**LAPORAN TUGAS AKHIR**

**IMPLEMENTASI PENGOLAHAN CITRA DIGITAL UNTUK MENGHITUNG JUMLAH KENDARAAN MASUK PADA JALAN RAYA**

Diajukan guna melengkapi sebagian syarat dalam mencapai gelar Sarjana Strata Satu (S1)

****

Disusun Oleh:

Nama : Ivan Prasetyo Nugroho

N.I.M. : 41420120059

Pembimbing : Rachmat Muwardi, B.Sc., S.T., M.Sc.

**PROGRAM STUDI TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS MERCU BUANA**

**JAKARTA**

**2022**

**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

Semakin majunya perkembangan teknologi saat ini, menjadikan pengaplikasian teknologi memasuki berbagai ranah aspek kehidupan. Salah satunya ialah di bidang tata kelola kota, khususnya pada pengaturan lalu lintas di jalan raya perkotaan. Semakin banyaknya pemilik kendaraan bermotor, berefek kepada semakin padat pula kendaraan bermotor di jalan raya. Kepadatan kendaraan bermotor pada umumnya terletak pada persimpangan karena kendaraan bermotor harus bergantian melewati persimpangan secara bergantian dengan kendaraan bermotor dari arah yang lain, di mana pengaturannya dilakukan oleh lampu lalu lintas (*traffic light*).

Permasalahan yang timbul dari gambaran kasus pada paragraf di atas ialah, pada persimpangan akan muncul antrean panjang pada daerah persimpangan sehingga menimbulkan kemacetan. Sebab kemacetan yang terjadi selain dari banyaknya jumlah kendaraan bermotor yang lewat, ialah dari kurang efektifnya sistem pengaturan lalu lintas pada daerah persimpangan. Efek dari kurang efektifnya suatu sistem pengaturan lalu lintas ialah, terjadi kepadatan di salah satu atau lebih dari masing-masing sisi persimpangan.

Melihat permasalahan yang terjadi, dibutuhkan suatu sistem di mana suatu sistem tersebut dapat menghitung kendaraan bermotor yang akan menuju ke suatu persimpangan dengan memanfaatkan proses pengolahan citra digital dari kamera yang dipasang pada jalan raya. Diharapkan dengan adanya sistem tersebut, dapat digunakan untuk peningkatan efisiensi dan efektivitas sistem pengaturan lalu lintas yang sudah ada.

Sistem penghitung kendaraan bermotor menghasilkan data berupa angka jumlah kendaraan bermotor yang akan lewat di suatu persimpangan. Data tersebut kemudian digunakan sebagai masukan suatu sistem lanjutan untuk memperhitungkan durasi menyala lampu merah, kuning, dan hijau pada suatu sisi persimpangan jalan. Data terukur juga dapat digunakan sebagai masukan ke dinas pemerintahan terkait dalam mengatur dan mengontrol jumlah kendaraan yang akan melewati suatu area.

1. **Rumusan Masalah**

Pada pembuatan Tugas Akhir ini terdapat beberapa permasalahan yang timbul, yaitu sebagai berikut:

1. Bagaimana proses yang dilakukan untuk mendeteksi dan menghitung mobil dari suatu masukan citra digital?
2. Apakah data hitung dari sistem yang telah dibuat memiliki ketepatan dengan data sebenarnya?
3. Bagaimana proses yang dilakukan untuk menilai kinerja dari sistem yang telah dibuat?
4. **Tujuan dan Manfaat**

Tujuan dan manfaat dirumuskan untuk memberikan arah kegiatan pembuatan sistem sehingga dapat dijadikan ukuran tingkat keberhasilannya. Tujuan dan manfaat dirumuskan meliputi tujuan umum pembuatan Tugas Akhir. Tujuan dan manfaat yang dimaksudkan adalah sebagai berikut:

1. Sistem yang terimplementasi diharapkan dapat mendeteksi dan menghitung mobil dari masukan citra digital.
2. Sistem yang terimplementasi diharapkan dapat menghasilkan data hitung yang tepat atau mendekati dengan data sebenarnya.
3. Sistem yang terimplementasi diharapkan dapat memiliki kinerja yang baik dalam mendeteksi dan menghitung mobil.
4. **Batasan Masalah**

Pada pembuatan Tugas Akhir ini terdapat permasalahan yang lebih fokus, maka dilakukan suatu pembatasan masalah, diantaranya:

1. Sistem yang penulis implementasi merupakan untuk kepentingan edukasi sebagai pengaplikasian materi yang penulis telah peroleh saat kuliah.
2. Kendaraan yang dideteksi ialah mobil saja.
3. Sistem yang penulis implementasi menggunakan algoritma YOLO (*You Only Look Once*) untuk mendeteksi objek pada masukan citra digital.
4. Data citra digital pada pengujian sistem, menggunakan data *video* yang diambil oleh penulis memiliki resolusi 360p dan *frame rate* 30fps.
5. Pengujian kinerja sistem menggunakan metode *Confussion Matrix*.
6. **Metodologi Penelitian**

Dalam proses untuk menyelesaikan tugas akhir, diperlukan serangkaian proses agar pengerjaan tugas akhir ini dapat selesai dengan hasil yang sebaik mungkin. Berikut ini ialah proses-proses yang akan dilakukan untuk membangun sebuah sistem pendeteksi dan penghitung jumlah kendaraan pada jalan raya:

1. Mengambil *video* sebagai masukan sistem.
2. Memilih algoritma *object detection* yang akan dipakai pada sistem.
3. Membuat desain diagram alur/*flowchart* dan diagram blok sebagai acuan dalam pembuatan sistem.
4. Membuat implementasi kode untuk menghasilkan sistem pendeteksi dan penghitung mobil mengacu pada diagram alur yang telah dirancang.
5. Membuat implementasi kode untuk mengevaluasi atau menguji hasil dari sistem yang telah dibuat dengan menggunakan metode *confussion matrix* dimana dilakukan perhitungan nilai akurasi, *recall*, dan *precision*.
6. Melakukan proses pengujian apakah data terukur dari sistem sesuai dengan data aktual.
7. Membuat laporan perancangan dan pengujian yang dilampirkan pada laporan tugas akhir.
8. **Sistematika Penulisan**

Laporan tugas akhir ini terdiri dari 5 bab, antara lain :

1. Bab I Pendahuluan

Pada bab ini akan dijelaskan mengenai latar belakang masalah, tujuan, manfaat, batasan masalah, metodologi penelitian dan sistematika penulisan dari tugas akhir yang dikerjakan.

1. Bab II Landasan Teori

Bab ini terdiri dari dua sub-bab yang antara lain ialah tinjauan pustaka dan dasar teori. Pada tinjauan pustaka akan dipaparkan beberapa penelitian yang sudah pernah dilakukan dengan topik terkait tugas akhir yang penulis kerjakan. Dan pada sub-bab dasar teori, akan dipaparkan penjelasan tentang bahasa pemrograman, algoritma *object detection* yang digunakan, dan metode evaluasi yang digunakan untuk menguji sistem yang penulis kerjakan.

1. Bab III Metodologi Penelitian / Perancangan Alat dan Sistem

Pada bab ini akan dijelaskan dijabarkan blok diagram, implementasi kode, pengujian sistem dengan metode *confussion matrix* yang disertai dengan proses perhitungan akurasi, *recall*, dan *precision*.

1. Bab IV Hasil dan Pembahasan

Pada bab ini akan dipaparkan analisis data-data yang didapatkan dari hasil pengujian.

1. Bab V Penutup

Menjelaskan mengenai kesimpulan akhir penelitian dan saran-saran yang direkomendasikan untuk perbaikan proses pengujian dan pengerjaan tugas akhir selanjutnya.

**BAB II**

**LANDASAN TEORI**

* 1. **Tinjauan Pustaka**

Tinjauan pustaka berisi informasi tentang teori dan uraian teori yang menjadi dasar referensi TA (Tugas Akhir), berupa pembahasan hasil-hasil penelitian sebelumnya yang sesuai dengan tujuan yang telah ditetapkan. Tinjauan pustaka ini dapat membantu mahasiswa dalam menyusun kerangka berpikir ilmiah, merumuskan *hipotesis*, dan menetapkan metode penelitian, pengumpulan dan analisis data. Kajian pustaka dapat merujuk pada hasil penelitian seperti laporan tugas akhir dan jurnal.



Dalam Tugas Akhir tidak hanya mengumpulkan data, kemudian menulisya tanpa ada suatu landasan, tetapi dengan adanya Tugas Akhir mempunyai teori-teori yang cocok dengan rumusan masalah yang akan menjadi Tugas Akhir. Hal ini dilakukan supaya lebih mudah untuk menjelaskan adanya teori dalam Tugas Akhir.

1. Video

Video merupakan suatu media elektronik untuk menyampaikan suatu pesan melalui gambar yang bergerak. Video tersusun dari beberapa gambar yang ditampilkan secara bergantian. Kecepatan gambar yang ditampilkan pada video biasa disebut *frame rate* yang memiliki satuan *Frame Per Second* (FPS). FPS merupakan merupakan satuan yang menyatakan berapa banyak gambar yang tertampil pada video dalam 1 detik. Contohnya suatu video memiliki *frame rate* sebesar 30fps, maka video tersebut menampilkan gambar secara bergantian sebanyak 30 gambar dalam 1 detik. Semakin besar nilai FPS video, maka pergerakan objek pada video akan semakin halus. Di masa sekarang, nilai *frame rate* video memiliki beberapa nilai yang beragam, yaitu 24fps, 30fps, 60fps, 70fps, 120fps, dan masih banyak nilai yang lainnya.

1. Pengolahan Citra Digital (*Image Processing*)



Gambar 2.1. Ilustrasi *image processing*

Pengolahan citra digital merupakan suatu proses dalam mengolah dan memproses suatu sinyal masukan berupa citra digital yang ditransformasikan menjadi gambar berbeda sebagai keluaran. Pengolahan citra digital dilakukan dengan tujuan memperbaiki atau meningkatkan kualitas citra agar dapat lebih mudah diinterpretasi, dimanipulasi atau dianalisis oleh suatu sistem lanjutan. Berdasarkan tujuan transformasinya, operasi pengolahan citra digital dikelompokkan menjadi sebagai berikut:

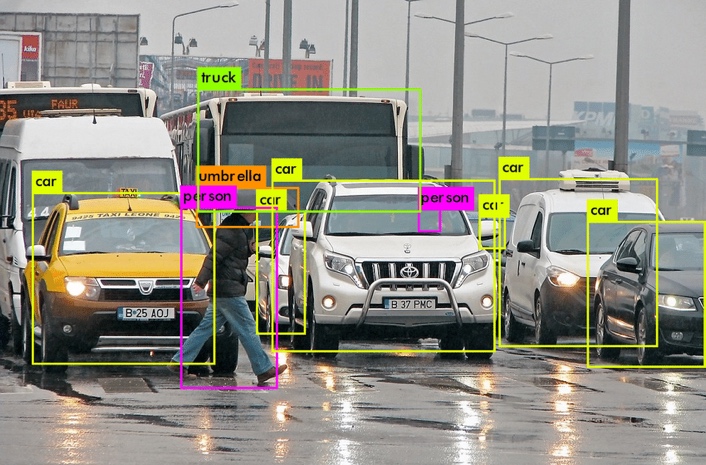
* Peningkatan kualitas citra (*image enhancement*)

Operasi peningkatan citra merupakan suatu operasi yang bertujuan untuk meningkatkan beberapa fitur tertentu di dalam suatu citra.

* Pemulihan citra (*image restoration*)

Operasi pemulihan citra merupakan suatu operasi untuk memperbaiki atau mengembalikan kondisi citra ke kondisi sebelumnya yang sudah diketahui yang diakibatkan oleh adanya gangguan yang menyebabkan beberapa fitur di dalam citra mengalami kerusakan atau perubahan yang tidak diinginkan.

1. Deteksi Objek

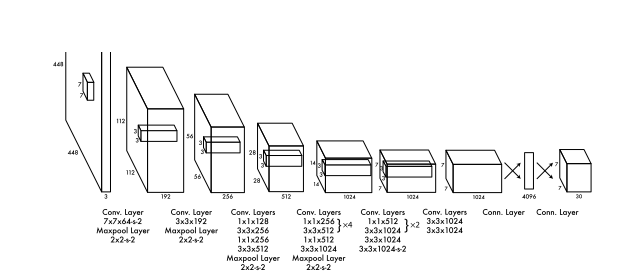


Gambar 2.2. Ilustrasi deteksi objek

Deteksi objek pada pengolahan citra digital merupakan penentuan keberadaan dan posisi suatu objek pada citra. Secara sederhana, deteksi objek ialah melakukan pemindaian pada keseluruhan citra dan menentukan kelas (keberadaan) suatu objek dan lokasi koordinat objek tersebut.

Alur pendeteksian objek ialah dimulai dari masukan, yaitu suatu citra. Citra masukan lalu dimasukkan ke sebuah model atau algoritma pada sistem yang telah dilatih untuk mendeteksi suatu objek. Setelah pemrosesan dari model atau algoritma telah selesai, akan menghasilkan koordinat yang merepresentasikan posisi suatu objek pada citra. Koordinat yang dihasilkan akan menghasilkan sebuah kotak pembatas (*bounding box*) di sekitar objek yang dideteksi. Terdapat beberapa model atau algoritma yang dapat digunakan untuk mendeteksi objek pada citra, yaitu di antaranya ialah *Faster R-CNN, YOLO, Mask R-CNN*, dan lain sebagainya.

1. *You Only Look Once* (YOLO)

**

Gambar 2.3. Arsitektur YOLO

*You Only Look Once* (YOLO) merupakan salah satu algoritma deteksi objek secara *real-time ­*yang berbasis *Convolutional Neural Network* (CNN). YOLO pertama kali dikembangkan oleh Redmon et al. pada tahun 2015. Dalam pendeteksian objek, YOLO hanya menggunakan satu lapisan jaringan syaraf atau biasa disebut dengan jaringan syaraf tunggal (*single neural network*). Jaringan tunggal pada YOLO tersebut akan membagi masukan citra menjadi beberapa *regions* dan memprediksi *bounding boxes* serta kemungkinan pada setiap *regions* secara bersamaan. YOLO memiliki kelebihan, yaitu YOLO dapat mendeteksi suatu objek dengan lebih cepat. Namun dengan kecepatan deteksinya itu pula muncul suatu kelemahan, yaitu YOLO dapat membuat lebih banyak melakukan kesalahan lokalisasi objek pada masukan citra, dan YOLO lemah dalam mendeteksi objek berukuran kecil pada citra. Kelemahan YOLO ini muncul dikarenakan pada tahap deteksi, YOLO tidak melakukan langkah proposal terlebih dahulu.

Algoritma YOLO membagi citra menjadi beberapa sel *grid* (petak kecil) yang memiliki ukuran s x s. Jika suatu bagian dari objek terdeteksi masuk ke dalam suatu sel *grid,* maka sel *grid* tersebut bertanggung jawab untuk melakukan deteksi pada objek tersebut. Tanggung jawab dari sel *grid* tersebut ialah memprediksi lokasi *bounding boxes* dan nilai *confidence score*, serta prediksi klasifikasi *class* dari objek terdeteksi. Nilai *confidence score* pada YOLO didefinisikan oleh rumus:

*Confidence Score =*

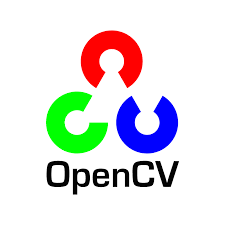
Apabila pada sel *grid* tidak terdapat objek terdeteksi, maka nilai *confidence score*-nya akan nol. Gabungan banyak dari sel *grid* ini akan membentuk *bounding boxes*. Setiap *bounding boxes* terdiri dari variabel x, y, w, h, dan *confidence score*. Variabel x dan y merupakan nilai titik koordinat pusat *bounding boxes*. Variabel w dan h merupakan nilai lebar dan tinggi *bounding boxes* relatif terhadap ukuran keseluruhan citra. Sedangkan variabel *confidence score*  merupakan nilai keyakinan bahwa di dalam suatu *bounding boxes* terdapat objek terdeteksi.

Nilai keyakinan suatu *class* terdeteksi di dalam *bounding boxes* disebut *class confidence score*. *Class confidence score* mengukur nilai keyakinan atau probabilitas terhadap hasil klasifikasi dan lokalisasi objek. *Class confidence score* untuk setiap *bounding boxes* dirumuskan sebagai berikut:

Keterangan:

* : probabilitas kondisional *class* i.
* : probabilitas *class* i.

1. OpenCV

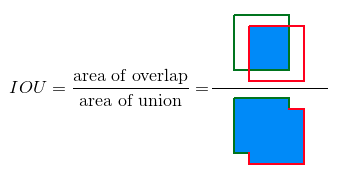


Gambar 2.4. Logo OpenCV

*Computer vision* ialahsuatu kecerdasan buatan yang memungkinkan untuk melakukan pengenalan objek pada suatu citra yang di dapat dari suatu sumber yang dapat berupa gambar atau video. OpenCV ialah salah satu *library* yang digunakan untuk digunakan untuk mengimplementasi pemrograman *computer vision* dengan didukung oleh fungsi-fungsi yang terdapat di *library* OpenCV.

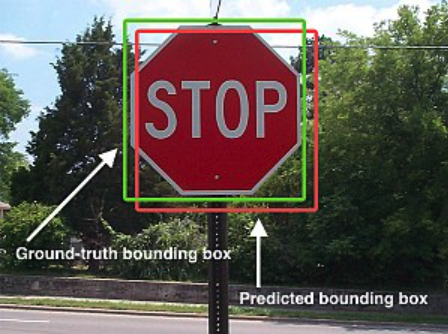
OpenCV didukung di beberapa bahasa pemrograman, diantaranya bahasa C, C++, Python, dan Java. OpenCV dapat dijalankan pada sistem operasi MacOS, Linux maupun Windows.

1. *Intersection over Union (IoU)*



Gambar 2.5. *Intersection of Union*

*Intersection over Union (IoU)* merupakan suatu parameter untuk mengevaluasi keakuratan deteksi objek pada suatu citra. IoU mengevaluasi irisan antara *bounding box*, yaitu antara *predicted bounding box* dan *ground truth bounding box*.



Ilustrasi IoU

1. Akurasi

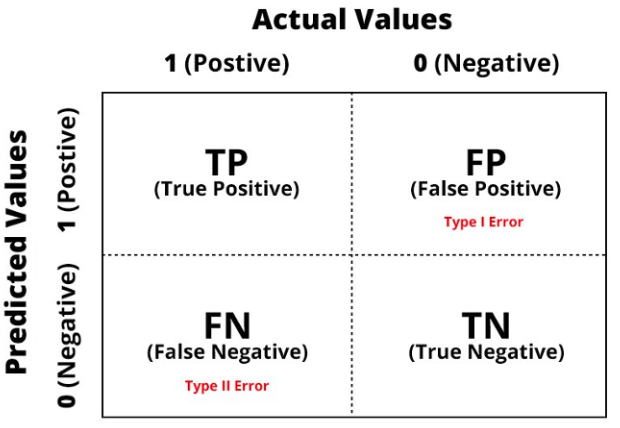
Pada pemodelan klasifikasi objek, akurasi merupakan *metrics* persentase model memprediksi objek dengan benar. Akurasi digunakan sebagai *metrics* untuk melihat performa model secara singkat. Berikut ini ialah rumus untuk menghitung nilai akurasi dari suatu model:

Akurasi =

Tetapi *metrics* akurasi memiliki kekurangan, yaitu ketika pengklasifikasian model harus mendeteksi *imbalanced data* dan *class* yang memiliki bobot yang sangat tinggi jika salah terprediksi. Contoh kasusnya ialah apabila terdapat 2 *class* dengan nama *class* A dan *class* B. Suatu model berhasil mendeteksi 99 *class* A dengan benar, namun model salah mendeteksi 1 *class* B yang dimana *class* B ini merupakan kasus yang sangat fatal apabila salah dideteksi.

1. *Confusion Matrix*

*Confusion Matrix* atau bisa disebut juga *error matrix* pada dasarnya memberikan informasi perbandingan hasil prediksi yang tereksekusi oleh model dengan hasil prediksi sebenarnya. *Confusion matrix* dapat digambarkan menjadi suatu tabel matriks yang menggambarkan kinerja dari suatu model terhadap data uji yang nilai sebenarnya telah diketahui.



Tabel matriks *confusion matrix*

Berikut ini ialah variabel-variabel terkait pada *confusion matrix*:

1. *Positive* (P): sampel positif.
2. *Negative* (N): sampel negatif.
3. *True Positive* (TP): sampel bernilai positif diprediksi benar.
4. *True Negative* (TN): sampel bernilai negatif diprediksi benar.
5. *False Positive* (FP): sampel bernilai positif diprediksi salah.
6. *False Negative* (FN): sampel bernilai negatif diprediksi salah.
7. *F1-Score*

*F1-Score* adalah *Harmonic Mean* antara nilai *precision* dan *recall*. Nilai rentang dari *F1-Score* sebesar [0, 1]. *F1-Score* bertujuan untuk mengetahui seberapa tepat pengklasifikasian pada suatu model. Berikut ini ialah rumus untuk menghitung nilai *F1-Score*:

*F1-Score* = 2 x

1. *Precision* dan *Recall*

Nilai *precision* ialah nilai perbandingan antara total prediksi benar pada sampel positif dengan total sampel positif yang diprediksi oleh model. Nilai *precision* menunjukkan seberapa baik model dalam memprediksi sampel positif dengan benar. Berikut ini ialah rumus dari *precision*:

*Precision =*

Nilai *recall* ialah nilai perbandingan antara total prediksi benar pada sampel positif dengan total smpel positif yang terdeteksi. Nilai *recall* menunjukkan seberapa baik model dalam memprediksi suatu *class* dengan benar. Berikut ini ialah rumus dari *recall*:

*Recall =*

Nilai antara *precision* dan *recall* saling berhubungan, berikut ini ialah hubungan antara kedua nilai tersebut terhadap penilaian dari performa suatu model:

1. *High Recall, Low Precision* ialah model memberi hasil prediksi yang kurang baik pada sampel positif, tetapi model dapat meminimalkan prediksi *false positive*.
2. *Low Recall, High Precision* ialah model memberi hasil prediksi yang baik pada sampel positif, tetapi akan banyak prediksi *false positive*.

**BAB III**

**METODOLOGI PENELITIAN /**

**PERANCANGAN ALAT DAN SISTEM**

1. **Gambaran Umum**

Pada bagian ini akan dijelaskan penjelasan sistem dan *time line* pembuatan sistem secara umum. Pada bab ini penulis akan menjelaskan tentang:

1. Gambaran umum sistem
2. Blok diagram sistem
3. Perancangan perangkat lunak
4. Proses evaluasi sistem

Secara umum, sistem dibuat menggunakan bahasa *phyton* sebagai bahasa pemrograman dan PyCharm sebagai *Integrated Development Environment* (IDE). Pada program yang dibuat, akan diimplementasi algoritma deteksi objek menggunakan *library* OpenCV dan YOLOv4. Lalu, hasil dari sistem yang dibuat akan dievaluasi menggunakan metode *confusion matrix.*

Pertama-tama video berupa video dengan resolusi 360p dengan *frame rate* 30fps akan digunakan sebagai masukan sistem. Sistem akan mendeteksi jumlah mobil dengan menggunakan algoritma YOLO dengan bantuan *library* OpenCV, dan akan menghitung jumlah mobil secara kontinyu selama video diputar. Apabila terdeteksi mobil pada *frame* video masukan, maka akan ditampilkan *bounding boxes* pada objek yang terdeteksi serta dilakukan penambahan jumlah mobil. Setelah proses dari sistem selesai, maka perlu dilakukan proses evaluasi yang mengukur kualitas dari sistem yang telah dibuat

dengan menggunakan metode *confusion matrix*. Setelah itu akan dilakukan perbandingan dengan cara menjalankan sistem pada *environment* yang berbeda, sebagai contoh dilakukan perbandingan apabila sistem dijalankan pada *environment* yang terpasang *Graphical Processing Unit* (GPU) dengan yang tidak terpasang GPU.

1. **Blok Diagram Sistem**
2. **Perancangan Perangkat Lunak**
3. **Proses Evaluasi Sistem**

**DAFTAR PUSTAKA**