

***REAL TIME DETECTION CACAR MONYET
(MONKEYPOX) DAN PENYAKIT KULIT SERUPA
DENGAN YOLO BERBASIS MOBILE***

SKRIPSI



ANDI ILHAMSYAH IDRIS

H071191016

PROGRAM STUDI SISTEM INFORMASI

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

2023

***REAL TIME DETECTION CACAR MONYET
(MONKEYPOX) DAN PENYAKIT KULIT SERUPA
DENGAN YOLO BERBASIS MOBILE***

SKRIPSI

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

ANDI ILHAMSYAH IDRIS

H071191016

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

MAKASSAR

2023

LEMBAR PERNYATAAN KEONTETIKAN

Yang bertandatangan di bawah ini :

Nama : Andi Ilhamsyah Idris

NIM : H071191016

Program Studi : Sistem Informasi

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Real Time Detection Cacar Monyet (Monkeypox) dan Penyakit Kulit Serupa dengan YOLO Berbasis Mobile

adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alih tulisan orang lain, dan belum pernah dipublikasikan dalam bentuk apapun.

Makassar, 10 Juni 2023



Andi Ilhamsyah Idris

NIM. H071191016

Real Time Detection Cacar Monyet (Monkeypox) dan Penyakit Kulit Serupa dengan YOLO Berbasis Mobile

Disusun dan diajukan oleh

ANDI ILHAMSYAH IDRIS

H071191016



Diteujui Oleh :

Pembimbing Utama

Pembimbing Pertama

Dr.Eng. Armin Lawi, S.Si., M.Eng.
NIP. 197204231997121001

A. Muh.Amil Siddik, S.Si., M.Si.
NIP. 199110032019031015

Kepala Program Studi



Dr. Hendra, S.Si., M.Kom.
NIP. 197601022002121001

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Andi Ilhamsyah Idris

NIM : H071191016

Program Studi : Sistem Informasi

Judul Skripsi : *Real Time Detection* Cacar Monyet (*Monkeypox*) dan Penyakit Kulit Serupa dengan *YOLO* Berbasis *Mobile*

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

DEWAN PENGUJI

Tanda Tangan

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

Sekretaris : A. Muh. Amil Siddik, S.Si., M.Si., (.....)

Anggota : Dr. Andi Muhammad Anwar, S.Si.. (.....)
M.Si.

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

Ditetapkan di : Makassar

Tanggal : 10 Juni 2023



KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah swt., karena atas berkat dan penyertaan-Nya penulis dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer. Penulis menyadari bahwa penulisan tugas akhir ini tidak lepas dari bantuan, bimbingan dan dukungan dari berbagai pihak, mulai dari awal masa perkuliahan sampai pada penyusunan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih yang setulus-tulusnya kepada Ayahanda **Muhammad Idris, S.E.**, dan **Ibunda Andi Irma, S.Pd.** sebagai orang tua yang dengan segenap hati telah mendidik dan membesarkan penulis serta mendoakan dan mendukung penulis. Ucapan terima kasih juga kepada saudara-saudara kandung penulis Kakak **Andi Eka Pratiwi Idris** dan **Andi Winda Noviyasari, S.E.** yang telah memberikan dukungan fisik maupun moril.

Dengan kerendahan hati penulis mengucapkan terima kasih yang setulus-tulusnya kepada :

1. Bapak **Prof. Dr. Ir. Jamaluddin Jompa, M.Sc.**, selaku rektor Universitas Hasanuddin Makassar.
2. Bapak **Dr. Eng Amiruddin, M.Si.**, selaku dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.
3. Bapak **Dr. Hendra, S.Si., M.Kom.**, selaku Ketua Program Studi Sistem Informasi.
4. Bapak **Dr.Eng. Armin Lawi, S.Si., M.Eng.**, selaku dosen pembimbing utama yang telah membimbing penulisan skripsi mulai dari awal sampai selesai.
5. Bapak **A. Muh. Amil Siddik, S.Si., M.Si.**, selaku dosen pembimbing akademik dan sekaligus dosen pembimbing pertama yang telah membimbing penulisan skripsi mulai dari awal sampai selesai.

6. Bapak **Dr. Andi Muhammad Anwar, S.Si., M.Si.**, selaku dosen penguji pertama yang telah memberikan arahan kepada penulis selama masa perkuliahan dan memberikan masukan-masukan dalam penyusunan skripsi.
7. Bapak **Muhammad Sadno, S.Si., M.Si.**, selaku dosen penguji kedua yang telah memberikan masukan-masukan dan arahan dalam penyusunan skripsi.
8. **Dosen Departemen Matematika** khususnya **Bapak dan Ibu Dosen Program Studi Sistem Informasi, Fakultas MIPA Universitas Hasanuddin** atas semua ilmu yang sangat bermanfaat yang telah diajarkan kepada penulis selama menempuh pendidikan.
9. Keluarga besar **Andi Banri Abbas** dan **Andi Maninggau**, yang selalu memberikan dukungan dan mendoakan penulis.
10. Saudara **Khaera Ummah Armita Katli**, atas kesabaran, perhatian, dan yang senantiasa memberikan dukungan kepada penulis dan menemani penulis dalam kondisi apapun.
11. **Seluruh teman-teman Program Studi Ilmu Komputer 2019** (berubah nama menjadi **Sistem Informasi** pada awal tahun 2021), khususnya **Alif Setya, Yusuf Syam, Muhammad Takdim, Fauzi Asham, Fatwa Anugerah, Rafli Masloman, Muammar Ahlan, Bayu Ajid, Fajri Rasid, Muhammad Ikhsan, Richard Enrico, Silverius Sony** dan **Theodarryl** yang telah mendukung dan berjuang bersama selama ini.

Akhir kata, penulis berharap Tuhan Yang Maha Esa melimpahkan berkat-Nya kepada semua pihak yang telah mendukung dan membantu. Peneliti menyadari bahwa pembuatan penelitian skripsi ini masih banyak kekurangan dan kelemahannya baik dari materi maupun teknik penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan penulis untuk menambah kesempurnaan penelitian skripsi ini.

Makassar, 10 Juni 2023



Andi Ilhamsyah Idris

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

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

Nama : Andi Ilhamsyah Idris

NIM : H071191016

Program Studi : Sistem Informasi

Departemen : Matematika

Fakultas : Matematika dan Ilmu Pengetahuan Alam

Jenis Karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah berjudul :

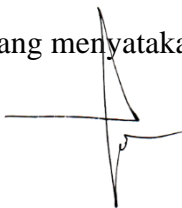
Real Time Detection Cacar Monyet (Monkeypox) dan Penyakit Kulit Serupa dengan YOLO Berbasis Mobile

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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar. Pada Tanggal 10 Juni 2023

Yang menyatakan



(Andi Ilhamsyah Idris)

ABSTRAK

Kulit memiliki peranan penting untuk menunjang hidup manusia, salah satunya sebagai indera peraba manusia. Oleh karena itu, mengingat pentingnya kulit sebagai pelindung organ tubuh, maka penting sekali untuk menjaga kesehatan kulit sejak dini. Tujuan penelitian ini adalah mendeteksi penyakit – penyakit yang ada pada kulit, seperti penyakit *monkeypox*, *chickenpox* dan *measles*. Penelitian ini menggunakan pendekatan *deep learning* dengan metode yang saat ini memiliki hasil paling signifikan dalam pengenalan citra yaitu *convolutional neural network*. Metode yang digunakan adalah *You Only Look Once (YOLO) v5* dan *v7* untuk mendeteksi obyek pada citra dan mengklasifikasikan citra menjadi beberapa *class*, seperti *chickenpox*, *monkeypox*, *measles* dan normal, serta melakukan perbandingan antara metode *YOLO v5* dan *v7*. Penelitian ini dibantu dengan layanan *Google Colab*, *framework Flask*, *Flutter* dan *Google Cloud Platform*. Hasil yang diperoleh dari penelitian ini yaitu metode yang digunakan menghasilkan kinerja akurasi dengan nilai *mean average precision (mAP)* masing – masing di atas 85%. Model *YOLOv5* memiliki *mAP* dengan rata - rata sebesar 94,5%, *precision* sebesar 98,6%, *recall* sebesar 99,9% dan *YOLOv7* memiliki *mAP* dengan rata – rata sebesar 87,9%, *precision* sebesar 98,1%, *recall* sebesar 98%. Model *YOLOv5* kemudian di-*deploy* menggunakan *framework flask* dan layanan *Google Cloud Platform* lalu diintegrasikan ke aplikasi *mobile* menggunakan *Flutter*.

Kata Kunci : *Monkeypox, Chickenpox, Measles, You Only Look Once, Flutter,*

ABSTRACT

The skin plays a crucial role in supporting human life, including its function as a sense of touch. Therefore, considering the importance of the skin as a protective organ, it is essential to maintain skin health from an early age. The purpose of this research is to detect diseases that affect the skin, such as monkeypox, chickenpox, and measles. This study employs a deep learning approach using the most significant image recognition method called Convolutional Neural Network (CNN). The You Only Look Once (YOLO) v5 and v7 methods are utilized to detect objects in images and classify them into several classes, including chickenpox, monkeypox, measles, and normal. Additionally, a comparison between the YOLO v5 and v7 methods is performed. The research is facilitated by Google Colab, Flask framework, Flutter, and Google Cloud Platform. The obtained results demonstrate the accuracy performance of the employed methods with mean average precision (mAP) values above 85%. The YOLOv5 model achieves an mAP of 94.5% on average, precision of 98.6%, and recall of 99.9%, while YOLOv7 attains an mAP of 87.9% on average, precision of 98.1%, and recall of 98%. The YOLOv5 model is then deployed using the Flask framework and Google Cloud Platform, and integrated into a mobile application using Flutter.

Keywords : *Monkeypox, Chickenpox, Measles, You Only Look Once, Flutter*

DAFTAR ISI

LEMBAR PERNYATAAN KEONTETIKAN.....	ii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS	vii
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Penelitian Relevan.....	6
2.2 Cacar Monyet (<i>Monkeypox</i>)	7
2.3 Penyakit Kulit Serupa.....	8
2.3.1 Penyakit Cacar Air (<i>Chickenpox</i>)	8
2.3.2 Penyakit Campak (<i>Measles</i>)	9
2.4 <i>Convolutional Neural Network</i> (CNN).....	9
2.4.1 <i>Convolutional Layer</i>	11
2.4.2 <i>Stride & Padding</i>	13
2.4.3 <i>Activation Layer</i>	14
2.4.4 <i>Pooling Layer</i>	15
2.3.5 <i>Fully Connected Layer</i>	16
2.5 <i>You Only Look Once</i> (YOLO).....	16
2.5.1 <i>You Only Look Once</i> (YOLOv5)	19
2.5.2 <i>You Only Look Once</i> (YOLOv7)	21
2.6 Ukuran Kinerja Model.....	23

2.6.1	<i>Confusion Matrix</i>	23
2.7	<i>Mean Average Precision (mAP)</i>	25
2.8	<i>Library</i>	25
2.8.1	<i>PyTorch</i>	25
2.8.2	<i>Matplotlib</i>	25
2.9	Rancang Bangun Aplikasi <i>Mobile</i>	26
2.9.1	<i>Framework Flutter</i>	26
2.9.2	<i>Flask</i>	26
BAB III METODE PENELITIAN.....		27
3.1	Waktu dan Tempat.....	27
3.2	Tahap Penelitian.....	27
3.2.1	Tahapan Pra-Penelitian.....	28
3.2.2	Tahapan Penelitian.....	28
3.3	Instrumen Penelitian.....	33
BAB IV HASIL DAN PEMBAHASAN.....		34
4.1	Deskripsi Data.....	34
4.2	<i>Preprocessing</i>	34
4.3	<i>Splitting Data</i>	35
4.4	Implementasi Algoritma <i>YOLO</i>	36
4.4.1	Algoritma <i>YOLOv5</i>	36
4.4.2	Algoritma <i>YOLOv7</i>	37
4.5	Evaluasi Model.....	38
4.5.1	Akurasi.....	39
4.5.2	<i>Confusion Matrix</i>	41
4.6	Model Terbaik.....	42
4.7	<i>Deployment Model</i>	44
4.8	Pengujian Aplikasi.....	48
4.8.1	Pengujian <i>Blackbox Testing</i>	49
BAB V KESIMPULAN DAN SARAN.....		51
5.1	Kesimpulan.....	51
5.2	Saran.....	51
DAFTAR PUSTAKA.....		52

DAFTAR GAMBAR

Gambar 2.2.1 Penyakit Monkeypox.....	8
Gambar 2.4.1 Arsitektur Convolutional Neural Network (CNN).....	11
Gambar 2.4.2 Contoh Bentuk Konvolusi.....	13
Gambar 2.4.3 Grafik Fungsi ReLu.....	14
Gambar 2.4.4 Ilustrasi Perhitungan pada Pooling Layer	15
Gambar 2.4.5 Ilustrasi proses fully connected layer	16
Gambar 2.5.1 Intersection Over Union	18
Gambar 2.5.2 Algoritma Deteksi YOLO	18
Gambar 2.5.3 Arsitektur YOLOv5.....	20
Gambar 2.5.4 Arsitektur YOLOv7.....	22
Gambar 3.2.1 Alur Penelitian.....	28
Gambar 3.2.2 Gambar Monkeypox di Dataset.....	29
Gambar 3.2.3 Gambar non-monkeypox di Dataset.....	30
Gambar 3.2.4 Rancangan Aplikasi.....	32
Gambar 4.4.1 File monkeypox.yaml.....	36
Gambar 4.4.2 Hyperparameter YOLOv5.....	37
Gambar 4.4.3 Citra hasil training model YOLOv5.....	37
Gambar 4.4.4 Hyperparameter YOLOv7.....	38
Gambar 4.4.5 Citra hasil dari training model YOLOv7.....	38
Gambar 4.5.1 Kurva akurasi untuk model YOLOv5	40
Gambar 4.5.2 Kurva akurasi untuk model YOLOv7	41
Gambar 4.5.3 Confusion Matrix YOLOv5	41
Gambar 4.5.4 Confusion Matrix YOLOv7	42
Gambar 4.6.1 Perbandingan algoritma object detection	43
Gambar 4.7.1 Load model YOLOv5.....	44
Gambar 4.7.2 Implementasi penulisan hasil dalam bentuk API	45
Gambar 4.7.3 Implementasi penulisan request ke API	45
Gambar 4.7.4 Contoh response request API	46
Gambar 4.7.5 Implementasi integrasi API ke dalam aplikasi	47

Gambar 4.7.6 Dependencies dalam pengembangan aplikasi.....	47
Gambar 4.7.7 Tampilan aplikasi halaman utama dan halaman kamera.....	47
Gambar 4.7.8 Hasil deteksi di aplikasi.....	48

DAFTAR TABEL

Tabel 2.5.1 Perbandingan performa YOLOv5	21
Tabel 2.5.2 Perbandingan performa YOLOv7	22
Tabel 2.6.1 Confusion Matrix Multiclass	23
Tabel 3.1.1 Timeline Kegiatan.....	27
Tabel 4.1.1 Data penyakit	34
Tabel 4.2.1 Inisiasi Hyperparameter	35
Tabel 4.3.1 Pembagian dataset.....	35
Tabel 4.5.1 Summary YOLOv5	39
Tabel 4.5.2 Summary YOLOv7	40
Tabel 4.6.1 Evaluasi akurasi pada masing – masing model.....	42
Tabel 4.8.1 Test case blackbox testing.....	49
Tabel 4.8.2 Hasil Pengujian	50

BAB I

PENDAHULUAN

1.1 Latar Belakang

Organ terluas yang menutupi seluruh permukaan tubuh manusia adalah kulit (MZ dkk., 2020). Kulit memiliki peran penting untuk menunjang hidup manusia, salah satunya sebagai indera peraba manusia. Kulit merupakan bagian yang secara langsung akan menerima ransangan seperti sentuhan, rasa sakit dan pengaruh lainnya dari luar, sehingga berbagai penyakit sering kali menyerang kulit. Oleh karena itu, mengingat pentingnya kulit sebagai pelindung organ tubuh, maka penting sekali untuk menjaga kesehatan kulit sejak dini (Ajrana, 2022). Kulit yang tidak terawat akan menimbulkan berbagai penyakit dan gangguan pada kulit.

Masalah kesehatan kulit merupakan salah satu masalah kesehatan yang sering dijumpai pada masyarakat Indonesia. Angka kejadian masalah kulit secara keseluruhan sampai saat ini belum tersedia (Zahtamal dkk., 2020). Namun demikian, angka kejadian berdasarkan jenis penyakit kulit dapat menjadi gambaran masalah kesehatan kulit, contohnya penyakit Cacar Monyet (*Monkeypox*). *Monkeypox* merupakan penyakit kulit yang kembali mewabah di pertengahan tahun 2022. Pada awal Mei 2022, kasus *monkeypox* dilaporkan di Inggris, Spanyol dan tempat lain di Eropa. Ukuran kluster wabah *monkeypox* bertambah setiap hari, seperti penyebaran geografis di seluruh Eropa dan Amerika Utara. Minggu pertama laporan awal, 24 negara melaporkan dugaan dan konfirmasi kasus virus *monkeypox* (Kraemer dkk., 2022). Penyakit *monkeypox* belum pernah ditemukan di Indonesia sejak pertama kali ditemukan pada manusia di Kongo pada tahun 1970. *Monkeypox* pertama kali ditemukan satu kasus di Indonesia pada 19 Agustus 2022 di Jakarta, kemudian naik jadi 63 orang pada 15 September 2022 (Sari & Hairunisa, 2022). Hasil data berdasarkan *World Health Organization* (WHO), pada 28 September 2022 terkonfirmasi kasus mencapai 67.539 kasus dan angka kematian mencapai 27 jiwa di 105 negara (Marisah dkk., 2022). Melihat perkembangan wabah *monkeypox* yang semakin parah, WHO menyatakan bahwa penyakit *monkeypox* merupakan

wabah darurat yang akan menjadi perhatian bagi kesehatan global dan telah memperkirakan bahwa kasus ini akan lebih banyak yang teridentifikasi.

Monkeypox adalah penyakit menular *zoonosis* yang diakibatkan oleh virus *monkeypox* yang termasuk dalam kelompok *Orthopoxvirus* (Qelina & Graharti, 2019). Virus ini awalnya ditularkan dari hewan ke manusia khususnya hewan pengerat dan primata seperti tikus, monyet atau tupai yang telah terinfeksi. Penularan virus yang terjadi dari hewan ke manusia melalui kontak langsung dengan cairan tubuh hewan yang terinfeksi dan juga memiliki kemampuan penularan dari manusia ke manusia. Penularan antar manusia melalui percikan liur yang masuk melalui mata, mulut, hidung atau luka di kulit.. *Monkeypox* bisa sembuh sendiri dalam beberapa waktu namun komplikasi dari *monkeypox* dapat menyebabkan kematian (Ludji & Buan, 2022).

Diagnosis *monkeypox* dapat dilakukan terutama dengan metode *polymerase chain reaction* (PCR) atau uji lesi kulit menggunakan mikroskop elektron (Sitaula & Shahi, 2022). Metode konfirmasi virus yang paling terpercaya adalah PCR yang juga telah digunakan untuk diagnosis *Covid-19* dalam beberapa tahun terakhir. Selain itu, teknik berbasis kecerdasan buatan (AI) dapat membantu mendeteksinya dengan bantuan pemrosesan dan analisis citra virus. Munculnya pertumbuhan model AI di berbagai *domain* seperti Gambar *rontgen* dada, analisis Gambar buah dan analisis sentiment, model AI untuk analisis Gambar medis telah diusulkan untuk berbagai jenis virus.

Metode *deep learning* yang saat ini memiliki hasil paling signifikan dalam pengenalan citra adalah *Convolutional Neural Network* (CNN). Hal tersebut dikarenakan CNN berusaha meniru sistem pengenalan citra pada visual *cortex* manusia sehingga memiliki kemampuan mengolah (Putra, 2016).

Pada penelitian yang dilakukan oleh Toan-Khoa Nguyen, Lien T. dkk., yang berjudul “*Analysis of Object Detection Models on Duckitown Robot Based on YOLOv5 Architectures*” menggunakan sebuah robot yang diberi nama “Duckietown” dan memiliki peran untuk mendeteksi sebuah objek pada sebuah video, dimana objek yang dideteksi terdapat 2 *class* yaitu *class cones* dan *class duckies*. Pelatihan model deteksi menggunakan jumlah dataset sebanyak 1000

sampel dengan pembagian 80% data train dan 20% data validasi. Model yang dilatih dalam penelitian tersebut menggunakan beberapa variasi ukuran YOLOv5 seperti tipe ukuran *small*, *medium*, *large* dan *extra large* dan fungsi optimasinya menggunakan optimasi SGD (Nguyen dkk., 2021). Pada penelitian yang dilakukan oleh Fahad Jubayer dkk., yang berjudul “*Detection of Mold on the Food Surface Using YOLOv5*” menggunakan model untuk mendeteksi objek jamur pada makanan khususnya jamur pada buah tomat dan roti, dimana model objek deteksi yang dilatih berupa *YOLOv5s* dengan dataset sebesar 850 Gambar (Jubayer dkk., 2021).

Berdasarkan penelitian – penelitian di atas, algoritma YOLO telah banyak digunakan oleh para peneliti dalam menganalisis suatu objek, sebab algoritma ini telah diklaim sebagai model terbaik dalam menyelesaikan permasalahan mengenai objek.

Maka dari itu, penulis tertarik untuk melakukan penelitian dengan menggunakan algoritma *YOLO* pada *dataset*. Sehingga peneliti memutuskan untuk membuat penelitian yang berjudul “*Real Time Object Detection Cacar Monyet (Monkeypox) dan Penyakit Kulit Serupa dengan Algoritma YOLO Berbasis Mobile*”.

1.2 Rumusan Masalah

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

1. Bagaimana membangun model *object detection* klasifikasi penyakit *monkeypox* dengan pemilihan algoritma *YOLO*?
2. Bagaimana mengevaluasi kinerja model *object detection* klasifikasi penyakit *monkeypox* dengan pemilihan algoritma *YOLO*?
3. Bagaimana men-*deploy* model *object detection* klasifikasi penyakit *monkeypox* dengan pemilihan algoritma *YOLO* ke dalam aplikasi *mobile*?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. *Dataset* yang digunakan adalah *dataset Monkeypox and Similar Disease*.
2. *Dataset* diambil dengan mengunduh Gambar *monkeypox* dan penyakit serupa yang diperoleh dari penelitian yang dilakukan oleh (Kumar, 2023) tentang “*An XNOR-ResNet and Spatial Pyramid Pooling-based YOLOv3-Tiny Algorithm for Monkeypox and Similar Skin Disease Detection*”.
3. Algoritma *YOLO* yang digunakan yaitu *YOLOv5* dan *YOLOv7*
4. Aplikasi *mobile* dibangun menggunakan *framework flutter* sehingga bisa dijalankan di beberapa *platform*.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan dari penelitian ini adalah:

1. Membangun model *object detection* klasifikasi penyakit *monkeypox* menggunakan kerangka kerja pemilihan algoritma *YOLO*.
2. Mengevaluasi kinerja model *object detection* klasifikasi penyakit *monkeypox* menggunakan kerangka kerja pemilihan algoritma *YOLO*.
3. Men-*deploy* model yang telah dibuat menggunakan kerangka kerja pemilihan algoritma *YOLO* ke dalam aplikasi *mobile*.

1.5 Manfaat Penelitian

Berdasarkan tujuan penelitian di atas, maka manfaat dari penelitian ini adalah sebagai berikut:

1. Menjadi sumber informasi mengenai kinerja model klasifikasi penyakit *monkeypox* menggunakan kerangka kerja pemilihan algoritma *YOLO*.
2. Menyediakan aplikasi *mobile* yang dapat mengenali citra *monkeypox*.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Relevan

Penelitian ini merujuk pada beberapa penelitian yang telah dilakukan sebelumnya. Berikut ini penelitian terdahulu yang berhubungan dengan skripsi ini antara lain:

Penelitian yang berjudul “*Monkeypox Virus Detection Using Pre-trained Deep Learning based Approaches*” diambil pada jurnal *Medical Systems* yang diteliti oleh Chiranjibi Sitaula dan Tej Bahadur Shahi pada tahun 2022 dengan tujuan membandingkan 13 model *pre-trained deep learning* yang berbeda dengan bantuan *transfer learning* pada *dataset monkeypox* (Sitaula & Shahi, 2022). Hasil evaluasi menunjukkan bahwa *ensemble learning* memberikan kinerja tertinggi (*precision*: 85.44%; *recall*: 85.47%; *F1-Score*: 85.40%; dan *accuracy*: 87.13%) selama deteksi virus *monkeypox*.

Penelitian mengenai *monkeypox* juga pernah dilakukan oleh Vanessa Alcalá-Rmz dkk dengan judul “*Convolutional Neural Network for Monkeypox Detection*” pada tahun 2022. Pada penelitian ini menggunakan Gambar yang diperoleh dari *database public* dari *monkeypox*. Gambar – Gambar ini telah diproses sebelumnya, masing – masing dibagi menjadi 80/20 untuk *training* dan *testing*. Penelitian ini menggunakan *MiniGoogleNet* dengan 6 kali percobaan dengan jumlah *epoch* yang berbeda. Model terbaik adalah salah satu dari *epoch 50* dengan akurasi 97.08% dan *loss function* 14.42% (Alcalá-Rmz dkk., 2022).

Penelitian yang berjudul “*An XNOR-ResNet and Spatial Pyramid Pooling-based YOLOv3-tiny Algorithm for Monkeypox and Similar Skin Disease Detection*” yang diteliti oleh Akhlil Kumar pada tahun 2022 dengan tujuan mendeteksi *monkeypox* penyakit kulit lain yang serupa menggunakan algoritma *YOLOv3-tiny*. *Mean Average Precision (mAP)* yang didapatkan pada penelitian ini adalah 76.51% (Kumar, 2023).

Selanjutnya penelitian terkait *monkeypox* juga dilakukan oleh Shams Nafisa Ali dkk., dengan judul “*Monkeypox Skin Lesion Detection Using Deep Learning*

Models: A Feasibility Study” pada tahun 2022. Pada penelitian ini, *dataset* yang digunakan adalah *Monkeypox Skin Lesion Dataset* (MSLD) yang terdiri dari Gambar kulit dari cacar monyet, cacar air dan campak. Augmentasi data digunakan untuk meningkatkan ukuran sampel dan eksperimen validasi silang 3 kali lipat disiapkan. Model *pre-trained deep learning* yang digunakan yaitu *VGG-16*, *ResNet50* dan *InceptionV3* digunakan untuk mengklasifikasikan cacar monyet dan penyakit lainnya (Ali dkk., 2022).

2.2 Cacar Monyet (*Monkeypox*)

Virus *monkeypox* termasuk dalam famili *poxviridae*, subfamily *chordopoxvirinae* dan genus *orthopoxvirus* (Gessain dkk., 2022). *Monkeypox* adalah *orthopoxvirus zoonosis* dengan presentasi yang mirip dengan cacar. *Monkeypox* ditularkan secara tidak sengaja ke manusia ketika mereka bertemu dengan hewan terinfeksi. Virus ditularkan melalui kontak langsung, tetesan pernapasan dan melalui benda seperti handuk dan tempat tidur (Rizk dkk., 2022). Nama *monkeypox* berasal dari penemuan awal virus pada monyet di laboratorium Denmark pada tahun 1958. Kasus pertama pada manusia didiagnosis pada tahun 1970 pada bayi laki-laki berusia 9 bulan di Kongo (Bunge dkk., 2022).

Infeksi *monkeypox* sebelumnya merupakan penyakit endemik di daerah Afrika. Saat ini penyakit *monkeypox* menyebar dengan cepat di negara – negara yang sebelumnya tidak tergolong dalam negara endemik (Budiyarto dkk., 2023).



Gambar 2.2.1 Penyakit *Monkeypox*

(Sumber: (Kumar, 2023))

2.3 Penyakit Kulit Serupa

Penyakit kulit serupa merupakan penyakit kulit yang memiliki wujud karakteristik yang identik dengan penyakit *monkeypox*. Penyakit yang memiliki wujud karakteristik yang identik dengan penyakit *monkeypox* diantaranya ada penyakit cacar air (*chickenpox*) dan campak (*measles*).

2.3.1 Penyakit Cacar Air (*Chickenpox*)

Penyakit cacar air atau *chickenpox* disebabkan oleh virus *varicella zoster*. Virus ini tergolong bagian dari alphaherpes merupakan jenis dari virus *imunogenik*. Penyakit ini dikenal dengan penyakit yang cara penularannya sangat cepat yaitu ketika seseorang yang terinfeksi penyakit ini mencipratkan cairan dari tubuhnya seperti bersin atau batuk ke orang lain yang sehat maka tidak menutup kemungkinan orang tersebut akan terjangkit virus (Anggeriyane dkk., 2022). Penyakit cacar air mempunyai ciri yang sangat khas yaitu ditandai dengan timbulnya benjolan kecil mirip bisul namun kecil disertai dengan air di dalamnya belapis tipis atau biasanya disebut dengan plentingan. Pada awalnya timbul kemerahan pada kulit, kemudian berubah menjadi bitnik-bintik berisi cairan yang menyebar keseluruh tubuh bahkan pada bagian mata (Susanto, 2022).

2.3.2 Penyakit Campak (*Measles*)

Penyakit campak dikenal juga sebagai *Morbili* atau *Measles*, merupakan penyakit yang sangat menular dari genus *morbillivirus* dan termasuk golongan virus RNA (Sihotang dkk., 2019). Penularan dapat terjadi melalui udara yang telah terkontaminasi oleh *droplet* orang yang telah terinfeksi. Gejala campak meliputi demam, batuk, mata merah, pilek dan ruam yang terasa gatal. Ruam berbentuk bitnik-bintik merah yang muncul di seluruh tubuh, biasanya dimulai di belakang telinga dan menyebar ke seluruh tubuh dalam beberapa hari. Ruam biasanya berlangsung selama sekitar 5 – 6 hari dan mungkin menimbulkan rasa gatal.

2.4 Convolutional Neural Network (CNN)

CNN adalah salah satu algoritma paling populer digunakan untuk *deep learning*, sebuah *machine learning* yang model pembelajarannya dikhususkan untuk melakukan klasifikasi langsung pada media dua dimensi seperti Gambar, video, teks atau suara (Ajrana, 2022). Algoritma CNN akan sangat berguna khususnya ketika digunakan untuk mencari pola pada suatu Gambar kemudian mengenali objek pada Gambar tersebut. Penelitian awal yang mendasari CNN ini pertama kali dilakukan oleh Hubel dan Wiesel mengenai visual *cortex* pada indera penglihatan kucing (Hubel & Wiesel, 1962). Pada dasarnya klasifikasi citra menggunakan *Multilayer Feedforward Perceptron* (MLP) sudah bisa dilakukan, akan tetapi ketika digunakan untuk melakukan klasifikasi data dalam jumlah banyak, akurasi yang didapatkannya pun menurun. Oleh karena itu, algoritma CNN dikembangkan karena algoritma ini mampu mempelajari langsung data yang ada pada Gambar, kemudian menggunakan pola yang didapatkan untuk mengklasifikasi.

CNN adalah perkembangan metode dari MLP. Namun, CNN memiliki jumlah dimensi yang lebih banyak dibandingkan dengan MLP (Sabilla, 2020). Berbeda dengan arsitektur MLP, arsitektur CNN ini sebenarnya lebih kompleks dan memiliki proses yang cukup panjang sebelum masuk tahap klasifikasi. Namun, secara garis besar ada 2 tahapan pemrosesan yang dilakukan oleh algoritma CNN ini, yaitu tahap *feature learning* dan tahap *classification*. Tahap *feature learning* merupakan tahap dimana Gambar yang dimasukkan akan diekstraksi untuk

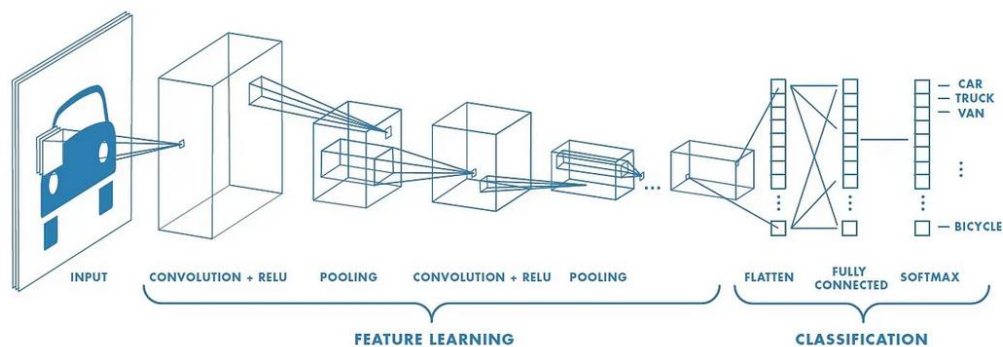
dipelajari *value* dari Gambar tersebut. Proses inilah yang membedakan algoritma CNN dengan MLP (Ajrana, 2022). Jika MLP menggunakan satu proses ekstraksi dalam sekali input untuk melakukan klasifikasi. Sedangkan CNN ini, melakukan klasifikasi bisa menggunakan banyak sekali ekstraksi dalam sekali input. Banyaknya ekstraksi ini disimpan dalam bentuk kedalaman Gambar (*depth*). Proses *feature learning* ini sangat bergantung pada kedalaman suatu Gambar. Semakin dalam suatu Gambar maka semakin banyak ekstraksi yang didapatkan sehingga pola yang didapat juga semakin jelas terbentuk (Li dkk., 2018).

CNN memiliki akurasi tinggi karena memiliki jumlah ekstraksi fitur yang dihasilkan oleh konvolusi dan jumlah neuron, serta penggabungan setiap neuron dengan menggunakan bobot yang diperbarui pada iterasi tertentu. Kombinasi terbaik akan menghasilkan akurasi tinggi (Sabilla, 2020). Adapun kombinasi terbaik dan sering dimodifikasi adalah:

1. Ukuran dari *convolution* dibatasi untuk mendapatkan jumlah *layer*. Semakin banyak iterasi, semakin lama waktu yang dibutuhkan. Akan tetapi, semakin sedikit iterasi juga akan mempengaruhi hasil pendekatan kebenaran akibat jumlah fitur yang sedikit.
2. Ukuran kernel berfungsi sebagai sub matriks. Sehingga pada matriks 2x2 diwakilkan satu nilai yang didapat dari perkalian matriks. Semakin kecil ukuran kernel, semakin detail hasil yang dihasilkan. Akan tetapi, hal tersebut mengakibatkan waktu komputasi semakin lama.
3. Jumlah *layer* berfungsi sebagai penampung hasil konvolusi. Semakin banyak *layer*, fitur yang dihasilkan juga semakin banyak. Akan tetapi, hal tersebut berakibat pada waktu komputasi yang semakin lama.
4. Jumlah *fully-connected* berfungsi menggabungkan ekstraksi fitur ke dalam kelas. Mekanisme yang digunakan adalah memberikan nilai perkalian acak dari bobot dan bias. Ketika nilai tersebut masih jauh, maka perkalian acak diperbarui dan diulang hingga mendapatkan bobot dan bias yang bagus untuk pendekatan kelas. Proses ini memakan waktu yang cukup lama tergantung jumlah fitur yang dihasilkan.

5. *Pooling layer* adalah lapisan pengurangan fitur yang dihasilkan oleh ekstraksi fitur. *Pooling* mengeliminasi dimensi yang tidak digunakan. Parameter dari *pooling* umumnya menggunakan rata – rata atau nilai maksimum.

Pada tahap klasifikasi, model *neural network* akan digunakan untuk melakukan klasifikasi objek berdasarkan kelasnya. Ilustrasi arsitektur dari CNN dapat dilihat pada Gambar 2.4.1.



Gambar 2.4.1 Arsitektur *Convolutional Neural Network* (CNN)

(Sumber: medium.com/@samuelsena)

Berdasarkan pemaparan di atas, banyak parameter CNN yang dapat dioptimalisasi untuk mendapatkan nilai presisi dan akurasi tinggi serta waktu yang singkat. Hal itu dilakukan dengan cara mencari kombinasi parameter yang paling tepat. Hasil kombinasi parameter yang sudah dioptimalisasi pada *dataset* tertentu dapat digunakan untuk *dataset* yang lainnya atau disebut dengan *transfer learning*.

Pada Gambar 2.4.1 merupakan Gambar arsitektur dari CNN. Secara umum tahapan klasifikasi CNN dibagi menjadi dua bagian besar yaitu *feature learning* dan *classification*. Berikut adalah penjelasan tahapan dari arsitektur CNN tersebut.

2.4.1 *Convolutional Layer*

Convolutional merupakan salah satu tahap pada arsitektur CNN. *Convolutional* merupakan suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang (Irfansyah dkk., 2021).

Dalam pengolahan citra, *convolutional* berarti mengaplikasikan sebuah *kernel* pada citra. *Kernel* adalah sebuah matriks kecil dengan tinggi dan lebarnya lebih kecil dari matriks citra yang akan dilakukan konvolusi. *Kernel* biasanya juga dikenal dengan istilah *filter* atau *convolution mask* (Widyaya & Budi, 2021).

Dalam *machine learning*, *input* citra berbentuk *array* dan dimensi dan *kernel* merupakan parameter berbentuk *array* multidimensi yang disesuaikan dengan model algoritma. Konvolusi dapat digunakan pada lebih dari satu dimensi. Sebagai contoh jika menggunakan Gambar dua dimensi I sebagai *input*, maka *kernel* K juga berbentuk dua dimensi:

$$S(i, j) = (I \times K)(i, j) = \sum_a \sum_b I(a, b)K(i - a, j - b) \quad (2.1)$$

Keterangan:

$S(i, j)$ = Fungsi hasil konvolusi

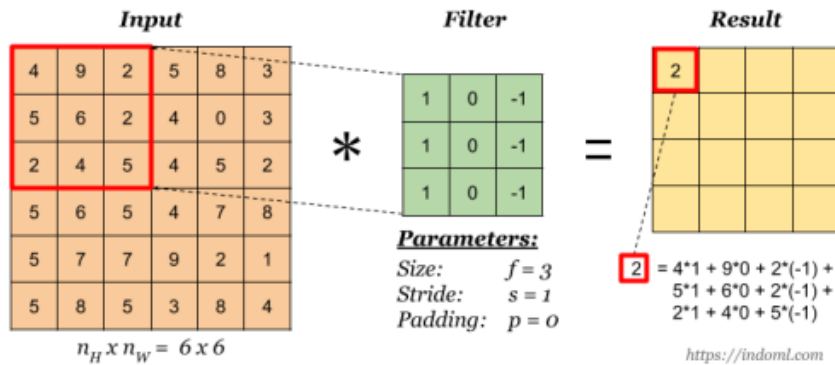
I = *Input*

K = *Kernel* atau *filter*

(i, j) = *Pixel Input*

(a, b) = *Pixel Kernel*

Kernel ini biasanya berukuran kecil dalam dimensi spasial, tetapi tersebar di seluruh kedalaman *input*. Saat data mengenai *convolution layer*, *layer* tersebut menggabungkan setiap *filter* melintasi dimensi spasial *input* untuk menghasilkan peta aktivasi 2D (O'Shea & Nash, 2015). Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input, Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial pada data. Bobot pada *layer* tersebut menspesifikasikan *kernel* konvolusi yang digunakan, sehingga *kernel* konvolusi dapat dilatih berdasarkan input pada CNN (Putra, 2016).



Gambar 2.4.2 Contoh Bentuk Konvolusi

(Sumber: <https://indoml.com/>)

Gambar 2.4.2 merupakan ilustrasi sebuah matriks dengan input berukuran 6x6 dimana dilakukan konvolusi dengan *kernel* atau *filter* berukuran 3x3 dengan *stride* atau perpindahannya sebanyak 1 dan *zero padding*. Hasil dari konvolusi tersebut ditunjukkan pada matriks *result* atau biasa disebut sebagai *feature map*.

2.4.2 Stride & Padding

Stride adalah parameter yang menentukan berapa jumlah pergeseran *filter*. Jika nilai *stride* adalah 1, maka konvolusi *filter* akan bergeser sebanyak 1 *pixel* secara *horizontal* kemudian akan bergeser secara *vertical*. Semakin kecil *stride* maka akan semakin detail informasi yang didapatkan dari sebuah input, namun membutuhkan komputasi yang lebih jika dibandingkan dengan *stride* yang besar dan perlu kita perhatikan bahwa dengan menggunakan *stride* yang kecil tidak selalu akan mendapatkan performa yang bagus.

Padding atau *zero padding* adalah parameter yang menentukan jumlah *pixel* yang akan ditambahkan di setiap sisi dari *input*. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi *output* dari *convolutional layer* (*feature map*). *Size filter* tidak selalu lebih kecil dari *input size*, suatu *padding* p diberikan kepada *input* citra agar *size* dari *input* lebih besar atau sama dengan *size* dari *filter*. Suatu *input* citra berukuran $n \times n$, *padding* p akan merubah *size input* menjadi $(n + 2p) \times (n + 2p)$.

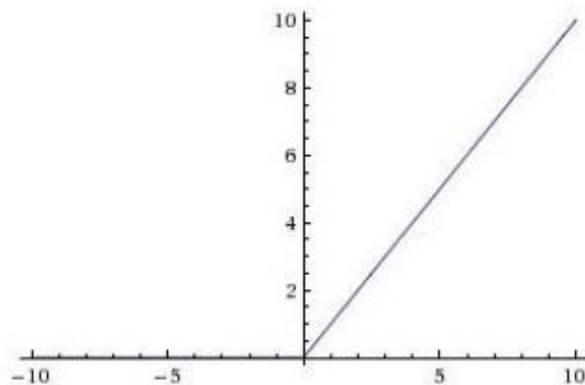
2.4.3 Activation Layer

Fungsi aktivasi biasa disebut sebagai lapisan pemetaan *non-linear*. Fungsi aktivasi biasa digunakan untuk meningkatkan kemampuan *network classification* (Chen dkk., 2018). Salah satu peranan dari fungsi aktivasi adalah untuk memberikan kemampuan *network* agar dapat melakukan tugas *non-linear*. Tanpa fungsi aktivasi, *neural network* hanyalah kombinasi operasi *linear* yang hanya melakukan tugas- tugas yang *linear* pula (Mukhtarom, 2020). Padahal kebanyakan kasus nyata di lapangan merupakan kasus *non-linear*.

Rectified Linear Unit (ReLU) merupakan salah satu fungsi aktivasi yang sering digunakan pada CNN. Fungsi aktivasi ReLu menjalankan operasi nilai ambang (*threshold*) untuk setiap elemen input dimana jika nilai lebih kecil dari nol akan diset menjadi nol. Persamaan 2.2 merupakan bentuk fungsi ReLu.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.2)$$

Pada fungsi aktivasi ReLu, semua nilai x *negative* akan dipetakan ke 0, seperti pada Gambar 2.4.3.



Gambar 2.4.3 Grafik Fungsi ReLu

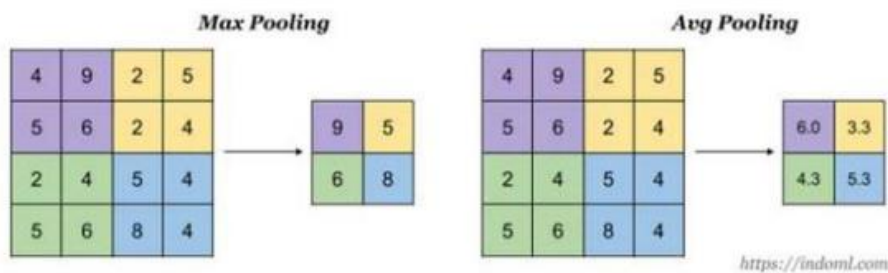
(Sumber: towardsdatascience/@dimashulga)

Pada fungsi ini input dari *neuron-neuron* berupa bilangan negatif, maka fungsi ini akan menterjemahkan nilai tersebut ke dalam nilai 0 dan jika input

bernilai positif maka *output* dari *neuron* adalah nilai aktivasi itu sendiri. Fungsi aktivasi ReLu memiliki kelebihan yaitu dapat mempercepat proses konfigurasi yang dilakukan dengan *Stochastic Gradient Descent* (SGD) jika dibandingkan dengan fungsi *sigmoid* dan *tanh*.

2.4.4 Pooling Layer

Pooling layer atau *subsampling* diletakkan di antara *convolutional layer* dan *ReLU layer* dengan tujuan untuk mengurangi jumlah parameter perhitungan, seperti lebar dan tinggi citra, tetapi bukan kedalaman citra. (Wulandari, 2019). Langkah melakukan *pooling* adalah dengan membagi hasil keluaran *convolutional layer* menjadi beberapa *grid* berdasarkan penentuan jumlah *stride* dan jenis *pooling* yang digunakan. Terdapat dua jenis *pooling* yang umum digunakan pada *pooling layer*, yaitu *Max Pooling* dan *Average Pooling*. Pada *max pooling*, nilai yang diambil merupakan nilai maksimal dari setiap *grid*. Sedangkan pada *average pooling*, nilai yang diambil merupakan nilai rata-rata dari setiap *grid*.



Gambar 2.4.4 Ilustrasi Perhitungan pada *Pooling Layer*

(Sumber: <https://indoml.com/>)

Perhitungan dimensi keluaran dari *pooling layer* dinyatakan dalam persamaan 2.3 berikut.

$$O = \frac{I-K}{S} + 1 \quad (2.3)$$

Keterangan:

O = dimensi dari keluaran *pooling layer*

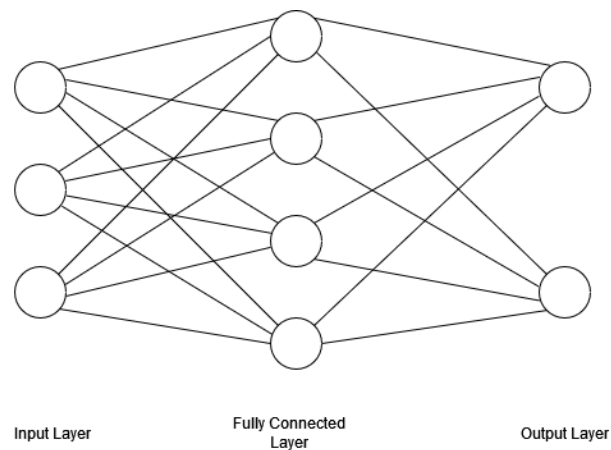
I = dimensi matriks masukan

K = dimensi matriks *kernel*

S = jumlah *stride*

2.3.5 Fully Connected Layer

Fully Connected Layer merupakan lapisan terakhir dalam CNN yang berarti setiap *neuron* pada lapisan sebelumnya dihubungkan dengan setiap *neuron* di *fully connected layer* (Wulandari, 2019). *Fully connected layer* adalah lapisan yang digunakan untuk melakukan transformasi pada dimensi data agar dapat diklasifikasikan secara *linear* (Putra, 2016). Gambar 2.4.5 menampilkan proses yang ada di dalam *fully connected layer*.



Gambar 2.4.5 Ilustrasi proses *fully connected layer*

Pada Gambar 2.4.5 terdapat *input layer* yang mana *input layer* tersebut merupakan hasil dari proses *flattening*. Proses *flattening* inilah yang menghasilkan sebuah vector yang akan digunakan sebagai input dari *fully connected layer*. Kemudian pada bagian *fully connected layer* memiliki beberapa *layer-layer* seperti *dense layer* dan *dropout layer*. Pada bagian *output layer* merupakan tahap terakhir yang berfungsi mendefinisikan jenis dari sebuah kelas.

2.5 You Only Look Once (YOLO)

YOLO adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara *real-time*. Sistem pendeteksian yang dilakukan dengan

menggunakan *repurpose classifier* atau *localizer* untuk melakukan deteksi (Rahmadkk., 2021).

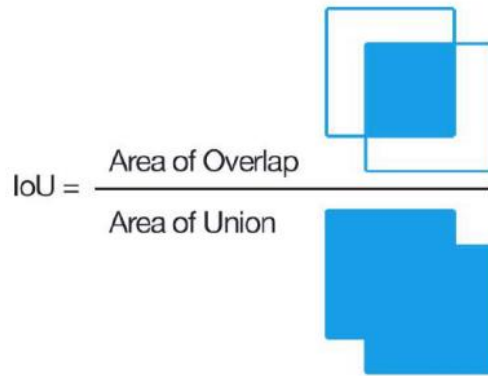
YOLO pertama kali diciptakan oleh Joseph Redmon pada tahun 2015 adalah sistem deteksi objek *real-time* berdasarkan CNN. *YOLO* bekerja dengan menerapkan jaringan saraf tunggal ke Gambar penuh. Jaringan ini membagi Gambar menjadi daerah dan memprediksi kotak pembatas dan probabilitas untuk setiap wilayah (Surbakti & Putri, 2022). *YOLO* menggunakan pendekatan jaringan saraf tiruan untuk mendeteksi objek pada sebuah citra. Jaringan ini membagi citra menjadi beberapa wilayah dan memprediksi setiap kotak pembatas dan probabilitas untuk setiap wilayah. Kotak – kotak pembatas kemudian dibandingkan dengan probabilitas yang diprediksi. *YOLO* memiliki beberapa kelebihan dibandingkan dengan sistem yang berorientasi pada *classifier*.

YOLO membagi input citra menjadi *grid – grid* dengan ukuran $S \times S$. Pada setiap *grid* bertanggung jawab untuk memprediksi adanya objek dan *bounding box* dan *confidence*. Nilai *confidence* akan memprediksikan bahwa adanya sebuah objek. Setiap *bounding box* terdapat parameter yaitu x, y, w, h dan *confidence*. Parameter x dan y adalah koordinat dari *bounding box*. Parameter w dan h adalah tinggi dan lebar *bounding box*. *Confidence* adalah nilai dari *Intersection Over Union* dari perhitungan dari *predicted box* dan *ground-truth box* (Setiawan, 2021). Secara umum nilai *confidence* memiliki persamaan sebagai berikut:

$$Confidence = Pr(Object) * IOU \quad (2.4)$$

$Pr(Object)$ merupakan nilai objektivitas, apabila tidak terdapat objek maka akan bernilai nol dan bernilai satu jika terdapat sebuah objek. *IOU* merupakan nilai rasio *Intersection Over Union* antara kotak *ground truth* dan kotak prediksi. *YOLO* mengkalkulasi probabilitas kelas kondisional dengan prediksi kotak confidence dengan persamaan.

$$Pr(Class i | Object) * Pr(Object) * IOU = Pr(Class i) * IOU \quad (2.5)$$

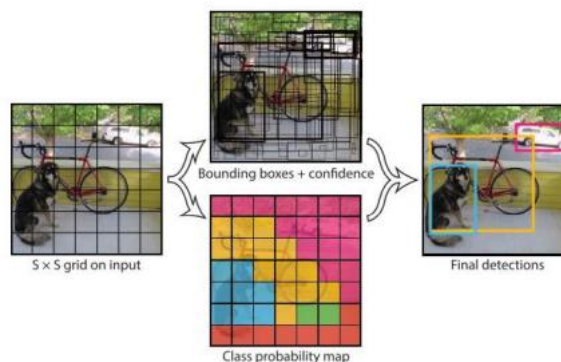


Gambar 2.5.1 *Intersection Over Union*

(Sumber: gabormelli.com)

IOU atau *Intersection Over Union* merupakan metrik yang mengukur akurasi model dalam mendeteksi objek pada kumpulan data latih (Effendi, 2022). *IOU* membandingkan *bounding box ground-truth* dan *bounding box* prediksi yang terletak pada objek dalam citra Gambar. Nilai *IOU* yang dihasilkan akan berfungsi sebagai nilai *confidence* untuk ke tahap proses *non max suppression*.

Non max suppression berfungsi untuk menghilangkan *bounding box* yang *overlap* / tumpang tindih terhadap objek dalam suatu Gambar. *Non max suppression* akan memanfaatkan dua nilai yaitu *IOU* dan *Threshold*. *Threshold* merupakan nilai yang ditentukan oleh peneliti sebagai ambang batas. Apabila nilai *IOU* dari suatu *bounding box* lebih kecil dari nilai *threshold* maka *bounding box* tersebut akan dihilangkan.



Gambar 2.5.2 Algoritma Deteksi *YOLO*

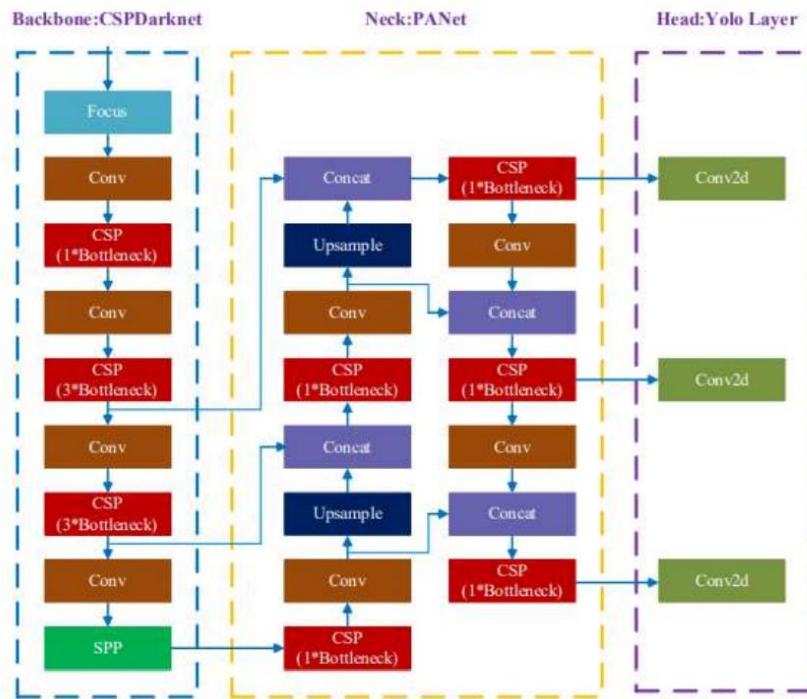
(Sumber: <https://github.com/pjreddie/darknet/wiki/YOLO:-Real-Time-Object-Detection>)

2.5.1 *You Only Look Once (YOLOv5)*

YOLOv5 merupakan *framework* yang didalamnya terdapat algoritma pendeteksian objek yang dikembangkan oleh peneliti dan CEO dari *Ultralytics* yaitu Glenn Jocher pada tahun 2020. *YOLOv5* menggunakan *framework PyTorch* yang ditulis menggunakan Bahasa pemrograman *python* (Baihaqi & Zonyfar, 2022). *YOLOv5* merupakan hasil dari pengembangan implementasi *YOLOv3* pada *framework PyTorch* yang dikembangkan oleh Gleen Jocher. *YOLOv5* memiliki lima *pre-trained* model dengan ukuran yang berbeda, yaitu *YOLOv5s* (ukuran paling kecil), *YOLOv5m*, *YOLOv5l* dan *YOLOv5x* (ukuran yang paling besar).

Menurut (Jiang dkk., 2022) *YOLO* merupakan salah satu yang cepat dalam hal deteksi. Prinsip dari algoritma *YOLO* yang digunakan adalah mendeteksi benda atau target dengan menilai blok piksel berdasarkan warna dan bentuk yang telah dilatih sehingga kerusakan – kerusakan pada Gedung dapat terdeteksi dan diklasifikasi. Secara umum arsitektur *YOLOv5* tidak jauh berbeda dengan *YOLO* versi sebelumnya. *YOLOv5* memiliki tiga kerangka kerja utama yaitu *backbone*, *neck* dan *head*. Bagian *backbone YOLOv5* menggunakan *CSPDarknet53* yang diadopsi 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* menggunakan *Path Aggregation Network (PANet)* untuk meningkatkan arus informasi. *PANet* ini pengembangan dari *feature pyramid network (FPN)* di *YOLOv3* yang mencakup lapisan *bottom up* dan *top down*. Hal ini dapat meningkatkan propagasi level rendah fitur dalam model.

Arsitektur *YOLOv5* memiliki tiga kerangka utama yang tersusun dari *backbone*, *neck* dan *head*. Berikut Gambar 2.5.3 tentang arsitektur dari algoritma *YOLOv5*.



Gambar 2.5.3 Arsitektur *YOLOv5*

(Sumber: <https://github.com/ultralytics/yolov5/issues/280>)

YOLOv5 memiliki beberapa tipe antara lain *YOLOv5n*, *YOLOv5s*, *YOLOv5m*, *YOLOv5l* dan *YOLOv5x*. Masing – masing tipe tersebut memiliki perbedaan dari segi kecepatan deteksi dan performa akurasi *mAP* nya. Perbedaan tipe *YOLO* tersebut didasarkan pada jumlah *depth multiple* dan *layer channel multiple* (Effendi, 2022). Berikut Tabel 2.5.1 perbandingan performa *YOLOv5*.

Tabel 2.5.1 Perbandingan performa *YOLOv5*

Model	Size	<i>mAP</i> 0.5:0.95	<i>mAP</i> 0.5	<i>Speed</i> CPU b1 (ms)	<i>Speed</i> V100 b1 (ms)	<i>Speed</i> V100 b32 (ms)	<i>Params</i> (M)	FLOPS@640 (B)
<i>YOLOv5n</i>	640	28.0	45.7	45	6.3	0.6	1.9	4.5
<i>YOLOv5s</i>	640	37.4	56.8	98	6.4	0.9	7.2	16.5
<i>YOLOv5m</i>	640	45.4	64.1	224	8.2	1.7	21.2	49.0
<i>YOLOv5l</i>	640	49.0	67.3	430	10.1	2.7	46.5	109.1
<i>YOLOv5x</i>	640	50.7	68.9	766	12.1	4.8	86.7	205.7

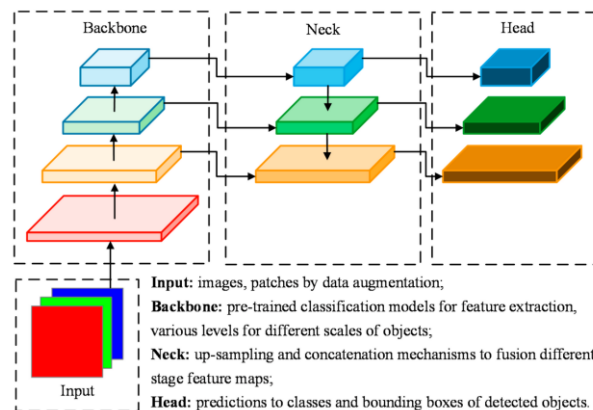
2.5.2 *You Only Look Once (YOLOv7)*

Model *YOLOv7* merupakan model yang terbaru dalam keluarga *YOLO*. Model *YOLO* adalah detector objek yang arsitekturnya berasal dari *YOLOv4*, *Scaled YOLOv4* dan *YOLO-R*. *YOLOv7* meningkatkan kecepatan dan akurasi dengan memperkenalkan beberapa reformasi arsitektur. Mirip dengan *YOLOv4*, *backbone YOLOv7* tidak menggunakan *backbone* terlatih *ImageNet*. Sebaliknya, model dilatih menggunakan dataset *COCO* seluruhnya. *YOLOv7* memiliki beberapa tipe antara lain *YOLOv7-tiny*, *YOLOv7*, *YOLOv7-X*, *YOLOv7-W6*, *YOLOv7-E6*, *YOLOv7-D6*, *YOLOv7-E6E* (Wang dkk., 2022). Berikut Tabel 2.5.2 tentang perbandingan performa *YOLOv7*.

Tabel 2.5.2 Perbandingan performa *YOLOv7*

Model	Parameters	FPS	AP test (%)
<i>YOLOv7-tiny</i>	6.2	286	38.7
<i>YOLOv7</i>	36.9	161	51.4
<i>YOLOv7-X</i>	71.3	114	53.1
<i>YOLOv7-W6</i>	70.4	84	54.9
<i>YOLOv7-E6</i>	97.2	56	56
<i>YOLOv7-D6</i>	154.7	44	56.6
<i>YOLOv7-E6E</i>	151.7	36	58.8

Arsitektur *YOLOv7* juga memiliki tiga kerangka utama seperti *backbone*, *neck* dan *head*. Berikut Gambar 2.5.4 tentang arsitektur dari algoritma *YOLOv7*.



Gambar 2.5.4 Arsitektur *YOLOv7*

(Sumber: <https://www.researchgate.net/publication/359169212>)

2.6 Ukuran Kinerja Model

2.6.1 Confusion Matrix

Evaluasi kinerja pada suatu sistem klasifikasi merupakan sesuatu yang penting untuk mengetahui seberapa baik sistem yang dibangun dalam mengklasifikasikan data. Berdasarkan jumlah kelas pada keluaran hasil sistem klasifikasi, jenis klasifikasi dikelompokkan menjadi empat jenis yaitu klasifikasi biner, *multi-case*, *multi-label* dan *hierarchical*. Pada pengukuran kinerja sistem klasifikasi umumnya digunakan *confusion matrix* untuk membandingkan hasil klasifikasi oleh sistem dengan hasil klasifikasi sesungguhnya. Berikut Tabel *Confusion Matrix multiclass* yang dapat dilihat pada Tabel 2.6.1

Tabel 2.6.1 *Confusion Matrix Multiclass*

	<i>Predicted</i>	
<i>Actual</i>	<i>Class 1</i>	<i>Class 2</i>
<i>Class 1</i>	$T_{(1,1)}$	$F_{(1,2)}$
<i>Class 2</i>	$F_{(2,1)}$	$T_{(2,2)}$

Tabel 2.6.1 tersebut bisa diketahui parameter *accuracy*, *precision*, *recall* dan *F1-Score*. Berikut adalah penjelasan dan rumus untuk masing – masing parameter tersebut.

- Accuracy* merupakan rasio prediksi positif dan negatif dengan keseluruhan data. Berdasarkan Tabel di atas maka akurasi dapat dirumuskan:

$$Accuracy = \frac{\sum_{i=1}^c T_i}{N} \quad (2.4)$$

Keterangan

$$i = 1, 2$$

- c = Banyaknya *class*
- N = Banyaknya *dataset*
- T_i = Jumlah data yang terklasifikasi *True* untuk kelas ke- i

b. *Precision* merupakan rasio prediksi benar positif jika dibandingkan dengan keseluruhan hasil yang diprediksi positif. Presisi dapat diwakili dengan rumus

$$Precision(i) = \frac{T_i}{T_i + \sum_{j=1}^c F_{ij}} \quad (2.5)$$

Keterangan

- i = 1, 2
- j = 1, 2
- c = Banyaknya *class*
- F = Jumlah data yang terklasifikasi *False* pada masing-masing *class*
- i = Menyatakan baris pada masing-masing *class*
- j = Menyatakan kolom pada masing-masing *class*

c. *Recall* atau bisa disebut juga dengan *sensitivity* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. *Recall* dapat dihitung dengan rumus

$$Recall(j) = \frac{T_i}{T_i + \sum_{i=1}^c F_{ij}} \quad (2.6)$$

d. *F1-Score* adalah kombinasi rata-rata dari *precision* dan *recall*. *F1-Score* bisa dihitung dengan menggunakan rumus

$$F1 - Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (2.7)$$

2.7 Mean Average Precision (mAP)

Mean Average Precision adalah metrik yang digunakan untuk mengukur keakuratan detector objek di semua kelas dalam dataset tertentu. Sebelum menentukan nilai *mAP*, perlu diketahui lebih dahulu nilai dari *Average Precision* (*AP*) yang diperoleh dengan menghitung *Area Under Curve* (*AUC*) dari kurva *precision-recall*. Namun kurva *precision-recall* sering kali berbentuk zig-zag yang naik dan turun yang menyebabkan sulitnya menentukan estimasi nilai *AUC*. Pada dasarnya, *mean average precision* (*mAP*) hanyalah *AP* rata-rata dari semua kelas yang dinyatakan pada persamaan berikut.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (2.8)$$

AP_i adalah kelas ke i dan N adalah total kelas yang dievaluasi (HARIS, 2022).

2.8 Library

Dalam membangun model arsitektur digunakan Bahasa *python* dengan dukungan *library* atau *framework* yang memadai. Berikut merupakan *library python* yang digunakan dalam membangun arsitektur model CNN.

2.8.1 PyTorch

PyTorch merupakan Pustaka sumber terbuka untuk pembelajaran mesin yang pertama kali dikembangkan oleh *Facebook Research*. *Pytorch* digunakan sebagai pengganti *numpy* untuk menggunakan kekuatan GPU dalam proses pembelajaran mesin.

2.8.2 Matplotlib

Matplotlib digunakan untuk membuat visualisasi data. Contoh penggunaannya di CNN yaitu mem-plot *dataset* beserta labelnya dan mem-plot hasil *training* maupun *testing* model CNN ke dalam sebuah grafik

2.9 Rancang Bangun Aplikasi Mobile

2.9.1 Framework Flutter

Flutter adalah sebuah *framework open-source* atau SDK yang dikembangkan oleh *google* untuk membangun antar muka aplikasi yang memiliki kinerja tinggi serta dapat dipublikasi ke *platform android* dan *iOS* dari *codebase* tunggal (Suhendro dkk., 2021). *Flutter* menggunakan Bahasa pemrograman *Dart* yang pastinya terasa familiar dengan Bahasa pemrograman *Java* atau *Javascript*. *Dart* merupakan Bahasa pemrograman yang dikembangkan oleh *google* untuk kebutuhan umum.

2.9.2 Flask

Flask merupakan sebuah *web framework* yang ditulis dengan Bahasa *python* dan tergolong sebagai jenis *microframework* karena tidak membutuhkan *tools* atau *library* tertentu serta memiliki *database* bawaan. *Flask* menggunakan *Jinja Template Engine* dan *Werkzeug WSGI ToolKit*.