

**DETEKSI *TERATOZOOSPERMIA* BERDASARKAN MORFOLOGI  
SPERMA MANUSIA MENGGUNAKAN YOLOv8**

**(Skripsi)**

**Oleh**

**NABILLAH AISYAH**

**2117051036**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2025**

**DETEKSI *TERATOZOOSPERMIA* BERDASARKAN MORFOLGI  
SPERMA MANUSIA MENGGUNAKAN YOLOv8**

**Oleh**

**NABILLAH AISYAH**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar  
SARJANA KOMPUTER**

**Pada**

**Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2025**

## ABSTRAK

### DETEKSI *TERATOZOOSPERMIA* BERDASARKAN MORFOLOGI SPERMA MANUSIA MENGGUNAKAN YOLOv8

Oleh

NABILLAH AISYAH

Infertilitas menjadi masalah yang semakin banyak ditemui di seluruh dunia, terutama pada pria yang berkontribusi signifikan terhadap banyak kasus yang terjadi. *Teratozoospermia* merupakan kondisi yang ditandai dengan morfologi sperma abnormal yang telah diidentifikasi sebagai salah satu penyebab utama infertilitas pria. Penelitian ini bertujuan untuk meningkatkan deteksi dan klasifikasi *teratozoospermia* dengan memanfaatkan teknik *deep learning* yaitu model YOLOv8. Dataset yang digunakan adalah dataset VISEM yang terdiri dari 85 video sperma yang diproses menjadi gambar, dianotasi, dan diaugmentasi untuk meningkatkan performa model. Data dibagi menjadi 3 jenis yaitu *training*, *validation*, dan *testing* dengan 2 jenis kelas label yaitu normal dan abnormal. Serangkaian teknik yang digunakan yaitu *resizing*, *cropping*, dan augmentasi untuk meningkatkan kualitas data. Selain deteksi, penelitian ini menerapkan algoritma Bytetrack untuk melakukan *tracking* pergerakan sperma dan menghitung jumlah sperma normal dan abnormal secara akurat. Evaluasi model dilakukan menggunakan metrik seperti *precision*, *recall*, *f1-score* dan *mean average precision* (mAP). Hasil terbaik diperoleh pada skenario pembagian data 80:10:10 dengan 120 *epoch* dengan *oversampling* yang menghasilkan nilai *precision* 96.5%, *recall* 97.4%, *f1-score* 96.9% dan mAP 98.9%. Model YOLOv8 menunjukkan performa baik dalam deteksi secara *real-time* dan dikombinasikan dengan Bytetrack mampu mendukung proses analisa sperma yang lebih akurat dan efisien untuk mendukung diagnosa infertilitas pria.

Kata Kunci : *Teratozoospermia*, *Deep Learning*, YOLOv8, Deteksi Objek, Bytetrack, Kasifikasi, Sperma.

## ABSTRACT

### ***DETECTION OF TERATOZOOSPERMIA BASED ON THE MORPHOLOGY OF HUMAN SPERM USING YOLOv8***

*By*

**NABILLAH AISYAH**

*Infertility is becoming an increasingly prevalent problem worldwide, with men contributing significantly to many of the cases. Teratozoospermia is a condition characterized by abnormal sperm morphology that has been identified as one of the main causes of male infertility. This study aims to improve the detection and classification of teratozoospermia by utilizing deep learning techniques, namely the YOLOv8 model. The dataset used is the VISEM dataset consisting of 85 sperm videos that are processed into images, annotated, and augmented to improve model performance. The data is divided into 3 types namely training, validation, and testing with 2 types of label classes namely normal and abnormal. A series of techniques were used such as resizing, cropping, and augmentation to improve the data quality. In addition to detection, this study applies the ByteTrack algorithm to track sperm movement and accurately count the number of normal and abnormal sperm. Model evaluation is performed using metrics such as precision, recall, f1-score and mean average precision (mAP). The best results were obtained in the 80:10:10 data division scenario with 120 epochs with oversampling which resulted in a precision value of 96.5%, recall 97.4%, f1-score 96.9% and mAP 98.9%. The YOLOv8 model shows good performance in real-time detection and combined with ByteTrack can support a more accurate and efficient sperm analysis process to support male infertility diagnosis.*

*Keywords: Teratozoospermia, Deep Learning, YOLOv8, Object Detection, ByteTrack, Classification, Sperm.*

Judul Skripsi : **DETEKSI TERATOZOOSPERMIA  
BERDASARKAN MORFOLOGI SPERMA  
MANUSIA MENGGUNAKAN YOLOv8**

Nama Mahasiswa : **Nabillah Aisyah**

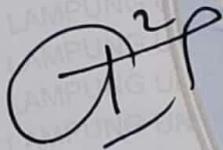
Nomor Pokok Mahasiswa : 2117051036

Program Studi : S1-Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam

**MENYETUJUI**

1. Komisi Pembimbing

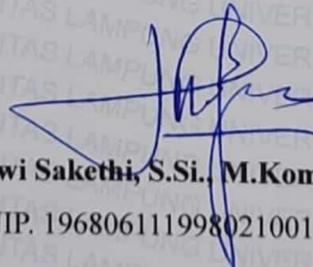


**Dr. Aristoteles, S.Si., M.Si**  
NIP. 19810521 200604 1 002



**Rahman Taufik, S.Pd., M.Kom**  
NIP. 19930627 202203 1 007

2. Ketua Jurusan Ilmu Komputer

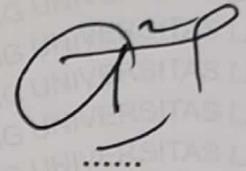


**Dwi Sakethi, S.Si., M.Kom**  
NIP. 196806111998021001

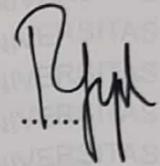
**MENGESAHKAN**

1. Tim Penguji

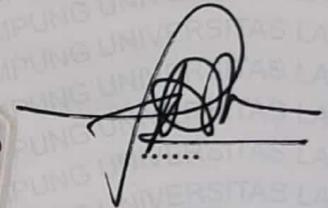
Ketua : **Dr. Aristoteles, S.Si., M.Si**



Sekretaris : **Rahman Taufik, S.Pd., M.Kom**



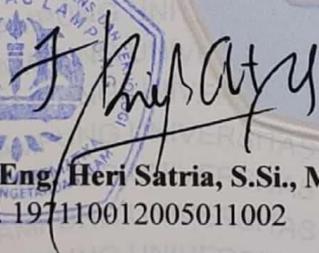
Penguji  
Bukan Pembimbing : **Favorisen R. Lumbanraja, Ph.D**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



**Dr. Eng. Heri Satria, S.Si., M.Si.**  
NIP. 197110012005011002



Tanggal Lulus Ujian Skripsi: **4 Juni 2025**

## PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Nabillah Aisyah

NPM : 2117051036

Dengan ini menyatakan bahwa skripsi saya yang berjudul “**Deteksi Teratozoospermia Berdasarkan Morfologi Sperma Manusia Menggunakan YOLOv8**” merupakan hasil karya saya sendiri dan bukan karya orang lain. Seluruh tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil jiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 13 Juni 2025



Nabillah Aisyah  
NPM. 2117051036

## RIWAYAT HIDUP



Penulis dilahirkan di Jakarta pada tanggal 7 Agustus 2003 sebagai anak keempat dari 4 bersaudara, dari pasangan Bapak Sahroni dan Ibu Nur Aini. Penulis menyelesaikan pendidikan formal pertama kali di Taman Kanak-kanak (TK) Gajah Mada dan selesai pada tahun 2008, lalu pendidikan Sekolah Dasar (SD) diselesaikan di SD Negeri 1 Kota Baru pada tahun 2015. Kemudian pendidikan menengah pertama di SMP Negeri 1 Bandar Lampung pada tahun 2018, lalu melanjutkan ke pendidikan menengah atas di SMA Negeri 1 Bandar Lampung yang diselesaikan pada tahun 2021.

Pada tahun 2021 penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SNMPTN. Selama menjadi mahasiswa mahasiswa, penulis melakukan beberapa kegiatan antara lain:

1. Menjadi Anggota Muda Ilmu Komputer (ADAPTER) Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2021/2022.
2. Menjadi Anggota Bidang Internal Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2021/2022 .
3. Menjadi Bendahara Bidang Internal Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2022/2023.
4. Menjadi Asisten Dosen Jurusan Ilmu Komputer pada mata kuliah Logika tahun 2022, Struktur Data dan Algoritma dan Basis Data pada tahun 2023, dan Pemrosesan Data Terdistribusi pada tahun 2024.

5. Menjadi anggota panitia Divisi Lomba Cepat Tepat Komputer pada acara pekan Raya Jurusan Ilmu Komputer tahun 2022 dan 2023.
6. Melaksanakan Kerja Praktik di PT. Jasaraharja Putera Cabang Bandar Lampung pada periode 2023/2024 dengan program kerja pengembangan *website* dan divisi *finance*.
7. Mengikuti Studi Independen MSIB di Karier.mu pada tahun 2024 di bidang *Data Analyst*.
8. Mengikuti Kuliah Kerja Nyata (KKN) periode II tahun 2024 di Desa Kedaton II, Kecamatan Batang Hari Nuban, Kabupaten Lampung Timur dengan program kerja edukasi penipuan online.

## MOTTO

1. *“Karena sesungguhnya sesudah kesulitan itu ada kemudahan, sesungguhnya sesudah kesulitan itu ada kemudahan.”*  
(QS. Al Insyirah: 5-6)
2. *“Allah tidak membebani seseorang melainkan dengan kesanggupannya.”*  
(QS. Al Baqarah 286)
3. *“Sesungguhnya Allah tidak akan mengubah nasib suatu kaum hingga mereka mengubah diri mereka sendiri.”*  
(QS. Ar-Ra'd : 11)
4. *“Tugas kita bukanlah untuk berhasil, tugas kita adalah untuk mencoba disitulah kita menemukan kesempatan untuk berhasil.”*  
(Buya Hamka)
5. *“It's fine to fake it until you make it, until you do, until it's true.”*  
(Taylor Swift)
6. *“Setetes keringat orang tuaku yang keluar, ada seribu langkahku untuk maju.”*  
(Nabillah Aisyah)

## **PERSEMBAHAN**

*Alhamdulillahirobbilalamin*

Puji Syukur kehadiran Allah Subhanahu Wa Ta'ala atas segala Rahmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan dengan sebaik-baiknya.

Shalawat serta salam selalu tercurahkan kepada junjungan Nabi Agung Muhammad Shallallahu 'Alaihi Wassalam.

Kupersembahkan karya ini kepada:

### **Kedua Orang Tuaku Tercinta**

Dengan segenap cinta dan hormat aku ucapkan terima kasih yang tak terhingga atas segala doa, dukungan, serta kasih sayang yang tiada pernah putus. Terima kasih atas setiap pengorbanan dan perjuangan dalam mendidik serta membesarkanku yang menjadi kekuatan terbesar dalam hidupku dan takkan pernah mampu kubalas sepenuhnya.

### **Seluruh Keluarga Besar Ilmu Komputer 2021**

Yang senantiasa membantu dan memberikan dukungan

**Alamamater Tercinta, Universitas Lampung dan Jurusan Ilmu Komputer**

Tempat bernaung mengemban semua ilmu untuk menjadi bekal hidup.

## SANWACANA

Puji Syukur penulis panjatkan kehadirat Allah Subhanahu Wa Ta'ala Tuhan Yang Maha Esa, karena atas berkat, rahmat, dan karunia-Nya. Shalawat serta salam senantiasa tercurahkan kepada junjungan Nabi Muhammad Shallallahu 'Alaihi Wassalam, sehingga penulis dapat menyelesaikan skripsi yang berjudul "**Deteksi Teratozoospermia Berdasarkan Morfologi Sperma Manusia Menggunakan YOLOv8**" dengan baik dan tepat waktu. Keberhasilan dan suksesnya penyelesaian skripsi ini tidak terlepas dari bimbingan dan bantuan serta kontribusi dari berbagai pihak yang selalu menjadi semangat dan motivasi penulis dalam menyelesaikan pembuatan skripsi ini. Dalam kesempatan yang berbahagia ini penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah Subhanahu Wa Ta'ala Tuhan Yang Maha Esa yang telah memberikan berkat, Rahmat, dan petunjuk-Nya selama proses penyusunan skripsi sehingga penulis pada akhirnya dapat menyelesaikannya dengan lancar.
2. Kedua Orang Tua yaitu mama tersayang dan bapak tercinta yang selalu dan senantiasa memberikan doa, cinta, dan pengorbanan yang tiada henti. Segala pencapaian ini tak lepas dari bimbingan dan kasih sayang kalian, semoga Allah membalas segala kebaikan dengan limpahan berkah dan kesehatan.
3. Ketiga kakakku tercinta yaitu kak Pipit, kak Abi, kak Ipan dan seluruh keluarga besar yang selalu mendoakan serta memberikan dukungan selama proses penulisan.
4. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan FMIPA Universitas Lampung.
5. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
6. Ibu Yunda Heningtyas, M.Kom. selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.

7. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D. selaku Ketua Prodi S1 Ilmu Komputer FMIPA Universitas Lampung.
8. Bapak Rico Andrian, S.Si., M.Kom. selaku Pembimbing Akademik penulis yang selalu mendukung peningkatan akademik penulis.
9. Bapak Dr. Aristoteles, S.Si., M.Si. selaku Dosen Pembimbing Utama yang telah memberikan arahan, ide, kritik, serta saran dalam menyelesaikan penelitian ini.
10. Bapak Rahman Taufik, S.Pd., M.Kom. selaku Dosen Pembimbing Kedua yang telah memberikan arahan dan bantuan untuk menyelesaikan penelitian ini.
11. Bapak Favorisen R. Lumbanraja S.Si., M.Kom., Ph.D. selaku Pembahas yang telah memberikan masukan serta saran yang bermanfaat untuk perbaikan skripsi ini.
12. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu, motivasi dan pengalaman hidup selama penulis menempuh pendidikan di Jurusan Ilmu Komputer Universitas Lampung.
13. Seluruh Staf dan Karyawan Jurusan Ilmu Komputer yaitu Ibu Ade Nora Maela, Bang Zainuddin, Mas Nofal, dan Mas Syam yang telah membantu segala urusan administrasi penulis di Jurusan Ilmu Komputer.
14. Sahabat seperjuangan semasa kuliah Vidya Sinta Billkis, Cindy Loria, Shafa Auliya, Siti Ayuni, Jihan Haya Mufialdo, dan Fathimah Abiyyi Khairunnisa yang selalu mendukung, menemani, dan berbagi cerita indah selama masa perkuliahan.
15. Teman-teman Himakom dan Pimpinan Tahun 2023 yang telah mengajarkan banyak hal dalam berorganisasi, memberikan banyak pengalaman, mencoba banyak hal baru selama perkuliahan sehingga lebih berkesan.
16. Keluarga Ilmu Komputer Angkatan 2021 yang telah menjadi bagian dari perjalanan ini, terima kasih atas kebersamaan dan dukungan yang diberikan selama perjalanan kuliah di Jurusan Ilmu Komputer Universitas Lampung.
17. Muhammad Donda Fauzaan yang selalu memberikan dukungan, bantuan, dan menemani di setiap langkah penulis baik dalam suka maupun duka dalam menyelesaikan penelitian ini.

18. Terakhir, terima kasih untuk diri sendiri Nabillah Aisyah karena telah mampu berusaha keras dan berjuang sejauh ini. Terima kasih tetap memilih berusaha dan merayakan dirimu sendiri sampai di titik ini, walau sering kali merasa putus asa atas apa yang diusahakan dan belum sesuai dengan keinginan, namun terima kasih tetap menjadi manusia yang selalu mau berusaha dan tidak lelah mencoba. Terima kasih karena memutuskan tidak menyerah sesulit apapun proses penyusunan skripsi ini dan telah menyelesaikannya sebaik dan semaksimal mungkin, karena ini merupakan suatu pencapaian yang patut dirayakan. Tetaplah jadi manusia yang mau berusaha dan tidak lelah untuk mencoba.

Penulis menyadari bahwa skripsi ini masih memiliki banyak kekurangan dan belum sepenuhnya sempurna. Meski demikian, besar harapan penulis agar karya ini dapat memberikan manfaat bagi civitas akademika Universitas Lampung, terkhusus mahasiswa Ilmu Komputer.

Bandar Lampung, 13 Juni 2025

Nabillah Aisyah  
NPM. 2117051036

## DAFTAR ISI

	<b>Halaman</b>
<b>DAFTAR ISI</b> .....	xv
<b>DAFTAR GAMBAR</b> .....	xix
<b>DAFTAR TABEL</b> .....	xxi
<b>DAFTAR KODE PROGRAM</b> .....	xxii
<b>I. PENDAHULUAN</b> .....	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	3
1.3. Batasan Masalah .....	3
1.4. Tujuan Penelitian .....	3
1.5. Manfaat Penelitian .....	4
<b>II. TINJAUAN PUSTAKA</b> .....	<b>5</b>
2.1. Penelitian Terdahulu.....	5
2.2. Deteksi .....	8
2.3 Sperma .....	8
2.4. Morfologi Sperma .....	9
2.5. <i>Teratozoospermia</i> .....	9
2.6. Pengolahan Citra dan Video.....	10
2.6.1 Pengolahan Citra Digital .....	10
2.6.2 Pengolahan Video Digital .....	12
2.7. <i>Resize Image</i> .....	13

2.8	Augmentasi .....	13
2.9	<i>Random Sampling</i> .....	14
2.10	<i>Oversampling</i> .....	14
2.11	<i>Artificial Intelligence</i> .....	15
2.12	<i>Machine Learning</i> .....	15
2.13	<i>Deep Learning</i> .....	16
2.14	<i>Overfitting</i> .....	17
2.15	<i>Underfitting</i> .....	17
2.16	<i>Hyperparameter</i> .....	18
2.16.1	<i>Optimizer</i> .....	18
2.16.2	<i>Learning Rate</i> .....	18
2.16.3	<i>Epoch</i> .....	19
2.16.4	<i>Batch Size</i> .....	19
2.17	<i>Hyperparameter Tuning</i> .....	20
2.18	<i>Transfer Learning</i> .....	20
2.19	CNN ( <i>Convolutional Neural Network</i> ) .....	21
2.19.1	<i>Convolution Layer</i> .....	21
2.19.2	<i>Pooling Layer</i> .....	23
2.19.3	<i>Rectified Linear Unit (ReLU)</i> .....	24
2.19.4	<i>Sigmoid Linear Unit (SiLU)</i> .....	25
2.19.5	<i>Fully-Connected Layer</i> .....	26
2.20	YOLO ( <i>You Only Look Once</i> ) .....	28
2.21	<i>You Only Look Once Versi 8 (YOLOv8)</i> .....	30
2.22	<i>Region Of Interest (RoI)</i> .....	33
2.23	<i>Intersection Over Union (IoU)</i> .....	34
2.24	<i>Confusion Matrix</i> .....	35

2.25. <i>Average Precision (AP)</i> .....	37
2.26. <i>Mean Average Precision (mAP)</i> .....	38
2.27. <i>ByteTrack</i> .....	38
<b>III. METODOLOGI PENELITIAN</b> .....	<b>41</b>
3.2. <i>Alat Pendukung</i> .....	42
3.3. <i>Tahapan Penelitian</i> .....	45
3.3.1 <i>Pengumpulan Dataset</i> .....	47
3.3.2 <i>Pre-processing</i> .....	47
3.3.4 <i>Pembagian Dataset</i> .....	49
3.3.5 <i>Augmentasi Data</i> .....	50
3.3.6 <i>Training Model YOLOv8</i> .....	50
3.3.7 <i>Evaluasi Kinerja Model</i> .....	51
3.3.8 <i>Tracking Bytetrack</i> .....	52
3.3.9 <i>Counting Sperma</i> .....	53
<b>IV. HASIL DAN PEMBAHASAN</b> .....	<b>54</b>
4.1 <i>Pengumpulan Dataset</i> .....	54
4.2 <i>Pre-processing dataset</i> .....	55
4.3 <i>Augmentasi Data</i> .....	64
4.4 <i>Training Model</i> .....	65
4.5 <i>Evaluasi Data Testing</i> .....	79
4.6 <i>Penentuan ROI (Region of Interest)</i> .....	80
4.7 <i>Tracking Objek</i> .....	82
4.7.1 <i>ByteTrack</i> .....	83
4.7.2 <i>Counting Sperma Normal dan Abnormal</i> .....	86
4.7.3 <i>Perhitungan IOU</i> .....	90
<b>V. PENUTUP</b> .....	<b>92</b>

5.1 Kesimpulan.....	92
5.2 Saran.....	93
<b>DAFTAR PUSTAKA.....</b>	<b>94</b>

## DAFTAR GAMBAR

	<b>Halaman</b>
1. Morfologi Sperma.....	8
2. Morfologi <i>Teratozoospermia</i> .....	10
3. Jenis Citra Digital (VISEM Dataset) .....	12
4. Keterkaitan AI, ML, dan DL .....	15
5. Arsitektur CNN .....	21
6. Ilustrasi <i>Feature Map</i> .....	22
7. Ilustrasi <i>Max Pooling</i> .....	23
8. Ilustrasi <i>Average Pooling</i> .....	24
9. Ilustrasi ReLU .....	25
10. Aktivasi Fungsi SiLU dan ReLU .....	26
11. <i>Fully Connected Layer</i> .....	27
12. Arsitektur YOLO.....	28
13. Ilustrasi <i>Bounding Box</i> (VISEM Dataset).....	29
14. Arsitektur YOLOv8.....	30
15. Arsitektur <i>Anchor Free Detection</i> .....	31
16. Arsitektur C2F.....	32
17. Arsitektur SPFF.....	33
18. Ilustrasi RoI.....	34
19. Ilustrasi IoU .....	35
20. Tahapan Penelitian .....	46
21. Pengumpulan Dataset Visem. ....	55
22. Hasil Ekstraksi Frame .....	56
23. <i>Resize Image</i> .....	58
24. Anotasi Sperma dan Non Sperma .....	59
25. Anotasi Normal dan Abnormal .....	60

26. Hasil <i>Cropping</i> Normal dan Abnormal .....	62
27. Grafik Hasil <i>Training Epoch</i> 30 Proporsi 80:10:10 .....	70
28. Grafik Hasil <i>Training Epoch</i> 90 Proporsi 80:10:10 .....	71
29. Grafik <i>Training Epoch</i> 120 Proporsi 80:10:10 .....	72
30. Grafik <i>Training Epoch</i> 30 Proporsi 70:15:15 .....	74
31. Grafik <i>Training Epoch</i> 90 Proporsi 70:15:15 .....	75
32. Grafik <i>Training Epoch</i> 120 Proporsi 70:15:15 .....	76
33. File best.pt sperma dan non-sperma.....	78
34. File best.pt normal dan abnormal.....	78
35. Penerapan ROI Pada Area Deteksi. ....	82
36. Hasil <i>Tracking</i> .....	86
37. Hasil <i>Counting</i> .....	88
38. Perhitungan IOU .....	91

## DAFTAR TABEL

	<b>Halaman</b>
1. Penelitian Terdahulu.....	5
2. Evaluasi Model <i>Confusion Matrix</i> .....	35
3. Jadwal Penelitian .....	41
4. Video Inputan <i>Tracking</i> .....	52
5. Total Objek Sebelum dan Sesudah Augmentasi.....	64
6. Penyesuaian Hyperparameter.....	67
7. Hasil <i>Training</i> 80:10:10 dengan <i>Batch Size</i> 16.....	69
8. Hasil <i>Training</i> 70:15:15 dengan <i>Batch Size</i> 16.....	73
9. Hasil <i>Testing</i> 80:10:10 dengan <i>Batch Size</i> 16 .....	79
10. Hasil <i>Testing</i> 70:15:15 dengan <i>Batch Size</i> 16 .....	80
11. Video Inputan <i>Tracking</i> .....	83
12. Hasil <i>Counting</i> Sperma Normal.....	88
13. Hasil <i>Counting</i> Sperma Abnormal .....	89

## DAFTAR KODE PROGRAM

	<b>Halaman</b>
1. Ekstraksi Video ke Gambar.....	56
2. <i>Resize</i> Gambar .....	58
3. Proses <i>Cropping</i> .....	61
4. Splitting Data .....	63
5. Proses <i>Tracking</i> .....	85
6. <i>Tracking</i> dengan ByteTrack .....	85
7. Proses <i>Counting</i> .....	87
8. Perhitungan IoU .....	90

## I. PENDAHULUAN

### 1.1. Latar Belakang

Pada reproduksi manusia diawali dengan proses fertilisasi, yaitu proses penyatuan sel gamet jantan berupa sperma dan sel gamet betina berupa sel ovum sehingga menghasilkan *zigot diploid* yang kemudian akan berkembang menjadi suatu embrio sebagai awal mula individu baru (Aristoteles *et al.*, 2023). Namun, tidak jarang proses reproduksi manusia berakhir dengan kegagalan. Salah satu penyebab kegagalan fertilisasi ini dipengaruhi oleh kualitas sperma, terutama dari segi morfologi.

Morfologi sperma abnormal atau *teratozoospermia* ditandai dengan berbagai kelainan morfologi, seperti kepala yang tidak berbentuk oval, dua kepala dalam satu sperma, leher yang terlipat, *midpiece* yang terlalu tebal atau terlalu tipis, ekor yang pendek, memiliki lebih dari satu ekor atau ekor yang patah (H. T. T. Nguyen *et al.*, 2022). Kelainan morfologi ini menjadi salah satu indikator penting dalam menentukan kualitas sperma dan potensinya dalam proses fertilisasi. Oleh karena itu, deteksi dan klasifikasi sperma menjadi hal penting dalam analisis laboratorium reproduksi.

Menurut Badan Kesehatan Dunia, *World Health Organization* (WHO) 50 hingga 80 juta pasangan usia subur mengalami infertilitas (Aristoteles *et al.*, 2023). Di Indonesia kasus infertilitas mencapai 10-15% dari seluruh pasangan usia subur berasal dari organ pria (Akbar, 2020). Pria memiliki presentase yang cukup besar dalam menyumbang angka kejadian infertilitas sekitar 25-40% (Annisa & Mahfudah, 2019). Namun, proses analisis morfologi sperma secara

manual menggunakan mikroskop memiliki kekurangan, seperti tingkat subjektivitas yang tinggi dan ketergantungan pada keahlian analisis sehingga rawan terjadi kesalahan. Maka, dalam dunia medis tantangan terbesar adalah mendeteksi dan mengklasifikasikan sperma dengan akurat dan efisien karena adanya variasi morfologi sperma yang beragam.

Seiring berkembangnya teknologi, pendekatan berbasis *deep learning* mulai banyak digunakan dalam klasifikasi objek berdasarkan morfologinya. Salah satu model yang digunakan yaitu YOLO (*You Only Look Once*). Terdapat penelitian terdahulu yang membahas mengenai evaluasi morfologi sperma yang pernah diteliti oleh (Sato *et al.*, 2022) yang membahas terkait identifikasi sperma dengan kualitas terbaik menggunakan YOLOv3. Identifikasi ini diteliti dengan dataset berupa video yang diambil di bawah mikroskop sebanyak 51 objek. Identifikasi yang dilakukan memiliki 2 kelas yaitu sperma normal dan abnormal, dimana sperma abnormal memiliki sensitivitas dan nilai prediktif positif (PPV) masing-masing sebesar 0,881 dan 0,853. Sedangkan sperma normal memiliki sensitivitas dan PPV sebesar 0,794 dan 0,689. Hal ini menunjukkan bahwa metode deteksi objek berbasis *deep learning* berpotensi besar dalam klasifikasi sperma.

Penelitian ini berfokus pada klasifikasi *teratozoospermia* berdasarkan morfologi sperma dengan menggunakan metode deep learning, khususnya model YOLOv8. Dengan adanya tantangan dalam proses klasifikasi sperma yang melibatkan deteksi objek, penelitian ini bertujuan untuk mengatasi kesulitan dalam membedakan antara sperma normal dan abnormal yang menjadi tantangan dalam analisis laboratorium. Dalam hal ini, proses deteksi sperma dilakukan dengan mengidentifikasi keberadaan sperma berdasarkan lokasi *bounding box* pada gambar mikroskopis dan mengklasifikasikan antara sperma yang memiliki morfologi normal dan abnormal (*teratozoospermia*). Model YOLOv8 digunakan untuk meningkatkan akurasi dan efisiensi dalam klasifikasi morfologi sperma karena kemampuannya yang unggul dalam mendeteksi dan mengklasifikasikan objek dengan akurasi tinggi secara *real-*

*time*. Dengan pendekatan ini, diharapkan dapat memperbaiki proses klasifikasi sperma dan memberikan solusi yang lebih efektif dalam penanganan *teratozoospermia*.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan, maka terdapat beberapa rumusan masalah pada penelitian ini, yaitu :

1. Bagaimana implementasi YOLOv8 dalam mendeteksi *teratozoospermia* berdasarkan morfologi sperma manusia?
2. Bagaimana kinerja model YOLOv8 dalam mendeteksi *teratozoospermia* pada morfologi sperma manusia?

## 1.3. Batasan Masalah

Berdasarkan latar belakang yang telah dijabarkan, maka terdapat beberapa batasan masalah pada penelitian ini, yaitu:

1. Mendeteksi *teratozoospermia* pada morfologi sperma dengan menggunakan model YOLOv8.
2. Deteksi yang dilakukan berdasarkan 2 kelas morfologi sperma yaitu normal dan abnormal.
3. Data citra yang digunakan bersumber dari VISEM sperma manusia dalam bentuk video berekstensi AVI (*Audio Video Interleave*).
4. Anotasi data dilakukan secara manual di Roboflow menggunakan *bounding box* untuk menandai sperma, non-sperma, normal dan abnormal.

## 1.4. Tujuan Penelitian

Berdasarkan latar belakang yang telah dijabarkan, maka terdapat beberapa tujuan pada penelitian ini, yaitu:

1. Mengimplementasikan model YOLOv8 dalam mendeteksi morfologi sperma dalam klasifikasi *teratozoospermia*.
2. Mengevaluasi tingkat keakuratan model YOLOv8 dalam membedakan antara sperma normal dan abnormal berdasarkan morfologi sperma.

### **1.5. Manfaat Penelitian**

1. Mengetahui hasil evaluasi dari deteksi morfologi sperma manusia menggunakan YOLOv8.
2. Mengetahui tingkat keakuratan model YOLOv8 dalam mendeteksi *teratozoospermia*.
3. Mengetahui hasil distribusi klasifikasi sperma normal dan abnormal.
4. Mendukung pengembangan model di masa depan untuk pro analisis morfologi sperma normal dan abnormal.

## II. TINJAUAN PUSTAKA

### 2.1. Penelitian Terdahulu

Penelitian ini memiliki keterkaitan dengan penelitian sebelumnya, sehingga terdapat hubungan antara persamaan dan perbedaan pada objek yang diteliti. Rangkuman dari penelitian terdahulu dapat dilihat pada Tabel 1.

Tabel 1. Penelitian Terdahulu

Penelitian	Data	Metode	Hasil
A new deep-learning model using YOLOv3 to support sperm selection during intracytoplasmic sperm injection procedure (Sato <i>et al.</i> , 2022)	Normal : 928 Abnormal : 2.248 Unclassifiable : 1.095 Vacuole : 354	<i>Grad-CAM (Gradient Neighted Class Activation)</i>  <i>Pre-processing RGB frame extraction, cropping, Resizing, dan Augmentation</i>	Sperma Normal – <i>Sensitivity</i> : 0.794 <i>Positive Predictive Value (PPV)</i> : 0.689  Sperma Abnormal <i>Sensitivity</i> : 0.881 <i>Positive Predictive Value (PPV)</i> : 0.853
		Model dan Algoritma : YOLOv3 dan SORT	
		Evaluasi : <i>Confusion Matrix</i>	

Tabel 1. Lanjutan

<b>Penelitian</b>	<b>Data</b>	<b>Metode</b>	<b>Hasil</b>
Study on Sperm-Cell Detection Using YOLOv5 Architecture with Labaled Dataset (Dobrovolny <i>et al.</i> , 2023)	Training : 368 images (3500 labels) Validation : 14 images	<i>One stage object detection</i>  <i>Pre-processing : Extraction video to image, resize, dan annotation</i>  Model : YOLOv5	YOLOv5 ( <i>small</i> ) <i>Precision : 61.6</i> <i>Recall : 64.9</i> mAP : 64.6  YOLOv5 ( <i>medium</i> ) <i>Precision : 71.7</i> <i>Recall : 57.8</i> mAP : 66.4  Evaluasi : -YOLOv5 ( <i>large</i> ) <i>Confusion Matrix dan mAP. Precision : 88.6</i> <i>Recall : 52.6</i> mAP : 72.1  -YOLOv5 ( <i>xtra</i> ) <i>Precision : 64.6</i> <i>Recall : 71.9</i> mAP : 68.6
Identification of Human Sperm based on Morphology Using the You Only Look Once Version 4 Algorithm (Aristoteles <i>et al.</i> , 2022)	Image : 1330 Sperm : 105.465 Non sperm : 22.425	<i>Cutmix, Mosaic, BoF and Bos</i>  <i>Preprocessing: Resize, Anotation</i>  Model: YOLOV4  Evaluasi: mAP dan <i>Confusion Matrix</i>	<i>Average Precision</i> Sperm : 90.31% Non-sperm : 68.19%  mAP : 79.58%

Tabel 1. Lanjutan

<b>Penelitian</b>	<b>Data</b>	<b>Metode</b>	<b>Hasil</b>
Sperm Abnormality Detection Using Sequential Deep Neural Network (Shahzad <i>et al.</i> , 2023)	Normal : 1086 (Acrosome), 1122 (Head), 1301 (Vacuole), 1471 (Tail/Neck) Abnormal : 454 (Acrosome), 418 (Head), 293 (Vacuole), 601 (Tail/Neck)	<i>Feedforward Neural Network (FNN)  Preprocessing: RGB Frame Extraction, Filtering Noise, Segmentation, Statical Measurement  Model: Sequential Deep Neural Network  Evaluasi : Confusion Matrix</i>	<i>Accuracy Akrosom : 89% Kepala : 90% Vakuola : 92%</i>
Transparent Tracking of Spermatozoa with YOLOv8 (B.-T. Nguyen <i>et al.</i> , 2023)	20 video berdurasi 30 detik	<i>Deep Neural Network  One Stage Object Detection  Preprocessing: Resize, Normalisasi  Model: YOLOv8  Evaluasi : Confusion Matrix &amp; mAP</i>	<i>Precision : 50% Recall : 62.8% mAP50 : 50.6% mAP50-90 : 19.1%</i>

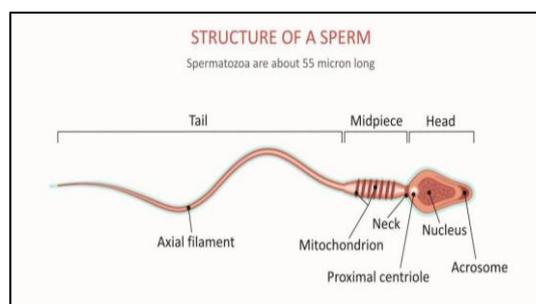
## 2.2. Deteksi

Deteksi adalah cabang dari *computer vision* yang bertujuan untuk mengenali dan menentukan posisi objek dalam gambar atau video. Beda halnya dengan klasifikasi gambar yang hanya menentukan jenis objek, deteksi objek tidak hanya mengklasifikasikan tetapi juga mengidentifikasi letak objek tersebut dengan menampilkan *bounding box* di sekitarnya (Tasnim & Qi, 2023). Dengan kata lain, proses ini tidak hanya berfokus pada pengenalan jenis objek dalam gambar, tetapi juga mencakup penentuan posisi objek tersebut secara akurat di dalam citra.

Teknologi ini banyak dimanfaatkan dalam berbagai bidang seperti kendaraan otonom, pengawasan, *augmented reality*, dan medis. Beberapa metode populer dalam deteksi objek adalah R-CNN, Fast R-CNN, Faster R-CNN, dan YOLO yang menggunakan jaringan saraf konvolusional (CNN) untuk mengklasifikasikan objek dan memprediksi posisinya secara akurat.

## 2.3 Sperma

Sperma adalah sel reproduksi jantan atau gamet yang berperan penting dalam proses pembuahan. Sperma memiliki tujuan untuk mentransfer DNA dari jantan menuju sel telur (ovum) (Prakash *et al.*, 2014). Sperma diproduksi di dalam testis melalui proses yang disebut *spermatogenesis*. Sel sperma memiliki 3 bagian utama yaitu kepala, bagian tengah (*midpiece*), dan ekor. Struktur pada sperma dapat dilihat pada Gambar 1.



Gambar 1. Morfologi Sperma (<https://doktersehat.com/>).

## 2.4. Morfologi Sperma

Morfologi adalah suatu cabang ilmu biologi yang mempelajari bentuk, struktur, dan ukuran pada organisme atau bagian-bagian makhluk hidup. Morfologi sperma adalah studi mengenai bentuk dan struktur sperma (Yuzkat *et al.*, 2021). Morfologi sperma memiliki kontribusi yang besar terhadap infertilitas pria karena menjadi indikator pertama dalam membuahi sel telur.

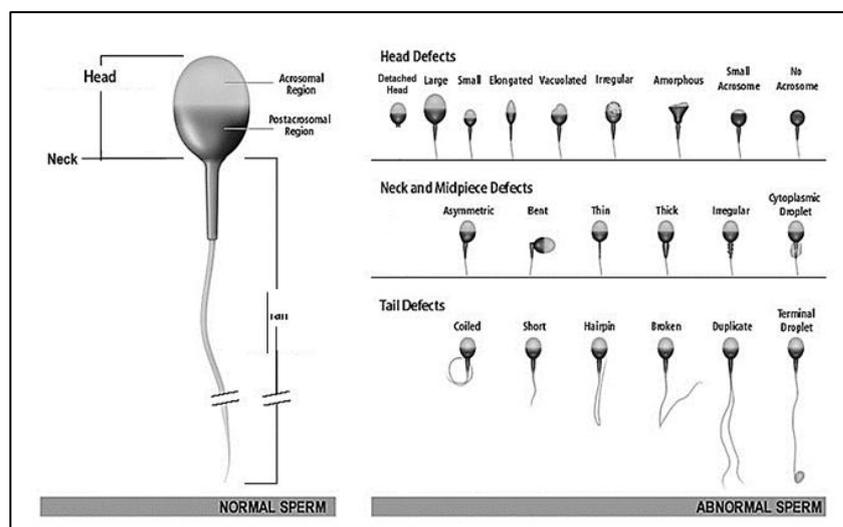
Secara umum, *spermatozoa* yang normal memiliki 3 bagian utama yaitu kepala, leher, dan ekor. Morfologi sperma yang normal ditandai dengan kepala berbentuk oval dimana akrosom menutupi sepertiga bagian kepala. Panjang kepala sperma berkisar antara 3-5 mikron dengan lebar kepala yang sekitar setengah hingga sepertiga dari panjangnya. Bagian tengah sperma (leher) memiliki lebar lebih kecil dari kepala dengan panjang dua kali lipat dari kepala dan sejajar dengan sumbu kepala. Ekor sperma terlihat jelas dan memiliki panjang sekitar sembilan kali lipat dari panjang kepala (Aliyah *et al.*, 2022).

## 2.5. *Teratozoospermia*

*Teratozoospermia* adalah adanya sperma atau spermatozoa dalam bentuk abnormal dan seringkali dikaitkan dengan infertilitas. Kondisi ini terjadi ketika lebih dari 30% sperma memiliki morfologi yang abnormal dalam proses konsepsi. Penyebab *teratozoospermia* dapat disebabkan dari faktor genetik dan lingkungan serta kondisi fisik seperti varikokel (Atmoko *et al.*, 2024). Sperma dengan morfologi abnormal memiliki peluang lebih kecil untuk berhasil membuahi sel telur secara alami, sehingga mengurangi kemungkinan terjadinya kehamilan.

Morfologi sperma abnormal dapat diidentifikasi melalui beberapa ciri fisik yang menyimpang dari bentuk normal. Sperma normal umumnya memiliki kepala berbentuk oval, leher (*midpiece*) yang terhubung, dan ekor yang

panjang. Namun, sperma abnormal terdapat beberapa kelainan pada bentuk kepala, leher (*midpiece*), dan ekor. Pada bagian kepala, sperma abnormal memiliki bentuk yang tidak oval, ukuran yang lebih besar dari sperma normal, dan memiliki dua kepala atau ekor dalam satu sperma. Pada bagian leher, sperma abnormal memiliki leher yang terlalu tebal atau tipis, leher yang bengkok. Sedangkan pada ekor memiliki dua ekor, ekor yang patah, dan ekor yang terlalu pendek (Ilhan & Serbes, 2022). Perbandingan morfologi sperma normal dan abnormal (*teratozoospermia*) dapat dilihat pada Gambar 2.



Gambar 2. Morfologi *Teratozoospermia* (Ilhan & Serbes, 2022).

## 2.6. Pengolahan Citra dan Video

Pada sub-bab ini adalah penjelasan terkait pengolahan pada citra *digital* dan video digital yang dijelaskan sebagai berikut.

### 2.6.1 Pengolahan Citra Digital

Pengolahan citra digital (*Digital Image Processing*) adalah cabang ilmu yang mempelajari cara pembentukan, pemrosesan, dan analisis citra untuk menghasilkan informasi yang dapat dimengerti oleh manusia

(Ratna, 2020). Citra yang diproses dapat berupa gambar diam (foto) atau gambar yang bergerak (video). Sedangkan pengolahan digital merupakan pemrosesan yang citra atau gambar yang dilakukan secara digital atau bantuan komputer. Pengolahan citra digital mencakup berbagai teknik, seperti peningkatan kualitas gambar, pengenalan pola, segmentasi objek, hingga kompresi data.

Citra digital dapat dipresentasikan sebagai matriks 2 dimensi  $f(x,y)$  yang terdiri dari  $M$  kolom dan  $N$  baris, sedangkan titik pertemuan antara kolom dan baris disebut piksel (*pixel = picture element*), yaitu elemen terkecil pada suatu citra. Citra  $f(x,y)$  dapat dinyatakan dalam bentuk fungsi matematis yang ditunjukkan pada Persamaan 1.

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

$$0 \leq x \leq M - 1$$

$$0 \leq y \leq N - 1$$

$$0 \leq f(x,y) \leq G - 1$$

Keterangan :

$M$  = jumlah piksel baris (*row*) pada array citra

$N$  = jumlah piksel kolom (*column*) pada array citra

$G$  = nilai skala keabuan (*graylevel*)

Pada pengolahan citra digital dapat dibagi menjadi 3 jenis utama, yaitu.

a. *Color Image* atau RGB (*Red, Green, Blue*)

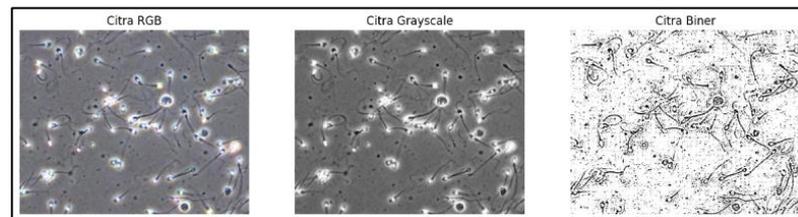
Jenis *color image* setiap piksel memiliki warna tertentu, yaitu merah (*Red*), hijau (*Green*), dan biru (*Blue*). Apabila setiap warna memiliki range 0-255, maka totalnya yaitu  $255^3 = 16.581.375$  dengan variasi warna yang berbeda pada setiap gambar.

b. *Black and White*

Jenis *black and white* (*grayscale*) setiap piksel memiliki warna gradasi yaitu putih hingga hitam. Citra ini memiliki lapisan keabuan dengan nilai piksel berkisar hitam (0) hingga putih (255). Setiap piksel pada rentang tersebut direpresentasikan oleh 8 bit atau 1 *byte*.

c. *Binary Image*

Jenis ini pada setiap pikselnya hanya terdiri warna hitam dan putih sehingga hanya memerlukan 1 bit per piksel. Nilai intensitas direpresentasikan melalui dua warna yaitu nilai piksel 0 digunakan untuk warna putih dan nilai piksel 1 digunakan untuk warna hitam. Hasil pengolahan citra digital pada data VISEM dapat dilihat pada Gambar 3.



Gambar 3. Jenis Citra Digital (VISEM Dataset).

### 2.6.2 Pengolahan Video Digital

Pengolahan video digital adalah teknologi yang digunakan untuk menganalisis dan memanipulasi data visual dari video guna mendapatkan informasi yang bermanfaat (Ratna, 2020). Proses ini mencakup berbagai teknik, seperti deteksi objek, pelacakan gerakan, pengenalan pola, dan peningkatan kualitas gambar. Dalam banyak aplikasi, pengolahan *video digital* digunakan untuk memahami lingkungan sekitar secara lebih mendalam dengan menganalisis gambar bergerak secara berkelanjutan. Misalnya, dalam sistem pengenalan atau deteksi, pengolahan video digital memberikan konteks tambahan yang lebih akurat dibandingkan hanya menggunakan data statis.

## 2.7. *Resize Image*

*Resize image* adalah suatu proses pengolahan gambar (citra) yang digunakan untuk mengubah ukuran pada gambar agar sesuai dengan ukuran yang diinginkan (Rokhmah *et al.*, 2020). Proses ini memungkinkan pengguna untuk menyesuaikan lebar, tinggi, diagonal, rasio aspek gambar dengan tujuan mengecilkan ukuran file, mengoptimalkan resolusi, serta menyesuaikan proporsi gambar. Hal ini dilakukan untuk mencapai efisiensi dalam penggunaan ruang penyimpanan tanpa mempengaruhi kualitas visual gambar.

Pada YOLOv8 tahap *resize image* adalah langkah penting dalam proses pra-pemrosesan. Proses ini dilakukan untuk memastikan gambar yang diinput sesuai dengan ukuran yang diharapkan oleh jaringan neural, mempercepat pemrosesan, dan mengoptimalkan akurasi (Muhlashin & Stefanie, 2023). YOLOv8 membutuhkan gambar input dengan ukuran tertentu misalnya 640 x 640 piksel agar model bekerja dengan baik.

## 2.8 Augmentasi

Augmentasi data merupakan sebuah proses untuk menghasilkan contoh data baru yang digunakan dalam pelatihan model. Teknik ini dilakukan dengan memodifikasi data asli sehingga dapat menciptakan variasi data yang lebih beragam. Augmentasi dilakukan ketika jumlah data terbatas, karena memungkinkan model untuk belajar dari data yang lebih bervariasi tanpa perlu melakukan pengumpulan data tambahan. Beberapa peneliti telah menerapkan teknik augmentasi data untuk meningkatkan keragaman dalam dataset, sehingga model yang dilatih menjadi lebih robust dan mampu melakukan generalisasi dengan lebih baik pada data baru (Hassan *et al.*, 2022). Teknik augmentasi yang umum dilakukan meliputi rotasi gambar, *flipping* (pembalikan), *zoom*, perubahan kecerahan, pemotongan (*cropping*), serta penggabungan beberapa gambar seperti teknik Mosaic dan CutMix.

## 2.9. *Random Sampling*

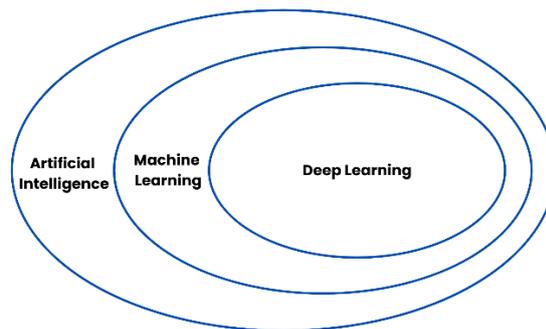
*Random sampling* adalah metode pengambilan sampel di mana setiap elemen dalam populasi memiliki peluang yang sama untuk dipilih. Teknik ini bertujuan untuk menghasilkan sampel yang representatif, sehingga hasil penelitian dapat digeneralisasi ke seluruh populasi. Dalam *random sampling*, pemilihan elemen dilakukan secara acak tanpa memperhitungkan karakteristik spesifik elemen tersebut. Metode ini sering digunakan dalam berbagai bidang, termasuk survei, eksperimen ilmiah, dan penelitian sosial, karena dapat meminimalkan bias dan memberikan gambaran yang objektif tentang populasi yang diteliti (Muhlashin & Stefanie, 2023). Selain itu, *random sampling* juga membantu memastikan bahwa hasil penelitian dapat diandalkan dan valid, karena setiap elemen dalam populasi memiliki peluang yang sama untuk dimasukkan dalam sampel.

## 2.10. *Oversampling*

*Oversampling* atau *Random Oversampling* (ROS) merupakan salah satu teknik penyeimbangan data yang digunakan ketika terjadi ketidakseimbangan antara jumlah data pada kelas mayoritas dan minoritas atau *class imbalance*. Teknik ini dilakukan dengan cara menduplikasi secara acak sampel-sampel dari kelas minoritas, kemudian menambahkannya ke dalam dataset pelatihan. Dengan demikian, ukuran dataset kelas minoritas menjadi bertambah hingga distribusi antar kelas menjadi lebih seimbang (Torres *et al.*, 2022). Pendekatan ini bertujuan untuk mencegah model menjadi bias terhadap kelas mayoritas dan meningkatkan kemampuan model dalam mengenali kelas minoritas, terutama dalam kasus klasifikasi dengan data yang tidak seimbang.

### 2.11. *Artificial Intelligence*

*Artificial Intelligence* (AI) merupakan bidang ilmu komputer yang sangat luas, bidang ini berhubungan dengan meniru fungsi kognitif manusia untuk menyelesaikan masalah di dunia nyata dengan tujuan membangun sistem yang dapat belajar dan berpikir seperti manusia (Holzinger *et al.*, 2019). *Artificial Intelligence* (AI) berfokus pada pengembangan teknologi yang memungkinkan komputer untuk memahami, belajar, membuat keputusan, dan memecahkan masalah secara mandiri, seperti yang dilakukan manusia dalam kehidupan sehari-hari. *Artificial Intelligence* memiliki bagian integral yaitu *machine learning* (ML), sementara itu *deep learning* (DL) adalah subkelas dari ML dalam teknologi AI. Keterkaitan AI dengan ML dan DL dapat dilihat pada Gambar 4.



Gambar 4. Keterkaitan AI, ML, dan DL (Muhlashin & Stefanie, 2023).

### 2.12. *Machine Learning*

*Machine learning* (ML) adalah algoritma dan model statistik yang digunakan oleh sistem komputer untuk melakukan tugas tertentu tanpa diprogram secara eksplisit (Mahesh, 2020). *Machine Learning* menunjukkan kemampuan sistem untuk belajar dari data latih yang spesifik terhadap masalah sehingga dapat mengotomatisasi proses pembuatan model dan menyelesaikan tugas-tugas yang diberikan. Dengan menggunakan data tersebut, sistem akan belajar mengenai pola dan membuat prediksi atau keputusan tanpa diprogram secara eksplisit untuk setiap skenario.

Pada *machine learning* terdapat 3 tipe utama yang umum digunakan, yaitu (Janiesch *et al.*, 2021):

**a. *Supervised Learning***

Tipe ini setiap model yang dilatih menggunakan data yang sudah diberi label. Dengan data tersebut, model akan belajar memetakan input ke output dengan benar sehingga bisa membuat prediksi yang akurat ketika diberikan data baru.

**b. *Unsupervised Learning***

Tipe ini data yang digunakan untuk melatih model tidak diberi label. Sehingga, model harus menemukan pola, struktur, atau pengelompokkan yang mendasari data tanpa ada panduan output yang benar.

**c. *Reinforcement Learning***

Tipe ini model tidak diberi instruksi langsung tentang langkah yang harus diambil, sehingga perlu menyediakan input dan output dalam memberikan tindakan yang diperbolehkan serta batasan terkait hasil tindakan tersebut.

### **2.13. *Deep Learning***

*Deep learning* merupakan salah satu metode pembelajaran mesin berbasis data, yang sangat efektif dalam menyelesaikan berbagai masalah komputasi kompleks (Bhatt *et al.*, 2021). Teknik ini menggunakan jaringan saraf tiruan (*neural networks*) dengan banyak lapisan untuk memproses data dan belajar dari pola yang ada. Dalam beberapa tahun terakhir, *deep learning* telah menunjukkan keberhasilannya dalam berbagai aplikasi, termasuk pengenalan gambar, pemrosesan bahasa alami, dan prediksi data. Dalam konteks yang lebih spesifik, *deep learning* dapat digunakan untuk menerapkan teknik dasar seperti pengenalan pola, prediksi kondisi optimal, dan pengendalian otomatis. Selain itu, efektifitas *deep learning* dalam konteks tersebut perlu diverifikasi melalui pengujian dan evaluasi untuk memastikan keandalannya dalam aplikasi dunia nyata.

#### 2.14. *Overfitting*

*Overfitting* adalah salah satu masalah dalam *supervised learning* yang terjadi ketika model mempelajari data pelatihan dengan terlalu baik sehingga sulit digeneralisasi ke data baru. *Overfitting* disebabkan oleh keterbatasan data pelatihan, seperti ukuran yang kecil atau adanya *noise*, serta kompleksitas algoritma dengan jumlah parameter yang besar (Ying, 2019). Kondisi ini menghasilkan performa model yang baik pada data pelatihan tetapi buruk pada data validasi atau pengujian. Untuk mengatasi *overfitting* digunakan teknik seperti regulasi untuk mengurangi kompleksitas model, dropout untuk mencegah model bergantung pada fitur tertentu, augmentasi data untuk meningkatkan variasi dalam data pelatihan, *early stopping* untuk menghentikan pelatihan sebelum model mulai terlalu menyesuaikan data pelatihan dan menggunakan dataset yang lebih besar serta representatif guna meningkatkan kemampuan generalisasi model.

#### 2.15. *Underfitting*

*Underfitting* adalah kondisi sebuah model *machine learning* gagal mempelajari pola yang mendasari data secara memadai karena kurangnya kapasitas model atau pelatihan yang memadai untuk memahami hubungan yang sebenarnya berdasarkan penghafalan atau tidak (Czajkowski & Kretowski, 2019). *Underfitting* berlawanan dengan *overfitting*, dimana model yang *underfitting* terlalu sedikit memperhatikan data pelatihan, sehingga menghasilkan performa yang buruk pada data pelatihan maupun data baru. *Underfitting* terjadi ketika model terlalu sederhana atau tidak dilatih cukup lama, maka tidak mampu menangkap kompleksitas data. Untuk mengatasi *underfitting* yaitu meningkatkan kompleksitas model dengan menambahkan parameter atau lapisan, melatih model lebih lama dengan jumlah *epoch* yang cukup, memilih fitur-fitur yang relevan dalam data, dan memastikan data pelatihan cukup banyak serta representatif.

## 2.16. *Hyperparameter*

*Hyperparameter* adalah parameter yang digunakan untuk mengoptimalkan hasil yang diharapkan dengan tujuan untuk meningkatkan performa model yang diimplementasikan. *Hyperparameter* ditentukan sebelum melakukan proses pelatihan yang akan membantu model mencapai akurasi dan efisiensi model (Prayitno *et al.*, 2024). Berikut beberapa *hyperparameter* yang mempengaruhi kinerja model yang terdiri dari *optimizer*, *learning rate*, *epoch*, dan *batch size*.

### 2.16.1. *Optimizer*

*Optimizer* merupakan algoritma yang digunakan untuk menemukan dan mengoptimalkan nilai bobot dan bias dalam jaringan saraf yang dilakukan pada proses pelatihan model *machine learning* dan *deep learning*. Tujuan utamanya adalah untuk meminimalisir kesalahan sehingga model dapat menghasilkan prediksi yang lebih akurat pada data baru. Beberapa jenis *optimizer* yang umum digunakan dalam *deep learning* yaitu *Stochastic Gradient Descent* (SGD), *Adam*, dan *RMSprop*. Pemilihan *optimizer* yang tepat dapat mempercepat proses konvergensi dan meningkatkan akurasi model (Julianto *et al.*, 2022).

### 2.16.2. *Learning Rate*

*Learning rate* merupakan salah satu *hyperparameter* penting dalam proses pelatihan model *deep learning*. Fungsinya untuk mengatur seberapa besar perubahan bobot yang dilakukan model setiap proses *training* berlangsung. Jika nilai *learning rate* terlalu kecil, maka proses pelatihan bisa menjadi sangat lambat karena model hanya melakukan sedikit penyesuaian di setiap iterasinya. Sebaliknya, jika *learning rate* terlalu besar, maka model bisa tidak stabil karena perubahan bobot yang terlalu drastis. Oleh karena itu, pemilihan

*learning rate* yang tepat sangat penting agar proses pelatihan berjalan optimal dan model mampu menghasilkan performa yang baik (Prayitno *et al.*, 2024).

### 2.16.3. *Epoch*

Jumlah *epoch* adalah salah satu *hyperparameter* penting dalam proses pelatihan model, yang menunjukkan seberapa sering seluruh data pelatihan digunakan untuk memperbarui bobot model. Semakin banyak *epoch*, maka semakin sering model belajar dari data yang sama. Namun, jumlah *epoch* yang digunakan harus disesuaikan dengan kompleksitas model dan ukuran dataset. Jika *epoch* terlalu sedikit, maka model dapat kekurangan waktu belajar dan hasilnya kurang optimal. Sebaliknya, jika terlalu banyak maka dapat terjadi *overfitting* atau terlalu menghafal data pelatihan dan tidak berkerja dengan baik pada data baru. Oleh karena itu, pemilihan jumlah *epoch* yang tepat sangat penting agar model dapat belajar secara efektif dan tetap mampu melakukan generalisasi (Prayitno *et al.*, 2024).

### 2.16.4. *Batch Size*

*Batch size* adalah jumlah sampel data yang digunakan dalam satu kali iterasi selama proses pelatihan model. Dalam pelatihan, dataset yang besar dibagi menjadi *batch* agar proses pembelajaran menjadi lebih efisien dan tidak membebani memori. Dengan kata lain, model tidak memproses data sekaligus melainkan mempelajari sedikit demi sedikit melalui setiap *batch*. Pemilihan ukuran *batch* yang tepat dapat mempengaruhi kecepatan, efisiensi, dan hasil akhir dari pelatihan model (Prayitno *et al.*, 2024).

### 2.17. *Hyperparameter Tuning*

*Hyperparameter tuning* merupakan langkah penting dalam proses pelatihan model *machine learning*. *Hyperparameter tuning* bertujuan untuk meningkatkan performa model dari segi *accuracy*, *precision*, maupun *recall* (Saputra *et al.*, 2025). Teknik ini seringkali digunakan karena dapat membantu menemukan kombinasi *hyperparameter* yang optimal tanpa harus mencoba satu per satu secara manual, hal tersebut dapat menghemat waktu dan tenaga dalam proses pelatihan model.

Dalam *computer vision*, proses tuning menjadi penting karena model harus mampu mendeteksi objek secara otomatis dengan cepat dan akurat, seperti penggunaan CNN untuk sistem deteksi objek. Model seperti YOLOv8 dan YOLO-NAS dapat menunjukkan peningkatan performa yang signifikan ketika *hyperparameter* yang digunakan sudah diatur secara optimal. Selain itu, *library* Tune dan Ray Tune dimanfaatkan untuk melakukan tuning agar menghasilkan kombinasi *hyperparameter* terbaik sehingga pengembangan model dapat dilakukan dengan optimal dan hemat waktu.

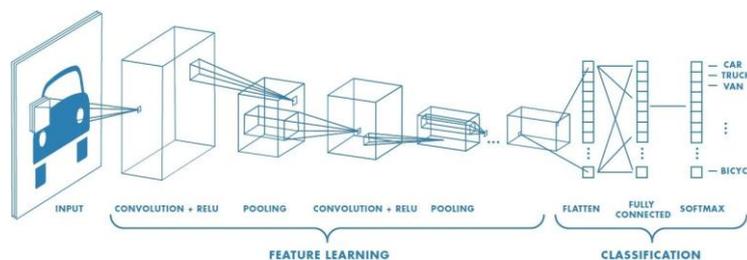
### 2.18. *Transfer Learning*

*Transfer learning* merupakan teknik dalam *machine learning* yang digunakan untuk memanfaatkan kembali model yang telah dilatih pada suatu tugas kemudian disesuaikan untuk menyelesaikan tugas lain yang serupa. Teknik ini membantu proses pelatihan model dari awal karena membutuhkan banyak data dan waktu yang cukup lama. Dengan memanfaatkan *transfer learning*, model yang sudah dilatih menggunakan dataset yang besar dan kompleks dapat digunakan sebagai titik awal untuk menyelesaikan tugas baru yang sejenis. Hal ini membuat proses pelatihan menjadi lebih efisien karena model tidak perlu belajar dari awal sehingga waktu pelatihan menjadi lebih singkat dan kebutuhan data yang lebih sedikit (Eka Okta Putra *et al.*, 2024). *Transfer learning* banyak digunakan dalam berbagai bidang, terutama *computer vision*

dan *natural language processing* karena dapat mempercepat pengembangan model serta mempertahankan performa yang baik.

## 2.19. CNN (*Convolutional Neural Network*)

CNN (*Convolutional Neural Network*) adalah metode pembelajaran mendalam (*deep learning*) yang dirancang khusus untuk memproses data berupa gambar. CNN bekerja dengan memberikan bobot dan bias terhadap berbagai objek dalam gambar, sehingga dapat membedakan satu objek dengan objek lainnya (Bhatt *et al.*, 2021). CNN memiliki performa yang paling efektif dalam proses klasifikasi citra, pengenalan, segmentasi, dan pengambilan gambar kembali (*image retrieval*). Namun, CNN memiliki kelemahan yaitu tidak secara eksplisit memperhitungkan posisi dan orientasi spesifik suatu objek dalam gambar apabila diputar atau dipindahkan ke posisi yang berbeda. Arsitektur pada CNN dapat dilihat pada Gambar 5.



Gambar 5. Arsitektur CNN (<https://medium.com/>).

### 2.19.1 *Convolution Layer*

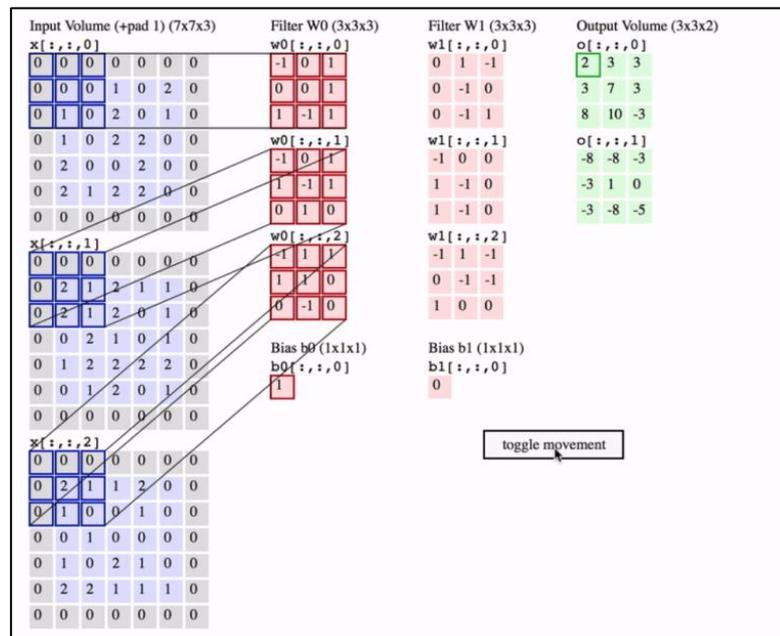
*Convolution layer* adalah bagian dari tahapan dalam *convolutional neural network* (CNN) untuk mendeteksi fitur-fitur penting dalam gambar. Fitur ini dapat berupa garis, tepi, atau pola, yang membentuk objek dalam gambar. Tujuannya dilakukan konvolusi pada gambar adalah untuk mengambil atau mengekstraksi berbagai fitur dari gambar input (Azmi *et al.*, 2023). Dengan itu, model dapat

mengenali dan memahami berbagai elemen visual dalam gambar, yang penting untuk tugas-tugas seperti pengenalan objek, segmentasi gambar, dan klasifikasi.

Lapisan konvolusi terdiri dari neuron-neuron yang disusun membentuk sebuah filter dengan ukuran panjang dan tinggi (*pixel*). Proses konvolusi menggunakan *kernel* sebagai filter yang dioperasikan di seluruh gambar input. Filter ini digeser dengan langkah-langkah tertentu yang disebut *stride*. Pada setiap posisi, filter ini menghitung *dot product* atau gabungan antara nilai-nilai dalam gambar (*matriks*) dan *kernel* sehingga menghasilkan matriks baru yang dikenal sebagai *feature map* atau *activation map* (Purwono *et al.*, 2022). Persamaan dan ilustrasi dari *feature map* atau *activation map* dinyatakan pada Gambar 6 dan Persamaan 2 dan 3.

$$\text{Activation map} = \text{Input} * \text{Filter} \quad (2)$$

$$= \sum_{y=0}^{\text{columns}} \left( \sum_{x=0}^{\text{columns}} \text{input}(x - p, y - q) \text{Filter}(x, y) \right) \quad (3)$$



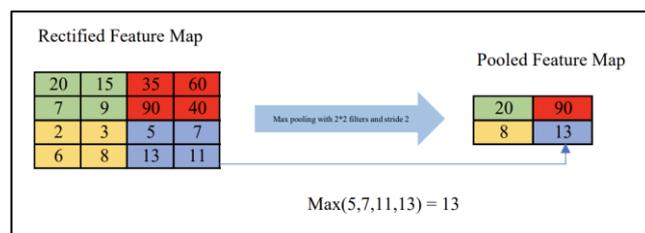
Gambar 6. Ilustrasi *Feature Map* (Yuda *et al.*, 2020).

### 2.19.2 Pooling Layer

*Pooling layer* adalah tahapan yang digunakan untuk mengurangi dimensi atau ukuran dari *feature map* yang digunakan sebagai nilai masukan yang berfungsi untuk menurunkan sumber daya komputasi dan *overfitting*. Layer ini berada secara terurut di lapisan kedua setelah *convolution layer*. Secara umum, *pooling layer* menggunakan filter berukuran 2x2 yang bergerak dua langkah pada setiap bagian input. Hasil dari proses konvolusi ini sering diproses menggunakan dua algoritma, yaitu *max pooling* dan *average pooling*. Pada *max pooling*, nilai maksimum dari setiap area yang dilalui filter akan diambil, sedangkan *average pooling* mengambil nilai rata-rata dari area yang dipindai oleh filter. Berikut merupakan persamaan dan ilustrasi dari *max pooling* dan *average pooling* (Zafar *et al.*, 2022). Fungsi matematis *max pooling* dan *average pooling* dinyatakan pada Persamaan 4 dan 5.

#### a. *max pooling*

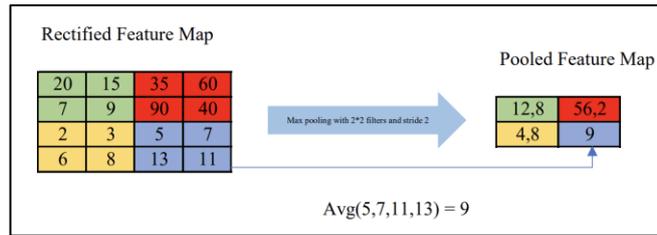
$$f_{max}(X) = \max_i x_i \quad (4)$$



Gambar 7. Ilustrasi *Max Pooling*.

#### b. *average pooling*

$$f_{ave}(X) = \frac{1}{N} \sum_i^N x_i = 1x_i \quad (5)$$



Gambar 8. Ilustrasi *Average Pooling*.

### 2.19.3 Rectified Linear Unit (ReLU)

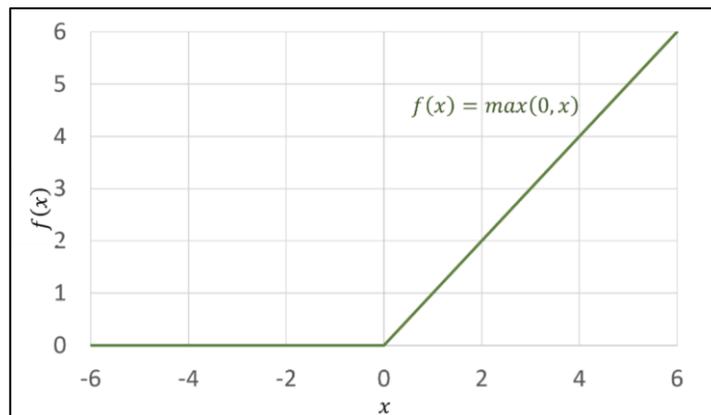
Fungsi ReLU (*rectified linear unit*) adalah fungsi non-linear dimana neuron hanya diaktifkan ketika output dari transformasi linear lebih besar dari nol, sehingga neuron tidak aktif secara bersamaan (Firmansyah & Hayadi, 2022). Relu digunakan untuk meminimalisir error dan saturasi serta optimal pada setiap *hidden layer* dari *neural network*. Cara kerja dari ReLU yaitu dengan melakukan *threshold* pada nilai 0 terhadap nilai piksel pada input citra, dimana akan menghasilkan output 0 ketika  $x < 0$  dan menghasilkan fungsi linear ketika  $x \geq 0$ , berikut persamaan fungsi dan operasi ReLU (Agarap, 2019). Fungsi matematis dari ReLU dinyatakan pada Persamaan 6.

$$f(x) = \max(0, x) \quad (6)$$

Keterangan :

$x$  = nilai inputan

$f(x)$  = hasil output berupa nilai 0 dan 1



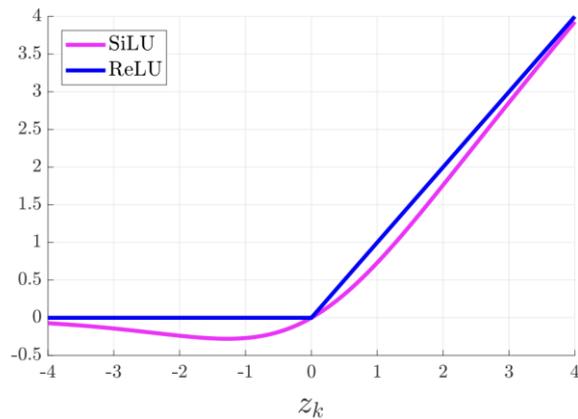
Gambar 9. Ilustrasi ReLU (Putra *et al.*, 2022).

Gambar 9 menunjukkan fungsi aktivasi ReLU yang meneruskan nilai positif tanpa adanya perubahan, sedangkan nilai negatif diubah menjadi nol. Visualisasinya berupa garis datar di nol untuk nilai negatif dan garis lurus untuk nilai positif. Efisiensi ReLU dibandingkan dengan fungsi *sigmoid* dan *tanh* terletak pada kemampuannya menjaga gradien tetap besar ketika input positif, sehingga secara efektif mengurangi masalah *vanishing gradient* yang terjadi pada fungsi aktivasi lain (Lin & Shen, 2018).

#### 2.19.4 *Sigmoid Linear Unit* (SiLU)

*Sigmoid Linear Unit* (SiLU) atau dikenal sebagai *swish* merupakan variasi non-linear dari fungsi aktivasi ReLU. Dalam YOLOv8, SiLU digunakan untuk menggantikan Leaky ReLU karena memiliki gradien yang lebih mulus dan dapat meningkatkan performa model. SiLU memiliki proses yang serupa dengan ReLU, apabila nilai input  $z_k$  memiliki nilai positif maka aktivasinya akan mendekati nilai input tersebut, sedangkan untuk nilai input negatif maka aktivasinya mendekati nol. Namun, berbeda dengan ReLU, SiLU memiliki tingkat minimum global sekitar -0.28 ketika  $z_k \approx -128$  yang menyebabkan SiLU memiliki kemampuan untuk menstabilkan pembelajaran jaringan secara alami yang mampu menghindari

*overfitting* dan menjaga model tetap stabil dalam proses pelatihan (Elfwing *et al.*, 2018). Ilustrasi dari fungsi aktivasi SiLU dapat dilihat pada Gambar 10.



Gambar 10. Aktivasi Fungsi SiLU dan ReLU (Elfwing *et al.*, 2018).

Fungsi aktivasi SiLU adalah versi komprehensif dari fungsi *sigmoid* dan ReLU. Fungsi SiLU memiliki perhitungan dengan mengalikan input dengan fungsi *sigmoid*, sehingga menyerupai ReLU yang bersifat kontinu dan “*undershooting*”. Secara sistematis, fungsi aktivasi SiLU didefinisikan pada Persamaan 7.

$$f(x) = x \cdot \text{sigmoid}(x) \quad (7)$$

Dimana  $z$  merupakan input dan  $\sigma(z)$  adalah fungsi sigmoid yang didefinisikan pada Persamaan (8).

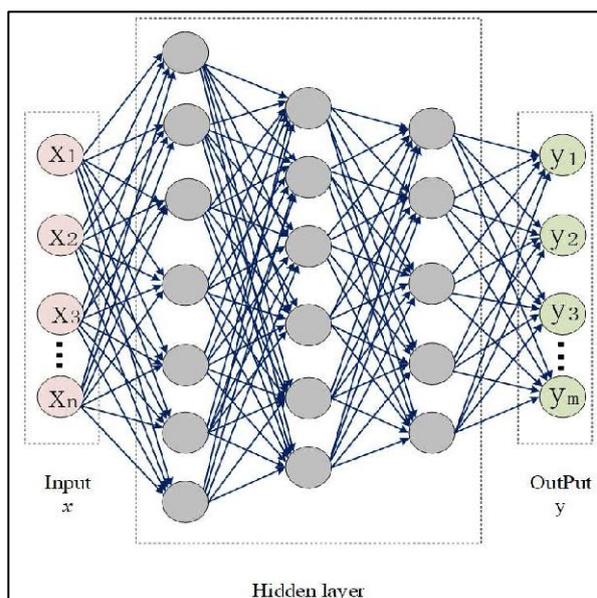
$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (8)$$

### 2.19.5 Fully-Connected Layer

*Fully connected layer* adalah lapisan output dari serangkaian proses konvolusi yang terletak di bagian bawah jaringan. Lapisan ini mirip dengan jaringan saraf *feedforward* dan menerima input dari lapisan *pooling* terakhir atau lapisan output konvolusi yang terlebih dahulu

diratakan sebelum dikirim ke lapisan berikutnya. Proses ini mengubah semua nilai yang dihasilkan dari lapisan *pooling* atau konvolusi terakhir menjadi vector atau matriks 3D yang memungkinkan distribusi merata dari output (Purwono *et al.*, 2022).

Lapisan *fully connected* digunakan untuk mempelajari kombinasi non-linear tingkat tinggi dari fitur-fitur yang dihasilkan oleh lapisan konvolusi. Dengan menerima input dari proses sebelumnya, lapisan ini bertugas mengidentifikasi fitur-fitur yang paling berkaitan dengan kelas tertentu. Fungsinya adalah menyatukan semua node menjadi satu dimensi, sehingga jaringan dapat melakukan prediksi dengan lebih akurat berdasarkan fitur-fitur yang telah dipelajari. Berikut merupakan ilustrasi dari proses *fully-connected* yang dapat dilihat pada Gambar 11.



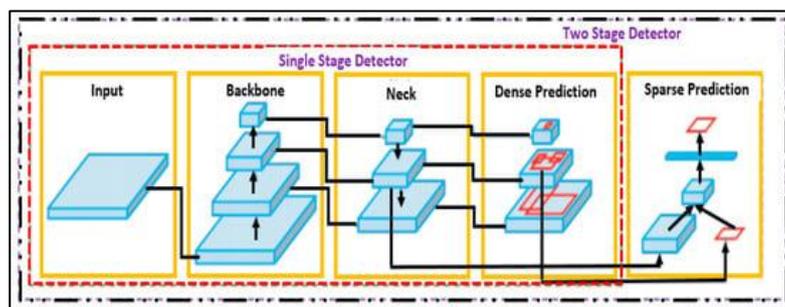
Gambar 11. *Fully Connected Layer* (Zhu & Song, 2020).

Gambar 11 menunjukkan tahapan *fully connected layer*, dimana setiap neuron pada lapisan *hidden layer* terhubung ke semua neuron di output layer. Input data diproses melalui *hidden layer* yang kemudian dilakukan proses ekstraksi pada fitur-fitur penting dan hasilnya diratakan (*flatten*) dan diteruskan ke *fully connected layer*. Tahap ini semua neuron saling terhubung untuk menghasilkan

kombinasi fitur yang digunakan oleh *output layer* untuk menghasilkan prediksi akhir. *Fully connected layer* bertanggung jawab dalam menggabungkan seluruh informasi dari *hidden layer* menjadi output yang sesuai (Zhu & Song, 2020).

## 2.20. YOLO (*You Only Look Once*)

Metode YOLO (*You Only Look Once*) pertama kali dikenalkan oleh Joseph Redmon pada sebuah penelitiannya yang berjudul “*You Only Look Once : Unified, Real-Time Object Detection*” di tahun 2015. YOLO adalah *framework deep learning* yang dirancang untuk mendeteksi objek secara *real-time* yang terinspirasi dari model GoogleNet dalam klasifikasi gambar. Model YOLO merupakan peningkatan dari *detector* berbasis *region* yang memanfaatkan *Convolutional Neural Network* (CNN) untuk mengekstrak fitur dari gambar dan memiliki performa yang unggul pada *dataset* standar seperti PASCAL VOC dan COCO (Garg *et al.*, 2018). Arsitektur model YOLO dapat dilihat pada Gambar 12.



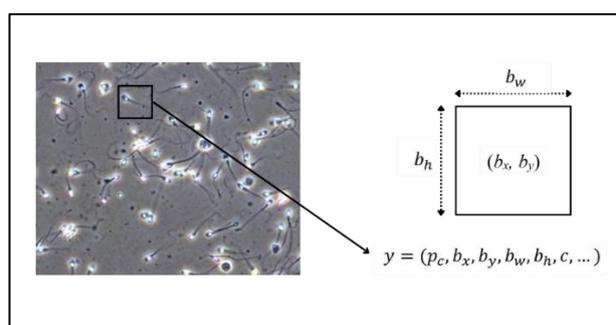
Gambar 12. Arsitektur YOLO (Hussain, 2023).

Pada metode YOLO (*you only look once*) terdiri dari 3 fitur sebagai berikut.

1. *Backbone* : Pada model YOLO, gambar diproses melalui *Backbone*. *Backbone* terdiri dari beberapa lapisan konvolusi dan *pooling*. Fungsinya adalah untuk mengekstraksi berbagai fitur penting dari gambar.

2. *Neck* : Setelah fitur diekstraksi oleh *Backbone*, fitur-fitur tersebut digabungkan dan diproses lebih lanjut di bagian *neck*. *Neck* terdiri dari beberapa lapisan konvolusi yang berfungsi untuk menggabungkan dan mengolah fitur dari *backbone*.
3. *Head* : Setelah fitur-fitur diproses di *neck*, fitur tersebut diteruskan ke bagian *head*. *Head* terdiri dari beberapa lapisan konvolusi dan bertujuan untuk menentukan lokasi dan kelas objek. Pada bagian *head*, YOLO akan melakukan prediksi lokasi dan kategori di dalam *bounding box* yang akan dihasilkan.

Dalam mendeteksi objek, YOLO akan membagi gambar menjadi beberapa kotak kecil atau sel grid, kemudian setiap sel akan memeriksa keberadaan objek di dalamnya, menentukan jenis objek, dan ukurannya menggunakan *bounding box*. *Bounding box* merupakan hasil proses konvolusi dari hasil input gambar yang diperoleh. *Bounding box* digunakan untuk memvisualisasikan objek dan menandai batas-batas di sekitar objek hasil dari identifikasi dan klasifikasi gambar yang dilakukan oleh komputer. Fungsi *bounding box* adalah untuk menghasilkan koordinat yang merepresentasikan posisi objek dalam gambar sekaligus memberi batasan dan label pada objek tersebut. Ilustrasi *bounding box* pada data VISEM dapat dilihat pada Gambar 13.



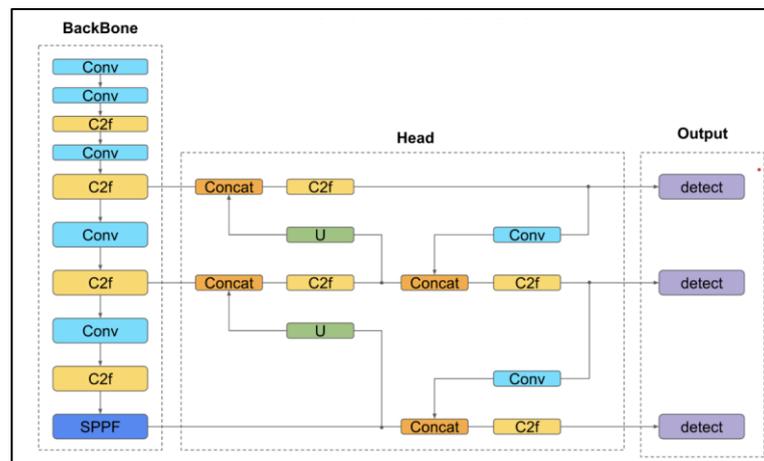
Gambar 13. Ilustrasi *Bounding Box* (VISEM Dataset).

Pada gambar 13 menjelaskan bahwa  $b_x$  dan  $b_y$  mewakili koordinat titik tengah dari kotak, sedangkan  $b_h$  dan  $b_w$  menunjukkan tinggi dan lebar kotak. Label  $c$  digunakan untuk menggambarkan banyaknya jumlah kelas pada objek yang terdeteksi. Semua atribut dari *bounding box* telah dilakukan

proses transformasi, sehingga nilai-nilai atribut tersebut dinyatakan dalam skala 0 hingga 1.

## 2.21. *You Only Look Once* Versi 8 (YOLOv8)

YOLOv8 adalah model deteksi objek YOLO (*You Only Look Once*) yang dirilis oleh Ultralytics pada tahun 2023. Model ini digunakan untuk mendeteksi dan mengklasifikasikan objek dalam gambar, serta membedakan objek satu sama lain (Sohan *et al.*, 2024). YOLOv8 adalah evolusi dari YOLOv7 yang memiliki berbagai perbaikan signifikan dalam hal performa dan akurasi deteksi objek. Evolusi tersebut ditunjukkan dengan penambahan lapisan pemrosesan gambar yang lebih kompleks serta penerapan teknik pembelajaran mesin yang lebih canggih. Arsitektur YOLOv8 dapat dilihat pada Gambar 14.



Gambar 14. Arsitektur YOLOv8 (Sohan *et al.*, 2024).

Pada model YOLOv8 terdiri dari 3 fitur sebagai berikut.

### a. Jaringan tulang punggung (*backbone*)

Jaringan tulang punggung (*backbone*) digunakan untuk mengekstraksi fitur berdasarkan gambar atau citra yang diinputkan menggunakan *Feature Pyramid Network* (FPN) yang memungkinkan model dapat mendeteksi objek dengan berbagai level skala atau resolusi.

b. Leher (*neck*)

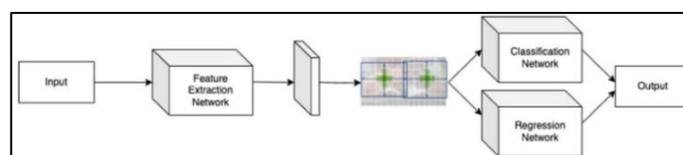
Leher (*neck*) digunakan untuk menyempurnakan fitur-fitur yang diinputkan menggunakan serangkaian *Cross-Layer Connection* (CLC) agar menghasilkan representasi fitur yang lebih akurat.

c. Kepala (*head*)

Kepala (*head*) digunakan untuk mengambil kembali fitur yang sudah disempurnakan pada *neck* yang kemudian memprediksi kotak pembatas atau grid, skor kelas objek, dan tingkat akurasi untuk setiap objek dalam gambar/citra. Pada YOLOv8 terdapat beberapa pembaharuan yang menjadi versi ini lebih optimal dibandingkan versi lain, yaitu:

a. ***Anchor-Free Detection***

*Anchor-free detection* merupakan metode deteksi objek yang dapat langsung memprediksi pusat objek tanpa mengandalkan kotak referensi (*anchor box*). Kelebihan dari metode ini yaitu dapat menyederhanakan proses dan mengurangi jumlah prediksi yang harus dibuat oleh model, sehingga mempercepat tahap *Non-Maximum Suppression* (NMS) (Sohan *et al.*, 2024). Tahap *Non-Maximum Suppression* (NMS) yaitu proses yang digunakan untuk menyaring dan memilih hasil prediksi terbaik setelah model melakukan inferensi. Arsitektur dari *Anchor-Free Detection* dapat dilihat pada Gambar 15.



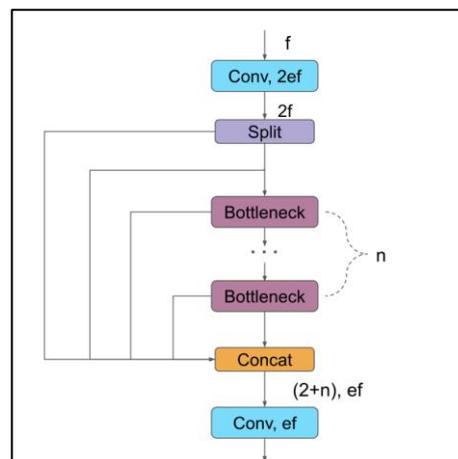
Gambar 15. Arsitektur *Anchor Free Detection* (Sun & Tian, 2023).

*Anchor-free detection models* (AFDM) memiliki cara kerja yang lebih sederhana dibandingkan dengan model berbasis *Anchor-Based Detection Models* (ABDM). *Anchor-free detection* tidak

menggunakan kotak *anchor* melainkan langsung memprediksi titik-titik penting (*keypoints*) dari objek dalam gambar. Setelah jaringan CNN (*Convolutional Neural Network*) mengekstraksi fitur-fitur dari gambar, maka akan dilakukan prediksi secara langsung pada fitur tersebut. Hal ini dapat mengurangi kompleksitas karena jumlah parameter yang lebih sedikit dan proses deteksi yang lebih cepat.

## b. C2f

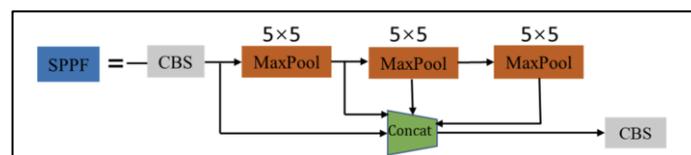
C2F adalah komponen baru dalam arsitektur YOLOv8 yang menggantikan lapisan C3 pada YOLOv5. Fungsi utama C2F adalah untuk menggabungkan (*concatenate*) output dari semua lapisan *Bottleneck* (Sohan *et al.*, 2024). Selain itu, C2F berfungsi sebagai modul penggabungan fitur (*feature fusion*) dan *upsampling* yang menggantikan arsitektur *neck* yang digunakan sebelumnya (Opara *et al.*, 2024). C2F secara efektif menggabungkan fitur dari berbagai lapisan *backbone* dan memperbesar (*upscale*) resolusinya sehingga lebih sesuai untuk deteksi objek. Dengan itu, C2F mampu membantu model dalam menangkap dan mengolah fitur pada berbagai skala dan resolusi, serta meningkatkan akurasi deteksi objek. Arsitektur dari C2F dapat dilihat pada Gambar 16.



Gambar 16. Arsitektur C2F (Sohan *et al.*, 2024).

### c. *Spatial Pyramid Pooling Fast (SPPF)*

*Spatial Pyramid Pooling Fast (SPPF)* adalah versi yang lebih cepat dari metode *Spatial Pyramid Pooling (SPP)* yang digunakan pada YOLOv5. SPPF mempercepat deteksi objek dengan menerapkan operasi *max pooling 5x5* sebanyak tiga kali secara berturut-turut untuk mempertahankan ukuran *receptive field* yang sama, sehingga model tetap dapat menangkap detail gambar tanpa mengurangi kualitas informasi yang diperoleh. Selain itu, SPPF menambahkan *average pooling* untuk menangkap informasi latar belakang. Dengan kombinasi *max pooling* dan *average pooling*, YOLOv8 dapat mendeteksi objek dengan berbagai skala dan resolusi secara lebih efisien dan akurat tanpa menghilangkan informasi yang penting dari gambar (Qiu *et al.*, 2022). Arsitektur dari SPPF dapat dilihat pada Gambar 17.

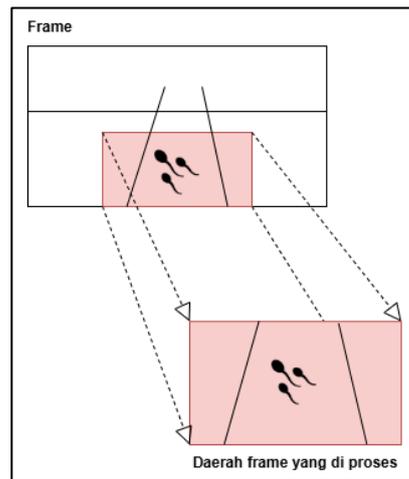


Gambar 17. Arsitektur SPPF (Qiu *et al.*, 2022).

## 2.22. *Region Of Interest (RoI)*

*Region of Interest (RoI)* merupakan salah satu teknik segmentasi dalam pengolahan citra digital yang memungkinkan pengguna untuk memfokuskan proses analisis pada area tertentu yang dianggap relevan dalam sebuah citra. Teknik ini bekerja dengan memberikan penekanan khusus pada wilayah yang dipilih, sehingga bagian tersebut memiliki kualitas lebih tinggi dibandingkan area lain di sekitarnya. *Region of Interest (RoI)* umumnya diterapkan sebelum proses deteksi objek dengan tujuan untuk membatasi area yang mengandung informasi penting, sehingga sistem tidak perlu memproses keseluruhan frame (Pratomo *et al.*, 2020). Dalam proses deteksi objek seperti kendaraan

pada citra atau video, pembatasan area pemrosesan dapat dilakukan dengan menentukan area jalan raya sebagai wilayah RoI. Tujuan dari pembatasan ini adalah untuk menghindari gangguan atau *noise* yang disebabkan oleh objek-objek di luar area tersebut, sehingga pengolahan citra menjadi lebih akurat dan efisien. Ilustrasi dari *Region of Interest* (RoI) dapat dilihat pada Gambar 18.



Gambar 18. Ilustrasi RoI (Pratomo *et al.*, 2020).

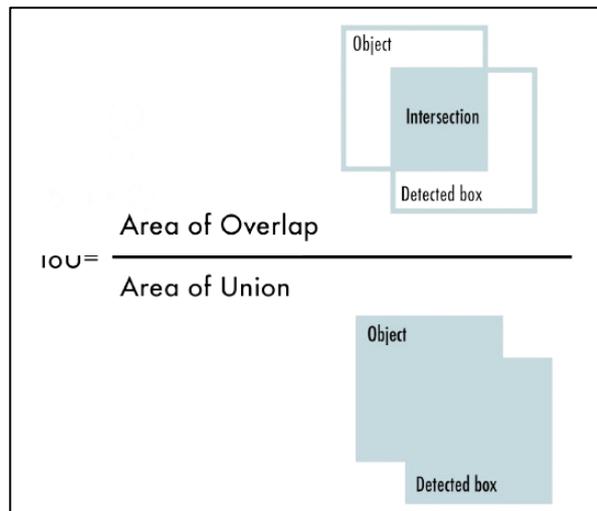
### 2.23. Intersection Over Union (IoU)

Intersection over Union (IoU) merupakan salah satu metrik evaluasi yang paling umum digunakan dalam tugas *object detection*. IoU digunakan untuk mengukur tingkat kesesuaian antara *bounding box* hasil prediksi model dengan *bounding box ground truth* (keadaan sebenarnya). Perhitungan IoU dilakukan dengan membagi luas area dari perpotongan (*intersection*) antara dua *bounding box* dengan luas gabungan (*union*) dari keduanya (Rezatofighi *et al.*, 2019). Secara sistematis, IoU dinyatakan pada persamaan 9.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (9)$$

Nilai IoU berada pada rentang antara 0 hingga 1. Semakin mendekati angka 1 maka tingkat kesesuaian antara prediksi dan *ground truth*. Sebaliknya, nilai IoU yang rendah menunjukkan bahwa *bounding box* prediksi jauh dari lokasi

objek sebenarnya. Berikut ilustrasi dari *Intersection of Union* (IoU) yang dapat dilihat pada Gambar 19.



Gambar 19. Ilustrasi IoU (Rezatofghi *et al.*, 2019).

#### 2.24. Confusion Matrix

*Confusion matrix* adalah sebuah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi, khususnya dalam *supervised learning*. Tabel ini memberikan gambaran visual tentang seberapa baik model mengklasifikasikan data berdasarkan kategori yang sebenarnya dan yang diprediksi. *Confusion matrix* terdiri dari empat komponen utama: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

Tabel 2. Evaluasi Model *Confusion Matrix*

	Actual	Positive	Negative
Predicted			
Positive		<i>True Positive</i>	<i>False Positif</i>
Negative		<i>False Negative</i>	<i>True Negative</i>

Keterangan:

*True Positive* (TP) = Data positif yang terdeteksi positif

*False Positive* (FP) = Data positif yang terdeteksi negatif

*True Negative* (TN) = Data negatif yang terdeteksi negatif

*False Negative* (FN) = Data negatif yang terdeteksi positif

Dengan menggunakan *confusion matrix*, kita bisa menghitung metrik evaluasi lain seperti *sensitivity*, *precision*, *accuracy*, dan *F-score*, yang memberikan wawasan lebih dalam mengenai kinerja model, terutama dalam menangani ketidakseimbangan data atau kasus kesalahan prediksi (Chagnon *et al.*, 2024).

a. *Sensitivity*

*Sensitivity*, juga dikenal sebagai *recall*, adalah kemampuan model untuk mendeteksi kasus positif dengan benar. *Sensitivity* dihitung sebagai proporsi antara prediksi TP dengan seluruh kasus positif, baik yang diprediksi benar maupun yang salah. Secara matematis, *sensitivity* dinyatakan pada Persamaan 10:

$$\frac{TP}{TP+FN} \quad (10)$$

b. *Precision*

*Precision* adalah ukuran yang digunakan untuk mengevaluasi seberapa relevan prediksi positif di antara semua prediksi yang dinyatakan positif oleh model. *Precision* dihitung sebagai proporsi antara prediksi TP dengan semua prediksi yang dinyatakan positif, baik yang benar maupun yang salah. Secara matematis, *precision* dinyatakan pada Persamaan 11:

$$\frac{TP}{TP+FP} \quad (11)$$

c. *Accuracy*

*Accuracy* adalah proporsi prediksi yang benar di antara semua prediksi yang dilakukan oleh model. Ini mengukur seberapa baik sebuah model klasifikasi dalam memprediksi kondisi secara keseluruhan. Secara matematis, *accuracy* dinyatakan pada Persamaan 12:

$$\frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

d. *F1-Score*

*F1-Score* adalah metrik yang menggabungkan *Precision* dan *Sensitivity* ke dalam satu ukuran tunggal. *F1-Score* sangat berguna ketika ada ketidakseimbangan antara kelas positif dan negatif atau ketika kita ingin mempertimbangkan performa model secara keseluruhan, bukan hanya fokus pada satu metrik. Secara matematis, *F1-Score* dinyatakan pada Persamaan 13:

$$2 \times \frac{\text{Sensitivity} \times \text{Presisi}}{\text{Sensitivity} + \text{Presisi}} \quad (13)$$

### 2.25. *Average Precision (AP)*

Average Precision (AP) merupakan salah satu metrik evaluasi yang sering digunakan dalam *object detection*. Metrik ini didasarkan pada area di bawah kurva *precision* dan *recall* (PR) yang telah diproses untuk menghilangkan zig-zag yang tidak stabil, sehingga hasil evaluasi menjadi lebih akurat dan mudah diinterpretasikan (Padilla *et al.*, 2021). AP menggambarkan *trade-off* atau keseimbangan antara *precision* dan *recall* pada berbagai tingkat *confidence* dari prediksi *bounding box*. Perhitungan AP diawali dengan mengurutkan seluruh nilai *confidence* dari prediksi *bounding box* yang dihasilkan model. Selanjutnya, *precision* dan *recall* dihitung pada berbagai ambang batas (*threshold*) berbeda untuk membentuk kurva *precision-recall*. Semakin tinggi AP, maka semakin baik performa model dalam mendeteksi objek dengan tepat dan konsisten. Secara sistematis, *Average Precision (AP)* dinyatakan pada persamaan 14.

$$\int_{r=0}^1 p(r) dr \quad (14)$$

Keterangan :

$p(r)$  = fungsi *precision* terhadap *recall*

$r$  = nilai *recall*, kisaran 0 hingga 1

$dr$  = perubahan kecil pada *recall*

## 2.26. Mean Average Precision (mAP)

*Mean Average Precision* (mAP) adalah metrik yang umum digunakan untuk mengevaluasi kinerja algoritma deteksi objek. *Mean Average Precision* (mAP) mengukur akurasi model dalam melokalisasi dan mengklasifikasikan objek dalam gambar dengan mempertimbangkan nilai presisi dan *recall* pada berbagai ambang batas *Intersection over Union* (IoU) (Tian *et al.*, 2024). Metrik mAP dihitung berdasarkan rata-rata nilai *average precision* (AP) untuk setiap kelas dalam dataset, kemudian memberikan gambaran menyeluruh tentang kemampuan model dalam mengidentifikasi objek secara akurat di berbagai kelas. Secara sistematis, *Mean Average Precision* (mAP) dinyatakan pada persamaan 15.

$$\frac{1}{N} \sum_{i=1}^N AP_i \quad (15)$$

Keterangan :

N = total kelas dalam *dataset*

$AP_i$  = nilai *average precision* untuk kelas ke- $i$

## 2.27. ByteTrack

*Bytetrack* merupakan algoritma pelacakan multi-objek yang memanfaatkan hasil deteksi objek untuk menentukan posisi dan pergerakan objek pada setiap *frame* video yang dibuat oleh Yifu Zhang bersama tim peneliti dari Tsinghua University, Shanghai AI Laboratory, Alibaba DAMO Academy, dan Huazhong University of Science and Technology pada tahun 2021. Algoritma ini menggunakan Kalman filter untuk memprediksi posisi objek dan menggabungkan informasi posisi, kecepatan, serta ciri visual dalam proses persamaan objek antar *frame*. Salah satu keunggulan *Bytetrack* adalah mampu menangani deteksi dengan *confidence score* rendah tanpa mengganggu stabilitas pelacakan. Objek yang tidak aktif dalam waktu tertentu akan dihapus. Dengan pembaruan yang dilakukan secara terus-menerus, maka *Bytetrack* mampu memberikan pelacakan yang akurat dan stabil terutama

dalam kondisi lingkungan yang padat dan kompleks (Liu *et al.*, 2025). Secara sistematis, *ByteTrack* dinyatakan pada persamaan 16, 17 18, 19 dan 20:

**a. Prediksi (*Prediction Step*)**

- Prediksi State

$$\hat{x}_{n+1,n} = F\hat{x}_{n,n} + GU_n \quad (16)$$

$\hat{x}_{n+1,n}$  : Prediksi *state* pada waktu  $n + 1$  berdasarkan waktu  $n$

$F$  : *State transition matrix*, menggambarkan hubungan antara *state* sekarang dan berikutnya.

$\hat{x}_{n,n}$  : Estimasi *state* setelah koreksi pada waktu  $n$

$G$  : Matriks control

$U_n$  : Vektor Input kendali pada waktu  $n$

- Prediksi Ketidakpastian

$$P_{n+1,n} = FP_{n,n}F^T + Q \quad (17)$$

$P_{n+1,n}$  : Prediksi *covariance matrix* pada waktu  $n + 1$

$P_{n,n}$  : Kovariansi estimasi setelah koreksi waktu  $n$

$F^T$  : Transpos dari  $F$

$Q$  : *Process noise covariance*

**b. Update (*Correction Step*)**

- Hitung Kalman Gain

$$Kn = P_{n,n-1}H^T(HP_{n,n-1}H^T + R_n)^{-1} \quad (18)$$

$Kn$  : Kalman gain pada waktu  $n$

$P_{n,n-1}$  : *Predicted error covariance* yaitu ketidakpastian dari estimasi *state* sebelum dikoreksi

$H$  : Matriks observasi

$H^T$  : Transpose dari  $H$

$R_n$  : Covariance dari noise pengukuran

$(.)^{-1}$  : Invers matriks

- Perbarui Estimasi State

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + K_n (Z_n - H\hat{x}_{n,n-1}) \quad (19)$$

$\hat{x}_{n,n}$  : *State* estimasi akhir setelah koreksi pada waktu  $n$

$\hat{x}_{n,n-1}$  : *State* prediksi awal sebelum koreksi

$Z_n$  : Pengukuran aktual pada waktu  $n$

$H\hat{x}_{n,n-1}$  : Prediksi hasil pengukuran dari state awal

- Perbarui Kovariansi Estimasi

$$P_{n,n} = (I - K_n H)P_{n,n-1}(I - K_n H)^T + K_n R_n K_n^T \quad (20)$$

$P_{n,n}$  : *Covariance* estimasi setelah update

$I$  : Matriks identitas, ukuran sama dengan  $P$

$R_n$  : *Covariance noise* pengukuran

$K_n R_n K_n^T$  : Koreksi tambahan jika terjadi *noise* tinggi



### 3.2. Alat Pendukung

Alat pendukung yang dilakukan pada penelitian ini meliputi perangkat keras dan perangkat lunak, yaitu:

#### a. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan untuk penelitian ini adalah laptop dengan spesifikasi sebagai berikut:

1. Laptop : Asus Vivobook
2. *Processor* : Intel® Core™ i7-12700H
3. Penyimpanan : SSD 512GB
4. Ram : 16GB
5. GPU : Intel® Iris® Xe Graphics

#### b. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam proses penelitian ini yaitu:

1. Sistem Operasi *windows 11 Home Single Language 64-bit*
2. *Python* versi 3.9.0

Python adalah bahasa pemrograman serbaguna yang populer digunakan untuk berbagai kebutuhan, mulai dari pengembangan aplikasi, pemrosesan data, hingga pengembangan kecerdasan buatan. Versi Python 3.9.0 menyediakan fitur-fitur baru dan perbaikan performa yang mendukung berbagai *library* dan *framework* terkini yang digunakan dalam pengolahan citra dan *machine learning*.

3. *Google Drive*

Google Drive merupakan layanan penyimpanan *cloud* yang disediakan oleh Google. Dalam konteks penelitian dan pengembangan *machine learning*, Google Drive sering digunakan untuk menyimpan dataset, file konfigurasi, maupun hasil pelatihan model. Drive ini juga terintegrasi dengan Google Colab, sehingga

memudahkan dalam mengakses dan mengelola file secara langsung dari *notebook*.

#### 4. *Google Colab*

Google Colab adalah layanan *cloud* gratis yang memungkinkan pengguna menulis dan menjalankan kode Python langsung dari *browser*. Colab menyediakan akses GPU gratis serta integrasi dengan berbagai *library* populer seperti TensorFlow dan PyTorch, sehingga sangat mendukung pengembangan dan pelatihan model *deep learning*.

#### 5. *Roboflow*

Roboflow adalah *platform* yang membantu pengguna dalam pengelolaan dataset untuk *computer vision*. Roboflow memungkinkan pengguna untuk mengupload, mengelola, mengonversi anotasi, dan melakukan augmentasi gambar dengan mudah. *Platform* ini juga mendukung ekspor dataset ke format yang kompatibel dengan berbagai model seperti YOLO, TensorFlow, dan lainnya.

#### 6. *Library Sklearn 1.2.2*

Scikit-learn atau sklearn adalah *library* Python yang digunakan untuk *machine learning*. Versi 1.2.2 menyediakan beragam fungsi seperti klasifikasi, regresi, clustering, dan evaluasi model. Sklearn banyak digunakan dalam *pipeline* analisis data dan sebagai alat evaluasi performa model.

#### 7. *Library OpenCV 4.6.0*

OpenCV (*Open Source Computer Vision Library*) adalah *library open source* yang dirancang untuk pemrosesan citra dan video. Pada versi 4.6.0, OpenCV menawarkan berbagai fitur seperti pembacaan gambar, pemrosesan frame dari video, serta teknik augmentasi dan transformasi gambar untuk keperluan pelatihan model.

8. *Library OS 3.9.0*

OS adalah library bawaan atau *built-in* dari Python yang digunakan untuk mengakses fitur-fitur sistem operasi. Pada Python 3.9.0, OS berfungsi dalam mengelola file, folder, dan jalur direktori yang dibutuhkan selama proses pemrosesan data dan penyimpanan file hasil.

9. *Library Numpy 1.23.0*

Numpy adalah library fundamental untuk komputasi numerik di Python. Library ini menyediakan objek array multidimensi, fungsi matematika tingkat tinggi, dan kemampuan manipulasi array. Versi 1.23.0 meningkatkan performa dan kompatibilitas dengan berbagai operasi numerik dalam *machine learning*.

10. *Library Pandas 1.1.4*

Pandas adalah *library* Python yang digunakan untuk manipulasi dan analisis data dalam bentuk tabel atau *data frame*. Versi 1.1.4 mendukung pembacaan data dari berbagai sumber seperti file CSV, Excel, dan SQL serta menyediakan fitur untuk memfilter, menggabungkan, dan mengelola data secara efisien.

11. *Library Matplotlib 3.3.0*

Matplotlib adalah *library* visualisasi data dalam Python. Dengan versi 3.3.0 maka pengguna dapat membuat berbagai jenis grafik dan plot, seperti grafik garis, histogram, scatter plot, dan visualisasi hasil pelatihan model seperti loss dan akurasi.

12. *Library PyTorch 1.8.0*

PyTorch adalah *framework deep learning open source* yang dikembangkan oleh Facebook AI. Versi 1.8.0 menyediakan dukungan yang luas untuk pembangunan dan pelatihan model *neural network*,

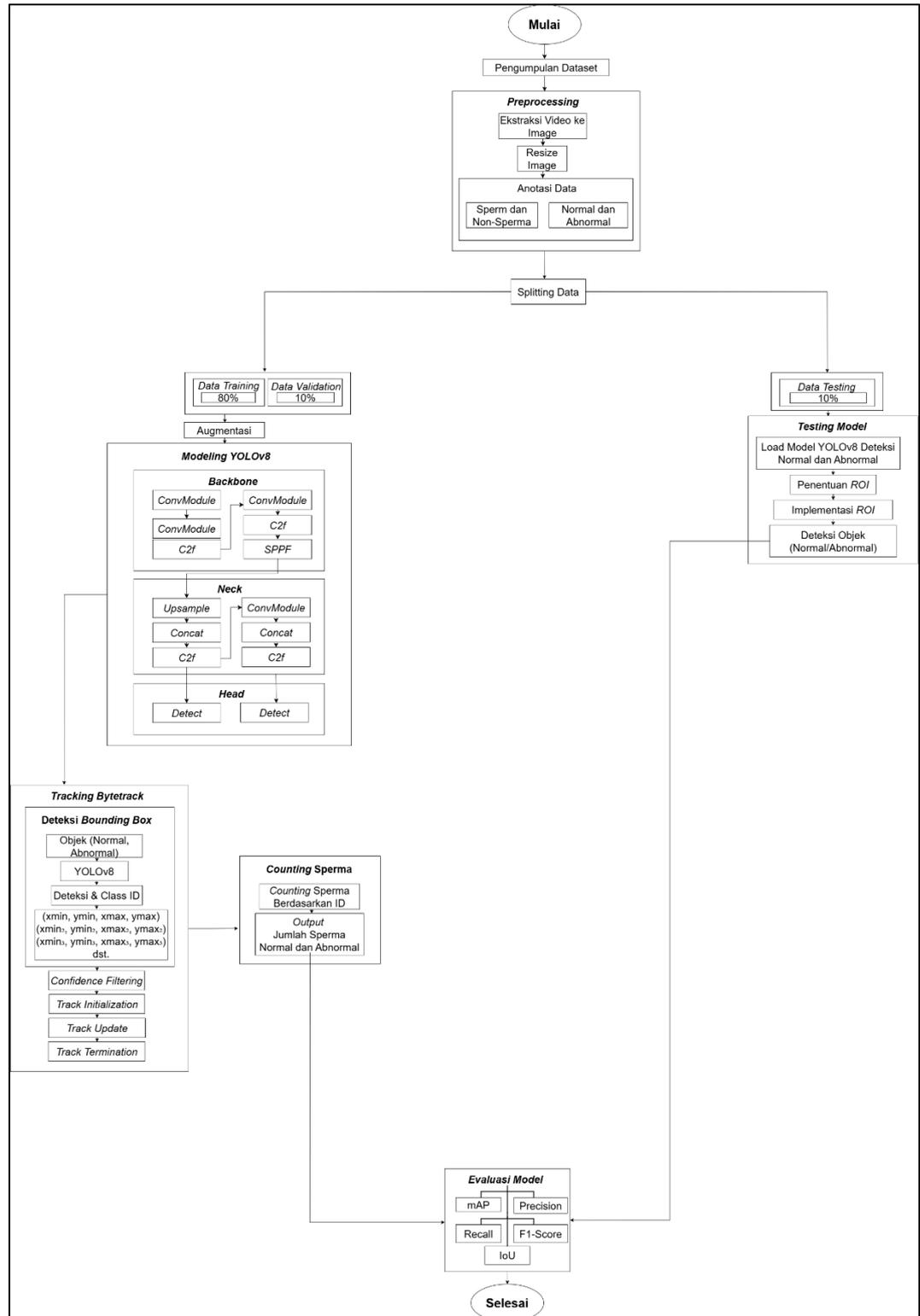
serta memungkinkan pengembangan model yang dinamis dan fleksibel melalui pendekatan imperative.

### 13. *Library Ultralytics* 8.3.11

Ultralytics adalah library resmi yang digunakan untuk mengakses dan menjalankan model YOLOv8. Versi 8.3.11 menyediakan antarmuka pemrograman yang mudah digunakan untuk pelatihan, validasi, prediksi, dan evaluasi model deteksi objek dengan efisiensi tinggi.

### **3.3. Tahapan Penelitian**

Berikut alur kerja dari penelitian Deteksi *Teratozoospermia* Berdasarkan morfologi sperma manusia menggunakan YOLOv8 ditunjukkan pada Gambar 20 berikut.



Gambar 20. Tahapan Penelitian.

### 3.3.1 Pengumpulan Dataset

*Dataset* yang digunakan pada penelitian ini diperoleh dari *open dataset* dengan laman <https://datasets.simula.no/visem/>. *Dataset* VISEM tersebut berisi data sebanyak 85 video yang berasal dari rekaman sperma dibawah mikroskop dari peserta laki-laki dengan usia 18 tahun ke atas di negara Norwegia. *Dataset* ini diterbitkan oleh Simula Research Laboratory dan Simula Metropolitan Center for Digital Engineering (SimulaMet). Penelitian ini dilakukan pada tahun 2008 hingga 2013 dengan tujuan untuk mengembangkan sistem analisis sperma berbasis *Computer Aided Sperm Analysis* (CASA). Kumpulan data ini memiliki muatan sebesar 35 *gigabyte*, dengan durasi setiap video yaitu dua hingga tujuh menit. Video tersebut memiliki resolusi 640 piksel untuk lebar (*width*) dan 480 piksel untuk tinggi (*height*). *Dataset* ini akan dibagi menjadi 2 kelas yaitu sperma normal dan abnormal (*teratozoospermia*).

### 3.3.2 Pre-processing

#### a. Ekstraksi Video ke Gambar

Pada tahap pertama *pre-processing* pada penelitian ini dilakukan proses ekstraksi video menjadi gambar. Setiap data video sperma yang terdapat pada dataset VISEM akan diekstrak untuk menghasilkan *frame-frame* gambar individual. Proses ekstraksi ini bertujuan untuk mengubah format data dari video menjadi kumpulan gambar agar lebih mudah diolah pada tahapan berikutnya. Berdasarkan 85 video sperma yang digunakan, proses ekstraksi menghasilkan sebanyak 1.330 *image*. Seluruh gambar hasil ekstraksi tersebut kemudian akan digunakan dalam proses *resize image* agar memiliki ukuran yang seragam sesuai dengan format yang dibutuhkan oleh model YOLOv8. Penyeragaman

ukuran gambar ini penting dilakukan agar proses pelatihan model dapat berjalan secara optimal dan konsisten.

### ***b. Resize Image***

Pada tahap ini, gambar yang diekstraksi dengan ukuran awal 640 x 480 piksel akan diubah menjadi 640 x 640 piksel untuk menyamakan ukuran dari setiap gambar/citra. Proses *resize image* ini bertujuan untuk menyamakan dimensi gambar agar menjaga konsistensi data, sehingga gambar-gambar tersebut siap untuk diproses lebih lanjut dengan kualitas yang optimal.

### ***c. Anotasi Data***

Pada tahap ini, setiap gambar melalui proses *resize* selanjutnya dianotasi menggunakan Roboflow. Anotasi dilakukan dalam dua skenario berbeda. Pertama, anotasi dilakukan untuk membedakan atau memberi label pada objek sperma dan non-sperma sebanyak 135 citra hasil ekstraksi yang dihasilkan label untuk sperma sebanyak 5.779 label dan non-sperma sebanyak 711 label. Kedua, anotasi dilakukan untuk mengklasifikasikan sperma dengan memberikan label pada sperma normal dan abnormal (*teratozoospermia*) sebanyak 1.330 citra hasil ekstraksi yang dihasilkan label untuk normal sebanyak 3.098 label dan abnormal sebanyak 1.869 label. Proses ini bertujuan agar setiap gambar memiliki informasi yang tepat sehingga model dapat mendeteksi dan mengklasifikasikan sperma berdasarkan morfologi dengan akurat. Dalam proses *labelling* dilakukan secara manual dengan menggambar *bounding box* di sekitar objek sperma dan memberi label seperti normal dan abnormal. Kemudian, file *labelling*

diunggah pada Google Drive untuk penyimpanan dan diakses pada tahap *modeling*.

#### ***d. Cropping Image***

Pada tahap ini, *cropping* dilakukan untuk memisahkan objek sperma normal dan abnormal dari gambar yang mengandung banyak sperma. Proses ini penting karena dalam satu gambar yang sudah diekstraksi terdapat beberapa sperma dengan morfologi yang berbeda yaitu normal dan abnormal, sehingga untuk dilakukan deteksi dan klasifikasi setiap sperma secara akurat diperlukan pemotongan gambar agar hanya satu sperma yang diisolasi dalam satu gambar. Tujuan dari *cropping* yaitu membantu mengurangi *noise* atau gangguan dari objek lain yang tidak relevan.

### **3.3.4 Pembagian Dataset**

Pada tahap ini, terdapat dua skenario dalam pelatihan model yaitu menggunakan proporsi 80:10:10 dan 70:15:15. *Dataset* akan dibagi menjadi tiga bagian yaitu *training dataset*, *validation dataset*, dan *testing dataset*. *Training dataset* mencakup 80% pada skenario pertama dan 70% pada skenario kedua dari total *dataset* untuk melatih model dalam mengenali pola-pola dari data seperti sperma normal dan abnormal (*teratozoospermia*). *Validation dataset* mencakup 10% skenario pertama dan 15% pada skenario kedua dari total *dataset* yang berfungsi untuk mengevaluasi kinerja model selama proses pelatihan dan membantu penyesuaian *hyperparameter* dalam mencegah *overfitting*. Terakhir, *testing dataset* mencakup 10% pada skenario pertama dan 15% pada skenario kedua dari total *dataset* yang digunakan setelah proses pelatihan selesai untuk mengukur performa akhir model pada data yang tidak digunakan selama proses pelatihan model. Pembagian *dataset* ini dilakukan secara acak agar setiap subset

memiliki representasi yang seimbang antara sperma normal dan sperma abnormal (*teratozoospermia*).

### 3.3.5 Augmentasi Data

Tahap ini dilakukan untuk memperbanyak variasi dan jumlah data sperma normal dan abnormal yang telah melalui proses *cropping* pada data *training* atau pelatihan. Augmentasi dilakukan menggunakan Roboflow dengan teknik seperti *flip* horizontal, rotasi 90°, *zoom* hingga 20%, rotasi acak antara -15° hingga +15°, *shear*, perubahan warna (*hue*, *saturasi*, *brightness*, *exposure*), serta penambahan *noise*. Hasilnya, jumlah data sperma normal meningkat dari 2.431 menjadi 4.862, dan sperma abnormal dari 1.508 menjadi 3.016.

### 3.3.6 Training Model YOLOv8

Pada penelitian ini, proses pelatihan dilakukan dalam dua tahap. Tahap pertama bertujuan untuk melatih model YOLOv8 dalam membedakan objek sperma dan non-sperma. Setelah model berhasil mengenali objek sperma dan non sperma, tahap kedua dilanjutkan dengan pelatihan untuk mengklasifikasikan sperma ke dalam dua kategori berdasarkan morfologinya yaitu sperma normal dan abnormal (*teratozoospermia*). Dataset yang digunakan telah melalui proses pra-pemrosesan dan dibagi menjadi tiga bagian, yaitu data latih (*training*), validasi (*validation*), dan uji (*testing*). Data latih digunakan untuk membangun model, data validasi berfungsi untuk memantau performa selama pelatihan dan mencegah *overfitting*, sementara data uji digunakan untuk mengevaluasi kemampuan generalisasi model terhadap data baru.

Model YOLOv8 mengenali objek melalui ekstraksi fitur menggunakan lapisan konvolusi, dilanjutkan dengan C2f untuk mempertahankan detail, serta SPPF (*Spatial Pyramid Pooling Fast*) untuk mengakomodasi variasi ukuran objek. Fitur yang diperoleh diperbesar melalui proses *upsampling*, kemudian digabungkan untuk menghasilkan prediksi berupa *bounding box* dan label kelas. *Hyperparameter* seperti *epoch*, *learning rate*, *batch size*, *momentum*, dan lainnya diatur untuk mencapai performa optimal. Dalam penelitian ini, proses *tuning* dilakukan menggunakan *Ray Tune* untuk mencari kombinasi parameter terbaik yang kemudian digunakan untuk proses *training* dalam melatih model YOLOv8. Evaluasi performa model selama pelatihan dilakukan menggunakan metrik seperti *precision*, *recall*, *f1-score* dan *mean average precision (mAP)*, dan model terbaik disimpan dalam file *best.pt*.

### 3.3.7 Evaluasi Kinerja Model

Pada tahap ini, performa model YOLOv8 yang telah dilatih akan dievaluasi menggunakan *testing dataset*. Beberapa metrik evaluasi yang umum digunakan adalah *precision*, *recall*, dan *F1-score* untuk mengukur kemampuan model dalam mendeteksi dan mengklasifikasikan sperma normal dan abnormal (*teratozoospermia*). Selain itu, *mean Average Precision (mAP)* akan digunakan untuk mengevaluasi kinerja deteksi objek secara keseluruhan. Metrik-metrik ini digunakan untuk membantu dalam menilai seberapa baik model mengenali objek berdasarkan akurasi, *precision*, *recall* dan efektivitas model secara keseluruhan.

### 3.3.8 Tracking Bytetrack

Pada tahap pelacakan (*tracking*) menggunakan algoritma ByteTrack yang dibuat oleh Yifu Zhang bersama tim peneliti dari Tsinghua University. Pada tahap *tracking* ini digunakan lima sampel video sebagai input untuk menguji kemampuan sistem dalam mendeteksi sperma normal dan abnormal. Kelima video ini dipilih karena memiliki tingkat kepadatan objek yang bervariasi, sehingga dapat merepresentasikan berbagai kondisi nyata dalam proses analisis. Setelah proses pelatihan model YOLOv8 selesai dilakukan, diperoleh file model hasil pelatihan terbaik yang disimpan dalam berkas *best.pt*. File ini kemudian diintegrasikan ke dalam sistem *tracking* dengan algoritma ByteTrack.

ByteTrack bekerja dengan melacak objek dari satu *frame* ke *frame* berikutnya secara *real-time*. Setiap objek yang terdeteksi oleh model YOLOv8 akan diberikan ID unik, sehingga pergerakannya dapat diikuti secara konsisten sepanjang durasi video. Pemilihan video sampel bertujuan untuk mengevaluasi performa sistem secara general dalam kondisi yang berbeda, sehingga dapat menunjukkan tingkat efektivitas deteksi dan pelacakan dalam skenario dunia nyata. Berikut merupakan video yang digunakan untuk melacak sperma normal dan abnormal menggunakan ByteTrack dapat dilihat pada Tabel 4.

Tabel 4. Video Inputan *Tracking*

No.	Video	Durasi Video	Justifikasi Video
1	12.avi	30 detik	Jumlah objek terlihat cukup banyak namun tidak padat (sedang)
2	30.avi	30 detik	Jumlah terlihat sedikit objek (sedikit)

Tabel 4. Lanjutan

3	32.avi	30 detik	Jumlah objek merata dan terlihat banyak (padat)
4	42.avi	30 detik	Jumlah objek terlihat cukup banyak namun tidak padat (sedang)
5	55.avi	30 detik	Jumlah objek sangat banyak dan memenuhi area frame (sangat padat)

### 3.3.9 Counting Sperma

Proses *counting* bertujuan untuk mengetahui perbandingan jumlah sperma normal dan abnormal yang terdeteksi dalam video. *Counting* dilakukan bersamaan dengan *tracking*, dimana setiap sperma yang berhasil diklasifikasikan langsung dihitung dan dicatat. Pada setiap objek sperma yang terdeteksi, maka informasi penting seperti ID objek, nomor *frame*, hasil klasifikasi (normal atau abnormal), dan posisi *bounding box* akan diambil dan disimpan. Data ini direkap ulang pada setiap *frame* yang diproses, kemudian disimpan dalam file .csv secara berkala. Proses ini berjalan terus dari *frame* pertama hingga *frame* terakhir video. Setelah seluruh video selesai dianalisis, sistem menghitung total jumlah sperma normal dan abnormal yang terdeteksi, dan hasil akhirnya akan disimpan dalam file.csv untuk mengetahui berapa banyak jumlah sperma normal dan abnormal dalam video yang diinputkan.

## V. PENUTUP

### 5.1 Kesimpulan

Penelitian mengenai Deteksi *Teratozoospermia* Berdasarkan Morfologi Sperma Manusia Menggunakan YOLOv8 telah dilakukan dengan melibatkan evaluasi terhadap kinerja model. Berdasarkan hasil yang diperoleh, terdapat beberapa poin kesimpulan yang dapat diambil terkait performa model YOLOv8 yang dikombinasikan dengan algoritma ByteTrack dalam melakukan deteksi, penghitungan, dan klasifikasi sperma menjadi dua kelas yaitu normal dan abnormal. Berikut merupakan beberapa kesimpulan yang dapat diambil dari penelitian ini, antara lain.

1. Penelitian ini berhasil mengimplementasikan model YOLOv8 untuk mendeteksi dan mengklasifikasikan kedalam dua kelas yaitu sperma normal dan abnormal berdasarkan morfologinya. Penelitian ini dilakukan menggunakan dua skenario pembagian dataset yaitu 80:10:10 dan 70:15:15. Pada skenario 80:10:10, performa terbaik dicapai pada *epoch* ke-120 dan *oversampling* dengan nilai mAP sebesar 0.989. Selain itu, pada skenario 70:15:15, hasil terbaik diperoleh pada *epoch* ke-120 dengan nilai mAP sebesar 0.84.
2. Penelitian ini berhasil melakukan deteksi dengan hasil matriks *precision*, *recall*, dan *F1-Score* terbaik pada skenario 80:10:10 pada *epoch* 120 yang di *oversampling* masing-masing sebesar 0.965, 0.974, 0.969. Sedangkan pada skenario 70:15:15 hasil matriks *precision*, *recall*, dan *F1-Score* terbaik pada *epoch* 120 masing-masing sebesar 0.741, 0.882, 0.805.

3. Penelitian ini mampu mengimplementasikan algoritma ByteTrack untuk melakukan pelacakan objek sperma normal dan abnormal secara *real-time*. Setiap objek yang terdeteksi diberikan ID unik, sehingga memungkinkan pelacakan pergerakan secara individual. Selain itu, sistem juga mampu menghitung jumlah masing-masing sperma normal dan abnormal berdasarkan video yang diinputkan apabila setiap objek memiliki *confidence score*  $> 0.25$ .

## 5.2 Saran

Berdasarkan penelitian yang telah dilakukan diperoleh saran atau rekomendasi yang dapat dijadikan untuk melanjutkan penelitian ini yaitu :

1. Menambahkan variasi dataset yang digunakan dalam melatih model yang memiliki kualitas visual yang jelas, baik dari segi pencahayaan maupun tidak *blur* guna meningkatkan keakuratan dalam proses deteksi.
2. Mengembangkan sistem berbasis web yang mampu melakukan pendeteksian objek secara otomatis dan *real-time* dengan waktu pemrosesan yang lebih singkat.
3. Melakukan pendeteksian objek dengan menggunakan model *deep learning* lain seperti faster R-CNN, SSD, dan sejenisnya ataupun memanfaatkan versi terbaru dari YOLO untuk membandingkan efektivitas dari performa model dalam mendeteksi objek.

## DAFTAR PUSTAKA

- Agarap, A. F. M. (2019). *Deep Learning using Rectified Linear Units (ReLU)*.  
<https://doi.org/10.48550/arXiv.1803.08375>
- Akbar, A. (2020). Gambaran Faktor Penyebab Infertilitas Pria Di Indonesia :  
Meta Analisis. *Jurnal Pandu Husada*, 1(2), 66.  
<https://doi.org/10.30596/jph.v1i2.4433>
- Aliyah, S. N., Santoso, H., & Zayadi, H. (2022). *Analisis Normalitas dan Abnormalitas Spermatozoa Segar Sapi Limousin (Bos taurus) dan Sapi Bali (Bos sondaicus) Sebelum Proses Pembekuan Di Balai Besar Inseminasi Buatan Singosari Malang Normality and Abnormality Analysis of Fresh Spermatozoa Limousin Cattle (Bos taurus) and Bali Cattle (Bos sondaicus) Before Freezing at the Singosari Artificial Insemination Center Malang*. 3(1), 47–55.
- Annisa, F., & Mahfudah, U. (2019). *Faktor Yang Mempengaruhi Kejadian Infertilitas Pada Pria*. 182–186.
- Aristoteles, Syarif, A., & Lumbanraja, F. R. (2023). Systematic Review: Perkembangan Machine Learning Pada Sperma Manusia. *TEKNOINFO*, 17(1), 112–118. <https://ejournal.teknokrat.ac.id/index.php/teknoinfo/index>
- Aristoteles, Syarif, A., Sutyarso, Lumbanraja, F. R., & Hidayatullah, A. (2022). Identification of Human Sperm based on Morphology Using the You Only Look Once Version 4 Algorithm. *IJACSA) International Journal of Advanced Computer Science and Applications*, 13(7).  
<https://datasets.simula.no/visem/>.

- Atmoko, W., Savira, M., Shah, R., Chung, E., & Agarwal, A. (2024). Isolated teratozoospermia: revisiting its relevance in male infertility: a narrative review. *Translational Andrology and Urology*, *13*(2), 260–273. <https://doi.org/10.21037/tau-23-397>
- Azmi, K., Defit, S., & Sumijan. (2023). *Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat*. *16*(1), 28–40.
- Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). Cnn variants for computer vision: History, architecture, application, challenges and future scope. *Electronics (Switzerland)*, *10*(20), 1-28. <https://doi.org/10.3390/electronics10202470>
- Chagnon, J., Hagenbuchner, M., Tsoi, A. C., & Scarselli, F. (2024). On the effects of recursive convolutional layers in convolutional neural networks. *Neurocomputing*, *591*, 1–18. <https://doi.org/10.1016/j.neucom.2024.127767>
- Czajkowski, M., & Kretowski, M. (2019). Decision tree underfitting in mining of gene expression data. An evolutionary multi-test tree approach. *Expert Systems with Applications*, *137*, 392–404. <https://doi.org/10.1016/j.eswa.2019.07.019>
- Dobrovolny, M., Benes, J., Langer, J., Krejcar, O., & Selamat, A. (2023). Study on Sperm-Cell Detection Using YOLOv5 Architecture with Labeled Dataset. *Genes*, *14*(2). <https://doi.org/10.3390/genes14020451>
- Eka Okta Putra, G., Queena Fredlina, K., & Nyoman Yudi Anggara Wijaya, I. (2024). IMPLEMENTASI TRANSFER LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK DALAM KLASIFIKASI PENYU. *Jurnal Mahasiswa Teknik Informatika*, *8*(1), 1077–1082.
- Elfwing, S., Uchibe, E., & Doya, K. (2018). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, *107*, 3–11. <https://doi.org/10.1016/j.neunet.2017.12.012>

- Firmansyah, I., & Hayadi, B. H. (2022). Komparasi Fungsi Aktivasi Relu dan Tanh Pada Multilayer Perceptron. *Informatika Dan Komputer*, 6(2), 200–206.
- Garg, D., Goel, P., Pandya, S., Ganatra, A., & Kotecha, K. (2018). *A Deep Learning Approach for Face Detection using YOLO*.
- Hassan, H., Ren, Z., Zhao, H., Huang, S., Li, D., Xiang, S., Kang, Y., Chen, S., & Huang, B. (2022). Review and classification of AI-enabled COVID-19 CT imaging models based on computer vision tasks. *Computers in Biology and Medicine*, 141, 1–21. <https://doi.org/10.1016/j.compbimed.2021.105123>
- Holzinger, A., Langs, G., Denk, H., Zatloukal, K., & Müller, H. (2019). Causability and explainability of artificial intelligence in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(4), 1–13. <https://doi.org/10.1002/widm.1312>
- Hussain, M. (2023). YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*, 11(7), 1–25. <https://doi.org/10.3390/machines11070677>
- Ilhan, H. O., & Serbes, G. (2022). Sperm morphology analysis by using the fusion of two-stage fine-tuned deep networks. *Biomedical Signal Processing and Control*, 71, 103246. <https://doi.org/https://doi.org/10.1016/j.bspc.2021.103246>
- Janiesch, C., Zschech, P., & Heinrich, K. (2021). *Machine learning and deep learning*. 31, 685–695. <https://doi.org/10.1007/s12525-021-00475-2/Published>
- Julianto, A., Sunyoto, A., Ferry, D., & Wibowo, W. (2022). OPTIMASI HYPERPARAMETER CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI PENYAKIT TANAMAN PADI. 3(2), 98–105. <https://doi.org/https://doi.org/10.46764/teknimedia.v3i2.77>

- Lin, G., & Shen, W. (2018). Research on convolutional neural network based on improved Relu piecewise activation function. *Procedia Computer Science, 131*, 977–984. <https://doi.org/10.1016/j.procs.2018.04.239>
- Liu, J., Xie, Y., Zhang, Y., & Li, H. (2025). Vehicle Flow Detection and Tracking Based on an Improved YOLOv8n and ByteTrack Framework. *World Electric Vehicle Journal, 16*(1), 1–24. <https://doi.org/10.3390/wevj16010013>
- Mahesh, B. (2020). Machine Learning Algorithms - A Review. *International Journal of Science and Research (IJSR), 9*(1), 381–386. <https://doi.org/10.21275/art20203995>
- Muhlashin, M. N. I., & Stefanie, A. (2023). Eye Disease Classification Based On Fundus Image Using Yolo V8 Algorithm Klasifikasi Penyakit Mata Berdasarkan Citra Fundus Menggunakan Algoritma Yolo V8. *Jurnal Media Computer Science, 3*(1), 33–40.
- Nguyen, B.-T., Nguyen, V.-L., & Tran, M.-T. (2023). *Transparent Tracking of Spermatozoa with YOLOv8. 3658*, 1–5. <http://ceur-ws.org>
- Nguyen, H. T. T., Dang, H. N. T., Nguyen, T. T. T., Nguyen, T. Van, Dang, T. C., Nguyen, Q. H. V., & Le, M. T. (2022). Correlations between abnormalities of morphological details and DNA fragmentation in human sperm. *Clinical and Experimental Reproductive Medicine, 49*(1), 40–48. <https://doi.org/10.5653/cerm.2021.04777>
- Opara, J. N., Moriwaki, R., & Chun, P. (2024). Delineating Landslide and Debris Flow Detection in Japan through Aerial Photography: A YOLO v8 Approach to Disaster Management. *Intelligence Informatics and Infrastructure, 5*(1), 111–123.
- Padilla, R., Passos, W. L., Dias, T. L. B., Netto, S. L., & Da Silva, E. A. B. (2021). A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics (Switzerland), 10*(3), 1–28. <https://doi.org/10.3390/electronics10030279>

- Prakash, S., Lakshmi, V., Karthik, M., Sc, G. M., Ganesh, L., Sc, M., & Lakshmanasamy Mudaliar, A. (2014). Morphological diversity of sperm: A mini review. *Iran J Reprod Med*, 12(4), 239–242.
- Pratomo, A. H., Kaswidjanti, W., & Mu'arifah, S. (2020). *IMPLEMENTASI ALGORITMA REGION OF INTEREST (ROI) UNTUK MENINGKATKAN PERFORMA ALGORITMA DETEKSI DAN KLASIFIKASI KENDARAAN*. 7(1), 155–162. <https://doi.org/10.25126/jtiik.202071718>
- Prayitno, A., Chotimah, C., Nugraha, S., & Istanto, T. (2024). *Sistem Keamanan Pintu Laboratorium Perikanan Berbasis Mikrokontroler*. 6(1), 1–12. <https://doi.org/https://doi.org/10.30595/jrre.v6i1.19710>
- Purwono, Ma'arif, A., Rahmaniari, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Haq, Q. M. U. (2022). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, 2(4), 739–748. <https://doi.org/10.31763/ijrcs.v2i4.888>
- Putra, D. S., Azmi, M., Muslikhin, & Purwanto, W. (2022). ANN Activation Function Comparative Study for Sinusoidal Data. *Journal of Physics: Conference Series*, 2406(1), 1–7. <https://doi.org/10.1088/1742-6596/2406/1/012029>
- Qiu, M., Huang, L., & Tang, B. H. (2022). ASFF-YOLOv5: Multielement Detection Method for Road Traffic in UAV Images Based on Multiscale Feature Fusion. *Remote Sensing*, 14(14), 1–22. <https://doi.org/10.3390/rs14143498>
- Ratna, S. (2020). Pengolahan Citra Digital Dan Histogram Dengan Phyton Dan Text Editor Phycharm. *Technologia*, 11(3), 181–186.
- Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. *Computer Vision and Pattern Recognition*, 2, 658–666. <https://doi.org/https://doi.org/10.48550/arXiv.1902.09630>

- Rokhmah, E. Y., Zaman, L., & Pranoto, Y. M. (2020). *Implementasi Content Aware Pada Pembuatan Thumbnail Menggunakan Metode Seam Carving and Salient Detection*. 2(2), 94–101.
- Saputra, H., Muchtar, K., Chitraningrum, N., Andria, A., & Febriana, A. (2025). Performance evaluation of hyper-parameter tuning automation in YOLOV8 and YOLO-NAS for corn leaf disease detection. *Sinergi (Indonesia)*, 29(1), 197–206. <https://doi.org/10.22441/sinergi.2025.1.018>
- Sato, T., Kishi, H., Murakata, S., Hayashi, Y., Hattori, T., Nakazawa, S., Mori, Y., Hidaka, M., Kasahara, Y., Kusuhara, A., Hosoya, K., Hayashi, H., & Okamoto, A. (2022). A new deep-learning model using YOLOv3 to support sperm selection during intracytoplasmic sperm injection procedure. *Reproductive Medicine and Biology*, 21(1), 1–8. <https://doi.org/10.1002/rmb2.12454>
- Sohan, M., Sai Ram, T., & Rami Reddy, Ch. V. (2024). *A Review on YOLOv8 and Its Advancements*. 529–545. [https://doi.org/10.1007/978-981-99-7962-2\\_39](https://doi.org/10.1007/978-981-99-7962-2_39)
- Sun, C., & Tian, X. (2023). Lunar Dome Detection Method Based on Few-Shot Object Detection. *Journal of Physics: Conference Series*, 2560(1), 1–8. <https://doi.org/10.1088/1742-6596/2560/1/012010>
- Tasnim, S., & Qi, W. (2023). Progress in Object Detection: An In-Depth Analysis of Methods and Use Cases. *European Journal of Electrical Engineering and Computer Science*, 7(4), 39–45. <https://doi.org/10.24018/ejece.2023.7.4.537>
- Tian, J., Jin, Q., Wang, Y., Yang, J., Zhang, S., & Sun, D. (2024). Performance analysis of deep learning-based object detection algorithms on COCO benchmark: a comparative study. *Journal of Engineering and Applied Science*, 71(1), 1–18. <https://doi.org/10.1186/s44147-024-00411-z>
- Torres, F. R., Trinidad, J. F. M., & Ochoa, J. A. C. (2022). An Oversampling Method for Class Imbalance Problems on Large Datasets. *Applied Sciences (Switzerland)*, 12(7). <https://doi.org/10.3390/app12073424>

- Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, 1168(2). <https://doi.org/10.1088/1742-6596/1168/2/022022>
- Yuda, R. P., Aroef, C., Rustam, Z., & Alatas, H. (2020). Gender Classification Based on Face Recognition using Convolutional Neural Networks (CNNs). *Journal of Physics: Conference Series*, 1490(1), 1–8. <https://doi.org/10.1088/1742-6596/1490/1/012042>
- Yuzkat, M., Ilhan, H. O., & Aydin, N. (2021). Multi-model CNN fusion for sperm morphology analysis. *Computers in Biology and Medicine*, 137, 1–12. <https://doi.org/10.1016/j.compbiomed.2021.104790>
- Zafar, A., Aamir, M., Mohd Nawawi, N., Arshad, A., Riaz, S., Alruban, A., Dutta, A. K., & Almotairi, S. (2022). A Comparison of Pooling Methods for Convolutional Neural Networks. *Applied Sciences (Switzerland)*, 12(17), 1–21. <https://doi.org/10.3390/app12178643>
- Zhu, J., & Song, J. (2020). An Intelligent Classification Model for Surface Defects on Cement Concrete Bridges. *Applied Sciences (Switzerland)*, 10(3), 2–15. <https://doi.org/10.3390/app10030972>