

**EKSPLOITASI PERFORMA MODEL *EFFICIENTDET* BERBASIS  
KOMPUTASI *PERSONAL COMPUTER* UNTUK DETEKSI OBJEK  
*UNMANNED AERIAL VEHICLE***

***PERFORMANCE EXPLOITATION IN PERSONAL COMPUTER  
COMPUTING COST MODEL-BASED EFFICIENTDET FOR UNMANNED  
AERIAL VEHICLE OBJECT DETECTION***

**TUGAS AKHIR**

Disusun sebagai syarat mata kuliah Tugas Akhir pada Program Studi S1 Teknik  
Telekomunikasi

Oleh  
**IGA NARENDRA PRAMAWIJAYA**  
**1101184256**



**Telkom  
University**

**FAKULTAS TEKNIK ELEKTRO  
UNIVERSITAS TELKOM  
BANDUNG  
2022**

**LEMBAR PENGESAHAN**

**TUGAS AKHIR**

**EKSPLOITASI PERFORMA MODEL EFFICIENTDET BERBASIS KOMPUTASI**

**PERSONAL COMPUTER UNTUK DETEKSI OBJEK UNMANNED AERIAL**

**VEHICLE**

**PERFORMANCE EXPLOITATION IN PERSONAL COMPUTER COMPUTING**

**COST MODEL-BASED EFFICIENTDET FOR UNMANNED AERIAL VEHICLE**

**OBJECT DETECTION**

**Telah disetujui dan disahkan sebagai Tugas Akhir**

**Program S1 Teknik Telekomunikasi**

**Fakultas Teknik Elektro**

**Universitas Telkom**

**Bandung**

**Disusun oleh:**

**Iga Narendra Pramawijaya**

**1101184256**

**Bandung, 13 April 2022**

**Menyetujui,**

**Pembimbing I**

**Pembimbing II**

**Suryo Adhi Wibowo, S.T., M.T., Ph.D. Dr. Koredianto Usman, S.T., M.Sc.**

**Nik pbb 1**

**Nik pbb 2**

## **LEMBAR PERNYATAAN ORISINALITAS**

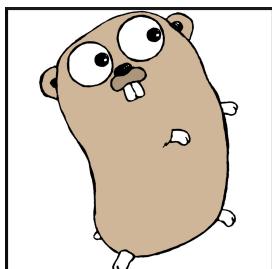
Nama : Iga Narendra Pramawijaya  
NIM : 1101184256  
Alamat : Kost Violet C3.07, Jl. Adhyaksa Raya, Bandung, Jawa Barat 40267  
No. Telepon : +6285339435369  
Email : hi@iganarendra.my.id

Menyatakan bahwa Tugas Akhir ini merupakan karya orisinal saya sendiri, dengan judul :

**EKSPLORASI PERFORMA MODEL *EFFICIENTDET* BERBASIS  
KOMPUTASI PERSONAL COMPUTER UNTUK DETEKSI OBJEK  
*UNMANNED AERIAL VEHICLE*  
*PERFORMANCE EXPLOITATION IN PERSONAL COMPUTER*  
*COMPUTING COST MODEL-BASED *EFFICIENTDET* FOR UNMANNED*  
*AERIAL VEHICLE OBJECT DETECTION***

Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhan kepadanya apabila kemudian ditemukan adanya pelanggaran terhadap kejujuran akademik atau etika keilmuan dalam karya ini, atau ditemukan bukti yang menunjukkan ketidakaslian karya ini.

Bandung, 13 April 2022



---

Iga Narendra Pramawijaya

---

1101184256

## ABSTRAK

Kendaraan udara tanpa awak atau *Unmanned Aerial Vehicles* (UAV) dilengkapi kamera dengan resolusi tinggi. Kamera pada UAV dapat mengambil foto atau citra UAV. Sebelumnya, [1] Zhu, dkk. merangkum model deteksi objek citra UAV. Berbagai model tersebut memakan beban komputasi tinggi. Model jaringan syaraf tiruan yang tidak dilatih menggunakan citra UAV akan menghasilkan performa yang buruk. Model deteksi objek yang dilatih kembali menggunakan citra UAV akan memakan beban komputasi yang tinggi. Model pengenalan objek *EfficientDet* lebih ringan secara komputasi. Maka dari itu penulis ingin meneliti peforma model pengenalan objek beban komputasi rendah *EfficientDet* pada citra UAV.

Dalam tugas akhir ini, sebuah sistem deteksi objek pada citra UAV yang dilatih menggunakan komputer konvensional akan dirancang untuk mendeteksi 10 kelas objek dengan menggunakan model *EfficientDet* versi D0. Setelah data didapatkan maka akan dilakukan *preprocessing* berupa konversi anotasi. Selanjutnya akan dilakukan proses pelatihan model. Setiap pelatihan, model akan diuji sehingga menghasilkan nilai validasi. Nilai validasi pada pelatihan terakhir yang akan dianalisa sebagai patokan performa.

Tugas akhir ini menggunakan himpunan data VisDrone yang terdiri atas 6471 gambar data latih dan 548 gambar data uji. Seluruh gambar berukuran 960×540 piksel. Tolak ukur performa model deteksi objek adalah nilai *average precision* (AP) area besar, AP nilai IoU 0,5; 0,75, dan *average recall* (AR) dengan maksimal deteksi 1, 10, dan 100. Model *EfficientDet d0* melampaui ARmax1 model TridentNet pada penelitian sebelumnya dengan 2,33%.

**Kata Kunci :** *EfficientDet*, Citra UAV, Beban Komputasi Rendah.

## ABSTRACT

Unmanned Aerial Vehicles are equipped with high-resolution cameras. The camera on the UAV can take UAV photos. Zhu, et al. [1] noveled object detection models for UAV images, but available models took high computational cost. Artificial neural network models not trained using the UAV image will result in poor performance. When object detection models retrained using UAV images, it will take up a high computational cost. EfficientDet object detection model is significantly lower in computing cost. Therefore, the author wants to examine the performance of the low computing cost model-based EfficientDet on UAV images.

In this thesis, an object detection system on UAV imagery is trained using conventional computers will be designed to detect 10 classes object using the D0 version of the EfficientDet model. After the data is obtained then preprocessing will be carried out in the form of annotation conversion. Next, model training process will be carried out several times. In each training, the model will be tested so as to produce a validation value. The last validation value will be analyzed as a benchmark for performance.

This thesis uses the VisDrone dataset consisting of 6471 images as training data and 548 images of test data with  $960 \times 540$  pixels resolution. In this thesis, large area average precision (AP) value, AP IoU value 0.5, 0.75, and average recall (AR) with maximum detection of 1, 10, and 100 will be benchmarks indicating model performance. As a result, EfficientDet d0 model surpassed TridentNet model according to latest research with ARmax1 value of 2.33%.

**Key Word :** *EfficientDet, UAV Images, Low Computing Cost.*

## KATA PENGANTAR

Puji dan syukur kepada Tuhan yang Maha Esa atas segala kasih dan anugerah yang telah dilimpahkan kepada penulis, sehingga Tugas Akhir yang berjudul **Deteksi Objek pada Citra Unmanned Aerial Vehicle menggunakan EfficientDet.** Tugas akhir ini disusun sebagai syarat untuk menempuh gelar sarjana dan menyelesaikan pendidikan pada program studi S1 Teknik Telkomunikasi Universitas Telkom, Bandung.

Pada penulisan tugas akhir ini, penulis menyadari bahwa masih terdapat banyak kekurangan karena keterbatasan ilmu yang dimiliki. Segala kritik dan saran yang bersiat membangun sangat diterima guna memperbaiki penulisan tugas akhir ini.

Sebagai penutup, penulis memohon maaf atas segala kesalahan yang penulis lakukan baik sengaja maupun yang tidak disengaja saat menyelesaikan Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi penulis, pembaca, dan semua kalangan di dunia pendidikan.

Bandung, 13 April 2022

Iga Narendra Pramawijaya

## **UCAPAN TERIMA KASIH**

Dalam penyusunan Tugas Akhir ini, Penulis ingin memberikan ucapan terima kasih yang sebesar-besarnya kepada seluruh pihak yang terlibat dalam pembuatan Tugas Akhir ini.

Bandung, 5 Pebruari 2021

Iga Narendra Pramawijaya

## **DAFTAR ISI**

**LEMBAR PENGESAHAN**

**LEMBAR PERNYATAAN ORISINALITAS**

**ABSTRAK**

iv

**KATA PENGANTAR**

vi

**UCAPAN TERIMA KASIH**

vii

**DAFTAR ISI**

viii

**DAFTAR GAMBAR**

xi

**DAFTAR SINGKATAN**

xiii

**I PENDAHULUAN**

1

1.1	Latar Belakang Masalah . . . . .	1
1.2	Rumusan Masalah . . . . .	2
1.3	Tujuan dan Manfaat . . . . .	2
1.4	Batasan Masalah . . . . .	2
1.5	Metode Penelitian . . . . .	3
1.6	Sistematika Penulisan . . . . .	4

**DAFTAR LAMPIRAN**

1

**II DASAR TEORI**

5

2.1	Citra digital dengan kanal warna RGB . . . . .	5
2.2	Citra <i>Unmanned Aerial Vehicle</i> (UAV) . . . . .	6

2.3	Jaringan Syaraf Tiruan ( <i>Artifical Neural Network</i> ) . . . . .	7
2.4	Deteksi objek . . . . .	9
2.5	<i>EfficientDet</i> . . . . .	10
<b>III PERANCANGAN SISTEM</b>		<b>12</b>
3.1	Rancangan Sistem . . . . .	12
3.1.1	Akuisisi Data . . . . .	14
3.1.2	<i>Preprocessing</i> . . . . .	14
3.1.3	Pelatihan Model . . . . .	14
3.1.4	Analisis Hasil . . . . .	15
3.2	Performansi Sistem . . . . .	15
3.2.1	<i>Average Precision</i> (AP) dan <i>Average Recall</i> (AR) . . . . .	15
3.3	Sistematika Data . . . . .	16
<b>IV ANALISIS SIMULASI SISTEM</b>		<b>17</b>
4.1	Analisis Hasil Deteksi Objek dengan Pelatihan sebanyak 10 <i>epochs</i>	17
4.1.1	Hasil <i>Average Precision</i> . . . . .	17
4.1.2	Hasil <i>Average Recall</i> . . . . .	18
4.2	Analisis Hasil Deteksi Objek dengan Pelatihan sebanyak 20 <i>epochs</i>	20
4.2.1	Hasil <i>Average Precision</i> . . . . .	20
4.2.2	Hasil <i>Average Recall</i> . . . . .	21
4.3	Analisis Hasil Deteksi Objek dengan Pelatihan sebanyak 30 <i>epochs</i>	23
4.3.1	Hasil <i>Average Precision</i> . . . . .	23
4.3.2	Hasil <i>Average Recall</i> . . . . .	25
4.4	Analisis Hasil Deteksi Objek dengan pelatihan sebanyak 40 <i>epochs</i>	26
4.4.1	Hasil <i>Average Precision</i> . . . . .	27
4.4.2	Hasil <i>Average Recall</i> . . . . .	28
4.5	Analisis Hasil Deteksi Objek dengan pelatihan sebanyak 50 <i>epochs</i>	29
4.5.1	Hasil <i>Average Precision</i> . . . . .	30

4.5.2    Hasil <i>Average Recall</i> . . . . .	31
4.6    Pembahasan . . . . .	33
<b>V KESIMPULAN DAN SARAN</b>	<b>34</b>
5.1    Kesimpulan . . . . .	34
5.2    Saran . . . . .	34
<b>DAFTAR PUSTAKA</b>	<b>35</b>
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

2.1	Contoh warna yang dihasilkan bila sebuah titik memiliki nilai RGB (205,100,50) (warna yang dihasilkan mungkin kurang akurat karena keterbatasan percetakan) . . . . .	5
2.2	Contoh gambar pada himpunan data <i>VisDrone</i> . . . . .	7
2.3	Cara kerja salah satu model deteksi objek (YOLO) . . . . .	10
2.4	Grafik kinerja dan FLOP dari berbagai model deteksi objek termasuk <i>EfficientDet</i> . . . . .	11
3.1	Diagram blok rancangan sistem. . . . .	12
3.2	Diagram alir sistem. . . . .	13
4.1	Skor AP50 setelah pelatihan sebanyak 10 epochs . . . . .	17
4.2	Skor AP75 setelah pelatihan sebanyak 10 epochs . . . . .	18
4.3	Skor API atau AP di area besar setelah pelatihan sebanyak 10 epochs	18
4.4	Skor ARmax1 setelah pelatihan sebanyak 10 epochs . . . . .	19
4.5	Skor ARmax10 setelah pelatihan sebanyak 10 epochs . . . . .	19
4.6	Skor ARmax100 setelah pelatihan sebanyak 10 epochs . . . . .	20
4.7	Skor AP50 setelah pelatihan sebanyak 20 epochs . . . . .	20
4.8	Skor AP75 setelah pelatihan sebanyak 20 epochs . . . . .	21
4.9	Skor API setelah pelatihan sebanyak 20 epochs . . . . .	21
4.10	Skor ARmax1 setelah pelatihan sebanyak 20 epochs . . . . .	22
4.11	Skor ARmax10 setelah pelatihan sebanyak 20 epochs . . . . .	22
4.12	Skor ARmax100 setelah pelatihan sebanyak 20 epochs . . . . .	23
4.13	Skor AP50 setelah pelatihan sebanyak 30 epochs . . . . .	23
4.14	Skor AP75 setelah pelatihan sebanyak 30 epochs . . . . .	24

4.15 Skor API setelah pelatihan sebanyak 30 epochs . . . . .	24
4.16 Skor ARmax1 setelah pelatihan sebanyak 30 epochs . . . . .	25
4.17 Skor ARmax10 setelah pelatihan sebanyak 30 epochs . . . . .	26
4.18 Skor ARmax100 setelah pelatihan sebanyak 30 epochs . . . . .	26
4.19 Skor AP50 setelah pelatihan sebanyak 40 epochs . . . . .	27
4.20 Skor AP75 setelah pelatihan sebanyak 40 epochs . . . . .	27
4.21 Skor API setelah pelatihan sebanyak 40 epochs . . . . .	28
4.22 Skor ARmax1 setelah pelatihan sebanyak 40 epochs . . . . .	28
4.23 Skor ARmax10 setelah pelatihan sebanyak 40 epochs . . . . .	29
4.24 Skor ARmax100 setelah pelatihan sebanyak 40 epochs . . . . .	29
4.25 Skor AP50 setelah pelatihan sebanyak 50 epochs . . . . .	30
4.26 Skor AP75 setelah pelatihan sebanyak 50 epochs . . . . .	31
4.27 Skor API setelah pelatihan sebanyak 50 epochs . . . . .	31
4.28 Skor ARmax1 setelah pelatihan sebanyak 50 epochs . . . . .	32
4.29 Skor ARmax10 setelah pelatihan sebanyak 50 epochs . . . . .	32
4.30 Skor ARmax100 setelah pelatihan sebanyak 50 epochs . . . . .	33
0.1 Bukti Persetujuan Pembimbing 1 (Suryo Adhi Wibowo, S.T., M.T., Ph.D) . . . . .	
0.2 Bukti Persetujuan Pembimbing 2 (Dr. Koredianto Usman, S.T., M.Sc)	

## **DAFTAR SINGKATAN**

UAV Unmanned Aerial Vehicle

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Kendaraan udara tanpa awak, atau *Unmanned Aerial Vehicles* (UAV) telah dilengkapi dengan berbagai instrumentasi. Berbagai instrumen tersebut digunakan untuk navigasi kendaraan itu sendiri, atau untuk pengambilan data dari jarak jauh. Pengambilan data menggunakan UAV sudah semakin sering dilakukan seiring perkembangan teknologi pada UAV dan meningkatnya kebutuhan pengambilan data dari tempat yang sulit dijangkau.

Salah satu instrumen yang populer digunakan pada UAV adalah kamera dengan resolusi tinggi. Kamera pada UAV ini dapat mengambil foto atau video yang disebut citra UAV. Citra UAV dapat diimplementasikan dalam berbagai hal. Pemantauan dan pengindraan jarak jauh [2] [3], mitigasi, dan pemetaan pasca bencana merupakan macam implementasi yang sering melibatkan citra dari UAV [4] [5]. Citra UAV memiliki berbagai karakteristik diantaranya (i) *ultra-high spatial resolution* [6], (ii) dipengaruhi oleh kondisi cuaca, dan (iii) dipengaruhi oleh ketinggian.

Model jaringan syaraf tiruan dapat membantu implementasi penggunaan citra UAV [7]. Model jaringan syaraf tiruan dapat dibentuk untuk mengenali objek-objek. Dari berbagai model pengenalan objek yang tersedia, *EfficientDet* lebih ringan secara komputasi dibandingkan model pengenalan objek yang lain [8]. Model jaringan syaraf tiruan khususnya pengenalan objek kendaraan dan manusia yang tersedia tidak dilatih menggunakan citra UAV, maka model yang tersedia tidak dihadapkan dengan berbagai kondisi unik dari karakteristik citra UAV, dan akan menghasilkan prediksi yang buruk. Pada penelitian sebelumnya, [1] Zhu, dkk. merangkum berbagai model deteksi yang telah dilatih kembali menggunakan citra UAV,

namun berbagai model tersebut memakan beban komputasi yang tinggi. Bila model deteksi objek yang sudah ada dilatih kembali menggunakan citra UAV, maka akan memakan beban komputasi yang tinggi karena tingginya resolusi dari citra UAV. Walau pada *EfficientDet* beban komputasi dapat ditekan, penelitian sebelumnya menggunakan sistem komputasi performa tinggi yang tidak terjangkau. Hal ini memungkinkan terjadinya penurunan performa saat model dilatih menggunakan sistem komputer konvensional. Maka dari itu penulis ingin meneliti peforma model pengenalan objek dengan beban komputasi rendah *EfficientDet* pada citra UAV.

## 1.2 Rumusan Masalah

Adapun permasalahan yang terjadi yaitu belum ada penelitian mengenai performansi model deteksi objek dengan sistem komputer konvensional menggunakan *Efficientdet* pada citra UAV.

## 1.3 Tujuan dan Manfaat

Tujuan dari Tugas Akhir ini adalah menimplementasi dan mananlisis performansi model deteksi objek *Efficientdet* pada citra UAV. Adapun manfaat dalam Tugas Akhir ini adalah:

1. Meningkatkan performa model deteksi objek rendah komputasi *Efficientdet* pada citra UAV
2. Dapat mengimplementasikan model deteksi objek *Efficientdet* untuk menge- nali berbagai objek pada citra UAV.

## 1.4 Batasan Masalah

Batasan masalah untuk membatasi penelitian ini adalah sebagai berikut:

1. Simulasi dan *retrain* menggunakan komputer jinjing (*laptop*) pribadi dengan spesifikasi:

- GPU Nvidia GeForce GTX 1660 Ti *with Max-Q Design* dengan memori sebesar 4 GB GDDR6 terdedikasi
- Prosesor 2,6 GHz (4,5 GHz turbo) Intel Core i7-9750H
- RAM 16 GB, DDR4 2933 MHz *dual channel*

2. Objek yang dideteksi dalam penelitian ini meliputi:

- pejalan kaki
- manusia
- sepeda
- mobil
- minibus
- bus
- truk
- sepeda motor
- sepeda motor roda tiga, dan
- sepeda roda tiga

3. Himpunan data yang digunakan dalam penelitian ini adalah *VisDrone* versi 2019 [1].

## 1.5 Metode Penelitian

Metode penelitian yang diterapkan dalam penyelesaian Tugas Akhir ini dengan melakukan simulasi, perhitungan *average recall* (AR) dan perhitungan *average precision* (AP) yang menunjukkan ketepatan model mendeteksi objek pada berbagai gambar yang diujicobakan. Semakin tinggi AP dan AR, semakin baik model deteksi objek.

## 1.6 Sistematika Penulisan

- **BAB I PENDAHULUAN**

Bab ini membahas motivasi utama penelitian ini, penelitian-penelitian lain terkait yang sudah pernah dilakukan sebelumnya, serta tujuan dari penelitian ini.

- **BAB II DASAR TEORI**

Bab ini membahas landasan teori dan literatur yang digunakan dalam proses penelitian analisis performansi dan implementasi model deteksi objek *EfficientDet* pada citra UAV.

- **BAB III PERANCANGAN SISTEM**

Bab ini berisi tahapan-tahapan yang dilakukan dalam proses penelitian berupa diagram alir penelitian, parameter yang menjadi referensi penetian, dan desain rancangan setiap skenario.

- **BAB IV ANALISIS SIMULASI SISTEM**

Bab ini berisi pembahasan hasil dari nilai *Average Recall (AR)* dan *Average Precision (AP)* setiap variasi skenario. Pada bab ini juga disertakan grafik untuk mempermudah proses analisis.

- **BAB V KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dan saran Tugas Akhir untuk pengembangan selanjutnya.

## BAB II

### DASAR TEORI

#### 2.1 Citra digital dengan kanal warna RGB

Dalam penelitian ini, citra UAV yang diteliti berbentuk citra digital dengan kanal warna merah, hijau, biru (*Red, Green, Blue* atau *RGB*). Citra digital dengan kanal warna *RGB* merupakan larik (*array*) dua dimensi (matriks) yang pada masing-masing elemennya memiliki tiga nilai warna, yaitu merah, hijau, dan biru. Setiap nilai warna merepresentasikan tingkat kekuatan warna pada titik tersebut. Contohnya bila pada sebuah titik memiliki nilai (205, 100, 50), berarti merah mendominasi warna pada titik tersebut, lalu diikuti warna hijau dan biru. Dapat dilihat pada Gambar 2.1, warna yang dihasilkan didominasi merah. Warna yang dihasilkan pada Gambar 2.1 tidak sepenuhnya merah, melainkan jingga. Hal ini disebabkan karena walaupun warna merah mendominasi dengan nilai 205, warna-warna lain (hijau dan biru) masih memiliki nilai dan memberikan kontribusi dalam campuran warna. Nilai warna citra pada penelitian ini bernilai 24 bit sehingga memiliki rentang 0 sampai 255 untuk masing-masing warna. Sehingga titik dengan nilai warna (0,0,0) berwarna hitam dan titik bernilai (255,255,255) berwarna putih.



**Gambar 2.1.** Contoh warna yang dihasilkan bila sebuah titik memiliki nilai *RGB* (205,100,50) (warna yang dihasilkan mungkin kurang akurat karena keterbatasan percetakan)

Suatu titik pada matriks citra disebut piksel, sehingga dapat dikatakan bahwa citra digital merupakan gabungan dari banyak piksel yang berbentuk matriks dua dimensi, dengan kedalaman kanal warna yang bervariasi. Keuntungan menggunakan skema RGB dibanding skema warna lain adalah setiap piksel bisa memiliki warna yang memiliki warna bervariasi dan dinamis namun dapat mudah diinterpretasi secara numerik, sehingga mempermudah pengiriman data citra melalui *pipeline* algoritma deteksi objek. Kekurangan yang umum dari skema RGB adalah kesulitan untuk *scaling* khususnya untuk citra yang tersusun atas piksel [9].

## 2.2 Citra *Unmanned Aerial Vehicle* (UAV)

Citra UAV dalam penelitian ini adalah gambar digital dengan kanal warna RGB yang diambil menggunakan kamera resolusi tinggi yang terdapat pada kendaraan udara tanpa awak (*Unmanned Aerial Vehicle* atau UAV). Pengambilan data menggunakan UAV sudah semakin sering dilakukan seiring perkembangan teknologi pada UAV dan meningkatnya kebutuhan pengambilan data dari tempat yang sulit dijangkau. Citra UAV dapat diimplementasikan dalam berbagai hal. Pemantauan dan pengindraan jarak jauh [2] [3], mitigasi, dan pemetaan pasca bencana merupakan macam implementasi yang sering melibatkan citra dari UAV [4] [5]. Citra UAV memiliki berbagai karakteristik diantaranya (i) *ultra-high spatial resolution* [6], (ii) dipengaruhi oleh kondisi cuaca, dan (iii) dipengaruhi oleh ketinggian. Pada penelitian ini, penulis akan menggunakan himpunan data *VisDrone* [1]. Himpunan data ini sebenarnya merupakan sebuah kompetisi untuk peneliti menunjukkan hasil terbaik mereka membuat atau memodifikasi model pengenalan objek untuk citra UAV. Penulis akan melihat dan membandingkan hasil dari penelitian ini dengan publikasi hasil kompetisi pada tahun 2018 [10]. Gambar 2.2 merupakan contoh gambar-gambar yang tersedia pada himpunan data *VisDrone*.



**Gambar 2.2.** Contoh gambar pada himpunan data *VisDrone*

### 2.3 Jaringan Syaraf Tiruan (*Artifical Neural Network*)

Jaringan syaraf tiruan adalah salah satu model pembelajaran mesin yang terinspirasi dari cara kerja otak manusia, dimana ribuan dari miliaran sel syaraf (neuron) yang saling terhubung memproses data secara paralel. Para peneliti menirukan neuron dengan membuat *node* titik pemrosesan data, dan memberikan pembebanan di masing-masing hubungan antar *node*, dan bias di masing-masing titik pemrosesan. Biasanya beban (*weight*) menggunakan operasi perkalian, dan bias menggunakan operasi penjumlahan. Jaringan syaraf tiruan terdiri atas lapisan *input*, lapisan tersembunyi (*hidden layer*) setebal satu sampai tiga lapis, dan lapisan *output*. Keluaran dari neuron  $i$  adalah  $h_i$  dengan persamaan:

$$h_i = \sigma\left(\sum_{j=1}^N V_{ij}x_j + T_i^{hid}\right), \quad (2.1)$$

- dimana  $\sigma(\cdot)$  disebut fungsi aktivasi atau fungsi transfer,  $N$  adalah jumlah *input* neuron,  $V_{ij}$  adalah beban,  $x_j$  adalah *input* menuju neuron, dan  $T_i^{hid}$  adalah *threshold* dari neuron-neuron yang tersembunyi. Fungsi dari fungsi transfer adalah agar *neural network* dapat mengerti dan menyelesaikan masalah yang non-linier (yang merupakan kelebihan dari model *perceptron*) dan untuk memberikan nilai permulaan kepada neuron sehingga jaringan tidak terpengaruh oleh neuron yang divergen

[11]. Jaringan syaraf tiruan juga dapat disebut *perceptron* multi lapis (*multilayer perceptron*) selama memiliki sifat *fully connected* [12]. Agar menghasilkan hasil prediksi yang optimal, nilai beban dan bias harus selalu disesuaikan. Penyesuaian itu didapat melalui pemrosesan secara berulang (rekursif) yang Watt, dkk [13] rumuskan sebagai berikut:

- Pilih jenis fungsi aktifasi  $\sigma(\cdot)$
- Hitung persamaan 2.1
- Bentuk hasil dari persamaan 2.1 dan fungsi  $\sigma()$  akan memicu nilai beban dan bias yang lebih baik, dengan melihat seberapa jauh beda hasil prediksi dengan hasil sebenarnya.

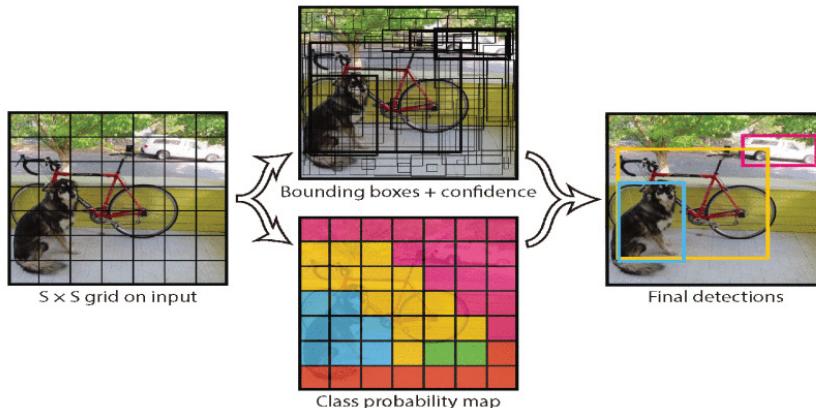
Jaringan syaraf tiruan sudah banyak diterapkan dalam berbagai jenis pengolahan informasi. Informasi yang diolah menggunakan jaringan syaraf tiruan biasanya yang memiliki pola dan kompleksitas tertentu seperti (i) data dalam bentuk tabel yang nantinya menghasilkan prediksi berupa angka (disebut sebagai model regresi) (ii) data (dapat berupa gambar, tabel, dan lainnya) yang nantinya menghasilkan prediksi mengelompokkan masukan ke dalam kelas tertentu (disebut model klasifikasi), (iii) data yang mengandung parameter waktu (disebut data *time series*) serta pola dan nantinya dapat memprediksi seluruh data masukan walaupun data masukan hanya sebagian kecil (disebut *model forecasting*), dan (iv) data berupa kalimat-kalimat yang dapat dicari maknanya (disebut *dataset kalimat*), yang biasanya untuk keperluan pemrosesan bahasa natual (*natural language processing*). Pada penelitian ini penulis akan menggunakan model klasifikasi yang tidak hanya mengelompokkan citra masukan, melainkan memberikan lokasi detil mengenai keberadaan objek di dalam citra. Selanjutnya hal ini disebut deteksi objek.

## 2.4 Deteksi objek

Deteksi objek adalah sebuah bagian dari pengenalan citra yang lebih kompleks, dimana model tidak hanya menklasifikasikan gambar, namun juga memperkirakan secara tepat kelas dan lokasi dari objek yang terdapat pada Gambar [14]. Tantangan ini sudah mulai dicobakan sejak tahun 1998 melalui penelitian Papageorgiou, dkk. Pada saat itu, dikembangkan teknik deteksi representasi *wavelet* yang dibuat him-punan bagian dan direpresentasi sebagai kelas, serta dibentuk menyerupai model masukan *support vector machine* [15]. Jaringan syaraf tiruan dengan pembelajaran yang dalam (*deep neural network*, yang selanjutnya akan disebut *deep learning*) telah menunjukkan hasil yang luar biasa dalam tantangan klasifikasi, dengan jaringan yang diberi nama *ImageNet* [16]. Hal ini menunjukkan potensi *deep learning* menyelesaikan tantangan selanjutnya yaitu deteksi objek. *ImageNet* secara tidak langsung menjadi populer sebagai *baseline* keberhasilan atau seberapa canggih sebuah model *deep learning* untuk gambar. Szegedy, dkk. dalam penelitian mereka [14] membuktikan bahwa *deep learning* tidak hanya mampu mempelajari fitur-fitur dari sebuah gambar untuk keperluan klasifikasi, namun juga mampu menangkap informasi geometrik yang kuat. Informasi geometrik diperlukan untuk memastikan keberadaan sebuah objek dalam gambar. Hal ini meningkatkan potensi *deep learning* untuk keperluan deteksi objek.

Melihat potensi ini, banyak peneliti tertarik mengembangkan sistem *deep learning* untuk pengenalan objek. Menurut Pathak, dkk. [17] sampai tahun 2018 sudah ada 14 kerangka kerja (*frameworks*), 6 layanan berbasis komputasi awan, dan banyak sekali model *deep learning* yang ditujukan sebagai detektor objek. Tidak terhitungnya jumlah model dikarenakan setiap peneliti atau bahkan insiyur dapat menyesuaikan lapisan-lapisan sesuai kebutuhan mereka. Ketika lapisan sudah ada yang berbeda, itu dapat disebut sebagai model *deep learning* yang baru. Kelemahan dari berbagai model yang sudah tersedia populer atau dikembangkan secara individu sebagian besar datang dari harga komputasi. Pada umumnya model *deep learning*

untuk deteksi objek mahal untuk dilatih karena membutuhkan *Floating Point Operations* (FLOP) yang tinggi. Hal ini dapat diatasi dengan *pretrained model*, yaitu model yang sudah dilatih sebelumnya menggunakan komputer yang sangat tinggi kemampuan komputasinya (*mainframe*, superkomputer, dan sejenisnya) menggunakan sebuah himpunan data tertentu (contoh: MS-COCO, himpunan data untuk ImageNet, CIFAR-100). Contoh jaringan yang memiliki akurasi tinggi namun cukup berat untuk dilatih kembali adalah YOLO (*You Only Look Once*)[18] yang cara kerjanya digambarkan pada Gambar 2.3. *Pretrained model* dapat digunakan langsung untuk aplikasi, karena sudah memiliki beban dan bias di masing-masing neuronnya. *Pretrained model* perlu dilatih kembali bila aplikasi model berbeda dengan jenis himpunan data pada proses *training* sebelumnya. Hal ini kembali mengungkit masalah beban komputasi yang tinggi, walaupun tidak setinggi sebelumnya.

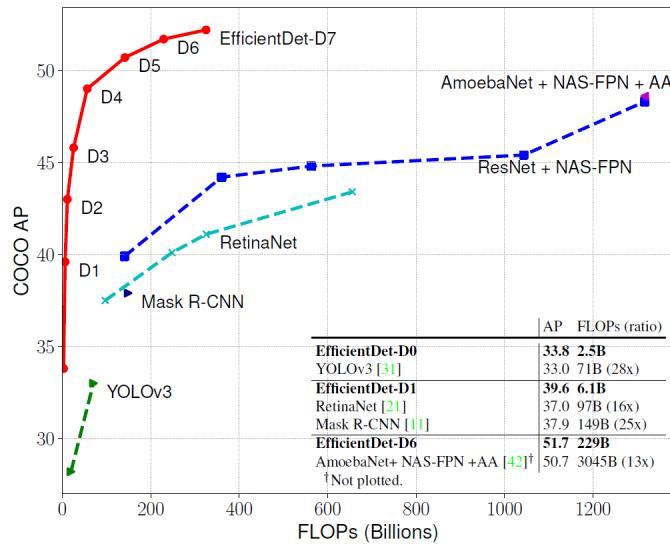


**Gambar 2.3.** Cara kerja salah satu model deteksi objek (YOLO)

## 2.5 *EfficientDet*

*EfficientDet* adalah salah satu model deteksi objek yang dikembangkan oleh Tan, dkk. yang memiliki keunggulan di bagian efisiensi model [8]. Model ini begitu menarik karena versi D7 nya dapat meraih akurasi yang mutakhir (*state-of-the-art*) sebesar 52.2% presisi rerata (*Average Precision (AP)*) pada himpunan data COCO *test-dev* dengan hanya 325 juta FLOP. Model *EfficientDet* berukuran 4-9 kali lebih kecil dibanding pendahulunya. Walaupun 325 juta FLOP terlihat besar, model

ini jumlah FLOP nya 13-42 kali lebih rendah dibandingkan dengan model-model deteksi objek *state-of-the-art* pendahulunya, salah satunya *AmoebaNet-based NAS-FPN detector*[8]. Model ini menggunakan jaringan piramida dua arah untuk melakukan ekstraksi ciri. Model piramida ini terdiri atas 8 tingkatan, dimana piramida dua arah (BiFPN) yang diadaptasi dari Chen, dkk.[19] diterapkan dari tingkatan ketiga sampai ketujuh. Jaringan piramida ini diperuntukan untuk membuat peta fitur yang lebih kecil namun lebih kaya akan informasi. Satu tingkatan piramida menuju lainnya dihubungkan menggunakan operasi konvolusi, sehingga ketika berpindah tingkatan, matriks peta fitur menjadi lebih kecil dimensinya, namun semakin padat informasinya. Selain itu, saat bergerak dari satu tingkatan menuju yang lainnya diberi pembebanan, artinya saat proses pembelajaran, jaringan *EfficientDet* akan mempelajari tingkatan mana yang bobotnya tinggi dan rendah. Sehingga bisa meningkatkan presisi melalui pemberian bobot yang lebih tinggi kepada tingkatan yang menghasilkan peta fitur yang representatif terhadap hasil prediksi. Gambar 2.4 menggambarkan hubungan performansi pada sumbu Y terhadap FLOP pada sumbu X [8].



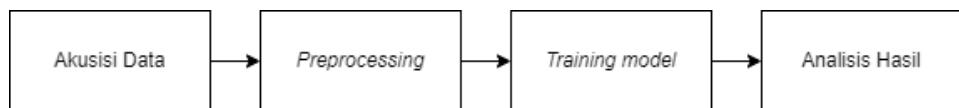
**Gambar 2.4.** Grafik kinerja dan FLOP dari berbagai model deteksi objek termasuk *EfficientDet*

## BAB III

### PERANCANGAN SISTEM

#### 3.1 Rancangan Sistem

Pada tugas akhir ini, sistem dirancang untuk mendekripsi objek-objek pada citra UAV. Sistem menggunakan model jaringan syaraf tiruan *EfficientDet-D0*. Gambaran secara umum rancangan sistem dapat dilihat pada Gambar 3.1.



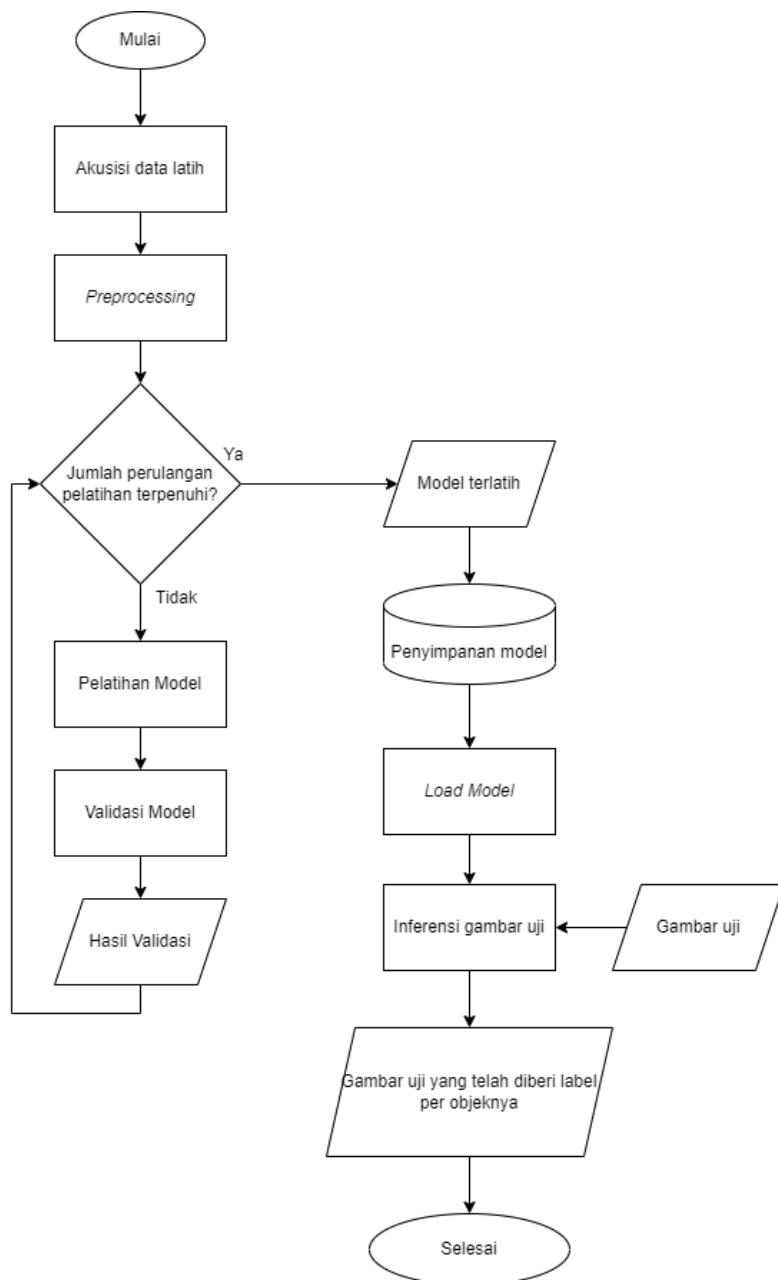
**Gambar 3.1.** Diagram blok rancangan sistem.

Secara keseluruhan, ada empat tahapan pada sistem, yaitu:

1. Akuisisi data, yaitu proses mendapatkan citra-citra UAV dari dataset VisDome.
2. *Preprocessing*, yaitu tahap untuk mengolah dan menyesuaikan dataset sehingga dapat dilatih menggunakan *EfficientDet*.
3. *Training Model*, merupakan proses pelatihan model untuk mengenali objek-objek pada citra UAV.
4. Analisis Hasil, merupakan tahap untuk mengukur performa deteksi menggunakan parameter *average precision* dan *average recall*.

Dalam prosesnya, tugas akhir ini menggunakan satu proses linear dari pelatihan dan validasi. Tidak seperti tugas akhir lain yang proses pelatihan dan validasinya terpisah, pada tugas akhir ini validasi langsung dilakukan setelah pelatihan berakhir. Proses validasi akan menghasilkan nilai yang menjadi tolak ukur kinerja dari model

deteksi objek. Proses pelatihan dan validasi dilakukan berulang. Nilai validasi yang akan ditinjau adalah nilai validasi terakhir yang dihasilkan model. Setelah perulangan proses latih dan validasi, model yang sudah dilatih akan disimpan lalu dilakukan proses inferensi. Proses inferensi artinya memberi label objek-objek yang terdeteksi model pada citra. Proses pelatihan dan validasi dapat dilihat pada Gambar 3.2.



**Gambar 3.2.** Diagram alir sistem.

### 3.1.1 Akuisisi Data

Data yang digunakan dalam tugas akhir adalah himpunan data VisDrone yang tersedia secara umum. Citra yang tersedia menggunakan format .jpg dan mempunyai ukuran 960×540 piksel. Terdapat 10 kelas objek yang dapat dideteksi. 10 kelas tersebut adalah pejalan kaki, manusia, sepeda, mobil, minibus, bus, truk, sepeda motor, sepeda motor roda tiga, dan sepeda roda tiga. Jumlah gambar yang digunakan sebagai data latih sebanyak 6471 gambar. Data validasi yang digunakan sebanyak 548 gambar.

### 3.1.2 *Preprocessing*

Proses *preprocessing* pada tugas akhir ini hanya terdiri dari satu tahap, yaitu mengubah format anotasi (catatan mengenai lokasi berbagai objek pada masing-masing gambar) menjadi *tfrecord* dari format PASCAL VOC. Konversi ini diperlukan karena model deteksi *EfficientDet* pada tugas akhir ini menggunakan bantuan kerangka kerja *automl* yang dapat membaca format anotasi *tfrecord* secara bawaan.

### 3.1.3 Pelatihan Model

Model deteksi objek *EfficientDet-D0* dilatih menggunakan data VisDrone yang sudah diubah anotasinya menjadi *tfrecord*. Dalam satu iterasi pelatihan, model memproses sebanyak 5696 gambar yang dipilih secara acak dari total 6471 gambar yang tersedia sebagai data latih. Selanjutnya, iterasi pelatihan akan disebut dengan *epoch*. Satu *epoch* diraih dengan 6500 proses kalkulasi atau yang selanjutnya disebut dengan *step*. Setelah dilatih, model akan langsung dinilai performanya menggunakan data validasi. Sebagai upaya peningkatan performa. Selanjutnya model akan di latih kembali dengan 5696 dari 6471 gambar yang diacak kembali. Setelah melalui beberapa proses perulangan latih-validasi, proses latih akan dihentikan, model disimpan untuk inferensi, dan skor validasi terakhir akan disimpan sebagai

tolak ukur performa.

### 3.1.4 Analisis Hasil

Setelah proses pelatihan terakhir, kita akan mendapatkan skor validasi. Skor validasi ini terdiri atas *average precision* dan *average recall*. Dua parameter ini akan menjadi tolak ukur performa model. Analisis dilakukan dengan menggunakan bantuan kurva dengan parameter *average precision* atau *average recall* pada sumbu tegak, dan jumlah *steps* pada sumbu datar.

## 3.2 Performansi Sistem

Seperti model deteksi objek lainnya, pada tugas akhir ini, performansi sistem diukur menggunakan *average precision* (AP) dan *average recall* (AR). Hasil skor validasi terdiri atas 6 macam AP dan 6 macam AR. Berbagai macam AP dan AR ini dipengaruhi area deteksi (kecil, sedang, besar), *Intersection over Union* (IoU), dan jumlah deteksi objek maksimum. Pada tugas akhir ini, mengikuti beberapa penelitian yang menggunakan himpunan data serupa, akan hanya diambil 6 nilai, yaitu AP untuk area besar, AP untuk nilai IoU 0.5, AP untuk nilai IoU 0.75, AR dengan maksimal deteksi 1, 10, dan 100.

### 3.2.1 Average Precision (AP) dan Average Recall (AR)

*Average precision* atau presisi rata-rata artinya sebuah nilai yang menunjukkan seberapa tepat sebuah model pendekripsi objek dalam menentukan objek. Nilai AP diadpat dari sebuah fungsi yang membandingkan kotak penanda objek pada anotasi *tfrecord* dengan kotak penanda objek yang dihasilkan model menggunakan IoU. Nilai rata-rata AP didapat melalui persamaan:

$$\frac{1}{n} \sum_{k=1}^{k=n} AP_k, \quad (3.1)$$

- dimana  $n$  adalah jumlah kelas dan  $AP_k$  adalah AP dari masing-masing kelas. Setelah dibandingkan, akan didapat nilai IoU dan selanjutnya ditentukan apakah model dinyatakan berhasil atau gagal dalam menentukan lokasi objek melalui suatu ambang batas nilai IoU. *Average recall* artinya sebuah nilai yang menunjukkan seberapa akurat sebuah model pendekripsi objek dalam melakukan deteksi. Metode yang digunakan hampir sama dengan saat mencari AP, namun AR lebih berfokus pada jumlah objek yang terdeteksi tepat (*True Positive, False Negative*), dan AP berfokus pada ketepatan model saat objek diberi label tertentu (*True Positive, False Positive*).

Nilai AR rata-rata didapat melalui persamaan:

$$\frac{1}{n} \sum_{k=1}^{k=n} AR_k, \quad (3.2)$$

- dimana  $n$  adalah jumlah kelas dan  $AR_k$  adalah AR dari masing-masing kelas objek yang di deteksi.

### 3.3 Sistematika Data

Berbagai gambar yang ada pada himpunan data *VisDrone* diambil dengan latar belakang urban. Seluruh citra diambil di daerah perkotaan, dengan berbagai kendaraan, jalan raya, lahan parkir, bangunan, fasilitas publik, taman, kolam renang, lapangan, dan lain sebagainya fasilitas yang umumnya dapat ditemukan di sebuah kota. Waktu saat pengambilan gambar bervariasi dari pagi hingga malam hari. Seluruh citra merupakan citra UAV yang diambil dari ketinggian. Tidak hanya waktu, kondisi cuaca saat citra diambil juga bervariasi dari cerah sampai hujan yang cukup mengurangi jarak pandang. Berbagai citra ini di sampel dari 14 kota berbeda di seluruh kawasan Republik Rakyat Cina. Himpunan data ini diambil dan dikelola oleh tim AISKEYEYE, Laboratorium Pembelajaran Mesin dan Penambangan Data , Universitas Tianjin, Republik Rakyat Cina.

## BAB IV

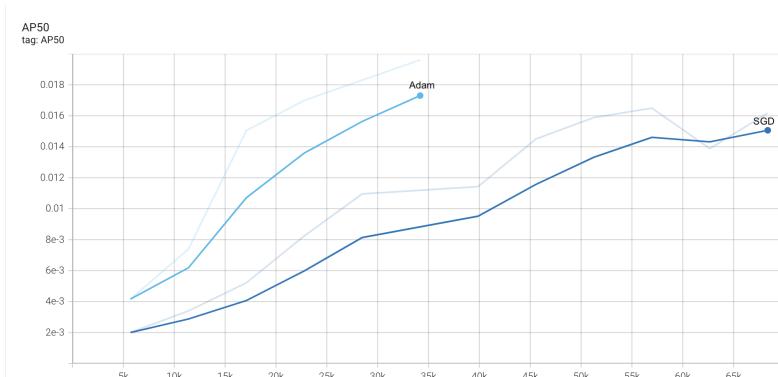
### ANALISIS SIMULASI SISTEM

#### 4.1 Analisis Hasil Deteksi Objek dengan Pelatihan sebanyak 10 epochs

Kali ini model *efficientdet D0* dilatih sebanyak 10 iterasi terlebih dahulu. Model dilatih menggunakan dataset *Visdrone*. Pelatihan menggunakan dua macam fungsi optimasi, yaitu *stochastic gradient descend* [20] dan Adam [21].

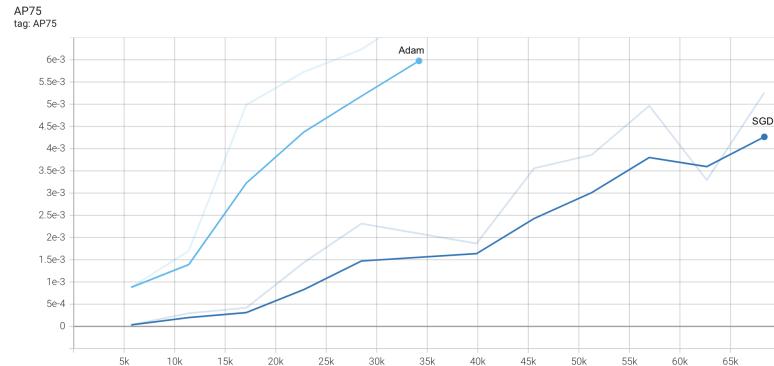
##### 4.1.1 Hasil Average Precision

Nilai AP50 dari model yang dilatih menggunakan *stochastic gradient descend* menghasilkan 0,015 atau 1,5%. Sementara model yang dilatih menggunakan fungsi optimasi Adam memberikan hasil yang sedikit lebih baik, yaitu di angka 1,73% atau 0,0173. Rincian kenaikan skor AP50 tertera pada gambar 4.1.



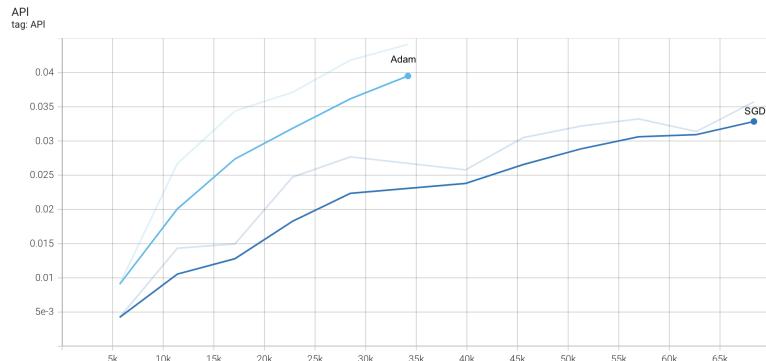
Gambar 4.1. Skor AP50 setelah pelatihan sebanyak 10 epochs

Untuk nilai AP75 keduanya menunjukkan skor yang lebih rendah lagi. AP75 untuk fungsi optimasi Adam hanya sebesar  $5,97 \times 10^{-3}$  dan *stochastic gradient descend* bahkan lebih kecil lagi, sebesar  $4,26 \times 10^{-3}$ . Rincian kenaikan skor AP75 tertera pada gambar 4.2.



**Gambar 4.2.** Skor AP75 setelah pelatihan sebanyak 10 epochs

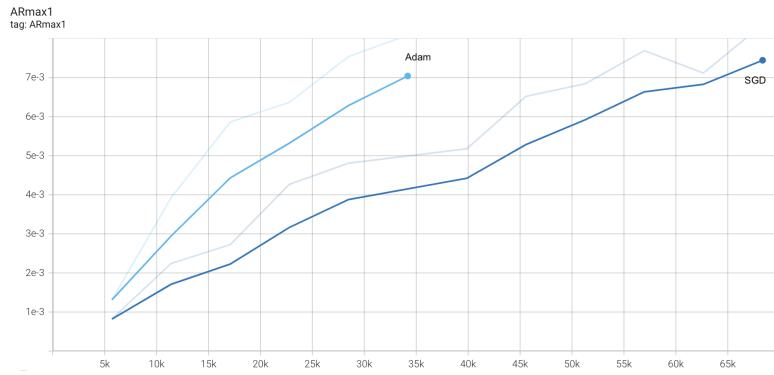
AP untuk area besar menunjukkan hasil yang lebih baik dari kedua jenis fungsi optimasi. Model yang dilatih menggunakan fungsi optimasi Adam mendapat skor 0,0395 atau 3,95%. Model dengan *stochastic gradient descend* mendapat skor 0,03285 atau 3,285% . Rincian kenaikan skor AP untuk area besar tertera pada gambar 4.3.



**Gambar 4.3.** Skor API atau AP di area besar setelah pelatihan sebanyak 10 epochs

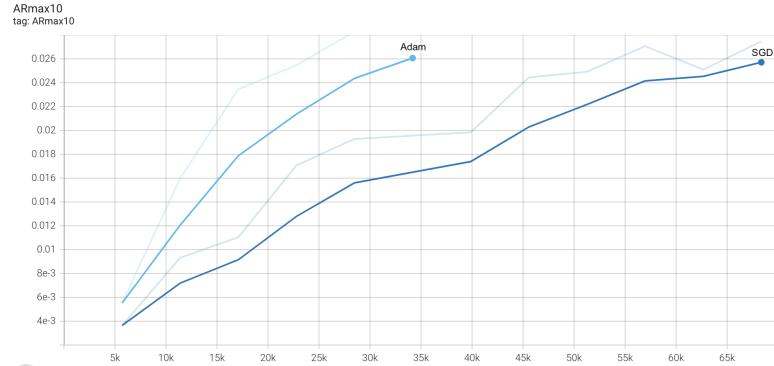
#### 4.1.2 Hasil Average Recall

Nilai AR dengan maksimal deteksi 1 dari model yang dilatih menggunakan fungsi optimasi *stochastic gradient descend* menghasilkan  $8,36 \times 10^{-3}$ . Sementara model yang dilatih menggunakan fungsi optimasi Adam memberikan hasil yang sedikit lebih buruk, yaitu di angka  $8,08 \times 10^{-3}$ . Rincian kenaikan skor ARmax1 tertera pada gambar 4.4.



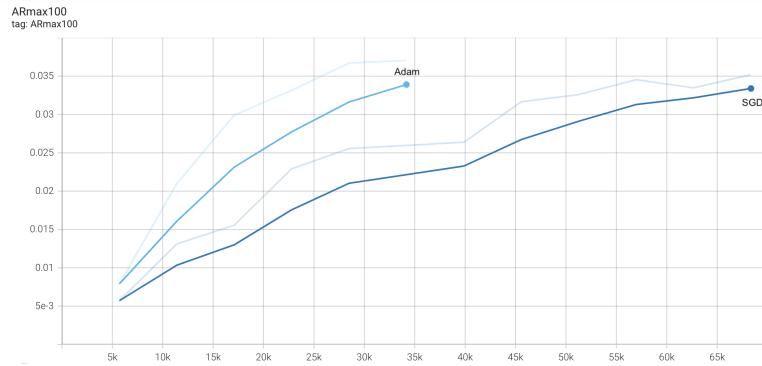
**Gambar 4.4.** Skor ARmax1 setelah pelatihan sebanyak 10 epochs

Nilai AR dengan maksimal deteksi 10 menunjukkan hasil yang lebih baik. Model yang dilatih menggunakan fungsi optimasi *stochastic gradient descend* menghasilkan 0,0257 atau 2,57%. Model yang menggunakan fungsi optimasi Adam menunjukkan hasil yang sedikit lebih baik. Optimasi Adam mampu sedikit mendongkrak hasil menjadi 0,0284 atau 2,84%. Rincian kenaikan skor ARmax10 tertera pada gambar 4.5.



**Gambar 4.5.** Skor ARmax10 setelah pelatihan sebanyak 10 epochs

AR dengan maksimal deteksi 100 menunjukkan hasil yang lebih baik lagi. Model berhasil meraih skor 0,0334 atau 3,34% dengan fungsi optimasi *stochastic gradient descend*. Sementara itu fungsi optimasi Adam unggul tipis dengan skor 0,0339 atau 3,39%. Rincian kenaikan skor ARmax100 tertera pada gambar 4.6.



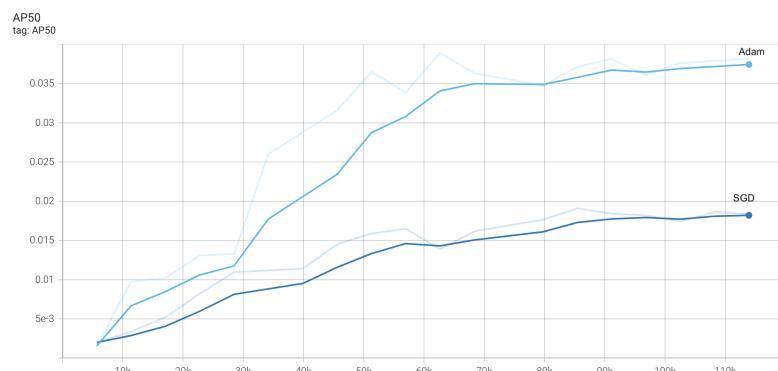
**Gambar 4.6.** Skor ARmax100 setelah pelatihan sebanyak 10 epochs

## 4.2 Analisis Hasil Deteksi Objek dengan Pelatihan sebanyak 20 epochs

Selanjutnya model *efficientdet D0* dilatih dengan iterasi yang lebih banyak yaitu 20 iterasi untuk melihat perkembangan skor. Model masih dilatih menggunakan dataset *Visdrone*. Pelatihan menggunakan dua macam fungsi optimasi, yaitu *stochastic gradient descend* [20] dan Adam [21].

### 4.2.1 Hasil Average Precision

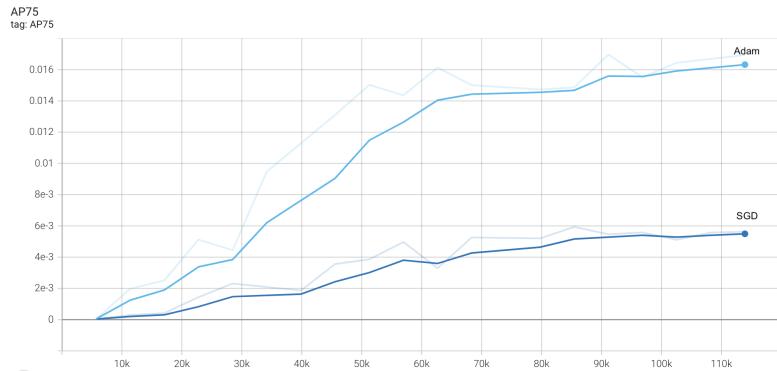
Terjadi selisih yang cukup terlihat untuk nilai AP50 antara fungsi optimasi *stochastic gradient descend* (SGD) dan Adam. SGD hanya mendapat skor sebesar 0,0184. Adam unggul jauh dengan 0,0382. Rincian dari proses pelatihan dan kenaikan skor AP50 tertera pada gambar 4.7.



**Gambar 4.7.** Skor AP50 setelah pelatihan sebanyak 20 epochs

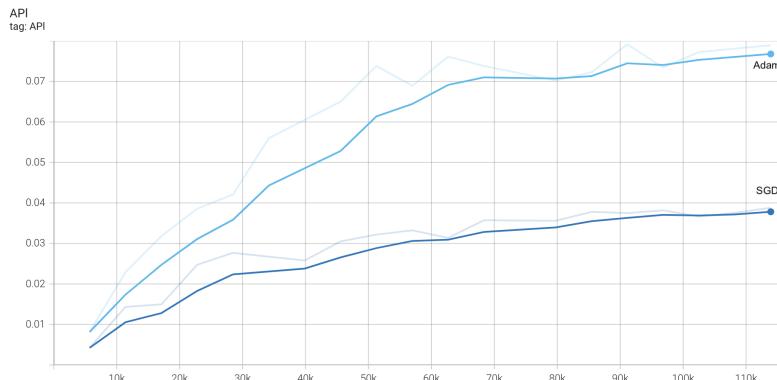
Selisih yang cukup jauh terlihat kembali muncul untuk nilai AP75 antara fung-

si optimasi *stochastic gradient descend* (SGD) dan Adam. SGD mendapat skor sebesar  $5,63 \times 10^{-3}$ . Adam masih unggul dengan skor sebesar 0.0169. Rincian kenaikan dalam *steps* tertera pada gambar 4.8.



**Gambar 4.8.** Skor AP75 setelah pelatihan sebanyak 20 epochs

Pola yang sama terjadi untuk skor AP dengan area besar atau API. SGD mendapat skor API sebesar 0,0388. Fungsi optimasi Adam cukup unggul dengan skor API sebesar 0,0789. Rincian kenaikan dalam *steps* tertera pada gambar 4.9.

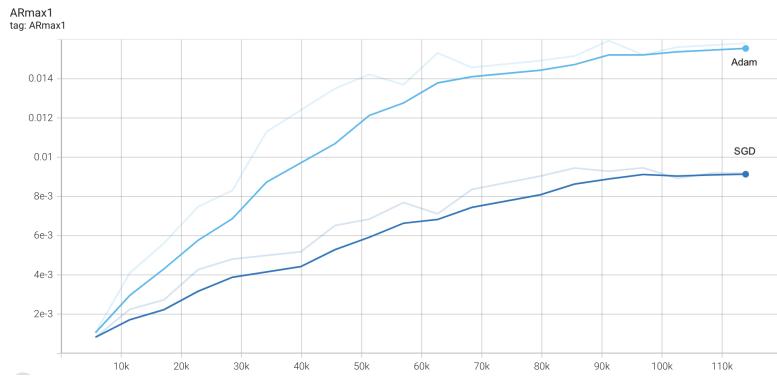


**Gambar 4.9.** Skor API setelah pelatihan sebanyak 20 epochs

#### 4.2.2 Hasil Average Recall

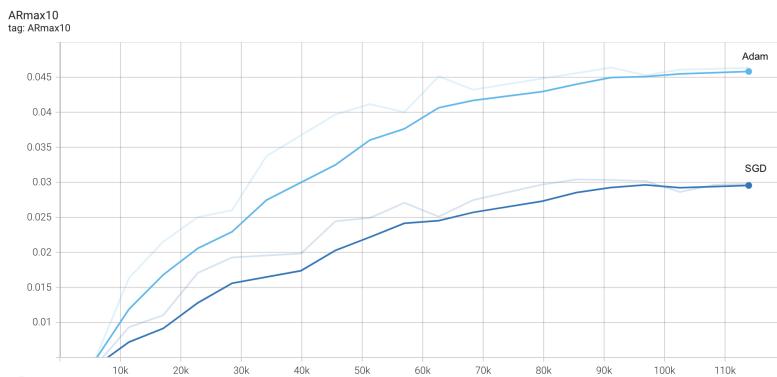
Dalam skor AR maksimal deteksi 1, fungsi optimasi Adam unggul tipis dari fungsi optimasi *stochastic gradient descend* (SGD). SGD mendapat skor sebesar  $9,18 \times 10^{-3}$ . Adam mendapat skor ARmax1 sebesar 0,0158. Hasil ini sudah mengungguli dua model *state-of-the-art* dari metrik ARmax1 [8]. Terlebih lagi model *Efficientdet d0* hanya dilatih dengan 20 iterasi (dibanding model lain yang dilatih

menggunakan ratusan iterasi) serta menggunakan sistem komputer konvensional yang terjangkau bagi masyarakat. Rincian kenaikan dalam *steps* tertera pada gambar 4.10.



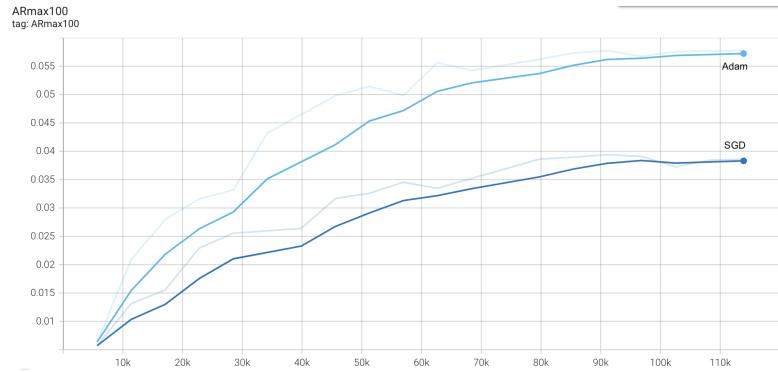
**Gambar 4.10.** Skor ARmax1 setelah pelatihan sebanyak 20 epochs

Pada Skor AR maksimal deteksi 10, fungsi optimasi Adam kembali unggul. Model dengan optimasi SGD mendapat skor ARmax10 sebesar 0,0298. Model dengan optimasi Adam unggul dengan skor ARmax10 sebesar 0,0463. Sayangnya kedua skor ini belum mampu melampaui model *state-of-the-art* untuk dataset *Visdrone*. Rincian nilai ARmax10 terhadap *steps* tertera pada gambar 4.11.



**Gambar 4.11.** Skor ARmax10 setelah pelatihan sebanyak 20 epochs

Untuk skor AR maksimal deteksi 100, terjadi peningkatan dari kedua model dengan fungsi optimasi yang berbeda, namun peningkatan ini belum cukup signifikan untuk mengalahkan berbagai model *state-of-the-art*. Model dengan optimasi SGD mendapat skor sebesar 0,0386. Sementara model dengan optimasi Adam mendapat skor sebesar 0,0572. Rincian kenaikan skor ARmax100 tertera pada gambar 4.12.



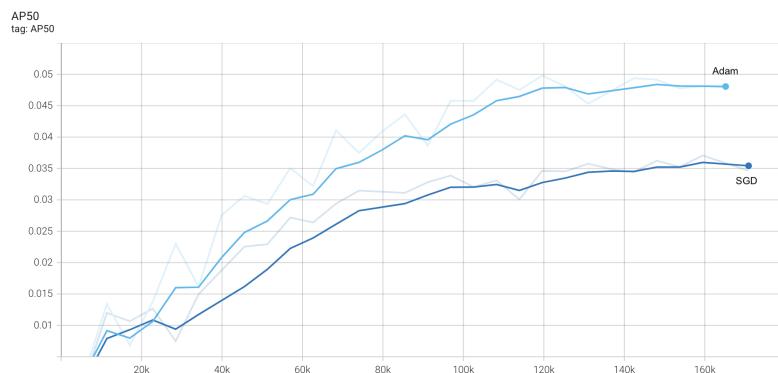
**Gambar 4.12.** Skor ARmax100 setelah pelatihan sebanyak 20 epochs

### 4.3 Analisis Hasil Deteksi Objek dengan Pelatihan sebanyak 30 epochs

Setelah melihat potensi dari skor ARmax1 bila dibandingkan dengan berbagai model yang sudah dicobakan sebelumnya, peneliti melanjutkan peningkatan iterasi pembelajaran ke 30 iterasi. Model masih dilatih menggunakan dataset *Visdrone*. Pelatihan menggunakan dua macam fungsi optimasi, yaitu *stochastic gradient descend* [20] dan Adam [21].

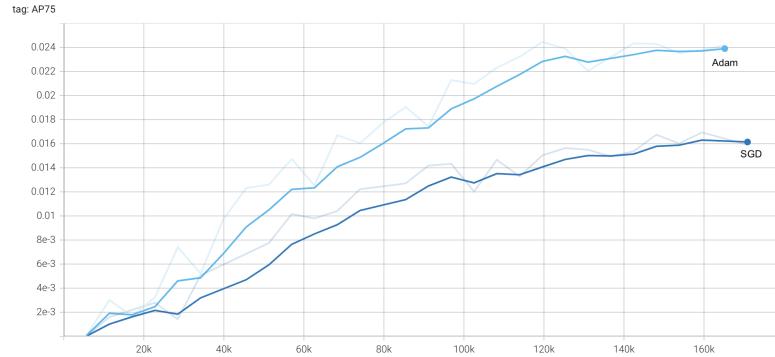
#### 4.3.1 Hasil Average Precision

Menambah 10 iterasi pembelajaran dari sebelumnya, dari metrik AP50 terlihat peningkatan. Model dengan fungsi optimasi SGD mendapat skor AP50 sebesar 0,0346. Model dengan fungsi optimasi Adam mendapat skor AP50 sebesar 0,0480. Rincian kenaikan skor AP50 tertera pada gambar 4.13.



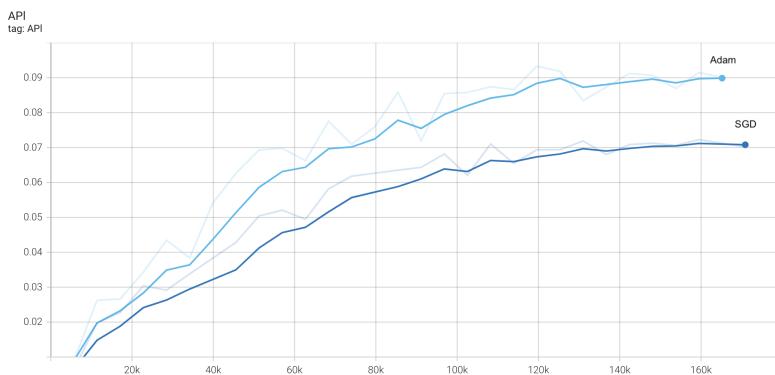
**Gambar 4.13.** Skor AP50 setelah pelatihan sebanyak 30 epochs

Sesuai dugaan, bila kita meningkatkan ambang batas deteksi, performa model akan terlihat tidak sebaik sebelumnya. Untuk skor metrik AP75, model dengan optimasi SGD mendapat skor sebesar 0,0161. Model dengan optimasi Adam mendapat skor AP75 sebesar 0,0239. Rincian kenaikan skor AP50 terhadap *steps* tertera pada gambar 4.14.



**Gambar 4.14.** Skor AP75 setelah pelatihan sebanyak 30 epochs

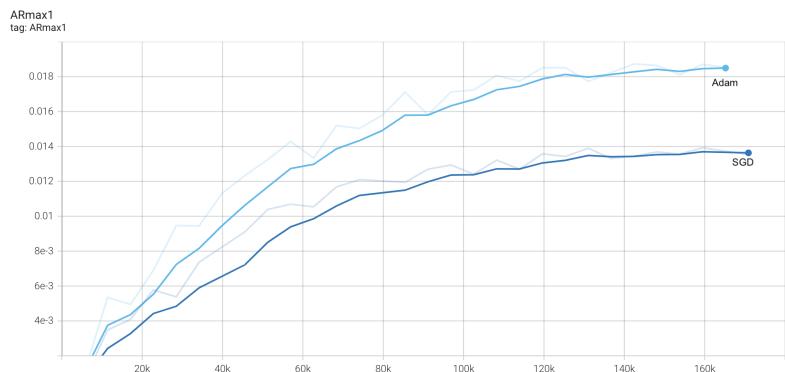
Untuk skor AP untuk area besar atau API, kedua model menunjukkan kenaikan yang baik namun belum cukup untuk mengungguli model yang sudah tersedia untuk dataset *Visdrone*. Model dengan optimasi SGD mendapat skor sebesar 0,0702. Model dengan optimasi Adam mendapat skor sebesar 0,0901. Rincian kenaikan skor API tertera pada gambar 4.15.



**Gambar 4.15.** Skor API setelah pelatihan sebanyak 30 epochs

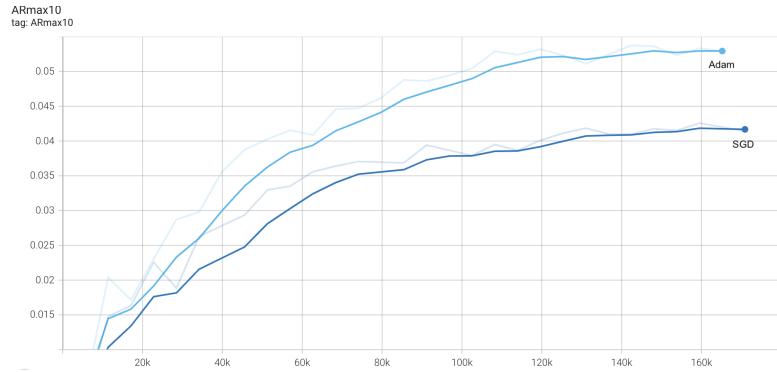
### 4.3.2 Hasil Average Recall

Untuk AR dengan maksimal deteksi 1, model dengan fungsi optimasi Adam berhasil mengungguli TridentNet yang sebelumnya menjadi model dengan nilai ARmax1 tertinggi [10]. *Efficientdet D0* dengan fungsi optimasi Adam dapat unggul hanya dengan 30 iterasi saja, dengan nilai ARmax1 sebesar 0,0185 atau 1.85%. Pada penelitian sebelumnya, TridentNet meraih ARmax1 sebesar 1.17%. Model yang dilatih dengan fungsi optimasi SGD mendapat skor 0,0135. Model dengan fungsi optimasi SGD juga mengungguli model tertinggi kedua dan ketiga pada penelitian sebelumnya. Rincian kenaikan skor ARmax1 tertera pada gambar 4.16.



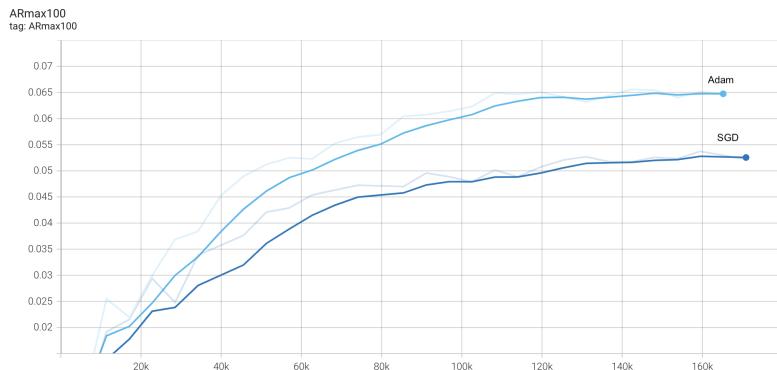
Gambar 4.16. Skor ARmax1 setelah pelatihan sebanyak 30 epochs

Bila kita menambah ambang maksimal deteksi AR menjadi 10, fungsi optimasi Adam dan SGD belum mampu mendongkrak jauh skor ARmax10 dari *efficientdet D0*. Model dengan fungsi optimasi Adam mendapat ARmax10 sebesar 0,0529. Model dengan fungsi optimasi SGD mendapat ARmax10 sebesar 0,0414. Rincian kenaikan skor ARmax10 tertera pada gambar 4.17.



**Gambar 4.17.** Skor ARmax10 setelah pelatihan sebanyak 30 epochs

Terakhir, dari skor ARmax100 menunjukkan pola yang sama dengan sebelumnya. Model dengan fungsi optimasi Adam mendapat skor ARmax100 sebesar 0,0647. Model dengan fungsi optimasi SGD mendapat skor ARmax100 sebesar 0,0522. Rincian kenaikan skor ARmax100 tertera pada gambar 4.18.



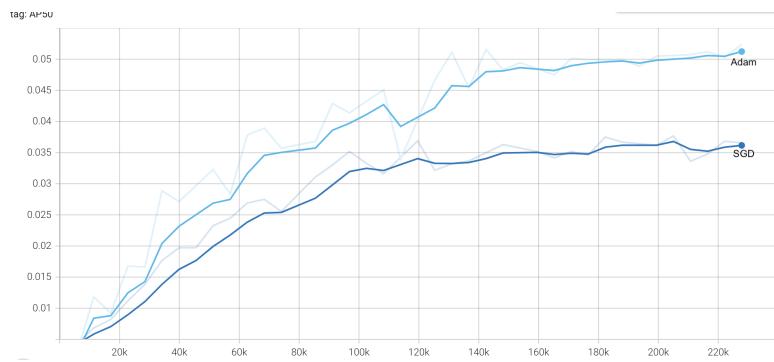
**Gambar 4.18.** Skor ARmax100 setelah pelatihan sebanyak 30 epochs

#### 4.4 Analisis Hasil Deteksi Objek dengan pelatihan sebanyak 40 epochs

Peneliti mencoba kembali menambah iterasi pembelajaran menjadi 40 iterasi. Peningkatan iterasi dilakukan karena kondisi komputer pribadi konvensional sekiranya masih dapat menjalankan pembelajaran model sampai 50 iterasi. Selain itu, nilai ARmax1 masih berpotensi meningkat. Model masih dilatih menggunakan dataset *Visdrone*. Pelatihan menggunakan dua macam fungsi optimasi, yaitu *stochastic gradient descend* [20] dan Adam [21].

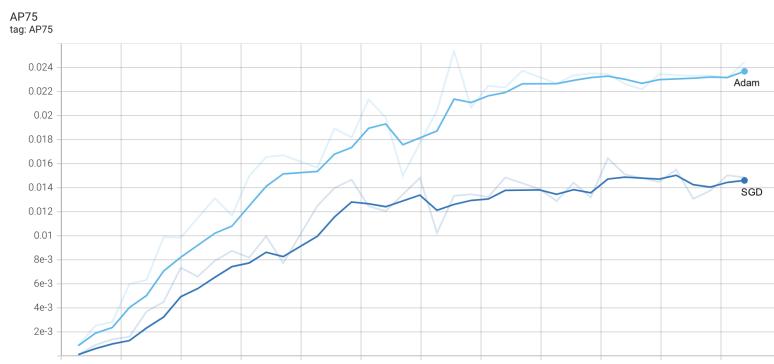
#### 4.4.1 Hasil Average Precision

Nilai AP50 dari kedua model masih menunjukkan pola yang sama dengan jumlah iterasi pelatihan sebelumnya, hanya saja terjadi peningkatan nilai. Model dengan fungsi optimasi SGD mendapat nilai AP50 sebesar 0,0366. Model dengan fungsi optimasi Adam mendapat nilai AP50 sebesar 0,0524. Rincian skor AP50 tertera pada gambar 4.19.



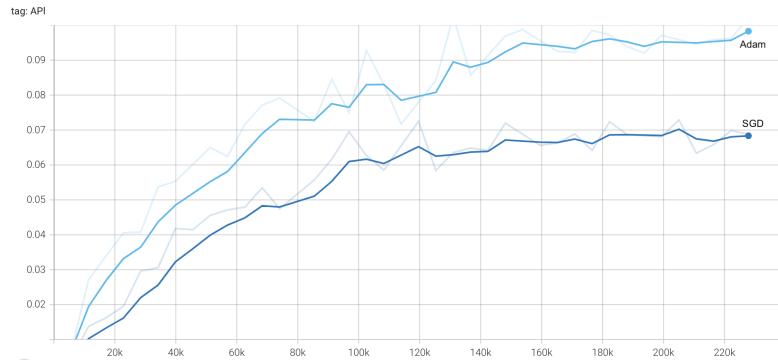
Gambar 4.19. Skor AP50 setelah pelatihan sebanyak 40 epochs

Untuk metrik AP75, Model dengan fungsi optimasi Adam mendapat skor AP75 sebesar 0,0245. Sementara untuk model dengan fungsi optimasi SGD mendapat skor AP75 sebesar 0,0149. Rincian skor AP75 tertera pada gambar 4.20.



Gambar 4.20. Skor AP75 setelah pelatihan sebanyak 40 epochs

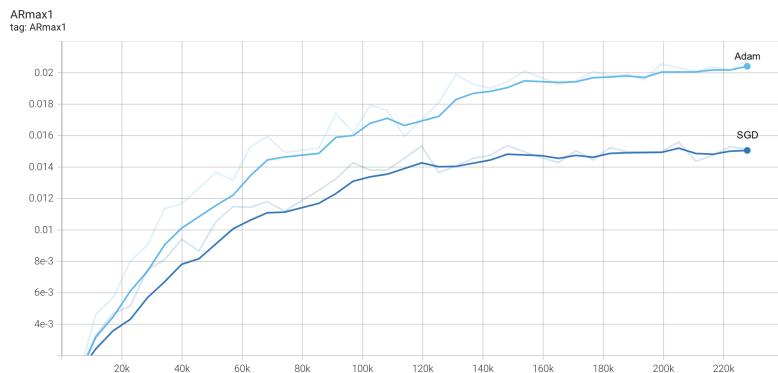
Untuk metrik AP dalam area besar atau API, Model dengan fungsi optimasi Adam mendapat skor API sebesar 0,1022. Sementara untuk model dengan fungsi optimasi SGD mendapat skor API sebesar 0,0689. Rincian skor API tertera pada gambar 4.21.



**Gambar 4.21.** Skor API setelah pelatihan sebanyak 40 epochs

#### 4.4.2 Hasil *Average Recall*

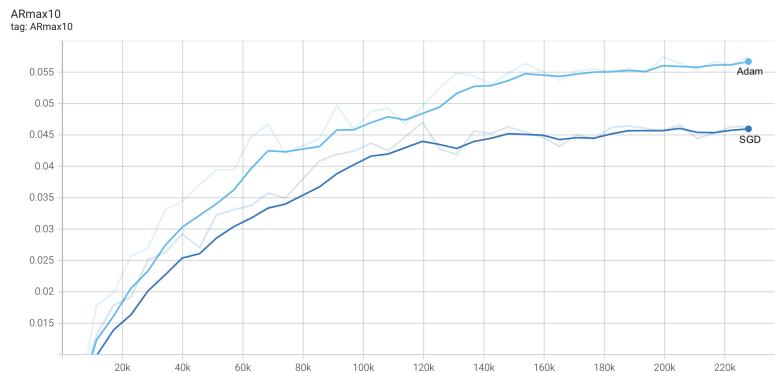
Untuk AR dengan maksimal deteksi 1, model dengan fungsi optimasi Adam kembali berhasil mengungguli TridentNet. Kali ini model dengan fungsi optimasi Adam mengungguli TridentNet hingga nyaris dua kali lipat [8]. *Efficientdet D0* dengan fungsi optimasi Adam berpotensi untuk memiliki nilai ARmax1 yang semakin tinggi bila ditingkatkan iterasi pembelajarannya. Nilai ARmax1 pada model yang menggunakan Adam sebesar 0,0208 atau 2,08%. Pada penelitian sebelumnya, TridentNet meraih ARmax1 sebesar 1,17%. Model yang dilatih dengan fungsi optimasi SGD mendapat skor 0,0151. Model dengan fungsi optimasi SGD juga mengungguli model tertinggi kedua dan ketiga pada penelitian sebelumnya. Rincian kenaikan skor ARmax1 tertera pada gambar 4.22.



**Gambar 4.22.** Skor ARmax1 setelah pelatihan sebanyak 40 epochs

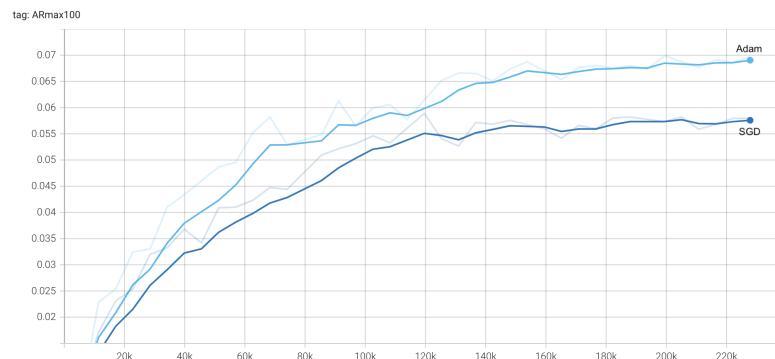
Untuk metrik ARmax10 atau AR maksimal deteksi 10, model dengan fungsi

optimasi Adam mendapat skor sebesar 0,0575. Model yang menggunakan fungsi optimasi SGD mendapat ARmax10 sebesar 0,0463. Rincian perkembangan skor ARmax10 tertera pada gambar 4.23.



**Gambar 4.23.** Skor ARmax10 setelah pelatihan sebanyak 40 epochs

Dari sisi metrik AR dengan maksimal deteksi 100, model dengan fungsi optimasi Adam mendapat skor sebesar 0,0698. Sementara model dengan fungsi optimasi SGD mendapat ARmax100 sebesar 0,0580. Rincian perkembangan skor ARmax100 tertera pada gambar 4.24.



**Gambar 4.24.** Skor ARmax100 setelah pelatihan sebanyak 40 epochs

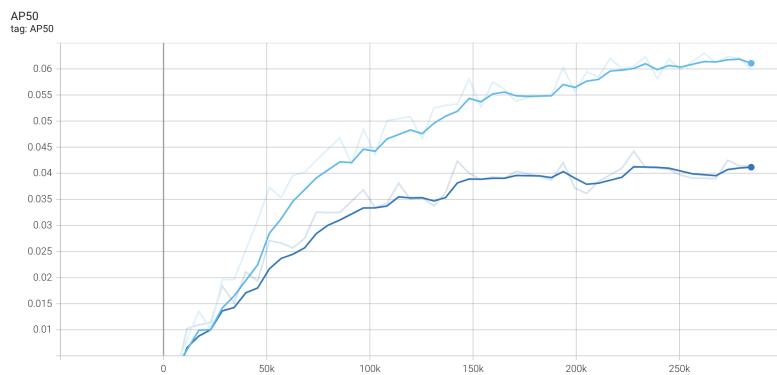
#### 4.5 Analisis Hasil Deteksi Objek dengan pelatihan sebanyak 50 epochs

50 iterasi pelatihan atau *epochs* dirasa sudah cukup menggambarkan kemampuan latih komputer pribadi konvensional. Perulangan pelatihan model sebanyak 50 kali ini memakan waktu proses hingga 10 hari 19 jam untuk model dengan fungsi optimasi SGD, dan 4 hari 17 jam untuk model dengan fungsi optimasi Adam, de-

ngan sistem komputer pribadi konvensional yang disebutkan pada batasan masalah. Semakin tinggi *epochs*, semakin lama juga waktu pemrosesan yang diperlukan. Tidak praktis bila komputer pribadi dibiarkan menyala dan dalam beban kerja tinggi selama lebih dari 10 hari. Maka dari itu, 50 *epochs* adalah jumlah iterasi pembelajaran terbanyak dalam penelitian ini. Dalam pengujian terakhir ini, model masih dilatih menggunakan dataset *Visdrone*. Pelatihan menggunakan dua macam fungsi optimasi, yaitu *stochastic gradient descend* [20] dan Adam [21].

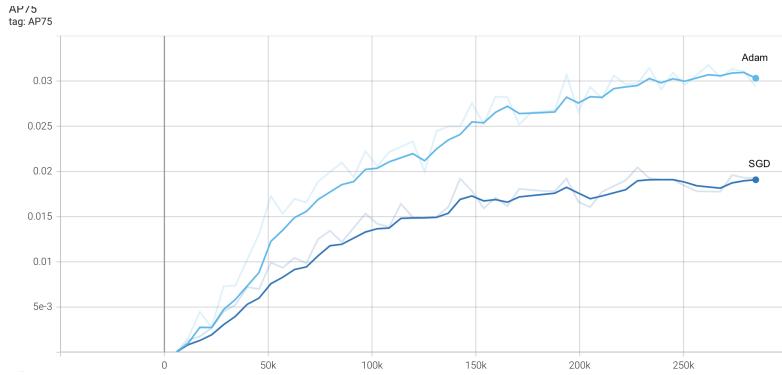
#### 4.5.1 Hasil Average Precision

Pada metrik AP50, model yang dilatih menggunakan fungsi optimasi Adam mendapat skor sebesar 0,0621. Model yang dilatih menggunakan fungsi optimasi SGD mendapat skor sebesar 0,0414. Rincian kenaikan skor AP50 tertera pada gambar 4.25.



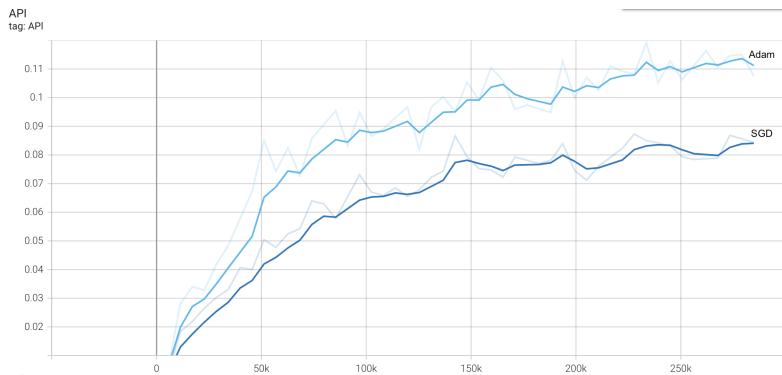
**Gambar 4.25.** Skor AP50 setelah pelatihan sebanyak 50 epochs

Selanjutnya, untuk ambang batas yang lebih besar yaitu AP75. Model yang dilatih dengan fungsi optimasi Adam mendapat skor sebesar 0,0303. Sementara itu, model yang dilatih dengan fungsi optimasi SGD mendapat skor sebesar 0,0193. Rincian kenaikan skor AP75 tertera pada gambar 4.26.



**Gambar 4.26.** Skor AP75 setelah pelatihan sebanyak 50 epochs

Terakhir, untuk AP dengan area besar atau API. Model yang dilatih dengan fungsi optimasi Adam mendapat skor sebesar 0,108. Model yang dilatih dengan fungsi optimasi SGD mendapat skor sebesar 0,0841. Rincian kenaikan skor API tertera pada gambar 4.27.

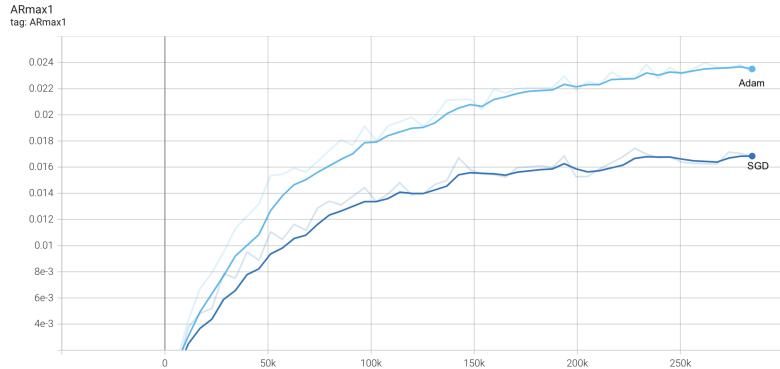


**Gambar 4.27.** Skor API setelah pelatihan sebanyak 50 epochs

#### 4.5.2 Hasil Average Recall

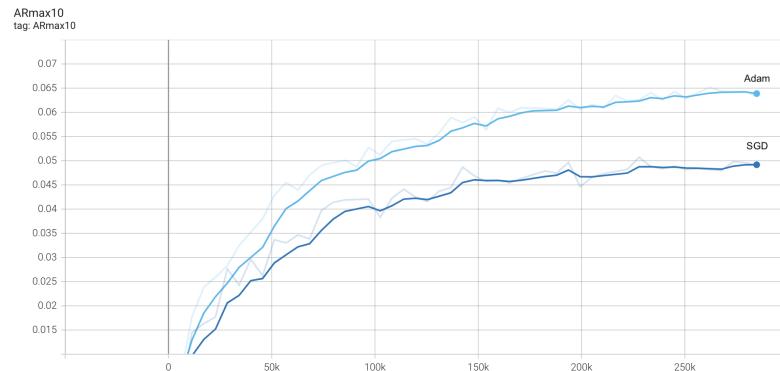
Untuk AR dengan maksimal deteksi 1, model dengan fungsi optimasi Adam berhasil memperbesar keunggulannya dibandingkan TridentNet [1]. *Efficientdet D0* dengan fungsi optimasi Adam berpotensi untuk memiliki nilai ARmax1 yang semakin tinggi bila ditingkatkan iterasi pembelajarannya. Nilai ARmax1 pada model yang menggunakan Adam sebesar 0,0233 atau 2,33%. Pada penelitian sebelumnya, TridentNet meraih ARmax1 sebesar 1,17%. Model yang dilatih dengan fungsi optimasi SGD mendapat skor 0,0169, yang juga unggul dari TridentNet. Rincian

kenaikan skor ARmax1 tertera pada gambar 4.28.



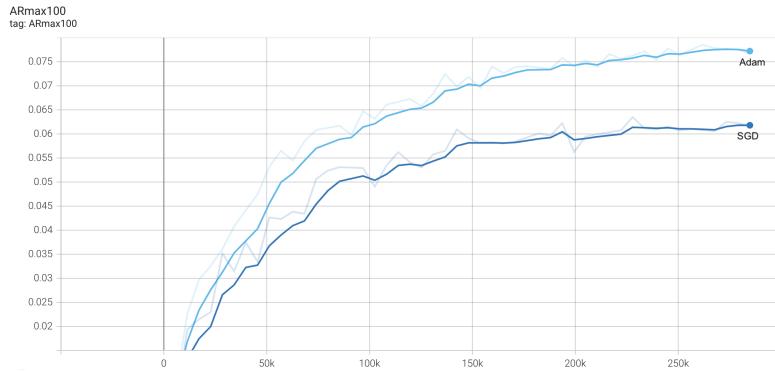
**Gambar 4.28.** Skor ARmax1 setelah pelatihan sebanyak 50 epochs

Selanjutnya, ketika ambang maksimal deteksi ditingkatkan ke 10. Model dengan fungsi optimasi Adam mendapat ARmax10 sebesar 0,0633. Model dengan fungsi optimasi SGD mendapat ARmax10 sebesar 0,0492. Rincian kenaikan skor ARmax10 tertera pada gambar 4.29.



**Gambar 4.29.** Skor ARmax10 setelah pelatihan sebanyak 50 epochs

Ketika ambang maksimal deteksi kembali ditingkatkan ke 100. Model dengan fungsi optimasi Adam mendapat ARmax100 sebesar 0,0767. Sementara itu, model dengan fungsi optimasi SGD mendapat ARmax100 sebesar 0,0618. Rincian kenaikan skor ARmax100 tertera pada gambar 4.30.



**Gambar 4.30.** Skor ARmax100 setelah pelatihan sebanyak 50 epochs

## 4.6 Pembahasan

Model *EfficientDet D0* yang dilatih dalam sistem komputer pribadi konvensional yang terbatas mampu unggul dari performa *average recall* bila dibandingkan dengan penelitian sebelumnya [10]. Perlu diketahui juga bahwa berbagai model pada penelitian sebelumnya dilatih dalam ratusan iterasi. Model *EfficientDet D0* mampu unggul hanya dengan 30 iterasi atau lebih. Nilai *average recall* penting untuk sebuah model deteksi objek khususnya pada implementasi UAV karena model akan mendekripsi lebih banyak objek dan mengacu pada lokalisasi akurasi [22]. Ketika sebuah model memiliki *recall* yang tinggi namun *precision* yang rendah, model akan mengklasifikasi lebih banyak objek atau sampel dengan tepat, namun akan banyak juga sampel yang teridentifikasi sebagai kelas lain (atau *false positive*) [23]. Dalam kasus deteksi objek pada UAV, memprioritaskan nilai *average recall* diperlukan khususnya pada kasus inferensi langsung saat UAV bergerak dan beroperasi. Prioritas ini dibutuhkan karena untuk UAV akan lebih penting untuk mendekripsi keberadaan objek terlebih dahulu. Prioritas deteksi keberadaan objek yang berarti memprioritaskan nilai *average recall* dan mentoleransi rendahnya nilai *average precision* dapat diterima pada kasus implementasi UAV. Kecepatan deteksi juga menjadi hal penting dalam model deteksi objek pada UAV. Hosang, dkk. dalam artikelnya juga menyebutkan pada model deteksi dengan waktu proses yang lebih rendah, nilai kualitas deteksinya dapat berkurang dan itu dapat ditoleransi [22].

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil simulasi dan analisis dapat ditarik kesimpulan bahwa model *EfficientDet d0* dapat mengungguli skor ARmax1 model dari penelitian sebelumnya dalam 30 iterasi atau lebih walaupun dalam kemampuan komputasi perangkat keras yang terbatas. Model *EfficientDet d0* cocok digunakan sebagai model deteksi objek berbiaya komputasi rendah untuk implementasi UAV karena memiliki nilai *recall* yang tinggi. Rendahnya biaya komputasi yang diperlukan akan mempermudah proses pelatihan, meningkatkan kecepatan deteksi pada UAV dan skalabilitas deteksi objek di masa mendatang.

#### **5.2 Saran**

Saran yang dapat diterapkan pada penelitian selanjutnya yang berkaitan dengan topik Tugas Akhir adalah terdapat potensi untuk meningkatkan hasil deteksi dari model. Tidak hanya untuk meningkatkan *recall*, namun juga untuk meningkatkan *precision*. Penelitian selanjutnya dapat meningkatkan reliabilitas model secara keseluruhan dengan menerapkan *preprocessing* yang lebih banyak untuk menyederhanakan citra masukan, serta modifikasi lapisan *pooling* pada model untuk meningkatkan kapabilitas model bila dihadapkan dengan citra resolusi tinggi dari UAV.

## DAFTAR PUSTAKA

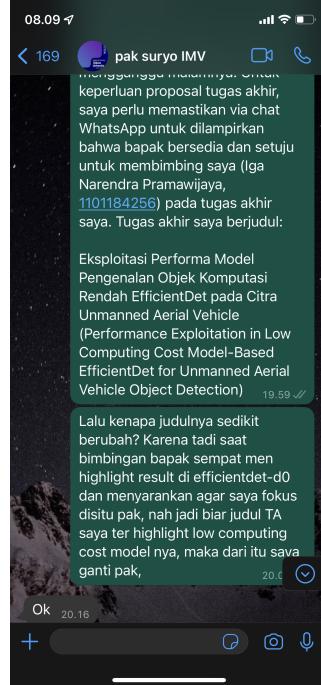
- [1] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, “Detection and tracking meet drones challenge,” 2021.
- [2] J. Wu, G. Yang, X. Yang, B. Xu, L. Han, and Y. Zhu, “Automatic counting of in situ rice seedlings from uav images based on a deep fully convolutional neural network,” *Remote Sensing*, vol. 11, no. 6, 2019. [Online]. Available: <https://www.mdpi.com/2072-4292/11/6/691>
- [3] S. Oleire-Oltmanns, I. Marzolff, K. D. Peter, and J. B. Ries, “Unmanned aerial vehicle (uav) for monitoring soil erosion in morocco,” *Remote Sensing*, vol. 4, no. 11, pp. 3390–3416, 2012. [Online]. Available: <https://www.mdpi.com/2072-4292/4/11/3390>
- [4] K. G. Panda, S. Das, D. Sen, and W. Arif, “Design and deployment of uav-aided post-disaster emergency network,” *IEEE Access*, vol. 7, pp. 102 985–102 999, 2019.
- [5] J. M. M. Neto, R. A. da Paixão, L. R. L. Rodrigues, E. M. Moreira, J. C. J. dos Santos, and P. F. F. Rosa, “A surveillance task for a uav in a natural disaster scenario,” in *2012 IEEE International Symposium on Industrial Electronics*, 2012, pp. 1516–1522.
- [6] G.-H. Kwak and N.-W. Park, “Impact of texture information on crop classification with machine learning and uav images,” *Applied Sciences*, vol. 9, no. 4, 2019. [Online]. Available: <https://www.mdpi.com/2076-3417/9/4/643>
- [7] C. Kyrkou, G. Plastiras, T. Theocharides, S. I. Venieris, and C.-S. Bouganis, “Dronet: Efficient convolutional neural network detector for real-time uav

- applications,” in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2018, pp. 967–972.
- [8] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10781–10790.
- [9] T. McREYNOLDS and D. BLYTHE, “Chapter 19 - illustration and artistic techniques,” in *Advanced Graphics Programming Using OpenGL*, ser. The Morgan Kaufmann Series in Computer Graphics, T. McREYNOLDS and D. BLYTHE, Eds. San Francisco: Morgan Kaufmann, 2005, pp. 501–530. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9781558606593500214>
- [10] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, Q. Nie, H. Cheng, C. Liu, X. Liu *et al.*, “Visdrone-det2018: The vision meets drone object detection in image challenge results,” in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.
- [11] S.-C. Wang, “Artificial neural network,” in *Interdisciplinary computing in java programming*. Springer, 2003, pp. 81–100.
- [12] A. V. Joshi, *Machine learning and artificial intelligence*. Springer, 2020.
- [13] J. Watt, R. Borhani, and A. K. Katsaggelos, *Machine learning refined: foundations, algorithms, and applications*. Cambridge University Press, 2020.
- [14] C. Szegedy, A. Toshev, and D. Erhan, “Deep neural networks for object detection,” 2013.
- [15] C. P. Papageorgiou, M. Oren, and T. Poggio, “A general framework for object detection,” in *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*. IEEE, 1998, pp. 555–562.

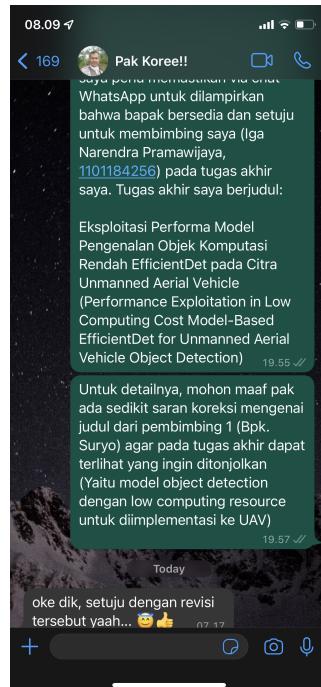
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
- [17] A. R. Pathak, M. Pandey, and S. Rautaray, “Application of deep learning for object detection,” *Procedia Computer Science*, vol. 132, pp. 1706–1717, 2018, international Conference on Computational Intelligence and Data Science. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050918308767>
- [18] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [19] J. Chen, H. Mai, L. Luo, X. Chen, and K. Wu, “Effective feature fusion network in bifpn for small object detection,” in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 699–703.
- [20] C. Lemaréchal, “Cauchy and the gradient method,” 2012.
- [21] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [22] J. H. Hosang, R. Benenson, P. Dollár, and B. Schiele, “What makes for effective detection proposals?” *CoRR*, vol. abs/1502.05082, 2015. [Online]. Available: <http://arxiv.org/abs/1502.05082>
- [23] A. Gad, “Evaluating object detection models using mean average precision,” Mar 2022. [Online]. Available: <https://www.kdnuggets.com/2021/03/evaluating-object-detection-models-using-mean-average-precision.html>

## **LAMPIRAN**

## Bukti persetujuan pembimbing



**Gambar 0.1.** Bukti Persetujuan Pembimbing 1 (Suryo Adhi Wibowo, S.T., M.T., Ph.D)



**Gambar 0.2.** Bukti Persetujuan Pembimbing 2 (Dr. Koredianto Usman, S.T., M.Sc)