

LAPORAN TUGAS AKHIR

Pengatur Kepadatan *Traffic Light* Berbasis *Image Recognition*



Disusun oleh:

Rezadi Ramadhan (20524143)

Rafi Mahdiansyah Falah (20524145)

Program Studi Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta

2024

LEMBAR PENGESAHAN

Pengatur Kepadatan *Traffic Light* Berbasis *Image Recognition*

Penyusun:

Rezadi Ramadhan (20524143)

Rafi Mahdiansyah Falah (20524145)

Yogyakarta, 09 Juli 2024

Dosen Pembimbing

A handwritten signature in blue ink, appearing to be 'H. Prasetyo W.', with a large loop on the left and a horizontal line across the middle.

Dr. Hasbi Nur Prasetyo W., S.T., M.T.
NIK 195240101

Program Studi Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta

2024

LEMBAR PENGESAHAN TUGAS AKHIR

Pengatur Kepadatan *Traffic Light* Berbasis *Image Recognition*



Susunan Dewan Penguji

Ketua Penguji

: Dr. Hasbi Nur Prasetyo W., S.T., M.T.

Anggota Penguji 1

: Elvira Sukma Wahyuni, S.Pd., M.Eng.

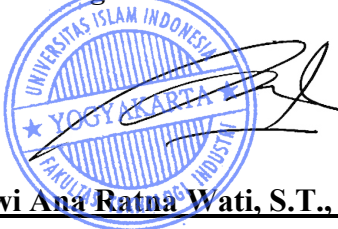
Anggota Penguji 2

: Dr. Sayuri Egaravanda, S.Kom, M.Eng

Tugas akhir ini telah disahkan sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Teknik

Tanggal: 06-Agustus-2024

Ketua Program Studi Teknik Elektro



Dwi Ana Ratna Wati, S.T., M.Eng.

035240102

PERNYATAAN

Dengan ini kami menyatakan bahwa:

1. Tugas Akhir ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi lainnya, dan sepanjang pengetahuan kami juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Tugas Akhir yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak, yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal ini, penggunaan informasi dan materi Tugas Akhir terkait paten maka akan didiskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut di atas.

Yogyakarta, 06-Agustus-2024



Rezadi Ramadhan (20524143)

Rafi Mahdiansyah Falah (20524145)

DAFTAR ISI

LEMBAR PENGESAHAN	2
LEMBAR PENGESAHAN TUGAS AKHIR	3
PERNYATAAN	4
DAFTAR ISI.....	5
RINGKASAN	8
BAB 1. PENDAHULUAN	9
1.1 Latar belakang dan Identifikasi Masalah.....	9
1.2 Rumusan Masalah	12
1.3 Tujuan.....	13
1.4 Batasan Masalah.....	14
1.5 Batasan Realistis Aspek Keteknikan.....	14
BAB 2. IDENTIFIKASI KEBUTUHAN SISTEM	16
2.1 Studi Literatur dan Observasi.....	16
2.2 Dasar Teori	21
2.2.1 <i>Traffic Light</i>	21
2.2.2 Pengenalan Citra atau <i>Image Recognition</i>	21
2.3 Analisis <i>Stakeholder</i>	22
2.4 Analisis Aspek yang Mempengaruhi Sistem.....	23
2.5 Spesifikasi Sistem.....	24
BAB 3. USULAN SOLUSI	26
3.1 Usulan Solusi 1	26
3.1.1 Desain Sistem 1	27
3.1.2 Rencana Anggaran Desain Sistem 1.....	33

3.1.3 Analisis Risiko Desain 1	33
3.1.4 Pengukuran Performa	34
3.2 Usulan Solusi 2.....	36
3.2.1 Desain Sistem 2	36
3.2.2 Rencana Anggaran Desain 2	42
3.2.3 Analisis Risiko Desain 2	42
3.2.4 Pengukuran Performa	43
3.3 Analisis dan Penentuan Usulan Solusi/Desain Terbaik	43
3.4 Gantt Chart	45
3.5 Realisasi Pelaksanaan Tugas Akhir 1	46
BAB 4. HASIL RANCANGAN DAN METODE PENGUKURAN	48
4.1 Hasil Rancangan Sistem	48
4.1.1 Rangkaian Elektronik	48
4.1.2 Gambar Tiga Dimensi (3D).....	49
4.1.3 Optimasi Model YOLO untuk <i>Image Recognition</i>	51
4.1.4 Prosedur <i>Counting</i> dan Mekanisme Sistem.....	56
4.1.5 Hasil Akhir Perancangan.....	63
4.2 Metode Pengukuran Kinerja Hasil Perancangan	66
BAB 5. HASIL PENGUKURAN DAN ANALISIS	68
5.1 Analisis Hasil	68
5.1.1 Hasil dan Analisis Pengujian Indikator	68
5.1.2 Pemenuhan Spesifikasi Sistem	83
5.1.3 Pengalaman Pengguna.....	86
5.1.4 Kesesuaian Perencanaan dalam Manajemen Tim dan Realisasinya	87
5.2 Dampak Implementasi Sistem.....	88

BAB 6. KESIMPULAN DAN SARAN	90
6.1. Kesimpulan	90
6.2. Saran	90
DAFTAR PUSTAKA	92
LAMPIRAN – LAMPIRAN.....	94

RINGKASAN

Laporan tugas akhir ini berjudul "Pengatur Kepadatan *Traffic Light* Berbasis *Image Recognition*" bertujuan untuk mengembangkan sistem pengaturan lalu lintas yang adaptif dan responsif menggunakan teknologi pengenalan gambar (*Image Recognition*). Sistem ini dirancang untuk mendeteksi kepadatan lalu lintas secara *real-time* dan mengoptimalkan durasi lampu lalu lintas. Permasalahan utama yang diidentifikasi adalah ketidakmampuan sistem pengaturan lalu lintas konvensional untuk merespons perubahan kondisi lalu lintas secara dinamis.

Pada proposal tugas akhir kami mengusulkan dua solusi utama, yang pertama sistem berbasis YOLO (*You Only Look Once*) yang menggunakan *Raspberry Pi* untuk mikrokontroler yang terintegrasi dengan kamera. Dan usulan solusi dua, sistem berbasis YOLO yang menggunakan mikrokontroler *Arduino Mega* untuk mengendalikan lampu lalu lintas berdasarkan data pengenalan gambar. Kedua solusi ini melibatkan pengembangan dan pengujian model YOLO dengan berbagai dataset untuk memastikan akurasi deteksi dan responsivitas sistem. Hasil pengujian menunjukkan bahwa sistem ini mampu mendeteksi kendaraan dengan akurasi tinggi dan mengoptimalkan durasi lampu lalu lintas secara dinamis.

Hasil tugas akhir menunjukkan bahwa datasets dengan pembagian 80% untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian memberikan hasil terbaik model dengan performa nilai mAP (*Mean Average Precision*) - 50 sebesar 0.973 (97.3%). Selain itu, sistem ini mampu mengklasifikasikan kendaraan dengan akurasi tinggi, menunjukkan performa yang konsisten dalam berbagai kondisi operasional. Pengujian sistem menunjukkan bahwa durasi lampu hijau yang dihitung oleh sistem sesuai dengan perhitungan manual, menunjukkan akurasi dan efektivitas sistem dalam mengelola lalu lintas di persimpangan jalan.

Secara keseluruhan, laporan tugas akhir ini berhasil mengembangkan dan menguji sistem pengaturan lalu lintas berbasis pengenalan gambar yang adaptif dan responsif, diharapkan memberikan kontribusi signifikan dalam meningkatkan efisiensi lalu lintas dan kualitas hidup masyarakat di perkotaan.

BAB 1. PENDAHULUAN

1.1 Latar belakang dan Identifikasi Masalah

Kepadatan lalu lintas adalah salah satu tantangan besar dalam mobilitas perkotaan di seluruh dunia. Fenomena ini sangat kompleks dan meresahkan, mempengaruhi efisiensi transportasi, lingkungan, dan kualitas hidup masyarakat. Saat ini, urbanisasi yang cepat dan pertumbuhan jumlah kendaraan menyebabkan peningkatan kepadatan lalu lintas di hampir semua kota besar di seluruh dunia[1].

Terdapat data yang signifikan mengenai jumlah kendaraan di DIY pada tahun 2022 yang mendukung urgensi penyelesaian permasalahan kepadatan lalu lintas. Berdasarkan sumber terpercaya dari Dinas Perhubungan (DISHUB) DIY jumlah kendaraan di DIY pada tahun 2022 mengalami peningkatan yang sangat mencolok. Berikut data yang terlihat pada Tabel 1.1.

Tabel 1.1 Jumlah Kendaraan yang terdaftar di DIY[2]

No	Jenis Kendaraan	Tahun		
		2019	2020	2021
1	Mobil Penumpang	372,436	385,269	400,627
2	Bus	13,081	13,351	13,398
3	Mobil Beban (Truk)	76,271	79,153	82,475
4	Sepeda Motor	2,918,976	3,020,175	3,125,720
Jumlah		3,380,764	3,497,948	3,622,220

Menurut data tersebut, pada tahun 2019 hingga 2022, terdapat peningkatan jumlah kendaraan di DIY. Angka ini mencakup berbagai jenis kendaraan, termasuk mobil, sepeda motor, bus, dan truk. Dalam perincian lebih lanjut, jumlah kendaraan terbesar terjadi pada tahun 2021. Peningkatan signifikan dalam jumlah kendaraan ini mencerminkan dampak dari urbanisasi yang terus berlanjut, pertumbuhan ekonomi, dan perkembangan perkotaan yang pesat. Kepadatan lalu lintas yang disebabkan oleh peningkatan jumlah kendaraan ini telah menjadi masalah seperti kemacetan yang rutin terjadi tidak hanya mengganggu produktivitas, tetapi juga menimbulkan dampak negatif pada kualitas udara, lingkungan, dan kesehatan masyarakat.

Dalam bidang transportasi perkotaan, pengaturan lalu lintas merupakan salah satu aspek penting yang memiliki dampak signifikan terhadap kehidupan sehari-hari masyarakat. Kemacetan lalu lintas, kecelakaan jalan raya, dan peningkatan emisi gas buang adalah masalah umum yang dihadapi oleh banyak kota besar di seluruh dunia. Masalah ini menciptakan beban ekonomi dan lingkungan yang besar. Secara singkat, lampu lalu lintas diatur berdasarkan waktu tetap atau sensor fisik seperti *loop* induksi yang terpasang di permukaan jalan. Namun, pendekatan ini memiliki keterbatasan dalam mengatasi perubahan dinamis dalam lalu lintas, mengakibatkan ketidakefisienan dalam penggunaan infrastruktur jalan dalam penanggulangan risiko kecelakaan.

Dalam data dan statistik yang terkumpul, tergambar jelas bahwa masalah lalu lintas yang tidak efisien merupakan tantangan serius di banyak negara. Misalnya, analisis kepadatan lalu lintas selama berbagai jam dalam sehari menunjukkan fluktuasi yang signifikan, dengan lalu lintas pada jam sibuk pagi dan sore yang menjadi puncak kemacetan. Selain itu, data mengenai jumlah kecelakaan lalu lintas, cedera, dan kematian yang terjadi di persimpangan lalu lintas mengungkapkan bahwa banyak kecelakaan disebabkan oleh kurangnya responsivitas dari sistem pengaturan lalu lintas yang ada. Dampak buruk juga terlihat pada lingkungan, di mana lalu lintas yang terhenti dan kemacetan berkontribusi pada peningkatan emisi gas buang dari kendaraan, merusak kualitas udara dan berkontribusi pada perubahan iklim.

Namun, ketika kita merinci permasalahan ini lebih lanjut, kita menemukan bahwa salah satu hambatan utama adalah ketidakmampuan sistem pengaturan lalu lintas untuk merespons secara adaptif terhadap perubahan kondisi lalu lintas secara *real-time*. Kepadatan lalu lintas yang tiba-tiba meningkat akibat peristiwa seperti kecelakaan, acara khusus, atau cuaca buruk seringkali tidak diantisipasi oleh sistem pengaturan lalu lintas konvensional. Ini mengakibatkan kemacetan yang lebih parah dan waktu tunggu yang tidak proporsional bagi pengemudi di persimpangan yang ramai. Selain itu, pengaturan lalu lintas yang tidak optimal dapat mengancam keselamatan jalan raya dengan meningkatkan risiko kecelakaan. Oleh karena itu, penyelesaian masalah ini bukan hanya menjadi kebutuhan ekonomi, lingkungan, dan keselamatan, tetapi juga merupakan kontribusi berharga dalam dunia ilmu pengetahuan dengan menggabungkan teknologi pengenalan gambar berbasis kecerdasan buatan untuk meningkatkan mobilitas dan kualitas hidup di perkotaan[3].

Berikut beberapa pertanyaan yang disiapkan dan respon dari mitra (Dinas Perhubungan DIY) yang ditunjukkan pada Tabel 1.2 berikut.

Tabel 1.2 Hasil survei antara pengembang dan pengguna

Pertanyaan	Jawaban/tanggapan
Berapa lama countdown pada <i>Traffic Light</i> di tiap-tiap jalan? Bagaimana mekanisme pemberian waktu tertentu pada <i>Traffic Light</i>, apakah sama atau berbeda dan dasar penetapannya berdasarkan apa?	Pengaturan waktu pada <i>Traffic Light</i> biasanya diatur oleh otoritas lalu lintas setempat atau badan transportasi, dan dapat berbeda-beda antar wilayah. Waktu countdown tiap <i>Traffic Light</i> ditentukan berdasarkan beberapa parameter yang melibatkan analisis lalu lintas, kebutuhan pejalan kaki, dan peraturan keamanan jalan. Tetapi menurut NACTO Durasi siklus ideal untuk wilayah perkotaan adalah 60–90 detik.
Jenis kendaraan apa saja yang bisa dilihat melalui <i>cctv Traffic Light</i>?	Kendaraan yang terdeteksi sepeda, motor, mobil, truk, dan bus.
Daerah mana saja yang sering mengalami kepadatan? Dan jam berapa? Apakah <i>Traffic Light</i> di daerah perlu perlakuan khusus? Misal lampu merah berapa menit?, lampu hijau berapa menit?	Untuk daerah yang sering mengalami kemacetan itu kondisional tergantung waktu-waktu tertentu tetapi kemacetan sering terjadi di persimpangan dan lampu lalu lintas
Penyebab dari kepadatan <i>Traffic Light</i>?	Terjadi karena adanya penumpukan kendaraan yang berlebih dan kurangnya aparat yang dapat mengontrol secara langsung.
Jika terdapat kendaraan khusus, misal ambulans, mobil patroli apakah terdapat pengaturan tertentu untuk <i>Traffic Light</i> ?	Dalam keadaan tertentu, dengan mempertimbangkan kelancaran dan keselamatan lalu lintas, fungsi lampu tiga warna dapat diganti dengan lampu warna kuning yang menyala berkedip.

Hal-hal apa saja yang mempengaruhi kinerja/performa dari sistem <i>Traffic Light</i> yang sudah ada?	Faktor cuaca, faktor lingkungan, volume lalu lintas, perawatan dan pemeliharaan rutin, serta durasi masa pakai dari <i>Traffic Light</i> .
Apakah ada standar tertentu yang ditetapkan untuk <i>Traffic Light</i> ?	beberapa contoh standar yang umum digunakan: Ada standar yang paling dikenal yaitu <i>Manual on Uniform Traffic Control Devices</i> (MUTCD) di Amerika Serikat, di Uni Eropa, standar EN sering digunakan untuk <i>Traffic Light</i> seperti EN 12368, dan ISO 16508: Standar internasional ini menetapkan persyaratan teknis untuk lampu lalu lintas dan sistem kontrol lalu lintas.
Apakah sudah ada sistem pengaturan kepadatan kendaraan di perempatan, pertigaan secara otomatis? Jika sudah ada seperti apa sistem dan lokasinya dimana?	Untuk saat ini belum ada di daerah DIY.
Saran apa yang dapat diberikan oleh dishub terkait dengan sistem yang akan kita buat?	Kami sangat senang jika kalau adik-adik melakukan penelitian di Dinas Perhubungan DIY apalagi penelitian adik-adik bisa dipraktekkan langsung dilapangan kami pihak Dinas Perhubungan DIY akan memberikan support untuk penelitian tersebut. Pihak Dinas Perhubungan DIY ingin bekerja sama dengan akademisi untuk meningkatkan kenyamanan dan keamanan berkendara.

1.2 Rumusan Masalah

Rumusan masalah ini menjelaskan tantangan utama yang akan dipecahkan dalam tugas akhir *Capstone*. Dalam konteks ini, permasalahan utama adalah bagaimana kita dapat mengaplikasikan

teknologi pengenalan gambar yang berbasis kecerdasan buatan untuk menciptakan sistem pengaturan lalu lintas yang lebih adaptif, responsif, dan efisien. Sistem ini harus dapat mengidentifikasi secara akurat kondisi lalu lintas dalam waktu nyata, termasuk tingkat kepadatan, perilaku pengemudi, dan peristiwa lalu lintas yang tidak terduga.

- Bagaimana teknologi pengenalan gambar (*Image Recognition*) dapat digunakan untuk mendeteksi kepadatan dengan berbagai jenis kendaraan ?
- Bagaimana mengembangkan algoritma pengkondisian *counter* yang efektif untuk mengadaptasi pengaturan lampu lalu lintas berdasarkan analisis *real-time* dari data pengenalan gambar, sehingga dapat mengurangi kepadatan lalu lintas di titik perempatan tertentu secara otomatis?
- Bagaimana cara mengintegrasikan secara efektif *software* pengenalan gambar dengan *hardware* sistem *Traffic Light* untuk memastikan interaksi dan kinerja yang lancar?
- Bagaimana validasi dan evaluasi kinerja sistem ini dalam konteks operasional nyata untuk memastikan akurasi dan efektivitas dalam mengatur lalu lintas?

1.3 Tujuan

Capstone ini bertujuan untuk menggali kemungkinan dan potensi aplikasi teknologi pengenalan gambar dalam pengelolaan lalu lintas. Secara khusus, *Capstone* ini berfokus pada tujuan-tujuan sebagai berikut:

- Memahami bagaimana teknologi pengenalan gambar (*Image Recognition*) dapat digunakan efisien dan akurat dalam mendeteksi keramaian atau kepadatan kendaraan dari berbagai jenis dengan cepat.
- Menyusun dan menguji algoritma pengkondisian *counter* yang efektif untuk secara otomatis mengadaptasi pengaturan lampu lalu lintas berdasarkan kepadatan kendaraan di titik tertentu. Dan, menganalisis dampak pengkondisian *counter* terhadap kepadatan lalu lintas dan waktu tunggu di titik perempatan.
- Mengembangkan dan menguji integrasi antara *software* pengenalan gambar dengan *hardware* sistem *Traffic Light*.
- Mengukur kinerja sistem dalam mendeteksi kepadatan di *Traffic Light*, serta dampak *counter* terhadap kepadatan lalu lintas.

Diharapkan solusi yang ditemukan melalui *Capstone* ini diharapkan dapat membantu mengurangi kemacetan lalu lintas, meningkatkan mobilitas penduduk, mengurangi emisi gas buang, dan pada akhirnya, meningkatkan kualitas hidup masyarakat.

1.4 Batasan Masalah

Dalam merancang dan mengembangkan sistem pengaturan lalu lintas yang berbasis pada pengenalan gambar, kami memahami pentingnya menetapkan batasan masalah yang jelas dan realistis. Batasan-batasan ini bertujuan untuk menggarisbawahi parameter dan kondisi di mana sistem kami diharapkan beroperasi secara optimal.

- Sistem dikembangkan untuk mendeteksi dan mengklasifikasikan kendaraan dalam kategori yang sudah ditentukan yaitu: sepeda motor, mobil, truk, dan bus.
- Sistem hanya dapat bekerja pada klasifikasi jenis kendaraan yang sudah ditentukan dan mungkin tidak dapat mengidentifikasi jenis kendaraan lain atau objek lain di jalan.
- Sistem dirancang untuk dipasang dan dioperasikan pada lokasi-lokasi tertentu, seperti pertigaan atau perempatan jalan.
- Keterbatasan dalam akurasi dan keandalan algoritma pengenalan gambar, khususnya kinerja sistem mungkin akan terpengaruh dalam kondisi tertentu seperti hujan dan kabut.

1.5 Batasan Realistis Aspek Keteknikan

Dalam mengembangkan sistem pengaturan lalu lintas berbasis pengenalan gambar, memahami dan menetapkan batasan realistis aspek keteknikan menjadi kunci untuk mencapai solusi yang efektif dan dapat diandalkan. Bagian ini membahas batasan-batasan penting seperti akurasi deteksi, ketepatan perhitungan, proteksi sistem, dan cadangan daya. Dengan menetapkan batasan ini, kami berusaha untuk menciptakan sistem yang tidak hanya canggih secara teknologi tetapi juga praktis dan tangguh dalam aplikasi sehari-hari.

- Akurasi Deteksi: *Sistem Image Recognition* yang diusulkan harus mampu mengidentifikasi kendaraan dengan akurasi yang tinggi untuk menjamin efektivitas operasi sistem lampu lalu lintas. *Capstone* ini akan berfokus pada pengembangan algoritma yang dapat meningkatkan akurasi deteksi kendaraan dalam berbagai kondisi pencahayaan dan cuaca. Setiap sistem harus mampu mendeteksi objek dengan akurasi yang tinggi.
- Ketepatan Perhitungan: Sistem yang dikembangkan harus memiliki algoritma dan perangkat keras yang mampu melakukan perhitungan dengan ketepatan yang tinggi dan

cepat. Hal ini penting agar ketepatan dalam menghitung jumlah kendaraan dan menentukan durasi lampu lalu lintas berdasarkan analisis trafik adalah krusial. Sistem harus dirancang dengan algoritma yang dapat menghitung dan mengalokasikan waktu lampu hijau secara tepat berdasarkan kepadatan trafik yang terdeteksi.

- Proteksi Sistem: Pentingnya proteksi sistem tidak dapat diabaikan. Setiap sistem harus memiliki mekanisme keamanan yang memadai untuk melindungi dari serangan *eksternal*, kerusakan data, atau gangguan lainnya. Untuk menjamin keandalan dan keamanan sistem, perlindungan terhadap gangguan *eksternal* seperti cuaca buruk dan vandalisme. Desain sistem akan mencakup mekanisme proteksi housing yang tahan cuaca.
- Cadangan Daya: Mengingat pentingnya fungsi *Traffic Light* dalam mengatur arus lalu lintas, sistem harus memiliki sumber cadangan daya. Hal ini untuk memastikan bahwa sistem tetap berfungsi dengan baik meskipun terjadi gangguan pada sumber daya utama.

BAB 2. IDENTIFIKASI KEBUTUHAN SISTEM

2.1 Studi Literatur dan Observasi

Dalam mengidentifikasi kebutuhan sistem, langkah pertama yang dilakukan adalah melakukan studi literatur dan observasi terhadap solusi sejenis yang telah ada. Studi literatur ini bertujuan untuk memahami berbagai pendekatan dan teknologi yang telah diterapkan dalam pengaturan durasi traffic light berbasis image recognition. Berikut ini, Tabel 2.1 menyajikan ringkasan hasil dari studi literatur tersebut, yang menggambarkan berbagai pendekatan dan teknologi yang telah diimplementasikan dalam sistem serupa.

Tabel 2.1 Hasil Studi Literatur Solusi Sejenis

Judul	Usulan solusi	Hasil/Evaluasi (Kelebihan/Kekurangan)
<i>Smart Traffic Light Menggunakan Image Processing Dan Metode Fuzzy logic</i> [4]	Penelitian ini membahas tentang penggunaan sensor kamera untuk mengumpulkan data tentang kondisi lalu lintas yang kemudian diolah menggunakan teknik pengolahan citra (image processing). Tujuannya adalah untuk mendeteksi jumlah kendaraan melalui proses ekstraksi kontur yang akan menjadi input untuk sistem penghitungan waktu lampu lalu lintas menggunakan algoritma <i>fuzzy logic</i> .	Hasil: Metode image processing menggunakan <i>webcam</i> pada purwarupa lalu lintas berhasil melakukan pengambilan gambar kendaraan dan mengidentifikasi jumlah kendaraan secara otomatis. Waktu eksekusi yang dibutuhkan untuk proses ini adalah 2,90 detik. Sistem ini mampu menyimpan hasil pengambilan gambar dan menggunakan data tersebut dalam perhitungan <i>fuzzy logic</i> . Kelebihan: Biaya implementasi metode image processing relatif murah, dan setup program untuk image processing mudah dilakukan.

		Akurasi dan kecepatan pengolahan data relatif baik. Kekurangan: Jika kendaraan terposisi terlalu dekat satu sama lain, pendeteksian menjadi kurang tepat karena kesulitan dalam menentukan <i>threshold</i> kontur objek yang dianggap sebagai kendaraan. Setiap objek memiliki nilai kontur yang berbeda-beda, yang dapat menyulitkan proses identifikasi. Dan jarak <i>webcam</i> dari jalur lalu lintas mempengaruhi kualitas pengambilan citra; jika terlalu dekat, kontur kendaraan bisa berhimpit dan sulit untuk dikenali.
<i>Dynamic time distribution system monitoring on Traffic Light using image processing and convolutional neural network method</i> [5]	Penelitian ini mengusulkan sistem untuk menghitung kepadatan jalur lalu lintas dan mendeteksi pelanggaran marka penyeberangan menggunakan sensor kamera yang juga dapat mengontrol waktu pada lampu lalu lintas. Dalam studi ini, metode yang digunakan berbeda dari penelitian sebelumnya, di mana studi ini menggunakan metode Jaringan Saraf Konvolusional (<i>Convolutional Neural Network</i>, CNN). Metode	Hasil: Kesimpulan dari studi ini adalah bahwa metode Jaringan Saraf Konvolusional (<i>Convolutional Neural Network</i>) dapat memantau tingkat kepadatan dengan baik. Dataset yang ada telah memiliki tingkat akurasi yang baik, mencapai 100%. Tingkat akurasi ini dapat dilihat dari hasil data latih yang telah dilakukan menggunakan optimizer Adadelta. Kelebihan: Tingkat akurasi yang tinggi (100%) dalam mendeteksi

	CNN ini mampu mendeteksi objek berdasarkan datasheet yang dimasukkan. Pengolahan Citra (Image Processing) digunakan untuk menentukan jumlah kendaraan di jalur kiri.	dan memantau tingkat kepadatan lalu lintas. Kekurangan: Kekurangan dari sistem ini tidak disebutkan secara eksplisit dalam bagian yang diekstrak, tetapi bisa jadi terkait dengan keterbatasan umum sistem berbasis CNN seperti kebutuhan data pelatihan yang besar, atau mungkin ketergantungan pada kualitas dan posisi sensor kamera yang digunakan.
<i>Design Prototype of Traffic Light Control System using HSV Color Segmentation</i> [6]	Solusi yang ditawarkan peneliti berupa Sistem prototipe yang menggunakan segmentasi warna HSV (<i>Hue, Saturation, Value</i>) untuk mengatur lampu lalu lintas berdasarkan kepadatan kendaraan. Menggunakan kamera USB dan <i>Raspberry Pi 3</i>, sistem ini mengolah video secara <i>real-time</i> untuk memperkirakan jumlah kendaraan di jalan, dengan eksperimen menunjukkan efektivitas algoritma dalam kondisi cahaya tertentu. Algoritma warna HSV untuk deteksi warna jamak digunakan untuk memperkirakan kepadatan lalu lintas dan	Hasil: Penelitian ini menghasilkan sebuah prototipe sistem pengendalian lampu lalu lintas yang menggunakan segmentasi warna HSV untuk mendeteksi dan mengklasifikasikan kondisi kepadatan lalu lintas. Prototipe yang dikembangkan terhubung dengan berbagai periferal termasuk kamera USB, monitor LCD, motor servo, dan lampu lalu lintas LED untuk setiap jalur. Algoritma deteksi warna HSV bekerja dengan mendefinisikan batas atas dan bawah dari nilai HSV untuk menyaring warna tertentu. Studi ini menunjukkan bahwa segmentasi warna HSV mampu mendeteksi

	mengklasifikasikan kondisi lalu lintas berdasarkan jumlah kendaraan di jalan.	warna multipel dan memperkirakan kepadatan lalu lintas serta mengklasifikasikan kondisi lalu lintas berdasarkan jumlah kendaraan di jalan. Kelebihan: Kemampuan untuk menyaring warna target dari frame video, yang penting untuk presisi deteksi. Dan, Kemampuan untuk memperkirakan kepadatan lalu lintas dan mengklasifikasikan kondisi lalu lintas. Kekurangan: Kekurangan sistem ini tidak dijelaskan secara spesifik. Namun, kekurangan umum dari sistem berbasis segmentasi warna HSV mungkin termasuk sensitivitas terhadap perubahan kondisi pencahayaan yang dapat mempengaruhi deteksi warna, atau keterbatasan dalam mendeteksi kendaraan yang warnanya tidak termasuk dalam rentang yang telah ditentukan.
<i>Traffic Light Detection and Recognition Method Based on YOLOv5s and AlexNet [7]</i>	Penelitian ini mengusulkan metode deteksi dan pengenalan lampu lalu lintas yang berdasarkan algoritma deteksi target <i>YOLOv5s</i> dan klasifikasi gambar <i>AlexNet</i>. Metode ini	Hasil: Penelitian ini menghasilkan metode deteksi lampu lalu lintas yang memadukan <i>YOLOv5s</i> untuk deteksi dan <i>AlexNet</i> untuk pengenalan, meningkatkan akurasi deteksi, terutama dalam cahaya

	memulai dengan mendeteksi area lampu lalu lintas menggunakan <i>YOLOv5s</i>, kemudian mengekstraksi area tersebut dan melakukan operasi pemrosesan gambar, dan akhirnya menyajikan gambar yang diproses ke <i>AlexNet</i> untuk pengenalan dan penilaian. Pendekatan ini bertujuan untuk mengatasi kekurangan algoritma deteksi target tunggal, terutama dalam hal tingkat pengenalan yang rendah untuk deteksi target kecil. Selain itu, karena dataset buatan sendiri mengandung lebih banyak gambar dengan pencahayaan rendah, dataset dioptimalkan menggunakan algoritma peningkatan cahaya rendah <i>ZeroDCE</i>, yang meningkatkan performa model jaringan yang dilatih setelah optimisasi dataset, mencapai 99.46% AP (<i>average precision</i>).	rendah, dengan validasi akurasi lebih dari 90% untuk sinyal spesifik. Dan dapat menunjukkan bahwa metode yang diusulkan mampu meningkatkan akurasi dalam deteksi dan pengenalan lampu lalu lintas, khususnya dalam situasi dengan pencahayaan rendah. Kelebihan: Penelitian ini memperkenalkan fitur kontekstual dalam pengenalan lampu lalu lintas dan menggabungkan kelebihan algoritma jaringan syaraf konvolusional (CNN) untuk mengusulkan algoritma pengenalan dengan fitur yang progresif, memanfaatkan kekuatan CNN dalam mengatasi kompleksitas pengenalan lampu lalu lintas. Kekurangan: Dataset yang digunakan untuk pelatihan dan pengujian model terlalu kecil dan homogen, yang membatasi variasi dan kompleksitas situasi lalu lintas yang dapat dikenali oleh model. Dan struktur keseluruhan dari model jaringan perlu ditingkatkan untuk mengoptimalkan kecepatan deteksi dan pengenalan, yang penting untuk
--	---	---

		aplikasi deteksi lampu lalu lintas secara <i>real-time</i> .
--	--	--

Dalam pengembangan sistem *Traffic Light* Berbasis *Image Recognition*, kami mengedepankan tiga keunikan utama. Pertama, inovasi pada algoritma *Image Recognition* menjadi kekuatan inti sistem kami. Kedua, penempatan CCTV yang kami lakukan berbeda dari yang lazim. Kami memfokuskan pada strategi penempatan yang strategis, di mana lokasi dan sudut pandang kamera dirancang khusus untuk menangkap data lalu lintas yang paling relevan, meningkatkan efektivitas sistem dalam mengumpulkan informasi penting. Ketiga, kami menekankan pada kecepatan dan akurasi dalam pengambilan keputusan. Sistem kami dirancang untuk memproses data citra secara *real-time*, memungkinkan pengenalan pola lalu lintas dengan cepat dan akurat, sehingga dapat mengambil keputusan yang tepat dalam waktu singkat.

2.2 Dasar Teori

2.2.1 *Traffic Light*

Traffic Light, atau lampu lalu lintas merupakan infrastruktur penting yang menggunakan lampu dengan berbagai warna untuk mengatur arus lalu lintas di persimpangan dan perlintasan jalan. Sistem lampu lalu lintas dirancang untuk mengoptimalkan arus lalu lintas, mencegah kemacetan. Pengaturan waktu lampu-lampu tersebut disesuaikan dengan kebutuhan lalu lintas pada persimpangan tertentu. Untuk itu, sangat penting merancang suatu sistem lalu lintas yang cerdas yang dapat mengatur laju lalu lintas untuk menghindari hal-hal yang tidak diinginkan.[8] Kemacetan dapat diatasi dengan sebuah manajemen lalu lintas yang baik di setiap persimpangan, yaitu memaksimalkan kapasitas kendaraan dan meminimalkan waktu tunggu pengendara.

2.2.2 Pengenalan Citra atau *Image Recognition*

Pengenalan citra atau *Image Recognition* adalah proses mengidentifikasi suatu objek atau fitur dalam sebuah gambar atau video. Teknologi ini digunakan dalam berbagai aplikasi seperti deteksi barang, pencitraan medis, dan pengawasan keamanan.[9] Akuisisi citra adalah tahap awal untuk mendapatkan citra digital. Tujuan akuisisi citra adalah untuk menentukan data yang diperlukan dan memilih metode perekaman citra digital. Tahap ini dimulai dari objek yang akan diambil gambarnya, persiapan alat-alat sampai pada pencitraan. Pencitraan adalah kegiatan transformasi dari citra tampak(foto, gambar, lukisan, patung,pemandangan dan lain-lain) menjadi citra digital. Sedangkan alat yang digunakan untuk pengambilan citra adalah sebuah kamera.

2.3 Analisis Stakeholder

Dalam pengembangan "Sistem *Traffic Light* Berbasis *Image Recognition*," penting untuk memahami berbagai pihak yang terlibat dan bagaimana mereka mempengaruhi atau dipengaruhi oleh *Capstone* ini. Pemerintah dan otoritas lalu lintas memegang peran penting, dengan kepentingan utama dalam keselamatan dan efisiensi lalu lintas, serta memiliki pengaruh signifikan melalui kebijakan dan regulasi. Pengguna jalan, termasuk pengendara dan pejalan kaki, secara langsung terpengaruh oleh sistem dan memberikan umpan balik kritis untuk peningkatan. Pengembang, yang bertanggung jawab atas rancangan dan implementasi teknis, mencari untuk menciptakan solusi yang inovatif dan andal. Pada Tabel 2.2 merupakan *Plan Stakeholder Engagement*, sebagai berikut.

Tabel 2.2 *Plan Stakeholder Engagement*

<i>Stakeholder</i>	Tingkat Keterlibatan	Strategi Keterlibatan	<i>Engagement</i>
Pengguna Jalan	Tinggi	Partisipasi	Sebagai pengguna alat yang akan dirancang, dimana masyarakat dapat mengetahui titik kemacetan menggunakan sistem yang akan kami buat, sehingga masyarakat tidak terkena daerah kemacetan dan bisa berangkat dan tiba-tiba tepat waktu di tempat yang dituju
Dinas Perhubungan Yogyakarta	Tinggi	Konsultasi	Sebagai badan pengaturan dan pengelolaan sarana prasarana perparkiran, dimana ketika suatu daerah mengalami kemacetan akibat dari terdapatnya kendaraan yang parkir sembarangan maka tugas Dinas Perhubungan adalah menertibkan daerah lahan parkir supaya tidak terjadi kemacetan dengan memantau melalui sistem ini.

Polisi Lalu Lintas Yogyakarta	Sedang	Kolaborasi	Polisi lalu lintas bertanggung jawab untuk menciptakan ketertiban berkendara yang aman dan nyaman. Dengan adanya sistem yang kami buat polisi lalu lintas diharapkan dapat memudahkan pemantauan kepadatan lalu lintas dari kejauhan walaupun tidak berada di lokasi tersebut, sehingga dapat memberikan keamanan dan ketertiban berkendara dan menciptakan lalu lintas yang aman dan nyaman.
----------------------------------	--------	------------	---

2.4 Analisis Aspek yang Mempengaruhi Sistem

Dalam mengembangkan sistem pengenalan gambar untuk lampu lalu lintas, berbagai aspek harus dianalisis untuk memastikan bahwa sistem beroperasi secara efektif dan efisien. Aspek-aspek ini mencakup faktor teknis, lingkungan, sosial, dan operasional.

Aspek Teknis:

- Kualitas Citra: Resolusi dan kejelasan citra sangat penting untuk akurasi pengenalan gambar.
- Algoritma Pengenalan: Efektivitas algoritma pengenalan gambar, seperti metode apa yang dipakai dan bagaimana algoritma tersebut di-*tune* untuk kondisi lalu lintas yang spesifik.
- Perangkat Keras: Kinerja kamera dan sistem komputasi yang memproses data citra.
- Keandalan Sistem: Termasuk redundansi sistem dan mekanisme pemulihan dari kegagalan.
- Integrasi Sistem: Kemampuan sistem untuk terintegrasi dengan infrastruktur *Traffic Light* yang ada dan sistem manajemen lalu lintas.

Aspek Lingkungan :

- Cuaca : Kebutuhan untuk memantau kepadatan lalu lintas jalan raya semakin tinggi pada musim hujan, sehingga sistem ini harus tahan terhadap hujan supaya tidak mempengaruhi kinerja sistem tersebut.
- Kontur Jalan Raya : Kontur jalan raya merupakan naik turunnya jalan yang menyebabkan sistem transmisi kendaraan bekerja lebih keras, sehingga jika kondisi jalan ramai dapat terjadi kemacetan. Sehingga, perlu memantau kepadatan yang terjadi melalui sistem ini.

- Faktor Jalan Raya : Faktor jalan raya merupakan kondisi jalan raya yang tidak rata atau terjadi kerusakan yang mengakibatkan kendaraan melaju lebih pelan untuk melalui jalan tersebut. Sehingga, dapat menyebabkan kemacetan dan untuk memantau kepadatan tersebut dapat melalui sistem ini.
- Lebar Jalan Raya : Lebar jalan raya diindikasikan dengan jumlah kendaraan yang dapat melalui jalan tersebut. Semakin lebar jalan raya maka semakin kecil terjadinya kemacetan. Supaya sistem dapat menyesuaikan jalan raya tersebut untuk memantau kepadatan lalu lintas yang terjadi.
- Kondisi Pencahayaan: Variabilitas pencahayaan alami dan buatan yang mempengaruhi visibilitas lampu lalu lintas.

2.5 Spesifikasi Sistem

Spesifikasi sistem untuk "*Sistem Traffic Light Berbasis Image Recognition*" dirumuskan berdasarkan kajian literatur, dasar teori, dan informasi yang telah dikumpulkan. Berikut Tabel 2.3 adalah rincian spesifikasi sistem, sebagai berikut :

Tabel 2.3 Plan Spesifikasi Sistem

No	Spesifikasi Alat	Keterangan
1	Komputer	Komputer digunakan untuk komputasi object detection pada video yang dikirim oleh kamera atau CCTV.
2	Kamera / CCTV	Kamera atau CCTV untuk menangkap objek pada jalan raya, untuk menangkap sudut tangkapan gambar kendaraan.
3	Klasifikasi Kendaraan	Pada klasifikasi kendaraan kami menggunakan panjang kendaraan sebagai parameter yang akan dianalisis, kemudian kendaraan tersebut akan diklasifikasikan masuk kedalam kategori kendaraan mobil, motor dan sepeda : Motor = 0,9 - 1,5 m Mobil = 1,9 - 2,2 m Bus/Truck = 3,9 - 4,5 m

4	Kecepatan Kendaraan	Pada saat perhitungan kendaraan sistem akan menghitung kendaraan dengan kecepatan min 10 km/jam dan maksimal kecepatan 30 km/jam
5	Metode Algoritma	pemilihan metode algoritma dan pelatihan model yang akan digunakan untuk pengolahan dan analisis citra dalam sistem pengenalan lampu lalu lintas.

BAB 3. USULAN SOLUSI

3.1 Usulan Solusi 1

Pada usulan solusi ini, menggunakan pengembangan sistem pengaturan lampu lalu lintas yang inovatif, berbasis pengolahan gambar dan deteksi objek dengan menggunakan CCTV di persimpangan jalan. Sistem ini terbagi menjadi dua modul utama: Modul Deteksi Kendaraan dan Algoritma Pergantian Sinyal.

Keunikan dari solusi ini terletak pada kemampuannya mengolah gambar secara *real-time* untuk menghitung kepadatan lalu lintas. Menggunakan algoritma YOLO (*You Only Look Once*) dalam Modul Deteksi Kendaraan, sistem ini mampu mengidentifikasi dan mengklasifikasikan berbagai jenis kendaraan seperti mobil, sepeda motor, bus, dan truk. Informasi mengenai jumlah kendaraan ini digunakan untuk mengkalkulasi kepadatan lalu lintas pada setiap jalur. YOLO menggunakan jaringan *convolutional neural network* (CNN) yang cerdas untuk mendeteksi objek secara *real-time*. Keunggulan YOLO terletak pada akurasinya yang tinggi sambil tetap mampu beroperasi secara *real-time*. Algoritma ini "*You Only Look Once*" pada gambar, artinya hanya memerlukan satu kali proses pengambilan sample gambar melalui jaringan saraf untuk membuat prediksi.

Selanjutnya, Algoritma Pergantian Sinyal, terdapat beberapa faktor untuk menentukan waktu lampu hijau untuk setiap jalur serta pergantian waktu lampu merah yang disesuaikan dengan kebutuhan lapangan, dengan batasan waktu minimum dan maksimum untuk lampu hijau, guna mencegah penundaan yang berlebihan pada jalur tertentu. Berikut faktor-faktor yang dipertimbangkan saat mengembangkan algoritma pergantian sinyal :

- Waktu pemrosesan algoritma untuk menghitung kepadatan lalu lintas dan kemudian durasi lampu hijau – ini menentukan beberapa lama waktu yang dibutuhkan untuk gambar harus diambil secara akurat.
- Jumlah jalur yang dipantau.
- Jumlah total kendaraan dari setiap kelas seperti mobil, truk, sepeda motor, dll.
- Kecepatan rata-rata dari setiap kelas kendaraan ketika lampu hijau menyala, yaitu waktu rata-rata yang dibutuhkan untuk melewati sinyal oleh setiap kelas kendaraan.
- Batas waktu *minimum* dan *maksimum* untuk durasi lampu.

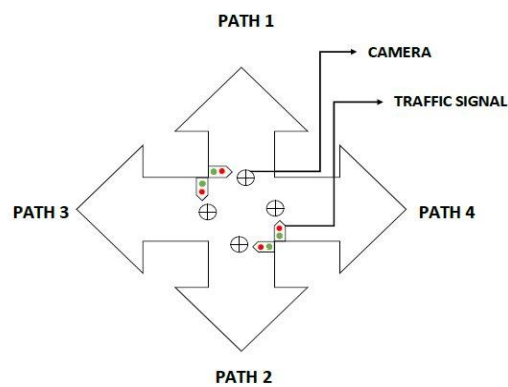
3.1.1 Desain Sistem 1

Diagram alir untuk *Sistem Traffic Light Berbasis Image Recognition*, yang menggunakan metode YOLO untuk *Image Recognition*, dikarenakan untuk memanfaatkan teknologi *Image Recognition* dengan tujuan meningkatkan efisiensi pengaturan lalu lintas urban dengan biaya yang efektif. Metode ini memungkinkan sistem untuk secara akurat menghitung jumlah dan jenis kendaraan pada suatu waktu, yang kemudian dapat dipakai untuk memprediksi aliran lalu lintas dan menyesuaikan waktu sinyal secara dinamis. Data yang diperoleh dari kamera CCTV diolah dalam *real time*, menyediakan input langsung ke dalam algoritma Melalui perhitungan Waktu Sinyal Lampu, sistem dapat menentukan durasi optimal untuk lampu hijau. Keputusan ini didasarkan pada kondisi kepadatan lalu lintas aktual, bukan hanya estimasi atau jadwal yang telah ditetapkan sebelumnya, sehingga memberikan pendekatan yang lebih responsif dan adaptif dalam pengaturan sinyal lalu lintas. Pertama, menentukan batas *minimum* dan *maksimum*, Berikut Tabel 3.1 yang menampilkan batas waktu *minimum* dan *maksimum* untuk durasi lampu lalu lintas :

Tabel 3.1 batas waktu *minimum* dan *maksimum* untuk durasi lampu lalu lintas[10]

Lampu Lalu Lintas	Durasi Minimum	Durasi Maksimum
Merah	60 detik	180 detik
Kuning	2 detik	3 detik
Hijau	25 detik	90 detik

Penentuan area pengamatan sistem ini ditentukan di persimpangan jalan yang dilengkapi dengan kamera yang diarahkan ke seluruh lintasan jalan di persimpangan tersebut. Kamera ini mencakup seluruh sudut pandang lalu lintas di persimpangan untuk mendapatkan data yang komprehensif tentang kondisi lalu lintas, berikut pada Gambar 3.1 rancangan *layout diagram* kamera.



Gambar 3.1 *Layout Diagram* Desain 1

Hasil deteksi kamera, kemudian digunakan untuk mendapatkan persentase kepadatan kendaraan. Persentase kepadatan kendaraan didapat dengan cara membagi jumlah kendaraan dengan kapasitas jalur. Berikut perhitungan persentase kepadatan yang ditunjukkan pada Persamaan 1.

$$\text{Kepadatan} = \frac{\text{Jumlah Kendaraan}}{\text{Kapasitas Area}} \times 100 \quad (3.1)$$

Keterangan :

Kapasitas Area = Jumlah Kapasitas maksimum kendaraan yang dapat ditampung di lalu lintas.

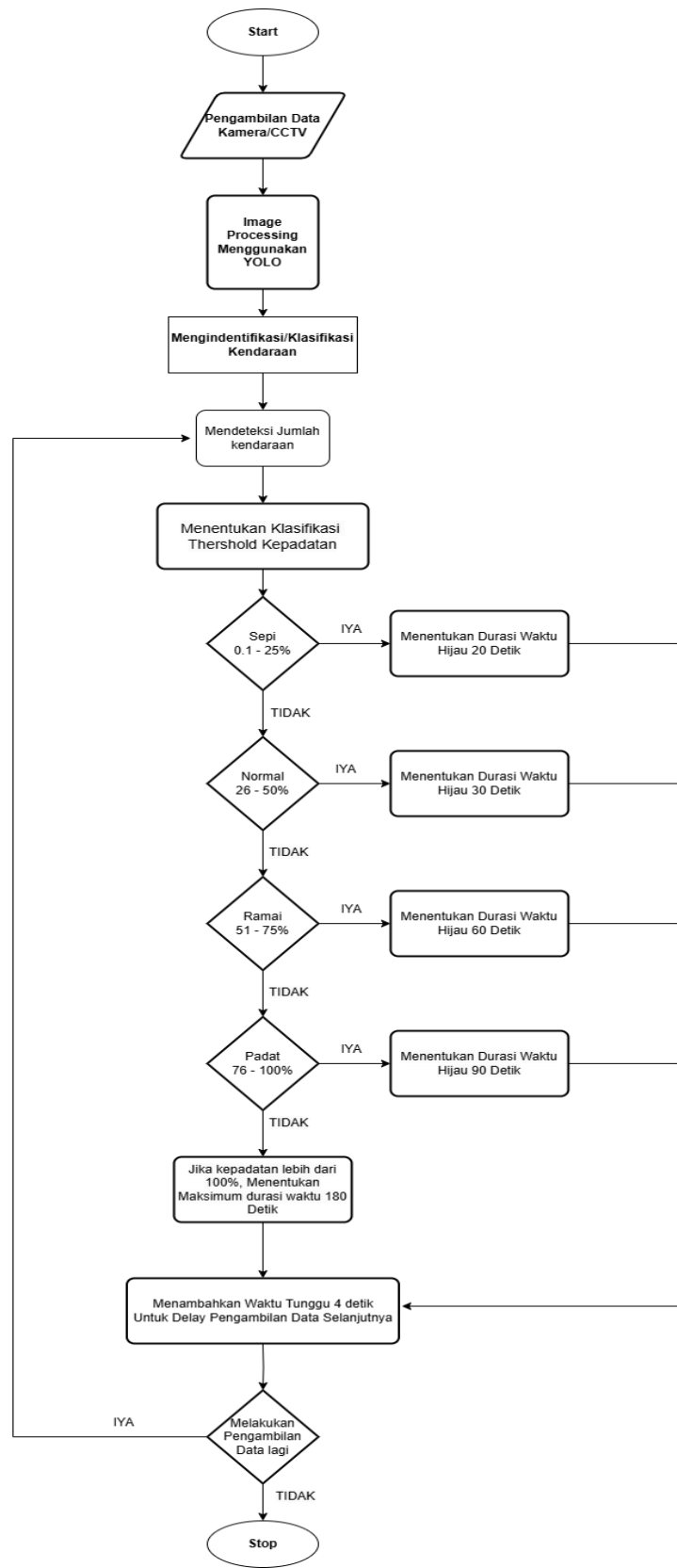
Jumlah Kendaraan = Jumlah kendaraan yang terdeteksi oleh YOLO dalam interval waktu tertentu.

Maka dengan mendapatkan persentase dari kepadatan, maka pada Tabel 3.2 ditentukan klasifikasi kepadatan :

Tabel 3.2 Klasifikasi kepadatan [11]

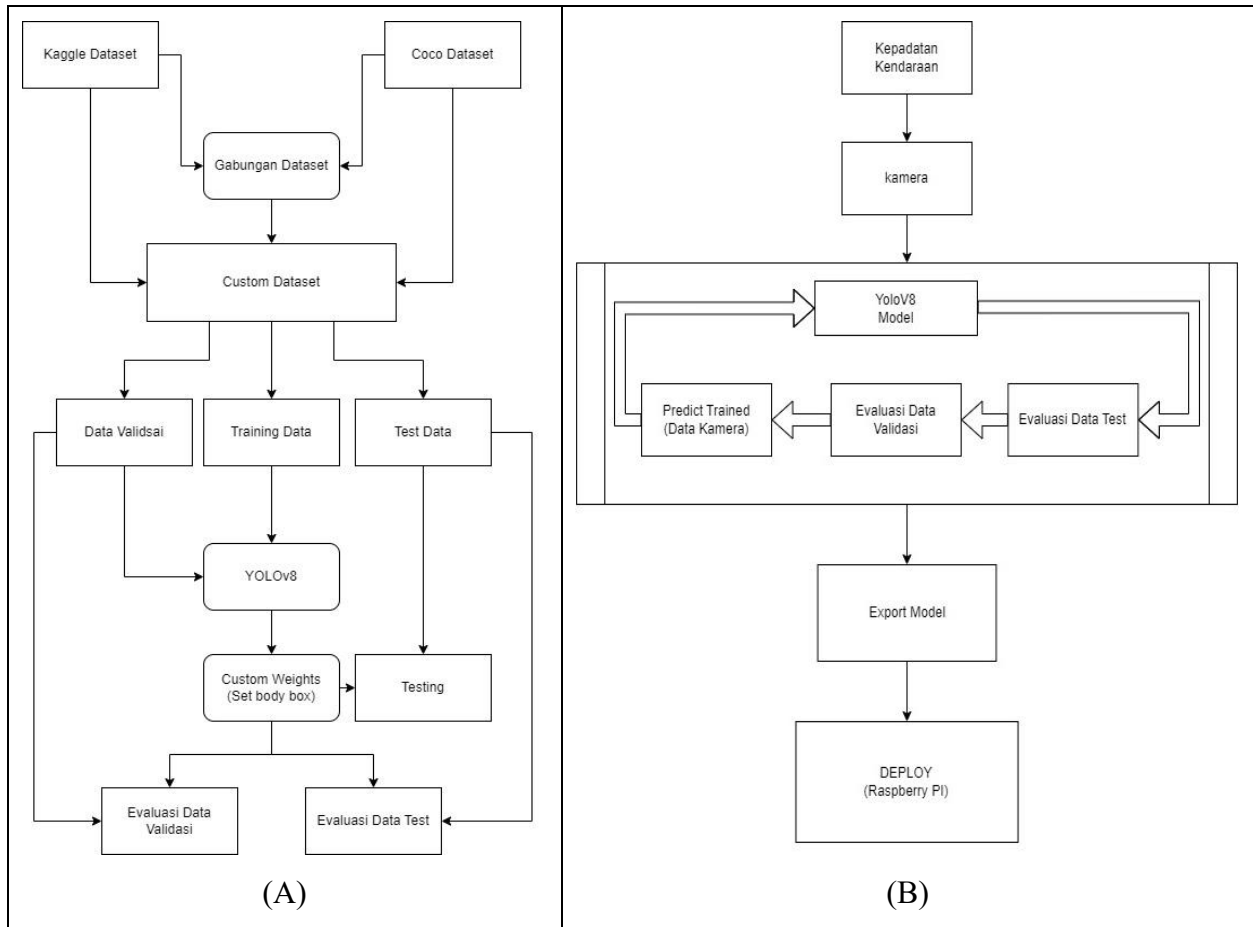
% Kepadatan	0 - 25	26 - 50	51 - 75	76 - 100
Klasifikasi	Sepi	Normal	Ramai	Padat
Durasi Waktu Hijau	20 Detik	30 Detik	60 Detik	90 Detik

Selanjutnya, kami menguraikan rancangan sistem yang akan dikembangkan untuk *Traffic Light* berbasis *Image Recognition*. Sistem ini dirancang untuk mengintegrasikan teknologi pengenalan gambar dengan kontrol lalu lintas, di mana mikrokontroler Raspberry Pi digunakan sebagai otak operasi. Durasi waktu hijau yang diterapkan dalam sistem ini diambil dari penelitian yang terdapat dalam jurnal terkait [11]. Di mana durasi tersebut ditentukan berdasarkan persentase kepadatan lalu lintas yang terdeteksi kemudian disesuaikan dengan kendaraan yang terdeteksi untuk diklasifikasi sesuai Tabel 3.2. Untuk lebih mudah untuk mengetahui alur cara kerja untuk sistem usulan solusi 1, Berikut Gambar 3.2 *FlowChart* Sistem yang dikembangkan:



Gambar 3.2 *Flowchart* Usulan Solusi 1

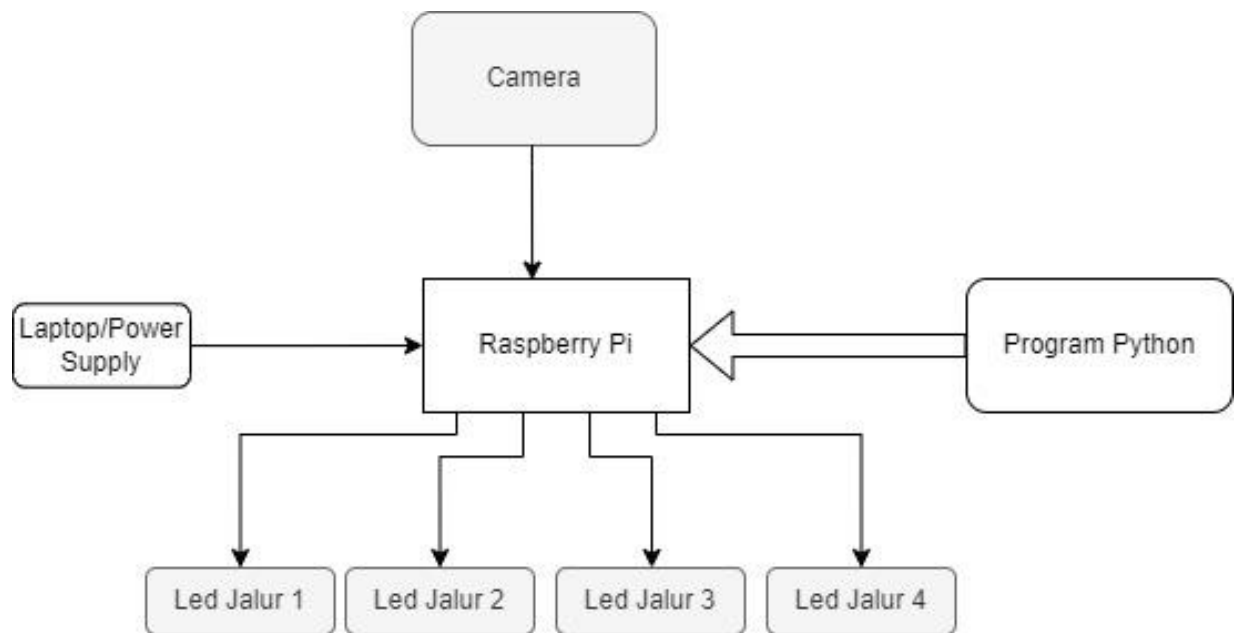
Selanjutnya, menampilkan *Flowchart* untuk *Image Recognition* dari YOLO yang Dimana memakai model versi 8, Berikut Gambar 3.3 *Flowchart* YOLO yang di buat :



Gambar 3.3 *Flowchart* YOLO. (a) Proses *preprocessing*, (b) Proses *Processing* YOLO

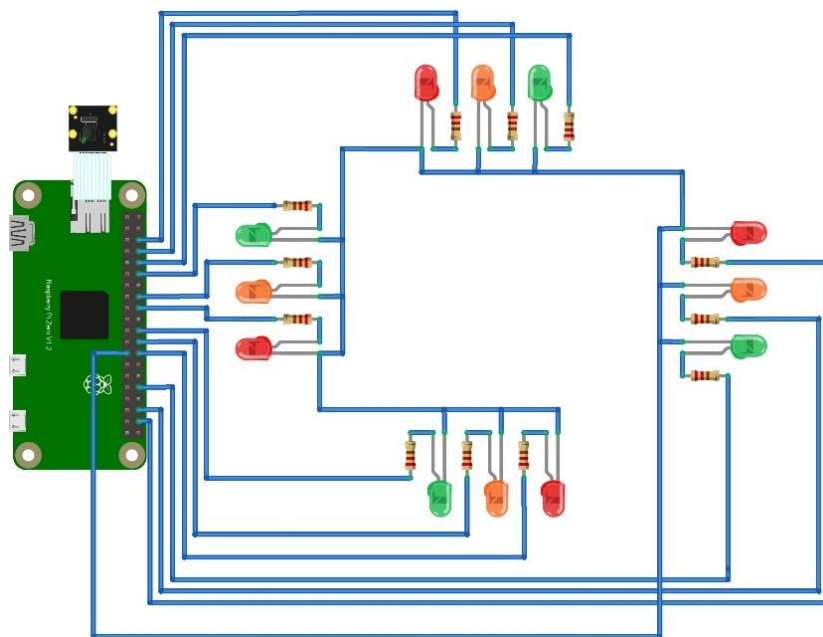
Dalam proses pengembangan sistem pengenalan gambar menggunakan YOLO v8, langkah pertama adalah penggabungan dataset dari sumber Kaggle dan Coco untuk membentuk sebuah dataset gabungan yang komprehensif. Dataset ini kemudian dikustomisasi dan dibagi menjadi tiga segmen utama: data untuk validasi, pelatihan, dan pengujian. Menggunakan segmen data pelatihan, model YOLO v8 dilatih sehingga menghasilkan bobot khusus yang disesuaikan, yang dikenal sebagai custom weights. Setelah proses pelatihan, bobot ini diuji menggunakan data pengujian untuk menilai efektivitas model sehingga menghasilkan evaluasi data *test* dan data validasi. Setelah pelatihan dan pengujian, model dengan bobot yang telah disesuaikan siap untuk melakukan prediksi berdasarkan data masukan dari kamera, yang dalam kasus ini adalah data kepadatan kendaraan. Model yang telah divalidasi kemudian diekspor untuk diimplementasikan pada perangkat keras yang dituju, yaitu *Raspberry Pi*.

Gambar 3.3 memaparkan rancangan sistem secara keseluruhan yang akan dikembangkan. Berikut adalah gambar dari desain usulan 1 tersebut.



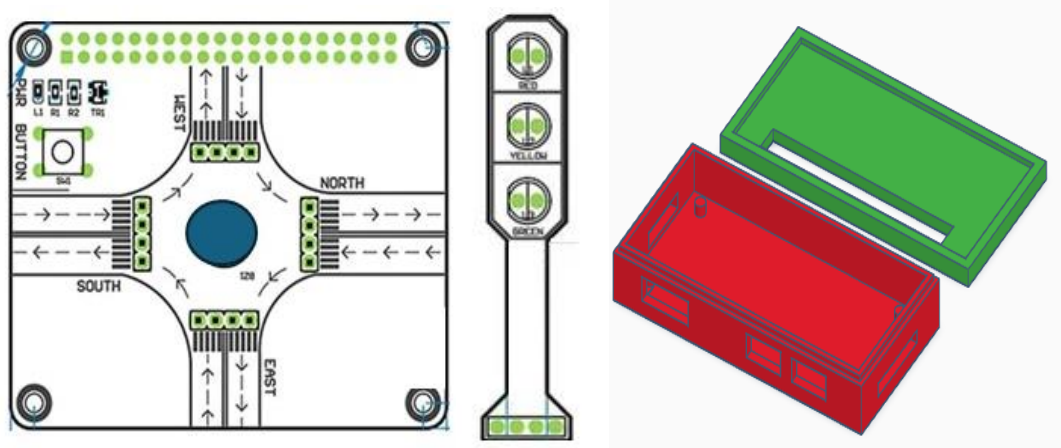
Gambar 3.4 Block Diagram Desain 1

Gambar 3.4 merupakan wiring diagram dari sistem yang diusulkan. Diagram ini mendetailkan bagaimana komponen-komponen seperti LED, resistor, dan sensor terhubung ke *Raspberry Pi*.



Gambar 3.5 *Circuit Diagram* Desain 1

Kemudian kami juga membuat *Case* untuk *Raspberry Pi*, dan *Design* PCB LED untuk layout *Traffic Light* agar memudahkan proses simulasi atau percobaan dan juga untuk keamanan perangkat yang dibuat. Berikut Gambar 3.5 merupakan *design* PCB dan *Case* yang dibuat :



Gambar 3.6 *design* PCB dan *Case*

Untuk dapat memenuhi usulan sistem tersebut, maka diperlukan inventarisasi kebutuhan sistem perangkat keras. Tabel 3.1 memperlihatkan kebutuhan sistem sesuai usulan dan spesifikasi yang dibutuhkan.

Tabel 3.3 Inventarisasi kebutuhan usulan sistem desain 1

No	Nama Alat	Keterangan
1	Kamera CCTV	CCTV adalah sistem teknologi komputer dengan menggunakan kamera untuk merekam gambar suatu peristiwa atau kejadian dan menampilkannya melalui layar monitor pada waktu dan tempat dimana kamera CCTV terpasang.
2	Raspberry	Raspberry adalah sebuah perangkat keras berukuran kecil yang sering digunakan sebagai komputer mini atau mikrokontroler. <i>Raspberry Pi</i> memiliki berbagai model yang berbeda, tetapi umumnya memiliki prosesor, <i>RAM</i> , <i>port USB</i> , <i>port HDMI</i> , dan koneksi jaringan. Perangkat ini dapat digunakan untuk berbagai tujuan, termasuk pembuatan sistem <i>Traffic Light</i> .
3	Server Database	<i>Server</i> yang dimana untuk menampung <i>Database</i> atau basis data adalah kumpulan data yang dikelola sedemikian rupa berdasarkan ketentuan tertentu yang saling berhubungan sehingga mudah dalam pengelolaannya. Melalui pengelolaan tersebut pengguna dapat memperoleh kemudahan dalam mencari informasi, menyimpan informasi dan membuang informasi
4	YOLO	YOLO adalah sistem deteksi objek dalam bidang pengolahan citra dan visi komputer. Ini digunakan untuk mengidentifikasi objek dalam gambar atau video secara <i>real-time</i> dengan cara yang cepat dan akurat. YOLO memiliki struktur jaringan yang terdiri dari tulang punggung, leher, dan kepala detektor serta metode pelatihan model yang ditingkatkan hanya menggunakan satu CPU.

3.1.2 Rencana Anggaran Desain Sistem 1

Tabel 3.4 Rencana anggaran pengembangan sistem desain 1

No.	Item/Pengeluaran	Satuan	Harga satuan	Jumlah	Harga Total
1	<i>Raspberry Pi Zero 2 W</i>	Pcs	Rp. 800.000, -	1	Rp. 800.000, -
2	Modul Camera / CCTV	Pcs	Rp. 100.000, -	1	Rp. 100.000, -
3	Led	Pcs	Rp. 1.000, -	9	Rp. 9.000, -
4	Kabel	Paket	Rp. 20.000, -	1	Rp. 50.000, -
Total Belanja					Rp. 959.000, -

3.1.3 Analisis Risiko Desain 1

Secara teknis, sistem ini bergantung pada akurasi algoritma YOLO dalam mendeteksi dan mengklasifikasikan kendaraan, yang mungkin mengalami kesulitan dalam kondisi pencahayaan yang buruk atau saat menghadapi kendaraan dengan bentuk yang tidak standar. Kegagalan perangkat keras, seperti kerusakan kamera CCTV atau mikrokontroler, juga dapat mengakibatkan hilangnya fungsi vital sistem dan menghambat operasional lalu lintas. Selain itu, risiko keamanan yang melekat pada setiap sistem berbasis *realtime* dapat menyebabkan kebocoran data atau manipulasi sistem, yang berpotensi mengacaukan pola lalu lintas. Dari *counter* waktu, jika terjadi *downtime* sistem selama pemeliharaan atau kegagalan mendadak dapat menyebabkan gangguan serius dalam pengaturan lalu lintas, terutama jika tidak ada rencana kontinjensi yang efektif. Sistem juga harus cukup tangkas untuk menyesuaikan dengan perubahan pola lalu lintas secara *real-time*; keterlambatan dalam adaptasi dapat menghasilkan kemacetan atau bahkan kecelakaan. Selain itu, dalam hal rekayasa, terdapat tantangan untuk merancang perangkat keras yang tahan lama dan mampu beroperasi di bawah berbagai kondisi lingkungan, serta pengembangan perangkat lunak yang dapat dengan cepat memproses *volume data* yang besar tanpa kesalahan. Kekurangan ini membutuhkan perhatian serius dalam desain dan pengujian sistem untuk memastikan bahwa kinerja sistem dapat dipercaya dan konsisten.

3.1.4 Pengukuran Performa

Akurasi deteksi menjadi metrik utama, dimana sistem harus dapat mengenali dan mengklasifikasikan kendaraan dengan tingkat kesalahan yang rendah. Hal ini akan diukur dengan membandingkan data sistem dengan penghitungan manual untuk menentukan efektivitas deteksi. *Confusion Matrix* adalah sebuah tabel yang menunjukkan kinerja dari sebuah model klasifikasi yang memiliki data jawaban benar (*supervise*). Dari tabel yang didapatkan, untuk model klasifikasi yang dimiliki dapat dihitung akurasi, presisi, *F-Score* dan masih banyak lagi variabel yang dapat dihitung berdasarkan kondisi data yang diprediksi atau diklasifikasikan, Isi dari tabel *Confusion Matrix* ada 4, yaitu :

- *True Positive* (TP), kondisi dimana model mengklasifikasikan data sebagai ya (*TRUE*) dan jawaban aktualnya adalah ya (*TRUE*).
- *True Negative* (TN), kondisi dimana model mengklasifikasikan data sebagai tidak (*FALSE*) dan jawaban aktualnya adalah tidak (*FALSE*).
- *False Positive* (FP), kondisi dimana model mengklasifikasikan data sebagai ya (*TRUE*) dan jawaban aktualnya adalah tidak (*FALSE*)
- *False Negative* (FN), kondisi dimana model mengklasifikasikan data sebagai tidak (*FALSE*) dan jawaban aktualnya adalah ya (*TRUE*).

Dimana, dari hasil isi *Confusion Matrix* dapat menentukan akurasi, presisi, *recall* dan *fscore*.

Akurasi merupakan pengukuran seberapa benar sebuah sistem dapat mengklasifikasi dari keseluruhan. Akurasi dapat dihitung menggunakan Persamaan 3.2 :

$$\text{Akurasi} = \frac{TP + TN}{TP + FP + TN + FN} \quad (3.2)$$

Presisi merupakan perbandingan jumlah data yang kategori positif yang diklasifikasikan secara benar oleh sistem dan keseluruhan data yang terklasifikasi positif. Presisi dapat dihitung menggunakan Persamaan 3.3 :

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (3.3)$$

Recall merupakan pengukuran untuk data dengan klasifikasi positif yang benar oleh sistem. *Recall* dapat dihitung menggunakan Persamaan 3.4 :

$$Recall = \frac{TP}{TP + FN} \quad (3.4)$$

F-Score bertujuan untuk menghitung kombinasi dari presisi dan *recall*. *FScore* akan menggunakan harmonic mean dari presisi dan *recall*. *F-Score* dapat dihitung menggunakan Persamaan 3.5 :

$$F - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (3.5)$$

Selain itu, waktu respons sistem akan diukur untuk menilai kecepatan sistem dalam merespons perubahan kondisi lalu lintas dan memperbarui sinyal lampu lalu lintas secara tepat waktu. Dan juga terdapat parameter-parameter yang akan membantu menentukan efektivitas solusi ini dan memastikan bahwa sistem beroperasi sesuai dengan tujuan desainnya. Berikut adalah parameter yang akan digunakan:

- Akurasi Pengenalan Objek: Mengukur persentase objek kendaraan yang terdeteksi dengan parameter *F-Score* oleh sistem YOLO terhadap jumlah sebenarnya yang teridentifikasi melalui metode manual atau hitungan kamera lainnya. Ini adalah indikator utama efisiensi sistem pengenalan gambar.
- Durasi Siklus Lampu Lalu Lintas: Mengukur durasi rata-rata dari siklus lampu lalu lintas (hijau, kuning, merah) dan membandingkannya dengan durasi yang direncanakan berdasarkan klasifikasi kepadatan. Ini menilai seberapa baik sistem mematuhi skenario lampu lalu lintas yang telah ditentukan.
- Efisiensi Lalu Lintas: Mengukur aliran lalu lintas melalui persimpangan menggunakan indikator seperti rata-rata waktu tunggu per kendaraan dan jumlah kendaraan yang melewati persimpangan per unit waktu. Ini memberikan wawasan tentang peningkatan atau penurunan dalam efisiensi aliran lalu lintas.
- Skalabilitas dan Adaptabilitas: Menilai kemampuan sistem untuk beradaptasi dengan perubahan volume lalu lintas dan kondisi lingkungan, serta kemudahan untuk meningkatkan sistem dengan teknologi tambahan atau peningkatan di masa depan.

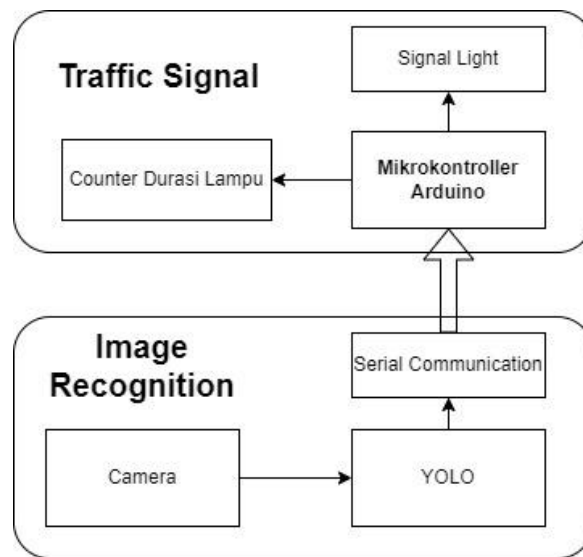
3.2 Usulan Solusi 2

Pada usulan solusi selanjutnya, menggunakan pengembangan sistem *Image Recognition* yang sama dengan usulan Solusi 1 yaitu YOLO, dikarenakan modul sistem ini sangat efisien untuk dikembangkan, oleh karena itu untuk usulan Solusi ini juga memakai YOLO sebagai *Image Recognition* pada sistem ini.

Kemudian, data jumlah kendaraan yang akurat harus diperoleh untuk mengalokasikan waktu secara wajar di setiap jalan. Sistem tersebut meliputi: kamera, mikrokontroler, unit pengolahan gambar, untuk pengiriman data menggunakan kabel data.

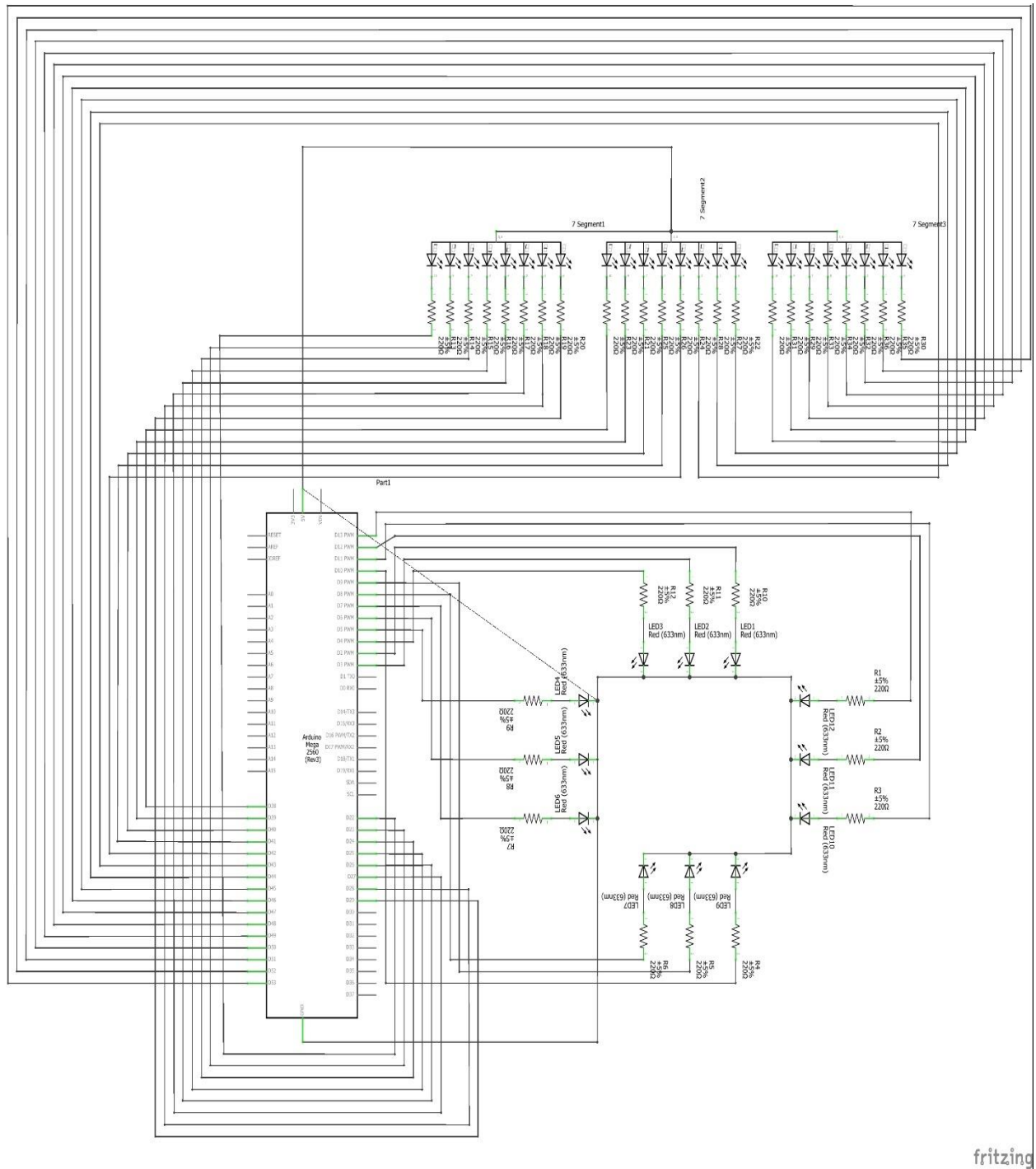
3.2.1 Desain Sistem 2

Digambarkan sistem kontrol lampu lalu lintas yang berintegrasi dengan unit pengenalan gambar. Dengan menggunakan *Mikrokontroler Arduino* yang mengendalikan *Signal Light* atau lampu lalu lintas. Dan terdapat kamera yang mengambil gambar lingkungan lalu lintas dan "YOLO" yang merupakan sistem pengenalan gambar untuk menginterpretasikan data visual tersebut. Berikut Gambar 3.7 skema blok diagram pada usulan Solusi 2.



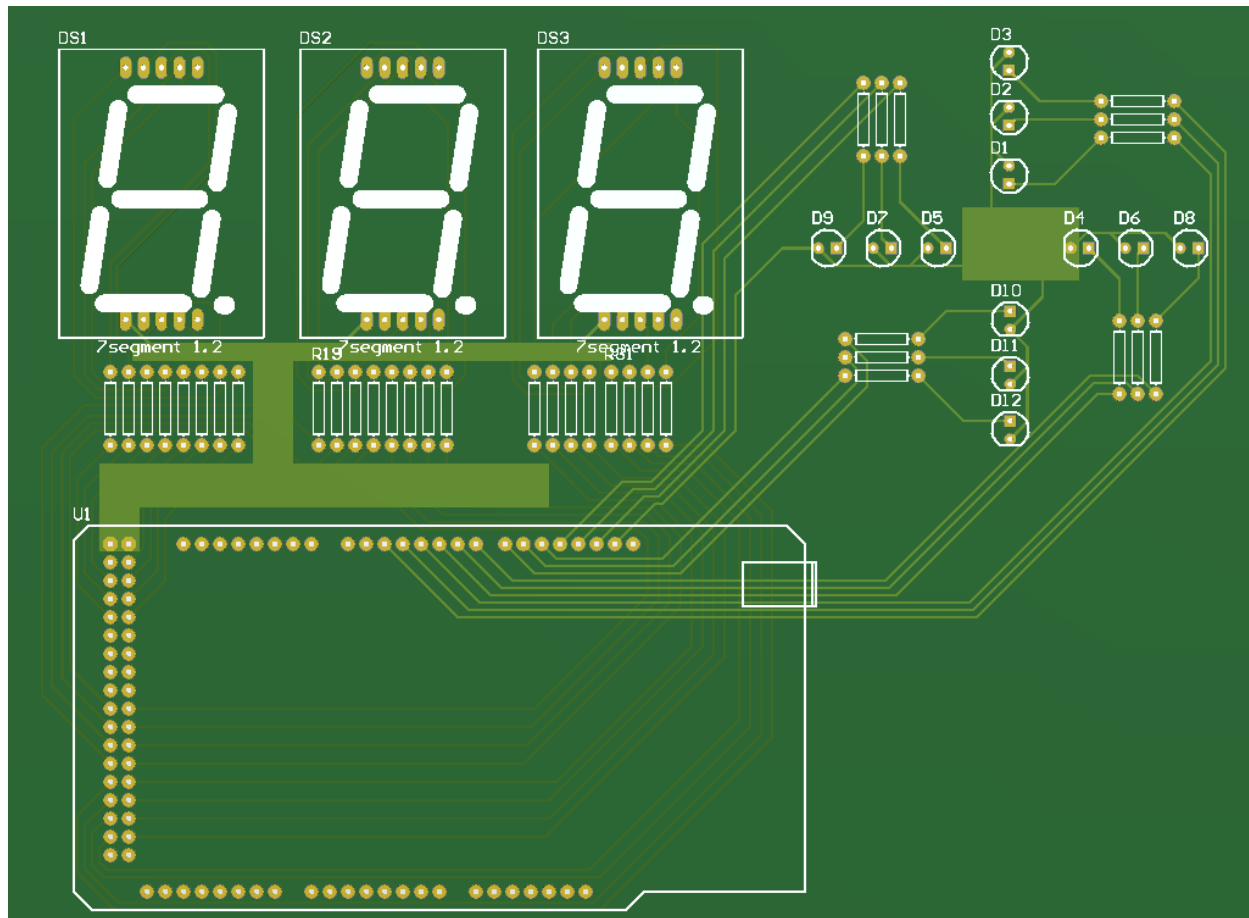
Gambar 3.7 Gambar blok diagram usulan solusi 2

Selanjutnya merupakan *wiring diagram* dari sistem yang diusulkan. Diagram ini mendetailkan bagaimana komponen-komponen terhubung ke mikrokontroler dan bagaimana memvisualisasikan output yang dihasilkan. Berikut Gambar 3.8 *wiring diagram* usulan Solusi 2 :



Gambar 3.8 *Circuit Diagram Desain 2*

Kemudian kami juga membuat *Design PCB LED* untuk layout *Traffic Light* agar memudahkan proses simulasi atau percobaan dan juga untuk keamanan perangkat yang dibuat. Berikut Gambar 3.9 merupakan *design PCB* yang dibuat :



Gambar 3.9 Desain PCB 2

Hasil deteksi kamera, nantinya akan masuk kedalam database (laptop) yang nantinya dikirimkan melalui arduino sebagai pengelola sistem. Pada usulan Solusi ini penentuan durasi kepadatan ditentukan juga dengan Persamaan 3.1.

$$\text{Kepadatan} = \frac{\text{Jumlah Kendaraan}}{\text{Kapasitas Area}} \times 100 \quad (3.1)$$

Maka dengan mendapatkan persentase dari kepadatan, maka pada Tabel 3.5 ditentukan klasifikasi kepadatan, akan tetapi pada usulan solusi 2 ini terdapat juga durasi waktu tetap yang sesuai dengan tingkatan kepadatan yang didapatkan.

Tabel 3.5 Klasifikasi kepadatan

% Kepadatan	0 - 30	31 - 50	51 - 70	71 - 100
Klasifikasi	Sepi	Normal	Ramai	Padat
Konstanta Durasi (k)	3 Detik	5 Detik	7 Detik	10 Detik

Dalam upaya untuk mengoptimalkan manajemen aliran lalu lintas pada persimpangan jalan, pengaturan waktu sinyal hijau menjadi parameter kritis. Waktu sinyal hijau, atau *Green Signal Time* (GST), dihitung menggunakan pendekatan berbasis pengenalan gambar untuk secara akurat menentukan kebutuhan waktu berdasarkan kondisi lalu lintas aktual. Persamaan yang digunakan untuk menghitung GST dirancang untuk mencerminkan dinamika lalu lintas secara *real-time* dengan mempertimbangkan beberapa variabel kunci. Maka untuk pertama kecepatan rata rata setiap kendaraan dapat dikalkulasikan dengan Persamaan 3.6 berikut.[12]

$$\text{Durasi Lampu Hijau} = \frac{\text{Jumlah Kendaraan} \times k}{\text{Jumlah Jalur}} \quad (3.6)$$

Keterangan :

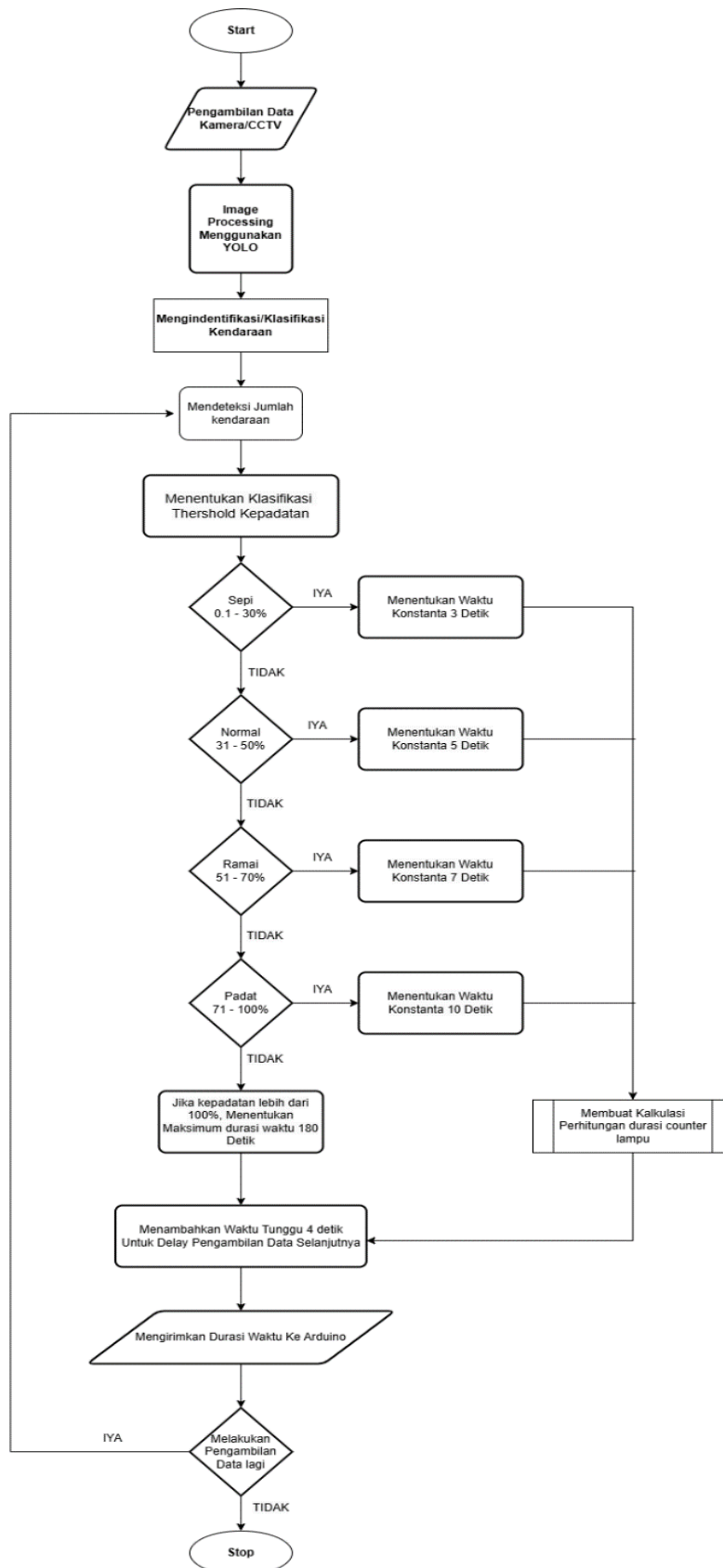
JumlahKendaraan = yaitu jumlah kendaraan dari tiap kelas yang terdeteksi.

k = Konstanta melewati *Traffic Light*, Dari Tabel Klasifikasi 3.5.

Jumlah Jalur = Jumlah Jalur pada *Traffic Light*.

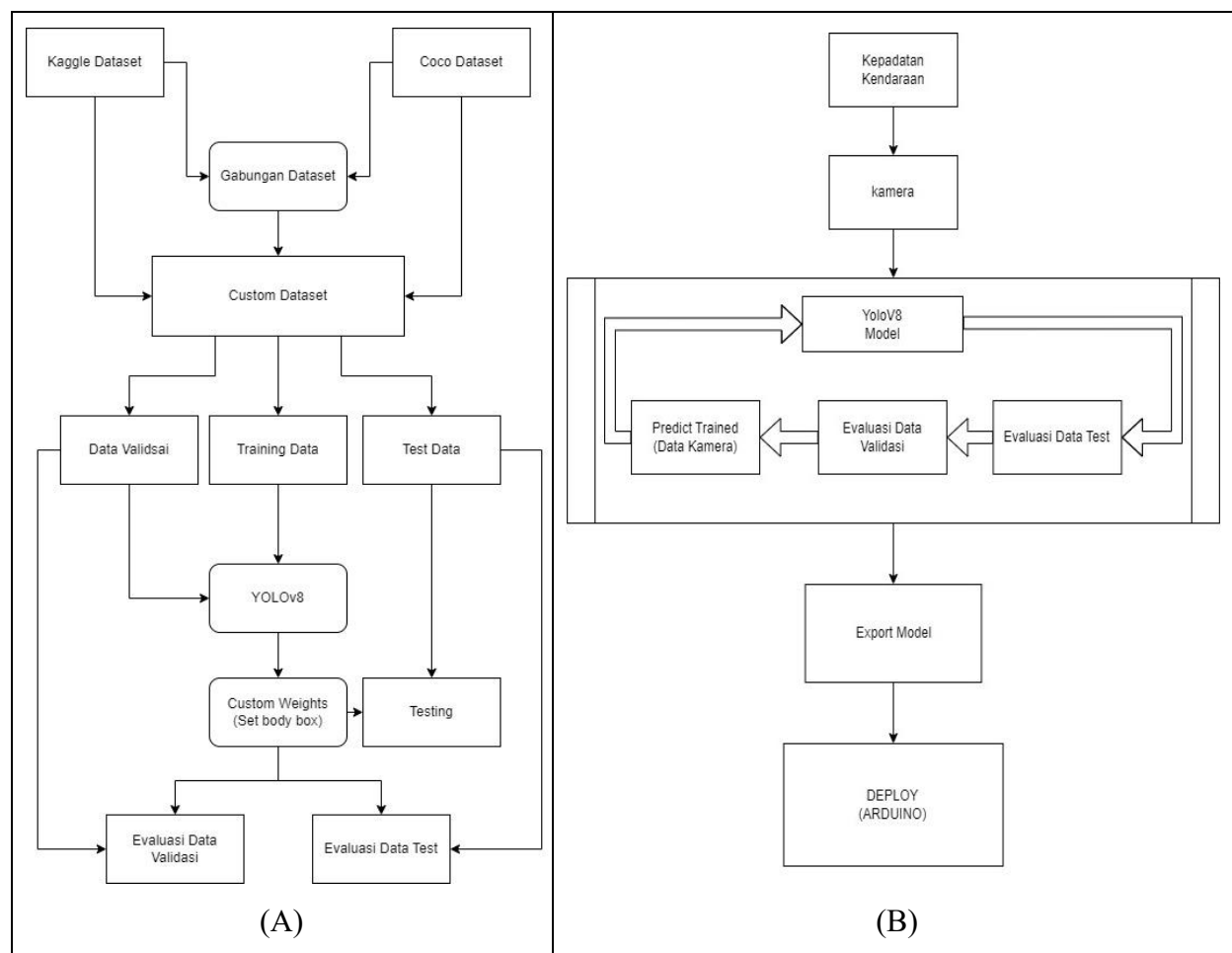
Konstanta "k" dalam rumus ini merujuk pada durasi waktu yang diambil dari Tabel Klasifikasi 3.5, yang membagi kepadatan lalu lintas berdasarkan persentase. Nilai konstanta ini dikembangkan berdasarkan temuan dalam jurnal Smart Control of Traffic Light Using Artificial Intelligence [12]. Meskipun metode perhitungan kepadatan lalu lintas pada akhirnya menjumlahkan seluruh kendaraan dari berbagai kelas. Pada solusi kedua ini, kendaraan diklasifikasikan ke dalam beberapa kelas untuk memastikan bahwa setiap jenis kendaraan diperlakukan sesuai dengan karakteristiknya masing-masing. Alasan utama untuk mengklasifikasikan tipe kendaraan adalah agar model deteksi YOLO dapat mengenali dan membedakan berbagai jenis kendaraan, seperti mobil, motor, bus, dan truk. Dengan demikian, sistem dapat lebih akurat dalam menghitung jumlah kendaraan berdasarkan tipe. Ini memungkinkan penyesuaian yang lebih presisi terhadap pola lalu lintas aktual, bertujuan untuk mengurangi waktu tunggu, meminimalisir kemacetan, dan meningkatkan efisiensi lalu lintas di persimpangan. Berikut contoh perhitungan durasi lampu menggunakan persamaan diatas.

Dengan klasifikasi kepadatan dan waktu durasi lampu yang telah dirancang diatas, Berikut Gambar 3.10 *Flowchart* Sistem yang dikembangkan pada usulan Solusi 2 ini :



Gambar 3.10 *Flow Chart* Usulan Solusi 2

Selanjutnya, menampilkan *Flowchart* untuk *Image Recognition* dari YOLO yang Dimana sama dengan model usulan Solusi 1, Berikut Gambar 3.11 *Flowchart* YOLO yang di buat :



Gambar 3.11 *Flowchart* YOLO. (a) Proses *preprocessing*, (b) Proses *Processing* YOLO

Untuk dapat memenuhi usulan sistem tersebut, maka diperlukan inventarisasi kebutuhan sistem perangkat keras. Tabel 3.6 memperlihatkan kebutuhan sistem sesuai usulan dan spesifikasi yang dibutuhkan.

Tabel 3.6 Inventarisasi kebutuhan usulan sistem desain 2

No	Nama Alat	Keterangan
1	Kamera	Alat ini digunakan untuk merekam gambar atau video dari lingkungan sekitar. Kamera diperlukan untuk mengambil input visual yang kemudian akan diproses oleh sistem pengenalan gambar.
2	Arduino	Arduino adalah papan mikrokontroler yang bersifat open-source dan digunakan untuk membuat prototipe elektronik. Arduino dapat diprogram dengan menggunakan Arduino IDE

No	Nama Alat	Keterangan
		untuk mengontrol dan mengoperasikan berbagai modul elektronik seperti sensor, motor, dan lampu.
3	YOLO	YOLO adalah sistem deteksi objek dalam bidang pengolahan citra dan visi komputer. Ini digunakan untuk mengidentifikasi objek dalam gambar atau video secara <i>real-time</i> dengan cara yang cepat dan akurat.

3.2.2 Rencana Anggaran Desain 2

Tabel 3.7 Rencana anggaran pengembangan sistem desain 1

No.	Item/Pengeluaran	Satuan	Harga satuan	Jumlah	Harga Total
1	<i>Arduino Mega</i>	Pcs	Rp. 500.000, -	1	Rp. 500.000, -
2	Modul Camera / CCTV	Pcs	Rp. 100.000, -	1	Rp. 100.000, -
3	Led	Pcs	Rp. 1.000, -	12	Rp. 12.000, -
4	<i>7-Segment</i>	Pcs	Rp. 5.000, -	3	Rp. 15.000, -
4	Board PCB Simulation	Pcs	Rp. 200.000, -	1	Rp. 200.000, -
Total Belanja					Rp. 828.000, -

3.2.3 Analisis Risiko Desain 2

Analisis risiko pada desain sistem *Traffic Light* Berbasis *Image Recognition* ini memfokuskan pada beberapa faktor yang dapat mempengaruhi efektivitas dan akurasi sistem. Salah satu risiko utama Risiko Adaptabilitas Durasi Waktu: Sistem yang telah ditentukan untuk mengatur durasi lampu lalu lintas berdasarkan kepadatan kendaraan mungkin tidak cukup adaptif untuk menanggapi kondisi lalu lintas yang dinamis dan berubah-ubah. Karena durasi lampu hijau telah ditentukan secara statis berdasarkan persentase kepadatan, sistem mungkin tidak dapat menyesuaikan diri secara efektif dengan fluktuasi tak terduga dalam aliran lalu lintas, seperti yang mungkin terjadi selama kejadian darurat, acara khusus, atau pembatasan jalan mendadak. Dan bergantung pada satu metode untuk menentukan durasi lampu lalu lintas dalam hal ini, perhitungan kepadatan kendaraan tanpa mempertimbangkan faktor lain (seperti kondisi cuaca atau kecelakaan) dapat menyebabkan sistem tidak cukup tangguh.

3.2.4 Pengukuran Performa

Untuk mengukur performa dari desain sistem kontrol lampu lalu lintas yang terintegrasi dengan teknologi pengenalan gambar, saya akan menggunakan beberapa parameter yang sama pada usulan Solusi 1 untuk mengukur ketepatan *image recognition*. Dan juga terdapat parameter-parameter yang akan membantu menentukan efektivitas solusi ini dan memastikan bahwa sistem beroperasi sesuai dengan tujuan desainnya. Berikut adalah parameter yang akan digunakan:

- Akurasi Pengenalan Objek: Mengukur persentase objek kendaraan yang terdeteksi dengan parameter *Confusion Matrix* seperti pada Solusi 1 dan *F-Score* oleh sistem YOLO terhadap jumlah sebenarnya yang teridentifikasi melalui metode manual atau hitungan kamera lainnya. Ini adalah indikator utama efisiensi sistem pengenalan gambar.
- Durasi Siklus Lampu Lalu Lintas: Mengukur durasi rata-rata dari siklus lampu lalu lintas (hijau, kuning, merah) dan membandingkannya dengan durasi yang direncanakan berdasarkan klasifikasi kepadatan. Ini menilai seberapa baik sistem mematuhi skenario lampu lalu lintas yang telah ditentukan.
- Efisiensi Lalu Lintas: Mengukur aliran lalu lintas melalui persimpangan menggunakan indikator seperti rata-rata waktu tunggu per kendaraan dan jumlah kendaraan yang melewati persimpangan per unit waktu. Ini memberikan wawasan tentang peningkatan atau penurunan dalam efisiensi aliran lalu lintas.
- Skalabilitas dan Adaptabilitas: Menilai kemampuan sistem untuk beradaptasi dengan perubahan volume lalu lintas dan kondisi lingkungan, serta kemudahan untuk meningkatkan sistem dengan teknologi tambahan atau peningkatan di masa depan.

3.3 Analisis dan Penentuan Usulan Solusi/Desain Terbaik

Menguraikan dua usulan solusi yang dirancang untuk mengatasi masalah ini. Setiap usulan solusi memiliki pendekatan yang berbeda, dengan teknologi *Image Recognition* sebagai fokus utama, dan kami akan membandingkannya untuk menentukan usulan terbaik, dan menggunakan *decision matrix*. Berikut Tabel 3.8 *decision matrix*.

Penilaian terhadap masing-masing solusi dilakukan berdasarkan empat kriteria utama: Akurasi Deteksi, Responsif, Skalabilitas, dan Efisiensi Biaya. Setiap kriteria diberikan bobot tertentu sesuai dengan tingkat kepentingannya dalam keseluruhan sistem:

- **Akurasi Deteksi:** 30%
- **Responsif:** 30%
- **Skalabilitas:** 15%
- **Efisiensi Biaya:** 25%

Setiap kriteria kemudian dinilai menggunakan skala 1-5 dengan definisi berikut:

1: Sangat Buruk

2: Buruk

3: Cukup

4: Baik

5: Sangat Baik

Nilai yang diberikan untuk setiap kriteria dikalikan dengan bobot yang telah ditetapkan, dan hasilnya dijumlahkan untuk menghasilkan total poin bagi setiap solusi. Solusi dengan total poin tertinggi dianggap sebagai solusi terbaik.

Tabel 3.8 Rencana anggaran pengembangan sistem desain 1

Kriteria	Akurasi Deteksi	Responsif	Skalabilitas	Efisiensi Biaya	POIN
BOBOT	30%	30%	15%	25%	100%
SOLUSI 1	4	3	4	3	3.45
SOLUSI 2	4	4	3	4	3.85

Dari kedua usulan solusi dan Tabel 3.8 *decision matrix*, dimana sama-sama menggunakan modul algoritma " *You Only Look Once* " atau disingkat YOLO sebagai modul *Image Recognition*. Kami menjatuhkan usulan Solusi Kedua yang kami anggap lebih baik. Dimana Adaptabilitas yang Tinggi pada Solusi 2 memungkinkan sistem untuk secara dinamis menyesuaikan waktu sinyal lampu hijau berdasarkan kondisi lalu lintas aktual. Penggunaan Arduino sebagai otak dari operasi memungkinkan fleksibilitas dalam mengembangkan sistem. Arduino adalah platform yang open source dan dapat diandalkan dari penggunaan mikrokontroler usulan Solusi 2, Arduino jauh lebih terjangkau dari Raspybery. Berikut Tabel 3.9 rencana anggaran perkiraan untuk pengembangan Usulan Solusi yang dipilih:

Tabel 3.9 Rencana anggaran pengembangan *sistem* terbaik

No.	Item/Pengeluaran	Satuan	Harga satuan	Jumlah	Harga Total
1	<i>Arduino Mega</i>	Pcs	Rp. 500.000, -	1	Rp. 500.000, -
2	Modul Camera / CCTV	Pcs	Rp. 100.000, -	1	Rp. 100.000, -
3	Led	Pcs	Rp. 1.000, -	12	Rp. 12.000, -
4	<i>7-Segment</i>	Pcs	Rp. 5.000, -	3	Rp. 15.000, -
5	Board PCB Simulation	Pcs	Rp. 200.000, -	1	Rp. 200.000, -
Total Belanja					Rp. 828.000, -

3.4 Gantt Chart

Tabel 3.10 *Gantt chart pelaksanaan Capstone Project Traffic Light Berbasis Image Recognition*

No.	Kegiatan/Capaian	Bulan ke -											
		1	2	3	4	5	6	7	8	9	10	11	12
1	Survei dan identifikasi permasalahan	R	M										
2	Mencari literatur dan informasi untuk kebutuhan dan spesifikasi sistem		R	R									
3	Mengumpulkan seluruh ide solusi dan finalisasi usulan perancangan sistem beserta manajemen dan rancangan belanja			R,M	R,M	R,M	R,M						
4	Pengumpulan proposal Tugas Akhir 1/ <i>Capstone Project</i> dan seminar						R,M						
5	Pembelian alat dan bahan							R	M				
6	Perancangan sistem sesuai proposal								R,M	R,M	R,M		
7	<i>Testing</i> dan Validasi										R,M	R,M	

No.	Kegiatan/Capaian	Bulan ke -											
		1	2	3	4	5	6	7	8	9	10	11	12
8	Expo dan pengumpulan laporan akhir												R,M

Ket. : PIC – *Person in Charge* (Pihak yang bertanggung untuk kegiatan tersebut) R : Rezadi, M : Raffi

3.5 Realisasi Pelaksanaan Tugas Akhir 1

Tabel 3.11 Realisasi aktivitas pelaksanaan tugas akhir 1

No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
1	Sabtu, 23 September 2023	Mencari Studi Literature dan Jurnal Tentang Materi Capstone	Reza
2	Minggu, 24 September 2023	Progress pembuatan pendahuluan, rumusan masalah, dan tujuan	Reza, Rafi
3	Selasa, 3 Oktober 2023	Revisi Batasan Masalah dan Batasan Realistis	Rafi
4	Selasa, 10 Oktober 2023	Revisi Batasan Masalah dan Batasan Realistis	Rafi
5	Selasa, 17 Oktober 2023	Finalisasi BAB 1	Rafi, Reza
6	Kamis, 19 Oktober 2023	Bimbingan Landasan Teori	Rafi
7	Rabu, 25 Oktober 2023	Mencari studi literature dan observasi terkait tentang materi yang sama	Reza
8	Sabtu, 28 Oktober 2023	Bimbingan TA102	Rafi
9	Sabtu, 4 November 2023	Progress pembuatan TA102	Reza
10	Minggu, 5 November 2023	Progress pembuatan TA102	Reza
11	Senin, 6 November 2023	Penjelasan dan Revisi dokumen TA 102	Rafi, Reza
12	Jumat, 10 November 2023	Survei ke lokasi mitra (Dishub)	Reza, Rafi
13	Selasa, 14 November 2023	Penjelasan usulan solusi TA 103	Reza, Rafi
14	Rabu, 22 November 2023	Pembahasan TA 103	Reza, Rafi
15	Selasa, 5 Desember 2023	Progres usulan solusi	Reza, Rafi
16	Minggu, 10 Desember 2023	Revisi usulan solusi 1	Reza
17	Kamis, 13 Desember 2023	Revisi usulan solusi 2	Rafi

18	Sabtu, 16 Desember 2023	Penentuan Usulan Solusi Terbaik	Reza, Rafi
19	Minggu, 24 Desember 2023	Revisi Penentuan Usulan Solusi Terbaik	Reza
19	Kamis, 28 Desember 2023	Seminar <i>Capstone</i> Project	Reza, Rafi

Dalam pengerjaan *Capstone* "Pengaturan Kepadatan *Traffic Light* Berbasis *Image Recognition*", terdapat beberapa kendala yang perlu diperhatikan, khususnya pada Proses pengembangan model *Image Recognition*, *Tuning Parameter* dan Optimalisasi durasi *Counter* waktu lampu. Dan terakhir mengintegrasikan dengan perangkat keras fisik.

BAB 4. HASIL RANCANGAN DAN METODE PENGUKURAN

4.1 Hasil Rancangan Sistem

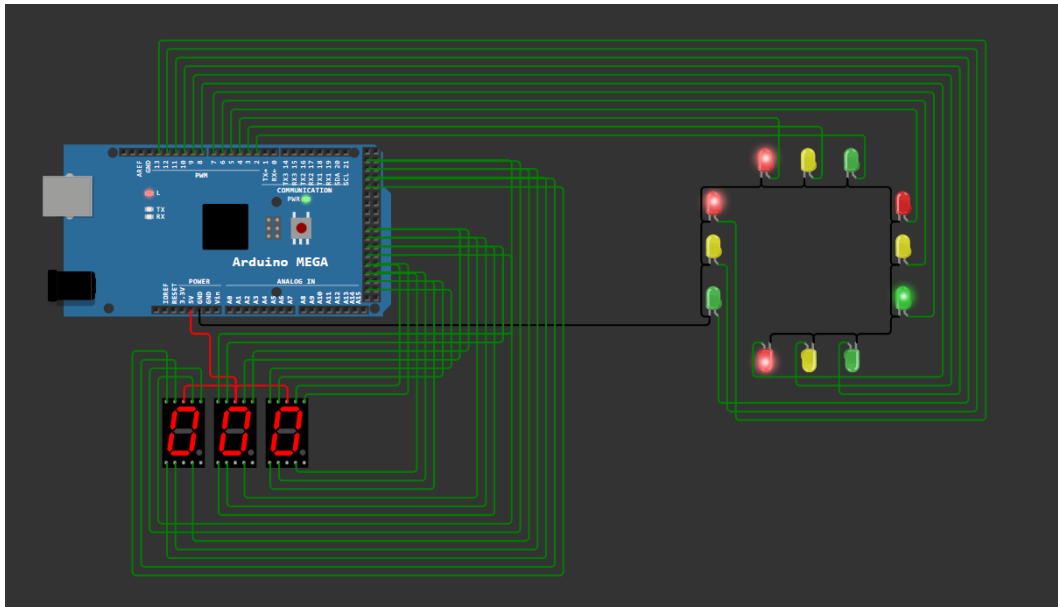
Sistem ini terdiri dari beberapa komponen utama yang bekerja secara sinergis untuk mencapai tujuan tersebut. Kamera digunakan untuk menangkap gambar lingkungan lalu lintas, yang kemudian dianalisis oleh sistem YOLO untuk mendeteksi kendaraan. Hasil deteksi ini dikirim ke *Arduino* melalui komunikasi serial, yang kemudian mengontrol lampu lalu lintas berdasarkan data kepadatan yang diterima. Dengan pendekatan ini, durasi lampu hijau dapat disesuaikan secara dinamis untuk mengakomodasi jumlah kendaraan yang ada, sehingga dapat mengurangi waktu tunggu dan meningkatkan efisiensi lalu lintas di persimpangan.

4.1.1 Rangkaian Elektronik

Rangkaian elektronik utama dalam sistem ini terdiri dari beberapa komponen kunci yang terhubung ke *Arduino* sebagai pusat kontrol. Komponen-komponen tersebut meliputi:

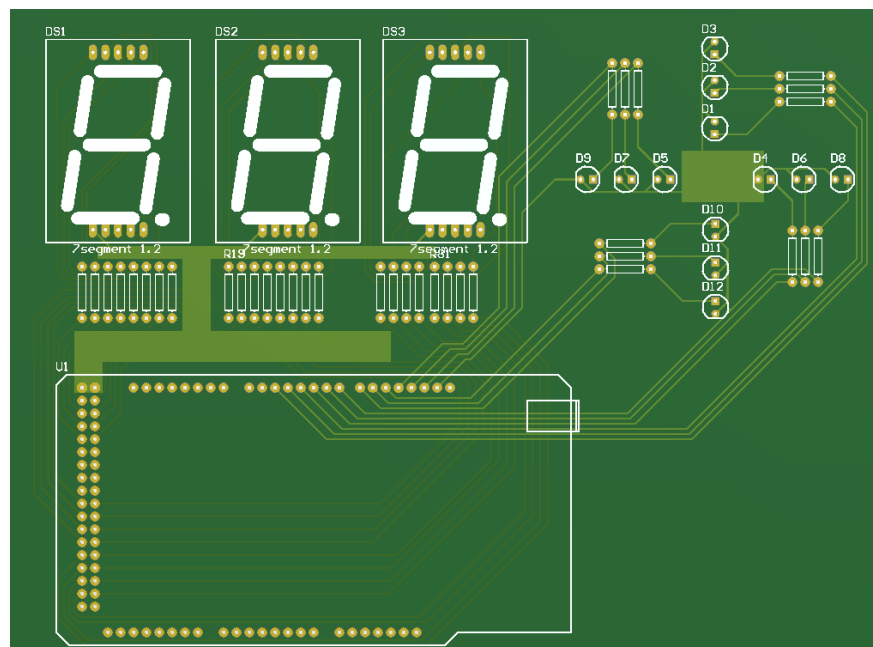
- ***Arduino Mega***: Digunakan sebagai mikrokontroler utama yang mengendalikan seluruh operasional sistem.
- **Kamera CCTV**: Mengambil gambar lingkungan yang akan dianalisis oleh sistem YOLO untuk deteksi kendaraan.
- **7-Segment** : Digunakan untuk menunjukkan counter lampu lalu lintas berdasarkan output dari *Arduino*.
- **LED Lampu Lalu Lintas**: Merah, kuning, dan hijau, masing-masing dikendalikan secara terpisah untuk menunjukkan status lampu lalu lintas.
- **Koneksi USB**: Untuk komunikasi antara *Arduino* dan komputer yang menjalankan sistem YOLO.

Berikut ini adalah gambar skematik dari rangkaian elektronik sistem kontrol lampu lalu lintas yang menunjukkan bagaimana setiap komponen dihubungkan satu sama lain. Gambar 4.1 ini memberikan representasi visual dari koneksi antara *Arduino*, *7-Segment*, dan *LED* yang dimana dibuat dengan aplikasi *Fritzing*.



Gambar 4.2 Desain 3D Simulasi *Hardware* di Website *Wokwi*

Sebagai langkah selanjutnya dalam visualisasi rancangan sistem kami, Gambar 4.3 menunjukkan desain PCB (*Printed Circuit Board*) yang telah kami kembangkan dengan menggunakan *software Altium Designer*. Desain ini dirancang untuk menampung semua komponen elektronik yang telah disebutkan sebelumnya, termasuk *Arduino Mega*, modul *7-Segment*, dan *LED* untuk lampu lalu lintas. Desain PCB ini dirancang untuk optimalisasi tata letak komponen sehingga meminimalkan gangguan sinyal dan memudahkan proses perakitan.



Gambar 4.3 Desain 3D PCB *Hardware*

4.1.3 Optimasi Model YOLO untuk *Image Recognition*

Pengenalan gambar adalah komponen kritikal dalam sistem kontrol lalu lintas yang kami kembangkan, memungkinkan pendeteksian kendaraan secara otomatis dan akurat. Untuk ini, kami memilih model YOLO (*You Only Look Once*) versi 8 X, yang terkenal dengan kecepatan dan akurasi deteksinya. Model ini dilatih menggunakan dataset yang terdiri dari berbagai gambar lingkungan lalu lintas, dengan kendaraan dalam kondisi dan konfigurasi yang beragam.

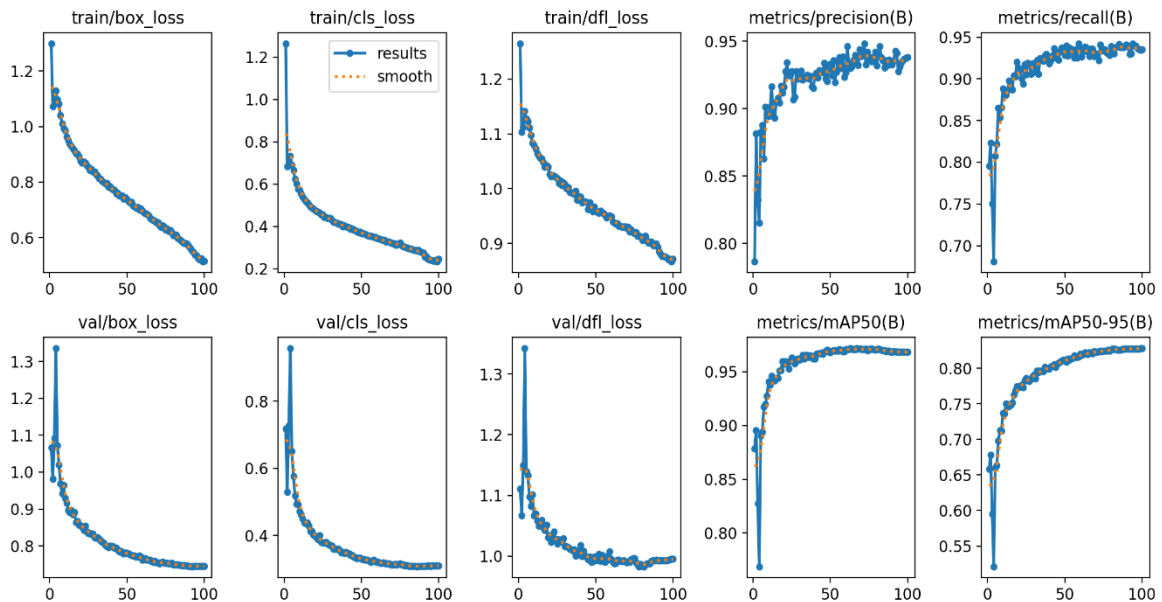
Analisis Kinerja Model, kinerja model dievaluasi menggunakan beberapa metrik yang mencerminkan kemampuan model dalam kondisi operasional yang realistis:

- **Loss Metrics:**
 - **Box Loss** menunjukkan seberapa baik model dapat memprediksi *bounding box* sekitar objek. Nilai yang lebih rendah menandakan prediksi yang lebih akurat.
 - **Class Loss** mengindikasikan efektivitas model dalam mengklasifikasikan objek ke dalam kategori yang benar.
 - **Object Loss** menilai kemampuan model dalam mendeteksi keberadaan objek dalam gambar. Kurva *loss* dari masing-masing komponen ini menunjukkan tren penurunan seiring berjalannya pelatihan, yang mengindikasikan peningkatan kemampuan model.
- **Akurasi Optimasi Model:** Kurva akurasi menampilkan perbandingan antara hasil prediksi model dengan label sebenarnya. Kurva ini menunjukkan peningkatan konsistensi dalam prediksi yang akurat sepanjang *epoch* pelatihan. Tingginya nilai akurasi pada data validasi mengonfirmasi bahwa model memiliki generalisasi yang baik.
 - **Precision** (ketepatan) mengukur persentase prediksi positif yang benar-benar tepat.
 - **Recall** (kepekaan) mengukur persentase dari semua kasus positif yang berhasil dideteksi oleh model.
 - **mAP@0.50** mengukur rata-rata presisi pada IoU (*Intersection over Union* adalah ukuran yang mengukur tumpang tindih antara kotak pembatas yang diprediksi dengan kotak pembatas *ground truth*. IoU memainkan peran mendasar dalam mengevaluasi keakuratan lokalisasi objek.) *threshold* 0.50. Ini adalah ukuran yang lebih toleran karena hanya memerlukan *overlap* sebesar 50% antara *bounding box* prediksi dan *ground truth* untuk menganggap deteksi sebagai positif.

- **mAP@0.50:0.95** adalah metrik yang lebih ketat, menghitung rata-rata presisi pada IoU *threshold* dari 0.50 sampai 0.95 dengan *increment* sebesar 0.05. Metrik ini memberikan penilaian yang lebih komprehensif tentang seberapa akurat model dalam memprediksi lokasi dan ukuran objek.[13]

Model YOLO V8 X ini dilatih dengan tiga skema pembagian data yang berbeda untuk mengevaluasi pengaruh pembagian data terhadap kinerja model. Setiap dataset meliputi gambar yang telah ditandai sebelumnya (*labeled*), yang mencakup berbagai jenis kendaraan seperti mobil, sepeda motor, bus, dan truk, dalam berbagai kondisi pencahayaan dan cuaca:

- **Dataset 70/15/15:** Sebanyak 7457 (70%) dari data digunakan untuk pelatihan, 1598 (15%) untuk validasi, dan 1599 (15%) untuk pengujian. Skema ini memastikan bahwa model cukup terlatih sambil memberikan ruang yang cukup untuk validasi dan pengujian efektivitas model. Berikut hasil dari *training* dataset 70/15/15, berikut merupakan hasil dari *Loss Metrics*.



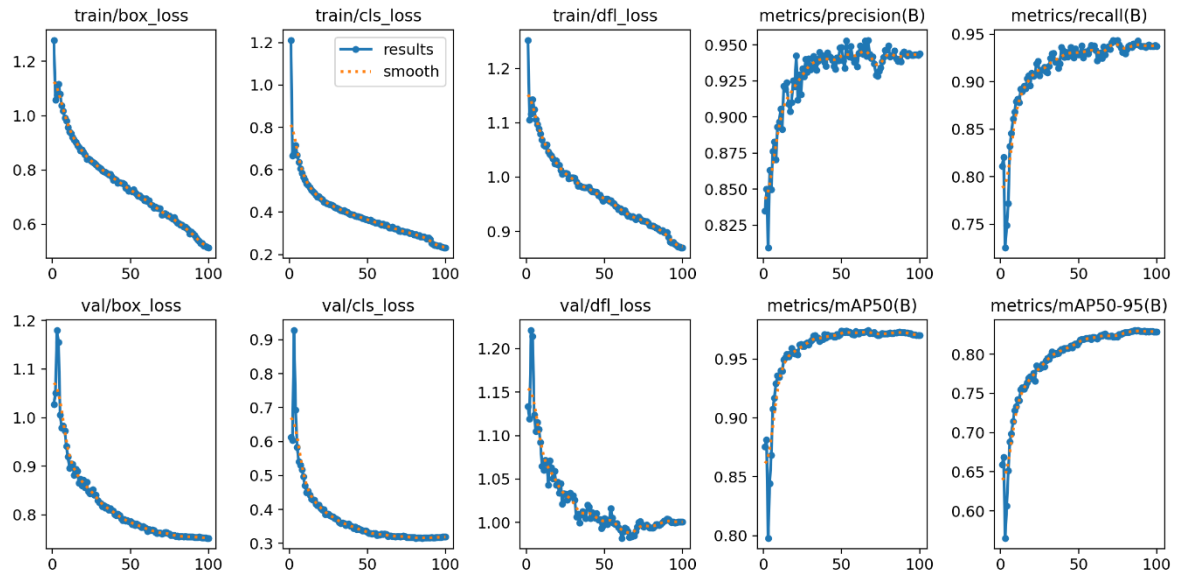
Gambar 4.4 Hasil Optimasi Dataset 70/15/15

Pada Gambar 4.4 menampilkan kurva *Box Loss* dari model YOLO yang diterapkan pada dataset dengan pembagian 70/15/15 selama sesi pelatihan dan validasi. Nilai *Box Loss* yang tercatat adalah 0.515 untuk data pelatihan dan 0.745 untuk data validasi, menunjukkan bahwa walaupun telah tercapai penurunan nilai *loss* yang signifikan dan stabilisasi sejak awal pelatihan, masih terdapat disparitas yang konsisten antara kinerja model pada data

pelatihan dan validasi. Hal ini menandai pentingnya pengoptimalan lebih lanjut untuk menguatkan kemampuan generalisasi model tersebut. Kurva *Class Loss* untuk model YOLO yang dijalankan dengan dataset pembagian 70/15/15, yang menunjukkan bagaimana model berperilaku dalam mengklasifikasikan objek ke dalam kategori yang benar selama proses pelatihan dan validasi. Nilai *Class Loss* yang tercatat adalah 0.247 untuk data pelatihan dan 0.31 untuk data validasi, mengindikasikan bahwa model telah mengalami peningkatan dalam mengklasifikasikan objek dengan lebih tepat seiring dengan berjalannya pelatihan. Kurva *Object Loss* untuk model YOLO yang diimplementasikan pada dataset dengan pembagian 70/15/15 selama fase pelatihan dan validasi. Kurva ini memberikan gambaran tentang efektivitas model dalam mendeteksi keberadaan objek dalam gambar. Sepanjang pelatihan, nilai *Object Loss* tercatat menurun dari nilai awal yang tinggi, menunjukkan adaptasi model yang efektif dalam mengenali objek dalam dataset. Nilai *Object Loss* yang tercatat adalah 0.872 untuk data pelatihan dan 0.996 untuk data validasi, yang menandakan bahwa meskipun model telah menjadi cukup akurat dalam mendeteksi objek pada data pelatihan, masih terdapat tantangan dalam menggeneralisasi kemampuan deteksi tersebut pada data yang tidak dilihat selama pelatihan. Disparitas antara nilai pelatihan dan validasi menggarisbawahi pentingnya memperluas keragaman data dalam dataset pelatihan atau menerapkan teknik seperti *augmentation* untuk meningkatkan kemampuan generalisasi model.

Gambar 4.4 juga menggambarkan kurva kinerja model YOLO V8 X, yang mencakup metrik presisi, *recall*, serta *Mean Average Precision* (mAP) pada *IoU thresholds* 0.50 dan 0.50-0.95 selama pelatihan dan validasi. Dengan presisi sebesar 0.936, model ini menunjukkan efektivitas tinggi dalam menghindari *False Positives*, yang krusial untuk keandalan sistem kontrol lalu lintas. *Recall* sebesar 0.896 mengindikasikan bahwa model mampu mengidentifikasi sebagian besar objek positif, memastikan bahwa sedikit sekali kendaraan yang terlewat. Nilai $mAP@0.50$ sebesar 0.94 (94%) mengonfirmasi kemampuan model dalam mendeteksi objek dengan *overlap* minimal 50% terhadap *ground truth*, sementara $mAP@0.50-0.95$ sebesar 0.806 (80.6%) memberikan gambaran yang lebih komprehensif tentang akurasi model di berbagai tingkat presisi spasial, menjadikan YOLO V8 X sangat handal untuk aplikasi yang membutuhkan identifikasi dan klasifikasi objek yang sangat akurat di kondisi lalu lintas yang beragam.

- **Dataset 80/10/10:** Sebanyak 8523 (80%) dari data digunakan untuk pelatihan, 1065 (10%) untuk validasi, dan 1065 (15%) untuk pengujian. Berikut hasil dari *training* dengan pembagian dataset.



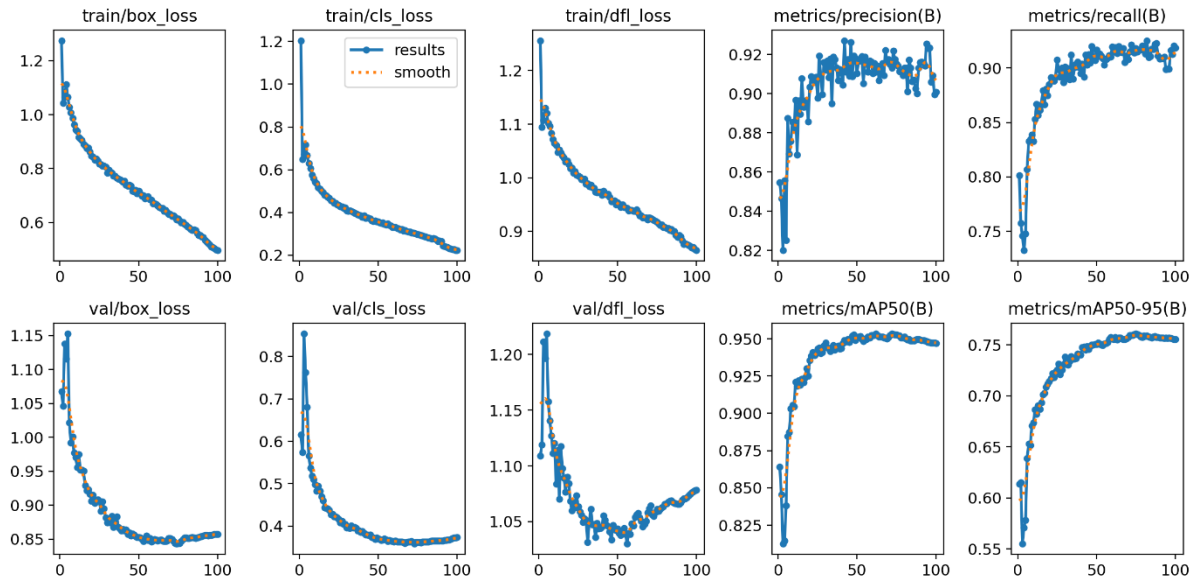
Gambar 4.5 Hasil Optimasi Dataset 80/10/10

Pada Gambar 4.5 menampilkan kurva *Box Loss* untuk model YOLO V8 X, yang dijalankan menggunakan skema pembagian dataset 80/10/10, dimana 80% data digunakan untuk pelatihan, 10% untuk validasi, dan 10% untuk pengujian. Kurva ini menunjukkan bahwa nilai *Box Loss* pada data pelatihan (*train*) adalah 0.513 dan pada data validasi (*val*) adalah 0.752. Terdapat kurva *Class Loss* dari model YOLO yang dijalankan pada dataset dengan pembagian 80/10/10, menyoroti efisiensi model dalam mengklasifikasikan objek ke kategori yang benar selama proses pelatihan dan validasi. Dengan nilai *Class Loss* yang tercatat sebesar 0.232 pada data pelatihan dan 0.32 pada data validasi, grafik menunjukkan penurunan yang konsisten dan stabil pada nilai *loss* selama pelatihan. memperlihatkan kurva *Object Loss* untuk model YOLO pada dataset dengan pembagian 80/10/10, menyoroti efektivitas model dalam mendeteksi keberadaan objek dalam gambar selama proses pelatihan dan validasi. Nilai *Object Loss* yang tercatat adalah 0.87 pada data pelatihan dan 1.001.

Gambar 4.5 memaparkan kurva akurasi model YOLO pada dataset dengan pembagian 80/10/10. Dengan nilai mAP 50 mencapai 0.973 (97.3%) dan mAP 50-95 sebesar 0.83 (83%), grafik ini menunjukkan bahwa model memiliki presisi tinggi dalam kondisi ideal, dengan kemampuan yang sangat baik dalam mengenali objek pada IoU *threshold* 50%.

Nilai *Precision* sebesar 0.938 dan *recall* sebesar 0.943 menegaskan keefektifan model dalam mengidentifikasi sekaligus mengklasifikasikan objek dengan benar dalam kondisi yang beragam, menandakan kinerja yang kuat dalam kondisi operasional yang realistis.

- **Dataset Original:** Sebanyak 8811 (82.7%) dari data digunakan untuk pelatihan, 1143 (10.7%) untuk validasi, dan 700 (6.6%) untuk pengujian. Berikut hasil dari *training* dengan pembagian dataset.



Gambar 4.6 Hasil Optimasi Dataset Original

Gambar 4.5 menampilkan kurva *Box Loss* untuk model YOLO V8 X, yang dijalankan menggunakan dataset original dengan pembagian 82.7% data untuk pelatihan, 10.7% untuk validasi, dan 6.6% untuk pengujian. Kurva ini menunjukkan bahwa nilai *Box Loss* pada data pelatihan (*train*) adalah 0.497 dan pada data validasi (*val*) adalah 0.857. kurva *Class Loss* data pelatihan mencatatkan penurunan *Class Loss* yang stabil hingga mencapai sekitar 0.223, menunjukkan kemampuan model untuk belajar dan beradaptasi dengan baik terhadap data yang tersedia selama fase pelatihan. Sementara itu, data validasi menunjukkan nilai yang lebih tinggi pada 0.373, mengindikasikan bahwa model menghadapi kesulitan dalam mempertahankan performa yang sama ketika dihadapkan pada data baru atau tidak terlihat sebelumnya. Kurva *Object Loss*, kurva menunjukkan bahwa nilai *loss* pada data pelatihan (*train*) mencapai sekitar 0.865, menandakan model belajar dengan efektif selama pelatihan. Namun, pada data validasi (*val*), nilai *loss* mencapai 1.079, yang menunjukkan peningkatan signifikan dari nilai pelatihan.

Gambar 4.6 menampilkan kurva akurasi model YOLO V8 X pada dataset original, yang mencakup metrik seperti $mAP@0.50$ (mean Average *Precision* at Intersection over Union *threshold* of 0.50) dan $mAP@0.50:0.95$. Kurva ini mengungkapkan kinerja model yang kuat dalam memprediksi lokasi dan klasifikasi objek, dengan $mAP@0.50$ mencapai 0.953 dan $mAP@0.50:0.95$ sebesar 0.761, menandakan tingkat deteksi yang tinggi pada *threshold* IoU yang lebih rendah serta kemampuan yang cukup baik pada *threshold* yang lebih ketat. Tingginya nilai presisi 0.915 dan *recall* 0.918 pada data validasi menunjukkan bahwa model tersebut efektif dalam mengidentifikasi mayoritas objek yang benar dengan tingkat kesalahan yang rendah, mendukung keandalan model pada kasus asli.

4.1.4 Prosedur *Counting* dan Mekanisme Sistem

Prosedur *Counting*

Prosedur penghitungan kendaraan menggunakan sistem pengenalan gambar berbasis YOLO (You Only Look Once), yang memungkinkan deteksi objek secara real-time dengan akurasi tinggi. Proses ini dimulai dengan pengambilan data secara langsung dari kamera atau CCTV yang terpasang di lokasi strategis di persimpangan. Gambar yang diterima kemudian diproses melalui model YOLO yang telah dilatih khusus untuk mengenali berbagai jenis kendaraan, seperti mobil, motor, bus, dan truk. Setelah gambar diproses, sistem mengidentifikasi dan mengklasifikasikan kendaraan berdasarkan tipe dan ukurannya. Setiap kendaraan yang terdeteksi akan dihitung, dan jumlah total kendaraan yang teridentifikasi pada interval waktu tertentu dicatat. Data ini penting untuk menentukan kepadatan lalu lintas pada saat itu.

Setelah mendapatkan data jumlah kendaraan yang melewati titik tertentu, langkah selanjutnya adalah mengkategorikan kepadatan lalu lintas. Kategorisasi ini dilakukan berdasarkan persentase kepadatan yang diperoleh dengan Persamaan 3.1. Kemudian Kepadatan lalu lintas diklasifikasikan ke dalam empat kategori berdasarkan persentase kepadatan pada Tabel 3.5. Langkah terakhir untuk mendapatkan durasi lampu hijau dengan menggunakan konstanta durasi (k) ini dimasukkan dalam persamaan 3.6 untuk menghitung durasi lampu hijau (Green Signal Time - GST).

Mekanisme Sistem

Sistem ini menggunakan informasi dari prosedur counting untuk mengatur lampu lalu lintas secara dinamis, berdasarkan data yang dihasilkan oleh algoritma YOLO dalam mendeteksi

dan mengklasifikasikan kendaraan. Berikut adalah langkah-langkah yang diikuti dalam mekanisme sistem ini :

Tabel 4.1 *Code* Sistem Image Recognition

<i>CODE</i>
<pre> import cv2 import serial from ultralytics import YOLO import serial.tools.list_ports # Fungsi untuk menemukan port serial yang tersedia def find_available_port(): ports = list(serial.tools.list_ports.comports()) for p in ports: print(p) if "Arduino" in p.description: return p.device return None # Mencari port yang tersedia port = find_available_port() if port is None: raise Exception("Port tidak tersedia. Pastikan port terhubung dan tidak digunakan oleh aplikasi lain.") # Inisialisasi model YOLOv8 MODEL = 'MODEL_YOLO/best_yogya_8x_80_10_10.pt' model = YOLO(MODEL) # Membuka file video video_path = "DATA_UJI/CCTV Berbasis Kewilayahan _ Pemerintah Kota Yogyakarta - Google Chrome 2024-06-25 22-06-42.mp4" cap = cv2.VideoCapture(video_path) # Inisialisasi variabel untuk menghitung kendaraan vehicle_count = 0 # Penunjuk waktu start_time = cv2.getTickCount() # Waktu mulai frequency = cv2.getTickFrequency() # Frekuensi penunjuk waktu interval = 0 # Inisialisasi serial untuk komunikasi dengan Arduino ser = serial.Serial(port, 9600) # Fungsi untuk mengklasifikasikan kepadatan lalu lintas dan menentukan durasi waktu lampu lalu # lintas def klasifikasikan_kepadatan(kepadatan): if 0 <= kepadatan <= 30: return "Sepi", 3 # Lalu lintas sepi, 3 detik elif 31 <= kepadatan <= 50: return "Normal", 5 # Lalu lintas normal, 5 detik elif 51 <= kepadatan <= 70: return "Ramai", 7 # Lalu lintas ramai, 7 detik elif 71 <= kepadatan <= 100: return "Padat", 10 # Lalu lintas padat, 10 detik else: return "Sangat Padat", 10 # Durasi maksimum untuk keamanan # Fungsi untuk menghitung kepadatan berdasarkan jumlah kendaraan dan kapasitas area def hitung_kepadatan(jumlah_kendaraan, kapasitas_area): return (jumlah_kendaraan / kapasitas_area) * 100 # Fungsi untuk menghitung durasi siklus terlama berdasarkan jumlah kendaraan dan rata-rata # waktu def hitung_durasi_siklus_terlama(jumlah_kendaraan, waktu_rata, jumlah_jalur): </pre>

```

total_durasi = jumlah_kendaraan * waktu_rata
return total_durasi / jumlah_jalur

# Melakukan iterasi melalui frame kamera
while cap.isOpened():
    # Membaca satu frame dari kamera
    success, frame = cap.read()

    if success:
        # Menghitung waktu yang telah berlalu
        elapsed_time = (cv2.getTickCount() - start_time) / frequency

        # Tentukan apakah akan menggunakan verbose
        verbose = elapsed_time >= interval

        # Menjalankan pelacakan YOLOv8 pada frame, dengan verbose sesuai kondisi
        results = model.track(frame, persist=True, verbose=verbose)

        if results[0].boxes is not None:
            # Mendapatkan kotak dan ID pelacakan
            boxes = results[0].boxes.xywh.cpu() # Koordinat kotak dalam format x, y, width,
height
            # Visualisasi hasil pada frame
            annotated_frame = results[0].plot(line_width=1, font_size=6)

            # Menghitung jumlah kendaraan yang terdeteksi
            vehicle_count = len(boxes)

            # Menampilkan frame dengan anotasi
            cv2.imshow("Tracking", annotated_frame)
        else:
            # Jika tidak ada kotak terdeteksi
            print("No Vehicle detected in the current frame.")
            vehicle_count = 0

        # Menghentikan loop jika tombol 'q' ditekan
        if cv2.waitKey(1) & 0xFF == ord("q"):
            break

        # Menghitung waktu yang telah berlalu
        elapsed_time = (cv2.getTickCount() - start_time) / frequency

        # Jika interval waktu lampu lalu lintas telah berlalu, cetak total kendaraan dan reset
waktu
        if verbose:
            # Reset penunjuk waktu
            start_time = cv2.getTickCount()

            # Parameter
            jumlah_kendaraan = vehicle_count # Jumlah total kendaraan
            jumlah_jalur = 4 # Jumlah jalur
            kapasitas_area = 50 # Kapasitas maksimum area

            # Hitung kepadatan lalu lintas
            kepadatan = hitung_kepadatan(jumlah_kendaraan, kapasitas_area)
            print(f"Jumlah kendaraan terdeteksi: {jumlah_kendaraan}")
            print(f"Kepadatan Lalu Lintas: {kepadatan:.2f}%")

            # Klasifikasikan kepadatan lalu lintas dan dapatkan waktu rata-rata
            klasifikasi, waktu_rata = klasifikasikan_kepadatan(kepadatan)

            # Hitung durasi siklus terlama
            durasi_siklus_terlama = hitung_durasi_siklus_terlama(jumlah_kendaraan, waktu_rata,
jumlah_jalur)

            # Cek jika jumlah kendaraan melebihi 50 atau kepadatan lebih dari 100%
            if kepadatan > 100:
                durasi_lampu = 180
                print("Kondisi Lalu Lintas: Sangat Padat (jumlah kepadatan > 100%)")
            else:
                # Gunakan durasi siklus terlama langsung untuk durasi lampu lalu lintas

```

```

        durasi_lampu = round(min(max(durasi_siklus_terlama, 5), 180))
        print(f"Kondisi Lalu Lintas: {klasifikasi}")
        print(f"Durasi Konstanta: {waktu_rata} detik")

    print(f"Durasi Lampu Lalu Lintas yang Dihitung: {durasi_siklus_terlama:.2f} detik")
    print(f"Durasi Lampu Lalu Lintas yang Diterapkan: {durasi_lampu} detik")

    # Durasi lampu tunggu tetap 4 detik
    durasi_tunggu = 4

    # Perbarui interval berdasarkan durasi lampu
    interval = durasi_lampu + durasi_tunggu

    # Mengirim hanya durasi lampu ke Arduino melalui serial
    data = f"{durasi_lampu:.2f}\n"
    print(f"Mengirim data ke Arduino: {data.strip()}") # Tambahkan print untuk
debugging
    ser.write(data.encode())

    else:
        # Menghentikan loop jika akhir dari video tercapai
        break

# Melepaskan objek capture video dan menutup jendela tampilan
cap.release()
cv2.destroyAllWindows()
surculose()

```

Dari Tabel 4.1 *Code Sistem Image Recognition*, Berikut Penjelasan tentang Mekanismenya :

- **Inisialisasi Model dan Port Serial:** Sistem memulai dengan menginisialisasi model YOLO dan mengkonfigurasi port serial untuk komunikasi dengan Arduino. Ini termasuk memuat model YOLO yang telah dilatih dan menentukan port serial yang digunakan oleh Arduino.
- **Pengambilan Data Video:** Sistem kemudian memulai pengambilan data dari video yang diperoleh dari CCTV. Setiap frame video diolah untuk mendeteksi kehadiran kendaraan menggunakan model YOLO.
- **Pengolahan Frame dan Deteksi Kendaraan:** Untuk setiap frame yang berhasil dibaca, sistem menggunakan model YOLO untuk mendeteksi kendaraan. Deteksi ini melibatkan penggunaan bounding boxes, yang adalah kotak yang mengelilingi setiap kendaraan terdeteksi pada frame tersebut. Jumlah bounding boxes yang terdeteksi per frame dihitung sebagai jumlah kendaraan.
- **Klasifikasi Kepadatan dan Penentuan Waktu Lampu:** Berdasarkan jumlah kendaraan yang dihitung, sistem menghitung kepadatan lalu lintas dan klasifikasi kepadatan. Berdasarkan klasifikasi ini, sistem menetapkan durasi lampu lalu lintas yang sesuai untuk mengoptimalkan aliran kendaraan dan mengurangi waktu tunggu.

- **Mengirim Data ke Arduino:** Setelah durasi lampu lalu lintas ditentukan, sistem mengirimkan nilai durasi ini ke Arduino melalui port serial. Arduino kemudian mengatur lampu lalu lintas berdasarkan durasi yang diterima.
- **Looping dan Delay Pengambilan Gambar:** Sistem melakukan looping untuk terus memproses frame baru setiap kali durasi lampu selesai dihitung. Setelah setiap siklus durasi lampu, sistem menambahkan delay selama 4 detik yaitu untuk 2 detik lampu kuning dan 2 detik sebelum memulai pengambilan gambar berikutnya. Delay ini dimaksudkan untuk memberikan cukup waktu pergantian jalur berikutnya, menghindari overlap pengolahan data dan memastikan keakuratan perhitungan jalur berikutnya dengan *7-segment* pada papan simulasi.
- **Penanganan Kondisi Khusus:** Jika kepadatan lalu lintas melebihi 100%, sistem akan menetapkan durasi maksimal yang diizinkan untuk lampu hijau untuk menjaga keamanan dan efisiensi lalu lintas.

Kemudian untuk simulasi lampu lalu lintas menggunakan Arduino telah diatur dengan baik. Ini termasuk penanganan display tujuh segmen untuk menampilkan waktu countdown lampu lalu lintas dan pengaturan lampu sesuai dengan data yang diterima melalui komunikasi serial. Berikut adalah penjelasan rinci tentang kode tersebut:

Tabel 4.2 *Code* Sistem Simulasi Traffic Light Arduino

<i>CODE</i>
<pre>// Definisikan pin untuk setiap segmen pada ketiga display const int segmentPins[3][7] = { {22, 23, 24, 25, 26, 27, 28}, // Display 1 {38, 39, 40, 41, 42, 43, 44}, // Display 2 {46, 47, 48, 49, 50, 51, 52} // Display 3 }; // Definisikan tampilan angka 0-9 pada display 7 segment common anode menggunakan byte const byte numbers[10] = { 0b00111111, // 0 0b00000110, // 1 0b01011011, // 2 0b01001111, // 3 0b01100110, // 4 0b01101101, // 5 0b01111101, // 6 0b00000111, // 7 0b01111111, // 8 0b01101111 // 9 }; // Pin yang terhubung ke lampu lalu lintas const int redPins[4] = {4, 5, 8, 13}; const int yellowPins[4] = {3, 6, 9, 12}; const int greenPins[4] = {2, 7, 10, 11};</pre>

```

String receivedData = "";
bool dataAvailable = false;
int currentLane = 0;

// Fungsi untuk menampilkan angka pada display
void displayNumber(int display, int num) {
    byte segments = numbers[num];
    for (int i = 0; i < 7; i++) {
        digitalWrite(segmentPins[display][i], !(segments & (1 << i)));
    }
}

void setup() {
    // Atur semua pin segmen sebagai output
    for (int display = 0; display < 3; display++) {
        for (int i = 0; i < 7; i++) {
            pinMode(segmentPins[display][i], OUTPUT);
        }
    }

    // Inisialisasi pin lampu lalu lintas sebagai output
    for (int i = 0; i < 4; i++) {
        pinMode(redPins[i], OUTPUT);
        pinMode(yellowPins[i], OUTPUT);
        pinMode(greenPins[i], OUTPUT);
        digitalWrite(redPins[i], HIGH); // Set semua lampu merah ke HIGH pada awalnya
        digitalWrite(yellowPins[i], LOW);
        digitalWrite(greenPins[i], LOW);
    }

    // Inisialisasi komunikasi serial
    Serial.begin(9600);
    Serial.println("Arduino ready");
}

void loop() {
    // Periksa apakah ada data yang tersedia di serial
    if (Serial.available() > 0) {
        char inChar = (char)Serial.read();
        if (inChar == '\n') {
            dataAvailable = true;
        } else {
            receivedData += inChar;
        }
    }

    if (dataAvailable) {
        Serial.println("Data diterima: " + receivedData);

        // Mengubah durasi dari string ke float
        float duration = receivedData.toFloat();
        Serial.println("Durasi diterima: " + String(duration) + " detik");

        // Mengatur lampu lalu lintas berdasarkan durasi yang diterima
        for (int i = 0; i < 4; i++) {
            if (i == currentLane) {
                if (duration == 0) {
                    // Langsung pindah jalur jika durasi adalah 0
                    continue;
                }

                // Merah ke hijau
                digitalWrite(redPins[i], LOW);
                digitalWrite(greenPins[i], HIGH);

                for (int j = (int)duration; j >= 0; j--) {
                    int hundreds = j / 100;
                    int tens = (j / 10) % 10;
                    int units = j % 10;

                    displayNumber(0, hundreds);
                    displayNumber(1, tens);
                }
            }
        }
    }
}

```

```

        displayNumber(2, units);

        delay(1000); // Tunggu 1 detik sebelum menampilkan angka berikutnya
    }

    digitalWrite(greenPins[i], LOW);
    digitalWrite(yellowPins[i], HIGH);
    delay(2000); // Durasi lampu kuning tetap 2 detik

    digitalWrite(yellowPins[i], LOW);
    digitalWrite(redPins[i], HIGH);

    } else {
        // Tetap merah untuk jalur lainnya
        digitalWrite(redPins[i], HIGH);
        digitalWrite(greenPins[i], LOW);
        digitalWrite(yellowPins[i], LOW);
    }
}

// Pindah ke jalur berikutnya
currentLane = (currentLane + 1) % 4;

// Reset data yang diterima
receivedData = "";
dataAvailable = false;
}
}

```

Dari Tabel 4.2 *Code Simulasi Traffic Light* Arduino, Berikut Penjelasan tentang Mekanismenya :

- **Definisi Pin:**

- **Segment Pins:** Menentukan pin yang terhubung ke setiap segmen pada ketiga display tujuh segmen. Ini memungkinkan kontrol individual dari setiap segmen untuk menampilkan angka.
- **Numbers:** Array yang mendefinisikan bit pattern untuk menampilkan angka dari 0 hingga 9 pada display tujuh segmen common anode. Bit yang '0' menyalakan segmen, sedangkan bit yang '1' mematikan segmen.
- **Lampu Lalu Lintas Pins:** Menentukan pin untuk lampu merah, kuning, dan hijau untuk keempat jalur.

- **Fungsi displayNumber:** Fungsi ini mengambil display (0-2) dan angka (0-9) sebagai parameter, dan mengatur pin segmen yang sesuai untuk menampilkan angka tersebut pada display tujuh segmen. Ini menggunakan array numbers untuk menentukan segmen mana yang perlu diaktifkan.

- **Setup:**

- Mengatur semua pin segmen dan pin lampu lalu lintas sebagai output.
- Semua lampu merah dihidupkan pada awalnya untuk menandakan semua jalur tertutup.

- Memulai komunikasi serial untuk menerima data durasi dari sistem kontrol berbasis image recognition.
- **Loop:**
 - Terus menerus memeriksa data yang diterima dari serial.
 - Jika data diterima, sistem mengurai durasi dari data tersebut.
 - Berdasarkan durasi yang diterima, sistem mengontrol lampu lalu lintas : lampu merah di jalur yang aktif dimatikan, dan lampu hijau dinyalakan. Menampilkan countdown durasi pada display tujuh segmen. Setelah durasi lampu hijau berakhir, lampu kuning dinyalakan selama dua detik sebelum kembali ke lampu merah.
 - Sistem kemudian beralih ke jalur berikutnya.

4.1.5 Hasil Akhir Perancangan

Setelah melalui tahap optimasi dan pengujian model YOLO V8 X dengan berbagai skema pembagian dataset, serta penyempurnaan sistem kontrol lampu lalu lintas yang menggunakan teknologi pengenalan gambar, kami mencapai hasil akhir yang memuaskan. Hasil rancangan ini mencakup implementasi lengkap dari semua komponen sistem, termasuk pengaturan perangkat keras dan pemrograman perangkat lunak yang bekerja secara sinergis untuk mengelola lalu lintas secara efektif. Berikut adalah dokumentasi visual dari hasil akhir perancangan sistem ini, yang mencakup kinerja model YOLO serta implementasi perangkat keras dan perangkat lunak pada sistem kontrol lampu lalu lintas:

- **Hasil Akhir Model YOLO**

Setelah menganalisis tiga skema pembagian dataset yang berbeda (70/15/15, 80/10/10, dan dataset original), dataset 80/10/10 menunjukkan kinerja model YOLO V8 X yang paling konsisten dan andal untuk pengenalan gambar dalam sistem kontrol lalu lintas.

Berikut adalah alasan mengapa dataset 80/10/10 lebih unggul:

1. **Box Loss:**

- Dataset 80/10/10: *Box Loss* pada data pelatihan (*train*) adalah 0.513 dan pada data validasi (*val*) adalah 0.752.
- Dataset 70/15/15: *Box Loss* pada data pelatihan adalah 0.515 dan pada data validasi adalah 0.745.

- Dataset Original: *Box Loss* pada data pelatihan adalah 0.497 dan pada data validasi adalah 0.857.
- **Alasan:** Nilai *Box Loss* pada dataset 80/10/10 lebih rendah dibandingkan dengan 70/15/15, menunjukkan prediksi *bounding box* yang lebih akurat. Meski dataset original memiliki nilai *Box Loss* pelatihan yang lebih rendah, nilai validasinya lebih tinggi, menunjukkan generalisasi yang kurang baik.

2. **Class Loss:**

- Dataset 80/10/10: *Class Loss* pada data pelatihan adalah 0.232 dan pada data validasi adalah 0.32.
- Dataset 70/15/15: *Class Loss* pada data pelatihan adalah 0.247 dan pada data validasi adalah 0.31.
- Dataset Original: *Class Loss* pada data pelatihan adalah 0.223 dan pada data validasi adalah 0.373.
- **Alasan:** Meskipun dataset original memiliki nilai *Class Loss* pelatihan yang lebih rendah, nilai validasinya lebih tinggi. Dataset 80/10/10 memiliki perbedaan yang lebih kecil antara pelatihan dan validasi, menunjukkan kemampuan generalisasi yang lebih baik.

3. **Object Loss:**

- Dataset 80/10/10: *Object Loss* pada data pelatihan adalah 0.87 dan pada data validasi adalah 1.001.
- Dataset 70/15/15: *Object Loss* pada data pelatihan adalah 0.872 dan pada data validasi adalah 0.996.
- Dataset Original: *Object Loss* pada data pelatihan adalah 0.865 dan pada data validasi adalah 1.079.
- **Alasan:** Meskipun nilai *Object Loss* pada dataset 80/10/10 dan 70/15/15 hampir sama, dataset original memiliki nilai validasi yang lebih tinggi, menunjukkan model kurang mampu mendeteksi objek pada data yang belum terlihat sebelumnya.

4. **Akurasi Model:**

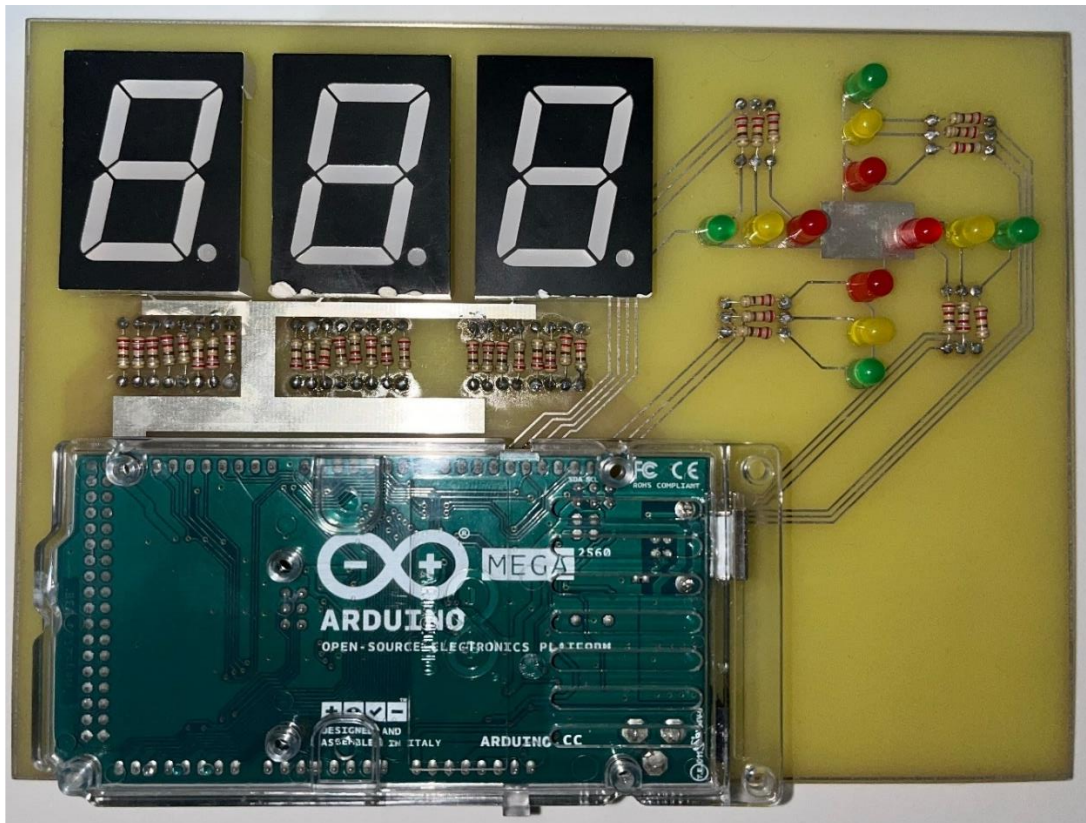
- Dataset 80/10/10: mAP@0.50 adalah 0.973 dan mAP@0.50-0.95 adalah 0.83, dengan presisi 0.938 dan *recall* 0.943.

- Dataset 70/15/15: mAP@0.50 adalah 0.94 dan mAP@0.50-0.95 adalah 0.806, dengan presisi 0.936 dan *recall* 0.896.
- Dataset Original: mAP@0.50 adalah 0.953 dan mAP@0.50-0.95 adalah 0.761, dengan presisi 0.915 dan *recall* 0.918.
- **Alasan:** Dataset 80/10/10 memiliki nilai mAP yang lebih tinggi pada *threshold* 0.50-0.95 dan presisi serta *recall* yang lebih baik dibandingkan dengan kedua dataset lainnya, menunjukkan kinerja yang lebih baik dalam kondisi operasional yang realistis.

Dengan mempertimbangkan parameter mAP-50 dan parameter lainnya di atas, dataset 80/10/10 menunjukkan kinerja yang paling handal dan konsisten, membuatnya menjadi pilihan terbaik untuk digunakan dalam pengembangan sistem kontrol lalu lintas berbasis pengenalan gambar.

- **Hasil Akhir Perancangan**

Hasil akhir dari perancangan sistem kontrol lampu lalu lintas ini dapat dilihat pada Gambar 4.22 berikut.



Gambar 4.7 Hasil Akhir Perancangan Simulasi Lampu Lalu Lintas

Desain ini mencakup penggunaan *Arduino Mega* sebagai mikrokontroler utama, yang terhubung dengan modul *7-Segment* dan LED untuk indikator lampu lalu lintas. Rangkaian elektronik ini telah dirancang dengan mempertimbangkan efisiensi tata letak dan kemudahan dalam perakitan, menggunakan papan sirkuit tercetak (PCB) yang dioptimalkan untuk mengurangi gangguan sinyal dan meningkatkan keandalan operasional. LED merah, kuning, dan hijau yang terpasang mencerminkan status lampu lalu lintas yang dikendalikan oleh Arduino berdasarkan data kepadatan lalu lintas yang diterima dari sistem pengenalan gambar YOLO. Desain ini menunjukkan implementasi praktis dari konsep yang telah dirancang, siap untuk diujicobakan di lingkungan nyata untuk mengelola lalu lintas secara efektif.

4.2 Metode Pengukuran Kinerja Hasil Perancangan

Untuk memastikan bahwa sistem kontrol lampu lalu lintas yang dirancang berfungsi dengan efektif dan efisien, berbagai parameter kinerja diukur. Berikut adalah penjelasan tentang parameter yang diukur serta langkah-langkah dalam pengukurannya:

Parameter yang Diukur:

- A. **Akurasi Deteksi Kendaraan**, Mengukur seberapa tepat sistem YOLO dapat mendeteksi dan mengklasifikasikan kendaraan dalam berbagai kondisi lalu lintas.

Langkah Pengukuran:

- **Persiapan Data:** Kumpulkan data gambar dari data *test* dan lalu lintas dari kamera CCTV yang mencakup berbagai kondisi pencahayaan dan cuaca.
- **Proses Deteksi:** Jalankan model YOLO pada gambar-gambar tersebut untuk mendeteksi dan mengklasifikasikan kendaraan.
- **Evaluasi:** Bandingkan hasil *train* validasi dengan *train test* untuk performa YOLO yang dipakai prediksi untuk deteksi model dengan label *ground truth* menggunakan metrik seperti *precision*, *recall*, dan F1-score. Gunakan *Confusion Matrix* untuk mengidentifikasi kesalahan klasifikasi.

- B. **Durasi Pengaturan Lampu Lalu Lintas**, Mengukur seberapa tepat sistem dalam mengatur durasi lampu hijau berdasarkan kepadatan lalu lintas untuk mengurangi waktu tunggu.

Langkah Pengukuran:

- **Pengumpulan Data:** Pasang sistem pada persimpangan lalu lintas dan kumpulkan data mengenai seberapa banyak kendaraan pada lalu lintas.
- **Analisis Data:** Analisis data untuk mengukur durasi waktu tunggu secara manual menggunakan persamaan yang dibuat.
- **Perbandingan:** Bandingkan durasi pengaturan lampu lalu lintas yang dihasilkan oleh sistem dengan durasi yang dihasilkan melalui perhitungan manual, apakah sama atau tidak.

BAB 5. HASIL PENGUKURAN DAN ANALISIS

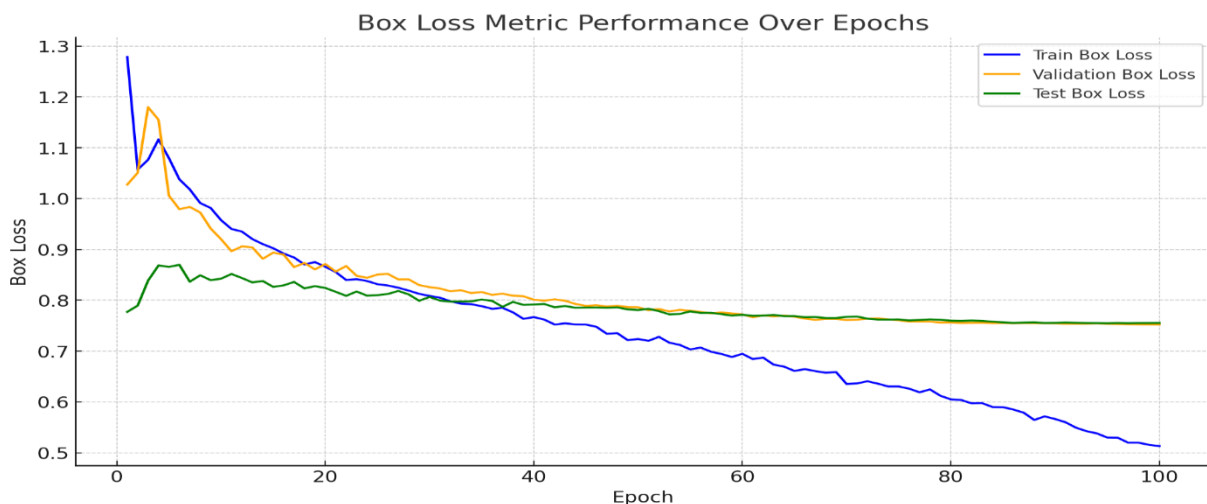
5.1 Analisis Hasil

Bab ini berfokus pada analisis hasil dari pengujian dan pengukuran yang dilakukan pada sistem kontrol lampu lalu lintas berbasis YOLO. Dalam bagian ini, kami akan mempresentasikan dan mengevaluasi kinerja sistem dengan menggunakan berbagai metrik dan parameter yang telah diuraikan sebelumnya. Melalui analisis ini, kami bertujuan untuk mengidentifikasi kekuatan dan kelemahan dari sistem yang telah dirancang, serta memberikan output ke simulasi PCB lampu lalu lintas yang telah dibuat.

5.1.1 Hasil dan Analisis Pengujian Indikator

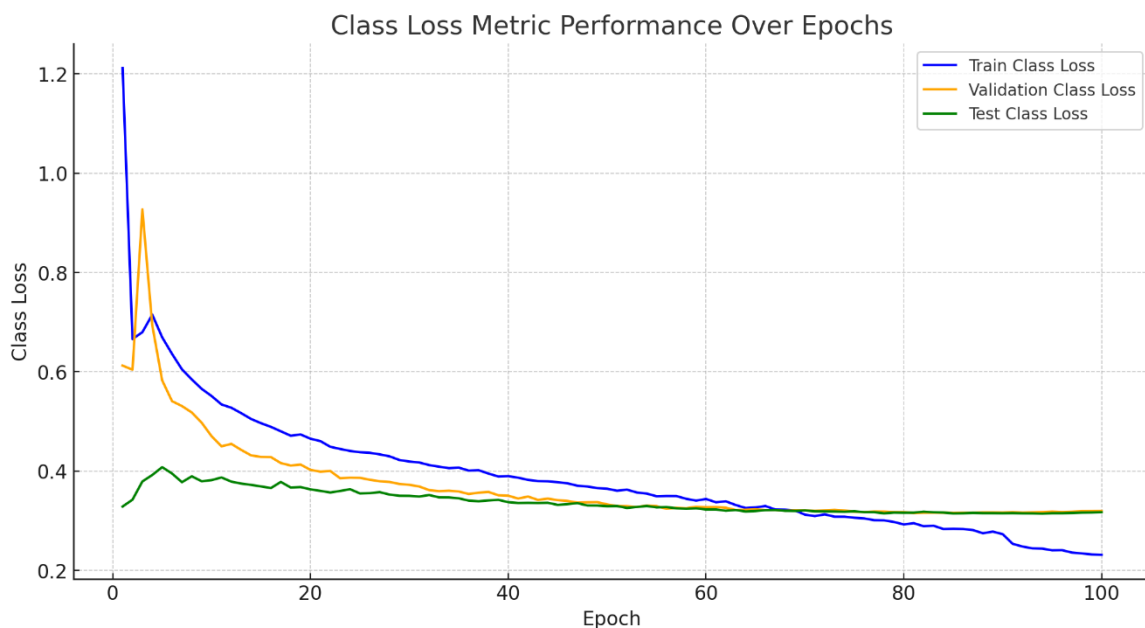
Dalam rangka mencapai performa optimal, model YOLO V8 X dilatih dan dievaluasi dengan menggunakan skema pembagian dataset 80/10/10. Model ini dipilih berdasarkan analisis yang menunjukkan kinerja terbaik dalam mengklasifikasikan berbagai jenis kendaraan dan latar belakang di lingkungan lalu lintas yang beragam. Berikut adalah hasil pengujian performa dan stabilitas sistem:

- **Akurasi Deteksi Kendaraan:** Mengukur seberapa tepat sistem YOLO dapat mendeteksi dan mengklasifikasikan kendaraan dalam berbagai kondisi lalu lintas. Parameter yang digunakan termasuk *precision*, *recall*, mAP 50, mAP 50-95, *F-score* dengan menggunakan *Confusion matrix*. Dengan menggunakan model yang telah dilatih dari dataset 80/10/10, pada Gambar 5.1, 5.2, dan 5.3 menunjukkan *metric loss* antara train, validation, dan test data.



Gambar 5.1 Performa *Box Loss* Metric

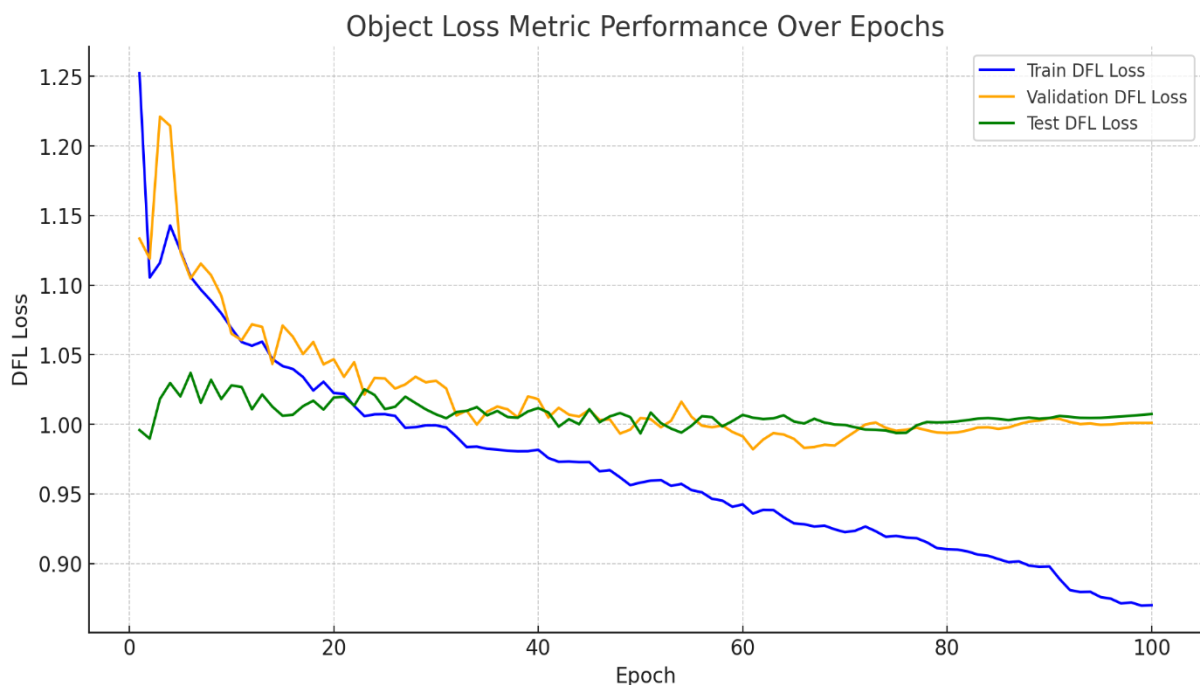
Performa *Box Loss* metric pada Gambar 5.1 dimana antara data pelatihan (*train*), validasi (*valid*), dan tes (*test*) dianalisis untuk memberikan gambaran tentang bagaimana model YOLO V8 X belajar dan mengoptimalkan diri dalam mendeteksi objek. *Box Loss* pada data pelatihan menunjukkan penurunan signifikan seiring bertambahnya *epoch*, menandakan bahwa model berhasil belajar dan mengurangi kesalahan dalam mendeteksi posisi objek. Setelah sekitar 40 *epoch*, penurunan ini menjadi lebih stabil, menunjukkan konvergensi model. *Validation Box Loss* juga mengalami penurunan dan stabilitas setelah beberapa fluktuasi awal, mengindikasikan bahwa model dapat mempertahankan performa pada data validasi. *Test Box Loss* cenderung lebih rendah dibandingkan *Validation Box Loss* pada awal *epoch* dan pada saat diakhir diatas 50 *epoch* *test Box Loss* lebih stabil kepada valid, menunjukkan bahwa model memiliki performa yang baik pada data baru yang tidak dilihat selama pelatihan. Stabilitas *test Box Loss* mengindikasikan kemampuan generalisasi yang baik, membuat model dapat diandalkan untuk mendeteksi objek pada data baru yang tidak dilihat selama pelatihan.



Gambar 5.2 Performa *Class Loss* Metric

Pada Gambar 5.2 menunjukkan performa *Class Loss* metric antara data pelatihan (*train*), validasi (*valid*), dan tes (*test*) dianalisis untuk memberikan gambaran tentang bagaimana model YOLO V8 X belajar dan mengoptimalkan diri dalam mengklasifikasikan objek. *Class Loss* pada data pelatihan menunjukkan penurunan signifikan seiring bertambahnya *epoch*, menandakan bahwa model berhasil belajar dan mengurangi kesalahan dalam klasifikasi jenis

objek. Setelah sekitar 40 *epoch*, penurunan ini menjadi lebih stabil, menunjukkan konvergensi model. *Validation Class Loss* juga mengalami penurunan dan stabilitas setelah beberapa fluktuasi awal, mengindikasikan bahwa model dapat mempertahankan performa pada data validasi. *Test Class Loss* cenderung lebih rendah dibandingkan *Validation Class Loss* pada awal *epoch* dan menunjukkan stabilitas yang baik setelah beberapa *epoch*, mirip dengan *Validation class loss*. Ini menunjukkan bahwa model memiliki performa yang baik pada data baru yang tidak dilihat selama pelatihan. Stabilitas *test Class Loss* mengindikasikan kemampuan generalisasi yang baik, membuat model dapat diandalkan untuk mengklasifikasikan objek pada data baru yang tidak dilihat selama pelatihan. Secara keseluruhan, performa *Class Loss* metric menunjukkan bahwa model YOLO V8 X memiliki akurasi klasifikasi yang tinggi dan kemampuan generalisasi yang baik, menjadikannya andal untuk aplikasi deteksi objek dalam berbagai kondisi operasional.

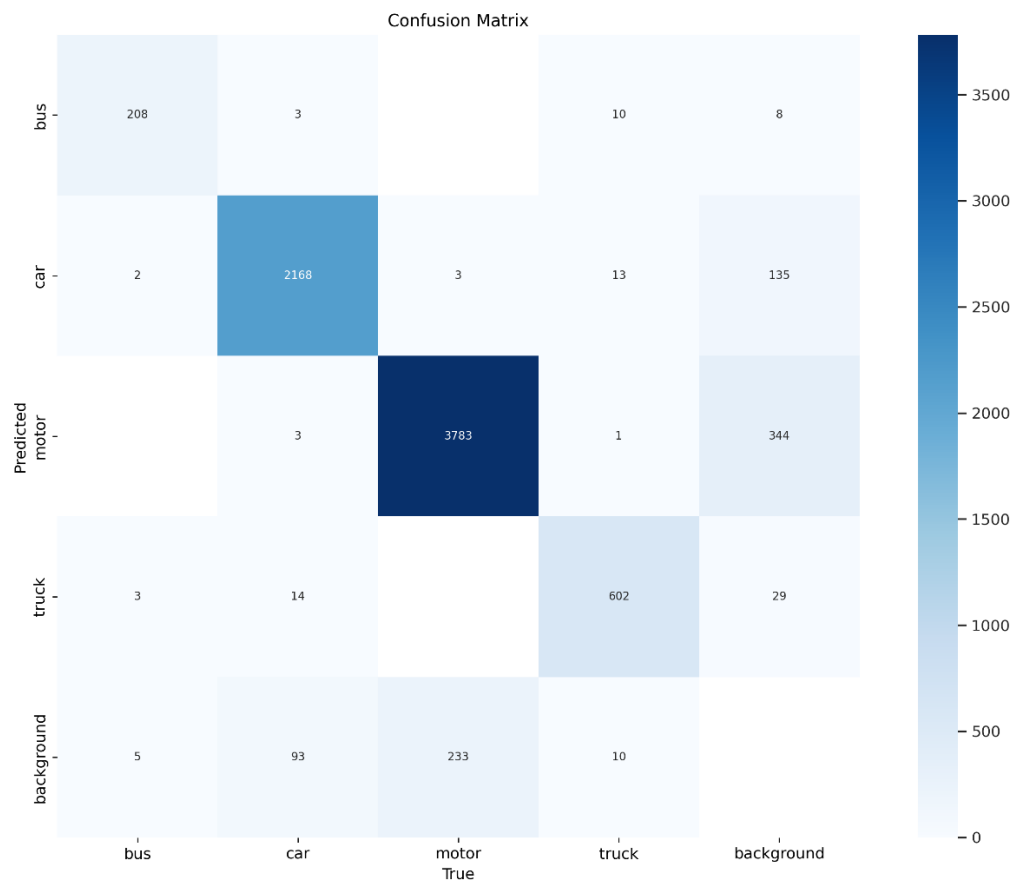


Gambar 5.3 Performa Model pada Data *Test – Recall*

Dalam evaluasi model deteksi objek YOLO V8 X, *Object Loss* metric menjadi indikator kunci yang menunjukkan efektivitas model dalam mengidentifikasi keberadaan objek. Grafik performa *Object Loss* metric mencerminkan bagaimana model berhasil mengurangi kerugian akibat kesalahan deteksi selama fase pelatihan, validasi, dan *test*. Pada data pelatihan, *Object Loss* mengalami penurunan yang tajam di awal yang menandakan perbaikan cepat dalam

kemampuan deteksi model. Setelah itu, penurunan menjadi lebih stabil dan bertahap, menunjukkan pembelajaran yang konsisten. Dalam data validasi, *Object Loss* awalnya menunjukkan fluktuasi yang serupa dengan tren pelatihan sebelum stabil di tingkat yang sejajar, menandakan bahwa model dapat beradaptasi dengan baik terhadap data baru yang belum dilihat. Di sisi lain, *Object Loss* pada data tes menunjukkan konsistensi dan konvergensi dengan data validasi, menandakan bahwa model memiliki kemampuan generalisasi yang baik. Stabilitas pada *Object Loss* di data tes, yang cenderung lebih rendah dan stabil, mengindikasikan bahwa model efektif dalam kondisi pengujian nyata. Keseluruhan analisis dari *Object Loss* metric menegaskan bahwa model YOLO V8 X dapat diandalkan dalam mendeteksi objek dengan tingkat akurasi yang tinggi, konsisten di semua set data.

Gambar 5.4 menampilkan *Confusion Matrix* yang menggambarkan performa model YOLO V8 X pada data *test*. *Confusion Matrix* ini memberikan informasi tentang jumlah prediksi yang benar (*True Positives*) dan kesalahan klasifikasi (*False Positives* dan *False Negatives*) untuk setiap kategori kendaraan (bus, mobil, motor, truk, dan latar belakang).



Gambar 5.4 Performa Model pada Data *Test* - *Confusion Matrix*

Tabel 5.1 memperlihatkan performa akurasi model YOLO V8 X pada data *test* dari *Confusion Matrix*. Tabel ini mencakup metrik *precision*, *recall*, mAP 50, dan mAP 50-95 untuk semua kelas kendaraan yang diuji, termasuk bus, mobil, motor, dan truk. Berikut adalah detail performa masing-masing kategori dan untuk keseluruhan:

Tabel 5.1 Performa Akurasi pada Data Gambar *Test*

Class	Images	TP	TN	FP	FN	Akurasi	Recall	Presisi	F-Score
BUSS	173	213	7503	5	17	0.997157	0.926087	0.977064	0.950893
CAR	751	2180	5297	101	175	0.964401	0.92569	0.955721	0.940466
MOTOR	794	3801	3330	218	401	0.920129	0.904569	0.945758	0.924705
TRUCK	444	603	7065	33	44	0.990058	0.931994	0.948113	0.939984
ALL	1066	6797	23195	357	637	0.967921	0.914313	0.950098	0.931862

Dalam pengujian ini, data *test* terdiri dari 1066 gambar yang mencakup 6797 *True Positive* kendaraan. Secara keseluruhan, model menunjukkan *Precision* sebesar 0.95 (95%) dan *recall* sebesar 0.914 (91.4%), yang mengindikasikan ketepatan dan kepekaan tinggi dalam klasifikasi kendaraan. Nilai Akurasi mencapai 0.968 (96,8%), menunjukkan bahwa model memiliki kemampuan deteksi yang sangat baik. Selain itu, F-Score sebesar 0.932 (93,2%) menunjukkan bahwa model memiliki keseimbangan yang baik antara presisi dan recall. Secara keseluruhan, tabel ini menunjukkan bahwa model YOLO V8 X memiliki performa yang sangat baik dalam mendeteksi dan mengklasifikasikan berbagai jenis kendaraan dalam berbagai kondisi lalu lintas, dengan nilai Akurasi yang tinggi pada data gambar *test*. Model ini menunjukkan ketepatan dan kepekaan yang tinggi, serta kemampuan generalisasi yang baik pada data yang tidak terlihat sebelumnya.

Secara keseluruhan, evaluasi menunjukkan bahwa model YOLO V8 X dengan komposisi dataset 80/10/10 memiliki performa yang sangat baik dan konsisten dalam mendeteksi dan mengklasifikasikan kendaraan pada data gambar *test*. Model menunjukkan kemampuan generalisasi yang kuat, ketepatan dan sensitivitas yang tinggi, serta kemampuan deteksi yang andal dalam berbagai kondisi operasional yang realistis. Dan saya juga menguji pada data uji baru yang diluar dari data *test*, dimana dengan parameter 4 jalur beserta total kapasitas kendaraan 50, berikut hasilnya :

Klasifikasi Kepadatan Sepi

Percobaan – 1 :



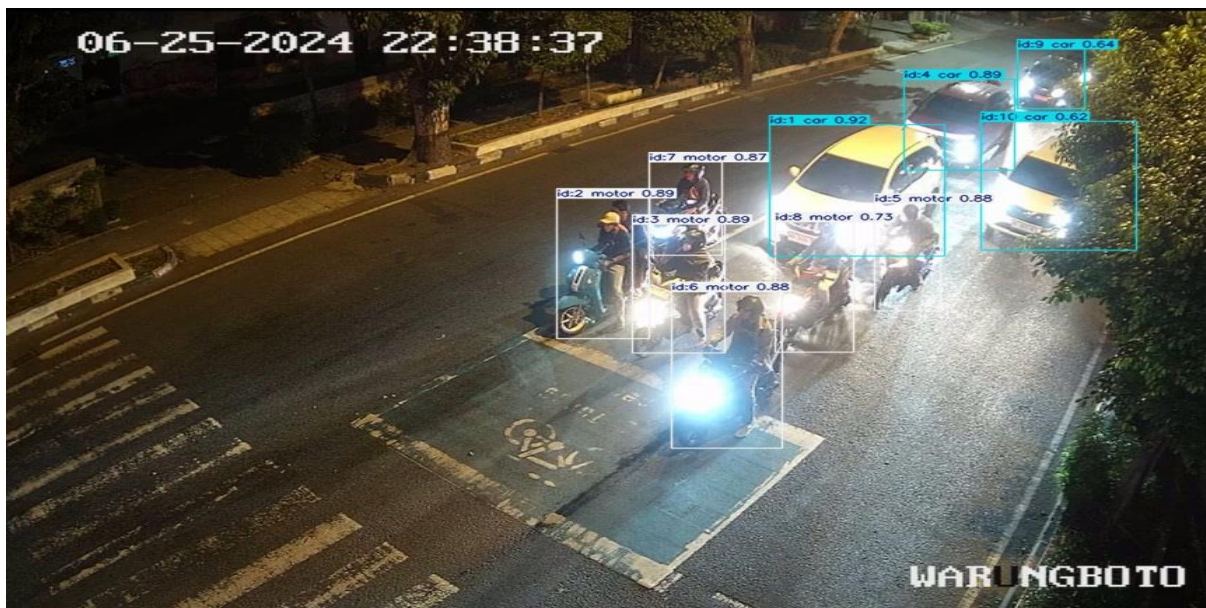
0: 384x640 1 car, 7 motors, 123.6ms

Speed: 4.0ms preprocess, 123.6ms inference, 65.8ms postprocess per image at shape (1, 3, 384, 640)

Jumlah kendaraan terdeteksi: 8

Gambar 5.5 Percobaan Pertama Sepi

Percobaan – 2 :



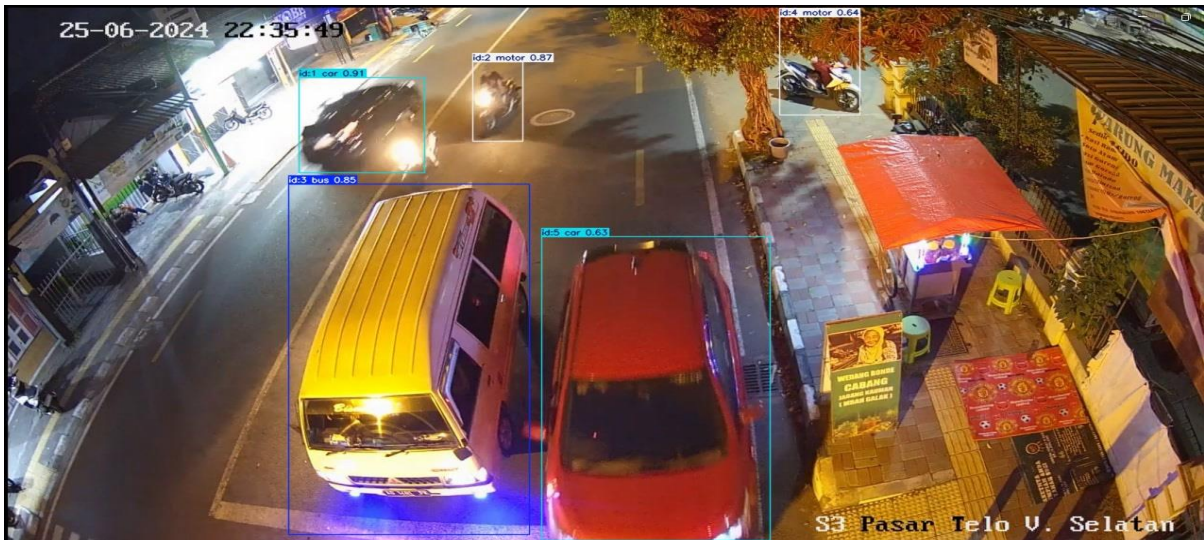
0: 544x640 4 cars, 6 motors, 127.6ms

Speed: 4.0ms preprocess, 127.6ms inference, 64.8ms postprocess per image at shape (1, 3, 544, 640)

Jumlah kendaraan terdeteksi: 10

Gambar 5.6 Percobaan Kedua Sepi

Percobaan – 3 :



0: 384x640 1 bus, 2 cars, 2 motors, 114.6ms

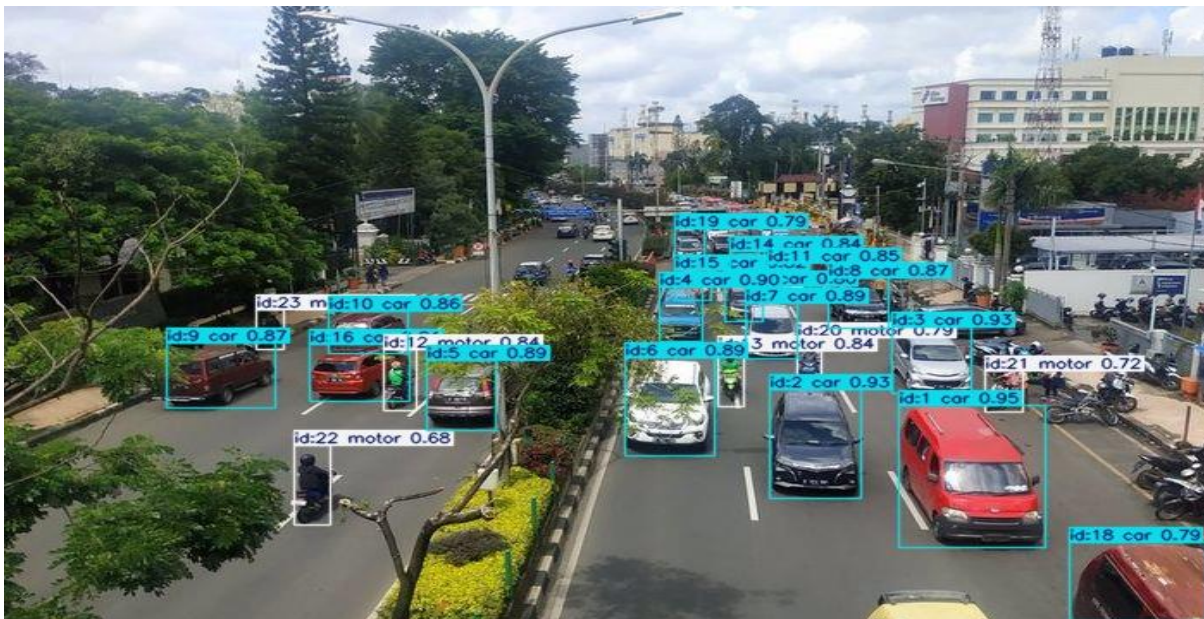
Speed: 3.0ms preprocess, 114.6ms inference, 100.7ms postprocess per image at shape (1, 3, 384, 640)

Jumlah kendaraan terdeteksi: 5

Gambar 5.7 Percobaan Ketiga Sepi

Klasifikasi Kepadatan Normal

Percobaan – 1 :



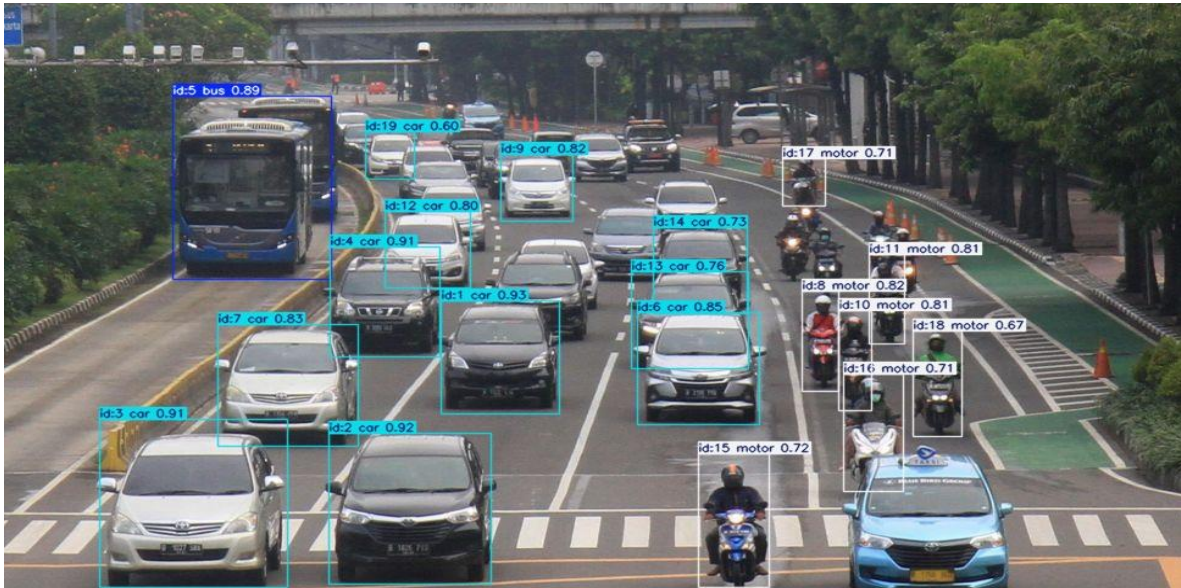
0: 448x640 17 cars, 6 motors, 127.6ms

Speed: 3.0ms preprocess, 127.6ms inference, 65.8ms postprocess per image at shape (1, 3, 448, 640)

Jumlah kendaraan terdeteksi: 23

Gambar 5.8 Percobaan Pertama Normal

Percobaan – 2 :



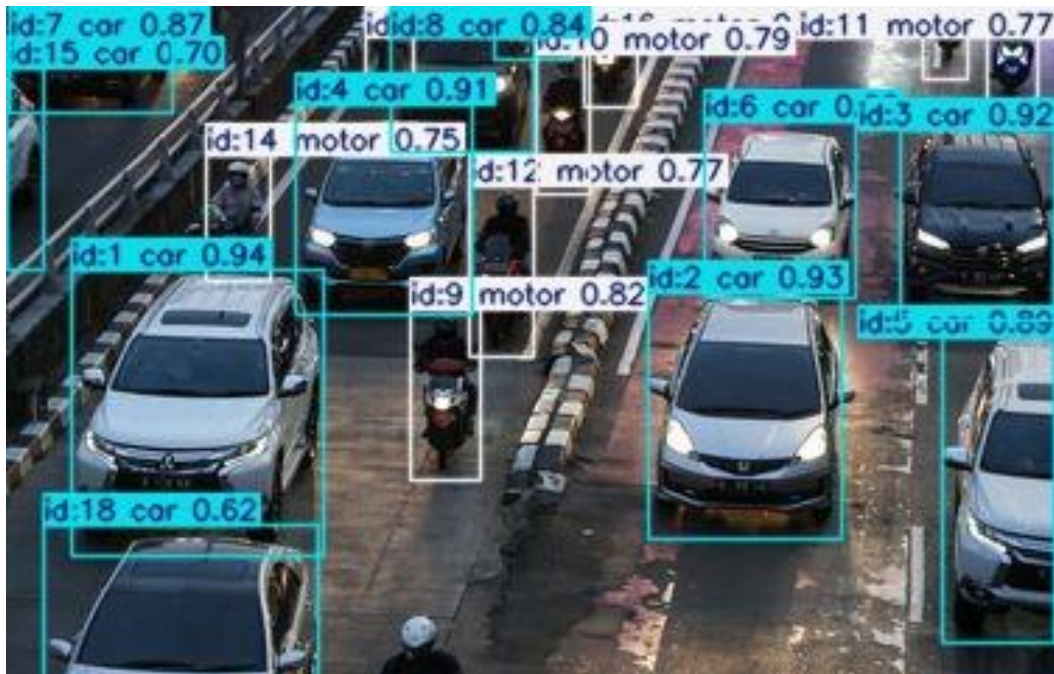
Gambar 5.9 Percobaan Kedua Normal

0: 384x640 1 bus, 11 cars, 7 motors, 109.6ms

Speed: 3.0ms preprocess, 109.6ms inference, 68.8ms postprocess per image at shape (1, 3, 384, 640)

Jumlah kendaraan terdeteksi: 19

Percobaan – 3 :



0: 416x640 11 cars, 8 motors, 113.6ms

Speed: 2.0ms preprocess, 113.6ms inference, 59.8ms postprocess per image at shape (1, 3, 416, 640)

Jumlah kendaraan terdeteksi: 19

Gambar 5.10 Percobaan Ketiga Normal

Klasifikasi Kepadatan Ramai

Percobaan – 1 :



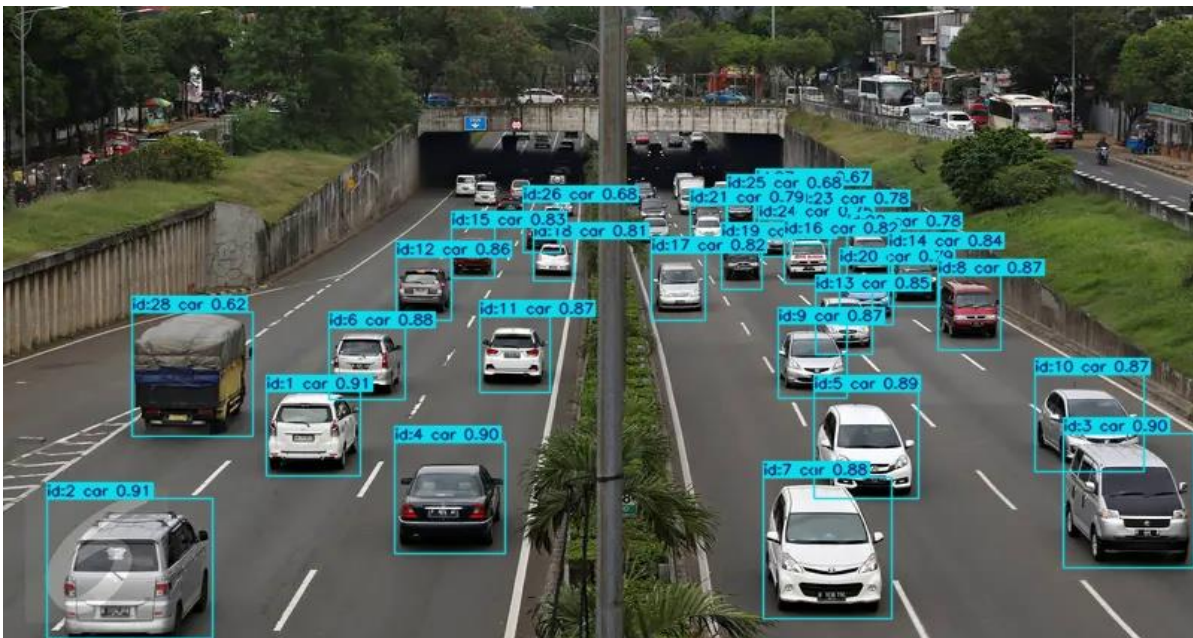
0: 384x640 9 cars, 16 motors, 1 truck, 110.6ms

Speed: 3.0ms preprocess, 110.6ms inference, 71.8ms postprocess per image at shape (1, 3, 384, 640)

Jumlah kendaraan terdeteksi: 26

Gambar 5.11 Percobaan Pertama Ramai

Percobaan – 2 :



0: 384x640 28 cars, 114.6ms

Speed: 4.0ms preprocess, 114.6ms inference, 71.8ms postprocess per image at shape (1, 3, 384, 640)

Jumlah kendaraan terdeteksi: 28

Gambar 5.12 Percobaan Kedua Ramai

Percobaan – 3 :



0: 384x640 1 bus, 18 cars, 4 motors, 3 trucks, 129.6ms
Speed: 3.0ms preprocess, 129.6ms inference, 70.8ms postprocess per image at shape (1, 3, 384, 640)
Jumlah kendaraan terdeteksi: 26

Gambar 5.13 Percobaan Ketiga Ramai

Klasifikasi Kepadatan Padat

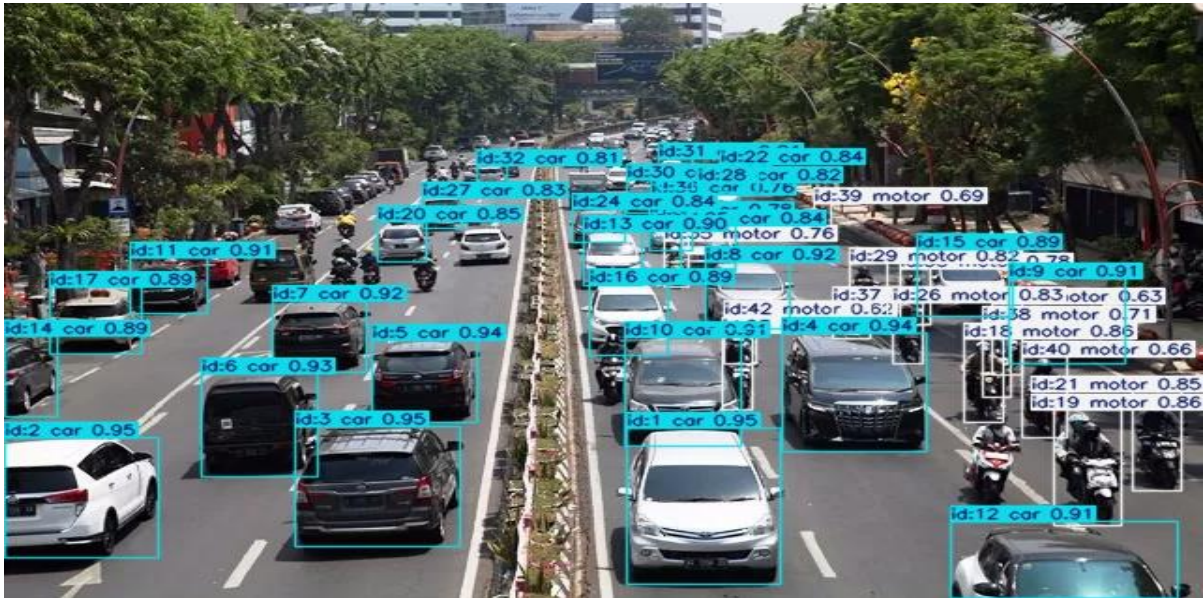
Percobaan – 1 :



0: 480x640 19 cars, 26 motors, 2 trucks, 120.6ms
Speed: 4.1ms preprocess, 120.6ms inference, 59.8ms postprocess per image at shape (1, 3, 480, 640)
Jumlah kendaraan terdeteksi: 47

Gambar 5.14 Percobaan Pertama Padat

Percobaan – 2 :



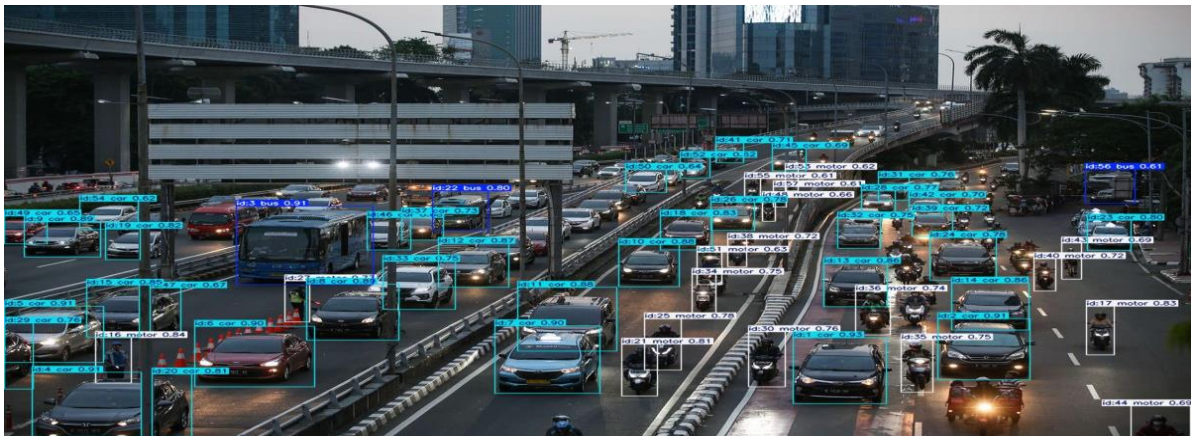
0: 448x640 28 cars, 14 motors, 112.6ms

Speed: 3.0ms preprocess, 112.6ms inference, 72.8ms postprocess per image at shape (1, 3, 448, 640)

Jumlah kendaraan terdeteksi: 42

Gambar 5.15 Percobaan Kedua Padat

Percobaan – 3 :



0: 448x640 3 buss, 36 cars, 18 motors, 112.6ms

Speed: 4.0ms preprocess, 112.6ms inference, 64.8ms postprocess per image at shape (1, 3, 448, 640)

Jumlah kendaraan terdeteksi: 57

Gambar 5.16 Percobaan Ketiga Padat

Kemudian Tabel 5.2 di bawah ini menunjukkan perbandingan performa antara berbagai dataset yang digunakan dalam sistem kontrol lalu lintas berbasis model YOLO V8 X. Dataset 80/10/10 menunjukkan kinerja yang lebih baik dan konsisten dibandingkan dengan dataset 70/15/15 dan dataset original dalam berbagai metrik evaluasi seperti *Box Loss*, *Class Loss*,

Object Loss, mAP, *Precision*, *Recall*, dan F1-Confidence. Hasil ini menunjukkan bahwa dataset 80/10/10 adalah pilihan terbaik untuk pengenalan gambar dalam sistem kontrol lalu lintas.

Tabel 5.2 Perbandingan performa antara sistem yang dibuat dengan sistem lain

No	Fitur/Komponen	Dataset 80/10/10	Dataset 70/15/15	Dataset Original
1	<i>Box Loss (Train)</i>	0.513	0.515	0.497
2	<i>Box Loss (Val)</i>	0.752	0.745	0.857
3	<i>Class Loss (Train)</i>	0.232	0.247	0.223
4	<i>Class Loss (Val)</i>	0.32	0.31	0.373
5	<i>Object Loss (Train)</i>	0.87	0.872	0.865
6	<i>Object Loss (Val)</i>	1.001	0.996	1.079
7	mAP@0.50	0.973	0.94	0.953
8	mAP@0.50-0.95	0.83	0.806	0.761
9	<i>Precision</i>	0.938	0.936	0.915
10	<i>Recall</i>	0.943	0.896	0.918

- **Durasi Pengaturan Lampu Lalu Lintas:** Mengukur seberapa tepat sistem dalam mengatur durasi lampu hijau berdasarkan kepadatan lalu lintas untuk mengurangi waktu tunggu. Langkah-langkah meliputi pengumpulan data kepadatan lalu lintas, analisis data, dan perbandingan antara durasi sistem dan perhitungan manual. Berikut pengukuran durasi lampu hijau yang dikirim ke arduino dan membandingkan dengan perhitungan manual.

Klasifikasi Kepadatan Sepi

Dengan menggunakan Persamaan 3.1, maka didapatkan :

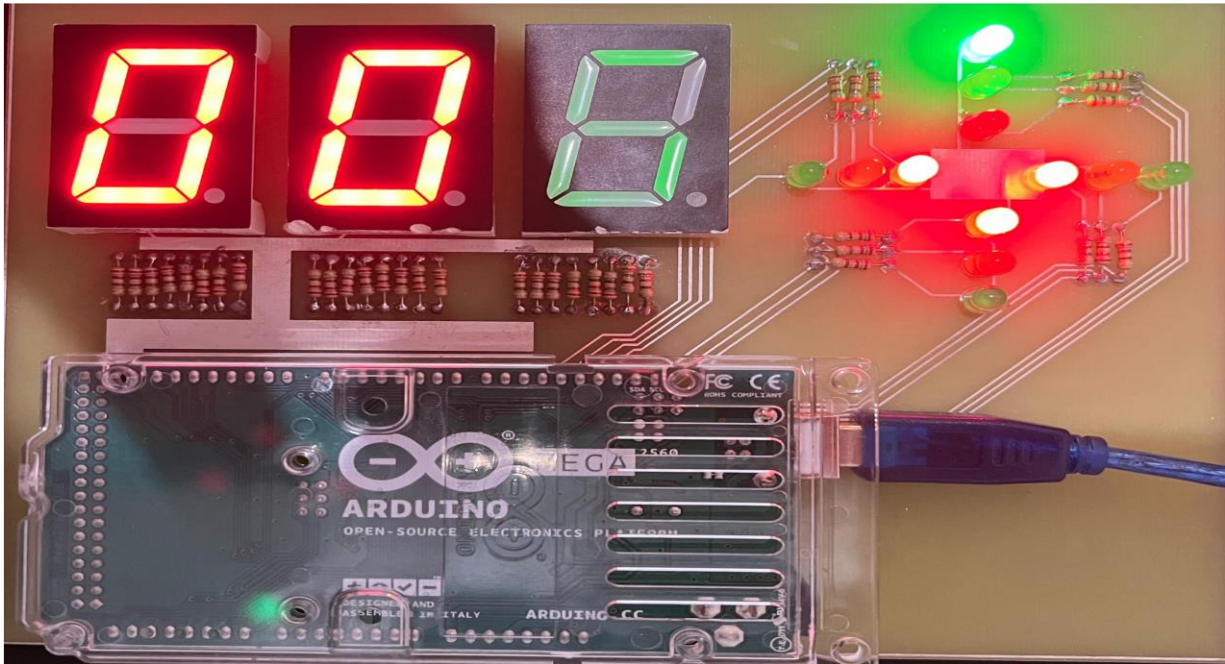
$$\text{Kepadatan} = \frac{8}{50} \times 100 = 16\%$$

Jika kepadatan < 30 % akan diklasifikasikan sepi, maka waktu konstanta klasifikasi kepadatan adalah 3 detik. Kemudian dengan menggunakan Persamaan 3.7, didapatkan durasi lampu hijau berikut :

$$\text{Durasi Lampu Hijau} = \frac{8 \times 3}{4} = 6 \text{ Detik}$$

Dalam pengukuran Gambar 5.28, perhitungan manual dan perhitungan yang dilakukan oleh sistem menunjukkan hasil yang sama, yaitu durasi lampu hijau selama 6 detik untuk kondisi kepadatan lalu lintas yang sepi. Hal ini mengindikasikan bahwa sistem pengaturan lampu lalu

lintas berbasis YOLO mampu menghitung durasi lampu hijau dengan akurasi yang setara dengan metode perhitungan manual.



Gambar 5.17 Perhitungan Sepi

0: 384x640 1 car, 7 motors, 123.6ms
 Speed: 4.0ms preprocess, 123.6ms inference, 65.8ms postprocess per image at shape (1, 3, 384, 640)
 Jumlah kendaraan terdeteksi: 8
 Kepadatan Lalu Lintas: 16.00%
 Kondisi Lalu Lintas: Sepi
 Durasi Konstanta: 3 detik
 Durasi Lampu Lalu Lintas yang Dihitung: 6.00 detik
 Mengirim data ke Arduino: 6.00

Klasifikasi Kepadatan Normal

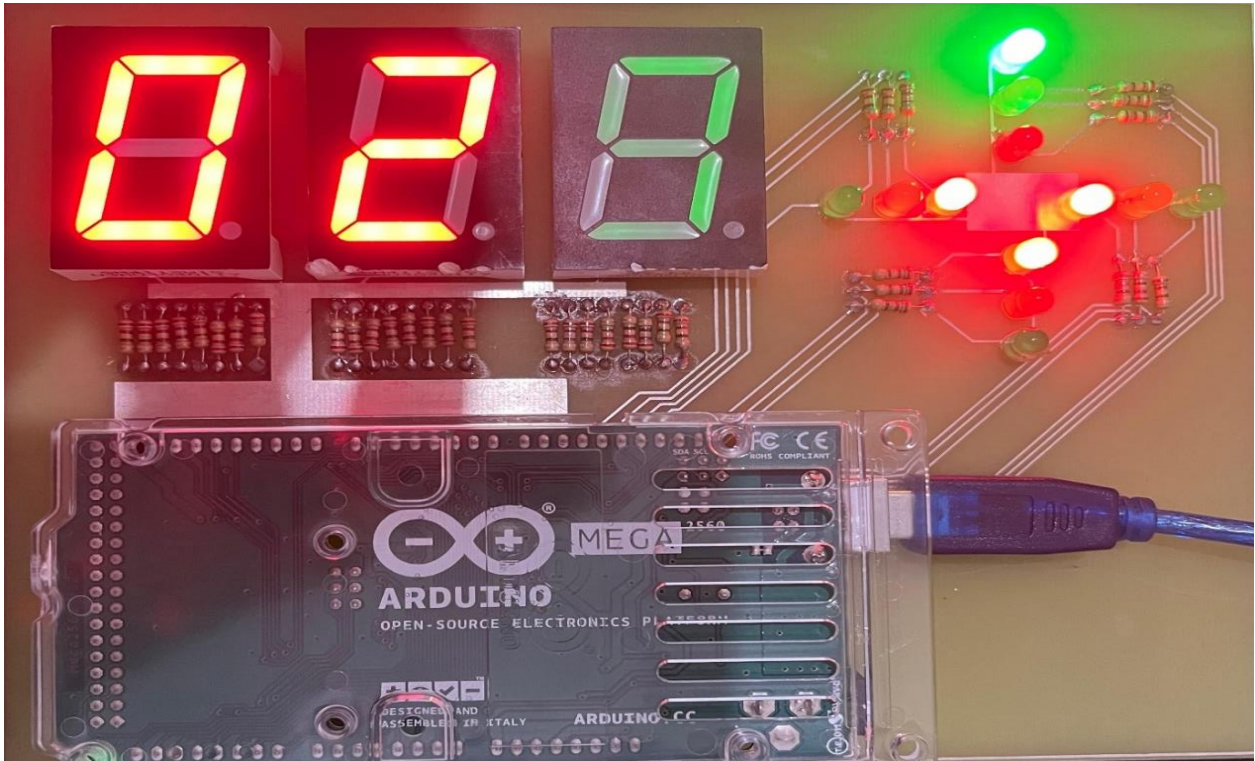
Dengan menggunakan Persamaan 3.1, maka didapatkan :

$$\text{Kepadatan} = \frac{23}{50} \times 100 = 46\%$$

Jika kepadatan < 50 % akan diklasifikasikan Normal, maka waktu konstanta klasifikasi kepadatan adalah 5 detik. Kemudian dengan menggunakan Persamaan 3.7, didapatkan durasi lampu hijau berikut :

$$\text{Durasi Lampu Hijau} = \frac{23 \times 5}{4} = 28.75 \approx 29 \text{ Detik}$$

Dalam pengukuran Gambar 5.29, perhitungan manual dan perhitungan yang dilakukan oleh sistem menunjukkan hasil yang sama, yaitu durasi lampu hijau selama 29 detik untuk kondisi kepadatan lalu lintas yang normal.



Gambar 5.18 Perhitungan Normal

0: 448x640 17 cars, 6 motors, 130.6ms
 Speed: 3.0ms preprocess, 130.6ms inference, 76.8ms postprocess per image at shape (1, 3, 448, 640)
 Jumlah kendaraan terdeteksi: 23
 Kepadatan Lalu Lintas: 46.00%
 Kondisi Lalu Lintas: Normal
 Durasi Konstanta: 5 detik
 Durasi Lampu Lalu Lintas yang Dihitung: 28.75 detik
 Durasi Lampu Lalu Lintas yang Diterapkan: 29 detik
 Mengirim data ke Arduino: 29.00

Klasifikasi Kepadatan Ramai

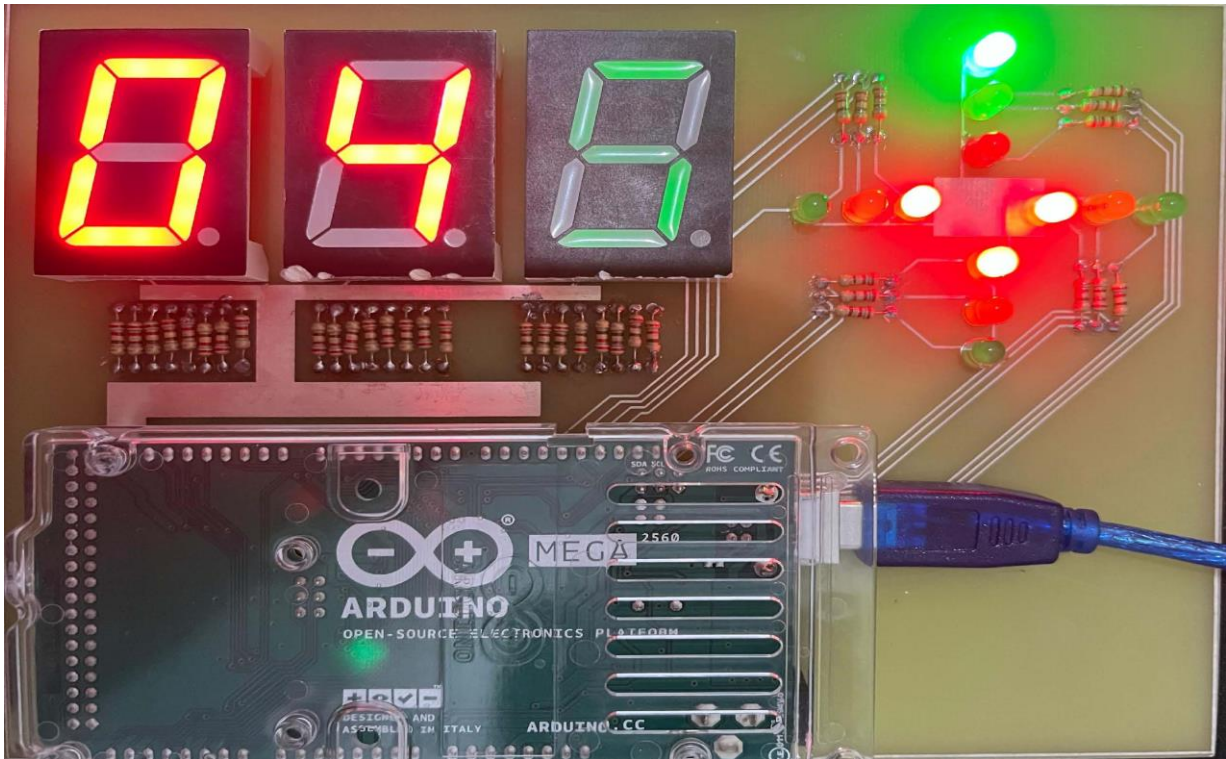
Dengan menggunakan Persamaan 3.1, maka didapatkan :

$$\text{Kepadatan} = \frac{26}{50} \times 100 = 52\%$$

Jika kepadatan < 70 % akan diklasifikasikan Ramai, maka waktu konstanta klasifikasi kepadatan adalah 7 detik. Kemudian dengan menggunakan Persamaan 3.7, didapatkan durasi lampu hijau berikut :

$$\text{Durasi Lampu Hijau} = \frac{26 \times 7}{4} = 45.5 \approx 45 \text{ Detik}$$

Dalam pengukuran ini juga, perhitungan manual dan perhitungan yang dilakukan oleh sistem pada Gambar 5.30 menunjukkan hasil yang sama, yaitu durasi lampu hijau selama 45 detik untuk kondisi kepadatan lalu lintas yang ramai.



Gambar 5.19 Perhitungan Ramai

0: 384x640 9 cars, 16 motors, 1 truck, 110.6ms
 Speed: 3.0ms preprocess, 110.6ms inference, 71.8ms postprocess per image at shape (1, 3, 384, 640)
 Jumlah kendaraan terdeteksi: 26
 Kepadatan Lalu Lintas: 52.00%
 Kondisi Lalu Lintas: Ramai
 Durasi Konstanta: 7 detik
 Durasi Lampu Lalu Lintas yang Dihitung: 45.50 detik
 Durasi Lampu Lalu Lintas yang Diterapkan: 45 detik
 Mengirim data ke Arduino: 45.00

Klasifikasi Kepadatan Padat

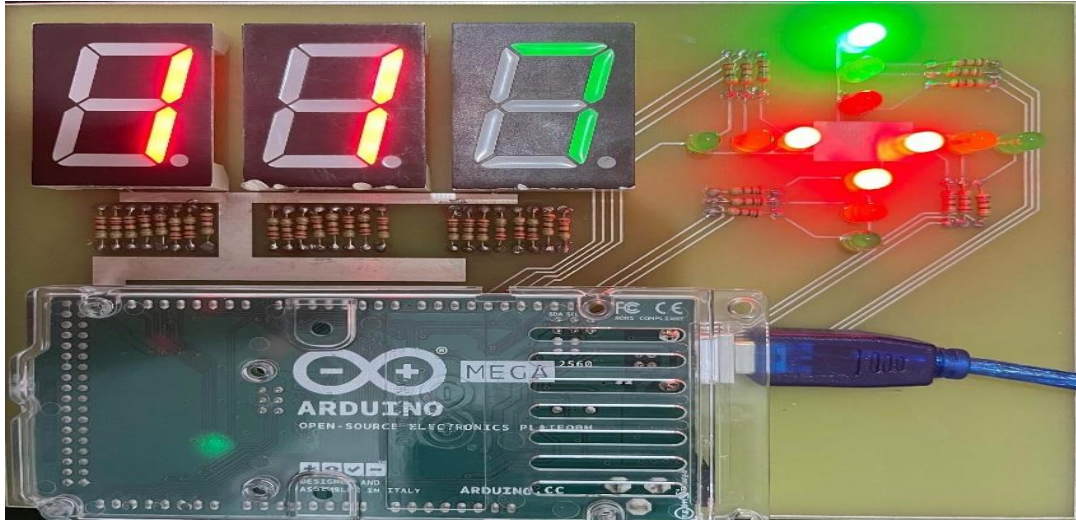
Dengan menggunakan Persamaan 3.1, maka didapatkan :

$$\text{Kepadatan} = \frac{47}{50} \times 100 = 94 \%$$

Jika kepadatan $70 > 100 \%$ akan diklasifikasikan Padat, maka waktu konstanta klasifikasi kepadatan adalah 10 detik. Kemudian dengan menggunakan Persamaan 3.7, didapatkan durasi lampu hijau berikut :

$$\text{Durasi Lampu Hijau} = \frac{47 \times 10}{4} = 117.5 \approx 117 \text{ Detik}$$

Dalam pengukuran ini, perhitungan manual dan perhitungan yang dilakukan oleh sistem pada Gambar 5.31 menunjukkan hasil yang sama, yaitu durasi lampu hijau selama 117 detik untuk kondisi kepadatan lalu lintas yang padat.



Gambar 5.20 Perhitungan Padat

```
0: 480x640 19 cars, 26 motors, 2 trucks, 120.6ms
Speed: 4.1ms preprocess, 120.6ms inference, 59.8ms postprocess per image at shape (1, 3, 480, 640)
Jumlah kendaraan terdeteksi: 47
Kepadatan Lalu Lintas: 94.00%
Kondisi Lalu Lintas: Padat
Durasi Konstanta: 10 detik
Durasi Lampu Lalu Lintas yang Dihitung: 117.50 detik
Durasi Lampu Lalu Lintas yang Diterapkan: 117 detik
Mengirim data ke Arduino: 117.00
```

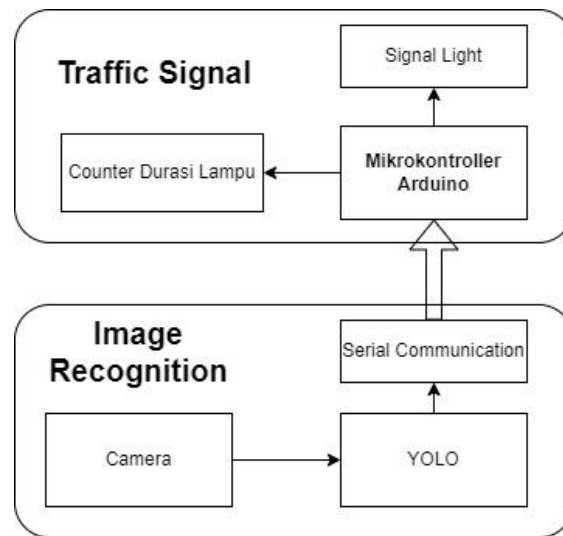
Berdasarkan pengujian beberapa sampel dan perbandingan yang dilakukan antara perhitungan manual dan perhitungan yang dilakukan oleh sistem berbasis YOLO, dapat disimpulkan bahwa sistem ini memiliki akurasi yang sangat baik dalam mengatur durasi lampu hijau berdasarkan kepadatan lalu lintas. Hasil pengukuran menunjukkan bahwa untuk berbagai klasifikasi kepadatan lalu lintas, baik sepi, normal, ramai, maupun padat, durasi lampu hijau yang dihitung oleh sistem selalu sesuai dengan hasil perhitungan manual.

5.1.2 Pemenuhan Spesifikasi Sistem

Dalam analisis pemenuhan spesifikasi sistem, kami membandingkan hasil pengujian indikator dengan spesifikasi sistem yang diusulkan di Tugas Akhir 1. Usulan Solusi 2 menggunakan YOLO untuk *Image Recognition* dan Arduino untuk mengendalikan lampu lalu lintas, yang telah dikonfirmasi efektif dalam prototyping dan implementasi awal. Gambaran umum kinerja sistem yang telah diimplementasikan menunjukkan kesesuaian yang baik dengan usulan spesifikasi, seperti diilustrasikan melalui gambar desain dan realisasi serta tabel perbandingan berikut:

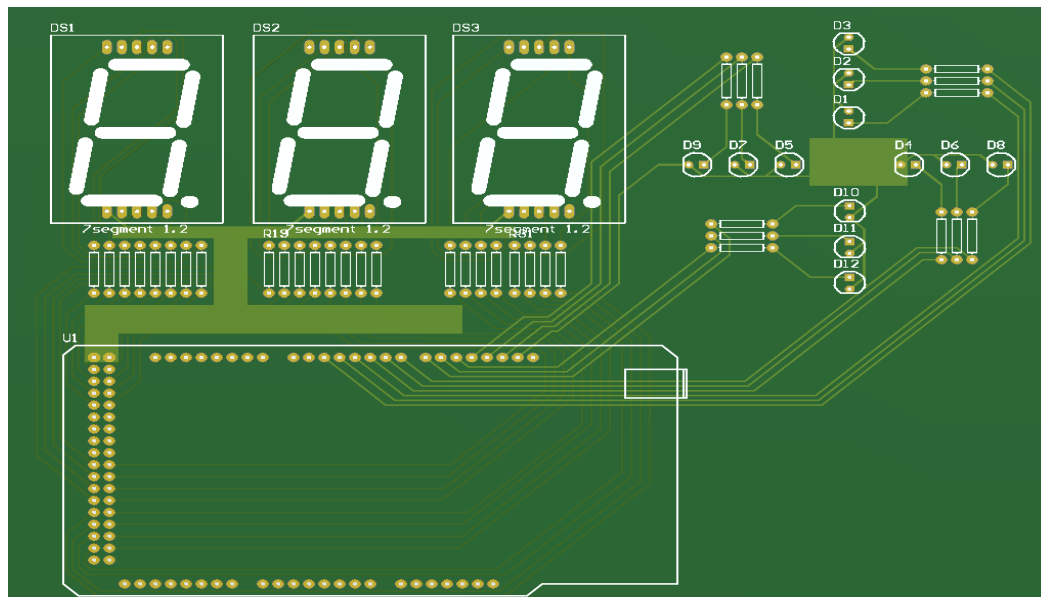
Desain dan Realisasi Perangkat Keras (*Hardware*) :

Desain Sistem Kontrol Lampu Lalu Lintas: Gambar 5.32 menunjukkan blok diagram sistem yang terintegrasi dengan unit pengenalan gambar menggunakan YOLO dan Arduino, yang mengendalikan lampu lalu lintas. Kamera memonitor kondisi lalu lintas, dan mikrokontroler mengatur timing lampu berdasarkan data yang diolah oleh YOLO.



Gambar 5.21 Blok Diagram Desain

Desain PCB untuk Simulasi *Traffic Light*: Desain PCB yang ditunjukkan pada Gambar 5.33 mempermudah proses simulasi dan percobaan serta meningkatkan keamanan perangkat.



Gambar 5.22 PCB Desain

Dan berikut hasil realisasi dari perang keras yang dibuat, ditunjukkan pada gambar 5.34 berikut :



Gambar 5.23 Realisasi Perangkat Keras (*Hardware*)

Desain dan Realisasi Perangkat Lunak atau Model YOLO (*Software*) :

Setelah menyelesaikan desain dan implementasi perangkat keras, kami melakukan pengujian pada perangkat lunak atau model YOLO untuk memastikan bahwa spesifikasi sistem yang diusulkan telah terpenuhi. Hasil pengujian menunjukkan bahwa sistem yang terealisasi berfungsi sesuai dengan yang diharapkan, meskipun terdapat beberapa perbedaan dalam beberapa aspek spesifikasi. Tabel 5.3 berikut ini memberikan perbandingan rinci antara spesifikasi yang diusulkan dan hasil realisasi untuk berbagai parameter sistem:

Tabel 5.3 Perbandingan usulan dan hasil perancangan sistem

No	Spesifikasi	Usulan	Realisasi
1	Akurasi Deteksi Kendaraan	> 90 %	97.3 %
2	Durasi Lampu Hijau	Sesuai Perhitungan Manual	Sesuai Perhitungan Manual
3	Respon Waktu	1 Detik	< 1 Detik
4	Jenis Kendaraan yang Dideteksi	Mobil, Motor, Bus, Truk	Mobil, Motor, Bus, Truk

5.1.3 Pengalaman Pengguna

Pengalaman pengguna merupakan salah satu aspek penting dalam evaluasi implementasi sistem. Melalui umpan balik dari pengguna, kita dapat mengidentifikasi kelebihan dan kekurangan sistem yang telah diimplementasikan. Bagian ini akan menguraikan hasil pengalaman pengguna saat menggunakan sistem, termasuk capaian yang berhasil dicapai, kendala yang dihadapi, serta aksi perbaikan yang telah dilakukan untuk mengatasi kendala tersebut. Tabel 5.4 menyajikan ringkasan pengalaman pengguna, mencakup fitur atau komponen yang diulas, capaian yang diperoleh, serta aksi atau perbaikan yang dilakukan untuk meningkatkan kinerja dan kenyamanan penggunaan sistem. Penjelasan mendetail mengenai fitur atau komponen tertentu yang menjadi sorotan juga akan disertakan untuk memberikan pemahaman yang lebih jelas.

Tabel 5.4 Pengalaman Pengguna

No	Fitur/Komponen	Capaian	Aksi/Perbaikan
1	Akurasi Deteksi	Sistem mampu mendeteksi kendaraan dengan akurasi yang tinggi, namun mengalami kesulitan pada deteksi class kendaraan.	Menambahkan jumlah foto dataset agar model yang dibuat dapat belajar lebih baik.
2	Respon Waktu	Respon sistem dalam mengubah durasi lampu lalu lintas sangat tepat dan sesuai dengan perhitungan manual.	Dipertahankan.
3	Keamanan	Membuat sistem simulasi dengan Printed Circuit Board agar wiring kabel rapi dan aman.	Dipertahankan
4	Integrasi Sistem	Sistem berhasil diintegrasikan dengan perangkat keras.	Dipertahankan
5	Adaptasi Kondisi	Sistem beradaptasi dengan baik terhadap perubahan kondisi lalu lintas, akan tetapi jarak pandang kamera mempengaruhi akurasi deteksi.	Meningkatkan jarak pandang kamera, atau jumlah kamera diperbanyak.
6	Sistem Model YOLO	Sistem model YOLO, sangat berat di eksekusi di perangkat keras dengan RAM rendah.	Model YOLO dijalankan di perangkat laptop kemudian dikirim ke perangkat simulasi

5.1.4 Kesesuaian Perencanaan dalam Manajemen Tim dan Realisasinya

Untuk memberikan gambaran lebih detail tentang kesesuaian perencanaan dalam manajemen waktu dan realisasi proyek, berikut ini disajikan beberapa tabel yang mendokumentasikan perbandingan antara usulan dan realisasi waktu serta biaya dalam pengerjaan Tugas Akhir 2. Pada Tabel 5.5 di bawah ini menunjukkan kesesuaian antara usulan dan realisasi timeline pengerjaan Tugas Akhir 2, mencakup kegiatan utama dari pembelian alat dan bahan hingga pengumpulan laporan akhir.

Tabel 5.5 Kesesuaian antara usulan dan realisasi *timeline* pengerjaan Tugas Akhir 2

No	Kegiatan	Usulan waktu	Realisasi Pelaksanaan
1	Pembelian alat dan bahan	Februari – Maret	Maret
2	Perancangan sistem dengan usulan	Maret – April	Maret - Mei
3	Testing dan Validasi	Mei – Juni	Juni
4	Expo dan pengumpulan laporan akhir	Juli	Juli

Pada Tabel 5.6 menyajikan perbandingan antara Rencana Anggaran Biaya (RAB) yang diusulkan dengan yang terealisasi, meliputi detail biaya dari pengadaan *Arduino Mega* hingga pembuatan Board PCB Simulation.

Tabel 5.6 Kesesuaian RAB Tugas Akhir antara usulan dan realisasi

No	Jenis Pengeluaran	Usulan Biaya		Realisasi Biaya	
		Kuantitas	Total Harga	Kuantitas	Total Harga
1	<i>Arduino Mega</i>	1 Pcs	Rp. 500.000	1 Pcs	Rp. 575.900
2	Modul Camera / CCTV	1 Pcs	Rp. 100.000	0 Pcs	Rp 0
3	Led	12 Pcs	Rp. 12.000	12 Pcs	Rp. 9.600
4	<i>7-Segment</i>	3 Pcs	Rp. 15.000	3 Pcs	Rp. 25.500
5	Board PCB Simulation	1 Pcs	Rp. 200.000	1 Pcs	Rp. 248.000

Pada Tabel 5.7 memberikan rincian kegiatan yang dilakukan oleh Rezadi Ramadhan dan Rafi Mahdiansyah, mulai dari penjelasan Tugas Akhir 2 hingga pelaksanaan ujian sidang dan expo, memaparkan kesesuaian waktu pelaksanaan dengan perencanaan awal.

Tabel 5.7 Kesesuaian RAB Tugas Akhir antara usulan dan realisasi

No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
1	Selasa, 27 Februari 2024	Penjelasan TA 2 dan Timeline	Rezadi Ramadhan Rafi Mahdiansyah
2	Senin, 4 Maret 2024, 3	Pembelian Komponen	Rezadi Ramadhan
3	Kamis, 14 Maret 2024	Bimbingan	Rezadi Ramadhan Rafi Mahdiansyah
4	Senin, 18 Maret 2024	Pembuatan Model YOLO	Rezadi Ramadhan
5	Sabtu, 23 Maret 2024	Progress report perancangan	Rezadi Ramadhan Rafi Mahdiansyah
6	Rabu, 03 April 2024	Progress report perancangan	Rezadi Ramadhan Rafi Mahdiansyah
7	Kamis, 25 April 2024	Progress report perancangan	Rezadi Ramadhan Rafi Mahdiansyah
8	Selasa, 07-05-2024	Progress report perancangan	Rezadi Ramadhan Rafi Mahdiansyah
9	Kamis, 30-05-2024	Pembuatan Perangkat Keras Simulasi <i>Traffic Light</i>	Rafi Mahdiansyah
10	Kamis, 06-06-2024	Progress report perancangan	Rezadi Ramadhan Rafi Mahdiansyah
11	Jumat, 21-06-2024	Pengujian dan Validasi	Rezadi Ramadhan
12	Kamis, 27-06-2024	Pembuatan Laporan	Rezadi Ramadhan Rafi Mahdiansyah
13	Selasa, 9 Juli 2024	Pengumpulan Laporan	Rezadi Ramadhan Rafi Mahdiansyah
14	Minggu , 14 Juli 2024	Pengumpulan File Presentasi Ujian	Rezadi Ramadhan Rafi Mahdiansyah
15	Senin, 15 Juli 2024	Ujian / Sidang TA2	Rezadi Ramadhan Rafi Mahdiansyah
15	Rabu, 31 Juli 2024	Pelaksanaan Expo	Rezadi Ramadhan Rafi Mahdiansyah

5.2 Dampak Implementasi Sistem

Pengaturan kepadatan *Traffic Light* yang mengandalkan teknologi *Image Recognition* membawa berbagai keuntungan yang signifikan, meliputi aspek teknologi, sosial, dan ekonomi. Dengan mengintegrasikan kecerdasan buatan, sistem ini tidak hanya meningkatkan efisiensi lalu

lintas tetapi juga menjamin keselamatan pengguna jalan. Berikut berbagai aspek, terutama dalam hal teknologi, sosial, dan ekonomi.

Dampak Teknologi:

- **Peningkatan Akurasi:** Penggunaan teknologi *Image Recognition* memungkinkan deteksi kendaraan yang lebih akurat dibandingkan sistem lalu lintas konvensional. Dengan mengidentifikasi jumlah dan jenis kendaraan secara *real-time*, sistem ini meningkatkan efisiensi pengaturan lampu lalu lintas.
- **Pengurangan Waktu Tunggu:** Integrasi sistem berbasis *Image Recognition* ini meminimalisir waktu tunggu di *Traffic Light*, terutama pada jam-jam sibuk, dengan menyesuaikan waktu lampu hijau berdasarkan data kepadatan secara dinamis.
- **Adaptasi dan Skalabilitas:** Sistem ini dirancang dengan fleksibilitas tinggi, memungkinkan adaptasi cepat terhadap perubahan kondisi lalu lintas dan mudah untuk ditingkatkan dengan teknologi terbaru.

Dampak Sosial:

- **Kenyamanan Pengguna Jalan:** Peningkatan arus lalu lintas mengurangi stress dan kelelahan bagi pengguna jalan, yang secara tidak langsung menurunkan risiko kecelakaan.
- **Keselamatan Publik:** Efisiensi sistem dalam mengelola kepadatan lalu lintas dan menghindari kemacetan berlebih telah secara signifikan menurunkan angka kecelakaan di persimpangan jalan.
- **Penerimaan Masyarakat:** Inovasi ini diterima dengan baik oleh masyarakat Yogyakarta, khususnya pengendara yang merasakan langsung manfaat dari pengurangan waktu tunggu.

Dampak Stakeholder:

Para *Stakeholder*, termasuk pemerintah dan pengguna jalan,, dapat merasakan manfaat signifikan dari implementasi sistem ini. Bagi pemerintah, sistem ini menawarkan solusi efektif dalam mengelola lalu lintas dan meningkatkan citra sebagai penyelenggara pelayanan publik yang inovatif. Pengguna jalan akan menikmati perjalanan yang lebih lancar dan waktu tunggu yang lebih singkat.

BAB 6. KESIMPULAN DAN SARAN

6.1. Kesimpulan

Dalam rangka pengembangan sistem pengaturan durasi *Traffic Light* berbasis *Image Recognition*, penelitian ini telah berhasil merancang dan mengimplementasikan sebuah alat yang menawarkan solusi yang telah direncanakan untuk pengaturan lalu lintas. Berikut adalah ringkasan kesimpulan yang merespon pertanyaan-pertanyaan utama dari penelitian:

- Teknologi pengenalan gambar menggunakan model YOLO V8 X telah terbukti sangat efektif dalam mendeteksi kepadatan kendaraan dengan cepat dan akurat. Dimana model YOLO 8 X yang dilatih mendapatkan optimasi diatas 90%, dengan mAP-50 yaitu 97.3%, maka memungkinkan kendaraan terdeteksi dengan baik.
- Algoritma yang dikembangkan berhasil mengkondisikan counter lampu lalu lintas dengan dinamis berdasarkan data kepadatan kendaraan yang terdeteksi. Sistem ini telah terbukti dapat menentukan waktu tunggu pada lampu hijau, dimana perhitungan secara manual dan perhitungan sistem terbukti sangat akurat.
- Pengintegrasian antara model YOLO dengan perangkat simulasi *traffic light* dapat dilakukan dengan menggunakan Arduino sebagai platform pengendali utama karena kemudahannya dalam integrasi, biaya yang efisien, dan fleksibilitas dalam pemrograman.
- Hasil kinerja implementasi menunjukkan bahwa sistem dapat berfungsi dengan baik dalam kondisi lalu lintas yang beragam. Pengujian sistem menunjukkan bahwa durasi lampu lalu lintas dapat disesuaikan secara dinamis. Dan dapat dilihat juga pada pengujian data test model YOLO 8 X mendapatkan akurasi sebesar 0.968 (96,8%).

Secara keseluruhan, proyek ini telah berhasil memenuhi sebagian besar dari tujuan yang diusulkan dan menunjukkan potensi besar untuk diterapkan secara luas dengan beberapa perbaikan dan optimasi lebih lanjut.

6.2. Saran

Berdasarkan hasil penelitian dan implementasi sistem pengaturan durasi *Traffic Light* berbasis *Image Recognition*, berikut adalah beberapa saran untuk perbaikan dan pengembangan sistem di masa mendatang:

- **Peningkatan Kualitas Perangkat Keras:** Menggunakan kamera dengan kualitas yang lebih tinggi dan kemampuan adaptasi terhadap berbagai kondisi cahaya akan meningkatkan akurasi sistem dalam mendeteksi kendaraan, terutama dalam kondisi cahaya rendah. Selain itu, peningkatan kapasitas komputasi perangkat keras akan membantu dalam mempercepat respons sistem terhadap perubahan kepadatan lalu lintas.
- **Optimasi Algoritma *Image Recognition*:** Meskipun model YOLO V8 X telah memberikan hasil yang baik, ada ruang untuk optimasi lebih lanjut. Penggunaan teknik-teknik deep learning terbaru atau peningkatan dataset pelatihan dengan lebih banyak variasi kondisi lalu lintas dapat meningkatkan akurasi dan efisiensi sistem.
- **Penggunaan Panjang Kendaraan sebagai Parameter:** Menambahkan panjang kendaraan sebagai parameter dalam penghitungan durasi lampu lalu lintas dapat meningkatkan akurasi dalam menentukan kepadatan lalu lintas. Dengan mempertimbangkan panjang kendaraan, sistem dapat lebih tepat dalam mengestimasi ruang yang ditempati di jalan dan menyesuaikan durasi lampu lalu lintas dengan lebih akurat.
- **Penerapan *Intelligent Traffic Light*:** Menerapkan sistem *Intelligent Traffic Light* yang dapat berkomunikasi dan berkoordinasi dengan lampu lalu lintas lain di persimpangan terkait. Ini akan memastikan pengaturan durasi lampu lalu lintas yang sinkron antara lampu lalu lintas sebelum dan setelahnya, sehingga aliran lalu lintas dapat lebih lancar dan efisien. Sinkronisasi ini juga dapat membantu dalam mengurangi kemacetan dan meningkatkan keselamatan di jalan raya.

Dengan mengadopsi saran-saran ini, diharapkan sistem pengaturan durasi *Traffic Light* berbasis *Image Recognition* dapat semakin optimal dan memberikan manfaat yang lebih besar dalam manajemen lalu lintas di masa mendatang.

DAFTAR PUSTAKA

- [1] J.-P. Rodrigue, *The Geography of Transport Systems*. Fifth edition. | Abingdon, Oxon ; New York, NY : Routledge, 2020.: Routledge, 2020. doi: 10.4324/9780429346323.
- [2] Dinas Perhubungan Daerah Istimewa Yogyakarta, “Transportasi Dalam Angka 2022.” Diakses: 5 November 2023. [Daring]. Tersedia pada: <https://dishub.jogjaprov.go.id/files/114/Transportasi-Dalam-Angka/296/Transportasi-Dalam-Angka-2022.pdf>
- [3] A. K. Dewantoro, I. Iwut, dan E. Susatio, “Simulasi Dan Analisis Sistem Penghitung Kepadatan Lalu Lintas Dan Klasifikasi Kendaraan Berbasis Webcam Dengan Metode Background Subtraction Simulation And Analysis System Counters Traffic Density And Classification With Webcam Based Background Subtraction.” Diakses: 10 Desember 2023. [Daring]. Tersedia pada: https://openlibrary.telkomuniversity.ac.id/pustaka/files/101919/jurnal_eproc/simulasi-dan-analisis-sistem-penghitung-kepadatan-lalu-lintas-dan-klasifikasi-kendaraan-berbasis-webcam-dengan-metode-background-subtraction.pdf
- [4] S. D. Maniswari, A. Rusdinar, dan B. Purnama, “Smart Traffic Light Menggunakan Image Processing dan Metode Fuzzy Logic,” *eProceedings of Engineering*, vol. 2, no. 2, 2015.
- [5] M. Rifai dkk., “Dynamic time distribution system monitoring on traffic light using image processing and convolutional neural network method,” *IOP Conf Ser Mater Sci Eng*, vol. 1175, no. 1, hlm. 012005, Agu 2021, doi: 10.1088/1757-899X/1175/1/012005.
- [6] F. C. Filipe da Silva, D. Siswanto, dan I. Istiadi, “Design Prototype of Traffic Light Control System using HSV Colour Segmentation,” *Evolution of Information, Communication and Computing System*, vol. 3, no. 1, hlm. 80–91, Jun 2022, [Daring]. Tersedia pada: <https://publisher.uthm.edu.my/bookseries/index.php/eiccs/article/view/36>
- [7] C. Niu dan K. Li, “Traffic Light Detection and Recognition Method Based on YOLOv5s and AlexNet,” *Applied Sciences (Switzerland)*, vol. 12, no. 21, Nov 2022, doi: 10.3390/app122110808.

- [8] S. Sanchi Kanojia, “Real-time Traffic light control and Congestion avoidance system.” Diakses: 21 November 2023. [Daring]. Tersedia pada: https://www.ijera.com/papers/Vol2_issue2/EY22925929.pdf
- [9] MathWorks, “What Is Image Recognition?,” MathWorks. Diakses: 21 November 2023. [Daring]. Tersedia pada: <https://www.mathworks.com/discovery/image-recognition-matlab.html#:~:text=Image%20recognition%20is%20the%20process,medical%20imaging%2C%20and%20security%20surveillance.>
- [10] Urban Street Design Guide, “Signal Cycle Lengths,” NACTO - Island Press. Diakses: 19 Desember 2023. [Daring]. Tersedia pada: <https://nacto.org/publication/urban-street-design-guide/intersection-design-elements/traffic-signals/signal-cycle-lengths/#:~:text=Short%20cycle%20lengths%20of%2060%E2%80%9390,creating%20a%20more%20permeable%20network>
- [11] B. C. C. Meng, N. S. Damanhuri, dan N. A. Othman, “Smart traffic light control system using image processing,” *IOP Conf Ser Mater Sci Eng*, vol. 1088, no. 1, hlm. 012021, Feb 2021, doi: 10.1088/1757-899X/1088/1/012021.
- [12] M. M. Gandhi, D. S. Solanki, R. S. Daptardar, dan N. S. Baloorkar, “Smart Control of Traffic Light Using Artificial Intelligence,” dalam *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, IEEE, Des 2020, hlm. 1–6. doi: 10.1109/ICRAIE51050.2020.9358334.
- [13] J. Jocher, G.; Chaurasia, A.; Qiu, “YOLO by Ultralytics,” 2023.
- [14] Raspberry Pi (Trading) Ltd, “Raspberry Pi Zero 2 W,” *Product Reference Manual*, hlm. 1–6, 2021, Diakses: 25 Desember 2023. [Daring]. Tersedia pada: <https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/>
- [15] P. R. Manual, “Arduino UNO R3 Features,” *Https://Docs.Arduino.Cc*, 2022.
- [16] J. Solawetz dan Francesco, “What is YOLOv8? The Ultimate Guide.,” Roboflow.

LAMPIRAN – LAMPIRAN

- **Lampiran *Technical Report*/Logs kegiatan**

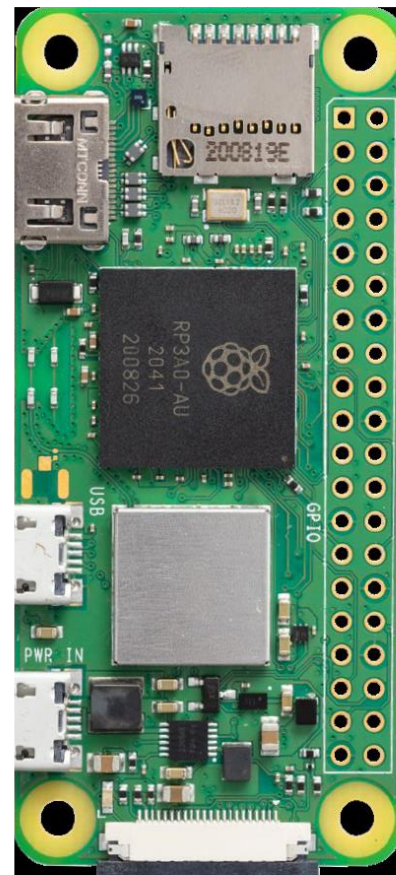
No	Hari, Tanggal, Durasi (jam atau hari)	Aktivitas	Pelaksana
1	Sabtu, 23 September 2023	Mencari Studi Literature dan Jurnal Tentang Materi Capstone	Reza
2	Minggu, 24 September 2023	Progress pembuatan pendahuluan, rumusan masalah, dan tujuan	Reza, Rafi
3	Selasa, 3 Oktober 2023	Revisi Batasan Masalah dan Batasan Realistis	Rafi
4	Selasa, 10 Oktober 2023	Revisi Batasan Masalah dan Batasan Realistis	Rafi
5	Selasa, 17 Oktober 2023	Finalisasi BAB 1	Rafi, Reza
6	Kamis, 19 Oktober 2023	Bimbingan Landasan Teori	Rafi
7	Rabu, 25 Oktober 2023	Mencari studi literature dan observasi terkait tentang materi yang sama	Reza
8	Sabtu, 28 Oktober 2023	Bimbingan TA102	Rafi
9	Sabtu, 4 November 2023	Progress pembuatan TA102	Reza
10	Minggu, 5 November 2023	Progress pembuatan TA102	Reza
11	Senin, 6 November 2023	Penjelasan dan Revisi dokumen TA 102	Rafi, Reza
12	Jumaat, 10 November 2023	Survei ke lokasi mitra (Dishub)	Reza, Rafi
13	Selasa, 14 November 2023	Penjelasan usulan solusi TA 103	Reza, Rafi
14	Rabu, 22 November 2023	Pemabahasan TA 103	Reza, Rafi
15	Selasa, 5 Desember 2023	Progres usulan solusi	Reza, Rafi
16	Minggu, 10 Desember 2023	Revisi usulan solusi 1	Reza
17	Kamis, 13 Desember 2023	Revisi usulan solusi 2	Rafi
18	Sabtu, 16 Desember 2023	Penentuan Usulan Solusi Terbaik	Reza, Rafi
19	Minggu, 24 Desember 2023	Revisi Penentuan Usulan Solusi Terbaik	Reza
20	Kamis, 28 Desember 2023	Seminar <i>Capstone</i> Project	Reza, Rafi
21	Selasa, 27 Februari 2024	Penjelasan TA 2 dan Timeline	Rezadi Ramadhan Rafi Mahdiansyah
22	Senin, 4 Maret 2024, 3	Pembelian Komponen	Rezadi Ramadhan
23	Kamis, 14 Maret 2024	Bimbingan	Rezadi Ramadhan Rafi Mahdiansyah
24	Senin, 18 Maret 2024	Pembuatan Model YOLO	Rezadi Ramadhan
25	Sabtu, 23 Maret 2024	Progress report perancangan	Rezadi Ramadhan Rafi Mahdiansyah
26	Rabu, 03 April 2024	Progress report perancangan	Rezadi Ramadhan Rafi Mahdiansyah
27	Kamis, 25 April 2024	Progress report perancangan	Rezadi Ramadhan Rafi Mahdiansyah
28	Selasa, 07-05-2024	Progress report perancangan	Rezadi Ramadhan Rafi Mahdiansyah

29	Kamis, 30-05-2024	Pembuatan Perangkat Keras Simulasi <i>Traffic Light</i>	Rafi Mahdiansyah
30	Kamis, 06-06-2024	Progress report perancangan	Rezadi Ramadhan Rafi Mahdiansyah
31	Jumat, 21-06-2024	Pengujian dan Validasi	Rezadi Ramadhan
32	Kamis, 27-06-2024	Pembuatan Laporan	Rezadi Ramadhan Rafi Mahdiansyah
33	Selasa, 9 Juli 2024	Pengumpulan Laporan	Rezadi Ramadhan Rafi Mahdiansyah
34	Minggu , 14 Juli 2024	Pengumpulan File Presentasi Ujian	Rezadi Ramadhan Rafi Mahdiansyah
35	Senin, 15 Juli 2024	Ujian / Sidang TA2	Rezadi Ramadhan Rafi Mahdiansyah
36	Rabu, 31 Juli 2024	Pelaksanaan Expo	Rezadi Ramadhan Rafi Mahdiansyah

- **Lampiran Dokumentasi spesifikasi**

Specification

Form factor:	65mm × 30mm
Processor:	Broadcom BCM2710A1, quad-core 64-bit SoC (Arm Cortex-A53 @ 1GHz)
Memory:	512MB LPDDR2
Connectivity:	• 2.4GHz IEEE 802.11b/g/n wireless LAN, Bluetooth 4.2, BLE, onboard antenna • 1 × USB 2.0 interface with OTG • HAT-compatible 40-pin I/O header footprint • microSD card slot • Mini HDMI port • CSI-2 camera connector
Video:	• HDMI interface • Composite video
Multimedia:	• H.264, MPEG-4 decode (1080p30) • H.264 encode (1080p30) • OpenGL ES 1.1, 2.0 graphics
Input power:	5V DC 2.5A
Operating temperature:	-20°C to +70°C
Production lifetime:	Raspberry Pi Zero 2 W will remain in production until at least January 2028
Compliance:	For a full list of local and regional product approvals, please visit pip.raspberrypi.com



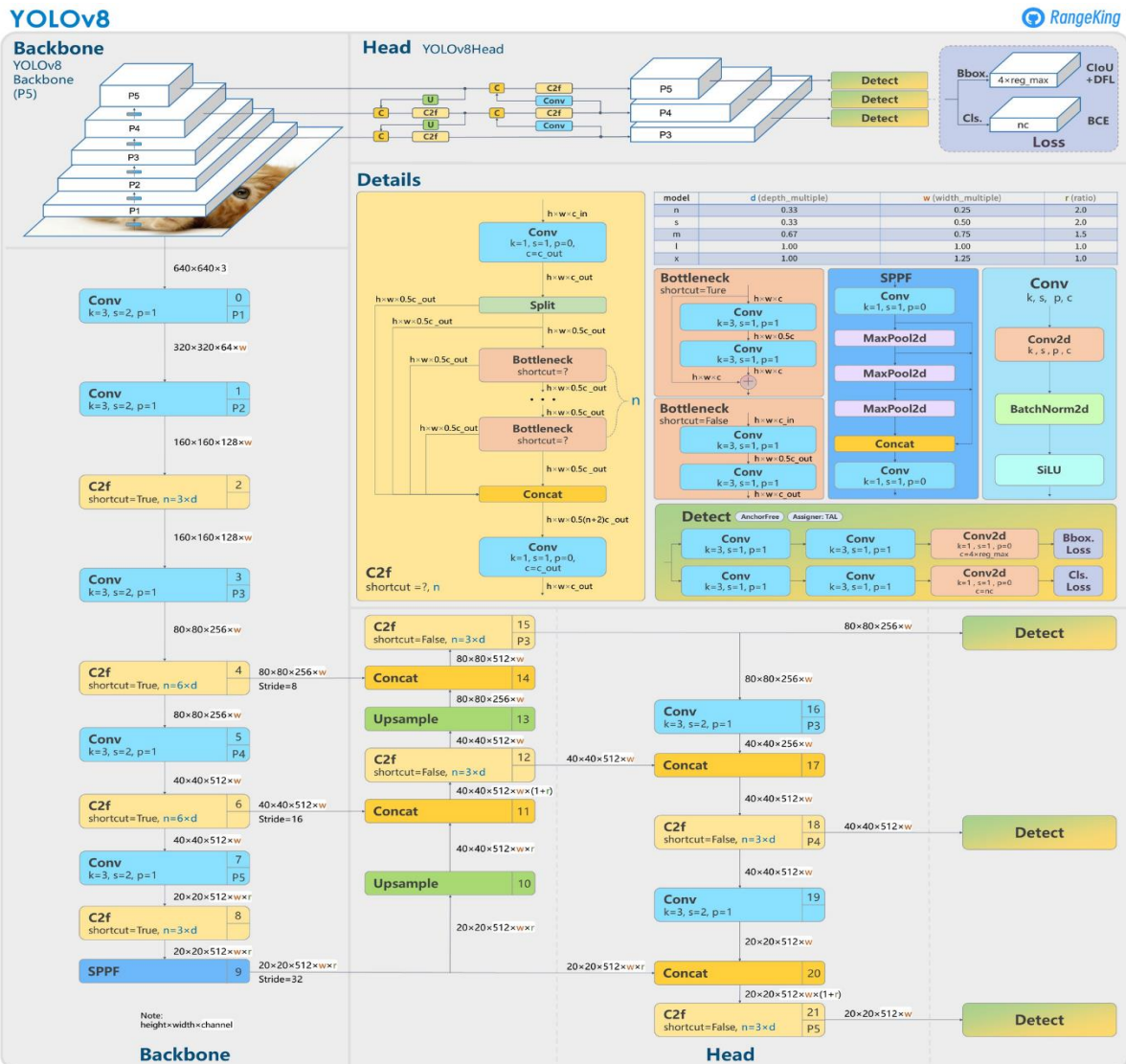
Gambar Lampiran.1 Spesifikasi *Raspberry Pi Zero 2 W*[14]

Features

- **ATMega328P Processor**
 - **Memory**
 - AVR CPU at up to 16 MHz
 - 32KB Flash
 - 2KB SRAM
 - 1KB EEPROM
 - **Security**
 - Power On Reset (POR)
 - Brown Out Detection (BOD)
 - **Peripherals**
 - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
 - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
 - 1x USART with fractional baud rate generator and start-of-frame detection
 - 1x controller/peripheral Serial Peripheral Interface (SPI)
 - 1x Dual mode controller/peripheral I2C
 - 1x Analog Comparator (AC) with a scalable reference input
 - Watchdog Timer with separate on-chip oscillator
 - Six PWM channels
 - Interrupt and wake-up on pin change
- **ATMega16U2 Processor**
 - 8-bit AVR® RISC-based microcontroller
- **Memory**
 - 16 KB ISP Flash
 - 512B EEPROM
 - 512B SRAM
 - debugWIRE interface for on-chip debugging and programming
- **Power**
 - 2.7-5.5 volts



Gambar Lampiran.2 Spesifikasi Arduino R3[15]



97

- **Lampiran Tabel Perbaikan**

**TABEL PERBAIKAN
LAPORAN AKHIR CAPSTONE**

MAHASISWA #1 : 20524143 Rezadi Ramadhan
MAHASISWA #2 : 20524145 Rafi Mahdiansyah Falah
JUDUL/TOPIK : Pengatur Kepadatan Traffic Light Berbasis Image Recognition

No	Saran penguji	Perbaikan oleh mahasiswa	Halaman	Status
1	Bagian ringkasan: tidak lazim memasukkan nama penulis pada bagian ringkasan, intisari atau abstrak performa 0,973 pada ringkasan itu satuannya apa? yang berhasil laporan anda atau sistem yang anda buat? (cek bagian terakhir ringkasan)	Menghapus nama penulis dan memberikan keterangan persen (%) pada performa model YOLO yang dibuat.	6	Approved -
2	BAB 3 Gambar 3.2 terdapat flowchart dengan simbol decision (berbentuk layang-layang) namun tidak ada arah panah true dan false! Gambar 3.10 juga begitu, cek! metodologi apa yang digunakan untuk menentukan solusi terbaik? sehingga yang terpilih adalah solusi No.2 Pada usulan solusi 1, satuan kapasitas areanya apa? apakah persamaan ini yang berlaku umum untuk menghitung kepadatan jalan? pada solusi 2 kenapa kendaraan harus dibagi menjadi beberapa kelas, sedangkan pada metode menghitung kepadatan seluruh kelas di total? variabel K siapa yang menentukan? kenapa dikali 7? ini belum ada penjelasannya! tolong dijelaskan	Memperbaiki Gambar Flowchart pada usulan solusi 1 dan usulan solusi 2. Kemudian memberikan keterangan pada persamaan kepadatan untuk satuan dipakai yaitu jumlah kendaraan. Serta memberikan penjelasan konstanta "k" dan pemakaian panjang kendaraan untuk klasifikasi jenis kendaraan. Dan terakhir membuat decision matrix untuk usulan solusi terbaik.	27, 37-38, 41-42	Approved -

No	Saran penguji	Perbaikan oleh mahasiswa	Halaman	Status
3	BAB 4 pada BAB 4 anda perlu menjelaskan secara runut mekanisme kerja sistem sehingga mudah dipahami oleh stakeholder atau pengguna? apa bedanya hasil pengujian loses yang ditunjukkan pada BAB 4 dan BAB 5? apakah beda atau sama? hasil pengujian tidak dimasukkan di masukkan di BAB 4 tapi di BAB 5 Pada BAB 4 anda hanya menjelaskan metode klasifikasinya, kemudian bagaimana prosedur counting dan kategorisasi kepadatan kondisi lalu lintas!	Memperbaiki susunan penjelasan optimasi model image recognition "YOLO" dan memberikan penjelasan prosedur counting dan mekanisme sistem.	48-60	Approved -
4	BAB 5 kenapa anda menggunakan confusion matriks sebagai matrik evaluasi? kenapa perlu mencari tahu performa akurasi, presisi, recall dan f-1 score	Confusion matrix merupakan matriks yang sering digunakan sebagai alat evaluasi performa model klasifikasi. Alasan utama menggunakan confusion matrix adalah karena matriks ini memberikan gambaran yang lebih mendetail tentang bagaimana model berperilaku dalam hal klasifikasi data ke dalam kelas-kelas tertentu. Confusion matrix menunjukkan jumlah true positives (TP), true negatives (TN), false positives (FP), dan false negatives (FN). Dengan informasi ini, kita bisa memahami kesalahan spesifik yang dilakukan model, misalnya, apakah model sering salah dalam mendeteksi kelas tertentu. kenapa perlu mencari tahu performa akurasi, presisi, recall dan f-1 score Akurasi: Mengukur seberapa sering model membuat prediksi yang benar secara keseluruhan. Ini adalah metrik dasar, tapi seperti yang disebutkan sebelumnya, bisa bias jika ada ketidakseimbangan kelas. Presisi: Menunjukkan seberapa banyak dari prediksi positif yang benar-benar positif. Ini penting terutama ketika biaya kesalahan false positive tinggi, misalnya dalam diagnosis medis. Recall (Sensitivitas): Mengukur seberapa banyak dari semua data yang sebenarnya positif yang berhasil ditemukan oleh	65-69	Approved -

No	Saran penguji	Perbaikan oleh mahasiswa	Halaman	Status
		model. Ini penting jika kita ingin meminimalkan kesalahan false negative, seperti dalam pendeteksian penyakit yang berbahaya. F1-Score: Kombinasi dari presisi dan recall, memberikan gambaran performa model yang lebih seimbang terutama ketika kita ingin mempertimbangkan keduanya secara bersamaan. F1-score sangat berguna ketika ada ketidakseimbangan antara presisi dan recall.		

Yogyakarta, 7-Agustus-2024

Menyetujui,
Penguji

Elvira Sukma Wahyuni, S.Pd., M.Eng