

PRAKATA

Puji syukur penulis ucapkan kepada Allah SWT, karena atas izin-Nya penulis dapat menyelesaikan skripsi yang berjudul “Judul Skripsi Anda Judul Skripsi Anda Judul Skripsi Anda Judul Skripsi Anda”. Penulis juga mengucapkan terima kasih kepada:

1. Ibu Dr. Erni Tri Astuti M. Math., selaku Direktur Politeknik Statistika STIS;
2. Bapak Ibnu Santoso, SST., M.T., selaku Ketua Program Studi D-IV Komputasi Statistik Politeknik Statistika STIS;
3. Bapak Robert Kurniawan, SST., M.Si, selaku dosen pembimbing yang telah bersedia meluangkan waktu dalam membimbing penyusunan skripsi ini;
4. Ibu Dr. Eng. Lya Hulliyyatus Suadaa SST., MT., dan Bapak Dr. Rindang Bangun Prasetyo S.S.T., M.Si. selaku dosen penguji atas koreksi dan saran yang disampaikan;
5. Orangtua dan keluarga yang telah memberikan banyak dukungan secara moril;
6. Serta semua pihak yang telah memberikan dukungan dalam penulisan skripsi ini.

Penulis menyadari skripsi ini masih mempunyai kekurangan, baik dari isi maupun susunannya. Oleh karena itu, saran dan kritik yang membangun sangat penulis harapkan demi perbaikan skripsi ini. Semoga skripsi ini dapat bermanfaat bagi semua pihak.

Jakarta, 12 Juni 2023

Rizky Theofilus

ABSTRAK

RIZKY THEOFILUS, “Deteksi Kebersihan Sungai dari Sampah menggunakan Convolutional Neural Network dengan Algoritma YOLOv8 (Studi Kasus: Sungai Ciliwung)”.

viii+77 halaman

Meningkatnya jumlah sampah padat di sungai menjadi salah satu masalah utama di daerah perkotaan, karena sungai yang dipenuhi sampah bisa berujung pada berbagai masalah, mulai dari banjir hingga berbagai penyakit. Tujuan dari penelitian ini adalah untuk membangun model image classification menggunakan CNN, menggunakannya untuk mendeteksi jenis sampah mengapung di sungai, dan mengimplementasikan model tersebut terhadap data baru. Algoritma yang digunakan adalah YOLOv8. Data untuk training set dan validation set diambil dari Google Images dan YouTube. Data untuk testing set diambil secara langsung oleh penulis. Sebagian dari data yang dikumpulkan untuk training juga digunakan sebagai data testing. Hasil yang diperoleh adalah epoch terbaik berada di epoch ke-177, dengan nilai Precision sebesar 84.02%, Recall sebesar 91.03%, Accuracy sebesar 77.6%, dan F1-Score sebesar 87.38%. Kemudian, perbandingan performa antara epoch terbaik dan epoch terakhir (epoch ke-200) menunjukkan bahwa epoch terbaik memiliki keunggulan dalam segi performa, dibandingkan dengan epoch terakhir. Kesimpulan yang dapat diambil adalah algoritma YOLOv8 bisa mendeteksi sampah mengapung di permukaan Sungai Ciliwung.

Kata kunci: Convolutional Neural Network, image classification, YOLOv8, sampah mengapung, Sungai Ciliwung.

DAFTAR ISI

	Halaman
PRAKATA.....	i
ABSTRAK	ii
DAFTAR ISI.....	iii
DAFTAR TABEL.....	v
DAFTAR GAMBAR	vi
DAFTAR LAMPIRAN	viii
 BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	6
1.3 Tujuan & Manfaat Penelitian	6
1.4 Batasan Penelitian	7
1.5 Sistematika Penulisan	8
 BAB II KAJIAN PUSTAKA	11
2.1 Landasan Teori.....	11
2.2 Penelitian Terkait	30
2.3 Kerangka Pikir	34
 BAB III METODOLOGI.....	37
3.1 Ruang Lingkup Penelitian.....	37
3.2 Alur Penelitian	37
3.3 Metode Pengumpulan Data	39
3.4 Metode Analisis	41
 BAB IV HASIL DAN PEMBAHASAN	45
4.1 Proses Pengumpulan dan Pengolahan Data	45

4.2	Analisis Hasil Training & Perbandingan Antar Epoch	
	Maksimal	46
4.3	Analisis Hasil <i>Validation</i>	59
4.4	Analisis Hasil Testing	60
BAB IV HASIL DAN PEMBAHASAN		67
5.1	Kesimpulan	67
5.2	Saran	67
DAFTAR PUSTAKA		69
LAMPIRAN		75
RIWAYAT HIDUP		77

DAFTAR TABEL

No. Tabel	Judul Tabel	Halaman
1.	Perbandingan kecepatan algoritma deteksi objek, menggunakan Pascal 2007..	24
2.	Daftar Penelitian.....	30
3.	Prebandingan Hasil Antar Epoch.....	57

DAFTAR GAMBAR

No. Gambar	Judul Gambar	Halaman
1.	Skema Neural Network. Masing-masing lingkaran melambangkan neuron. Neuron terbagi menjadi 3 lapisan, yaitu <i>input layer</i> , <i>hidden layer</i> , dan <i>output layer</i>	16
2.	Skema Feed Forward Neural Network.....	17
3.	Sejarah perkembangan ilmu dan algoritma deteksi objek.....	20
4.	Pemrosesan gambar menggunakan YOLO. (1) Merubah ukuran gambar menjadi 448x448, (2) menjalankan <i>convolutional network</i> terhadap gambar, dan (3) membuat deteksi dan menetapkan ambang batas deteksi berdasarkan <i>confidence</i>	25
5.	Arsitektur YOLOv8.....	27
6.	Alur penelitian	38
7.	Lokasi pengambilan gambar	40
8.	Lokasi dan jangkauan pengambilan gambar (Lat. -6.2312566, Lon. 106,8637216).....	41
9.	Alur pengumpulan dan pemrosesan data.....	45
10.	Results Graph (50 epoch).....	47
11.	Confusion Matrix (50 epoch).....	48
12.	Results Graph (100 epoch).....	50
13.	Confusion Matrix (100 epoch).....	51

14. Results Graph (200 epoch).....	54
15. Confusion Matrix (200 epoch).....	55
16. Val Prediction.....	59
17. Contoh sampah terdeteksi untuk: a. epoch terbaik (epoch ke-177) dan b. epoch terakhir (epoch ke-200).....	61
18. Contoh sampah terdeteksi untuk: a. epoch terbaik (epoch ke-177) dan b. epoch terakhir (epoch ke-200).....	62
19. Contoh sampah terdeteksi untuk: a. epoch terbaik (epoch ke-177) dan b. epoch terakhir (epoch ke-200).....	63
20. Contoh sampah terdeteksi untuk: a. epoch terbaik (epoch ke-177) dan b. epoch terakhir (epoch ke-200).....	64

DAFTAR LAMPIRAN

No. Lampiran	Judul Lampiran	Halaman
1.	Data.....	75
2.	<i>Source Code</i>	75
3.	Dataset Roboflow.....	75
4.	Folder Hasil <i>Training</i> , <i>Validasi</i> , dan <i>Testing</i>	76

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sampah adalah barang buangan atau sisa yang sudah tidak digunakan/dipakai lagi oleh pemiliknya. Secara umum, sampah dibagi menjadi dua, yaitu sampah organik dan sampah anorganik. Sampah organik adalah limbah yang berasal dari sisa makhluk hidup (alam) seperti hewan, manusia, atau tumbuhan yang telah mengalami pembusukan atau pelapukan. Sampah ini tergolong sampah yang ramah lingkungan, karena dapat diuraikan oleh bakteri secara alami dan berlangsung secara cepat. Sementara itu, sampah anorganik adalah sampah yang berasal dari sisa manusia yang sulit untuk di urai oleh bakteri, sehingga membutuhkan waktu yang cukup lama (hingga ratusan tahun) untuk dapat di uraikan. (Fadillah et al., 2019)

Sungai adalah aliran air yang besar dan memanjang yang mengalir secara terus-menerus dari hulu (sumber) menuju hilir (muara). Sungai merupakan tempat mengalirnya air secara gravitasi menuju ke tempat yang lebih rendah. Arah aliran sungai sesuai dengan sifat air mulai dari tempat yang tinggi ke tempat rendah. Sungai bermula dari gunung atau dataran tinggi menuju ke danau atau lautan. (Sulistyowati, 2018)

Sungai merupakan salah satu bagian dari siklus hidrologi. Air dalam sungai umumnya terkumpul dari presipitasi, seperti hujan, embun, mata air, limpasan bawah tanah dan di beberapa negara tertentu juga berasal dari lelehan es atau salju. Selain air, sungai juga mengalirkan sedimen dan polutan, seperti sampah dan

limbah industri. Ekosistem sungai adalah elemen utama dalam siklus air universal, dan sangat penting bagi kehidupan manusia, karena sungai menghubungkan aliran air di darat dengan lingkungan laut (Neo et al., 2022). Yang berarti, sungai berperan sebagai penghubungan antara daratan dan lautan.

Sungai Ciliwung adalah salah satu sungai yang berada di Tatar Pasundan, Pulau Jawa, Indonesia. Panjang aliran utama sungai ini adalah hampir 120 km dengan daerah tangkapan airnya (daerah aliran sungai) seluas 387 km² (Muzaidi, Anggraini, dan Prayugo, 2018). Sungai ini relatif lebar dan di bagian hilirnya dulu dapat dilayari oleh perahu kecil pengangkut barang dagangan. Wilayah yang dilintasi Sungai Ciliwung adalah Kabupaten Bogor, Kota Bogor, Kota Depok, dan Provinsi DKI Jakarta. Ciliwung menjadi salah satu sungai terpenting di Tatar Pasundan karena posisinya yang melalui DKI Jakarta, dan karena menjadi salah satu penyebab banjir tahunan di wilayah hilirnya. (Hendrayanto, 2008)

Salah satu kasus terbaru adalah banjir Sungai Ciliwung yang terjadi pada tahun 2022. Dikutip dari berita yang ditulis di Kompas.com (Sutrisna dan Movanita, 2022), Sungai Ciliwung meluap pada hari Senin, 10 Oktober 2022. Badan Penanggulangan Bencana Daerah (BPBD) mencatat pada tanggal tersebut, terdapat 68 RT di 9 kelurahan yang terendam banjir dengan ketinggian bervariasi, mulai dari 40 cm hingga ada yang mencapai 2.7 meter.

Penyebab banjir dibagi menjadi 2, yaitu faktor alam dan faktor non-alam. Faktor alam terdiri dari curah hujan yang tinggi, kapasitas sungai yang kecil, kapasitas drainase yang kurang memadai, erosi, pengaruh fisiografi, dan pengaruh air pasang. Sementara itu, faktor non-alam terdiri dari adanya kawasan kumuh di bibir sungai, tumpukan sampah, adanya perubahan kondisi DAS (daerah aliran

sungai), terjadi drainase lahan, kerusakan bangunan pengendali air, kerusakan hutan (tidak ada vegetasi alam), serta sistem pengendalian banjir yang tidak tepat. (Yutantri, Suryandari, Putri, dan Widyawati, 2023)

Salah satu penyebab utama banjir di wilayah DKI Jakarta adalah masyarakat yang membuang sampah sembarangan di sungai. Sampah mengapung di sungai bisa menyebabkan air sungai tergenang, menyumbat drainase, dan mengurangi kapasitas sungai. Penyebab banjir yang lain adalah masyarakat membangun rumah di tepi sungai. Hal ini dapat mengakibatkan bekurangnya kapasitas sungai dan daerah resapan. Selain itu, masyarakat yang tinggal di tepi sungai juga akan membuang sampah di sungai, mengakibatkan masalah di atas. Tidak hanya itu, sampah mengapung, seperti plastik, juga bisa menyebabkan masalah bagi fungsi ekosistem dan kesehatan manusia (Isnanto, 2023).

Neural Network (jaringan syaraf) adalah sebuah cabang *machine learning*, yang terinspirasi dari sistem syaraf yang terdapat di otak binatang dan manusia (Hardesty, 2017). Neural Network tersusun oleh titik-titik yang disebut “neuron buatan” yang dihubungkan dengan garis-garis yang mirip seperti sinaps dalam jaringan syaraf otak manusia. Cara kerjanya adalah sebuah neuron menerima sinyal berupa angka riil dari neuron yang terhubung, memproses sinyal tersebut, kemudian mengirimkannya lagi ke neuron lain (Bishop, 2006). CNN merupakan salah satu jenis Neural Network, yang banyak digunakan dalam pemrosesan sinyal dan klasifikasi gambar. Teknik visi komputer banyak didominasi oleh CNN karena akurasi yang tinggi dalam klasifikasi gambar (Lawrence, 1992).

Image classification (klasifikasi gambar) adalah proses memasukkan gambar ke dalam sebuah kategori tertentu, dan memastikan bahwa setiap gambar

memiliki kategori masing-masing. *Image classification* juga dikenal sebagai *object detection* (deteksi objek) (Gavali dan Banu, 2019). Tujuan dari deteksi objek adalah untuk mengembangkan model dan teknik komputasi untuk menyediakan salah satu informasi paling penting yang diperlukan oleh aplikasi visi komputer. Metrik paling penting untuk deteksi objek adalah kecepatan dan akurasi (termasuk akurasi klasifikasi dan akurasi lokalisasi). Seiring perkembangan zaman, munculnya teknik *deep learning* mempercepat kemajuan teknik deteksi objek. (Zou, Chen, Shi, Guo, dan Ye, 2023)

Algoritma YOLO (singkatan dari “You Only Look Once”) adalah algoritma deteksi objek dan segmentasi objek yang dibangun oleh Joseph Redmon dan Ali Farhadi di University of Washington. Pertama kali diluncurkan pada tahun 2015, YOLO menjadi dikenal karena kecepatan dan akurasinya. Algoritma ini didesain untuk memberikan hasil yang memiliki kecepatan dan akurasi tinggi dalam waktunya, serta mudah digunakan dalam berbagai perangkat keras dan aplikasi. YOLOv8 merupakan versi ke-8 dari algoritma YOLO. (Jocher, 2023)

Meningkatnya jumlah sampah padat menjadi salah satu masalah utama di daerah perkotaan. Urbanisasi dan pertumbuhan populasi menjadi indikator utama dalam peningkatan sampah padat, terutama di permukaan air seperti sungai, danau, dan laut (Chen, 2018). Tingkat pencemaran sungai dapat dilihat dan dinilai dari banyaknya sampah pada sungai tersebut. Sampah dan limbah industri juga dapat dikenali oleh mata manusia tanpa bantuan peralatan tambahan, baik dalam segi bentuk objek sampah yang beraneka macam, maupun warna objek sampah yang beraneka ragam. Dengan mengetahui informasi dasar seperti tingkat kebersihan sungai berdasarkan banyaknya sampah di sungai, dinas terkait dapat dengan sigap

sehingga tidak sampai berakibat bencana dan menimbulkan kerugian. Dengan adanya karakteristik visual tersebut maka memungkinkan proses tersebut dapat dikenali oleh komputer, dengan menggunakan pengolahan citra digital. (Setyawan dan Kristian, 2020)

Pada era globalisasi ini, terjadi banyak pencemaran sungai akibat kelalaian manusia yang tidak menjaga lingkungan. membuang sampah di sungai dan membangun rumah di tepi sungai. Jumlah dan jenis sampah mengapung di sungai bisa menggambarkan kualitas sungai, dan bisa dijadikan alat ukur kualitas air sungai. Sampah mengapung yang ada di sungai bisa merusak sungai dan lingkungan sekitarnya, mengurangi keindahan sungai, serta menurunkan kualitas air sungai. Selain itu, sampah mengapung di sungai yang tidak dibersihkan akan terus mengalir hingga ke laut, menyebabkan masalah yang lebih rumit (Jambeck et al, 2015). Padahal, seiring berkembangnya kawasan perkotaan dan meningkatnya jumlah penduduk, warga semakin membutuhkan pasokan air bersih untuk kehidupan sehari-hari. Kondisi ini menuntut masyarakat untuk bisa menjaga sungai agar tetap bersih. Untuk itu, perlu adanya sebuah metode untuk melihat kebersihan sungai dari sampah mengapung, dan untuk mendeteksi sampah dari foto dan video yang diambil. Berdasarkan latar belakang di atas, kami hendak mengajukan penelitian ini, yang kami beri judul “Deteksi Kebersihan Sungai dari Sampah menggunakan Convolutional Neural Network dengan Algoritma YOLOv8 (Studi Kasus: Sungai Ciliwung)”.

1.2 Identifikasi Masalah

Salah satu penyebab banjir adalah tumpukan sampah di sungai, yang menyebabkan sumbatan terhadap sistem drainase dan menyebabkan air meluap. Selain itu, banyaknya sampah mengapung mengindikasikan kebersihan sungai yang kurang baik. Terutama di Sungai Ciliwung, salah satu sungai paling terkenal di DKI Jakarta, karena sering menyebabkan banjir di ibukota. Oleh karena itu, diperlukan penerapan *image classification* terhadap sungai Ciliwung untuk mendeteksi sampah mengapung di permukaan sungai.

Berdasarkan masalah yang telah dikemukakan di atas, masalah yang akan diteliti dalam penelitian ini adalah sebagai berikut:

1. Bagaimanakah cara mendeteksi sampah yang ada di Sungai Ciliwung pada tahun 2020 - 2022 menggunakan *image classification* dengan bantuan algoritma YOLOv8?
2. Bagaimanakah cara mengimplementasikan model yang telah dibangun terhadap data baru yang diperoleh dari sumber data yang berbeda, pada Sungai Ciliwung di tahun 2023?

1.3 Tujuan & Manfaat Penelitian

Berdasarkan identifikasi masalah, maka ditetapkan tujuan dari penelitian ini yaitu sebagai berikut:

- Membangun model *image classification* menggunakan Convolutional Neural Network (CNN) dengan algoritma YOLOv8, dan menggunakannya untuk

mendeteksi sampah mengapung yang ada di Sungai Ciliwung pada tahun 2020 – 2022.

- Mengimplementasikan model yang telah dibangun terhadap data baru yang diperoleh dari sumber data yang berbeda, pada Sungai Ciliwung di tahun 2023.

Kemudian, manfaat yang diharapkan dapat diperoleh dari penelitian ini adalah sebagai berikut:

- Bagi penulis, diharapkan penelitian ini dapat menjadi pembelajaran pribadi untuk melaksanakan penelitian yang lebih baik, dan untuk menulis laporan penelitian yang lebih baik ke depannya.
- Bagi Politeknik Statistika STIS, diharapkan penelitian ini dapat menjadi tulisan yang berguna bagi kemajuan perstatistikaan Indonesia, terutama dalam hal sampah dan kebersihan sungai, serta dapat menjadi sumber atau rujukan bagi penelitian serupa di masa mendatang.
- Bagi pemerintah setempat, diharapkan penelitian ini dapat menjadi acuan dalam menyusun metode penanggulangan tumpukan sampah di sungai.

1.4 Batasan Penelitian

Batasan penelitian perlu ditetapkan agar penelitian ini tidak melenceng dari masalah yang dirumuskan, sehingga dapat memenuhi tujuan yang ditetapkan dan dapat memberikan manfaat yang diharapkan. Dalam penelitian ini, batasan penelitian yang akan digunakan adalah sebagai berikut:

- Studi kasus penelitian dibatasi pada bagian Sungai Ciliwung di Jakarta Timur, terutama bagian sungai yang berada di belakang kampus Politeknik Statistika STIS.
- Data yang digunakan adalah data foto dan video yang menampilkan sampah yang mengapung di permukaan sungai. Data diperoleh dari media daring, atau diambil langsung oleh penulis menggunakan *smartphone* pribadi. Media daring yang digunakan adalah Google Images untuk data berupa foto, dan YouTube untuk data berupa video.
- Metode yang digunakan adalah *Image Processing* menggunakan Convolutional Neural Network (CNN). Bahasa pemrograman yang digunakan adalah bahasa pemrograman Python. Pemrograman dilakukan di aplikasi web Google Colab.
- Algoritma yang digunakan adalah algoritma YOLOv8.
- Alat *labeling & preprocessing* yang digunakan adalah Roboflow.

1.5 Sistematika Penulisan

Penulisan skripsi ini terdiri dari lima bab yang merujuk pada pedoman penulisan skripsi di Politeknik Statistika STIS. Bagian awal buku skripsi memuat Halaman Sampul, Halaman Judul, Halaman Pernyataan, Halaman Pengesahan, Prakata, Abstrak, Daftar Isi, Daftar Tabel, Daftar Gambar, dan Daftar Lampiran. Bagian isi terdiri atas lima bab, yaitu Bab Pendahuluan, Bab Kajian Pustaka, Bab Metodologi, Bab Hasil dan Pembahasan, serta Bab Kesimpulan dan saran.

Bab I adalah Pendahuluan yang akan menjelaskan latar belakang penelitian, identifikasi masalah, tujuan penelitian, batasan penelitian dan sistematika penulisan.

Bab II adalah Kajian Pustaka yang terdiri dari landasan teori yang menjadi dasar penelitian, penelitian terkait, serta kerangka pikir penelitian.

Bab III adalah Metodologi yang terdiri area studi, metode pengumpulan data, metode pelabelan, fitur yang digunakan, dan metode analisis data yang dibagi berdasarkan tujuan penelitian.

Bab IV adalah Hasil dan Pembahasan yang terdiri dari hasil analisis, serta pemaparan pembahasan yang telah didapatkan

Bab terakhir yaitu Bab V adalah Kesimpulan dan Saran yang terdiri dari kesimpulan dari hasil dan pembahasan, serta rekomendasi berupa saran dari penulis bagi para penerima manfaat penelitian ini. Selain itu, sumber-sumber lain berupa buku, jurnal, hasil penelitian, dan bahan referensi lainnya yang menjadi rujukan dalam penelitian ini disajikan dalam daftar pustaka.

“...sengaja dikosongkan...”

BAB II

KAJIAN PUSTAKA

2.1 Landasan Teori

Sampah

Sampah adalah barang buangan atau sisa yang sudah tidak digunakan/dipakai lagi oleh pemiliknya. Menurut Konvensi Basel, sampah adalah “bahan atau barang yang telah dibuang, atau direncanakan akan dibuang, atau diwajibkan oleh hukum negara untuk dibuang” (Basel Convention, 1989). Sementara itu, menurut Divisi Statistik PBB, UNSD Glossary of Environment Statistics menyatakan sampah sebagai “barang yang bukan produk utama (untuk dipasarkan) dan tidak lagi dibutuhkan oleh produsen, baik untuk tujuan produksi, transformasi, atau konsumsi, serta ingin dibuang oleh produsen. Sampah bisa berupa sesuatu yang dihasilkan selama proses pengambilan bahan mentah, proses merubah bahan mentah menjadi barang setengah jadi atau barang jadi, proses konsumsi barang jadi, atau aktivitas manusia lain; tidak termasuk barang yang didaur ulang atau digunakan ulang di lokasi produksi” (UNSD, 1997).

Pada dasarnya, sampah adalah sesuatu yang subjektif. Walaupun sampah adalah benda fisik, membuat sampah adalah proses fisik dan psikis. Apa yang dianggap sebagai sampah bagi seseorang bisa saja merupakan sesuatu yang bermanfaat bagi orang lain. Ada beberapa jenis sampah yang masih memiliki manfaat, seperti daur ulang sampah untuk menghasilkan barang baru, atau digunakan kembali untuk hal lain. (Doron, 2018)

Namun, sampah juga memiliki dampak terhadap lingkungan, terutama jika tidak dikelola dengan baik. Saat ini sampah merupakan masalah lingkungan yang sangat serius, karena selalu ada sampah yang dihasilkan setiap harinya oleh penduduk Indonesia. Yang menjadi masalah adalah sampah-sampah tersebut malah dibuang sembarangan di berbagai tempat, yang akan berakibat merusak lingkungan sekitar. Jumlah produksi sampah setiap tahun akan bertambah seiring dengan bertambahnya jumlah penduduk. Terlebih lagi, jumlah sampah di Indonesia tergolong sangat tinggi. (Fadillah et al., 2019)

Secara umum, sampah dibagi menjadi dua, yaitu sampah organik dan sampah anorganik. Berikut adalah pembagiannya: (Fadillah et al., 2019)

- Sampah organik adalah limbah yang berasal dari sisa makhluk hidup (alam) seperti hewan, manusia, atau tumbuhan yang telah mengalami pembusukan atau pelapukan. Sampah ini tergolong sampah yang ramah lingkungan, karena dapat diuraikan oleh bakteri secara alami dan berlangsung secara cepat.

Contoh: sampah daun, ranting pohon, bangkai hewan, sisa makanan

- Sampah anorganik adalah sampah yang berasal dari sisa barang buatan manusia. Sampah anorganik sulit untuk diurai oleh bakteri, sehingga membutuhkan waktu yang cukup lama (bahkan hingga ratusan tahun) untuk dapat di uraikan. Ada beberapa jenis sampah anorganik yang berbahaya bagi kesehatan manusia, terutama jika ditangani dengan cara yang tidak sesuai. Contohnya adalah membakar sampah plastik atau baterai bekas, yang dapat menyebabkan masalah pernafasan. Selain itu, kebiasaan membuang sampah sembarangan juga menyebabkan masalah pencemaran lingkungan. Karena sampah anorganik sulit

diurai, maka sampah tersebut tidak akan hilang dengan sendirinya dan dapat menjadi sumber dari berbagai macam masalah.

Contoh: sampah plastik, kertas, Styrofoam, besi, kaca, kaleng, kain

Sungai

Sungai adalah aliran air yang besar dan memanjang yang mengalir secara terus-menerus dari hulu (sumber) menuju hilir (muara). Sungai merupakan salah satu bagian dari siklus hidrologi. Air dalam sungai umumnya terkumpul dari presipitasi, seperti hujan, embun, mata air, limpasan bawah tanah dan di beberapa negara tertentu juga berasal dari lelehan es atau salju. Selain air, sungai juga mengalirkan sedimen dan polutan, seperti sampah dan limbah industri. (Sulistiyowati, 2018)

Sungai merupakan salah satu bagian dari siklus hidrologi. Air dalam sungai umumnya terkumpul dari presipitasi, seperti hujan, embun, mata air, limpasan bawah tanah dan di beberapa negara tertentu juga berasal dari lelehan es atau salju. Selain air, sungai juga mengalirkan sedimen dan polutan, seperti sampah dan limbah industri. Ekosistem sungai adalah elemen utama dalam siklus air universal, dan sangat penting bagi kehidupan manusia, karena sungai menghubungkan aliran air di darat dengan lingkungan laut (Neo et al., 2022). Yang berarti, sungai berperan sebagai penghubung antara daratan dan lautan.

Sungai Ciliwung adalah salah satu sungai terpenting di Tatar Pasundan, Pulau Jawa, Indonesia. Wilayah yang dilintasi Sungai Ciliwung adalah Kabupaten Bogor, Kota Bogor, Kota Depok, dan Provinsi DKI Jakarta. Sungai ini terkenal,

terutama karena sungai ini melalui wilayah ibu kota, DKI Jakarta, dan kerap menimbulkan banjir tahunan di wilayah hilirnya. (Hendrayanto, 2008)

Banjir

Banjir adalah suatu peristiwa yang terjadi akibat menumpuknya air yang jatuh dan tidak dapat di tampung oleh tanah. Peristiwa ini terjadi karena air yang jatuh ke dataran tidak memiliki daerah tangkapan yang baik. Pada umumnya, banjir terjadi di daerah perkotaan, karena kebutuhan akan ruang perkotaan semakin meningkat sebagai akibat dari meningkatnya aktivitas sosial-ekonomi masyarakat, namun luasan ruang yang ada sangat terbatas, semakin langka dan mahal. Hal ini kemudian mendorong pemanfaatan ruang yang bersifat memaksimalkan kondisi ruang yang ada, termasuk membangun di tempat yang tidak seharusnya dibangun, seperti di bibir sungai. Selain itu, menurunnya ketinggian permukaan tanah yang disebabkan penyerapan air tanah yang berlebihan, ditambah curah hujan yang tinggi terutama di musim penghujan namun tidak terserap ke tanah, membuat banjir semakin parah dan semakin sering (Yohana, Griandini, dan Muzambeq, 2017). Banjir merupakan bencana yang sering terjadi di Indonesia, terutama di saat musim penghujan.

Faktor penyebab banjir dibagi menjadi 2, yaitu faktor alam dan faktor non-alam. (Yutantri et al., 2023, hal 200)

- Faktor alam adalah faktor penyebab banjir yang berasal dari alam, tanpa pengaruh dari kegiatan manusia. Faktor alam terdiri dari curah hujan yang tinggi, kapasitas sungai yang kecil, kapasitas drainase yang kurang memadai, erosi, pengaruh fisiografi, dan pengaruh air pasang. Peningkatan curah hujan dan cuaca

ekstrem yang disebabkan oleh perubahan iklim terutama mampu meningkatkan tingkat kerusakan yang disebabkan oleh banjir, karena perubahan iklim bisa berujung banjir yang lebih parah atau lebih sering (Hirabayashi et al., 2013)

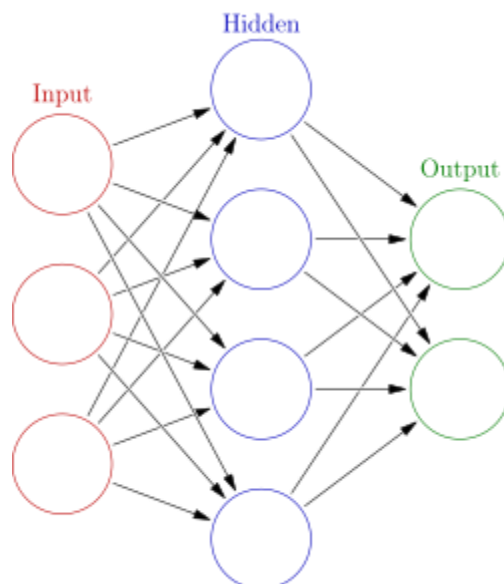
- Faktor non-alam adalah faktor penyebab banjir yang disebabkan oleh kegiatan manusia yang mempengaruhi alam. Faktor non-alam terdiri dari adanya kawasan kumuh di bibir sungai, tumpukan sampah di sungai, adanya perubahan kondisi DAS (daerah aliran sungai), terjadi drainase lahan, kerusakan bangunan pengendali air, kerusakan hutan yang mengakibatkan tidak adanya vegetasi alam, serta sistem pengendalian banjir yang tidak tepat. Perubahan lingkungan yang disebabkan oleh manusia seringkali meningkatkan intensitas dan frekuensi banjir, seperti penebangan hutan, berkurangnya tanah basah, atau perubahan jalur air (Hirabayashi et al., 2013).

Neural Network (NN)

Neural Network (jaringan syaraf), juga dikenal sebagai Neural Net atau Artificial Neural Network (ANN, jaringan syaraf buatan) adalah sebuah cabang *machine learning*, yang terinspirasi dari sistem syaraf yang terdapat di otak binatang dan manusia (Hardesty, 2017). Neural Network merupakan sebuah sistem yang dapat mengubah strukturnya untuk memecahkan masalah berdasarkan informasi yang mengalir melalui jaringan tersebut. Oleh karena sifatnya yang adaptif tersebut, Neural Network juga sering disebut dengan jaringan adaptif (Nasution, Harumy, Haryanto, Fachrizal, Julham, dan Turnip, 2015). Neural Network digunakan untuk model prediksi, kontrol adaptif, dan aplikasi lain yang

bisa di-*train* menggunakan dataset. Selain itu, Neural Network juga digunakan untuk menyelesaikan masalah dalam kecerdasan buatan (artificial intelligence).

Neural Network tersusun oleh titik-titik yang disebut “neuron buatan” yang dihubungkan dengan garis-garis yang mirip seperti sinaps dalam jaringan syaraf otak manusia. Cara kerjanya adalah sebuah neuron menerima sinyal berupa angka riil dari neuron yang terhubung, memproses sinyal tersebut, kemudian mengirimkannya lagi ke neuron lain. Biasanya neuron terbagi menjadi 3 lapisan: lapisan masukan (*input layer*), lapisan keluaran (*output layer*), dan lapisan tersembunyi (*hidden layer*). Antar lapisan bisa memiliki fungsi yang berbeda-beda. Sinyal berjalan mulai dari lapisan masukan dan keluar terakhir dari lapisan keluaran, dan mungkin melalui beberapa lapisan tersembunyi. Sebuah jaringan disebut jaringan dalam (deep neural network) jika memiliki lebih dari 2 lapisan tersembunyi (Bishop, 2006).



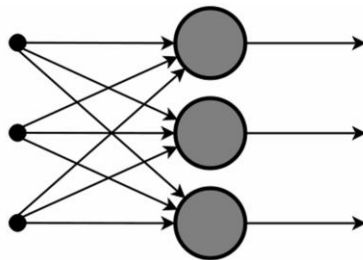
Sumber: Glosser.ca (Wikipedia)

Gambar 1. Skema Neural Network. Masing-masing lingkaran melambangkan neuron. Neuron terbagi menjadi 3 lapisan, yaitu *input layer*, *hidden layer*, dan *output layer*

Ada beberapa jenis Neural Network yang berbeda satu sama lain, yaitu sebagai berikut: (Islam, Chen, dan Jin, 2019)

- Feed Forward Neural Network.

Ini adalah bentuk paling sederhana dari Neural Network, dimana data input bergerak dalam satu arah, dari lapisan input ke lapisan output, dan mungkin melalui lapisan tersembunyi. Responsif terhadap *noise* dan mudah dirawat, Feed Forward NN digunakan dalam visi komputer dan pengenalan suara, dimana proses klasifikasi kelas target adalah sesuatu yang rumit (Lippmann, 1987). Namun, karena sifatnya yang menghubungkan satu neuron di satu lapisan dengan semua neuron di lapisan berikutnya, Feed Forward NN sangat rentan terhadap overfitting (Kurtzman, 2019).



Sumber: Islam, Chen, dan Jin, 2019

Gambar 2. Skema Feed Forward Neural Network

- Radial Basis Function Neural Network

Dalam Radial Basis Function, jarak antar neuron diperhitungkan. Pada umumnya jarak yang digunakan adalah jarak Euclidean, namun jarak lain juga bisa digunakan. Neuron dengan jarak terdekat ke pusat diutamakan. Model ini banyak digunakan dalam sistem pemulihan tenaga (listrik). (Murphy, 2012)

- Kohonen Self-Organizing Neural Network

Tujuan dari peta Kohonen adalah memetakan vektor yang berasal dari beragam dimensi, menjadi satu peta diskrit yang terbantu dari neuron. Dalam peta tersebut, neuron yg paling dekat dengan neuron lain akan saling mendekat membentuk kluster. Proses ini akan terus berulang sampai semua neuron menjadi anggota satu kluster. Kohonen NN biasa digunakan untuk menemukan pola dalam data. (Hubel dan Wiesel, 1959)

- Recurrent Neural Network (RNN) – Long Short Term Memory

Prinsip utama RNN adalah memasukkan kembali output sebuah lapisan ke inputnya untuk memprediksi keluaran baru. Jika prediksi yang dihasilkan salah, maka bisa diperbaiki dengan cepat.

- Convolutional Neural Network (CNN)

CNN, atau yang juga dikenal sebagai Shift Invariant/Space Invariant Artificial Neural Network (SIANN), adalah versi regularisasi dari Feed Forward NN. CNN bisa membatalkan gradien menghilang dan gradien meledak, dengan memilih menggunakan beban standar daripada koneksi yang lebih sedikit. CNN juga bisa mengatasi masalah overfitting dari Feed Forward NN, dengan cara membatasi parameter selama training, atau mengurangi konektivitas (Kurtzman, 2019).

CNN banyak digunakan dalam pemrosesan sinyal dan klasifikasi gambar. Teknik visi komputer banyak didominasi oleh CNN karena akurasi yang tinggi dalam klasifikasi gambar (Lawrence, 1992). Bukan hanya itu, CNN juga diaplikasikan dalam pengenalan gambar dan video (Valueva, Nagornov, Lyakhov, Valuev, dan Chervyakov, 2020), sistem rekomendasi (van den Oord et al., 2013),

segmentasi gambar, analisis gambar medis, Natural Language Processing (Collobert dan Weston, 2008), antarmuka otak-komputer (Avilov, Rimbert, Popov, dan Bougrain, 2020), dan runtun waktu finansial (Tsantekidis, Passalis, Tefas, Kannianen, Gabbouj, dan Iosidifis, 2017).

- **Modular Neural Network**

Modular NN berupa gabungan beberapa jaringan yang bekerja masing-masing, dan berkontribusi terhadap output. Setiap jaringan memiliki inputnya masing-masing yang berbeda dari jaringan lain, dan satu jaringan tidak akan memberitahu jaringan lain mengenai tugasnya. Kelebihan Modular NN adalah bisa membagi tugas yang besar menjadi bagian-bagian kecil, yang berarti menurunkan kompleksitas tugas, mengurangi koneksi, dan menghilangkan interaksi antar-jaringan, yang berujung pada peningkatan kecepatan komputasi. Akan tetapi, kecepatan komputasi masih bergantung pada jumlah neuron dan keterlibatan masing-masing neuron dalam menghitung hasil.

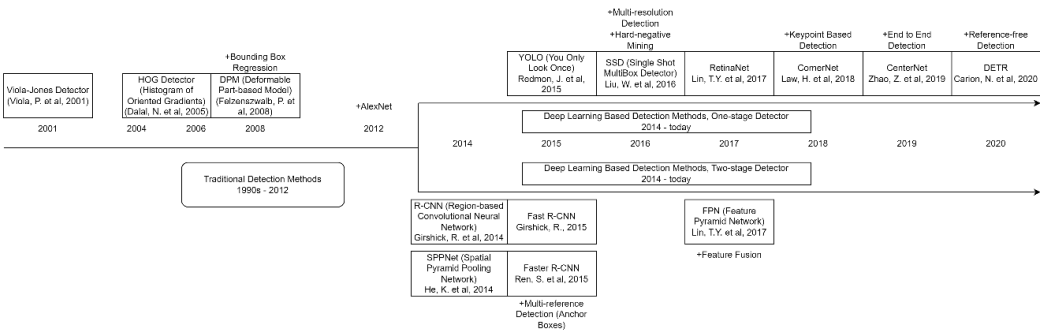
Image Classification (Klasifikasi Gambar)

Otak manusia dapat dengan mudah mengklasifikasikan gambar. Namun klasifikasi yang demikian sulit dilakukan oleh komputer, terutama jika gambar tersebut memiliki *noise* (elemen dalam gambar yang memengaruhi gambar, sebagian maupun keseluruhan). *Image processing* (pemrosesan gambar) adalah proses digitalisasi suatu keadaan/pemandangan berupa gambar, dan melakukan sebuah operasi/metode untuk mengambil informasi yang berguna dari gambar tersebut. *Image classification* (klasifikasi gambar) adalah *image processing* dalam jangkauan yang lebih luas. *Image classification* berarti proses memasukkan gambar

ke dalam sebuah kategori tertentu, dan memastikan bahwa setiap gambar memiliki kategori masing-masing. *Image classification* juga dikenal sebagai *object detection* (deteksi objek). (Gavali dan Banu, 2019)

Tujuan dari deteksi objek adalah untuk mengembangkan model dan teknik komputasi untuk menyediakan salah satu informasi paling penting yang diperlukan oleh aplikasi visi komputer. Metrik paling penting untuk deteksi objek adalah kecepatan dan akurasi (termasuk akurasi klasifikasi dan akurasi lokalisasi). Deteksi objek berperan sebagai dasar dari tugas visi komputer lainnya, seperti segmentasi, *image captioning* (membuat deskripsi tertulis dari sebuah gambar), *object tracking* (mendeteksi objek dan mengikuti arah gerakannya), dsb. Seiring perkembangan zaman, munculnya teknik *deep learning* mempercepat kemajuan teknik deteksi objek. (Zou et al, 2023)

Dalam dua dekade terakhir, telah diterima secara luas bahwa perkembangan teknik deteksi objek dibagi menjadi dua periode utama, yaitu periode metode tradisional (sebelum 2014) dan periode metode berbasis *deep learning* (setelah 2014). Berikut adalah pembagian metode yang paling berpengaruh dalam setiap periode: (Zou et al, 2023)



Sumber: Zou et al., 2023, dengan perubahan

Gambar 3. Sejarah perkembangan ilmu dan algoritma deteksi objek

- Metode Tradisional: metode yang dibangun menggunakan fitur yang dibuat sendiri
 - a. Viola-Jones Detector (VJ), dikemukakan pada tahun 2001, adalah algoritma yang pertama kali berhasil mendeteksi wajah manusia, tanpa ada gangguan seperti segmentasi warna kulit (Viola et al., 2001)
 - b. HOG Detector (Histogram of Oriented Gradients), dikemukakan pada tahun 2005, merupakan dasar bagi banyak model deteksi objek dan aplikasi visi komputer (Dalal et al., 2005)
 - c. DPM Detector (Deformable Part-based Model), dikemukakan pada tahun 2008 sebagai kelanjutan dari metode HOG, merupakan puncak dari deteksi objek dengan metode tradisional. Walaupun akurasi tidak lagi sebagai metode deteksi objek berdasarkan *deep learning*, dasar pemikirannya masih banyak digunakan (Felzenszwalb et al., 2008)
- 3. Metode Berbasis Deep Learning: metode yang didasarkan pada penggunaan *deep learning* dan CNN
 - a. One-stage Detector: deteksi satu langkah
 - 1) YOLO (You Only Look Once), dikemukakan pada tahun 2015, merupakan model deteksi satu langkah pertama di zaman *deep learning*. Terkenal karena kecepatannya, namun akurasi lokalisasinya bisa turun terutama untuk objek berukuran kecil. Masalah ini diselesaikan dengan versi baru, dan dengan metode SSD (Redmon et al., 2015)
 - 2) SSD (Single Shot MultiBox Detector), dikemukakan pada tahun 2016, memperkenalkan teknik multi-reference dan multi-resolution yang

mampu meningkatkan akurasi deteksi, terutama untuk objek berukuran kecil (Liu et al., 2016)

- 3) RetinaNet, dikemukakan pada tahun 2017, memperkenalkan “focal loss” yang memungkinkan detektor satu langkah untuk meraih akurasi yang setara dengan detektor dua langkah, dengan kecepatan yang tidak berubah (Lin et al., 2017)
- 4) CornerNet, dikemukakan pada tahun 2018. Dengan pendekatan “sudut pojok kotak”, CornerNet mampu melampaui semua detektor satu langkah pada masanya (Law et al., 2018)
- 5) CenterNet, dikemukakan pada tahun 2019. Berbeda dengan CornerNet, CenterNet menganggap pusat objek sebagai sebuah titik, dan mengabaikan atribut lain (Zhao et al., 2019)
- 6) DETR, dikemukakan pada tahun 2020, menggunakan Transformer sebagai dasarnya. Ini menandakan era baru deteksi objek, tanpa menggunakan *anchor box* atau *anchor point* (Carion et al., 2020)

b. Two-stage Detector: deteksi dua langkah

- 1) R-CNN (Region-based Convolutional Neural Network), dikemukakan pada tahun 2014, dilakukan dengan cara membagi gambar menjadi bagian-bagian. R-CNN mengalami peningkatan nilai mAP dibandingkan dengan DPMv5, namun waktu deteksi yang dibutuhkan sangat lama (Girshick et al., 2014)
- 2) SPPNet (Spatial Pyramid Pooling Network), dikemukakan pada tahun 2014 sebagai solusi dari masalah yang dimiliki oleh R-CNN,

meningkatkan kecepatan tanpa mengorbankan akurasi deteksi (He et al., 2014)

- 3) Fast R-CNN, dikemukakan pada tahun 2015 sebagai kelanjutan dari R-CNN dan SPPNet, mengambil kelebihan dari masing-masing metode ditambah dengan kemampuan untuk melatih detector dan bounding box regressor dalam satu jaringan. Namun kecepatan deteksi masih dipengaruhi oleh deteksi daerah (Girshick, 2015)
- 4) Faster R-CNN, dikemukakan pada tahun 2015 sebagai solusi dari masalah yang dimiliki oleh Fast R-CNN, merupakan metode pertama yang hampir mencapai realtime. Dari R-CNN ke Faster R-CNN, sebagian besar bagian dari sistem deteksi objek digabungkan menjadi satu framework. (Ren et al., 2015)
- 5) FPN (Feature Pyramid Network), dikemukakan pada tahun 2017. FPN menjadi tumpuan untuk membangun model deteksi objek baru ke depannya (Lin et al., 2017)

YOLO

YOLO (“You Only Look Once”) adalah sebuah model deteksi objek dan segmentasi gambar yang terkenal karena kecepatan dan akurasi. Model ini dibangun oleh Joseph Redmon dan Ali Farhadi di University of Washington. YOLO diciptakan sebagai jawaban atas kelemahan yang dimiliki oleh teknologi deteksi objek yang tersedia. Termasuk metode termutakhir pada waktu itu, yang menggunakan R-CNN (Region-based Convolutional Neural Network), Fast R-

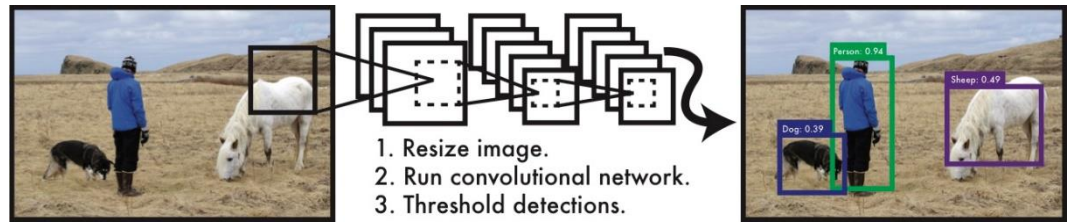
CNN, bahkan Faster R-CNN mendapat kritik karena alur deteksi yang digunakan lambat dan sulit dioptimisasi.

Tabel 1. Perbandingan kecepatan algoritma deteksi objek, menggunakan Pascal 2007

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img

Sumber: Redmon et al., 2015, dengan perubahan

Dengan algoritma YOLO, komputer hanya perlu melihat gambar sekali (“you only look once”) untuk memprediksi objek apa yang ada di gambar, lokasinya, lalu membuat bounding box di sekitar objek. Dengan hanya menggunakan satu convolutional network untuk prediksi bounding box dan probabilitas kelas, YOLO memiliki beberapa keunggulan dari metode deteksi objek lainnya. Pertama, YOLO sangat cepat. YOLO hanya perlu menjalankan neural network terhadap gambar selagi testing berjalan. Kedua, YOLO melihat keseluruhan gambar dalam membuat prediksi. YOLO bisa menghindari kesalahan mendeteksi latar belakang sebagai objek karena kekurangan konteks, lebih baik daripada metode Fast R-CNN. Dan yang ketiga, YOLO dapat mempelajari representasi umum dari objek, terutama jika di-train menggunakan karya seni atau foto alam. (Redmon et al., 2015)



Sumber: Redmon et al., 2015, dengan perubahan

Gambar 4. Pemrosesan gambar menggunakan YOLO. (1) Merubah ukuran gambar menjadi 448x448, (2) menjalankan *convolutional network* terhadap gambar, dan (3) membuat deteksi dan menetapkan ambang batas deteksi berdasarkan *confidence*

Model ini terus mengalami perkembangan setiap tahunnya, mengikuti kebutuhan dan perkembangan ilmu deteksi objek dan segmentasi. Sejarah perkembangan YOLO adalah sebagai berikut: (Terven, Córdova-Esparza, & Romero-González, 2023)

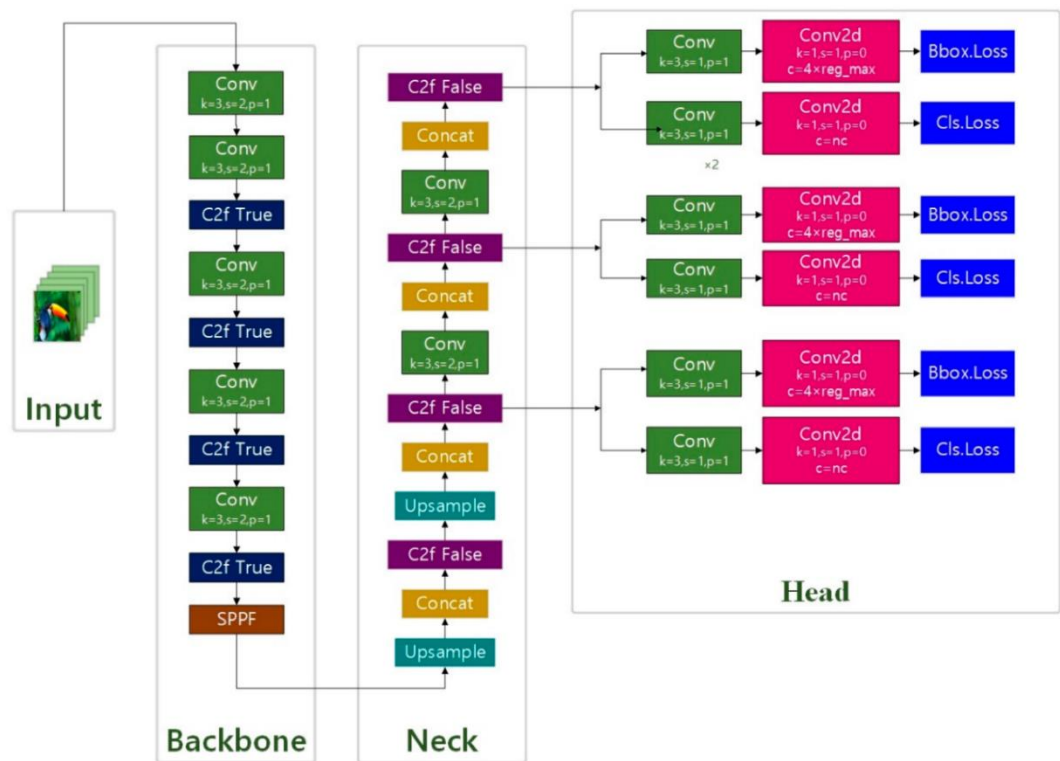
- YOLO, versi pertama yang diluncurkan pada tahun 2016.
- YOLOv2, versi kedua yang diluncurkan pada tahun 2017. Versi ini merupakan pengembangan dari versi sebelumnya, dengan menerapkan normalisasi *batch*, kotak jangkar, dan klaster dimensi.
- YOLOv3, versi ketiga yang diluncurkan pada tahun 2018. Versi ini menggunakan jaringan *backbone* yang lebih efisien, menggunakan banyak jangkar sekaligus, dan menerapkan *spatial pyramid pooling*.
- YOLOv4, versi keempat yang diluncurkan pada tahun 2020, oleh Bochkovskiy, Wang, & Liao. Versi ini memperkenalkan inovasi baru, seperti augmentasi data mosaik, kepala deteksi bebas-jangkar yang baru, serta *loss function* yang baru.
- YOLOv5, versi kelima yang dirilis oleh Jocher, pendiri Ultralytics. Versi ini lebih berkembang daripada sebelumnya, dan memiliki fitur baru seperti

optimisasi *hyperparameter*, eksperimen *tracking* terpadu, dan ekspor otomatis menjadi format yang sering digunakan.

- Scaled-YOLOv4, versi alternatif dari YOLOv4 yang dikembangkan di Pytorch, tidak seperti YOLOv4 yang dikembangkan di Darknet. Tujuan utama dari versi ini adalah memperkenalkan *scaling*, untuk menyesuaikan performa dengan kemampuan perangkat. Versi *scale-up* (peningkatan) dikenal dengan YOLOv4-large, didesain untuk digunakan oleh *cloud GPU*. Sementara itu versi *scale-down* (penurunan) dikenal dengan YOLOv4-tiny, didesain untuk GPU yang berkemampuan rendah, bisa digunakan dengan 46 FPS.
- YOLOv6, versi keenam yang diluncurkan pada tahun 2022 secara open-source dengan bantuan Meituan. Versi ini banyak digunakan oleh robot pesan antar nir-awak yang digunakan oleh beberapa perusahaan.
- YOLOv7, versi ketujuh yang dilengkapi dengan tugas tambahan, seperti estimasi pose dalam dataset titik COCO.
- YOLOv8, versi terbaru yang diluncurkan pada tahun 2023 oleh Ultralytics. Model ini dibangun di atas kesuksesan versi-versi sebelumnya. Model ini berisi fitur baru, serta peningkatan performa, fleksibilitas, dan efisiensi. YOLOv8 didesain sebagai algoritma yang cepat, akurat, dan mudah digunakan, cocok untuk mendukung banyak tugas kecerdasan buatan, seperti deteksi, segmentasi, estimasi pose, tracking, dan klasifikasi. Ada 5 versi *scaling* YOLOv8, diurutkan dari yang terkecil, yaitu YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), & YOLOv8x (extra large) (Jocher, Chaurasia, & Qiu, 2023)

- YOLOv8.1, peningkatan terhadap v8 yang diluncurkan pada tahun 2024. Versi ini berisi penerapan model OBB (Oriented Bounding Box), peningkatan performa segmentasi, optimisasi performa, peningkatan arsitektur model dan fitur training, serta peningkatan API & CLI.

Berikut adalah asitektur yang diterapkan dalam algoritma YOLOv8 (Wang, Liu, Li, Zhou, & Ding, 2023):



Sumber: Wang, Liu, Li, Zhou, & Ding, 2023

Gambar 5. Arsitektur YOLOv8

Struktur algoritma YOLOv8 dibagi menjadi 3 bagian yaitu tulang belakang (*backbone*), leher (*neck*), dan kepala (*head*). *Backbone* bertugas untuk mengambil fitur-fitur penting dari input. Pada umumnya *backbone* adalah sebuah CNN yang telah dilatih untuk tugas klasifikasi berskala besar, seperti ImageNet. Fitur penting

tersebut diambil dalam tingkatan yang berbeda-beda, dimana fitur tingkat rendah (pojok gambar, tekstur) diambil di lapisan awal, dan fitur tingkat tinggi (bagian objek, informasi semantik) diambil di lapisan akhir.

Neck adalah penghubung antara *backbone* dan *head*, berfungsi untuk merata-ratakan dan memperbaiki informasi spasial dan semantik pada input yang telah diambil oleh *backbone*, serta mengantarkannya pada *head*. Neck bisa mengandung *convolutional layer* tambahan, FPN, atau mekanisme lain yang dapat membantu meningkatkan representasi fitur-fitur tersebut.

Head merupakan bagian terakhir dalam algoritma deteksi objek, yang bertugas untuk membuat prediksi deteksi objek berdasarkan informasi yang diperoleh dari *backbone* dan *neck*. Pada umumnya *head* tersusun dari subjaringan yang memiliki tugas masing-masing, seperti klasifikasi, lokalisasi, segmentasi, dan estimasi pose.

Sesuai gambar di atas, input dimasukkan terlebih dahulu ke bagian backbone, yang berisi *convolutional layer* dan *True C2f layer*, untuk ekstraksi fitur. Kemudian, fitur yang telah diperoleh diarahkan ke bagian neck, yang berisi upsampling layer untuk peningkatan fitur, *convolutional layer*, *concat layer*, dan *False C2f layer*. Terakhir, fitur yang telah ditingkatkan diarahkan ke bagian head, yang berisi *convolutional layer* dan *2d convolutional layer*, untuk pembuatan prediksi dan menghitung *loss*, sebelum hasil prediksi ditampilkan. (Terven, Córdova-Esparza, & Romero-González, 2023)

Roboflow

Roboflow adalah alat bantu untuk model visi komputer. Roboflow diluncurkan pada Januari 2020, dan saat ini telah digunakan oleh lebih dari 250 ribu developer untuk membuat dataset, melatih model, dan *deploy*. Roboflow mendorong developer untuk membangun aplikasi visi komputer, dengan cara membantu mempermudah proses membangun aplikasi, terutama labeling dan training (Roboflow, 2023).

Roboflow memungkinkan edit dan manipulasi dataset menjadi lebih efisien, menggunakan bantuan AI untuk membuat bounding box, polygon, dan segmentasi. Roboflow menerima berbagai macam format input gambar dan video, seperti JPG, PNG, BMP, MOV, MP4, dan AVI, yang mempermudah impor gambar untuk pembuatan dataset. Selain itu, Roboflow juga bisa mengekspor dataset dengan berbagai macam format juga, seperti COCO JSON, VGG, Vott JSON, Marmot XML, YOLO PyTorch, YOLO Darknet TXT, dan Kaggle CSV, yang mempermudah dataset digunakan di berbagai macam framework. Salah satu kelebihan utama Roboflow adalah antarmuka yang menarik dan pengaturan dasar yang cerdas, sehingga labeling bisa dilakukan dengan cepat dan akurat, tanpa perlu latihan mendalam atau pengetahuan khusus. Ditambah lagi, Roboflow memiliki protokol privasi dan keamanan yang ketat, sehingga data yang tersimpan terjamin aman dan terlindungi dengan baik. (Shandilya, Srivastav, Yemets, Datta, dan Nagar, 2023)

Roboflow dapat digunakan untuk membantu pembuatan model, seperti:

- Membuat dataset, terutama yang berhubungan dengan data visual. Lakukan filtering, tagging, segmentation, preprocessing, augmentation, dan melihat versi sebelumnya dari dataset.
- Membantu labeling dengan cepat dan mudah, dengan model pre-trained dan SAM untuk labeling otomatis, serta histori labeling.
- Menyediakan alat untuk membangun model yang akurat, termasuk GPU untuk training model, serta lebih dari 50 ribu model pre-trained untuk didasarkan.
- Membantu deploy aplikasi yang telah dibangun ke mana saja, termasuk NVIDIA Jetson, iOS, kamera OAK, Raspberry Pi, browser, cloud, dll., menyediakan container yang dioptimasi sesuai perangkat, serta load balancing.
- Bekerja sama dengan orang lain untuk membangun aplikasi bersama, dengan job assignment, instruksi, notifikasi, dan metrik performa anggota.

2.2 Penelitian Terkait

Tabel berikut berisi daftar penelitian terkait yang akan digunakan sebagai rujukan dalam penelitian ini.

Tabel 2. Daftar Penelitian

No.	Nama	Metode	Hasil
(1)	(2)	(3)	(4)
1.	Improved YOLO Based Detection Algorithm for Floating Debris in Waterway Feng Lin, Tian Hou, Qiannan Jin, Aiju You (2021)	Menambahkan <i>feature map attention</i> (FMA) ke dalam algoritma YOLOv5s	Menghasilkan algoritma FMA-YOLOv5s. Algoritma baru tersebut bisa digunakan untuk mendeteksi sampah mengapung di permukaan sungai secara waktu-nyata

2.	<p>An automated solid waste detection using the optimized YOLO model for riverine management</p> <p>Mohd Khairuddin, Uswah Khairuddin (2022)</p>	<p>Mengembangkan algoritma YOLOv4 supaya memiliki kemampuan untuk mendeteksi sampah mengapung secara otomatis.</p>	<p>Dengan mAP sebesar 89%, algoritma yang sudah dikembangkan tersebut mampu mendeteksi 5 jenis sampah mengapung, yaitu botol plastik, kalen aluminium, kantong plastik, styrofoam, dan container plastik (ember, Tupperware, dsb.)</p>
3.	<p>The Determination of River Cleanness Level Based on the Number of Waste Based of Digital Image Tabulation using Deep Convolutional Neural Network</p> <p>Aditya Yuli Setyawan, Yosi Kristian (2020)</p>	<p>Menggunakan Deep Convolutional Neural Network untuk memperoleh tingkat kebersihan sungai</p>	<p>Tingkat kebersihan sungai dapat diukur dengan baik, namun karena tiap sungai memiliki jumlah sampah yang berbeda-beda, penelitian lanjutan diperlukan untuk melengkapi hasil</p>
4.	<p>Low resource deep learning to detect waste intensity in the river flow</p> <p>Ferdinandus Fidel Putra, Yulius Denny Prabowo (2021)</p>	<p>Menggunakan algoritma YOLOv3 untuk mendeteksi sampah mengapung, mempermudah proses monitoring sampah, dan mendeteksi sampah mengapung melalui video</p>	<p>Algoritma berhasil mendeteksi sampah mengapung melalui video dengan <i>confidence</i> 98.74%</p>
5.	<p>Identification of illegally dumped plastic waste in a highly polluted river in Indonesia using Sentinel-2 satellite imagery</p> <p>Anjar Dimara Sakti, Emenda Sembiring, Pitri Rohayani, Kamal Nur Fauzan, Tania Septi Anggraini,</p>	<p>Menggunakan metode Random Forest dan Adjusted Plastic Index (API), serta menggunakan data dari citra satelit Sentinel-2 untuk mendeteksi pembuangan sampah plastik ilegal di lingkungan sungai</p>	<p>Menggunakan API mampu meningkatkan akurasi dalam mengidentifikasi sampah plastik jika dibandingkn dengan menggunakan Plastic Index (PI). Seiring meningkatnya teknologi <i>machine learning</i> dan implementasi API, penelitian lanjutan diperlukan untuk menlampau Batasan yang ada dan untuk dapat digunakan di area selain studi kasus</p>

	Cokro Santoso, Vinka Aprilla Patricia, Kalingga Titon Nur Ihsan, Attar Hikmahtiar Ramadan, Sanjiwana Arjasakusuma, Danang Surya Candra ()		
--	--	--	--

Sumber utama yang digunakan dalam penelitian ini adalah jurnal yang berjudul “Improved Yolo based detection algorithm for floating debris in waterway” yang ditulis oleh Feng Lin, Tian Hou, Qiannan Jin, & Aiju You. Penelitian ini mencoba menggabungkan algoritma YOLOv5 dengan fitur FMA (*feature map attention*), menghasilkan algoritma FMA-YOLOv5s. Algoritma baru ini bisa mendeteksi sampah mengapung dengan lebih baik jika dibandingkan dengan YOLOv5. Pada dasarnya, metode *image processing* tradisional sulit untuk melihat sampah mengapung karena lingkungan observasi yang kompleks, seperti cahaya yang memantul dari permukaan sungai, sampah yang hanya setengah mengapung, tanaman air, dan sebagainya. Penelitian ini berisi teori dan contoh penggunaan algoritma YOLO, terutama YOLOv5, yang sangat diperlukan dalam penelitian yang sedang dikerjakan ini, sehingga akan dijadikan sumber rujukan utama.

Sumber utama lain yang akan digunakan adalah sebuah jurnal yang berjudul “An automated solid waste detection using the optimized YOLO model for riverine management” yang ditulis oleh Nur Athirah Zailan dkk. Penelitian ini mengajukan model YOLOv4 yang telah dikembangkan untuk bisa bekerja dalam kondisi yang beragam, seperti perbedaan tingkat cahaya, latar belakang yang kompleks, dan

halangan langsung. Penelitian ini memiliki isi yang mirip dengan rujukan utama di atas, namun menggunakan versi YOLO yang lebih rendah, yaitu versi 4. Namun, jurnal ini berisi penjelasan hasil yang mendalam, sehingga akan dijadikan rujukan kedua.

Selain kedua sumber utama di atas, sumber lain yang akan digunakan adalah sebuah jurnal yang berjudul “The Determination of River Cleanness Level Based on the Number of Waste Based of Digital Image Tabulation using Deep Convolutional Neural Network” yang ditulis oleh Aditya Yuli Setyawan & Yosi Kristian. Penelitian ini membahas tentang bagaimana memperoleh tingkat kebersihan sungai menggunakan Deep Convolutional Neural Network berbasis pengolahan citra digital sebagai metode pendeteksian sampah berdasarkan banyaknya sampah di sungai. Jurnal ini berisi teori image processing menggunakan CNN, namun algoritma yang digunakan disusun sendiri oleh penulis.

Sumber lain yang akan digunakan adalah sebuah jurnal yang berjudul “Low resource deep learning to detect waste intensity in the river flow” yang ditulis oleh Ferdinandus Fidel Putra & Yulius Denny Prabowo. Penelitian ini bertujuan untuk membuat aplikasi yang bisa mendeteksi, melacak, dan mengukur sampah di sungai menggunakan YOLOv3 terhadap input berupa video. Hasil yang diperoleh adalah sebuah aplikasi yang mampu menampilkan jumlah sampah dalam video dengan *confidence* yang tinggi, hingga mencapai 98.74%. Penelitian ini menggunakan versi YOLO yang lebih rendah, yaitu YOLOv3, namun memiliki hasil yang sangat positif.

Kemudian, masih ada sumber lain yang akan digunakan, yaitu sebuah artikel yang berjudul “Identification of illegally dumped plastic waste in a highly

polluted river in Indonesia using Sentinel-2 satellite imagery” yang ditulis oleh Anjar Dimara Sakti dkk. Penelitian ini bertujuan untuk mengidentifikasi pembuangan sampah ilegal di lingkungan sungai, dengan menggunakan indeks plastik yang telah disesuaikan (API, Adjusted Plastic Index) dan data citra satelit Sentinel-2. Ini adalah penelitian pertama yang membuat API dan random forest machine learning menggunakan data Sentinel-2 untuk mengidentifikasi pembuangan sampah plastik ilegal.

Perbedaan mendasar yang ada antara penelitian ini dengan penelitian terkait adalah sebagai berikut:

- Penelitian 1, 2, dan 4 menggunakan versi YOLO yang lebih lawas dibandingkan dengan penelitian ini. Penelitian 1 menggunakan YOLOv5, penelitian 2 menggunakan YOLOv4, dan penelitian 4 menggunakan YOLOv3, sementara penelitian ini menggunakan YOLOv8.
- Penelitian 3 dan 5 menggunakan data dari citra satelit, sementara penelitian ini yang menggunakan foto dan video.

2.3 Kerangka Pikir

Masalah utama dalam penelitian ini adalah menemukan cara untuk mendeteksi sampah mengapung di sungai. Kemudian, setelah berkonsultasi dan mencari informasi, telah ditentukan bahwa lokus yang digunakan adalah Sungai Ciliwung, dan metode yang digunakan adalah Convolutional Neural Network dengan algoritma YOLOv8. Setelah penulis menyelesaikan pengumpulan data dan mulai menggunakan algoritma, penulis mengalami kesulitan dalam memasukkan data yang telah diperoleh ke dalam algoritma. Setelah kajian pustaka lebih lanjut,

penulis menemukan aplikasi Roboflow sebagai alat yang sangat membantu, bukan hanya dalam menyusun dataset, namun juga dalam labelling. Maka, penulis menggunakan Roboflow untuk labelling, preprocessing, dan membuat dataset, dan memasukkan dataset tersebut ke algoritma YOLOv8. Hasil yang diperoleh kemudian dianalisis lebih lanjut untuk menentukan hasil akhir penelitian, membuat pembahasan, dan menetapkan kesimpulan.

Alasan menetapkan Sungai Ciliwung sebagai lokus penelitian adalah karena Sungai Ciliwung adalah salah satu sungai yang terkenal sebagai penyebab banjir di Provinsi DKI Jakarta, dan ada bagian sungai yang terjangkau dari lokasi penulis. Alasan menggunakan CNN sebagai metode merupakan hasil dari kajian pustaka. Hasil dari kajian teori dan pencarian penelitian terkait menunjukkan bahwa CNN banyak digunakan dalam image classification. Kemudian, alasan menggunakan YOLO sebagai algoritma utama juga merupakan hasil dari kajian pustaka. Penelitian terkait menggunakan algoritma YOLOv5, namun setelah penulis mencari mengenai YOLO, penulis menemukan bahwa ada versi yang lebih baru, yaitu YOLOv8. Oleh karena itu, penulis juga ingin mengetahui perbedaan kedua versi ini, dan harapannya dapat memperbandingkan hasil dari kedua versi tersebut.

“...sengaja dikosongkan...”

BAB III

METODOLOGI

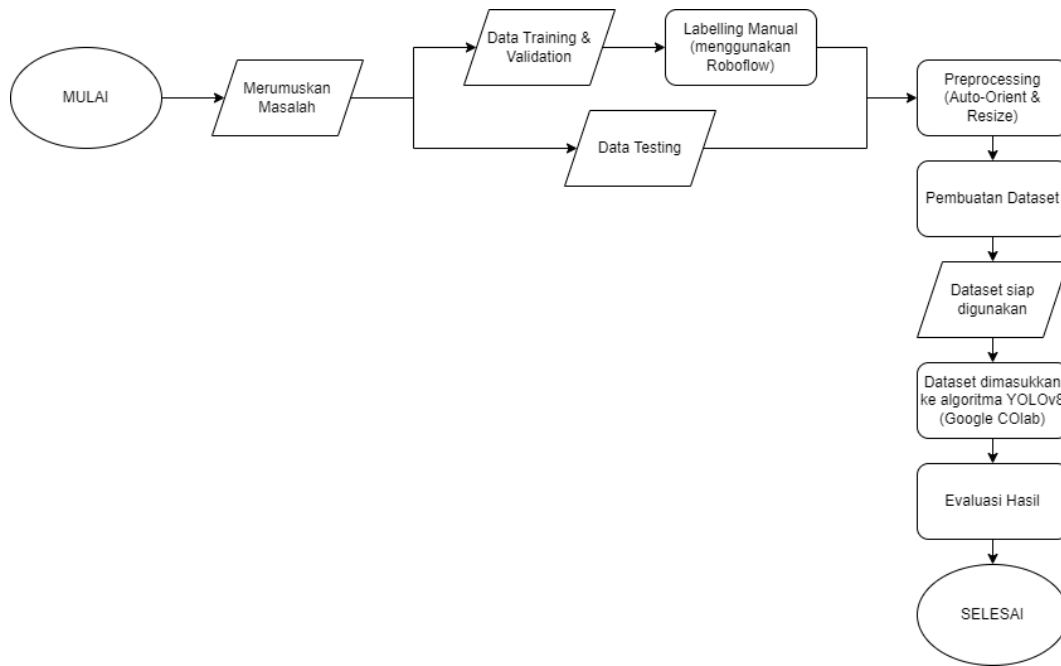
3.1 Ruang Lingkup Penelitian

Penelitian ini memiliki ruang lingkup sebagai berikut:

- Data yang digunakan untuk *training set* dan *validation set* adalah data tahun 2020 – 2022.
- Data yang digunakan untuk *testing set* adalah sebagian data *training* yang tidak diberi label, serta data baru yang diambil pada tahun 2023.
- Data diproses melalui aplikasi Roboflow untuk labeling, *preprocessing*, dan pembuatan dataset. Label yang digunakan hanya 1, yaitu “*wastes*” (sampah).
- Algoritma yang digunakan adalah algoritma YOLOv8. Algoritma ditulis dan dijalankan melalui Google Colab, dengan menggunakan GPU untuk *runtime*.
- Jumlah epoch yang digunakan untuk *training* model adalah 50 epoch, 100 epoch, dan 200 epoch. Dari 3 training tersebut, akan dipilih *training* yang memiliki hasil terbaik. Dari hasil *training* yang dipilih, akan diambil 2 epoch untuk *validating* dan *testing*, yaitu epoch terbaik dan epoch terakhir.
- Rasio data yang digunakan adalah 90:10 untuk *training* dan *validation*.
- Ukuran gambar yang digunakan dalam algoritma adalah 900 x 900.

3.2 Alur Penelitian

Alur penelitian ini dapat dijelaskan dalam grafik berikut.



Gambar 6. Alur penelitian

Setelah merumuskan masalah, langkah pertama yang dilakukan adalah mengumpulkan data. Setelah semua data dikumpulkan, langkah selanjutnya adalah memproses data di aplikasi Roboflow. Untuk data *training* dan *validation*, akan dilakukan *labeling* secara manual, untuk menandakan suatu objek sebagai “sampah” (“*wastes*”). Kemudian, akan dilakukan *preprocessing* terhadap keseluruhan data. *Preprocessing* yang dilakukan adalah Auto-Orient dan Resize.

Auto-Orient berfungsi untuk membuat *bounding box* yang sudah dibuat sebelumnya tetap pada objek yang diberi label, sehingga posisinya mengikuti objek, bukan koordinat gambar. Ini membuat *bounding box* akan bergerak mengikuti objek yang telah diberi label, bahkan jika orientasi gambar diubah. Yang berarti *bounding box* akan mengikuti posisi objek pada gambar, bukan orientasi gambar (*portrait* atau *landscape*) (Dwyer, 2020). Sementara itu, Resize berfungsi untuk

menyetarakan ukuran gambar menjadi satu ukuran seragam. Tujuannya adalah untuk mengurangi ukuran gambar dan mempercepat proses training. (Nelson, 2020)

Setelah preprocessing selesai, semua data akan dijadikan satu dataset yang mudah untuk diimpor ke mana saja. Dataset ini akan digunakan dalam algoritma YOLOv8 yang ada di Google Colab. Hasil running algoritma kemudian akan dianalisis.

3.3 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah pengambilan gambar, baik secara daring maupun secara langsung menggunakan perangkat keras pribadi.

- Untuk data yang akan digunakan sebagai *training set* dan *validation set*, data berupa gambar yang diambil dari Google Images dan video yang diambil dari YouTube. Data yang diambil adalah foto dan video yang menampilkan keadaan Sungai Ciliwung pada tahun 2020 – 2022 dengan sampah mengapung yang terlihat dengan jelas. Kata kunci pencarian yang digunakan adalah “sampah sungai ciliwung 2020”, “sampah sungai ciliwung 2021”, dan “sampah sungai ciliwung 2022”. Rasio data *training* dan *validation* adalah 90:10. Foto dan video diambil sebanyak mungkin untuk memperbanyak jumlah data yang digunakan untuk *training* dan *validation*. Jumlah data yang diperoleh adalah 547 data *training* dan 60 data *validation*.
- Untuk data yang akan digunakan sebagai *testing set*, data berupa gambar yang diambil secara langsung menggunakan *smartphone* pribadi. Data diambil pada tanggal 19 – 24 November 2023. Foto yang diambil adalah keadaan Sungai

Ciliwung dengan sampah mengapung yang terlihat dengan jelas. Sampel bagian sungai yang digunakan adalah bagian sungai yang berada di belakang kampus Politeknik Statistika STIS. Selain itu, sebagian dari data *training* dan *validation* juga diambil sebagai data testing. Jumlah data yang diperoleh adalah 114 data baru dan 76 data *training* tanpa label. Total data testing yang digunakan adalah 190 gambar.

Jumlah gambar yang diambil per hari adalah sebagai berikut:

Minggu, 19 November 2023 = 21 gambar

Senin, 20 November 2023 = 17 gambar

Selasa, 21 November 2023 = 22 gambar

Rabu, 22 November 2023 = 24 gambar

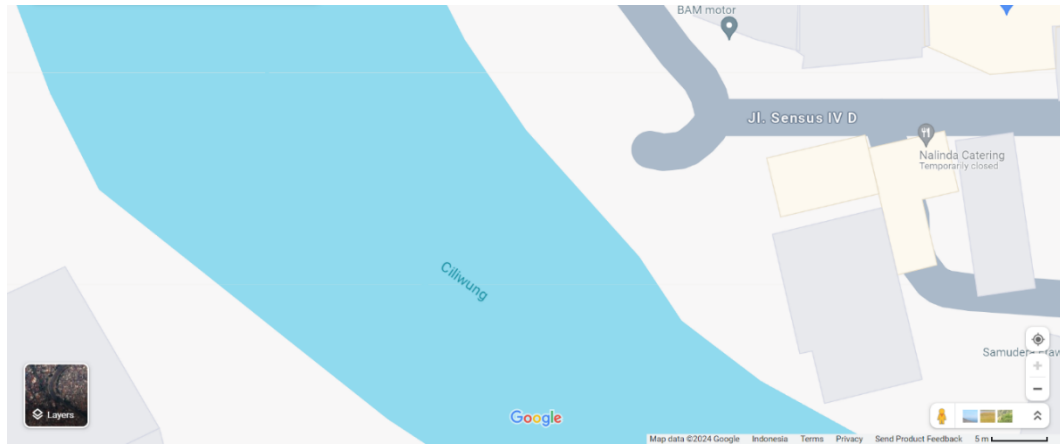
Jumat, 24 November 2023 = 30 gambar

Lokasi pengambilan gambar adalah sebagai berikut:



Sumber: Google Maps (maps.google.com), 2023

Gambar 7. Lokasi pengambilan gambar



Sumber: Google Maps (maps.google.com), 2023

Gambar 8. Lokasi dan jangkauan pengambilan gambar (Lat. -6.231276, Lon. 106.863590)

Spesifikasi smartphone yang digunakan adalah sebagai berikut:

Merek	Infinix HOT 12 Play
Model	Infinix X6816
Tahun rilis	2022
Versi Android	Android 11 “Red Velvet Cake”
Chipset	Mediatek MT6765V/CB Helio G37
Dimensi	171 x 78 x 8.9 mm
Berat	209 g
Jenis layar	IPS LCD, 90 Hz
Ukuran layar	6.82 inches, 111.7 cm ²
Resolusi	720x1640 piksel
CPU	OctaCore Processor
Kamera	Depan: 8M Belakang: 13M Dual Camera
Kualitas video	1080p@30fps
Penyimpanan	RAM 4GB + 2GB FusiMem ROM 128GB
Baterai	6000 mAh

3.4 Metode Analisis

Data yang telah dikumpulkan perlu dianalisis, agar data tersebut dapat digunakan sebagai dasar pengambilan kesimpulan. Metode analisis yang digunakan dalam penelitian ini adalah sebagai berikut:

- Data yang telah dikumpulkan dimasukkan ke dalam aplikasi Roboflow. Data berupa video dipecah-pecah menjadi gambar per *frame*. Dalam aplikasi Roboflow, data kemudian diberi label (*labeling*), *preprocessing*, melatih model, dan menyusun dataset. Dataset yang dihasilkan dihubungkan ke algoritma CNN.
- Algoritma CNN YOLOv8 dijalankan melalui Google Colab menggunakan GPU. Dataset yang dihasilkan dihubungkan ke Google Colab, dan algoritma kini dapat dijalankan.
- Dari antara hasil yang diperoleh dari proses *training* algoritma, hasil yang perlu diperhatikan adalah grafik Results dan Confusion Matrix. Nilai yang dibutuhkan adalah nilai Precision, Recall, Accuracy, dan F1-Score.
- Results Graph diperlukan untuk melihat grafik-grafik yang mampu menjelaskan hasil training. Yang perlu dilihat adalah grafik **box_loss**, **cls_loss**, dan **dfl_loss**, baik untuk *training set* maupun *validation set*. Kriteria yang baik adalah jika grafik-grafik tersebut mengalami tren menurun. Selain itu, juga akan dilihat grafik **metrics/precision**, **metrics/recall**, **metrics/mAP50**, dan **metrics/mAP50-95**. Kriteria yang baik adalah jika grafik-grafik tersebut mengalami tren meningkat.
- Confusion Matrix diperlukan untuk mendapatkan nilai True Positive (TP), False Positive (FP), False Negative (FN), dan True Negative (TN). Nilai-nilai tersebut diperlukan untuk menghitung Precision, Recall, Accuracy, dan F1-Score.

Untuk menghitung Precision, rumusnya adalah sebagai berikut:

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

Untuk menghitung Recall, rumusnya adalah sebagai berikut:

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

Untuk menghitung Accuracy, rumusnya adalah sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

Dan untuk menghitung F1-Score, rumusnya adalah sebagai berikut:

$$F1 = \frac{2 \times P \times R}{P+R} \quad (4)$$

4. Akan dilakukan *training* sebanyak 3 kali dengan maksimal epoch yang berbeda-beda, yaitu 50 epoch, 100 epoch, dan 200 epoch. Dari ketiga *training* tersebut, akan dipilih *training* yang memiliki hasil yang terbaik. Hasil terbaik yang dimaksud adalah epoch terbaik yang tidak terlalu dekat dengan epoch terakhir, serta nilai Precision, Recall, Accuracy, dan F1 Score yang tinggi. Dari hasil *training* yang dipilih, yang akan digunakan untuk *validating* dan *testing* adalah epoch terbaik dan epoch terakhir. Hasil *validating* dilihat dari grafik Validation Predictions. Hasil *testing* dilihat dari Prediction with Test Images.

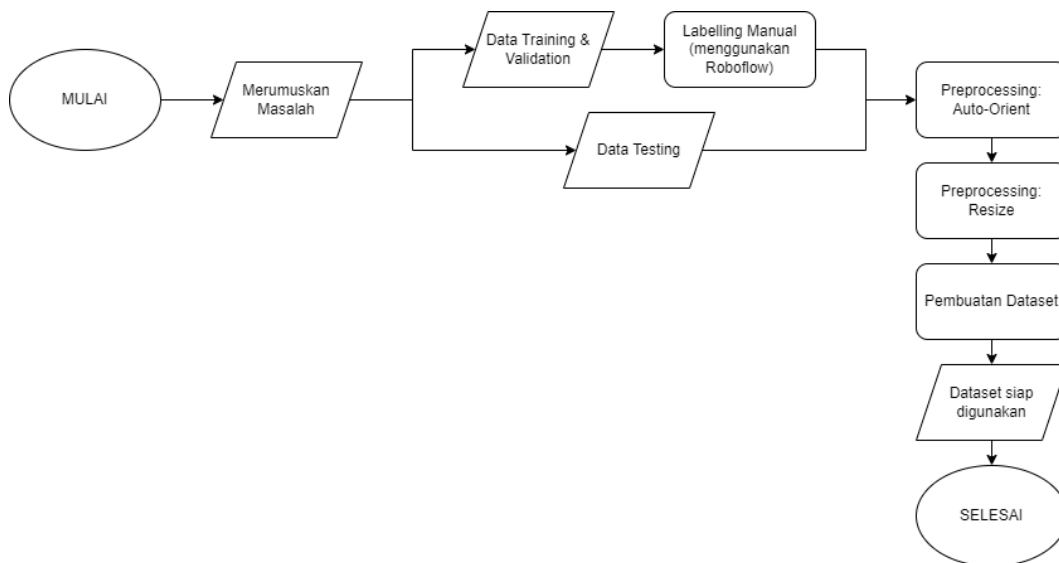
“...sengaja dikosongkan...”

BAB IV

HASIL DAN PEMBAHASAN

4.1 Proses Pengumpulan dan Pengolahan Data

Berikut adalah proses mengumpulkan dan mengolah data dalam bentuk grafik:



Gambar 9. Alur pengumpulan dan pemrosesan data

Data dikumpulkan dengan cara mengambil foto secara daring menggunakan mesin pencari Google, mengambil video secara daring menggunakan mesin pencari Google dan situs video YouTube, serta mengambil gambar secara langsung menggunakan perangkat keras *smartphone* milik peneliti. Setelah semua data yang dibutuhkan telah dikumpulkan, data kemudian dimasukkan ke dalam aplikasi Roboflow untuk *labelling* dan *preprocessing*.

Data foto dan video yang diperoleh secara daring akan digunakan sebagai data *training* dan data *validation*. Data berikut akan diberi label sebelum dilakukan

preprocessing. Hanya 1 label yang akan digunakan, yaitu label “sampah” (“*wastes*”).

Sementara itu, data yang diambil secara langsung akan digunakan sebagai data *testing*. Sebagian dari data yang diambil secara daring juga akan digunakan sebagai data *testing*. Data *testing* tidak diberi label, dan langsung diarahkan untuk *preprocessing*. Tujuannya adalah untuk mengukur kemampuan model dalam menentukan sampah mengapung, tanpa terlebih dahulu mengetahui sampah mengapung berdasarkan label yang telah diberikan sebelumnya.

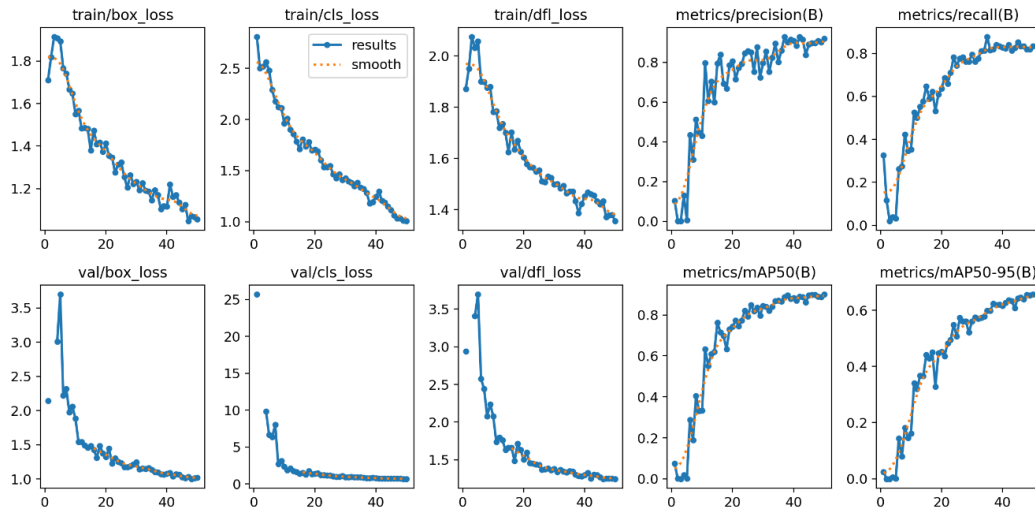
Preprocessing yang dilakukan adalah Auto-Orient dan Resize. Auto-Orient berfungsi untuk membuat *bounding box* yang sudah dibuat sebelumnya tetap pada objek yang diberi label, sehingga posisinya mengikuti objek, bukan koordinat gambar. Sementara itu, Resize berfungsi untuk menyetarakan ukuran gambar menjadi satu ukuran seragam.

Setelah proses mengumpulkan dan memproses data selesai, diperoleh jumlah data sebagai berikut: 547 data *training*, 60 data *validation*, dan 190 data *testing*. Keseluruhan data kemudian dijadikan 1 dataset, dan dimasukkan ke dalam algoritma YOLOv8 yang telah disiapkan di Google Colab. Algoritma diperoleh dari situs penyedia YOLOv8, Ultralytics, dengan beberapa modifikasi.

4.2 Analisis Hasil Training & Perbandingan Antar Epoch Maksimal

Training model dilakukan dengan beberapa epoch maksimal, yaitu 50 epoch, 100 epoch, dan 200 epoch. Tujuannya adalah untuk melakukan perbandingan performa antara epoch maksimal. Berikut adalah masing-masing hasil training dan perbandingannya dengan hasil lainnya:

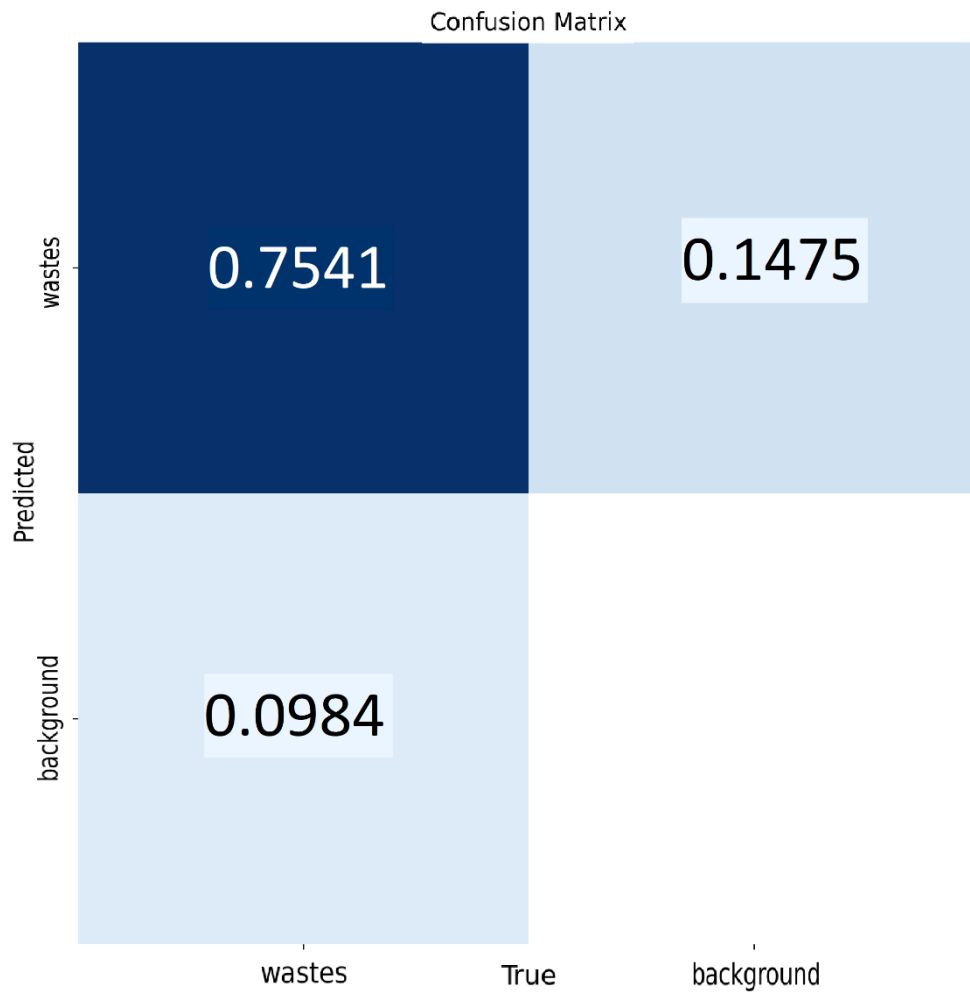
1. 50 epoch



Gambar 10. Results Graph (50 epoch)

Gambar 10 menunjukkan Results Graph dari *training* menggunakan 50 epoch. Dari Results Graph di atas, dapat dilihat bahwa grafik **box_loss**, **cls_loss**, dan **dfl_loss**, baik untuk *training set* maupun *validation set*, mengalami tren menurun. Hal ini menunjukkan bahwa banyaknya loss yang terjadi dalam satu epoch berkurang seiring dengan peningkatan jumlah epoch yang telah dilatih.

Kemudian, dapat dilihat bahwa grafik **metrics/precision**, **metrics/recall**, **metrics/mAP50**, dan **metrics/mAP50-95** mengalami tren meningkat, yang berarti nilai Precision, Recall, dan Mean Average Precision setiap epoch meningkat seiring dengan peningkatan jumlah epoch yang telah dilatih. Sehingga, dapat disimpulkan bahwa tidak ada masalah dalam sesi *training* kali ini.



Gambar 11. Confusion Matrix (50 epoch)

Gambar 11 menunjukkan Confusion Matrix dari *training* menggunakan 50 epoch. Berdasarkan Confusion Matrix di atas, diperoleh nilai-nilai sebagai berikut:

$$TP = 0.7541$$

$$FP = 0.1475$$

$$FN = 0.0984$$

$$TN = 0$$

Dengan nilai di atas, maka nilai Precision adalah sebagai berikut:

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

$$P = \frac{0.7541}{0.7541 + 0.1475}$$

$$P = \frac{0.7541}{0.9016}$$

$$P = 0.8364 = 83.64\%$$

Sementara itu, nilai Recall adalah sebagai berikut:

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

$$R = \frac{0.7541}{0.7541 + 0.0984}$$

$$R = \frac{0.7541}{0.8525}$$

$$R = 0.8846 = 88.46\%$$

Kemudian, nilai Accuracy adalah sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{0.7541}{1}$$

$$Accuracy = 0.7541 = 75.41\%$$

Terakhir, nilai F1-Score adalah sebagai berikut:

$$F1 = \frac{2 \times P \times R}{P + R}$$

$$F1 = \frac{2 \times 0.8364 \times 0.8846}{0.8364 + 0.8846}$$

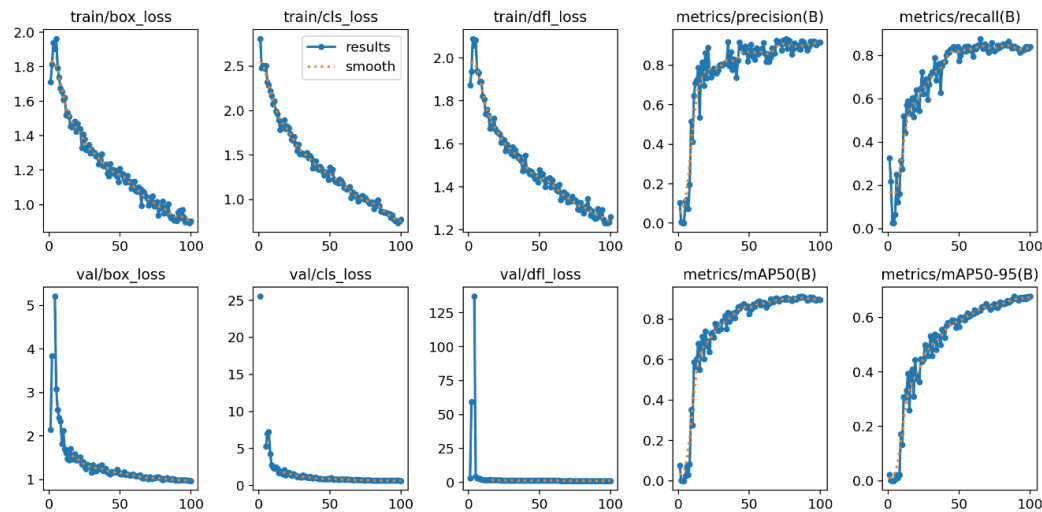
$$F1 = \frac{1.4798}{1.721}$$

$$F1 = 0.8598 = 85.98\%$$

Berdasarkan hasil di atas, dapat disimpulkan bahwa:

- Precision = 83.64%, yang berarti bahwa dari semua hasil yang diambil oleh algoritma, 83.64% adalah benar gambar sampah mengapung yang diberi label “wastes”.
- Recall = 88.46%, yang berarti bahwa dari semua hasil yang diberi label “wastes”, hanya 88.46% yang diambil oleh algoritma.
- Accuracy = 75.41%, yang berarti bahwa algoritma memiliki akurasi sebesar 75.41%, untuk membedakan sampah mengapung dan bukan sampah mengapung.
- F1-Score = 85.98%

2. 100 epoch

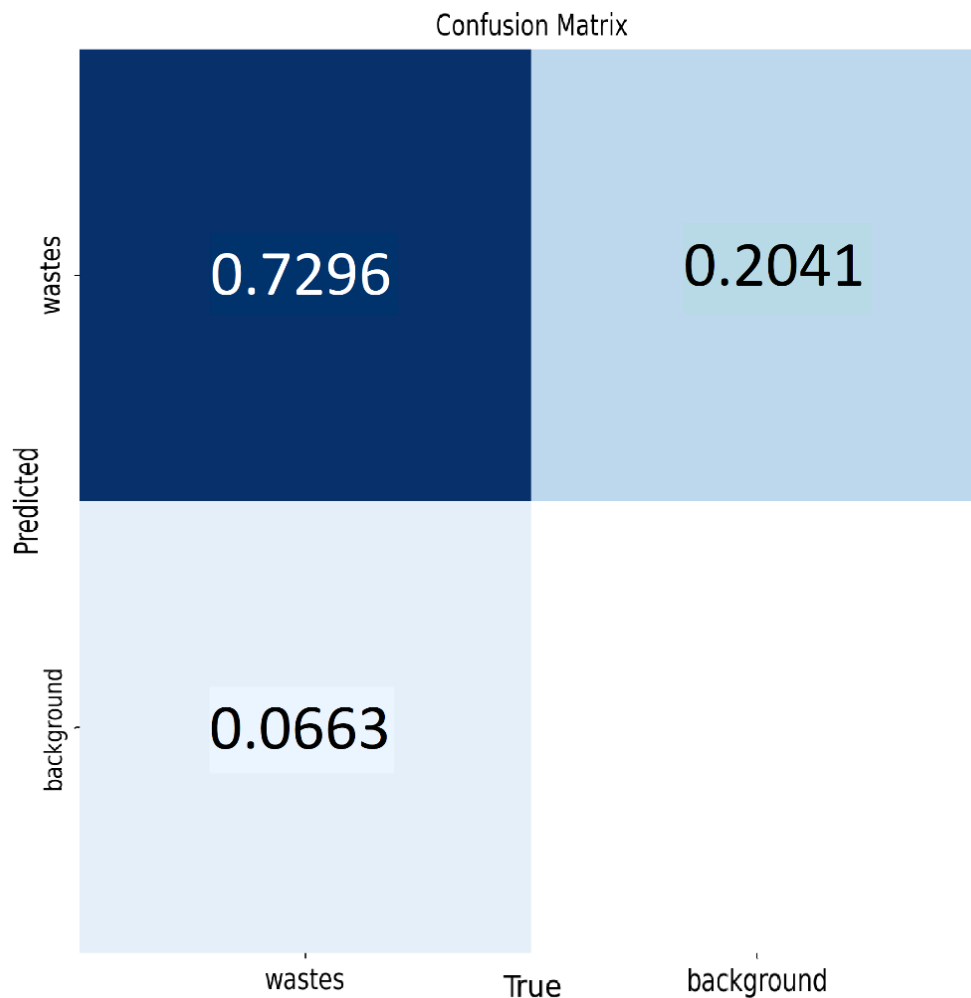


Gambar 12. Results Graph (100 epoch)

Gambar 12 menunjukkan Results Graph dari training menggunakan 100 epoch. Dari Results Graph di atas, dapat dilihat bahwa grafik **box_loss**, **cls_loss**, dan **dfl_loss**, baik untuk *training set* maupun *validation set*, mengalami tren

menurun. Hal ini menunjukkan bahwa banyaknya loss yang terjadi dalam satu epoch berkurang seiring dengan peningkatan jumlah epoch yang telah dilatih.

Kemudian, dapat dilihat bahwa grafik **metrics/precision**, **metrics/recall**, **metrics/mAP50**, dan **metrics/mAP50-95** mengalami tren meningkat, yang berarti nilai Precision, Recall, dan Mean Average Precision setiap epoch meningkat seiring dengan peningkatan jumlah epoch yang telah dilatih. Sehingga, dapat disimpulkan bahwa tidak ada masalah dalam sesi *training* kali ini.



Gambar 13. Confusion Matrix (100 epoch)

Gambar 13 menunjukkan Confusion Matrix dari *training* menggunakan 100 epoch. Berdasarkan Confusion Matrix di atas, diperoleh nilai-nilai sebagai berikut:

$$TP = 0.7296$$

$$FP = 0.2041$$

$$FN = 0.0663$$

$$TN = 0$$

Dengan nilai di atas, maka nilai Precision adalah sebagai berikut:

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

$$P = \frac{0.7296}{0.7296 + 0.2041}$$

$$P = \frac{0.7296}{0.9337}$$

$$P = 0.7814 = 78.14\%$$

Sementara itu, nilai Recall adalah sebagai berikut:

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

$$R = \frac{0.7296}{0.7296 + 0.0663}$$

$$R = \frac{0.7296}{0.7959}$$

$$R = 0.9167 = 91.67\%$$

Kemudian, nilai Accuracy adalah sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{0.7296}{1}$$

$$Accuracy = 0.7296 = 72.96\%$$

Terakhir, nilai F1-Score adalah sebagai berikut:

$$F1 = \frac{2 \times P \times R}{P + R}$$

$$F1 = \frac{2 \times 0.7814 \times 0.9167}{0.7814 + 0.9167}$$

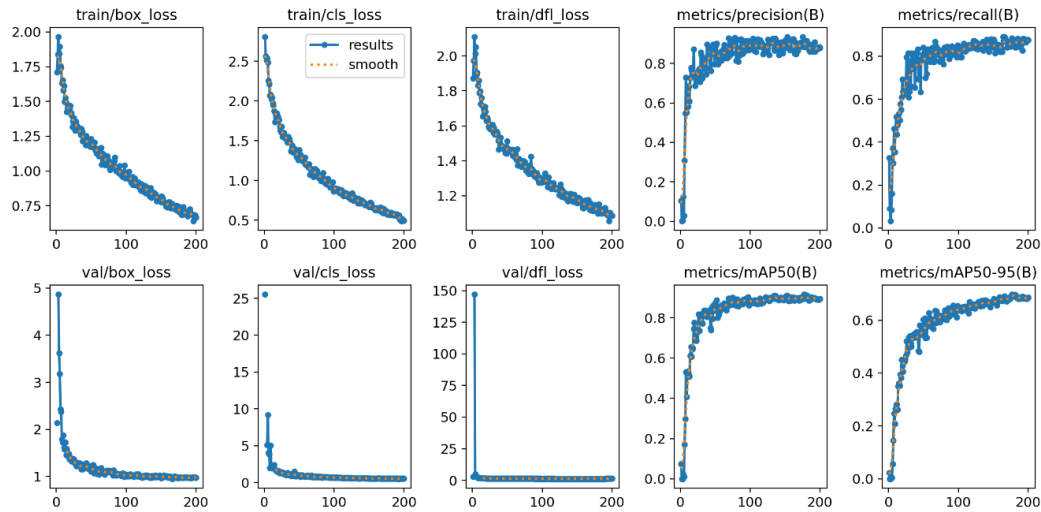
$$F1 = \frac{1.4326}{1.6981}$$

$$F1 = 0.8437 = 84.37\%$$

Berdasarkan hasil di atas, dapat disimpulkan bahwa:

- Precision = 78.14%, yang berarti bahwa dari semua hasil yang diambil oleh algoritma, 78.14% adalah benar gambar sampah mengapung yang diberi label “wastes”.
- Recall = 91.67%, yang berarti bahwa dari semua hasil yang diberi label “wastes”, hanya 91.67% yang diambil oleh algoritma.
- Accuracy = 72.96%, yang berarti bahwa algoritma memiliki akurasi sebesar 72.96%, untuk membedakan sampah mengapung dan bukan sampah mengapung.
- F1-Score = 84.37%

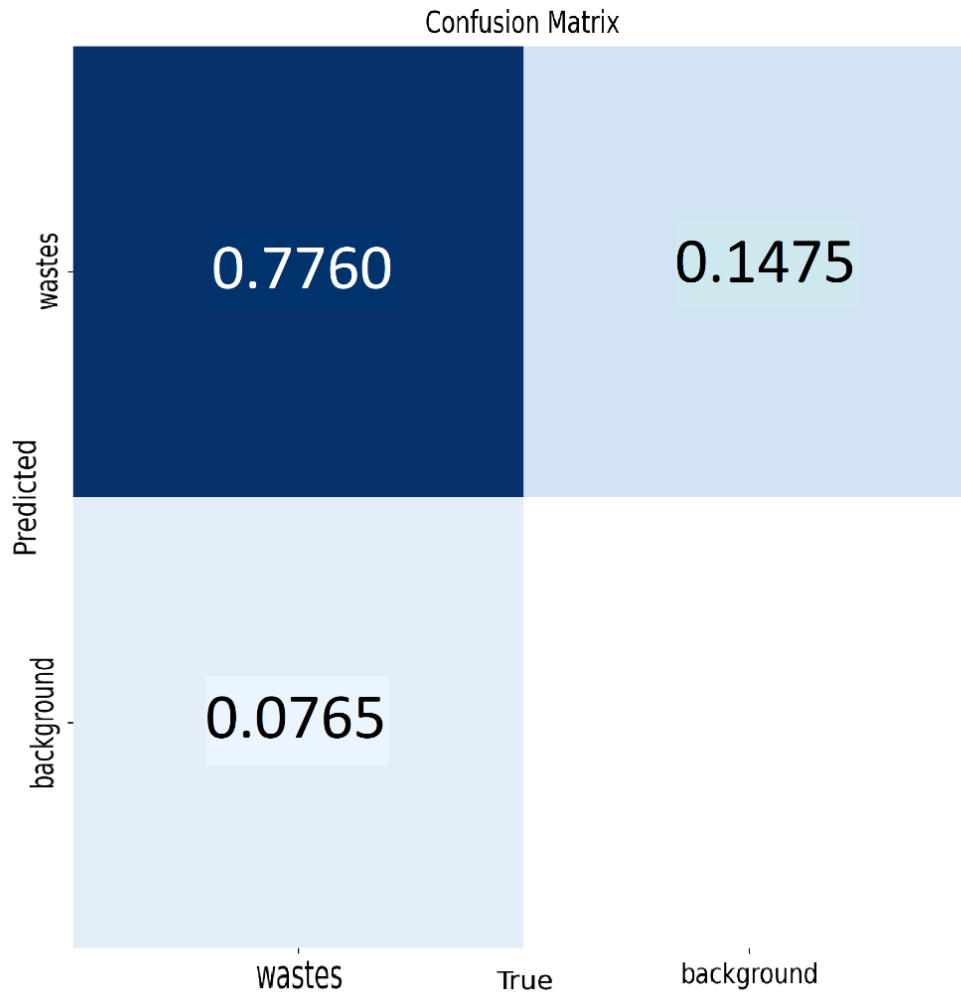
3. 200 epoch



Gambar 14. Results Graph (200 epoch)

Gambar 14 menunjukkan Results Graph dari training menggunakan 200 epoch. Dari Results Graph di atas, dapat dilihat bahwa grafik **box_loss**, **cls_loss**, dan **dfl_loss**, baik untuk *training set* maupun *validation set*, mengalami tren menurun. Hal ini menunjukkan bahwa banyaknya loss yang terjadi dalam satu epoch berkurang seiring dengan peningkatan jumlah epoch yang telah dilatih.

Kemudian, dapat dilihat bahwa grafik **metrics/precision**, **metrics/recall**, **metrics/mAP50**, dan **metrics/mAP50-95** mengalami tren meningkat, yang berarti nilai Precision, Recall, dan Mean Average Precision setiap epoch meningkat seiring dengan peningkatan jumlah epoch yang telah dilatih. Sehingga, dapat disimpulkan bahwa tidak ada masalah dalam sesi *training* kali ini.



Gambar 15. Confusion Matrix (200 epoch)

Gambar 15 menunjukkan Confusion Matrix dari *training* menggunakan 200 epoch. Berdasarkan Confusion Matrix di atas, diperoleh nilai-nilai sebagai berikut:

$$TP = 0.776$$

$$FP = 0.1475$$

$$FN = 0.0765$$

$$TN = 0$$

Dengan nilai di atas, maka nilai Precision adalah sebagai berikut:

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

$$P = \frac{0.7760}{0.7760 + 0.1475}$$

$$P = \frac{0.7760}{0.9235}$$

$$P = 0.8402 = 84.02\%$$

Sementara itu, nilai Recall adalah sebagai berikut:

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

$$R = \frac{0.7760}{0.7760 + 0.0765}$$

$$R = \frac{0.7760}{0.8525}$$

$$R = 0.9103 = 91.03\%$$

Kemudian, nilai Accuracy adalah sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{0.7760}{1}$$

$$Accuracy = 0.7760 = 77.6\%$$

Terakhir, nilai F1-Score adalah sebagai berikut:

$$F1 = \frac{2 \times P \times R}{P + R}$$

$$F1 = \frac{2 \times 0.8402 \times 0.9103}{0.8402 + 0.9103}$$

$$F1 = \frac{1.5297}{1.7505}$$

$$F1 = 0.8738 = 87.38\%$$

Berdasarkan hasil di atas, dapat disimpulkan bahwa:

- Precision = 84.02%, yang berarti bahwa dari semua hasil yang diambil oleh algoritma, 84.02% adalah benar gambar sampah mengapung yang diberi label “wastes”.
- Recall = 91.03%, yang berarti bahwa dari semua hasil yang diberi label “wastes”, hanya 91.03% yang diambil oleh algoritma.
- Accuracy = 77.6%, yang berarti bahwa algoritma memiliki akurasi sebesar 77.6%, untuk membedakan sampah mengapung dan bukan sampah mengapung.
- F1-Score = 87.38%

4. Perbandingan hasil antar epoch

Ketiga hasil yang telah diperoleh di atas akan dibandingkan satu sama lain, untuk memperoleh hasil terbaik sesuai dengan kriteria yang telah ditetapkan sebelumnya. Berikut adalah perbandingan hasil *training* antar epoch maksimal yang berbeda-beda dalam bentuk tabel.

Tabel 3. Perbandingan Hasil Antar Epoch

Last Epoch	50	100	200
Best Epoch	50	89	177
True Positive (TP)	0.7541	0.7296	0.7760
False Positive (FP)	0.1475	0.2041	0.1475
False Negative (FN)	0.0984	0.0663	0.0765
True Negative (TN)	0	0	0
Precision	83.64%	78.14%	84.02%
Recall	88.46%	91.67%	91.03%
Accuracy	75.41%	72.96%	77.6%
F1-Score	85.98%	84.37%	87.38%

Dari tabel di atas, dapat disimpulkan bahwa:

1. Epoch terbaik (Best Epoch) untuk training 50 & 100 epoch terletak sangat dekat dengan epoch terakhir (Last Epoch), yaitu epoch ke-50 untuk 50 epoch, dan epoch ke-89 untuk 100 epoch. Hal ini mengindikasikan bahwa epoch terbaik masih bisa lebih tinggi lagi. Kemudian, dilakukan training dengan 200 epoch, dan dapat dilihat bahwa epoch terbaik berada di epoch ke-177, jauh dari epoch terakhir.
2. Diperlukan nilai TP yang tinggi, serta nilai FP dan FN yang rendah, untuk memaksimalkan nilai Precision, Recall, Accuracy, dan F1-Score. Berdasarkan tabel di atas, terlihat bahwa nilai TP tertinggi ada di hasil *training* dengan 200 epoch, yaitu 0.7760. Nilai FP terendah ada di hasil *training* dengan 50 & 200 epoch, kedua hasil tersebut memiliki nilai yang sama, yaitu 0.1475. Kemudian, nilai FN terendah berada di hasil *training* dengan 100 epoch, yaitu 0.0663.
3. Sementara itu, nilai TN untuk semua epoch bernilai 0. Hal ini terjadi karena hanya ada 1 label yang digunakan dalam proses *training*, yaitu “wastes”. Sehingga, tidak ada label lain yang dapat digunakan untuk perbandingan hasil, dan label pembanding yang diberikan oleh Roboflow adalah latar belakang gambar (“*background*”).
4. Nilai Precision, Accuracy, dan F1-Score tertinggi berada di hasil *training* dengan 200 epoch. Akan tetapi, nilai Recall tertinggi berada di hasil *training* dengan 100 epoch. Berdasarkan hasil berikut, dapat disimpulkan bahwa model hasil *training* dengan 200 epoch lebih baik jika dibandingkan dengan hasil *training* dengan 50 & 100 epoch.

Dengan memperhatikan hasil-hasil di atas, maka diambil keputusan bahwa hasil *training* yang akan digunakan untuk tahap berikutnya adalah hasil *training* dengan 200 epoch.

4.3 Analisis Hasil Validation



Gambar 16. Val Prediction

Gambar 16 merupakan hasil Val Prediction. Dari hasil yang diperoleh di atas, dapat dilihat hal-hal sebagai berikut:

1. Secara umum algoritma berhasil mendeteksi sampah mengapung dengan cukup baik. Algoritma bahkan dapat mendeteksi sampah berukuran kecil, seperti yang terlihat di gambar A, dengan *confidence* sebesar 0.7.
2. Gambar B, E, H, I, & M menunjukkan algoritma mendeteksi gunung sampah mengapung berukuran besar dengan *confidence* sebesar 0.8 – 0.9.
3. Gambar C, D, F, G, J, K, L, N, dan O menunjukkan algoritma mendeteksi gunung sampah mengapung berukuran sedang dan terletak di belakang objek besar di depan, dengan *confidence* sebesar 0.9 – 1.0.
4. Gambar P menunjukkan algoritma mendeteksi kumpulan sampah mengapung dengan *confidence* di antara 0.5 – 0.9.

4.4 Analisis Hasil Testing

Ada 190 gambar di dalam *testing set*, sehingga yang akan ditampilkan berikut ini hanyalah beberapa contoh kasus yang terjadi dalam Prediction with Test Images. Berikut adalah beberapa contoh hasil *testing*, serta perbandingan antara epoch terbaik (epoch ke-177) dan epoch terakhir (epoch ke-200):



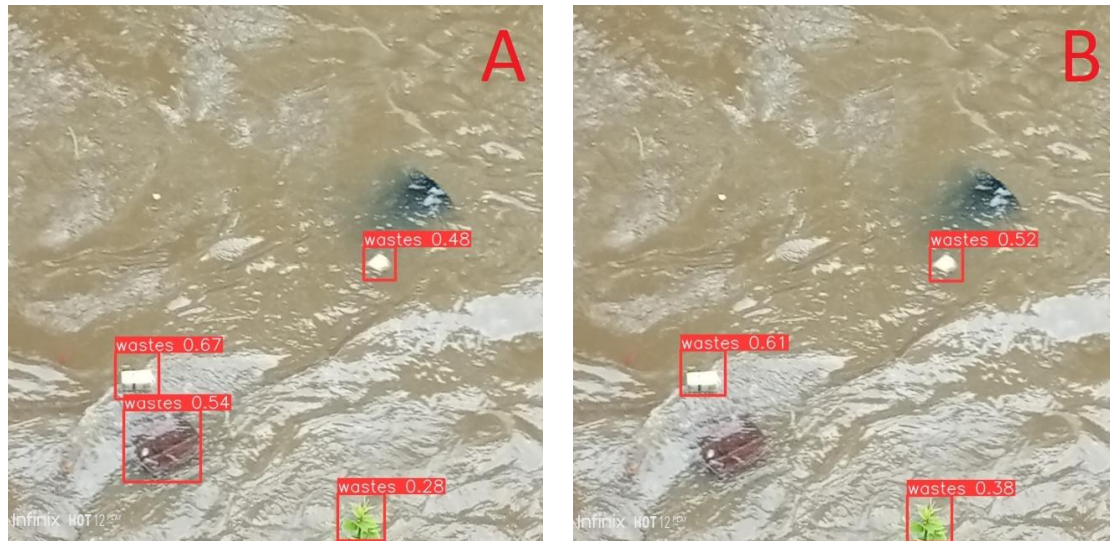
Gambar 17. Contoh sampah terdeteksi untuk: a. epoch terbaik (epoch ke-177) dan b. epoch terakhir (epoch ke-200)

Dari Gambar 17 di atas, terlihat bahwa secara umum epoch terbaik dan epoch terakhir mendeteksi sampah di lokasi yang sama dengan *confidence* yang sama di beberapa deteksi. Sampah mengapung dalam jumlah besar di permukaan sungai bagian kiri atas memiliki *confidence* sebesar 0.96, dan bagian bawah memiliki *confidence* sebesar 0.91. Namun, bagian kanan atas memiliki *confidence* yang berbeda di antara kedua epoch, dimana epoch terbaik memiliki *confidence* yang lebih baik (0.61) dibandingkan epoch terakhir (0.52). Sehingga, dapat dikatakan bahwa untuk contoh kasus di atas, performa epoch terbaik lebih baik dibandingkan epoch terakhir.



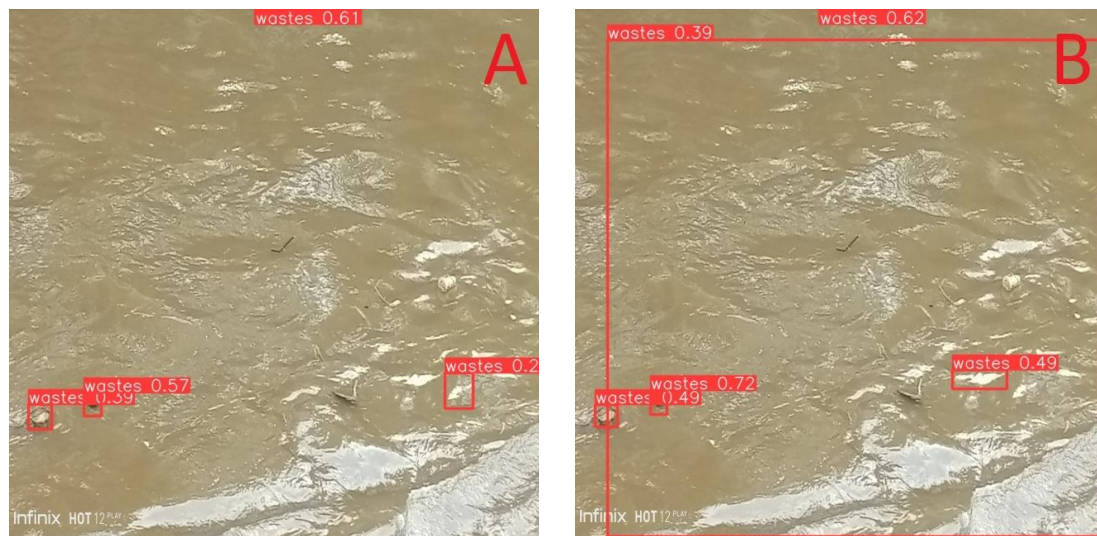
Gambar 18. Contoh sampah terdeteksi untuk: a. epoch terbaik (epoch ke-177) dan b. epoch terakhir (epoch ke-200)

Dari Gambar 18 di atas, terlihat bahwa ada bagian dimana kedua epoch memiliki performa yang sama, dan ada bagian yang memberikan hasil yang berbeda. Sampah mengapung berwarna hijau memiliki *confidence* yang sama di antara kedua epoch, yaitu sebesar 0.92. Kemudian, sampah mengapung kecil berwarna biru dapat dideteksi dengan lebih baik oleh epoch terbaik (*confidence* 0.62) dibandingkan epoch terakhir (*confidence* 0.54). Selanjutnya, sampah mengapung berwarna kuning-merah dapat dideteksi dengan lebih baik oleh epoch terakhir (*confidence* 0.76) dibandingkan epoch terbaik (*confidence* 0.32). Terakhir, sampah mengapung kecil berwarna kuning dapat dideteksi dengan lebih baik oleh epoch terakhir (*confidence* 0.87) dibandingkan epoch terbaik (*confidence* 0.82). Sehingga, dapat dikatakan bahwa untuk contoh kasus di atas, performa epoch terakhir lebih baik dibandingkan epoch terbaik.



Gambar 19. Contoh sampah terdeteksi untuk: a. epoch terbaik (epoch ke-177) dan b. epoch terakhir (epoch ke-200)

Dari Gambar 19 di atas, terlihat bahwa secara umum, epoch terbaik dan epoch terakhir memiliki performa yang berbeda untuk setiap sampah mengapung yang terdeteksi. Sampah mengapung berwarna putih di bagian atas dapat dideteksi dengan lebih baik oleh epoch terakhir (*confidence* 0.52) dibandingkan epoch terbaik (*confidence* 0.48). Rumput di bagian bawah gambar juga dapat dideteksi dengan lebih baik oleh epoch terakhir (*confidence* 0.38) dibandingkan epoch terbaik (*confidence* 0.28). Selanjutnya, sampah mengapung berwarna putih di bagian kiri dapat dideteksi dengan lebih baik oleh epoch terbaik (*confidence* 0.67) dibandingkan epoch terakhir (*confidence* 0.61). Kemudian, sampah mengapung berwarna coklat dapat dideteksi oleh epoch terbaik dengan *confidence* sebesar 0.54, namun tidak dapat dideteksi oleh epoch terakhir. Akan tetapi, kedua epoch tidak dapat mendeteksi sampah plastik hitam setengah mengapung. Sehingga, dapat dikatakan bahwa untuk contoh kasus di atas, performa epoch terbaik lebih baik dibandingkan epoch terakhir.



Gambar 20. Contoh sampah terdeteksi untuk: a. epoch terbaik (epoch ke-177) dan b. epoch terakhir (epoch ke-200)

Dari Gambar 20 di atas, terlihat bahwa secara umum, epoch terbaik dan epoch terakhir memiliki performa yang berbeda untuk setiap sampah mengapung yang terdeteksi. Sampah mengapung kecil berwarna hitam di sebelah kiri dapat dideteksi dengan lebih baik oleh epoch terakhir (*confidence* 0.72) dibandingkan epoch terbaik (*confidence* 0.57). Sampah mengapung kecil berwarna coklat di sebelah kiri dapat dideteksi dengan lebih baik oleh epoch terakhir (*confidence* 0.49) dibandingkan epoch terbaik (*confidence* 0.39). Kemudian, epoch terakhir mampu mengumpulkan berbagai macam sampah ke dalam satu kelompok deteksi, walaupun memiliki *confidence* yang rendah, yaitu 0.39. Juga, tidak ada terlalu banyak sampah mengapung dalam kumpulan deteksi tersebut. Tidak hanya itu, kedua epoch melakukan kesalahan deteksi, dimana pantulan cahaya matahari di permukaan sungai dideteksi sebagai sampah mengapung oleh epoch terbaik (*confidence* 0.61 & 0.2) dan juga oleh epoch terakhir (*confidence* 0.62 & 0.49).

Sehingga, dapat dikatakan bahwa untuk contoh kasus di atas, performa epoch terakhir lebih baik dibandingkan epoch terbaik. Sehingga, dapat dikatakan bahwa untuk contoh kasus di atas, performa epoch terakhir lebih baik dibandingkan epoch terbaik.

Terlihat dari contoh di atas, bahwa algoritma berhasil mendeteksi sampah mengapung di permukaan sungai. Di antara kedua epoch yang digunakan, terlihat bahwa secara umum, epoch terakhir (epoch ke-200) memiliki performa yang sedikit lebih baik dengan epoch terbaik (epoch ke-177). Epoch terakhir mampu mendeteksi lebih banyak sampah dengan *confidence* yang cukup baik. Namun, ada beberapa kasus dimana epoch terakhir melakukan kesalahan deteksi.

Dengan memperhatikan keseluruhan hasil yang diperoleh, baik yang ditampilkan di atas, maupun yang tidak dapat ditampilkan karena keterbatasan ruang, maka diambil keputusan bahwa hasil *testing* yang akan digunakan untuk tahap berikutnya adalah epoch terbaik (epoch ke-177).

“...sengaja dikosongkan...”

BAB IV

HASIL DAN PEMBAHASAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan pada bagian sebelumnya, berikut adalah beberapa hal yang dapat disimpulkan:

1. Image Classification dengan CNN menggunakan algoritma YOLOv8 bisa digunakan untuk mendeteksi sampah mengapung yang ada di permukaan Sungai Ciliwung pada tahun 2020 – 2022
2. Model YOLOv8 bisa diimplementasikan terhadap data baru dari sumber yang berbeda, pada Sungai Ciliwung di tahun 2023. Model yang digunakan berupa hasil training menggunakan 200 epoch, dengan epoch terbaik adalah epoch ke-177. Nilai Precision sebesar 84.02%, Recall sebesar 91.03%, Accuracy sebesar 77.6%, dan F1-Score bernilai 87.38%.

5.2 Saran

Kemudian, saran yang dapat diberikan berdasarkan hasil di atas adalah sebagai berikut:

1. Menggunakan lokus yang berbeda, seperti sungai lain yang terkenal menjadi penyebab banjir di daerah lain, contohnya Sungai Cisadane di Tangerang, Sungai Cibanten di Serang, Sungai Ciberes & Cisanggarung di Cirebon, Sungai Citarum di Bandung, dan sebagainya.

2. Menggunakan metode analisis yang berbeda, seperti meningkatkan versi YOLO yang digunakan menjadi YOLOv8.1 atau YOLOv9, atau menggunakan algoritma/bahasa pemrograman yang berbeda seperti R-CNN (Regional CNN), Fast R-CNN, Faster R-CNN, atau SSD (Single Shot Detector).
3. Menggunakan sumber data yang berbeda, seperti citra satelit, atau live detection menggunakan kamera pengawas (CCTV) atau drone.

DAFTAR PUSTAKA

- Avilov, O., Rimbert, S., Popov, A., & Bougrain, L. (2020). Deep Learning Techniques to Improve Intraoperative Awareness Detection from Electroencephalographic Signals, *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2020, 142 – 145. <https://doi.org/10.1109/EMBC44109.2020.9176228>
- Basel Convention. (1989). *Basel Convention on the Control of Transboundary Movements of Hazardous Wastes and Their Disposal*. Basel: Basel Convention.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer. ISBN 978-0-387-31073-2.
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers, *European Conference on Computer Vision*, 213 – 229. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-58452-8_13
- Chen, Y. C. (2018). Effects of urbanization on municipal solid waste composition. *Waste Management*, 79(1), 828–836. <https://doi.org/10.1016/j.wasman.2018.04.017>
- Collobert, R. & Weston, J. (2008). A unified architecture for natural language processing, *Proceedings of the 25th international conference on Machine learning - ICML '08*, 160 – 167. <https://doi.org/10.1145/1390156.1390177>
- Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection, *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 1, 886 – 893. <https://doi.org/10.1109/CVPR.2005.177>
- Doron, A. (2018). *Waste of a Nation: Garbage and Growth in India*. Harvard University Press.
- Dwyer, B. (2020, 05). When Should I Auto-Orient My Images? [Online] Available: <https://blog.roboflow.com/exif-auto-orientation/>
- Fadillah, I., Lutfienzy A, Kamil, M. F. E., Shalahuddin, M., Setiawan, I., Azidah N, Hanifatul M, Niffa A, Rahmatu S., & Fikri, K. (2019). Perubahan Pola Pikir Masyarakat tentang Sampah melalui Sosialisasi Pengolahan Sampah Organik dan Non Organik di Dusun Pondok, Kecamatan Gedangsari, Kab. Gunungkidul. *Prosiding Konferensi Pengabdian Masyarakat*, 1(1), 239 – 242. Retrieved from <https://sunankalijaga.org/prosiding/index.php/abdimas/article/view/201>

- Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model, *2008 IEEE computer society conference on computer vision and pattern recognition*, 1, 1 – 8. <https://doi.org/10.1109/CVPR.2008.4587597>
- Gavali, P. & Banu, J. S. (2019). Chapter 6 - Deep Convolutional Neural Network for Image Classification on CUDA Platform, *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, 99 – 122. <https://doi.org/10.1016/B978-0-12-816718-2.00013-0>
- Girshick, R. (2015). Fast r-cnn, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1440 – 1448.
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580 – 587.
- Hardesty, L. (2017). Explained: Neural networks. [Online]. Available: <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 346 – 361. <https://doi.org/10.1109/TPAMI.2015.2389824>
- Hendrayanto. (2008). Transboundary watershed management. A case study of upstream-downstream relationships in Ciliwung watershed. *Proceedings of International Workshop on Integrated Watershed Management for Sustainable Water Use in a Humid Tropical Region*, JSPS-DGHE Joint Research Project, Tsukuba, 8(2).
- Hirabayashi, Y., Mahendran, R., Koirala, S., Konoshima, L., Yamazaki, D., Watanabe, S., Kim, H., & Kanae, S. (2013). Global flood risk under climate change, *Nature Climate Change*, 3(9) 816 – 821. <https://doi.org/10.1038/nclimate1911>
- Hubel, H. D. & Wiesel, T. N. (1959). Receptive Fields of Single neurones in the Cat's Striate Cortex, *Journal of Physiology*, 148, 574 – 591.
- Islam, M., Chen, G., & Jin. Z. (2019). An Overview of Neural Network, *American Journal of Neural Networks and Applications*. 5(1), 7 – 11. <https://doi.org/10.11648/j.ajjna.20190501.12>
- Isnanto, B. A. (2023, 05). 13 Penyebab Banjir: Faktor Alam hingga Ulah Manusia. [Online]. Available: <https://www.detik.com/edu/detikpedia/d-6710273/13-penyebab-banjir-faktor-alam-hingga-ulah-manusia>
- Jambeck, J. R., Geyer, R., Wilcox, C., Siegler, T. R., Perryman, M., Andrady, A., Narayan, R., & Law, K. L. (2015). Plastic Waste Inputs from Land into the

- Ocean, *Science*, 347(6223), 768–771.
<https://www.science.org/doi/10.1126/science.1260352>
- Jocher, G., Chaurasia, A., & Qiu, J. (1 Januari 2023). YOLO by Ultralytics - GitHub. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- Kurtzman, T. (2019). Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening, *PLOS ONE*, 14(8). <https://doi.org/10.1371/journal.pone.0220113>
- Law, H. & Deng, J. (2018) Cornernet: Detecting objects as paired keypoints, *Proceedings of the European conference on computer vision (ECCV)*, 734 – 750.
- Lawrence, J. (1992). Data Preparation for a Neural Network, *AI EXPERT.*, 6(11), 34 – 41.
- Lin, F., Hou, T., Jin, Q., & You, A. (2021). Improved Yolo based detection algorithm for floating debris in waterway, *Entropy*, 23(9), 250 – 263.
<https://doi.org/10.3390/e23091111>
- Lin, T.Y., Dollár, P., Girshick, R. B., He, K., Hariharan, B., & Belongie, S. J. (2017). Feature pyramid networks for object detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1(2), 2117 – 2125.
- Lin, T.Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2980 – 2988.
- Lippmann, R. P. (1987). An introduction to computing with neural nets, *IEEE Accost. Speech Signal Process. Mag.*, 4 – 22.
- Lippmann, R., Martin, E., & Paul, D. (1987). Multi-style training for robust isolated-word speech recognition, *ICASSP '87, IEEE International Conference on Acoustics, Speech, and Signal Processing*, 705-708.
<https://doi.org/10.1109/icassp.1987.1169544>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector, *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 21 – 37. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-46448-0_2
- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge, Massachusetts: The MIT Press.
- Muzaidi, I., Anggarini, E., and Prayugo, H. M. (2018). Studi Kasus Pencemaran Air Sungai Teluk Dalam Banjarmasin akibat Limbah Domestik. *Media Teknik Sipil*, 16(2), 108 – 114.
<http://ejournal.umm.ac.id/index.php/jmts/article/view/6267>

- Nasution, D., Harumy, T. H. F., Haryanto, E., Fachrizal, F., Julham, & Turnip, A. (2015). A classification method for prediction of qualitative properties of multivariate EEG-P300 signals, *2015 International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)*. <https://doi.org/10.1109/icacomit.2015.7440180>
- Nelson, J. (2020, 01). You Might Be Resizing Your Images Incorrectly. [Online] Available: <https://blog.roboflow.com/you-might-be-resizing-your-images-incorrectly/>
- Neo, E.X., Hasikin, K., Mokhtar, M.I., Lai, K.W., Azizan, M.M., Razak, S.A., and Hizaddin, H.F. (2022). Towards integrated air pollution monitoring and health impact assessment using federated learning: a systematic review. *Front Public Health*, 10(851553), 1 – 19. <https://doi.org/10.3389/fpubh.2022.851553>
- Putra, F. F. & Prabowo, Y. D. (2021). Low resource deep learning to detect waste intensity in the river flow, *Bulletin of Electrical Engineering and Informatics*, 10(5), 2724 – 2732.
- Redmon, J., Divvala, S., Girahick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779 – 788. <https://doi.org/10.48550/arXiv.1506.02640>
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks, *Advances in neural information processing systems*, 28, 779 – 788.
- Roboflow: Roboflow. (2023). About. [Online]. Available: <https://roboflow.com/about>
- Sakti, A.D., Sembiring, E., Rohayani, P., Fauzan, K. N., Anggraini, T. S., Santoso, C., Patricia, V. A., Ihsan, K. T. N., Ramadan, A. H., Arjasakusuma, S., & Candra, D. S. (2023). Identification of illegally dumped plastic waste in a highly polluted river in Indonesia using Sentinel-2 satellite imagery, *Scientific Reports* 13, 5039(2023). <https://doi.org/10.1038/s41598-023-32087-5>
- Setyawan, A. Y. & Kristian, Y. (2020). The Determination of River Cleanness Level Based on the Number of Waste Based of Digital Image Tabulation using Deep Convolutional Neural Network, *JICTE (Journal of Information and Computer Technology Education)*, 4(2), 21 – 26. <https://doi.org/10.21070/jicte.v4i2.971>
- Shandilya, S. K., Srivastav, A., Yemets, K., Datta, A., & Nagar, A. K. (2023). YOLO-based segmented dataset for drone vs. bird detection for deep and

- machine learning algorithms, *Data in Brief*, 50(2023), 1 – 10.
<https://doi.org/10.1016/j.dib.2023.109355>
- Sulistiyowati, E. S. (2018). *Dinamika Hidrosfer*. Klaten: Saka Mitra Kompetensi.
- Sutrisna, T. & Movanita, A. N. K. (10 Oktober 2022). 68 RT di Jakarta Terendam Banjir akibat Luapan Kali Ciliwung, Ini Lokasinya. [Online]. Available: <https://megapolitan.kompas.com/read/2022/10/10/13110001/68-rt-di-jakarta-terendam-banjir-akibat-luapan-kali-ciliwung-ini?page=all>
- Tsantekidis, A., Passalis, N., Tefas, A., Kannianen, J., Gabbouj, M., & Iosifidis, A. (2017). Forecasting Stock Prices from the Limit Order Book Using Convolutional Neural Networks, *2017 IEEE 19th Conference on Business Informatics (CBI)*, 7 – 12. <https://doi.org/10.1109/CBI.2017.23>
- Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS, *Machine Learning & Knowledge Extraction (MAKE)*, 5(4), 1680 – 1716.
<https://doi.org/10.3390/make5040083>
- UNSD (United Nations Statistics Division). (1997). *Glossary of Environment Statistics*. New York: United Nations.
- Valueva, M. V., Nagornov, N. N., Lyakhov, P. A., Valuev, G. V., & Chervyakov, N. I. (2020). Application of the residue number system to reduce hardware costs of the convolutional neural network implementation, *Mathematics and Computers in Simulation*. 177, 232 – 243.
<https://doi.org/10.1016/j.matcom.2020.04.031>
- van den Oord, A., Dieleman, S., Schrauwen, B., Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., & Weinberger, K. Q. (2013). *Deep content-based music recommendation*. Curran Associates, Inc. pp. 2643–2651.
- Viola, P. & Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features, *2001 IEEE computer society conference on computer vision and pattern recognition (CVPR)*, 1, I – I.
<https://doi.org/10.1109/CVPR.2001.990517>
- Wang, N., Liu, H., Li, Y., Zhou, W., & Ding, M. (2023). Segmentation and Phenotype Calculation of Rapeseed Pods Based on YOLO v8 and Mask R-Convolution Neural Networks, *Plants*, 12(3328), 1 – 16.
<https://doi.org/10.3390/plants12183328>
- Yohana, C., Griandini, D., & Muzambeq, S. (2017). Penerapan Pembuatan Teknik Lubang Biopori Resapan Sebagai Upaya Pengendali Banjir. *Jurnal Pemberdayaan Masyarakat Madani (JPMM)*, 1(2), 296 – 308.
<https://doi.org/10.21009/JPMM.001.2.10>

- Yutantri, V., Suryandari, R. Y., Putri, M. N., & Widyawati, L. F. (2023). Persepsi Masyarakat terhadap Faktor-Faktor Penyebab Banjir di Perumahan Total Persada Raya Kota Tangerang. *Journal of Regional and Rural Development Planning (Jurnal Perencanaan Pembangunan Wilayah Dan Perdesaan)*, 7(2), 199 – 214. <https://doi.org/10.29244/jp2wd.2023.7.2.199-214>
- Zailan, N. A., Azizan, M. M., Hasikin, K., Khairuddin, A. S. M., & Khairuddin, U. (2022). An automated solid waste detection using the optimized YOLO model for riverine management, *Frontiers in Public Health*, 10(1), 1 – 14. <https://doi.org/10.3389/fpubh.2022.907280>
- Zhao, Z. Q., Zheng, P., Xu, S.t., & Wu, X. (2019). Object detection with deep learning: A review, *IEEE transactions on neural networks and learning systems*, 30(11), 3212 – 3232.
- Zou, Z., Chen, K., Shi, Z., Guo, Y. & Ye, J. (2023). Object Detection in 20 Years: A Survey, *Proceedings of the IEEE*, 11(3), 257 – 276. <https://doi.org/10.1109/JPROC.2023.3238524>

LAMPIRAN

Lampiran 1. Data

Berikut adalah lampiran berupa link Google Drive yang berisi data yang digunakan dalam skripsi kali ini.

<https://drive.google.com/drive/folders/1CXB2tghdp5vkdJ1rGtLYsW7S99yC-z8K?usp=sharing>

Lampiran 2. Source Code

Berikut adalah lampiran berupa link Google Colab yang berisi algoritma YOLOv8 yang digunakan untuk keseluruhan proses *running*, mulai dari *training*, *validation*, hingga *testing*.

a. 50 epoch

<https://colab.research.google.com/drive/1txmytpFIPPiSjRHFoSFp4JdooYDF2fJo?usp=sharing>

b. 100 epoch

<https://colab.research.google.com/drive/1yYtGAznd6jtgBDq06VTh-Xlx0vuaeR1J?usp=sharing>

c. 200 epoch

<https://colab.research.google.com/drive/1Ths9sxdp112xjtt29tYe5NxfHUba07G?usp=sharing>

Lampiran 3. Dataset Roboflow

Berikut adalah lampiran berupa link Roboflow yang berisi dataset hasil pre-processing terhadap data.

<https://universe.roboflow.com/yes-h4okd/floating-wastes>

Lampiran 4. Folder Hasil *Training*, *Validasi*, dan *Testing*

Berikut adalah lampiran berupa link Google Drive yang berisi hasil *running* algoritma YOLOv8, terutama hasil *validation* dan *testing*.

a. 50 epoch

https://drive.google.com/drive/folders/1XTLMxGyWp9mRUvtSUEYFOHjqg_UjHqP3?usp=drive_link

b. 100 epoch

https://drive.google.com/drive/folders/1ZaYpCXnuXd6pTU1gp9vHnoW2Rgl_72Ea?usp=drive_link

c. 200 epoch

https://drive.google.com/drive/folders/1YirZvm1wEoEMoJm4Y6jONhA6mfWzNLIz?usp=drive_link

RIWAYAT HIDUP

Penulis bernama Rizky Theofilus Hutabarat, lahir di Jambi pada tanggal 21 Mei 2002. Penulis merupakan anak laki-laki dari pasangan Bapak St. J.P Hutabarat dan Ibu Y.R. br. Simanjuntak dan merupakan anak pertama dari 3 bersaudara. Penulis tinggal di Tebing Tinggi, Kabupaten Tanjung Jabung Barat, Jambi hingga saat ini. Penulis mulai mengenyam pendidikan di TK Yayasan Pendidikan Mayang Mangurai (YPMM) pada tahun 2006, kemudian melanjutkan ke jenjang pendidikan dasar di SD S YPMM mulai tahun 2008 hingga lulus tahun 2014. Pada tahun 2017, penulis menyelesaikan pendidikan menengah pertama di SMP S YPMM. Pada tahun yang sama penulis melanjutkan sekolah di SMAN Titian Teras H. Abdurrahman Sayoeti. Pada tahun 2020, penulis diterima untuk mengikuti pendidikan di Politeknik Statistika STIS.

Akhirnya pada tahun 2024, atas izin Tuhan Yang Maha Esa, dukungan dan do'a dari keluarga dan teman, penulis berhasil menyelesaikan pendidikan Program Diploma IV di Politeknik Statistika STIS.