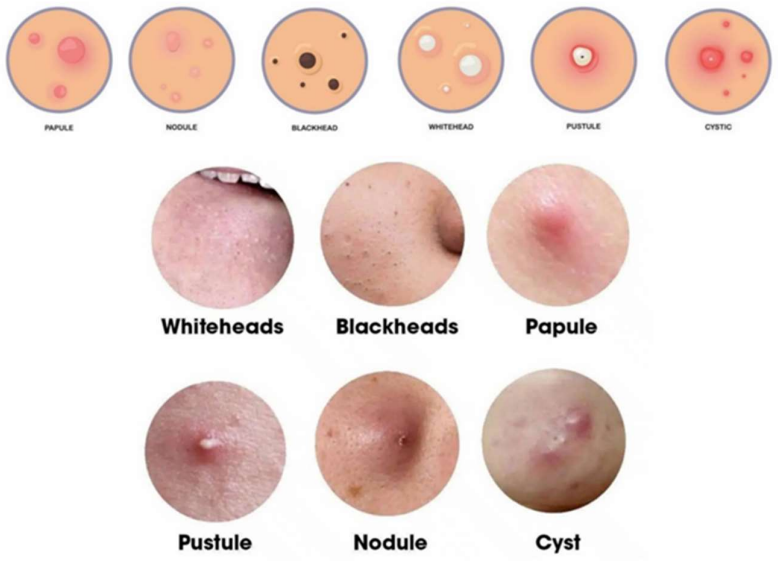
# BAB II
# TINJAUAN PUSTAKA

## 2.1 Landasan Teori

### 2.1.1 Jerawat dan Klasifikasinya

Jerawat, yang secara ilmiah dikenal sebagai acne vulgaris, merupakan masalah kulit yang muncul akibat penyumbatan pori-pori oleh minyak dan sel-sel kulit mati (Hasanah & Hasan, 2022). Ini adalah penyakit kulit inflamasi kronis yang mempengaruhi folikel pilosebasea, ditandai dengan kemunculan lesi seperti komedo, papula, pustula, nodul, dan kista (Truchuelo dkk., 2017). Selain itu, faktor hormonal juga berkontribusi, terutama pada wanita yang mengalami gangguan hormonal seperti sindrom ovarium polikistik (PCOS), yang dapat memicu timbulnya jerawat secara kronis (Kabakchieva, 2024).




Gambar II.1 Jenis Jerawat (Chi dkk., 2024)

Jerawat dapat diklasifikasikan menjadi beberapa jenis berdasarkan karakteristik lesinya. Gambar II.1 menunjukan detail berbagai jenis jerawat.

*a. Blackhead*

*Blackhead* atau yang biasa dikenal dengan komedo terbuka merupakan jenis jerawat ringan yang memiliki ujung berwarna hitam atau gelap alibat penyumbatan pori-pori oleh sebum dan kulit mati yang teroksidasi.

*b. Whitehead*

Sama seperti *blackhead, whitehead* atau yang dikenal dengan komedo tertutup juga merupakan jenis jerawat ringan. *Whitehead* memiliki ujung berwarna putih atau menyerupai warna kulit karena penyumbatan folikel tanpa adanya oksidasi sebum (Hasanah dkk., 2022).

c. Papula

Papula merupakan salah satu bentuk jerawat inflamasi ditandai dengan benjolan merah tanpa nanah dengan ukuran diameter kurang dari 5mm. Papula berwarna merah muda atau merah (Virna dkk., 2024) menandakan adanya peradangan yang disebabkan oleh bakteri *propionibacterium acnes*.

d. Pustula

Sama seperti papula, jerawat pustula merupakan jerawat inflamasi yang disebabkan oleh bakteri *propionibacterium acnes.* Pustula juga memiliki ukuran kurang dari 5 mm. Pustula memiliki ujung berwarna putih atau kuning dengan area peradangan di sekitarnya yang tampak kemerahan (Hasanah & Hasan, 2022).

e. Nodul

Nodul merupakan bentuk jerawat yang lebih parah dan mendalam, ditandai dengan benjolan besar dengan diameter lebih dari 5 mm dan nyeri yang jauh terbentuk di dalam kulit. Kondisi ini sering meninggalkan bekas luka yang lebih dalam akibat kerusakan jaringan (Kabakchieva, 2024).

f. Jerawat Batu (*Cyst*)

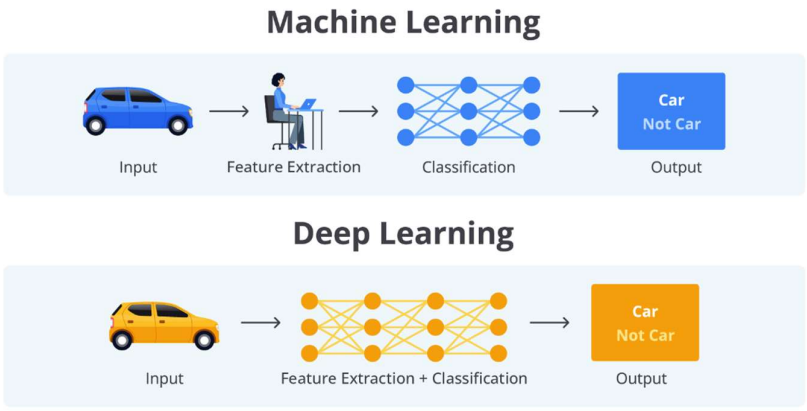
Jerawat batu atau yang disebut dengan *cystic acne* merupakan jerawat yang paling parah. Jerawat ini terbentuk ketika kista tumbuh jauh di bawah kulit. Bakteri, minyak, dan sel kulit mati dapat berkumpul di pori-pori, menciptakan lingkungan yang mendukung pertumbuhan jerawat (Islam dkk., 2023).

### 2.1.2 *Deep Learning*

*Deep Learning* (DL) adalah cabang *dari Machine Learning* (ML) yang menggunakan arsitektur jaringan saraf tiruan (*artificial neural network*) dengan banyak lapisan (*deep neural networks*) untuk mempelajari representasi data. Berbeda dengan algoritma *machine learning* tradisional*, deep learning* mampu mengekstrak fitur secara otomatis dari data mentah, sehingga mengurangi kebutuhan akan rekayasa fitur manual (Huang dkk., 2019). Gambar II.2 menunjukan perbedaan algoritma *machine learning* vs *deep learning.*

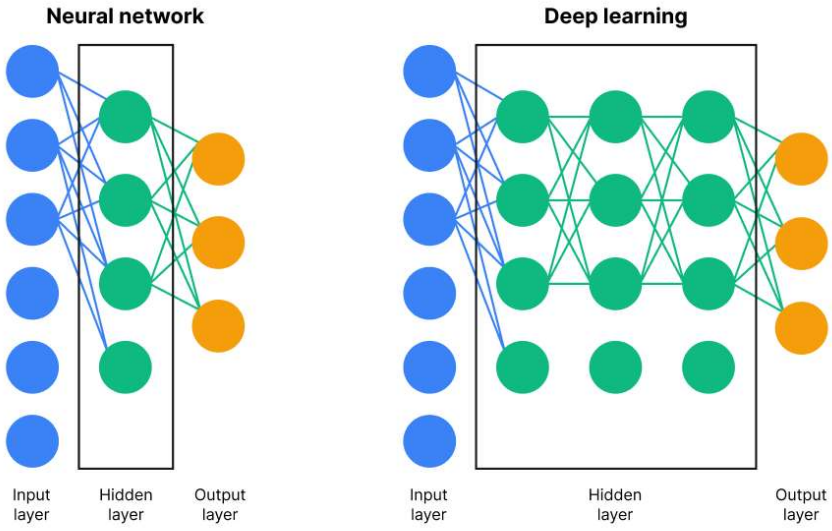
DL berfokus pada bagaimana jaringan saraf tiruan mempelajari pola dari data melalui proses forward dan backward propagation. Dalam DL, terdapat beberapa komponen utama yang mendukung kinerja model. Pertama, fungsi aktivasi berperan dalam mengubah input linear menjadi output non-linear, dengan contoh umum seperti *ReLU, Sigmoid*, dan *Tanh*. Kedua, optimasi dilakukan untuk

meminimalkan fungsi kerugian (*loss function*), yang sering kali menggunakan algoritma seperti *Stochastic Gradient Descent* (SGD) (Sun, 2019). Terakhir, regularisasi diterapkan untuk mengurangi risiko overfitting, menggunakan teknik seperti dropout dan batch normalization guna meningkatkan generalisasi model.




Gambar II.2 Machine Learning Vs Deep Learning (Afinda, 2024)

Jaringan DL terdiri dari beberapa lapisan utama seperti yang ditunjukan oleh gambar II.3.




Gambar II.3 Arsitektur Jaringan Deep Learning (Afinda, 2024)

a. *Input Layer*

Lapisan ini merupakan lapisan pertama dalam jaringan yang menerima data mentah, misalnya gambar, teks, atau suara. *Input* ini kemudian diteruskan ke jaringan tersembunyi *(hidden layer).*

b. *Hidden Layer*

Lapisan tersembunyi dalam jaringan saraf tiruan memproses data dengan menerapkan bobot dan bias pada setiap unit neuron. Struktur jaringan ini dapat bervariasi tergantung pada jenis tugas yang dihadapi. Salah satu arsitektur yang umum digunakan adalah *Multi-Layer Perceptron* (MLP), yang terdiri dari beberapa lapisan neuron dan sering diterapkan dalam tugas klasifikasi sederhana. Untuk pemrosesan data berbasis citra, *Convolutional Neural Network* (CNN) menjadi pilihan utama karena kemampuannya dalam mengekstraksi fitur melalui lapisan konvolusi. Sementara itu, untuk data sekuensial seperti teks dan suara, *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM) digunakan karena kemampuannya dalam menangkap ketergantungan temporal dan pola jangka panjang dalam data.

c. *Output Layer*

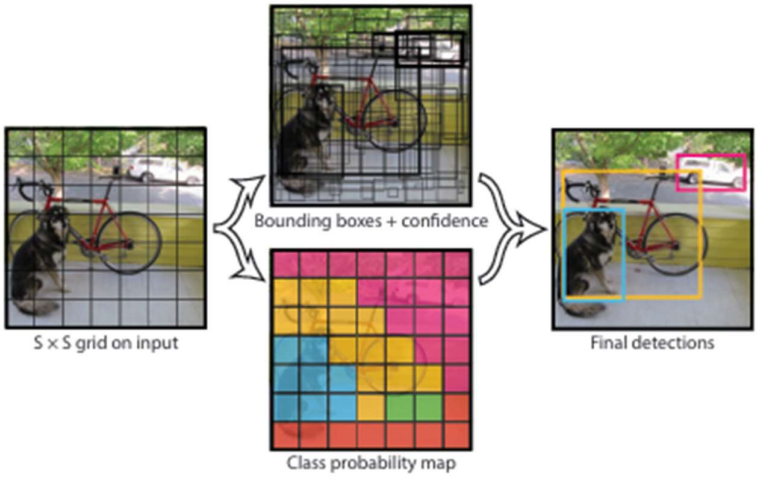
Lapisan terakhir yang memberikan hasil akhir dari prediksi model. Fungsi aktivasi yang umum digunakan di lapisan ini adalah *softmax* untuk klasifikasi atau *linear* untuk regresi.

### 2.1.3 *You Only Look Once* (YOLO)

YOLO adalah salah satu algoritma deteksi objek berbasis *deep learning* yang paling populer karena kemampuannya dalam melakukan deteksi secara *real-time.*

YOLO pertama kali diperkenalkan oleh Redmon dkk. (2016) sebagai metode yang mengubah pendekatan deteksi objek dari *region proposal-based* (seperti Faster R-CNN) menjadi metode berbasis *single-stage*, di mana seluruh gambar dianalisis sekaligus dalam satu proses prediksi (Redmon dkk., 2016). Model ini menggunakan CNN untuk mendeteksi dan mengklasifikasikan objek dalam satu langkah, menjadikannya lebih cepat dibandingkan dengan metode dua tahap seperti Faster R-CNN.

Secara umum, YOLO bekerja dengan membagi gambar input menjadi grid berukuran $S \times S$. Setiap sel grid bertanggung jawab untuk mendeteksi objek yang pusatnya berada dalam sel tersebut. Model ini terdiri dari tiga komponen utama dalam melakukan prediksi seperti yang tertera pada Gambar II.4.




Gambar II.4 Proses Prediksi Model YOLO (Redmon dkk., 2016)

a. *Bounding Box*

*Bounding box* adalah persegi panjang yang mengelilingi objek yang terdeteksi dalam gambar. *Bounding box* ini direpresentasikan oleh empat parameter utama,

yaitu *x* dan *y* yang menunjukan titik pusat *bounding box* dalam sistem koordinat gambar, serta *w* dan *h* menunjukan lebar dan tinggi *bounding box* relatif terhadap ukuran gambar. Setiap sel memprediksi ***B*** *bounding boxes,* dengan masing-masing berisi *x,y,w,h* dan *confidence score. Bounding box* yang memiliki nilai *confidence* tertinggi akan dipilih sebagai hasil deteksi.

Misalnya, jika YOLO membagi gambar 448 x 448 menjadi 7 x 7 *grid* ( $S = 7$), maka setiap sel memiliki ukuran 64 x 64. Jika suatu objek (misalnya mobil) berada dalam sel (3,3), YOLO akan menghasilkan *bounding box* dengan *x,y* dalam skala relatif terhadap sel, dan *w,h* dalam skala relatif terhadap seluruh gambar.

b. *Confidence Score*

Dalam YOLO, *confidence score* digunakan untuk mengukur tingkat keyakinan model bahwa suatu *bounding box* mengandung objek, dengan nilai berkisar antara 0 dan 1. Perhitungan nilai ini didasarkan pada probabilitas keberadaan objek serta *Intersection over Union* (IoU) antara *bounding box* prediksi dan *ground truth*. Jika *confidence score* mendekati 1, model sangat yakin bahwa *bounding box* tersebut berisi objek, sedangkan jika mendekati 0, kemungkinan besar prediksi tersebut salah (*false positive*). Untuk meningkatkan akurasi, YOLO menerapkan *confidence threshold* sehingga hanya *bounding box* dengan nilai di atas ambang batas (misalnya 0.5) yang dipertahankan. Selain itu, *Non-Maximum Suppression* (NMS) digunakan untuk menghilangkan duplikasi *bounding box* pada objek yang sama.

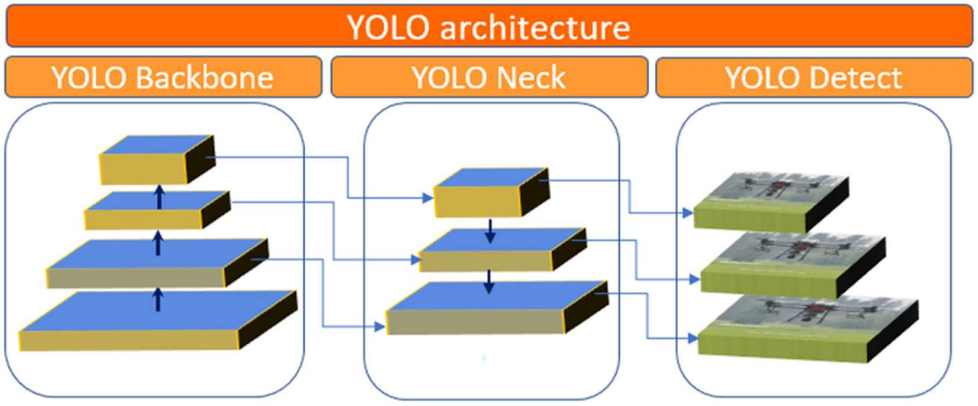
c. *Class Prediction* (Kelas Objek)

Selain menentukan keberadaan objek, YOLO juga memprediksi kategori objek dalam setiap *bounding box* melalui vektor probabilitas kelas. Model akan memilih

kelas dengan probabilitas tertinggi sebagai hasil akhir prediksi. Misalnya, jika YOLO mendeteksi sebuah objek dengan probabilitas 0.85 sebagai mobil, 0.12 sebagai truk, dan 0.03 sebagai manusia, maka objek tersebut diklasifikasikan sebagai mobil. Untuk memastikan bahwa total probabilitas semua kelas berjumlah 1, YOLO menggunakan fungsi aktivasi *Softmax*, yang mengubah skor prediksi mentah menjadi distribusi probabilitas. Dengan pendekatan ini, YOLO dapat mengidentifikasi objek dengan akurasi tinggi berdasarkan kategori yang telah dipelajarinya dari dataset pelatihan seperti COCO atau Pascal VOC.

### 2.1.4 Arsitektur YOLO

Algoritma YOLO terdiri dari tiga komponen utama: *Backbone, Neck,* dan *Head* (Khanam & Hussain, 2024) seperti yang ditunjukan pada Gambar II.5




Gambar II.5 Arsitektur YOLO (Delleji dkk., 2022).

Setiap bagian memiliki peran penting dalam meningkatkan efisiensi dan akurasi deteksi objek.

a. *Backbone*

*Backbone* adalah bagian dari jaringan yang bertanggung jawab untuk mengekstraksi fitur dari gambar input. Biasanya, *Backbone* menggunakan *Convolutional Neural Networks (CNNs)* untuk menghasilkan representasi fitur yang kaya sebelum diteruskan ke lapisan berikutnya.

b. *Neck*

*Neck* adalah lapisan perantara yang menghubungkan Backbone dengan Head. Tujuan utamanya adalah meningkatkan kualitas fitur dengan teknik *seperti Feature Pyramid Network* (FPN) dan *Path Aggregation Network* (PANet)

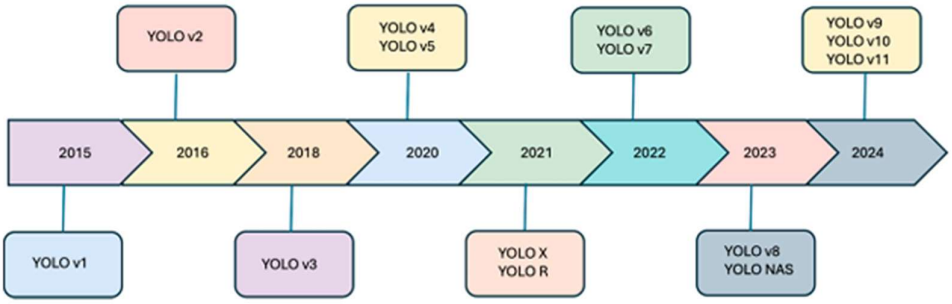
c. *Head (Detect)*

*Head* adalah bagian terakhir dari arsitektur YOLO yang bertanggung jawab untuk menghasilkan prediksi akhir, yaitu: Koordinat *Bounding box, confidence score,* dan *Class Probability.*

### 2.1.5 YOLOv11 dan YOLO Versi Sebelumnya

Seiring perkembangan YOLO, arsitektur ini mengalami berbagai penyempurnaan, seperti penggunaan *anchor-free detection, transformer-based enhancements,* dan *Neural Architecture Search (NAS)*. Gambar II.6 menunjukan evolusi algoritma YOLO dari tahun ke tahun.

Evolusi algoritma YOLO mencapai puncaknya dengan peluncuran YOLOv11, yang menandai kemajuan signifikan dalam teknologi deteksi objek real-time. Iterasi ini mengembangkan kekuatan pendahulunya dan memperkenalkan kemampuan baru yang memperluas aplikasinya dalam berbagai tugas computer vision (CV), termasuk estimasi postur dan segmentasi instance.




Gambar II.6 Evolusi YOLO dari tahun ke tahun (Ali & Zhang, 2024)

Evolusi algoritma YOLO mencapai puncaknya dengan peluncuran YOLOv11, yang menandai kemajuan signifikan dalam teknologi deteksi objek *real-time*. Iterasi ini mengembangkan kekuatan pendahulunya dan memperkenalkan kemampuan baru yang memperluas aplikasinya dalam berbagai tugas *computer vision* (CV), termasuk estimasi postur dan segmentasi *instance*.

Untuk memberikan gambaran yang lebih komprehensif mengenai evolusi YOLO dari versi pertama hingga versi sebelas, Tabel II.1 merangkum perbandingan karakteristik utama dari masing-masing versi berdasarkan studi (Lee & Kim, 2024).

Tabel II.1 Komparasi YOLO Series

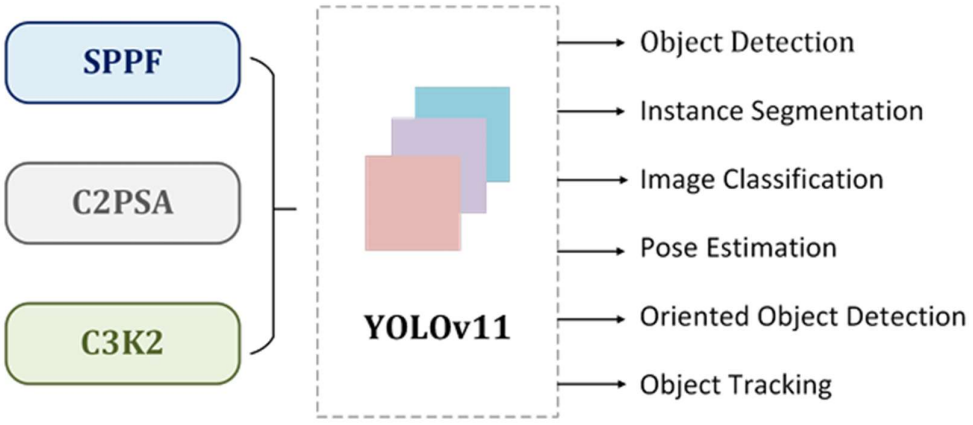
| **Versi** | **Backbone** | **Ciri Arsitektur** | **Keunggulan** | **Kekurangan** |
|---|---|---|---|---|
| YOLOv1 | Darknet | Grid SxS, 2 FC Layer | Kecepatan tinggi, arsitektur sederhana | Deteksi objek kecil lemah, IOU rendah |
| YOLO v2 | Darknet-19 | Batch norm, anchor box, multi-scale | Lebih akurat dari v1, mendukung COCO | Masih kurang untuk objek kecil dan kompleks |

| **Versi** | **Backbone** | **Ciri Arsitektur** | **Keunggulan** | **Kekurangan** |
|---|---|---|---|---|
| YOLO v3 | Darknet-53 | *Residual, FPN, multi-scale* | Baik untuk objek kecil, inference cepat | Ukuran model besar, anchor masih rigid |
| YOLO v4 | CSPDarknet53 | *PANet, Mish, SPP, Bag-of-Freebies* | Akurasi tinggi dengan efisiensi inference | Kompleksitas bertambah |
| YOLO v5 | CSPDarknet53 | *AutoAnchor,* modul skala *(s/m/l/x)* | Modular, mudah dipakai, ringan | Tidak dirilis oleh penulis asli |
| YOLO v6 | CSPStrackRep | *Efficient decoupled head, quantization* | Akurasi & efisiensi tinggi di edge device | Kompatibilitas dengan model lama terbatas |
| YOLO v7 | E-ELAN, RepVGG | *Model scaling, trainable bag-of-freebies* | Sangat cepat, akurasi tinggi | Model besar, konsumsi memori tinggi |
| YOLO v8 | C2f + SPPF | *Anchor-free, streamlined Python interface* | Flexible: deteksi, segmentasi, klasifikasi | Meninggalkan anchor membuat sulit tuning awal |
| YOLO v9 | GELAN + PGI | *Auxiliary training, GELAN, CBLinear, CBFuse* | Akurasi tertinggi (mAP 93.5%) | Latensi tinggi saat training, arsitektur kompleks |
| YOLO v10 | C2f + C2fCIB | *NMS-free, PSA attention* | Tidak butuh NMS, *inference* cepat | Implementasi dan debug lebih kompleks |

| Versi | Backbone | Ciri Arsitektur | Keunggulan | Kekurangan |
|---|---|---|---|---|
| YOLO v11 | C2PSA + C3k2 | *Multi-task:* segmentasi, tracking, klasifikasi | Lebih cepat dari YOLOv8 dengan akurasi lebih tinggi | Belum sepenuhnya terstandarisasi di komunitas |

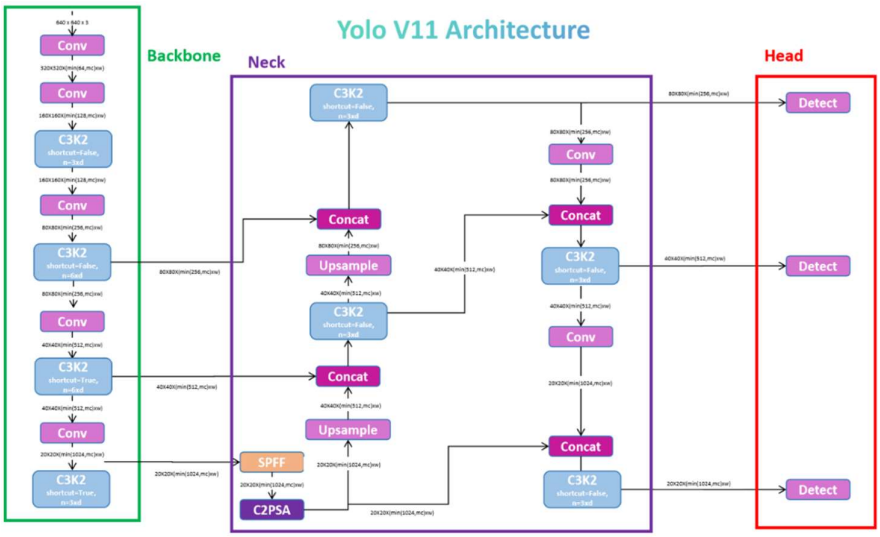
Desain YOLOv11 menekankan keseimbangan antara daya dan kepraktisan, bertujuan untuk mengatasi tantangan spesifik di berbagai industri dengan peningkatan akurasi dan efisiensi. Model ini mencerminkan perkembangan berkelanjutan dalam teknologi deteksi objek *real-time*, mendorong batasan aplikasi CV dan membuka peluang baru untuk implementasi di berbagai sektor (Khanam & Hussain, 2024).

Gambar II.7 menunjukan beberapa inovasi arsitektur model YOLOv11 yang mengintegrasikan blok C3k2 (*Cross Stage Partial* dengan ukuran kernel 2), SPPF (*Spatial Pyramid Pooling - Fast*), dan C2SPA (*Convolutional block with Parallel Spatial Attention*).




Gambar II.7 Modul Utama YOLOv11 (Khanam & Hussain, 2024)

Arsitektur YOLOv11 dikembangkan untuk memaksimalkan kecepatan dan akurasi, dengan mengandalkan kemajuan yang telah diperkenalkan dalam versi YOLO sebelumnya seperti YOLOv8, YOLOv9, dan YOLOv10. Inovasi dalam arsitektur ini bertujuan untuk meningkatkan kemampuan model untuk memproses informasi spasial sambil mempertahankan inferensi dengan kecepatan tinggi. Gambar II.8 menunjukkan rincian arsitektur YOLOv11 yang mencakup setiap komponen utamanya, yaitu *backbone, neck*, dan *head.*




Gambar II.8 Arsitektur YOLOv11 (Nikhileswara Rao, 2024)

Pada bagian *backbone*, YOLOv11 mempertahankan struktur lapisan konvolusi seperti pada versi terdahulu untuk mereduksi ukuran gambar sekaligus membangun dasar ekstraksi fitur. Inovasi penting terletak pada penggantian blok C2f (yang digunakan di YOLOv8 dan YOLOv10) dengan blok C3k2. Blok C3k2 dirancang lebih efisien secara komputasi karena menggunakan dua konvolusi kecil dibandingkan satu konvolusi besar, dengan ukuran kernel yang lebih kecil (“k2”)

untuk meningkatkan kecepatan pemrosesan tanpa mengorbankan kinerja (Khanam & Hussain, 2024).

Pada bagian neck, YOLOv11 tetap menggunakan modul *Spatial Pyramid Pooling Fast* (SPFF) untuk menggabungkan fitur dari berbagai area gambar pada skala berbeda. SPFF ini memungkinkan model mendeteksi objek dengan ukuran beragam, khususnya objek kecil, dengan tetap mempertahankan efisiensi waktu proses secara *real-time* (Nikhileswara Rao, 2024). Modul ini bekerja melalui beberapa operasi *max-pooling* dengan ukuran kernel yang bervariasi untuk menangkap konteks dari berbagai resolusi. Selain itu, YOLOv11 mengintegrasikan blok perhatian C2PSA untuk meningkatkan fokus model pada area-area penting dalam gambar, sehingga dapat meningkatkan deteksi objek kecil atau yang sebagian tertutup melalui pemetaan spasial yang lebih relevan (Khanam & Hussain, 2024). Sementara itu, bagian head masih mengadopsi pendekatan prediksi multi-skala seperti versi YOLO sebelumnya. *Head* ini menghasilkan kotak deteksi dari tiga skala berbeda yaitu, rendah, sedang, dan tinggi, yang masing-masing merujuk pada keluaran dari peta fitur P3, P4, dan P5. Pendekatan ini memastikan objek kecil dikenali melalui peta resolusi tinggi (P3), sedangkan objek lebih besar ditangkap dengan lebih baik melalui peta resolusi rendah seperti P5.

YOLOv11 mendukung berbagai tugas visi komputer, menunjukkan versatilitasnya dalam berbagai aplikasi. Model ini unggul dalam deteksi objek, mampu mengidentifikasi dan melokalisasi objek dalam gambar atau video, serta memberikan kotak pembatas untuk setiap item yang terdeteksi. Selain itu, YOLOv11 dapat melakukan segmentasi *instance*, memisahkan objek individu

hingga tingkat piksel, yang berguna dalam pencitraan medis dan deteksi cacat. Model ini juga mampu mengklasifikasikan gambar, mendeteksi pose dengan melacak titik kunci, serta mendeteksi objek berorientasi untuk lokalisasi yang lebih tepat. Dengan berbagai varian yang dirancang untuk tugas spesifik, YOLOv11 menawarkan fungsionalitas inti seperti inferensi, validasi, pelatihan, dan ekspor, menjadikannya alat yang serbaguna dalam visi komputer (Jocher & Qiu, 2024; Khanam & Hussain, 2024).

Model ini hadir dalam beberapa ukuran arsitektur seperti YOLO11n (*nano*), YOLO11s (*small*), YOLO11m (*medium*), YOLO11l (*large*), dan YOLO11x (*extra-large*) yang detailnya ditunjukan Tabel II.2 (Jocher & Qiu, 2024; Khanam & Hussain, 2024).

Tabel II.2 Varian YOLOv11 Object Detection

| Model | Ukuran Gambar (Pixels) | Akurasi (mAP 50-95) | Kecepatan | Parameter (M) |
|---|---|---|---|---|
| YOLO11n | 640 | 39.5 | Sangat Cepat | 2.6 |
| YOLO11s | 640 | 47.0 | Cepat | 9.4 |
| YOLO11m | 640 | 51.5 | Cukup Cepat | 20.1 |
| YOLO11l | 640 | 53.4 | Agak Lambat | 25.3 |
| YOLO11x | 640 | 54.7 | Lambat | 56.9 |

Variasi ini dapat disesuaikan dengan kebutuhan performa dan sumber daya komputasi. Misalnya YOLO11n cocok untuk perangkat dengan keterbatasan sumber daya karena lebih ringan dan cepat, sedangkan YOLO11x menawarkan akurasi lebih tinggi dengan ukuran model yang lebih besar dan waktu inferensi yang lebih panjang.

### 2.1.6 Evaluasi Multi-Kelas dalam *Object Detection*

Berdasarkan studi oleh (Kothala & Guntur, 2024) dan (Shrawne dkk., 2024), klasifikasi multi-kelas pada objek visual serupa masih menjadi tantangan utama dalam *object detection,* terutama ketika objek bersifat kecil dan memiliki bentuk tumpang tindih. Oleh karena itu, evaluasi performa dalam deteksi objek multi-kelas menjadi krusial. Evaluasi model dalam deteksi objek multi-kelas tidak hanya dilakukan secara global, tetapi juga harus memperhatikan performa per kelas, terutama ketika objek yang dideteksi memiliki visual yang mirip dan jumlah data yang tidak seimbang.

Dalam kasus deteksi jenis jerawat, beberapa kelas seperti *papule, pustule, nodule*, dan *cyst* memiliki ciri visual yang hampir serupa, sehingga sering tertukar oleh model. Maka dari itu, selain menggunakan metrik global seperti *mean Average Precision* (mAP), dibutuhkan metrik per kelas dan *confusion matrix* untuk memahami kinerja klasifikasi secara lebih mendalam.

*a. Confusion Matrix*

*Confusion matrix* merupakan alat evaluasi penting untuk memahami pola kesalahan model. Dalam deteksi jerawat, *confusion matrix* dapat menunjukkan kecenderungan model dalam memprediksi visual yang hampir sama, seperti *papule* sebagai *pustule*, atau *cyst* sebagai *nodule*, dan mengabaikan kelas minor karena jumlah datanya sedikit.

Studi oleh (Bakirci dkk., 2024) menunjukan bahwa *confusion matrix* efektif dalam mengevaluasi performa YOLOv11 pada deteksi kendaraan yang bentuknya mirip (SUV, *hatchback*) secara per kelas.

b. Evaluasi Metrik Per Kelas

Selain *confusion matrix*, evaluasi juga dapat dilakukan menggunakan metrik *precision, recall, F1-score, mean Average* dan *Precision (mAP). Precision* berfungsi untuk mengukur ketepatan prediksi model dalam mengklasifikasikan objek sebagai kelas tertentu. Persamaan (1) merupakan rumus yang untuk evaluasi menggunakan *precision* (Kim dkk., 2021).

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

Dengan TP sebagai *True Positive*, FP sebagai *False Positive*, dan FN sebagai *False Negative*. Sedangkan *Recall* digunakan untuk mengukur kemampuan model dalam menemukan semua objek dari kelas tertentu, menggunakan rumus pada persamaan (2) (Wang dkk., 2022).

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Keseimbangan antara *precision* dan *recall* menjadi aspek krusial dalam menghadapi data yang tidak seimbang, sehingga diperlukan evaluasi menggunakan *F1-score*. Rumus *F1-score* disajikan pada Persamaan (3) (Deepak Khairkar dkk., 2024).

$$F1 - Score = \frac{2\ x\ Precision\ x\ Recall}{Precision+R} \quad (3)$$

Selain *F1-score*, metrik yang juga sangat penting untuk mengevaluasi performa model deteksi objek adalah mAP. mAP mengukur rata-rata presisi dari setiap kelas pada berbagai ambang nilai *Intersection over Union* (IoU), sehingga memberikan gambaran menyeluruh terhadap akurasi model dalam mendeteksi dan

mengklasifikasikan objek secara tepat. Rumus mAP ditunjukkan pada persamaan (4) (Ahmed dkk., 2024).

$$mAP = \frac{\sum_{i=1}^{c} AP_i}{c} \quad (4)$$

*c. Class Imbalance*

Masalah *class imbalance* adalah kondisi ketika jumlah data antar kelas tidak seimbang, menyebabkan model cenderung hanya belajar dari kelas mayoritas. Jika tidak ditangani, hal ini dapat menjadi penyebab signifikan menurunnya performa model. Solusi seperti augmentasi dan evaluasi granular sangat diperlukan agar model tidak hanya akurat secara umum, tetapi juga dalam mengenali semua kelas secara proporsional (Shrawne dkk., 2024; Thapa dkk., 2025).

## 2.2 Penelitian Terkait

Penelitian dalam bidang deteksi jerawat telah mengalami perkembangan pesat seiring dengan kemajuan teknologi *deep learning* dan *computer vision.* Berbagai metode telah diusulkan, mulai dari pendekatan tradisional berbasis pengolahan citra hingga teknik modern yang memanfaatkan arsitektur *deep learning* seperti CNN, YOLO, Faster-RCNN, dan *ensemble learning.* YOLO, khususnya telah menjadi pilihan populer untuk tugas deteksi objek *real-time* karena kecepatan dan akurasinya yang tinggi. Untuk memahami kontribusi dan perbandingan antara penelitian-penelitian terdahulu, Tabel II.3 adalah rangkuman dari penelitian-penelitian yang relevan mengenai deteksi dan klasifikasi jerawat melalui teknik *deep learning.*

Tabel II.3 Penelitian Terkait

| Penulis | Metode | Fokus Pembahasan | Hasil |
|---|---|---|---|
| (Huey Gan dkk., 2024) | Menggunakan YOLOv5 untuk menganalisis kondisi kulit wajah, termasuk deteksi jerawat, pigmentasi, dan pori-pori yang membesar. | Penelitian ini berfokus pada pengembangan sistem analisis kulit yang disesuaikan untuk kulit Malaysia, mengatasi kekurangan alat analisis yang ada di pasar | Eksperimen menunjukkan bahwa precision, recall, dan mAP50 dari YOLOv5 dalam mendeteksi kondisi kulit mencapai sekitar 54%, 44,6%, dan 42,4%. |
| (Veby Agustin dkk., 2024) | Membandingkan performa YOLOv5 dan YOLOv8 dalam mendeteksi jerawat dan mengklasifikasikan tingkat keparahannya. | Penelitian ini membahas perbandingan antara dua versi YOLO dalam konteks deteksi jerawat, serta pengaruh hyperparameter dan ukuran model terhadap performa deteksi. | YOLOv5 menunjukkan performa lebih tinggi dalam mendeteksi jerawat dibandingkan YOLOv8, dengan hyperparameter yang lebih konservatif. |
| (Quattrini dkk., 2022) | Menggunakan model CNN (DenseNet121) untuk mengklasifikasikan wajah dengan dan tanpa jerawat, setelah melakukan segmentasi semantik untuk memfokuskan pada area kulit. | Penelitian ini menekankan pentingnya segmentasi semantik dalam meningkatkan akurasi deteksi jerawat dan bagaimana model dapat digunakan untuk diagnosis dermatologis yang lebih cepat. | Model mencapai rata-rata F1 score sebesar 60.84% dalam membedakan wajah yang terpengaruh jerawat dan yang tidak. |
| (Le dkk., 2024) | Penelitian ini menggunakan YOLOv8 untuk object detection | Fokusnya adalah menggabungkan deteksi jerawat dengan | Hasilnya, model mencapai mAP 0,78 (AP tertinggi 0,94) dan akurasi |

| Penulis | Metode | Fokus Pembahasan | Hasil |
|---|---|---|---|
| | lima jenis jerawat, yaitu *acne, blackheads, papules, pustules, dan sebo-crystan-conglo* serta menilai tingkat kerusakan kulit (level 0–4) dari dataset DermNet-NZ | klasifikasi jenis dan tingkat keparahan. | keseluruhan 78,1%, dengan penilaian tingkat kerusakan kulit yang menunjukkan hasil positif. |
| (Gao dkk., 2025) | Mengembangkan Algoritma AcneDGNet yang menggabungkan deteksi lesi dan penilaian keparahan jerawat menggunakan deep learning | Penelitian ini berfokus pada pengembangan dan evaluasi algoritma *deep learning* yang dapat secara akurat menyelesaikan deteksi lesi jerawat dan penilaian tingkat keparahan secara bersamaan dalam skenario perawatan kesehatan yang berbeda (*online* dan *offline*) | Akurasi penilaian keparahan jerawat mencapai 89.5% dalam skenario *online* dan 89.8% dalam skenario *offline*, menunjukkan kinerja yang lebih baik dibandingkan dermatologis junior. |
| (Virna dkk., 2024) | Menggunakan arsitektur MobileVNet untuk mengklasifikasikan 5 jenis jerawat. | Penelitian ini mengeksplorasi tiga skenario pembagian dataset berbeda: 70/30, 80/20, 90/10 untuk mengevaluasi kinerja dan generalisasi model. Penelitian ini juga menekankan ukuran dataset dan variasi pencahayaan serta kualitas gambar yang | Akurasi Pelatihan dengan skenario terbaik (90:10) mencapai 51% - 80%. |

| Penulis | Metode | Fokus Pembahasan | Hasil |
|---|---|---|---|
| | | dapat mempengaruhi hasil klasifikasi | |
| (Islam dkk., 2023) | Mengembangkan model CNN terintegrasi ganda untuk mengklasifikasikan jerawat ke dalam tujuh kelas. | Fokus utama dari penelitian ini adalah untuk mengenali penyakit jerawat dan mengevaluasi jenisnya. | 97.53% dalam klasifikasi jerawat |
| (Sangha & Rizvi, 2021) | Menggunakan YOLOv5 untuk melatih model deteksi objek pada dataset foto jerawat yang telah dianotasi. | Penelitian ini berfokus untuk melatih model deteksi objek untuk mendeteksi jerawat untuk *single-class* (jerawat) dan multi-kelas (empat tingkat keparahan) | mAP@0.5 sebesar 37.97% |
| (D. Zhang dkk., 2024) | Mengembangkan model deteksi jerawat otomatis menggunakan YOLOv7 yang ditingkatkan dengan modul ELAN dan EPSA. | Fokus utama penelitian ini adalah mengembangkan model otomatis yang lebih efisien untuk mendeteksi jerawat, menggantikan deteksi manual yang memakan waktu dan berisiko tinggi untuk kesalahan. | mAP sebesar 83,7% |
| (Isa & Mangshor, 2021) | Menggunakan YOLOv4 untuk mendeteksi dan mengenali empat jenis jerawat: *cyst*, papula, pustula, dan *whitehead* | Penelitian ini berfokus untuk mengembangkan aplikasi mobile yang dapat mengenali jenis jerawat secara akurat dan otomatis. | Mencapai akurasi rata-rata 91,25% |
| (Faruq Aziz & Saputri, 2024) | Menggunakan YOLOv9 untuk mendeteksi berbagai jenis lesi | Fokus utama dari penelitian ini mengembangkan metode yang efisien | Precision 60.5%, Recall 86.0%, dan mean Average |

| Penulis | Metode | Fokus Pembahasan | Hasil |
|---|---|---|---|
| | kulit, termasuk jerawat. | dan akurat untuk mendeteksi berbagai jenis lesi kulit, termasuk jerawat, untuk diagnosis dermatologis yang tepat | Precision (mAP) 81.4%. |

Berdasarkan Tabel II.3, berbagai penelitian dalam 5 tahun terakhir (2021-2025), terlihat bahwa deteksi jerawat berbasis *deep learning* mengalami perkembangan pesat. Berbagai arsitektur model seperti CNN, ResNet50, DenseNet21, serta variaon YOLO (YOLOv4 – YOLOv9) telah digunakan dalam upaya peningkatan akurasi deteksi jerawat. Dari keseluruhan studi yang dikaji, model YOLO menjadi pendekatan yang paling dominan digunakan karena kemampuannya dalam deteksi objek secara *real-time* dengan akurasi yang cukup tinggi.

Meski demikian, masih terdapat sejumlah tantangan yang belum sepenuhnya teratasi. Sebagian besar penelitian hanya berfokus pada deteksi tingkat keparahan jerwat atau hanya mampu mendeteksi *single-class* jerawat tanpa memperhatikan variasi jenis jerawat. Beberapa penelitian memang telah mengembangkan pendekatan *multi-class,* namun cakupan jenis jerawat masih terbatas dan belum mencakup keseluruhan tipe jerawat yang umum ditemukan. Selain itu, terdapat pula penelitian yang mencampurkan jerawat dan lesi kulit lain, sehingga model deteksi yang dikembangkan menjadi kurang fokus terhadap deteksi jerawat itu sendiri. Tantangan lainnya adalah performa model yang masih belum optimal, yang menunjukan bahwa masih ada ruang untuk untuk perbaikan dan peningkatan kualitas model deteksi jerawat. Pada Tabel II.4 disajikan matriks algoritma

penelitian terdahulu mengenai model deteksi jerawat serta kontribusi dari penelitian ini.

Tabel II.4 Matriks Penelitian

| **Peneliti** | **Algoritma** | | | | | | | | | | **Tujuan Penelitian** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CNN | YOLOv5 | YOLOv8 | ResNet | YOLOv9 | AcneDGNet | YOLOv7 | YOLOv4 | MobileVNet | YOLOv11 | Deteksi Single class | Deteksi Multi-class | Klasifikasi |
| (H. Zhang & Ma, 2022) | v | v | | | | | | | | | v | | v |
| (Wen dkk., 2022) | v | | | | | | | | | | v | | |
| (Huey Gan dkk., 2024) | | v | | | | | | | | | | v | v |
| (Veby Agustin dkk., 2024) | | v | v | | | | | | | | v | | v |
| (Le dkk., 2024) | | | v | | | | | | | | | v | v |
| (Gao dkk., 2025) | | | | | | v | | | | v | v | | |
| (Virna dkk., 2024) | | | | | | | | | v | | | | v |
| (Islam dkk., 2023) | v | | | | | | | | | | | | v |
| (Sangha & Rizvi, 2021) | | v | | | | | | | | | v | | |
| (D. Zhang dkk., 2024) | | | | | | | v | | | | v | | |
| (Isa & Mangshor, 2021) | | | | | | | | v | | | | | v |
| (Faruq Aziz & Saputri, 2024) | | | | | v | | | | | | | v | |
| **Usulan Penelitian** | | | | | | | | | | **v** | | **v** | |

Berdasarkan analisis pada Tabel II.4, penelitian ini bertujuan untuk mengatasi beberapa celah utama. Pertama, penelitian ini mengusulkan penggunaan YOLOv11, versi terbaru dari YOLO, yang diharapkan dapat memberikan performa lebih baik dibandingkan YOLOv5, YOLOv7, YOLOv8, atau YOLOv9. Kedua, penelitian ini berfokus pada deteksi multi-kelas jerawat, mencakup enam jenis lesi jerawat, yang masih jarang dibahas dalam penelitian sebelumnya. Ketiga, penelitian

ini bertujuan untuk efisiensi kinerja sistem secara keseluruhan dengan memanfaatkan keunggulan YOLOv11 dalam aspek kecepatan dan akurasi. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi yang signifikan dalam bidang dermatologi berbasis kecerdasan buatan, khususnya dalam deteksi multi-kelas jerawat pada citra wajah.