

**TUGAS AKHIR - EC184801**

# **OPTIMISASI PENDETEKSIAN OBJEK KECIL YOLOv7 UNTUK MENDETEKSI OBJEK-OBJEK *AIRBORNE***

**Dion Andreas Solang**

NRP 0721 19 4000 0039

Dosen Pembimbing

**Reza Fuad Rachmadi, S.T., M.T., Ph.D**

NIP 19850403201212 1 001

**Dr. I Ketut Eddy Purnama, S.T., M.T.**

NIP 19690730199512 1 001

**Program Studi Strata 1 (S1) Teknik Komputer**

Departemen Teknik Komputer

Fakultas Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2023



**FINAL PROJECT - EC184801**

***YOLOv7 SMALL OBJECT DETECTION OPTIMIZATION  
TO DETECT AIRBORNE OBJECTS***

**Dion Andreas Solang**

NRP 0721 19 4000 0039

Advisor

**Reza Fuad Rachmadi, S.T., M.T., Ph.D**

NIP 19850403201212 1 001

**Dr. I Ketut Eddy Purnama, S.T., M.T.**

NIP 19690730199512 1 001

**Undergraduate Study Program of Computer Engineering**

Department of Computer Engineering

Faculty of Faculty of Intelligent Electrical and Informatics Technology

Sepuluh Nopember Institute of Technology

Surabaya

2023

*[Halaman ini sengaja dikosongkan]*



**FINAL PROJECT - EC184801**

***YOLOv7 SMALL OBJECT DETECTION OPTIMIZATION  
TO DETECT AIRBORNE OBJECTS***

**Dion Andreas Solang**

NRP 0721 19 4000 0039

Advisor

**Reza Fuad Rachmadi, S.T., M.T., Ph.D**

NIP 19850403201212 1 001

**Dr. I Ketut Eddy Purnama, S.T., M.T.**

NIP 19690730199512 1 001

**Undergraduate Study Program of Computer Engineering**

Department of Computer Engineering

Faculty of Faculty of Intelligent Electrical and Informatics Technology

Sepuluh Nopember Institute of Technology

Surabaya

2023

*[Halaman ini sengaja dikosongkan]*

# LEMBAR PENGESAHAN

## OPTIMISASI PENDETEKSIAN OBJEK KECIL YOLOv7 UNTUK MENDETEKSI OBJEK-OBJEK *AIRBORNE*

### TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat  
memperoleh gelar Sarjana Teknik pada  
Program Studi S-1 Teknik Komputer  
Departemen Teknik Komputer  
Fakultas Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh: **Dion Andreas Solang**  
NRP. 0721 19 4000 0039

Disetujui oleh Tim Penguji Tugas Akhir:

Reza Fuad Rachmadi, S.T., M.T., Ph.D  
NIP: 19850403201212 1 001

(Pembimbing I)

.....

Dr. I Ketut Eddy Purnama, S.T., M.T.  
NIP: 19690730199512 1 001

(Pembimbing II)

.....

TBA.  
NIP: 66666666 999999 6 999

(Penguji I)

.....

TBA.  
NIP: 66666666 999999 6 999

(Penguji II)

.....

TBA.  
NIP: 66666666 999999 6 999

(Penguji III)

.....

Mengetahui,  
Kepala Departemen Teknik Komputer FTEIC - ITS

Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.  
NIP. 19700313199512 1 001

**SURABAYA**  
**Mei, 2023**

*[Halaman ini sengaja dikosongkan]*

# APPROVAL SHEET

## *YOLOv7 SMALL OBJECT DETECTION OPTIMIZATION TO DETECT AIRBORNE OBJECTS*

### FINAL PROJECT

Submitted to fulfill one of the requirements  
for obtaining a degree Bachelor of Engineering at  
Undergraduate Study Program of Computer Engineering  
Department of Computer Engineering  
Faculty of Faculty of Intelligent Electrical and Informatics Technology  
Sepuluh Nopember Institute of Technology

By: **Dion Andreas Solang**  
NRP. 0721 19 4000 0039

Approved by Final Project Examiner Team:

Reza Fuad Rachmadi, S.T., M.T., Ph.D  
NIP: 19850403201212 1 001

(Advisor I)

.....

Dr. I Ketut Eddy Purnama, S.T., M.T.  
NIP: 19690730199512 1 001

(Co-Advisor II)

.....

TBA.  
NIP: 66666666 999999 6 999

(Examiner I)

.....

TBA.  
NIP: 66666666 999999 6 999

(Examiner II)

.....

TBA.  
NIP: 66666666 999999 6 999

(Examiner III)

.....

Acknowledged,  
Head of Computer Engineering Department ELECTICS - ITS

Dr. Supeno Mardi Susiki Nugroho, S.T., M.T.  
NIP. 19700313199512 1 001

**SURABAYA**  
**May, 2023**



*[Halaman ini sengaja dikosongkan]*

## PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama Mahasiswa / NRP : Dion Andreas Solang / 0721 19 4000 0039  
Departemen : Teknik Komputer  
Dosen Pembimbing / NIP : Reza Fuad Rachmadi, S.T., M.T., Ph.D / 19850403201212 1 001

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "OPTIMISASI PENDETEKSIAN OBJEK KECIL YOLOv7 UNTUK MENDETEKSI OBJEK-OBJEK *AIRBORNE*" adalah hasil karya sendiri, berfsifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, May 2023

Mengetahui  
Dosen Pembimbing

Mahasiswa

Reza Fuad Rachmadi, S.T., M.T., Ph.D  
NIP. 19850403201212 1 001

Dion Andreas Solang  
NRP. 0721 19 4000 0039

*[Halaman ini sengaja dikosongkan]*

## STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP : Dion Andreas Solang / 0721 19 4000 0039  
Department : Computer Engineering  
Advisor / NIP : Reza Fuad Rachmadi, S.T., M.T., Ph.D / 19850403201212 1 001

Hereby declared that the Final Project with the title of "*YOLOv7 SMALL OBJECT DETECTION OPTIMIZATION TO DETECT AIRBORNE OBJECTS*" is the result of my own work, is original, and is written by following the rules of scientific writing.

If in future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with provisions that apply at Sepuluh Nopember Institute of Technology.

Surabaya, May 2023

Acknowledged  
Advisor

Student

Reza Fuad Rachmadi, S.T., M.T., Ph.D  
NIP. 19850403201212 1 001

Dion Andreas Solang  
NRP. 0721 19 4000 0039

*[Halaman ini sengaja dikosongkan]*

## ABSTRAK

Nama Mahasiswa : Dion Andreas Solang  
Judul Tugas Akhir : OPTIMISASI PENDETEKSIAN OBJEK KECIL YOLOv7 UNTUK  
MENDETEKSI OBJEK-OBJEK *AIRBORNE*  
Pembimbing : 1. Reza Fuad Rachmadi, S.T., M.T., Ph.D  
2. Dr. I Ketut Eddy Purnama, S.T., M.T.

Pada penelitian ini kami mengajukan Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Kata Kunci: Roket, *Anti-gravitasi*, Energi, Angkasa.

*[Halaman ini sengaja dikosongkan]*

## ABSTRACT

*Name* : Dion Andreas Solang  
*Title* : *YOLOv7 SMALL OBJECT DETECTION OPTIMIZATION TO DETECT AIR-BORNE OBJECTS*  
*Advisors* : 1. Reza Fuad Rachmadi, S.T., M.T., Ph.D  
2. Dr. I Ketut Eddy Purnama, S.T., M.T.

*In this research, we proposed Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.*

*Keywords: Rocket, Anti-gravity, Energy, Space.*



*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

Puji dan syukur kehadiran Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque.

Penelitian ini disusun dalam rangka Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Saudara tercinta yang telah Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero.
2. Bapak Nikola Tesla, S.T., M.T., selaku Quisque ullamcorper placerat ipsum. Cras nibh.
3. Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl.

Akhir kata, semoga Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus.

Surabaya, Mei 2023

Dion Andreas Solang

*[Halaman ini sengaja dikosongkan]*

# DAFTAR ISI

<b>ABSTRAK</b>	<b>i</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>KATA PENGANTAR</b>	<b>v</b>
<b>DAFTAR ISI</b>	<b>vii</b>
<b>DAFTAR GAMBAR</b>	<b>ix</b>
<b>DAFTAR TABEL</b>	<b>xi</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	1
1.3 Tujuan . . . . .	2
1.4 Batasan Masalah . . . . .	2
<b>2 TINJAUAN PUSTAKA</b>	<b>3</b>
2.1 Teori Dasar . . . . .	3
2.1.1 Arsitektur Famili YOLO . . . . .	3
2.1.2 YOLOv7 . . . . .	4
2.2 Rekalkulasi <i>Anchor</i> . . . . .	5
2.3 Augmentasi Mosaik . . . . .	5
2.4 Penelitian Terkait . . . . .	6
2.4.1 YOLO-Z . . . . .	6
2.4.2 exYOLO . . . . .	6
<b>3 METODOLOGI</b>	<b>7</b>
3.1 Metode Pencarian Solusi Optimisasi . . . . .	7
3.2 Kandidat Modifikasi . . . . .	8
3.2.1 Augmentasi Mosaik . . . . .	8
3.2.2 Rekalkulasi <i>Anchor On-Training</i> . . . . .	8
3.2.3 Modifikasi Neck . . . . .	8

3.2.4	Penambahan YOLO Head . . . . .	9
3.3	Dataset . . . . .	9
3.3.1	Sumber Dataset . . . . .	9
3.3.2	Sampling Dataset . . . . .	10
3.4	Skema Training Model . . . . .	10
3.5	Timeline Pelaksanaan Penelitian . . . . .	10
<b>4</b>	<b>HASIL DAN PEMBAHASAN</b>	<b>13</b>
4.1	Performa Awal . . . . .	13
4.2	Pengaruh Augmentasi Mosaic dan Rekalkulasi Anchor . . . . .	13
4.2.1	Augmentasi Mosaic . . . . .	13
4.2.2	Rekalkulasi Anchor . . . . .	14
4.2.3	Performa Augmentasi Mosaik dan Rekalkulasi Anchor . . . . .	14
4.3	Pengaruh Penggantian <i>Box Loss Function</i> . . . . .	15
4.4	Pengaruh Perubahan Koneksi <i>Neck-Backbone</i> . . . . .	15
4.5	Pengaruh Penambahan <i>Head</i> . . . . .	15
4.6	Pengaruh Pengubahan <i>Head</i> menjadi <i>Anchor-Free</i> . . . . .	15
<b>5</b>	<b>PENUTUP</b>	<b>17</b>
5.1	Kesimpulan . . . . .	17
5.2	Saran . . . . .	17
	<b>DAFTAR PUSTAKA</b>	<b>19</b>
	<b>BIOGRAFI PENULIS</b>	<b>21</b>

## DAFTAR GAMBAR

1.1	Contoh Dataset Objek <i>Airborne</i> . . . . .	2
2.1	Prediksi <i>Anchor Box</i> dan <i>offset</i> dari koordinat latis (Redmon & Farhadi, 2018) .	3
2.2	<i>Feature Pyramid Network</i> pada <i>Neck</i> YOLO . . . . .	4
2.3	Contoh Augmentasi Data Mosaik (Jocher et al., 2022) . . . . .	5
2.4	Contoh Objek <i>Cone</i> yang Terlihat Jauh dari Kamera . . . . .	6
3.1	Urutan Pengerjaan Penelitian . . . . .	7
3.2	Menambah <i>upsampling</i> pada <i>neck</i> . . . . .	9
3.3	Menggunakan <i>feature map</i> dari layer yang lebih di belakang . . . . .	9
3.4	Penambahan <i>Layer Head</i> pada YOLO . . . . .	10
4.1	Contoh Augmentasi Mosaic pada Dataset Training . . . . .	13
4.2	Persebaran Anchor. Kiri: Anchor Lama. Kanan: Anchor Hasil Rekalkulasi . .	14

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

3.1	Distribusi Dataset Training dan Test . . . . .	10
3.2	Distribusi Kelas Dataset . . . . .	11
3.3	Distribusi Sampling Dataset . . . . .	11
3.4	Tabel timeline . . . . .	11
4.1	Performa Modifikasi Augmentasi Mosaic dan Rekalkulasi Anchor . . . . .	14
4.2	Performa Modifikasi Augmentasi Mosaic dan Rekalkulasi Anchor . . . . .	15
4.3	Performa Modifikasi Augmentasi Mosaic dan Rekalkulasi Anchor . . . . .	15



*[Halaman ini sengaja dikosongkan]*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Seiring berkembangnya teknologi *autonomous vehicles*, terdapat banyak keinginan untuk mengaplikasikan teknologi tersebut di berbagai bidang. Salah satu aplikasi teknologi ini di bidang komersil adalah *Amazon Prime Air*. *Amazon Prime Air* memanfaatkan *Autonomous Aerial Vehicle* (AAV) untuk melakukan pengantaran barang dari warehouse ke rumah kostumer secara *autonomous* (Amazon, 2022). Untuk melakukan hal ini, AAV yang digunakan harus mempunyai kemampuan penerbangan *autonomous* yang mumpuni.

Salah satu tantangan terbesar dari penerbangan *autonomous* adalah kemampuan *Sense and Avoid* (SAA) dari AAV. Meskipun di udara terdapat ruang gerak yang luas, tetap terdapat resiko AAV akan menabrak objek di udara. Objek - objek tersebut dapat berupa helikopter, pesawat, burung, misil, dan lain - lain. Objek - objek ini juga sering disebut dengan objek *airborne* (“Airborne Object Tracking Challenge”, 2021).

Salah satu sensor yang dapat digunakan untuk melakukan SAA adalah kamera. Kamera memiliki bobot yang cukup ringan, sehingga dapat dibawah oleh AAV. Selain itu, kamera juga memiliki harga yang relatif lebih murah dibandingkan sensor - sensor seperti LiDAR atau Radar.

Dengan memilih kamera sebagai sensor, maka dibutuhkan suatu model computer vision untuk diaplikasikan pada kamera tersebut. Objek - objek *airborne* akan tampak sangat kecil pada kamera seperti yang dapat dilihat pada Gambar 1.1. Beberapa dataset kamera *airborne* yang memiliki resolusi 20482448 pixel, objeknya dapat berukuran 4 (0.00008% luas resolusi) hingga 1000 pixel (0.01% luas resolusi) sehingga terlihat sangat kecil (“Airborne Object Tracking Dataset”, 2021). Oleh karena itu, dibutuhkan suatu model yang dapat mendeteksi objek - objek yang sangat kecil sehingga dapat mendeteksi objek *airborne*.

YOLOv7 merupakan model state-of-the-art untuk melakukan pendeteksian objek secara real-time. YOLOv7 memiliki akurasi tertinggi dari semua model pendeteksi objek dengan kecepatan deteksi 30 FPS (yang terpublikasi) pada GPU Nvidia V100. Terdapat versi scaled dari YOLOv7 yang memiliki jumlah parameter yang lebih kecil dan dapat diaplikasikan pada device edge computing (Wang et al., 2022). Oleh karena itu, YOLOv7 ini cocok untuk digunakan pada AAV di mana dibutuhkan suatu pendeteksi objek yang real-time.

### 1.2 Rumusan Masalah

YOLOv7 bukan merupakan model deteksi objek umum sehingga YOLOv7 tidak didesain untuk melakukan deteksi objek kecil seperti objek-objek *airborne*. Oleh karena itu, dibuatlah rumusan masalah seperti berikut:

- Apa solusi yang dapat diaplikasikan pada YOLOv7 agar kemampuan deteksi objek *airborne*-nya dapat dioptimalisasi?

### 1.3 Tujuan



Gambar 1.1: Contoh Dataset Objek *Airborne*

Adapun tujuan dari tugas akhir ini adalah untuk menemukan solusi untuk mengoptimisasi kemampuan YOLOv7 mendeteksi objek airborne.

## 1.4 Batasan Masalah

Optimisasi kemampuan deteksi objek kecil hanya akan dilakukan dengan memodifikasi YOLOv7. Modifikasi yang diaplikasikan tidak boleh menyebabkan YOLOv7 untuk tidak dapat melakukan pendeteksian secara *real time*. Target pengaplikasian model ini adalah untuk AAV dengan *computational resource* yang terbatas sehingga model hasil modifikasi harus cukup ringan untuk hal tersebut.

## BAB II

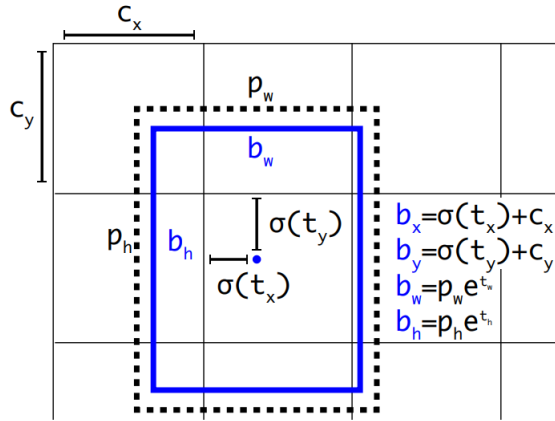
### TINJAUAN PUSTAKA

#### 2.1 Teori Dasar

##### 2.1.1 Arsitektur Famili YOLO

Arsitektur famili YOLO pada dasarnya terbagi akan 3 bagian yaitu *head*, *neck*, dan *backbone*. Setiap bagian ini mempunyai fungsi masing-masing. Berikut adalah penjelasan fungsi dan cara kerja dari ketiga bagian tersebut.

##### *Layer Head YOLO dan Anchor Box*

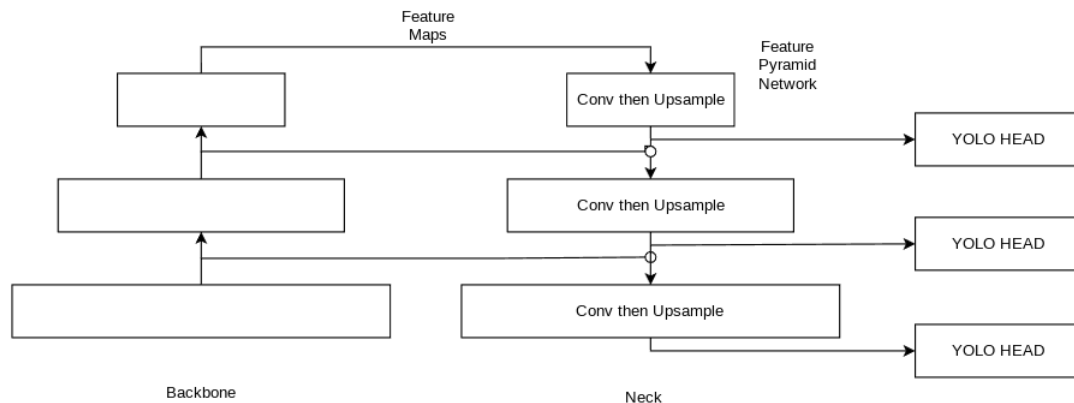


Gambar 2.1: Prediksi *Anchor Box* dan *offset* dari koordinat latis (Redmon & Farhadi, 2018)

Arsitektur famili YOLO yang dipublikasikan setelah YOLOv2 terus menggunakan *anchor box* untuk melakukan deteksi (Redmon & Farhadi, 2016; Redmon & Farhadi, 2018; Bochkovski et al., 2020; Wang et al., 2020; Jocher et al., 2022; Wang et al., 2021; Wang et al., 2022). *Anchor boxes* merupakan beberapa *Bounding Box* yang telah terdefiniskan. Arsitektur YOLO akan memprediksi probabilitas *anchor box* berada pada suatu koordinat latis beserta dengan *offset anchor box* tersebut untuk menepatkan *anchor box* pada objek yang dideteksi. Penggunaan *anchor box* ini dapat meningkatkan akurasi deteksi karena *neural network* hanya perlu mencari titik tengah objek dan *error* dimensi *boudning box* dengan menggunakan *offset* (Redmon & Farhadi, 2018). Hal ini lebih sederhana daripada mencari titik-titik *bounding box* secara independen sehingga lebih mudah untuk dipelajari oleh *neural network*.

Prediksi *bounding boxes* terjadi di bagian *head* dari arsitektur YOLO. Bagian *head* dari YOLO akan mengambil beberapa hasil *upsampling* yang terjadi pada *neck* YOLO, dan kemudian melakukan prediksi *anchor boxes* dari hasil tersebut. Hasil prediksi *Head* YOLO pada suatu tingkatan *upsampling* berupa tensor dengan ukuran  $N \times N \times [A \times (4 + 1 + C)]$  dengan  $N$  sebagai dimensi hasil *upsampling*-nya,  $A$  sebagai jumlah *anchor boxes* untuk *scaling* tersebut, dan  $C$  sebagai jumlah kelas prediksi. Angka 4 merepresentasikan 4 *offset*  $b_x, b_y, b_w, b_h$  seperti pada Gambar 2.1 dan angka 1 merepresentasikan *objectness score* dari prediksi *bounding box*.

## Neck YOLO



Gambar 2.2: *Feature Pyramid Network* pada Neck YOLO

Neck dari YOLO merupakan *layer-layer* dimana head YOLO mengambil fitur untuk dilakukan deteksi *bounding box*. Pada YOLOv3 Redmon and Farhadi (2018), arsitektur neck dibuat menyerupai *Feature Pyramid Network* (FPN) seperti pada Gambar 2.2. Pada versi-versi YOLO selanjutnya, bentuk neck ini tidak banyak berubah dan pada dasarnya tetap mempertahankan bentuk *pyramid*-nya.

Penaikkan tingkatan *pyramid* dari FPN merupakan *upsampling* dari *feature map* yang dihasilkan *backbone*. Output tiap tingkatan pada FPN di neck inilah yang diinputkan pada head YOLO. Melakukan prediksi pada tingkatan *upsampling* yang berbeda-beda dapat membuat *neural network* mendapatkan lebih banyak informasi semantik dan informasi yang lebih detail sehingga dapat lebih akurat dalam mendeteksi objek besar maupun kecil.

## Backbone YOLO

Backbone dari YOLO merupakan bagian yang mengekstrak fitur dari citra yang diinputkan. Hasil ekstraksi fitur ini akan diinputkan pada neck yang kemudian akan diupsampling olehnya. Model-model YOLO dapat menggunakan *feature extractor* dari model-model klasifikasi citra sebagai backbone-nya. Sebagai contoh, salah satu varian YOLO, YOLO-Z menggunakan DenseNet sebagai backbone-nya sedangkan arsitektur YOLO dasarnya, YOLOv5 menggunakan backbone YOLOv5v7.0 (Benjumea et al., 2021).

### 2.1.2 YOLOv7

YOLOv7 merupakan pendeteksi objek *real time* dengan skor akurasi tertinggi pada dataset COCO di tahun 2022. Pada YOLOv7, dilakukan beberapa perubahan untuk meningkatkan akurasi dan kecepatan deteksinya. Perubahan-perubahan tersebut dilakukan pada arsitekturnya dan pada *bag-of-freebies*-nya.

Perubahan arsitektur dilakukan pada backbone. YOLOv7 menggunakan *Extended Efficient Layer Aggregation Network* (E-ELAN) sebagai backbone, berbeda dengan leluhurnya YOLOv4 yang menggunakan CSP-Darknet. E-ELAN merupakan arsitektur *neural network* yang efisien karena E-ELAN didesain dengan mengontrol *gradient path* terpanjang yang terpendek. Karena efisiensinya, arsitektur E-ELAN ini dapat meningkatkan kecepatan deteksi dan akurasi. (Wang et al., 2022)

*Bag-of-freebies* merupakan kumpulan metode peningkatan akurasi yang tidak meningkatkan

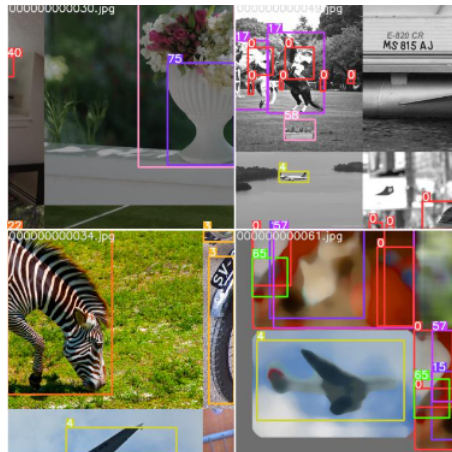
*cost inference* (Bochkovskiy et al., 2020). Pada YOLOv7, ditambahkan beberapa *bag-of-freebies* yang dapat dilatih seperti *re-parameterized convolution* dan *extra auxiliary head* di tengah-tengah *neural network*. Selain kedua itu, YOLOv7 juga menambahkan *trainable bag-of-freebies* dari YOLOR seperti EMA, *Implicit Knowledge*, dan *conv-bn topology Batch Normalization* (Wang et al., 2022).

## 2.2 Rekalkulasi *Anchor*

*Anchor box* dari model-model *pre-trained* YOLO pada umumnya mengoptimisasi *anchor box* modelnya pada dataset COCO. Ukuran *anchor box* yang akan digunakan pada model YOLO dapat dikonfigurasi agar lebih sesuai dengan dataset yang akan digunakan untuk melatih model YOLO. Penyesuaian ini dapat meningkatkan IoU(*Intersection Over Union*) prediksi model dengan *ground truth* sehingga meningkatkan akurasi.

Penyesuaian *anchor box* dapat dilakukan pada saat sebelum training atau pada saat training. Penyesuaian *anchor box* sebelum training dapat dilakukan dengan cara mengkonfigurasi secara manual tiap ukuran *anchor box* atau dengan menggunakan algoritma *clustering*. Penggunaan algoritma *clustering* akan lebih baik karena setiap ukuran *anchor box*-nya disesuaikan dengan pengelompokan-pengelompokan ukuran *bounding box* natural yang terdapat pada dataset. Untuk penyesuaian saat training, dapat digunakan algoritma dari Zhong et al. (2018). Algoritma ini akan mengoptimisasi ukuran-ukuran *anchor* bukan hanya berdasarkan dataset, namun berdasarkan kemampuan dari *neural network* pendeteksi objeknya. Untuk melakukan hal tersebut, algoritma ini akan memanfaatkan *back propagation localization loss* untuk rekalkulasi *anchor*.

## 2.3 Augmentasi Mosaik



Gambar 2.3: Contoh Augmentasi Data Mosaik (Jocher et al., 2022)

Augmentasi mosaik merupakan teknik augmentasi yang baru dikenalkan pada YOLOv4. Teknik augmentasi ini akan memilih 4 gambar dari dataset, memotong gambar-gambar tersebut dan menggabungkannya secara acak pada satu gambar seperti pada Gambar 2.3. Hasil dari penggabungan itu membuat gambar terlihat seperti mosaik. Teknik augmentasi ini mampu meningkatkan akurasi model (Bochkovskiy et al., 2020).

## 2.4 Penelitian Terkait

### 2.4.1 YOLO-Z

YOLO-Z merupakan arsitektur famili YOLO yang modifikasi dari YOLOv5 (Benjumea et al., 2021). Modifikasi-modifikasi yang dilakukan meliputi pergantian *backbone*, *neck*, dan jumlah *anchor Backbone* dari YOLOv5r5.0 menjadi DenseNet yang di-*downscale*. *Neck* dari YOLO-Z juga diganti dari PanNet menjadi FPN dan biFPN tergantung pada *scale* YOLO-Z yang digunakan.

Modifikasi pada YOLO-Z didesain untuk mendeteksi objek kecil untuk tujuan melakukan deteksi *cone* yang nampak jauh pada lintasan *autonomous racing* secara *real time* (lihat Gambar 2.4). Modifikasi-modifikasi dibuktikan dapat meningkatkan kemampuan pendeteksian objek kecil (Benjumea et al., 2021). Oleh karena itu, untuk meningkatkan kemampuan mendeteksi objek kecil YOLOv7, beberapa modifikasi yang dilakukan YOLO-Z pada YOLOv5 dapat diaplikasikan.



Gambar 2.4: Contoh Objek *Cone* yang Terlihat Jauh dari Kamera

### 2.4.2 exYOLO

exYOLO merupakan hasil modifikasi arsitektur YOLOv3 (Xiao, 2021). Pada exYOLO, dilakukan modifikasi *neck* dengan menambahkan suatu *Receptive Field Block* sebelum penggabungan *feature map* yang akan diupsampling. Modifikasi-modifikasi ini membuat exYOLO memiliki skor mAP yang lebih tinggi daripada YOLOv3 pada dataset PASCAL VOC 2007.

## BAB III

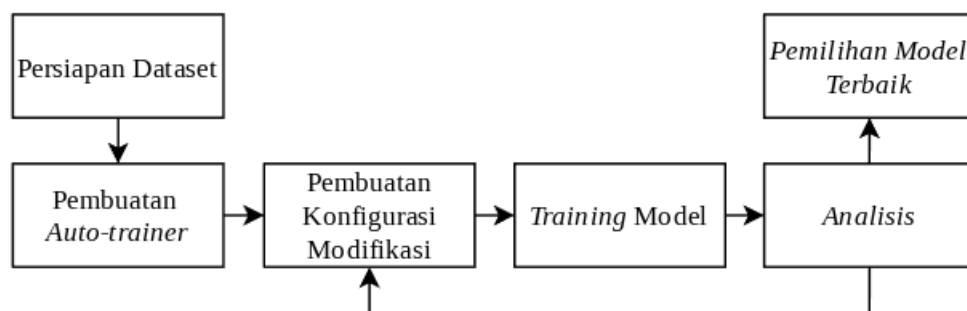
# METODOLOGI

### 3.1 Metode Pencarian Solusi Optimisasi

Untuk mencari solusi optimisasi pendeteksian objek kecil YOLOv7 yang terbaik, akan dilakukan penambahan atau perubahan *bag-of-freebies*, *bag-of-specials*, dan arsitektur YOLOv7. Setiap modifikasi-modifikasi itu akan diaplikasikan secara independen dan kombinatif. Yang dimaksud dengan kombinatif adalah modifikasi-modifikasi akan dikombinasikan menjadi 1 model YOLOv7.

Setiap kombinasi modifikasi, independen atau tidak, akan diuji kemampuannya mendeteksi objek *airborne*. Solusi optimisasi terbaik akan ditentukan berdasarkan metrik  $AP_{50}$ . Subbab 3.2 akan membahas tentang kandidat modifikasi-modifikasi yang dapat dilakukan.

Tahapan pencarian solusi optimisasi sendiri dapat dibagi menjadi enam tahap. Tahap-tahap tersebut adalah Persiapan Dataset, Pembuatan *Auto-trainer*, Pembuatan Konfigurasi Modifikasi, *Training Model*, Analisis, dan Pemilihan Model Terbaik. Urutan pengerjaan dari tahap-tahap ini dapat dilihat pada Gambar 3.1.



Gambar 3.1: Urutan Pengerjaan Penelitian

Pada tahap persiapan dataset, akan dilakukan pengunduhan dataset dari “Airborne Object Tracking Dataset” (2021). Gambar-gambar pada dataset ini kemudian akan di-*sampling* dan didistribusikan menjadi dataset *training*, validasi, dan pengujian. Pada pendistribusian dataset ini juga akan dilakukan *balancing* antar kelas dan dataset positif negatif. *Balancing* dilakukan agar tidak ada kelas yang mendominasi pada dataset.

Selanjutnya, di tahap pembuatan *auto-trainer*, akan dilakukan pembuatan program yang dapat dengan otomatis membangun dan melatih *neural network* modifikasi YOLOv7. Pembuatan *auto-trainer* ini dilakukan agar proses-proses pengerjaan tahapan-tahapan selanjutnya menjadi dapat dilakukan dengan lebih mudah.

Setelah itu, akan dilakukan pembuatan konfigurasi-konfigurasi modifikasi. Konfigurasi modifikasi YOLOv7 akan dibuat agar dapat diinputkan pada *auto-trainer*. Konfigurasi modifikasi akan berisi kombinasi modifikasi-modifikasi yang ada pada subbab 3.2.

Tahapan selanjutnya adalah *Training* model. Pada tahap ini, konfigurasi-konfigurasi modi-



fikasi pada tahapan sebelumnya akan dibangun dan kemudian dilatih. Hasil dari tahap ini adalah *weights neural network* modifikasi YOLOv7, histori pelatihannya, dan metrik-metriknya pada dataset uji.

Pada tahap analisis, hasil dari tahap sebelumnya akan dianalisis untuk mencari tahu performa model-model hasil modifikasi. Analisis juga dilakukan untuk menemukan *gap* kandidat modifikasi lain yang masih bisa dieksplorasi untuk meningkatkan kemampuan pendeteksian objek kecil YOLOv7. Ketika suatu kandidat modifikasi seperti itu ditemukan, maka akan dilakukan kembali pembuatan konfigurasi modifikasi untuk menguji kandidat modifikasi tersebut.

Tahapan terakhir adalah pemilihan model terbaik. Pada tahapan ini akan dipilih model yang memiliki performa terbaik dari antara model-model hasil modifikasi lainnya. Pemilihan model akan dilakukan dengan berdasarkan pada skor mAP tertinggi. Untuk mempertahankan solusi optimisasi yang dapat melakukan deteksi secara *real time*, model yang akan dipilih hanyalah model yang dapat melakukan deteksi dengan kecepatan minimum 10 FPS pada *onboard computing device* seperti Jetson TX2 atau *compute device* lainnya.

## 3.2 Kandidat Modifikasi

### 3.2.1 Augmentasi Mosaik

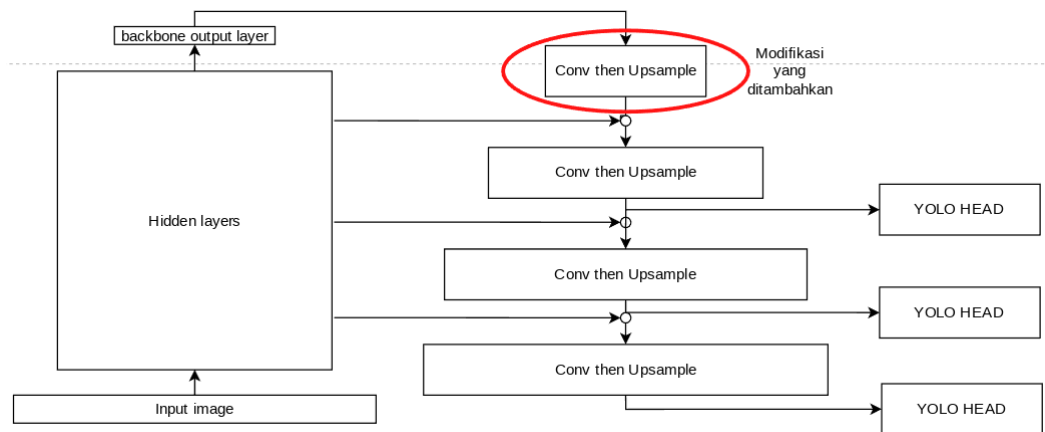
Seperti yang dibahas pada subbab 2.3, augmentasi mosaik pada dataset mampu meningkatkan akurasi deteksi objek-objek kecil dari model. Oleh karena itu, akan dilakukan eksperimen menambahkan augmentasi mosaik pada YOLOv7 untuk melihat apakah augmentasi ini akan meningkatkan akurasi, khususnya pada dataset objek kecil.

### 3.2.2 Rekalkulasi *Anchor On-Training*

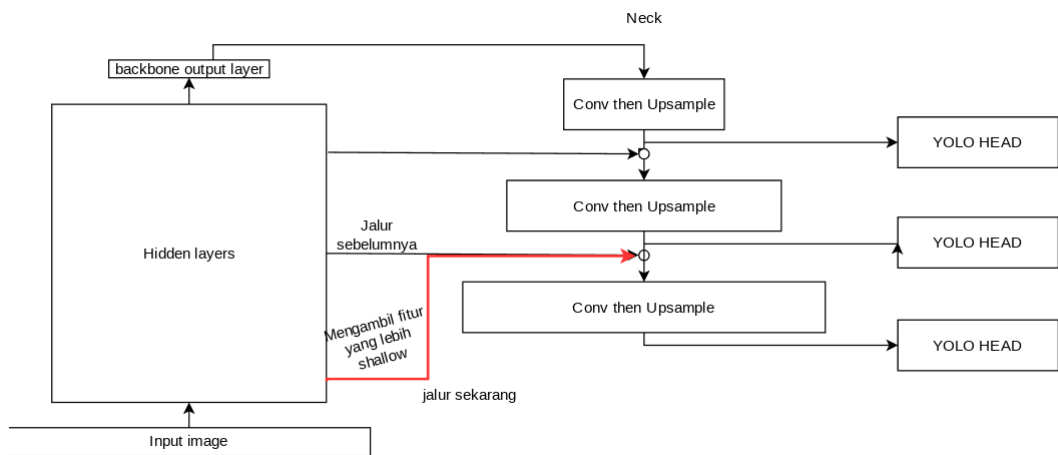
Yang dimaksud dengan rekalkulasi *anchor on-training* adalah ketika ukuran-ukuran *anchor box* direkalkulasi pada saat training. Berbeda dengan *clustering pre-training* seperti pada YOLOv2 (Redmon & Farhadi, 2016), ukuran-ukuran *anchor* akan di-*learning* bersama dengan pendeteksi objeknya. Untuk melakukan hal ini, digunakan algoritma optimisasi *anchor box* yang mirip dengan algoritma Zhong et al. (2018). Pada bagian *head*, akan ditempelkan suatu layer yang akan mengoutputkan faktor *rescaling* dari tiap *anchor box*. Bagian tersebut akan di-*training* bersama dengan YOLOv7 *anchor box* akan teroptimisasi tidak hanya pada dataset, namun pada keseluruhan *neural network* juga.

### 3.2.3 Modifikasi Neck

Seperti pada penelitian-penelitian terkait di subbab 2.4, modifikasi *neck* dapat dilakukan untuk meningkatkan akurasi pendeteksian objek kecil. Modifikasi *neck* dapat dilakukan dengan menambahkan layer upsampling seperti pada Gambar 3.2 atau dengan memindahkan sumber feature map untuk dari beberapa layer neck lebih jauh ke belakang seperti pada Gambar 3.3. Penambahan layer upsampling dapat membuat neural network untuk mendapatkan feature-map yang lebih detail sehingga dapat melakukan pendeteksian objek kecil dengan lebih baik. Pemindahan sumber feature map ke belakang dapat dilakukan untuk mengantisipasi fitur objek kecil yang dapat hilang ketika layer neural network semakin dalam. Dengan memindahkannya lebih ke belakang, YOLOv7 akan melakukan deteksi pada layer dengan abstraksi yang lebih rendah (sedikit informasi yang hilang).



Gambar 3.2: Menambah *upsampling* pada *neck*



Gambar 3.3: Menggunakan *feature map* dari layer yang lebih di belakang

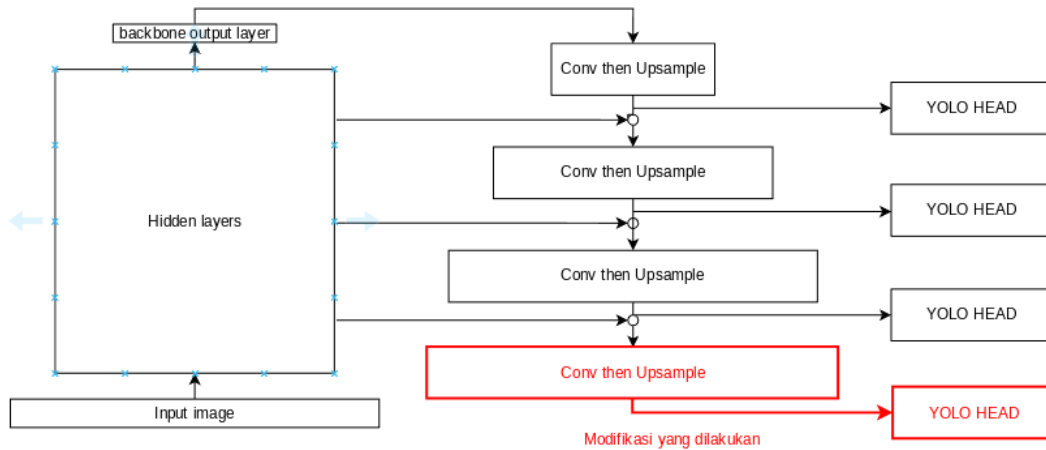
### 3.2.4 Penambahan YOLO Head

Penambahan YOLO head dapat membuat YOLOv7 melakukan deteksi pada skala yang lebih tinggi. Hal ini akan berpengaruh pada kemampuan pendeteksian objek kecil. Dengan melakukan pendeteksian pada skala yang lebih beragam, YOLOv7 dapat mendeteksi objek yang besar maupun kecil. Perhatikan bahwa penambahan YOLO Head akan diikuti dengan penambahan *layer upsampling* pada *neck* seperti di Gambar 3.4.

## 3.3 Dataset

### 3.3.1 Sumber Dataset

Dataset untuk objek-objek *airborne* didapatkan dari “Airborne Object Tracking Dataset” (2021) dan dihost pada suatu server AWS S3 Bucket. Dataset ini berisi video-video monokromatik penerbangan UAV. Terdapat 4 kelas pada dataset ini yaitu pesawat, helikopter, burung, dan *other*. Distribusi dataset training dan uji dapat dilihat pada tabel 3.1 sedangkan distribusi kelas dari dataset dapat dilihat pada Tabel 3.2.



Gambar 3.4: Penambahan *Layer Head* pada YOLO

Tabel 3.1: Distribusi Dataset Training dan Test

Pembagian Dataset	Ukuran (TB)	Sekuen penerbangan UAV	Jumlah Gambar	Jumlah Label
Training	11,3	4154	4975765	2891891
Validation + Test	2.1	789	943852	496075
Total	13,4	4943	5919617	3387966

### 3.3.2 Sampling Dataset

Karena jumlah dataset pada “Airborne Object Tracking Dataset” (2021) berukuran sangat besar, hanya sebagian dari dataset tersebut akan digunakan untuk *training* dan *test*. Akan diambil total 60000 gambar dari dataset dengan pembagian sesuai dengan Tabel 3.3

### 3.4 Skema Training Model

Untuk melatih berbagai modifikasi YOLOv7, akan dibuat suatu *auto-trainer*. *Auto-trainer* ini akan menerima suatu *file* konfigurasi modifikasi YOLOv7, dan dengan otomatis membangun arsitektur YOLOv7 yang termodifikasi dan melatihnya. Setelah mendapatkan model modifikasi YOLOv7 yang sudah dilatih, *auto-trainer* akan menguji model tersebut dengan dataset uji. Metrik-metrik pengujian, grafik histori *training loss vs validation loss*, dan *weights* dari model kemudian akan dikirim ke user. Dengan membuat *auto-trainer* ini, proses pelatihan model dan pelaporan hasil menjadi terotomasi sehingga akan mempermudah proses penelitian.

### 3.5 Timeline Pelaksanaan Penelitian

Tabel 3.2: Distribusi Kelas Dataset

Pembagian	Total Objek	Pesawat	Helikopter	Burung	<i>Other</i>
Training	2,89 juta	0,79 juta	1,22 juta	0,33 juta	0,54 juta
Validation + Test	0,50 juta	0,13 juta	0,17 juta	0,06 juta	0,14 juta
Total	3,39 juta	0,92 juta	1,39 juta	0,39 juta	0,69 juta

Tabel 3.3: Distribusi Sampling Dataset

Pembagian	Total	Presentase				
	Gambar	Pesawat	Helikopter	Burung	<i>Other</i>	Negatif
Training	54000	23,75%	23,75%	23,75%	23,75%	5%
Validation	3000	20%	20%	20%	20%	20%
Test	3000	20%	20%	20%	20%	20%

Tabel 3.4: Tabel timeline

Kegiatan	Minggu															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Persiapan Dataset																
Pemb. <i>Auto-trainer</i>																
Pemb. Konfigurasi Modifikasi																
Training Model																
Analisis																
Pemb. Laporan																

*[Halaman ini sengaja dikosongkan]*

## BAB IV

# HASIL DAN PEMBAHASAN

Pada bab ini, akan dipaparkan pengaruh modifikasi-modifikasi yang dilakukan pada YOLOv7.

### 4.1 Performa Awal

Untuk mengukur pengaruh dari modifikasi-modifikasi yang dilakukan pada YOLOv7, maka hal pertama yang harus dilakukan adalah mengukur performa YOLOv7 tanpa segala modifikasi yang diajukan pada bab 3.2. Arsitektur YOLOv7 *plain* ini di-*train* pada 500 sampel data dari “Airborne Object Tracking Dataset” (2021) dengan 300 epoch dan batch size 1. Dengan aturan tersebut, ditemukan bahwa model *plain* tidak mampu untuk mendeteksi objek apapun pada dataset uji, dengan kriteria “terdeteksi”  $IoU > 0.5$  ( $mAP@.5 = 0$ ).

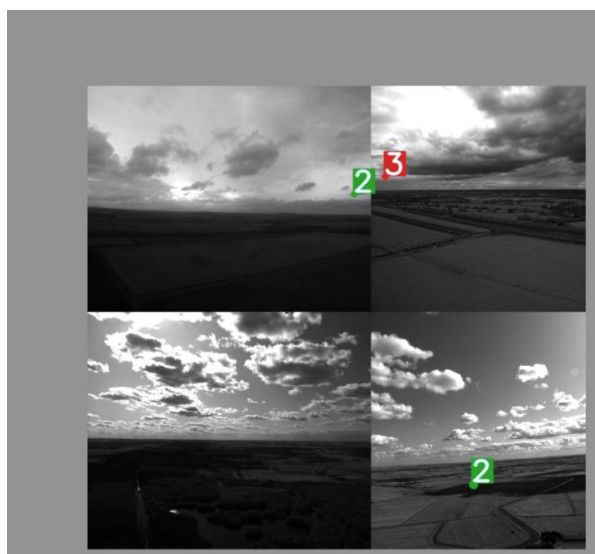
Untuk keperluan komparasi dengan performa-performa dari modifikasi pada YOLOv7, model *plain* ini akan selanjutnya disebut sebagai YOLOv7-*plain*.

### 4.2 Pengaruh Augmentasi Mosaic dan Rekalkulasi Anchor

Terdapat 3 modifikasi yang diujikan pada bagian ini, yaitu YOLOv7-*plain* yang ditambahkan augmentasi mosaic, YOLOv7-*plain* yang direkalkulasi anchor, dan YOLOv7-*plain* yang ditambahkan augmentasi mosaic dan rekalkulasi anchor.

#### 4.2.1 Augmentasi Mosaic

Proses melakukan augmentasi mosaic cukup *straightforward*, augmentasi ini dilakukan pada beberapa data training. Contoh hasil augmentasi dapat dilihat pada gambar 4.1



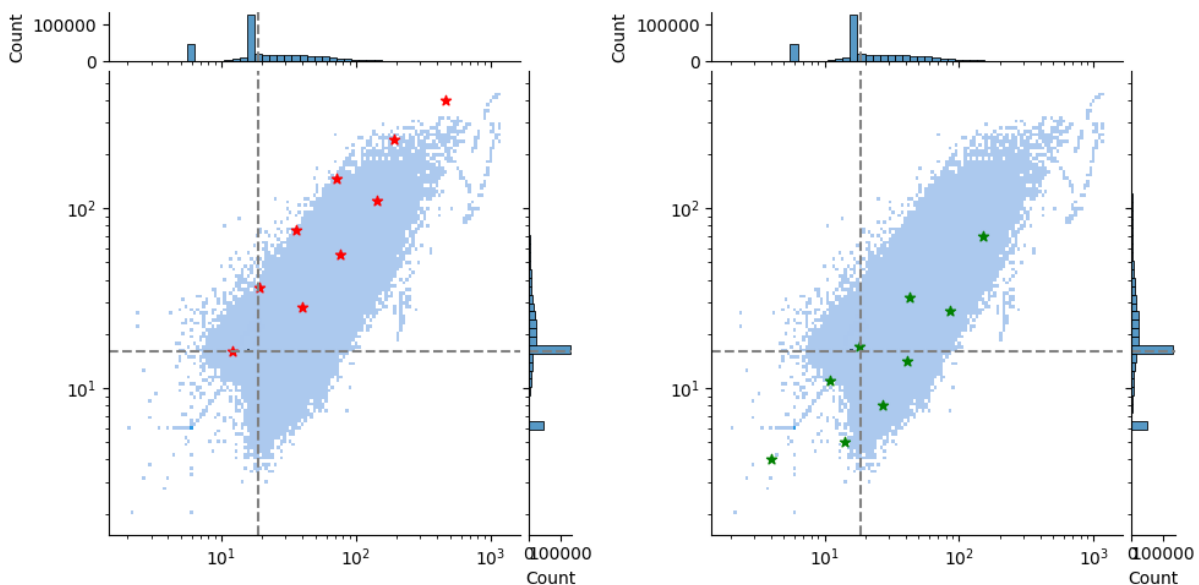
Gambar 4.1: Contoh Augmentasi Mosaic pada Dataset Training

## 4.2.2 Rekalkulasi Anchor

Rekalkulasi anchor dilakukan dengan mengkluster data training ke 9 centroid menggunakan algoritma k-means. Sembilan centroid tersebut digunakan sebagai anchor, 3 untuk tiap head pada arsitektur YOLO (terdapat 3 head). Persebaran anchor sebelum dan sesudah direkalkulasi dapat dilihat pada Tabel 4.1 dan Gambar 4.2

Tabel 4.1: Performa Modifikasi Augmentasi Mosaic dan Rekalkulasi Anchor

Layer Head	Anchor Lama	Hasil Rekalkulasi Anchor
1	[12,16], [19,36], [40,28]	[4,4], [14,5], [11,11]
2	[36,75], [76,55], [72,146]	[27,8], [18,17], [41,14]
3	[142,110], [192,243], [459,401]	[43,32], [86,27], [149,70]



Gambar 4.2: Persebaran Anchor. Kiri: Anchor Lama. Kanan: Anchor Hasil Rekalkulasi

Jika kita memperhatikan persebaran anchor sebelum dan sesudah direkalkulasi pada Gambar 4.2, dapat kita lihat bahwa anchor hasil rekalkulasi lebih mencakup seluruh persebaran dataset daripada anchor lama. 8 dari 9 anchor lama bertempat di kuadran pertama dari garis median(garis putus-putus). Hal ini berarti 8 anchor tersebut hanya mampu mendeteksi sekitar 25% dari objek-objek pada dataset. Sedangkan, anchor hasil rekalkulasi menempatkan anchor di setiap kuadran.

## 4.2.3 Performa Augmentasi Mosaik dan Rekalkulasi Anchor

Performa dari tiap modifikasi dapat dilihat pada tabel 4.2. Pada tabel tersebut, terlihat bahwa YOLOv7 mampu untuk mendeteksi beberapa objek pada dataset uji ketika diberi augmentasi mosaic pada data train dan direkalkulasi anchornya.

Hanya modifikasi nomor 3 yang mampu melakukan deteksi, maka modifikasi tersebut dijadikan baseline untuk modifikasi-modifikasi lainnya. Untuk mempermudah komparasi penambahan modifikasi-modifikasi selanjutnya, maka model ini akan disebut sebagai YOLOv7-base.

Tabel 4.2: Performa Modifikasi Augmentasi Mosaic dan Rekalkulasi Anchor

No	Modifikasi	mAP@50
0	<b>yolov7-plain</b>	0%
1	<b>mosaic</b>	0%
2	<b>rekalkulasi anchor</b>	0%
3	<b>mosaic + rekalkulasi anchor</b>	<b>11,2%</b>
Peningkatan		<b>+11,2%</b>

### 4.3 Pengaruh Penggantian *Box Loss Function*

Dengan menggunakan YOLOv7-base sebagai baseline, Box Loss function dari YOLOv7 diganti menjadi EIoU. Telah juga dilakukan percobaan menggunakan convexication pada EIoU. Hasil dari pengujian dapat dilihat pada Tabel 4.3

Tabel 4.3: Performa Modifikasi Augmentasi Mosaic dan Rekalkulasi Anchor

No	Modifikasi	mAP@50
0	<b>yolov7-base + CIoU (original)</b>	<b>11,2%</b>
1	<b>yolov7-base + EIoU</b>	0%
2	<b>yolov7-base + EIoU + Covexication</b>	4,92%
Peningkatan		<b>-6,28%</b>

Ternyata, meskipun EIoU memiliki performa lebih baik daripada CIoU pada Peng and Yu (2021) dengan convexication tidak lebih baik daripada CIoU.

### 4.4 Pengaruh Perubahan Koneksi *Neck-Backbone*

### 4.5 Pengaruh Penambahan *Head*

### 4.6 Pengaruh Pengubahan *Head* menjadi *Anchor-Free*



*[Halaman ini sengaja dikosongkan]*

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil pengujian yang Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. sebagai berikut:

1. Pembuatan Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus.
2. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa.
3. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna.

#### **5.2 Saran**

Untuk pengembangan lebih lanjut pada Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. antara lain:

1. Memperbaiki Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus.
2. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa.
3. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna.

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- Airborne object tracking challenge*. (2021). Retrieved September 1, 2022, from <https://www.aicrowd.com/challenges/airborne-object-tracking-challenge>
- Airborne object tracking dataset*. (2021). Retrieved September 1, 2022, from <https://registry.opendata.aws/airborne-object-tracking>
- Amazon. (2022). *Amazon prime air prepares for drone deliveries*. Retrieved September 1, 2022, from <https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries>
- Benjumea, A., Teeti, I., Cuzzolin, F., & Bradley, A. (2021). Yolo-z: Improving small object detection in yolov5 for autonomous vehicles.
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection.
- Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., Michael, K., TaoXie, Fang, J., imyhxy, & et al. (2022). Ultralytics/yolov5: V7.0 - yolov5 sota realtime instance segmentation. <https://doi.org/10.5281/zenodo.7347926>
- Peng, H., & Yu, S. (2021). A systematic iou-related method: Beyond simplified regression for better localization. *CoRR*, *abs/2112.01793*. <https://arxiv.org/abs/2112.01793>
- Redmon, J., & Farhadi, A. (2016). Yolo9000: Better, faster, stronger.
- Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement.
- Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2020). Scaled-YOLOv4: Scaling cross stage partial network.
- Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.
- Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2021). You only learn one representation: Unified network for multiple tasks.
- Xiao, J. (2021). exYOLO: A small object detector based on YOLOv3 object detector. *Procedia Computer Science*, *188*, 18–25. <https://doi.org/10.1016/j.procs.2021.05.048>
- Zhong, Y., Wang, J., Peng, J., & Zhang, L. (2018). Anchor box optimization for object detection.

*[Halaman ini sengaja dikosongkan]*

## BIOGRAFI PENULIS



Dion Andreas Solang, lahir pada Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

*[Halaman ini sengaja dikosongkan]*