

BAB II

LANDASAN TEORI

2.1 Tinjauan terhadap penelitian terkait.

Berikut beberapa penelitian terkait yang menjadi referensi pada penelitian ini:

Berdasarkan hasil riset, penelitian tentang efisiensi pada peternakan sapi ini sudah banyak dilakukan, namun masih banyak yang menggunakan teknologi peternakan pintar hanya untuk memantau kondisi disebuah peternakan dengan sedikit control tetapi masih kurang efisien. Pada salah satu jurnal yang berjudul “*Smart Cow Health Monitoring, Farm Environmental Monitoring and Control Using Wireless Sensor Networks*” membuat sebuah sistem yang berfungsi untuk memonitoring kondisi ternak secara *real-time* agar bias memantau kondisi sapi secara langsung dengan komponen seperti kamera, sensor kelembapan untuk mengukur kelembapan lingkungan pada area peternakan. Sedangkan untuk pengontrolan ada semprot air yang bias dikendalikan oleh peternak secara langsung untuk meningkatkan kelembapan area peternakan dengan cara menyemprotkan butiran air pada area peternakan.

Riset literatur yang didapatkan, yaitu “*Smart Diary Farming Through the Internet of Things*” membuat sebuah sistem yang cukup rumit dibandingkan dengan literatur yang hanya memantau beberapa komponen pada lingkungan seperti suhu dan kelembapan pada area peternakan. Pada literatur ini,

menggunakan beberapa komponen yang lebih banyak untuk memantau kondisi pada area peternakan, seperti “Digital Tracking for Each Now, Genomic testing: Herd Inteligent, A Digitally Signalled Birth, Sensor Driven Crop Management”, sehingga bias membuat nilai efisiensi pada peternakan menjadi meningkat dikarenakan banyaknya komponen pemantau pada peternakan. Perbedaan antara penelitian ini dengan yang sudah ada berdasarkan hasil pencarian beberapa referensi yang didapat, yaitu terletak pada fitur klasifikasi sapi yang memiliki ciri-ciri terkena penyakit mata pink-eye. Pada referensi ini hanya mempunyai beberapa kemampuan monitoring seperti suhu, kelembapan dan pendeteksi lokasi pada binatang ternak, sedangkan pada penelitian ini, membuat sebuah rancangan agar dapat mengklasifikasi sapi yang mempunyai ciri-ciri terkena penyakit mata pink-eye.

(Silvi Ratna, 2020) dalam penelitiannya berjudul, “Sistem Monitoring Kesehatan berbasis Internet of Things (IoT)”. Internet of Things merupakan jaringan komunikasi yang dimana alat-alat dan sensor saling berhubungan satu sama lain atau dengan sistem yang lebih besar. Jaringan IoT mengumpulkan miliaran data dari perangkat yang berbeda setiap yang digunakan dalam kehidupan sehari-hari. Saat ini, ada sekitar 20 miliar perangkat di dunia yang berinteraksi satu sama lain, dan pada tahun 2025 diperkirakan akan mencapai 75 miliar perangkat.

Aplikasi IoT di sektor kesehatan meningkat dari hari ke hari, orang yang tinggal di daerah pedesaan tidak dapat mengambil manfaat dari layanan kesehatan preventif karena kurangnya infrastruktur. Akibatnya, kematian terjadi sangat dini

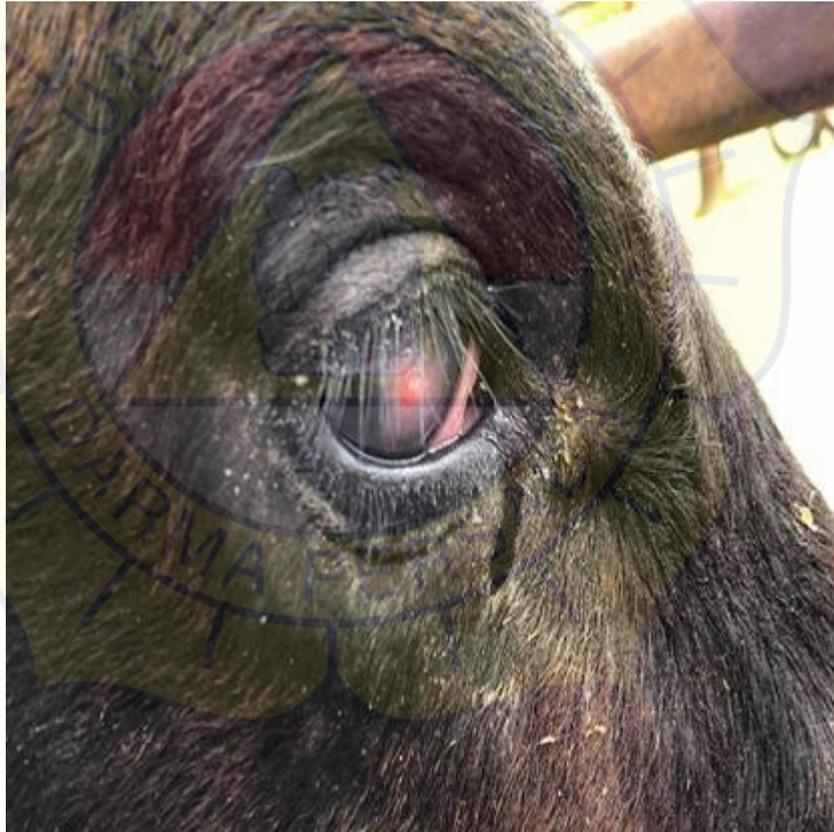
di wilayah yang jauh dari puskesmas atau rumah sakit. Selain itu dengan penuaan populasi dunia cepat, kebutuhan para lansia untuk bantuan kehidupan semakin meningkat. Perangkat yang menyediakan pemantauan secara terus menerus terhadap pasien ini relatif mahal dan membutuhkan tenaga terlatih untuk menggunakannya. Untuk membantu pasien tersebut untuk ditindaklanjuti secara terus menerus dengan alat kesehatan yang dapat dipakai untuk menjaga kesehatan mereka, alat yang dipakai ini secara terus menerus untuk mengukur detak jantung telah terjadi, maka akan mengirimkan informasi tentang kondisi kesehatan pasien kepada anggota keluarga dan dokter. HR dan HRV digunakan terutama sebagai alat diagnostik untuk penyakit jantung dan non-jantung seperti gagal jantung, penuaan, penyakit parkinson, diabetes.

IoT adalah realistik baru yang sepenuhnya mengubah kehidupan kita sehari-hari. Ini juga merupakan cara untuk merevolusi perawatan kesehatan modern dengan memberikan perawatan yang lebih personal dan preventif. Berkat teknologi IoT, informasi diantara berbagai perangkat pintar dapat dihubungkan dari mana saja dengan media internet sebagai media koneksinya. Dalam teknologi IoT, dengan layanan kesehatan pintar dapat memberikan diagnosis penyakit jarak jauh, dengan biaya murah, konsumsi daya rendah, dan memiliki kinerja tinggi, perangkat yang dapat mengumpulkan data jantung pasien untuk dapat dikirim ke pihak keluarga pasien atau dokter. Pemantauan kesehatan seseorang secara realtime melalui perangkat IoT dapat dipakai dengan banyak perangkat sensor-sensor alat kesehatan lainnya.

2.2 Tinjauan Teori

2.2.1 Penyakit Mata *pink-eye* pada Sapi

Pink-eye merupakan penyakit mata akut yang menular pada hewan ternak seperti sapi maupun kambing. Infeksi ini bersifat epizootic yang ditandai dengan memerahnya conjunctiva dan kekeruhan mata. Penyakit ini tidak sampai menyebabkan kematian, akan tetapi dapat menyebabkan kerugian yang cukup besar bagi peternak, karena akan menyebabkan kebutaan, penurunan berat badan dan biayapengobatan yang mahal.



Gambar 2.1 penyakit mata *pink-eye* pada sapi

Penyakit *pink-eye* ini disebabkan oleh sebuah virus, bakteri, chlamydia maupun rikketsia, namun penyebab penyakit ini biasanya

diakibatkan oleh bakteri *Moraxella bovis*. Penularan penyakit ini dapat terjadi melalui :

1. Kontak antara ternak peka dengan ternak penderita.
2. Serangga yang bias memindahkan mikroorganismenya.
3. Iritasi debu.
4. Sumber-sumber lain yang dapat menyebabkan goresan atau luka mata.

Gejala klinis pada penyakit ini seperti :

1. Mata berair, kemerahan pada bagian mata yang putih dan kelopaknya.
2. Bengkak pada kelopak mata.
3. Menjulingkan mata untuk menghindari sinar matahari.
4. Selaput bening mata/kornea menjadi keruh.
5. Pembuluh darah tampak menyilangnya.
6. Terjadi borok atau lubang pada selaput mata.
7. Borok dapat pecah dan mengakibatkan kebutaan.
8. Sembuh dalam waktu 1-4 minggu.

2.2.2 Machine Learning

Machine learning merupakan wujud kecerdasan buatan yang memungkinkan system untuk belajar melalui data tanpa melalui pemrograman secara eksplisit (Hurwitz, et al.,2018). Machine learning menggunakan algoritma yang secara berulang dari data untuk mendeskripsikan data, meningkatkan, dan memprediksi hasil. Prinsip ini menirukan cara belajar manusia yang mana manusia belajar melalui informasi yang didapat dari buku, cerita, gambar, dan sebagainya. Oleh

karena itu, selama algoritma mempelajari data pelatihan untuk menghasilkan model yang tepat.

Dalam Machine learning, model berupa keluaran yang dihasilkan melalui pelatihan pada algoritma machine learning berdasarkan data. Setelah melakukan pelatihan, lalu diberikan sebuah model dengan masukan data, maka akan menghasilkan keluaran dari hasil pembelajaran. Sebagai contoh algoritma prediktif akan membuat model prediksi. Sehingga saat diberikan data pada model prediksi, selanjutnya model menghasilkan prediksi berdasarkan data yang dilatih.

Pada tahun 1998., Tom Mitchell memaparkan definisi Machine learning dengan lebih formal dan mudah dimengerti.

“A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E” (Mitchell, 1997).

Machine learning bertujuan agar computer belajar, berfikir, dan mengambil keputusan. Sehingga computer dapat belajar melalui data, maka konsep ini memberitahu computer bagaimana caranya belajar.

2.2.3 Jenis-Jenis Algoritma Machine Learning

Berdasarkan jenisnya, algoritma machine learning dikelompokkan kedalam 3 kategori, diantaranya :

2.2.3.1 Supervised Learning

Supervised Learning merupakan kumpulan dataset yang memiliki nilai input dan output yang saling berpasangan. Data ini memiliki fitur berlabel yang mendefinisikan arti dari data. Supervised learning dimaksudkan untuk menemukan pola dalam data yang telah diberi label melalui proses training sehingga komputer akan belajar memahami dataset agar menghasilkan model yang diharapkan.

2.2.3.2 Unsupervised Learning

Unsupervised learning merupakan algoritma machine learning yang datasetnya hanya menerima input variable, tanpa harus memiliki output variable. Unsupervised learning paling cocok pada masalah pada dataset yang tidak memiliki label. Sehingga dalam memodelkan data, algoritma ini mengklasifikasikan data berdasarkan pola atau cluster yang ditemukannya. Unsupervised learning biasa digunakan pada masalah yang berkaitan dengan association dan clustering. Pada kasus nyata, unsupervised learning digunakan untuk mendeteksi spam email. Terdapat terlalu banyak variable dalam email yang benar dan spam. Sehingga, machine learning mengklasifikasi berdasarkan kelompok dan kaitannya untuk mengidentifikasi email yang tidak diinginkan.

2.2.3.3 Reinforcement Learning

Reinforcement learning bertujuan untuk membuat computer dapat belajar sendiri melalui lingkungan (Purnama,2019). Computer akan melakukan pembelajaran sendiri melalui interaksi yang dilakukan dengan lingkungannya. Selama proses training, computer akan melakukan kegiatan trial and error. Sehingga ketika kegiatan trial and error dilakukan computer akan menerima feedback. Feedback yang didapatkan dari aksi akan digunakan sebagai panduan sekaligus peta untuk melakukan aksi selanjutnya. Setiap mencoba setiap opsi dan kemungkinan yang ada, algoritma ini akan melakukan pengamatan dan evaluasi pencapaian, proses ini dilakukan secara berulang-ulang sampai memenuhi keputusan yang akurat. Saat ini, reinforcement learning banyak dimanfaatkan pada berbagai mesin permainan dan robot.

2.2.4 Deep Learning

Pada mulanya deep learning diperkenalkan oleh Geoffrey Hinton pada tahun 2006 yang merupakan salah satu varian neural network yang disebut Deep Beliefs Nets. Deep Beliefs Nets memperkenalkan bagaimana melakukan training model neural network dengan dua buah layer yang ditambahkan satu layer di atasnya, lalu training hanya dilakukan pada layer teratas dan begitu seterusnya. Dengan melakukan teknik ini maka penulis dapat melakukan training model neural network dengan layer yang lebih banyak dari model-model sebelumnya. Ide tersebut memunculkan istilah

deep learning untuk membedakan antara arsitektur neural network konvensional (single layer) dan neural network multilayer (Purnama, 2019). Deep learning merupakan bagian dari algoritma machine learning yang dikembangkan dari neural network multiple layer yang merepresentasi secara otomatis dari data seperti di gambar, teks atau video tanpa aturan kode yang diperkenalkan atau pengetahuan domain manusia.

2.2.5 Deteksi Objek

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. Yang kedua menganggap deteksi objek sebagai masalah regresi atau klasifikasi, mengadopsi kerangka kerja terpadu untuk mencapai hasil akhir (kategori dan lokasi) secara langsung. (Zhao, et al., 2019).

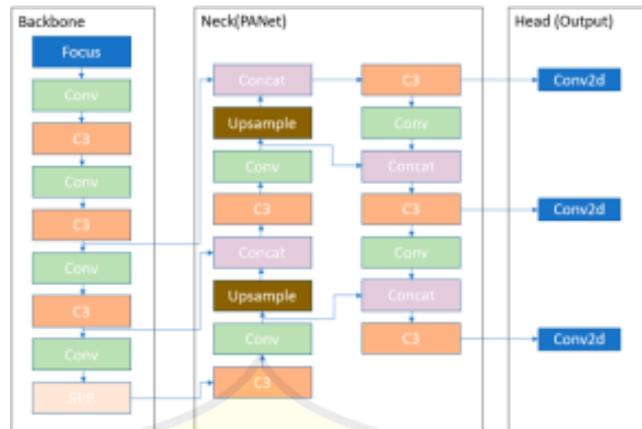
2.2.6 Algoritma *You Only Look Once* (YOLO)

Algoritma YOLO (*You Only Look Once*) merupakan algoritma *deep learning* untuk deteksi objek secara real-time berbasis *Convolutional Neural Network* (CNN) yang menggunakan pendekatan yang berbeda dengan algoritma lain, yaitu menerapkan sebuah jaringan syaraf tunggal (*Single Neural Network*) pada keseluruhan citra. Jaringan ini menggunakan fitur

menghasilkan kecepatan yang mengesankan dan dengan demikian merupakan keseimbangan yang baik antara kecepatan dan akurasi.

2.2.7 Arsitektur Algoritma YOLOv4

YOLOv4 berbeda dari rilis sebelumnya. YOLOv4 menggunakan PyTorch sebagai gantinya dari Darknet. YOLOv4 menggunakan CSPDarknet53 sebagai backbone. Backbone ini memecahkan pengulangan informasi gradien di backbone besar dan mengintegrasikan perubahan gradien ke dalam peta fitur yang mengurangi kecepatan inferensi, meningkatkan akurasi, dan mengurangi ukuran model dengan mengurangi parameter. YOLOv4 menggunakan Path Aggregation Network (PANet) sebagai neck untuk meningkatkan arus informasi. PANet mengadopsi Feature Pyramid Network baru (FPN) yang mencakup beberapa lapisan bottom up dan top down. Hal ini dapat meningkatkan propagasi level rendah fitur dalam model. PANet meningkatkan lokalisasi di lapisan bawah, yang meningkatkan akurasi lokalisasi objek. Selain itu, head di YOLOv4 sama dengan YOLOv3 yang menghasilkan tiga keluaran peta fitur yang berbeda untuk mencapai skala multi prediction. Hal ini juga membantu meningkatkan prediksi objek kecil hingga besar secara efisien dalam model. Gambar diumpankan ke CSPDarknet53 untuk ekstraksi fitur dan diumpankan lagi ke PANet untuk fusi fitur. Akhirnya, lapisan YOLO mengeluarkan hasil. Gambar dari arsitektur YOLO v4 ditampilkan sebagai berikut.

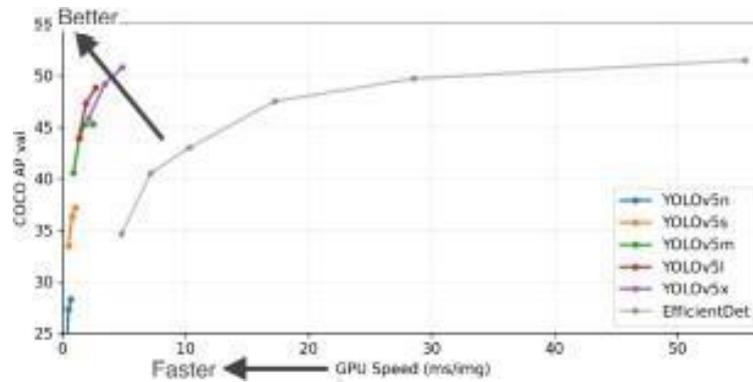


Gambar 2.3 Arsitektur YOLOv4

Conv menunjukkan lapisan konvolusi. C3 terdiri dari tiga lapisan konvolusi dan modul yang mengalir oleh berbagai hambatan. Spatial Pyramid Pooling (SPP) adalah lapisan pooling yang digunakan untuk menghilangkan batasan ukuran tetap pada jaringan. Upsample digunakan dalam upsampling fusi lapisan sebelumnya di node terdekat. Concat adalah lapisan pengiris dan digunakan untuk mengiris layer sebelumnya. Tiga Conv2d terakhir adalah modul deteksi objek yang digunakan di head dari jaringan.

2.2.8 Tipe Algoritma YOLOv4

YOLOv4 memiliki beberapa tipe dengan masing-masing memiliki perbedaan dari segi kecepatan deteksi dan performa mAPnya. Berikut gambar dari perbedaan dari masing- masing tipe YOLOv4.



Gambar 2.4 Performa tipe YOLOv4

Tabel 2.1 Performa tipe model YOLOv4

Model	size (pixels)	mAPval 0.5:0.95	mAPval 0.5	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv4n	640	28.4	46.0	45	6.3	0.6	1.9	4.5
YOLOv4s	640	37.2	56.0	98	6.4	0.9	7.2	16.5
YOLOv4m	640	45.2	63.9	224	8.2	1.7	21.2	49.0
YOLOv4l	640	48.8	67.2	430	10.1	2.7	46.5	109.1
YOLOv4x	640	50.7	68.9	766	12.1	4.8	86.7	205.7

Pada **Gambar 2.4** digambarkan bahwa semakin keatas arah grafik model maka akan semakin baik dalam mAP model tetapi semakin ke kiri maka semakin cepat untuk deteksi. Pada **Tabel 2.1** ditampilkan beberapa indikator yang bisa menjadi pertimbangan dalam memilih model. Tipe Model YOLOv4 dilatih dengan data COCO dengan size 640 piksel. Dengan melihat dari Speed atau kecepatannya, akan semakin besar bila params juga semakin besar. Params merupakan parameter training. Oleh karena itu komputasi

akan lebih lama jika parameter semakin besar dan deteksi akan semakin lambat.

2.2.9 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 diibatasi; 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 :

1. Linux/Unix.
2. Windows.
3. Mac OS.
4. Java Virtual Machine.
5. OS/2.
6. Amiga.
7. Symbian.

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

1. Django, web framework.
2. Scipy dan Scikit, pusaka untuk membuat aplikasi machine learning dan kecerdasan buatan.
3. Tornado, pustaka untuk membuat aplikasi we, websocket, dan *asynchronousprogramming*.
4. Celery, pustaka untuk membuat *asynchronous task*.
5. Matplotlib, pustaka untuk membuat grafik keperluan *saintific*.
6. bioPython, pustaka untuk menganalisa DNA dan *Genome* makhluk hidup.
7. TensorFlow, pustaka untuk membuat aplikasi yang ditenagai oleh *deep learning*.
8. 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.

2.2.10 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 menanggapi 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 :

1. Membaca dan menulis gambar.
2. Mendeteksi wajah dan fitur-fiturnya
3. Mendeteksi bentuk seperti lingkaran, persegi, segitiga dan lain-lain didalam gambar.
4. Pengenalan teks didalam gambar seperti plat motor dan sebagainya.
5. Memodifikasi warna dan kualitas gambar misalnya CamScanner.
6. Mengembangkan aplikasi *Augmented Reality*.

Beberapa keuntungan OpenCV, antara lain :

1. banyak tutorial yang tersedia sehingga mudah untuk dipelajari.
2. Bekerja di hamper semua Bahasa pemrograman, seperti :
 - a. C++
 - b. Android SDK
 - c. JAVA
 - d. Python
 - e. C
3. 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 adlah 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. & Kahler, A., 2008).

2.2.11 Streamlit

Streamlit adalah aplikasi framework open-source untuk *machine learning* dan *data science*. Pembuatan UI untuk *image captioning* sendiri cukup mudah karena streamlit memiliki library pada python sehingga hal ini membuat pembangunannya mudah dan memiliki UI yang cantik.

2.2.12 Roboflow

Roboflow merupakan sebuah platform yang dikhususkan untuk membantu AI engineer dalam mengolah 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 hasil deteksi dari model yang sudah dilatih.

Roboflow menyediakan 20 library model yang dapat menjadi pilihan model saat akan membuat projek computer vision, yaitu YOLOX, Vision Transformer, YOLOv5, YOLOR, OpenAI Clip, Scaled-YOLOv4, Resnet32, EfficientDet-D0-D7, YOLOv4-tiny, YOLOv4 Darknet, EfficientNet, Detectron2, EfficientDet, Faster R-CNN, YOLO v3 PyTorch, YOLO v3 Keras, MobileNetSSDv2, YOLOv4, PyTorch, MobileNetV2 Classification, dan ResNet32.

2.2.14 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.

2.2.15 Parameter Evaluasi *Object Detection*

Dari berbagai kumpulan data beranotasi yang digunakan dalam deteksi objek, *Average Precision* (AP) adalah metrik yang paling umum digunakan untuk mengukur akurasi deteksi. Sebelum mempertimbangkan variasi AP, perlu dilihat beberapa konsep umum seperti yang didefinisikan dibawah ini.

1. *True Positive* (TP): Kebenaran deteksi dari *ground-truth bounding box*.
2. *False Positive* (FP): Kesalahan deteksi dari objek yang tidak ada atau kesalahan tempat deteksi dari objek yang ada.
3. *False Negative* (FN): *Ground-truth bounding box* yang tidak terdeteksi.

Penting untuk dicatat bahwa hasil *True Negative* (TN) tidak berlaku dalam konteks deteksi objek, karena ada banyak kotak pembatas yang tidak boleh dideteksi pada gambar tertentu. Untuk deteksi objek, IOU mengukur area tumpang tindih antara kotak pembatas yang diprediksi dan kotak pembatas *ground-truth* dibagi dengan luas unit di antara keduanya.

Untuk mengklasifikasikan deteksi sebagai benar atau salah dapat dengan cara membandingkan IOU dengan ambang batas tertentu t . Untuk $IOU > t$, deteksi dianggap benar. Untuk $IOU < t$ maka deteksi dianggap salah. Sebaliknya, evaluasi metode deteksi objek paling sering didasarkan pada istilah presisi P dan *recall* R .

Presisi adalah fungsi model yang hanya mengidentifikasi objek terkait. Presisi merupakan persentase prediksi positif yang benar. Recall adalah kemampuan model yang menemukan semua kasus yang relevan (semua kotak pembatas *ground-truth*). Ini adalah persentase prediksi positif yang benar dalam semua kebenaran dasar yang diberikan.

Mean Average Precision (mAP) adalah *matrix* yang digunakan untuk mengukur akurasi deteksi objek di semua kelas dalam basis data tertentu. mAP hanyalah AP rata-rata di semua kelas yang dapat diformulakan.

Confusion Matrix merupakan sebuah table untuk mengukur permasalahan klasifikasi pada machine learning yang menampilkan jumlah prediksi benar dan salah seperti yang telah dijelaskan sebelumnya.

2.2.16 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 dibagi menjadi 3, yaitu *train set*, *validation set*, dan *test set*.

- *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.
- *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.

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

2.2.17 Augmentasi Data

Augmentasi data dapat dilakukan secara *online* selama pelatihan dan *offline* sebagai langkah pra-pemrosesan. Augmentasi data *online* atau real-time dilakukan selama pelatihan jaringan, dimana gambar yang telah diaugmentasi akan dan disintesis setelah melewati lapisan pertama jaringan. Augmentasi *online* akan lebih sedikit pengerjaannya karena tidak ada Langkah pra-pemrosesan data. Manfaat lainnya adalah model dapat secara acak menampilkan gambar yang dihasilkan hanya sekali dan disimpan sementara di cache GPU, sehingga menghemat ruang penyimpanan. Augmentasi data *offline* dilakukan dengan mengubah data di luar jaringan dengan algoritma pemrosesan data, mirip dengan fase *preprocessing*. Menerapkan metode *offline* memakan waktu dan membutuhkan ruang penyimpanan tambahan seiring dengan peningkatan komputasi, karena gambar harus dibagi menjadi beberapa kelompok dan kemudian diproses terlebih dahulu. Tetapi metode *offline* juga memberikan manfaat yaitu dapat meningkatkan reproduksibilitas model karena memiliki salinan bagaimana setiap gambar ditambah. Misalnya, mungkin model berperforma lebih baik pada gambar terang daripada gambar gelap, jadi perlu adanya pengumpulan lebih banyak data pelatihan cahaya rendah. Selain itu, augmentasi secara *offline* akan membuat waktu pelatihan model lebih cepat karena tidak adanya proses augmentasi saat pelatihan dan akan

hemat biaya untuk penggunaan GPU.

Beberapa macam augmentasi data yang bisa digunakan untuk ekspansi data adalah sebagai berikut.

- *Flip*: Membalik gambar secara acak dengan arah vertikal atau horizontal.
- *Rotate*: Memutar gambar secara acak searah jarum jam atau berlawanan arah jarum jam hingga jumlah derajat yang dikonfigurasi tepat.
- *Crop*: Memotong gambar pada bagian tertentu.
- *Shear*: Mendistorsi gambar melintasi sumbu horizontal atau vertikal.
- *Exposure*: Menyesuaikan eksposur gambar menjadi lebih terang atau lebih gelap.
- *Blur*: Menggunakan Gaussian Blur pada gambar.
- *Noise*: Memberikan derau secara acak pada gambar.

2.2.18 Underfitting and Overfitting

Underfitting dan *overfitting* adalah dua masalah utama yang terjadi selama fase pelatihan. Masalah *underfitting* berarti NN pelatihan tidak memadai dan akurasi pembelajaran rendah. Masalah *overfitting*, disisi lain, berarti bahwa tugas pelatihan bekerja dengan baik hanya untuk data pelatihan dan bukan untuk data pengujian. Hal ini mengurangi fleksibilitas dan kemampuan beradaptasi model. Fenomena *underfitting* dan *overfitting* biasa terjadi dalam pembelajaran mesin. Sejauh ini, pandangan utama adalah bahwa *underfitting* dan *overfitting* dikaitkan dengan kompleksitas arsitektur NN, jumlah dan jenis sampel data yang digunakan dalam set pelatihan, dan jumlah *epoch* pelatihan.

2.2.19 Epoch dan Batch size

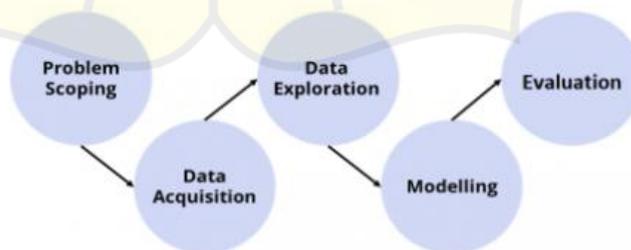
Ukuran *batch* adalah *hyperparameter* yang menentukan jumlah sampel untuk diproses sebelum parameter model internal diperbarui. Sebuah *batch* membuat prediksi dengan menganggapnya sebagai iterasi dari *for-loop* untuk satu atau lebih sampel. Pada akhir *batch*, variabel output yang diharapkan dibandingkan dan kesalahan dihitung. Dari kesalahan ini, digunakan algoritma pembaharuan untuk meningkatkan model. Dataset pelatihan dapat dibagi menjadi satu atau lebih *batch*. Ketika semua sampel pelatihan digunakan untuk membuat satu *batch*, algoritma pembelajaran disebut *batch gradient descent*. Ketika *batch* adalah ukuran satu sampel, algoritma pembelajaran disebut *stochastic*

gradient descent. Ketika ukuran batch lebih dari satu sampel dan kurang dari ukuran dataset pelatihan, algoritma pembelajaran disebut *mini-batch gradient descent*.

Jumlah *epoch* adalah *hyperparameter* yang menentukan seberapa sering algoritma pembelajaran iterasi melalui dataset pelatihan. *Epoch* berarti bahwa setiap sampel dalam dataset pelatihan memiliki kesempatan untuk memperbarui parameter internal model. *Epoch* terdiri dari satu atau lebih *batch*. Misalnya, seperti yang disebutkan di atas, *epoch* dengan *batch* disebut algoritma pembelajaran *batch gradient descent*. Jumlah *epoch* secara tradisional sangat besar, seringkali ratusan atau ribuan, dan algoritma pelatihan dapat dijalankan sampai kesalahan model diminimalkan.

2.2.20 AI Project Life Cycle

AI Project Life Cycle merupakan tahapan atau metode kerja untuk membuat suatu proyek AI. *AI Project Life Cycle* dapat dibagi menjadi 5 Stage seperti pada gambar berikut.



Gambar 2.5 Alur *AI Project Life Cycle*

1. Problem Scoping

Dalam tahap ini dilakukan identifikasi masalah yang akan diberikan solusi menggunakan AI. Tahap ini merupakan proses dimana membuat problem 18 statement dari tema yang luas dengan cara mendefinisikan 4W yaitu *What, Who, Why, dan Where* dari masalah yang telah diidentifikasi.

2. Data Acquisition

Pada tahap ini dilakukan proses pengumpulan data. tahap ini adalah tentang memperoleh data untuk proyek yang sedang dibangun. Data dapat berupa informasi atau fakta dan statistik dikumpulkan bersama untuk referensi atau analisis. Kapan pun kita ingin proyek AI dapat memprediksi output, kita perlu melatihnya terlebih dahulu menggunakan data. Data yang diperlukan untuk melatih model agar proyek AI menjadi efisien harus relevan terhadap masalah yang telah diidentifikasi sebelumnya. Oleh karena itu perlu menentukan tipe data sesuai masalah yang telah ditentukan. Data dapat diambil dari berbagai sumber seperti website, sensor, survey, web scraping, kamera, API, dan lain-lain.

3. Data Exploration

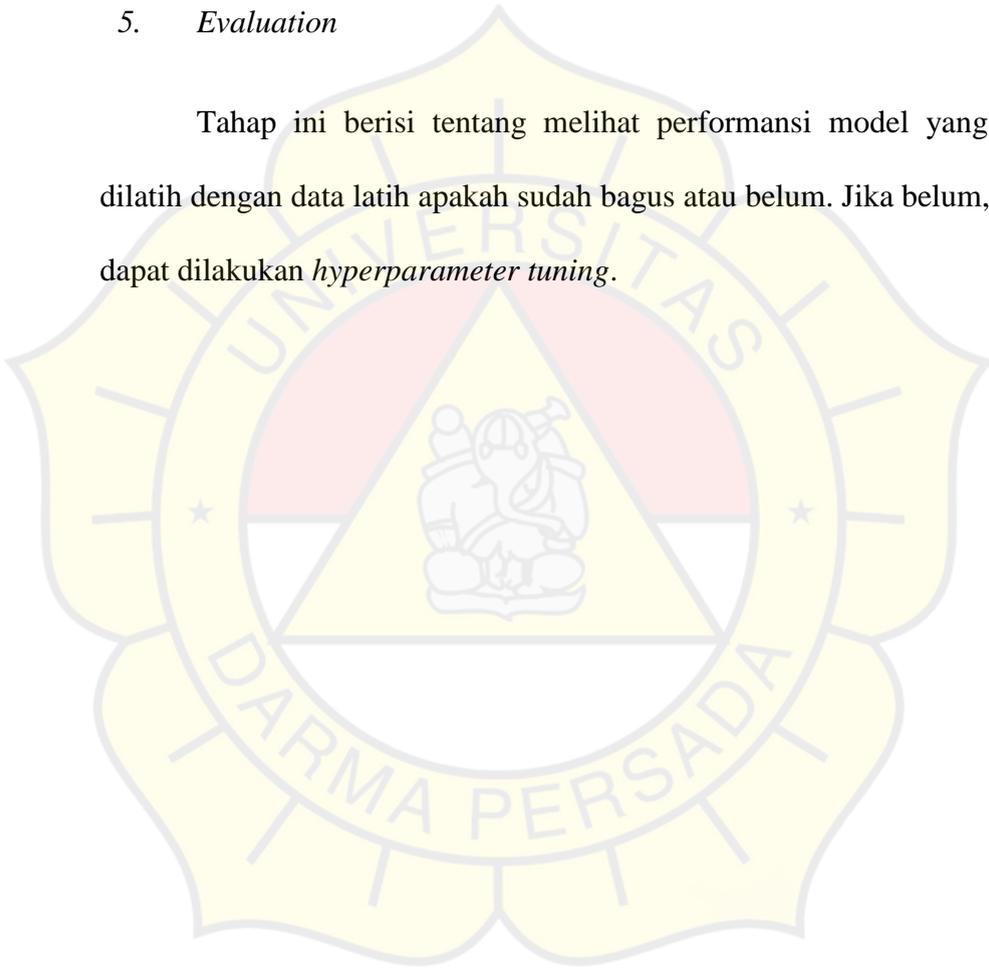
Tahap ini bertujuan untuk melakukan visualisasi data dari data-data yang sudah terkumpul untuk melihat tren atau pola pada data.

4. Modelling

Dalam membangun projek berbasis AI, diperlukan untuk bekerja dengan algoritma AI. Dalam memilih algoritma, perlu melihat pendekatan model apakah rule-based atau pendekatan *learning*. Jika telah menemukan model AI yang tepat, maka akan dilakukan pelatihan model dengan data yang sudah terkumpul sebelumnya.

5. *Evaluation*

Tahap ini berisi tentang melihat performansi model yang telah dilatih dengan data latih apakah sudah bagus atau belum. Jika belum, maka dapat dilakukan *hyperparameter tuning*.





TEKNOLOGI INFORMASI

UNIVERSITAS DARMA PERSADA