk memilih model yang sesuai dengan sumber daya perangkat yang digunakan. Tabel 2.1 menunjukkan beberapa perbedaan dari 5 varian YOLOv5, berdasarkan ukuran, mAP, kecepatan, *params*, dan FLOPs.

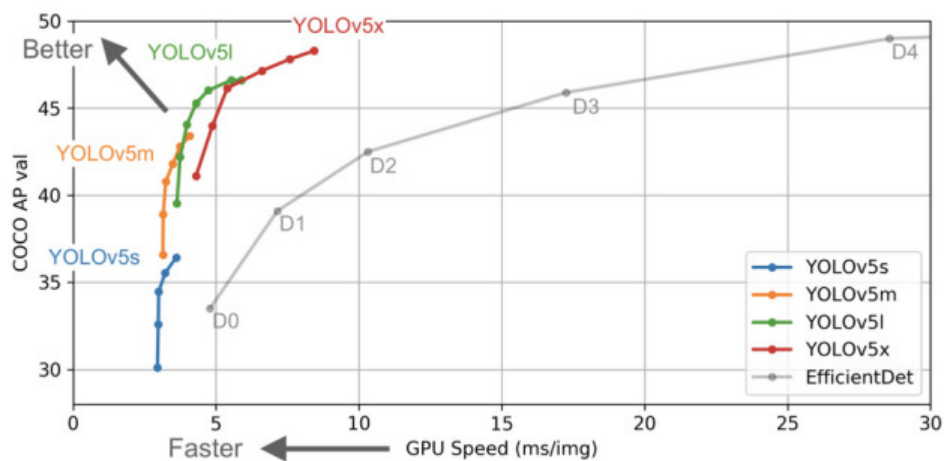
Tabel 2.1 Perbandingan Akurasi Model

Model	size (pixels)	mAP val 50-95	mAP val 50	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	45	63	6	19	45
YOLOv5s	640	37.4	56.8	98	64	9	72	165
YOLOv5m	640	45.4	64.1	224	82	17	212	490
YOLOv5l	640	49.0	67.3	430	101	27	465	1091
YOLOv5x	640	50.7	68.9	766	121	48	867	2057

Sumber: YOLOv5 Github Repo

Masing-masing dari varian YOLOv5 memberikan tingkat akurasi yang tinggi, dengan kecepatan yang berbeda-beda. Seperti tampak pada Gambar 2.3.

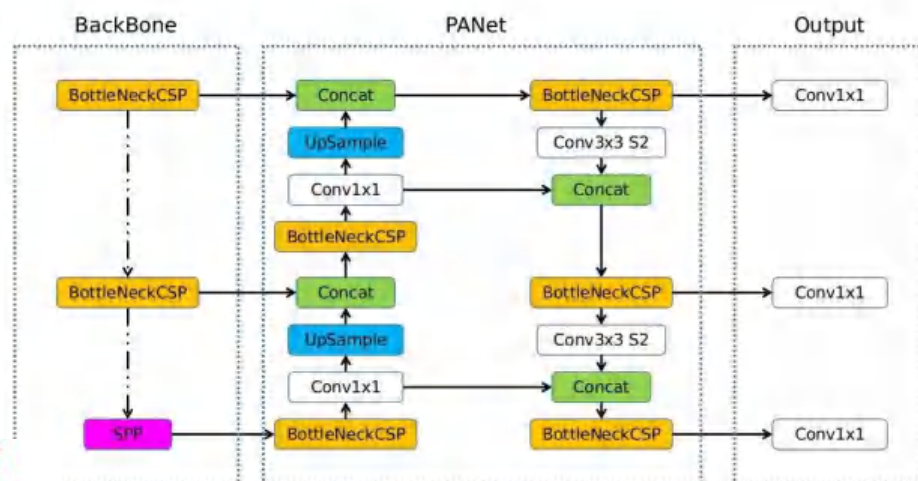




Gambar 2.3 Grafik tingkat akurasi dan kecepatan varian YOLOv5

Sumber: Roboflow blog

YOLOv5 dengan konsisten melanjutkan desain algoritma yang menjadi ciri khas dari seri YOLO, dimulai dari proses *input* gambar, diteruskan ke *backbone* untuk ekstraksi fitur, kemudian menghasilkan map fitur dengan berbagai resolusi yang nantinya digabung melalui jaringan fusi fitur (*neck*), ini akan menghasilkan tiga map fitur dengan ukuran 80x80, 40x40 dan 20x20 yang kemudian dikirim ke *head* untuk dilakukan perhitungan konfidensi dan perubahan *bounding-box* pada setiap *pixel* dalam map fitur dengan menggunakan *anchor* yang telah ditetapkan sebelumnya dan menghasilkan multi-dimensi *layer* yang mencakup informasi tentang kelas objek, *class confidence*, koordinat *box*, tinggi dan lebar objek. proses ini disebut sebagai proses inferensi (H. Liu dkk., 2022).



Gambar 2.4 Arsitektur YOLOv5

Sumber: YOLOv5 Github Repo



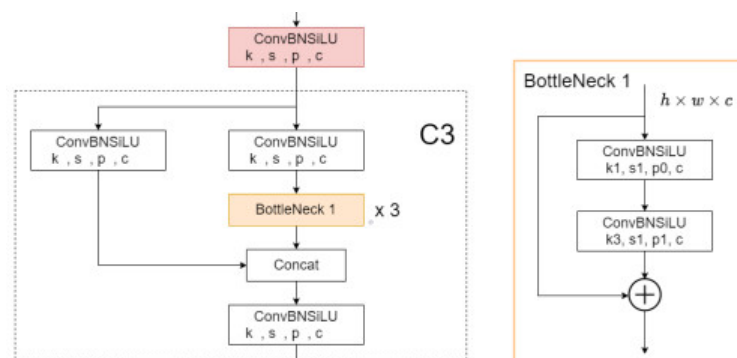
YOLOv5 memiliki 3 komponen utama seperti yang telah disebutkan sebelumnya, yaitu *Backbone* yang menggunakan *CSP-Darknet53*, *Neck* yang menggunakan struktur SPPF dan CSP-PAN, kemudian *Head* menggunakan YOLOv3 *Head*.

2.3.1. Backbone

Bagian utama dari struktur jaringan YOLOv5 adalah *Backbone*. Bagian ini menjadi tempat untuk *pre-trained network* mengekstrak berbagai representasi fitur dari gambar. Sehingga membantu mengurangi resolusi spasial gambar dan meningkatkan resolusi fitur (saluran). Jaringan *Backbone* adalah jaringan saraf *convolutional* yang menggabungkan berbagai gambar *fine-grained* dan membentuk fitur seperti *Focus*, conv, C3 dan *Spatial Pyramid Pooling* (SPP) (Qiao dkk., 2023).

Seperti yang disebutkan sebelumnya, *CSP-Darknet53* digunakan sebagai *backbone* dari YOLOv5. *CSP-Darknet53* merupakan jaringan saraf *convolutional* dengan menggunakan *DarkNet-53* untuk mendeteksi objek. *Cross Stage partial network* (CSPNet) telah menyelesaikan masalah mengenai informasi gradien berulang dalam *backbone* berskala besar, dan mengintegrasikan perubahan gradien ke dalam map fitur, sehingga mengurangi parameter dan *floatation-point operations per second* model (FLOPS), yang tidak hanya memastikan kecepatan dan akurasi inferensi, tetapi juga mengurangi ukuran model (C.-Y. Wang dkk., 2020).

Dengan memanfaatkan pendekatan *Cross Stage Partial Network* (CSPNet), map fitur pada lapisan dasar dipecah menjadi dua bagian dan kemudian menggabungkan bagian tersebut ke dalam hierarki multi-tahap. Penggunaan strategi *split* dan *merge* memungkinkan aliran gradien menyebar melalui jaringan (Bochkovskiy dkk., 2020).



Gambar 2.5 Arsitektur Backbone Model

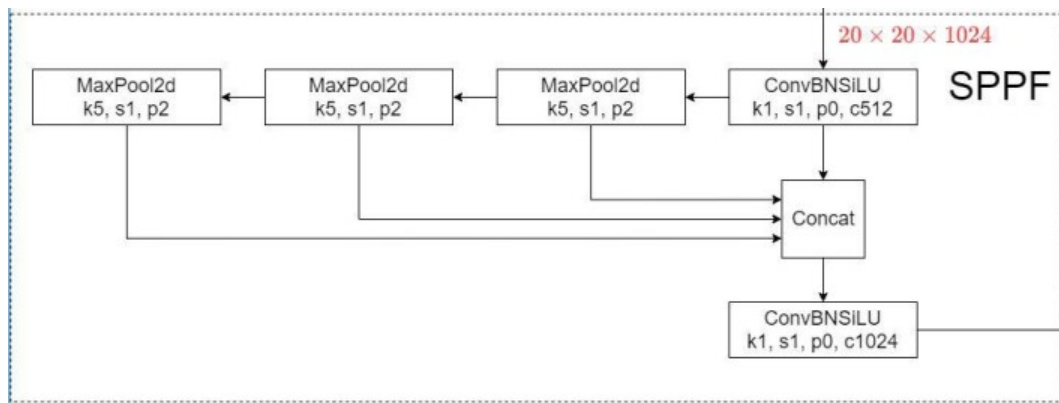
Sumber: YOLOv5 Github Repo



2.3.2. Neck

Bagian *Neck* menghubungkan antara *backbone* dan *head*. Bagian ini menggunakan SPPF dan CSP-PAN. CSP-PAN menggunakan pendekatan dari *path aggregation network* (PANet) untuk meningkatkan aliran informasi (S. Liu dkk., 2018). PANet merupakan struktur *feature pyramid network* (FPN) yang baru dengan jalur fusi fitur *bottom-up* yang telah disempurnakan, yang mampu meningkatkan akurasi dalam mendeteksi objek dengan skala yang berbeda (H. Liu dkk., 2022).

Spatial pyramid pooling (SPP) merupakan *pooling layer* yang digunakan untuk mengubah persyaratan *fixed-sized* (contohnya seperti 224x224) gambar *input* (He dkk., 2014). Sehingga dengan SPP ini, gambar *input* bisa menjadi berbagai ukuran. Pada YOLOv5, menggunakan varian SPP yang lainnya, yaitu SPPF untuk meningkatkan kecepatan jaringan (Chen & Chen, 2023).



Gambar 2.6 Struktur Blok SPPF

Sumber: YOLOv5 Github repo

2.3.3. Head

YOLOv5 menggunakan *Head* yang sama seperti YOLOv3 dan YOLOv4. Bagian ini terdiri dari tiga lapisan konvolusi yang memprediksi lokasi *bounding-box* (x , y , tinggi, dan lebar), skor, dan kelas objek. Koordinat (x, y) menggambarkan pusat kotak dalam kaitannya dengan batas sel *grid*. Lebar dan tinggi diprediksi berdasarkan ukuran total gambar. Akhirnya, prediksi *confidence* mencerminkan *intersection over union* (IoU) antara kotak yang diprediksi dan kotak aktualnya (Liu dkk., 2015).

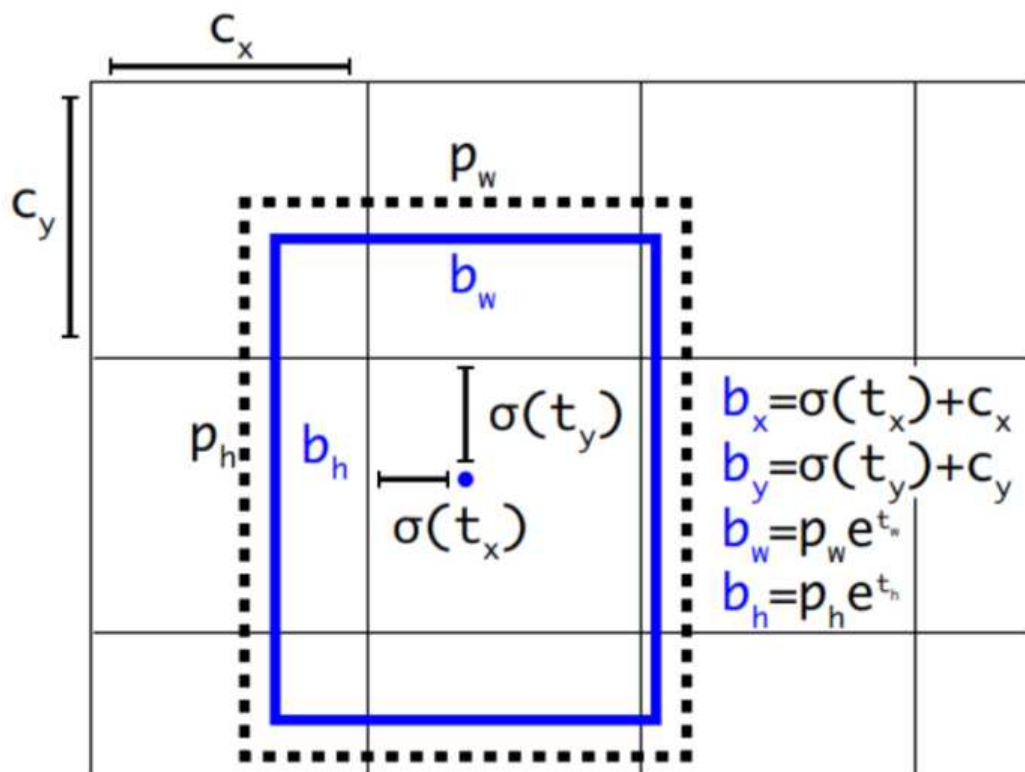


Salah satu usulan jaringan, untuk memprediksi nilai t_x dan t_y serta koordinat pusat (x, y) menggunakan Persamaan 2.2 dan Persamaan 2.3:

$$x = (t_x \times w_a) - x_a \quad (2.2)$$

$$y = (t_y \times h_a) - y_a \quad (2.3)$$

Rumus ini tidak membatasi *anchor box* sehingga bisa berakhir di mana saja pada gambar, terlepas dari lokasi mana yang memprediksi kotak tersebut. Misal, prediksi $t_x = 1$ akan menggeser kotak ke kanan sejauh lebar *anchor box*, sedangkan prediksi $t_x = -1$ akan menggeser kotak ke kiri sejauh jumlah yang sama. Alih-alih memprediksi pergeseran, YOLO memprediksi koordinat relatif terhadap lokasi pada sel *grid*. Sehingga ini bisa membatasi *ground truth* tetap berada di antara 0 dan 1 (Redmon & Farhadi, 2017).



Gambar 2.7 Kotak pembatas dengan dimensi sebelumnya dan prediksi lokasi

Jaringan memprediksi 5 *bounding box* di setiap sel pada *output feature map*. Setiap *bounding box* memprediksi 5 koordinat, yaitu t_x, t_y, t_w, t_h dan t_0 . Pada Gambar 2.7 memperlihatkan lebar dan tinggi kotak pembatas yang diprediksi sebagai *offsets* atau pergeseran posisi dari titik pusat kluster. Jika sel tersebut 1 atau bergeser dari sudut kiri atas gambar sejauh (c_x, c_y) dan *bounding* memiliki lebar dan tinggi p_w, p_h , maka prediksi tersebut sesuai dengan & Farhadi, 2017):



$$b_x = \sigma(t_x) + c_x \quad (2.4)$$

$$b_y = \sigma(t_y) + c_y \quad (2.5)$$

$$b_w = p_w e^{t_w} \quad (2.6)$$

$$b_h = p_h e^{t_h} \quad (2.7)$$

2.3.4. Activation Function

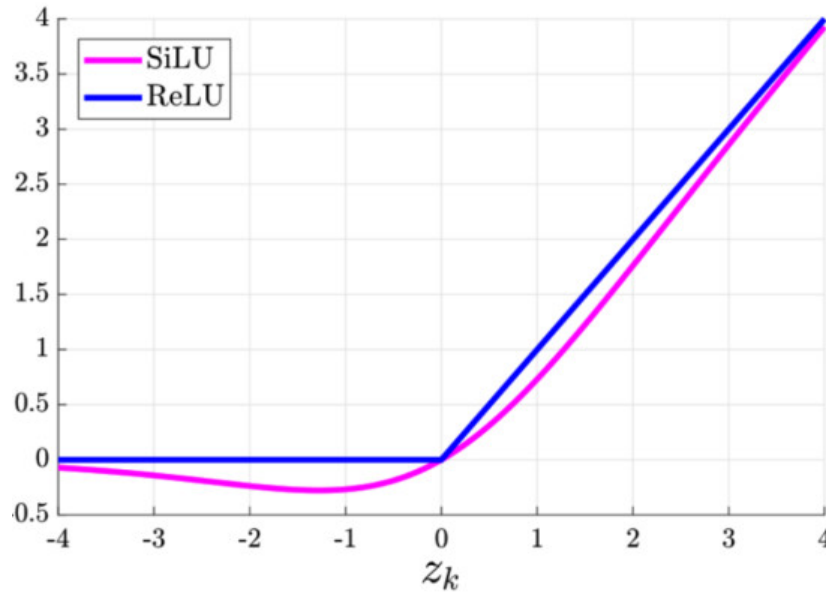
Activation function dalam *artificial neural network* berperan sebagai “portal” matematis, dengan konsep utamanya sebagai *perceptron*. Fungsi aktivasi ini memegang peranan krusial sebagai komponen dalam kerangka *deep learning*. selain itu, *activation function* juga berkontribusi pada efisiensi komputasi selama pelatihan model, dan menentukan *output* dari model *deep learning* yang dapat memengaruhi akurasi secara keseluruhan.

Activation function juga dianggap sebagai bagian yang sangat penting dalam arsitektur *neural network*. Ini dikarenakan fungsi aktivasi memiliki dampak yang besar pada kemampuan *neural network*, yang artinya *activation function* dapat memengaruhi seberapa cepat dan seberapa baik jaringan saraf tiruan dapat mempelajari pola dari data pelatihan. Tujuan utama dari fungsi aktivasi adalah mengubah sinyal *input* dari sebuah *node* dalam jaringan saraf tiruan menjadi sinyal *output* (El-Amir & Hamdy, 2020).

Activation function berada di antara *input* untuk diberikan kepada *neuron* dan *output* yang dikirimkan ke lapisan berikutnya. Fungsi ini dapat memiliki sifat yang sangat sederhana, seperti menghidupkan atau mematikan *output neuron* sesuai dengan aturan atau ambang batas tertentu. Sebaliknya, *activation function* juga dapat mengambil bentuk transformasi yang kompleks, mengubah sinyal *input* menjadi sinyal *output* yang diperlukan agar jaringan saraf dapat berfungsi dengan optimal.

YOLOv5 menggunakan *Sigmoid-Weighted Linear Unit* (SiLU) atau juga disebut dengan *swish* sebagai *activation function*. Ini dikarenakan SiLU memberikan rentang *output* jaringan menjadi antara 0 dan 1, yang mana ini membuat SiLU memiliki performa yang baik dalam aplikasi deteksi dibandingkan amachandran dkk., 2017).





Gambar 2.8 Fungsi aktivasi ReLU dan SiLU

Pada nilai magnitudo yang besar, penggunaan *Sigmoid-weight Linear Unit* (SiLU) dan *Rectifier Liner Unit* (ReLU) menghasilkan aktivasi yang serupa atau mendekati kesamaan (Elfwing dkk., 2018). Yang menjadi perbedaan adalah, SiLU tidak menunjukkan sifat monoton yang selalu naik, karena SiLU memiliki nilai minimum global, terlepas dari nilai *input*-nya ada batas minimum yang dapat dicapai oleh aktivasi SiLU. Selain itu SiLU juga memiliki fitur *Self-Stabilizing*, yang berarti fungsi ini cenderung menstabilkan diri sendiri, sehingga meskipun nilai *input*-nya bervariasi, aktivasi dapat tetap dalam kisaran tertentu

2.3.5. Loss Function

Loss function merupakan metrik yang digunakan untuk mengukur sejauh mana perbedaan antara prediksi model dan label *ground truth*. Pada YOLOv5, *loss function* yang digunakan ada 3, yaitu *objectness loss* (l_{obj}), *classification loss* (l_{cls}), dan *location loss function* ($l_{l_{iou}}$). Persamaan *loss function* adalah sebagai berikut (C. Wang dkk., 2021):

$$Loss = (l_{obj}) + (l_{cls}) + (l_{l_{iou}}) \quad (2.8)$$

$$l_{obj} = \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{obj} [\hat{C}_i \log(C_i) + (1 - \hat{C}_i) \log(1 - C_i)] - \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{noobj} [\hat{C}_i \log(C_i) + (1 - \hat{C}_i) \log(1 - C_i)] \quad (2.9)$$

$$l_{cls} = - \sum_{i=0}^{S^2} I_{i,j}^{obj} \sum_{j=0}^B [\hat{P}_i(c) \log(P_i(c)) + (1 - \hat{P}_i(c)) \log(1 - P_i(c))] \quad (2.10)$$



$$l_{CIoU} = \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{obj} \left[1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \right] \quad (2.11)$$

$$v = \frac{4}{\pi^2} \left(\frac{\arctan w^{gt}}{w^{gt}} - \frac{\arctan w}{h} \right)^2 \quad (2.12)$$

$$\alpha = \frac{v}{(1 - IoU) - v} \quad (2.13)$$

Di mana, S merupakan nilai *grid*, B merupakan jumlah kotak dalam setiap *grid*, λ_{noobj} merepresentasikan bobot. $I_{i,j}^{obj}$ dan $I_{i,j}^{noobj}$ digunakan untuk menentukan apakah *anchor box* ke- j pada *grid* ke- i mengandung objek atau tidak. Jika itu mengandung objek, maka $I_{i,j}^{obj}$ adalah 1, dan $I_{i,j}^{noobj}$ adalah 0. jika sebaliknya, maka $I_{i,j}^{obj}$ itu 0 dan $I_{i,j}^{noobj}$ itu 1, hanya *confidence loss* yang dihitung. $\hat{C}_i$ menunjukkan apakah *anchor frame* bertanggung jawab atas prediksi seluruh jaringan. C_i mengindikasikan tingkat *confidence* prediksi dan kotak penanda. $P_i(c)$ menunjukkan kemungkinan *box* sebelumnya telah memiliki wilayah target. Variabel c jarak diagonal dari daerah penutupan minimum yang berisi kotak prediksi dan kotak aktual. IoU berarti rasio pembagian dan penggabungan dari *bounding box* prediksi dan *bounding box* aktual. $\rho(\cdot)$ adalah jarak euclidean. Variabel b , w dan h menunjukkan koordinat tengah, lebar dan tinggi dari kotak prediksi. Sedangkan variabel b^{gt} , w^{gt} dan h^{gt} merupakan koordinat tengah, lebar dan tinggi dari kotak aktual. α menunjukkan faktor bobot, dan variabel v sama seperti rasio panjang ke lebar.

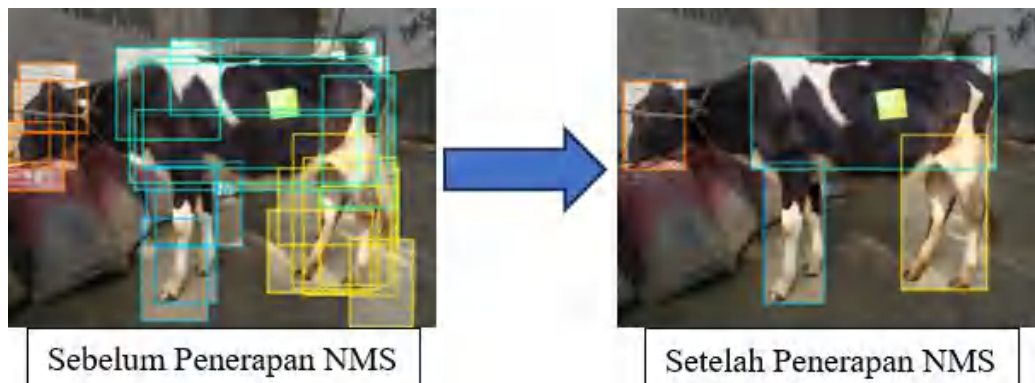
2.4. Transfer Learning

Peningkatan kinerja model lebih lanjut dapat dicapai dengan menggunakan metode yang dikenal sebagai *transfer learning*. konsep dasar di balik *transfer learning* adalah penggunaan satu set filter yang telah mempelajari dataset yang jauh lebih besar, dengan lebih banyak fitur daripada membangun model dari awal. Bobot nilai ini kemudian dapat diterapkan pada jaringan baru dan model dapat dilatih dengan dataset menggunakan filter yang telah dipelajari sebelumnya. Misalnya, menggunakan model yang telah dilatih sebelumnya menggunakan seribu kelas pada dataset yang hanya berisi dua kelas seperti kucing atau anjing. Namun, pada tahap beberapa lapisan jaringan yang telah ada sebelumnya harus dihapus dan aru harus ditambahkan agar memungkinkan model melakukan klasifikasi kelas tersebut (Moroney, 2020).



2.5. Non-maximum Suppression (NMS)

Non-maximum Suppression (NMS) telah menjadi bagian penting dari bidang visi komputer selama beberapa dekade. NMS banyak digunakan dalam deteksi tepi, deteksi titik fitur dan deteksi objek. NMS merupakan kelas algoritma yang digunakan untuk memilih satu entitas (misalnya, *bounding boxes*) dari banyak entitas yang saling tumpang tindih. Dengan kata lain, NMS digunakan untuk mengeliminasi *bounding box* yang tidak relevan (Redmon dkk., 2015).



Gambar 2.9 Contoh penerapan NMS

Sumber: Roboflow

Cara kerja dari NMS ini dimulai dengan sebuah daftar kotak deteksi $\mathcal{B}$ dengan skor $\mathcal{S}$. Setelah memilih deteksi dengan skor maksimum $\mathcal{M}$, deteksi tersebut dihapus dari set $\mathcal{B}$ dan ditambahkan ke set deteksi akhir $\mathcal{D}$. Selain itu, kotak apa pun yang memiliki tumpang tindih dengan $\mathcal{M}$ lebih dari sebuah ambang batas $\mathcal{N}_t$ di set $\mathcal{B}$ juga dihapus. Proses ini diulang untuk kotak-kotak yang tersisa di $\mathcal{B}$ (Bodla dkk., 2017).

2.6. Evaluasi Kinerja Model

Untuk mengevaluasi kinerja model YOLOv5 setelah dilakukan pelatihan data, *mean Average Precision* (mAP), *precision*, *recall* akan digunakan dan *intersection over union* (IoU) dari citra yang diprediksi dan citra aktual yang telah ditandai digunakan untuk menilai apakah target telah berhasil diprediksi. Variabel *true positive* (TP), *false positive* (FP) dan *false negative* (FN) menunjukkan nilai *True Positive*, *False Positive* dan *False Negative*. Indikator evaluasi model sebagai

Qiao dkk., 2023):



2.6.1. Precision

Precision merupakan indikator evaluasi yang menunjukkan seberapa banyak jumlah prediksi yang benar dibandingkan dengan total jumlah sampel yang diprediksi sebagai positif dalam hasil prediksi.

$$Precision = \frac{tp}{tp+fp} \times 100\% \quad (2.14)$$

2.6.2. Recall

Recall merupakan indikator evaluasi yang menunjukkan seberapa banyak jumlah prediksi yang benar dibandingkan dengan total jumlah sampel positif yang sebenarnya dalam dataset uji.

$$Recall = \frac{tp}{tp+fn} \times 100\% \quad (2.15)$$

2.6.3. Intersection Over Union (IoU)

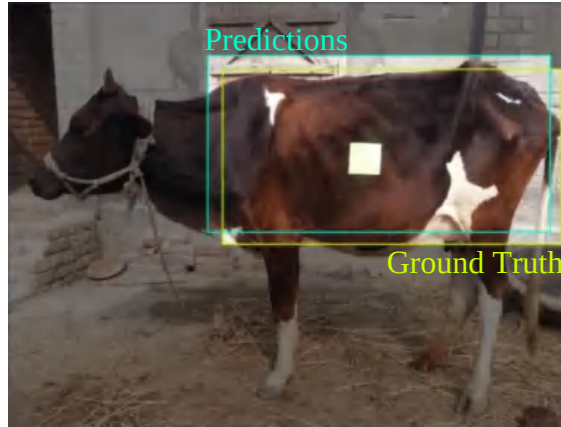
Intersection Over Union merupakan metrik yang sering digunakan untuk membandingkan antar dua bentuk sembarang. IoU memetakan properti geometris objek yang dibandingkan ke dalam properti area, seperti lebar, tinggi, dan posisi dua *bounding box*, kemudian menghitung *normalized size* yang berfokus pada area (atau volume). Properti ini memastikan IoU tidak terpengaruh oleh skala masalah yang sedang dipertimbangkan (Rezatofighi dkk., 2019).

Intersection over union, sebagai metrik evaluasi yang sangat umum digunakan dalam mendeteksi objek, berperan sebagai standar untuk menilai prediksi yang benar dan yang salah. Dalam menggunakan IoU sebagai metrik evaluasi, penting untuk memilih ambang batas akurasi yang sesuai. *Threshold* atau ambang batas yang digunakan adalah 0.5, sehingga jika hasil $\text{IoU} \geq 0.5$, maka akan menghasilkan *true positive* atau target berhasil di prediksi, namun jika $\text{IoU} < 0.5$, maka akan menghasilkan *false positive* atau target gagal diprediksi. Selain menggunakan 0.5 sebagai ambang batas IoU, juga menggunakan rentang ambang batas antara 0.5 hingga 0.9.

Persamaan *Intersection over Union* (IoU) untuk membandingkan antar dua bentuk sembarang $A, B \subseteq \mathbb{S} \in \mathbb{R}^n$ yang ditunjukkan pada Persamaan 2.16:



$$IoU = \frac{\text{area of intersection}}{\text{area of union}} = \frac{|A \cap B|}{|A \cup B|} \quad (2.16)$$



Gambar 2.10 Contoh *intersection over union*

Sumber: Roboflow

Langkah-langkahnya dimulai dengan menemukan bentuk A dan B yang sudah mencakup seluruh objek dari C, di mana $C \subseteq S \in \mathbb{R}^n$ (Rezatofighi dkk., 2019). Kemudian mengalkulasinya menggunakan Persamaan 2.10.

Untuk menghitung *area of Intersection* dan *area of union* pada Persamaan 2.10, terlebih dahulu memastikan titik-titik koordinat dari *bounding box* prediksi dalam hal ini dinotasikan B^p dan *bounding box ground truth* dinotasikan B^g , di mana koordinat dari $B^p = (x_1^p, y_1^p, x_2^p, y_2^p)$ dan koordinat dari $B^g = (x_1^g, y_1^g, x_2^g, y_2^g)$. Kemudian pada kotak prediksi B^p , terlebih dahulu menentukan bahwa $x_2^p > x_1^p$ dan $y_2^p > y_1^p$ sehingga

$$\begin{aligned}\hat{x}_1^p &= \min(x_1^p, x_2^p), & \hat{x}_2^p &= \max(x_1^p, x_2^p), \\ \hat{y}_1^p &= \min(y_1^p, y_2^p), & \hat{y}_2^p &= \max(y_1^p, y_2^p)\end{aligned}$$

Setelah menentukan titik-titik koordinat dari *bounding box*, selanjutnya Persamaan 2.17 dan Persamaan 2.18 untuk melakukan kalkulasi area:

$$B^g: A^g = (x_2^g - x_1^g) \times (y_2^g - y_1^g) \quad (2.17)$$

$$B^p: A^p = (\hat{x}_2^p - \hat{x}_1^p) \times (\hat{y}_2^p - \hat{y}_1^p) \quad (2.18)$$

Berdasarkan alur di atas, maka *area of intersection* dan *area of union* dapat dihitung menggunakan persamaan 2.19 dan Persamaan 2.20:

$$x_1^i = \max(\hat{x}_1^p, x_1^g), \quad x_2^i = \min(\hat{x}_2^p, x_2^g),$$

$$y_1^i = \max(\hat{y}_1^p, y_1^g), \quad y_2^i = \min(\hat{y}_2^p, y_2^g)$$

$$Intersection = \begin{cases} (x_2^i - x_1^i) \times (y_2^i - y_1^i) & \text{if } x_2^i > x_1^i, y_2^i > y_1^i \\ 0 & \text{otherwise} \end{cases} \quad (2.19)$$

$$Union = A^p + A^g - \text{area of intersection} \quad (2.20)$$



2.6.4. Mean Average Precision (mAP)

Mean Average Precision merupakan indikator evaluasi yang umum digunakan dalam tugas deteksi objek, termasuk model seperti YOLO. mAP memberikan gambaran secara menyeluruh tentang kinerja model dalam mendeteksi dan mengklasifikasikan objek ke berbagai kelas pada ambang batas *Intersection over union* (IoU) tertentu untuk membedakan antara *true positives* dan *false positive* (Rezatofighi dkk., 2019). mAP merupakan nilai rata-rata dari *average precision* (AP), yang ketika nilainya semakin tinggi, maka semakin baik pula kinerja model tersebut dalam mendeteksi objek (Qiao dkk., 2023). Ini merujuk pada rata-rata presisi pada ambang batas IoU yang berbeda, dalam hal ini diberikan denotasi mAP50 berdasarkan IoU *threshold* 0.5 dan mAP50-90 berdasarkan *threshold* $IoU = \{.55, \dots, .95\}$ (Rezatofighi dkk., 2019). Persamaan 2.21 akan menghitung AP dan Persamaan 2.22 akan menghitung mAP.

$$AP = \int_0^1 P(r)dr \quad (2.21)$$

$$mAP = \frac{1}{n} \sum_{i=1}^n AP(i) \quad (2.22)$$

Nilai AP berkisar dari 0 hingga 1, di mana nilai yang lebih tinggi menunjukkan kinerja yang lebih baik. $P(r)$ mewakili presisi pada *recall* tertentu (r).

2.7. Evaluasi Prediksi Berat Sapi

Dalam mengevaluasi akurasi prediksi berat badan sapi, penting untuk mempertimbangkan kemungkinan keberadaan *outlier* dalam data. Hawkins (1980) menyimpulkan bahwa *outlier* merupakan observasi yang sangat berbeda dari observasi lainnya, sehingga menimbulkan dugaan bahwa observasi tersebut mungkin berasal dari proses yang berbeda. Nilai yang ada di *outlier* bisa jauh lebih tinggi maupun lebih rendah dibandingkan dengan nilai-nilai lain dalam dataset. Hal ini bisa terjadi karena berbagai alasan, termasuk kesalahan pengukuran, kejadian langka atau karena faktor lain yang tidak terduga (Aggarwal, 2017).

Ada beberapa teknik yang sering digunakan untuk mendeteksi *outlier* dalam analisis data. Yang paling umum digunakan seperti *Z-score*, *Interquartile Range* (IQR), *box plot* atau *histogram*. *Z-score* untuk mendeteksi *outlier* berdasarkan deviasi dari rata-rata. Data dianggap *outlier* jika *Z-score* berada di luar batas biasanya di atas 3 atau di bawah -3. Untuk menghitung standar deviasi



terlebih dahulu menghitung rata-rata dari data, kemudian standar deviasi dapat dihitung dengan Persamaan 2.23:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{mean})^2}{n}} \quad (2.23)$$

Di mana x_i merupakan data ke- i dan n merupakan jumlah data. Setelah itu *Z-score* dapat dihitung menggunakan Persamaan 2.24 :

$$Z = \frac{x - \text{mean}}{\sigma} \quad (2.24)$$

Di mana x merupakan nilai data individual dari dataset dan σ merupakan standar deviasi. IQR mengidentifikasi *outlier* dengan menghitung jarak antar kuartil. Data dianggap *outlier* jika berada di bawah $Q_1 - k(Q_3 - Q_1)$ atau di atas $Q_3 + k(Q_3 - Q_1)$ di mana k sejumlah konstanta (Obikee dkk., 2014).

Untuk mengevaluasi prediksi berat badan sapi, dua metrik yang digunakan adalah *Mean Absolute Error* (MAE) dan *Mean Absolute Percentage Error* (MAPE).

2.7.1. Mean Absolute Error (MAE)

Mean absolute error (MAE) adalah salah satu metrik yang digunakan untuk mengukur seberapa akurat suatu model prediktif dengan cara menghitung rata-rata nilai absolut dari kesalahan (*error*) antara prediksi yang dihasilkan oleh model dan nilai aktualnya (Suryanto dkk., 2019). MAE didefinisikan dengan rumus yang ditunjukkan oleh Persamaan 2.25:

$$MAE = \frac{1}{n} \sum_{i=1}^n |A_i - P_i| \quad (2.25)$$

Keterangan :

- n untuk jumlah data
- A untuk data aktual
- P untuk data prediksi
- $|A_i - P_i|$ adalah nilai absolut dari kesalahan untuk data ke- i

2.7.2. Mean Absolute Percentage Error (MAPE)

Mean absolute percentage error (MAPE) merupakan metode untuk mengukur rata-rata persentase besarnya kesalahan yang dihasilkan oleh model, atau jarak prediksi-prediksi tersebut biasanya keliru. MAPE sering digunakan untuk praktik karena interpretasinya yang sangat intuitif dalam hal kesalahan relatif



(de Myttenaere dkk., 2016). Persamaan 2.26 untuk menghitung *mean absolute percentage error* (MAPE) :

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - P_i}{A_i} \right| \quad (2.26)$$

Keterangan :

- n untuk jumlah data
- A untuk data aktual
- P untuk data prediksi
- $\left| \frac{A_i - P_i}{A_i} \right|$ adalah nilai absolut dari kesalahan untuk data ke- i

2.8. Rancang Bangun Aplikasi *Mobile* (Android)

2.8.1. Android

Android merupakan sistem operasi *mobile* yang dirancang khusus untuk perangkat *mobile* seperti *smartphone* dan tablet. Android dikembangkan oleh *Open Handset Alliance* (OHA), konsorsium perusahaan-perusahaan teknologi yang dipimpin oleh Google. Android sendiri bersifat *open source*, yang memungkinkan pengembang untuk membangun, mengakses, dan memodifikasi aplikasi *mobile*. Android pertama kali diluncurkan pada tahun 2008, dan telah mengalami evolusi yang signifikan, dengan peningkatan keamanan, performa, dan fitur di setiap versi baru.

2.8.2. *TensorFlow Lite*

TensorFlow merupakan *open-source library* dari *machine learning* yang dibuat oleh tim pengembang Google. *TensorFlow* sering digunakan untuk melakukan berbagai pembelajaran mesin dan penelitian jaringan saraf tiruan. *Tensorflow* sering membutuhkan GPU dan ukuran penyimpanan yang besar saat membuat model, sehingga sebagian besar perangkat *mobile* tidak dapat menggunakan *library* ini. Untuk mengatasi masalah tersebut, model dari *TensorFlow* perlu dikonversi ke format yang mendukung penggunaan pada perangkat Mobile (Moroney, 2020).

Tensorflow lite adalah fitur yang dirancang khusus untuk perangkat *mobile*.

nengalokasi penggunaan memori secara khusus untuk mengurangi beban ncy pada perangkat *mobile* dengan dampak yang minimum pada



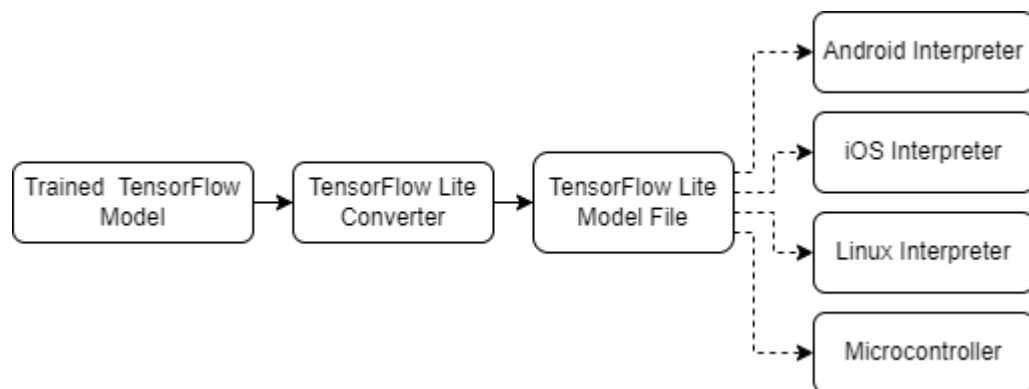
akurasi. Fitur ini juga memperkenalkan format *file* baru yang didukung *flat buffer*, yaitu *tflite*.

TensorFlow Lite memiliki beberapa keuntungan, seperti berikut:

- Proses konversi model yang mudah dan cepat, khususnya penggunaan model untuk perangkat *mobile*.
- *Toolkit machine learning* yang lebih sederhana untuk perangkat iOS dan Android.
- Penggunaan model *machine learning* dapat digunakan langsung pada perangkat *mobile*, tanpa perlu terhubung ke server secara terus-menerus.
- Memungkinkan melakukan inferensi secara *offline* pada perangkat *mobile*.

Selain beberapa keuntungan yang telah disebutkan, *TensorFlow Lite* juga memiliki beberapa kekurangan lainnya, sebagai berikut:

- Beberapa operasi atau fungsi dari *TensorFlow* secara “penuh” mungkin tidak dapat dijalankan pada *TensorFlow Lite*.
- Dapat mengurangi presisi, sehingga berpengaruh pada hasil inferensi.
- Beberapa fitur mungkin tidak optimal saat diimplementasikan pada semua perangkat *mobile*.
- Setelah model dikonversi ke *TensorFlow Lite*, mungkin ukurannya bisa menjadi lebih besar atau lebih kecil, namun beberapa tingkat kompresi ini dapat memengaruhi performa dan akurasi model.



Gambar 2.11 Ilustrasi *TensorFlow Lite*



Secara umum *TensorFlow Lite* akan mengambil model *TensorFlow* dan h model tersebut ke dalam format *tflite*. Setelah model dikonversi, model mengalami perubahan ukuran dan serangkaian *interpreter* untuk berbagai