

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Berbagai penelitian sebelumnya telah mengeksplorasi penggunaan model deep learning, khususnya arsitektur *YOLO*, untuk deteksi dan klasifikasi penyakit pada segmen mata anterior. Fokus utama penelitian-penelitian ini adalah meningkatkan akurasi deteksi, efisiensi komputasi, serta kemampuan *real-time* model pada citra mata bagian luar yang diambil menggunakan perangkat *slit-lamp* maupun kamera *smartphone*. Seiring dengan tantangan visual pada citra non-fundus, seperti adanya pantulan cahaya dan lesi yang transparan, integrasi *Attention Mechanism* dan *Explainable Artificial Intelligence (XAI)* menjadi krusial untuk memastikan model berfokus pada area patologis yang tepat sekaligus memberikan transparansi diagnostik bagi tenaga medis.

Penelitian terdahulu menunjukkan bahwa model *YOLO* mampu melakukan klasifikasi multi-kelas dan mendeteksi lesi pada permukaan okular dengan presisi tinggi. Salah satu studi penting menerapkan *YOLOv5* pada penyakit mata anterior dengan dukungan *Grad-CAM++* untuk memvisualisasikan area fitur seperti katarak dan penyakit kornea. Pendekatan tersebut membuktikan bahwa mekanisme perhatian dan visualisasi fitur dapat meningkatkan kepercayaan terhadap sistem otomatis pada citra mata luar. Namun, penerapan arsitektur generasi terbaru seperti *YOLOV12* yang memiliki keunggulan ekstraksi fitur tekstur mikro pada citra mata anterior non-fundus masih sangat terbatas. Oleh karena itu, penelitian ini hadir untuk mengisi celah tersebut dengan mengimplementasikan *YOLOV12* berbasis *Attention* dan *XAI* guna mengoptimalkan deteksi penyakit mata bagian luar secara otomatis dan terukur. Berikut adalah ringkasan penelitian terdahulu dalam Tabel 2.1 yang menjadi acuan dalam penelitian ini:

Tabel 2.1 Penelitian terdahulu

No	Tahun	Masalah	Dataset	Model	Hasil	Temuan Utama
1	2025	Urgensi diagnosis otomatis penyakit mata bagian luar	Citra Segmen Anterior	Explainability AI (XAI),	$mAP50 = 89\%$	AI sangat potensial dalam manajemen penyakit kornea, katarak, dan mata kering secara personal [7]
2	2024	Deteksi penyakit mata luar via smartphone	OSD Dataset (Smartphone Images)	Improved YOLOv5+CBAM	$mAP50 = 90.5\%$	Membuktikan citra non-fundus (luar mata) memiliki akurasi deteksi sangat tinggi dengan arsitektur YOLO [12].
3	2025	Deteksi objek kecil dengan tekstur kompleks	SMA-Dataset (Complex Texture)	SMA-YOLO (YOLOV12)	$mAP50 = 83.7\%$	YOLOV12 + Attention meningkatkan akurasi deteksi objek kecil dan mengurangi waktu komputasi [15].
4	2025	Deteksi cacat pada fitur tekstur mikro	PCB Defect (High-Res Images)	Improved YOLOV12	Accuracy= 90.4%	YOLOV12 memiliki kemampuan ekstraksi fitur tekstur halus yang sangat tinggi, cocok untuk lesi mata [16].
5	2025	Transparansi diagnosis patologi mata	Retinal & Anterior Dataset	YOLOV12 + Grad-CAM	Acc= 90.8%	Menunjukkan bahwa YOLOV12 yang digabung dengan XAI menghasilkan diagnosa yang akurat dan

No	Tahun	Masalah	Dataset	Model	Hasil	Temuan Utama
						transparan [26].
6	2025	Klasifikasi penyakit mata anterior	ASDD (<i>Slit-lamp</i> Images)	YOLOv5 + <i>Grad-CAM++</i>	Accuracy =89.9%	Analisis <i>Attention</i> memastikan model fokus pada area patologis (lesi) dan bukan pada <i>noise</i> [13].
7	2025	Kompleksitas komputasi performa <i>real-time</i> pada perangkat terbatas.	Citra Lubang	YOLOV12-ECA	<i>mAP50</i> = 78,8%	Integrasi mekanisme <i>ECA</i> meningkatkan efisiensi kanal secara cepat (7.0 ms) [14].
8	2024	Lokalisasi kekeruhan pada katarak	<i>Cataract Slit-lamp</i> Dataset	YOLO + <i>Attention</i>	F1 score >85%	<i>Attention</i> membantu memetakan posisi kekeruhan lensa mata secara lebih presisi[25].
9	2024	Sifat <i>Black-box</i> pada AI medis	Medical Image Diagnosis Dataset	YOLO + <i>XAI</i>	<i>mAP50</i> =88%	Penggunaan <i>Grad-CAM</i> dan Eigen CAM memberikan penjelasan visual pada setiap keputusan deteksi [27].
10	2026	Interpretasi lokal dan global diagnosis AI	Public Eye Disease Dataset	CNN + LIME & SHAP	Accuracy= 85%	Teknik <i>XAI</i> memberikan keyakinan klinis bagi dokter dalam memvalidasi hasil AI [28].

Berdasarkan tinjauan pustaka yang terangkum dalam Tabel 2.1 di atas, terlihat adanya evolusi signifikan dalam penggunaan model deteksi objek untuk diagnosa

penyakit mata bagian luar. Penelitian terdahulu pada citra segmen anterior non-fundus menunjukkan bahwa arsitektur *YOLO* generasi sebelumnya, seperti *YOLOv5* yang dikombinasikan dengan modul *Attention* atau *XAI* , mampu mencapai akurasi dalam rentang 89% hingga 90.4% [13]. Meskipun angka tersebut cukup impresif, tantangan utama pada citra non-fundus tetap terletak pada kompleksitas fitur tekstur mikro dan gangguan noise pencahayaan yang sering kali mengaburkan area lesi kornea yang transparan.

Integrasi model *YOLOV12* sebagai arsitektur state-of-the-art dalam literatur terbaru terbukti mampu memberikan performa yang lebih stabil dengan angka *mAP* sekitar 83.7% dan akurasi mencapai 90.4% dalam mengenali fitur tekstur halus [15][16]. Keunggulan *YOLOV12* dalam ekstraksi fitur mikro ini menjadi dasar kuat untuk menggunakannya pada kasus mata anterior yang memerlukan ketelitian tinggi. Selain itu, pemanfaatan mekanisme *Attention* seperti *Convolutional Block Attention Module (CBAM)* telah terbukti secara klinis mampu memfokuskan model pada area patologis secara lebih presisi dibandingkan model standar tanpa bantuan modul perhatian[12]. Di sisi lain, *Efficient Channel Attention (ECA)* juga digunakan pada beberapa pengembangan arsitektur *YOLO* untuk membantu model dalam memperkuat representasi fitur kanal dengan kompleksitas komputasi yang relatif ringan [14].

Celah penelitian (research gap) yang berhasil diidentifikasi adalah belum adanya studi yang secara komprehensif menggabungkan kekuatan *YOLOV12*, efisiensi *Attention Mechanism*, dan transparansi *Explainable AI (XAI)* secara simultan pada deteksi otomatis penyakit mata luar (non-fundus). Sebagian besar penelitian sebelumnya masih berfokus pada peningkatan akurasi model tanpa memberikan interpretasi yang jelas terhadap proses pengambilan keputusan sistem. Selain itu, implementasi *Attention Mechanism* pada model deteksi penyakit mata anterior juga masih relatif terbatas. Penelitian ini menawarkan pendekatan yang mengintegrasikan *YOLOV12* dengan mekanisme perhatian dan visualisasi *XAI* untuk meningkatkan kemampuan deteksi sekaligus interpretabilitas model. Melalui pendekatan tersebut, sistem tidak hanya mampu melakukan klasifikasi dan deteksi secara efektif, tetapi juga memberikan visualisasi area penting yang menjadi dasar

keputusan model. Dengan demikian, penelitian ini diharapkan dapat membantu meningkatkan kepercayaan tenaga medis terhadap penggunaan sistem AI dalam proses skrining kesehatan mata.

2.2 Teori Terkait Penelitian

Subbab ini membahas teori-teori utama yang menjadi dasar dalam pengembangan sistem deteksi penyakit mata anterior pada citra non-fundus berbasis *YOLOV12* dan *Explainable Artificial Intelligence (XAI)*. Teori yang dijabarkan difokuskan pada konsep representasi citra digital, arsitektur *YOLO* sebagai metode deep learning untuk deteksi objek, evaluasi performa model menggunakan metrik deteksi seperti *precision*, *recall*, *mAP*, dan *F1-score*, serta pendekatan *Explainable AI* untuk meningkatkan interpretabilitas hasil prediksi model. Subbab ini memberikan landasan konseptual yang mendukung implementasi sistem deteksi otomatis yang akurat, efisien, dan transparan bagi tenaga medis.

2.2.1 Representasi Citra Digital

Citra digital merupakan representasi diskrit dari suatu objek nyata yang dinyatakan dalam bentuk matriks dua dimensi yang tersusun atas elemen-elemen yang disebut piksel (pixel) [29]. Secara matematis, citra digital dapat direpresentasikan sebagai fungsi dua variabel $f(x, y)$, di mana x dan y menyatakan koordinat spasial, sedangkan $f(x, y)$ menunjukkan nilai intensitas pada titik tersebut [30]. Representasi ini menjadi dasar dalam bidang pengolahan citra digital dan computer vision karena memungkinkan informasi visual dikonversi ke dalam bentuk numerik yang dapat diproses oleh komputer. Pada citra grayscale, nilai intensitas umumnya berada pada rentang 0-255 untuk representasi 8-bit, di mana 0 merepresentasikan hitam dan 255 merepresentasikan putih. Sementara itu, pada citra berwarna (RGB), setiap piksel terdiri atas tiga kanal warna, yaitu merah (*Red*), hijau (*Green*), dan biru (*Blue*), sehingga dapat direpresentasikan sebagai rumus 2.1 berikut:

$$f(x, y) = [R(x, y), G(x, y), B(x, y)].$$

Rumus 2.1 Representasi Citra RGB

Sumber:[31]

Representasi multi-kanal ini memungkinkan sistem untuk mengekstraksi informasi warna dan pola visual yang lebih kompleks, yang sangat penting dalam analisis citra medis dan aplikasi visi komputer lanjutan. Dalam konteks deteksi penyakit mata anterior pada citra non-fundus, representasi digital ini menjadi tahap fundamental sebelum dilakukan proses *preprocessing*, deteksi, maupun ekstraksi fitur menggunakan arsitektur *YOLOV12*. Setiap piksel dan area citra diproses oleh jaringan untuk mempelajari pola spasial, bentuk lesi, tekstur, dan variasi warna yang relevan dengan kondisi patologis, kemudian diolah menjadi fitur diskriminatif yang mendukung proses klasifikasi atau sistem pendukung keputusan [32].

2.2.2 CLAHE pada Ruang Warna CIELAB

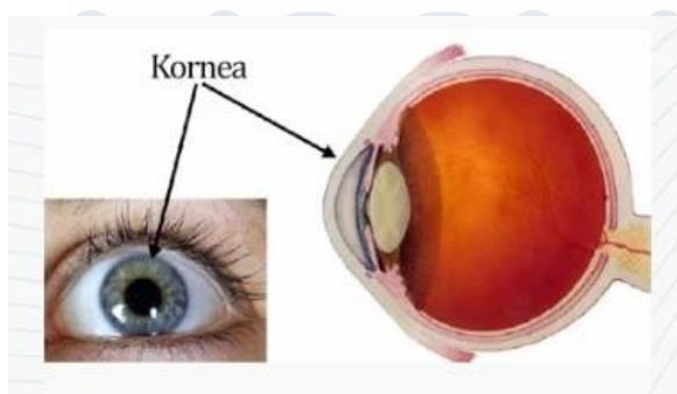
Contrast Limited Adaptive Histogram Equalization (CLAHE) merupakan metode peningkatan kualitas citra yang digunakan untuk memperbaiki kontras pada suatu gambar [33]. Metode ini dikembangkan dari Histogram Equalization dan Adaptive Histogram Equalization (AHE) dengan menambahkan batas kontras (clip limit) untuk mengurangi peningkatan noise yang berlebihan [34]. CLAHE bekerja dengan membagi citra menjadi beberapa bagian kecil, kemudian melakukan peningkatan kontras pada setiap bagian secara terpisah. Pendekatan tersebut memungkinkan detail-detail kecil pada citra menjadi lebih terlihat dibandingkan metode peningkatan kontras global. Dalam bidang citra medis, CLAHE sering digunakan karena mampu memperjelas area yang memiliki pencahayaan tidak merata tanpa merusak struktur visual penting pada citra [35]. Oleh karena itu, metode ini sesuai digunakan untuk meningkatkan kualitas citra mata anterior yang memiliki variasi pencahayaan dan detail patologis yang halus.

Pada penelitian ini, CLAHE diterapkan pada ruang warna CIELAB (Lab*) untuk mempertahankan informasi warna asli citra selama proses peningkatan kontras [36]. Ruang warna CIELAB memisahkan informasi luminansi pada channel L^* dari informasi warna pada channel a^* dan b^* . Channel L^* merepresentasikan tingkat kecerahan citra, sedangkan channel a^* dan b^* merepresentasikan komponen

warna [37]. Proses CLAHE diterapkan hanya pada channel L^* sehingga peningkatan kontras dapat dilakukan tanpa mengubah distribusi warna asli pada citra mata [38]. Setelah proses *enhancement* selesai, seluruh channel digabungkan kembali dan dikonversi ke format RGB. Pendekatan ini membantu menonjolkan fitur patologis halus seperti kemerahan, pembengkakan, dan area inflamasi yang penting dalam proses deteksi penyakit mata anterior menggunakan model YOLOV12.

2.2.3 Anatomi Mata Bagian Anterior

Bagian anterior mata merupakan struktur yang terletak pada segmen depan bola mata dan memiliki peran penting dalam proses transmisi serta pemfokusan cahaya menuju retina seperti yang ditunjukkan pada Gambar 2.1 Anatomi Mata Bagian Anterior. Struktur utama pada bagian ini meliputi kornea, konjungtiva, iris, sklera, dan lensa yang bekerja secara sinergis dalam sistem optik penglihatan. Selain berfungsi sebagai media optik, bagian ini juga berperan dalam proteksi struktur intraokular serta regulasi jumlah cahaya yang masuk ke dalam mata. Bagian anterior mata juga membantu menjaga kestabilan tekanan intraokular agar fungsi visual tetap optimal. Gangguan pada struktur ini dapat menyebabkan penurunan kualitas penglihatan hingga gangguan visual yang signifikan [39].

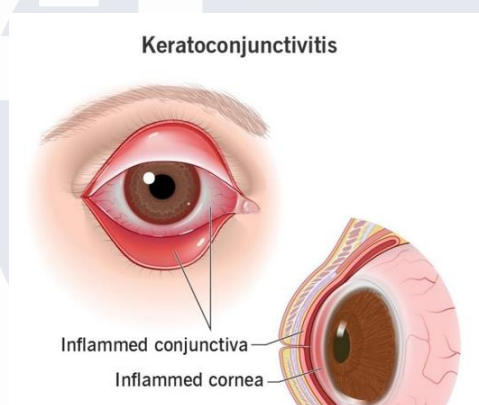


Gambar 2.1 Kornea

Sumber: [40]

Salah satu struktur utama pada bagian anterior mata adalah kornea yang terletak paling depan dan bersifat transparan berbentuk kubah seperti yang

ditunjukkan pada Gambar 2.1 Anatomi Mata Bagian Anterior. Kornea merupakan lapisan terluar mata yang berfungsi sebagai media refraksi utama. Struktur ini bersifat avaskular atau tidak memiliki pembuluh darah, sehingga memperoleh nutrisi melalui aqueous humor dan air mata. Kornea terdiri dari beberapa lapisan, yaitu epitel, stroma, dan endotel, yang berperan dalam menjaga kejernihan optik dan stabilitas refraksi cahaya. Secara fungsional, kornea menyumbang sekitar dua pertiga dari total kekuatan pembiasan cahaya pada mata. Gangguan seperti kekeruhan atau inflamasi pada kornea dapat menyebabkan penurunan ketajaman penglihatan [41].



Gambar 2.2 Konjungtiva

Sumber: [42]

Konjungtiva adalah membran mukosa tipis dan transparan yang melapisi permukaan dalam kelopak mata serta menutupi bagian anterior sklera [43]. Struktur ini berfungsi sebagai pelindung terhadap infeksi dan iritasi serta membantu menjaga kelembapan permukaan mata melalui produksi mukus dan kontribusi pada lapisan air mata. Inflamasi pada konjungtiva dikenal sebagai konjungtivitis dan sering ditandai dengan kemerahan serta pembengkakan [17]. Iris merupakan struktur berbentuk cincin berpigmen yang terletak di belakang kornea dan berfungsi mengatur jumlah cahaya yang masuk ke mata melalui perubahan ukuran pupil. Warna mata seseorang ditentukan oleh kandungan pigmen pada iris. Mekanisme kontraksi dan relaksasi otot pada iris memungkinkan penyesuaian terhadap intensitas cahaya lingkungan. Sklera adalah jaringan ikat kuat berwarna putih yang

membentuk sebagian besar dinding luar bola mata. Struktur ini memberikan perlindungan mekanis serta mempertahankan bentuk bola mata.

Sklera berlanjut ke kornea pada bagian anterior melalui daerah transisi yang disebut limbus [44]. Lensa merupakan struktur bikonveks transparan yang terletak di belakang iris dan berfungsi untuk memfokuskan cahaya ke retina melalui proses akomodasi. Perubahan bentuk lensa dikendalikan oleh otot siliaris untuk memungkinkan penglihatan pada berbagai jarak. Gangguan pada lensa, seperti katarak, menyebabkan kekeruhan yang menghambat transmisi cahaya [45]. Secara keseluruhan, struktur anatomi mata bagian anterior memiliki peran penting dalam kualitas pembentukan citra visual. Oleh karena itu, pemahaman anatomi ini menjadi dasar dalam analisis citra medis mata anterior, terutama dalam identifikasi perubahan warna, tekstur, dan morfologi yang berkaitan dengan penyakit mata.

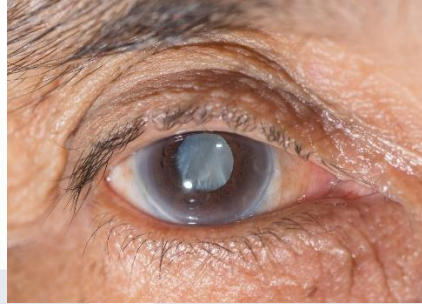
2.2.4 Patologi dan Jenis Penyakit Mata Anterior

Patologi adalah cabang ilmu kedokteran yang mempelajari perubahan abnormal pada jaringan atau organ akibat penyakit, termasuk penyebab, mekanisme, dan dampaknya terhadap fungsi tubuh. Dalam konteks mata anterior, patologi merujuk pada perubahan pada struktur bagian depan mata, seperti kornea, konjungtiva, iris, lensa, dan kelopak mata, yang dapat memengaruhi penglihatan. Pemahaman patologi ini penting untuk mendukung pengembangan sistem deteksi berbasis deep learning, karena model seperti YOLOV12 harus mengenali fitur visual khas dari setiap jenis penyakit. Berikut beberapa kategori penyakit mata anterior beserta ciri patologisnya:

2.2.4.1 Katarak (*Cataract*)

Katarak adalah kekeruhan pada lensa mata yang mengganggu transmisi cahaya menuju retina, sehingga penglihatan menjadi kabur dan berkabut [46]. Tingkat kekeruhan bisa bervariasi, mulai dari ringan hingga total, dan dapat terjadi pada satu atau kedua mata. Penyebabnya bisa terkait penuaan, trauma, atau kondisi metabolik seperti diabetes. Secara visual, katarak mengubah warna pupil dari hitam menjadi abu-abu atau putih susu, yang dapat terlihat saat pemeriksaan dengan *slit-lamp*. Deteksi dini sangat penting untuk mencegah gangguan penglihatan

permanen. Dalam konteks deteksi otomatis, model harus mampu mengenali gradasi kekeruhan ini meskipun ada gangguan cahaya pada kornea. Gambar 2.3 menunjukkan contoh karakteristik visual katarak yang ditandai dengan adanya kekeruhan pada area lensa mata.



Gambar 2.3 Ciri Khas Katarak

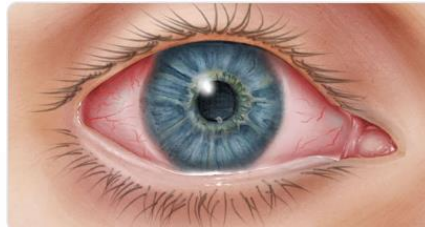
Sumber :[47]

Gambar 2.3 memperlihatkan kekeruhan lensa yang menjadi ciri khas katarak. Dalam praktik klinis, pantulan cahaya atau pencahayaan yang tidak merata sering menutupi lesi, sehingga diagnosis memerlukan ketelitian tinggi. Integrasi *YOLOV12* memungkinkan model fokus mendeteksi area lensa yang mengalami kekeruhan, sementara *Attention Mechanism* membantu menyoroti fitur tekstur halus yang kritis. Dengan tambahan *Explainable AI (XAI)*, model dapat menghasilkan *heatmap* yang menunjukkan area lensa yang dikenali sebagai patologis. Pendekatan ini meningkatkan akurasi dan transparansi diagnosis otomatis, mendukung tenaga medis dalam pengambilan keputusan.

2.2.4.2 Konjungtivitis (*Conjunctivitis*)

Konjungtivitis adalah peradangan pada selaput konjungtiva yang menyebabkan pelebaran pembuluh darah, sehingga mata tampak merah [48]. Selain itu, kondisi ini sering disertai sekret atau lendir yang dapat menempel pada kelopak mata. Penyebabnya beragam, mulai dari infeksi bakteri, virus, hingga reaksi alergi atau iritasi lingkungan. Secara visual, mata merah akibat konjungtivitis bisa sulit dibedakan dari mata merah biasa. Deteksi otomatis membutuhkan kemampuan model untuk mengekstraksi pola vaskularisasi dan warna yang khas. Identifikasi dini penting untuk mencegah komplikasi lebih lanjut, seperti penyebaran infeksi.

Gambar 2.4 menunjukkan salah satu ciri khas konjungtivitis berupa injeksi konjungtiva yang menyebabkan kemerahan pada mata.



Gambar 2.4 Injeksi Konjungtiva

Sumber: [49]

Gambar 2.4 menunjukkan injeksi konjungtiva yang khas pada konjungtivitis. Dalam praktik klinis, perbedaan antara mata merah akibat infeksi dan iritasi ringan sering sulit dikenali. *YOLOV12* membantu model mendeteksi pola pembuluh darah yang tidak biasa, sementara *Attention Mechanism* menyoroti area dengan intensitas merah yang signifikan. *XAI* memungkinkan visualisasi fitur yang dikenali sebagai patologis, sehingga dokter dapat memverifikasi hasil prediksi. Pendekatan ini meningkatkan keandalan sistem deteksi otomatis dan membantu diagnosis lebih cepat.

2.2.4.3 Uveitis

Uveitis adalah peradangan pada jaringan uvea, termasuk iris, badan siliar, dan koroid, yang dapat menyebabkan kemerahan di sekitar kornea (injeksi siliar) dan perubahan kejernihan bilik mata depan [24]. Penyakit ini sering muncul dengan gejala halus yang sulit diamati secara kasat mata. Penyebabnya bisa autoimun, infeksi, atau trauma mata. *Uveitis* berpotensi menimbulkan komplikasi serius, termasuk penurunan penglihatan permanen jika tidak ditangani. Deteksi dini dan tepat sasaran sangat penting, sehingga model deteksi otomatis perlu menyoroti area kecil dan fokus pada perubahan pigmen atau sel radang. Sistem berbasis *YOLOV12* dapat mendeteksi lesi secara cepat dengan akurasi tinggi. Gambar 2.5 menunjukkan karakteristik visual uveitis berupa kemerahan di sekitar iris dan bilik mata depan.



Gambar 2.5 *Uveitis*

Sumber: [50]

Gambar 2.5 memperlihatkan kemerahan khas akibat *Uveitis* di sekitar iris dan bilik mata depan. Perubahan visual yang halus sering menimbulkan kesulitan deteksi manual, terutama bagi tenaga medis non-spesialis. *YOLOV12* mampu fokus mendeteksi area iris dan pupil, sementara *Attention Mechanism* menyoroti lesi mikro yang mungkin terlewat oleh pengamatan manusia. *XAI* dapat memvisualisasikan area yang dianggap patologis, sehingga dokter dapat mengevaluasi prediksi model dengan lebih percaya diri. Pendekatan ini membantu meningkatkan sensitivitas deteksi dan transparansi diagnosis.

2.2.4.4 Gangguan Kelopak Mata (*Eyelid Disease*)

Gangguan kelopak mata mencakup kondisi seperti Blefaritis (peradangan tepi kelopak) dan Hordeolum (benjolan akibat infeksi) [51]. Gejalanya meliputi pembengkakan, eritema, dan kerak pada bulu mata atau margin kelopak. Penyakit ini dapat memengaruhi fungsi kelopak mata dan kenyamanan penglihatan. Diagnosis manual terkadang sulit, terutama jika gejala ringan atau tersembunyi di area perifer. Deteksi otomatis membutuhkan model yang mampu mengenali perubahan bentuk, warna, dan tekstur di kelopak mata. Identifikasi dini membantu mencegah komplikasi, termasuk infeksi yang menyebar ke jaringan sekitar. Gambar 2.6 menunjukkan contoh karakteristik visual gangguan kelopak mata berupa benjolan dan peradangan.



Gambar 2.6 Benjolan Khas Kelopak Mata

Sumber: [52]

Gambar 2.6 menunjukkan peradangan dan benjolan khas pada kelopak mata. Variasi gejala, seperti eritema dan kerak, sering menimbulkan kesulitan deteksi manual. *YOLOV12* dapat mendeteksi area kelopak yang mengalami perubahan, sementara *Attention Mechanism* menyoroti fitur kritis seperti tekstur dan pembengkakan. *XAI* memberikan visualisasi yang membantu tenaga medis mengevaluasi prediksi model. Pendekatan ini meningkatkan akurasi deteksi sekaligus transparansi, mendukung keputusan diagnostik yang lebih tepat.

2.2.5 Analisis Kelas Penyakit

Dalam penelitian ini, pengelompokan kelas penyakit mata anterior disusun berdasarkan terminologi klinis dan karakteristik visual diagnostik yang umum digunakan dalam literatur oftalmologi [53]. Setiap kelas tidak hanya merepresentasikan label citra, tetapi juga menggambarkan kondisi patologi tertentu yang memiliki ciri visual khas pada area mata anterior. Pendekatan ini membantu meningkatkan konsistensi klasifikasi penyakit pada proses pelatihan model deep learning serta mendukung interpretasi hasil deteksi agar lebih relevan dengan konteks medis. Tabel 2.2 menunjukkan pemetaan kelas penyakit beserta karakteristik visual utama dan tingkat urgensi klinis masing-masing penyakit.

Tabel 2.2 Pemetaan Kelas dan Karakteristik Diagnostik

Kelas Penyakit	Terminologi Klinis	Karakteristik Visual Utama	Tingkat Urgensi
<i>Conjunctivitis</i>	Konjungtivitis	Mata merah, berair, dan bengkak pada konjungtiva	Sedang[54]
<i>Eyelid</i>	Kel. Kelopak Mata	Pembengkakan kelopak, benjolan, atau perubahan bentuk	Rendah-Sedang[55]
Normal	Mata Normal	Sklera putih bersih, kornea jernih, dan pupil normal	Tidak ada[55]
<i>Uveitis</i>	<i>Uveitis Anterior</i>	Kemerahan di sekitar kornea dan peradangan pada mata bagian dalam	Tinggi[56]
<i>Cataract</i>	Katarak	Lensa mata tampak keruh atau putih	Sedang-Tinggi[54]

Berdasarkan Tabel 2.2, setiap penyakit mata anterior memiliki karakteristik visual dan tingkat urgensi klinis yang berbeda. Kelas *Uveitis* memiliki tingkat urgensi tertinggi karena dapat menyebabkan komplikasi serius seperti glaukoma sekunder dan penurunan penglihatan permanen apabila terlambat ditangani. Sementara itu, *Cataract* juga termasuk penyakit dengan urgensi tinggi karena menjadi salah satu penyebab utama kebutaan di dunia. Kelas *Conjunctivitis* dan *Eyelid Disease* umumnya memiliki tingkat urgensi lebih rendah, namun tetap memerlukan deteksi dini untuk mencegah komplikasi lanjutan. Perbedaan tingkat urgensi ini menjadi pertimbangan dalam analisis performa model, khususnya pada nilai *recall* untuk meminimalkan kesalahan deteksi pada penyakit berisiko tinggi.

2.2.6 Analisis Karakteristik Diagnostik Antar Kelas

Analisis kelas diagnostik dilakukan untuk memahami karakteristik visual pembeda antar penyakit mata anterior serta potensi kesulitan klasifikasi pada sistem deteksi otomatis. Setiap kelas penyakit memiliki pola visual tertentu yang dapat

dikenali melalui perubahan warna, tekstur, maupun morfologi pada mata. Namun, beberapa penyakit memiliki kemiripan visual yang cukup tinggi sehingga berpotensi menyebabkan kesalahan klasifikasi. Oleh karena itu, analisis karakteristik visual diperlukan untuk membantu model dalam mempelajari fitur yang paling relevan dari setiap kelas. Selain itu, analisis ini juga menjadi dasar dalam penggunaan *Attention Mechanism* pada model deep learning. Tabel 2.3 menunjukkan karakteristik visual utama, potensi konfusi, dan faktor pembeda antar kelas penyakit.

Tabel 2.3 Analisis Karakteristik Visual dan Tantangan Diskriminasi Diagnostik antar Kelas

Kelas	Ciri Visual Pembeda	Potensi Konfusi dengan	Faktor Pembeda Kunci
<i>Conjunctivitis</i>	Mata merah menyebar pada bagian putih mata, terkadang ada papil atau folikel pada kelopak bagian dalam	<i>Uveitis</i>	Kemerahan pada <i>Conjunctivitis</i> menyebar merata, sedangkan <i>Uveitis</i> lebih fokus di sekitar kornea [48]
<i>Eyelid</i>	Kelopak mata bengkak, merah, terdapat benjolan, atau bentuk kelopak tidak normal	Normal	Adanya benjolan, pembengkakan jelas, atau cairan pada kelopak[57]
Normal	Sklera putih, kornea jernih, iris normal, dan pupil bulat teratur	Semua kelas tahap awal	Tidak ditemukan tanda-tanda penyakit atau kelainan[58]
<i>Uveitis</i>	Kemerahan di sekitar kornea, dapat muncul bercak putih kecil, hipopion, atau perlengketan iris	<i>Conjunctivitis</i>	Kemerahan lebih terpusat di area sekitar kornea, bukan menyebar di seluruh mata[59]
<i>Cataract</i>	Lensa mata tampak keruh putih atau abu-abu, refleks merah pupil berkurang atau hilang	Normal	Adanya kekeruhan pada lensa atau perubahan tampilan pupil[60]

Berdasarkan Tabel 2.3, tantangan klasifikasi terbesar terdapat pada kelas *Conjunctivitis* dan *Uveitis* karena keduanya sama-sama menunjukkan gejala

kemerahan pada mata. Perbedaannya terletak pada distribusi hiperemia, di mana *Conjunctivitis* cenderung bersifat menyebar secara luas sedangkan *Uveitis* lebih dominan terpusat di sekitar kornea. Selain itu, kelas *Cataract* juga memiliki variasi visual yang tinggi tergantung jenis dan stadium penyakit, sehingga fitur antar citra menjadi kurang seragam. Kondisi tersebut menyebabkan model membutuhkan kemampuan ekstraksi fitur yang lebih detail agar dapat membedakan setiap kelas secara akurat. Oleh karena itu, penelitian ini menggunakan *YOLOV12* yang dikombinasikan dengan *CBAM* untuk membantu model lebih fokus pada area patologis penting. Selanjutnya, *Grad-CAM* digunakan untuk memvisualisasikan area citra yang paling berpengaruh terhadap hasil prediksi model sehingga interpretasi deteksi menjadi lebih transparan.

2.2.7 Evaluasi Model Deteksi

Evaluasi performa model object detection bertujuan untuk mengukur sejauh mana sistem mampu mendeteksi objek dengan tepat baik dari sisi klasifikasi maupun lokasi *bounding box*. Dalam penelitian deteksi objek, metrik evaluasi yang umum digunakan meliputi *Precision*, *Recall*, *F1-score*, dan *Mean Average Precision (mAP)*. Metrik-metrik ini dihitung berdasarkan konsep True Positive (TP), False Positive (FP), dan False Negative (FN), yang ditentukan dengan membandingkan hasil prediksi model terhadap ground truth menggunakan ambang batas Intersection over Union (IoU) [26].

Precision mengukur proporsi prediksi positif yang benar terhadap seluruh prediksi positif yang dihasilkan model. Secara matematis dirumuskan sebagai:

$$Precision = \frac{TP}{TP+FP}$$

Rumus 2.2 *Precision*

Sumber:[61]

Nilai *precision* yang tinggi menunjukkan bahwa model jarang menghasilkan deteksi palsu (false detection). Dalam konteks medis, *precision*

penting untuk meminimalkan kesalahan diagnosis akibat deteksi objek yang sebenarnya tidak ada [29]. *Recall* mengukur kemampuan model dalam mendeteksi seluruh objek yang seharusnya terdeteksi. Rumusnya adalah:

$$Recall = \frac{TP}{TP + FN}$$

Rumus 2.3 Recall

Sumber : [62]

Nilai *recall* yang tinggi menunjukkan bahwa model mampu menemukan sebagian besar objek aktual dalam citra. Dalam aplikasi medis, *recall* yang tinggi sangat penting untuk mengurangi risiko objek penyakit yang terlewat (missed detection) [26]. *F1-score* merupakan rata-rata harmonik antara *precision* dan *recall*, yang digunakan untuk menyeimbangkan keduanya:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Rumus 2.4 F1 Score

Sumber : [8]

F1-score memberikan gambaran komprehensif terhadap performa model, terutama ketika terdapat ketidakseimbangan jumlah kelas (imbalanced dataset) [26]. Selain menggunakan metrik *precision*, *recall*, dan *F1-score*, performa model object detection juga dipengaruhi oleh nilai confidence *Threshold* [63]. *Confidence Threshold* merupakan ambang batas yang digunakan untuk menentukan apakah suatu hasil prediksi diterima atau diabaikan berdasarkan tingkat keyakinan model [64]. Perubahan nilai *Confidence Threshold* akan memengaruhi keseimbangan antara *precision* dan *recall* [65]. *Threshold* yang lebih rendah cenderung meningkatkan *recall* karena lebih banyak prediksi diterima, namun dapat menurunkan *precision* akibat meningkatnya *false positive* [66]. Sebaliknya, *Threshold* yang lebih tinggi meningkatkan *precision* karena hanya prediksi dengan

tingkat keyakinan tinggi yang diterima, tetapi dapat menurunkan *recall* karena sebagian objek yang sebenarnya benar tidak lagi terdeteksi [67]. Oleh karena itu, pemilihan *Confidence Threshold* perlu disesuaikan dengan tujuan aplikasi [68]. Salah satu pendekatan yang umum digunakan adalah memilih *Threshold* yang menghasilkan nilai *F1-score* tertinggi sebagai titik keseimbangan antara *precision* dan *recall* [69]. *Mean Average Precision (mAP)* merupakan metrik utama dalam evaluasi object detection. Nilai ini dihitung dengan terlebih dahulu menentukan *Average Precision (AP)* untuk setiap kelas, yang diperoleh dari luas area di bawah kurva *Precision–Recall*:

$$AP = \int_0^1 Precision(Recall) d(Recall)$$

Rumus 2.5 *Average Precision*

Sumber : [50]

Selanjutnya, *mAP* dihitung sebagai rata-rata dari seluruh nilai *AP* pada semua kelas:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

Rumus 2.6 *MAP*

Sumber : [70]

di mana *N* adalah jumlah kelas. Dalam praktiknya, *mAP* sering dihitung pada ambang *IoU* tertentu, seperti *mAP@0.5* atau *mAP@[0.5:0.95]*, untuk mengevaluasi ketepatan lokasi *bounding box* secara lebih komprehensif [26]. Secara keseluruhan, kombinasi metrik *precision*, *recall*, *F1-score*, dan *mAP* memberikan evaluasi menyeluruh terhadap kemampuan model dalam mendeteksi dan mengklasifikasikan objek, baik dari aspek akurasi klasifikasi maupun presisi lokasi spasial. Dalam konteks deteksi penyakit mata anterior, penggunaan metrik ini penting untuk memastikan sistem mampu mendeteksi area lesi secara akurat dan konsisten.

2.3 Framework/Algoritma yang digunakan

Penelitian ini menggunakan kerangka kerja *Cross-Industry Standard Process for Data Mining (CRISP-DM)* sebagai panduan dalam seluruh tahapan pengolahan data, pembangunan model, dan evaluasi performa sistem deteksi penyakit mata anterior. Penerapan *CRISP-DM* memastikan proses penelitian berlangsung secara sistematis, terstruktur, dan dapat direproduksi, sekaligus mendukung integrasi model *YOLOV12* dengan pendekatan *Explainable Artificial Intelligence (XAI)* untuk menghasilkan output yang akurat sekaligus interpretatif.

2.3.1 *Cross-Industry Standard Process for Data Mining (CRISP-DM)*

CRISP-DM merupakan kerangka kerja standar yang banyak digunakan dalam analisis data dan pengembangan model berbasis data, termasuk pada domain medis [71]. Framework ini bersifat fleksibel dan iteratif, sehingga dapat diadaptasikan pada berbagai jenis data dan tujuan penelitian. Dalam penelitian ini, *CRISP-DM* berperan sebagai landasan metodologis untuk membangun sistem deteksi penyakit mata anterior berbasis citra non-fundus, yang tidak hanya fokus pada akurasi, tetapi juga interpretabilitas prediksi melalui *XAI*. Kerangka *CRISP-DM* terdiri dari enam tahapan utama, yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment*, yang saling terkait dan membentuk siklus iteratif [72]. Setiap tahap dapat disesuaikan atau diperbaiki berdasarkan hasil tahapan sebelumnya untuk mencapai performa sistem yang optimal.



Gambar 2.7 Framework *CRISP-DM*

Sumber: [73]

Merujuk pada Gambar 2.7, framework *CRISP-DM* dalam penelitian ini terdiri atas enam tahapan yang saling terhubung dan bersifat iteratif, sehingga setiap langkah dapat ditinjau kembali atau disesuaikan bila diperlukan. Siklus ini memastikan bahwa proses pengolahan data, pembangunan model, dan evaluasi sistem deteksi penyakit mata anterior berjalan secara sistematis dan selaras dengan tujuan penelitian. Penjelasan masing-masing tahapan dalam kerangka *CRISP-DM* untuk penelitian ini adalah sebagai berikut:

1. *Business Understanding*

Tahap ini fokus pada pemahaman tujuan penelitian dan kebutuhan pengguna akhir, yaitu tenaga medis [74]. Dalam konteks penelitian ini, peneliti mengidentifikasi kebutuhan akan sistem yang dapat mendeteksi penyakit mata anterior secara otomatis, cepat, dan akurat, sekaligus memberikan interpretasi visual dari prediksi model agar dapat dipahami oleh tenaga medis. Indikator keberhasilan meliputi tingkat akurasi deteksi, *Mean Average Precision (mAP)*, *precision*, *recall*, dan *F1-score*, serta kualitas visualisasi *XAI*.

2. *Data Understanding*

Tahap ini mencakup eksplorasi awal terhadap dataset citra non-fundus mata anterior. Analisis dilakukan untuk memahami karakteristik data, variasi kondisi mata, ukuran dataset, distribusi kelas penyakit, serta mendeteksi potensi kendala seperti citra buram, pencahayaan tidak merata, atau anotasi yang tidak konsisten. Pemahaman ini penting agar model yang dibangun dapat belajar dari data yang representatif.

3. *Data Preparation*

Pada tahap ini, citra diproses agar siap digunakan oleh model *YOLOV12*. Aktivitas utama meliputi *preprocessing* (normalisasi, resize, dan augmentasi citra), anotasi *bounding box* untuk area lesi, dan pembagian dataset menjadi subset pelatihan, validasi, dan pengujian. Tahap ini memastikan data berada dalam format yang optimal untuk pelatihan model deteksi berbasis deep learning.

4. *Modeling*

Tahap ini mencakup pembangunan dan pelatihan model *YOLOV12* untuk mendeteksi penyakit mata anterior. Model dilatih untuk mengenali beberapa kategori penyakit sekaligus memprediksi lokasi area abnormal menggunakan *bounding box*. Selain itu, integrasi *Explainable AI* diterapkan menggunakan teknik visualisasi seperti *Grad-CAM*, sehingga setiap prediksi disertai peta perhatian (*Attention mAP*) yang menampilkan area citra yang menjadi fokus model.

5. *Evaluation*

Setelah model dilatih, tahap evaluasi dilakukan untuk menilai performa sistem. Metrik yang digunakan meliputi *Mean Average Precision (mAP)*, *precision*, *recall*, dan *F1-score*. Evaluasi juga mencakup analisis interpretabilitas visual dari *XAI* untuk memastikan bahwa hasil deteksi tidak hanya akurat, tetapi juga dapat dipahami secara klinis oleh tenaga medis.

6. *Deployment*

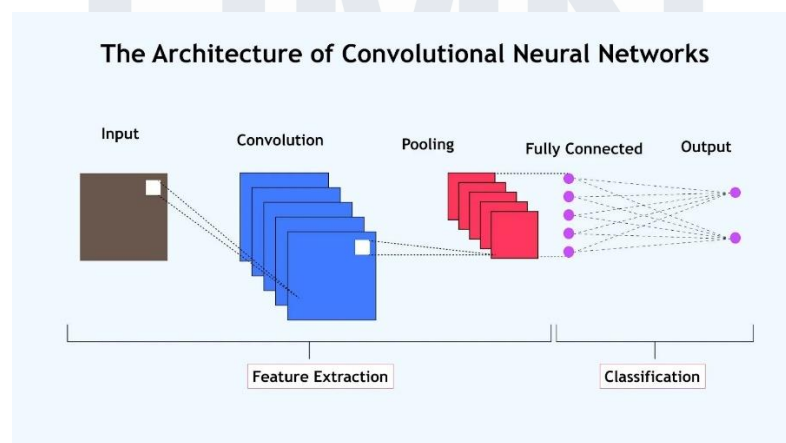
Tahap terakhir adalah penerapan model dalam sistem yang dapat digunakan secara nyata, misalnya sebagai *decision support system* bagi tenaga medis [75]. Hasil deteksi dapat ditampilkan dengan overlay *bounding box* pada

citra mata anterior non-fundus, disertai peta perhatian *XAI* yang membantu memahami prediksi. Sistem ini dirancang agar dapat dijalankan pada perangkat dengan sumber daya terbatas, termasuk laptop atau perangkat mobile dengan GPU ringan.

Dengan penerapan CRISP-DM, setiap tahapan penelitian saling terkait dan berperan penting dalam membangun sistem deteksi penyakit mata anterior yang efisien, akurat, dan interpretatif, sehingga hasil penelitian tidak hanya berupa model AI, tetapi juga solusi klinis yang dapat digunakan secara praktis.

2.3.2 *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) merupakan salah satu arsitektur deep learning yang dirancang khusus untuk memproses data berbentuk grid, seperti citra dua dimensi [48]. CNN bekerja dengan memanfaatkan operasi konvolusi untuk mengekstraksi fitur spasial secara otomatis dari citra input. Arsitektur ini terinspirasi dari mekanisme kerja sistem visual biologis dan telah terbukti efektif dalam berbagai tugas pengenalan pola dan klasifikasi citra [11]. Dalam implementasinya, CNN terdiri atas beberapa lapisan utama yang saling terhubung untuk membentuk proses pembelajaran yang sistematis. Setiap lapisan memiliki fungsi spesifik, mulai dari ekstraksi fitur hingga proses klasifikasi akhir. Struktur umum dari arsitektur CNN tersebut dapat dilihat pada Gambar 2.8.



Gambar 2.8 Arsitektur CNN

Berdasarkan Gambar 2.8, arsitektur CNN diawali dengan tahap input berupa citra yang kemudian diproses melalui convolutional layer untuk mengekstraksi fitur penting. Hasil dari proses konvolusi akan melewati fungsi aktivasi seperti ReLU guna memperkenalkan non-linearitas dalam model. Selanjutnya, pooling layer digunakan untuk mereduksi dimensi fitur tanpa menghilangkan informasi utama, sehingga membantu mengurangi kompleksitas komputasi. Tahapan convolution dan pooling ini dikenal sebagai proses feature extraction, seperti yang ditunjukkan pada bagian tengah gambar. Setelah fitur diperoleh, data akan diteruskan ke fully connected layer yang berperan dalam proses classification untuk menghasilkan prediksi akhir. Dengan demikian, CNN mampu melakukan ekstraksi fitur dan klasifikasi secara otomatis dalam satu arsitektur terpadu.

Operasi konvolusi pada CNN dapat dirumuskan sebagai berikut:

$$S(i, j) = (X * K)(i, j) = \sum_m \sum_n X(i - m, j - n) K(m, n).$$

Rumus 2.7 Konvolusi CNN

Sumber: [76]

Dimana X adalah citra input, K adalah kernel atau filter konvolusi, dan $S(i, j)$ merupakan nilai pada feature map hasil konvolusi. Proses ini memungkinkan model mendeteksi pola lokal seperti tepi (edges), tekstur, maupun bentuk tertentu pada citra. Setiap kernel belajar mengenali fitur yang berbeda selama proses pelatihan melalui mekanisme backpropagation. Secara umum, CNN terdiri atas beberapa lapisan utama, yaitu convolutional layer, activation function (biasanya Rectified Linear Unit/ReLU), pooling layer, dan fully connected layer [77].

Convolutional layer berfungsi mengekstraksi fitur, sedangkan fungsi aktivasi ReLU memperkenalkan non-linearitas agar jaringan mampu memodelkan hubungan kompleks. Pooling layer digunakan untuk melakukan reduksi dimensi fitur sekaligus mempertahankan informasi penting, sehingga mengurangi kompleksitas komputasi dan risiko *overfitting*. Pada tahap akhir, *fully connected layer* melakukan proses klasifikasi berdasarkan fitur yang telah dipelajari. Keunggulan utama CNN adalah kemampuannya melakukan automatic feature

extraction, sehingga menggantikan proses *feature engineering* manual yang sebelumnya banyak digunakan pada metode *machine learning* tradisional [78]. Pendekatan ini memungkinkan sistem secara adaptif mempelajari representasi fitur yang optimal langsung dari data citra, termasuk pada aplikasi citra medis seperti deteksi dan klasifikasi penyakit.

2.3.3 Object Detection dan Arsitektur YOLO

Object detection merupakan tugas dalam bidang computer vision yang bertujuan untuk mengidentifikasi keberadaan suatu objek dalam citra sekaligus menentukan lokasinya melalui *bounding box* [79]. Berbeda dengan klasifikasi citra yang hanya menentukan kelas objek, *object detection* memberikan informasi spasial berupa koordinat lokasi objek.

Secara umum, *bounding box* direpresentasikan dalam format:

$$B = (x, y, w, h)$$

Rumus 2.8 Representasi *Bounding Box*

Sumber: [80]

di mana x dan y menunjukkan koordinat pusat kotak, sedangkan w dan h menyatakan lebar dan tinggi kotak prediksi. Untuk mengevaluasi kualitas prediksi *bounding box*, digunakan metrik *Intersection over Union (IoU)*, yang dirumuskan sebagai:

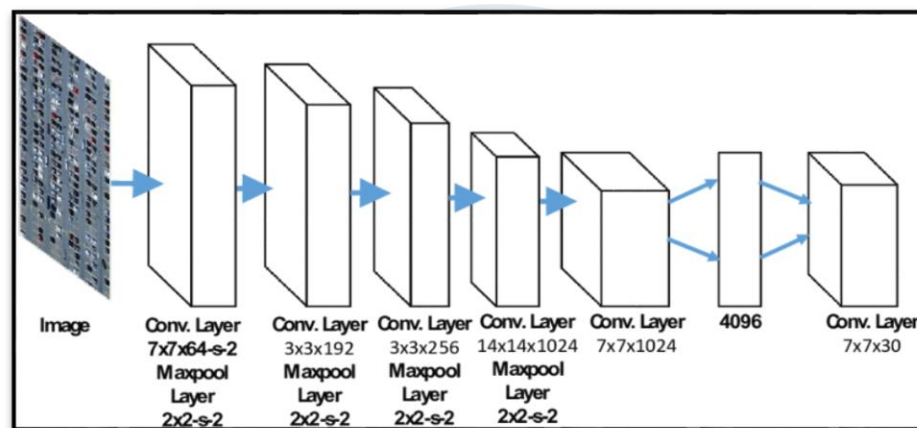
$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Rumus 2.9 Perhitungan *Intersection over Union (IoU)*

Sumber: [81]

Nilai *IoU* mengukur rasio antara area tumpang tindih (overlap) dari *bounding box* prediksi dengan ground truth terhadap total area gabungan keduanya. Semakin tinggi nilai *IoU* (mendekati 1), semakin baik kesesuaian antara prediksi dan anotasi sebenarnya. Dalam praktiknya, ambang batas *IoU* tertentu (misalnya 0.5) digunakan untuk menentukan apakah suatu deteksi dianggap benar (true

positive) [26]. Salah satu arsitektur *object detection* yang populer adalah *You Only Look Once (YOLO)*, yang menggunakan pendekatan *single-stage detector* untuk melakukan deteksi objek secara cepat dan efisien. *YOLO* membagi citra menjadi beberapa grid, di mana setiap grid bertanggung jawab untuk memprediksi *bounding box* dan probabilitas kelas objek. Ilustrasi arsitektur dasar *YOLO* dapat dilihat pada Gambar 2.9.



Gambar 2.9 Arsitektur *YOLO*

Sumber :[82]

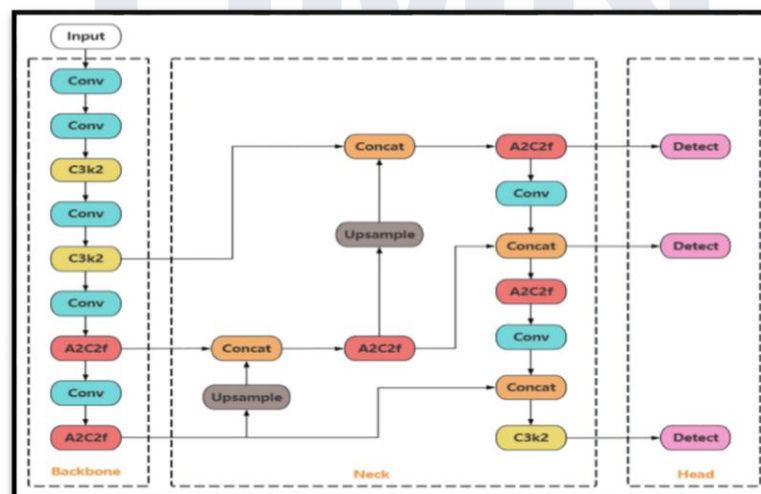
Berdasarkan Gambar 2.9, arsitektur *YOLO* diawali dengan citra input yang diproses melalui beberapa lapisan konvolusi (*convolutional layer*) untuk mengekstraksi fitur. Setiap lapisan konvolusi diikuti dengan operasi *max pooling* yang berfungsi untuk mengurangi dimensi fitur sekaligus mempertahankan informasi penting. Seiring bertambahnya kedalaman jaringan, fitur yang dihasilkan menjadi semakin kompleks dan representatif terhadap objek dalam citra. Pada tahap akhir, fitur tersebut diteruskan ke *fully connected layer* yang menghasilkan prediksi berupa koordinat *bounding box* dan probabilitas kelas. Output akhir berupa representasi lokasi dan kategori objek dalam citra secara simultan dalam satu kali proses inferensi.

Arsitektur pada Gambar 2.9. menunjukkan bahwa *YOLO* melakukan proses deteksi objek secara *end-to-end* dalam satu jaringan terpadu. Meskipun struktur ini masih tergolong sederhana dibandingkan dengan versi terbaru, pendekatan ini menjadi dasar penting dalam pengembangan *YOLO* selanjutnya. *YOLOv1*

memperkenalkan konsep *single-stage object detection*, kemudian *YOLOv2* dan *YOLOv3* meningkatkan akurasi melalui penggunaan *anchor box* dan *multi-scale prediction* [83]. Selanjutnya, *YOLOv4* hingga *YOLOv7* berfokus pada peningkatan efisiensi, kecepatan, dan optimasi ekstraksi fitur [84], sedangkan *YOLOv8* hingga *YOLOV12* membawa pengembangan pada stabilitas pelatihan, efisiensi inferensi, dan sensitivitas terhadap objek kecil serta kompleks [18]. Perkembangan ini membuat *YOLO* semakin adaptif terhadap berbagai jenis dataset kompleks, termasuk citra medis, sehingga sangat sesuai untuk diterapkan pada deteksi penyakit mata anterior menggunakan citra non-fundus dalam penelitian ini.

2.3.4 You Only Look Once Versi 12 (YOLOV12)

Arsitektur *YOLOV12* merupakan pengembangan dari metode You Only Look Once (*YOLO*) yang mengadopsi pendekatan *single-stage object detection*, di mana proses deteksi objek dilakukan dalam satu kali tahap inferensi [85]. Pendekatan ini memungkinkan model untuk secara simultan memprediksi lokasi dan kelas objek secara efisien. *YOLOV12* dirancang untuk meningkatkan akurasi deteksi tanpa mengorbankan kecepatan komputasi, sehingga cocok untuk aplikasi *real-time* [86]. Secara umum, arsitektur *YOLOV12* terdiri atas tiga komponen utama, yaitu backbone, neck, dan head. Ketiga komponen tersebut bekerja secara terintegrasi dalam satu alur pemrosesan. Struktur arsitektur *YOLOV12* yang digunakan dalam penelitian ini dapat dilihat pada Gambar 2.10.



Gambar 2.10 Arsitektur YOLOV12

Sumber : [87]

Berdasarkan Gambar 2.10, bagian backbone berfungsi sebagai ekstraktor fitur utama dari citra input. Proses ini dilakukan melalui serangkaian operasi konvolusi (Conv) yang bertujuan untuk menangkap pola dasar seperti tepi dan tekstur. Selain itu, terdapat modul seperti C3k2 dan A2C2f yang berperan dalam memperdalam representasi fitur tanpa meningkatkan kompleksitas secara signifikan [88]. Semakin dalam lapisan yang dilalui, semakin kompleks fitur yang dihasilkan oleh jaringan. Backbone pada YOLOV12 dirancang untuk meningkatkan efisiensi komputasi sekaligus mempertahankan kualitas fitur yang dihasilkan. Dengan demikian, tahap ini menjadi fondasi penting dalam keseluruhan proses deteksi objek.

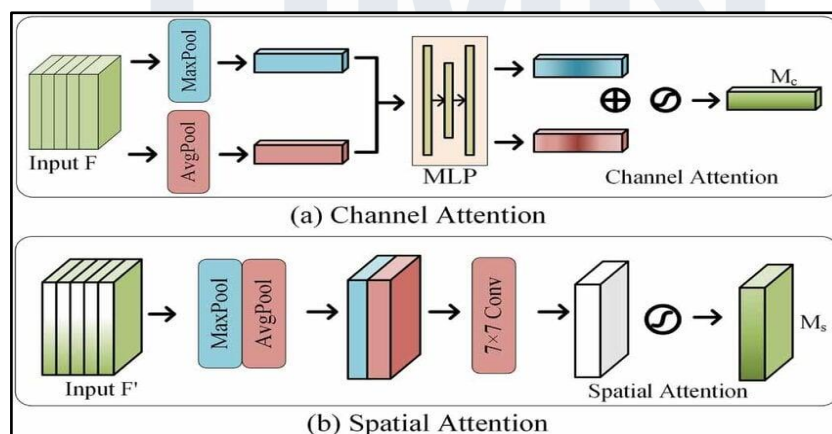
Selanjutnya, fitur yang dihasilkan oleh backbone akan diteruskan ke bagian neck untuk proses penggabungan fitur multi-skala. Pada tahap ini, digunakan operasi seperti upsampling dan concatenation (Concat) untuk mengintegrasikan fitur dari berbagai level resolusi. Pendekatan ini memungkinkan model untuk mempertahankan informasi spasial sekaligus memahami konteks global dari citra. Struktur neck pada YOLOV12 mengadopsi konsep Feature Pyramid Network (FPN) dan Path Aggregation Network (PAN) yang efektif dalam meningkatkan kemampuan deteksi objek kecil maupun besar. Proses penggabungan fitur ini sangat penting dalam meningkatkan ketelitian prediksi. Dengan adanya neck, informasi yang dihasilkan menjadi lebih kaya dan representatif.

Tahap akhir dalam arsitektur YOLOV12 adalah bagian head, yang berfungsi untuk menghasilkan prediksi objek. Pada bagian ini, fitur yang telah diproses akan digunakan untuk memprediksi *bounding box*, *confidence score*, serta klasifikasi objek melalui modul Detect. YOLOV12 biasanya menghasilkan beberapa output pada skala yang berbeda untuk meningkatkan akurasi deteksi. Pendekatan multi-skala ini memungkinkan model mendeteksi objek dengan ukuran yang bervariasi secara lebih optimal. Selain itu, proses ini dilakukan secara langsung tanpa tahap tambahan, sehingga tetap mempertahankan kecepatan inferensi. Dengan demikian, keseluruhan arsitektur YOLOV12 mampu memberikan keseimbangan antara

mengeksktraksi fitur dasar citra, kemudian fitur diperkaya menggunakan modul A2C2f dan diperkuat oleh *CBAM* . Selanjutnya, neck menggabungkan fitur dari berbagai skala melalui mekanisme FPN dan PAN sebelum diteruskan ke *detection head*. Pada tahap akhir, *detection head* menghasilkan prediksi berupa lokasi *bounding box*, kelas penyakit, dan *confidence score* pada tiga skala deteksi sehingga model mampu mendeteksi objek dengan ukuran yang bervariasi secara lebih optimal.

2.3.5 Convolutional Block Attention Module (CBAM)

Convolutional Block Attention Module (CBAM) merupakan salah satu mekanisme *Attention* yang dirancang untuk meningkatkan kemampuan model dalam mengekstraksi fitur yang relevan secara selektif [89]. Modul ini bekerja dengan memberikan bobot perhatian pada feature map sehingga model dapat lebih fokus pada informasi penting dan mengabaikan informasi yang kurang relevan [90]. Pendekatan ini terbukti efektif dalam meningkatkan performa model pada berbagai tugas visi komputer, termasuk deteksi objek. Dalam konteks citra medis, keberadaan noise, variasi pencahayaan, serta batas lesi yang tidak tegas menjadi tantangan utama dalam proses ekstraksi fitur. Ilustrasi alur kerja *CBAM* yang terdiri dari dua tahap utama dapat dilihat pada Gambar 2.12. Dengan adanya mekanisme ini, model mampu menyesuaikan fokus secara dinamis terhadap karakteristik citra yang kompleks.



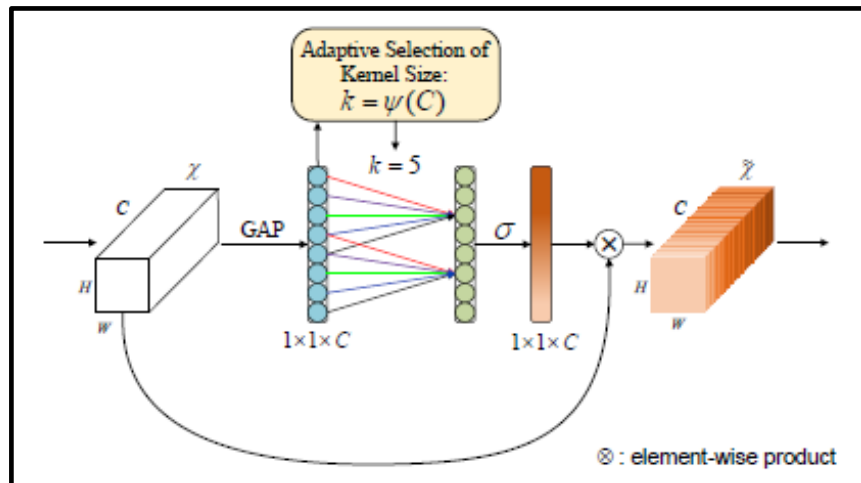
Gambar 2.12 *CBAM Attention*

Sumber : [91]

CBAM terdiri dari dua komponen utama, yaitu *Channel Attention* dan *Spatial Attention*, yang bekerja secara berurutan untuk meningkatkan kualitas representasi fitur [92]. Pada *Channel Attention* seperti ditunjukkan pada Gambar 2.12(a), feature map diproses menggunakan average pooling dan max pooling untuk menghasilkan representasi global, yang kemudian diteruskan ke Multi-Layer Perceptron (MLP). Output dari kedua jalur tersebut digabungkan melalui operasi penjumlahan dan diaktivasi menggunakan fungsi sigmoid untuk menghasilkan bobot perhatian kanal. Selanjutnya, pada tahap *Spatial Attention* yang ditunjukkan pada Gambar 2.12(b), feature map hasil sebelumnya diproses kembali menggunakan average pooling dan max pooling pada dimensi kanal. Hasil penggabungan kedua fitur tersebut kemudian dilewatkan melalui konvolusi berukuran 7×7 untuk menghasilkan peta perhatian spasial. Kombinasi kedua mekanisme ini memungkinkan model memahami informasi apa yang penting dan di mana lokasi fitur tersebut secara lebih akurat.

2.3.6 *Efficient Channel Attention (ECA)*

Efficient Channel Attention (ECA) merupakan salah satu mekanisme *Attention* yang dirancang untuk meningkatkan representasi fitur dengan fokus pada hubungan antar kanal secara efisien [93]. Berbeda dengan *CBAM* yang mengombinasikan *channel* dan *Spatial Attention*, *ECA* hanya berfokus pada *Channel Attention* tanpa menambahkan kompleksitas yang signifikan. Pendekatan ini bertujuan untuk mempertahankan performa model dengan jumlah parameter yang minimal. Dalam konteks citra medis, efisiensi komputasi menjadi aspek penting, terutama untuk implementasi pada perangkat dengan sumber daya terbatas seperti *mobile device*. Ilustrasi alur kerja *ECA* yang menekankan interaksi antar kanal dapat dilihat pada Gambar 2.13. Dengan mekanisme ini, model tetap mampu menyoroti fitur penting tanpa meningkatkan beban komputasi secara signifikan.



Gambar 2.13 ECA Attention

Sumber: [94]

ECA bekerja dengan memanfaatkan informasi global dari feature map melalui operasi *Global Average Pooling* (GAP), seperti ditunjukkan pada Gambar 2.13. Hasil dari GAP menghasilkan representasi kanal berukuran $1 \times 1 \times C$ yang kemudian diproses menggunakan konvolusi satu dimensi (*1D convolution*) untuk menangkap hubungan lokal antar kanal. Berbeda dengan CBAM yang menggunakan *Multi-Layer Perceptron* (MLP), ECA menghindari reduksi dimensi dan langsung memodelkan interaksi kanal secara efisien. Ukuran kernel pada konvolusi ini ditentukan secara adaptif berdasarkan jumlah kanal ($k = \psi(C)$), sehingga mampu menyeimbangkan antara kompleksitas dan performa [95]. Output dari proses ini kemudian dilewatkan ke fungsi aktivasi sigmoid untuk menghasilkan bobot perhatian kanal. Bobot tersebut selanjutnya dikalikan secara *element-wise* dengan feature map awal untuk menghasilkan representasi fitur yang telah diperkuat.

2.4 Tools/software yang digunakan

Dalam penelitian ini, beberapa perangkat bantu (tools) digunakan untuk mendukung proses pengolahan data, pembangunan model, evaluasi, hingga implementasi sistem deteksi penyakit mata anterior. Tools yang digunakan mencakup Google Colab, Python beserta pustaka pendukungnya, serta teknologi *Deployment* seperti *FastAPI* dan Flutter. Pemilihan tools ini didasarkan pada

kebutuhan penelitian yang memerlukan komputasi tinggi, fleksibilitas pengembangan, serta kemudahan integrasi sistem. Selain itu, tools yang digunakan mampu mendukung pengembangan model deep learning secara efisien dan terstruktur. Setiap tools memiliki peran spesifik yang saling terintegrasi dalam membangun pipeline penelitian secara end-to-end. Dengan demikian, keseluruhan tools yang digunakan mampu meningkatkan efisiensi, akurasi, serta konsistensi penelitian.

2.4.1 Google Colab

Google Colab merupakan lingkungan pengembangan berbasis cloud yang memungkinkan eksekusi kode Python secara interaktif melalui browser tanpa memerlukan instalasi local [96]. Dalam penelitian ini, Google Colab digunakan karena menyediakan akses ke GPU dan TPU yang sangat membantu dalam mempercepat proses pelatihan model *YOLOV12*. Selain itu, platform ini mendukung kolaborasi secara *real-time* sehingga memudahkan proses berbagi eksperimen dan hasil penelitian. Fleksibilitasnya dalam mendukung berbagai pustaka Python menjadikan proses *preprocessing*, augmentasi, hingga evaluasi dapat dilakukan dalam satu lingkungan terintegrasi. Dengan demikian, Google Colab berperan penting dalam meningkatkan efisiensi dan kecepatan eksperimen penelitian. Manfaat penggunaan Google Colab dalam penelitian ini meliputi:

1. Akses GPU/TPU: untuk mempercepat pelatihan model *YOLOV12*.
2. Kolaborasi : *real-time* melalui notebook yang dapat dibagikan.
3. Visualisasi interaktif : untuk menampilkan hasil deteksi dan *XAI*.
4. Fleksibilitas pustaka : untuk mendukung seluruh pipeline penelitian.

Penggunaan Google Colab memungkinkan seluruh proses eksperimen dilakukan dalam satu platform tanpa memerlukan konfigurasi perangkat keras secara lokal. Hal ini sangat membantu dalam menghemat waktu dan biaya penelitian, terutama pada tahap pelatihan model yang membutuhkan komputasi tinggi. Selain itu, kemudahan akses melalui browser membuat penelitian dapat dilakukan secara fleksibel dari berbagai perangkat. Integrasi dengan berbagai pustaka Python juga mempercepat proses pengembangan model secara end-to-end.

Dengan dukungan fitur yang lengkap, Google Colab menjadi solusi yang efektif untuk penelitian berbasis deep learning. Oleh karena itu, penggunaan Google Colab sangat relevan dalam mendukung penelitian ini.

2.4.2 Python

Python merupakan bahasa pemrograman utama yang digunakan dalam penelitian ini karena memiliki sintaks yang sederhana dan mudah dipahami [97]. Bahasa ini juga memiliki ekosistem pustaka yang sangat lengkap untuk mendukung pengolahan data, analisis citra, serta pengembangan model deep learning. Dalam penelitian ini, Python digunakan untuk mengimplementasikan seluruh pipeline mulai dari *preprocessing* citra hingga evaluasi model. Berbagai pustaka yang tersedia memungkinkan proses pengembangan dilakukan secara efisien dan terintegrasi. Selain itu, Python juga mendukung berbagai framework deep learning yang umum digunakan dalam penelitian AI. Dengan dukungan tersebut, Python menjadi pilihan utama dalam pengembangan sistem deteksi penyakit mata anterior.

1. *OpenCV*: Pengolahan citra, augmentasi, dan normalisasi citra mata anterior.
2. *PyTorch / TensorFlow*: Implementasi dan pelatihan model *YOLOV12*.
3. *NumPy dan Pandas*: Mendukung berbagai proses pengolahan data
4. *Matplotlib dan Seaborn*: Visualisasi hasil deteksi dan *heatmap XAI*.
5. *Scikit-learn*: Perhitungan metrik evaluasi seperti *precision*, *recall*, *Mean Average Precision (mAP)*, dan *F1-score*.

Penggunaan Python memungkinkan integrasi berbagai tahapan penelitian dalam satu bahasa pemrograman yang konsisten. Hal ini mempermudah proses debugging, pengembangan, serta pemeliharaan kode program. Selain itu, dukungan komunitas yang besar membuat Python memiliki banyak referensi dan dokumentasi yang dapat digunakan. Integrasi antar pustaka juga memungkinkan pengolahan data dan pelatihan model dilakukan secara lebih efisien. Dengan fleksibilitas yang tinggi, Python mampu mengakomodasi kebutuhan penelitian yang kompleks. Oleh karena itu, Python menjadi fondasi utama dalam pengembangan sistem deteksi penyakit mata anterior.

2.4.3 FastAPI

FastAPI merupakan framework berbasis Python yang digunakan untuk membangun layanan *backend* dalam bentuk RESTful API pada tahap *Deployment* [98]. Framework ini dipilih karena memiliki performa tinggi dan mampu menangani proses inferensi model secara cepat dan efisien. Dalam penelitian ini, *FastAPI* berfungsi sebagai penghubung antara aplikasi mobile dan model deteksi penyakit mata anterior. Sistem *backend* yang dibangun dijalankan pada lingkungan *Virtual Private Server (VPS)* sebagai infrastruktur *deployment* untuk memastikan layanan dapat diakses secara online. VPS digunakan sebagai server utama yang menjalankan API, model *YOLOV12*, serta proses *preprocessing* dan inferensi citra.

Sistem *backend* menerima input citra dari pengguna melalui aplikasi client, kemudian dilakukan *preprocessing* sebelum diteruskan ke model *YOLOV12* untuk proses deteksi. Setelah proses inferensi selesai, hasil deteksi akan dikirim kembali ke aplikasi dalam bentuk data terstruktur melalui RESTful API. Dengan demikian, *FastAPI* memungkinkan komunikasi antara client dan server berjalan secara cepat dan terintegrasi dalam satu sistem. Penggunaan VPS juga memungkinkan sistem dapat diakses secara jarak jauh serta mendukung ketersediaan layanan secara online. Manfaat penggunaan *FastAPI* dalam penelitian ini meliputi:

1. Performa tinggi dalam menangani request inferensi model.
2. Kemudahan integrasi dengan model deep learning berbasis Python.
3. RESTful API untuk komunikasi antara client dan server.
4. Scalability untuk pengembangan sistem lebih lanjut.

Penggunaan *FastAPI* yang berjalan pada VPS memungkinkan sistem *backend* bekerja secara efisien dalam menangani permintaan dari pengguna. Proses pengiriman dan penerimaan data dapat dilakukan secara real-time sehingga meningkatkan responsivitas sistem. Selain itu, penggunaan VPS memastikan layanan tetap berjalan secara stabil dan dapat diakses dari berbagai perangkat. Struktur API yang jelas memudahkan integrasi dengan aplikasi *frontend* berbasis mobile. Kemampuan *FastAPI* dalam menangani beban kerja yang tinggi juga mendukung pengembangan sistem yang lebih skalabel. Dengan demikian,

kombinasi *FastAPI* dan VPS menjadi komponen penting dalam implementasi sistem berbasis client-server pada penelitian ini.

2.4.4 Flutter

Flutter merupakan framework pengembangan aplikasi mobile yang digunakan untuk membangun antarmuka pengguna dalam penelitian ini [99]. Framework ini memungkinkan pengembangan aplikasi lintas platform sehingga dapat digunakan pada perangkat Android. Dalam penelitian ini, Flutter berfungsi sebagai *frontend* yang berinteraksi langsung dengan pengguna dalam proses pengambilan citra mata. Aplikasi yang dikembangkan mampu mengirimkan citra ke *backend* dan menerima hasil deteksi secara *real-time*. Selain itu, Flutter juga digunakan untuk menampilkan hasil deteksi serta visualisasi *Explainable AI* agar mudah dipahami oleh pengguna. Manfaat penggunaan Flutter dalam penelitian ini meliputi:

1. Cross-platform untuk Android.
2. User interface interaktif untuk menampilkan hasil deteksi.
3. Integrasi API dengan *backend* FastAPI.
4. Responsivitas tinggi untuk pengalaman pengguna yang lebih baik.

Penggunaan Flutter memungkinkan pengembangan aplikasi dilakukan dengan lebih efisien karena hanya menggunakan satu basis kode untuk berbagai platform. Hal ini sangat menghemat waktu dan sumber daya dalam proses pengembangan sistem. Selain itu, tampilan antarmuka yang dihasilkan dapat disesuaikan agar lebih intuitif dan mudah digunakan oleh pengguna. Integrasi dengan *backend* juga berjalan dengan baik melalui API yang telah disediakan.