

## **BABI**

### **PENDAHULUAN**

#### **A. Latar Belakang**

Ternak Sapi merupakan usaha yang banyak di geluti oleh sebagian orang terutama di Indonesia. Sapi memiliki potensi ekonomi yang sangat tinggi baik sebagai temak potong, ternak bibit maupun bahan pangan, dan juga salah satu hewan kurban dalam islam.

Baru-baru ini munculnya penyakit yang menyerang hewan ternak yang disebut Penyakit Mulut dan Kuku (PMK) atau *Foot and Mouth Disease* (FMD) merupakan penyakit hewan menular akut yang menyerang ternak sapi, kerbau, kambing, domba, kuda dan babi dengan tingkat penularan mencapai 90-100% yang disebabkan oleh virus dengan *genus Aphovirus* dari famili *Picornaviridae*. Hewan yang terinfeksi virus ini akan mengalami demam dan beberapa tanda gejala lainnya. Yaitu di dalam dan sekitar mulut, lidah, bibir, dan celah kuku pada hewan yang bersangkutan akan mengalami luka lepuh yang berisi cairan dan pada hewan betina gejala yang sama akan terjadi pada ambing dan puting susu (Pusat Data dan Analisa Tempo, 2020). Selain itu, juga akan muncul *hipersalivasi*, saliva terlihat menggantung, air liur berbusa di lantai kandang (Direktorat Kesehatan Hewan, 2022).

Wabah PMK kembali menyerang ribuan sapi di Indonesia dan ditemukan pertama kali di Kabupaten Gresik pada 28 April 2022 sebanyak 402 ekor sapi potong (Sutawi, 2022). Sehingga penyebarannya yang semakin terus meluas. Sebaran kasus Penyakit Mulut dan Kuku juga terjadi di Provinsi Sulawesi Selatan. Diketahui, sudah ada lebih dari 1.500 hewan ternak yang telah terinfeksi PMK. Secara rinci, Dinas Peternakan dan Kesehatan Hewan Provinsi Sulawesi Selatan mencatat ada 1.589 hewan ternak yang telah terkonfirmasi positif PMK. Jumlah kasus itu tersebar di 14 kabupaten/kota.

Salah satunya di Kabupaten Bantaeng, dilansir dari laman Siaga PMK, hingga 15 Maret 2023 kasus PMK di Kabupaten Bantaeng tercatat data sebanyak 339 ekor sakit, 254 sembuh, (Siaga PMK, 2023). Kasus PMK pada sapi ditemukan 139 ekor di Kabupaten Bantaeng dan 82 ekor sapi potong bersyarat selama tahun 2022.

Penularan penyakit mulut dan kuku juga terjadi di Sentra Ternak Bantaeng. Sebagai satu-satunya peternakan sapi terbesar yang menyediakan sapi potong dan sapi kurban di Kabupaten Bantaeng upaya mengantisipasi merebaknya penyebaran virus penyakit mulut dan kuku terus dilakukan.

Berdasarkan latar belakang diatas perlu adanya tindakan cepat dalam mendeteksi PMK pada sapi untuk mempermudah para peternak melakukan pertolongan pertama pada sapi yang terinfeksi, ehingga menghambat penyebaran virus, mengingat penularannya yang sangat mudah menyerang. Oleh karena itu, pengembangan sistem yang dapat mendeteksi dan mengenali PMK terutama gejala timbulnya PMK sapi dapat membantu peternak. Sehingga peneliti melakukan penelitian "Sistem Pakar Pendeteksian Penyakit Mulut dan Kuku Pada Sapi Berbasis Citra Di Sentral Ternak Bantaeng" dalam sistem pakar tersebut menggunakan pendeteksi citra objek ataupun *object detection* digunakan *dataset* berbentuk gambar/foto selaku input.

## **B. Rumusan Masalah**

Berdasarkan latar belakang di atas maka yang menjadi pokok permasalahan dalam penelitian ini sebagai berikut:

1. Bagaimana pembuatan sistem pakar deteksi PMK pada Sapi Berbasis Citra?

## **C. Tujuan Penelitian**

Tujuan dari penelitian ini yaitu:

1. Untuk mengetahui pembuatan sistem pakar deteksi PMK pada Sapi Berbasis Citra!

## **D. Manfaat Penelitian**

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi Akademisi

Penelitian ini diharapkan dapat memberikan kontribusi dan referensi mengenai proses pembuatan sistem pakar deteksi PMK berbasis citra yang dihasilkan saat pengujian sistem deteksi PMK, sehingga dapat dikembangkan untuk penelitian pembuatan sistem deteksi penyakit hewan lainnya.

2. Bagi Peternak Sapi

Penelitian ini sebagai sarana untuk menambah wawasan dan acuan bagi peternak dalam pendeteksian PMK sapi.

3. Bagi Pemerintah dan Instansi

Terkait Hasil dari penelitian ini diharapkan dapat dijadikan sebagai salah satu bagian informasi dan sumbangan pemikiran terhadap arah kebijakan yang ditempuh oleh pemerintah daerah, khususnya terkait PMK sapi di Kabupaten Bantaeng.

## **E. Ruang Lingkup Penelitian**

Batasan masalah dari penelitian ini adalah :

1. *Dataset* yang digunakan merupakan gambar berupa Iuka sapi yang terdeteksi PMK dan sapi terdeteksi sehat atau tidak terdeteksi PMK.
2. *Dataset* didapatkan dari Pusat Kesehatan Hewan Kabupaten Bantaeng dan Sentra Ternak Bantaeng,
3. *Klasifikasi* citra mencakup empat kelas, yaitu Mulut dan Kuku sapi untuk Gejala PMK dan Mulut dan Kuku untuk yang Sehat.
4. *Klasifikasi* citra menggunakan algoritma Yolov5

## **F. Sistematika Penulisan**

Uraian pembahasan secara sistematika per bab, dimana setiap bab terdiri atas sub bagian seperti berikut :

**BAB I** Bab ini menjelaskan tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup penelitian yang dilakukan serta sistematika penulisan dari hasil laporan penelitian.

**BAB II**, Bab ini menjelaskan tentang landasan teori, penelitian terkait dan kerangka pikir

**BAB III** Bab ini menjelaskan tentang waktu dan tempat penelitian, alat dan bahan, perancangan sistem, Teknik pengujian sistem dan Teknik analisis data

**BAB IV**, Bab ini menjelaskan hasil dan pembahasan yang diteliti dari muJai pengumpulan dataset, pembagian dataset, pelabelan dataset, training dataset dan hasil pendeteksian.

**BAB V**, Bab ini menjelaskan kesimpulan dan saran hasil yang didapatkan

## BAB II

### TINJAUAN PUSTAKA

#### A. Landasan Teori

##### 1. Penyakit Mulut dan Kuku

Indonesia sudah beberapa kali mengalami wabah PMK sejak pertama kali masuk pada tahun 1887 melalui impor sapi dari Belanda (Direktorat Kesehatan Hewan, 2022). Direktorat Kesehatan Hewan juga menyebutkan bahwa wabah PMK terakhir terjadi di pulau Jawa di tahun 1983 dan kemudian dapat dimusnahkan melalui program vaksinasi massal. Sehingga pada tahun 1986 Indonesia dinyatakan sebagai negara bebas PMK melalui Surat Keputusan Menteri Pertanian Nomor 260 Tahun 1986 dan pada tahun 1990 diakui OIE dengan Resolusi nomor XI. Namun, Wabah PMK kembali menyerang ribuan sapi di Indonesia dan ditemukan pertama kali di Kabupaten Gresik pada 28 April 2022 sebanyak 402 ekor sapi potong. Kasus kedua ditemukan 102 ekor sapi di Kabupaten Lamongan dan 595 ekor sapi perah dan kerbau di Kabupaten Sidoarjo terinfeksi PMK pada 1 Mei 2022 (Sutawi, 2022).



Gambar 1 Grafik kasus penyakit mulut dan kuku di Indonesia (Direktorat Kesehatan Hewan, 2022)

Berdasarkan data pada grafik penyebaran virus PMK terus mengalami peningkatan dan terjadi dalam waktu yang cukup singkat. Ternak yang terserang penyakit mulut dan kuku diantaranya adalah sapi, kerbau, kambing, domba dan babi. Tercatat hingga akhir bulan juni 2022 terdapat 19 provinsi dan 221 kabupaten/kota tertular penyakit mulut dan kuku dengan jumlah kasus 291.538 ekor sakit, 96.060 ekor sembuh, 2.944 ekor potong bersyarat dan 1.733 ekor mati. Kasus PMK tertinggi ditunjukkan oleh Kabupaten Probolinggo dengan jumlah kasus sebanyak 11.433 ekor ternak. Penyebaran kasus yang sangat cepat dapat disebabkan oleh mobilitas ternak, produk maupun manusia yang tinggi. Menanggapi hal tersebut pihak pemerintah setempat sudah berupaya melakukan pencegahan penyebaran penyakit PMK melalui vaksinasi di daerah-daerah endemik dan meminimalisir mobilitas ternak melalui penutupan pasar hewan (Zainuddin et al, 2022)



Gambar 2 Sapi Terjangkit PMK yang didapatkan di Pusat Kesehatan Hewan Kabupaten Bantaeng

PMK biasanya sangat menular dan menyerang hewan-hewan peka atau berkuku belah seperti sapi, babi, kambing, domba, kerbau, dan beberapa hewan liar seperti rusa, babi hutan, antelop. Meskipun PMK tidak menyebabkan angka kematian yang tinggi pada hewan dewasa, penyakit ini dapat menimbulkan gejala klinis yang bervariasi tergantung serotipe virus PMK yang menyerang. Gejala klinis yang akan segera muncul sejak pertama kali terinfeksi, yaitu kenaikan suhu

tubuh dan diikuti lemas, nafsu makan menurun, terdapat luka lepuh pada tubuh, *salivasi* akan meningkat dan disertai busa di sekitar bibir serta saliva yang menggantung di area mulut (Grubman & Baxt, 2004).

## 2. Sistem Pakar

Sistem pakar merupakan salah satu cabang dari kecerdasan buatan (*artificial intelligence*) yang membuat penggunaan secara luas *knowledge* (pengetahuan) yang khusus untuk menyelesaikan masalah tingkat manusia yang pakar. Menurut Feigenbaum dalam Arhani (2005:2) mendefinisikan sistem pakar sebagai "suatu program komputer cerdas yang menggunakan *knowledge* (pengetahuan) dan prosedur *inferensi* untuk menyelesaikan masalah yang cukup sulit sehingga membutuhkan seorang yang ahli untuk menyelesaikannya." Menurut Martin dan Oxman dalam Kusri (2006:11) menyatakan bahwa "sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam bidang tersebut." Ciri-ciri sistem pakar menurut Kusri (2006:14- 15) adalah:

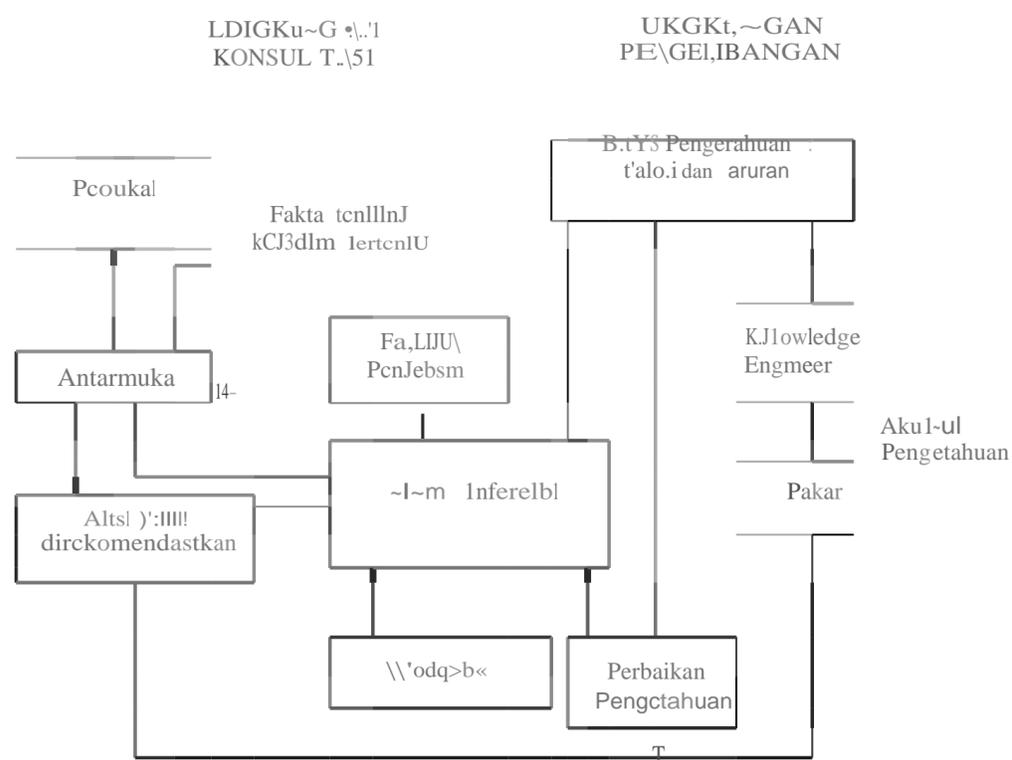
- a. Terbatas pada bidang yang spesifik.
- b. Dapat memberikan penalaran untuk data data yang tidak lengkap atau tidak pasti.
- c. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
- d. Berdasarkan pada rule atau kaidah tertentu.
- e. Dirancang untuk dapat dikembangkan secara bertahap.
- f. Outputnya bersifat nasihat atau anjuran.
- g. Output tergantung dari dialog dengan user.
- h. *Knowledge basedan inferensi engine* terpisah.

Turban dalam Arhani (2005:11) menyatakan bahwa "konsep dasar dari suatu sistem pakar mengandung beberapa unsur/elemen, yaitu keahlian, ahli, pengalihan keahlian, inferensi, aturan dan kemampuan menjelaskan." Terdapat

tiga orang yang terlibat dalam lingkungan sistem pakar menurut Turban dalam Arhami (2005:12-13), yaitu:

- a. Pakar, adalah orang yang memiliki pengetahuan, khusus pendapat pengalaman dan metode, serta kemampuan untuk mengaplikasikan keahliannya tersebut guna menyelesaikan masalah.
- b. *Knowledge engineer* (perekayasa sistem), adalah orang yang membantu pakar dalam menyusun area permasalahan dengan menginterpretasikan dan mengintegrasikan jawaban-jawaban pakar atas pertanyaan yang diajukan, menggambarkan analogi, mengajukan *counterexample* dan menerangkan kesulitan-kesulitan konseptual.
- c. *User* (Pemakai), sistem pakar memiliki beberapa pemakai, yaitu: pemakai bukan pakar, pelajar, pembangun sistem pakar yang ingin meningkatkan dan menambahkan basis pengetahuan, dan pakar.

Menurut Turban dalam Arhami (2005:13) menyatakan bahwa " Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*)



Gambar 3 Arsitektur Sistem Pakar Sumber : Sri Hartati dan Sari Iswati (2008). Sistem Pakar dan Pengembangannya

Pada gambar di atas menjelaskan bahwa, sistem pakar disusun menjadi dua bagian utama, yaitu: pertama, lingkungan pengembangan (*development environment*), kedua lingkungan konsultasi. Di bawah ini akan dijelaskan secara ringkas komponen-komponen yang membentuk sistem pakar.

- a. Antarmuka pengguna merupakan mekanisme yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem. Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.
- b. Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan menyelesaikan masalah. Komponen ini disusun atas dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang obyek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui.
- c. Akuisisi pengetahuan adalah akumulasi, transfer dan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Dalam tahap ini *knowledge engineer* berusaha menyerap pengetahuan untuk selanjutnya ditransfer ke dalam basis pengetahuan. Pengetahuan berasal dari pakar, dilengkapi dengan buku, basis data, laporan penelitian dan pengalaman pemakai.
- d. Mesin Inferensi (*Inference Engine*) Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin Inferensi menurut Turban dalam Arhami (2005: 19) adalah "program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam *workplace*, dan untuk memformulasikan kesimpulan".

### 3. Basis Citra Digital

citra adalah gambar pada bidang dua dimensi. Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus dari intensitas cahaya pada bidang dwimarta. Sumber cahaya yang menerangi objek, kemudian objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya tersebut ditangkap oleh alat-alat optik, misalnya pada mata manusia, kamera, dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam.

pengolahan citra adalah istilah umum untuk berbagai teknik yang berfungsi untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Foto adalah salah satu contoh gambar dua dimensi yang dapat diolah dengan mudah. Setiap foto dalam bentuk citra digital dapat diolah menggunakan perangkat lunak tertentu. Pengolahan citra merupakan bagian penting yang mendasari berbagai aplikasi nyata, seperti pengenalan pola, penginderaan jarak jauh melalui satelit, dan *machine vision*

### 4. Object Detection

*Computer vision* merupakan salah satu cabang ilmu yang bertujuan untuk membuat keputusan untuk mengenali objek fisik nyata dan keadaan dari sebuah gambar atau citra (Shapiro & Stockman, 2001). Dalam artian sederhananya, *computer vision* adalah bagaimana komputer dapat melihat, memvisualisasikan, menganalisa citra digital baik dalam bentuk gambar ataupun video. Dengan kata lain, *computer vision* seperti peran mata manusia pada komputer dan juga otak manusia pada komputer. Kemampuan komputer ini dapat mendekati kemampuan manusia dalam menangkap informasi visual yaitu kemampuan:

- a. Object detection: mengenali objek dan memberikan batasan
- b. Recognition: memberikan label pada objek
- c. Interpreting motion: menafsirkan gerak
- d. Description, 3D interference, Dan kemampuan lainnya

*Object detection* merupakan suatu aspek yang terdapat pada *computer vision*. *Object detection* ini berfungsi untuk mendeteksi objek yang ada pada suatu gambar. Hal ini dapat diperlakukan sebagai pengenalan objek kelas dua, dimana satu kelas mewakili objek sedangkan satu kelas lagi mewakili non-objek. Dalam

hal penelitian ini, objek yang dideteksi adalah nematoda, sedangkan nonobjek merupakan lingkungan sekitarnya.

Deteksi objek bertujuan untuk menentukan dan mengklasifikasikan objek yang ada dalam suatu gambar dan melabelinya dengan kotak pembatas (*bounding box*) persegi panjang untuk menunjukkan keberadaannya. Kerangka metode deteksi objek dapat dikategorikan menjadi dua jenis. Satu mengikuti jalur deteksi objek tradisional dengan menghasilkan proposal wilayah di awal dan kemudian mengklasifikasikan setiap proposal ke dalam kategori objek yang berbeda.

## 5. *Dataset*

*Dataset* atau dengan kata lain kumpulan data, seperti namanya, dataset akan berisi sekumpulan data yang banyak yang akan berfungsi sebagai data uji pada penelitian ataupun pengamatan beserta atributnya. *Dataset* dapat diambil dan dikumpulkan dari mana saja sesuai kebutuhan penelitian dan sesuai dengan penelitian yang dilakukan. Dalam hal penelitian penulis kali ini, dataset yang digunakan adalah dataset baru yaitu dataset yang akan berisi kumpulan data atau objek yang akan terlihat seperti basil mikroskop untuk dilakukan pendeteksian nematoda dan membedakan nematoda parasit dengan nematoda non-parasit, sehingga dapat menghasilkan sistem yang dapat membantu usaha pertanian dan perkebunan.

*Dataset* Pada umumnya dataset yang digunakan pada kasus deteksi objek adalah dalam bentuk gambar maupun video. Dataset merupakan kumpulan data yang akan menjadi input untuk melatih model deep learning. Dataset dapat akan dibagi menjadi, yaitu train set, validation set, dan test set.

- a. *Train set*: Sampel data yang digunakan untuk menyesuaikan dengan model. Train set merupakan *Dataset* aktual yang digunakan untuk melatih model (bobot dan bias untuk jaringan saraf). Model melihat data ini dan mempelajarinya.
- b. *Validation set*: Sampel data yang digunakan untuk memberikan evaluasi yang tidak biasa tentang kecocokan model dengan set data pelatihan saat menemukan

hyperparameter model. Dataset validasi tidak digunakan untuk latih model tetapi data ini berjalan selama pelatihan untuk pengembangan model.

- c. *Test set*: Sampel data yang digunakan untuk menguji performa model yang telah dilatih dengan train set.

## 6. *Data Augmentation*

*Data augmentation* merupakan suatu proses pengolahan *dataset* yang biasanya berupa gambar dengan cara mengubah atau memodifikasi data yang sudah dikumpulkan atau yang sudah ada sedemikian rupa sehingga akan terdeteksi sebagai gambar yang berbeda oleh komputer. *Data augmentation* ini berfungsi untuk memperbanyak *dataset* atau kumpulan data yang sudah ada dengan tidak menduplikasi data yang ada secara cuma-cuma melainkan melakukan beberapa perubahan berskala yang dapat diterapkan pada semua data gambar yang ada. *Data augmentation* juga dapat memberikan hasil training yang lebih baik dengan banyaknya data untuk mencegah terjadinya *overfitting* atau terlalu terpaku dengan beberapa gambar yang sudah dipelajari. *Data augmentation* biasanya dilakukan dengan operasi transformasi dan dapat digabungkan beberapa operasi sekaligus, antara lain:

- a. *Rotasi*: melakukan rotasi dengan skala tertentu dari 0-180°
- b. *Zoom*: melakukan operasi memperbesar atau memperkecil gambar sesuai skala
- c. *Flip*: melakukan operasi membalik gambar
- d. *Shift*: melakukan penggeseran gambar yang sudah ada ke kanan atau kiri dengan skala tertentu
- e. *Shear*: melakukan transformasi geser secara acak

*Augmentasi* adalah sebuah proses memodifikasi atau memanipulasi citra, dimana citra asli dalam bentuk standar akan diubah bentuk dan posisinya. *Augmentasi* juga memiliki manfaat untuk meningkatkan keragaman data yang ada untuk proses pelatihan, tanpa benar-benar harus menambah data baru. Hal ini sangat berguna untuk mengatasi permasalahan train model yaitu data *hungry* (membutuhkan banyak data). *Augmentasi* data memiliki banyak jenis tekniknya

*seperti,flippirig, cropping, cutoff, copy-paste, padding, dll.* Oleh karena banyaknya jenis teknik: augrmentasi maka data baru akan lebih banyak didapatkan dengan mengkombinasikan teknik augmentasi yang ada. Sehingga dalam satu citra bisa memiliki banyak jenis augmentasinya yang menyebabkan proses pembelajaran mesin dapat belajar mengenali berbagai citra yang berbeda beda sehingga bisa meningkatkan performa dari suatu model (Sanjaya & Ayub, 2020)

## 7. Roboflow

*Roboflow* merupakan sebuah platform yang tersedia di *web* yang memiliki banyak fungsi yang berhubungan dengan dataset. Dengan rnenggunakan *roboflow* ini dapat membagikan dataset sekaligus memproses dataset tersebut. Beberapa firur yang digunakan pada penelitian ini diantaranya adalah melakukan annotate atau menandai objek yang akan di deteksi menggunakan *bounding box*, selain itu dapat digunakan juga *pre-processing* pada *dataset* misalnya melakukan *grayscale*, dan juga augmentasi dengan menggunakan *roboflow*. *Roboflow* ini menyediakan beberapa jenis augmentasi konvensional *seperti flip, rotate, brightness, exposure, shear* dan lain-lain.

*Roboflow* yang dikhususkan untuk membantu *AI engineer* dalam mengolab *dataset* untuk pengaplikasian projek *computer vision*. *Roboflow* menyediakan layanan yang dapat membantu pengembang *computer vision* dalam menjalankan projek *computer vision* dengan alur kerja. Hal yang pertama dilakukan adalah upload dimana dataset diunggah pada *platform roboflow* yang selanjutnya akan *diorganisir* agar data tidak sulit untuk dilakukan anotasi atau pemberian *bounding box*. Selanjutnya data tersebut dapat menjadi input untuk latih model. Model yang telah dilatih dapat dilakukan deployment dan dapat dilihat basil deteksi dari model yang sudab dilatib. *Roboflow* menyediakan 20 library model yang dapat menjadi pilihan model saat akan membuat projek *computer vision*, yaitu *YOLOX, Vision Transformer, YOLOv5, YOLOR, OpenAJ Clip, Scaled-YOLOv4, Resnet32, EfficientDet-DO-D7, YOLOv4-tiny, YOLOv4 Darknet, EfficientNet, Detectron2, Efficieniliet, Faster R-CNN, YOLO v3 PyTorc/l, YOLO v3 Keras, MobileNetSSDv2, YOLOv4, PyTorch, MobileNetV2 Classification, dan ResNet32.*

## 8. YOLO (*You Only Look Once*)

Algoritma pengenalan object yang utama saat ini adalah seri R-CNN dan seri YOLO. *Region Convolutional Neural Network* (R-CNN) menggunakan *Region Proposal Network* (RPN) dimana cara kerjanya dengan membuat proposal *bounding box* terlebih dahulu, kemudian menjalankan pengklasifikasian pada *bounding box* dan selanjutnya menghapus deteksi duplikat dan memperbaiki *bounding box*. Karena alur kerja ini yang dilakukan satu persatu sebanyak proposal yang membuat RCNN sulit dan lambat untuk di optimalkan. Sedangkan YOLO merupakan algoritma dengan pendekatan baru dalam *object detection*. *Computer vision* menggunakan jaringan syaraf tiruan (CNN).

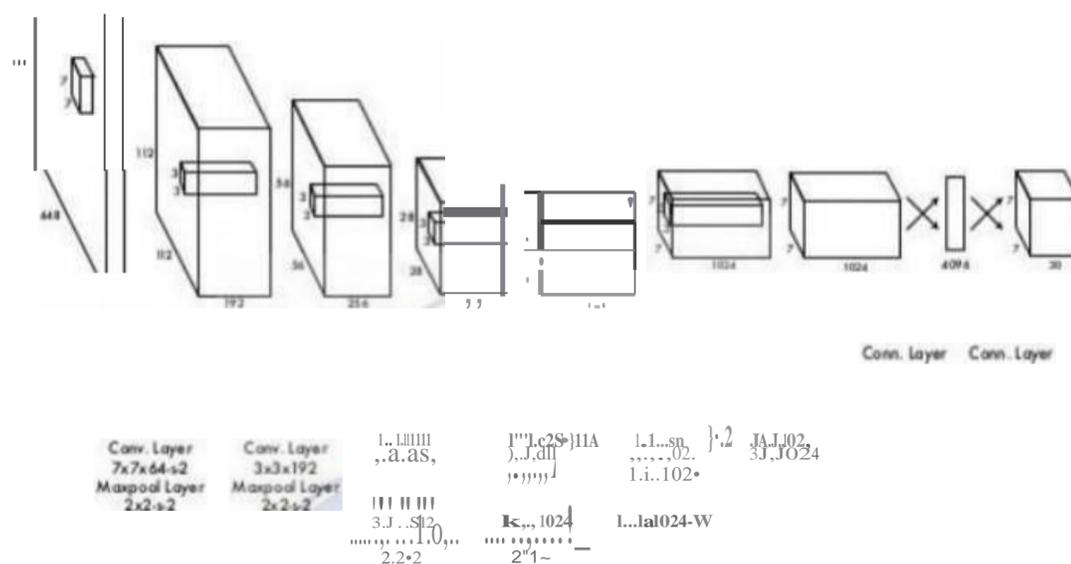
YOLOv1 diperkenalkan pertama kali melalui jurnal yang berjudul "*You Only Look Once: Unified, Real-Time Object Detection*" oleh Joseph Redmon, Santosh Divvala, Ross Girshick, dan Ali Farhadi pada tahun 2015 (Redmon et al., 2016). Ide pertama dari kreator YOLO adalah dengan mendesign sebuah algoritma yang memanfaatkan jaringan tunggal *Convolutional Neural Network* yang berguna memprediksi beberapa *bounding box* dan probabilitas kelas untuk *bounding box* tersebut, dan YOLO berlatih pada citra penuh secara langsung mengoptimalkan kinerja pendeteksian, yang dimana dengan cara ini bisa mengalahkan kecepatan algoritma pendahulunya seperti R-CNN, Fast R-CNN, Faster R-CNN.

Kerangka kerja YOLO dibagi menjadi 3 bagian utama, yaitu :

- a. *Backbone* .*Convolutional Neural Network* yang menggabungkan dan membentuk fitur gambar pada berbagai jenis detail gambar.
- b. *Neck*: Serangkaian lapisan jaringan yang mencampur dan menggabungkan fitur gambar dan meneruskannya ke lapisan prediksi.
- c. *Head* : Menggunakan fitur dari *Neck* untuk menghasilkan kotak pembatas, dan memprediksi kategori

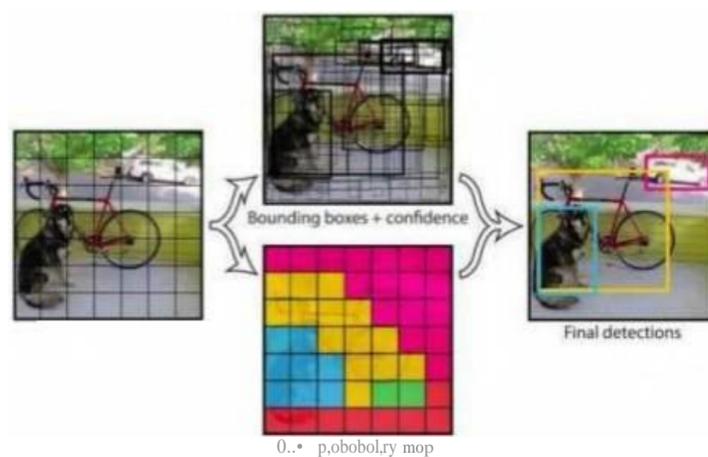
Pertama, Algoritma YOLO akan membagi citra dalam grid berukuran  $S \times S$ , apabila dalam region  $S$  terdapat objek maka region tersebut akan melakukan deteksi. Setiap *region* akan memprediksi *bounding box* dan peta kelas pada masing masing *grid* (Justitian et al., 2022).

Arsitektur YOLO tersendiri terinspirasi dari model *GoogLeNet* yang digunakan dalam tugas klasifikasi gambar (Joseph Redmon, 2016). Terdiri dari 3 jenis lapisan : *Convolutional*, *Maxpool*, dan *Fully Connected layer*



Gambar 4 Arsitektur YOLO (Aditya Yanuar, 2018)

Arsitektur ini mengambil gambar sebagai input dan mengubah ukurannya menjadi 448\*448 kemudian diteruskan ke jaringan CNN (conv layer). Model ini memiliki 24 lapisan konvolusi, 4 lapisan max-pooling dan SS2 lapisan yang terhubung penuh. Arsitektur ini menggunakan Leaky ReLU sebagai fungsi aktivasinya di seluruh arsitektur kecuali lapisan terakhir yang menggunakan fungsi aktivasi linear (Aditya Yanuar, 2018).



Gambar 5 YOLO Conceptual Design (Redmon et al., 2016).

*Loss function* / *cost function* merupakan fungsi yang digunakan untuk mengukur seberapa baik performa yang dihasilkan model dalam mendeteksi sebuah objek. Fungsi *loss* yang baik adalah fungsi yang menghasilkan error yang paling rendah. Jika model memiliki kelas lebih dari dua / multiclass maka perlu adanya cara untuk mengukur perbedaan nilai antara probabilitas hasil prediksi dan probabilitas nilai aktual. Fungsi *loss* yang digunakan dalam algoritma YOLO adalah *Mean Square Error* (MSE). *Mean Square Error* (MSE) merupakan fungsi *loss* yang paling sering umum untuk digunakan dan sering disebut juga sebagai L2 Loss, dimana cara kerjanya yaitu dengan menghitung rata-rata perbedaan kuadrat nilai aktual dan nilai prediksi. Hasil dari MSE akan selalu bernilai positif terlepas dari nilai prediksi dan nilai aktualnya. Untuk mencapai model dengan performa terbaik, peneliti harus mencoba mengurangi nilai L2 Loss sekecil mungkin atau mencapai nilai sempurna yaitu 0 (Arwinda et al., 2020)

## 9. Deskripsi YOLOv5

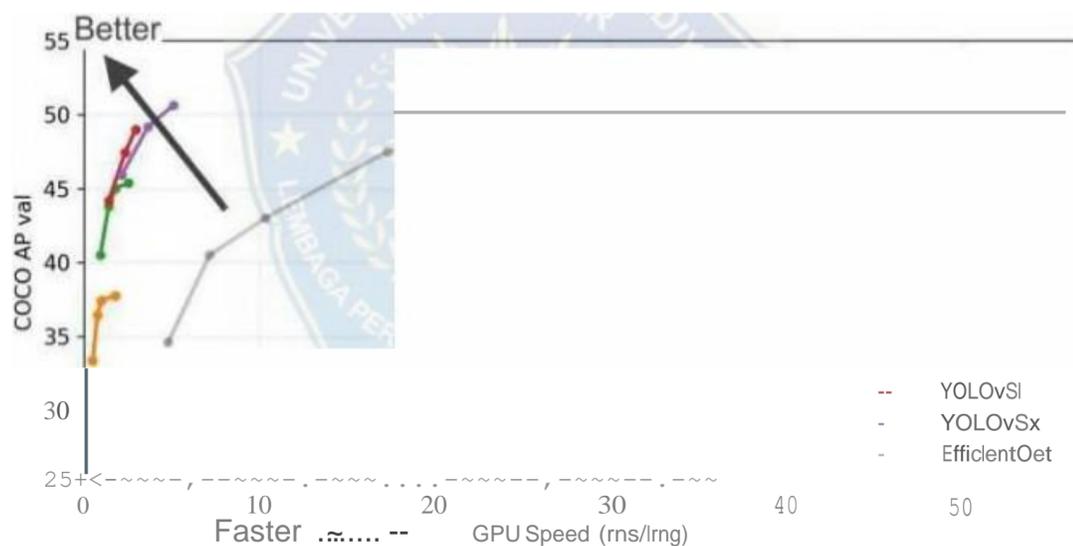
Tidak lama setelah YOLOv4 rilis pada April 2020, Glenn Jocher dan tim Ultralytics LLC melakukan publikasi versi terbaru dari Seri YOLO, versi terbaru tersebut diberi nama YOLOv5 (Jocher, 2020). Glenn Jocher merupakan peneliti, pengembang dan CEO Ultralytics LLC. Secara umum arsitektur YOLOv5 tidak jauh berbeda dengan YOLO versi sebelumnya. YOLOv5 dikembangkan menggunakan bahasa pemrograman *Python*, bukan seperti YOLO versi sebelumnya yang menggunakan bahasa pemrograman *C*.

Hal ini yang membuat instalasi dan integrasi pada perangkat IoT lebih mudah digunakan. Selain itu YOLOv5 dikembangkan menggunakan kerangka *library PyTorch* dimana memiliki komunitas yang lebih besar dari komunitas Darknet, yang menyebabkan *PyTorch* akan mendapatkan dukungan kontribusi dan potensi pertumbuhan yang lebih baik di masa depan (Thuan, 2021).

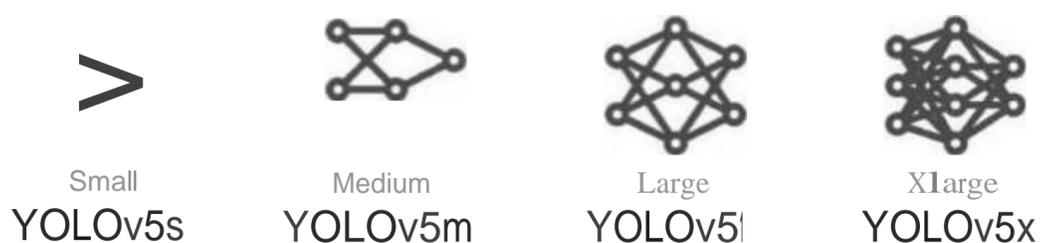
Meskipun dikembangkan dalam dua bahasa pemrograman dan kerangka kerja yang berbeda, kinerja antara YOLOv4 dan YOLOv5 tidak jauh berbeda, akan tetapi setelah beberapa saat YOLOv5 terbukti lebih berperforma baik akurasi

maupun kecepatan daripada YOLOv4 dalam beberapa kasus, dan juga sebagian komunitas computer vision lebih percaya akan performa YOLOv5 (Thuan, 2021). YOLO memiliki tiga kerangka kerja utama yaitu *backbone*, *neck* dan *head*. Bagian *backbone* YOLOv5 menggunakan CSPDarknet53 yang di adopsi dari CSPDarknet YOLOv4, bagian ini memecahkan pengulangan informasi gradien di backbone dan mengintegrasikan perubahan gradien ke dalam peta fitur sehingga dapat meningkatkan akurasi, mengurangi kecepatan inferensi, dan mengurangi ukuran bobot model dengan mengurangi jumlah parameter. Bagian neck YOLOv5

Tipe YOLOv5 YOLOv5 memiliki beberapa tipe antara lain YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, dan masing-masing tipe tersebut memiliki perbedaan dari segi kecepatan deteksi dan performa akurasi mAPnya. Pada penelitian ini Penulis akan menggunakan YOLOv5. Berikut gambar dari masing-masing tipe YOLOv5



Gambar 6 Tipe YOLOv5 (Glenn Jocher, 2022)



Gambar 7 Parameter Tipe YOLO (Glenn Jocher, 2022)

Pada gambar merupakan gambaran parameter dari tipe YOLO. Parameter merupakan variabel konfigurasi internal model yang nilainya diberikan dari data pretrained model yang digunakan. Parameter ini digunakan untuk menentukan keterampilan model dalam membuat prediksi (Glenn Jocher, 2022). Semakin tinggi tipe YOLO maka akan memiliki jaringan *neural network* yang lebih banyak sehingga parameter yang digunakan semakin besar.

#### 10. Google Colab

*Google Colaboratory* merupakan sebuah alat yang mirip dan berdasarkan dari *Jupyter Notebooks*. *Jupyter* adalah alat *open source berbasis browser* yang mengintegrasikan bahasa pemrograman, library, dan alat visualisasi. *Notebook Jupyter* berfungsi baik secara lokal maupun di cloud. Setiap dokumen terdiri dari beberapa sel, di mana setiap sel berisi bahasa skrip atau kode markdown, dan outputnya disematkan dalam dokumen. Tipe output termasuk teks, tabel, grafik, dan grafik. *Google Colaboratory* (Colab) adalah proyek yang bertujuan untuk menyebarkan pendidikan dan penelitian pembelajaran mesin. *Notebook Colab* didasarkan pada *Jupyter* dan digunakan sebagai objek *Google Documents*: 17 notebook dapat dibagikan dan pengguna dapat berkolaborasi pada notebook yang sama.

Kolaborasi menyediakan runtime Python 2 dan 3 yang telah dikonfigurasi sebelumnya dengan pembelajaran mesin dan library AI, seperti TensorFlow, Matplotlib, dan Keras. Mesin virtual di bawah runtime (VM) dinonaktifkan setelah jangka waktu tertentu, dan semua data dan konfigurasi pengguna hilang. Namun, notebook memungkinkan untuk mentransfer file dari hard disk VM ke akun *Google Drive* pengguna. Layanan *Google* ini menyediakan akselerasi GPU runtime, juga sepenuhnya dikonfigurasi dengan perangkat lunak yang sebelumnya diuraikan. *Infrastruktur Google Colaboratory di-host di platform Google Cloud*.

## 11. Confusion Matrix

*Confusion Matrix* adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada empat istilah yang merupakan representasi hasil proses klasifikasi pada *confusion matrix* yaitu *True Positif*, *True Negatif*, *False Positif*, dan *False Negatif*. *Confusion Matrix* adalah metode evaluasi yang digunakan dalam menghitung tingkat kebenaran dari proses klasifikasi.

Ada empat istilah yang merupakan representasi hasil proses *klasifikasi* pada *Confusion Matrix* yaitu *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*.

- a. TP (*True Positive*) ialah jumlah data yang kelas aktual dan prediksinya merupakan kelas positif
- b. FN (*False Negative*) ialah total data yang kelas aktualnya merupakan kelas positif sedangkan kelas prediksinya merupakan kelas negatif.
- c. FP (*False Positive*) ialah banyaknya data yang kelas aktualnya merupakan kelas negatif sedangkan kelas prediksinya merupakan kelas positif.
- d. TN (*True Negative*) ialah banyaknya data yang kelas aktualnya merupakan kelas negatif sedangkan kelas prediksinya merupakan kelas negatif

*Confusion Matrix* adalah metode evaluasi yang digunakan dalam menghitung tingkat kebenaran dari proses klasifikasi. Ada tiga metrik yang bermanfaat untuk mengukur performa dari classifier atau algoritma yang digunakan untuk melakukan prediksi.

- a. *Akurasi* merupakan metode pengujian berdasarkan tingkat kedekatan antara nilai prediksi dengan nilai aktual. Dengan mengetahui jumlah data yang *diklasifikasikan* secara benar

- b. Presisi merupakan metode pengujian dengan melakukan perbandingan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi yang terambil oleh sistem baik yang relevan maupun tidak.
- c. *Recall* merupakan metode pengujian yang membandingkan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi relevan yang ada dalam koleksi informasi (baik yang terambil atau tidak terambil oleh sistem)

## 12. Bahasa Pemrograman Python

*Python* dikembangkan oleh Guido van Rossum pada tahun 1990 di CWI, Amsterdam sebagai kelanjutan dari Bahasa pemrograman ABC. Versi terakhir yang dikeluarkan CWI adalah 1.2 Tahun 1995, Guido pindah ke CNRI sambil terus melanjutkan pengembangan *Python*. *Python* adalah Bahasa pemrograman interpretative multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai Bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. *Python* mendukung multi paradigm pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperative, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai Bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. seperti halnya pada Bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai Bahasa skrip meski pada praktiknya penggunaan Bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan menggunakan Bahasa skrip. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform system operasi. Saat ini, kode python dapat dijalankan platform *system operasi*, diantaranya : *Linux/Unix, Windows, Mac OS, Java Virtual Machine, OS/2., Amiga, Symbian.*

Banyak hal yang dapat dijelajahi ketika menggunakan bahasa pemrograman python. Beberapa *package Python* yang populer antara lain :

- a. *Django, llleframework*
- b. *Scipy* dan *Scikit*, pusaka untuk membuat aplikasi *machine learning* dan kecerdasan buatan.
- c. *Tornado*, pustaka untuk membuat aplikasi *lve, websocker; dan asynchronous programming*.
- d. *Celery*, pustaka untuk membuat asynchronous task.
- e. *Matplotlib*, pustaka untuk membuat grafik keperluan saintific.
- f. *biol'ython*, pustaka untuk menganalisa DNA dan Genome makhluk hidup.
- g. *TensorF/0lll*, pustaka untuk membuat aplikasi yang ditenagai oleh deep learning
- h. *Opencv*, pustaka untuk membuat *aplikasi computer vision*.

Pada penelitian ini, saya menggunakan *OpenCV* sebagai library pada program *computer vision* yang akan saya kerjakan. Bahasa ini tidak hanya digunakan oleh para pemula, tetapi juga para periset tingkat dunia. Dukungan pustaka yang sangat banyak menjadikan banyak persoalan dapat dipecahkan dengan mudah menggunakan bahasa ini.

### 13. Numl'y

*Nu,nJJY* merupakan *library* pada bahasa pemrograman *Python* yang lebih menekankan pada komputasi ilmiah. *Num(P)* dapat melakukan pembentukan objek N-dimensional array yang hampir sama dengan list di python. Namun *NumPy* memiliki kelebihan penggunaan memori yang lebih kecil dan durasi runtime yang lebih singkat dari pada list di pyhton. Selain itu dengan menggunakan *Numpy* operasi untuk *Aljabar Linear* lebih mudah.

#### 14. OpenCV

*OpenCV* merupakan kepanjangan dari "*Open Source Computer Vision*" yang merupakan program *open source* berbasis C++ dan saat ini banyak digunakan sebagai program computer vision. yang merupakan pustaka berbasis Open Source yang mengandung lebih dari 500 fungsi yang ditujukan untuk menangani visi computer dan dapat diunduh di <https://opencv.org/>. perangkat lunak ini dirilis dengan lisensi BSD dan dapat digunakan untuk kepentingan bisnis maupun komersial. Platform yang didukung mencakup Windows, Linux, Mac OS, iOS, dan Android. *OpenCV* digunakan terutama untuk operasi yang berkaitan dengan gambar, seperti :

- a. Membaca dan menulis gambar.
- b. Mendeteksi wajah dan fitur-fiturnya
- c. Mendeteksi bentuk seperti lingkaran, persegi, segitiga dan lainlain didalam gambar.
- d. Pengenalan teks didalam gambar seperti plat motor dan sebagainya.
- e. Memodifikasi warna dan kualitas gambar misalnya Cain Scanner.
- f. Mengembangkan aplikasi *Augmented Reality*.

Beberapa keuntungan *OpenCV*, antara lain :

- a. banyak tutorial yang tersedia sehingga mudah untuk dipelajari.
- b. Bekerja di hamper semua Bahasa pemrograman, seperti : C++ b. Android SDK, JAVA dan Python e.
- c. Gratis digunakan Menurut situs resmi *OpenCV*,

pustaka ini telah diunduh lebih dari 14 jutakali pada oktober 2018. Komunitas pemakainya mencapai lebih dari 47 ribu. Beberapa diantara mereka adalah pekerja di perusahaan besar seperti IBM, Microsoft, Intel, SONY, Siemens, dan Google serta pusat-pusat riset seperti Stanford, MIT, CMU, Cambridge, dan INRIA (Bradsky, G. & Kabler, A., 2008).

## 15. Flowchart

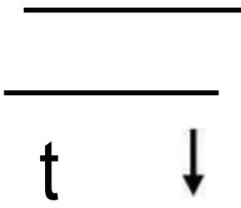
*Flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program,. Biasanya mempengaruhi penyelesaian masalah yang kbusunya perlu dipelajari dan dievaluasi lebih lanjut. Indrajani (2011 :22). Flowchart dapat digunakan untuk menyajikan kegiatan manual, kegiatan pemrosesan ataupun keduanya. *Flowchart* di bedakan menjadi 5 jenis flowchart, antara lain *system flowchart*, *document flowchart*, *schematic flowchart*, *program flowchart*, *process flowchart*.

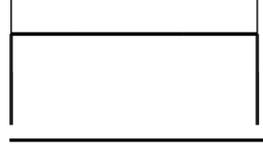
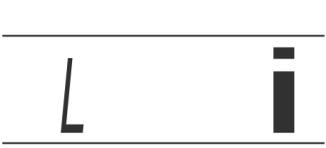
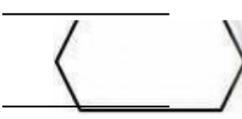
- a. System Flowchart dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan• urutan dari prosedur-prosedur yang ada di dalam sistem. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem.
- b. Docwnent *Flowchart* Bagan alir dokwnen (*documentjlo1vchart*) atau disebut juga bagan alir formulir (*form flowcharts* atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan embusannya.
- c. *Schematic Flowchart* Bagan alir skematik (*schematic flowcharts* merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaanya adalah, bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem, juga rnenggunakan gambar• gambar computer dan peralatan lainnya yang digunakan. Penggunaan gambar-garnbar ini mudah untuk dipahami, tetapi sulit dan lama menggambaranya.
- d. Program *Flowchart* Bagan ali program (*prograinf/Ol,vchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem. Bagan alir program dapat terdiri dari dua macam, yaitu bagan alir logika program (*program logic flowharty* dan bagan alir program computer terinci (*detailed computer program flowcharts*. Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program computer secara logika.

Bagan alat logika program ini dipersiapkan oleh analis sistem. Bagan alir program komputer terinci (*detailed computer program flowcharts*) digunakan untuk menggambarkan instruksi-instruksi program computer secara terinci. Bagan alir ini dipersiapkan oleh pemogram.

- e. *Process Flowchart* Bagan alir proses (*process flowcharts*) merupakan bagan alir yang banyak digunakan di teknik *industry*. Bagan alir ini juga berguna bagi analis sistem untuk menggambarkan proses dalam suatu prosedur.

Tabel 1 Symbol Flowchart

No	Symbol	Nama	Keterangan
1		<i>Input/output</i>	Untuk menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
2		<i>Disk Storage</i>	Untuk menyatakan input berasal dari disk atau output di simpan ke disk
3		<i>Document</i>	Untuk mencetak document
4		<i>Arus/Flow</i>	Untuk menyatakan jalannya arus suatu proses
5		<i>Connector</i>	Untuk menyatakan sambungan dan proses ke proses lainnya dalam halaman/lembaran sarna

6		<i>Office connector</i>	Untuk rnenyatakan sambungan dari suatu proses ke proses lainnya
7		<i>proses</i>	Sebuah fungsi pemrosesan yang dilaksanakan oleh computer untuk menghasilkan perubahan
8		<i>Symbol manual</i>	Untuk menyatakan suatu Tindakan (proses) yang tidak dilakukakn oleh cornputeri (manual)
9		<i>Decisionllogika</i>	Untuk rnenunjukkan suatu kondisi tertentu dengan dua kemungkinan ya/tidak
10		<i>predejkined</i>	Untuk menyatakan penyediaan tempat penyunpanan suatu pengolahan untuk memberi harga awal
11		<i>Teriminal</i>	Untuk menyatakan permulaan atau akhir suatu program
12		<i>Offline storage</i>	Untuk menunjukkan bahwa data dalam symbol ini akan disimpan ke suatu media tertentu

## B. Penelitian Terikait

Tabel 2 Penelitian Terikait

No	Peneliti	Judul Penelitian	Metode Penelitian	Basil Penelitian (Kesimpulan)
1	Helviani Zebua, 2023	Pembuatan Model Deteksi Gejala Awai Penyakit Mulut dan Kuku Pada Sapi	<i>Convoltional Neural Network (CNN)</i>	Hasil uji citra gejala awal penyakit rmlut dan kuku pada Sapi 01enunjukkan persentase akurasi tertinggi mencapai 91%.
2	Sona Nova Ria, Miftahul Walid, Busro Akramul Uma01, 2022	Pengolahan Citra Digital Untuk Identifikasi Jenis Penyakit Kulit Menggunakan Metode	<i>Convolutional Neural Network (CNN)</i>	Hasil uji citra Digital Untuk Identifikasi Jenis Penyakit Kulit menunjukkan persentase akurasi tertinggi mencapai 85%
3	Prayitna, Dimas Surya. 2022	Deteksi penyakit daun tomat dengan Algoritrna <i>You Only Look Once (YOLO)</i>	<i>You Only Look Once (YOLO)</i>	Nilai precision, recall dan F-1 score pada iterasi ke-3.000 secara berurutan adalab 0.71, 0.53, dan 0.63. Untuk nilai mean Average Precision (mAP) nya

---

					adalah sebesar 68.17% dengan average IoU sebesar 55.53%.
4	Feri Agustina, Muhammad Sukron,2022	Deteksi Kematangan Buah Pepaya Menggunakan Algoritma YOLO Berbasis Android	<i>Algoritma YOLO</i>		Penggunaan metode YOLO mendeteksi objek dengan cepat, dan mempunyai performa yang lebih baik, karena dapat secara cepat dan akurat menunjukkan informasi objek.
5	Moh. Rosidi Zamronia, Qabilah Cita K. N. Sb, Agung Wahyudic, 2022	Sistem Pakar Diagnosa Penyakit Sapi Sebagai Upaya Pencegahan Penyebaran Wabah PMK Di Lamongan	<i>algoritma fonvard chaining.</i>		Hasil dari Sistem Pakar Diagnosa Penyakit Sapi dengan metode <i>Forward chaining</i> dapat diimplementasikan untuk aplikasi sistem pakar yang berbasis web dengan baik.

---

Penelitian pertama dilakukan oleh Helviani Zebua Teknik Elektro Dan Informatika Fakultas Sains dan Teknologi Universitas Iambi Judul "Pembuatan Model Deteksi Gejala Awal Penyakit Mulut dan Kuku Pada Sapi Berbasis Citra Menggunakan Metode *Convolutional Neural Network*" Penyakit Mulut dan Kuku (PMK) merupakan penyakit yang sangat menular dan disebabkan oleh virus penyakit mulut dan kuku, yaitu *Foot and Mouth Disease Virus (FMDV)* pada hewan berkuku belah. Sejak tahun 1887, Indonesia beberapa kali mengalami wabah PMK. Tersebar nya PMK ke populasi hewan rentan di Indonesia dengan

cepat disebabkan oleh benda yang terkontaminasi virus PMK serta lalu lintas hewan dan produknya. Untuk meminimalisir penyebaran PMK, diperlukan pendeteksian PMK sedini mungkin. Dalam kasus pendeteksian objek, salah satu metode yang dapat digunakan adalah *Convolutional Neural Network (CNN)*. CNN dapat bekerja lebih baik dalam pembuatan model dengan dataset citra dibandingkan dengan jenis algoritma neural network lainnya karena jaringan pada metode CNN menghususkan neuron untuk mengenali bentuk citra (pada tahap konvolusi). Penelitian ini dilakukan untuk menemukan arsitektur model deteksi gejala awal PMK pada sapi dengan metode CNN dengan akurasi model terbaik dalam pendeteksian gejala awal PMK pada sapi. Hasil penelitian ini menunjukkan bahwa penambahan jumlah *epoch* dan nilai *batch size* dapat mempengaruhi akurasi dari model CNN. Pada penelitian ini, jumlah epoch 20 dan nilai *batch size* 32 menghasilkan akurasi terbaik dengan nilai akurasi training sebesar 99% dan akurasi pengujian pada 22 citra adalah sebesar 91%.

Penelitian kedua dilakukan oleh Sona Nova Ria, Miftahul Walid, Busro Akramul Umam, 2022 Program Teknologi informatika, FakuJtas Teknik, Universitas Islam 1udul "Pengolahan Citra Digital Untuk Identifikasi Jenis Penyakit Kulit Menggunakan Metode *Convolutional Neural Network (CNN)*" Penyakit kulit merupakan penyakit yang paling umum dan paling cepat menginfeksi tubuh manusia. Hal ini terjadi karena kulit merupakan organ pertama yang menerima rangsangan dari luar berupa sentuhan, suhu dan rangsangan lainnya. Penyakit kulit terdiri dari beberapa jenis yang memiliki warna dan tekstur yang hampir sama dengan kasat mata. Oleh karena itu, diperlukan suatu pendekatan untuk mengidentifikasi jenis penyakit kulit dengan bantuan sistem pengolah citra, dan jaringan syaraf tiruan. Metode identifikasi yang digunakan dalam penelitian ini adalah *Convolutional Neural Network (CNN)*. Citrakulit yang terinfeksi digunakan sebagai citra masukan untuk pemrosesan citra. Sebelum diidentifikasi dilakukan image preprocessing yaitu resizing, grayscale, menggunakan metode *Convolutional Neural Network*. Proses pengujian pada penelitian ini menggunakan 70 jenis citra penyakit kulit, data validasi dan 35 jenis citra penyakit kulit untuk data pengujian. Hasil dari penelitian ini adalah metode Convolutional Neural

Network dapat mengenali setiap jenis citra penyakit kulit dengan akurasi 98% pada proses pengujian validasi dan 85% pada proses pengujian.

Penelitian ketiga dilakukan oleh Prayitna, Dimas Surya. 2022 Program Studi Teknik Informatika, Fakultas Sains dan Teknologi UIN Sunan Gunung Djati Bandung Judul "Deteksi penyakit daun tomat dengan Algoritma *You Only Look Once (YOLO)*" Tomat merupakan salah satu sayuran yang banyak diproduksi di Indonesia dan menjadi salah satu sayuran yang paling banyak dimanfaatkan di Indonesia. Selain karena banyaknya produksi tomat di Indonesia, tomat juga merupakan tanaman yang sering terserang berbagai macam penyakit. Beberapa penyebabnya adalah bakteri, jamur serta virus. Di Indonesia sendiri, seringkali penyakit yang menjangkit pada daun tomat adalah daun yang membusuk serta bercak-bercak yang disebabkan oleh bakteri. Penyakit-penyakit ini dapat berakibat pada mutu tanaman tomat yang berkurang. Masalahnya, penyakit-penyakit yang menjangkit tanaman ini masih diidentifikasi secara manual. Penelitian ini memiliki tujuan untuk mendeteksi penyakit pada daun tomat dengan algoritma *You Only Look Once (YOLO)* dan mengetahui hasil pengujian untuk mendeteksi penyakit daun tomat dengan algoritma *You Only Look Once (YOLO)*. Metode *You Only Look Once v3 (YOLOv3)* adalah metode deteksi objek yang digunakan dalam penelitian ini. *You Only Look Once (YOLO)* merupakan metode deteksi objek yang melakukan tiga tahapan: 1. mengubah ukuran gambar menjadi 448 x 448, 2. menjalankan jaringan saraf tunggal (single neural network) dan yang terakhir melakukan *Non-Maximum Suppression* untuk mengeliminasi kotak pembatas yang tidak memiliki nilai paling maksimal. Nilai precision, recall dan F-1 score pada iterasi ke-3.000 secara berurutan adalah 0.71, 0.53, dan 0.63. Untuk nilai *mean Average Precision (mAP)* nya adalah sebesar 68.17% dengan *average IoU* sebesar 55.53%.

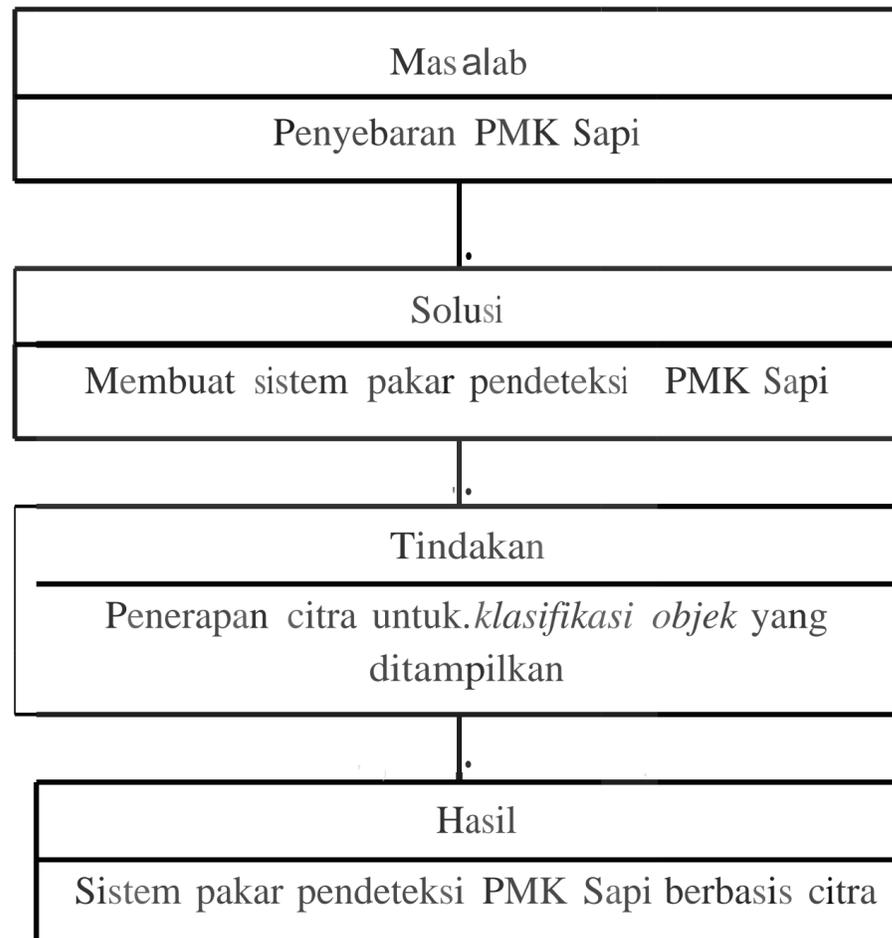
Penelitian keempat dilakukan oleh Feri Agustina, Muhammad Sukron, 2022 Program Studi Teknik Informatika, Fakultas Teknik Universitas George Mason Judul "Deteksi Kematangan Buah Pepaya Menggunakan Algoritma YOLO Berbasis Android" Indonesia dikenal dengan negara tropis, dimana semua tumbuhan tumbuh subur. Salah satu pohon buah yang mudah sekali tumbuh adalah

pohon pepaya. Salah satu kelebihan tanaman papaya yaitu tidak mengenal musim sehingga papaya mampu berbuah terus-menerus tanpa bergantung musim. Dalam produksi massal buah pepaya, proses identifikasi kematangan buah pepaya merupakan hal yang sulit. Dengan cara manual, kematangan buah pepaya ditentukan berdasarkan analisis visual warna kulit buah oleh mata manusia yang dapat mengalami kelelahan dan membutuhkan waktu yang cukup lama untuk mengklasifikasi dan mengidentifikasi tingkat kematangan buah pepaya. Maka dibutuhkan solusi untuk dapat mengenali kematangan buah pepaya dengan bantuan komputer untuk mengklasifikasikan kematangan buah pepaya dengan benar yaitu dengan menggunakan kemampuan teknologi pendeteksi objek. Pendeteksi objek atau lebih dikenal dengan *object detection* merupakan penerapan komputer untuk mendeteksi dan mengidentifikasi objek tertentu. Penggunaan metode YOLO mendeteksi objek dengan cepat, dan mempunyai performa yang lebih baik, karena dapat secara cepat dan akurat menunjukkan informasi objek.

Penelitian kelima dilakukan oleh Moh. Rosidi Zamronia, Qabilah Cita K. N. Sb, Agung Wahyudic, 2022 Program Studi Teknik Informatika, UNISLA Lamongan, Indonesia Judul " Sistem Pakar Diagnosa Penyakit Sapi Sebagai Upaya Pencegahan Penyebaran Wabah PMK Di Lamongan " Pada 5 Mei 2022, Indonesia dilanda wabah PMK pada peternakan. Hingga saat ini, pada Agustus 2022, terdapat 20 provinsi, 215 kabupaten/kota, 1.808 kecamatan, dan 7.434 desa yang masih aktif terjangkit penyakit kuku dan mulut (PMK). Munculnya PMK merupakan wabah yang mengakibatkan kerugian besar bagi peternak sapi. Apalagi jika Anda tidak memiliki pengetahuan tentang penularan penyakit tersebut. Salah satu upaya pencegahan penyebaran penyakit ternak adalah dengan menerapkan sistem pakar diagnosa penyakit ternak dengan *metode forward chaining*. Dimana sistem ini akan membantu peternak untuk memberikan pengetahuan gejala penyakit dan diagnosa penyakit, sehingga peternak mampu melakukan tindakan awal untuk menangani ternaknya.

### C. Kerangka Pikir

Untuk menyelesaikan permasalahan pada bab sebelumnya maka penulis membuat gambaran singkat sebagai alur penyusunan laporan ini dengan kerangka pemikiran sebagai berikut:



Gambar 8 Kerangka Pikir

## **BAB III**

### **METODE PENELITIAN**

#### **A. Tempat dan Waktu Penelitian**

Dalam penelitian ini penulis akan melakukan penelitian pada tempat Sentra Temak Bantaeng (STB), Desa Lonrong, Kecamatan Eremerasa, Kabupaten Bantaeng, Sulawesi Selatan.

#### **B. Alat dan Bahan**

a. Alat

Alat yang akan di gunakan dalam penelitian ini yaitu:

b. Perangkat Keras

Adapun perangkat keras yang akan digunakan yaitu laptop Lenovo

c. Perangkat Lunak

Adapun perangkat lunak (software) yang akan digunakan pada penelitian ini yaitu :

*Microsoft windows 16 64 bit*

*Microsoft office word 16 64 bit*

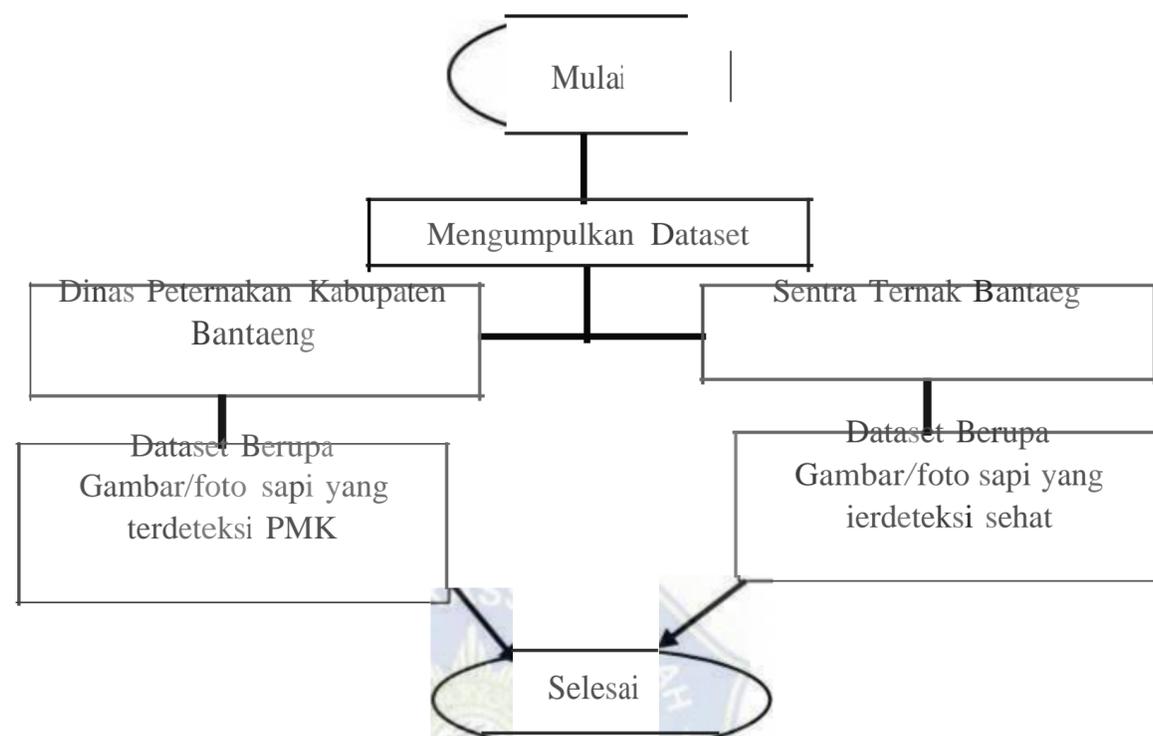
*Text editor visual studio code*

d. Bahan

Adapun yang menjadi bahan penelitian yang akan digunakan adalah dataset gambar terdeteksi PMK sapi yang diambil Pusat Kesehatan Hewan Kabupaten Bantaeng dan dataset gambar muJut dan kuku sapi sehat atau tidak terjangkit PMK yang di ambil di Sentra Ternak Bantaeng.

### C. Perancangan Sistem

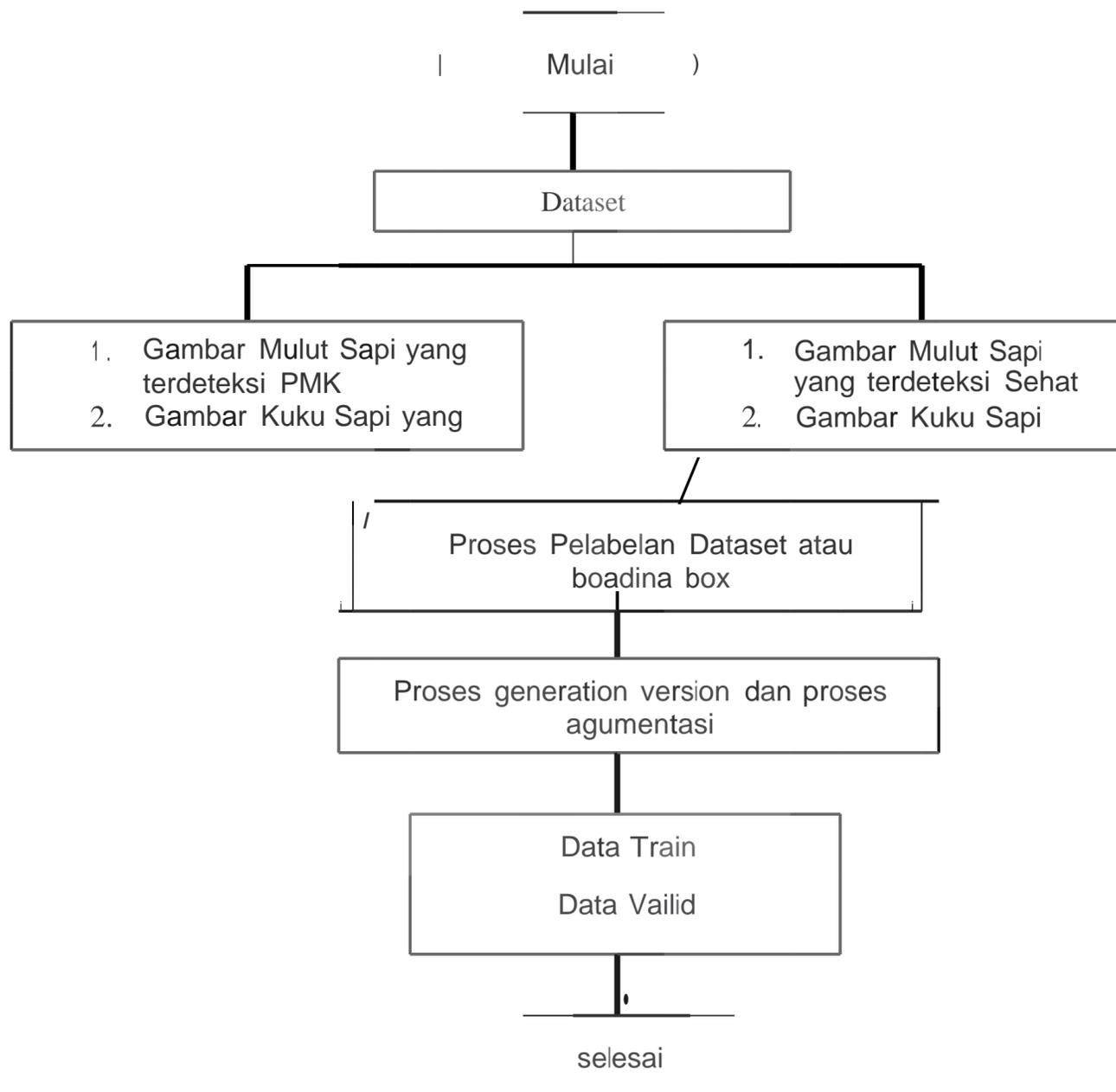
*Flowchart* merupakan serangkaian bagian yang menggambarkan alir system dari pendeteksian penyakit PMK sapi .berikut ini *Flowchart* untuk tahapan penelitian :



Gambar 9 Alur Pengumpulan Dataset

Berikut penjelasan tahap-tahap pada flowchart diatas :

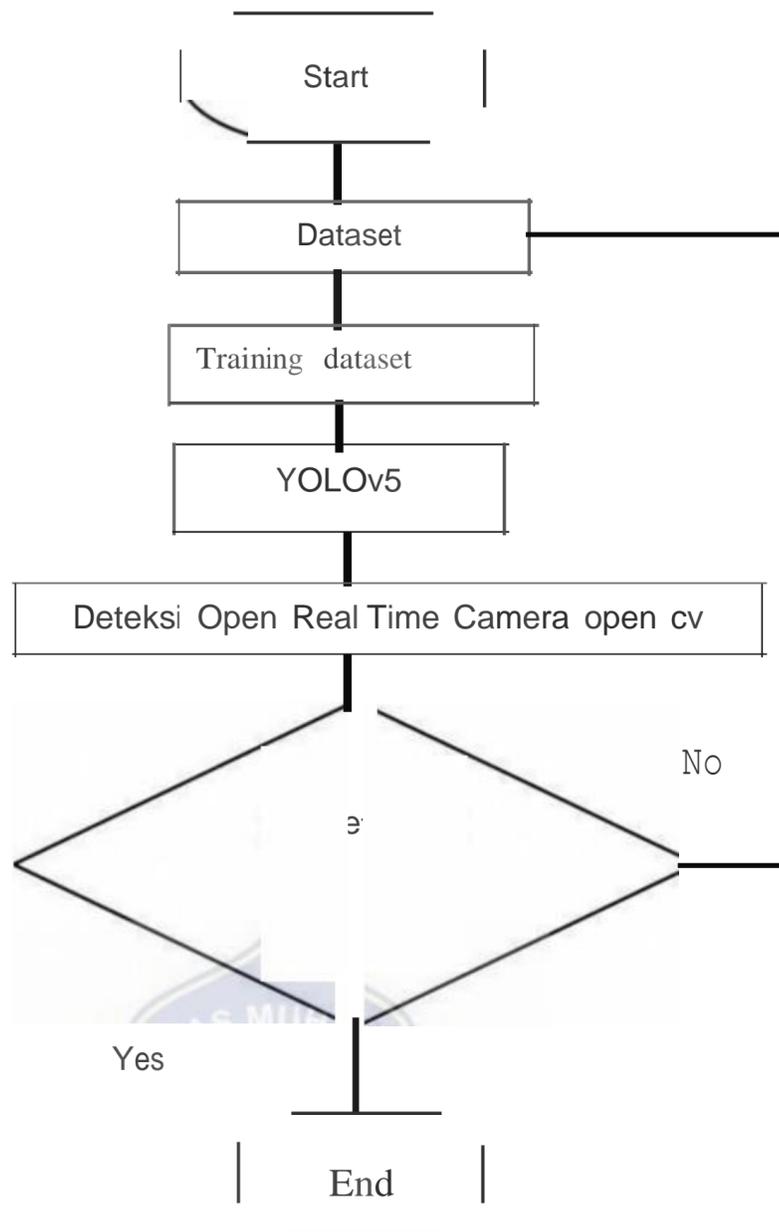
1. Mulai mengumpulkan dataset
2. Mengumpulkan dataset
3. Pengambilan dataset di pusat kesehatan hewan kabupaten bantaeng ,dataset yang di ambil merupakan gambar sapi terdeteksi PMK
4. Pengambilan dataset di sentra ternak bantaeng merupakan gambar sapi yang sehat atau tidak terjangkit PMK
5. Selesai mengumpulkan dataset



Gambar 10 Alur Pelabelan Dataset

Berikut penjelasan tahap-tahap *padaflowc/lart* diatas :

1. Mulai pelabelan
2. dataset yang sudah di kumpulkan dari sapi yang terdeksi PMK dan sapi yang tidak terdeksi PMK atau sehat akan di lakukan proses labelin
3. akan dilakukan Proses Annatation atau beading box dataset di aplikasi robotlow
4. dari Proses *Annatation* atau *boading box* dan *generation vesion* dan *agumentasi* tersebut akan menghasilkan data train.data testing dan data valid.
5. Selesai pelabelan *dataset*



Gambar 11 Alur Sistem Pakar Pendeteksian  
PMK

Berikut penjelasan tahap-tahap pada flowchart diatas :

1. Start atau mulai
2. Training Images atau hasil dataset yang sudah ditraining
3. Kemudian Training Images kemudian terproses ke algoritma yolo v5
4. Kemudian akan terbuka aplikasi kamera dalam system dalam bentuk real time untuk memproses objek apakah terdeteksi atau tidak
5. Danjika tidak terdeteksi akan ke training images untuk memproses Kembali dan apabila terdeteksi akan menampilkan hasil terdeteksi dengan akurasi
6. Kemudian Selesai

#### D. Teknik Pengujian Sistem

Dalam pembuatan sistem pakar pendeteksi penyakit mulut dan kuku objek dilakukan beberapa hal yang wajib dilakukan yaitu pengujian dan evaluasi model machine learning untuk mengukur kinerja model dalam mengklasifikasikan data ke dalam berbagai kategori atau kelas. Ini adalah komponen penting dari evaluasi model *klasifikasi*, termasuk dalam kasus pengujian pada tugas seperti deteksi objek menggunakan model seperti YOLOv5.

*Confusion matrix* berisi informasi tentang bagaimana model yang melakukan prediksi untuk setiap kelas dalam dataset. Yang membantu memahami sejauh mana model berhasil mengklasifikasikan data ke dalam kelas yang benar dan mengidentifikasi jenis kesalahan yang dibuat oleh model.

*Confusion matrix* terdiri dari empat sel:

1. *True Positive* (TP): Data yang benar-benar positif dan diklasifikasikan dengan benar oleh model sebagai positif.
2. *True Negative* (TN): Data yang benar-benar negatif dan diklasifikasikan dengan benar oleh model sebagai negatif.
3. *False Positive* (FP): Data yang sebenarnya negatif tetapi salah diklasifikasikan oleh model sebagai positif (kesalahan jenis 1).
4. *False Negative* (FN): Data yang sebenarnya positif tetapi salah diklasifikasikan oleh model sebagai negatif (kesalahan jenis 2).

Dengan menggunakan angka-angka ini, Anda dapat menghitung berbagai metrik evaluasi kinerja seperti akurasi, presisi, recall (sensitivitas), nilai F1, dan lainnya. Metrik ini memberikan pandangan yang lebih mendalam tentang kinerja model Anda daripada sekadar melihat akurasi saja.

*Confusion matrix* sangat penting dalam menganalisis hasil dan menentukan apakah model Anda berhasil mengklasifikasikan data dengan benar, serta dalam mengidentifikasi jenis kesalahan yang paling sering terjadi.

## E. Teknik Analisis Data

Untuk mencapai hasil yang diinginkan, peneliti melakukan serangkaian tahapan pengolahan dataset sebelum dilakukan metode yang ditentukan. Secara garis besar tahapan yang dilakukan sebagai berikut:

### 1. Pengumpulan Data

*Dataset* yang akan digunakan merupakan data yang diambil dari pusat kesehatan hewan Kabupaten Bantaeng dan Sentra Temak Bantaeng. Dari data yang tersebut diambil dari data berupa penyakit mulut dan kuku pada sapi dan sapi sehat atau tidak terjangkit PMK.

### 2. Penyeleksian data

Peneliti melakukan penyeleksian data dengan tujuan untuk memperoleh data yang akan dijadikan sampel yang benar-benar sesuai dengan tujuan penelitian. Data yang diseleksi berupa Gambar penyakit mulut dan kuku sapi dan mulut dan kuku sapi sehat

### 3. Preprocessing

Ditahap ini dataset akan dilakukan *preprocessing* yang terdiri dari beberapa proses yaitu pelabelan dan testing dataset.

### 4. Pengujian

Pada tahap ini melakukan pengujian terhadap dataset training yaitu pengujian dan evaluasi model *machine learning* untuk mengukur kinerja model dalam mengklasifikasikan data ke dalam berbagai kategori atau kelas.

## BAB IV

### BASIL DAN PEMBAHASAN

Pada bah ini, penulis akan menjelaskan tabap-tabap dalam memperoleh hasil akhir penelitian berdasarkan metode yang telah dijelaskan pada bah sebelumnya.

#### A. Pengumpulan Dataset

Pada tabap awal penelitian, penulis melakukan pengumpulan dataset berupa gambar sapi yang terdeteksi PMK dan sapi terdeteksi sehat atau tidak terjangkit PMK. Jumlah keseluruhan data gambar yang dikumpulkan sebanyak 8,289. Data gambar tersebut di dapatkan dari pusat kesehatan hewan Kabupaten Bantaeng dan Sentra Ternaka Bantaeng.

Tabel 3 Table daftar dataset gambar yang dikumpulkan

Tempat Pengambilan Data	Objek Pengambilan Data	Jumlah Data yang didapatkan
Pusat Kesehatan Hewan		
Kabupaten Bantaeng	Gambar Sapi Terdeteksi PMK	3,856 dataset gambar
Sentra Ternaka		
Bantaeng	Gambar Sapi Terdeteksi Sehat	4,433 dataset gambar
Jumlah Keseluruhan data yang diperoleh		8,289 dataset gambar

Data gambar yang didapatkan dari pusat kesehatan hewan Kabupaten Bantaeng merupakan data gambar sapi terdeteksi PMK, Data gambar diambil oleh dokter hewan setempat saat kasus PMK sedang mewabah di Kabupaten bantaeng yang terjadi pada awal hingga pertengahan tahun 2022. Data gambar yang didapatkan dari pusat kesehatan hewan Kabupaten Bantaeng adalah sebanyak 3,856 gambar.

Berikut Gambar Sapi yang Terdeteksi PMK



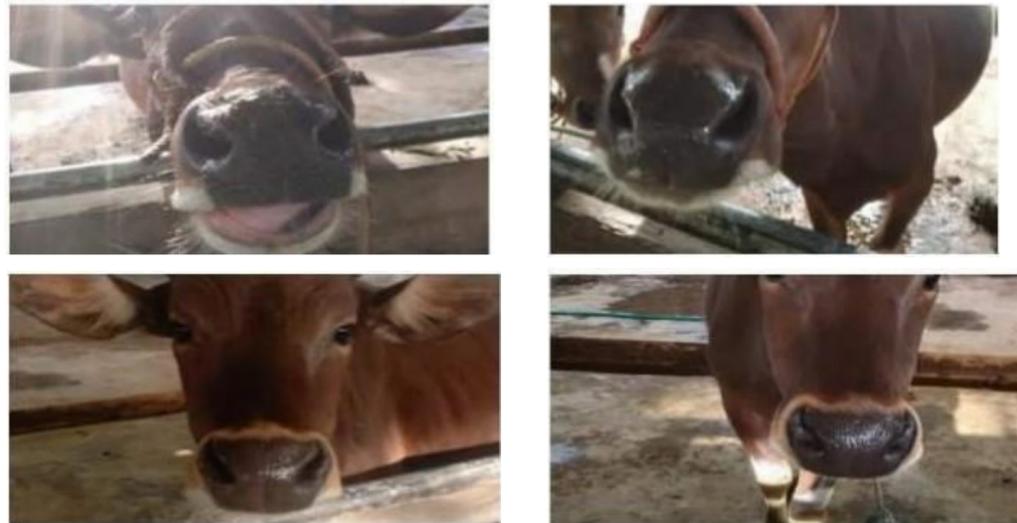
Gambar 12 Mulut sapi yang terdeteksi PMK



Gambar 13 kuku sapi yang terdeteksi PMK

Data gambar yang didapatkan dari Sentra Ternak Bantaeng merupakan data gambar sapi yang terdeteksi sehat atau tidak terdeteksi PMK. Data gambar diambil secara langsung oleh penulis pada tanggal 11 Juni 2023 menggunakan kamera *smartphone* dan data yang didapatkan adalah sebanyak 4,433 gambar.

Berikut gambar sapi yang terdeteksi sehat atau tidak terdeteksi PMK



Gambar 14 mulut sapi yang terdeteksi sehat atau tidak terdeteksi PMK



Gambar 15 mujut sapi yang terdeteksi sehat atau tidak terdeteksi PMK

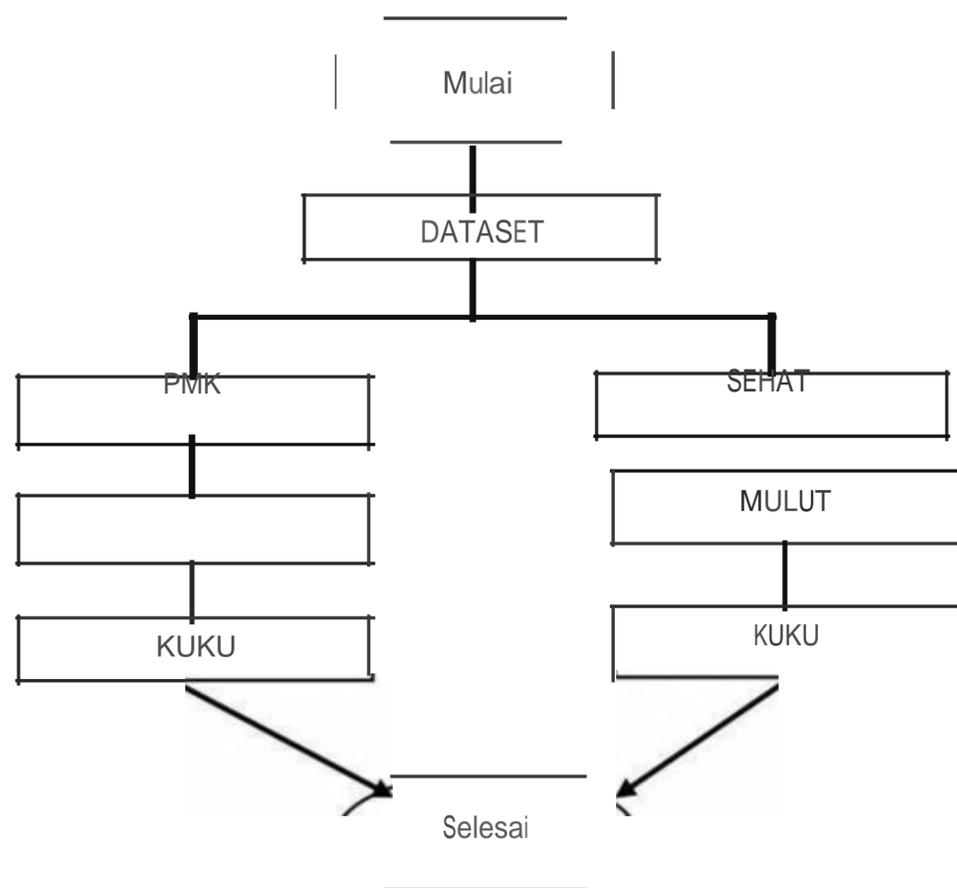
## B. Pembagian Dataset

Proses selanjutnya yaitu pembagian *dataset*, dalam tahap ini dataset yang berupa gambar sapi terdeteksi PMK dan sapi terdeteksi sehat akan di bagi menjadi 4 *class* yaitu mulut terdeteksi PMK, kuku terdeteksi PMK, mulut terdeteksi sehat dan kuku terdeteksi sehat, akan di masukkan kedalam folder yang sudah ditentukan karena jumlah *dataset* dari 4 *class* tersebut yang dimiliki berbeda-beda. berikut keterangan jumlah *dataset*.

Tabel 4 keterangan Jumlah Dataset

:No	:Dataset	jumlah
1	Gambar mulut terdeteksi PMK	2,138
2	Gambar kuku terdeteksi PMK	1,718
3	Gambar mulut terdeteksi sehat	2,219
4	Gambar kuku terdeteksi sehat	2,214
Total Dataset		= 8,289

Setelah dilakukan pembagian, *dataset* yang akan digunakan memiliki struktur direktori seperti berikut.

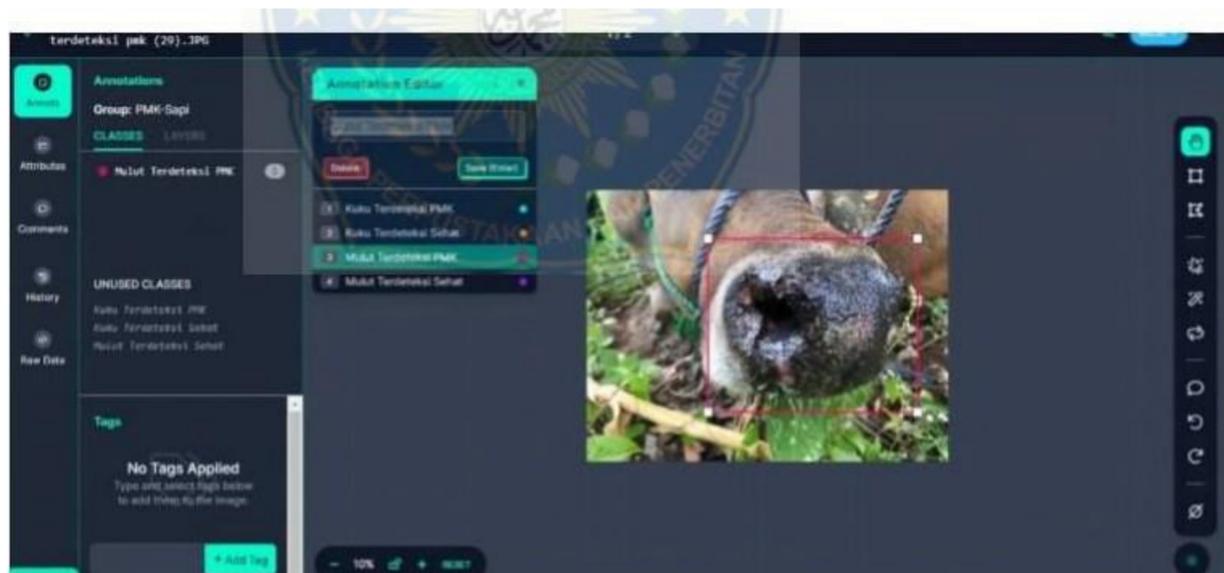


Gambar 16 Struktur Dataset

### C. Pelabelan Dataset

*Dataset* gambar sapi terdeteksi PMK dan sapi terdeteksi sehat yang dikumpulkan dan sudah dibagi kedalam direktori masing-masing akan di lakukan proses *labeling* dataset untuk mengetahui letak gambar sapi terdeteksi PMK dan terdeteksi sehat atau tidak terdeteksi PMK. Disini penulis akan melakukan pelabelan pada dataset menggunakan aplikasi *roboflow*. *Roboflow* merupakan sebuah platform yang tersedia di *web browser* yang memiliki banyak fungsi yang berhubungan dengan dataset, yang digunakan pada penelitian ini diantaranya adalah melakukan *annotate* atau menandai objek yang akan di deteksi menggunakan *bounding box*.

*Proses annotate* yaitu gambar yang sudah di simpan akan di *bounding box*., untuk menentukan gambar yang terdeteksi penyakit PMK dan gambar terdeteksi sehat atau tidak terdeteksi PM~ disini penulis sudah membagi dataset menjadi empat kelas yaitu Mulut Terdeteksi PMK .Kuku Terdeteksi PMK, Mulut Terdeteksi Sehat dan Kuku Terdeteksi Sehat.



Gambar 17 proses bounding box

Gambar diatas menjelaskan proses *bounding box* yang di lakukan di roboflow cara bounding box dilakukan untuk membatasi objek yang akan di deteksi dan memilih *class* objek jadi dari basil membatasi objek akan tampil *class* objek tersebut untuk dikenali.

Dari hasil bounding box dataset dari empat kelas akan secara otomatis membagi kedalam tiga kelas yaitu training set, validation set dan testing set berikut tabel keterangannya:

Tabel 5 dataset training set, validation set dan testing set

Dataset	Training Set	Validation Set	Testing Set
Mulut Terdeteksi PMK	1,058	487	593
Kuku Terdeteksi PMK	1,068	446	204
Mulut Terdeteksi Sehat	1,766	240	213
Kuku Terdeteksi Sehat	1,587	356	271

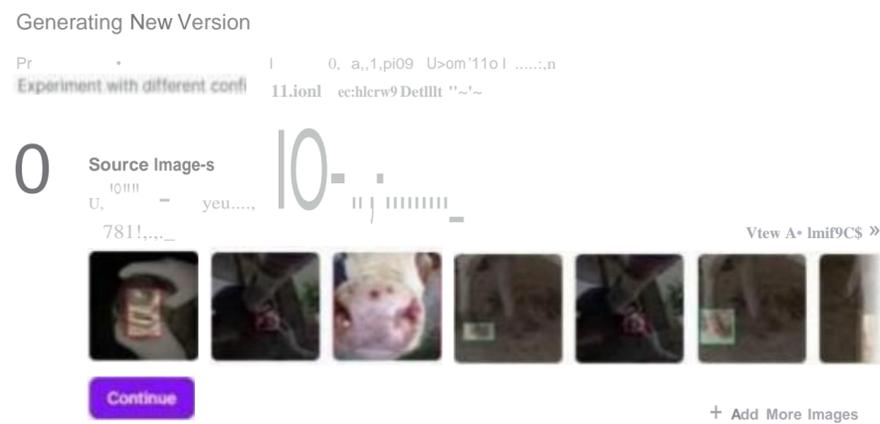
1. *Training Set* yaitu data yang digunakan untuk model training
2. *Validation Set* yaitu data yang digunakan untuk proses validasi model dan mencegah overfitting
3. *Testing Set* yaitu data yang digunakan untuk testing model sebagai simulasi penggunaan model di dunia nyata .

#### D. Generating New Version

*Generating new version* merupakan langkah untuk membuat versi kumpulan dataset untuk mengatur *training set*, *validation set* dan *testing set* dan menentukan langkah-langkah pra-pemrosesan serta tambahan untuk versi set data baru. *Dataset* yang sudah *bounding box* dan yang sudah dibagi kedalam *training set*, *validation set* dan *testing set* akan di lakukan *generating new version* untuk dilakukan penyiapan *export dataset* kedalam *algoritma yolov5*.

Dalam proses generation new version penulis melakukan

### 1. Source images



Gambar 18 source images

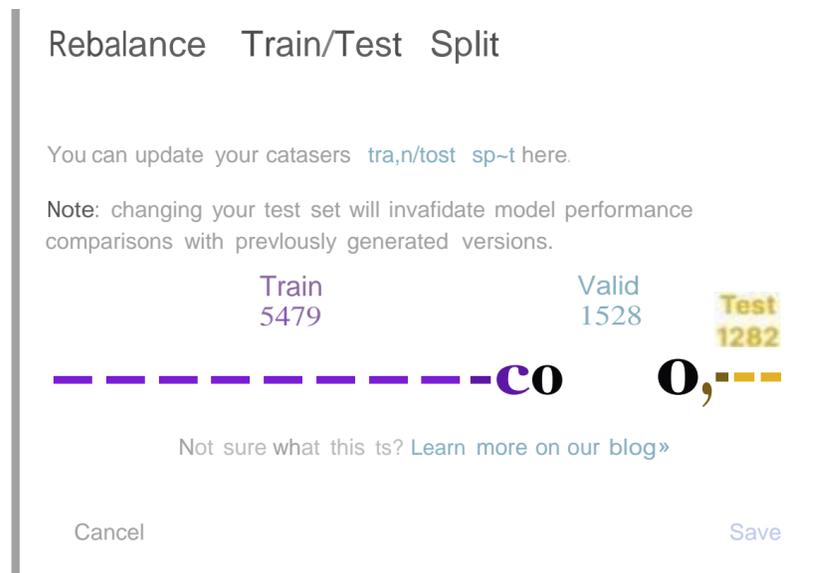
Dalam proses source images menjelaskan apakah ingin menambahkan images atau tidak ,

### 2. Train/test split



Gambar 19 train/test split

Dalam proses train/test split yaitu untuk mengatur berapa jumlah images dalam class train set, validation set dan testing set ,untuk mengatur berapa persen yang di inginkan di setiap class



Gambar 20 proses mengatur jumlah images train,valid dan testing

Gambar diatas menunjukkan proses mengatur jumlah images yang diinginkan pada *class train, valid dan testing*.

### 3. *Auto-orient : applied*



Gambar 21 proses auto-orient

*Auto-orient* merupakan fitur atau metode yang digunakan dalam *Roboflow* untuk mengatur orientasi objek secara otomatis dalam proses pembelajaran mesin (*machine learning*). Fitur ini memungkinkan sistem untuk secara cerdas menyesuaikan orientasi objek pada gambar sehingga data pelatihan menjadi lebih akurat.

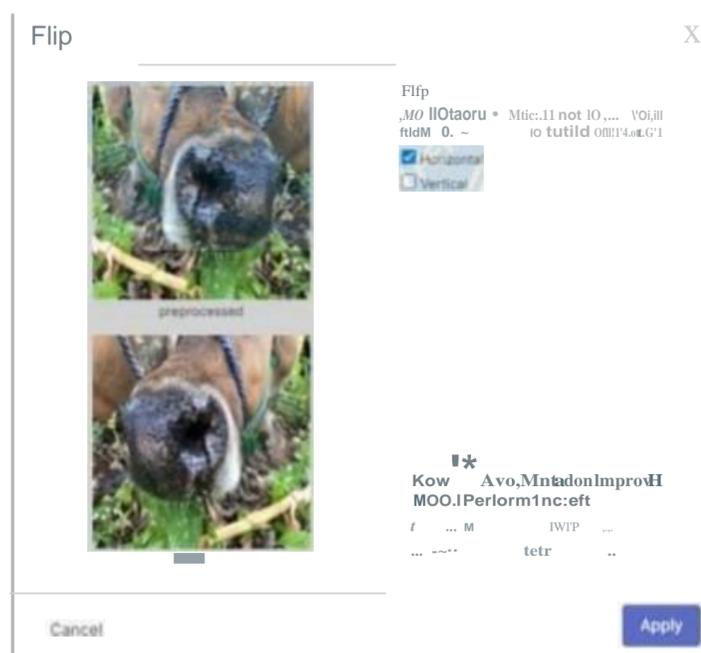
#### 4. *Resize*



Gambar 22 proses resize

*Resize* adalah proses mengubah ukuran gambar dalam dataset untuk memenuhi kebutuhan atau persyaratan tertentu. *Resize* dilakukan dengan tujuan agar semua gambar memiliki dimensi atau ukuran yang sama.

#### 5. *flip*

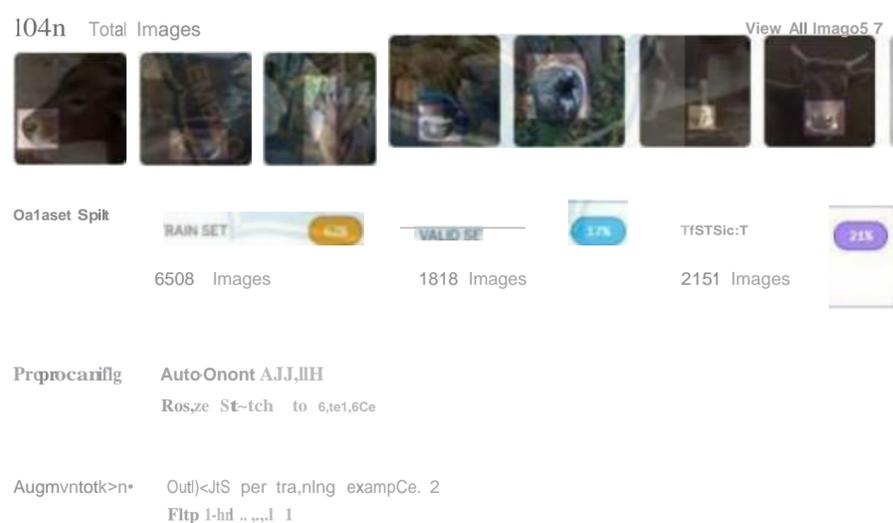


Gambar 23 proses flip

*flip horizontal* adalah salah satu fungsi atau operasi yang dapat digunakan untuk memanipulasi citra. Ketika kita menerapkan "*flip horizontal*" pada suatu citra, itu berarti kita membalikkan citra secara horizontal melintang sumbu vertikal. Dengan melakukan *flip horizontal*, setiap piksel dalam citra akan

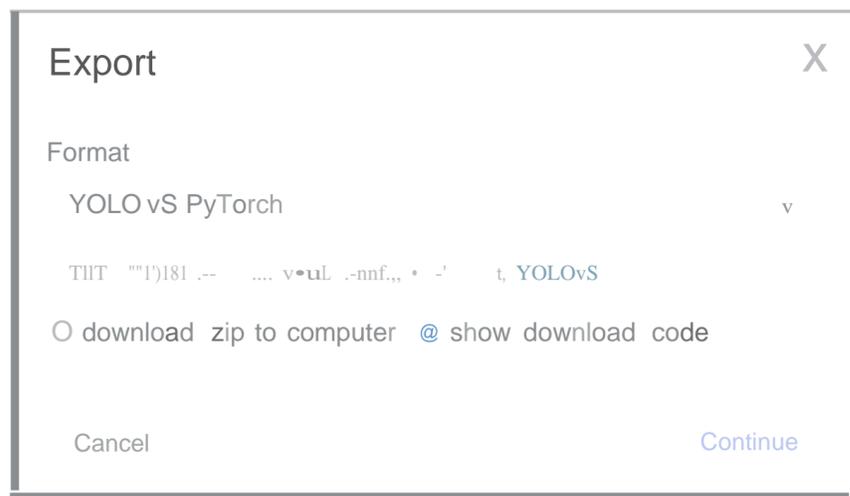
dipindahkan ke posisi yang berlawanan di sepanjang sumbu *vertikal*. Dengan demikian, sisi kiri gambar menjadi sisi kanan dan sebaliknya. Hasilnya adalah citra yang terlihat seperti cerminan dari aslinya.

*Augmentasi* data adalah teknik yang digunakan untuk meningkatkan variasi dan keberagaman dataset pelatihan dengan membuat variasi baru dari contoh-contoh yang ada. Dalam konteks "*Augmentations output per training example 2*", angka '2' menunjukkan bahwa setiap contoh pelatihan akan menghasilkan dua hasil augmentasi atau variasi baru. Dengan kata lain, setiap sampel data latihan akan diperluas menjadi tiga versi: satu asli dan dua hasil augmentasi. Peningkatan jumlah output augmentasi dapat membantu memperkaya dataset latihan, meningkatkan kemampuan model untuk mengenali pola-pola yang lebih beragam, serta memberikan ketahanan terhadap *overfitting* (ketika model terlalu khusus pada data latibannya).



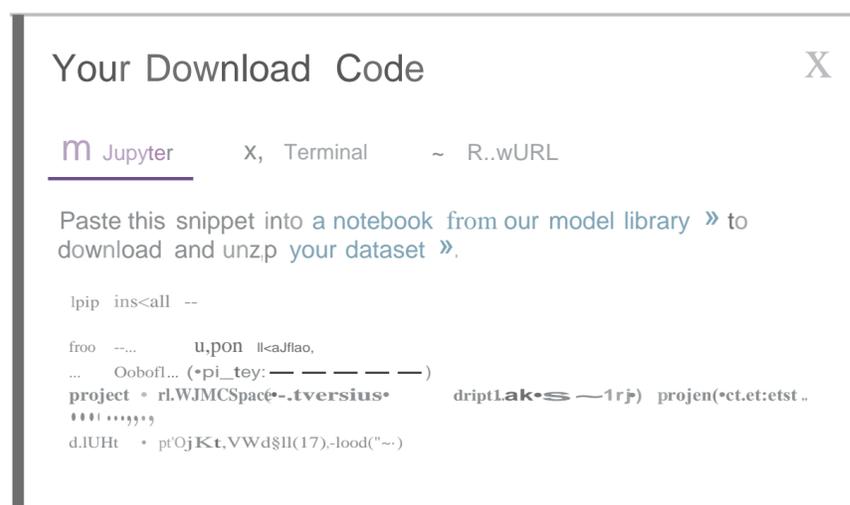
Gambar 24 hasil generation version

Gambar di atas adalah hasil dari proses generation version yang telah dilakukan mulai dari Source images yang berfungsi menambahkan dataset, split untuk mengatur jumlah image dalam *class train, valid dan testing, auto-orientation.resize* atau mengubah ukuran gambar dan augmentasi pada flip yang menambahkan gambar dalam bentuk *horizontal*



Gambar 25 export dataset hasil generation version

Gambar diatas menjelaskan basil dari proses generation version akan di export menjadi dataset *yolo v5 PyTorch* ,dalam proses tersebut akan diberi pilihan antara *download zip to computer* dan *show download code* disin penulis memilih dalam bentuk code.



Gambar 26 download code yolov5

Gambar diatas adalah bentuk dari code yang di *download* ,code tersebut merupakan dataset basil *generation version* yang akan di gunakan dalam proses training dataset. Dibawah ini *score code* basil *export ke yolov5* yang di *download*

```

pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="uaNawsDPOHlYJ8RldPDa")
project= rf.workspace("universitas-muhammadiyah-makassar•
ee72j ") .project ("deteksi-pmk-sapi")
data set = project.version (17) .download ("yolov5")

```

## E. Training Dataset

Untuk *training dataset* penulis menggunakan *google colab* untuk *mentraining dataset*, karena penulis menggunakan metode *algoritma YOLOv5* dalam pendeteksian penyakit PMK, maka melakukan training di *YOLOv5-Custom Training.ipynb*.

Sebelum langkah ke proses training dataset di *google colab* yang dilakukan pertama kali yaitu sambungkan ke GPU. Dalam konteks training dataset di *Google Co/ab*, GPU mengacu pada *akselerator* keras yang digunakan untuk meningkatkan kecepatan proses pelatihan *model machine learning*. Dengan menggunakan GPU, waktu yang diperlukan untuk melatih model bisa jauh lebih cepat dibandingkan jika hanya menggunakan CPU (*Central Processing Unit*).

### 1. Step pertama *Install Requirements*

*Install Requirements* adalah proses untuk menginstal semua persyaratan yang diperlukan sebelum menjalankan ke proses training. Dibawah ini adalah perintah yang dilakukan:

```
#clone YOLOv5 and
!git clone https://github.com/ultralytics/yolov5 # clone repo
%cd yolov5
%pip install -qr requirements.txt # install dependencies
%pip install -q roboflow
import torch
import os
from IPython.display import Image, clear_output # to display
images

print(f"Setup complete. Using torch {torch.__version__}
      {(torch.cuda.get_device_properties(0)).name if
torch.cuda.is_available() else 'CPU'}")
```

proses diatas menjelaskan

- `#clone YOLOv5 and:` perintah menjelaskan bahwa langkah pertama ada Jab mengkloning (menduplikasi) proyek atau repositori YOLOv5.
- `!git clone https://github.com/ultraJytics/yolov5:` Perintah ini digunakan untuk mengkloning repositori YOLOv5 dari URL yang diberikan ke daJam direktori.
- `%cd yolov5:` Perintah ini digunakan untuk berpindah ke direktori "yolov5" setelah proses pengklonan selesai.
- `%pip install -qr requirements.txt` Perintah ini digunakan untuk menginstal dependensi yang dibutuhkan oleh proyek menggunakan pip, dengan membaca file "requirements.txt".  
`%pip install -q roboflow:` Perintah ini digunakan untuk menginstal library Roboflow menggunakan pip.
- `import torch, import os, dan from !Python.display import Image, clear_output:` Ini digunakan untuk irnpor modul-modul yang akan digunakan daJam kode selanjutnya.
- `print(f'Setup complete. Using torch {torch.__version__} {torch.cuda.get_device_properties(0).name if torch.cuda.is_available() else 'CPU'})` : Pernyataan print ini menampilkan pesan bahwa semua persiapan sudah selesai dilakukan dan mencetak versi Torch yang sedang digunakan serta informasi tentang perangkat GPU jika tersedia.

Dengan demikian, kode tersebut bertujuan untuk mempersiapkan lingkungan kerja dengan menduplikasi repositori YOLOv5, menginstal dependensi dan melihat informasi tentang versi Torch yang digunakan serta perangkat GPU.

```

Cloning into 'yolov5' ...
remote: Enumerating objects: 15927, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 15927 (delta 20), reused 24 (delta 8), pack-reused 15880
Receiving objects: 100% (15927/15927), 14.66 MiB | 22.99 MiB/s, done.
Resolving deltas: 100% (10915/10915), done.
/content/yolov5
-----
188.5/188.5 kB 3.7 MB/s eta 0:00:00
6U.3/612.3 kB 9.0 MB/s eta 0:00:00
-----
62.7/62.7 kB 6.4 MB/s eta 0:00:00
-----
57.6/57.6 kB 1.5 MB/s eta 0:00:00
-----
155.3/155.3 kB 7.3 MB/s eta 0:00:00
-----
178.7/178.7 kB 21.3 MB/s eta 0:00:00
-----
58.8/58.8 kB 7.1 MB/s eta 0:00:00
-----
67.8/67.8 kB 7.8 MB/s eta 0:00:00
-----
59.3/59.3 kB 6.9 MB/s eta 0:00:00
-----
Preparing metadata (setup.py) ... done
-----
54.5/54.5 kB 6.8 MB/s eta 0:00:00
-----
Building wheel for wget (setup.py) ... done
Setup complete. Using torch 2.0.1+cu118 (Tesla T4)

```

Gambar 27 hasil dari proses Install Requirements

## 2. Step 2 Assemble Our Dataset

Maksud dari Assemble Our Dataset adalah mengumpulk:an atau menyusun kembali dataset kita. Yang merupakan kumpulan data yang digunakan dalam proses pembelajaran mesin atau analisis data.

```

#setup environment
os.environ["DATASET DIRECTORY"] = "/content/datasets"

```

Maksud dari kode `os.environ ["DATASET_DIRECTORY"]`

`content/datasets"` adalah untuk mengatur variabel lingkungan pada sistem operasi yang sedang digunakan. Dalam hal ini, variabel lingkungan `"DATASET_DIRECTORY"` diatur dengan nilai `"/content/datasets"`.

Variabel lingkungan atau environment variable adalah variabel yang dapat digunakan oleh program komputer untuk menyimpan informasi seperti direktori file, lokasi konfigurasi, atau nilai-nilai lain yang relevan dalam penggunaan program tersebut.

Dengan mengatur variabel lingkungan menggunakan `os.environ`, kita dapat memastikan bahwa program memiliki akses ke direktori dataset yang ditentukan ("`/content/datasets`" dalam contoh tersebut) dan bisa bekerja dengan benar. Jadi, melalui perintah tersebut, kita menetapkan direktori dataset sebagai nilai dari variabel lingkungan `'DATASET DIRECTORY'`. Ini membantu program mendapatkan referensi ke direktori dataset tanpa harus secara harcode menuliskan path-nya setiap kali dibutuhkan.

Selanjutnya code yang didownload dari roboflow hasil dari generation version yang di export kedalam bentuk yolov5 akan di install, di bawah ini adalah codenya:

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="uaNawsDPOHlYJ8RldPDa")
project= rf.workspace("universitas-muhammadiyah-makassar-ee7zj").project("deteksi-pmk-sapi")
dataset= project.version(17).download("yolov5")
```

`!pip install roboflow` adalah perintah untuk menginstal library Roboflow menggunakan `pip`, yang merupakan package manager untuk Python. Library Roboflow digunakan untuk mengakses dan memanipulasi data di platform Roboflow. Selanjutnya adalah penggunaan library Roboflow dalam mendownload dataset dari proyek "universitas-muhammadiyah-makassar-ee7zj" dengan nama proyek "deteksi-pmk-sapi" pada versi 17. Dataset ini akan didownload dengan format YOLOv5.

```

Itq1lTr°"nt Hr"eid) uthffiff c...nU . 2.tU.11.7 ht /YSr/loct1/tb/...d'Oll,lttdJt-D<bIH throa robc>flow) (2912.12.1)
lt,qlr,...,t already **ttallta: Cblrdoet*...e.e $t /"r/loc.tl1SotpythOn),Jtdit-Pt<111... (ff'OH robof0lif) (A.t.1)
~tcw:irf9t"t alrtdy **tilftct cye:iff"...e,tt.t " /w,r/loct1lib ...,V.0.,:Jltid1,tt,tt*1*t (ff'Om r,oboflow) (e.10.t)
~1lHr.Ntdt alrHO) uthfhd ldl'***1.st lll /w,r/loc.tl Ub P)l'Of0).lt d.ht-PtUCH (fro. f'000flow) (0.1t)
~eq1lHr. nt lrtady tttilftd kl*liol-tr>*1,J1 lll /llr/loc1111b/lllNl'Of0),lt/dlt pt<\..tl (ff'OH roboflow) (1.*.1)
i. qu1N11tlt alrady Utlllftcl ...tplotUb 1rt /llr/looJ lllllypt."Of0J,ltfdit-N<"iAch (frOH roboflow) (J.7.H)
lequi,...,nt lHroldy ... tt1ftld MlHP)'>.111.S " /ut.r/local/llb,rthof111td11t.p.a(*.,... (fro- rot,oflow) (1,23.S)
~qusr ... nt .l..Hy uti.Jltct otw"l<V-P)l'°"~".1.1 lll /uv lou1/Uo/P)"t"OnJlt dJjt-P.KY&d (ll'Cla .ObOfow) (.4,1,1.16)
ltqfir....,t ~lrtady a.tl1ftd Pillow>*7.1J lll /llr/loc111b/pythor),ll1d11t.p.e.latt1 (ff'OH roboflo..i) (9.*.0)
leqwl,...eent alrtdy ...tl1ftd pyoar*S-l->*.7 $t /v1r/loc11/llb1s,tl'111J,tt/dlt~k*tt' (froe t'COOfow) (J.*.1)
~ul.....nt alreildy s.attsttd pyt~dat'1Jt11 ln /v1r local/lt*t"t"l'1119/dlt-p.c.~1,1 (from l'ObOf0lf) (2.1.J)
~1<1ui-. i ..... , utl>Hld py.- ... .. i. /uv/ou1/ll>... 1.1t1*1>t-o<°°°° ff- -low) (1.e.t)
~t-qlruent lh'Hd) uthfled: re,Q,1ftU 1" /uvllloc.1111b1)'ttonJlt'Ust-p.c:1111H (frCm r'OOOfow) (0.J.t)
~tql,...,nt ll,...ldy ll\1Jftct. sla tn lu,r/loc1111b/P)'J101'11.11'dil~p.ac.~ts (~ f'!ObOfow) (1.16.8)
~lle-qldr. nt lrtN)' wthf1td SV'P'trvllanln (wv local/llD/pthor.3.le'dbt-p<u.ct's (frotl roboflow) (1. U.J)
ll-ulreHnt iihHdy snt.sHea: urllbl>*t.26.6 ln /w:rrou.Hl.lb.pytl101\11t101tt-p.cQju (froM roboflow) (2.a.,)
~flqlr...etlt lrttdy ..tj-fl.s: we,t iA /V1rtloc.all1Jb/P)fto,\11td1-t~t~t.e* (ff'OH roboflow) (1.1)
~L't'1t'1t lJrudy s.tisfteo: tqoa>~ ..1.a $t mv louJ11el~TOJ1.P.ltm-1),Kkaett (free l'ObOfow) (4.66.1)
~Srt:11tnt al,...ld' 11tS1fttd PyY~>~S.1 1 lll /utr/loc1111b1 ~.lt/dltt0,c'lJtt l'OH roooflow) (6.8.1)
leqrl-t already s.n.b.Jieo: Nqlllits-toolbutl ia /wr/ou1 lib OJUOn).1.8/dist~EH (frm robof.ow) (t.e.<~)
itqu lrt1letlt alrHO) uti sff N Ut"Ul&ll'py*Lo, s lll /u:u,loc11 lib cythof'1.18/d.in .._ups (fro. ut plot Ue->roboflow) (1.1. t)
~lQuil"MEnt ~lr-eady utisfJe: font.oc.ls.....22.e in /usrloc.i/ lib oyU"3.ltdist-pacbe-s (fr09 iwu,lotlib->roboflow) (*..l. l)
ltqwl'ffnt ~l'fady utifled: ~U&t.1>*M.I iA lolsnloc U.Jpytl0dl. lltolr. .P<UiH (ff'Cla Htplo; Ut.~>roboflow) (ll.1)
Rtqu.lrf'el'nt ~lr~adysattsfiee> c-rs.t-nQf'all1t'f'r,0*2 in lllr,loc11/llbJpytl10P'13.18/dist-plc.k:.,,1 (fr09 reqotsts->l'00of0111) t3.2.8)
~qu1,...,nt ~lrudy ut h fied: ope1lCV~pyuon~. lHu s,l.. I **..l. l.7, 1n rUS'l'ouJ Ublpython.J.leid.n pack.laf (fra- supt-Nb Uln>robof:°") (*.t.1.76)
~equ1,...,nt lll'9.cly satisfied: scipy<2.t.1.>+9.J ln (l.s.Jloc.tl1~ pyt~).1&dt,t-o.Mt=iel (fra. supervisJon->robof0lif) (1,11,1)
INdl11 Roboflow wof"~ksp.ace **
lotctna Roboflow projct.t "
Oic,w,loalJin1 D~tnet l/erslon llp i" oenk.s.l'P'C-S.it-11 to yol.O\SP\to1"dl'9" (Otl-176a / Otl91191) bytuEnnc.dng 0~uwt 'terslon lip to Detals.J~S.apl-11

```

Gambar 28 basil dari proses instal library Roboflow dan dataset yang akan di training

### 3. Step Train Our Custom YOLOv5 model

Maksud dari Train Our Custom YOLOv5 model adalah melatih atau menghasilkan model kustom menggunakan arsitektur YOLOv5. YOLO (You Only Look Once) adalah salah satu pendekatan populer dalam deteksi objek pada bidang computer vision. Arsitektur YOLOvS merupakan versi terbaru dan diperbarui dari seri YOLO yang memiliki performa lebih baik dalam mendeteksi objek dengan kecepatan tinggi.

Proses pelatihan ini melibatkan pengumpulan data latihan yang mencakup gambar-gambar dengan anotasi boks-boks pembatas (bounding boxes) untuk setiap objek target. Data ini kemudian digunakan untuk memberikan pemahaman kepada model tentang ciri-ciri visual dari objek-objekt tersebut. Setelah proses pelatihan selesai, hasilnya akan menjadi sebuah "model" yang siap digunakan dalam aplikasi deteksi objek secara real-time atau pada dataset lainnya.

Jadi proses selanjutnya adalah perintah untuk melatih model deteksi objek menggunakan framework YOLOv5.

```
python train.py --img 640 --batch 16 --epochs 100 --data  
{dataset.location}/data.yaml --weights yolov5s.pt --cache
```

Berikut adalah penjelasan dari setiap argumen dalam perintah tersebut:

"--img 640": Menentukan ukuran gambar input yang akan digunakan selama pelatihan. Dalam hal ini, ukuran gambar input adalah 640x640 piksel.

"--batch 16": Menentukan jumlah sampel (gambar) yang akan diproses dalam satu iterasi pelatihan. Dalam hal ini, batch size adalah 16.

"--epochs 100": Menentukan jumlah epoch atau iterasi pelatihan yang akan dilakukan. Sebanyak 100 epoch akan dilakukan untuk melatih model.

"--data {dataset.location}/data.yaml": Mengarahkan ke file konfigurasi datayaml yang berisi informasi mengenai dataset, seperti path ke direktori data dan label kelas.

"--weights yolov5s.pt": Menggunakan bobot awal pre-trained model "yolov5s.pt" sebagai titik awal pelatihan. Ini memungkinkan kita untuk melanjutkan proses pelatihan dari checkpoint sebelumnya jika ada.

"--cache": Memungkinkan caching dataset pada memori untuk akselerasi pembacaan data saat melakukan training.

Dengan menjalankan perintah di atas, model YOLOv5 akan dilatih menggunakan parameter dan konfigurasi tertentu seperti ukuran gambar input, batch size, jumlah epoch, serta lokasi dataset dan weight pre-trained.

Kemudian hasil dari training yang dilakukan akan mendapatkan hasil di gambar bawah ini :

```

100 epochs completed in 3.097 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.5118
Optimizer stripped from runs/train/exp/weights/best.pt, 14.5118

Validating runs/train/exp/weights/best.pt ...
Fusing layers ...
Model summary: 157 layers, 7820913 parameters, 0 gradients, 15.8 GFLOPs

```

Class	Images	Instances	P	R	AP <sub>50</sub>	AP <sub>50-95</sub>
all	1587	1649	0.606	0.614	0.589	0.205
Kuku Terdeteksi PMK	1587	441	0.957	0.954	0.952	0.187
Kuku terdeteksi Sehat	1587	432	0.835	0.847	0.831	0.122
Mulut Terdeteksi PHK	1587	459	0.967	0.958	0.963	0.219
Mulut Terdeteksi sehat	1587	310	0.813	0.858	0.837	0.293

Results saved to runs/train/exp

Gambar 29 Hasil dari training dataset

Gambar diatas menjelaskan bahwa pelatihan model YOLOv5 sebanyak 100 epochs dalam waktu 3.007 jam. Selama proses tersebut, optimizer sudah dihapus dari berkas weights/last.pt dan weights/best.pt dengan ukuran masing-masing 14.5MB. Setelah itu, dilakukan validasi menggunakan model yang tersimpan di berkas weights/best.pt. Beberapa langkah pra-pemrosesan juga dilakukan seperti penggabungan lapisan (fusing layers). Kemudian, diberikan hasil ringkasan model yang terdiri dari 157 lapisan, total parameter sejumlah 7.020.913, tidak ada perubahan gradien (0 gradients), dan kecepatan komputasi sebesar 15.8 GFLOPs.

Hasil evaluasi kinerja deteksi objek pada dataset validasi sebagai berikut:

Total gambar	1587
Total instansi objek	: 1649
Rata-rata presisi (P)	: 0.606
Rata-rata recall (R)	: 0.614
Mean Average Precision at IoU	:0.50 (mAP50): : 0.589
Mean Average Precision at IoU	:0.50 -IoU=95(mAP50-95) : 0.205

Lalu, terdapat evaluasi untuk setiap kelas objek:

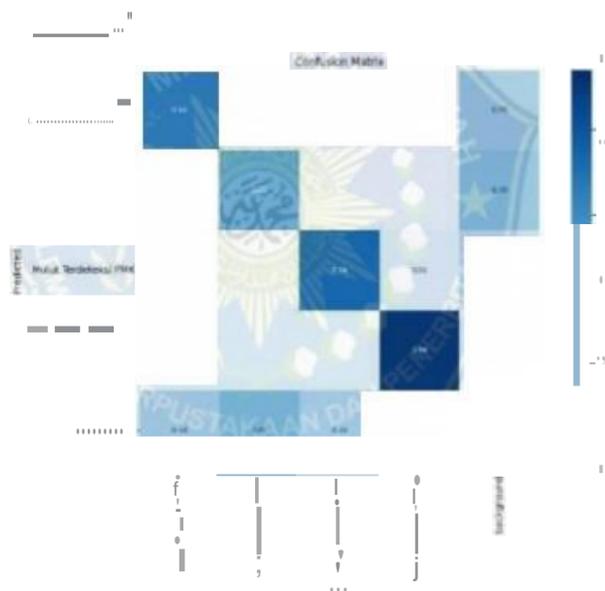
Kuku Terdeteksi PMK :Jumlah gambar: 1587  
:Jumlah instansi objek: 448  
:Presisi(P): 0.577  
:Recall(R): 0.543  
:mAP50 : 0.542  
:mAP50-95: 0.187

Kuku Terdeteksi Sehat :Jumlah gambar: 1587  
:Jumlah instansi objek: 432  
:Presisi(P): 0.359  
:Recall(R): 0.47  
:mAP50 : 0.34  
:mAP50-95 : 0.122

Mulut Terdeteksi PMK :Jumlah gambar: 1587  
:Jumlah instansi objek: 459  
:Presisi (P): 0.675  
:Recall (R): 0.584  
:mAP50 : 0.635  
:mAP50-95 : 0.219

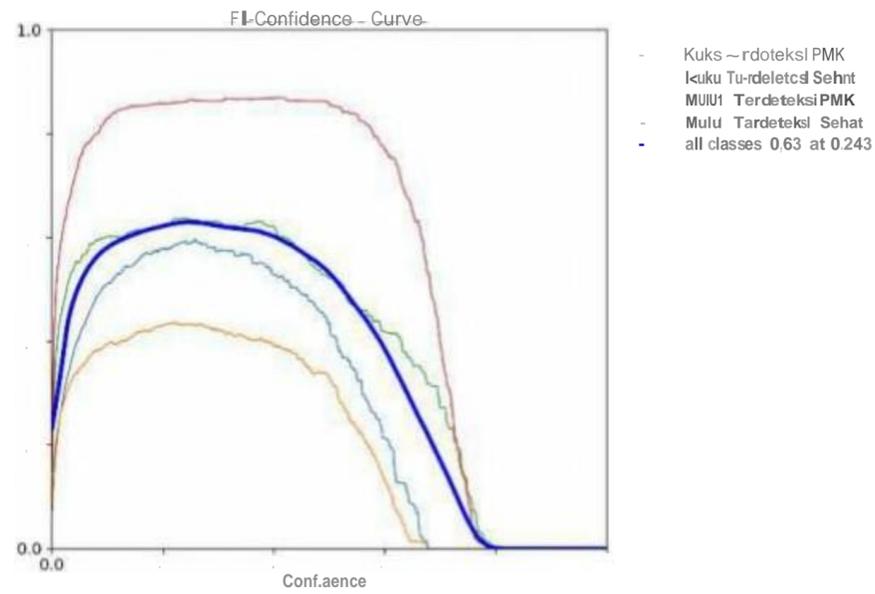
Mulut Terdeteksi Sehat :Jumlah gambar =1587  
:Jumlah instansi objek =310  
:Presisi (P) =813  
:Recall (R) =858  
:mAP50 = 0.837  
mApSO-gS = 0.293

Hasil dari proses training dataset yang tersimpan di run/train/exp akan menunjukkan *confosional matrix*, *F1-Confidece Curve*, *precision confidence curve*, *precision-Recall Curve* dan *Recall-Confidence Curve* berikut ada adalag gambarnya:



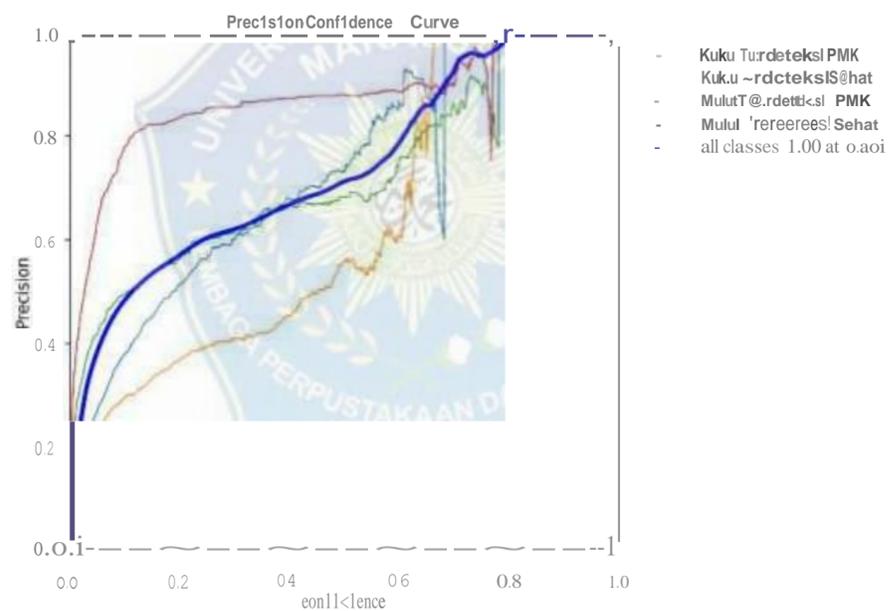
Gambar 30 hasil confosional matrix

Gambar diatas menjelaskan *bahwa Confusion matrix* adalah tabel yang digunakan untuk menggambarkan kinerja sistem klasifikasi pada dataset uji, di mana setiap baris mewakili kelas yang benar, dan setiap kolom mewakili kelas yang diprediksi oleh model.



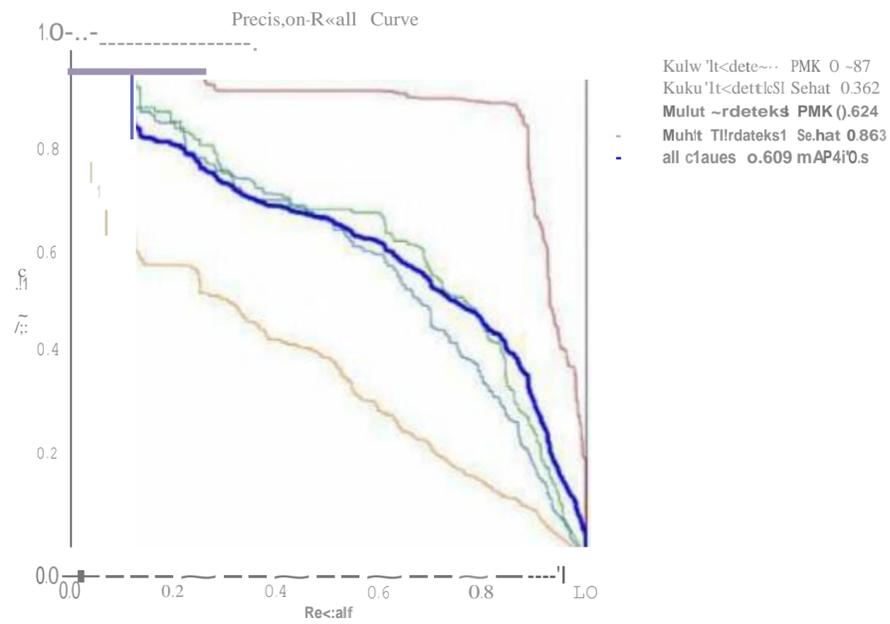
Gambar 31 hasil F1-Confidence Curve

Gambar diatas menjelaskan bahawa *F1-Confidence Curve* merupakan suatu metrik evaluasi gabungan antara presisi dan recall yang mengukur kualitas keseluruhan dari sistem deteksi objek.



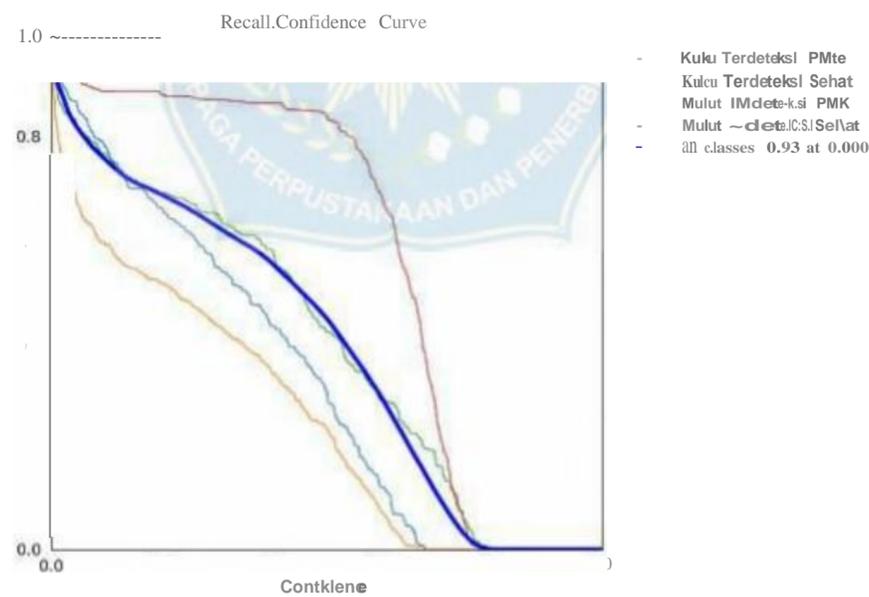
Gambar 32 basil precision confidence curve

Gambar diatas ada grafik *Precision-Confidence Curve*, yang berfungsi untuk dapat melihat bagaimana kinerja model Anda dalam menghasilkan deteksi yang akurat pada berbagai tingkat kepercayaan.



Gambar 33 hasil precision-Recall Curve

Gambar diatas merupakan *precision-Recall Curve* salah satu evaluasi yang umum digunakan dalam tugas deteksi objek seperti yang diimplementasikan dalam YOLOv5. Ini adaJah kurva yang menggambarkan hubungan antara presisi (precision) dan recall pada berbagai ambang batas kepercayaan yang digunakan daJam deteksi objek.



Gambar 34 hasil Recall-Confidence Curve

Gambar diatas merupakan suatu kurva yang menggambarkan hubungan antara recall dan ambang batas kepercayaan (*confidence threshold*) daJam tugas deteksi objek.

Kemudian proses selanjutnya adalah mengubah basil training yang dalam bentuk file best.pt menjadi file onnx.pt. ONNX (*Open Neural Network Exchange*) adalah format file yang dirancang untuk mengizinkan interoperabilitas yang lebih baik antara berbagai kerangka kerja (*framework*) dan alat (*tools*) pembelajaran mesin. ONNX menyediakan standar untuk mewakili model pembelajaran mesin, termasuk model yang digunakan dalam tugas deteksi objek seperti YOLOv5. dengan perintah berikut:

```
python export.py --weights
/content/yolov5/runs/train/exp/weights/best.pt --include onnx
--opset 11
```

Berikut ini penjelasan mengenai setiap code yang digunakan:

" --weights content/yolov5/runs/train/exp/weights/best.pt Argumen ini menentukan path (lokasi) file berat (weight) model YOLOv5 yang akan diekspor ke format ONNX. Dalam contoh ini, pathnya adalah content/yolov5/runs/train/exp/weights/best.pt.

"--include onnx: Argumen ini memberitahu skrip bahwa kita ingin menyertakan format ekspor ONNX dalam proses perubahan model.

"--opset 11: Argumen ini menentukan versi opset ONNX yang akan digunakan saat melakukan konversi model menjadi format ONNX. Dalam contoh ini, opset yang dipilih adalah versi 11.

Dengan menjalankan perintah tersebut, skrip Python (export.py) akan dijalankan dan memproses file berat model YOLOv5 (best.pt). Hasilnya nanti akan menghasilkan sebuah file dalam format ONNX dengan menggunakan opsi opset 11.

## F. Proses Pendeteksian

Dalam mendeteksi menggunakan OpenCV dan numpy dalam deteksi PMK dan tidak terdeteksi PMK dengan menggunakan model ONNX. Berikut score codenya:

```
import numpy as np
import cv2

classes= ["Kuku Terdeteksi PMK" , "Kuku Terdeteksi Sehat"
,"Mulut Terdeteksi PMK" ,"Mulut Terdeteksi Sehat"]
cap = cv2.VideoCapture(0)
net= cv2.dnn.readNetFromONNX("best.onnx")

while True:
    _, img = cap.read()

    blob= cv2.dnn.blobFromImage(img, scalefactor=1/255,
size=[640, 640], mean=[0, 0, 0], swapRB=True, crop=False)
    net.setInput(blob)
    detections= net.forward()[0]

    classes ids = []
    confidences = []
    boxes = []
    rows= detections.shape[0]

    img_width, img_height = img.shape[~], img.shape[0]
    x_scale = img_width/640
    y_scale = img_height/640

    for i in range(rows):
        row= detections[i]
        confidence= row[4]
        if confidence> 0.2:
            classes score= row[5]
            ind= np.argmax(classes score)
            if classes score[ind] > 0.2:
                classes ids.append(ind)
                confidences.append(confidence)
                ex, cy, w, h = row[:4]
                xl = int((cx-w/2)*x scale)
                yl = int((cy-h/2)*y_scale)
                width= int(w * x scale)
                height= int(h * y_scale)
                box= np.array([xl, yl, width, height])
```

```

        boxes.append(box)

indices= cv2.dnn.NMSBoxes(boxes, confidences, 0.2, 0.2)

for i in indices:
    xl, yl, w, h = boxes[i]
    label= classes[classes_ids[i]]
    conf = confidences[i]
    text = label + " : %.2f" % conf
    cv2.rectangle(img, (xl, yl), (xl+w, yl+h), (255, 0,
0) , 2)

    cv2.putText(img, text, (xl, yl-2),
cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 0, 255), 2)

cv2.imshow("Deteksi Objek", img)
if cv2.waitKey(1) & 0xff == 27:
    break

```

Berikut penjelasannya:

`import numpy as np`: Ini mengimpor library numpy dengan alias np. Numpy digunakan untuk melakukan operasi numerik yang efisien, seperti manipulasi array dan perhitungan matematika.

`import cv2`: Ini mengimpor library OpenCV (Open Source Computer Vision Library). OpenCV digunakan untuk pemrosesan gambar dan analisis komputer, termasuk deteksi objek.

`classes = ["Kuku Terdeteksi PMK", "Kuku Terdeteksi Sehat", "Mulut Terdeteksi PMK", "Mulut Terdeteksi Sehat"]`: Ini mendefinisikan daftar kelas atau label yang akan digunakan untuk menunjukkan hasil dari deteksi objek.

`cap= cv2.VideoCapture(0)`: Ini membuat objek VideoCapture untuk merekam video dari webcam atau sumber video lainnya dengan indeks 0 (biasanya merupakan webcam default).

`net= cv2.dnn.readNetFromONNX("best.onnx")`: Ini membaca model jaringan saraf tiruan (neural network) dari file ONNX ("best.onnx") Model ini telah dilatih sebelumnya untuk melakukan deteksi objek menggunakan teknik deep learning.

Looping while: Membaca frame demi frame dari video yang diambil oleh VideoCapture.

Menggunakan metode `read()` pada objek VideoCapture untuk membaca frame saat ini.

Melakukan preprocessing terhadap gambar dengan fungsi `.blobFromImage()`, yaitu mengubah gambar menjadi blob agar sesuai dengan input model.

Melakukan deteksi objek menggunakan `net.forward()`, mengambil hasil deteksi pada indeks ke-0 karena hanya satu gambar yang diproses dalam setiap iterasi.

Memproses hasil deteksi:

Mengekstraksi informasi kelas, kepercayaan (confidence), dan kotak pembatas (bounding box) dari output deteksi.

Menghitung ulang koordinat kotak pembatas sesuai dengan skala asli gambar.

Menyiripkan semua kelas, kepercayaan, dan kotak pembatas yang memenuhi threshold confidence tertentu.

Menggunakan fungsi `cv2.dnn.NMSBoxes()` untuk menghilangkan tumpang tindih antara kotak pembatas yang terdeteksi secara berlebihan dan mempertahankan hanya satu kotak dengan nilai confidence tertinggi.

Menampilkan hasil:

Meloopti semua indeks objek yang tersisa setelah NMS dilakukan.

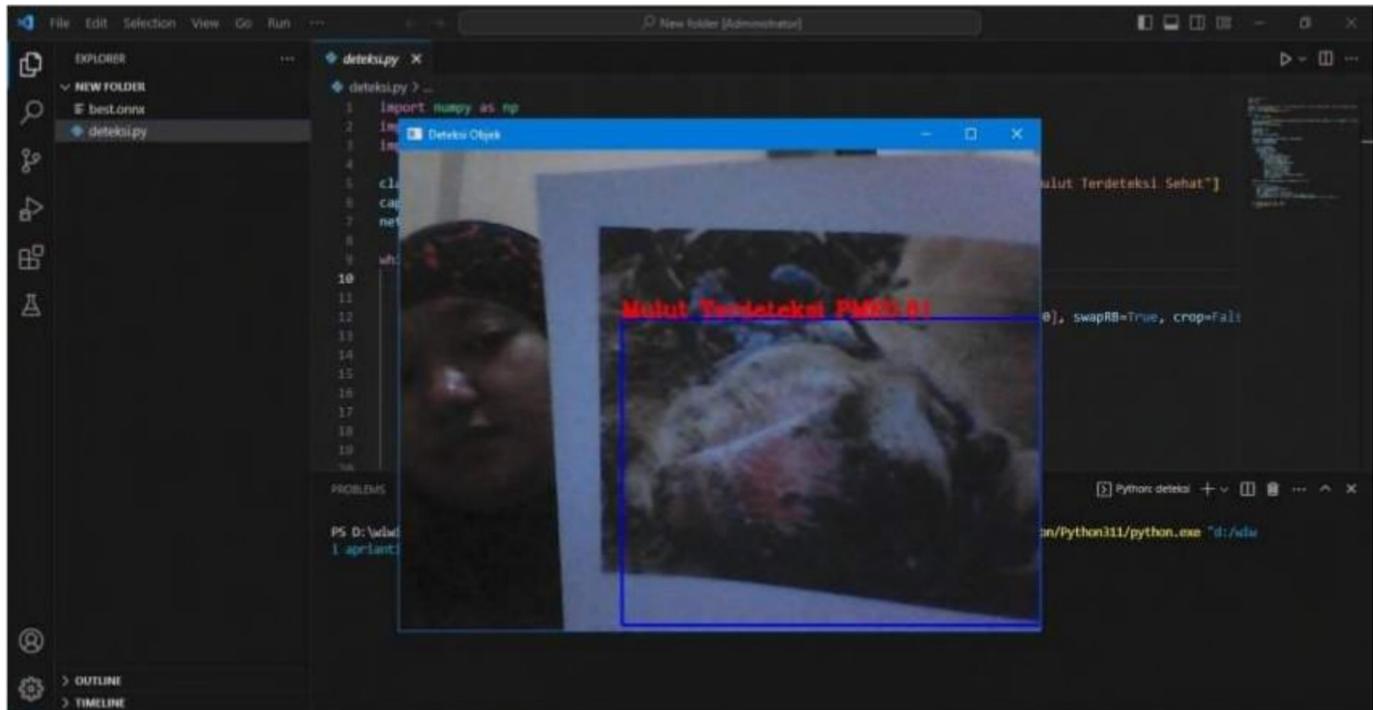
Mendapatkan koordinat dan label dari daftar kelas berdasarkan indeks tersebut.

Menggambar persegi panjang pada gambar asli dengan fungsi `cv2.rectangle()`.

Menulis teks label serta tingkat kepercayaannya di atas persegi panjang menggunakan fungsi `cv2.putText()`.

`cv2.imshow("Deteksi Objek", img)`: Ini menampilkan frame video beserta objek yang terdeteksi dalam jendela pop-up bernama "Deteksi Objek".

Terakhir, jika tombol ESC ditekan (`if cv2.waitKey(1) & 0xff == 27`), maka program akan berhenti.



Gambar 35 proses deteksi real time camera open cv

### G. Proses Uji Coba Deteksi

Pada tahap ini penulis melakukan uji coba deteksi dari uji coba sample gambar sapi terdeteksi PMK ,sapi terdeteksi sehat atau tidak terjangkit PMK dan juga uji coba deteksi objek lain yang tidak termasuk dalam penyakit pmk dan sehat.

Tabel 6 uji coba deteksi benda lain

NO	Objek Benda Lain	Akurasi
1.	PuJpen	0,26
2.	Mose	0,27
3.	Botol	0,25
4.	Pembungkus makanan	0,23
5.	Gelas	0,24
6.	Toples	0,21
7.	Tas	0,30
8.	Bantal	0,20
9.	Buku kecil	0,28
10.	Buku	0,34

Tabel diatas menjelaskan basil deteksi yang dilakukan dengan benda lain ,sehingga basil yang didapatkan yaitu akurasi nilai tertinggi di dapatkan dalam mendeteksi buku dengan akurasi 0,34 dan nilai terendab didapatkan di akurasi 0,20 yang mendeteksi bantal.

Tabel 7 uji coba objek pmk

NO	Objekpmk	Akurasi
1.	Mulutpmk	0,50
2.	Mulutpmk	0,52
3.	Mulut pmk	0,40
4.	Mulut pmk	0,45
5.	Mulut pmk	0,60
6.	Kukupmk	0,50
7.	Kukupmk	0,46
8.	Kukupmk	0,47
9.	Kukupmk	0,40
10.	Kukupmk	0,45

Tabel diatas menjelaskan basil deteksi yang dilakukan pada sample gambar penyakit yang terdeteksi PMK ,nilai tertinggi dalam deteksi mendapatkan akurasi 0,60 pada objek Mulut PMK dan terendah didapatkan di kuku PMK dengan akurasi 0,40.

Tabel 8 uji coba objek sehat

NO	Objek sehat	Akurasi
1.	Mulut sehat	0,60
2.	Mulut sehat	0,70
3.	Mulut sehat	0,80
4.	Mulut sehat	0,60
5.	Mulut sehat	0,55
6.	Kuku sehat	0,45

7.	Kuku sehat	0,50
8.	Kuku sehat	0,51
9.	Kuku sehat	0,40
10.	Kuku sehat	0,54

Tabel diatas menjelaskan basil deteksi gambar pada sapi terdeteksi sehat hasil yang dapatkan yaitu nilai tertinggi akurasi berada di mulut sehat dengan 0,80 dan nilai akurasi terendah didapatkan di kuku sehat dengan akurasi 0,40

Kesimpulan dari hasil uji coba yang di lakukan

1. Hasil uji coba dari benda lain memiliki akurasi tertinggi yang didapatkan yaitu 0,34 yang mendeteksi sebuah objek buku sedangkan nilai akurasi terendah memiliki 0,20 yang mendeteksi objek bantal.
2. Hasil uji dari sampel gambar PMK sapi memiliki nilai tertinggi akurasinya 0,60 yang mendekteksi sampel gambar mulut sapi terdeteksi PMK dan nilai akurasi terendah memiliki 0,40 yang mendeteksi kuku yang terdeksi PMK
3. Hasil uji dari sampel gambar sapi terdeteksi sehat yang memiliki akurasi tertinggi yaitu 0,80 yang mendeteksi muJu sapi sehat dan akurasi terendah yaitu 0,40 yang mendeteksi sampel kuku terdeteksi sehat.

## BABV

### KESIMPULAN DAN SARAN

#### A. Kesimpulan

Berdasarkan penelitian yang dilakukan dapat kita simpulkan sebagai berikut:

1. Dari hasil proses pelabelan dataset didapatkan data training 75% dengan jumlah 82 K images, validation set 14% 1.5 K images dan testing set 12% images 451 K .
2. Hasil pengujian dataset dengan 100 kali training mendapatkan gambaran confusional matrix yaitu kinerja sistem klasifikasi pada dataset uji, mulut terdeteksi PMK memiliki 0,74 ,kuku terdeteksi PMK 0,52 ,mulut terdeteksi sehat 0,94 ,dan kuku terdeteksi sehat 0,64.
3. Hasil percobaan dilakukan pada sample testing gambar penyakit mulut dan kuku memiliki akurasi tertinggi yang didapatkan dari hasil percobaan yaitu 0,60 mulut 0,50 kuku dan sehat memiliki akurasi ,mulut sehat memiliki 0,80 dan kuku 0,54.dan Hasil percobaan dengan benda lain mendapatkan akurasi 0,34 dan 0,30

#### B. Saran

Berdasarkan penelitian yang dilakukan ada beberapa saran penulis yang penulis berikan sebagai upaya perbaikan untuk penelitian selanjutnya yaitu sebagai berikut:

1. Menyeimbangkan jumlah dataset disetiap kelas yang dimiliki untuk memperoleh hasil training dengan akurasi yang lebih baik lagi.

## DAFTAR PUSTAKA

- Agustina, Feri, and Muhammad Sukron. "Deteksi Kematangan Buab Pepaya Menggunakan Algoritma YOLO Berbasis Android." *Jurnal Ilmiah Infokam* 18.2 (2022): 70-78.
- Arhami, Muhammad. 2005. *Konsep Dasar Sistem Pakar*. Yogyakarta: Andi
- Arwindo, Dicky Giacini, Eva Yulia Puspaningrum, and Yisti Vita Via. "Identifikasi penggunaan masker menggunakan algoritma CNN YOLOv3-Tiny." *Presiding Seminar Nasional Informatika Bela Negara*. Vol. 1. 2020
- Budiman, I., Saori, S., Anwar, RN, Fitriani, F., & Pangestu, MY (2021). Analisis Pengendalian Mutu Di Bidang Industri Makanan (Studi Kasus: Umkm Mochi Kaswari Lampion Kota Sukabumi). *Jurnal Inovasi Penelitian* , 1 (10), 2185-2190.
- Direktorat Kesehatan Hewan. (2022). *Kesiagaan Darurat Veteriner Indonesia Seri Penyakit Mulut dan Kuku (Kiat Vetindo) PMK* (3.1). Direktorat Kesehatan Hewan.
- Efford, N. (2000) *Pengolahan Citra Digital. Pengantar Praktis Menggunakan Java TM*. Pendidikan Pearson, Upper Saddle River.
- Fahrezi, Ahmad, et al. "Pengujian Black Box Testing pada Aplikasi Inventori Barang Berbasis Web di PT. AINO Indonesia." *LOGIC: Jurnal Dmu Komputer dan Pendidikan* 1.01 (2022): 1-5.
- Grubman, M. J., & Baxt, B. (2004). Foot and Mouth Disease. *Clinical Microbiology Reviews*, 17(2), 465-493. <https://doi.org/10.1128/CMR.17.2.465-493> .2004
- Hamjaya Putra Hamdu. (2019). *Laporan Surveilans Eksotik Penyakit Mulut dan Kuku (PMK) dan Bovine Spongiform Encephalopathy (BSE)*. Balai Besar Veteriner Maros

- Ichsanudin, Muhamad Nur, Muhammad Yusuf, and Suraya Suraya. "PENGUJIAN FUNGSIONAL PERANGKAT LUNAK SISTEM INFORMASI PERPUSTAKAAN DENGAN METODE BLACK BOX TESTING BAGI PEMULA." *PENYWPANAN: Jurnal Ilmiah Teknik dan Ilmu Komputer* 1.2 (2022): 1-8.
- Ilahiyah, S., & Nilogiri, A. (2018). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *JUSTINDO (Jurnal Sistem & Teknologi Informasi Indonesia. St'I)*, 50.
- Indrajani. 2011. Perancangan Basis Data dalam All in 1. Jakarta: PT. Elex Media Komputindo
- Jocber, G., Stoken, A., Borovec, J., Changyu, L., Hogan, A., Diaconu, L., ... & Rene Claramunt, E. (2020). Ultralytics/yolov5: v3.0. *Zenodo*
- Justitian, Eka Restu, Intan Yuniar Purbasari, and Fetty Tli Anggraeny. "Perbandingan Akurasi Deteksi Kelelahan pada Pengendara Menggunakan YOLOv3-Tiny YOLOv4-Tiny." *Jurnal Informatika dan Sistem Informasi* 3.1 (2022): 21-30.
- Khotimab, H., Nafiiyah, N., & Masruroh, M. (2020). Klasifikasi Kematangan Buah Mangga Berdasarkan Citra HSV dengan KNN. *Jurnal Elektronika Listrik dan Teknologi Informasi Terapan*, 1(2), 1-4.
- Kusrini, 2006. Sistem Pakar Teori dan Aplikasi. Yogyakarta: Andi
- Nurudin, M., Jayanti, W., Saputro, R. D., Saputra, M. P., & Yulianti, Y (2019). Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis. *Jurnal Informatika Universitas Pamulang*, 4(4), 143. <https://doi.org/10.32493/informatikav4i4.3841>
- Prayitna, Dimas Surya. *Deteksi penyakit daun tomat dengan Algoritma You Only Look Once (YOLO)*. Diss. UIN Sunan Gunung Djati Bandung, 2022.

- Putry, N. M. (2022). KOMPARASI ALGORITMA KNN DAN NAIVE BAYES UNTUK KLASIFIKASI DIAGNOSIS PENYAKIT DIABETES MELLITUS. *Evolusi: Jurnal Sains Dan Manajemen*, 10(1).
- Rahmatullah, Syaifur, and Dini Silvia Purnia-STMIK Nusa Mandiri. "Sistem Pakar Untuk Mendiagnosa Penyakit Mata Dengan Metode Forward Chaining." *SPEED-Sentra Penelitian Engineering dan Edukasi* 10.2 (2018).
- Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- Ria, SN, Walid, M., & Umam, BA (2022). Pengolahan Citra Digital Untuk Identifikasi Jenis Penyakit Kulit Menggunakan Metode Convolutional Neural Network (CNN). *Energy-Jurnal Ilmiah Ilmu-Ilmu Teknik*, 12 (2), 62-67.
- Pusat Data dan Analisa Tempo. (2020). *Indonesia dan Penanganan Penyakit Kuku dan Mulut*. Tempo Publishing.
- Sanjaya, J., & Ayub, M. (2020). Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup. *Jurnal Teknik Informatika dan Sistem Informasi*, 6(2).
- Shapiro, Linda G., and G. Stockman. "Computer vision prentice hall." *Inc., New Jersey* (2001) ..
- Situngkir, T. Y. P. (2022). KLASIFIKASIPENYAKIT PADADAUN KENTANG MENGGUNAKAN PENGOLAHAN CITRA DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) (Doctoral dissertation, UPN'Veteran" Yogyakarta).
- Sulistyanto, H., & SN, A. (2014). Urgensi Pengujian pada Kemajemukan Perangkat Lunak dalam Multi Perspektif. *KomuniTi*, 6(1), 65-74.

- Sutawi. (2022, May 24). *Bioterrorisme Penyakit Mulut dan Kuku* (PMK). <https://www.agropustaka.id/pemikiran/bioterrorisme-penyakit-mulut-dan-kuku-pmk/>.
- Tbuan, Do. "Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detection algorithm." (2021)
- Wibisono, W., & Baskoro, F. (2002). Pengujian Perangkat Lunak Dengan Menggunakan Model Behaviour UML Waskitbo Wibisono, Fajar Baskoro. *Juti*, 1(1), 43-50
- Yanuar, Aditya. "YOLO (You Only Look Once)." *di http.r/machinelearning.mipa.ugm.ac.id/2018/08/051)0[0-yoi,-on,/y-lookoncel( diakses Januari 2019)* (2018).
- Zamroni, Moh Rosidi, dan Agung Wahyudi. "Sistem Pakar Diagnosa Penyakit Sapi Sebagai Upaya Pencegahan Penyebaran Wabah PMK Di Larnongan." *JURNAL ILMIAH INFORMATIKA* 10.02 (2022): 145-152.
- Zebua, Helviani. *PEMBUATAN MODEL DETEKSI GEJALA AWAL PENYAKIT MULUT DAN KUKU PADA SAPI BERBASIS CITRA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK*. Diss. Sistem Informasi, 2023.