

BAB I

PENDAHULUAN

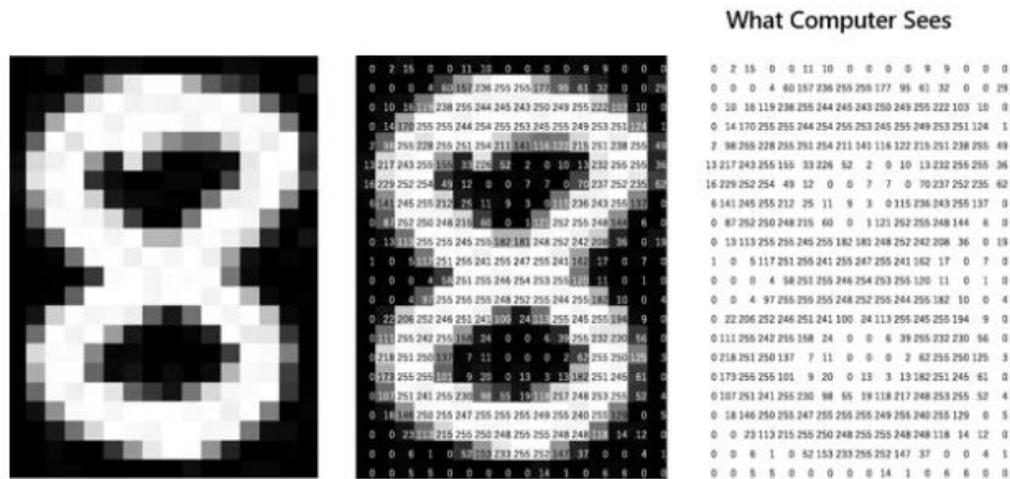
1.1 Latar Belakang Masalah

Menurut KBBI, revolusi berarti perubahan yang cukup mendasar dalam suatu bidang, sedangkan industri berarti kegiatan memproses atau mengolah barang dengan menggunakan sarana dan peralatan, misalnya mesin. Revolusi industri sering kali menjadi sebuah acuan untuk perubahan mendalam yang disebabkan oleh kemunculan teknologi-teknologi mutakhir. Perubahan cara-cara baru dalam melihat dunia baik dari sistem ekonomi maupun struktur sosialnya merupakan salah satu dampak dari revolusi industri (Schwab, 2019). Terjadinya perubahan yang sangat signifikan pada sistem kehidupan manusia merupakan dampak dari revolusi industri. Hal ini ditandai dengan kehadiran alat-alat yang sangat menunjang kehidupan manusia seiring dengan banyaknya permintaan baru setiap harinya.

Kini masyarakat tengah menghadapi revolusi industri keempat dimana teknologi digital sudah semakin mengagumkan. Hadirnya sensor buatan yang semakin kecil, internet yang semakin meluas, serta kecerdasan buatan ataupun pembelajaran mesin telah mengubah masyarakat dan perekonomian global (Schwab, 2019).

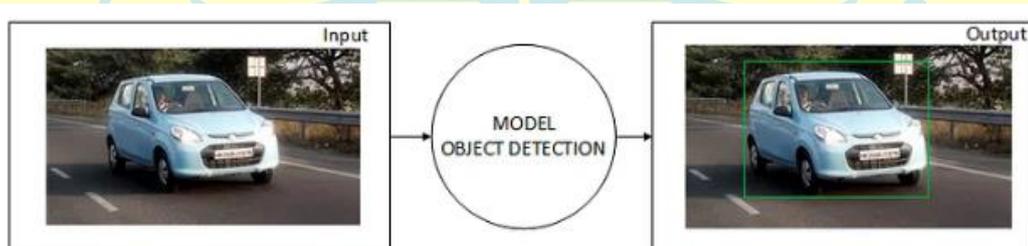
Salah satu perwujudan kecerdasan buatan atau pembelajaran mesin yang ada saat ini adalah sistem deteksi objek oleh komputer. Deteksi objek merupakan sebuah pengimplementasian dari teknik visi komputer guna mengidentifikasi objek yang ada di dalam sebuah gambar atau video (Nufus et al., 2021). Tidak seperti manusia yang dapat dengan mudahnya mengenali objek yang ada dihadapannya, komputer melihat sebuah gambar sebagai susunan dari angka-angka. Misalnya terdapat gambar tulisan tangan nomor 8, maka komputer akan membaca gambar tersebut sebagai sebuah piksel yang diwakili oleh satu angka. Dari susunan-susunan piksel tersebut jadilah sebuah matriks angka 2 dimensi

yang membentuk angka 8 (Kadam, 2020). Gambar 1.1 merupakan visualisasi bagaimana cara komputer melihat sebuah gambar.



Gambar 1.1 Cara Komputer Melihat Gambar (Kadam, 2020)

Secara sederhana, konsep sebuah komputer dalam melakukan deteksi objek dengan cara memindai terlebih dahulu seluruh bagian gambar kemudian komputer akan menentukan mana yang merupakan objek yang dimaksud dan mana yang bukan objek atau hanya latar saja. Ilustrasi deteksi objek dapat dilihat pada gambar 1.2.



Gambar 1.2 Ilustrasi Deteksi Objek (Salim, 2020)

Pertama-tama, sistem akan diberikan tugas untuk mendeteksi sebuah objek berupa mobil. Komputer akan menerima input berupa gambar yang pada ilustrasi tersebut tidak hanya berisi mobil saja, akan tetapi juga terdapat jalan, pohon, rambu lalu lintas, dan pembatas jalan. Dari sini, komputer akan meneruskan ke dalam sebuah model yang telah dilatih sebelumnya untuk mendeteksi apakah ada sebuah mobil yang terdapat dalam gambar tersebut. Kemudian outputnya akan berupa koordinat yang merepresentasikan posisi objek yang ada di dalam gambar.

Dan sistem akan membuat sebuah *bounding box* di sekitar objek sebagai representasi dari titik koordinat tersebut (Salim, 2020).

Deteksi objek sudah banyak diterapkan di kehidupan sehari-hari, beberapa di antaranya adalah alat pengukur suhu di stasiun yang bisa mendeteksi suhu manusia lewat kamera CCTV, aplikasi pengenalan masalah pada kulit wajah seseorang, *automated checkout* di supermarket, dan *traffic counting*. *Traffic counting* sendiri merupakan sebuah survei yang dilakukan untuk menghitung volume kendaraan yang melewati jalan tertentu dalam jangka waktu tertentu. *Traffic counting* memiliki banyak manfaat, diantaranya adalah pengumpulan data lalu lintas, pemahaman karakteristik lalu lintas, identifikasi komposisi kendaraan, serta evaluasi kinerja lalu lintas di suatu jalan (Ruslianto et al., 2021). Data-data tersebut kemudian dapat diolah menjadi sebuah informasi yang dapat digunakan untuk kebutuhan membuat desain jalan, mengukur kinerja jalan, dan menetapkan kebijakan-kebijakan yang sekiranya perlu diberlakukan mengingat volume kendaraan yang terus meningkat setiap tahunnya.

Umumnya, *traffic counting* melakukan penghitungan secara manual untuk menghitung setiap jenis kendaraannya satu per satu. Hal ini sangat membuang waktu bagi orang yang mengerjakannya. Jika rentang waktu pada *traffic counting* selama 1 hari atau 24 jam, berarti ia harus menghitung semua jenis kendaraan di tempat tersebut selama 24 jam. Belum lagi jika *counter* harus menghitung jalan raya bervolume tinggi atau multi jalur, hal ini akan berdampak pada keakuratan dan akan sangat tidak efisien. Maka dari itu, perhitungan *traffic counting* secara manual membutuhkan waktu dan biaya yang tinggi (Penny, 2021). Belum lagi tipe kendaraan menurut Bina Marga dibagi menjadi 8 kelas, yaitu sepeda motor, mobil sedan, angkutan sedang, mikro truk, bus, truk 2 sumbu, truk 3 sumbu, dan kendaraan tidak bermotor. Pembagian tipe kendaraan menurut Bina Marga terdapat pada tabel 1.1 (Arnis, 2012).

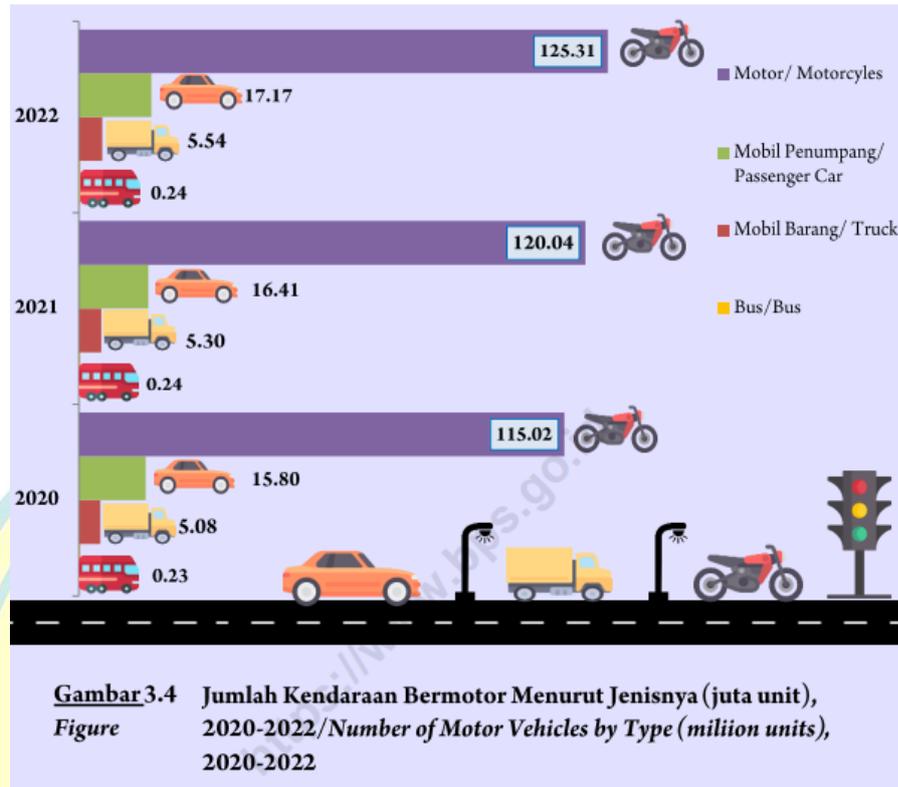
Tabel 1.1 Tipe Kendaraan Menurut Bina Marga

Kelas	Nama Kelas	Contoh Kendaraan
1	Sepeda motor	Sepeda motor, skuter, kendaraan roda 3
2	Sedan, jeep, wagon	Sedan, jeep, station wagon

3	Angkutan sedang	Opelet, <i>pick up</i> opelet, suburban, kombi, dan minibus
4	<i>Pick up</i> , mikro truk	<i>Pick up</i> , mikro truk, dan mobil hantaran
5a	Bus kecil	Bus kecil
5b	Bus besar	Bus besar
6	Truk 2 sumbu	Truk 2 as
7a	Truk 3 sumbu	Truk 3 as
7b	Truk gandeng	Truk gandengan
7c	Truk semitrailer	Truk tempelan
8	Tidak bermotor	Sepeda, becak, dokar, keretek, dan andong

Menurut survei yang dilakukan oleh Badan Pusat Statistik, jumlah kendaraan bermotor di Indonesia pada tahun 2022 mencapai 148.212.865 unit (BPS, 2022). Angka ini selalu bertambah dari tahun ke tahun mengikuti pertumbuhan penduduk di Indonesia karena kebutuhan akomodasi untuk mobilisasi seseorang juga semakin meningkat. Hal ini sangat tidak efektif dan efisien jika seseorang harus menghitung jenis kendaraan secara manual mengingat banyaknya jumlah kendaraan yang ada di negeri ini.

Dari permasalahan tersebut, dibutuhkan sebuah sistem yang dapat menghitung otomatis kendaraan berdasarkan jenisnya untuk membantu *counter*. Sistem yang dikembangkan berupa *object detection* dimana indikator keberhasilannya adalah komputer harus bisa menghitung jumlah kendaraan berdasarkan jenisnya dari video yang diinput. Video yang dimaksud adalah video yang berisi berbagai jenis kendaraan untuk dihitung setiap jenis kendaraannya.



Gambar 1.3 Jumlah Kendaraan Bermotor (BPS, 2023)

Sistem deteksi jenis kendaraan bukan lagi sesuatu hal yang baru. Sebelumnya, sudah banyak penelitian yang membahas sistem deteksi jenis kendaraan ini, diantaranya adalah Alfian dkk. dengan judul “*Counting Various Vehicles using YOLOv4 and DeepSORT*”. Penelitian ini menghasilkan nilai mAP sebesar 66.17% dengan 4000 iterasi (Kusumah et al., 2023). Berdasarkan hasil penelitian tersebut, metode ini dapat diterapkan untuk mengklasifikasikan dan menghitung kendaraan dalam lingkup terbatas. Selain itu, Yovi Pratama dan Errissya Rasywir dalam penelitiannya yang berjudul “Eksperimen Penerapan Sistem *Traffic Counting* dengan Algoritma YOLO (*You Only Look Once*) V.4.” menunjukkan hasil yang didapatkan dari penelitian ini cenderung baik walaupun tidak seluruh hasil penggunaan algoritma ini sempurna pada semua data.

Beberapa algoritma yang dapat menunjang pengembangan sistem deteksi jenis kendaraan, di antaranya adalah algoritma Faster R-CNN, MobileNet, YOLO, Mask R-CNN, dan sebagainya (Salim, 2020). Penelitian ini menggunakan algoritma YOLO versi 4 dimana YOLO merupakan akronim dari You Only Look Once. Dinamakan demikian karena YOLO dapat melihat gambar sekaligus dan

langsung memberikan jawaban tentang dimana objek berada dan jenis objeknya. YOLO membagi gambar menjadi kotak-kotak kecil dan langsung mencoba mengenali objek di setiap kotak. Hal ini membuat YOLO dapat mendeteksi objek secara *real-time* dan memakan waktu yang lebih sedikit daripada algoritma lainnya (Anggarwal, 2020).

Penelitian ini menggunakan algoritma YOLO karena YOLO merupakan algoritma *pre-trained learning*, yaitu model yang sudah dilatih sebelumnya. *Transfer learning* merupakan proses dimana model *machine learning* yang telah digunakan untuk memecahkan sebuah kasus sebelumnya dapat digunakan dalam kasus lain karena adanya keterkaitan masalah. *Transfer learning* merupakan teknik dimana model jaringan saraf yang sudah dilatih sebelumnya untuk memecahkan suatu masalah yang memiliki kemiripan dengan masalah yang sedang dipecahkan, sehingga memungkinkan satu atau lebih lapisan (layer) dari model yang sudah dilatih dapat digunakan dalam model baru yang sedang dikembangkan. Keuntungan menggunakan *transfer learning* adalah mengurangi waktu latih model dan meminimalisir kesalahan yang general (Jauhari, 2020).

Adapun teknik evaluasi yang digunakan untuk mengukur kinerja sistem adalah *Mean Average Precision* atau mAP. Umumnya, mAP digunakan untuk mengukur evaluasi dengan kasus deteksi objek. Dikatakan demikian karena mAP tidak memuat nilai *true negative* (TN) dalam *confusion matrix*-nya karena dalam deteksi objek keluarannya tidak akan menghasilkan *true negative* (Kristian, 2021).

Untuk mengembangkan sistem deteksi jenis, penelitian ini menggunakan *framework* Flask untuk menyertakan model *machine learning* pada sebuah situs web karena sistem ini dikembangkan menggunakan bahasa pemrograman Python. Flask termasuk ke dalam *micro-framework* karena Flask tidak membutuhkan *tools* ataupun *library* tertentu pada saat pengaplikasiannya. Harapannya, pihak-pihak yang melakukan *traffic counting*, seperti Kementerian Perhubungan dapat memudahkan dalam melakukan surveinya karena adanya sistem *traffic counting* yang menjadi *output* pada penelitian ini.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah sebelumnya, dapat diidentifikasi beberapa masalah berikut.

1. Perhitungan *traffic counting* secara manual membutuhkan biaya, tenaga, dan waktu yang tinggi. Hal ini menegaskan bahwa sulitnya menghitung berbagai jenis kendaraan dalam satu waktu secara manual.
2. Sistem pendeteksi jenis kendaraan yang sudah ada memiliki akurasi yang tidak terlalu tinggi, sehingga memungkinkan terjadinya kesalahan deteksi jenis kendaraan saat menggunakan sistem tersebut.
3. Sistem pendeteksi jenis kendaraan yang sudah ada tidak bisa mendeteksi jenis kendaraan dari berbagai macam sudut pandang video yang digunakan untuk *testing*, sehingga sulit mendeteksi jenis kendaraan berdasarkan video yang dimiliki.

1.3 Pembatasan Masalah

Adapun deteksi objek akan difokuskan pada permasalahan sebagai berikut.

1. Sumber data deteksi objek yang digunakan berasal dari internet dan bersifat *open source*. Sedangkan untuk pengujian oleh pengguna merupakan input berupa unggah video.
2. Data yang digunakan mencakup 8 jenis kendaraan yang kemudian ditetapkan menjadi 8 kelas *output*, yaitu motor, sedan_jeep_wagon, angkutan_sedang, mikro_truk, bus, truk_2_sumbu, truk_3_sumbu, tidak_bermotor.
3. Aplikasi sistem deteksi jenis kendaraan yang dikembangkan berupa situs web lokal sederhana menggunakan *framework* Flask.

1.4 Perumusan Masalah

Berdasarkan latar belakang, identifikasi, dan batasan masalah yang telah dipaparkan, maka dapat dirumuskan permasalahan yang akan dibahas pada penelitian ini, yaitu “Bagaimana Hasil Deteksi Objek Menggunakan YOLOv4 untuk Mendeteksi Jenis Kendaraan?”

1.5 Tujuan Penelitian

Tujuan penelitian yang akan dicapai pada penelitian ini adalah sebagai berikut.

1. Mengukur nilai *Mean Average Precision* (mAP) yang didapatkan dari hasil deteksi objek jenis kendaraan menggunakan algoritma YOLOv4.
2. Merancang sebuah aplikasi berupa situs web lokal sederhana untuk menghitung jenis kendaraan dengan inputan berupa video.

1.6 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat, antara lain sebagai berikut.

1. Dapat mendeskripsikan nilai mAP yang didapatkan dari hasil deteksi objek jenis kendaraan menggunakan YOLOv4, sehingga dapat dijadikan perbandingan terhadap algoritma lainnya.
2. Hasil dari penelitian dapat digunakan oleh pihak tertentu untuk menghitung jenis kendaraan guna mencapai tujuan dari *traffic counting*.
3. Sebagai referensi untuk penelitian selanjutnya.

