

BAB II

TINJAUAN PUSTAKA

A. Hasil Penelitian Terdahulu

Penelitian terkait pengenalan wajah pernah dilakukan oleh Muhammad Syarif Hidayatulloh dengan judul “*Sistem Pengenalan Wajah Menggunakan Metode YOLO (You Only Look Once)*” dengan tujuan mengidentifikasi wajah. Dalam penelitian ini digunakan metode YOLO dan menggunakan dataset pribadi sebanyak 8 wajah berbeda, yang masing-masing wajah berjumlah 10 citra yang kemudian diberi label menggunakan LabelImg untuk memberi kotak pembatas (*bounding box*) beserta nama class pada setiap citra atau disebut anotasi citra. Setelah proses anotasi selesai dataset akan di-*training*. *Training* model dilakukan menggunakan *framework Darknet* di *Google Colab*. Hasil yang di dapat dari penelitian tersebut bahwa akurasi deteksi menggunakan YOLO 100% akurat dengan berbagai sudut pandang dan jarak antar kamera dan objek (Hidayatulloh, 2021).

Penelitian yang dilakukan oleh Chairisni Lubis dan Novario Jaya Perdana ini memanfaatkan algoritma YOLO dan CNN dalam pendeteksian dan pengenalan ekspresi wajah yang mana pada proses klasifikasi dengan menggunakan Deep Learning, lapisan terakhir berfungsi untuk memperkuat perbedaan antara kelompok data (input) dan menghilangkan ciri yang tidak penting dengan menggunakan suatu fungsi diskriminan. Sebelum Deep

Learning melakukan pengenalan ekspresi wajah harus dilakukan pendeteksian pada citra menggunakan algoritma YOLO. Model latih dilatih dengan jumlah 150 epoch dengan tingkat keyakinan rata-rata sebesar 85%, YOLO dapat mendeteksi wajah pada citra dengan baik. Model CNN berarsitektur ResNet50 yang digunakan dalam penelitian ini memiliki performa yang kurang baik. Hal ini dibuktikan dengan akurasi model yang hanya sebesar 46,42%. Secara keseluruhan sistem hanya 3 konten yang mendapatkan hasil yang sesuai dengan kelas ekspresinya. Semua konten mewakili 6 kelas ekspresi yaitu ekspresi marah, jijik, takut, senang, sedih, dan terkejut (Lubis & Perdana, 2020).

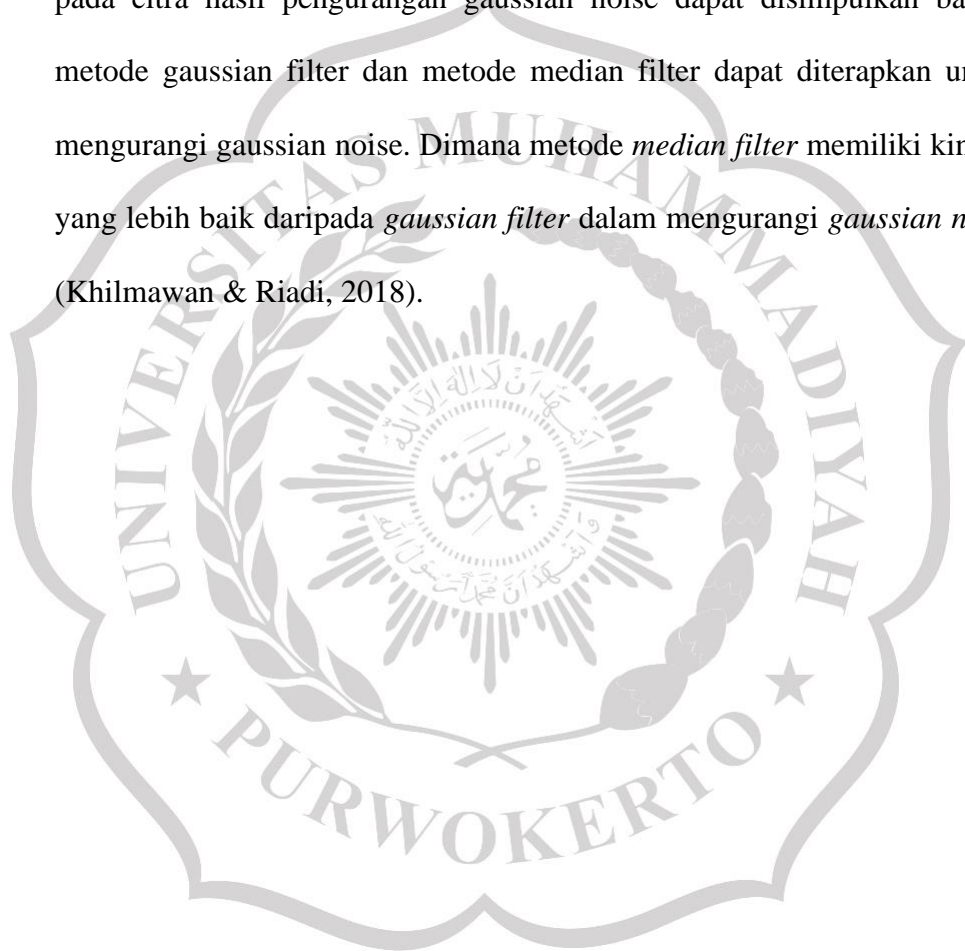
Penelitian yang dilakukan oleh Kevin Adiputra Shianto, Kartika Gunadi dan Endang Setyati ini memanfaatkan metode YOLO dan Faster R-CNN untuk mendeteksi jenis mobil. Untuk identifikasi gambar, salah satu metode yang terkenal adalah Faster R-CNN yang cukup cepat dan tepat untuk melakukan identifikasi gambar. Namun untuk ketepatannya masih belum maksimal. Metode lain yang tersedia adalah YOLO dimana metode ini akan lebih cepat dalam melakukan identifikasi. Jenis mobil yang akan dideteksi adalah sedan, *sport utility vehicle* (SUV), *multi purpose vehicle* (MPV), dan bus. Sistem yang akan dibuat menggabungkan hasil prediksi dari YOLO dan Faster R-CNN menggunakan sebuah feed forward neural network. Hasil dari neural network akan menggunakan data hasil YOLO dan Faster-RCNN sebagai inputnya lalu mengeluarkan sebuah region sebagai outputnya. Hasil dari penelitian ini YOLO memiliki akurasi

prediksi yang lebih rendah dari Faster R-CNN tetapi memiliki prediksi yang lebih akurat daripada Faster R-CNN. Faster R-CNN dengan YOLO mencapai akurasi 70% – 80% (Shianto et al., 2019).

Penelitian yang dilakukan oleh Daling Anggriani Mboeik yaitu membandingkan metode Gaussian Filtering dan Bilateral Filter. Penelitian ini telah menghasilkan sebuah program aplikasi untuk image smoothing yaitu gaussian filter dan bilateral filter. Penelitian ini menggunakan 30 sampel citra RGB bernoise sebagai citra uji. Citra uji di load dan ditampilkan pada program. Dari hasil pengujian, citra yang dismoothing menggunakan bilateral filter lebih baik dalam pengurangan noise. Hal ini berdasarkan hasil uji dimana, rata-rata nilai SNR matrik 3x3 dengan $S = 25$, $R = 7$ adalah 73.12858 db sedangkan nilai PSNRnya adalah 61.54534 (Mboeik, 2016).

Penelitian yang dilakukan oleh Muhammad Rizqi Khilmawan dan Aditya Akbar Riadi melakukan pengurangan noise pada citra tulang menggunakan metode median filter dan gaussian filter. Citra peneliti memiliki sering mengalami penurunan kualitas atau mutu, karena mengandung cacat atau terkena derau (*noise*). Warnanya kurang kontras, kurang tajam, kabur (*bluring*) dan sebagainya. Dalam kondisi yang demikian diperlukan perbaikan citra, yang bertujuan untuk mendapatkan tampilan citra dengan bentuk visualisasi yang lebih baik. Banyak teknik atau metode yang digunakan dalam perbaikan citra, antara lain dengan menggunakan metode median filtering dan gaussian filtering. Dari beberapa metode perbaikan citra tersebut memiliki kelebihan dan kekurangan masing-masing. Citra

akan diuji diberikan gaussian noise pada citra x-ray, kemudian dianalisa performanya secara kuantitatif. Dalam penelitian ini akan dilakukan proses pengukuran dalam mengurangi noise pada citra yang dilakukan dengan metode Peak Signal-to-Noise Ratio (PSNR) yang dilakukan saat proses filtering. Berdasarkan dari hasil pengujian secara kualitatif dan kuantitatif pada citra hasil pengurangan gaussian noise dapat disimpulkan bahwa metode gaussian filter dan metode median filter dapat diterapkan untuk mengurangi gaussian noise. Dimana metode *median filter* memiliki kinerja yang lebih baik daripada *gaussian filter* dalam mengurangi *gaussian noise* (Khilmawan & Riadi, 2018).



Tabel 2.1 Penelitian Terdahulu

No.	Author	Data Penelitian	Metode	Hasil	Penelitian yang akan dilakukan
1.	Muhammad Syarif Hidayatulloh (Hidayatulloh, 2021)	Data yang digunakan merupakan olahan pribadi berupa foto berformat .jpg.	YOLO	Akurasi deteksi menggunakan YOLO 100% akurat dengan berbagai sudut pandang dan jarak antar kamera dan objek.	Klasifikasi wajah menggunakan YOLOv3
2.	Chairisni Lubis dan Novario Jaya Perdana (Lubis & Perdana, 2020)	Data yang digunakan oleh aplikasi berupa data dalam bentuk citra dua dimensi yang dibagi menjadi 2 yaitu citra latih dan citra validasi.	YOLO dan CNN	Model YOLO dengan tingkat keyakinan rata-rata sebesar 85%. Model CNN berarsitektur ResNet50 yang memiliki performa yang kurang baik, dengan akurasi model yang hanya sebesar 46,42%.	Klasifikasi wajah menggunakan YOLOv3
3.	Kevin Adiputra Shianto, Kartika Gunadi dan Endang Setyati (Shianto et al., 2019)	Data yang digunakan adalah gambar jenis mobil sedan, <i>sport utility vehicle</i> (SUV), <i>multi purpose vehicle</i> (MPV), dan bus.	YOLO dan Faster R-CNN	YOLO memiliki akurasi prediksi yang lebih rendah dari Faster R-CNN tetapi memiliki prediksi yang lebih akurat daripada Faster R-CNN. Faster R-CNN dengan YOLO mencapai akurasi 70% – 80%.	Klasifikasi wajah menggunakan YOLOv3
4.	Daling Anggriani Mboeik (Mboeik, 2016)	Data yang adalah citra RGB bernoise	Gaussian Filter dan Bilateral Filter	Hasil perbandingan citra hasil metode gaussian filter dengan bilateral filter, menunjukkan bahwa	Perbaikan kualitas citra dengan Gaussian Filter

kinerja metode bilateral filter lebih baik dalam penapisan derau.

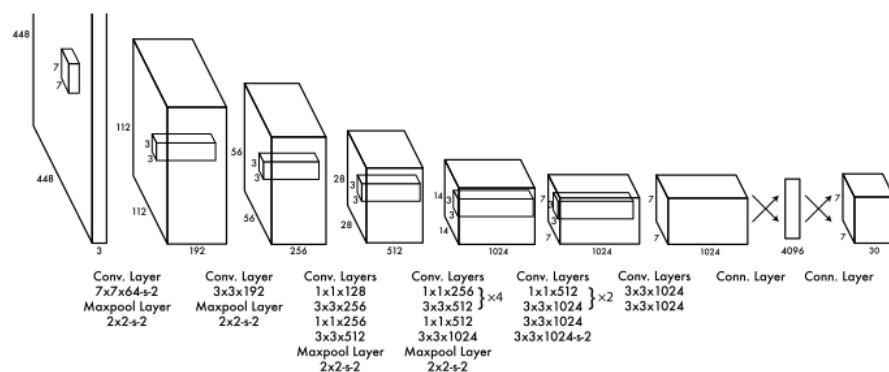
No.	Author	Data Penelitian	Metode	Hasil	Penelitian yang akan dilakukan
5.	Muhammad Rizqi Khilmawan dan Aditya Akbar Riadi (Khilmawan & Riadi, 2018)	data yang digunakan adalah citra yang mengandung cacat atau terkena derau (noise) warnanya kurang kontras, kurang tajam, kabur (bluring).	Gaussian filter dan Median Filter	metode median filter memiliki kinerja yang lebih baik daripada gaussian filter dalam mengurangi gaussian noise	Perbaikan kualitas citra dengan Gaussian Filter

B. Landasan Teori

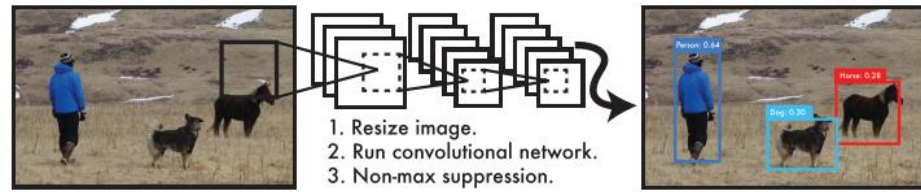
1. Algoritma YOLO (You Only Look Once)

You Only Look Once (Yolo) adalah sebuah algoritma yang dirancang untuk mendeteksi objek dalam *real-time*. Algoritma ini menggunakan sebuah *classifier* atau *localizer* yang sudah dimodifikasi untuk mendeteksi objek. Model yang digunakan diterapkan pada citra di berbagai lokasi dan skala. Daerah dengan skor citra tertinggi akan dianggap sebagai hasil deteksi objek (Jupiyandi et al., 2019).

YOLO memiliki jaringan deteksi yang terdiri dari 24 lapisan konvolusi, diikuti oleh 2 lapisan yang terhubung secara penuh. Beberapa lapisan konvolusi menggunakan lapisan reduksi 1x1 sebagai alternatif untuk mengurangi kedalaman *feature maps* yang diikuti oleh lapisan konvolusi 3x3. Ini ditunjukkan pada Gambar 1. Terdapat tiga langkah dalam mendeteksi objek menggunakan YOLO yang diilustrasikan pada gambar 2.



Gambar 2.1 Arsitektur YOLO (Redmon et al., 2016)



Gambar 2.2 Sistem deteksi YOLO (Redmon et al., 2016)

YOLO menguraikan gambar masukan ke dalam grid $S \times S$. Jika pusat suatu objek ada dalam sel *grid*, maka sel *grid* bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel *grid* memprediksi kotak pembatas (*bounding boxes*) B dan nilai keyakinan (*confidence score*) untuk kotak tersebut, serta probabilitas kelas kondisional C . Nilai keyakinan ini menggambarkan seberapa akurat perkiraan kotak tersebut. YOLO mendefinisikan *confidence* sebagai $Pr(Object)$.

Jika sel tidak mendeteksi adanya objek, maka nilai keyakinannya akan bernilai nol. Jika tidak, sistem ingin nilai keyakinan sama dengan *Intersection Over Union* (IoU) antara kotak prediksi dan *ground truth*. Prediksi pada YOLO menggunakan arsitektur yang mirip dengan *Convolutional Neural Networks* (Jaringan Saraf Tiruan Konvolusi). (Redmon et al., 2016). YOLO hanya menggunakan lapisan konvolusi dan lapisan pooling sebagai komponennya. Lapisan konvolusi terakhir disesuaikan sesuai dengan jumlah kelas dan jumlah kotak prediksi yang diinginkan (Uijlings et al., 2018).

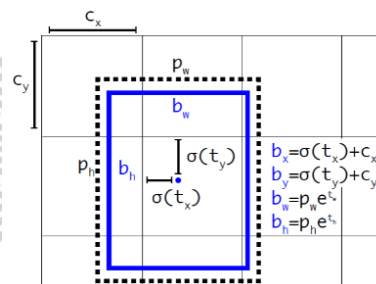
Faktor utama yang mempengaruhi prediksi akhir adalah nilai *confidence class*, yang didasarkan pada *Class confidence score* dan

nilai *bounding box*. *Class confidence score* mengukur nilai kepercayaan terhadap klasifikasi dan lokalisasi objek. *Class confidence score* memberikan nilai kepercayaan kelas tertentu untuk setiap kotak, yang mengkodekan kemungkinan kelas yang muncul di kotak dan seberapa sesuai kotak yang diprediksi dengan objek. Persamaan pada *class confidence score* untuk setiap kotak prediksi ditunjukkan pada Persamaan (1):

$$\Pr(\text{Class}_i | \text{Object}) * \Pr(\text{Object}) * \text{IoU} \frac{\text{truth}}{\text{pred}} = \Pr(\text{Class}_i) * \text{IoU} \frac{\text{truth}}{\text{pred}} \quad (1)$$

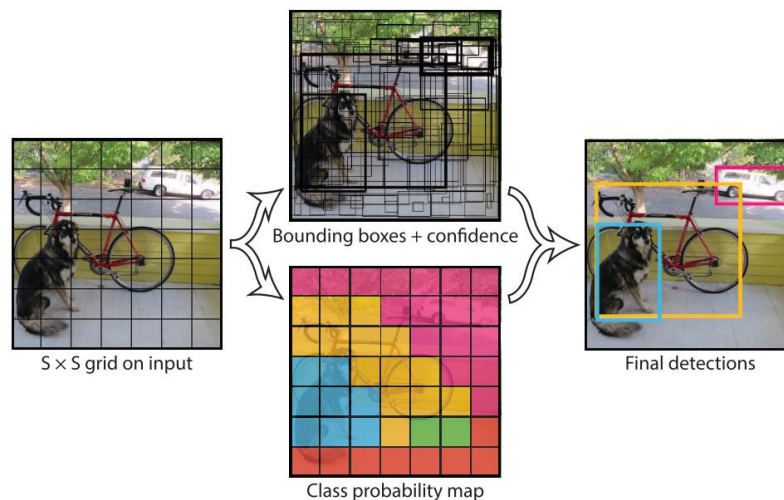
$\Pr(\text{Class}_i | \text{Object})$: probabilitas kondisional kelas i .

$\Pr(\text{Object})$: probabilitas kelas i .



Gambar 2.3 Bounding Box pada YOLO (Redmon et al., 2016)

Dari persamaan tersebut, akan mendapatkan nilai *confidence* dari kelas spesifik. Nilai ini akan merepresentasikan probabilitas kelas yang muncul di dalam kotak dan seberapa akurat kotak yang diprediksi. Hal ini membagi gambar menjadi grid dan setiap grid memprediksi *bounding boxes*, nilai *confidence* dari setiap kotak dan kelas probabilitas. Ilustrasi dapat dilihat pada gambar 2.4.



Gambar 2.4 Proses deteksi pada YOLO (Redmon et al., 2016)

Penelitian ini menggunakan YOLOv3 dimana Darknet-53 sebagai *feature extractor*. Darknet-53 adalah sebuah arsitektur jaringan neural konvolusi yang dikembangkan untuk sistem deteksi objek YOLO. Darknet-53 terdiri dari 53 lapisan konvolusi dan digunakan untuk mengekstrak fitur dari citra input. Arsitektur ini didesain untuk meningkatkan akurasi dan kecepatan dalam mendeteksi objek dibandingkan dengan arsitektur yang lebih sederhana. Darknet-53 merupakan salah satu bagian penting dari sistem deteksi objek YOLO.

2. OpenCV

OpenCV adalah salah satu library yang menyediakan fungsi-fungsi untuk pemrograman computer vision dalam waktu nyata. OpenCV merupakan perangkat lunak open-source yang dapat digunakan untuk tujuan akademis dan komersial. Library ini menyediakan antarmuka untuk bahasa pemrograman C, C++, Python, dan Java yang dapat

dijalankan di sistem operasi Windows, Mac, Linux, dan Android. (Budiarjo, 2020). OpenCV memiliki lebih dari 2500 algoritma yang telah dioptimalkan untuk dipakai.

3. Gaussian Filter

Gaussian merupakan nama seorang matematikawan Jerman, Karl Friedrich Gauss, seorang matematikawan Jerman yang terkenal karena kontribusinya dalam bidang matematika. Gaussian Filter adalah salah satu filter linier yang menggunakan nilai pembobot yang dipilih berdasarkan bentuk fungsi Gaussian untuk setiap anggotanya. Filter ini sering digunakan untuk menghapus noise dari citra atau sinyal yang diinput ke sistem.

Gaussian Filter adalah salah satu metode yang efektif untuk menghilangkan noise yang disebabkan oleh distribusi normal, seperti yang sering ditemukan pada citra hasil digitalisasi menggunakan kamera. Noise ini biasanya merupakan akibat pantulan cahaya dan kepekaan sensor cahaya pada kamera. Gaussian filter dapat mengurangi noise ini dengan cara menghitung rata-rata dari setiap piksel dan tetangganya, sehingga menghasilkan citra yang lebih bersih dan jelas (Mboeik, 2016).

Untuk menghitung atau menentukan nilai-nilai dari elemen dalam gaussian filter yang akan dibentuk berlaku persamaan (Verma & Mishra, 2015):

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2)$$

Dimana:

1. σ (sigma) adalah nilai standar deviasi distribusi normal yang digunakan. Makin besar nilai σ maka makin banyak titik tetangga yang dikutkan dalam perhitungan.
2. x dan y adalah posisi koordinat mask dimana koordinat (0,0) adalah posisi titik tengah dari mask yang mempunyai nilai paling besar.
3. π adalah konstanta dengan nilai 3,14.
4. e adalah konstanta bilangan natural dengan nilai 2,718281828.

4. RoboFlow

RoboFlow menyediakan alat-alat untuk mengubah gambar mentah menjadi model visi komputer yang terlatih dan dapat digunakan dalam aplikasi. Tugas-tugas yang dapat dilakukan oleh RoboFlow di antaranya adalah:

1. Menyiapkan dataset gambar yang dibutuhkan untuk melatih model visi komputer.
2. Mengkonfigurasi arsitektur jaringan neural yang digunakan untuk melatih model visi komputer.
3. Menjalankan proses pelatihan model visi komputer dan mengevaluasi hasilnya.
4. Menggunakan model visi komputer yang terlatih untuk mendeteksi objek dalam gambar atau video.

5. Mengintegrasikan model visi komputer ke dalam aplikasi untuk digunakan dalam aplikasi.
6. Anotasi gambar atau unggah anotasi yang ada.
7. Melihat apakah label ada dalam bingkai (perbaiki sekali klik).
8. Gambar pra-pemrosesan: mengubah ukuran, skala abu-abu, orientasi otomatis, penyesuaian kontras (Roboflow, 2021).

5. Python

Python adalah bahasa pemrograman tingkat tinggi yang dapat diinterpretasikan, berorientasi objek, dan memiliki semantik dinamis. Struktur data tingkat tinggi bawaan, dikombinasikan dengan pengetikan dinamis dan pengikatan dinamis, membuatnya sangat cocok untuk pengembangan aplikasi yang cepat, serta untuk digunakan sebagai bahasa skrip atau lem untuk menghubungkan komponen yang ada bersama-sama. Sintaks Python yang sederhana dan mudah dipahami menekankan keterbacaan, sehingga mengurangi biaya pemeliharaan program. Python mendukung modul dan paket, yang mendorong modularitas program dan penggunaan ulang kode. Penerjemah Python dan perpustakaan standar yang luas tersedia dalam bentuk sumber atau biner secara gratis untuk semua platform utama, dan dapat didistribusikan secara bebas.

6. Google Colaboratory

Colaboratory, atau disingkat "Colab", adalah produk dari Google Research. Colab memungkinkan siapa saja untuk menulis dan

mengeksekusi kode python melalui browser, dan sangat cocok untuk pembelajaran mesin, analisis data, dan pendidikan. Secara teknis, Colab adalah layanan notebook Jupyter yang dihosting oleh Google, yang tidak memerlukan persiapan apa pun untuk digunakan, dan memberikan akses gratis ke sumber daya komputasi termasuk GPU.

7. *Intersection over Union (IoU)*

Intersection over Union (IoU) adalah metrik evaluasi untuk mengukur seberapa akurat sebuah detektor objek dalam mengidentifikasi objek pada dataset tertentu. IoU dapat digunakan dengan ketentuan sebagai berikut:

- a. Memiliki *ground-truth bounding box* pada dataset objek
- b. Prediksi *bounding box* pada dataset objek

Nilai IoU berkisar antara 0 dan 1, di mana nilai 1 menunjukkan bahwa kotak deteksi sama persis dengan ground truth, sedangkan nilai 0 menunjukkan bahwa kotak deteksi tidak ada hubungannya dengan ground truth. Semakin dekat nilai IoU ke 1, semakin akurat deteksi objek tersebut. Seperti pada persamaan dibawah dan ilustrasi persamaan IoU dapat dilihat pada Gambar 8 (Hanafi, 2020).

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

Gambar 2. 5 Intersection over Union

8. *Confusion Matrix*

Confusion matrix merupakan sebuah metode yang biasa digunakan untuk melakukan pengukuran pada suatu *classifier* dalam melakukan prediksi dari kelas yang berbeda. Pengujian dilakukan untuk memperoleh hasil. Untuk perhitungan tingkat akurasi akan digunakan proses perhitungan *Precision*, *Recall* dan *Accuracy* (Jamhari et al., 2020). Pengujian menggunakan confusion matrix empat kondisi yaitu: *True Positive* (TP), *True negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Tabel 2.2 menunjukkan tabel confusion matrix akan memiliki 4 nilai.

Tabel 2.2 Tabel Cofusion Matrix

	Kelas <i>Positive</i> Prediksi	Kelas <i>Negative</i> Prediksi
<i>Positive</i>	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
<i>Negative</i>	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

True Positive (TP) merupakan jumlah banyaknya prediksi wajah yang benar, *False Positive (FP)* merupakan jumlah banyaknya hasil prediksi wajah yang tak terduga, *False Negative (FN)* merupakan hasil yang tidak sesuai, *True Negative (TN)* merupakan jumlah data yang tidak terprediksi (Jamhari et al., 2020).

Hasil evaluasi menghasilkan *Precision*, *Recall* dan *Accuracy*. Nilai *Precision* adalah rasio *item* relevan yang dipilih terhadap semua *item* yang terpilih (Jamhari et al., 2020). Penghitungan *Precision* dapat dilakukan menggunakan persamaan (3):

$$\text{Precision} = \frac{(TP)}{(TP+FP)} \times 100\% \quad (3)$$

Nilai *Recall* adalah persentase sebuah program memprediksi sebuah data ke bukan kelas aktualnya, *recall* biasa disebut juga dengan sensitifitas. Perhitungan *recall* dapat dilakukan menggunakan persamaan (4):

$$\text{Recall} = \frac{(TP)}{(TP+FN)} \times 100\% \quad (4)$$

Nilai *accuracy* merupakan persentase dari suatu kelas terprediksi dengan benar oleh model yang sudah dibuat. Perhitungan *accuracy* dapat dilakukan menggunakan persamaan (5):

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100\% \quad (5)$$