

TUGAS AKHIR - EC184801

DETEKSI DAN MONITORING KEMACETAN BERBASIS CITRA STREAMING VIDEO MENGGUNAKAN UNMANNED AERIAL VEHICLE TIPE FIXED WING DENGAN METODE YOLO

Putu Wisnu Bhaskara Putrawan NRP 07211640000031

Dosen Pembimbing Dr. Eko Mulyanto Yuniarno, ST., MT. Arief Kurniawan ST., MT.

DEPATERMEN TEKNIK KOMPUTER Fakultas Teknologi Elektro dan Informatika Cerdas Institut Teknologi Sepuluh Nopember Surabaya 2020



TUGAS AKHIR - EC184801

DETEKSI DAN MONITORING KEMACETAN BERBASIS CITRA STREAMING VIDEO MENGGUNAKAN UNMANNED AERIAL VEHICLE TIPE FIXED WING DENGAN METODE YOLO

Putu Wisnu Bhaskara Putrawan NRP 07211640000031

Dosen Pembimbing Dr. Eko Mulyanto Yuniarno, ST., MT. Arief Kurniawan ST., MT.

DEPATERMEN TEKNIK KOMPUTER Fakultas Teknologi Elektro dan Informatika Cerdas Institut Teknologi Sepuluh Nopember Surabaya 2020



FINAL PROJECT - EC184801

CONGESTION DETECTION AND MONITORING BASED ON VIDEO STREAMING USING UNMANNED AERIAL VEHILCE FIXED WING TYPE WITH YOLO METHOD

Putu Wisnu Bhaskara Putrawan NRP 07211640000031

Advisors Dr. Eko Mulyanto Yuniarno, ST., MT. Arief Kurniawan ST., MT.

Departement of Computer Engineering Faculty of Intelligent Electrical and Informatics Technology Sepuluh Nopember Institute of Technology Surabaya 2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "Deteksi dan Monitoring Kemacetan Berbasis Citra Streaming Video Menggunakan Unmanned Aerial Vehicle tipe Fixed Wing dengan Metode YOLO" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak dijinkan dan bukan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2020

Putu Wisnu Bhaskara Putrawan NRP, 07211640000031



LEMBAR PENGESAHAN

Deteksi dan Monitoring Kemacetan Berbasis Citra Streaming Video menggunakan Unmanned Aerial Vehicle tipe Fixed Wing dengan Metode YOLO

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh: Putu Wisnu Bhaskara Putrawan (NRP: 07211640000031)

Tanggal Ujian: 7 Juli 2020 Periode Wisuda: September 2020

Disetujui oleh:

<u>Dr. Eko Mulyanto Yuniarno ST., MT.</u> (Pembimbing I)

NIP: 196806011995121009

<u>Arief Kurniawan ST., MT.</u> (Pembimbing II)

NIP: 197409072002121001

Reza Fuad Rachmadi, ST., MT., Ph.D. (Penguji I)

NIP: 198504032012121001

<u>Dr. Surya Sumpeno ST., M.Sc.</u> (Penguji II)

NIP. 196906131997021003

Eko Pramunanto, ST. MT. (Penguji III)

NIP. 196612031994121001

Mengetahui

Kepala Départemen Teknik Komputer

DEDAT Suprno Mardi Susiki Nugroho, ST., MT.

EKNIK KOMPUTERIP. 19700313/1995121001



ABSTRAK

Nama Mahasiswa : Putu Wisnu Bhaskara Putrawan

Judul Tugas Akhir : Deteksi dan Monitoring Kemacetan Ber-

basis Citra Streaming Video menggunakan Unmanned Aerial Vehicle tipe Fixed

Wing dengan Metode YOLO

Pembimbing : 1. Dr. Eko Mulyanto Y. S.T., M.T.

2. Arief Kurniawan, S.T., M.T.

Kemacetan menjadi masalah utama baik di kota-kota besar maupun di daerah pinggiran kota. Kemacetan biasanya terjadi pada saat liburan panjang dan pada saat arus mudik. Banyak upaya telah dilakukan untuk memantau kemacetan namun masih kurang efisien dan tidak dapat memberikan data secara detail dari kemacetan tersebut. Hal ini membuat penanganan kemacetan menjadi kurang cepat. Metode yang diterapkan saat ini untuk deteksi dan monitoring kemacetan adalah pemasangan CCTV dengan sistem IP-Camera. Penerapan sistem IP-Camera tidak dapat diterapkan di daerah pinggiran kota karena minimnya akses internet. Sistem ini juga tidak dapat mencakup area yang luas karena pemasangan kamera dilakukan secara statis. Maka dari itu, pada tugas akhir ini akan dikembangkan suatu pendeteksi kemacetan yang dapat memberikan data secara detail dan realtime dengan jangkauan yang luas menggunakan UAV (Unmanned Aerial Vehicle) tipe Fixed Wing.

Kata Kunci : Pendeteksi Kemacetan, Realtime, UAV (Unmanned Aerial Vehicle), Fixed Wing

 $Halaman\ ini\ sengaja\ dikosongkan$

ABSTRACT

Name : Putu Wisnu Bhaskara Putrawan

Final Project Title : Congestion Detection and Monitoring ba-

sed on Video Streaming Footage using Unmanned Aerial Vehicle type Fixed

Wing with YOLO Method

Advisors : 1. Dr. Eko Mulyanto Y. S.T., M.T.

2. Arief Kurniawan, S.T., M.T.

Congestion or traffic jam is a major problem both in big cities and in suburban areas. In Indonesia, congestion usually occurs during long holidays and when the "mudik" event. Many attempts have been made to monitor traffic jam but it is still inefficient and cannot provide detailed data of these jam. This makes congestion difficult to overcome. The method currently applied for congestion detection and monitoring is the installation of CCTV with the IP-Camera system. The application of the IP-Camera system cannot be implemented in suburban areas due to lack of internet access. This system also cannot cover a large area because the installation of the camera is done statically. Therefore, in this final project a congestion detection will be developed which can provide detailed and realtime data with a wide range using UAV (Unmanned Aerial Vehicle) type Fixed Wing.

Keywords: Congestion Detection, Realtime, UAV (Unmanned Aerial Vehicle), Fixed Wing

 $Halaman\ ini\ sengaja\ dikosongkan$

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa karena atas segala berkah dan kemudahanNya, penulis dapat menyelesaikan penelitian yang berjudul "Deteksi dan Monitoring Kemacetan Berbasis Citra Streaming Video menggunakan Unmanned Aerial Vehicle tipe Fixed Wing dengan Metode YOLO".

Penelitian ini disusun dalam rangka untuk memenuhi tugas akhir Depatremen Teknik Komputer ITS, Bidang Studi Computer Vision dan Embedded System, serta digunakan sebagai persyaratan menyelesaikan pendidikan di jenjang S1. Penelitian ini dapat diselesaikan dengan bantuan dari berbagai pihak. Oleh karena itu, penulis mengucapkan banyak terimakasih kepada:

- 1. Tuhan Yang Maha Esa.
- 2. Kedua orangtua penulis, Bapak Nyoman Putrawan dan Ibu Luh Sri Artini yang selalu memberikan dukungan baik secara material maupun secara spiritual demi kelancaran proses penelitian ini.
- 3. Bapak Kepala Departemen Teknik Komputer Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.
- Bapak Dr. Eko Mulyanto Yuniarno, S.T., M.T. dan Bapak Arief Kurniawan S.T., M.T. selaku dosen pembimbing yang sudah memberikan dukungan dan bimbingan selama mengerjakan penelitian ini.
- 5. Tim Bayucaraka ITS yang membantu penulis dalam hal kebutuhan manufaktur UAV dan tempat istirahat.
- 6. Teman-teman YOLO yang sudah bersama-sama belajar metode objek deteksi YOLO dari awal.

Penulis berharap segenap kritik dan saran yang bersifat membangun. Harapannya dari penelitian ini dapat diterapkan langsung dalam kehidupan sehari hari dan dapat bermanfaat untuk penelitian-penelitian selanjutnya.

Surabaya, Juli 2020

Penulis

DAFTAR ISI

| Al | bstra | ık | i | | | |
|---------------|-------|---|------|--|--|--|
| Abstract | | | | | | |
| K. | ATA | PENGANTAR | v | | | |
| D | AFT | AR ISI | vii | | | |
| DAFTAR GAMBAR | | | | | | |
| D | AFT | AR TABEL | xi | | | |
| N | OMI | ENKLATUR | xiii | | | |
| 1 | PE | NDAHULUAN | 1 | | | |
| | 1.1 | Latar belakang | 1 | | | |
| | 1.2 | Permasalahan | 2 | | | |
| | 1.3 | Tujuan | 2 | | | |
| | 1.4 | Batasan masalah | 2 | | | |
| | 1.5 | Sistematika Penulisan | 3 | | | |
| 2 | DA | SAR TEORI | 5 | | | |
| | 2.1 | Related Works | 5 | | | |
| | 2.2 | Convolutional Neural Networks | 6 | | | |
| | 2.3 | YOLO (You Only Look Once) | 7 | | | |
| | 2.4 | Darkflow | 9 | | | |
| | 2.5 | Unmanned Aerial Vehicle | 9 | | | |
| | 2.6 | Mission Planner | 12 | | | |
| 3 | DE | SAIN DAN IMPLEMENTASI SISTEM | 15 | | | |
| | 3.1 | Desain Sistem | 15 | | | |
| | 3.2 | Akuisisi Data | 16 | | | |
| | | 3.2.1 Desain UAV | | | | |
| | | 3.2.2 Sistem Elektronik UAV | 18 | | | |
| | | 3.2.3 Sistem Kamera UAV | 21 | | | |
| | | 3.2.4 Spesifikasi <i>UAV</i> yang Digunakan | 24 | | | |

| | | 3.2.5 Gambar Kendaraan Tampak Atas | 24 |
|----|-------|---|------------|
| | 3.3 | Training Data | 25 |
| | | 3.3.1 Labelling Gambar | 25 |
| | | 3.3.2 Ekstraksi Fitur menggunakan Darkflow | 27 |
| | | 3.3.3 Model Pendeteksi Kendaraan | 27 |
| | 3.4 | Penghitungan Kepadatan Jalan | 27 |
| | 3.5 | Status Kemacetan | 31 |
| 4 | PE | NGUJIAN DAN ANALISA | 35 |
| | 4.1 | Pengujian Performa dan Rute Terbang UAV | 35 |
| | 4.2 | Pengujian Durasi dan Kecepatan Terbang UAV | 37 |
| | 4.3 | Pengujian Deteksi Kendaraan | 39 |
| | | 4.3.1 Pengujian Deteksi Kendaraan dari Video <i>Pla-</i> | 4.0 |
| | | yback UAV | 40 |
| | | 4.3.2 Pengujian Deteksi Kendaraan dari Video <i>Streaming VTX</i> | 42 |
| | | 4.3.3 Pengujian Deteksi Kendaraan pada Tingkat | |
| | | Pencahayaan yang Berbeda | 44 |
| | | 4.3.4 Pengujian Deteksi Kendaraan dengan Keting- | |
| | | gian dan Kecepatan Terbang UAV yang Ber- | |
| | | variasi | 47 |
| | 4.4 | Pengujian Luas Area Optimal yang dapat Dimonito- | |
| | | ring UAV | 50 |
| | 4.5 | Pengujian Sistem Pendeteksi Kemacetan | 50 |
| 5 | PE | NUTUP | 5 5 |
| | 5.1 | Kesimpulan | 55 |
| | 5.2 | Saran | 56 |
| D. | AFT | AR PUSTAKA | 57 |
| D | logro | 6 Danulia | 50 |

DAFTAR GAMBAR

| 2.1 | Arsitektur dari <i>CNN</i> | 6 |
|------|--|----|
| 2.2 | Lapisan dari CNN | 7 |
| 2.3 | Arsitktur dari YOLO | 8 |
| 2.4 | Cara Kerja YOLO | 8 |
| 2.5 | $Quadcopter\ UAV\ \dots\dots\dots\dots\dots\dots\dots$ | 10 |
| 2.6 | Fixed Wing UAV | 11 |
| 2.7 | Gerakan Pada UAV | 12 |
| 2.8 | Mission Planner | 13 |
| 3.1 | Desain sistem keseluruhan | 15 |
| 3.2 | Alur Akuisisi Data | 16 |
| 3.3 | Desain UAV | 17 |
| 3.4 | UAV yang Digunakan | 18 |
| 3.5 | Komponen Elektronik pada UAV | 19 |
| 3.6 | Konfigurasi $Motor\ BLDC$ pada UAV | 20 |
| 3.7 | Pixhawk Cube | 20 |
| 3.8 | Modul GPS HERE GNSS | 21 |
| 3.9 | Kamera yang Digunakan Pada UAV | 22 |
| 3.10 | | 22 |
| | Konfigurasi VTX pada UAV | 23 |
| | Gambar Kendaraan Tampak Atas | 25 |
| | Proses Training Data | 26 |
| 3.14 | Proses Pelabelan Gambar | 26 |
| | Penghitungan Panjang Jalan | 28 |
| 3.16 | Simulasi Jumlah Mobil Maksimal dalam Suatu Jalan | 28 |
| 3.17 | Simulasi Jumlah Bus Maksimal dalam Suatu Jalan . | 29 |
| 3.18 | Simulasi Jumlah Truk Maksimal dalam Suatu Jalan | 30 |
| 3.19 | Simulasi Jumlah Motor Maksimal dalam Suatu Jalan | 31 |
| 3.20 | Desain Interface dari Sistem Pendeteksi Kemacetan | 32 |
| 3.21 | Desain Interface pada Saat Jalan Padat | 33 |
| 3.22 | Desain Interface pada Saat Jalan Macet | 33 |
| 4.1 | Uji Coba UAV di Pusat Robotika ITS | 35 |
| 4.2 | Pembuatan Waypoint pada Mission Planner | 36 |
| 4.3 | Hasil Terbang Autonomous dari UAV | 37 |

| 4.4 | LiPo Battery Checker |
|------|--|
| 4.5 | Akurasi dari Model yang Dibuat 40 |
| 4.6 | Percobaan Terbang 1 41 |
| 4.7 | Percobaan Terbang 2 |
| 4.8 | Deteksi Kendaraan pada Video Streaming 43 |
| 4.9 | Deteksi Motor pada Video Streaming |
| 4.10 | Deteksi Mobil pada Video Streaming 44 |
| 4.11 | Hasil Deteksi Kendaraan pada Pagi Hari 45 |
| 4.12 | Hasil Deteksi Kendaraan pada Siang Hari 46 |
| 4.13 | Hasil Deteksi Kendaraan pada Sore Hari 46 |
| 4.14 | Hasil Deteksi Kendaraan pada Malam Hari 47 |
| 4.15 | Panjang Jalan Maksimal yang Dapat Dimonitoring |
| | oleh UAV |
| 4.16 | Sistem Pendeteksi Kemacetan 51 |
| 4.17 | Data yang Disajikan |
| 4.18 | Percobaan pada Video Youtube 1 |
| 4.19 | Percobaan pada Video Youtube 2 |
| 4.20 | Percobaan pada Video Youtube 3 53 |

DAFTAR TABEL

| 3.1 | Spesifikasi UAV | 24 |
|-----|---|----|
| 3.2 | Daya Tampung Jalan untuk Mobil | 29 |
| 3.3 | Daya Tampung Jalan untuk Bus | 30 |
| 3.4 | Daya Tampung Jalan untuk Truck | 30 |
| 3.5 | Daya Tampung Jalan untuk Motor | 31 |
| | | |
| 4.1 | Tabel Pengujian Durasi Terbang UAV | 38 |
| 4.2 | Tabel Pengujian Kecepatan Terbang UAV | 39 |
| 4.3 | Akurasi Deteksi pada Percobaan Terbang 1 | 41 |
| 4.4 | Akurasi Deteksi pada Percobaan Terbang 2 | 42 |
| 4.5 | Tabel Uji Coba Ketinggian pada Sistem Pendeteksi | |
| | Kendaraan | 48 |
| 4.6 | Tabel Uji Coba Kecepatan Terbang pada Sistem Pen- | |
| | deteksi Kendaraan | 50 |

 $Halaman\ ini\ sengaja\ dikosongkan$

NOMENKLATUR

 $\begin{array}{ll} TP & : \mbox{ True Positive} \\ TN & : \mbox{ True Negative} \\ FP & : \mbox{ False Positive} \\ FN & : \mbox{ False Negative} \end{array}$

 $Halaman\ ini\ sengaja\ dikosongkan$

BAB 1 PENDAHULUAN

Penelitian ini dibuat berdasarkan latar belakang dari berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dalam penelitian.

1.1 Latar belakang

Semakin banyaknya kendaraan pada saat ini membuat kemacetan menjadi permasalahan utama di jalan raya baik di wilayah perkotaan sampai di wilayah pinggiran kota [1]. Pada wilayah kotakota besar, kemacetan terjadi tanpa mengenal waktu. Sedangkan pada daerah pinggiran kota, kemacetan sering terjadi pada waktuwaktu tertentu seperti hari libur panjang atau mudik lebaran. Pada daerah kota, kemacetan dipantau menggunakan CCTV yang dipasang di tempat tempat rawan kemacetan. Pada saat ini, kamera CCTV vang dipasang menggunakan sistem IP-Camera vaitu sebuah sistem kamera wireless yang pengiriman datanya dilakukan melalui jaringan internet. Karena CCTV hanya dipasang pada titik-titik tertentu, CCTV hanya dapat memantau kemacetan pada tititk-titik tersebut dan membutuhkan banyak CCTV untuk memantau area yang luas. Namun pada daerah pinggiran kota, kebanyakan titiktitik kemacetan tidak dapat dipantau secara langsung dikarenakan sedikitnya tempat tempat yang dapat dipasangi kamera seperti lampu lalu lintas. Hal ini membuat tindakan untuk mengurai kemacetan seperti pengalihan jalur menjadi lambat. Selain itu, faktor lainnya adalah kebanyakan daerah pinggiran kota merupakan daerah yang minim jaringan internet sehingga jika diberi CCTV dengan sistem IP-Camera akan sulit untuk diterapkan [2]. Karena terbatasnya tempat untuk memasang kamera dan minimnya akses internet pada daerah pinggiran kota, perlu adanya teknik akuisisi data secara aerial agar dapat mendeteksi kemacetan dengan area yang luas dan bersifat temporer.

Salah satu media yang dapat mengambil data secara aerial yaitu sebuah robot terbang atau UAV. $Unmanned\ Aerial\ Vehicle$ atau disingkat UAV merupakan teknologi robot yang dapat terbang dan

beroperasi secara manual dengan remot kontrol atau autonomus menggunakan sistem komputer tertanam. UAV dapat dilengkapi dengan berbagai sensor tambahan seperti kamera dan modul antena video jarak jauh untuk memberikan data berupa streaming video untuk keperluan monitoring secara realtime [3]. UAV dibagi menjadi 2 tipe yaitu Rotary Wing dan Fixed Wing dimana untuk UAV tipe Rotary Wing terbang menggunakan baling baling besar sebagai pengganti sayap sedangkan UAV tipe Fixed Wing terbang menggunakan sayap sebagai pemberi gaya angkat utama UAV juga dapat terbang dengan cakupan area yang luas dan dapat beroperasi bahkan di daerah daerah terpencil tanpa memerlukan jaringan internet [4].

1.2 Permasalahan

Dari sistem monitoring dan pendeteksi kemacetan yang sudah ada sebelumnya, penggunaan kamera CCTV berbasis *IP-Camera* masih kurang efektif karena *recovery area* dari sistem ini sangatlah sempit. Selain itu sistem ini juga membutuhkan jaringan internet dimana tidak semua ruas jalan terdapat jaringan internet terutama di daerah pinggiran kota. Permasalahan lainnya adalah sistem monitoring kemacetan yang ada saat ini belum bisa menampilkan data kemacetan secara detail seperti jumlah kendaraan berdasarkan jenisnya. Perlu adanya suatu sistem pendeteksi dan monitoring kemacetan yang dapat mencakup area yang luas. Sistem tersebut harus bisa beroperasi tanpa memerlukan jaringan internet dalam pengiriman datanya dan dapat menampilkan data monitoring kemacetan secara detail.

1.3 Tujuan

Dari rumusan masalah tersebut, tujuan dari tugas akhir ini adalah untuk membuat sistem pendeteksi dan monitoring kemacetan yang dapat mencakup area yang luas berbasis citra *streaming video* dengan memanfaatkan teknologi *Unmanned Aerial Vehicle* tipe *Fixed Wing*.

1.4 Batasan masalah

Adapun batasan masalah pada Tugas Akhir ini dapat dilihat sebagai berikut :

- 1. Luas area yang dapat dimonitoring oleh *UAV* adalah area persegi yang memiliki ukuran maksimal 3km x 3km.
- 2. Jenis kendaraan yang dapat dideteksi hanya 4 jenis yaitu mobil, motor, bus, dan truck.
- 3. Daerah yang akan dipantau kemacetannya hanya daerah sekitar lingkungan kampus ITS.
- Waktu pengujian dilakukan pada waktu pagi, siang, dan sore hari.

1.5 Sistematika Penulisan

Laporan penelitian Tugas Akhir ini disusun secara sistematis dan terstruktur sehingga pembaca atau seseorang yang ingin melanjutkan penelitian ini dapat dengan mudah memahami dan mempelajarinya. Alur sistematika penulisan laporan penelitian ini yaitu:

1. BAB I Pendahuluan

Bab ini berisi tentang uraian latar belakang, permasalahan, alasan pemilihan judul, sistematika laporan, tujuan, dan metodologi penelitian.

2. BAB II Dasar Teori

Bab ini berisi tentang uraian sistematis dari teori-teori yang berhubungan dengan permasalahan yang dibahas pada tugas akhir ini. Teori teori yang digunakan sebagai dasar dalam pengerjaan antara lain informasi tentang deep learning, teori tentang UAV, dan teori penunjang lainnya.

3. BAB III Perancangan Sistem dan Impementasi Bab ini menjelaskan tentang sistem yang akan dibuat mulai dari sistem pendeteksi kendaraan sampai dengan sistem UAV itu sendiri. Blok diagram digunakan untuk memperjelas alur dari sistem tersebut.

4. BAB IV Pengujian dan Analisa

Pada bab ini, dijelaskan tentang proses pengujian sistem yang sudah dibuat dan melakukan analisa terhadap sistem tersebut. Spesifikasi dari masing masing sistem juga dijelaskan pada bab ini.

5. BAB V Penutup

Bab terakhir berisi kesimpulan yang diambil berdasarkan penelitian dan pengujian yang telah dilakukan. Kritik dan saran yang bersifat membangun juga dituliskan pada bab ini.

BAB 2 DASAR TEORI

Untuk mendukung penelitian ini, dibutuhkan beberapa teori pendukung sebagai acuan dan referensi dari berbagai sumber. Dengan demikian penelitian ini dapat berjalan lebih terarah.

2.1 Related Works

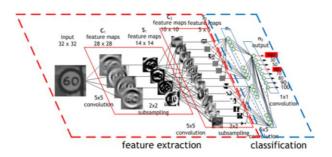
Pada sub bab ini akan dijelaskan mengenai beberapa penelitian terkait yang telah dilakukan sebelumnya. Penelitian ini berguna sebagai sumber referensi yang akan menjadi acuan pada penelitian tugas akhir ini.

- 1. Penelitian yang berjudul Applications of Unmanned Aerial Vehicle (UAV) Technology for Research and Education in UAE oleh Alkabi menyebutkan bahwa teknologi UAV dapat terbang di area yang luas dan bahkan dapat beroperasi di daerah terpencil tanpa memerlukan jaringan internet[4].
- 2. Penelitian yang berjudul Detecting and tracking vehicles in traffic by unmanned aerial vehicles yang dilakukan oleh Liang Wang melakukan proses monitoring kemacetan menggunakan UAV jenis rotary wing[2]. Penelitian tersebut sudah dapat memberikan data kemacetan secara realtime namun daya jelajah dari UAV itu sendiri tidak terlalu jauh. Sistem pendeteksi kendaraanya pun masih belum bisa membedakan kendaraan berdasarkan jenisnya.
- 3. Pada penelitian yang berjudul Guidance of Unmanned Aerial Vehicle oleh Yanushevsky, UAV dapat dilengkapi dengan berbagai sensor tambahan seperti kamera dan modul antena video jarak jauh untuk memberikan data berupa streaming video untuk keperluan monitoring secara realtime[3].
- 4. Berdasarkan penelitian yang berjudul Traffic Congestion And Possible Solutions A CASE STUDY OF ASANSOL oleh Maji, semakin banyaknya kendaraan pada saat ini membuat kemacetan menjadi permasalahan utama di jalan raya baik di wilayah perkotaan sampai di wilayah pinggiran kota[1].

- 5. Penelitian yang berjudul A Traffic Monitoring System for a Mixed Traffic Flow Via Road Estimation and Analysis oleh Nguyen Viet Hung menyebutkan bahwa Traffic Monitoring System merupakan sesuatu yang sangat dibutuhkan untuk mengkontrol arus kendaraan[5].
- 6. Dalam penelitian yang berjudul "Pengembangan Sistem Penghitung Jumlah Kendaraan Memanfaatkan Citra Aerial yang Diambil dengan Pesawat Tanpa Awak" oleh Susilo Budi Prakoso, Kendaraan di jalan dihitung menggunakan UAV tipe Rotary Wing. Sistem yang dibuat sudah bisa menghitung kendaraan yang lewat dalam suatu area pada kamera namun sistem tersebut belum bisa membedakan jenis kendaraan yang lewat [6].

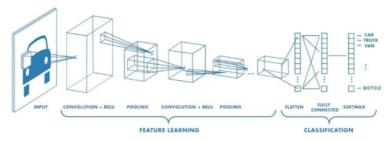
2.2 Convolutional Neural Networks

Convolutional Neural Networks atau biasa disebut CNN adalah salah satu metode machine learning yang merupakan variasi dari Multilayer Perceptron yang dibuat berdasarkan inspirasi dari jaringan syaraf manusia. CNN digunakan untuk menyelesaikan hal hal yang bersifat khusus seperti pattern recognition. CNN termasuk dalam jenis Deep Neural Network karena dalamnya tingkat jaringan dan banyak diimplementasikan dalam data citra. Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi.



Gambar 2.1: Arsitektur dari CNN

Sama halnya dengan Neural Network pada umumnya, CNN memiliki beberapa lapisan tersembunyi (hidden layers) dari sebuah input berupa vector tunggal. Pada Gambar 2.1 dapat dilihat arsitektur dari *CNN*. CNN terdiri dari berbagai lapisan yang dimana setiap lapisan memiliki Application Program Interface (API). Untuk lebih jelasnya bisa dilihat pada Gambar berikut.

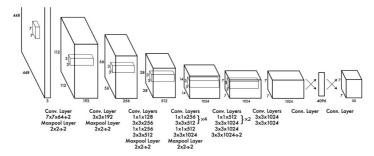


Gambar 2.2: Lapisan dari CNN

2.3 YOLO (You Only Look Once)

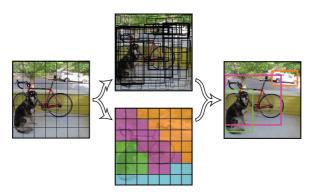
You only look once (YOLO) adalah sebuah struktur yang berfungsi untuk mendeteksi sebuah objek yang berskala Real-time[7]. YOLO menggunakan pendekan yang sangat berbeda dengan algoritma sebelumnya, yakni menerapkan jaringan syaraf tunggal pada keseluruhan gambar. Jaringan ini akan membagi gambar menjadi wilayah-wilayah kemudian memprediksi kotak pembatas dan probabilitas, untuk setiap kotak wilayah pembatas ditimbang probabilitasnya klasifikasian sebagai objek atau bukan. Kebanyakan sistem deteksi sebelumnya menggunakan pengklasifikasian atau localizer untuk melakukan deteksi dengan menerapkan model ke gambar di beberapa lokasi dan skala dan memberi nilai pada gambar sebagai bahan untuk pendeteksian. YOLO adalah jaringan syaraf yang pintar untuk melakukan deteksi secara real-time. YOLO memiliki arsitektur yang sederhana yaitu jaringan saraf convolutional. Arsitektur dari YOLO dijelaskan pada Gambar 2.3.

Dalam sistem deteksi *YOLO*, input frame atau gambar dibagi menjadi S x S grid dengan nilai S adalah kelipatan 32. Jika titik tengah sebuah objek yang di deteksi berada di dalam sebuah *grid*



Gambar 2.3: Arsitktur dari YOLO

cells, maka grid cell bertugas untuk mendeteksi objek tersebut. Setiap grid cell memprediksi bounding box dan confidence score untuk bounding box tersebut. Rasio ukuran dari bounding box ditentukan dari nilai anchors pada layer terakhir dari arsitektur YOLO. Cofidence score adalah sebuah nilai ukuran kepastian dan akuratnya objek yang dideteksi. Rentang nilai dari confidence score adalah 0 sampai 1. Apabila nilai mendekati 0 maka sistem seperti tidak yakin dalam menentukan class dari objek tersebut. Apabila nilai mendekati 1, sistem deteksi akan yakin bahwa objek yang di deteksi sesuai dengan class-nya.



Gambar 2.4: Cara Kerja YOLO

YOLO memiliki beberapa keunggulan diantaranya, proses komputasinya yang sangat cepat. Hal ini dikarenakan YOLO hanya menjalankan satu layer neural network untuk memprediksi deteksi pada satu frame atau gambar. Walaupun begitu, akurasi yang didapatkan YOLO jauh lebih baik dari object detector lainnya seperti SSD. Selain itu, YOLO mempertimbangkan deteksi secara general ketika membuat prediksi terhadap gambar atau frame. YOLO melihat gambar secara utuh dan membuat informasi mengenai class yang sesuai dengan tekstur atau pola pada gambar tersebut. Keunggulan lain dari YOLO adalah YOLO mempelajari representasi objek yang bersifat general sehingga kecil kemungkinannya untuk error ketika diterapkan pada objek dengan bentuk yang agak berbeda atau input yang tidak biasa.

2.4 Darkflow

Darkflow merupakan Implementation Tool untuk YOLO[8]. Dengan menggunakan Darkflow, metode YOLO bisa diterapkan untuk mendeteksi objek pada gambar atau pada video. Selain itu Darkflow juga berfungsi untuk tools untuk melakukan training model YOLO. Darkflow menggunakan TensorFlow dan dapat di install baik di sistem operasi Linux maupun Windows. Darkflow dibuat dengan bahasa Python sehingga lebih fleksibel diterapkan di sistem operasi lainnya. Namun sampai saat ini, Darkflow hanya dapat men-support YOLO dan YOLOv2.

2.5 Unmanned Aerial Vehicle

Unmanned Aerial Vehicle (UAV) adalah salah satu jenis robot penjelajah udara tanpa awak. UAV juga sering disebut dengan nama Pesawat Udara Nir Awak (PUNA) dalam Bahasa Indonesia. Karena tidak memiliki awak, UAV harus dikendalikan dari jarak jauh menggunakan remote control dari luar kendaraan atau biasa disebut Remotely Piloted Vehicle (RPV). Selain itu, UAV juga dapat bergerak secara otomatis berdasarkan program yang sudah ditanamkan pada sistem komputernya. UAV dapat ditambahkan beberapa sensor tambahan untuk membantu proses akuisisi data seperti kamera, sensor ultrasonic, dan sensor-sensor lainnya.

Secara umum UAV dapat dibedakan menjadi 2 kategori yaitu Fixed Wing dan Rotary Wing. Berikut ini adalah penjelasan masing

masing kategori tersebut:

1. Rotary Wing UAV

Rotary Wing adalah jenis UAV yang tidak memiliki sayap namun terdiri dari rotor sebagai penggerak baling baling. Jenis ini memiliki banyak variasi diantaranya 4 rotor (quadcopter), 6 rotor (hexacopter), dan 8 rotor (octocopter). Semakin banyak jumlah rotor maka pergerakan dari UAV ini akan semakin halus dan stabil namun hal ini juga membutuhkan baterai dengan kapasitas besar.



Gambar 2.5: Quadcopter UAV

Cara kerja dari Rotary Wing UAV yaitu gaya angkat yang dihasilkan berasal dari rotor sehingga UAV dapat terbang. UAV ini dapat melakukan gerakan maju, mundur kanan, kiri, dan diam di tempat (hover). Keuntungan terbesar dari Rotary Wing UAV adalah kemapuan lepas landas dan mendarat secara vertical sehingga pengguna dapat menerbangkannya dari mana saja tanpa memerlukan landasan pacu. Kemampuan untuk berputar dan bergerak secara dinamis kesegala arah serta mempertahankan posisi diam di udara menjadikan data foto yang diambil dapat diambil berulang kali pada posisi yang tetap. kekurangan dari model ini adalah konsumsi daya yang cukup besar karena memiliki sistem yang kompleks.

2. Fixed Wing UAV

Fixed Wing merupakan salah satu jenis UAV yang memiliki

sayap seperti pesawat terbang pada umumnya. UAV tipe Fi- $xed\ Wing$ memiliki sayap yang kokoh dan memiliki airfoil atau
bentuk penampang sayap yang menganut hukum bernouli sehingga mampu memberikan gaya angkat saat UAV bergerak.
Kecepatan aliran udara pada sisi atas sayap harus lebih cepat
dibandingkan dengan aliran udara pada sisi bawah sayap agar
dapat menghasilkan gaya angkat.

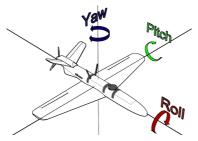


Gambar 2.6: Fixed Wing UAV

Terdapat 2 jenis konfigurasi Fixed Wing berdasarkan posisi baling-balingnya yaitu konfigurasi pusher dan puller. Pada konfigurasi pusher baling-baling berada di belakang badan UAV dan bekerja sebagai pendorong. Sedangkan pada konfigurasi puller, baling-baling berada di depan UAV dan bekerja sebagai penarik. Terdapat beberapa variasi Fixed Wing berdasarkan ada atau tidaknya ekor dan posisi ekor. Variasi pertama adalah bentuk Talon yang memiliki ekor di posisi belakang sayap UAV seperti ekor pada pesawat umumnya. Variasi kedua adalah Canard yang memiliki ekor di depan sayap UAV. Variasi ketiga adalah Fliying Wing yang tidak memiliki ekor (hanya sayap dan badan UAV).

Kontrol dari UAV berasal dari papan kontrol yang tertanam dalam pesawat. Sistem kontrol yang biasa disebut control surface terdiri dari aileron atau penggerak pada sayap, elevator atau penggerak pada ekor horizontal, dan rudder atau penggerak pada ekor vertikal. Berdasarkan Gambar 2.7, aileron berfungsi untuk menghasilkan gerakan roll, elevator menghasilkan gerakan pitch, dan rudder menghasilkan gerakan yaw. Pada saat terbang, UAV mengkombi-

nasikan gerakan pitch, yaw, dan roll untuk melakukan manuver.



Gambar 2.7: Gerakan Pada UAV

Bagian itu menjadikan UAV dapat terbang bebas dan berputar di tiga sumbu yang tegak lurus sama lain dan berpotongan di pusat gravitasi UAV itu sendiri. Baik model Fixed Wing maupun model Rotary Wing dapat melakukan ketiga gerakan tersebut. Model Fixed Wing memiliki kelebihan yaitu terbuat dari struktur yang lebih sederhana dibandingkan dengan tipe Rotary Wing. Model Fixed Wing dapat dibuat dari bermacam macam bahan mulai dari gabus, kayu, hingga bahan komposit. Model Fixed Wing juga memiliki daya jelajah yang lebih jauh dan lebih lama dibandingkan dengan model Rotary Wing.

2.6 Mission Planner

Mission Planner merupakan sebuah aplikasi ground station yang dibuat oleh Michael Oborne untuk ArduPilot open source board seperti Pixhawk yang digunakan pada penelitian ini[9]. Aplikasi ini hanya compatible dengan sistem operasi Windows. Mission Planner dapat digunakan untuk konfigurasi plane, copter, dan rover. Tampilan menu utama dari Mission Planner ditunjukkan pada Gambar 2.8.

Fitur utama dari Mission Planner yaitu dalam aplikasi ini kita dapat membuat flight plan atau jalur terbang dari UAV dan melakukan monitoring parameter secara realtime ketika UAV terbang menjalankan misi secara autonomous. Mission Planner dapat menampilkan data UAV seperti ketinggian, kecepatan, posisi, kemiringan, dan lain-lain. Fitur lainnya adalah Mission Planner dapat digunakan sebagai media untuk proses upload firmware pada Pi-



Gambar 2.8: Mission Planner

xhawkatau $flight\ controller$ lainnya. $Mission\ Planner\ melakukan komunikasi dengan\ UAV\ menggunakan komunikasi serial pada\ USB$ baik komunikasi secara wire maupun komunikasi secara wireless.

 $Halaman\ ini\ sengaja\ dikosongkan$

BAB 3 DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanankan sesuai dengan sistem yang sudah dirancang beserta implementasinya. Desain sistem ini merupakan konsep dasar yang digunakan dalam pembuatan dan perancangan infrastruktur yang kemudian disajikan dalam bentuk blok-blok diagram. Bagian implementasi merupakan realisasi dari blok diagram pada desain sistem.

3.1 Desain Sistem

Tugas akhir ini merupakan gabungan dari bidang penelitian $Computer\ Vision$, Telematika, dan $Embedded\ System$ yang bertujuan untuk mendeteksi kemacetan di suatu jalan menggunakan UAV secara realtime dengan metode YOLO untuk menghitung jumlah kendaraan pada jalan tersebut. Diagram alur pada Gambar 3.1 adalah gambaran dari sistem yang akan dibuat.



Gambar 3.1: Desain sistem keseluruhan

Berdasargan gambar tersebut, langkah awal yang harus dilakukan adalah dengan akuisisi data dengan mengumpulkan data kendaraan tampak atas. Ada 4 jenis kendaraan yang harus ada pada data tersebut yaitu mobil, motor, bus, dan truck. Gambar tersebut dapat diperoleh dari hasil monitoring UAV. Selanjutnya adalah tahap training data. Data gambar yang sudah di dapat kemudian diberi label sesuai dengan jenis kendaraan yang terdapat pada gambar tersebut. Selanjutnya, dilakukan ekstraksi fitur menggunakan Darkflow. Hasil dari training tersebut menghasilkan output weight yang berupa meta file dan pb file yang nantinya akan digunakan YOLO untuk mendeteksi kendaraan pada input video streaming dari UAV. Dari hasil deteksi tersebut dapat diperoleh jumlah kendaraan dalam 1 frame. Jumlah tersebut digunakan untuk menentukan kepadatan jalan dengan rumus yang telah ditentukan. Nilai kepadatan jalan yang diperoleh akan digunakan untuk menentukan status kemacetan pada suatu jalan.

3.2 Akuisisi Data

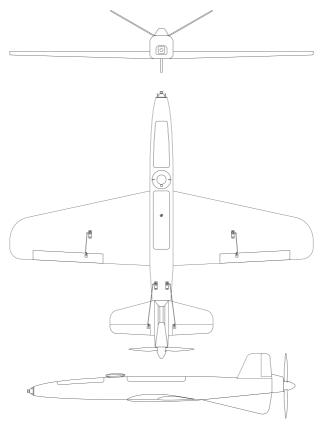
Dalam penelitian tugas akhir ini, teknik akuisisi data dilakukan dengan mengambil gambar melalui rekaman video monitoring UAV. Pada Gambar 3.2 dijelaskan tahapan dalam proses akuisisi data.



Gambar 3.2: Alur Akuisisi Data

3.2.1 Desain UAV

UAVyang digunakan merupakan UAVjenis $Fixed\ Wing\ yang$ merupakan hasil kerajinan tangan sendiri. Pada Gambar 3.3 dapat dilihat desain dari UAVyang digunakan memiliki penggerak baling-baling di bagian belakang. Hal ini bertujuan untuk mengurangi getaran yang dihasilkan oleh baling baling terhadap kamera monitoring.



Gambar 3.3: Desain UAV

Desain UAV ini memiliki panjang badan 114.6 cm dihitung mulai dari ujung depan hingga ujung belakang dan lebar 130 cm

dihitung mulai dari ujung sayap kiri sampai ukung sayap kanan. Dimensi penampang badan dibuat dengan ukuran maksimal $10 \, \mathrm{cm}$ x $10 \, \mathrm{cm}$ dan dimensi ruang dalam UAV bagian tengah adalah sebesar $7 \, \mathrm{cm}$ x $7 \, \mathrm{cm}$. Ukuran tersebut dipilih agar dapat menampung seluruh sistem yang dibutuhkan seperti kamera dan komponen elektronik lainnya. Aspek aerodinamika juga diperhitungkan pada saat mendesain UAV ini seperti luas sayap, $Center\ of\ Gravity$, dan bobot muatan. UAV yang digunakan dapat dilihat pada Gambar 3.4.

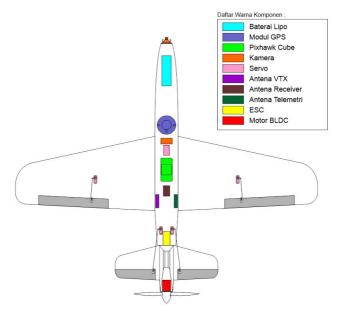


Gambar 3.4: UAV yang Digunakan

Bentuk ekor dari UAV menggunakan konfigurasi Y-Tail yang bertujuan untuk menambah efisiensi dan dapat menghasilkan manuver berbelok yang lebih tajam dibandingkan dengan bentuk ekor tradisional. Konfigurasi ini dipilih karena dibutuhkan UAV yang dapat terbang dengan durasi yang lama dan dapat melakukan manuver lincah mengikuti bentuk jalan yang akan dimonitoring.

3.2.2 Sistem Elektronik UAV

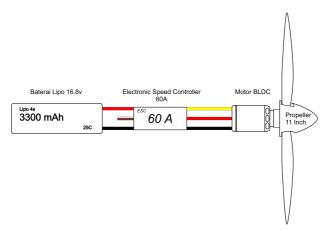
Sistem elektronik pada UAV ini terdiri dari sistem penggerak dan sistem autonomous dimana kedua sistem ini saling berhubungan satu sama lain. Pada Gambar 3.5 dapat dilihat tata letak komponen elektronik pada UAV yang digunakan.



Gambar 3.5: Komponen Elektronik pada UAV

Sistem penggerak dari UAV terdiri dari Propeller, $Motor\ BLDC$, $ESC\ (Elelctronic\ Speed\ Controller)$, Baterai LiPo, dan servo sebagai penggerak $control\ surface$. UAV ini menggunakan $Motor\ BLDC$ dengan Spesifikasi diameter 35mm, panjang 48mm, dan memiliki daya sebesar 610 watt. $ESC\$ yang digunakan memiliki maksimal batas arus sebesar 60 ampere. Baterai yang digunakan memiliki beda potensial sebesar 16.8 volt. $Propeller\$ atau baling-baling yang digunakan pada UAV ini memiliki 3 bilah dengan diameter sebesar 11 inch. Kombinasi ini memungkinkan sistem penggerak UAV dapat menghasilkan tenaga untuk mengangkat beban sebesar kurang lebih 3 kg. Data ini didapat dari datasheet $Motor\ BLDC\$ yang digunakan.

Servo yang digunakan adalah servo tipe mikro dengan metal gear. Servo ini memiliki torsi yang cukup besar untuk menggerakkan control surfaces pada UAV. Total ada 4 servo yang digunakan pada UAV ini.



Gambar 3.6: Konfigurasi $Motor\ BLDC$ pada UAV

Sistem autonomous untuk UAV menggunakan Microcontroller Pixhawk Cube seperti pada Gambar 3.7. Pixhawk Cube didesain tidak hanya untuk UAV tetapi juga dapat digunakan untuk tipe autonomous vehicle lain seperti kapal selam, autonomous car, dan antenna tracker. Dengan berbasis 32bit STM32F427 Cortex-M4F core with FPU, pengolahan data dari sensor pada UAV dapat dilakukan dengan cepat. Dalam penelitian ini, Pixhawk Cube di set dengan firmware ArduPlane V4.0.5 yang sementara ini merupakan firmware terbaru untuk UAV tipe Plane.



Gambar 3.7: Pixhawk Cube

Sistem navigasi pada UAV menggunakan modul gps HERE GNSS yang dapat dilihat pada Gambar 3.8. Modul GPS ini memiliki beberapa kelebihan yaitu update rate sebesar 10 Hz, tingkat presisi tinggi (akurasi 1 meter), waktu fix position yang cepat (26 detik pada cold start), dan dilengkapi dengan internal compass. Hal ini dibutuhkan untuk menjaga posisi UAV selalu pada jalur yang telah dibuat. Selain itu gps ini juga memiliki resistansi terhadap noise yang tinggi sehingga aman saat dioperasikan pada tempat tempat yang memiliki tingkat noise yang tinggi seperti daerah yang dekat dengan SUTET.



Gambar 3.8: Modul GPS HERE GNSS

Sensor bawaan pada pixhawk perlu di kalibrasi seperti compass dan gyro. Kalibrasi sensor tersebut dilakukan di program Mission Planner dengan prosedur yang sudah disediakan di dalam program tersebut. Penting untuk melakukan kalibrasi agar sistem navigasi autonomous UAV tidak mengalami kesalahan yang akan berdampak pada performa terbangnya nanti.

3.2.3 Sistem Kamera UAV

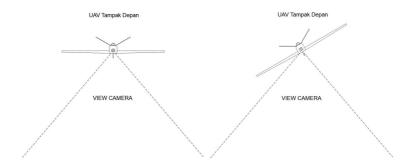
Gambar kendaraan tampak atas diambil dari video pada kamera UAV yang diterbangkan secara autonomous diatas jalan sekitar Gedung Robotika ITS. Kamera yang digunakan adalah kamera $Runcam\ 2$ yang dapat mengambil data video dengan resolusi HD 1080p 60fps.

Kamera ini dipasang menghadap bawah pada bagian bawah



Gambar 3.9: Kamera yang Digunakan Pada UAV

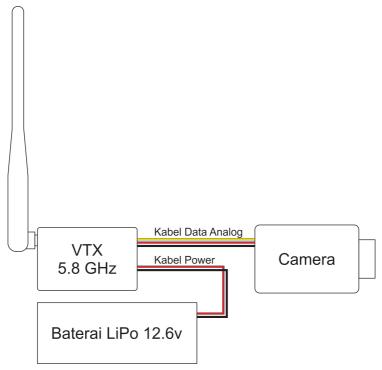
UAV dan diberi 1-axis stabilizer agar pada saat UAV berbelok, kamera masih menghadap bawah. Sistem stabilizer kamera dijelaskan pada Gambar 3.10. Stabilizer kamera UAV dibuat menggunakan servo yang dihubungkan ke Pixhawk Cube. Pixhawk akan memberikan feedback dari data kemiringan UAV kepada servo dan servo akan bergerak berlawanan sesuai dengan sudut kemiringan UAV. Kecepatan respon servo yang digunakan adalah 0.11 detik / 60 derajat. UAV terbang pada ketinggian 40 meter dari tanah dengan kecepatan kurang lebih 10 m/s atau 36 km/jam.



Gambar 3.10: Cara Kerja Stabilizer Kamera pada UAV

Pengiriman data video dari UAV dilakukan menggunakan module khusus yang bernama FPV Module. Alat ini bekerja seperti antena TV dan bekerja pada frekuensi 5.8 GHz. Dengan daya sebesar 600mW, modul ini dapat mengirimkan data video streaming

kurang lebih 2 Km LOS (Line Of Sight) dengan antena standar. Hal ini membuat UAV dapat terbang dengan radius maksimal 2 km. Modul FPV terdiri dari 2 perangkat yaitu Video Transmitter dan Video Receiver. Video Transmitter atau yang biasa disebut VTX dipasang pada UAV dan bertugas untuk memancarkan data video yang didapat dari kamera UAV. Konfigurasi VTX pada UAV dapat dilihat pada UAV dapat dilihat pada UAV



Gambar 3.11: Konfigurasi VTX pada UAV

Data video yang dikirimkan oleh VTX bersifat broadcast yang artinya semua orang yang memiliki *Video Receiver* yang berada di dalam jangkauan sinyal dapat melihat apa yang sedang di rekam oleh *UAV*. Terdapat 2 jenis *Video Receiver* yaitu tipe analog dan digital. Perbedaannya adalah *Video Receiver* analog menggunakan

kabel AV sebagai kabel datanya sehingga tipe ini tidak dapat menampilkan data video pada laptop atau PC. Video Receiver digital dapat dihubungkan langsung pada laptop atau PC melalui kabel USB yang tersedia. Video Receiver digital akan terbaca sebagai USB Camera 2.0 atau setara dengan Webcam.

Hasil video streaming yang didapat kemudian di screen capture di beberapa frame yang terdapat mobil atau kendaraan lain di dalamnya. Gambar tersebut nantinya akan digunanakan untuk proses training dari YOLO.

3.2.4 Spesifikasi UAV yang Digunakan

Pemilihan desain mekanik dan sistem elektronik menghasilkan suatu sistem UAV dengan spesifikasi tertentu. Spesifikasi performa UAV yang digunakan dalam pengujian ini dijelaskan pada Tabel 3.1.

| Aspek | Nilai |
|--------------------|----------------------|
| Kecepatan Optimal | 12 meter / detik |
| Ketinggian Optimal | 60 meter |
| Durasi Terbang | 15 menit 12 detik |
| Jarak Terbang | 2 km radius |
| Luas Area yang | 2.736×2.736 |
| dapat dimonitoring | meter |

Tabel 3.1: Spesifikasi UAV

Dengan spesifikasi tersebut, diharapkan UAV yang telah dibuat dapat memonitoring kemacetan pada suatu jalan secara realtime dengan baik.

3.2.5 Gambar Kendaraan Tampak Atas

Gambar kendaraan tampak atas diperlukan sebagai dataset. Gambar ini terdiri dari 4 jenis kendaraan yaitu mobil, motor, bus, dan truck. Ukuran dari masing masing jenis kendaraan harus juga bervariasi agar nantinya sistem deteksi kendaraan dapat mengidentifikasi kendaraan pada ketinggian terbang UAV yang bervariasi. Semakin tinggi UAV terbang maka semakin kecil gambar kendaraan yang harus di deteksi. Gambar-gambar ini hanya menampilkan kendaraan dalam posisi vertikal atau mendekati vertikal. Hal

ini bertujuan agar kendaraan yang dideteksi merupakan kendaraan yang hanya berada di jalan tersebut dan bukan kendaraan yang sedang parkir dengan posisi horizontal atau kendaraan yang melintas dari jalan lain saat berada di persimpangan. Contoh gambar yang digunakan untuk dataset dapat dilihat pada Gambar 3.12.



Gambar 3.12: Gambar Kendaraan Tampak Atas

Gambar tersebut memiliki fitur yang bisa dikatakan sedikit sehingga perlu banyak variasi mulai dari warna, bentuk, hingga ukuran untuk membuat sistem pendeteksi kendaraan yang akurat. Total gambar yang didapat untuk dataset adalah 89 gambar dengan variasi yang beragam untuk setiap jenis kendaraan.

3.3 Training Data

Training data dilakukan setelah UAV telah mengambil gambar kendaraan tampak atas. Ada beberapa langkah yang harus dilakukan dalam proses training. Langkah langkah tersebut dapat digambarkan dalam blok diagram pada Gambar 3.13.

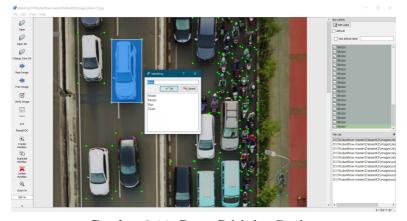
3.3.1 *Labelling* Gambar

Proses *labelling* merupakan proses untuk memberi tanda atau anotasi pada setiap gambar yang akan di training. Proses pelabelan gambar dilakukan menggunakan program *labelImg* versi 1.8.0. Hasil dari proses ini adalah file anotasi per gambar yang memiliki format .xml yang isinya adalah koordinat dari *bounding box* beserta labelnya. Proses pelabelan gambar dengan *labelImg* dapat dilihat pada



Gambar 3.13: Proses Training Data

Gambar 3.14.



Gambar 3.14: Proses Pelabelan Gambar

Kendaraan pada gambar masing masing ditandai menggunakan bounding box sesuai dengan jenisnya. Bounding box yang dibuat harus memiliki ukuran yang sesuai dengan ukuran kendaraan pada suatu gambar. Bounding box tidak direkomendasikan berukuan jauh lebih besar atau lebih kecil dari gambar untuk menghindari overfitting atau underfitting pada sistem pendeteksi kendaraan yang akan dibuat. Semua gambar yang akan ditraining dikumpulkan dalam satu folder dalam direktori Darkflow. Untuk file anotasi gambar

juga ditempatkan dalam satu folder terpisah namun masih di dalam direktori Darkflow.

3.3.2 Ekstraksi Fitur menggunakan Darkflow

Dataset yang sudah jadi kemudian akan diekstrak fiturnya menggunakan Darkflow. Directory dari Darkflow terlebih dahulu diakses menggunakan $command\ promp\ Anaconda\ Navigator$. Versi YOLO yang digunakan pada penelitian ini adalah YOLO v1 Untuk melakukan training, file .cfg terlebih dahulu disiapkan. File .cfg ini akan menjadi model awal yang berisi $convolutional\ layer$ dan beberapa parameter yang akan digunakan dalam proses training. Karena ada 4 jenis kendaraan yang di deteksi, maka parameter class pada file .cfg harus disesuaikan. Selain itu parameter filter pada layer terakhir juga harus di sesuaikan berdasarkan rumus filter = num*(num+class) dengan nilai $num\ default$ adalah 5. File label.txt pada $directory\ Darkflow$ juga harus disesuaikan dengan nama class yang akan di training. Dalam hal ini file label.txt diisi dengan "Mobil", "Motor", "Bus", dan "Truck".

Dalam proses training, spesifikasi perangkat yang digunakan adalah laptop Alienware 15 R3 dengan CPU Core i7 7700 HQ dengan clock speed sebesar 3.4 GHz. GPU yang digunakan adalah Nvidia GTX 1070 dengan memori 8192 MB. Jumlah epoch yang digunakan dalam training data ini adalah 2000 epoch. Optimizer yang digunakan pada training ini yaitu Adam Optimizer.

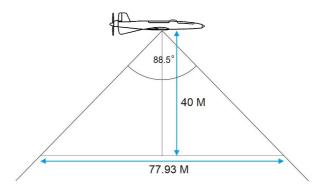
3.3.3 Model Pendeteksi Kendaraan

Hasil dari proses training ini disimpan dalam bentuk file .meta dan file .pb yang nantinya kedua file tersebut digunakan sebagai weight pada model pendeteksi kendaraan. File tersebut akan disimpan pada folder build graph di direktori Darkflow. Pada training ini hasil training dari checkpoint ke 8000 disimpan dalam bentuk .pb file dan .meta file.

3.4 Penghitungan Kepadatan Jalan

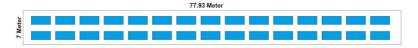
Data jumlah kendaraan yang di deteksi oleh sistem deteksi kendaraan digunakan untuk menentukan kepadatan dari suatu jalan. Karena tidak adanya rumus penghitungan kepadatan berdasarkan kumlah kendaraan dalam satu frame gambar, dibuatlah meto-

de baru untuk menghitung kepadatan jalan yang bernama Metode WTDC (Wisnu Traffic Density Counter). Pada metode WTDC, kepadatan suatu jalan ditentukan oleh luas jalan yang tampak pada kamera dibagi dengan jumlah kendaraan yang di deteksi. Luas jalan yang di deteksi dapat dihitung dengan rumus panjang jalan dikali dengan lebar jalan. Panjang jalan didapat dari perhitungan sudut pandang kamera. Dengan resolusi layar 16:9 dan sudut pandang horizontal 120 derajat, didapat sudut pandang vertikal sebesar 88.5 derajat. Ilustrasi perhitungan tersebut dapat dilihat pada Gambar 3.15.



Gambar 3.15: Penghitungan Panjang Jalan

Berdasarkan gambar tersebut, panjang jalan yang tampak pada kamera yaitu sepanjang 77.93 meter dengan ketinggian terbang UAV 40 meter dari jalan. Lebar jalan yang digunakan adalah 7 meter mengacu pada lebar jalan di pinggiran kota seperti jalan pantura.



Gambar 3.16: Simulasi Jumlah Mobil Maksimal dalam Suatu Jalan

Dalam Gambar 3.16 dapat dilihat jika sebuah jalan disimulasikan hanya berisi mobil dengan jarak 1 meter untuk setiap mo-

bilnya. Dimensi mobil yang digunakan adalah $4 \times 1,7$ meter yaitu ukuran standar untuk kategori mobil keluarga. Ukuran mobil keluarga dipilah karena pada saat liburan atau mudik, banyak dijumpai mobil keluarga di jalanan. Berikut ini adalah daya tampung jalan jika hanya ada mobil dalam jalan tersebut.

 No
 Keterangan
 Nilai

 1
 Panjang Jalan
 77.93 Meter

 2
 Lebar Jalan
 7 meter

 3
 Luas Jalan
 546 meter persegi

30 Mobil

1

Jumlah Kendaraan

Bobot Ukuran

 $\frac{4}{5}$

Tabel 3.2: Daya Tampung Jalan untuk Mobil

Karena mobil digunakan sebagai acuan, maka bobot ukuran mobil diberi nilai 1. Bobot ukuran ini nantinya akan digunakan dalam perhitungan status kemacetan.



Gambar 3.17: Simulasi Jumlah Bus Maksimal dalam Suatu Jalan

Dalam Gambar 3.17, jalan yang sama disimulasikan hanya berisi kendaraan bus dengan jarak 1 meter antar bus. Dimensi bus yang digunakan adalah 12.5×2.5 meter yaitu ukuran standar untuk jenis bus dengan daya tampung 50 orang. Ukuran bus tersebut digunakan karena pada jalan pinggiran kota atau pada perbatasan kota sering dijumpai bus antar provinsi dan bus pariwisata. Berikut ini adalah daya tampung jalan jika hanya terdapat bus dalam jalan tersebut.

Bobot ukuran bus didapat dari jumlah mobil maksimal yang dapat dideteksi dalam suatu jalan dibagi dengan jumlah bus maksimal yang dapat dideteksi dalam suatu jalan. Perhitungan ini menghasilkan bobot ukuran bus sebesar 3.

Gambar 3.18 menggambarkan simulasi jalan yang dimonitoring hanya berisi kendaraan truck. Jarak antar truck dibuat sebe-

Tabel 3.3: Daya Tampung Jalan untuk Bus

| No | Keterangan | Nilai |
|----|------------------|-------------------|
| 1 | Panjang Jalan | 77.93 Meter |
| 2 | Lebar Jalan | 7 meter |
| 3 | Luas Jalan | 546 meter persegi |
| 4 | Jumlah Kendaraan | 10 Bus |
| 5 | Bobot Ukuran | 3 |



Gambar 3.18: Simulasi Jumlah Truk Maksimal dalam Suatu Jalan

sar 1 meter. Dimensi truck yang digunakan adalah 8.5×2.5 meter yang dimana ukuran tersebut merupakan ukuran standar truck dengan dimensi bak 2×5.6 meter. Ukuran tersebut digunakan karena jumlah truck jenis ini paling banyak ditemukan di daerah pinggiran kota. Daya tampung jalan untuk truck dapat dilihat pada tabel berikut.

Tabel 3.4: Dava Tampung Jalan untuk Truck

| No | Keterangan | Nilai |
|----|------------------|-------------------|
| 1 | Panjang Jalan | 77.93 Meter |
| 2 | Lebar Jalan | 7 meter |
| 3 | Luas Jalan | 546 meter persegi |
| 4 | Jumlah Kendaraan | 16 Truck |
| 5 | Bobot Ukuran | 1.875 |

Bobot ukuran truck didapat dari jumlah mobil maksimal yang dapat dideteksi dalam suatu jalan dibagi dengan jumlah truck maksimal yang dapat dideteksi dalam suatu jalan. Perhitungan ini menghasilkan bobot ukuran truck sebesar 1,875.

Gambar 3.19 menunjukkan simulasi jalan yang dideteksi hanya berisi kendaraan berupa motor. Motor yang digunakan untuk simulasi berjarak 1 meter antar motor dan memiliki dimensi 2×0.6 meter. Daya tampung jalan untuk motor dapat dilihat pada tabel



Gambar 3.19: Simulasi Jumlah Motor Maksimal dalam Suatu Jalan

berikut.

Tabel 3.5: Daya Tampung Jalan untuk Motor

| No | Keterangan | Nilai |
|----|------------------|-------------------|
| 1 | Panjang Jalan | 77.93 Meter |
| 2 | Lebar Jalan | 7 meter |
| 3 | Luas Jalan | 546 meter persegi |
| 4 | Jumlah Kendaraan | 156 Motor |
| 5 | Bobot Ukuran | 0.192 |

Bobot ukuran motor didapat dari jumlah mobil maksimal yang dapat dideteksi dalam suatu jalan dibagi dengan jumlah motor maksimal yang dapat dideteksi dalam suatu jalan. Perhitungan ini menghasilkan bobot ukuran truck sebesar 0,192.

Dengan demikian, rumus untuk mendapatkan kepadatan jalan pada Metode WTDC yaitu :

$$Kepadatan = \frac{Luas \ Jalan}{(J_{mobil} \times B_{mobil}) + \dots + (J_{truck} \times B_{truck})} (3.1)$$

J = Jumlah Kendaraan

 $B = Bobot\ Ukuran\ Kendaraan$

Hasil dari perhitungan ini nantinya akan digunakan untuk mengambil keputusan apakah suatu jalan dikatakan macet atau tidak berdasarkan kepadatan jalannya.

3.5 Status Kemacetan

Status kemacetan diperoleh dari perhitungan kepadatan jalan menggunakan metode yang dibahas pada sub bab sebelumnya. Faktor yang sangat berpengaruh pada status kemacetan suatu jalan adalah lebar jalan. Pada kondisi jumlah kendaraan yang terdeteksi

sama dengan lebar jalan yang berbeda, status kemacetan yang dihasilkan akan berbeda. Sebagai contoh: dengan jumlah mobil yang terdeteksi sejumlah 30, jalan dengan lebar 7 meter akan berstatus macet sedangkan jalan dengan lebar 10.5 meter akan berstatus lancar. Hal ini dapat disimpulkan tentunya dengan asumsi ketinggian terbang UAV yang konstan.



Gambar 3.20: Desain Interface dari Sistem Pendeteksi Kemacetan

Status kemacetan dan jumlah kendaraan ditampilkan pada pojok kiri atas dari video streaming seperti pada Gambar 3.20. Jumlah kendaraan yang terdeteksi dan status kemacetan di-update secara realtime sehingga apa yang ditampilkan merupakan kondisi terbaru jalan yang di monitoring. Jika status kemacetan menunjukkan bahwa jalanan yang dimonitoring sedang padat, maka tampilan interface akan seperti Gambar 3.21.

Pada gambar tersebut terlihat bahwa tulisan "lancar" berganti menjadi "padat" dengan warna tulisan juga berubah menjadi warna kuning. Jika status kemacetan menunjukkan bahwa jalanan yang dimonitoring sedang macet, maka tampilan *interface* akan seperti Gambar 3.22.

Tulisan "padat" pada interface berubah menjadi "macet" dan warna status kemacetan berubah menjadi merah. Perbedaan warna ini berfungsi agar lebih mudah membedakan kondisi jalan yang



Gambar 3.21: Desain Interface pada Saat Jalan Padat



Gambar 3.22: Desain Interface pada Saat Jalan Macet

sedang di monitoring.

 $Halaman\ ini\ sengaja\ dikosongkan$

BAB 4 PENGUJIAN DAN ANALISA

Pada bab ini dipaparkan hasil pengujian serta analisa dari kedua desain sistem dan implementasinya. Pengujian dibagi menjadi empat bagian antara lain:

- 1. Pengujian Performa dan Rute Terbang UAV
- 2. Pengujian Durasi dan Kecepatan Terbang UAV
- 3. Pengujian Deteksi Kendaraan
- 4. Pengujian Luas Area Optimal yang dapat Dimonitoring UAV
- 5. Pengujian Sistem Pendeteksi Kemacetan

Dengan dilaksanakannya keempat pengujian tersebut, dapat ditarik kesimpulan dari pelaksanaan tugas akhir ini.

4.1 Pengujian Performa dan Rute Terbang UAV

Uji coba terbang dilakukan di halaman belakang Pusat Robotika ITS. Tempat ini dipilih karena jauh dari keramaian dan memiliki area tanpa halangan yang cukup luas. Hal ini perlu diperhatikan karena uji coba terbang UAV untuk pertama kali merupakan sesuatu yang cukup beresiko. Demi alasan keamanan, sistem autonomous dari UAV diubah ke sistem manual terlebih dahulu. UAV takeoff dengan cara dilempar tangan dan dikontrol secara manual terlebih dahulu.



Gambar 4.1: Uji Coba *UAV* di Pusat Robotika ITS

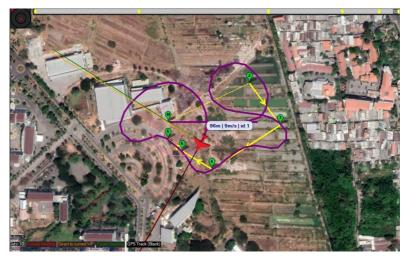
Setelah UAV dirasa tidak memiliki masalah saat uji coba manual, UAV akan didaratkan secara manual dan kemudian diberi program untuk menjalankan misi autonomous. Rute terbang harus dibuat sederhana dan tidak terlalu jauh antar titik atau waypoint. Pada Gambar 4.2 dapat dilihat interface dari program Mission Planner pada saat pembuatan waypoint. Waypoint kemudian di upload ke microcontroller Pixhawk Cube via antenna telemetri yang memiliki range frekuensi 433 MHz. UAV kembali diterbangkan namun kali ini dengan mode autonomous. Hasil terbang UAV secara autonomous dapat dilihat pada Gambar 4.3.



Gambar 4.2: Pembuatan Waypoint pada Mission Planner

Pada Gambar 4.3 dapat dilihat jalur terbang UAV ditandai dengan garis berwarna ungu sedangkan posisi UAV ditandai dengan gambar pesawat berwarna merah. Uji coba mode autonomous berjalan dengan lancar tanpa ada masalah sistem pada UAV. UAV sudah dapat mengikuti waypoint yang diberikan walaupun waypoint nomer 2 sedikit terlewati karena posisi UAV yang kurang pas saat menuju waypoint 2.

Pengujian ini dilakukan untuk melihat apakah jalur terbang UAV sudah sesuai dengan jalur yang dibuat pada program Mission Planner. Perubahan parameter Turn Control dan Airspeed sangat berpengaruh pada seberapa on-track UAV saat terbang. Turn time yang terlalu kecil akan membuat UAV overshoot atau keluar jalur



Gambar 4.3: Hasil Terbang Autonomous dari UAV

sedangkan Airspeed yang terlalu cepat membuat radius belok UAV menjadi sangat jauh.

4.2 Pengujian Durasi dan Kecepatan Terbang UAV

Pengujian ini dilakukan untuk mengetahui durasi terbang UAV yang aman agar UAV dapat kembali dengan selamat dan tidak merusak baterai yang digunakan. Selain itu, pengujian ini juga bertujuan untuk menetukan throttle maksimal agar UAV dapat terbang dengan waktu yang lama. Baterai yang digunakan pada UAV ini yaitu baterai LiPo 4 sel dengan tegangan full charge maksimal 16.8 volt dan tegangan aman minimal adalah sebesar 14.8 volt. Tegangan baterai diukur menggunakan battery checker seperti pada Gambar 4.4.

Battery Checker akan berbunyi ketika tegangan pada baterai LiPo telah melewati batas tegangan bawah untuk per sel (3.7 Volt). Durasi percobaan ditentukan oleh lamanya motor dihidupkan dengan nilai throttle tertentu hingga checker berbunyi. UAV akan dihadapkan ke atas untuk melihat apakah dalam nilai throttle tersebut, UAV dapat terangkat dari dorongan yang dihasilkan oleh propeller. Data percobaan ditampilkan pada Tabel 4.1.



Gambar 4.4: LiPo Battery Checker

Tabel 4.1: Tabel Pengujian Durasi Terbang UAV

| Percobaan | Throttle | Pesawat Terangkat? | Durasi |
|-----------|----------|-----------------------|-------------------|
| 1 | 100% | Ya | 2 Menit 3 Detik |
| 2 | 90% | Ya | 4 Menit 1 Detik |
| 3 | 80% | Ya | 7 Menit 7 Detik |
| 4 | 70% | Ya | 11 Menit 36 Detik |
| 5 | 60% | Tidak | 15 Menit 12 Detik |
| 6 | 50% | Tidak | 20 Menit 58 Detik |
| 7 | 40% | Tidak | 26 Menit 7 Detik |

Pada percobaan tersebut dapat diambil kesimpulan bahwa nilai throttle atau power minimum untuk membuat UAV bisa take-off adalah 70% dengan durasi 11 menit 36 detik sampai tegangan baterai melewati batas aman bawah tegangannya. Pada saat UAV terbang secara autonomous, nilai throttle tidak konstan namun berubah ubah dan tidak melebihi nilai maksimum yang diberikan. Nilai throttle pada saat cruise akan lebih rendah dari nilai throttle maksimal. Hal ini memungkinkan durasi terbang yang lebih lama dibandingkan dengan nilai hasil percobaan. Uji coba terbang terakhir menunjukkan UAV dapat terbang selama 15 menit dengan nilai throttle maksimal 70% dan nilai throttle pada saat cruise sebesar 40%.

Kecepatan terbang UAV berubah sesuai dengan throttle atau power yang digunakan. Semakin besar nilai throttle yang digunakan,

semakin cepat UAV terbang. Berdasarkan data yang diperoleh dari mission planner saat UAV terbang, pengaruh nilai throttle dengan kecepatan terbang UAV dijelaskan pada Tabel 4.2.

Percobaan Throttle Kecepatan Terbang 100% 22 m/s1 90% 2 20 m/s3 80% $18 \mathrm{m/s}$ 70% 4 $15 \mathrm{m/s}$ 60% 5 $12 \mathrm{m/s}$ 6 50% $10 \mathrm{m/s}$

 $8 \mathrm{m/s}$

40%

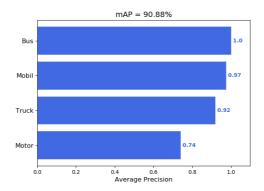
Tabel 4.2: Tabel Pengujian Kecepatan Terbang UAV

Berdasarkan tabel tersebut, UAV ini memiliki kecepatan maksimal sebesar 22 meter/detik dan $stall\ speed$ atau kecepatan minimal sebesar 8 meter/detik. Bila diinginkan durasi terbang yang lebih lama maka dapat menambah kapasitas baterai yang digunakan. Namun menambah kapasitas baterai berarti menambah bobot yang dibawa oleh UAV. Karena bobot UAV bertambah maka perlu dilakukan uji $wing\ loading\ untuk$ menentukan modifikasi luas sayap yang optimal. Semakin kecil $wing\ loading\ maka\ bebah\ motor\ brushless\ untuk\ mengangkat\ pesawat\ semakin kecil sehingga konsumsi baterai juga semakin kecil.$

4.3 Pengujian Deteksi Kendaraan

Dengan menggunakan dataset yang ada, model ini memiliki nilai mAP yang dijelaskan pada Gambar 4.5. Akurasi untuk mendeteksi kendaraan bus pada dataset adalah sebesar 100 persen. Sedangkan untuk mobil, akurasi yang didapat sebesar 97 persen. Akurasi untuk mendeteksi truck sebesar 92 persen. Akurasi deteksi motor mendapat nilai terkecil yaitu 74 persen, hal ini dikarenakan motor jika dilihat menggunakan kamera pada UAV akan tampak sangat kecil dan biasanya hanya berupa objek hitam kecil. Proses deteksi kendaraan dilakukan secara offboard yaitu seluruh proses dilakukan secara realtime pada PC atau laptop di darat.

Dengan skor mAP yang telah diperoleh, model harus diujikan



Gambar 4.5: Akurasi dari Model yang Dibuat

pada data testing. Tujuannya adalah untuk melihat apakah nilai mAP tersebut terbukti pada sumber gambar lain. Pengujian ini dibagi menjadi 2 bagian yaitu:

- 1. Pengujian Deteksi Kendaraan dari Video Playback UAV
- 2. Pengujian Deteksi Kendaraan dari Video Streaming VTX
- 3. Pengujian Deteksi Kendaraan pada Tingkat Pencahayaan yang Berbeda
- Pengujian Deteksi Kendaraan dengan Ketinggian dan Kecepatan Terbang UAV yang Bervariasi

4.3.1 Pengujian Deteksi Kendaraan dari Video $Playback\ UAV$

Hasil video streaming disimpan dan dijadikan data testing untuk model pendeteksi kendaraan yang telah dibuat. Video streaming diambil dengan UAV yang terbang di sekitaran wilayah Pusat Robotika ITS. Model pendeteksi kendaraan yang dibuat sudah dapat mendeteksi dengan benar mobil yang berada di jalan sekitar gedung Robotika ITS.

Uji coba terbang dilakukan sebanyak 2 kali di wilayah kampus ITS. *UAV* terbang melalui ruas jalan di depan gedung Informatika ITS dan ruas jalan di depan gedung Robotika ITS. Pada percobaan terbang pertama, hanya ada dua jenis kendaraan yang lewat yaitu mobil dan motor. Jumlah motor pada jalan dengan hitung manual

adalah 13 sedangkan jumlah mobil jika dihitung manual adalah sebanyak 4. Akurasi deteksi pada bercobaan pertama dijelaskan pada Tabel 4.3 dan hasil deteksinya dapat dilihat pada Gambar 4.6.



Gambar 4.6: Percobaan Terbang 1

Tabel 4.3: Akurasi Deteksi pada Percobaan Terbang 1

| Kendaraan | TP | TN | FP | FN |
|-----------|----|----|----|----|
| Mobil | 4 | 11 | 1 | 0 |
| Motor | 11 | 4 | 0 | 2 |

Pada percobaan terbang kedua, kendaraan yang lewat sama seperti percobaan terbang sebelumnya yaitu mobil dan motor. Jumlah mobil pada percobaan terbang kedua ini lebih sedikit dibandingkan percobaan pertama. Total kendaraan yang terlihat selema percobaan kedua adalah 3 mobil dan 12 motor. Akurasi deteksi pada bercobaan kedua dijelaskan pada Tabel 4.2 dan hasil deteksinya dapat dilihat pada Gambar 4.7.



Gambar 4.7: Percobaan Terbang 2

Tabel 4.4: Akurasi Deteksi pada Percobaan Terbang 2

| Kendaraan | TP | TN | FP | FN |
|-----------|----|----|----|----|
| Mobil | 4 | 11 | 1 | 0 |
| Motor | 11 | 4 | 0 | 2 |

4.3.2 Pengujian Deteksi Kendaraan dari Video $Stre-aming\ VTX$

Pada implementasinya, model yang sudah dibuat harus bisa mendeteksi kendaraan dari citra streaming video yang dikirim dari UAV menggunakan modul VTX. Modul ini juga memiliki receiver yang dipasang pada Laptop via USB. Modul receiver akan terbaca sebagai webcam sehingga dapat dengan mudah diimplementasikan pada sistem. Namun kekurangan dari modul ini adalah kualitas gambar yang dihasilkan setara dengan kamera dengan 0.2 Mega Pixel dan resolusi maksimal adalah 640x480 pixel. Hasil dari penerapan sistem pendeteksi kendaraan pada video streaming yang dikirim

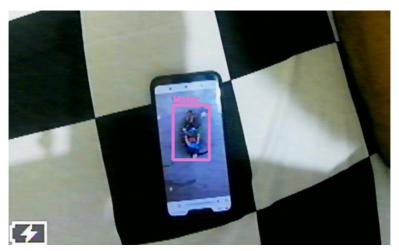
melalui VTX dapat dilihat pada Gambar 4.8.



Gambar 4.8: Deteksi Kendaraan pada Video Streaming

Terlihat pada gambar tersebut terdapat sebuah mobil berwarna putih yang parkir di bawah pohon. Mobil tersebut tidak terlihat jelas namun sistem dapat mengidentifikasikan mobil yang terlihat sebagi objek mobil. Beberapa pengujian tidak dapat dilakukan karena adanya wabah corona sehingga percobaan selantutnya menggunakan gambar kendaraan tampak atas yang dilihat melalui video streaming seperti pada Gambar 4.9

Pada gambar tersebut, terlihat gambar motor pada *smart-phone* dapat terdeteksi sebagai objek motor. Jika gambar posisinya lebih jauh, maka motor tidak terdeteksi. hal ini dikarenakan cahaya layar dari *smartphone* membuat gambar motor menjadi samar saat dijauhkan. Motor merupakan objek yang paling sulit di deteksi karena fiturnya yang sedikit dan warnanya menyerupai warna jalan ketika dilihat dari atas.



Gambar 4.9: Deteksi Motor pada Video Streaming

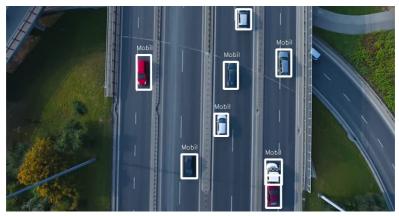


Gambar 4.10: Deteksi Mobil pada Video Streaming

4.3.3 Pengujian Deteksi Kendaraan pada Tingkat Pencahayaan yang Berbeda

Pada pengujian ini dilakukan 4 percobaan deteksi kendaraan pada kondisi pagi hari, siang hari, sore hari, dan malam hari.

Tujuan dari pengujian ini adalah untuk mengetahui kondisi terbaik untuk operasional sistem pendeteksi dan monitoring kemacetan. Gambar diundih via Google image karena kondisi saat ini yang tidak memungkinkan untuk mengambil data secara langsung. Hasil percobaan deteksi kendaraan pada kondisi pagi hari dapat dilihat pada Gambar 4.11.



Gambar 4.11: Hasil Deteksi Kendaraan pada Pagi Hari

Pada Gambar 4.11 dapat dilihat bahwa sistem dapat mendeteksi kendaraan dengan baik pada pagi hari dikarenakan kendaraan masih tampak jelas. Terdapat total 8 mobil pada gambar tersebut dan sistem berhasil mendeteksi semua mobil tersebut walaupun ada 2 mobil yang memiliki warna yang serupa dengan jalan. Hasil percobaan deteksi kendaraan pada kondisi siang hari dapat dilihat pada Gambar 4.12.

Pada Gambar 4.12 terlihat bahwa sistem pendeteksi kendaraan yang telah dibuat mampu mendeteksi kendaraan dengan baik dikarenakan banyaknya cahaya pada siang hari yang membuat kendaraan terlihat sangat jelas. Terdapat total 2 mobil pada gambar tersebut dan sistem berhasil mendeteksi semua mobil tersebut. Hasil percobaan deteksi kendaraan pada kondisi sore hari dapat dilihat pada Gambar 4.13.



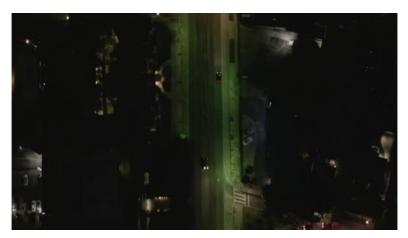
Gambar 4.12: Hasil Deteksi Kendaraan pada Siang Hari



Gambar 4.13: Hasil Deteksi Kendaraan pada Sore Hari

Gambar 4.13 mewakili kondisi jalan pada sore menjelang malam dimana masih terdapat sedikit cahaya. Pada gambar terdapat

total 6 mobil, 3 motor, dan 1 bus. Sistem dapat mendeteksi 4 mobil dan 1 bus. Terdapat 2 kendaraan mobil berwarna hitam yang masih dapat terdeteksi oleh sistem pendeteksi kendaraan. namun ada satu mobil dan 2 motor yang tidak terdeteksi. Hasil percobaan deteksi kendaraan pada kondisi malam hari dapat dilihat pada Gambar 4.14.



Gambar 4.14: Hasil Deteksi Kendaraan pada Malam Hari

Pada Gambar 4.14 terdapat 2 buah mobil yang terlihat secara visual. Namun sistem pendeteksi kendaraan tidak dapat mendeteksi kendaraan tersebut dikarenakan warna mobil pada gambar membaur dengan warna jalan. Selain itu, UAV juga tidak dapat dioperasikan pada malam hari dikarenakan alasan keamanan. Dari hasil yang didapat dapat disimpulkan bahwa sistem pendeteksi kendaraan dapat bekerja dengan baik pada pagi, siang, dan sore hari.

4.3.4 Pengujian Deteksi Kendaraan dengan Ketinggian dan Kecepatan Terbang UAV yang Bervariasi

Sistem pendeteksi kendaraan juga harus bisa mendeteksi kendaraan dengan benar pada ketinggian terbang UAV yang bervariasi. Semakin tinggi UAV terbang maka semakin luas jalan yang

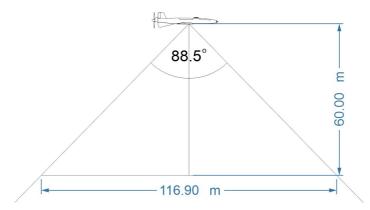
dapat dimonitoring. Namun hal ini juga diikuti dengan semakin kecilnya gambar kendaraan yang ditangkap oleh camera. Kecepatan terbang dari UAV juga dapat mempengaruhi kualitas hasil deteksi kendaraan. Maka dari itu, tujuan dari pengujian ini adalah untuk menentukan ketinggian dan kecepatan terbang optimal dari UAV agar dapat mendeteksi kendaraan dengan baik dan dapat memonitoring jalan dengan maksimal. Untuk mencari ketinggian terbang optimal dari UAV, dilakukan 7 percobaan yaitu pada ketinggian terbang 40 meter, 50 meter, 60 meter, 70 meter, 80 meter, 90 meter, dan 100 meter. Kecepatan rata rata terbang yang digunakan pada pengujian ini adalah sebesar 10 meter/detik. Ketinggian 100 meter digunakan sebagai ketinggian maksimal karena ketinggian ini masih tidak melanggar peraturan operasional drone di Indonesia yang membatasi ketinggian maksimal untuk operasional drone atau UAV maksimal 120 meter. Ketinggian terbang atau titik 0 meter dihitung dari tanah atau ketinggian UAV pada saat dihidupkan. Hasil percobaan tersebut dapat dilihat pada Tabel 4.5.

Tabel 4.5: Tabel Uji Coba Ketinggian pada Sistem Pendeteksi Kendaraan

| Kendaraan | Ketinggian Terbang UAV | | | | | | |
|-----------|------------------------|-----|-----|-----------------|-----|-----|------|
| Rendaraan | 40m | 50m | 60m | $70 \mathrm{m}$ | 80m | 90m | 100m |
| Mobil | Т | Т | Т | Т | - | - | - |
| Motor | Т | Т | Т | - | - | - | - |
| Bus | Т | Т | Т | T | Т | Т | - |
| Truck | Т | Т | Т | Т | Т | - | - |

Pada Tabel 4.5, dapat terdeteksinya suatu kendaraan ditandai dengan huruf "T". Mobil dapat terdeteksi pada ketinggian 40 meter sampai 70 meter. Motor dapat terdeteksi dengan maksimal pada ketinggan 40 meter sampai 60 meter. Untuk kendaraan bus, sistem pendeteksi dapat mendeteksi dengan akurat dari ketinggian 40 meter sampai 90 meter. Truck dapat terdeteksi dari ketinggian 40 meter sampai 80 meter. Pada ketinggian 100 meter, sistem pendeteksi kendaraan sudah tidak dapat mendeteksi kendaraan. Dari percobaan ini, diperoleh ketinggian maksimal dari UAV untuk mendeteksi semua jenis kendaraan secara optimal setinggi 60 meter. Dengan

demikian, sesuai dengan cara perhitungan panjang jalan pada bab sebelumnya, panjang jalan maksimal yang dapat dimonitoring oleh UAV adalah sepanjang 116.9 meter dan luas maksimal jalan yang dapat dimonitoring adalah 818 meter persegi (dengan lebar jalan = 7 meter) . Cara perhitungan panjang jalan dijelasikan dalam Gambar 4.15.



Gambar 4.15: Panjang Jalan Maksimal yang Dapat Dimonitoring oleh UAV

Sedangkan untuk pengujian kecepatan optimal dari UAV, dilakukan 6 percobaan dengan menggunakan ketinggian optimal UAV yaitu 60 meter. Percobaan dilakukan dengan kecepatan rata-rata 8 m/s, 10 m/s, 12 m/s, 15 m/s, 20 m/s, dan 22 m/s. Kecepatan maksimal 22 meter/detik dipilih karena merupakan kecepatan maksimal dari UAV. Sedangkan kecepatan 8 meter/detik merupakan stall speed atau kecepatan minimum dari UAV. Hasil percobaan tersebut dapat dilihat pada Tabel 4.6.

Pada Tabel 4.6, dapat terdeteksinya suatu kendaraan ditandai dengan huruf "T". Kendaraan mobil dapat terdeteksi pada kecepatan terbang 8 sampai 15 meter/detik. Motor hanya dapat terdeteksi dengan baik pada kecepatan 8 sampai 12 meter/detik. Bus dapat terdeteksi dengan baik pada kecepatan 8 sampai 20 meter/detik. Truck hanya terdeteksi dengan baik pada kecepatan 8 sampai 12 meter/detik. Dari percobaan ini, diperoleh kecepatan optimal dari

Tabel 4.6: Tabel Uji Coba Kecepatan Terbang pada Sistem Pendeteksi Kendaraan

| Kendaraan | Kecepatan UAV | | | | | |
|-----------|------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Rendaraan | $8 \mathrm{m/s}$ | $10 \mathrm{m/s}$ | $12 \mathrm{m/s}$ | $15 \mathrm{m/s}$ | $20 \mathrm{m/s}$ | $22 \mathrm{m/s}$ |
| Mobil | Т | Т | Т | Т | - | - |
| Motor | Т | Т | Т | - | - | - |
| Bus | Т | Т | Т | Т | Т | - |
| Truck | Т | Т | Т | - | - | - |

UAV adalah sebesar 12meter/detik.

4.4 Pengujian Luas Area Optimal yang dapat Dimonitoring UAV

Pengujian ini bertujuan untuk mengetahui seberapa luas daya jelajah dari UAV yang digunakan. Jarak tempuh optimal dari UAV ini didapat di estimasi menggunakan rumus berikut :

$$Jarak = (Kecepatan\ Optimal \times Durasi\ Terbang)$$
 (4.1)

Dimana kecepatan optimal UAV yang diperoleh dari percobaan sebelumnya adalah 12m/s dan durasi terbang yang sesuai dengan kecepatan tersebut pada Tabel 4.1 adalah 15 menit 12 detik. Perhitungan tersebut menghasilkan jarak tempuh optimal sebesar 10.944 meter. Maka, UAV ini dapat memonitoring jalan secara optimal sepanjang 10.944 meter. Jika jalan diasumsikan membentuk sebuah area persegi, maka estimasi luas area yang dapat dimonitoring adalah 2.736 x 2.736 meter.

4.5 Pengujian Sistem Pendeteksi Kemacetan

Pengujian ini dilakukan untuk menguji metode WTDC. Namun metode ini tidak dapat diterapkan secara optimal menggunakan data yang telah diambil karena tidak ada kondisi macet pada data tersebut mengingat kondisi saat ini yang tidak kondusif. Selain itu data yang telah diambil tidak mengandung jenis kendaraan bus dan truck. Contoh sistem pendeteksi kemacetan dapat dilihat pada Gambar 4.16.



Gambar 4.16: Sistem Pendeteksi Kemacetan

Sistem ini dapat menampilkan data kendaraan beserta status kemacetan secara detail. Status kemacetan akan "lancar" ketika nilai kepadatan jalan lebih dari 12.8 dan akan "macet" ketika nilai kepadatan kurang dari 12.8. Semakin kecil nilai maka semakin padat jalan tersebut. Jumlah kendaran juga ditampilkan dalam sistem ini. Data tersebut disajikan pada frame di pojok kiri atas seperti pada Gambar 4.17. Sayangnya karena kondisi saat ini yang tidak kondusif maka data pengujian selanjutnya diganti dengan video yanng di dapat dari Youtube. Namun dibutuhkan kalibrasi ulang pada perhitungan kepadatan jalan karena luas jalan yang berbeda beda di setiap video. Pengujian akan dilakukan menggunakan 3 video dengan kepadatan jalan yang berbeda beda mulai dari jalanan lancar, jalanan ramai, dan jalanan macet.

Pada pengujian pertama menggunakan video dari *Youtube*, secara visual kondisi jalanan pada video tersebut adalah lancar. Dengan menggunakan sistem pendeteksi kemacetan yang telah dibuat, didapati bahwa kondisi jalan itu adalah lancar. Hasil dari sistem pendeteksi pada video pertama dapat dilihat pada Gambar 4.18.

Pada pengujian kedua, video yang dicoba memiliki kondisi jalan yang ramai secara visual. Dengan menggunakan sistem pendeteksi kemacetan yang telah dibuat, hasil yang di dapat adalah jalan



Gambar 4.17: Data yang Disajikan



Gambar 4.18: Percobaan pada Video Youtube 1

tersebut terdeteksi sebagai padat. Hasil dari penerapan sistem tersebut dapat dilihat pada Gambar 4.19.

Pada pengujian ketiga, video yang dicoba memiliki kondisi jalan yang padat secara visual. Dengan menggunakan sistem pen-



Gambar 4.19: Percobaan pada Video Youtube 2

deteksi kemacetan yang telah dibuat, hasil yang di dapat adalah jalan tersebut terdeteksi sebagai macet. Hasil dari penerapan sistem pendeteksi dan monitoring kemacetan tersebut dapat dilihat pada Gambar 4.20.



Gambar 4.20: Percobaan pada Video Youtube 3

Dari percobaan tersebut, sistem sudah dapat menentukan status kemacetan sesuai dengan hasil perhitungan kepadatan jalan pada metode yang telah dibuat sebelumnya. Data jumlah kendaraan juga sudah dapat ditampilkan sehingga memudahkan pada saat monitoring kemacetan pada suatu jalan.

BAB 5 PENUTUP

5.1 Kesimpulan

Dari hasil pengujian yang sudah dilakukan dapat ditarik beberapa kesimpulan sebagai berikut:

- 1. UAV yang dibuat dapat terbang sesuai jalur yang dibuat dan memiliki durasi terbang 15 menit dengan coverage area maksimal seluas $2.7 \times 2.7 \text{ km}$.
- Sistem pendeteksi kendaraan yang dibuat dapat mendeteksi 4 macam kencaraan yaitu mobil, motor, bus, dan truck dengan mAP sebesar 90%. Akurasi sistem deteksi pada saat mendeteksi kendaran dari kamera realtime UAV adalah sebesar 80%.
- 3. Ketinggian optimal dari *UAV* yang dibuat agar dapat mendeteksi kendaraan dengan baik adalah 60 meter dengan kecepatan terbang optimal 12 meter/detik.
- 4. Rotasi kendaraan pada saat proses training berpengaruh pada hasil deteksi kendaraan. Jika dataset yang digunakan hanya mengandung gambar kendaraan vertikal, maka sistem pendeteksi tidak bisa mendeteksi gambar kendaraan horizontal.
- 5. Berdasarkan hasil pengujian, Sistem pendeteksi kendaraan dapat mendeteksi 8 dari 8 kendaraan yang tampak pada jalanan pagi hari dan 2 dari 2 kendaraan yang tampak pada jalan siang hari. Sedangkan pada sore hari berdasarkan hasil deteksi pada gambar yang diujikan sistem hanya dapat mendeteksi 5 dari 9 kendaraan yang tampak pada gambar. Pada malam hari sistem tidak dapat mendeteksi kendaraan apapun pada gambar.
- 6. Sistem pendeteksi dan monitoring kemacetan dapat mendeteksi kemacetan dari jumlah kendaraan yang di deteksi secara realtime dengan fps yang didapat berkisar antara 26 sampai 30~fps pada resolusi tamplan 640~x 480~px.

5.2 Saran

Untuk pengembangan penelitian selanjutnya terdapat beberapa saran sebagai berikut :

- 1. Akuisisi data untuk dataset harus langsung dari hasil video yang diambil menggunakan UAV agar hasil deteksi dapat lebih akurat dan data yang digunakan untuk dataset juga lebih sedikit.
- Jumlah epoch yang semakin banyak tidak menjamin hasil training semakin akurat. Jumlah epoch yang tepat dapat membuat waktu dalam proses training data lebih cepat dan menghasilkan nilai loss yang optimal.
- Gunakan YOLO versi terbaru agar kualitas deteksi dapat lebih baik. Versi awal dari YOLO tidak cocok digunakan untuk mendeteksi banyak objek dalam suatu gambar.
- 4. Dibutuhkan sistem transmisi data video yang dapat menampilkan video dengan resolusi yang lebih baik daripada yang digunakan pada penelitian ini agar hasil monitoring yang didapat lebih jelas dan memudahkan saat proses deteksi kendaraan.
- 5. Untuk menambah durasi terbang dari UAV, dibutuhkan kapasitas baterai lipo yang lebih besar daripada yang digunakan pada pengujian ini. Dikarenakan ukuran baterai yang semakin besar, ukuran UAV juga harus diperbesar untuk menunjang hal tersebut. sebagai acuan lebar sayap harus diatas 1.8 meter untuk membawa payload tambahan seberat 500 gram.

DAFTAR PUSTAKA

- [1] S. Maji, "Traffic congestion and possible solutions a case study of asansol," Quest Journals, pp. 42–46, 2017. (Dikutip pada halaman 1, 5).
- [2] H. Y. LiangWang, Fangliang Chen, "Detecting and tracking vehicles in traffic by unmanned aerial vehicles," <u>Automation in Construction 72</u>, pp. 294–308, 2016. (Dikutip pada halaman 1, 5).
- [3] R. Yanushevsky, "Guidance of unmanned aerial vehicle," <u>CRC Press</u>, pp. 1–5, 2011. (Dikutip pada halaman 2, 5).
- [4] A. Alkaabi, "Applications of unmanned aerial vehicle (uav) technology for research and education in uae," <u>International Journal of Social Sciences Arts and Humanities</u>, pp. 4–11, 2017. (Dikutip pada halaman 2, 5).
- [5] N. H. D. T. M. H. Nguyen Viet Hung, Le Chung Tran and N. T. Dzung, "A traffic monitoring system for a mixed traffic flow via road estimation and analysis," pp. 375–378, IEEE, 2016. (Dikutip pada halaman 6).
- [6] S. B. Prakoso, <u>Pengembangan Sistem Penghitung Jumlah Kendaraan Memanfaatkan Citra Aerial yang Diambil dengan Pesawat Tanpa Awak</u>. Institut Teknologi Sepuluh Nopember, 2018. (Dikutip pada halaman 6).
- [7] R. G. A. F. Joseph Redmon, Santosh Divvala, "You only look once: Unied,real-time object detection," <u>Available online: https://pjreddie.com/media/files/papers/yolo1.pdf</u>, 2016. (Dikutip pada halaman 7).
- [8] T. H. Trieu, "Darkflow," <u>GitHub Repository</u>. Available online: https://github.com/thtrieu/darkflow. (Dikutip pada halaman 9).
- [9] M. Oborne, "Mission planner." https://ardupilot.org/planner, 2019. (Dikutip pada halaman 12).

 $Halaman\ ini\ sengaja\ dikosongkan$

BIOGRAFI PENULIS



Putu Wisnu Bhaskara Putrawan, lahir pada 2 April 1998 di Jakarta. Lulus dari SMA Negeri 1 Sidoarjo pada tahun 2016 dan melanjutkan pendidikan S1 di Departemen Teknik Komputer, FTEIC-ITS Surabaya. Aktif menjadi anggota Tim Robot Terbang Bayucaraka ITS 2017-2020. Tertarik pada bidang UAV, Robotika, dan Computer Vision. Selama masa kuliah penulis banyak mengikuti lomba mulai dari tingkat Nasional sampai Internasional. Pernah menjadi juara 1 pada

TUBITAK UAV Competition di Turkey pada tahun 2019. Bagi pembaca yang memiliki kritik, saran atau pertanyaan mengenai tugas akhir ini dapat menghubungi penulis melalui email wisnubodo@gmail.com

 $Halaman\ ini\ sengaja\ dikosongkan$