

**DETEKSI *COLORECTAL POLYP* BERDASARKAN
CITRA *COLONOSCOPY* MENGGUNAKAN *YOU
ONLY LOOK ONCE 11 (YOLO11)***

SKRIPSI



Niken Kurniasari

NIM 082111733009

PROGRAM STUDI S1-TEKNIK BIOMEDIS

DEPARTEMEN FISIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS AIRLANGGA

2025

LEMBAR PERNYATAAN

**DETEKSI COLORECTAL POLYP BERDASARKAN CITRA COLONOSCOPY
MENGUNAKAN YOU ONLY LOOK ONCE 11 (YOLO11)**

SKRIPSI

Sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana Sains Bidang Teknik pada
Fakultas Sains dan Teknologi Universitas
Airlangga

Niken Kurniasari
NIM 082111733009

Disetujui oleh:

Pembimbing I



Endah Purwanti, S.Si., M.T.
NIP 197710312009122003

Pembimbing II



dr. Tri Asih Imro'ati, Sp.PD., K-GEH,
FINASIM
NIP 197711152003122007

LEMBAR PENGESAHAN NASKAH SKRIPSI

Judul : *Deteksi Colorectal Polyp Berdasarkan Citra Colonoscopy Menggunakan You Only Look Once 11 (YOLO11)*
Penyusun : Niken Kurniasari
NIM : 082111733009
Pembimbing I : Endah Purwanti, S.Si., M.T.
Pembimbing II : dr. Tri Asih Imro'ati, Sp.PD., K-GEH, FINASIM
Tanggal Ujian : 28 Mei 2025

Disetujui oleh,

Pembimbing I



Endah Purwanti, S.Si., M.T.
NIP 197710312009122003

Pembimbing II



dr. Tri Asih Imro'ati, Sp.PD., K-GEH,
FINASIM
NIP 197711152003122007

Mengetahui,

Sekretaris Departemen Fisika
Fakultas Sains dan Teknologi
Universitas Airlangga



Dr. Nuril Ukhrowivah, M.Si.
NIP 197004141999032001

Ketua Program Studi S1 Teknik Biomedis
Fakultas Sains dan Teknologi
Universitas Airlangga



Dr. Riries Rulaningtyas S.T., M.T.
NIP 197903152003122002

PEDOMAN PENGGUNAAN SKRIPSI

Skripsi ini tidak dipublikasikan, namun tersedia di perpustakaan dalam lingkungan Universitas Airlangga, diperkenankan untuk dipakai sebagai referensi kepustakaan, tetapi pengutipan harus seizin penyusun dan harus menyebutkan sumbernya sesuai kebiasaan ilmiah. Dokumen skripsi ini merupakan hak milik Universitas Airlangga.

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Niken Kurniasari

NIM : 082111733009

Program Studi : S1 Teknik Biomedis

Judul Skripsi : Deteksi *Colorectal Polyp* Berdasarkan Citra *Colonoscopy* Menggunakan
You Only Look Once 11 (YOLO11)

Menyatakan dengan sesungguhnya bahwa bagian atau keseluruhan skripsi ini tidak pernah diajukan untuk mendapatkan gelar akademis pada bidang studi dan/atau universitas lain dan tidak pernah dipublikasikan/ditulis oleh individu selain penyusun kecuali bila dituliskan dengan format kutipan dalam isi skripsi.

Demikian pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku di Universitas Airlangga. Pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Surabaya, 24 Juni 2025

Penulis



Niken Kurniasari

NIM 082111733009

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT berkat rahmat, hidayah dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “Deteksi *Colorectal Polyp* Berdasarkan Citra *Colonoscopy* Menggunakan *You Only Look Once 11 (YOLO11)*”. Naskah skripsi ini disusun sebagai dasar dan syarat pengerjaan serta penulisan skripsi yang merupakan salah satu syarat akademik dalam memperoleh gelar Sarjana Teknik pada program studi Teknik Biomedis, Fakultas Sains dan Teknologi, Universitas Airlangga. Penyelesaian naskah skripsi ini telah mendapatkan banyak bantuan, arahan, serta bimbingan dari berbagai pihak. Oleh karena ini penulis berkenan menyampaikan banyak terima kasih kepada:

1. Orang tua tercinta, Bapak Chodori dan Ibu Supiyati Minarti, yang telah berjuang demi masa depan penulis, memberikan dukungan, dan doa-doa terbaik yang tiada putusnya untuk penulis.
2. Ibu Endah Purwanti, S.Si., M.T, selaku dosen pembimbing I, yang telah memberikan arahan, nasihat, bimbingan, dan saran berharga selama masa penyusunan naskah dan penelitian skripsi.
3. dr. Tri Asih Imro'ati, Sp.PD., K-GEH, FINASIM, selaku dosen pembimbing II, yang telah memberikan masukan, bimbingan, dan doa selama penyusunan naskah dan penelitian skripsi.
4. Bapak Herri Trilaksana, S.Si., M.Si., Ph.D., selaku Kepala Departemen Fisika Fakultas Sains dan Teknologi Universitas Airlangga.
5. Dr. Riries Rulaningtyas, S.T., M.T., selaku Koordinator Program Studi S1 Teknik Biomedis, yang telah memberikan pengaruh positif terhadap perkembangan program studi dan mahasiswa khususnya.
6. Prof. Dr. Khusnul Ain, S.T., M.Si., selaku dosen wali, yang telah memberikan arahan dan saran selama masa perkuliahan penulis.
7. Bapak dan Ibu dosen Universitas Airlangga, terutama S1 Teknik Biomedis, yang telah membimbing dan memberikan wawasan, ilmu serta pengalamannya selama perkuliahan penulis.
8. Seluruh sahabat terkasih, Elda, Giselle, Nismara, Inov, Chania, Diah, Disal, Mutiara, Ghisella, Virgo, dan Alvian yang telah menjadi *partner* dalam bertumbuh di segala kondisi, menjadi pendukung, penghibur, dan pendengar yang baik dalam segala

masalah penulis terutama pada masa perantauan, serta memotivasi penulis dalam menyelesaikan skripsi dan pendidikan.

9. Teman-teman Teknik Biomedis Universitas Airlangga Angkatan 2021, Neutra, yang telah berjuang dan berproses bersama hingga habisnya masa perkuliahan.
10. Teman-teman Keilmuan Aksi Bersama, yang telah mendukung dan memberikan semangat penulis dalam masa skripsi.
11. Semua pihak yang tidak dapat penulis sebutkan satu persatu, terima kasih atas segala bantuannya dalam penyelesaian naskah skripsi sekaligus pendidikan penulis.
12. Diri sendiri, yang telah berhasil berjuang sejauh ini dan tidak pernah memutuskan untuk menyerah dalam kondisi apapun yang telah terjadi. *Sorry for crying and complaining so much lately, but also thank me for being strong all this time.*

Penulis berharap proposal skripsi ini sedikit banyaknya dapat memberi manfaat baik kepada penyusun sendiri dan juga kalangan lain.

Surabaya, 20 Mei 2025

Niken Kurniasari

Niken Kurniasari, 2025. **Deteksi *Colorectal Polyp* Berdasarkan Citra *Colonoscopy* Menggunakan *You Only Look Once 11 (YOLO11)***. Skripsi ini di bawah bimbingan Endah Purwanti, S.Si., M.T. dan dr. Tri Asih Imroati, Sp.PD., K-GEH, FINASIM, Program Studi S1 Teknik Biomedis, Departemen Fisika, Fakultas Sains dan Teknologi, Universitas Airlangga, Surabaya.

ABSTRAK

Colorectal polyp merupakan prekursor dari sebagian besar terbentuknya *colorectal cancer*. Penyakit ini merupakan salah satu kasus kanker yang menempati 4 besar di Indonesia, berdasarkan *Global Cancer Observatory (GLOBOCAN)* tahun 2022. Oleh karena itu, perlunya pendeteksian lebih dini sehingga dapat mencegah perkembangan keganasan *colorectal polyp* serta resiko pertumbuhan kanker dapat dihentikan. *Colonoscopy* menjadi salah satu metode deteksi awal *colorectal polyp* yang umum digunakan, namun proses ini sangat bergantung pada keahlian dan konsistensi ahli gastroenterologi, serta rentan terhadap kelelahan dan bias antar pengamat. Penelitian ini bertujuan untuk mengetahui pengaruh pemberian *enhancement* dalam performa *You Only Look Once 11 (YOLO11)* untuk membantu dokter dalam mendeteksi *colorectal polyp* dari citra *colonoscopy*. Data yang digunakan merupakan gabungan data sekunder dan primer sebagai representatif terhadap kondisi masyarakat di lapangan yang diolah dengan menggunakan skenario *Adaptive Gamma Correction (AGC)* untuk menyamaratakan intensitas dan *Highlight Removal (HR)* untuk menghilangkan pantulan Cahaya pada gambar. Hasil penelitian menunjukkan citra dari skenario AGC-HR menghasilkan kinerja terbaik melalui matriks evaluasi dengan *precision* 0.96, *recall* 0.89, dan *f1-score* 0.92 dibandingkan skenario lain. Namun hasil uji signifikansi menunjukkan performa skenario AGC-HR tidak signifikan terhadap skenario asli yaitu *p-value* 0,299. Secara keseluruhan, tidak ada satu pun skenario alternatif yang menunjukkan perbedaan signifikan terhadap satu sama lain sehingga dapat disimpulkan bahwa perubahan dalam skenario-skenario tersebut tidak memberikan pengaruh yang berarti terhadap hasil yang diamati.

Kata kunci: colorectal polyp, colonoscopy, YOLO11, enhancement.

Niken Kurniasari, 2025. **Colorectal Polyp Detection Based on Colonoscopy Image Using You Only Look Once 11 (YOLO11)**. Undergraduate research paper, supervised by Endah Purwanti, S.Si., M.T. and dr. Tri Asih Imroati, Sp.PD., K-GEH, FINASIM, Biomedical Engineering Undergraduate Study Program, Department of Physics, Faculty of Science and Technology, Universitas Airlangga, Surabaya.

ABSTRACT

Colorectal polyp is a precursor to most colorectal cancers. This disease is one of Indonesia's top 4 cancer cases, based on the Global Cancer Observatory (GLOBOCAN) in 2022. Therefore the need for early detection so that it can prevent the development of colorectal polyp malignancy and the risk of cancer growth can be stopped. Colonoscopy is one of the commonly used methods of early colorectal polyp detection, but this process is highly dependent on the expertise and consistency of gastroenterologists, and is prone to fatigue and inter-observer bias. This study aims to determine the effect of providing enhancement in the performance of You Only Look Once 11 (YOLO11) to assist doctors in detecting colorectal polyps from colonoscopy images. The data used is a combination of secondary and primary data as a representative of community conditions in the field that are processed using the Adaptive Gamma Correction (AGC) scenario to equalize the intensity and Highlight Removal (HR) to remove light reflections in the image. The results showed that the images from the AGC-HR scenario produced the best performance through the evaluation matrix with precision 0.96, recall 0.89, and f1-score 0.92 compared to other scenarios. However, the significance test results show that the performance of the AGC-HR scenario is not significant against the original scenario with a p-value of 0.299. Overall, none of the alternative scenarios showed significant differences against each other so it can be concluded that the changes in the scenarios did not have a significant effect on the observed results.

Keywords: colorectal polyp, colonoscopy, YOLO11, enhancement.

DAFTAR ISI

	Halaman
LEMBAR PERNYATAAN.....	ii
LEMBAR PENGESAHAN NASKAH SKRIPSI	iii
PEDOMAN PENGGUNAAN SKRIPSI.....	iv
LEMBAR PERNYATAAN KEASLIAN.....	v
KATA PENGANTAR	vi
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
DAFTAR SINGKATAN	xiv
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian.....	4
1.4.1 Manfaat Teoritis.....	4
1.4.2 Manfaat Praktis.....	4
1.5 Batasan Masalah	4
BAB II STUDI PUSTAKA	6
2.1 Saluran <i>Gastrointestinal</i>	6
2.2 <i>Colorectal Polyp</i>	7
2.2.1 Gejala.....	8
2.2.2 Jenis Polip.....	8
2.2.3 Diagnosis	9
2.2.4 Penanganan.....	10
2.3 Pengolahan Citra Digital	11
2.3.1 Citra Digital	11
2.3.2 Sistem Pengolahan.....	12
2.4 <i>Machine Learning</i>	18
2.4.1 <i>Supervised Learning</i>	18

2.4.2	<i>Unsupervised Learning</i>	19
2.4.3	<i>Semi-supervised Learning</i>	19
2.4.4	<i>Reinforcement Learning</i>	19
2.5	<i>Deep Learning</i>	20
2.6	<i>You Only Look Once (YOLO)</i>	21
2.7	Penelitian Terdahulu	27
BAB III METODE PENELITIAN		29
3.1	Tempat dan Waktu	29
3.2	Alat dan Bahan	29
3.3	Prosedur Penelitian	30
3.3.1	Studi Literatur	31
3.3.2	Pengumpulan	31
3.3.3	Anotasi <i>Dataset</i>	31
3.3.4	<i>Pre-processing</i> Data	32
3.3.5	Pemisahan/ <i>Splitting</i> Data	32
3.3.6	Deteksi Model Berbasis YOLO	32
3.3.7	<i>Evaluation</i> dan <i>Testing Model</i>	33
3.3.8	Interpretasi dan Analisis	33
3.4	Anggaran Biaya Penelitian	34
BAB IV HASIL DAN PEMBAHASAN		35
4.1.	Hasil Pengumpulan Data	35
4.2.	Anotasi <i>Dataset</i>	36
4.3.	Hasil <i>Preprocessing</i>	37
4.2.1	Hasil <i>Resizing</i>	37
4.2.1	Hasil Normalisasi Data	38
4.2.1	Hasil <i>Enhancement</i>	39
4.4.	Hasil <i>Splitting dataset</i>	40
4.5.	Hasil Deteksi Citra	41
4.6.	Analisa Perbandingan Hasil Performa Deteksi	42
BAB V KESIMPULAN DAN SARAN		49
5.1.	Kesimpulan	49
5.2.	Saran	49
DAFTAR PUSTAKA		51
LAMPIRAN		57

DAFTAR TABEL

Nomor	Judul Tabel	Halaman
3.1	Pendistribusian <i>dataset</i> yang digunakan dalam penelitian.	30
3.2	<i>Confusion matrix</i> .	33
4.1	Tabel analisis hasil deteksi citra.	42
4.2	Hasil <i>p-value</i> uji normalitas.	45
4.3	Hasil <i>mann-whitney u test</i> sebagai uji signifikansi.	46

DAFTAR GAMBAR

Nomor	Judul Gambar	Halaman
2.1	Arsitektur dinding saluran gastrointestinal (Biga dkk., 2019).	6
2.2	Saluran gastrointestinal (Shamsudhin dkk., 2017).	7
2.3	Perkembangan <i>colorectal polyp</i> (Hossain dkk., 2023).	8
2.4	Perbandingan ketiga citra (Florestiyanto dkk., 2022).	11
2.5	Algoritma <i>Adaptive Gamma Correction</i> (Rahman dkk., 2016).	14
2.6	Klasifikasi subkelas.	15
2.7	Tipe algoritma <i>machine learning</i> (Sarker, 2021).	18
2.8	Representasi model <i>deep learning</i> (Chollet, 2018).	20
2.9	Model YOLO (Redmon dkk., 2016).	21
2.10	Normalisasi titik tengah <i>bounding box</i> .	22
2.11	Penentuan <i>Intersection Over Union</i> (IOU) (Padilla dkk., 2020).	22
2.12	Penerapan <i>anchor box</i> (Alain dkk., 2023).	23
2.13	Arsitektur utama YOLO11 (Jegham dkk., 2024).	25
3.1	Skema penelitian.	30
4.1	Folder Kvasir-SEG.	35
4.2	Citra polip pada <i>dataset</i> RSUA.	36
4.3	Citra non-polip pada <i>dataset</i> RSUA, A: Edema, B: Patchy erythema, C: Diverticulae, D: Normal.	36
4.4	Hasil pelabelan <i>dataset</i> Kvasir-seg	37
4.5	Pelabelan <i>dataset</i> RSUA menggunakan Roboflow.	37
4.6	Proses <i>resizing dataset</i> , A: Kvasir-SEG, B: Polip RSUA, C: Non-polip RSUA.	38
4.7	Proses normalisasi <i>dataset</i> , A: Kvasir-SEG, B: Polip RSUA, C: Non-polip RSUA	39
4.8	Hasil <i>enhancement</i> menggunakan AGC pada, A: Polip RSUA, B: Non-polip RSUA, C: Kvasir-SEG.	40
4.9	<i>Directory</i> model YOLO11.	40
4.10	Contoh data.yaml YOLO11.	41
4.11	Hasil deteksi citra pada A: Kvasir-SEG, B: Polip RSUA, C: Non-polip RSUA.	41
4.12	Grafik perbandingan matriks evaluasi antar skenario.	43
4.13	Waktu pendeteksian antar skenario model.	47
4.14	<i>Dataset</i> dengan performa model yang kurang.	48

DAFTAR SINGKATAN

AGC	: <i>Adaptive Gamma Correction</i>
C2PSA	: <i>Cross Stage Partial with Spatial Attention</i>
C3k2	: <i>Cross Stage Partial with Kernel Size 2</i>
CNN	: <i>Convolutional Neural Network</i>
CRC	: <i>Colorectal Cancer</i>
CSP	: <i>Cross Stage Partial</i>
CTC	: <i>Computed Tomographic Colonography</i>
DNA	: <i>Deoxyribonucleic Acid</i>
EMR	: <i>Endoscopic Mucosal Resection</i>
FN	: <i>False Negative</i>
FP	: <i>False Positive</i>
GI	: <i>Gastrointestinal</i>
HR	: <i>Highlight Removal</i>
HSV	: <i>Hue, Saturation, and Value</i>
IOU	: <i>Intersection Over Union</i>
JSON	: <i>JavaScript Object Notation</i>
NMS	: <i>Non-Maximum Suppression</i>
RGB	: <i>Red, Green, and Blue</i>
RSUA	: <i>Rumah Sakit Universitas Airlangga</i>
SiLU	: <i>Sigmoid Linear Unit</i>
SPPF	: <i>Spatial Pyramid Pooling Fast</i>
Tb	: <i>Threshold Brightness</i>
Tc	: <i>Threshold Contrast</i>
TN	: <i>True Negative</i>
TP	: <i>True Positive</i>
WHO	: <i>World Health Organization</i>
YOLO	: <i>You Only Look Once</i>

DAFTAR LAMPIRAN

Nomor	Judul Lampiran
1.	Kodingan mengubah koordinat <i>bounding boxes</i> JSON ke .txt YOLO
2.	Kodingan fungsi AGC
3.	Kodingan fungsi HR
4.	Kodingan contoh penerapan AGC pada <i>dataset</i> Kvasir-SEG
5.	Kodingan contoh penerapan HR pada <i>dataset</i> Kvasir-SEG
6.	Kodingan contoh penerapan AGC-HR pada <i>dataset</i> Kvasir-SEG
7.	Kodingan contoh penerapan HR-AGC pada <i>dataset</i> Kvasir-SEG
8.	Kodingan fungsi <i>splitting dataset</i>
9.	Kodingan membuat data.yaml
10.	Kodingan memanggil model YOLO11
11.	Kodingan <i>training</i> model YOLO11
12.	Kodingan <i>validation</i> model YOLO11
13.	Kodingan <i>testing</i> model YOLO11
14.	Kodingan evaluasi performa model YOLO11
15.	Kodingan uji normalisasi dengan Shapiro-Wilk
16.	Kodingan uji signifikansi dengan <i>mann-whitney u test</i>

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saluran pencernaan manusia rentan terhadap berbagai kelainan mulai dari masalah kesehatan ringan hingga kanker yang mengancam jiwa. Sistem pencernaan terdiri dari saluran *gastrointestinal* (GI) dan organ-organ pencernaan, seperti hati, kantong empedu, dan pankreas. Saluran *gastrointestinal* juga dikenal sebagai saluran cerna, mencakup organ-organ yang dapat memproses makanan, seperti mulut, tenggorokan, kerongkongan, lambung, usus halus (*duodenum*, *jejunum*, dan *ileum*), usus besar atau *colon*, dan rektum. Penyakit saluran *gastrointestinal* seperti tukak lambung, pendarahan, *colorectal cancer*, tumor, dan penyakit lainnya sudah dikenal luas di seluruh dunia. Menurut *Pan American Health Organization*, pada tahun 2019 gangguan pencernaan telah menyumbang 375.170 kematian, yang mana terdiri dari 215.168 kematian pada pria dan 160.002 kematian pada wanita di kawasan Amerika (PAHO, 2021).

Berdasarkan data *World Health Organization* (WHO), pada tahun 2020 diperkirakan terdapat lebih dari 1,9 juta kasus baru *colorectal cancer* dan lebih dari 930.000 kematian dikarenakan *colorectal cancer* di seluruh dunia (World Health Organization, 2023). Di Indonesia, kasus *colorectal cancer* (CRC) telah menempati 4 besar kanker yang sering terjadi berdasarkan *Global Cancer Observatory* (GLOBOCAN) tahun 2022, yaitu mencapai 35.676 (8,7%) (Ferlay dkk., 2024). CRC disebabkan oleh pertumbuhan abnormal dalam kolon (bagian terpanjang dari usus besar) atau rektum (bagian akhir dari usus besar sebelum anus). Kebanyakan CRC berawal dari tumbuhnya *colorectal polyp* yang merupakan prekursor dari sebagian besar CRC. Polip ini sebenarnya bersifat jinak atau nonkanker, tetapi sebagian polip dapat mengalami mutasi DNA sehingga berkembang secara abnormal dan menyusup ke dalam submukosa untuk mengembangkan CRC (Shaukat dkk., 2020). Namun, jika polip tersebut terdeteksi sejak dini, polip tersebut dapat disembuhkan sepenuhnya sebelum berkembang menjadi kanker.

Colorectal polyp merupakan tonjolan atau benjolan yang tumbuh secara abnormal pada lapisan usus besar dan rektum saat tubuh memproduksi sel-sel yang tidak diinginkan dalam jumlah berlebihan. Bentuk polip bermacam-macam ada yang menyerupai tangkai kecil seperti jamur yang terdiri atas pertumbuhan mukosa di dalamnya dan ada yang berupa lesi datar. Polip berkembang perlahan seiring waktu. Keganasan *colorectal polyp* berkembang dari *benign polyp* yang terbentuk seiring waktu dan akhirnya bisa berkembang

menjadi CRC (Hossain dkk., 2023). Tahap lanjutnya dari *benign polyp* sebelum menjadi CRC yaitu polip ganas (*malignant polyp*) yang mencakup lesi datar yang telah mengalami invasi neoplastik pada submukosa tanpa perluasan ke muskularis propria (Shaukat dkk., 2020). Meskipun lesi yang hanya terbatas pada mukosa, lamina propria, dan muskularis mukosa bersifat prakanker, lesi tersebut tidak boleh dianggap sebagai kanker kolon invasif. Pendeteksian polip ganas lebih dini, dapat mencegah perkembangan keganasannya sehingga resiko pertumbuhan kanker dapat dihentikan.

Polip sering kali tidak menimbulkan gejala yang signifikan sehingga perlu adanya pencegahan terbaik untuk CRC. *Colonoscopy* merupakan salah satu bentuk *endoscopy* yang diakui secara luas dan dianggap efektif dalam pemeriksaan saluran GI, yang meliputi esofagus, lambung, duodenum, *colon*, dan rektum (Hossain dkk., 2023). *Colonoscopy* dilakukan dengan menggunakan alat berbentuk tabung fleksibel yang disebut kolonoskop, dimana dilengkapi kamera definisi tinggi di ujungnya dan memiliki saluran tambahan yang digunakan untuk memasukkan peralatan serta cairan pembersih lensa kolonoskop dan mukosa *colon* (Stauffer & Pfeifer, 2023). *Colonoscopy* dilakukan melalui rektum, usus besar, dan *colon*. *Colonoscopy* digunakan dalam memandu langkah-langkah pengobatan kanker, termasuk dalam perencanaan intervensi bedah. Oleh karena itu, perlu dilakukan pemeriksaan dengan *colonoscopy* untuk *screening* dan pendeteksian dini *colorectal polyp*. Dalam praktik klinisnya, pemeriksaan *colonoscopy* dilakukan secara manual melalui observasi langsung oleh ahli gastroenterologi. Proses ini memakan waktu yang lama, rentan terhadap kelelahan, dan bergantung pada perbedaan antara pengamat dan konsistensi individu pengamat (Hossain dkk., 2023).

Perkembangan teknologi yang pesat, khususnya dalam bidang kesehatan, telah membawa perubahan yang cukup besar pada berbagai metode diagnostik, termasuk *colonoscopy*. Penerapan *Artificial Intelligence* dapat membantu dokter dalam mendeteksi letak *polyp*. Pendeteksian ini dilakukan dengan citra medis yang dihasilkan dari pemeriksaan *colonoscopy*. Teknologi medis yang didukung dengan kecanggihan *Artificial Intelligence* dapat membantu meringankan kinerja dokter dan ahli gastroenterologi, dengan pelatihan-pelatihan khusus yang diterapkan dapat meningkatkan kinerja teknologi medis dalam mendeteksi penyakit *colorectal polyp* (Yasmin dkk., 2023). Algoritma yang digunakan, menjadi metode akurat dalam mendeteksi dan mengevaluasi keberagaman citra medis dalam jumlah data yang besar.

Dalam penanganan *colorectal polyp*, deteksi lokasi yang akurat sangat penting dalam menentukan jenis penanganan yang tepat, terutama untuk tindakan pembedahan. Deteksi berfokus pada identifikasi lokasi keberadaan polip secara keseluruhan yang mana dapat memungkinkan para ahli untuk mengetahui letak polip yang memerlukan intervensi dengan mudah. Pengidentifikasian lokasi polip yang tepat serta komputasi model yang lebih singkat menjadikan pertimbangan dalam pemilihan sistem deteksi di mana dapat mendukung keputusan bedah dan membantu mengurangi waktu analisis serta risiko komplikasi.

Pada tahun 2023, Elkarazle dkk. melakukan perbandingan dalam menentukan keunggulan metode deteksi *colorectal polyp* dari citra *colonoscopy*. Pada penelitian ini disebutkan bahwa adanya faktor kelelahan dan kurangnya pengalaman ahli medis dalam melakukan prosedur *colonoscopy* sehingga *colorectal polyp* tidak teridentifikasi oleh ahli yang menyebabkan komplikasi lebih lanjut dan berkembang menjadi *colorectal cancer* (CRC). Pada kasus ini, ahli medis mengalami kesulitan dalam mengidentifikasi polip yang terletak di sisi kanan usus besar, dikenal sebagai usus besar proksimal, karena bentuk dari usus besarnya sendiri. Selain itu, kondisi citra *colonoscopy* yang terkena pantulan cahaya yang berlebih serta keburaman juga menjadi salah satu penyebab kesulitannya ahli medis dalam mengidentifikasi polip.

Pembangunan sistem deteksi yang dilakukan oleh Elkarazle dkk., menggunakan pemodifikasian metode YOLO. Dari penelitian ini, menghasilkan *precision* sebesar 90.61, *recall* sebesar 91.04, dan *F1 score* sebesar 90.82. Selain itu, pada bulan April 2024, Lalinia & Sahafi juga melakukan penelitian mengenai pembangunan sistem deteksi *colorectal polyp* melalui perbandingan semua versi YOLOv8 sehingga hasil terbaik didapatkan dengan YOLOv8m. Penelitiannya diharapkan mampu meminimalkan diagnosis yang terlewat, memfasilitasi deteksi dini, dan berkontribusi pada pencegahan CRC. Dari hasil akhir deteksi, dihasilkan presisi 95,6%, *recall* 91,7%, dan skor F1 92,4% serta mAP50 sebesar 85,4% dan mAP50-95 sebesar 62% dengan waktu inferensi 10,6 milidetik dan 25 juta parameter yang menunjukkan keseimbangan yang terpuji antara akurasi dan efisiensi komputasi.

Berdasarkan urgensi dan penelitian yang telah dipaparkan di atas, penulis melakukan pengaplikasian *You Only Look Once* (YOLO) yang telah terbukti efektif dalam pendeteksian objek serta *image enhancement* sebagai bentuk perbaikan kualitas citra. Desain deteksi YOLO mengintegrasikan kecepatan dan akurasi melalui satu jalur dalam jaringan saraf konvolusional (Redmon dkk., 2016). Pengaplikasian YOLO dan *image enhancement* diharapkan dapat mendeteksi *colorectal polyp* menggunakan citra *colonoscopy* dengan

bounding box secara efisien. Penelitian ini diharapkan dapat mendeteksi *colorectal polyp* dengan *bounding box* sebagai indikasi letak tonjolan polip.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan di atas, didapatkan rumusan masalah sebagai berikut:

1. Bagaimana cara membangun sistem deteksi letak tonjolan *colorectal polyp* dengan implementasi metode YOLO11?
2. Bagaimana performa *machine learning* dalam mendeteksi letak tonjolan *colorectal polyp* dengan implementasi metode YOLO11 berdasarkan analisa *accuracy*, *recall*, dan *precision*?
3. Bagaimana pengaruh pemberian *enhancement* dalam performa YOLO11 untuk mendeteksi *colorectal polyp*?

1.3 Tujuan Penelitian

Dari rumusan masalah yang telah dipaparkan di atas, didapat tujuan sebagai berikut:

1. Mengetahui cara membangun sistem deteksi letak tonjolan *colorectal polyp* dengan implementasi metode YOLO11.
2. Mengetahui performa sistem deteksi letak tonjolan *colorectal polyp* menggunakan metode YOLO11 berdasarkan analisa *accuracy*, *recall*, dan *precision*.
3. Mengetahui pengaruh pemberian *enhancement* dalam performa YOLO11 untuk mendeteksi *colorectal polyp*.

1.4 Manfaat Penelitian

1.4.1 Manfaat Teoritis

Menjadi referensi untuk mengembangkan pendeteksian penyakit khususnya *colorectal polyp* dengan bantuan *artificial intelligence*.

1.4.2 Manfaat Praktis

Meningkatkan pelayanan kesehatan khususnya dalam membantu ahli medis untuk mendiagnosa *colorectal polyp* dengan cepat melalui *artificial intelligence*.

1.5 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut:

1. Penelitian menggunakan data sekunder berupa *colonoscopy image* yang diperoleh dari Rumah Sakit Universitas Airlangga dan bank *Simula Datasets* karena keterbatasan *resources*.

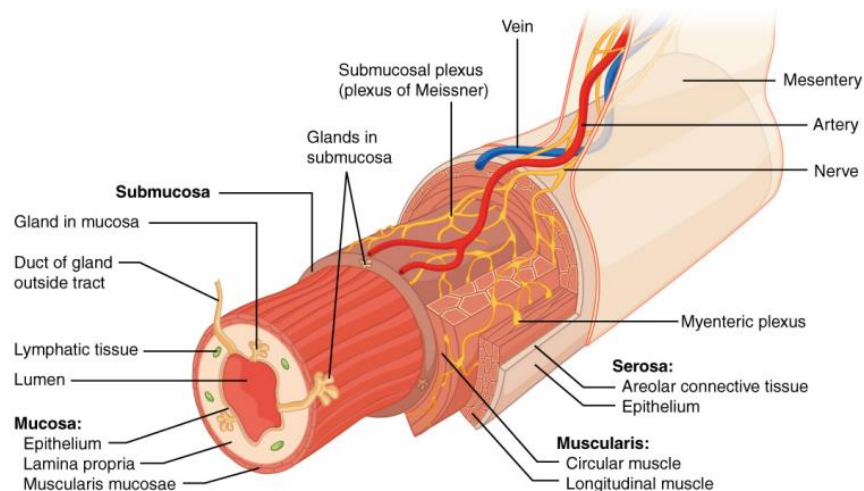
2. Penelitian menggunakan *dataset* yang memiliki 1 *groundtruth* sehingga fokus penelitian hanya pada deteksi 1 *bounding box*.
3. Penelitian memiliki output berupa *bounding box* sebagai hasil deteksi letak *colorectal polyp* tanpa diikuti klasifikasi kelas lain.
4. Arsitektur YOLO yang dibangun menggunakan *google colaboratory* dengan bahasa pemrograman *python*.
5. Penelitian ini melakukan perbandingan terkait performa sistem setelah dan sebelum pemberian *enhancement*.

BAB II

STUDI PUSTAKA

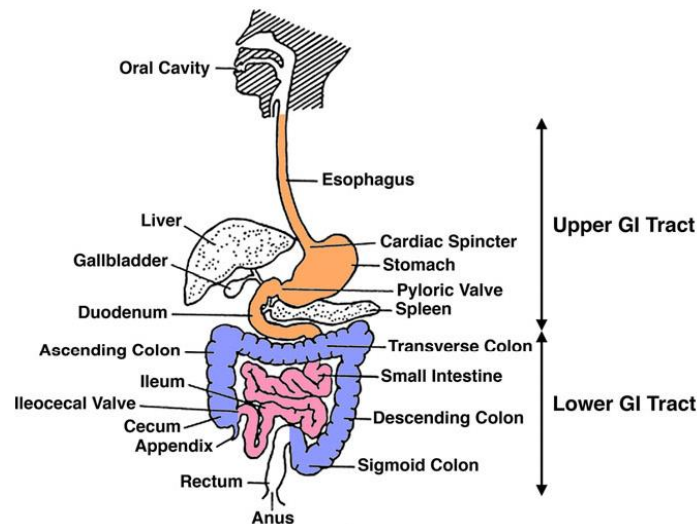
2.1 Saluran *Gastrointestinal*

Saluran *gastrointestinal* (GI) merupakan bagian dari sistem pencernaan. Saluran ini meliputi mulut, faring (tenggorokan), esofagus, lambung, usus halus, usus besar, rektum, dan anus. Sistem GI menjalankan berbagai fungsi otonom yang kompleks dan terkoordinasi dalam skala panjang dan waktu. Fungsi utamanya mencakup pencernaan, penyerapan, ekskresi, serta perlindungan, yang dilakukan oleh sejumlah organ dengan peran spesifik, mulai dari mulut hingga anus (Hall & Hall, 2021). Saluran pencernaan mulai berkembang pada minggu ketiga kehamilan. Pada tahap ini, terjadi proses gastrulasi, di mana tiga lapisan embrionik (ektoderm, mesoderm, dan endoderm) terbentuk. Lapisan endoderm membentuk epitel saluran pencernaan, serta organ seperti kantong empedu, pankreas, dan hati. Mesoderm membentuk jaringan ikat dan otot polos dinding saluran cerna. Sementara itu, ektoderm terbagi menjadi ektoderm permukaan, saluran saraf, dan krista saraf (Ogobuiro dkk., 2023).



Gambar 2.1 Arsitektur dinding saluran *gastrointestinal* (Biga dkk., 2019).

Saluran *gastrointestinal* terbagi menjadi empat lapisan, yaitu mukosa (epitel, lamina propria, dan mukosa muskular), submukosa, muskularis propria (lapisan otot melingkar bagian dalam, ruang intermuskular, dan lapisan otot longitudinal luar), dan serosa dapat dilihat pada gambar 2.1 (Rao & Wang, 2010).



Gambar 2.2 Saluran *gastrointestinal* (Shamsudhin dkk., 2017).

Perkembangan teknologi pangan dan gaya hidup masyarakat dapat mempengaruhi penyebaran penyakit yang berkembang. Saluran GI menjadi salah satu organ tubuh manusia yang terdampak dari perkembangan tersebut. Dalam mempermudah penggolongan penyakit, saluran *gastrointestinal* diklasifikasikan menjadi dua, yaitu *upper gastrointestinal tract* dan *lower gastrointestinal tract*. *Upper gastrointestinal tract* terdiri atas rongga mulut, kelenjar ludah, esofagus, lambung, dan usus halus. Fungsi bagian ini mencakup pengangkutan bolus makanan, pencernaan enzimatik, penyerapan nutrisi, serta sebagai perlindungan terhadap lingkungan eksternal (Treuting et al., 2018). Sedangkan *lower gastrointestinal tract* meliputi *secum*, usus buntu, usus besar, rektum, dan anus. Fungsi utama bagian ini adalah mendehidrasi dan menyimpan bahan feses (Treuting et al., 2018)

2.2 Colorectal Polyp

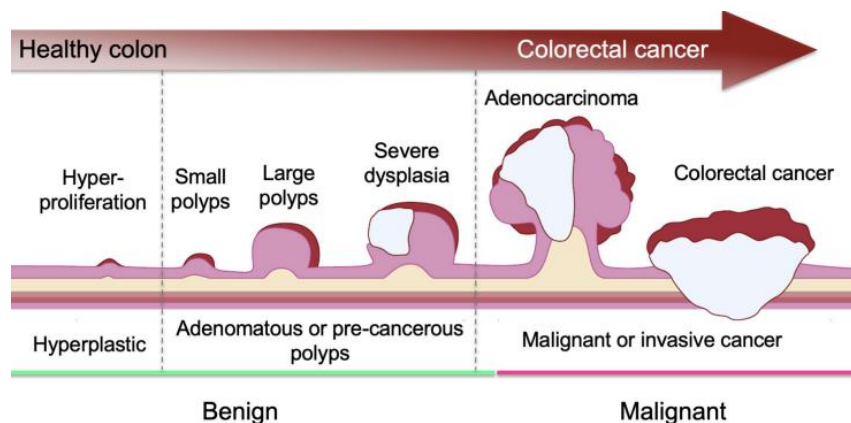
Colorectal polyp merupakan tonjolan atau benjolan yang tumbuh secara abnormal pada lapisan usus besar dan rektum saat tubuh memproduksi sel-sel yang tidak diinginkan dalam jumlah berlebihan. Umumnya, polip bersifat jinak dan tidak berbahaya. Namun, jika tidak ditangani sejak dini, polip dapat berpotensi berkembang menjadi kanker seiring berjalannya waktu. Polip dapat menjadi lebih agresif dan invasif sehingga berakibat fatal jika ditemukan pada stadium lanjut (Hossain dkk., 2023). Banyak faktor yang dapat menjadi penyebab timbulnya penyakit ini, seperti pola makan, gaya hidup, usia, dan jenis kelamin. Orang yang kelebihan berat badan, berusia di atas 50 tahun, merokok, atau memiliki riwayat keluarga polip biasanya cenderung memiliki risiko yang lebih tinggi. Namun, risiko ini dapat diminimalisir dengan pendeteksian pada tahap awal.

2.2.1 Gejala

Pasien *colorectal polyp*, biasanya jarang mengalami gejala yang spesifik. Pada fase awal, kebanyakan pasien tidak menimbulkan keluhan atau gejala sama sekali. Namun, terdapat beberapa gejala yang kemungkinan dapat terjadi seperti pendarahan rektum, nyeri perut, dan obstruksi jika ditemukan polip dengan ukuran yang sangat besar. Pendarahan rektum, biasanya tersembunyi dan merupakan keluhan yang paling sering terjadi. Sedangkan kram atau nyeri perut terjadi pada lesi atau tonjolan polip yang besar. Sementara itu, gejala obstruksi yang dialami pasien polip dapat berupa nyeri perut, pembengkakan, atau perubahan kebiasaan (Sayuti, 2019).

2.2.2 Jenis Polip

Polip memiliki variasi bentuk, ukuran, dan jenis permukaannya. Berdasarkan bentuknya, polip dibagi menjadi dua macam yaitu polip bertangkai dan *sessile polyp*. Polip yang menempel pada permukaan melalui tangkai panjang dan tipis, disebut polip bertangkai. Namun, jika tidak memiliki tangkai disebut polip *sessile*. Berdasarkan asalnya, polip terbagi menjadi polip epitel dan non-epitel. Secara histopatologi, *colorectal polyp* dapat diklasifikasikan menjadi beberapa kategori, polip neoplastik dan non-neoplastik. Jenis polip neoplastik biasanya terdapat polip adenomatosa dan polip displasia darah, jenis ini memiliki potensi menjadi kanker tetapi sebagian besarnya bersifat jinak saat pertama kali terdeteksi. Sedangkan polip non-neoplastik, seperti polip hiperplastik mukosa, inflamasi, dan hamartomatosa, tidak memiliki resiko keganasan (Basnet dkk., 2021).



Gambar 2.3 Perkembangan *colorectal polyp* (Hossain dkk., 2023).

Keganasan *colorectal polyp* terdapat dua fase, dilihat pada gambar 2.3, yaitu *benign polyp* yang terdiri dari polip non-neoplastik dan neoplastik serta *malignant polyp* yang terdiri dari CRC invasif (polip adenokarsinoma). Hiperproliferasi dianggap sebagai tahapan dari hiperplastik. Polip ini memiliki ukuran yang kecil dan sering ditemukan di ujung usus besar. Polip hiperplastik memiliki pembuluh yang lebih terang dibandingkan dengan area

sekitarnya serta pada permukaannya terlihat pola melingkar dengan titik kecil yang dikelilingi mukosa terang. Selanjutnya pada tahapan polip adenomatosa atau polip prakanker, jika semakin besar polip akan semakin memungkinkan berkembang menjadi kanker. Oleh karena itu, ahli urologi menyarankan tindakan pengangkatan polip yang berukuran lebih besar dari 5 mm untuk menghindari CRC di masa mendatang (Hossain dkk., 2023). Polip adenomatosa memiliki ciri pembuluh yang berwarna coklat tua dengan area tengah yang lebih terang. Sebagian besar, polip ini memiliki permukaan oval dengan dikelilingi pembuluh berwarna coklat. Setelah melewati tahap adenomatosa, keganasan sel-sel dalam *colorectal polyp* berkembang menjadi CRC ganas dan kemudian menjadi invasif. Adenokarsinoma merupakan jenis *malignant polyp* yang berpotensi menyebar dan menimbulkan komplikasi kesehatan yang berbahaya jika tidak ditangani dengan cepat.

2.2.3 Diagnosis

Dalam mendiagnosis adanya pertumbuhan *colorectal polyp*, perlunya pemeriksaan *screening* sehingga dapat menurunkan angka kematian akibat penyakit polip. Proses *screening* yang ideal yaitu layak, akurat, aman, dapat diterima, dan hemat biaya (Huck & Bohl, 2016). Salah satu proses diagnosis yang paling umum, yaitu *colonoscopy*. *Colonoscopy* merupakan salah satu bentuk *endoscopy* yang diakui secara luas dan dianggap efektif dalam pemeriksaan saluran GI. *Colonoscopy* menjadi metode yang pembanding dengan semua metode *screening* lain. Keuntungan kolonoskopi meliputi sensitivitas tinggi, kemampuan untuk mengangkat lesi saat terdeteksi, dan akses penuh ke bagian proksimal dan distal kolon (Li dkk., 2021). Metode ini memungkinkan deteksi dan eksisi lesi premaligna dari seluruh kolon dan rectum (Huck & Bohl, 2016). Jika metode utama untuk CRC *screening* pada pasien dengan risiko rata-rata adalah *colonoscopy*, tindak lanjut dan pemantauan jangka panjang akan bergantung pada hasil *colonoscopy* pertama dan evaluasi histopatologi yang relevan.

Pada tindakan *colonoscopy* terdapat tiga tahap pelaksanaan yaitu sebelum, selama, dan sesudah. Sebelum proses *colonoscopy* berlangsung, terdapat persiapan usus yang perlu dilakukan guna memperlancar proses tersebut. Persiapan usus yang buruk dapat menyebabkan kegagalan proses *colonoscopy* (Ahmad dkk., 2023). Efektifitas persiapan usus sangat berdampak pada hasil, kualitas, dan efisiensi prosedur *colonoscopy*. Oleh karena itu, diharapkan lebih dari 90% adanya persiapan usus yang memadai. Proses persiapan usus dilakukan dengan mengosongkan usus dengan bantuan obat dokter selama sehari sebelum prosedur *colonoscopy*. Selain itu, diet rendah serat selama dua atau tiga hari juga dapat

dilakukan selama proses persiapan usus. Kemudian selama proses *colonoscopy*, kolonoskop akan dimasukkan ke dalam usus besar melalui anus. Saat bergerak maju, kateter memompa udara ke dalam usus besar agar mengembang. Hal ini dapat menyebabkan efek nyeri karena tekanan gas pada pasien (Cleveland Clinic, 2022). Sementara itu, kamera mentransmisikan video usus besar ke monitor. Sesudah proses *colonoscopy* dilakukan, pasien akan dilakukan *monitoring* kesehatan sekaligus menunggu hasil pemeriksaan *colonoscopy* dalam beberapa hari.

Dalam perkembangan teknologi, terdapat teknologi diagnosis yang lebih canggih dalam mendiagnosis *colorectal polyp*, yaitu *Computed Tomographic Colonography* (CTC) yang disebut juga sebagai *virtual colonoscopy*. Teknologi ini merupakan prosedur non-invasif yang memberikan detail citra usus besar dengan menggunakan CT scan. Persiapan usus berkualitas tinggi diperlukan dalam visualisasi dinding kolon, karena bahan feses yang tertahan dapat mengaburkan polip pada hasil pencitraan. Selain itu, obesitas pasien juga dapat mengakibatkan pembatasan penggunaan CTC karena kemungkinan muat dan pemindahan meja CT sangat susah (Huck & Bohl, 2016). Jika polip atau kelainan ditemukan selama pemeriksaan, *colonoscopy* mungkin tetap diperlukan untuk tindakan lanjutan, yang dapat meningkatkan biaya keseluruhan dalam penggunaan teknologi CTC. (Van der Meulen dkk., 2018)

2.2.4 Penanganan

Beberapa diagnosis yang telah dilakukan dalam upaya pendeteksian dini *colorectal polyp* memerlukan pelatihan dan keterampilan yang cukup memiliki penguasaan. *Polypectomy colonoscopy* merupakan salah satu alternatif pengobatan untuk pengangkatan *colorectal polyp*, yang meliputi *snare polypectomy*, biopsi forsep panas, dan biopsi eksisi. *Snare polypectomy* umumnya digunakan untuk polip bertangkai, sedangkan untuk polip *sessile* kecil dapat diangkat menggunakan biopsi forsep panas atau biopsi eksisi (Kaltenbach dkk., 2020). Teknik lain yang juga dapat dilakukan adalah *Endoscopic Mucosal Resection* (EMR), yaitu polipektomi dengan bantuan injeksi salin (*saline assisted polypectomy*). Alternatif lain termasuk koagulasi plasma argon dan ablasi laser, yang sering digunakan untuk menghilangkan banyak polip kecil yang ditemukan pada gangguan hereditas atau polip *sessile* besar dengan sisa jaringan. Prosedur ini efektif, cepat, dan relatif aman jika dilakukan oleh ahli yang berpengalaman.

Hasil pendeteksian pasien yang terdiagnosa *malignant polyp*, paling baik ditangani dengan reseksi bedah karena risiko *residual cancer* atau metastasis kelenjar getah bening

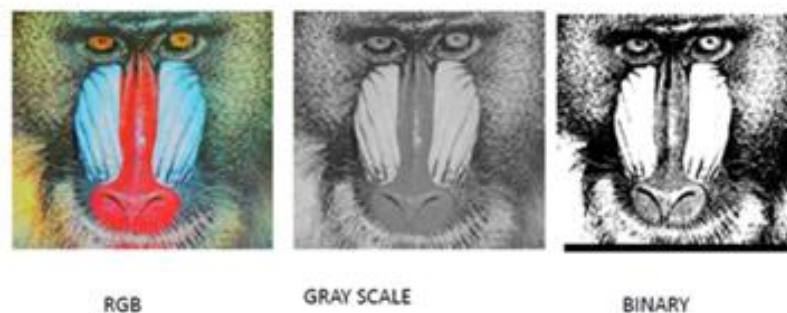
sangat tinggi dialami (Shaukat dkk., 2020). Jenis pengobatan ini menjadi pilihan terakhir jika terdapat ciri-ciri keganasan yang telah mencapai akar polip atau tidak terlihat adanya batas tegas antara area normal dan ganas. Tindakan ini dilakukan jika *polypectomy colonoscopy* tidak dapat dilakukan (polip *sessile* berukuran besar) atau adanya keraguan apakah polip yang telah diangkat seluruhnya melalui *colonoscopy* memiliki ciri-ciri histologis yang menunjukkan diferensiasi adenokarsinoma yang buruk.

2.3 Pengolahan Citra Digital

2.3.1 Citra Digital

Citra digital merupakan tampilan visual dalam format digital yang terbentuk dari kumpulan piksel, di mana setiap pikselnya memiliki nilai numerik tertentu (Dijaya, 2023). Citra digital dapat diperoleh melalui perangkat yang mengubah objek nyata atau data menjadi format digital, seperti kamera digital, *scanner*, atau simulasi komputer. Berdasarkan gambar 2.4, terdapat tiga jenis citra digital, yaitu citra berwarna yang terdiri dari piksel-piksel berwarna, citra hitam putih yang hanya terdiri atas piksel hitam dan putih, serta citra *grayscale* yang terdiri dari piksel-piksel bercorak abu-abu.

Citra hitam putih atau biner merupakan representasi citra digital dengan dua jenis warna, yaitu hitam dan putih yang diwakili oleh angka biner (0 dan 1). Sementara citra *grayscale* direpresentasikan dengan menggunakan skala derajat keabuan yang mana hasil pemangkatan nilai bit yang ada terhadap angka 2 (2^n) (Florestiyanto dkk., 2022). Sedangkan citra berwarna terdiri dari piksel-piksel yang berisi tiga angka sesuai dengan tingkat merah, hijau, dan biru (*Red, Green, Blue* atau RGB). Ketiga tingkatan warna tersebut merupakan warna primer aditif yang dapat digunakan untuk membuat warna apapun dengan campuran ketiga warna tersebut. Warna primer memiliki penyimpanan informasi visual sebesar 8 bit (1 *byte*) dengan rentang intensitas 0 hingga 255 (Florestiyanto dkk., 2022).



Gambar 2.4 Perbandingan ketiga citra (Florestiyanto dkk., 2022).

Secara sistematis, citra digital pada umumnya digunakan dalam bidang pengolahan citra sebagai suatu fungsi kontinu dari intensitas cahaya $f(x,y)$ yang berisi baris M dan kolom

N dengan (x,y) adalah koordinat diskrit, sesuai dengan persamaan 2.1. Representasi skala abu-abu dinyatakan sebagai triplet $f(x,y,z)$ dimana x dan y adalah koordinat spasial dan z adalah nilai fungsi pada koordinat (x,y) (Gonzalez & Woods, 2018).

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

Dalam visualisasiannya, citra digital memiliki beberapa elemen pendukung yaitu *brightness* yang menggambarkan intensitas cahaya yang dipancarkan piksel dalam citra yang dapat ditangkap oleh sistem penginderaan, *contrast* sebagai sebaran terang-gelap sebuah citra, *contour* yang menggambarkan keadaan atas adanya perubahan intensitas pada piksel-piksel yang bertetangga sehingga memungkinkan dalam penginderaan tepi-tepi objek di dalam citra, *color* sebagai persepsi yang ditangkap sistem visual terhadap panjang gelombang cahaya yang dipantulkan oleh citra, *shape* yang menyatakan *property intrinsic* dari objek 3 dimensi, dan *texture* sebagai distribusi spasial dari derajat keabuan dalam sekumpulan piksel yang bertetangga (Florestiyanto dkk., 2022).

2.3.2 Sistem Pengolahan

Pengolahan citra (*image processing*) merupakan proses mengolah piksel-piksel citra untuk tujuan tertentu. Proses ini melibatkan serangkaian operasi yang bertujuan untuk meningkatkan kualitas citra, mengekstraksi informasi penting, dan mempersiapkan citra dalam proses pengaplikasiannya (Dijaya, 2023). Praproses citra merupakan tahapan yang sangat penting dalam pengolahan citra, di mana citra asli diperbaiki dan ditingkatkan sebelum tahap lanjutan. Ada beberapa tahap dalam praproses citra, seperti normalisasi dan perbaikan kualitas.

Proses normalisasi adalah merupakan sebuah proses yang digunakan untuk mempersiapkan data agar bisa digunakan untuk memenuhi kebutuhan pemakai dan mendukung proses berikutnya agar mendapatkan hasil yang lebih baik. Inputan data yang tidak sama dapat mempengaruhi hasil akhir pada proses pengolahannya. Ada beberapa teknik normalisasi yang tersedia, salah satunya yaitu *Min-Max Normalization*. *Min-Max Normalization* merupakan teknik yang menyediakan transformasi linear pada rentang data asli. Teknik ini mengubah skala fitur dari rentang berapapun menjadi rentang yang baru, $[0, 1]$ atau $[-1, 1]$. Penerapan normalisasi ini dapat membantu mempercepat proses

pembelajaran dalam *layers* model dan membantu dalam konvergen lebih cepat (García dkk., 2015).

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\max_A' - \min_A') + \min_A' \quad (2.2)$$

Dengan :

v' = nilai hasil normalisasi

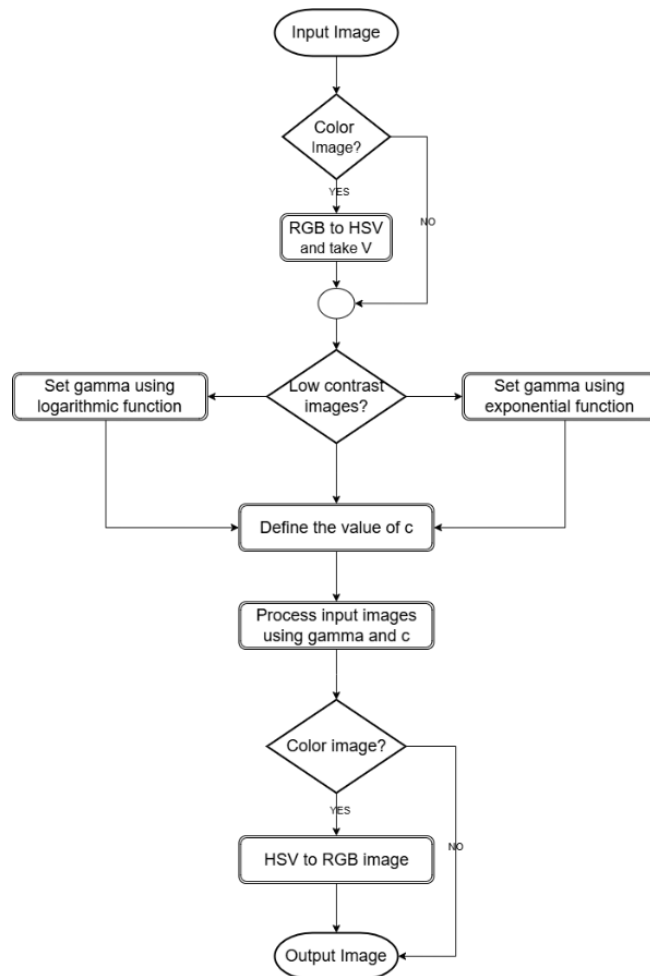
v = nilai awal

\min_A, \max_A = rentang awal

\min_A', \max_A' = rentang baru

Kualitas citra digital sering kali kurang optimal yang dapat mempengaruhi proses pengolahannya. Kualitas yang rendah dapat disebabkan oleh beberapa faktor, seperti *noise image*, resolusi rendah, kurangnya kontras, hingga pencahayaan yang kurang merata. Oleh karena itu, diperlukan proses *enhancement* atau peningkatan kualitas citra untuk membantu memperjelas detail visualnya. *Image enhancement* merupakan salah satu proses terpenting dalam pemrosesan citra digital. Secara umum, prinsip kerjanya adalah dengan memodifikasi informasi suatu citra menjadi lebih sesuai untuk metode pemrosesan tertentu (Qi dkk., 2022). Dalam proses *enhancement* terdapat beberapa metode yang dapat dilakukan, di antaranya yaitu *Adaptive Gamma Correction* dan *Highlight Removal* yang mana keduanya digunakan untuk perbaikan *brightness* pada citra.

Adaptive Gamma Correction (AGC) merupakan metode perbaikan kualitas citra dengan meningkatkan kontras citra dan menjaga kecerahannya, dimana penentuan parameter yang digunakan didapatkan berdasarkan informasi gambar masukan seperti pada gambar 2.5 (Rahman dkk., 2016). Dalam proses AGC, digunakan *color transformation* dengan mengubah citra input masukan menjadi HSV (*Hue, Saturation, and Value Channels*). HSV memiliki kelebihan dalam memisahkan informasi warna dan kecerahan (*brightness*) menjadi saluran rona (*Hue*), saturasi (*Saturation*), dan nilai (*Value*) suatu gambar. Pengubahan ini bertujuan untuk mencegah peningkatan berlebih dengan AGC pada *V channel* untuk menghindari distorsi warna citra dengan mempertahankan rona dan saturasi (Nie dkk., 2023).



Gambar 2.5 Algoritma *Adaptive Gamma Correction* (Rahman dkk., 2016).

Setiap gambar masukan memiliki karakteristik informasi masing-masing. Oleh karena itu, dalam proses AGC dilakukan pengklasifikasian gambar masukan berdasarkan kontras menjadi *low-contrast* (q_1) dan *high-contrast* (q_2). Hal ini digunakan untuk menentukan perlakuan apa yang sesuai terhadap masing-masing variasi citra yang diproses sehingga hasilnya seragam.

$$g(I) = \begin{cases} q_1, & D \leq \frac{1}{\tau} \\ q_2, & \text{lainnya} \end{cases} \quad (2.3)$$

Dengan :

$$D = (\mu + 2\sigma) - (\mu - 2\sigma) = 4\sigma$$

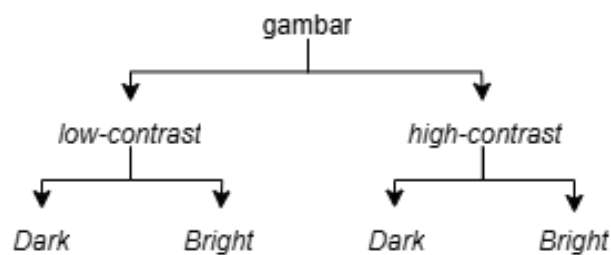
τ = parameter untuk menentukan kontras gambar (bernilai 3)

μ = intensitas rata-rata gambar

σ = intensitas standar deviasi gambar

Pada penelitian ini dilakukan modifikasi pada parameter klasifikasi persamaan 2.3 yaitu dari $D((\mu + 2\sigma) - (\mu - 2\sigma)) \leq 1/\tau$ menjadi $\sigma \leq 1/\tau$. Pada persamaan tersebut, mengklasifikasikan gambar sebagai gambar berkontras rendah ketika intensitas pikselnya berada dalam rentang yang kecil. Bentuk sederhana dari kriteria gambar yang diklasifikasikan sebagai *low-contrast* yaitu $4\sigma \leq 1/\tau$. Oleh karena itu, modifikasi dilakukan karena dalam banyak literatur, σ digunakan sebagai indikator utama untuk kontras global karena sudah dapat menyatakan persebaran rata-rata nilai piksel (Alfiyan, 2020)

Selain berdasarkan kontras, pada setiap kelas dilakukan pembagian subkelas, *bright* dan *dark* sehingga pembagian kelas menjadi seperti berikut.



Gambar 2.6 Klasifikasi subkelas.

Perbaikan kualitas AGC menggunakan koreksi gamma tradisional dilakukan seperti berikut

$$I_{out} = cI_{in}^{\gamma} \quad (2.4)$$

Dengan I_{out} dan I_{in} adalah intensitas gambar masukan dan keluaran serta c dan γ merupakan parameter kontrol dalam AGC. Berdasarkan klasifikasi pada persamaan 2.3, gambar yang masuk dalam kelas $q1$ (*low – contrast*) memiliki σ yang berarti sebagian besar piksel memiliki intensitas yang sama. Sehingga, perlu adanya penyebaran pada rentang yang lebih luas untuk meningkatkan kontras. Dalam proses ini, penggunaan γ dalam *gamma correction* sangat berpengaruh karena semakin tinggi γ maka persebaran intensitas semakin luas dan kontras semakin tinggi dengan perhitungan berikut.

$$\gamma = -\log_2(\sigma) \quad (2.5)$$

Perhitungan standar deviasi pada gambar digunakan untuk mengukur kontras gambar. Sedangkan, gambar yang masuk dalam kelas $q2$ (*high – contrast*) memiliki penyebaran intensitas yang signifikan sehingga penyesuaian kecerahan biasanya lebih penting daripada peningkatan kontras pada gambar. Oleh karena itu, adanya perbedaan perhitungan γ agar tidak memberikan banyak persebaran kontras menjadi sebagai berikut.

$$\gamma = \exp \left[\frac{(1 - (\mu + \sigma))}{2} \right] \quad (2.6)$$

Dengan $\gamma \in [0.90, 1.65]$, dengan artian nilai γ berada pada rentang kecil sekitar 1 sehingga sedikit pengaruh pada kontras gambar.

Parameter c digunakan sebagai faktor pencerah dan penggelapan intensitas gambar keluaran. Penggunaan parameter ini digunakan untuk mengklasifikasi gambar menjadi subkelas *bright* dan *dark* dengan perhitungan berikut.

$$c = \frac{1}{1 + \text{Heaviside}(0.8 - \mu) \times (k - 1)} \quad (2.7)$$

Dengan :

$$k = I_{in}^{\gamma} (1 - I_{in}^{\gamma}) \times \mu^{\gamma}$$

$$\text{Heaviside}(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases}$$

Pengklasifikasian subkelas dilakukan dengan syarat intensitas rata-rata gambar (μ) ≥ 0.8 maka akan masuk subkelas *bright* sebaliknya akan masuk subkelas *dark*.

$$c = \begin{cases} 1, & \mu \geq 0.8 \\ \frac{1}{1 + \text{Heaviside}(0.8 - \mu) \times (k - 1)}, & \text{lainnya} \end{cases} \quad (2.8)$$

Sementara itu, *image enhancement* dengan perbaikan *brightness* lain dapat dilakukan dengan menggunakan *Highlight Removal* (HR). *Highlight Removal* menggunakan prinsip penghapusan area pantulan cahaya tertentu. Pantulan cahaya merupakan masalah yang cukup krusial sehingga dapat mengganggu proses pendeteksian dalam medis terutama selama proses *colonoscopy* (Zhang dkk., 2022). Syarat dalam terbentuknya suatu pantulan yaitu adanya intensitas cahaya tinggi yang ditandai dengan saturasi putih atau minimal ada satu saluran warna yang tersaturasi jenuh (R,G, atau B). Oleh karena itu, proses HR digunakan dua model warna yaitu *grayscale* yang mewakili analisa intensitas dan RGB dengan pemrosesan algoritma logika OR untuk menentukan daerah *absolute highlight* dan *relative highlight*. Penggunaan *grayscale* dalam pendeteksian *absolute highlight* dikarenakan kemudahan dalam penentuan saturasi putih dalam keluaran endoskopi (Shen dkk., 2020). Apabila nilai *grayscale* lebih dari *threshold* kontras atau kurang dari *threshold* kecerahan, maka piksel tersebut terdeteksi sebagai *absolute highlight* (Wu & Su, 2017). Sedangkan pada keluaran endoskopi sering terjadi *misalignment color* karena gerakan lensa

yang cepat, akibatnya pantulan yang terbentuk tampak berwarna merah, hijau, atau biru yang sangat jenuh. Oleh karena itu, perlu pendeteksian *relative highlight* menggunakan RGB *channels* dengan syarat nilai G atau B lebih dari *threshold* intensitas. Dalam kasus endoskopi, gambar keluaran yang dihasilkan memiliki warna yang berdominan terhadap *red channels* sehingga nilai intensitas R tidak digunakan karena mendekati saturasi gambar (Yudhikar, 2021).

Adanya penggunaan *adaptive thresholding*, yang berarti nilai *threshold* (T_c , T_b , T_2) disesuaikan secara dinamis berdasarkan statistik gambar seperti intensitas rata-rata (μ) dan standar deviasi (σ). Hal ini dapat menyebabkan kinerja metode ini lebih kuat terhadap variasi kondisi pencahayaan dan kontras dalam gambar *colonoscopy*. Intensitas rata-rata digunakan untuk mengukur rata-rata kecerahan dalam setiap gambar. Sedangkan, standar deviasi digunakan untuk mengukur kontras setiap gambar.

$$T1 = \min_{intensity} + \left(\max_{intensity} - \min_{intensity} \right) \times 0.85 \quad (2.9)$$

$$T_c = T_{c_{base}} + 0.08 \times \left(\frac{\sigma}{255} \right) \quad (2.10)$$

$$T_b = T_{b_{base}} + 0.5 \times \left(\frac{\mu}{255} \right) \quad (2.11)$$

$$T_2 = T_{2_{base}} \times \left(1 + 0.1 \times \left(\frac{\sigma}{\mu} \right) \right) \quad (2.12)$$

Threshold T_1 digunakan untuk menentukan *threshold* atas *highlight* berbasis intensitas dari gambar *grayscale*, dimana akan dipilih apabila gambar berada di atas 15% dari rentang *brightness* gambar agar bisa focus pada area terang, bukan semua terang. Pada *threshold* T_c dan T_b digunakan dalam mengatur *threshold* yang berbasis kontras dan *brightness* agar *highlight* tahan *noise* kontras dan lebih adaptif terhadap *brightness* gambar. Sedangkan *threshold* T_2 digunakan untuk *relative highlight detection* yang dapat menentukan *baseline* syarat G dan B *relative*.

Penentuan area *specular highlight* dilakukan dengan menghitung intensitas agar dapat diklasifikasikan sebagai area *specular highlight*. Apabila area ini sudah ditentukan, maka adanya proses penghilangan area melalui dilatasi serta perbaikan area. Dilatasi dilakukan dengan memperluas wilayah spekular lebih dari yang diperlukan untuk mengompensasi penghilangan area *highlight*. Akhir dari proses HR yaitu perbaikan dengan pengisian (*inpainting*). Proses ini dilakukan dengan memulihkan informasi yang hilang dalam gambar dan biasanya mengacu pada interpolasi piksel yang hilang dengan menggunakan informasi

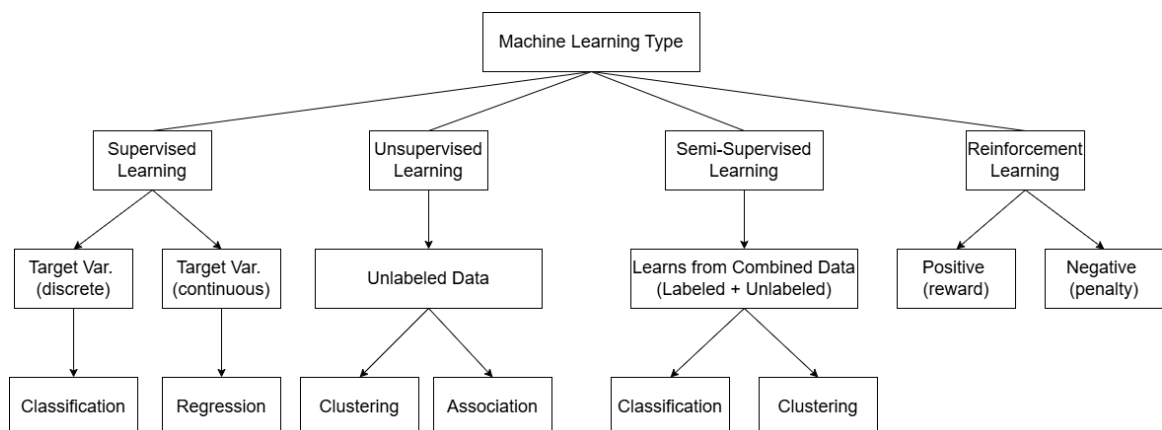
lingkungan sekitar (Au & Takei, 2001). *Inpainting* dilakukan dengan konsep *Navier-Stokes* (NS) yang diambil dari dinamika fluida. Proses ini akan mengisi area yang hilang (hasil penghilangan area) dengan *smoothness intensity* (ΔI) menggunakan arah *isophote* ($\nabla^\perp I$). Pengisian yang berhasil dikondisikan dengan persebaran ΔI sejajar dengan arah $\nabla^\perp I$ yang didefinisikan sebagai persamaan berikut.

$$\nabla^\perp I \cdot \nabla \Delta I = 0 \quad (2.13)$$

2.4 Machine Learning

Machine learning merupakan peralihan evolusi AI menuju kemampuan yang lebih canggih dan fleksibel. *Machine learning* memungkinkan sistem belajar sendiri dari data sehingga lebih adaptif dan efektif dalam menangani tugas yang membutuhkan pola yang lebih rumit. Sistem pembelajaran ini dilakukan dengan menyajikan data sebagai contoh yang relevan dengan permasalahan yang dihadapi dan menemukan struktur statistiknya, sehingga menghasilkan aturan untuk mengotomatiskan penyelesaian masalah tersebut. Oleh karena itu, *machine learning* memiliki kaitan yang sangat erat dengan statistik matematika, namun berbeda dari statistik dalam beberapa hal penting (Chollet, 2018). *Machine learning* cenderung menangani kumpulan data besar dan kompleks yang tidak praktis untuk analisis statistik klasik.

Berdasarkan prinsip pembelajaran, algoritma *machine learning* dibagi menjadi *supervised learning*, *unsupervised learning*, *semi-supervised learning*, dan *reinforcement learning*.



Gambar 2.7 Tipe algoritma *machine learning* (Sarker, 2021).

2.4.1 Supervised Learning

Pembelajaran ini berfokus pada pemetaan data input (dalam bentuk atribut) ke target yang diketahui (dalam bentuk label kelas) berdasarkan data *training* yang sudah diberi label (Sarker, 2021). Oleh karena itu penting adanya data *training* yang telah terlabel dengan benar

karena kualitas hasilnya sangat tergantung dari validitas label datanya. Secara umum, *supervised learning* sering digunakan dalam permasalahan “klasifikasi” sebagai pemisahan data dan “regresi” sebagai penyesuaian data (Chollet, 2018). Dalam regresi, algoritma dilatih untuk menghasilkan output kontinu (y) berdasarkan input data (x) menggunakan hipotesis $f(x)$. Sementara dalam klasifikasi, algoritma dilatih untuk menghasilkan output diskrit (y) dari input data (x) dengan menggunakan hipotesis $f(x)$.

2.4.2 *Unsupervised Learning*

Pembelajaran ini berjalan dengan menganalisis kumpulan data yang tidak berlabel sehingga perlu adanya pengawasan terhadap model yang bekerja guna menemukan informasi yang dibutuhkan (Sarker, 2021). Pembelajaran ini umumnya digunakan dalam *clustering*, Dimana terdapat sejumlah data input yang akan dikelompokkan kedalam kluster-kluster berdasarkan kriteria tertentu. *Unsupervised learning* adalah inti dari analisis data yang merupakan langkah penting dalam memahami sekumpulan data dengan baik sebelum mencoba memecahkan masalah dengan *supervised learning* (Chollet, 2018). Dengan menggunakan *unsupervised learning*, dapat menemukan relasi implisit antara *unlabeled data* yang tersedia.

2.4.3 *Semi-supervised Learning*

Jenis pembelajaran ini menggabungkan antara *supervised* dan *unsupervised* untuk menghasilkan suatu aturan dalam penyelesaian masalah. Oleh karena itu, pengaplikasian pembelajaran ini dapat digunakan dalam *classification* dan *clustering*. *Semi-supervised learning* bertujuan untuk menghasilkan prediksi yang lebih akurat dibandingkan jika hanya menggunakan data berlabel pada model tersebut (Sarker, 2021).

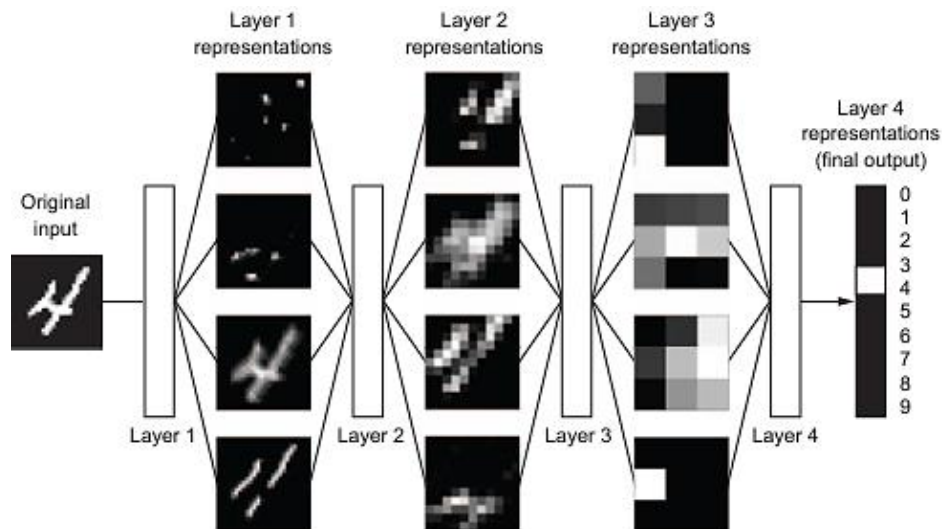
2.4.4 *Reinforcement Learning*

Reinforcement learning adalah jenis algoritma *machine learning* yang memungkinkan *software* dan *machine* melakukan evaluasi perilaku optimal dalam lingkungan tertentu untuk meningkatkan efisiensinya secara otomatis (Sarker, 2021). Dalam pembelajaran ini, *machine* diatur untuk melakukan tindakan berdasarkan permasalahan yang dihadapi, karena tidak ada jawaban yang pasti benar. *Machine* dapat belajar dari pengalaman sebelumnya untuk menghindari kesalahan dan meningkatkan kinerjanya seiring waktu. Jenis pembelajaran ini tidak memiliki data *training* dikarenakan data-data yang diperoleh berasal dari pengalaman sehingga sistem ini mirip dengan *supervised learning*. Hal yang membedakannya yaitu input yang digunakan tidak hanya satu jenis.

2.5 Deep Learning

Deep learning merupakan sub-bidang khusus dalam *machine learning*. *Deep learning* sering kali melibatkan puluhan atau bahkan ratusan lapisan representasi yang berurutan dan semuanya dipelajari secara otomatis dari data *training*. *Deep learning* adalah metode pembelajaran yang menggunakan jaringan syaraf buatan (*artificial neural network*) dengan *multi-layer* (Chollet, 2018). Istilah jaringan syaraf merujuk pada neurobiology yang dikembangkan dengan inspirasi dari pemahaman terhadap otak manusia, dimana neuron-neuron saling terhubung membentuk jaringan syaraf yang kompleks.

Pemahaman *deep learning* terhadap tujuan dari input yang diberikan dilakukan dengan pemetaan input yang rumit menjadi yang lebih sederhana. Setiap pemetaan ini diteruskan melalui berbagai lapisan dalam model. Dengan cara ini, *deep learning* mampu mempelajari pola-pola yang lebih kompleks dan abstrak dari data input, sehingga meningkatkan kemampuan komputer dalam mengidentifikasi dan memahami informasi yang ada dalam data tersebut. Oleh karena itu, penambahan lebih banyak lapisan dan lebih banyak unit dalam satu lapisan dapat merepresentasikan peningkatan kompleksitas fungsi (Goodfellow dkk., 2016).



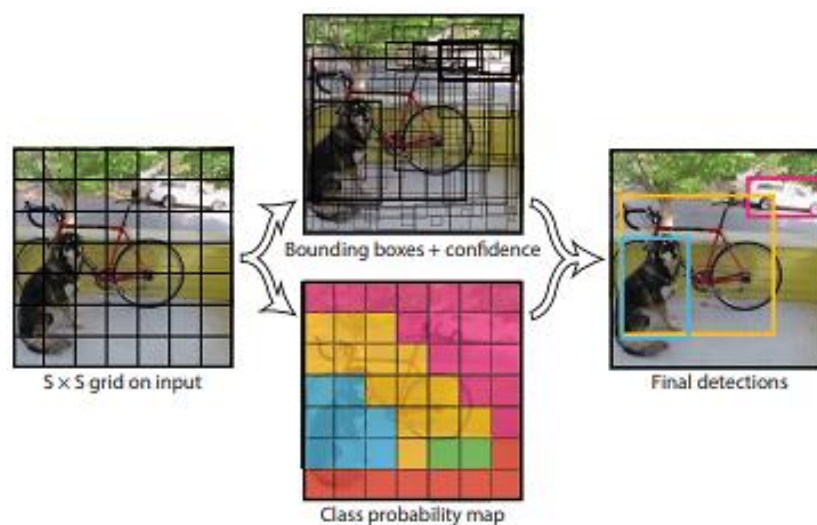
Gambar 2.8 Representasi model *deep learning* (Chollet, 2018).

Berdasarkan gambar 2.8, jaringan *deep learning* dapat dianggap sebagai proses penyaringan informasi secara bertahap. Informasi data input akan melewati lapisan filter, di mana setiap lapisan menyaring dan memproses informasi agar lebih jelas dan mudah dimengerti oleh komputer (Chollet, 2018). Setiap lapisan dalam *deep learning* akan mengidentifikasi detail yang berbeda dari data input yang diberikan, mulai dari elemen sederhana seperti garis dan tepi, hingga struktur yang lebih kompleks seperti bentuk dan

pola. Hasil akhir proses ini berupa informasi yang lebih “murni” sehingga komputer dapat memprosesnya dalam penyelesaian masalah yang dihadapi.

2.6 *You Only Look Once (YOLO)*

You Only Look Once atau yang lebih dikenal YOLO adalah algoritma yang mampu melakukan deteksi dan klasifikasi objek secara simultan dalam satu kali pemrosesan gambar (Redmon dkk., 2016). Pada prinsip deteksi objek, keterlibatan lokalisasi objek sangat diperlukan dalam pembangunan sistem. Lokalisasi objek dinyatakan dalam bentuk koordinat *ground truth* yang sesuai dengan lokasi objek yang dideteksi. Secara sederhana, setiap sel grid dalam deteksi akan menghasilkan serangkaian *bounding box* beserta *confidence* masing-masing kelas berdasarkan fitur yang diekstraksi dari *Convolutional Neural Network* (CNN). Penggunaan dan penggabungan lapisan CNN digunakan dalam menangkap pola dan ekstraksi fitur yang berguna pada skala spasial yang berbeda. Dengan cara ini, beban komputasi yang diperlukan dalam deteksi objek dapat diminimalkan secara signifikan. Dalam kinerjanya, metode ini menghilangkan kebutuhan untuk melakukan pemeriksaan berulang terhadap gambar, sehingga meningkatkan efisiensi dalam deteksi objek (Redmon et al., 2016).

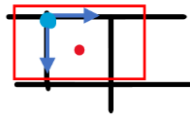


Gambar 2.9 Model YOLO (Redmon dkk., 2016).

Desain YOLO menggabungkan kecepatan dan akurasi waktu nyata dalam satu lintasan jaringan saraf konvolusional sambil mempertahankan presisi rata-rata yang tinggi (Redmon dkk., 2016). Pada dasarnya sistem ini akan membagi gambar input menjadi $S \times S$ grid. Jika pusat objek berada dalam suatu sel grid, maka sel grid tersebut bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel grid akan memprediksi 2 *bounding box* dan 2 *confidence*. Berdasarkan gambar 2.7, terdapat gambar input yang memiliki 7×7 grid

sehingga total prediksinya yaitu $7 \times 7 \times 2 = 98$ *bounding box* dengan nilai *confidence* yang dilambangkan oleh ketebalan garis *bounding box*-nya. Semakin tebal garis maka nilai *confidence*-nya semakin tinggi, seperti halnya dengan direpresentasikan pada gambar 2.9 atas.

Pada hasil output terdiri dari 5 prediksi: x , y , lebar (*width*), tinggi (*height*), dan skor keyakinan (*confidence*). Koordinat (x, y) menunjukkan posisi pusat *bounding box* relatif terhadap batas sel grid. Sementara itu, *width* dan *height* menunjukkan lebar dan tingginya *bounding box* yang diprediksi relatif terhadap keseluruhan gambar. Penentuan *bounding box* dilakukan dengan menormalisasi prediksi x , y , lebar (*width*), dan tinggi (*height*) ke 0 hingga 1. *Width* dan *height* dinormalisasikan dengan tinggi dan lebar gambar input yang digunakan. Sedangkan koordinat (x, y) menormalisasikan titik tengah *bounding box* dari 0 hingga 1 yang diukur dari posisi kiri atas sel grid yang ditempati titik tengah.



Gambar 2.10 Normalisasi titik tengah *bounding box*

Nilai koordinat (x, y) setelah normalisasi yaitu $x = (x_{\text{titik tengah}} - x_{\text{sel grid}}) / \text{ukuran sel grid}$ dan $y = (y_{\text{titik tengah}} - y_{\text{sel grid}}) / \text{ukuran sel grid}$.

Pada masing-masing *bounding box* memiliki kategori berbeda-beda, jika terdapat prediksi *bounding box* yang banyak akan dilakukan *threshold* dan *Non-Maximum Suppression* (NMS). NMS digunakan ketika terdapat banyak *bounding box* yang *overlap*, maka nilai *confidence* yang tinggi akan diambil karena menunjukkan *ground truth* yang tinggi juga. Kemudian perhitungan IOU akan dilakukan seperti gambar 2.11 untuk *suppress bounding box* lain yang memiliki nilai $\text{IOU} > \text{threshold}$.

$$\text{IOU} = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of intersection}}{\text{area of union}}$$

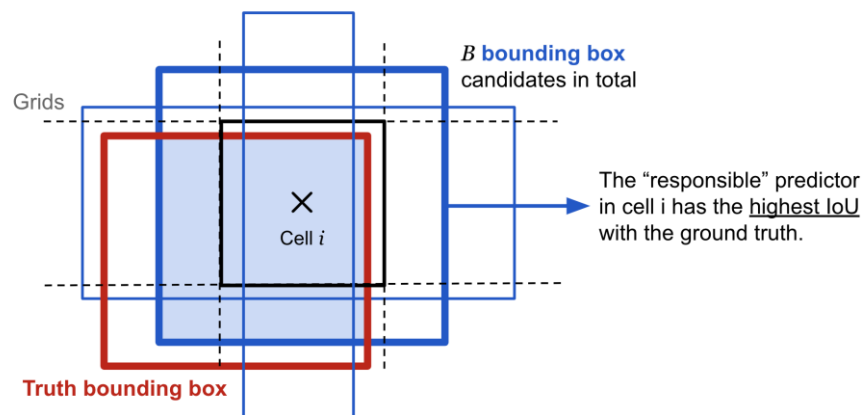
Gambar 2.11 Penentuan *Intersection Over Union* (IOU) (Padilla dkk., 2020).

Hasil akhir lain yaitu nilai *confidence* yang merepresentasikan seberapa yakin *box* dari model tersebut berisi objek serta seberapa akurat model tersebut menganggap *box*-nya sesuai prediksi. Secara umum, *confidence* didefinisikan sebagai

$$\text{Pr}(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} \quad (2.14)$$

Jika tidak ada objek dalam sel grid tersebut, maka *confidence*-nya bernilai nol karena probabilitas objeknya nol ($\Pr(\text{Object}) = 0$). Namun sebaliknya, jika terdapat objek dalam sel grid, maka nilai dari *confidence* sama dengan perpotongan atas gabungan (IOU) antara *box* yang diprediksi dengan kebenarannya (Redmon dkk., 2016). Objek yang memiliki metrik IOU maksimum akan dikodekan dan dideteksi oleh sistem. IOU digunakan sebagai metrik evaluasi untuk perhitungan presisi *detector* dan pengklasifikasian objek pada *dataset* tertentu (Jana dkk., 2018). Metrik ini melibatkan dua komponen utama, yaitu *ground truth bounding box* yang berlabel manual dari objek dalam gambar dan *predicted bounding box* yang dihasilkan oleh algoritma deteksi dan klasifikasi.

Dalam sel grid dapat dilakukan pembentukan sejumlah *anchor box* yang digunakan untuk membantu menentukan letak *bounding box* yang sah terisi objek deteksi. Penentuan ini dilakukan dengan perhitungan IOU dari *overlapping* sejumlah *anchor box* dengan *ground truth bounding box*. Nilai $\text{IOU} > \text{threshold}$ menunjukkan objek terdeteksi digunakan juga dalam menentukan nilai performa beberapa parameter. Nilai IOU tertinggi menunjukkan *overlap* dengan *ground truth* yang lebih besar (Alain dkk., 2023).



Gambar 2.12 Penerapan *anchor box* (Alain dkk., 2023).

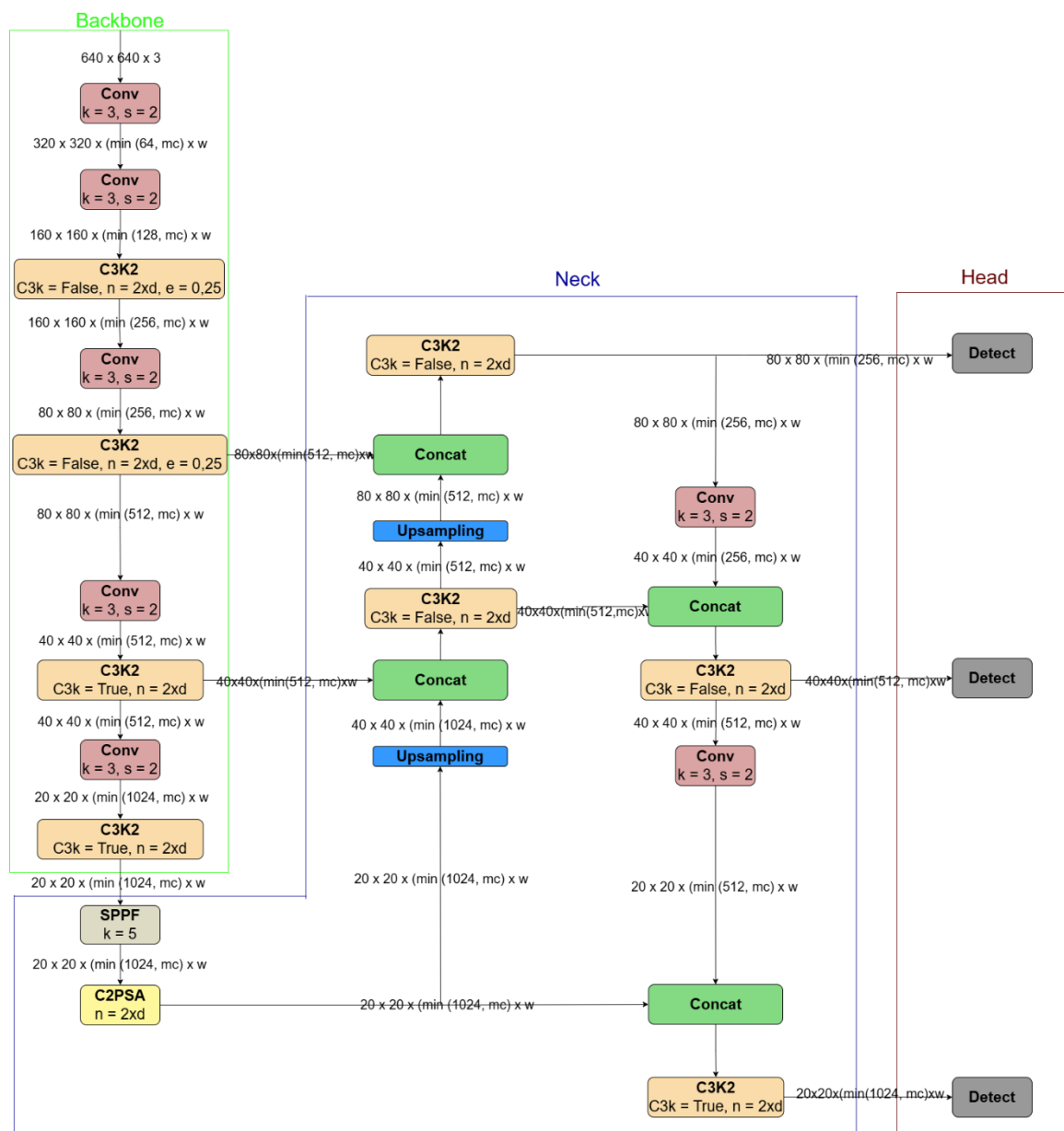
Arsitektur YOLO telah mengalami revolusi secara signifikan sejak tahun 2016. Pada tahun 2015, diperkenalkan konsep terobosan deteksi objek waktu nyata melalui satu lintasan jaringan yang disebut YOLOv1 (Ragab dkk., 2024). Model YOLOv1 menjadi model pertama atau asli yang mana menggabungkan masalah pembentukan *bounding box* dan identifikasi label kelas dalam satu jaringan. Pembentukan *bounding box* dengan pemrosesan model CNN dilakukan secara paralel dengan masing-masing output masuk ke dalam *head section* untuk melakukan deteksi. Kemudian pada tahun 2016, dirilisnya YOLOv2 yang arsitekturnya membuat sejumlah perbaikan berulang di atas YOLOv1 termasuk *BatchNorm*, resolusi tinggi, dan *anchor box*. Pengenalan fungsi konvolusi ekstraksi fitur baru yang

digunakan dalam *backbone* YOLOv2 yaitu DarkNet-19 dengan 19 layer konvolusi dan 5 layer *max-pooling*. Versi selanjutnya dirilis pada tahun 2018, YOLOv3 dibangun di atas model versi sebelumnya dengan sejumlah perbaikam, seperti penambahan skor objektivitas ke prediksi *bounding box*, integrasi koneksi tambahan ke *backbone layers*, dan penerapan *prediction* pada tiga skala granularitas terpisah untuk meningkatkan kinerja deteksi objek kecil (Nelson, 2024). Dalam jurnal yang ditulis oleh Ragab dkk., menyebutkan bahwa arsitektur YOLOv3 memperkenalkan jaringan baru yang disebut Darknet-53, dimana kinerjanya lebih mendalam dengan penggunaan GPU dan memiliki akurasi yang lebih daripada arsitektur sebelumnya. Pada tahun 2019, terdapat perbaikan metode dari YOLOv3 menjadi YOLOv4 yang mana memperkenalkan teknik augmentasi data mosaik (Ragab dkk., 2024).

Teknik ini membantu meningkatkan ketahanan model terhadap variasi skala dan orientasi objek yang berbeda. Selanjutnya pada tahun 2020, diperkenalkan YOLOv5 dengan arsitektur jaringan baru yang disebut CSPDarknet yang lebih efisien daripada DarkNet-53. Versi YOLO ini hanya memerlukan instalasi *torch* dan beberapa pustaka Python yang ringan sehingga *training* yang dilakukan sangat cepat. Kemudian pada tahun 2021, perilsan YOLOv6 sebagai perbaikan dari versi 5 dengan penggunaan *loss function* baru yang disebut *GIoU* yang membantu meningkatkan akurasi model (Ragab dkk., 2024). Pada tahun 2022, perilsan YOLOv7 dengan arsitektur Panoptic YOLO yang dapat mendeteksi objek dan segmentasi semantic dalam gambar. Selanjutnya YOLOv8 dikembangkan oleh Ultralytics pada tahun 2023, versi ini memperkenalkan sejumlah fitur baru, di antaranya BiFPN, *loss function CIoU*, dan *Cross-GPU training*. YOLO v8 adalah model tanpa *anchor* yang berarti melakukan prediksi secara langsung pada pusat suatu objek. Pada tahun 2024 terjadi perilsan 3 versi YOLO, yaitu YOLOv9, YOLOv10, dan YOLOv11. YOLOv9 mencapai mAP yang lebih tinggi daripada model YOLO populer yang ada, seperti YOLOv8, YOLOv7, dan YOLOv5, saat dibandingkan dengan kumpulan data MS COCO. Sementara YOLOv10 merupakan teknologi terbaru dalam deteksi objek yang menawarkan latensi lebih rendah daripada model YOLO sebelumnya dengan parameter lebih sedikit. Dalam hal kinerja, YOLOv10-S bekerja 1,8x lebih cepat daripada RT-DETR-R18 dengan AP serupa pada COCO, serta menggunakan parameter dan FLOP yang 2,8x lebih sedikit (Nelson, 2024).

Deteksi objek yang paling terbaru yaitu YOLO11. Versi ini dikembangkan oleh Ultralytics berdasarkan kemajuan dari versi YOLO sebelumnya. Pada YOLO11, arsitektur *backbone* dan *neck* mengalami peningkatan kemampuan ekstraksi fitur untuk deteksi objek

yang lebih presisi dan kinerja yang lebih kompleks, seperti yang ditunjukkan pada gambar 2.10. Dengan desain model yang lebih maju, YOLO11m mencapai Presisi Rata-rata (mAP) yang lebih tinggi pada *dataset* COCO dengan 22% lebih sedikit parameter dibandingkan YOLOv8, menjadikannya efisien secara komputasi tanpa mengurangi akurasi. YOLO11 juga dapat diterapkan di berbagai lingkungan, seperti perangkat *edge*, platform *cloud*, dan sistem yang mendukung GPU NVIDIA, memastikan fleksibilitas maksimum (Ultralytics, 2024). Secara umum, arsitektur YOLO terdiri dari tiga komponen, yaitu *backbone* sebagai ekstraktor fitur utama yang memanfaatkan jaringan saraf konvolusional, *neck* sebagai penggabung dan penyempurna representasi fitur di berbagai skala, dan *head* sebagai deteksi objek berdasarkan fitur maps yang telah disempurnakan.



Gambar 2.13 Arsitektur utama YOLO11 (Jegham dkk., 2024).

Backbone merupakan arsitektur YOLO yang bertanggung jawab untuk mengekstraksi fitur dari input pada berbagai skala. Pada YOLO11, lapisan *backbone* terdiri atas *convolutional layers*. *Convolutional layers* yang digunakan sama dengan arsitektur YOLO versi sebelumnya yang mana berfungsi untuk menurunkan sampel gambar. Dalam penggunaan, adanya pengenalan *C3k2 layer* yang menggantikan *C2F* dalam versi sebelumnya. *C3k2 layer* merupakan implementasi komputasi yang lebih efisien dari *CSP Bottleneck* yang menggunakan dua *convolutional layers* dengan ukuran kernel yang lebih kecil sehingga dapat mempercepat komputasi, meningkatkan aliran gradien, dan mengurangi *overhead*. Secara sistem, *CSP (Cross Stage Partial) Bottleneck* bekerja dengan membagi fitur maps menjadi dua, yaitu satu melewati rangkaian *Bottleneck* dan satunya lagi langsung masuk ke output (Khanam & Hussain, 2024).

Neck memiliki tanggung jawab terhadap penggabungan fitur dari berbagai resolusi dan meneruskannya ke *head* untuk diprediksi. *Neck* melibatkan *upsampling* dan *concatenation* fitur maps dari berbagai resolusi. *Upsampling* berguna untuk meningkatkan resolusi hasil fitur maps dalam proses *backbone*. Sedangkan *concatenation* merupakan *layers* penghubung output hasil dari *C3k2 backbone* dan *upsampling* dengan meningkatkan saluran atau *channel*. Seperti halnya dengan *backbone*, *neck* juga mengimplementasikan *C3k2 layer* agar mempercepat komputasi dan memfokuskan perhatian spasial pada area tertentu dengan *C2PSA*. *C2PSA (Cross Stage Partial with Spatial Attention)* merupakan tambahan baru yang memungkinkan YOLO11 untuk meningkatkan akurasi deteksi dengan peningkatan fokus model pada wilayah tertentu. Sedangkan, *SPPF* atau *Spatial Pyramid Pooling Fast Module* merupakan blok yang melakukan pengumpulan dan penggabungan data dalam berbagai skala. Sementara itu, pada bagian *head* merupakan bagian akhir yang mana berperan sebagai pendeteksian objek. Pada bagian ini, terdapat *CBS layers* yang mencakup *convolution*, *batch normalization*, dan *SiLU*. *CBS* berfungsi dalam mengekstrak fitur-fitur yang relevan dalam deteksi objek, menstabilkan dan menormalkan aliran data, dan menggunakan fungsi aktivasi *Sigmoid Linear Unit (SiLU)* untuk non-linearitas, yang meningkatkan kinerja model dengan memastikan hasil akhir di antara 0 dan 100 persen (Khanam & Hussain, 2024). Berdasarkan ketiga bagian komponen dari YOLO11, hasil deteksi akhir sistem berupa penggabungan koordinat *bounding box*, skor objektivitas, dan skor kelas dalam penilaian keakuratan sistem deteksi.

2.7 Penelitian Terdahulu

Penelitian terdahulu, yang dilakukan oleh Lalinia & Sahafi melakukan pembangunan sistem deteksi *colorectal polyp* yang efektif dan efisien. Berdasarkan data gangguan saluran *gastrointestinal* di Eropa, adanya kasus Kesehatan yang signifikan yaitu sekitar 1 juta setiap tahunnya. Salah satu penyakitnya yaitu CRC yang mana bermula dari *colorectal polyp*. Prosedur *colonoscopy* yang menjadi standar emas untuk mengurangi insiden dan mortalitas CRC masih rentan terhadap kesalahan manusia (Lalinia & Sahafi, 2024). Oleh karena itu, Lalinia & Sahafi memiliki inisiatif untuk membangun sistem deteksi dini *colorectal polyp* yang diharapkan mampu meminimalkan diagnosis yang terlewat, memfasilitasi deteksi dini, dan berkontribusi pada pencegahan CRC.

Data yang digunakan dalam proses pelatihan dan evaluasi penelitian ini adalah lima data sekunder, Kvasir-SEG, CVC-ClinicDB, CVC-ColonDB, ETIS, dan EndoScene. Pembagian data 80:10:10 dilakukan pada kelima *dataset* tersebut. Pembangunan sistem deteksi *colorectal polyp* dilakukan dengan metode YOLOv8, dengan perbandingan semua versi YOLOv8 sehingga hasil terbaik didapatkan dengan YOLOv8m. Dalam proses deteksi, penelitian ini memberikan hasil yang efektif dalam mengenali *colorectal polyp* dengan benar, yaitu presisi 95,6%, daya ingat 91,7%, dan skor F1 92,4% serta mAP50 sebesar 85,4% dan mAP50-95 sebesar 62% dengan waktu inferensi 10,6 milidetik dan 25 juta parameter yang menunjukkan keseimbangan yang terpuji antara akurasi dan efisiensi komputasi. Namun, hasil penelitian ini masih memiliki kekurangan pada terbatasnya ketersediaan kumpulan data *polip* public yang luas. Ketersediaan dan kualitas citra yang digunakan sangat mempengaruhi hasil akurasi dari model. Hal tersebut juga dibuktikan dalam penelitian lain yang dilakukan oleh Elkarazle dkk.

Dalam penelitian yang dilakukan oleh Elkarazle dkk., digunakan beberapa metode sebagai perbandingan dalam menentukan keunggulan metode deteksi *colorectal polyp* dari citra *colonoscopy*. Penelitian ini didasarkan pada faktor kelelahan dan kurangnya pengalaman ahli medis dalam melakukan prosedur *colonoscopy* sehingga *colorectal polyp* tidak teridentifikasi oleh ahli yang menyebabkan komplikasi lebih lanjut dan berkembang menjadi CRC. Pada kasus ini, ahli medis mengalami kesulitan dalam mengidentifikasi polip yang terletak di sisi kanan usus besar, dikenal sebagai usus besar proksimal, karena bentuk dari usus besarnya sendiri. Selain itu, kondisi citra *colonoscopy* yang terkena pantulan cahaya yang berlebih serta keburaman juga menjadi salah satu penyebab kesulitannya ahli medis dalam mengidentifikasi polip.

Data yang digunakan yaitu ETIS-Larib *dataset* yang terdiri dari 196 data dengan resolusi 1225 x 996 piksel. *Dataset* yang digunakan merupakan data asli tanpa filter dengan beberapa kesediaan data buram yang dapat membantu menjawab akar permasalahan penelitian ini. Pembangunan sistem deteksi yang dilakukan oleh Elkarazle dkk., menggunakan salah satu metode YOLO. YOLO memiliki kemampuan dalam mendeteksi objek lebih cepat daripada algoritma deteksi lainnya. Pemodelan metode, dilakukan pada YOLOv3 dengan CPSNet dan YOLOv4 dengan CPSPDarkNet5. Selain itu, juga dilakukan penggantian aktivasi ReLU menjadi SiLU serta augmentasi data untuk menghindari *overfitting*. Dari penelitian ini, menghasilkan *precision* sebesar 90.61, *recall* sebesar 91.04, dan *F1 score* sebesar 90.82. Kekurangan yang dimiliki dalam penelitian ini adalah model mengalami kesulitan dalam membedakan pantulan cahaya putih, material feses, dan objek berbentuk polip lainnya dari polip yang sebenarnya.

BAB III

METODE PENELITIAN

3.1 Tempat dan Waktu

Penelitian ini dilakukan di Rumah Sakit Universitas Airlangga yang bertempat di Jl. Dharmahusada Permai, Surabaya, serta Laboratorium Teknik Biomedis Fakultas Sains dan Teknologi, Universitas Airlangga bertempat di Jalan Ir. Soekarno. Mulyorejo, Kec. Mulyorejo, Surabaya, Jawa Timur. Penelitian ini berlangsung dalam kurun waktu 5 bulan, pada interval bulan Desember 2024 sampai dengan Mei 2025.

3.2 Alat dan Bahan

Perangkat *hardware* seperti laptop/komputer untuk melakukan proses *training* (*machine learning*) dengan spesifikasi tinggi untuk mendapatkan suatu model yang dapat memprediksi data baru. Berikut ini adalah spesifikasi perangkat yang digunakan.

- Laptop : Asus VivoBook K413JQ
- Windows : Windows 11
- Processor : Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
- GPU : NVIDIA® GeForce® MX350 2GB GDDR5
- RAM : 8 GB
- SSD : 512 GB

Software dan *library* yang digunakan pada penelitian ini adalah:

- *Google Collaboratory*

Tools dari Google yang berbasis *cloud* untuk menulis dan menjalankan program dengan bahasa Python.

- *Ultralytics*

Library Python yang membantu untuk memudahkan penggunaan model YOLO untuk deteksi objek.

- Roboflow

Platform yang digunakan dalam proses pengumpulan, pelabelan, dan pengolahan data sebelum masuk ke dalam model.

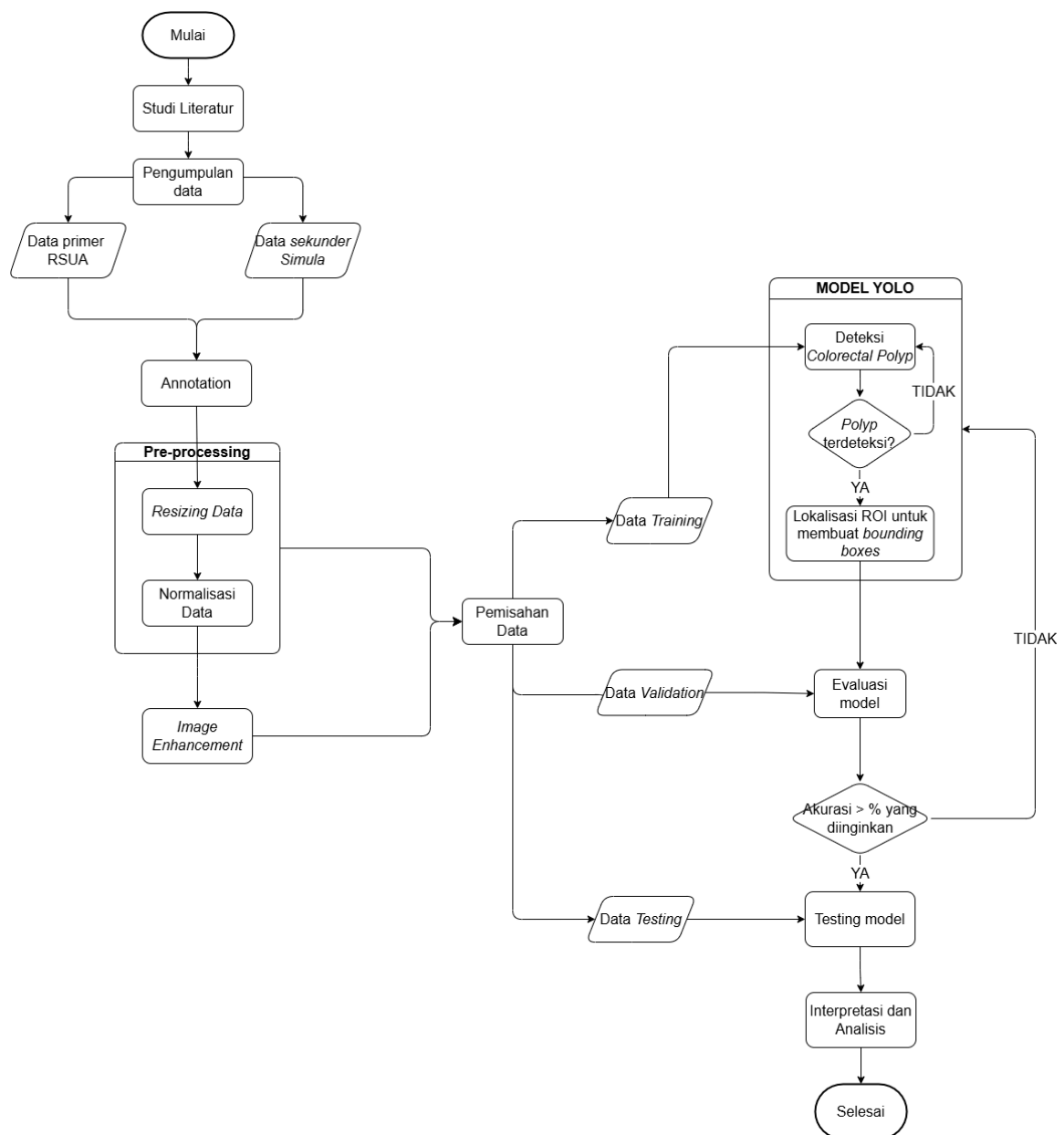
Bahan utama yang digunakan pada penelitian ini adalah sekumpulan data sekunder berupa *colonoscopy image* yang diperoleh dari Rumah Sakit Universitas Airlangga yang bersifat *private* serta citra *colonoscopy* yang diperoleh dari bank *Simula Datasets*, berikut merupakan pendistribusiannya.

Tabel 3.1 Pendistribusian *dataset* yang digunakan dalam penelitian.

	Kvasir-seg	RSUA
	950	236
<i>Training</i>	831	
<i>Validation</i>	236	
<i>Testing</i>	121	

3.3 Prosedur Penelitian

Penelitian ini digunakan untuk membuat sistem deteksi lokasi *colorectal polyp* dari citra *colonoscopy* dengan menggunakan *machine learning*, berupa YOLO. Berikut skema penelitian yang dilakukan:

**Gambar 3.1** Skema penelitian.

3.3.1 Studi Literatur

Penulis melakukan studi literatur terkait permasalahan kesehatan yang ada di Indonesia, melalui laman resmi kementerian kesehatan serta perpustakaan Kampus C, Universitas Airlangga. Setelah penetapan topik permasalahan, penulis mempelajari beberapa *text book* dan jurnal terkait dengan penyakit *colorectal polyp* serta melakukan konsultasi kepada dokter penyakit dalam, gastroenterologis. Selain itu, penetapan metode yang digunakan dalam penelitian ini didasari dari penanganan penyakit *colorectal polyp*, yaitu pembedahan, sehingga peneliti melakukan *literature review* terhadap jurnal dan *text book* mengenai metode *deep learning*, YOLO, untuk membantu para ahli dalam menentukan letak *polyp*.

3.3.2 Pengumpulan

Pada penelitian ini, data yang digunakan adalah data primer berupa *colonoscopy image* yang diperoleh dari Rumah Sakit Universitas Airlangga serta data sekunder berupa citra *colonoscopy* yang diperoleh dari bank *Simula Datasets*. Perolehan *Simula Datasets* berasal dari data Kvasir - *SEG* (<https://datasets.simula.no/kvasir-seg/>) yang sudah dalam kondisi teranotasi berupa koordinat *bounding box* di dalam gambar tersebut. Sedangkan data Rumah Sakit Universitas Airlangga dilakukan anotasi dengan pertimbangan dokter ahli gastroenterologi. Pengumpulan data dilakukan dengan minimal sampel yang dihitung menggunakan pendekatan *Lemeshow* sebagai berikut (Lwanga dkk., 1991).

$$n = \frac{z^2 \times P(1 - P)}{\alpha^2} \quad (3.1)$$

Dengan,

n = ukuran minimal sampel yang dibutuhkan

z = selang kepercayaan (95%)

P = estimasi proporsi populasi

α = taraf kesalahan

3.3.3 Anotasi Dataset

Pada penelitian ini, anotasi *dataset* dilakukan dengan menambahkan label atau informasi tambahan (metadata) pada data yang digunakan sebagai masukan *machine learning*. Anotasi berupa *bounding box* yang digunakan sebagai *groundtruth* dalam penentuan letak *colorectal polyp*.

3.3.4 *Pre-processing Data*

Pada tahap ini, data yang sudah dikumpulkan akan melalui proses *pre-processing data*, dimana data akan dipersiapkan untuk memasuki model YOLO yang akan dibuat.

- *Resizing Data*

Resizing data dilakukan dengan mengubah ukuran dimensi data citra *colonoscopy* menjadi ukuran tetap sesuai dengan model YOLO. Tujuan dari tahap ini bertujuan untuk menyamakan dimensi input model agar proses *training* lebih efisien. Beberapa model dirancang untuk ukuran input yang konsisten. *Resizing data* dapat membantu menyamakan ukuran dan mengurangi kompleksitas komputasi.

- *Normalisasi Data*

Ketika menemui *dataset* dengan jumlah besar, normalisasi data sangat penting untuk memastikan konsistensi dan kualitas data. Pada tahap ini, data citra *colonoscopy* disamaratakan rentang pikselnya antara 0 hingga 1. Normalisasi data digunakan untuk mempermudah proses pembelajaran model dan memastikan bahwa gambar dapat diproses secara konsisten.

- *Image Enhancement*

Tahap *enhancement* bertujuan untuk meningkatkan karakteristik citra. Data citra *colonoscopy* disesuaikan tampilannya agar dapat dianalisis lebih lanjut. Dalam pengoperasionalannya, *image enhancement* dilakukan dengan meningkatkan fitur tertentu pada citra input sehingga hasil proses pengolahan gambarnya menjadi tinggi.

3.3.5 *Pemisahan/Splitting Data*

Data hasil *pre-processing*, terdapat dua data yang akan digunakan, yaitu data *training*, data *validation*, dan data *testing*. Sesuai dengan tabel 3.1 pendistribusian data, *splitting data* hanya digunakan untuk *training*, *validation*, dan *testing data* dengan rasio perbandingan 70:20:10. Tahap ini bertujuan untuk mencegah bias saat mengevaluasi model serta memastikan bahwa model yang akan dibangun mampu ditingkatkan performanya dalam menggeneralisasi data yang belum pernah dilihat sebelumnya.

3.3.6 *Deteksi Model Berbasis YOLO*

Pada tahap ini, dilakukan pembuatan model pendeteksian penyakit *colorectal polyp* dengan menggunakan YOLO dimana metode ini berfungsi untuk mendeteksi atau mengenali objek dengan menghasilkan *bounding box* beserta *confidence* berdasarkan fitur yang

diekstraksi dari CNN. Algoritma YOLO diakses melalui *Ultralytics library* dengan *framework Google Colaboratory*.

3.3.7 Evaluation dan Testing Model

Evaluasi model dilakukan untuk pengecekan apakah model yang digunakan mengalami *underfitting* dan *overfitting*, yang mana berpengaruh pada kinerja model dalam melakukan generalisasi pada data baru. Proses evaluasi ini menggunakan data evaluasi yang tidak digunakan pada *training model* dan data *validation* sehingga data yang ada dianggap baru dan asing. Pada proses ini, model akan membuat banyak kesalahan dalam memprediksi data-data baru yang ditemukan. Pada proses pengujian model menggunakan data *testing* untuk mendapatkan informasi detail mengenai *accuracy* dan *loss* pada seluruh data yang digunakan.

3.3.8 Interpretasi dan Analisis

Proses ini membantu dalam pemahaman tentang keterkaitan antara variabel dan menghasilkan pengetahuan tambahan mengenai deteksi penyakit *colorectal polyp* berdasarkan citra *colonoscopy* penderita. Performa model yang dibangun dapat dilihat berdasarkan parameter-parameter seperti, nilai *precision*, *recall*, dan *f1-score*. Hasil akhir diolah pada *confusion matrix* yang berisi pengelompokan hasil, sebagai berikut:

- True Positive (TP)* : Jumlah positif yang diprediksi benar
- False Positive (FP)* : Jumlah positif yang diprediksi salah
- True Negative (TN)* : Jumlah negative yang diprediksi benar
- False Negative (FN)* : Jumlah negative yang diprediksi salah

Tabel 3.2 *Confusion matrix*.

<i>Predicted value</i>	<i>Actual value</i>	
	TP	FP
	FN	TN

Confusion matrix ini dapat digunakan untuk menghitung nilai dari parameter performa model.

$$precision = \frac{TP}{TP + FP} \quad (3.2)$$

$$recall = \frac{TP}{TP + FN} \quad (3.3)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3.4)$$

Pada penelitian ini juga dilakukan uji signifikansi untuk mengetahui seberapa bermakna hasil performa masing-masing skenario melalui nilai *mean* IoU yang didapatkan dari perhitungan seperti pada gambar 2.11. Pengujian signifikansi dilakukan dengan mengukur *p-value* atau probabilitas yang menentukan apakah hasil dari suatu analisis statistik signifikan atau tidak yang dilakukan dengan *Mann-Whitney U Test* (Quraissy & Madya, 2021). Pemilihan uji signifikansi ini didasari oleh data yang diproses merupakan data independen. Pengujian ini dilakukan *pairing* semua skenario yang ada. Syarat signifikansi dalam proses ini yaitu perhitungan *p-value* yang bernilai kurang dari 0.05 (*p-value* < 0.05).

$$z = \frac{U - \mu_U}{\sigma_U} \quad (3.5)$$

3.4 Anggaran Biaya Penelitian

Penelitian ini memerlukan biaya penelitian sebesar:

- Biaya etik	= Rp. 850.000,00
- Biaya pengolahan data	= Rp. 944.000,00
- Biaya operasional	= Rp. 500.000,00
<hr/>	
	Rp. 2.294.000,00 +

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Pengumpulan Data

Dalam penelitian ini, jumlah minimal sampel yang diperlukan pada penelitian ini sejumlah 384 citra. *Dataset* yang digunakan dalam penelitian merupakan penggabungan dua data sekunder berupa citra *colonoscopy* yang diperoleh dari bank *Simula Dataset* serta *colonoscopy image* yang diperoleh dari Rumah Sakit Universitas Airlangga yang bersifat *private* dengan jumlah keseluruhan 1.186 citra yang hanya memiliki 1 *groundtruth*. Data tersebut terdiri 950 citra dari Kvasir-SEG dan 236 citra dari RSUA. Sejumlah data ini sudah dapat mewakili data pengolahan citra karena sudah melebihi jumlah minimal sampel, yang mana semakin banyak data maka semakin baik pelatihan kinerja model.

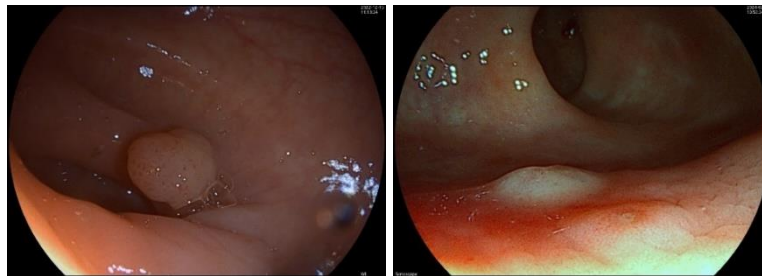
Dataset Kvasir-SEG memiliki variasi resolusi gambar berkisar 332 x 487 hingga 1920 x 1072 piksel. *Dataset* ini merupakan kumpulan data dari endoskopi di Vestre Viken Health Trust, Norwegia, yang telah diberi anotasi oleh ahli medis gastroenterologi. Dalam *dataset* ini terdiri dari citra polip beserta *masking*-nya yang disimpan dalam dua folder terpisah dengan format JPEG serta informasi gambar seperti titik koordinat *masker* dan *bounding boxes* terkait yang disimpan dalam file JSON.

Name	Date modified	Type	Size
images	1/24/2025 8:07 PM	File folder	
masks	1/24/2025 8:07 PM	File folder	
kvasir_bboxes	2/5/2025 3:17 PM	JSON File	129 KB

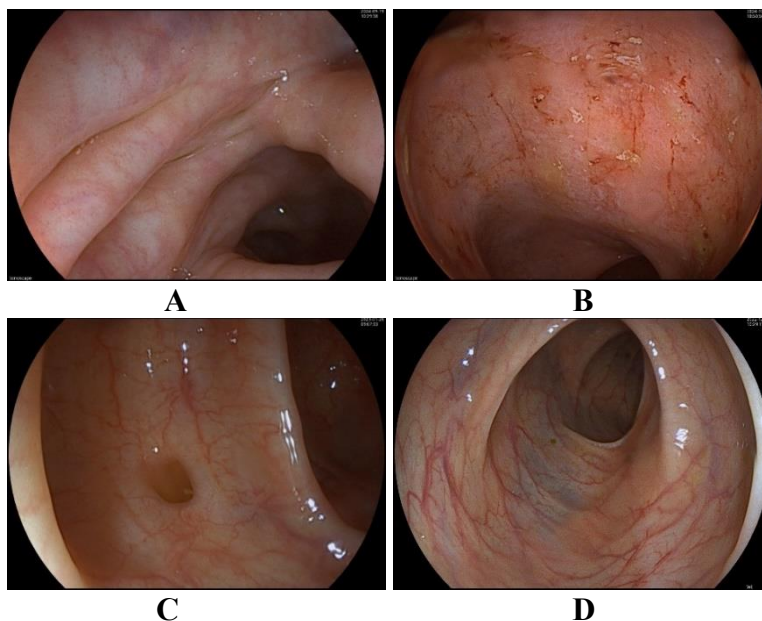
Gambar 4.1 Folder Kvasir-SEG.

Pada penelitian ini juga menggunakan citra *colonoscopy* yang diperoleh dari Rumah Sakit Universitas Airlangga dengan melalui uji etik dan beberapa persyaratan sehingga data bersifat *private*. Pengumpulan *dataset* dilakukan dengan mengambil kumpulan data pada komputer yang terhubung langsung dengan endoskop serta adanya pendampingan ahli gastroenterologi. Alat endoskop yang digunakan dalam pengambilan data adalah SonoScape tipe HD-550 dengan *disposable biopsy forceps* dari AGS yang memiliki diameter 2.8mm, panjang 2300mm, dan dapat membuka selebar 6.8mm. *Dataset* yang digunakan pada penelitian ini terbagi menjadi 118 data polip dan 118 non-polip. *Dataset* diperoleh dari 72 pasien, dengan jumlah 36 pasien laki-laki dan 36 pasien perempuan, dalam rentang usia yang paling muda berada di 18 tahun dan yang paling tua 79 tahun. *Dataset* yang diambil berada

pada rentang waktu Desember 2022 hingga Januari 2025. Kumpulan *dataset* RSUA memiliki variasi resolusi gambar sebesar 1504 x 1080 piksel. Citra non-polip yang diambil, digunakan sebagai pengecoh sistem pendeteksian polip.



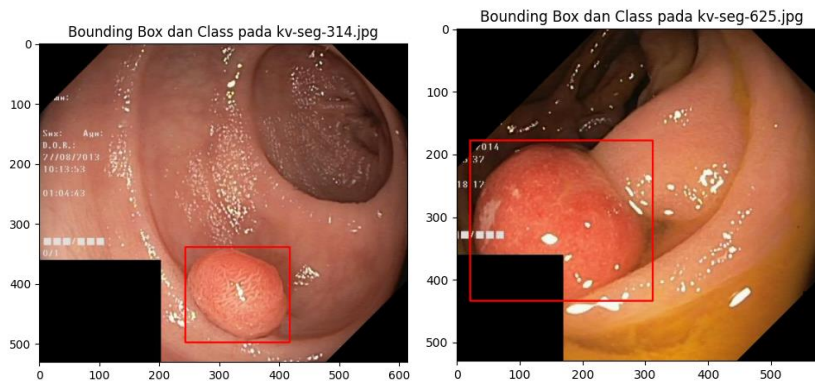
Gambar 4.2 Citra polip pada *dataset* RSUA.



Gambar 4.3 Citra non-polip pada *dataset* RSUA, A: Edema, B: Patchy erythema, C: Diverticulae, D: Normal.

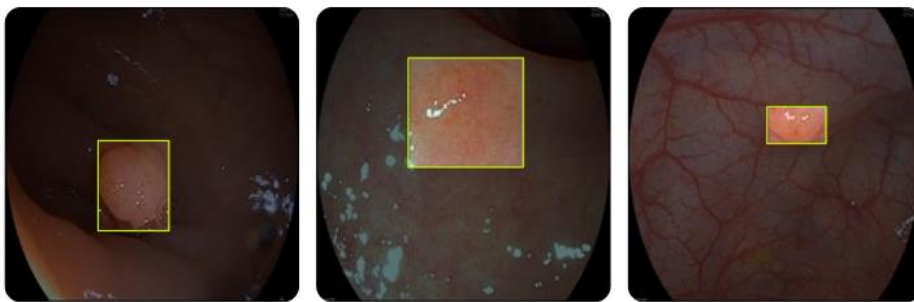
4.2. Anotasi *Dataset*

Pada *dataset* Kvasir-seg telah diberi anotasi oleh ahli gastroenterologi dengan bekerjasama dengan *Cancer Registry of Norway* (CRN) menggunakan aplikasi Labelbox. Labelbox merupakan alat yang digunakan untuk membantu dalam pemberian label atau anotasi pada area yang diminati dalam gambar. Hasil anotasi disimpan dalam file JSON yang berisi *image id*, ukuran *bounding boxes*, koordinat *bounding boxes*, dan kelas gambar. Dalam penggunaannya, adanya pengubahan koordinat *bonding boxes* dari format JSON menjadi .txt sebagai masukan YOLO.



Gambar 4.4 Hasil pelabelan *dataset* Kvasir-seg

Dataset yang berasal dari Rumah Sakit Universitas Airlangga merupakan data mentahan yang dijadikan sebagai data yang merepresentasikan data lapangan secara langsung. *Dataset* ini diberi pelabelan yang telah diverifikasi oleh dokter ahli gastroenterologi dari Rumah Sakit Universitas Airlangga dengan menggunakan Roboflow, seperti pada gambar 4.5.



Gambar 4.5 Pelabelan *dataset* RSUA menggunakan Roboflow.

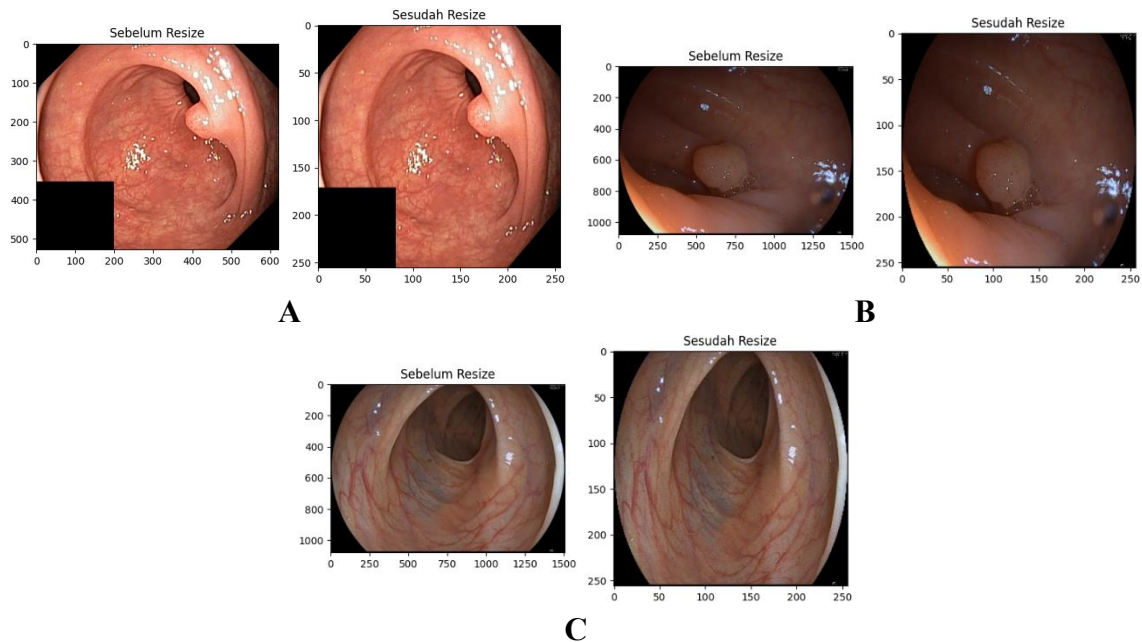
4.3. Hasil *Preprocessing*

Sebelum masuk dalam eksekusi model, *dataset* yang telah terkumpul akan masuk dalam tahap *preprocessing* sebagai proses awal persiapan data sebelum diolah dalam sistem deteksi. Pada tahap ini dilakukan *resizing*, *normalization* dan *enhancement*.

4.2.1 Hasil *Resizing*

Proses *preprocessing* tahap awal yaitu *resizing data*. Pada kedua data yang digunakan, memiliki variasi resolusi gambar yang berbeda. Kedua *dataset* akan dilakukan perubahan resolusi menjadi 256 x 256 piksel sehingga keluaran dari proses ini adalah citra dengan resolusi yang sama. Pada *dataset* RSUA, proses *resizing* dilakukan melalui Roboflow. *Dataset* yang telah diberi label akan masuk ke “*version*” untuk *preparing* data sebelum digunakan dalam model. Pada bagian ini, data akan dilakukan *resizing* sesuai dengan resolusi *input layer* pada arsitektur YOLO11. Hasilnya akan di unduh dalam ekstensi zip. Sedangkan pada *dataset* Kvasir-SEG, dilakukan pengkodean dalam *google colaboratory*

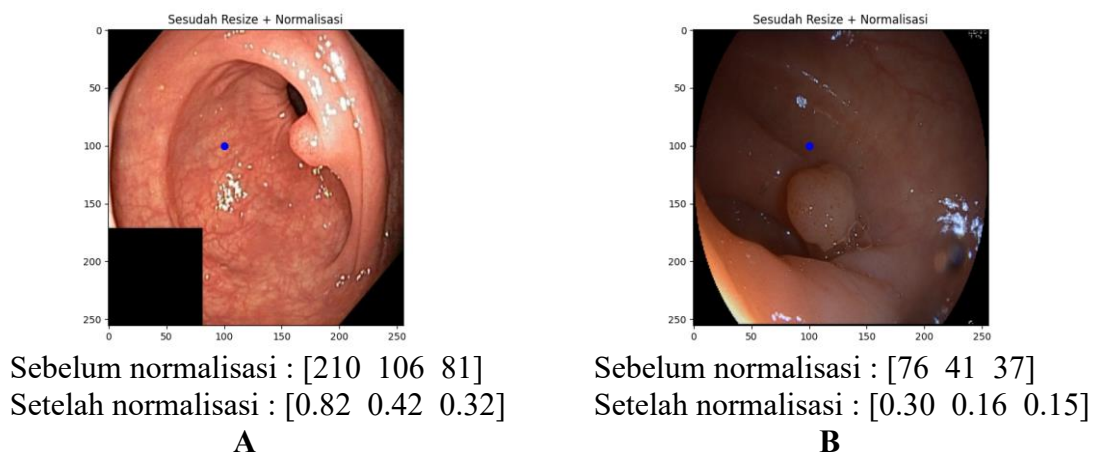
menggunakan *function* opencv. Berikut merupakan perbandingan hasil *resizing dataset* dari Kvasir-SEG dan RSUA.

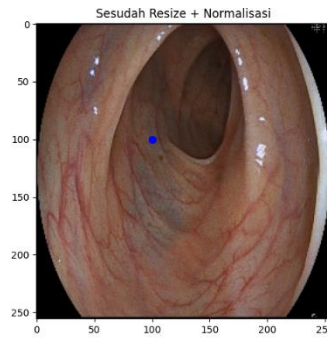


Gambar 4.6 Proses *resizing dataset*, A: Kvasir-SEG, B: Polip RSUA, C: Non-polip RSUA.

4.2.1 Hasil Normalisasi Data

Proses selanjutnya yaitu dilakukan normalisasi data, dimana *dataset* yang akan digunakan dilakukan penyeragaman rentang nilai batas atas dan bawah dari piksel sehingga model yang akan digunakan dapat lebih stabil. Pada gambar digital, biasanya memiliki rentang piksel antara 0 hingga 255 (8-bit). Sedangkan, pada model *deep learning* biasanya bekerja pada rentang angka yang kecil dan stabil. Oleh karena itu, pada proses ini digunakan normalisasi nilai piksel *dataset* diubah menjadi rentang 0.0 sampai 1.0. Berikut merupakan contoh perbandingan hasil sebelum dan sesudah normalisasi data pada koordinat [100,100] yang ditunjukkan dengan titik biru.





Sebelum normalisasi : [85 56 49]

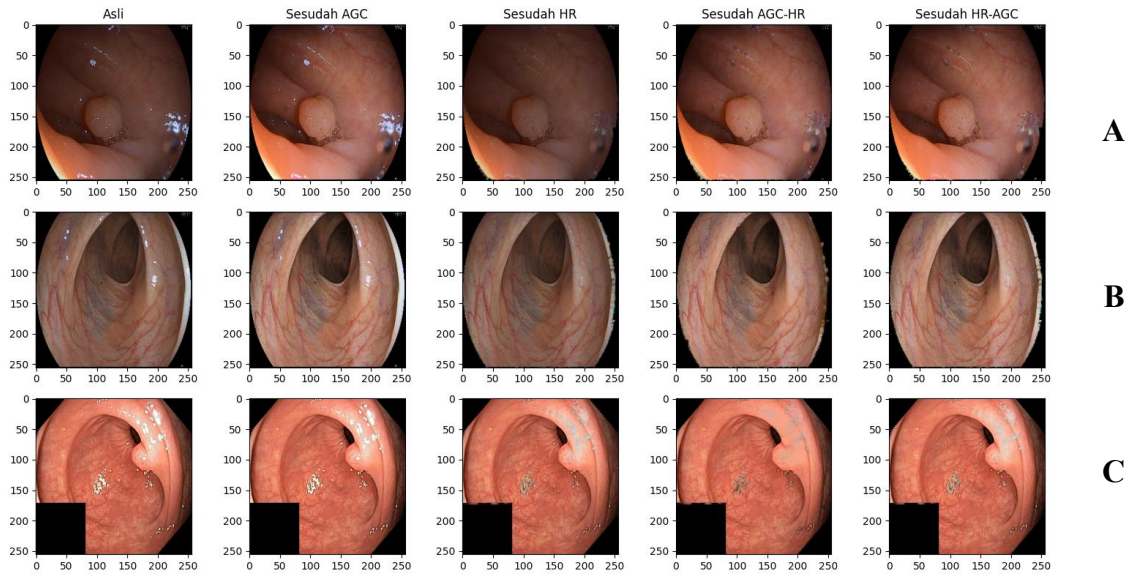
Setelah normalisasi : [0.33 0.22 0.19]

C

Gambar 4.7 Proses normalisasi *dataset*, A: Kvasir-SEG, B: Polip RSUA, C: Non-polip RSUA.

4.2.1 Hasil *Enhancement*

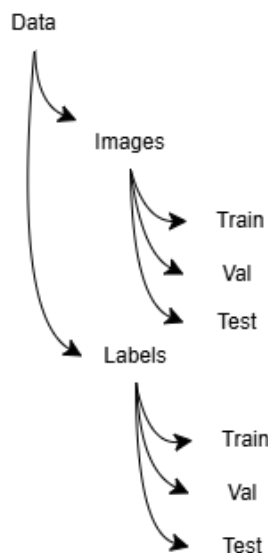
Pada proses *preprocessing*, adanya penerapan *image enhancement* yaitu *Adaptive Gamma Correction* (AGC) dan *Highlight Removal* (HR) untuk meningkatkan kualitas citra masukan model. Pada *dataset* yang digunakan dalam penelitian ini, digunakan dua kombinasi data dengan berbagai variasi informasi pada masing-masing gambar. Oleh karena itu, adanya penggunaan *adaptive threshold* dalam kedua metode *enhancement* untuk mempermudah metode dalam beradaptasi pada setiap variasi gambar yang masuk. AGC digunakan dalam menyetarakan intensitas gambar *dataset* sesuai dengan masing-masing informasi setiap gambar. Adanya klasifikasi kelas digunakan untuk menentukan perhitungan gamma setiap gambar. Di samping itu, pada citra hasil *colonoscopy* terdapat banyak pantulan dari pencahayaan endoskop dengan permukaan *colon* yang berair. Beberapa pantulan dapat menyebabkan distraksi terhadap pendeteksian *colorectal polyp*. Oleh karena itu, adanya perlakuan HR sebagai metode penghapusan area pantulan cahaya tertentu. Dalam penelitian ini, dilakukan kombinasi skenario dari masing-masing metode *enhance* untuk melihat pengaruh perlakuan yang dilakukan. Berikut merupakan hasil dari penerapan *enhancement* pada masing-masing data.



Gambar 4.8 Hasil *enhancement* menggunakan AGC pada, A:Polip RSUA, B: Non-polip RSUA, C: Kvasir-SEG.

4.4. Hasil *Splitting dataset*

Setelah melalui serangkaian proses *preprocessing*, *dataset* akan dilakukan pembagian atau *splitting*. Proses ini merupakan teknik membagi data menjadi 3 bagian seperti pada tabel 3.1. Proses pengelompokan dilakukan melalui *google collab* dengan memasukkan data gambar secara acak ke dalam setiap folder sesuai dengan masing-masing *bounding box*, kecuali citra non-polip karena tidak memiliki *bounding box*. *Splitting dataset* ini disesuaikan dengan *directory* model YOLO11 yang mana disimpan ke dalam *data.yaml*.



Gambar 4.9 *Directory* model YOLO11.

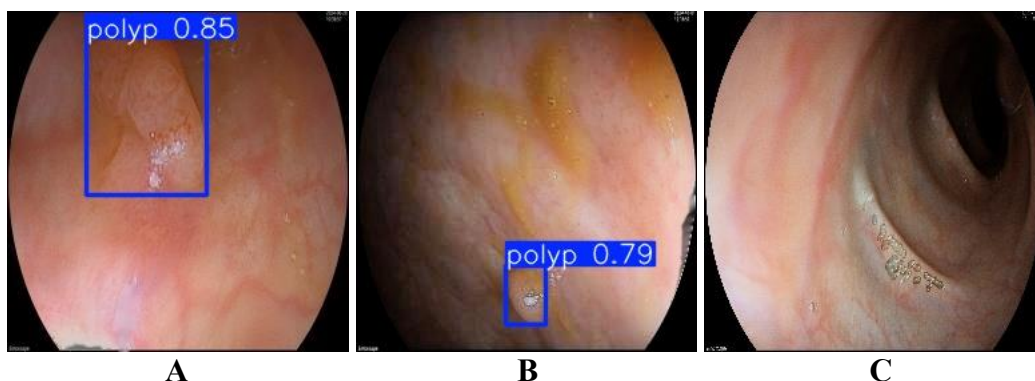
4.5. Hasil Deteksi Citra

Pada penelitian ini dilakukan kombinasi 5 skenario dengan *dataset* yang berbeda yaitu *dataset* asli tanpa *enhancement*, *dataset* yang telah dilakukan AGC, *dataset* yang telah dilakukan HR, *dataset* yang telah dilakukan AGC kemudian HR, dan *dataset* yang telah dilakukan HR kemudian AGC. Pada masing-masing skenario, dilakukan pembuatan *data.yaml* sebagai data masukan dari model YOLO11.

```
names:
- polyp
nc: 1
test: /content/dataset_split_kvrs/images/test
train: /content/dataset_split_kvrs/images/train
val: /content/dataset_split_kvrs/images/val
```

Gambar 4.10 Contoh *data.yaml* YOLO11.

Data.yaml tersebut digunakan dalam masukan *training*, *validation*, dan *testing* dari model. Penggunaan “*yolo11s.pt*” dalam penelitian ini mempertimbangkan keseimbangan antara akurasi dengan kecepatan. Varian YOLO “s” merupakan versi model dengan ukuran “*small*” yang ringan namun memiliki keakuratan yang stabil. Pada proses *training* model, parameter yang dikontrol yaitu *epoch* sebanyak 100 kali, 8 *batch size*, 256 *image size* dengan *processor* GPU. Selain itu, proses *training* menggunakan parameter yang selaras dengan YOLO11 yaitu menggunakan *learning rate* 0.002 dengan *optimizer* AdamW karena model berukuran kecil. Proses *training* akan dievaluasi dengan proses *validation* menggunakan data yang tidak digunakan dalam *training*. *Validation* berguna untuk mengevaluasi apakah model mengalami *overfitting* atau tidak. Bobot yang dihasilkan dalam *training* akan digunakan dalam *testing* untuk mendeteksi citra baru. Berikut merupakan hasil dari deteksi citra polip dengan kombinasi 5 skenario.



Gambar 4.11 Hasil deteksi citra pada A: Kvasir-SEG, B: Polip RSUA, C: Non-polip RSUA.

Pada gambar 4.11 menunjukkan hasil deteksi pada 5 skenario yang menampilkan *bounding box*, *confidence level*, dan kelas pada citra polip *dataset* yang digunakan.

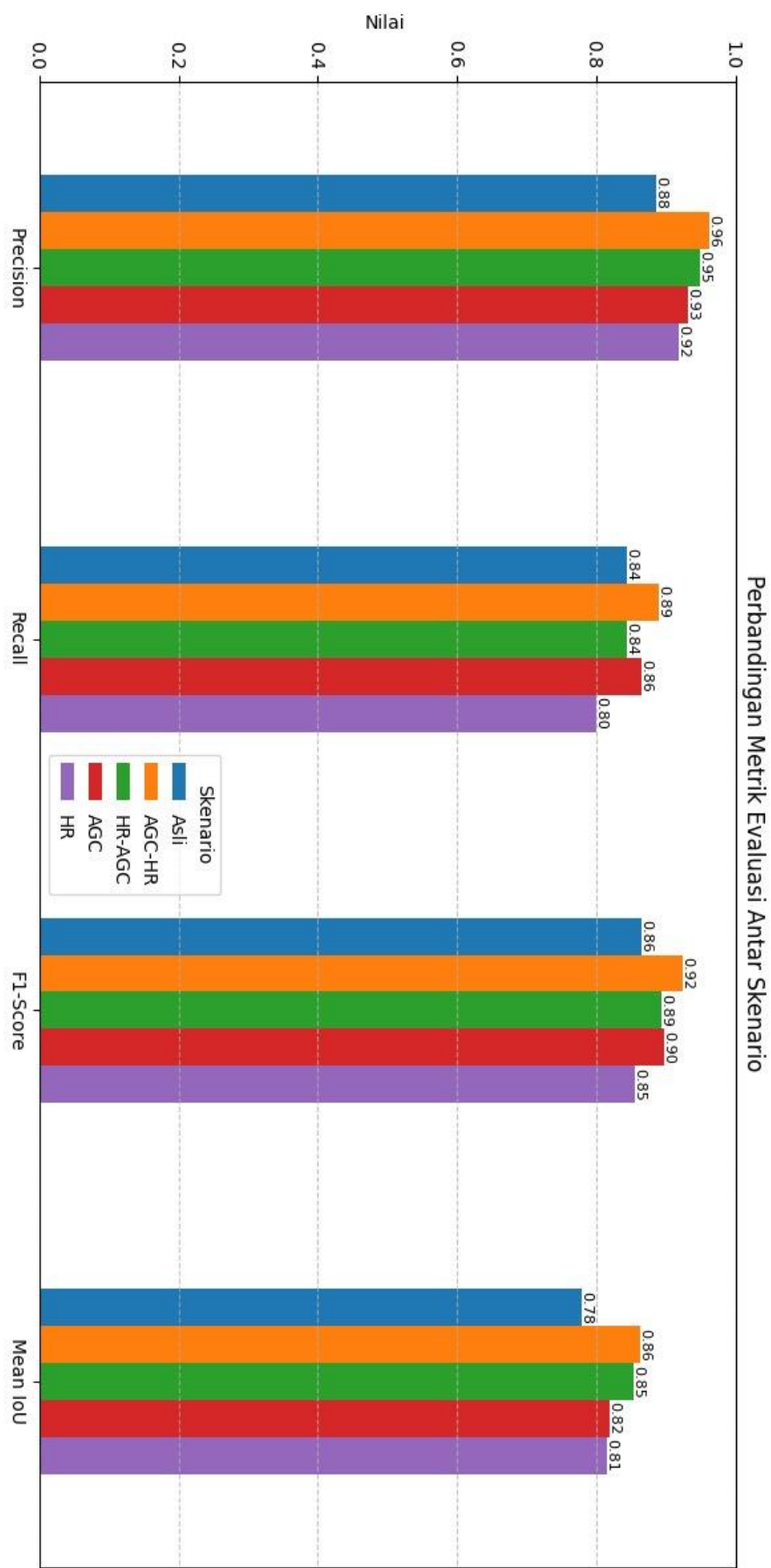
Tabel 4.1 Tabel analisis hasil deteksi citra.

<i>Training</i>			
<i>Dataset</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Asli	0.92	0.88	0.90
AGC	0.91	0.85	0.88
HR	0.93	0.82	0.87
AGC-HR	0.94	0.83	0.88
HR-AGC	0.93	0.87	0.90
<i>Testing</i>			
<i>Dataset</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Asli	0.88	0.84	0.86
AGC	0.93	0.86	0.90
HR	0.92	0.80	0.85
AGC-HR	0.96	0.89	0.92
HR-AGC	0.95	0.84	0.89

Pada tabel 4.1 merupakan hasil performa dari lima kombinasi skenario yang dilakukan pada penelitian. Performa model dianalisa berdasarkan *precision*, *recall*, dan *F1-score* hasil prediksi yang dibandingkan dengan *ground truth* citra.

4.6. Analisa Perbandingan Hasil Performa Deteksi

Pada penelitian ini dilakukan analisa performa model dengan menggunakan *precision*, *recall*, dan *f1-score* pada hasil *testing*. Hasil *testing* menunjukkan hasil yang paling objektif untuk menilai kemampuan model secara nyata, karena data yang digunakan tidak pernah dilihat model sebelumnya. Analisa performa model dilakukan pada 5 skenario *dataset* dengan melakukan perbandingan hasil matriks evaluasi, seperti berikut.



Gambar 4.12 Grafik perbandingan matriks evaluasi antar skenario.

Pada grafik matriks evaluasi performa model, beberapa skenario memiliki *precision* tinggi (>0.90). Hal tersebut menunjukkan bahwa model memiliki kemampuan dalam menghindari *False Positive* yang cukup baik. Namun, terdapat variasi *recall* cukup besar antar skenario. Dalam sistem deteksi, *precision* menunjukkan seberapa akurat model dalam mendeteksi objek yang benar serta *recall* digunakan untuk mengetahui seberapa lengkap model dalam mendeteksi semua objek yang ada. Pada deteksi objek, dibutuhkan keseimbangan antara keduanya (*f1-score*) karena menggambarkan performa yang seimbang antara kemampuan menangkap objek dan menghindari salah deteksi. Adanya analisa *f1-score* yang digunakan untuk mewakili performa model di dunia nyata karena memberikan ukuran menyeluruh dari kualitas deteksi, terutama ketika harus menyeimbangkan antara tidak melewatkan polip (*recall*) dan tidak salah mendeteksi (*precision*).

Pada penentuan performa terbaik, nilai *f1-score* memiliki perbedaan yang kecil, terutama pada skenario HR-AGC dan AGC serta skenario HR dan asli. Oleh karena itu, dilakukan penambahan perhitungan matriks *Mean IoU* pada setiap skenario. *Mean IoU* digunakan dalam menentukan seberapa presisi dan akurat area hasil prediksi *bounding box* dengan *ground truth*. Pada beberapa skenario *dataset* yang telah dilakukan, skenario AGC-HR menunjukkan performa terbaik secara keseluruhan. Perolehan nilai *f1-score* tertinggi sebesar 0.92, menunjukkan bahwa skenario ini merupakan skenario yang paling seimbang antara *precision* dan *recall*-nya. Nilai *recall* pada skenario AGC-HR mencapai 0.89, tertinggi di antara semua skenario, yang menunjukkan kemampuan model dalam mendeteksi objek yang benar secara konsisten. Nilai *mean IoU* yang dicapai termasuk ke dalam kriteria tertinggi dibandingkan skenario lain. Dalam keseluruhan matriks evaluasi, skenario AGC-HR unggul dalam mempertahankan performa yang stabil pada keseluruhan matriks. Performa terendah dari kelima skenario tersebut yaitu asli, dengan *precision* 0.88, *recall* 0.84, *f1-score* 0.85, dan *Mean IoU* 0.78. Berdasarkan nilai *f1-score*, skenario asli memiliki perbedaan yang tipis dengan skenario HR yang artinya skenario asli lebih imbang. Namun nilai *mean IoU* yang dihasilkan sangat rendah jika dibandingkan yang lain. Hal ini mengindikasikan banyaknya objek polip yang tidak berhasil terdeteksi dengan baik karena *overlapping* dari prediksi dengan *groundtruth* sedikit. Oleh karena itu, pemberian *enhancement* sangat penting dalam citra polip untuk memastikan model melihat informatif yang ada. Hal demikian juga menjadikan skenario HR menempati urutan kedua dari bawah. Hasil skenario HR menunjukkan adanya degradasi performa yang cukup signifikan akibat penggunaan *enhancement* HR. Hal ini bisa terjadi karena hilangnya beberapa karakteristik

penting pada citra polip. Degradasi performa menunjukkan bahwa tidak semua *enhancement* berdampak positif pada model.

Sementara itu, skenario HR-AGC menempati urutan performa terbaik kedua, setelah AGC-HR. Skenario ini secara konsisten meningkatkan *precision* (0.95), *recall* (0.84), *f1-score* (0.89), dan *mean IoU* (0.85). Berdasarkan nilai *f1-score*, skenario HR-AGC memiliki perbedaan yang tipis dengan skenario AGC, namun adanya penambahan matriks *mean IoU* menjadikan skenario HR-AGC lebih unggul dan menjadikan skenario AGC menempati urutan ketiga. Kedua skenario ini artinya masih dapat membuat model menjadi kompetitif dan lebih baik dibandingkan *baseline* (asli).

Pada hasil penelitian ini, dilakukan uji normalitas untuk menentukan persebaran distribusi normal atau tidak normal dari data hasil. Uji normalitas dilakukan dengan menggunakan Shapiro-Wilk sebagaimana jumlah data yang digunakan kurang dari 5000 (Mishra dkk., 2019).

Tabel 4.2 Hasil *p-value* uji normalitas.

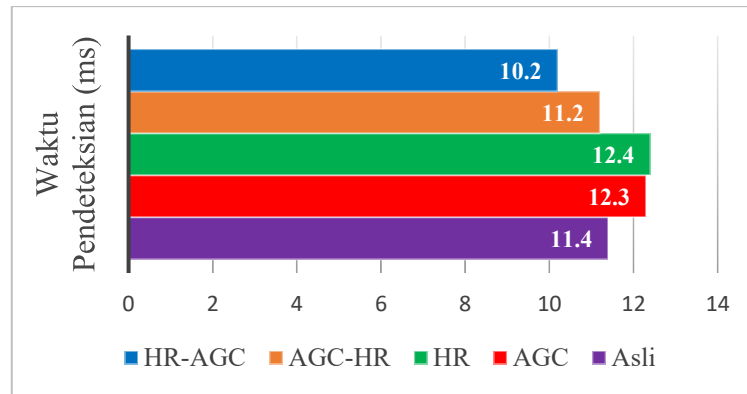
Skenario	Hasil (<i>p-value</i>)
Skenario Asli	9.29416e-15
Skenario AGC-HR	1.68677e-16
Skenario HR-AGC	5.24538e-16
Skenario AGC	2.65935e-15
Skenario HR	5.58492e-15

Pada hasil uji normalitas, semua skenario yang dilakukan pada penelitian menghasilkan persebaran distribusi tidak normal. Nilai *p-value* menunjukkan kurang dari 0.05 *threshold* sehingga mempengaruhi uji lanjutan yang digunakan sebagai pengujian signifikansi. Hasil ujinya sebagai berikut.

Tabel 4.3 Hasil *mann-whitney u test* sebagai uji signifikansi.

Skenario	Hasil Uji
Skenario AGC-HR dengan skenario asli	0.299 (Tidak Signifikan)
Skenario HR-AGC dengan skenario asli	0.552 (Tidak Signifikan)
Skenario AGC dengan skenario asli	0.731 (Tidak Signifikan)
Skenario HR dengan skenario asli	0.851 (Tidak Signifikan)
Skenario AGC-HR dengan scenario HR-AGC	0.612 (Tidak Signifikan)
Skenario AGC-HR dengan scenario AGC	0.122 (Tidak Signifikan)
Skenario AGC-HR dengan scenario HR	0.201 (Tidak Signifikan)
Skenario HR-AGC dengan scenario AGC	0.311 (Tidak Signifikan)
Skenario HR-AGC dengan scenario HR	0.395 (Tidak Signifikan)
Skenario AGC dengan scenario HR	0.825 (Tidak Signifikan)

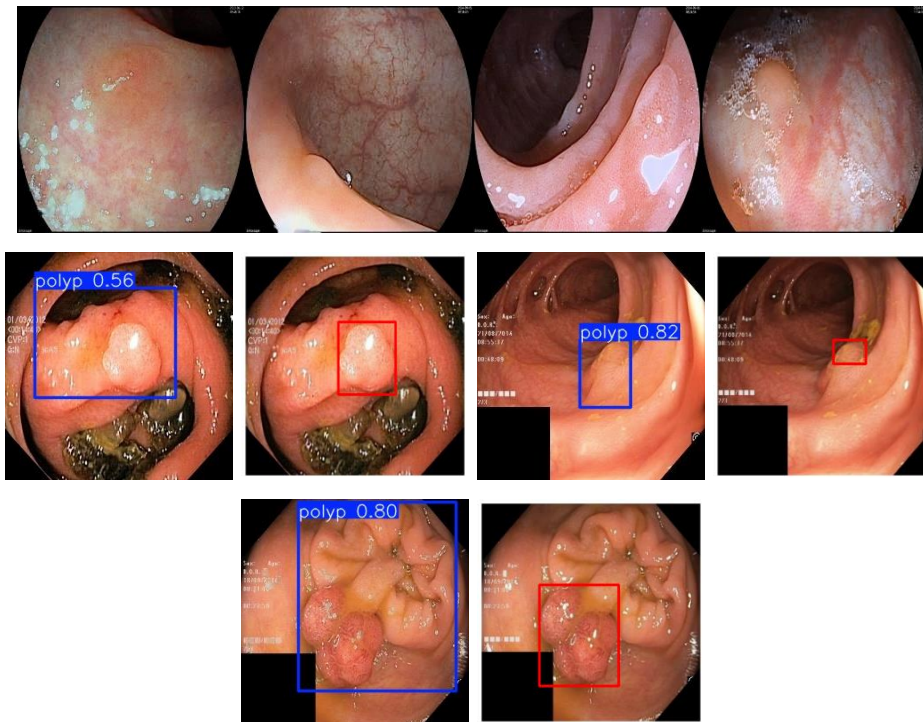
Berdasarkan hasil yang didapatkan, semakin kecil nilai *p-value* maka semakin signifikan hasilnya. Semakin signifikan maka pemilihan scenario performa model terbaik semakin jelas. Pada perbandingan semua skenario menunjukkan nilai yang tidak signifikan. Secara keseluruhan, tidak ada satu pun skenario alternatif yang menunjukkan perbedaan signifikan terhadap skenario asli, sehingga dapat disimpulkan bahwa perubahan dalam skenario-skenario tersebut tidak memberikan pengaruh yang berarti terhadap hasil yang diamati.



Gambar 4.13 Waktu pendeteksian antar skenario model.

Proses pendeteksian pada setiap skenario berkisar selama 11ms untuk setiap citra. Waktu tercepat diperoleh pada skenario HR-AGC yaitu 10.2 ms. Meskipun tercepat, HR-AGC tidak memiliki performa terbaik dalam hal metrik evaluasi. Skenario AGC-HR tetap menjadi skenario terbaik secara keseluruhan karena memberikan kombinasi kecepatan deteksi (11.2 ms) dan performa matriks yang baik. Berdasarkan waktu yang dibutuhkan untuk proses pendeteksian, model dapat diterapkan untuk sistem deteksi *colorectal polyp* secara *real-time* karena waktu terlambat (12.4 ms) pun masih jauh di atas batas minimal *real-time* (≤ 33.3 ms per-frame) (Zhivkov dkk., 2023). Secara keseluruhan, penerapan kombinasi *enhancement* AGC dan HR terbukti mampu meningkatkan performa deteksi model.

Performa model deteksi *colorectal polyp* dapat dipengaruhi oleh beberapa faktor. Pada penelitian ini, terdapat beberapa hasil pendeteksian yang salah pada citra yang memiliki polip *flat*, polip *sessile*, dan polip kecil. Pada hasil pendeteksian, ketiga polip tersebut masuk ke dalam FN, yang berarti ada polip tapi tidak terdeteksi. Selain itu, pada citra yang berkembang bersamaan dengan kondisi patologis lain (pseudopolip), sistem deteksi memiliki nilai IoU di bawah *threshold* sehingga dianggap FP, yang artinya bukan polip tapi terdeteksi. Hal ini dikarenakan oleh kemiripan visual antara pseudopolip dan patalogis lain yang membingungkan model dalam mengenali batas objek secara akurat. Sedangkan berdasarkan hasil evaluasi performa, nilai IoU yang terbaik dimiliki oleh citra polip *pedunculated*. Pada gambar 4.15 merupakan perbandingan *predict bounding box* (biru) dan *groundtruth bounding box* (merah) yang menandakan FP. Selain dari segi morfologi *dataset*, performa model deteksi juga dipengaruhi oleh keseimbangan jumlah data yang digunakan, baik data primer maupun data dari ketiga jenis.



Gambar 4.14 *Dataset dengan performa model yang kurang.*

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian Deteksi *Colorectal Polyp* Berdasarkan Citra *Colonoscopy* Menggunakan *You Only Look Once* (YOLO11) dapat disimpulkan bahwa:

1. Sistem deteksi letak tonjolan *colorectal polyp* dibangun dengan metode YOLO11 melalui tahap pengumpulan data *colonoscopy*, anotasi *dataset*, *training* model YOLO11 yang dievaluasi dengan proses validasi, serta evaluasi performa deteksi dari proses *testing* dengan matriks evaluasi *precision*, *recall*, dan *f1-score*.
2. Model YOLO11s yang digunakan dalam mendeteksi *colorectal polyp* berdasarkan citra *colonoscopy* memiliki performa yang sangat baik berdasarkan matriks evaluasi yang dihasilkan. Skenario AGC-HR menjadi skenario terbaik dengan *precision* 0.96, *recall* 0.89, dan *f1-score* 0.92. Hasil uji signifikansi menunjukkan performa skenario AGC-HR tidak signifikan terhadap skenario asli dengan *p-value* 0.299. Namun, berdasarkan matriks evaluasi skenario AGC-HR menjadi skenario terbaik dibandingkan skenario lain. Hal tersebut menunjukkan bahwa model mampu melakukan deteksi secara akurat dan konsisten.
3. Pembuktian dari pengaruh *enhancement* AGC dan HR terhadap performa YOLO11 untuk mendeteksi *colorectal polyp* dibuktikan dengan uji signifikansi. Secara keseluruhan, tidak ada satu pun skenario alternatif yang menunjukkan perbedaan signifikan terhadap skenario asli, karena nilai *p-value* yang dihasilkan tidak ada yang kurang dari 0.05 sehingga dapat disimpulkan bahwa perubahan dalam skenario-skenario tersebut tidak memberikan pengaruh yang berarti terhadap hasil yang diamati.

5.2. Saran

Berdasarkan analisa dan hasil yang diperoleh pada penelitian ini, terdapat beberapa saran untuk memperbaiki penelitian di masa mendatang, yaitu:

1. Pembangunan model deteksi *colorectal polyp* secara *real-time* sehingga dapat diimplementasikan pada kolonoskop secara langsung saat proses *colonoscopy*.
2. Penyeimbangan *dataset* yang digunakan sebagai *input* model deteksi *colorectal polyp* berupa data polip *sessile*, *flat*, dan kecil sehingga model lebih mampu mendeteksi polip dengan kriteria tersebut.

3. Mempertimbangkan dari ahli gastroenterologi dalam mengkombinasikan variasi *dataset* lain sehingga dapat memaksimalkan performa model.
4. Membandingkan metrik evaluasi performa berdasarkan asal *dataset* sehingga dapat mengetahui performa model pada setiap *dataset* yang digunakan.

DAFTAR PUSTAKA

- Ahmad, A., Marshall, S., Bassett, P., Thiruvilangam, K., Dhillon, A., & Saunders, B. P. (2023). Evaluation of bowel preparation regimens for colonoscopy including a novel low volume regimen (Plenvu): CLEANSE study. *BMJ Open Gastroenterology*, 10(1). <https://doi.org/10.1136/bmjgast-2022-001070>
- Alain, P., Sielenou, D., & Girard, S. (2023). *Detection of Traffic Scene Objects using YOLO Algorithm Theory and Practical Guide*.
- Alfiyan, F. (2020). Improvement of Enhanced Image Using Mean and Standard Deviation Increment Method Based on Visual Representation Statistics. Dalam *Journal of Telematics and Informatics (JTI)* (Vol. 8, Nomor 1).
- Au, W., & Takei, R. (2001). *Image Inpainting With The Navier-Stokes Equations*.
- Basnet, D., Makaju, R., Gurung, R. B., & Dhakal, R. (2021). Colorectal Polyps A Histopathological Study in Tertiary Care Center. *Nepalese Medical Journal*, 4(1), 414–418. <https://doi.org/10.3126/nmj.v4i1.37707>
- Biga, L. M., Bronson, S., Dawson, S., Harwell, A., Hopkins, R., Kaufmann, J., LeMaster, M., Matern, P., Morrison-Graham, K., Oja, K., Quick, D., Runyeon, J., OERU, O., & OpenStax. (2019). 23. the Digestive System. Dalam *Anatomy & Physiology*. OpenStax/Oregon State University.
- Chollet, F. (2018). *Deep Learning with Python*. Manning.
- Cleveland Clinic. (2022, November 30). *Colonoscopy: Prep & Procedure Details*. <https://my.clevelandclinic.org/health/diagnostics/4949-colonoscopy>
- Dijaya, R. (2023). *Buku Ajar Pengolahan Citra Digital* (1 ed.). Umsida Press.
- Elkarazle, K., Raman, V., Then, P., & Chua, C. (2023). Detection of Colorectal Polyps from Colonoscopy Using Machine Learning: A Survey on Modern Techniques. Dalam *Sensors* (Vol. 23, Nomor 3). MDPI. <https://doi.org/10.3390/s23031225>
- Ferlay, J., Colombet, M., Soerjomataram, I., Parkin, D. M., Piñeros, M., Znaor, A., & Bray, F. (2024). *Global Cancer Observatory: Cancer Today*. <https://gco.iarc.who.int/today>

- Florestiyanto, M. Y., Yuwono, bambang, & Prasetyo, D. B. (2022). *Dasar Pengolahan Digital Edisi 2022*. Lembaga Penelitian dan Pengabdian kepada Masyarakat UPN Veteran Yogyakarta.
- García, S., Luengo, J., & Herrera, F. (2015). *Data Preprocessing in Data Mining* (Vol. 72). springer. <https://doi.org/10.1007/978-3-319-10247-4>
- Gonzalez, R. C. ., & Woods, R. E. . (2018). *Digital image processing*. Pearson.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Networks: Modern Practices. Dalam *Deep Learning*. MIT Press. <https://www.deeplearningbook.org/>
- Hall, J. E., & Hall, M. E. (2021). *Guyton and Hall Textbook of Medical Physiology 14th Ed* (14 ed.). Elsevier.
- Hossain, M. S., Rahman, M. M., Syeed, M. M., Uddin, M. F., Hasan, M., Hossain, M. A., Ksibi, A., Jamjoom, M. M., Ullah, Z., & Samad, M. A. (2023). DeepPoly: Deep Learning-Based Polyps Segmentation and Classification for Autonomous Colonoscopy Examination. *IEEE Access*, 11, 95889–95902. <https://doi.org/10.1109/ACCESS.2023.3310541>
- Huck, M. B., & Bohl, J. L. (2016). Colonic Polyps: Diagnosis and Surveillance. *Clinics in Colon and Rectal Surgery*, 29(4), 296–305. <https://doi.org/10.1055/s-0036-1584091>
- Jana, A. P., Biswas, A., & Mohana. (2018). YOLO based Detection and Classification of Objects in video records. *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT-2018)*. <https://doi.org/doi:10.1109/rteict42901.2018.9012375>
- Jegham, N., Koh, C. Y., Abdelatti, M., & Hendawi, A. (2024). *Evaluating the Evolution of YOLO (You Only Look Once) Models: A Comprehensive Benchmark Study of YOLO11 and Its Predecessors*. <http://arxiv.org/abs/2411.00201>
- Kaltenbach, T., Anderson, J. C., Burke, C. A., Dominitz, J. A., Gupta, S., Lieberman, D., Robertson, D. J., Shaikat, A., Syngal, S., & Rex, D. K. (2020). Endoscopic Removal of Colorectal Lesions—Recommendations by the US Multi-Society Task Force on Colorectal Cancer. *Gastroenterology*, 158(4), 1095–1129. <https://doi.org/10.1053/j.gastro.2019.12.018>

- Khanam, R., & Hussain, M. (2024). *YOLOv11: An Overview of the Key Architectural Enhancements*. <http://arxiv.org/abs/2410.17725>
- Lalinia, M., & Sahafi, A. (2024). Colorectal polyp detection in colonoscopy images using YOLO-V8 network. *Signal, Image and Video Processing*, 18(3), 2047–2058. <https://doi.org/10.1007/s11760-023-02835-1>
- Li, K., Fathan, M. I., Patel, K., Zhang, T., Zhong, C., Bansal, A., Rastogi, A., Wang, J. S., & Wang, G. (2021). Colonoscopy polyp detection and classification: Dataset creation and comparative evaluations. *PLoS ONE*, 16(8 August). <https://doi.org/10.1371/journal.pone.0255809>
- Lwanga, S. K., Lemeshow, S., & World Health Organization. (1991). *Sample Size determination in Health Studies*. 0. <https://iris.who.int/handle/10665/40062>
- Mishra, P., Pandey, C. M., Singh, U., Gupta, A., Sahu, C., & Keshri, A. (2019). Descriptive statistics and normality tests for statistical data. *Annals of Cardiac Anaesthesia*, 22(1), 67–72. https://doi.org/10.4103/aca.ACA_157_18
- Nelson, J. (2024, Juli 17). *What is YOLO? The Ultimate Guide [2024]*. Roboflow. <https://blog.roboflow.com/guide-to-yolo-models/>
- Nie, C., Xu, C., Li, Z., Chu, L., & Hu, Y. (2023). Specular Reflections Detection and Removal for Endoscopic Images Based on Brightness Classification. *Sensors*, 23(2). <https://doi.org/10.3390/s23020974>
- Ogobuiro, I., Gonzales, J., Shumway, K. R., & Tuma, F. (2023). Physiology, Gastrointestinal. *StatPearls*. <https://www.ncbi.nlm.nih.gov/books/NBK537103/>
- Padilla, R., Netto, S. L., Da Silva, E. A. B., & Netto, S. L. (2020). *A Survey on Performance Metrics for Object-Detection Algorithms*. <https://doi.org/10.1109/IWSSIP48289.2020>
- PAHO. (2021). *Burden of Digestive Diseases - PAHO/WHO | Pan American Health Organization*. <https://www.paho.org/en/noncommunicable-diseases-and-mental-health/noncommunicable-diseases-and-mental-health-data-38>
- Qi, Y., Yang, Z., Sun, W., Lou, M., Lian, J., Zhao, W., Deng, X., & Ma, Y. (2022). A Comprehensive Overview of Image Enhancement Techniques. Dalam *Archives of*

- Computational Methods in Engineering* (Vol. 29, Nomor 1, hlm. 583–607). Springer Science and Business Media B.V. <https://doi.org/10.1007/s11831-021-09587-6>
- Quraissy, A., & Madya, S. (2021). Analisis Nonparametrik Mann Whitney Terhadap Perbedaan Kemampuan Pemecahan Masalah Menggunakan Model Pembelajaran Problem Based Learning. *VARIANSI: Journal of Statistics and Its Application on Teaching and Research*, 3(1), 51–57. <https://doi.org/10.35580/variensiunm23810>
- Ragab, M. G., Abdulkadir, S. J., Muneer, A., Alqushaibi, A., Sumiea, E. H., Qureshi, R., Al-Selwi, S. M., & Alhussian, H. (2024). A Comprehensive Systematic Review of YOLO for Medical Object Detection (2018 to 2023). *IEEE Access*, 12, 57815–57836. <https://doi.org/10.1109/ACCESS.2024.3386826>
- Rahman, S., Rahman, M. M., Abdullah-Al-Wadud, M., Al-Quaderi, G. D., & Shoyaib, M. (2016). An adaptive gamma correction for image enhancement. *Eurasip Journal on Image and Video Processing*, 2016(1). <https://doi.org/10.1186/s13640-016-0138-1>
- Rao, J. N., & Wang, J.-Y. (2010). *Intestinal Architecture and Development*. <https://www.ncbi.nlm.nih.gov/books/NBK54098/>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December*, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. Dalam *SN Computer Science* (Vol. 2, Nomor 3). Springer. <https://doi.org/10.1007/s42979-021-00592-x>
- Sayuti, M. (2019). KANKER KOLOREKTAL. Dalam *Jurnal Averrous* (Vol. 5, Nomor 2).
- Shamsudhin, N., Zverev, V. I., Keller, H., Pane, S., Egolf, P. W., Nelson, B. J., & Tishin, A. M. (2017). Magnetically guided capsule endoscopy: Dalam *Medical Physics* (Vol. 44, Nomor 8, hlm. e91–e111). John Wiley and Sons Ltd. <https://doi.org/10.1002/mp.12299>
- Shaukat, A., Kaltenbach, T., Dominitz, J. A., Robertson, D. J., Anderson, J. C., Cruise, M., Burke, C. A., Gupta, S., Lieberman, D., Syngal, S., & Rex, D. K. (2020). Endoscopic Recognition and Management Strategies for Malignant Colorectal Polyps:

- Recommendations of the US Multi-Society Task Force on Colorectal Cancer. *Gastroenterology*, 159(5), 1916-1934.e2. <https://doi.org/10.1053/j.gastro.2020.08.050>
- Shen, D. F., Guo, J. J., Lin, G. S., & Lin, J. Y. (2020). Content-aware specular reflection suppression based on adaptive image inpainting and neural network for endoscopic images. *Computer Methods and Programs in Biomedicine*, 192. <https://doi.org/10.1016/j.cmpb.2020.105414>
- Stauffer, C. M., & Pfeifer, C. (2023). Colonoscopy. *Operative Pediatric Surgery: Sixth Edition*, 601–609. <https://doi.org/10.1201/b13237-70>
- Treuting, P. M., Arends, M. J., & Dintzis, S. M. (2018a). Lower Gastrointestinal Tract. Dalam *Comparative Anatomy and Histology: A Mouse, Rat, and Human Atlas, Second Edition* (hlm. 213–228). Elsevier. <https://doi.org/10.1016/B978-0-12-802900-8.00012-9>
- Treuting, P. M., Arends, M. J., & Dintzis, S. M. (2018b). Upper Gastrointestinal Tract. Dalam *Comparative Anatomy and Histology: A Mouse, Rat, and Human Atlas, Second Edition* (hlm. 191–211). Elsevier. <https://doi.org/10.1016/B978-0-12-802900-8.00011-7>
- Ultralytics. (2024, September 27). *YOLO11 NEW - Ultralytics YOLO Docs*. <https://docs.ultralytics.com/models/yolo11/>
- Van der Meulen, M. P., Lansdorp-Vogelaar, I., Lucas Goede, S., Kuipers, E. J., Dekker, E., Stoker, J., & Van Ballegooijen, M. (2018). Colorectal cancer: Cost-effectiveness of colonoscopy versus CT colonography screening with participation rates and costs. Dalam *Radiology* (Vol. 287, Nomor 3, hlm. 901–911). Radiological Society of North America Inc. <https://doi.org/10.1148/radiol.2017162359>
- World Health Organization. (2023, Juli 11). *Colorectal cancer*. <https://www.who.int/news-room/fact-sheets/detail/colorectal-cancer>
- Wu, C. H., & Su, M. Y. (2017). Specular Highlight Detection from Endoscopic Images for Shape Reconstruction. *Applied Mechanics and Materials*, 870, 357–362. <https://doi.org/10.4028/www.scientific.net/amm.870.357>
- Yasmin, F., Hassan, M. M., Hasan, M., Zaman, S., Bairagi, A. K., El-Shafai, W., Fouad, H., & Chun, Y. C. (2023). GastroNet: Gastrointestinal Polyp and Abnormal Feature

Detection and Classification With Deep Learning Approach. *IEEE Access*, 11, 97605–97624. <https://doi.org/10.1109/ACCESS.2023.3312729>

Yu, B., Chen, W., Zhong, Q., & Zhang, H. (2021). Specular Highlight Detection Based on Color Distribution for Endoscopic Images. *Frontiers in Physics*, 8. <https://doi.org/10.3389/fphy.2020.616930>

Zhang, G., Lin, J., Cao, E., Pang, Y., & Sun, W. (2022). A Medical Endoscope Image Enhancement Method Based on Improved Weighted Guided Filtering. *Mathematics*, 10(9). <https://doi.org/10.3390/math10091423>

Zhivkov, T., Sklar, E. I., Botting, D., & Pearson, S. (2023). 5G on the Farm: Evaluating Wireless Network Capabilities and Needs for Agricultural Robotics. *Machines*, 11(12). <https://doi.org/10.3390/machines11121064>

LAMPIRAN

Lampiran 1. Kodingan mengubah koordinat *bounding boxes* JSON ke .txt YOLO

```

import json
import os

def convert_json_to_yolo(json_path, output_dir):

    os.makedirs(output_dir, exist_ok=True)

    with open(json_path, "r") as f:
        data = json.load(f)

    image_dict = {img["id"]: img["file_name"] for img in data["images"]}

    for ann in data["annotations"]:
        image_id = ann["image_id"]
        bbox = ann["bbox"]
        category_id = ann["category_id"]

        if category_id == 1:
            category_id = 0

        image_file = image_dict.get(image_id, None)
        if image_file is None:
            continue

        image_width = 256
        image_height = 256

        x_center = (bbox[0] + bbox[2] / 2) / image_width
        y_center = (bbox[1] + bbox[3] / 2) / image_height
        width = bbox[2] / image_width
        height = bbox[3] / image_height

        output_file = os.path.join(output_dir, image_file.replace(".jpg",
".txt"))
        with open(output_file, "a") as f:
            f.write(f"{category_id} {x_center:.6f} {y_center:.6f} {width:.6f}
{height:.6f}\n")

    print(f"Konversi selesai! Hasil disimpan di {output_dir}")

```

Lampiran 2. Kodingan fungsi AGC

```

def adaptive_gamma_correction(image):
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    v_channel = hsv[:, :, 2] / 255.0

    mean_intensity = np.mean(v_channel) # u/ Rata-rata kecerahan gambar.
    std_dev = np.std(v_channel) # u/ mengukur kontras

    tau = 3 # menentukan ambang batas dalam mengklasifikasikan gambar.

    if std_dev <= 1 / tau:
        image_class = 'low-contrast'
    else:
        image_class = 'high-contrast'

    if image_class == 'low-contrast':
        gamma = -math.log2(std_dev)
    else:

```

```

gamma = math.exp((1 - (mean_intensity + std_dev)) / 2)

if mean_intensity >= 0.8:
    c = 1 # faktor skala untuk memastikan hasil koreksi tetap dalam
    rentang yang dapat ditampilkan.
else:
    heaviside = 1 if 0.8 - mean_intensity > 0 else 0
    k = v_channel**gamma + (1 - v_channel**gamma) *
    (mean_intensity**gamma) # fungsi sigmoid untuk memperhalus hasil koreksi.
    c = 1 / (1 + heaviside * (k - 1))

corrected_v = c * np.power(v_channel, gamma)
corrected_v = np.clip(corrected_v * 255, 0, 255).astype(np.uint8)

hsv[:, :, 2] = corrected_v
corrected_image = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

return corrected_image

```

Lampiran 3. Kodingan fungsi HR

```

def adaptive_highlight_detection(image, Ts_base=0.1, Tv_base=0.8, T2_base=1.2):

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY).astype(float)
    hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    h, s, v = cv2.split(hsv_image)
    s = s / 255.0
    v = v / 255.0

    mean_intensity = np.mean(gray)
    std_intensity = np.std(gray)
    min_intensity, max_intensity = np.min(gray), np.max(gray)

    T1 = min_intensity + (max_intensity - min_intensity) * 0.85
    T1 /= 255.0

    Ts = Ts_base + 0.08 * (std_intensity / 255.0) #kontras
    Tv = Tv_base + 0.5 * (mean_intensity / 255.0) #rata-rata brightness
    T2 = T2_base * (1 + 0.1 * (std_intensity / mean_intensity)) #distribusi
    intensitas

    norm_image = gray / 255.0

    abs_highlights = (norm_image > T1) & (norm_image > Ts) & (norm_image < Tv)

    G = image[:, :, 1].astype(float) / (gray + 1e-5)
    B = image[:, :, 0].astype(float) / (gray + 1e-5)

    rel_highlights = (G > T1 * T2) | (B > T1 * T2)

    highlight_mask = np.logical_or(abs_highlights,
    rel_highlights).astype(np.uint8) * 255

    return highlight_mask

```

Lampiran 4. Kodingan contoh penerapan AGC pada dataset Kvasir-SEG

```

from google.colab import files

images_path = '/content/kvasir_seg_1_gt_removed/kvasir-seg/Kvasir-SEG/images'

```

```

image_files = [f for f in os.listdir(images_path) if f.endswith(('png', 'jpg',
'.jpeg'))]

output_folder_AGC = '/content/output_AGC_kvasir_seg_lgt_removed'
output_enhance_AGC = '/content/output_AGC_kvasir_seg_lgt_removed/images'

os.makedirs(output_folder_AGC, exist_ok=True)
os.makedirs(output_enhance_AGC, exist_ok=True)

target_size = (256,256)
for i, img in enumerate(image_files):
    img_path = os.path.join(images_path, img)

    input_image = cv2.imread(img_path)
    resized_image = cv2.resize(input_image, target_size)
    normal_sed = simple_normalization(resized_image)

    output_AGC = adaptive_gamma_correction(normal_sed)

    repaired_path = os.path.join(output_enhance_AGC, f'{img}.jpg')

    cv2.imwrite(repaired_path, output_AGC)

zip_output = '/content/output_AGC_kvasir_seg_lgt_removed.zip'
output_folder = '/content/output_AGC_kvasir_seg_lgt_removed'

shutil.make_archive(zip_output.replace('.zip', ''), 'zip', output_folder)

files.download(zip_output)

```

Lampiran 5. Kodingan contoh penerapan HR pada *dataset* Kvasir-SEG

```

from google.colab import files

images_path = '/content/kvasir_seg_lgt_removed/kvasir-seg/Kvasir-SEG/images'

image_files = [f for f in os.listdir(images_path) if f.endswith(('png', 'jpg',
'.jpeg'))]

output_folder_HR = '/content/output_HR_kvasir_seg_lgt_removed'
output_enhance_HR = '/content/output_HR_kvasir_seg_lgt_removed/images'

os.makedirs(output_folder_HR, exist_ok=True)
os.makedirs(output_enhance_HR, exist_ok=True)

target_size = (256,256)
for i, img in enumerate(image_files):
    img_path = os.path.join(images_path, img)

    input_image = cv2.imread(img_path)
    resized_image = cv2.resize(input_image, target_size)
    normal_sed = simple_normalization(resized_image)

    highlight_mask = adaptive_highlight_detection(normal_sed)

    repaired_image_NS = cv2.inpaint(normal_sed, highlight_mask, inpaintRadius=3,
flags=cv2.INPAINT_NS)

    repaired_path = os.path.join(output_enhance_HR, f'{img}.jpg')

    cv2.imwrite(repaired_path, repaired_image_NS)

zip_output = '/content/output_HR_kvasir_seg_lgt_removed.zip'
output_folder = '/content/output_HR_kvasir_seg_lgt_removed'

shutil.make_archive(zip_output.replace('.zip', ''), 'zip', output_folder)

```



```
files.download(zip_output)
```

Lampiran 6. Kodingan contoh penerapan AGC-HR pada *dataset* Kvasir-SEG

```
from google.colab import files

images_path = '/content/kvasir_seg_1_gt_removed/kvasir-seg/Kvasir-SEG/images'

image_files = [f for f in os.listdir(images_path) if f.endswith(('png', 'jpg', 'jpeg'))]

output_folder = '/content/output_AGC_HR_kvasir_seg_1gt_removed'
output_enhance = '/content/output_AGC_HR_kvasir_seg_1gt_removed/images'

os.makedirs(output_folder, exist_ok=True)
os.makedirs(output_enhance, exist_ok=True)

target_size = (256,256)
for i, img in enumerate(image_files):
    img_path = os.path.join(images_path, img)

    input_image = cv2.imread(img_path)
    resized_image = cv2.resize(input_image, target_size)
    normal_sed = simple_normalization(resized_image)

    output_AGC = adaptive_gamma_correction(normal_sed)

    highlight_mask = adaptive_highlight_detection(output_AGC)

    repaired_image_NS = cv2.inpaint(output_AGC, highlight_mask, inpaintRadius=3,
    flags=cv2.INPAINT_NS)

    repaired_path = os.path.join(output_enhance, f'{img}')

    cv2.imwrite(repaired_path, repaired_image_NS)

zip_output = '/content/output_AGC_HR_kvasir_seg_1gt_removed.zip'
output_folder = '/content/output_AGC_HR_kvasir_seg_1gt_removed'

shutil.make_archive(zip_output.replace('.zip', ''), 'zip', output_folder)

files.download(zip_output)
```

Lampiran 7. Kodingan contoh penerapan HR-AGC pada *dataset* Kvasir-SEG

```
from google.colab import files

images_path = '/content/kvasir_seg_1_gt_removed/kvasir-seg/Kvasir-SEG/images'

image_files = [f for f in os.listdir(images_path) if f.endswith(('png', 'jpg', 'jpeg'))]

output_folder = '/content/output_HR_AGC_kvasir_seg_1gt_removed'
output_enhance = '/content/output_HR_AGC_kvasir_seg_1gt_removed/images'

os.makedirs(output_folder, exist_ok=True)
os.makedirs(output_enhance, exist_ok=True)

target_size = (256,256)
for i, img in enumerate(image_files):
    img_path = os.path.join(images_path, img)

    input_image = cv2.imread(img_path)
```

```

resized_image = cv2.resize(input_image, target_size)
normal_sed = simple_normalization(resized_image)

highlight_mask = adaptive_highlight_detection(normal_sed)

repaired_image_NS = cv2.inpaint(normal_sed, highlight_mask, inpaintRadius=3,
flags=cv2.INPAINT_NS)

output_AGC = adaptive_gamma_correction(repaired_image_NS)

repaired_path = os.path.join(output_enhance, f'{img}.png')

cv2.imwrite(repaired_path, output_AGC)

zip_output = '/content/output_SRR_AGC_kvasir_seg_lgt_removed.zip'
output_folder = '/content/output_SRR_AGC_kvasir_seg_lgt_removed'

shutil.make_archive(zip_output.replace('.zip', ''), 'zip', output_folder)

files.download(zip_output)

```

Lampiran 8. Kodingan fungsi *splitting dataset*

```

import pandas as pd
import os
import random
from glob import glob
import shutil

def split_dataset1(images_path, labels_path, output_path, train_ratio=0.7,
val_ratio=0.2, test_ratio=0.1):

    os.makedirs(os.path.join(output_path, 'images', 'train'), exist_ok=True)
    os.makedirs(os.path.join(output_path, 'images', 'val'), exist_ok=True)
    os.makedirs(os.path.join(output_path, 'images', 'test'), exist_ok=True)
    os.makedirs(os.path.join(output_path, 'labels', 'train'), exist_ok=True)
    os.makedirs(os.path.join(output_path, 'labels', 'val'), exist_ok=True)
    os.makedirs(os.path.join(output_path, 'labels', 'test'), exist_ok=True)

    image_files = glob(os.path.join(images_path, "*.jpg")) +
glob(os.path.join(images_path, "*.png"))

    assert len(image_files) > 0, "Tidak ada file gambar yang ditemukan."

    random.shuffle(image_files)

    total_images = len(image_files)
    train_count = int(train_ratio * total_images)
    val_count = int(val_ratio * total_images)

    train_images = image_files[:train_count]
    val_images = image_files[train_count:train_count + val_count]
    test_images = image_files[train_count + val_count:]

    def copy_files(image_set, subset):
        for img_file in image_set:
            shutil.copy(img_file, os.path.join(output_path, 'images', subset,
os.path.basename(img_file)))

            label_file_path = os.path.join(labels_path,
os.path.splitext(os.path.basename(img_file))[0] + '.txt')
            if os.path.exists(label_file_path):
                shutil.copy(label_file_path, os.path.join(output_path, 'labels',
subset, os.path.basename(label_file_path)))
            else:

```

```

        print(f"Warning: Label tidak ditemukan untuk {img_file}")

    copy_files(train_images, 'train')
    copy_files(val_images, 'val')
    copy_files(test_images, 'test')

    print(f"Dataset dibagi menjadi train ({len(train_images)}), val
    ({len(val_images)}), dan test ({len(test_images)})")

```

Lampiran 9. Kodingan membuat data.yaml

```

import yaml

dataset = {
    'train': '/content/dataset_split_asli_kvrs_lgt_rem/images/train',
    'val': '/content/dataset_split_asli_kvrs_lgt_rem/images/val',
    'test': '/content/dataset_split_asli_kvrs_lgt_rem/images/test',
    'nc': 1,
    'names': ['polyp']
}

yaml_file = '/content/dataset.yaml'

with open(yaml_file, 'w') as file:
    yaml.dump(dataset, file)

print(f"File YAML dataset berhasil dibuat: {yaml_file}")

```

Lampiran 10. Kodingan memanggil model YOLO11

```

!pip install ultralytics
import ultralytics
ultralytics.checks()
from ultralytics import YOLO

```

Lampiran 11. Kodingan *training* model YOLO11

```

model = YOLO("yolo11s.pt")

train_results = model.train(
    data="/content/dataset.yaml",
    epochs=100,
    imgsz=256,
    batch=8,
    device=0,
    project="/content/runs/detect",
    name="train",
    exist_ok=True
)

```

Lampiran 12. Kodingan *validation* model YOLO11

```

!yolo task=detect mode=val model=/content/runs/detect/train/weights/best.pt
data=/content/dataset1.yaml project=/content/runs/detect name=val conf=0.5
iou=0.5

```

Lampiran 13. Kodingan *testing* model YOLO11

```
!yolo task=detect mode=predict model=/content/runs/detect/train/weights/best.pt
conf=0.5 iou=0.5 source=/content/dataset_split_asli kvrs_lgt_rem/images/test
project=/content/runs/detect name=predict save=True save_txt=True save_conf=True
```

Lampiran 14. Kodingan evaluasi performa model YOLO11

```
import os
import glob
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from collections import defaultdict

def compute_iou(box1, box2):
    def to_corners(box):
        x, y, w, h = box
        return [x - w/2, y - h/2, x + w/2, y + h/2]

    box1 = to_corners(box1)
    box2 = to_corners(box2)

    x11 = max(box1[0], box2[0])
    y11 = max(box1[1], box2[1])
    x12 = min(box1[2], box2[2])
    y12 = min(box1[3], box2[3])

    inter_area = max(0, x12 - x11) * max(0, y12 - y11)
    box1_area = (box1[2] - box1[0]) * (box1[3] - box1[1])
    box2_area = (box2[2] - box2[0]) * (box2[3] - box2[1])

    union_area = box1_area + box2_area - inter_area
    return inter_area / union_area if union_area != 0 else 0

def load_labels(label_path):
    labels = []
    with open(label_path, 'r') as f:
        for line in f:
            parts = line.strip().split()
            cls = int(parts[0])
            bbox = list(map(float, parts[1:5]))
            labels.append((cls, bbox))
    return labels

def evaluate_with_iou_list(pred_dir, gt_dir, iou_threshold=0.5):
    predictions = []
    for pred_file in glob.glob(f"{pred_dir}/*.txt"):
        filename = os.path.basename(pred_file)
        image_id = filename.replace('.txt', '')
        gt_file = os.path.join(gt_dir, filename)

        pred_labels = []
        with open(pred_file, 'r') as f:
            for line in f:
                parts = line.strip().split()
                cls = int(parts[0])
                bbox = list(map(float, parts[1:5]))
                conf = float(parts[5]) if len(parts) > 5 else 1.0
                pred_labels.append((image_id, cls, conf, bbox))
        predictions.extend(pred_labels)

    predictions.sort(key=lambda x: x[2], reverse=True)
    class_gt_counts = defaultdict(int)
    gt_boxes = defaultdict(list)
```

```

for gt_file in glob.glob(f"{gt_dir}/*.txt"):
    image_id = os.path.basename(gt_file).replace('.txt', '')
    for cls, bbox in load_labels(gt_file):
        gt_boxes[(image_id, cls)].append({'bbox': bbox, 'used': False})
        class_gt_counts[cls] += 1

stats = []
iou_detailed = []
acc_tp = 0
acc_fp = 0

for image_id, cls, conf, pred_box in predictions:
    matched = False
    best_iou = 0
    best_gt = None

    candidates = gt_boxes.get((image_id, cls), [])
    for gt in candidates:
        if not gt['used']:
            iou = compute_iou(pred_box, gt['bbox'])
            if iou >= iou_threshold and iou > best_iou:
                best_iou = iou
                best_gt = gt

    if best_gt is not None:
        best_gt['used'] = True
        matched = True
        iou_detailed.append([image_id, cls, conf, best_iou])
    else:
        iou_detailed.append([image_id, cls, conf, 0.0])

    tp = 1 if matched else 0
    fp = 0 if matched else 1
    acc_tp += tp
    acc_fp += fp

    precision = acc_tp / (acc_tp + acc_fp) if (acc_tp + acc_fp) > 0 else 0
    recall = acc_tp / sum(class_gt_counts.values()) if
sum(class_gt_counts.values()) > 0 else 0

    stats.append([image_id, cls, conf, tp, fp, acc_tp, acc_fp, precision,
recall])

df_metrics = pd.DataFrame(stats, columns=["Image", "Class", "Confidence",
"TP", "FP", "Acc TP", "Acc FP", "Precision", "Recall"])
df_iou_detail = pd.DataFrame(iou_detailed, columns=["Image", "Class",
"Confidence", "IoU"])

return df_metrics, df_iou_detail

pred_dir = '/content/predict_asli_lgt_rem/labels'
gt_dir = '/content/dataset_split_asli_kvrs_lgt_rem/labels/test'

df_metrics, df_iou_detail = evaluate_with_iou_list(pred_dir, gt_dir)
df_metrics.to_csv("/content/ detection_metrics_asli_lgt_rem.csv", index=False)
df_iou_detail.to_csv("/content/IOUasli.csv", index=False)

def count_total_groundtruth(gt_dir):
    total_gt = 0
    for file_name in os.listdir(gt_dir):
        if file_name.endswith('.txt'):
            file_path = os.path.join(gt_dir, file_name)
            with open(file_path, 'r') as f:
                lines = f.readlines()
                total_gt += len(lines)
    return total_gt
gt_dir = '/content/dataset_split_asli_kvrs_lgt_rem/labels/test'

```

```

total_gt = count_total_groundtruth(gt_dir)
print(f"Total Ground Truth: {total_gt}")

def compute_final_metrics(df: pd.DataFrame, total_gt: int):
    tp = df['TP'].sum()
    fp = df['FP'].sum()
    fn = total_gt - tp

    precision = tp / (tp + fp) if (tp + fp) > 0 else 0
    recall = tp / (tp + fn) if (tp + fn) > 0 else 0
    f1 = 2 * (precision * recall) / (precision + recall) if (precision + recall) > 0 else 0

    return {
        'True Positive (TP)': tp,
        'False Positive (FP)': fp,
        'False Negative (FN)': fn,
        'Precision': round(precision, 4),
        'Recall': round(recall, 4),
        'F1 Score': round(f1, 4)
    }

final_result = compute_final_metrics(df_metrics, total_gt)

print(f"Final Precision: {final_result['Precision']}")
print(f"Final Recall: {final_result['Recall']}")
print(f"Final F1-Score: {final_result['F1 Score']}")

```

Lampiran 15. Kodingan uji normalisasi dengan Shapiro-Wilk

```

import pandas as pd
import scipy.stats as stats
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import statsmodels.api as sm

files = {
    "IOUAGCHR": "IOUAGCHR.csv",
    "IOUasli": "IOUasli.csv",
    "IOUHRAGC": "IOUHRAGC.csv",
    "IOUAGC": "IOUAGC.csv",
    "IOUHR": "IOUHR.csv"
}

def cek_normalitas(data, label):
    print(f"\n===== {label} =====")
    iou = data['IoU'].dropna()

    stat, p = stats.shapiro(iou)
    print(f"Shapiro-Wilk W = {stat:.4f}, p-value = {p:.5e}")
    if p > 0.05:
        print("Data berdistribusi normal.")
    else:
        print("Data tidak berdistribusi normal.")

    plt.figure(figsize=(12, 5))
    plt.subplot(1, 2, 1)
    sns.histplot(iou, kde=True, bins=15, color='skyblue')
    plt.title(f"Histogram IoU - {label}")

    plt.subplot(1, 2, 2)
    sm.qqplot(iou, line='s', ax=plt.gca())
    plt.title(f"QQ-Plot IoU - {label}")

    plt.tight_layout()
    plt.show()

```

```
for nama, path in files.items():
    df = pd.read_csv(path)
    cek_normalitas(df, nama)
```

Lampiran 16. Kodingan uji signifikansi dengan *mann-whitney u test*

```
from scipy.stats import mannwhitneyu

df1 = pd.read_csv("IOUAGCHR.csv")
df2 = pd.read_csv("IOUHRAGC.csv")

iou1 = df1['IoU'].dropna()
iou2 = df2['IoU'].dropna()

min_len = min(len(iou1), len(iou2))
iou_1 = iou1[:min_len]
iou_2 = iou2[:min_len]

print("AGCHR-HRAGC")
stat, p = mannwhitneyu(iou_1, iou_2, alternative='two-sided')
print(f"Statistik uji: {stat}, p-value: {p}")

if p < 0.05:
    print("Ada perbedaan signifikan antara hasil deteksi data asli vs enhancement")
else:
    print("Tidak ada perbedaan signifikan antara dua metode")
```