

**SISTEM CERDAS MENGGUNAKAN ALGORITMA YOLOv4 UNTUK MENDETEKSI
PENGUNAAN ROMPI KESELAMATAN KERJA DI AREA KERJA**

**UJIAN AKHIR SEMESTER
RISET INFORMATIKA**



Oleh :

FARKHAN

NPM. 20081010060

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
JAWA TIMUR**

2022

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT karena atas karunianya penulis dapat menyelesaikan laporan proyek akhir dengan judul “Sistem Cerdas Menggunakan Algoritma Yolov4 Untuk Mendeteksi Penggunaan Rompi Keselamatan Kerja Di Area Kerja” sebagai hasil perkuliahan mata kuliah Riset Informatika.

Laporan ini dapat diselesaikan tepat pada waktunya tidak terlepas dari bantuan dan dukungan dari berbagai pihak, yakni Bapak Budi Nugroho, S.Kom., M.Kom. selaku dosen pengampu, dan pihak lain yang tidak disebutkan. Untuk itu penulis ucapkan terima kasih atas kontribusi bantuan dalam berbagai bentuk.

Penulis menyadari bahwa masih banyak kesalahan dalam penyusunan laporan ini. Maka dari itu penulis sangat mengharapkan kritik dan saran seluas-luasnya dari pembaca yang kemudian akan penulis jadikan sebagai evaluasi. Demikian semoga laporan proyek akhir ini bisa diterima sebagai ide atau gagasan yang menambah kekayaan intelektual dan dapat bermanfaat bagi pembaca dan juga untuk penulis sendiri.

Surabaya, 21 Desember 2022

Penulis

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	4
1.3. Batasan Masalah	4
1.4. Tujuan	5
BAB II KAJIAN PUSTAKA	6
2.1. Citra Digital	6
2.2. Pengolahan Citra Digital.....	7
2.3. Konvolusi	7
2.4. You Only Look Once.....	8
2.5. YOLOv4.....	10
BAB III METODOLOGI PENELITIAN	12
3.1. Tahapan Penelitian	12
3.2. Studi Literatur	13
3.3. Pengumpulan Data	13
3.4. Pra-proses Data	13
3.4.1. Anotasi Data	13
3.4.2. Akuisisi Data	14
3.5. Pelatihan Model	15
3.5.1. Membangun Arsitektur YOLOv4	15
3.5.2. Konfigurasi Hyperparameter YOLOv4.....	16
3.5.3. Memuat Bobot Pre-trained YOLOv4 dan Dataset	17
3.5.4. Pelatihan Dataset pada Model YOLOv4.....	18
3.6. Evaluasi Model	18
3.7. Penanaman Model ke dalam Aplikasi	18
3.8. Skenario Penelitian.....	19

BAB I PENDAHULUAN

1.1. Latar Belakang

Di Indonesia, keselamatan selalu memiliki kaitan dengan kesehatan, hal ini karena sistem keselamatan yang digunakan adalah terkait dengan program K3 atau Keselamatan dan Kesehatan Kerja. K3 memiliki filosofi dasar, yaitu menjamin keutuhan dan kesempurnaan kepada para pekerja melalui perlindungan atas keselamatan dan kesehatan dalam menjalankan pekerjaannya (Kepmenaker, 19993). K3 adalah suatu bentuk upaya perlindungan kepada tenaga kerja dan orang lain yang memasuki atau berada di area tempat kerja terhadap risiko dan bahaya dari akibat kecelakaan kerja. Dalam UU No. 13 Tahun 2003 pada Pasal 86 tentang Keselamatan dan Kesehatan Kerja, menjelaskan bahwa setiap pekerja atau buruh memiliki hak untuk memperoleh perlindungan atas keselamatan dan kesehatan kerja dan pada Pasal 87 dijelaskan bahwa setiap perusahaan memiliki kewajiban dalam menerapkan sistem manajemen Keselamatan dan Kesehatan Kerja dan jika terdapat pekerja yang mengalami kecelakaan, maka perusahaan berkewajiban untuk menanggung biayanya (Restuputri et al., 2015).

Setiap pekerja perlu memiliki pemahaman dan mampu menerapkan konsep dasar K3 untuk menghindari terjadinya risiko dan hazard atau bahaya dalam melaksanakan pekerjaannya. Risiko merupakan kemungkinan kecelakaan yang akan atau dapat terjadi dan bisa mengakibatkan kerusakan, sedangkan hazard atau bahaya adalah suatu keadaan atau kegiatan yang berpotensi mengakibatkan terjadinya cedera atau luka (Pratiwi, 2020). Kecelakaan kerja merupakan kecelakaan yang terjadi di dalam hubungan kerja, seperti kecelakaan yang terjadi pada perjalanan berangkat dari rumah menuju tempat kerja, kecelakaan yang terjadi ketika melaksanakan pekerjaan di tempat kerja, hingga kecelakaan yang terjadi ketika perjalanan pulang ke rumah dari tempat kerja. Beberapa sebab utama kecelakaan kerja dapat terjadi di antaranya adalah keadaan atau kondisi tidak aman atau *unsafe condition*, tindakan pekerja yang tidak aman atau *unsafe condition*, dan interaksi manusia dengan sarana pendukung kerja (Putra, 2017).

Menurut Peraturan Menteri Tenaga Kerja dan Transmigrasi Republik Indonesia No. 8 Tahun 2010, para pekerja memiliki kewajiban menggunakan Alat Pelindung Diri atau APD saat memasuki area kerja. APD merupakan alat yang memiliki fungsi sebagai pelindung bagian

tubuh yang berisiko terhadap adanya potensi bahaya dari kecelakaan kerja pada ruang lingkup tempat kerja. APD memang tidak dapat melindungi pekerja secara sempurna, tetapi dapat meminimalkan tingkat keparahan yang mungkin dapat terjadi (Adiwibowo et al., 2020). Pada kondisi sebenarnya, penggunaan APD sering kali dianggap remeh atau kurang penting oleh para pekerja, terutama pada pekerja yang menganggap ruang lingkup kerjanya adalah aman. Padahal penggunaan APD ini dapat berpengaruh terhadap keselamatan dan kesehatan kerja bagi pekerja itu sendiri. Beberapa penelitian menunjukkan hasil bahwa faktor manusia memegang peranan penting dalam munculnya kecelakaan kerja (Ulum et al., 2021). Mengutip dari *cnnindonesia.com* Menteri Ketenagakerjaan (Menaker), Ida Fauziyah, dalam Peringatan Bulan K3 Nasional di Kilometer Nol Sabang mengatakan “Merujuk pada data BPJS Ketenagakerjaan tahun 2019 terdapat 114.000 kasus kecelakaan kerja, tahun 2020 terjadi peningkatan pada rentang Januari hingga Oktober 2020 BPJS Ketenagakerjaan mencatat terdapat 177.000 kasus kecelakaan kerja.” Angka kecelakaan yang tinggi salah satunya disebabkan oleh tidak disiplinnya pekerja dalam menggunakan APD. Untuk mengurangi hal tersebut dibutuhkanlah pengawas yang bertugas memantau pelaksanaan K3 dan pengecekan APD. Namun, pelaksanaan tersebut masih menggantungkan manusia dan menggunakan cara manual untuk melaksanakannya. Secara umum, pengawasan yang menggantungkan manusia dalam durasi lama akan menurun tingkat efisiensinya dan membuang cukup banyak waktu.

Salah satu solusi untuk mengatasi penggunaan sumber daya manusia dalam pengawasan adalah dengan otomatisasi. Dalam proses pemantauan para pekerja yang tidak menggunakan APD otomatis memiliki peran yang penting di antaranya dapat mengurangi beban kerja pengawasan terhadap pekerja yang tidak menggunakan APD. Solusi yang tepat untuk menyelesaikan masalah ini adalah menggunakan *object detection* dengan metode *deep learning* yang dapat diterapkan pada video kamera pengawas di lingkungan kerja wajib alat pelindung diri (Rubaiyat et al., 2017). Teknologi tersebut dapat dimanfaatkan sebagai alat atau sistem cerdas pemrosesan citra, yaitu mendeteksi keberadaan objek secara akurat dan dalam waktu yang cepat. Dalam penelitian ini, fokus objek APD yang akan dideteksi adalah rompi keselamatan kerja, yaitu diketahui apakah terdapat objek rompi keselamatan kerja atau tidak pada citra gambar yang akan dideteksi.

Akhir-akhir ini, teknologi *deep learning* menjadi topik perbincangan hangat dan semakin banyak digunakan karena berhasil mutakhir yang diperoleh seperti pada *image classification*, *object detection*, hingga *natural language processing*. *Deep learning*

memberikan penawaran kinerja yang lebih baik dibandingkan metode-metode yang lain dalam memecahkan masalah-masalah yang sulit diselesaikan oleh manusia (Kamilaris & Prenafeta-Boldu, 2018). Seiring semakin pesatnya perkembangan *deep learning*, banyak model-model baru yang ditemukan untuk pengenalan dan pendeteksian objek pada *computer vision*. Beberapa di antaranya adalah *Region based Convolutional Neural Networks* (R-CNN, fast R-CNN, Faster R-CNN), *spatial Pyramid Network* (SPPNet), *RetinaNet*, dan *You Only Look Once* (YOLO) (Aini et al., 2021).

Dari beberapa algoritma deteksi objek tersebut, YOLO adalah yang tercepat dalam proses deteksi kemudian disusul oleh SSD dan RCNN. Namun, dapat dikatakan bahwa kondisi juga memengaruhi algoritma mana yang akan dipilih. Jika kumpulan data relatif kecil dan tidak memerlukan hasil *realtime*, maka yang terbaik adalah RCNN. YOLO menunjukkan kinerja terbaik secara keseluruhan dan patut digunakan jika membutuhkan analisis deteksi objek pada video langsung atau *realtime*. Selain itu, YOLO adalah yang paling baru dirilis dari ketiganya dan secara aktif disumbangkan secara luas oleh komunitas sumber terbuka (Srivastava et al., 2021). YOLO adalah model terpadu yang memiliki fungsi untuk melakukan deteksi objek pada citra yang diterimanya. YOLO memiliki kecepatan dan keakuratan yang diklaim lebih baik untuk diterapkan pada *computer vision*. Desain YOLO memungkinkan pelatihan ujung ke ujung dan kecepatan waktu nyata sambil mempertahankan presisi rata-rata yang tinggi. YOLO memberikan prediksi *bounding box* dari objek yang berada pada citra, kemungkinan lokasi dan probabilitas dari semua *class* pada satu waktu sehingga menjadi pilihan yang paling baik untuk digunakan pada permasalahan deteksi objek (Redmon et al., 2016).

Pencipta YOLO, yaitu Joseph Redmon memberikan beberapa perkembangan di antaranya adalah perbaikan atau peningkatan akurasi dan kecepatan waktunya. Perkembangan yang pertama adalah merilis YOLO9000 atau sering dikenal sebagai YOLOv2. Pada YOLOv2, perkembangan yang diberikan adalah penambahan normalisasi *Batch*, menaikkan resolusi dari *classifier* menjadi 448x448 yang semula berukuran 224x224, menghapus lapisan yang terhubung penuh dan mengganti dengan penambahan *Anchor Boxes*, menambahkan pelatihan multi-skala atau *multi-scale training* dan menggunakan darknet-19 (Redmon & Farhadi, 2017). Perkembangan yang selanjutnya yang diberikan oleh Joseph Redmon adalah menggunakan darknet-33 yang menggantikan darknet-19 menggunakan tiga skala deteksi yang sebelumnya hanya satu skala, menggunakan regresi

logistik untuk memprediksi skor dari *bounding box* untuk dapat mendeteksi objek lebih baik. Pada YOLOv3 fungsi *softmax* diganti dengan pengklasifikasi logistik independen untuk menentukan kelas gambar input. (Redmon & Farhadi, 2018). Perkembangan berikutnya tidak dikembangkan oleh Joseph Redmon melainkan oleh Alexy Bochkovskiy menjadi YOLOv4. Pada YOLOv4 perkembangan yang dilakukan adalah meningkatkan *Average Precision* (AP) sebesar 10% dan *Frame Per Second* (FPS) sebesar 12% dibandingkan dengan YOLOv3, pada YOLOv4 terdapat beberapa *backbone* yang dapat dipilih, yaitu CSPResNeXt50, CSPDarknet53, dan EfficientNet-B3. Namun, setelah beberapa percobaan, CSPDarknet53 menjadi model atau *backbone* yang paling optimal. YOLOv4 menggunakan teknik pengembangan baru yang diberi nama Bag-of-Freebies (BoF) dan Bag-of-Special (BoS) pada tahap pelatihannya, tujuannya adalah agar dapat meningkatkan performa serta akurasi model tanpa memengaruhi lama waktu proses *training* (Bochkovskiy et al., 2020).

Berdasarkan penjelasan dari latar belakang tersebut dan berdasarkan kebutuhan akan akurasi kecepatan deteksi objek, maka dalam penelitian ini penulis melakukan penelitian dengan judul “Sistem Cerdas Menggunakan Algoritma YOLOv4 Untuk Mendeteksi Penggunaan Rompi Keselamatan Kerja Di Area Kerja”, untuk mengetahui hasil deteksi penggunaan rompi kerja menggunakan YOLOv4 dan mengetahui performa dari YOLOv4 dalam melakukan deteksi penggunaan rompi kerja. Keluaran yang dihasilkan dalam penelitian ini, yaitu model identifikasi penggunaan rompi kerja di area kerja menggunakan YOLOv4 yang ditanamkan pada sebuah aplikasi.

1.2. Rumusan Masalah

Berdasarkan penjabaran latar belakang yang telah dijabarkan, maka didapatkan rumusan masalah yang akan dibahas antara lain sebagai berikut.

1. Bagaimana penerapan algoritma *You Only Look Once* (YOLOv4) untuk mendeteksi penggunaan rompi keselamatan kerja di area kerja?
2. Bagaimana hasil evaluasi performa dari algoritma YOLOv4 dalam melakukan deteksi penggunaan rompi keselamatan kerja di area kerja?

1.3. Batasan Masalah

Adapun batasan masalah yang digunakan penulis agar pembahasan dalam penelitian tidak menyimpang dari pembahasan adalah sebagai berikut.

1. Pada penelitian ini obyek dari alat pelindung diri yang akan dideteksi hanya berfokus pada rompi keselamatan kerja.
2. Dataset yang digunakan adalah dataset yang diambil dari sumber terbuka kaggle dan dianotasi menyesuaikan format YOLOv4.
3. Uji coba deteksi pada penelitian dilakukan menggunakan gambar dan rekaman video dari internet, sedangkan uji coba secara langsung didemokan menggunakan peneliti sebagai bahan uji coba.

1.4. Tujuan

Penelitian ini bertujuan untuk.

1. Mengimplementasikan algoritma YOLOv4 untuk mendeteksi penggunaan rompi keselamatan kerja di area kerja.
2. Mengetahui seberapa baik performa dari algoritma YOLOv4 pada deteksi penggunaan rompi keselamatan kerja di area kerja.
3. Membuat model pembelajaran mesin berbasis algoritma YOLOv4 yang dapat ditanamkan pada sistem deteksi penggunaan rompi keselamatan kerja di area kerja.

BAB II KAJIAN PUSTAKA

2.1. Citra Digital

Citra digital merupakan citra yang berbentuk gambar dua dimensi dan dapat dilakukan pengolahan atau komputasi pada komputer dengan cara komputer menyimpan dan mengolah data numerik dari nilai masing-masing piksel citra (Munantri et al., 2020). Citra digital dapat berupa file gambar yang tersimpan pada komputer dengan format file (bmp, jpg, png, atau format lainnya) (Sindar & Sinaga, 2017).

Citra digital pada dasarnya adalah kumpulan matriks yang tersusun dari indeks baris matriks dan kolom matriks yang mewujudkan suatu titik dari citra tersebut. Elemen matriks dari citra dapat disebut sebagai piksel pada citra digital berupa nilai bilangan integer yang mempresentasikan intensitas suatu amplitudo dari gray level pada piksel tersebut (Fikriya et al., 2017). Citra digital dapat diilustrasikan sebagai fungsi $f(x, y)$ yang memiliki ukuran M baris dengan N kolom dan x, y merupakan koordinat spasial. Amplitudo dari f pada koordinat (x, y) itulah disebut gray level dari piksel atau titik citra tersebut.

Bentuk fungsi matriks dari citra digital dapat dituliskan sebagai berikut.

$$f(x) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

Pada citra digital sistem koordinat mengikuti asas pemindaian layar TV dengan koordinat (x,y) di mana x adalah posisi kolom dan y adalah posisi baris (Nafi'iyah, 2015).

Citra digital memiliki beberapa jenis diantaranya adalah citra berwarna, citra biner, dan citra keabuan.

a. Citra Berwarna

Citra berwarna sering disebut sebagai citra RGB, hal tersebut didasarkan pada penyajian warna dengan bentuk komponen merah (R), hijau (G), dan biru (B). Setiap komponen warna pada citra digital memiliki nilai dengan rentang 0 hingga 255.

b. Citra Biner

Citra biner adalah citra yang memiliki dua nilai piksel yaitu 1 sebagai warna putih dan 0 sebagai warna hitam sehingga citra biner hanya terdiri dari dua warna.

c. Citra keabuan

Citra keabuan atau disebut sebagai citra grayscale merupakan citra digital yang nilai dari setiap pikselnya hanya terdiri dari satu nilai kanal yang menggambarkan tingkat intensitasnya (Tarigan et al., 2016).

2.2. Pengolahan Citra Digital

Pengolahan citra digital merupakan bidang ilmu yang mempelajari berbagai teknik seperti pembentukan citra, pengolahan citra dan analisis dari citra untuk mendapatkan informasi agar mudah dipahami oleh pengguna (Ratna, 2020). Pengolahan pada citra digital memiliki beberapa tujuan diantaranya adalah melakukan perbaikan atau peningkatan pada kualitas dari citra (image enhancement), perbaikan atau pemulihan citra pada kondisi semula (image restoration), ekstraksi fitur dari citra (feature extraction), pengurangan atau pereduksian ukuran dari file citra (image compression), dan lain lain (Basuki & Ramadijanti, 2005).

Langkah pada pengolahan digital dimulai dari proses pengambilan citra (image acquisition) yang dapat dilakukan dengan alat pemindai, sensor kamera dan lain-lain. Proses selanjutnya yaitu melakukan persiapan (preprocessing) seperti resizing atau melakukan enhancement. Setelah dilakukan preprocessing maka dilakukan pembagian citra kedalam bagian-bagian tertentu untuk memisahkan objek-objek yang diinginkan. Proses ini sering disebut dengan istilah image segmentation. Setelah objek dilakukan pemisahan Langkah berikutnya yaitu melakukan representation dan description untuk mengetahui kebenaran objek yang telah dipisahkan. Langkah terakhir dari proses pengolahan citra digital yaitu melakukan pengenalan dan interpretasi untuk memberikan label dan arti kedalam objek (Rianto & Harjoko, 2017).

2.3. Konvolusi

Konvolusi adalah salah satu metode yang digunakan untuk melakukan pengolahan citra digital. Konvolusi merupakan operator matematika yang memiliki fungsi menggabungkan dua array. Array dapat berbeda ukuran tetapi dengan dimensi yang sama. Nilai keluaran dari konvolusi dihasilkan dari kombinasi linear nilai masukan pada piksel tertentu. Konvolusi

merupakan teknik penghalusan atau perjelas citra dengan cara mengganti nilai piksel dengan nilai piksel yang berdekatan dengan nilai aslinya. Berikut adalah fungsi $f(x)$ dan $g(x)$ pada konvolusi:

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(a) \cdot g(x - a) da \quad (2.2)$$

Tanda (*) adalah operator konvolusi dan variabel a merupakan pembantu.

Dalam operasi konvolusi, pengolahan citra yang dilakukan merupakan operasi diskrit dan bentuk satu dimensi, berikut adalah bentuk operasinya:

$$h(x) = f(x) * g(x) = \sum_{a=-\infty}^{\infty} f(a) \cdot g(x - a) \quad (2.3)$$

Dalam bentuk dua dimensi, berikut adalah fungsi operasi konvolusinya:

Fungsi Diskrit:

$$h(x, y) = f(x, y) * g(x, y) = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} f(a, b) \cdot g(x - a, y - b) \quad (2.4)$$

Fungsi Integral:

$$h(x, y) = f(x, y) * g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b) \cdot g(x - a, y - b) dadb \quad (2.5)$$

Pada fungsi penapis, $g(x, y)$ merupakan konvolusi *filter*, *mask*, *kernel*, atau *template*. Konvolusi *kernel* pada fungsi diskrit berbentuk matriks yang biasa berukuran 3x3 (Soeparno & Kun, 2012).

2.4. You Only Look Once

You Only Look Once atau YOLO adalah salah satu algoritma yang memanfaatkan model jaringan syaraf tiruan pada CNN yang digunakan untuk melakukan deteksi objek diciptakan oleh Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi pada tahun 2016 (Redmon et al., 2016). Dalam melakukan proses deteksi objek YOLO menggunakan jaringan konvolusi kemudian memprediksi beberapa kemungkinan kotak pembatas (bounding box) beserta kelasnya secara bersamaan. YOLO secara penuh melatih gambar dan mengoptimalkan kerja deteksi secara langsung dan cepat. Dengan membingkai deteksi sebagai masalah regresi membuat alur menjadi tidak rumit dan menjadi lebih cepat. Tidak seperti pada metode sliding window dan region proposal-based selama proses pelatihan dan pengujian YOLO melihat semua gambar sehingga secara tidak langsung mengkodekan informasi kelas dari gambarnya.

YOLO bekerja dengan cara membagi gambar menjadi grid atau region dengan ukuran $s \times s$. Jika terdapat objek pada salah satu grid maka grid tersebut bertanggung jawab untuk

mendeteksi objeknya dengan memprediksi bounding box B dan menghitung nilai confidence yang mencerminkan keyakinan model terhadap keberadaan objek dan keakuratan box. Nilai confidence di definisikan dengan persamaan berikut:

$$confidence = Pr(object) * IOU_{pred}^{truth}$$

Pada persamaan tersebut nilai dari Pr(Object) merupakan nilai dari objektivitas dan akan bernilai nol jika tidak terdapat objek yang teridentifikasi. IOU adalah nilai dari rasio intersection Over Union antara kotak yang benar dengan kotak prediksi. Terdapat lima nilai prediksi dalam setiap bounding box yaitu xy.w.h, dan conf. Nilai (x,y) merupakan koordinat yang mewakili pusat kotak terhadap batas dari grid, nilai (wh) merupakan nilai relatif dari lebar dan tinggi yang diprediksi terhadap gambar sedangkan conf adalah nilai konfiden. Tidak hanya itu, setiap grid memiliki tugas untuk memprediksi kemungkinan C kelas kondisional dengan notasi Pr(Class, Object). Kemungkinan hanya akan di proses pada region atau grid yang terdapat objek. Untuk menghitung kemungkinan kelas kondisional dengan konfidensial dari kotak prediksi didefinisikan pada persamaan berikut:

$$Pr(Class_i|Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(Class_i) * IOU_{pred}^{truth}$$

Dari setiap kotak akan diberikan nilai dari sebuah kelas secara spesifik yang memberikan besaran konfiden dari kelas yang muncul pada kotak tersebut dan keakuratan kotak prediksi pada objek (Redmon et al., 2016).

YOLOV2 atau YOLO9000 merupakan pengembangan dari YOLO versi sebelumnya. Perubahan pada YOLOv2 yaitu terletak pada arsitektur yang digunakan dan cara memprediksi objek. Pada YOLOv2 tetap melakukan pembagian grid gambar sesuai dengan YOLO sebelumnya yaitu berdasarkan $s \times 5$. Arsitektur yang digunakan pada YOLOV2 adalah Darknet-19 dengan 19 convolutional layer, 5 Pooling layer. Pada arsitektur Darknet-19 fully connected layer dihilangkan dan menggunakan anchor box untuk melakukan prediksi dari bounding box agar dapat mengambil ukuran gambar yang berbeda-beda. Hasil nilai dari bounding box setiap anchor diseleksi berdasarkan nilai IOU, jika memiliki nilai IOU terbesar maka ditetapkan sebagai hasil deteksi. Pada YOLOv2 juga dikecilkan ukuran gambar masukan dari 448x448 menjadi 416x416 sehingga dihasilkan peta fitur ganjil yaitu 13x13 yang memiliki pusat tunggal. YOLOv2 menggunakan algoritma K-Means pada dimensi bounding box untuk

mendapatkan prioritas yang lebih baik yang mengarah pada skor IOU yang baik, yang tidak bergantung pada ukuran kotak (Redmon & Farhadi, 2017).

YOLOv3 adalah YOLO versi terakhir yang dikembangkan oleh Joseph Redmon dan Ali Farhadi. YOLOv3 memiliki perbedaan dengan YOLOv2 yaitu pada penggunaan ukuran deteksi dan arsitektur yang digunakan. Pada YOLOv3 menggunakan tiga jenis ukuran yang diambil dari proses ekstraksi fitur yaitu 13x13, 26x26, dan 52x52 dengan arsitektur Darknet-53. Pada arsitektur Darknet-53 menggunakan convolutional layer berjumlah 53 dengan menggunakan residual block berjumlah 5 yang menggantikan lapisan pooling. YOLOv3 tetap mempertahankan penggunaan algoritma K-Means pada bounding box yaitu pada anchor prior. Jika pada YOLOv2 bernilai K-5 maka pada YOLOv3 nilai dari K-3. Prediksi bounding box pada YOLOv3 menggunakan 4 koordinat pada setiap anchor boxes yaitu tx,y,w, th. Dalam melakukan prediksi skor objektivitas YOLOv3 menggunakan logistic regression untuk menghindari tumpang tindihnya label yang diberikan pada deteksi objek (Redmon & Farhadi, 2018).

2.5. YOLOv4

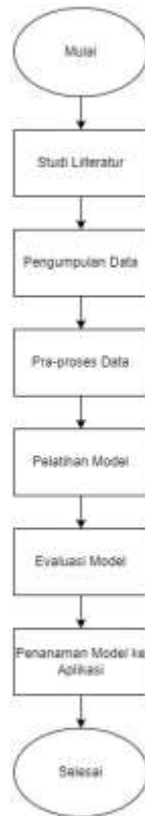
Pada penelitian ini metode yang digunakan adalah YOLOv4. YOLOv4 dikembangkan oleh peneliti yang berbeda dengan YOLO versi - versi sebelumnya yaitu dikembangkan oleh Alexey Bochkovskiy, Chien-Yao Wang, dan Hong-Yuan Mark Liao pada tahun 2020. Pengembangan YOLOv4 ditujukan untuk merancang kecepatan operasi dari objek detector yang lebih cepat dalam sistem produksi dan optimalisasi komputasi paralel, daripada indikator teoretis kompleksitas yang rendah (BFLOP). Penyempurnaan yang dilakukan pada YOLOv4 yaitu peningkatan mean average precision (mAP) sebesar 10% dan jumlah frame per second sebesar 12%. Arsitektur yang digunakan pada YOLOv4 memiliki 4 blok berbeda yaitu backbone, neck, dense prediction, dan sparse prediction. Backbone yang dipakai dalam YOLOv4 yaitu CSPDarknet53 (Cross Spatial Partial Darknet53) yang membagi lapisan menjadi dua bagian, satu untuk melewati convolutional layer dan yang lain tidak akan melewatinya. Neck memiliki fungsi membantu menambahkan lapisan diantara backbone dan dense prediction. YOLOv4 dalam arsitekturnya menggunakan path aggregation network, spatial attention module, dan spatial pyramid pooling yang kesemuanya telah dimodifikasi untuk meningkatkan akurasi. Head atau Dense Prediction berfungsi untuk menemukan bounding box untuk klasifikasi.

Dalam pengembangan YOLOV4 terdapat Bag of Freebies yaitu Kumpulan teknik atau metode yang mengubah strategi pelatihan atau biaya pelatihan untuk peningkatan akurasi model. Strategi yang digunakan pada Bag of Freebies yaitu menggunakan augmentasi data berupa cut mix dan mosaic, Semantic Distribution Bias pada dataset, dan regresi pada fungsi objektif pada bounding box. Bag of Specials merupakan metode tambahan di setiap detector objek dengan tujuan meningkatkan akurasi. Bag of specials memiliki 2 teknik, yang pertama untuk backbone yang menggunakan mish activation, cross stage partial connection, yang kedua untuk deteksi, yang menggunakan SPP-block, SAM blok, dan lain-lain. Dalam ujinya menggunakan dataset MS COCO YOLOv4 dengan algoritma lain memiliki keunggulan yang lebih pada YOLOv4 yang ditunjukkan pada gambar 2.18 (Bochkovskiy et al., 2020).

BAB III METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Dalam pelaksanaan penelitian ini, berikut merupakan metode yang akan digunakan untuk melaksanakan eksperimen penelitian, yaitu.



Gambar 3.1 Diagram Alur Tahapan Penelitian

Gambar 3.1 merupakan diagram alur tahapan penelitian yang akan dilaksanakan oleh penulis. Tahapan dimulai dari tahapan melakukan studi literatur kemudian melakukan pengumpulan data-data yang dibutuhkan yang akan diproses pada proses utama dalam penelitian yang dilakukan. Proses utama dalam penelitian ini dimulai dari tahapan pra-proses data, selanjutnya dilakukan perancangan dan pembuatan algoritma YOLOv4 untuk dilakukan pelatihan model dari hasil pra-proses data. Jika model telah dilakukan pelatihan, maka dilanjutkan melakukan evaluasi model untuk mengetahui model yang terbaik. Setelah diketahui model yang terbaik berdasarkan hasil evaluasi, maka dilanjutkan ke dalam tahap penanaman model ke dalam aplikasi.

3.2. Studi Literatur

Pada tahapan studi literatur, kegiatan yang dilakukan, yaitu mencari, membaca, dan mempelajari publikasi artikel ilmiah, makalah ilmiah, dan penelitian-penelitian sebelumnya dengan tema yang serupa dengan penelitian yang dilakukan oleh penulis yang tersedia secara terbuka di internet. Studi literatur dilakukan dengan tujuan mendapatkan dasar teori pendukung dalam melaksanakan sebuah penelitian seperti teori mengenai pengolahan citra, algoritma YOLOv4, hingga penanaman model ke dalam aplikasi.

3.3. Pengumpulan Data

Dataset yang digunakan dalam penelitian ini diambil dari situs *online* terbuka yang menyediakan beragam dataset, yaitu situs Kaggle dengan berbagai sumber akun yang memublikasikan ke dalam Kaggle. Dataset dilakukan pengunduhan dan dikumpulkan menjadi sebuah folder untuk disimpan dan diolah pada tahapan berikutnya. Dari dataset yang telah diunduh melalui Kaggle, perlu dilakukan tahapan pra-proses data untuk dilakukan penyesuaian dengan kebutuhan seperti dilakukan pengolahan terhadap dataset menyesuaikan dengan kebutuhan penelitian.

3.4. Pra-proses Data

Pada tahapan pra-proses data, akan dilakukan beberapa tahapan untuk mengolah dataset yang telah dikumpulkan. Pada tahapan ini terdiri dari dua bagian dalam melakukan pra-proses data. Langkah pertama, yaitu melakukan anotasi data sesuai dengan format YOLOv4. Dari hasil dataset yang telah dilakukan anotasi data dilanjutkan melakukan akuisisi data.

3.4.1. Anotasi Data

Dari dataset yang telah dikumpulkan sebelumnya, dilakukan proses anotasi pada dataset, yaitu melakukan pelabelan pada setiap gambar sesuai dengan format YOLOv4. Setiap gambar pada dataset dilakukan pelabelan menjadi dua kategori, yaitu memakai rompi keselamatan kerja dan tidak memakai rompi keselamatan kerja.

Pada tahapan anotasi data menggunakan aplikasi labelIMG untuk melakukan pelabelan. LabelIMG dibuat menggunakan bahasa pemrograman *python* yang dijalankan menggunakan terminal untuk membuka *interface* dan menjalankan aplikasi LabelIMG.

LabelIMG menyediakan beberapa jenis anotasi yang disimpan di antaranya CreateML, file XML dalam format PASCAL VOC, yaitu format yang digunakan oleh ImageNet. Selain itu, LabelIMG juga mendukung format YOLO yang akan digunakan pada penelitian ini. sebelum menjalankan aplikasi LabelIMG dibutuhkan pemasangan *library*, yaitu *python* sebagai *library* dasar dan Qt5 sebagai *library interface*, kemudian unduh kumpulan file dari aplikasi LabelIMG pada github milik tzutalin. Pada kumpulan file aplikasi Labelimg yang telah diunduh, dilakukan perubahan pendefinisian class atau kategori sesuai dengan kebutuhan penelitian yaitu pada folder data dengan nama file predefined classes yang di isi sesuai dengan kategori class dari penelitian.

Cara melakukan anotasi menggunakan Labelimg yaitu pertama, membuka terminal dan menjalankan file Labelimg.py. Setelah interface aplikasi terbuka, ubah format anotasi menjadi YOLO dan mengatur folder untuk menyimpan hasil anotasi. Langkah selanjutnya yaitu membuka folder dari dataset dan memuat seluruh gambar yang ada didalam folder tersebut. Setelah gambar yang akan di anotasi sudah ditampilkan maka dilakukan proses pemberian bounding box pada setiap bagian gambar yang di inginkan dan pilih class kategori yang sesuai kemudian simpan. Hasil anotasi yang disimpan berupa file text dengan format txt. Berikut merupakan format anotasi dari YOLOv4:

$$< \text{object} - \text{class} > < x > < y > < \text{width} > < \text{height} >$$

Objet class adalah nilai integer dari claus kategori objek yang bernilai mulai dari 0 hingga total dari claxs-1,x dan y merupakan titik tengah dari hounding box, width merupakan lebar dari bounding box dan height merupakan tinggi dari bounding box.

3.4.2. Akuisisi Data

Tahapan akuisisi data yang dilakukan, yaitu melakukan pembagian atau split dataset yang telah dianotasi menjadi data latih, data validasi dan data tes. Data latih adalah dataset yang digunakan dalam pelatihan model yang dibuat sedangkan data validasi adalah dataset yang digunakan untuk melakukan validasi pada model dan mengetahui performa dari model dan data tes adalah data yang digunakan untuk menguji hasil dari model yang telah dilakukan pelatihan dan validasi.

Dalam tahapan akuisisi data dibuatkan sebuah program menggunakan bahasa pemrograman python yang dapat melakukan pembagian dataset secara otomatis untuk mempersingkat dan mempermudah proses split data. Data yang telah dilakukan

pembagian disimpan berupa file text dengan format txt yang berisi dari nama gambar sesuai proses split

Pada penelitian ini dataset dibagi dengan rasio 80% dari dataset untuk data latih. 10% untuk data validasi dan 10% untuk data tes.

Tabel 3.1 Tabel Pembagian Dataset

Jenis Data	Rasio	Jumlah Data
Total Data		5000
Data Latih	80%	4000
Data Validasi	10%	500
Data Tes	10%	500

3.5. Pelatihan Model

Pelatihan model adalah tahapan yang dilakukan untuk melatih model menggunakan data latih yang telah disiapkan pada tahapan sebelumnya. Dalam pelatihan model algoritma yang digunakan adalah YOLOV4. Setelah dilakukan pra-proses data, maka dilanjutkan pada tahapan pelatihan model untuk digunakan dalam melakukan deteksi objek. Tahapan pelatihan model di antaranya:

3.5.1. Membangun Arsitektur YOLOv4

Sebelum melakukan tahapan pelatihan model maka diperlukan pembuatan arsitektur YOLOv4. YOLOV4 menggunakan koneksi CSP yang dibangun dengan Darknet-53 di sebagai backbone dalam ekstraksi fitur sesuai yang dijelaskan pada bab sebelumnya.

Dasar dari arsitektur YOLOv4 diperoleh melalui cloning atau unduhan dari pembuat algoritma YOLOV4 yang disimpan di folder yang telah ditentukan. Dari arsitektur dasar yang telah diunduh dilakukan improvisasi untuk menyesuaikan kebutuhan dan keterbatasan dari media yang digunakan dalam melakukan pelatihan model. Improvisasi yang dilakukan yaitu mengubah proses penyimpanan bobot saat proses pelatihan model.

Pada arsitektur dasar dari pembuat algoritma YOLOv4 bobot selalu disimpan adalah bobot dengan mAP paling tinggi dengan nama file yolov4- obj_best.weights, kemudian setiap 1000 epoch dengan nama file yolov4- obj <epoch weights dan akan menyimpan bobot dengan nama yolov4- obj last.weights disetiap 100 iterasi. Pada arsitektur dasar dilakukan improvisasi yaitu menyimpan setiap 100 iterasi dengan nama file yolov4-

obj_<epoch>.weights agar dapat digunakan melanjutkan pelatihan model jika proses pelatihan berhenti mendadak.

Tools yang digunakan dalam tahapan pelatihan model yaitu Google Colaboratory sebagai media untuk melakukan pengkodean dan menjalankan program yang dibuat dengan Google Drive sebagai media penyimpanannya. Improvisasi yang dilakukan sangat diperlukan karena dalam proses pelatihan model menggunakan GPU pada Google Colaboratory dan memanfaatkan koneksi internet yang tidak dapat dipastikan kestabilannya sehingga dapat terputus sewaktu-waktu dalam proses pelatihan model. Dengan adanya improvisasi tersebut mengurangi kehilangan bobot yang diinginkan ketika sudah mencapai batas penggunaan GPU, atau terputusnya koneksi internet sehingga dapat melanjutkan proses pelatihan model dilain waktu.

3.5.2. Konfigurasi Hyperparameter YOLOv4

Dalam metode pembelajaran mesin salah satunya pada Algoritma YOLOv4 terdapat nilai yang harus dilakukan penyesuaian diantaranya adalah parameter dan hyperparameter. Perbedaan antara parameter dan hyperparameter adalah pada parameter nilai yang telah ditetapkan akan berubah dengan sendirinya dalam proses pelatihan sedangkan nilai dari hyperparameter yang telah ditetapkan tidak akan terjadi perubahan selama proses pelatihan berjalan. Nilai yang dapat diatur pada hyperparameter YOLOV4 di antaranya adalah:

Tabel 3.2 Hyperparameter pada YOLOv4

Hyperparameter	Deskripsi
Batch	Variabel yang berisi total data latih yang akan di proses dalam pelatihan model setiap satu iterasi pelatihan.
Subdivision	Variabel yang mengatur jumlah mini batch di setiap batch. Mini batch adalah jumlah data yang akan diproses GPU dalam kurun satu waktu. Mini batch dihitung dengan rumus: mini batch=batch/subdivision.
Width dan height	Width dan height merupakan ukuran resolusi dari lapisan input yang akan dilalui gambar sehingga gambar berubah resolusi sesuai nilai width dan height yang ditetapkan. Width merupakan lebar

	sedangkan height merupakan tinggi dari ukuran resolusi.
Max_batches	Variabel yang mengatur jumlah maksimal iterasi pada proses pelatihan model.
Steps	Variabel yang mengatur iterasi pada saat learning rate mengecil sesuai dengan scale factor.
Scale	Variabel yang berisi nilai scale factor yang berfungsi sebagai pengecil learning rate.
Momentum	Variabel yang berfungsi membantu mengetahui Langkah berikutnya berdasarkan pengetahuan pada Langkah sebelumnya pada proses pelatihan model.
Decay	Variabel yang berisi nilai untuk menghilangkan ketidakseimbangan data pada dataset.
Learning rate	Variabel yang berfungsi mengatur kecepatan jaringan dalam memproses bobot pada saat pelatihan model.
Filters	Variable yang berisi nilai jumlah dari filter yang digunakan dan sesuai dengan jumlah dari class. Jumlah dari class kategori objek yang akan dilakukan pelatihan model.
Classes	Jumlah dari class kategori objek yang akan dilakukan pelatihan model

Hyperparameter pada algoritma YOLOv4 tersimpan pada file konfigurasi dengan ekstensi file (.cfg) pada folder cfg dalam arsitektur YOLOv4 yang sudah dibangun. Pada penelitian ini improvisasi yang dilakukan adalah melakukan tuning terhadap hyperparameter yang akan dibahas pada sub bab skenario peneliti. Tujuan dari hyperparameter adalah untuk menemukan nilai hyperparameter dari skenario penelitian yang memberikan hasil performa model terbaik.

3.5.3. Memuat Bobot Pre-trained YOLOv4 dan Dataset

Setelah arsitektur YOLOV4 telah selesai dibangun, proses berikutnya ialah melakukan *transfer learning* menggunakan bobot yang tela dilatih sebelumnya oleh pembuat algoritma YOLOv4 menggunakan dataset MS COCO. *Transfer learning* merupakan teknik yang menggunakan model sebelumnya yang sudah dilatih untuk digunakan sebagai titik mulai pada proses pelatihan yang baru.

Bobot pre-trained didapatkan dari proses unduh dari github pembuat algoritma YOLOv4 dan disimpan dalam sub folder dengan nama "data" di satu folder yang sama dengan arsitektur YOLOv4 yang telah dibuat sebelumnya pada Google Drive. Selain memuat Pre-trained YOLOv4, dataset juga diunggah ke dalam google drive berupa file archive berformat zip dan dilakukan ekstraksi ke dalam folder dengan nama "data" di satu folder yang sama dengan arsitektur YOLOv4.

3.5.4. Pelatihan Dataset pada Model YOLOv4

Setelah semua skenario dari hyperparameter sudah dikonfigurasi dan telah diterapkan di setiap model YOLOv4, Langkah selanjutnya yaitu melakukan pelatihan pada dataset. Proses pelatihan dilakukan menggunakan Google Colaboratory menggunakan akselerator GPU dari Google Colaboratory untuk mempersingkat waktu proses pelatihan dataset. Akan tetapi, penggunaan akselerator GPU dari Google Colaboratory memiliki batasan penggunaan dan harus menggunakan koneksi internet sehingga improvisasi yang dilakukan pada model YOLOv4 dapat mempermudah untuk melanjutkan proses pelatihan data di sesi berikutnya tanpa kehilangan iterasi bobot yang paling mendekati batasan penggunaan Google Colaboratory atau ketika koneksi internet terputus. Pada proses pelatihan model, bobot baru akan otomatis tersimpan yaitu setiap 100 iterasi, bobot terbaik, dan bobot terakhir. Pelatihan dataset dilakukan sesuai jumlah skenario penelitian. Dari hasil pelatihan dataset di setiap skenario pelatihan dipilih bobot dengan mAP tertinggi untuk dilakukan pengujian dengan data tes dan melakukan evaluasi model.

3.6. Evaluasi Model

Setelah pelatihan model selesai dilakukan dan telah mendapat bobot dari setiap skenario, tahapan selanjutnya yaitu melakukan evaluasi pada setiap model. Evaluasi yang dilakukan diantaranya adalah menggunakan Mean Average Precision (mAP), Intersection Over Union (IoU), Confusion Matrix, Precision, Recall, dan F1 Score.

3.7. Penanaman Model ke dalam Aplikasi

Setelah diketahui hasil dari evaluasi model terhadap semua skenario, penulis akan memilih satu bobot model yang memiliki performa paling baik di antara semua skenario untuk dilakukan penanaman model ke dalam aplikasi. Akan tetapi, dikarenakan dalam penelitian ini

fokus yang diteliti bukan pada pembuatan sistem. maka penulis tidak membahas proses perancangan dan pembuatan sistem aplikasi secara lebih detail.

Model yang telah ditanamkan ke dalam aplikasi dapat berguna untuk membantu pengawasan terhadap keselamatan di area kerja kritis yang mengharuskan menggunakan Alat Pelindung diri (APD) yaitu safety helmet. Gambaran umum dari cara kerja sistem aplikasi yaitu pengguna dapat menghubungkan komputer ke dalam kamera atau cctv sebelum menjalankan aplikasi. Aplikasi yang telah dibuka dan terhubung dengan kamera atau cctv maka akan langsung mendeteksi area yang ditangkap oleh kamera atau cctv, selanjutnya ditampilkan kedalam layar komputer dengan hasil deteksi yang telah dilakukan.

Aplikasi yang dirancang untuk penanaman model dibangun dengan library dari python yaitu menggunakan Flask. Untuk memuat model dari YOLOv4, penulis akan menggunakan fungsi yang telah ada pada bahasa pemrograman python.

3.8. Skenario Penelitian

Skenario penelitian yang dilakukan yaitu membandingkan hasil evaluasi model berdasarkan improvisasi tuning hyperparameter pada YOLOv4 untuk mengetahui model dengan tuning hyperparameter terbaik. Tuning hyperparameter yang dilakukan hanya pada dua variabel hyperparameter yaitu pada ukuran resolusi input (width dan height) dan learning rate dari setiap model. Pada variabel hyperparameter yang lain akan bernilai sama pada setiap model. Berikut adalah skenario penelitian yang dilakukan:

Tabel 3.3 Skenario Penelitian

Skenario Penelitian	Ukuran Resolusi Input	Learning Rate
1.	512 x 512	1
2.		0.1
3.		0.01
4.	416 x 416	1
5.		0.1
6.		0.01
7.	320 x 320	1
8.		0.1
9.		0.01

Dalam skenario penelitian ini, diberikan 9 skenario yang terbagi menjadi 3 macam ukuran resolusi input dan 3 macam *learning rate*. Seluruh skenario penelitian akan dilatih menggunakan bobot pre-trained dan dataset yang sama.