



**ANALISIS PERBANDINGAN AKURASI DAN PERFORMA  
ALGORITMA DETEKSI OBJEK PADA YOLO v3 DENGAN SINGLE  
SHOT MULTIBOX DETECTOR (SSD)**

**Fathur Rahman Haikal      4617010028**

**LAPORAN SKRIPSI**

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER  
POLITEKNIK NEGERI JAKARTA  
2021**



**ANALISIS PERBANDINGAN AKURASI DAN PERFORMA  
ALGORITMA DETEKSI OBJEK PADA YOLO v3 DENGAN SINGLE  
SHOT MULTIBOX DETECTOR (SSD)**

**LAPORAN SKRIPSI**

**Dibuat untuk Melengkapi Syarat-Syarat yang Diperlukan untuk  
Memperoleh Diploma Empat Politeknik**

**Fathur Rahman Haikal  
4617010028**

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER  
POLITEKNIK NEGERI JAKARTA**

**2021**



## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

### HALAMAN PERNYATAAN ORISINALITAS

Skripsi/Tesis/Disertasi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Fathur Rahman Haikal

NPM : 4617010028

Tanggal : 30 Juli 2021

Tanda Tangan :



© Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

LEMBAR PENGESAHAN

Skripsi diajukan oleh:

Nama : Fathur Rahman Haikal  
 NIM : 4617010028  
 Program Studi : Teknik Informatika  
 Judul Skripsi : Analisis Perbandingan Akurasi dan Performa Algoritma Deteksi Objek Pada YOLO v3 dengan *Single Shot Multibox Detector* (SSD).

Telah diuji oleh tim penguji dalam Sidang Skripsi pada Hari Selasa, 10 Agustus 2021 dandinyatakan LULUS

Disahkan Oleh:

Pembimbing	: Iklima Ermis Ismail, S.Kom, M.Kom.	(.....)
Penguji I	: Euis Oktavianti, S.Si, M.T.I.	(.....)
Penguji II	: Drs. Abdul Aziz, M.M.S.I.	(.....)
Penguji III	: Bambang Warsuta, S.Kom., M.T.I.	(.....)

POLITEKNIK  
 NEGERI  
 Mengetahui :  
 Jurusan Teknik Informatika dan Komputer  
 Ketua

Mauldy Laya, S.Kom., M.Kom.

NIP. 197802112009121003





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

**KATA PENGANTAR**

Puji dan syukur penulis ucapkan kepada Allah *subhanahu wa ta'ala* atas limpahan karunia dan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Analisis Perbandingan Akurasi dan Performa Algoritma Deteksi Objek Pada YOLO v3 dengan *Single Shot Multibox Detector* (SSD)”. Pada kesempatan kali ini, izinkan penulis untuk menyampaikan ucapan terimakasih atas dukungan nya dalam menyelesaikan skripsi ini kepada:

- a. Allah *subhanahu wa ta'ala*, Tuhan yang Maha Esa, yang telah memberikan penulis kesehatan dan akal sehat sehingga laporan ini dapat terselesaikan dengan baik.
- b. Ibu Iklima Ermis Ismail, S.Kom., M.Kom. selaku dosen pembimbing skripsi yang telah meluangkan waktu, tenaga dan pikiran untuk membimbing dan membantu penulis dalam menyusun laporan skripsi.
- c. Orang tua dan keluarga penulis yang setiap saat mendoakan penulis serta memberikan dukungan dan bantuan moral maupun material kepada penulis.
- d. Sahabat dan teman-teman yang telah banyak membantu penulis dalam menyelesaikan penyusunan skripsi ini beserta teman-teman TI angkatan 2017
- e. Kontributor di Github yang telah menyediakan sejumlah sumber untuk penelitian ini secara *open source*.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan Bapak/Ibu semua dan semoga laporan skripsi ini dapat bermanfaat bagi orang banyak dan pengembangan ilmu.

Depok, 30 Juli 2021

Fathur Rahman Haikal



## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya yang bertanda tangan di bawah ini:

Nama : Fathur Rahman Haikal  
 NIM : 4617010028  
 Program Studi : Teknik Informatika  
 Jurusan : Teknik Informatika dan Komputer  
 Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Jakarta **Hak Bebas Royalti Non Eksklusif (Non-exclusive Royalty- Free Right)** atas skripsi saya yang berjudul:

**ANALISIS PERBANDINGAN AKURASI DAN PERFORMA ALGORITMA DETEKSI OBJEK PADA YOLO v3 DENGAN SINGLE SHOT MULTIBOX DETECTOR (SSD)**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non Eksklusif ini Politeknik Negeri Jakarta berhak menyimpan, mengalih media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Depok. Pada tanggal: 30 Juli 2021

Yang menyatakan  
 (Fathur Rahman Haikal)





## Analisis Perbandingan Akurasi dan Performa Algoritma Deteksi Objek Pada YOLO v3 dengan *Single Shot Multibox Detector* (SSD)

### Abstrak

Deteksi objek bertujuan untuk menemukan dan mengklasifikasikan objek yang ada dalam satu gambar, dan memberi label dengan kotak pembatas persegi panjang untuk menunjukkan konfidensi keberadaan objek. Sejumlah penelitian sebelumnya telah membandingkan berbagai algoritma deteksi objek. Penelitian sebelumnya masih menggunkan perangkat dengan sumberdaya tinggi, sehingga diperlukan untuk melakukan penelitian performa dari deteksi objek dengan sumber daya rendah. Penelitian ini akan menguji dan membandingkan akurasi dan performa dari algoritma berbasis Regression / Classification yaitu YOLO (You Only Look Once) dengan SSD (Single Shot Multibox Detector). Digunakan framework Flutter pada perangkat mobile untuk mengukur performa algoritma deteksi objek, dengan parameter FPS dan ms/Inference. Penelitian ini menggunakan parameter confusion matrix dan loss function untuk pengukuran akurasi, serta menggunakan Flutter Devtools untuk pengukuran performa. Penelitian ini menyimpulkan bahwa YOLO memiliki keunggulan dari SSD pada FPS, namun SSD memiliki keunggulan dalam menghasilkan jumlah inferensi

Kata kunci: COCO, Flutter, Mobile, Perbandingan, Performa, SSD, YOLO.

POLITEKNIK  
NEGERI  
JAKARTA

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



## DAFTAR ISI

<b>HALAMAN PERNYATAAN ORISINALITAS.....</b>	<b>1</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>Error! Bookmark not defined.</b>
<b>KATA PENGANTAR.....</b>	<b>3</b>
<b>HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS.....</b>	<b>4</b>
<b>Abstrak.....</b>	<b>5</b>
<b>BAB I PENDAHULUAN.....</b>	<b>11</b>
1.1 Latar Belakang.....	11
1.2 Rumusan Masalah.....	13
1.3 Batasan Masalah.....	13
1.4 Tujuan dan Manfaat.....	13
1.5 Metode Pelaksanaan.....	14
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>15</b>
2.1 Penelitian Sejenis.....	15
2.2 YOLO.....	16
2.3 Single Shot Multibox Detector (SSD).....	19
2.4 Deteksi Objek.....	21
2.5 Aplikasi Mobile.....	21
2.6 Epoch.....	22
2.7 Pengukuran Hasil Uji.....	22
2.7.1 Confusion Matrix.....	22
2.7.2 Loss Function.....	25
2.7 Tensorflow Lite.....	26
<b>BAB III PERENCANAAN DAN REALISASI.....</b>	<b>28</b>
3.1 Deskripsi Rencana Pengujian.....	28
3.3. Rencana dan Penerapan Pengujian pada Android.....	32
<b>BAB IV PEMBAHASAN.....</b>	<b>38</b>
4.1 Deskripsi Pengujian Saat Training.....	38
4.2 Prosedur Pengujian.....	38
4.3 Analisa Data/Evaluasi.....	46
<b>BAB V PENUTUP.....</b>	<b>48</b>

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta





## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

5.1	Kesimpulan.....	48
5.2	Saran .....	48
<b>DAFTAR PUSTAKA .....</b>		<b>49</b>





## DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu .....	15
Tabel 2. 2 Perbandingan COCO AP dan YOLO v3.....	18
Tabel 4. 1 Profiling YOLO dan SSD pada Samsung M31 .....	43
Tabel 4. 2 Profiling YOLO dan SSD pada Xiaomi POCO M3 Pro 5G .....	44



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

**DAFTAR GAMBAR**

Gambar 2. 1 Perbandingan performa YOLO .....	17
Gambar 2. 2 Tahapan Proses Deteksi .....	18
Gambar 2. 3 Multibox detector .....	21
Gambar 2. 4 Confusing Matrix .....	23
Gambar 2. 5 Rumus Precision.....	23
Gambar 2. 6 Rumus Recall .....	24
Gambar 2. 7 Rumus Mean Average Precision .....	24
Gambar 2. 8 Rumus Classification Loss .....	25
Gambar 2. 9 Rumus Objectness Loss .....	26
Gambar 2. 10 Rumus Localization Loss atau Box Loss .....	26
Gambar 3. 1 Kode perhitungan Objectness loss .....	29
Gambar 3. 2 Kode perhitungan classification loss .....	30
Gambar 3. 3 Kode perhitungan box loss .....	30
Gambar 3. 4 Kode perhitungan precision .....	31
Gambar 3. 5 Kode perhitungan Recall .....	31
Gambar 3. 6 Smartphone Xiaomi POCO M3 Pro 5G Note 4 (sebelah kiri) dan Samsung M31 (sebelah kanan) .....	32
Gambar 3. 7 Objek yang dideteksi.....	33
Gambar 3. 8 Kode Fungsi StartImageStream .....	34
Gambar 3. 9 Kode Fungsi detectObjectOnFrame .....	35
Gambar 3. 10 BoundingBox, klasifikasi dan confidence pada flutter .....	35
Gambar 3. 11 Kode untuk Halaman Utama .....	36
Gambar 3. 12 Halaman Utama.....	36
Gambar 3. 13 Hasil dari deteksi object pada Samsung M31 menggunakan SSD.....	37
Gambar 3. 14 Hasil deteksi object pada Xiaomi POCO M3 Pro 5G note 4 menggunakan YOLO .....	37
Gambar 4. 1 Grafik Model Metrik Recall .....	39
Gambar 4. 2 Grafik Model Metrik Precision dan mAP .....	39





## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 4. 3 Grafik Classification Loss pada Data Training dan Validasi .....	40
Gambar 4. 4 Grafik Objectness Loss pada Data Training dan Validasi .....	41
Gambar 4. 5 Grafik Localization Loss pada Data Training dan Validasi .....	42
Gambar 4. 6 Grafik Perbandingan Train dan Validation dari Loss Function Classification.....	46
Gambar 4. 7 Grafik Perbandingan Train dan Validation Objectness.....	46
Gambar 4. 8 Grafik Perbandingan Train dan Validation Localization .....	46





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Deteksi objek adalah suatu kegiatan yang bertujuan untuk memperoleh pemahaman tentang klasifikasi, estimasi konsep dan lokasi dari objek-objek dalam sebuah gambar. Sebagai salah satu dasar masalah visi komputer, deteksi objek mampu memberikan informasi berharga untuk pemahaman semantik gambar dan video, dan terkait dengan banyak aplikasi, termasuk klasifikasi gambar (Y. Jia, et al., 2014)

Pengembangan serta penerapan dari penelitian deteksi objek merupakan hal yang membutuhkan waktu dan biaya (Zhao et al., 2019), sehingga pemilihan model yang tepat serta perangkat untuk diterapkan merupakan keharusan. Diperoleh dari penelitian oleh Liu et al. (2020), kebutuhan akan model yang ringan serta dapat dijalankan pada perangkat rendah daya sudah menjadi kebutuhan kritikal, sehingga diperlukan penelitian yang membandingkan berbagai algoritma untuk memberikan acuan untuk pemilihan algoritma terbaik untuk perangkat rendah daya.

Performa tinggi algoritma deteksi objek yang diterapkan pada perangkat rendah daya seperti *mobile* dan sistem tanam merupakan hal terpenting pada beberapa kasus, salah satunya adalah *Virtual Reality* pada perangkat *mobile* (Liu, et al. 2020). *Virtual Reality* memerlukan algoritma deteksi objek dengan performa tinggi karena diterapkan pada perangkat *mobile*, serta memerlukan algoritma dengan serta latensi rendah guna menjalankan seluruh proses deteksi hingga *render* gambar secara *real-time*.

Selain *virtual reality*, sejumlah penerapan deteksi objek pada perangkat *mobile* membutuhkan performa tinggi untuk menjalankan seluruh prosesnya secara *real-time*, seperti pemilahan langsung buah stroberi dengan deteksi jamur (*powdery mildew*) dengan perangkat *mobile* (Mahmud et al, 2020) dan deteksi buah *real-time* pada pohonnya menggunakan perangkat *mobile* (Bresilla, et al. 2019).





## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Mengenai algoritma deteksi objek, kerangka kerja metode deteksi objek dapat dikategorikan menjadi dua jenis (Zhao et al., 2019), metode pertama adalah *Region Proposal Based Framework*. Metode ini adalah *pipeline* deteksi objek tradisional, menghasilkan proposal *Region* terlebih dahulu dan kemudian mengklasifikasikan setiap proposal menjadi kategori objek yang berbeda. Metode ini memiliki beberapa algoritma seperti Faster R-CNN (Ren, et al., 2018) dan R-FCN (He, et al., 2016).

Metode yang menjadi pengembangan dari metode *Region Proposal Based Framework* adalah *Regression/Classification based*. Metode ini mengenali deteksi objek sebagai masalah regresi atau klasifikasi, mengadopsi kerangka kerja untuk mencapai hasil akhir yaitu kategori dan lokasi (Zhao et al., 2019). Contoh dari penerapan metode *Regression/Classification Based* adalah YOLO (Redmond et al., 2016) dan SSD (Liu et al., 2016).

Secara performa, metode *Regression/Classification Based* sudah jauh lebih baik daripada *Region Proposal Based*, hal ini disebabkan oleh latensi yang dihasilkan lebih rendah (Zhao et al., 2019) sehingga memungkinkan untuk dijalankan secara *real-time*. Algoritma yang dipilih untuk dibandingkan pada penelitian ini adalah algoritma berdasarkan metode *Regression/Classification Based*.

Sejumlah penelitian telah dilakukan dengan tujuan untuk membandingkan berbagai algoritma deteksi objek. Penelitian yang dilakukan oleh Sanchez (2020) dan Srivastava (2021), dilaksanakan dengan tujuan mengukur akurasi dari algoritma yang berjenis *Region Proposal Based Method* dengan *Regression/Classification based method*, selain itu juga digunakan spesifikasi komputer yang tinggi, bukan dilakukan pada perangkat rendah daya seperti *mobile*.

Pemilihan algoritma serta penerapannya rendah daya merupakan keputusan terpenting yang harus ditentukan di awal suatu proyek pengembangan serta penerapan objek, sehingga penelitian ini ditujukan untuk melakukan melakukan pengujian dan perbandingan dari dua algoritma deteksi objek *regression/classification based* yaitu *You Only Look Once (YOLO) v3* dan *Single*





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

*Shot Multibox Detection* (SSD) untuk diterapkan pada perangkat sumber daya rendah seperti *smartphone*. Penelitian ini diharapkan dapat menjadi acuan referensi individu dalam membuat pondasi algoritma berbasis deteksi objek.

## 1.2 Rumusan Masalah

Rumusan masalah yang menjadi dasar pemikiran dalam penelitian ini adalah:

1. Bagaimana akurasi dari model pendeteksian objek pada YOLO dan SSD?
2. Bagaimana performa model pendeteksian objek pada YOLO dan SSD ketika diterapkan pada *smartphone*?
3. Bagaimana implementasi model YOLO dan SSD pada perangkat *mobile* dalam menguji performa dari model?

## 1.3 Batasan Masalah

Penelitian ini memiliki batasan masalah diantaranya sebagai berikut:

1. Model akan diuji di *Google Collab* untuk diperoleh nilai akurasi secara setara menggunakan dataset yang sama (COCO 2017)
2. Model akan diterapkan untuk aplikasi pada perangkat *smartphone* (*Android*) untuk diukur performanya.
3. Model dibangun menggunakan *tensorflow*, diterapkan pada aplikasi *android* menggunakan *framework flutter*.

## 1.4 Tujuan dan Manfaat

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Mengetahui akurasi dari algoritma pendeteksian objek pada YOLO dan SSD
2. Mengetahui performa algoritma pendeteksian objek pada YOLO dan SSD ketika diterapkan pada *smartphone*
3. Mengetahui implementasi model YOLO dan SSD pada perangkat *mobile* dalam menguji performa dari model

Sedangkan, manfaat yang didapatkan pada penelitian ini:



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

1. Membantu dan mempermudah pengambilan keputusan dalam menentukan algoritma yang lebih sesuai untuk dipilih.
2. Mengetahui hasil dari akurasi dan performa algoritma saat diterapkan di lab maupun *mobile*.
3. Dapat dijadikan acuan untuk pengukuran berbagai algoritma lainnya.

### 1.5 Metode Pelaksanaan

Adapun pengumpulan data pada penelitian ini antara lain, adalah:

#### 1. Studi literatur

Studi literatur dilakukan untuk mencari sumber data melalui penelitian terdahulu yang serupa, seperti jurnal, buku maupun artikel. Studi literatur dilakukan guna mengumpulkan data untuk pelaksanaan penelitian mengenai deteksi objek.

#### 2. Pengujian

Pengujian dilakukan dengan mengukur akurasi dan performa. Akurasi diperoleh dengan melakukan training dan testing, lalu melihat nilai akurasi kedua model berdasarkan data COCO yang telah disediakan. Performa diperoleh dengan cara menerapkan model dari algoritma ke aplikasi, lalu mengukur kecepatan model dalam melakukan inferensi.

#### 3. Analisis Hasil Pengujian

Melakukan analisis perbandingan kinerja yang dapat dicapai oleh algoritma YOLO v3 dan SSD dalam melihat akurasi dan performa yang ditampilkan.

#### 4. Dokumentasi Kerja

Melakukan pengambilan screenshot, log pengujian dan data-data penunjang lainnya yang dapat menggambarkan proses kerja.





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Sejenis

Terdapat beberapa penelitian yang telah dilaksanakan sebelumnya yang berfokus pada perbandingan dari berbagai jenis algoritma deteksi objek. Berikut adalah ringkasannya.

*Tabel 2. 1 Penelitian Terdahulu*

No	Peneliti	Judul Penelitian	Pembeda
1	Zhao, Z.Q., et al	Object detection with deep learning: A review.	Berfokus tentang tinjauan secara umum dari seluruh algoritma dari berbagai generasi. Tidak memberikan perbandingan secara langsung mengenai performa dari masing-masing algoritma ketika sudah diterapkan menjadi model.
2	Sanchez, S.A., et al	Comparative analysis of deep learning image detection algorithms	Penelitian berfokus untuk membandingkan akurasi antara masing-masing algoritma. Belum menyertakan konfigurasi pengujian dan belum mengujikan performa.
3	Srivastava, S., et al	A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework.	Penelitian berfokus membandingkan performa masing-masing model, namun pengukuran yang diambil masih belum lengkap, serta model tidak diterapkan ke perangkat selain PC.
4	Wilcox, M., et al.	Comparing performance parameters of mobile app development strategies.	Penelitian berfokus pada perbandingan sejumlah parameter performa. Parameter pengukuran performa Response Time (ms/Inference) dan FPS diambil dari penelitian ini.

Diperoleh dari penelitian pertama yang berjudul “Object detection with deep learning: A review.” bahwa YOLO dan SSD sudah merupakan algoritma deteksi





## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

objek berjenis *Regression/Classification based*, yang merupakan generasi lanjutan dari *Region/Proposal based*, dengan performa yang jauh lebih cepat.

Penelitian kedua yang berjudul “Comparative analysis of deep learning image detection algorithms” berfokus untuk membandingkan akurasi dari masing-masing algoritma menggunakan berbagai dataset, namun masih membandingkan F-RCNN yang merupakan algoritma berjenis *Region/Proposal based* secara langsung dengan YOLO dan SSD yang merupakan *Regression/Classification based*, yang mana tidak ideal karena kedua jenis tersebut sudah beda generasi sebagaimana disebutkan di penelitian pertama.

Penelitian ketiga yang berjudul “A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework.” sudah menggunakan TensorFlow yang juga akan diterapkan pada penelitian ini, namun penelitian ketiga belum menyertakan konfigurasi perangkat yang dipakai untuk pengujian. Penelitian ketiga juga masih melakukan perbandingan antara *Regression/Classification based* dengan *Region/Proposal based* secara langsung yang sudah diketahui bahwa kedua jenis algoritma deteksi objek tersebut beda generasi.

Penelitian ini akan berfokus pada pengujian dua algoritma *Regression Classification based* yang merupakan generasi yang sama, yaitu YOLO dan SSD, mengukur akurasinya serta menerapkan modelnya ke perangkat mobile untuk diukur performanya secara nyata. Parameter performa diperoleh dari penelitian keempat yang berjudul “Comparing performance parameters of mobile app development strategies.”.

## 2.2 YOLO

*You Only Look Once* (YOLO) adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara *real-time*. Sistem pendeteksian yang dilakukan adalah dengan menggunakan *repurpose classifier* atau *localizer* untuk melakukan deteksi. Sebuah model diterapkan pada sebuah citra di beberapa lokasi dan skala.

Daerah dengan citra yang diberi score paling tinggi akan dianggap sebagai sebuah pendeteksian. YOLO menggunakan pendekatan jaringan syaraf tiruan (JST) untuk



## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

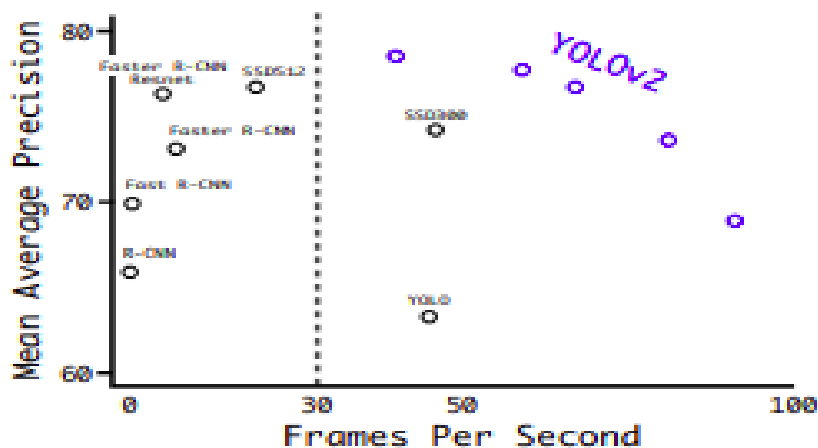
### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

mendeteksi objek pada sebuah citra. Jaringan ini membagi citra menjadi beberapa wilayah dan memprediksi setiap kotak pembatas dan probabilitas untuk setiap wilayah. Kotak-kotak pembatas ini kemudian dibandingkan dengan setiap probabilitas yang diprediksi.

YOLO memiliki beberapa kelebihan dibandingkan dengan sistem yang berorientasi pada *classifier*. Hal ini terlihat dari seluruh citra pada saat dilakukan sebuah uji dengan prediksi yang diinformasikan secara global pada citra. Hal tersebut juga membuat prediksi dengan sintesis jaringan saraf ini tidak seperti sistem *Region Convolutional Neural Network* (R-CNN) yang membutuhkan ribuan untuk sebuah citra sehingga membuat YOLO lebih cepat hingga beberapa kali daripada RCNN (Jupiyandi et al., 2019).

YOLO sudah mengalami beberapa iterasi dari pengembangannya, dimulai dari v1 hingga versi terbaru yang sedang dalam penelitian sekarang adalah v5, namun yang resmi dipublikasikan baru sampai versi keempat (Wang, Bochkovskiy and Liao, 2020). Penelitian ini dilakukan menggunakan versi ketiga pada YOLO. Adapun alasan dari pemilihan versi tersebut adalah tersedianya beberapa sumber yang dapat mempermudah penerapan dari model ke dalam perangkat bergerak. Secara performa, YOLO v1 sudah dapat dibandingkan jauh lebih baik daripada R-CNN namun secara presisi belum lebih baik dari SSD (*Single-Shot Detection*).



Gambar 2. 1 Perbandingan performa YOLO





### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

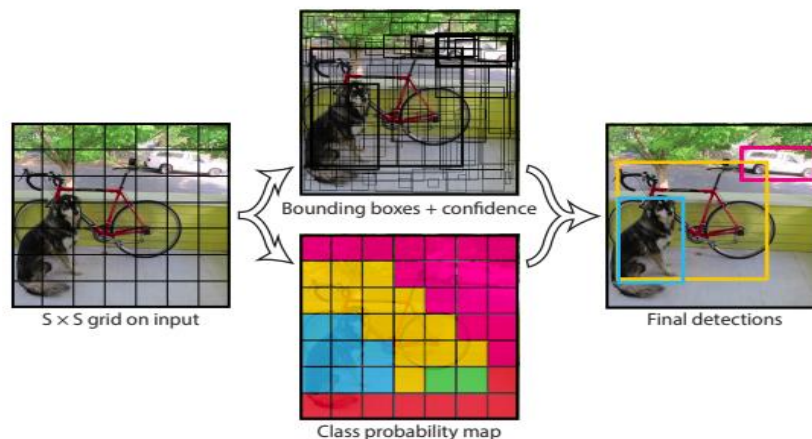
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar X.X menunjukkan perbandingan performa YOLOv2 dengan YOLO (v1) serta SSD. Bila ditinjau baik dari MAP maupun FPS, YOLOv2 sudah jauh lebih unggul daripada YOLOv1 dan SSD (Redmon *et al.*, 2016).

Tabel 2. 2 Perbandingan COCO AP dan YOLO v3

	backbone	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
<i>Two-stage methods</i>							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	<b>52.1</b>
<i>One-stage methods</i>							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	<b>40.8</b>	<b>61.1</b>	<b>44.1</b>	<b>24.1</b>	<b>44.2</b>	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

Tabel diatas menunjukkan perbandingan rerata presisi menggunakan standar COCO AP antara YOLO v3 dengan versi sebelumnya serta algoritma lainnya (Redmon and Farhadi, 2018).



Gambar 2. 2 Tahapan Proses Deteksi

Terdapat 2 tahapan besar dalam proses mendeteksi objek YOLO. Tahapan tersebut adalah (Redmon *et al.*, 2016):

1. Resolusi gambar dibuat menjadi 448 x 448.





#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2. Gambar dibagi menjadi kotak-kotak berukuran  $S \times S$  yang disebut grid cell.
3. Setiap kotak dijadikan sebagai titik tengah dan diberikan dalam tiga bentuk Bounding Boxes. Bounding box adalah proses terakhir dari alur pendeteksian objek yang bergerak pada *image sequence*. Bounding Boxes ini digabungkan dengan nilai *confidence* akan dihitung pada masing-masing bounding box yang kemudian akan diseleksi berdasarkan nilai yang didapat (Redmon *et al.*, 2016).

Penggabungan ini mempunyai kerja masing-masing yaitu memprediksikan apakah ada objek atau tidak. Jika seandainya diprediksi adanya objek, maka *Bounding Boxes* akan diberikan nilai 1 dan jika tidak diberikan nilai 0. *Bounding Boxes* yang bernilai 1 akan menghasilkan bounding box. Setiap *Bounding Box* memiliki lima prediksi:  $x$ ;  $y$ ;  $w$ ;  $h$ ; dan *confidence*. Prediksi  $(x,y)$  sebagai titik koordinat tengah grid cell sedangkan prediksi  $(w,h)$  width dan height sebagai batas ukuran keseluruhan gambar (objek). Prediksi *confidence* adalah bentuk nilai keyakinan dari prediksi yang memiliki nilai masing-masing: 1)  $S = 7$ ; 2)  $B = 3$ ; dan  $C = 20$ .

Terdapat dua proses dalam menemukan bounding box, yaitu:

- a. *BBoxes* dan *Confidence*: Mengimplementasikan algoritma YOLO untuk Untuk menentukan nilai *confidence* dapat dilihat dari persamaan 2.7 berikut ini.  

$$Confidence = Pr(Object) * IOU (Gtruth/Gpred) \quad (2.7)$$
- b. *Class probability*: Mengimplementasi dalam mengklasifikasi kelas pada objek.

Menjalankan operasi *convolutional neural network*. Setelah didapatkan *bounding boxes* dan *confidence* maka dijalankan proses CNN pada setiap *bounding boxes*. Pada *convolution layer* pertama kali digunakan matriks  $7 \times 7$  stride 2, selanjutnya matriks  $3 \times 3$  stride 2, pada proses *pooling layer* digunakan matriks  $2 \times 2$  stride, terakhir diikuti oleh *fully connected layer*.

### 2.3 Single Shot Multibox Detector (SSD)

Menurut Liu *et.al* (2016), *Single Shot Detector* (SSD) adalah metode yang digunakan untuk mendeteksi sebuah objek atau gambar dengan menggunakan



## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

*single deep neural network* dan salah satu algoritma deteksi objek karena kemudahan implementasi dan akurasi yang sepadan terhadap komputasi yang dibutuhkan. Metode ini termasuk kedalam deteksi object secara *real time*. Oleh karena itu menurut Ning et.al (2017), metode ini menggunakan konstruksi rangka yang termasuk kedalam jenis *Convolutional Neural Network* (CNN). Konstruksi pembangunan ini dibagi menjadi 2 bagian besar, *Feature Extraction Layer* dan *Convolutional Layer*. *Feature Extraction Layer* adalah bagian dalam melakukan *encoding* dari sebuah objek atau gambar menjadi *features* yang merepresentasikan gambar tersebut sedangkan, *convolutional layer* merupakan bagian yang terdiri dari neuron yang disusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (pixels).

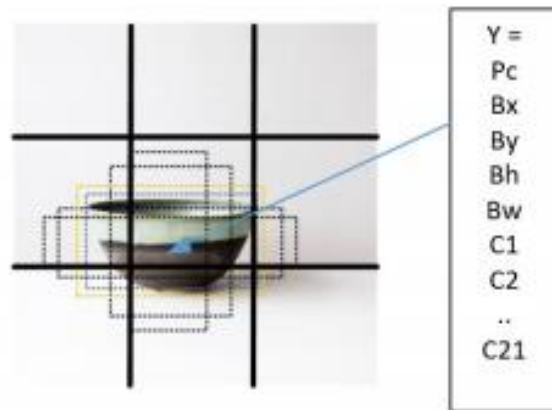
*Single Shot Detector* (SSD) yang menggunakan metode *multibox detector* memiliki tujuan untuk meregresi bounding box untuk menentukan prediksi lokasi objek sekaligus mengklasifikasikannya pada suatu masukan citra. Menurut Liu (2016), cara kerja *multibox detector* adalah dengan menyesuaikan classifier pada setiap lapisan konvolusi yang berukuran  $m * n * b * (4 + c)$  dengan  $m * n$  adalah ukuran *feature map* (contohnya 3x3 yang akan menghasilkan 9 sel atau grid),  $b$  adalah jumlah *predict bounding box* per grid, dan  $c$  adalah jumlah kelas ditambah 4 offset untuk ukuran setiap *predict bounding box*. Ilustrasi dapat dilihat pada Gambar 2. Selanjutnya SSD akan memeriksa apakah ada objek yang dikenali pada setiap grid pada *feature map*, ketika mendapatkan objek maka SSD akan membentuk *predict bounding box* baru dengan ukuran rasio yang berbeda-beda dengan jumlah maksimal 6 bounding box pada setiap grid. Pada setiap *predict bounding box* berisikan  $P_c$  untuk probabilitas softmax kelas target,  $B_x$  dan  $B_y$  adalah titik tengah pada grid tersebut sehingga tidak akan pernah melebihi nilai lebih dari 1,  $B_h$  &  $B_w$  adalah ukuran lebar dan tinggi dari bounding box tersebut, dan  $C_1$  hingga  $C_{21}$  nilai skor prediksi tiap kelas (Liu et. al, 2016).





#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 2. 3 Multibox detector

## 2.4 Deteksi Objek

Deteksi objek adalah teknologi komputer yang berkaitan dengan penglihatan komputer dan pemrosesan gambar yang berhubungan dengan mendeteksi contoh objek semantik dari kelas tertentu (seperti manusia, gedung, atau mobil) dalam gambar dan video digital. *Object Detection* atau deteksi objek merupakan bagian dari *Computer Vision*. *Object Detection* mengacu pada kemampuan komputer untuk mendeteksi sejumlah objek pada suatu gambar (Ikeuchi, 2014).

Menurut Jeong & Moon (2011) deteksi objek dilakukan dengan cara mengambil *image feature* seperti garis, sudut, kontur dan warna dari sebuah gambar. Deteksi objek merupakan bagian dari *Object Recognition* atau identifikasi objek. Oleh karena itu dapat disimpulkan bahwa untuk deteksi objek pasti harus diidentifikasi terlebih dahulu objek tersebut. Namun, pada penelitian ini hanya dilakukan deteksi objek saja tanpa adanya identifikasi objek. Oleh karena itu, teknologi deteksi objek dipilih untuk mempermudah proses pemotongan gambar.

## 2.5 Aplikasi Mobile

Aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain tanpa adanya pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti telepon seluler dan *smartphone*. Pengertian lainnya menyebutkan bahwa aplikasi mobile merupakan aplikasi yang telah dirancang khusus untuk





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

platform mobile dalam hal ini seperti misalnya iOS, android, atau windows mobile (Bruce and Pressman, 2014).

### 2.5.1 Flutter

*Flutter* merupakan *framework* yang digunakan untuk pembuatan aplikasi mobile dan php dalam *web service* dan MySQL untuk mengelola local database (Krisnada & Tanone, 2019). Menurut Dian & Muhar (2018), *Flutter* adalah sebuah *framework open source* yang dikembangkan oleh Google untuk membuat atau mengembangkan aplikasi yang dapat berjalan dalam sistem operasi Android dan iOS. *Flutter* menggunakan bahasa pemrograman Dart dalam pengkodean. *framework Flutter* memiliki perbedaan pada “*build*” (pembangunan) aplikasi. Pada *framework* ini semua kodenya di compile dalam kode native-nya (*Android NDK, LLVM, AOT compiled*) tanpa ada interpreter pada prosesnya sehingga proses compile-nya menjadi lebih cepat.

### 2.6 Epoch

Epoch adalah istilah yang digunakan dalam pembelajaran mesin dan menunjukkan sebuah siklus latih yang telah diselesaikan oleh algoritma pembelajaran mesin. *Epoch* menggambarkan satu siklus algoritma *machine learning* dari keseluruhan set data training. Artinya, Satu epoch menandakan sebuah algoritma *machine learning* telah ‘belajar’ dari data training secara keseluruhan (Wibawa, 2000).

### 2.7 Pengukuran Hasil Uji

#### 2.7.1 Confusion Matrix



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

		Real Label	
		Positive	Negative
Predicted Label	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Gambar 2. 4 Confusing Matrix

*Confusion matrix* adalah sebuah metode yang biasanya digunakan dalam melakukan kalkulasi pada akurasi konsep data mining (Rahman *et al.*, 2017), *recall*, *precision*, dan *error rate* (Arini, Wardhani and Octaviano, 2020). Menurut Rahman *et al* (2017), di dalam *confusing matrix* akan menampilkan dan menjelaskan sejumlah data yang diuji dan diklasifikasikan menjadi data yang benar dan salah. Oleh karena itu, confusion matrix juga dikatakan sebagai matrik dari prediksi yang akan dibandingkan dengan kelas yang asli dari data telah dimasukkan (Herdiawan, 2018).

#### 2.7.1.1

#### Precision

$$\text{Precision} = \frac{\sum TP}{\sum TP + FP}$$

Gambar 2. 5 Rumus Precision

*Precision* adalah ukuran jumlah dokumen teks yang relevan pada sebuah data yang dapat terkendali diantara semua dokumen teks yang terpilih oleh sistem (Subari and Ferdinandus, 2015). Menurut Manning, Raghavan and Schütze (2009) *precision* berfungsi mengevaluasi kemampuan sistem untuk menemukan peringkat yang baik guna mendefinisikan sebagai persentase dokumen yang di





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

*retrieve* dan benar-benar relevan terhadap *query* sehingga jika disimpulkan melihat Gambar... maka nilai *true* positif dibagi dengan semua yang terprediksi positif.

### 2.7.1.2 Recall

$$\text{Recall} = \frac{\sum TP}{\sum TP + FN}$$

Gambar 2. 6 Rumus Recall

Seperti yang sudah dijelaskan pada penelitian sebelumnya, menurut Subari dan Ferdinandus (2015), *Recall* adalah proporsi dari kuantitas dokumen pada teks atau data yang relevan sehingga menimbulkan kendali diantara semua dokumen teks atau data yang relevan pada koleksi. Pada aktivitas ini, *recall* akan mengevaluasi kemampuan sistem dalam pencarian semua item yang relevan dari koleksi dokumen dan akan melakukan pendefinisian sebagai persentase dokumen yang relevan terhadap *query* (Manning, Raghavan and Schütze, 2009) sehingga nilai yang dihasilkan sama seperti *precision* yaitu *true* positif dibagi dengan semua yang aslinya positif

### 2.7.1.3 Mean Average Precision

$$MAP = \frac{1}{(Q)} \sum_{j=1}^{|Q|} \frac{1}{M_j} \sum_{K=1}^{M_j} Precision(R_{jk})$$

Gambar 2. 7 Rumus Mean Average Precision

*Mean average precision* adalah perhitungan yang memberikan sebuah nilai sendiri terhadap seluruh titik *recall* dari seluruh pengukuran (Rochmawati and Kusumaningrum, 2016). Pengukuran ini sudah memiliki bukti dapat





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

memperlihatkan tingkat perbedaan dan stabilitas yang baik, sehingga dapat menghitung beberapa hasil yang benar dan salah dari *precision* dan akan dilakukan perhitungan persentase keaslian (Manning, Raghavan and Schütze, 2009).

## 2.7.2 Loss Function

*Loss function* merupakan algoritma pengukuran pada *machine learning*. Fungsi dari *loss function* ini adalah bagaimana mengukur jarak antara output model dan nilai target (kebenaran). Kerja pada *loss function* akan memberi tahu *tensorflow* seberapa baik atau buruk prediksi dibandingkan dengan hasil yang diinginkan sehingga *loss function* juga berfungsi untuk membandingkan target dengan prediksi dan memberikan jarak numerik antara keduanya. *Loss function* diukur setiap *epochs*. Semakin sedikit *loss function* setelah setiap epoch-nya semakin baik model dalam memprediksi labelnya.

### 2.7.2.1 Classification Loss

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Gambar 2. 8 Rumus Classification Loss

Aktivitas dalam mengukur loss dilihat dari classification (salah klasifikasi) per epochs. Kegiatan ini bagaimana menghitung selisih dari perhitungan semua klasifikasi yang salah lalu dikuadratkan. (Redmon and Farhadi, 2018).

### 2.7.2.2 Objectness Loss



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

$$\sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2$$

Gambar 2. 9 Rumus Objectness Loss

Aktivitas dalam mengukur loss dilihat dari objectness/confidence per epochs. Kegiatan ini menghitung selisih dari confidence rate dengan 100% lalu dikuadratkan. (Redmon and Farhadi, 2018).

#### 2.5.2.3 Localization Loss atau Box Loss

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

where

$\mathbb{1}_{ij}^{obj} = 1$  if the  $j$  th boundary box in cell  $i$  is responsible for detecting the object, otherwise 0.

$\lambda_{coord}$  increase the weight for the loss in the boundary box coordinates.

Gambar 2. 10 Rumus Localization Loss atau Box Loss

Kegiatan dalam mengukur loss juga dilihat dari localization (koordinat dan tinggi lebar dari *bounding box*) per epochs. Aktivitas yang dilakukan bagaimana menghitung selisih tinggi dan lebar dari box prediksi dengan *box asli* (ground truth) yang dijumlahkan dengan selisih dari koordinat prediksi dengan koordinat pusat asli. (Redmon and Farhadi, 2018).

## 2.7 Tensorflow Lite

*Tensorflow Lite* adalah versi lain dari *Tensorflow* biasa yang merupakan *library machine learning*, dirancang khusus untuk perangkat mobile (Andrea, 2018). *Tensorflow Lite* merupakan *framework deep learning open source* pada inferensi *on-device* sehingga jenis *tensorflow* ini menyediakan kerangka (*framework*) yang telah mendapatkan perlakuan untuk menyesuaikan dengan *tensorflow* biasa untuk diterapkan pada aplikasi mobile dan sejenisnya. Terdapat *Tensorflow Lite*



## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

*Interpreter* yang merupakan model dalam menginterpretasikan dan membaca data dengan memasukan data tersebut pada aplikasi dalam *model machine learning* (Pratama and Pratama, 2021)

Meskipun versi dari *Tensorflow* ini jauh lebih ringan dibandingkan *Tensorflow* yang biasa, *Tensorflow Lite* memiliki beberapa kekurangan: 1) versi perangkat *Android* haruslah versi 6.0 (SDK 23.0); 2) masih belum mendukung data yang berjenis “string” ; dan 3) data tensor haruslah salah satu jenis data yang bersifat primitif yaitu “float”, “int”, “long”, dan “byte” sehingga jika ada jenis data “string” haruslah dikonversi menjadi kata kata yang bersifat primitive. Salah satu bentuk konversi dengan menggunakan tokenize pada suatu perangkat lunak (Bosley, 2013).







**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## BAB III

### PERENCANAAN DAN REALISASI

#### 3.1 Deskripsi Rencana Pengujian

Pengujian berfokus pada performa dan akurasi kedua model. Dataset, konfigurasi hardware pengujian serta objek uji yang digunakan adalah sama. Perbedaan pada pengujian terletak pada spek hardware sebagai variabel kontrol, sehingga dapat diperoleh data yang secara adil membandingkan kedua algoritma. Pengujian dibagi ke dalam dua tahap:

1. Pengujian akurasi dilakukan menggunakan dataset COCO 2017 yang akan dijadikan data training dan validation untuk kedua algoritma, dilakukan di Google Colab. Pengukuran diambil setiap satu epoch saat training. Parameter yang digunakan untuk pengukuran akurasi adalah Confusion Matrix serta Loss Function. Model hasil training diambil dan dikonversi menjadi format TFLite untuk diterapkan ke dalam aplikasi flutter.
2. Pengujian performa dilakukan dengan cara mengukur pada kecepatan model menghasilkan inferensi setelah model diterapkan ke aplikasi dan aplikasi di-*instal* ke perangkat *smartphone*. Tipe *smartphone* uji yang digunakan adalah Samsung M31 dengan SoC Exynos 9611 dan RAM 6GB, Xiaomi POCO M3 Pro 5G Note 4 dengan SoC Mediatek Dimensity 700 dan RAM 6GB. Kedua *smartphone* memiliki RAM 6GB serta kamera 48MP sehingga kedua *smartphone* uji dapat dinilai setara.

#### 3.2 Rencana Pengujian Akurasi

Model akan diuji di *Google Colab* untuk diperoleh nilai akurasinya secara setara menggunakan dataset yang sama (COCO 2017).

##### 3.2.1 Persiapan Data



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

COCO Dataset dipilih sebagai dataset untuk diimplementasikan. COCO Dataset atau COCO API adalah dataset dengan gambaran yang besar. Dataset yang digunakan memiliki 81 label, lengkap dengan gambar yang telah diberi *bounding box*. Dataset ini dirancang untuk mendeteksi objek, segmentasi, *deteksi body keypoint*, segmentasi barang, dan pembuatan teks. Dataset ini biasa digunakan pada Matlab, Python, dan Lua API. Hal ini dikarenakan dapat membantu memuat, memarsing, dan memvisualisasikan anotasi dalam COCO (Saleh, Ammi and Szoniecky, 2018).

### 3.2.1 Perhitungan Akurasi

Kedua algoritma YOLO dan SSD akan di-training di Google Colab untuk diukur akurasi menggunakan pengukuran Confusion Matrix dan Loss Function. Model dilatih dengan dataset, batch size, jumlah epoch serta image size yang sama guna memberikan keseimbangan kepada kedua model guna memperoleh hasil yang valid. Kode untuk pengukuran akurasi diambil dari [github.com/ultralytics/yolov3](https://github.com/ultralytics/yolov3). Berikut ini adalah sejumlah fungsi yang dipakai untuk menghitung akurasi dari model saat training.

#### 3.2.1.1 Objectness Loss

Berikut ini adalah kode dan rumus dari perhitungan *Objectness Loss*.

$$\sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2$$

```
# Objectness
tobj[b, a, gj, gi] = (1.0 - self.gr) + self.gr * iou.detach().clamp(0).type(tobj.dtype) # iou ratio
```

Gambar 3. 1 Kode perhitungan *Objectness loss*

Perhitungan *objectness* dilakukan dengan cara mengurangi 1 dengan hasil prediksi lalu dikuadratkan.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

### 3.2.1.2 Classification Loss

Berikut ini adalah kode dan rumus dari perhitungan *Classification Loss*.

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

```
# Classification
if self.nc > 1: # cls loss (only if multiple classes)
    t = torch.full_like(ps[:, 5:], self.cn, device=device) # targets
    t[range(n), tcls[i]] = self.cp
    lcls += self.BCEcls(ps[:, 5:], t) # BCE
```

Gambar 3. 2 Kode perhitungan classification loss

Classification loss hanya dilakukan bila kelas lebih dari satu. Classification Loss dilakukan dengan mengurangi seluruh jumlah dari sebuah kelas *true* dengan jumlah prediksi dari kelas tersebut.

### 3.2.1.3 Box Loss

Berikut ini adalah kode dan rumus dari perhitungan *Box Loss*.

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right]$$

where

$\mathbb{1}_{ij}^{\text{obj}} = 1$  if the  $j$ -th boundary box in cell  $i$  is responsible for detecting the object, otherwise 0.

$\lambda_{\text{coord}}$  increase the weight for the loss in the boundary box coordinates.

```
# Regression
pxy = ps[:, :2].sigmoid() * 2. - 0.5
pwh = (ps[:, 2:4].sigmoid() * 2) ** 2 * anchors[i]
pbox = torch.cat((pxy, pwh), 1) # predicted box
iou = bbox_iou(pbox.T, tbox[i], x1y1x2y2=False, CIoU=True) # iou(prediction, target)
lbox += (1.0 - iou).mean() # iou loss
```

Gambar 3. 3 Kode perhitungan box loss

Box Loss dilakukan dengan cara menghitung selisih dari lokasi prediksi (pxy) dan panjang lebar prediksi (pwh) dengan selisih lokasi dan Panjang lebar asli.





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

### 3.2.1.4 Precision

Berikut ini adalah kode dan rumus dari perhitungan *precision*.

$$\text{Precision} = \frac{\sum TP}{\sum TP + FP}$$

```
# Precision
precision = tpc / (tpc + fpc) # precision curve
p[ci] = np.interp(-px, -conf[i], precision[:, 0], left=1)
```

Gambar 3. 4 Kode perhitungan *precision*

Perhitungan *Recall* dilakukan dengan cara membagi *True Positive* dengan jumlah dari terlabel positif (*true positif* dan *false positif*)

### 3.2.1.5 Recall

Berikut ini adalah kode dan rumus dari perhitungan *recall*.

$$\text{Recall} = \frac{\sum TP}{\sum TP + FN}$$

```
# Recall
recall = tpc / (n_1 + 1e-16) # recall curve
r[ci] = np.interp(-px, -conf[i], recall[:, 0], left=0) # negative x, xp because xp decreases
```

Gambar 3. 5 Kode perhitungan *Recall*

Perhitungan *Recall* dilakukan dengan cara membagi *True Positive* dengan jumlah dari asli positif (*true positif* dan *false negative*)



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

### 3.3. Rencana dan Penerapan Pengujian pada Android

Pengujian performa dilakukan dengan cara mengukur pada kecepatan model menghasilkan inferensi setelah model diterapkan ke aplikasi dan aplikasi di-*instal* ke perangkat *smartphone*. Tipe *smartphone* uji yang digunakan adalah Samsung M31 dengan SoC Exynos 9611 dan RAM 6GB, Xiaomi POCO M3 Pro 5G Note 4 dengan SoC Dimensity 700 dan RAM 6GB. Meskipun kedua *smartphone* uji memiliki kemampuan kamera berbeda, aplikasi membatasi kualitas gambar hingga 13MP, sehingga performa yang diperoleh semakin seimbang, cukup dengan *hardware* sebagai variabel kontrol.

#### 3.3.1 Spesifikasi Android

Smartphone uji yang dipakai adalah Samsung M31 serta Xiaomi POCO M3 Pro 5G Note 4 Pro. Samsung M31 memiliki RAM 6 dengan memiliki SoC Exynos 9611 dengan kamera 48MP. Xiaomi POCO M3 Pro 5G Note 4 Pro memiliki RAM 6 GB serta SoC yang jauh lebih tua yaitu Dimensity 700 dan kamera 48MP.



Gambar 3. 6 Smartphone Xiaomi POCO M3 Pro 5G Note 4 (sebelah kiri) dan Samsung M31 (sebelah kanan)

#### 3.3.2 Prosedur Pengujian

Pengujian dilakukan dengan cara menjalankan aplikasi Flutter pada *smartphone*, memilih model yang tersedia di halaman utama, lalu objek dideteksi model yang dipilih.

#### 3.3.3. Prosedur penelitian performa

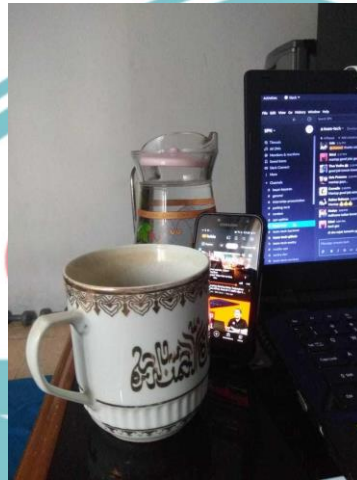
Didasari dengan metode pengukuran yang diperoleh dari penelitian oleh Willocx



### Hak Cipta :

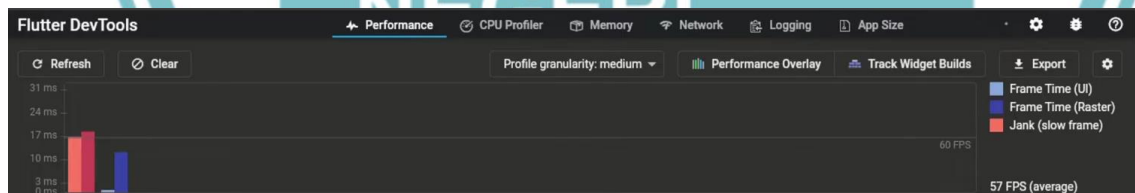
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

(2016), parameter pengujian performa yang dipilih adalah FPS dan ms/inference. Pengujian dilakukan dengan cara menjalankan aplikasi Flutter pada *smartphone*, memilih model yang tersedia di halaman utama, lalu objek dideteksi model yang dipilih, masing-masing model akan dijalankan lima kali di masing-masing ponsel uji.



Gambar 3. 7 Objek-objek yang dideteksi

Model akan digunakan untuk menguji sejumlah objek yang ada pada gambar tersebut. Ketepatan label tidak dapat diukur saat pengujian performa, namun jumlah dan kecepatan inferensi serta FPS akan diukur.



Gambar 3. 8 Pengukuran performa dengan cara Profiling pada Flutter menggunakan Flutter Devtools

Berikut adalah cara pengukuran FPS dan ms/Inference pada *Flutter*. Digunakan *Flutter Devtools* untuk memperoleh FPS. FPS terdapat pada bagian pojok kanan bawah pada gambar, serta ms/inference dihitung dengan mengukur rata-rata tinggi dari bar chart biru muda (Frame Time UI) dibagi dengan jumlah bar chart (Frame Time UI).





#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

### 3.3.4. Penerapan Model pada Flutter

Setelah model selesai di-*training*, model diekspor ke dalam format TFLite, lalu diterapkan ke aplikasi Flutter. Bagian dari program flutter yang akan menangkap frame dari kamera dan memberikannya ke model untuk dibuat inferensi.

Sebagian kode untuk penerapan model pada flutter diambil dari [github.com/shaqian/flutter\\_realtime\\_detection](https://github.com/shaqian/flutter_realtime_detection) Menggunakan library TensorFlow Lite, inferensi dilakukan dengan cara menggunakan fungsi `startImageStream` untuk memperoleh gambar dari kamera. Setiap frame dari kamera akan diberikan ke fungsi `detectObjectOnFrame` dengan parameter Model yang dipilih, besar dan lebar gambar yang dibuat 13 Megapixel, mengikuti spesifikasi kamera smartphone uji terendah.

```
controller.startImageStream((CameraImage img) {
  if (!isDetecting) {
    isDetecting = true;
```

Gambar 3. 9 Kode Fungsi StartImageStream

Kode diatas menggunakan library *Tensor flowLite*, inferensi dilakukan dengan cara menggunakan fungsi `startImageStream` untuk memperoleh gambar dari kamera.

```
Tflite.detectObjectOnFrame() {
  bytesList: img.planes.map((plane) {
    return plane.bytes;
  }).toList(),
  model: widget.model == yolo ? "YOLO" : "SSDMobileNet",
  imageHeight: img.height,
  imageWidth: img.width,
  imageMean: widget.model == yolo ? 0 : 127.5,
  imageStd: widget.model == yolo ? 255.0 : 127.5,
  numResultsPerClass: 1,
  threshold: widget.model == yolo ? 0.2 : 0.4,
}.then((recognitions) {
  int endTime = new DateTime.now().millisecondsSinceEpoch;
  print("Detection took ${endTime - startTime}");

  widget.setRecognitions(recognitions, img.height, img.width);

  isDetecting = false;
```



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Gambar 3. 10 Kode Fungsi *detectObjectOnFrame*

Setiap frame dari kamera akan diberikan ke fungsi *detectObjectOnFrame* dengan parameter Model yang dipilih, besar dan lebar gambar yang dibuat 13 Megapixel, mengikuti spesifikasi kamera smartphone uji terendah.

Berikut ini adalah kode untuk menampilkan bounding box, klasifikasi serta confidence hasil inferensi dari fungsi *detectObjectOnFrame*

```
@override
Widget build(BuildContext context) {
  List<Widget> _renderBoxes() {
    return results.map((re) {
      var _x = re["rect"]["x"];
      var _w = re["rect"]["w"];
      var _y = re["rect"]["y"];
      var _h = re["rect"]["h"];
      var scaleW, scaleH, x, y, w, h;

      return Positioned(
        left: math.max(0, x),
        top: math.max(0, y),
        width: w,
        height: h,
        child: Container(
          padding: EdgeInsets.only(top: 5.0, left: 5.0),
          decoration: BoxDecoration(
            border: Border.all(
              color: Color.fromRGB(37, 213, 253, 1.0),
              width: 3.0,
            ), // Border.all
          ), // BoxDecoration
          child: Text(
            "${re["detectedClass"]} ${(re["confidenceInClass"] * 100).toStringAsFixed(0)}%",
            style: TextStyle(
              color: Color.fromRGB(37, 213, 253, 1.0),
              fontSize: 14.0,
              fontWeight: FontWeight.bold,
            ), // TextStyle
          ), // Text
        ), // Container
      ); // Positioned
    }).toList();
  }
}
```

Gambar 3. 11 BoundingBox, klasifikasi dan confidence pada flutter

Model akan mengembalikan inferensi dalam bentuk JSON, dengan objek "detectedClass" sebagai klasifikasi, "confidenceInClass" sebagai *objectness*, serta ["rect"]["x"] dan ["rect"]["y"] sebagai titik tengah objek, dan ["rect"]["w"] dan ["rect"]["h"] sebagai tinggi dan lebar objek.



## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

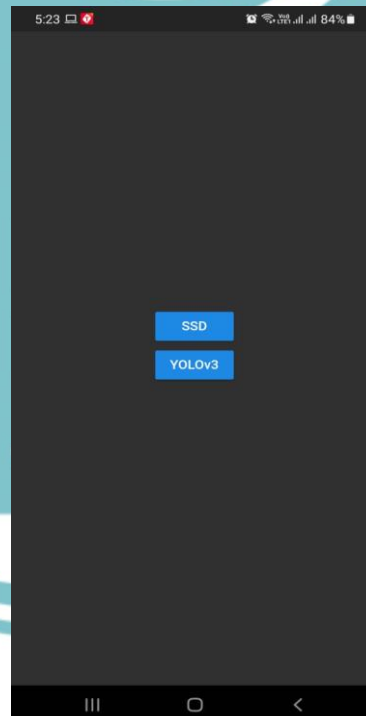
### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

```
@override
Widget build(BuildContext context) {
  Size screen = MediaQuery.of(context).size;
  return Scaffold(
    body: _model == ""
      ? Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            RaisedButton(
              child: const Text(ssd),
              onPressed: () => onSelect(ssd),
            ),
            RaisedButton(
              child: const Text(yolo),
              onPressed: () => onSelect(yolo),
            ),
          ],
        ),
      )
  );
}
```

Gambar 3. 12 Kode untuk Halaman Utama

Berikut ini adalah kode dan tampilan halaman utama, terdapat dua *button* untuk pemilihan model yang akan dipakai.



Gambar 3. 13 Halaman Utama

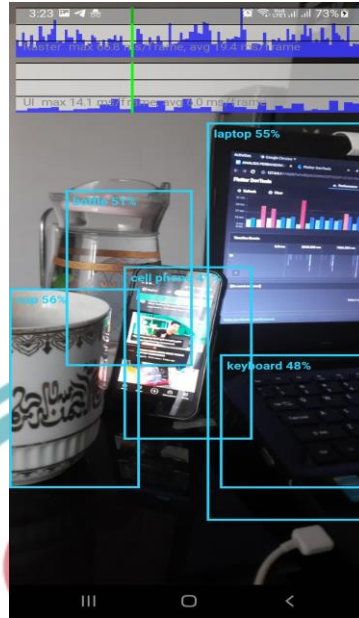




## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

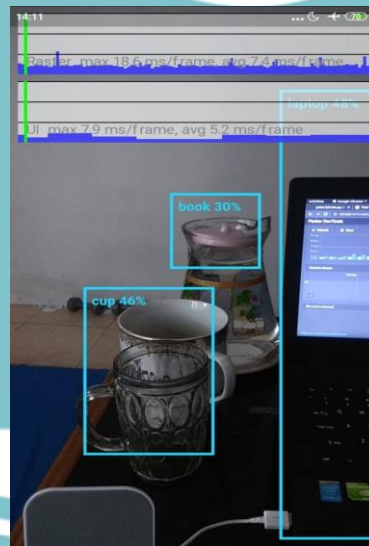
### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 3. 14 Hasil dari deteksi object pada Samsung M31 menggunakan SSD

Berikut ini adalah kode dan tampilan penerapan model SSD pada Samsung Galaxy M31, menggunakan indikator FPS di bagian atas layer.



Gambar 3. 15 Hasil deteksi object pada Xiaomi POCO M3 Pro 5G note 4 menggunakan YOLO

Berikut ini adalah kode dan tampilan penerapan model YOLO pada Xiaomi POCO M3 Pro 5G Note 4 Pro, menggunakan indikator FPS di bagian atas layer.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## BAB IV PEMBAHASAN

### 4.1 Deskripsi Pengujian Saat Training

Pengujian dilakukan untuk meminimalkan terjadinya kesalahan dalam sistem dan memastikan bahwa sistem memenuhi kebutuhan penggunaan. Pengujian atau tes yang sistematis menggambarkan beberapa hal: deskripsi pengujian; prosedur pengujian; data hasil pengujian; dan analisis atau evaluasi hasil pengujian.

### 4.2 Prosedur Pengujian

Prosedur pengujian akan menjelaskan rencana pengujian yang akan dilakukan. Prosedur pengujian pada penelitian ini dibagi menjadi dua yaitu, prosedur pengujian algoritma YOLO dan pengujian algoritma SSD. Sebelum diimplementasikan ke dalam aplikasi, model diuji secara otomatis menggunakan program. Parameter pengujian tersebut adalah *precision* dan *recall* terhadap data asli serta menghitung *loss function*. Kedua pengukuran tersebut diambil setiap satu siklus *epoch* saat training. Siklus *epoch* terjadi sebanyak 500 kali keseluruhan. Setelah berhasil diuji, model diimplementasikan ke aplikasi android. Model akan diuji secara langsung dengan waktu sebagai variabel. Setiap label dalam model akan diuji.

### 4.3 Hasil Pengujian YOLO dan SSD

Algoritma diuji sebanyak dua kali, ketika training dan setelah diterapkan ke android. Untuk pengujian saat *training*, digunakan *Confusion Matrix* dan *Loss Function* sebagai pengukuran. Metrik dari *Confusion Matrix* yang digunakan adalah *precision* dan *recall*, serta pengukuran yang digunakan untuk *Loss Function* adalah *Objectness*, *Localization* dan *Classification*.

Untuk pengujian setelah model diterapkan ke aplikasi, dilakukan menggunakan *hold-out testing*, serta pengukuran hanya menggunakan *confidence rate*. Pengujian dilakukan secara manual dengan waktu sebagai variabel.

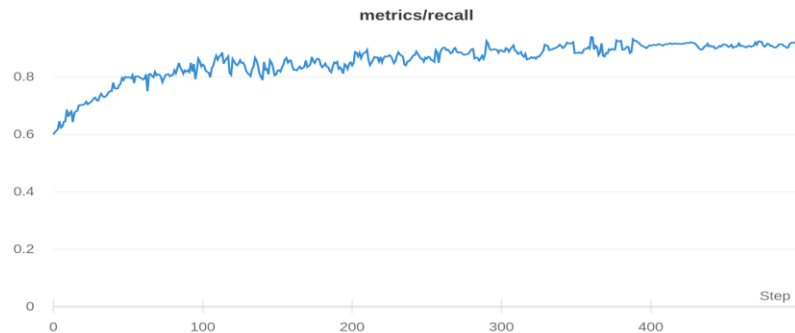


#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

### 4.3.2 Hasil Pengukuran Confusion Matrix and Loss

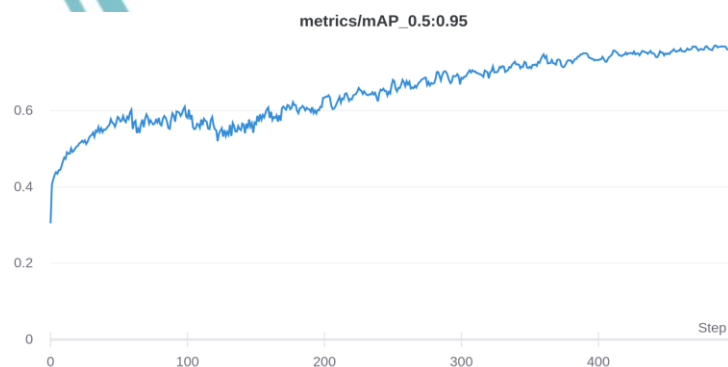
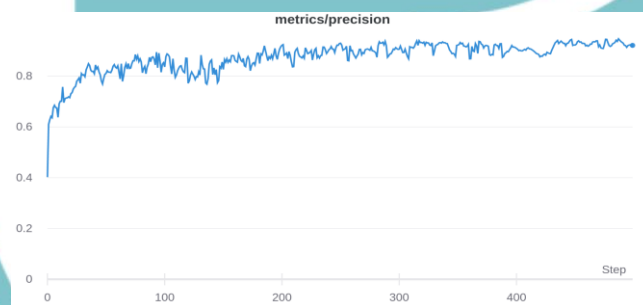
#### Penilaian Model Menggunakan Metrik *Recall*



Gambar 4. 1 Grafik Model Metrik Recall

Diawali dengan nilai 0.6 di epoch pertama, ketepatan *recall* dari model konsisten meningkat hingga lebih dari 0.8 sejak *epoch* ke 150. Menandakan performa ketepatan model meningkat setiap epochnya.

#### Penilaian Model Menggunakan Metrik *Precision* dan *mAP*



Gambar 4. 2 Grafik Model Metrik Precision dan mAP



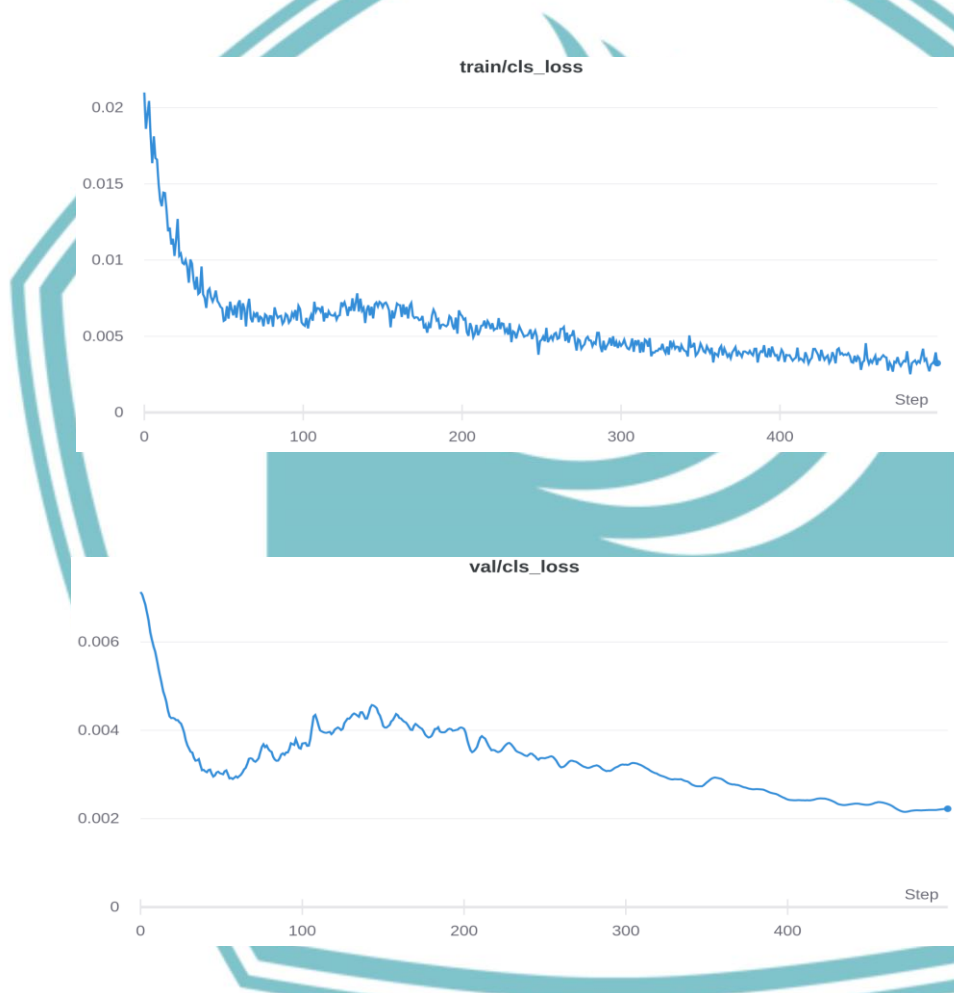


#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Diawali dengan nilai 0.4 di epoch pertama, ketepatan *precision* dari model konsisten meningkat hingga lebih dari 0.8 sejak *epoch* ke 150. Sementara hasil dari *Mean Average Precision* baru lebih dari 0.6 sejak *epoch* ke-200, namun terus meningkat sampai epoch terakhir. Menandakan performa ketepatan model meningkat setiap *epoch*-nya.

#### Perbandingan Classification Loss pada Data Training dan Validasi



Gambar 4. 3 Grafik Classification Loss pada Data Training dan Validasi

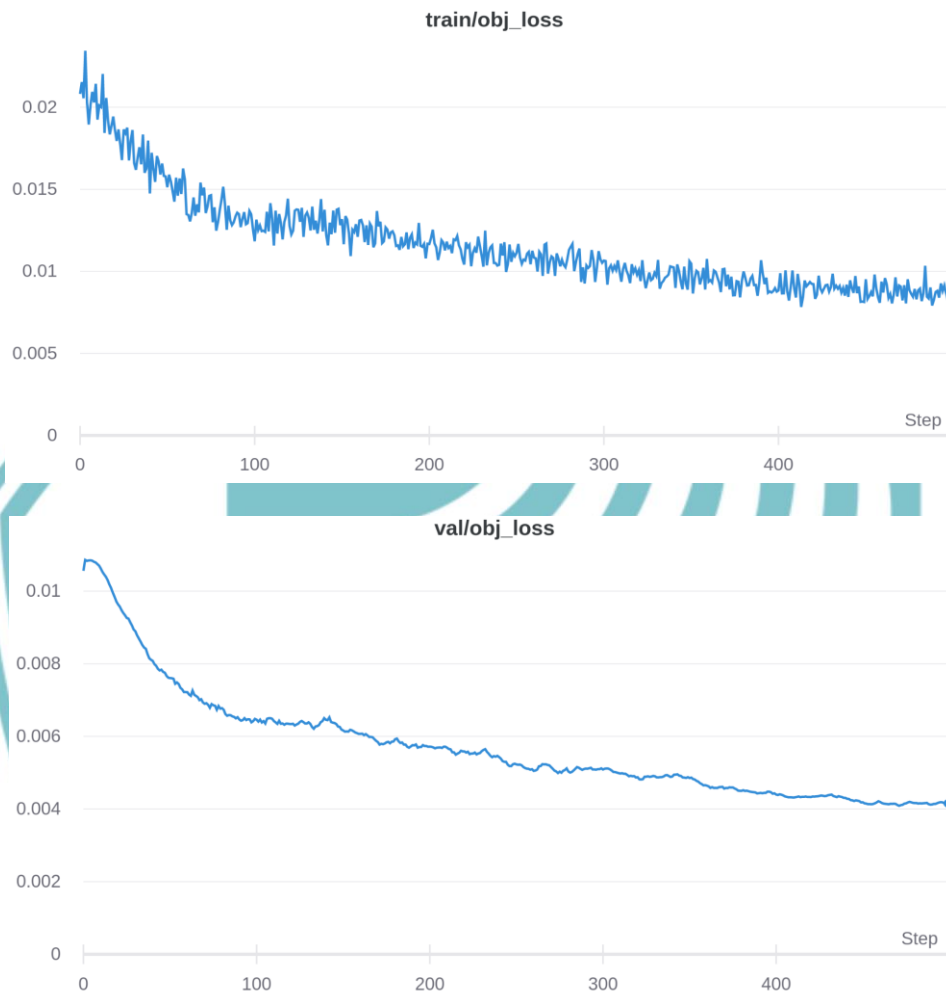
Model menghasilkan *classification loss* yang sangat tinggi senilai  $>0.006$  di beberapa epoch awal. Epoch ke-50 menghasilkan *local minima* namun nilai *loss* terus menurun setelah naik sampai  $>0.04$  di epoch ke-100 di pengujian dan validasi. Menandakan kemampuan model untuk memberi klasifikasi meningkat setiap epochnya.



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

### Perbandingan Objectness Loss pada Data Training dan Validasi



Gambar 4. 4 Grafik Objectness Loss pada Data Training dan Validasi

Model menghasilkan *objectness loss* yang sangat tinggi senilai  $>0.01$  di beberapa epoch awal, namun secara konsisten menurun. Menandakan model semakin yakin untuk memberi klasifikasi meningkat setiap epochnya.

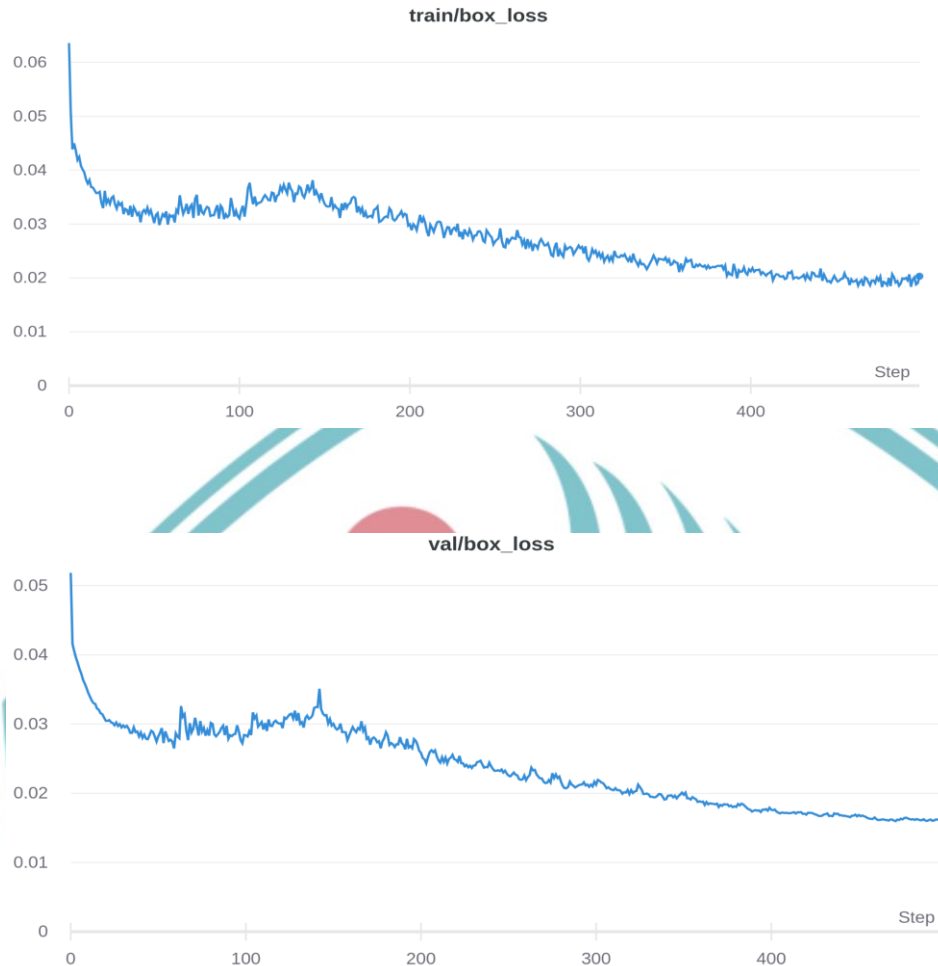
### Perbandingan Localization Loss pada Data Training dan Validasi



## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



Gambar 4. 5 Grafik Localization Loss pada Data Training dan Validasi

Model menghasilkan *localization loss* yang sangat tinggi senilai  $>0.05$  di beberapa epoch awal, namun menurun setelah epoch ke-50. Epoch ke-150 menghasilkan *peak* di pengujian dan validasi. Menandakan performa ketepatan model untuk memberi box meningkat setiap epochnya. Pengujian diadakan menggunakan 1 gambar untuk setiap label. Variabel dari pengujian tersebut adalah waktu, berapa lama model diberi waktu untuk mengklasifikasi





#### 4.4 Hasil Profiling YOLO dan SSD pada Flutter

Berikut ini adalah hasil dari pengujian YOLO dan SSD setelah diterapkan pada kedua *smartphone* uji

Tabel 4. 1 Profiling YOLO dan SSD pada Samsung M31

PERCOBAAN	YANG DIUJIKAN	ALGORITMA	
		YOLO	SSD
1	FPS	55	59
	Inference Count	17	25
	Average Inference (ms)	11	4
2	FPS	56	52
	Inference Count	23	25
	Average Inference (ms)	8	10
3	FPS	58	48
	Inference Count	22	25
	Average Inference (ms)	12	9
4	FPS	51	50
	Inference Count	22	25
	Average Inference (ms)	12	10
5	FPS	56	50
	Inference Count	23	25
	Average Inference (ms)	10	11

Tabel diatas menunjukkan bahwa:

1. Diperoleh rata-rata FPS untuk YOLO diperoleh 55.2 sedangkan untuk SSD adalah 51.8, YOLO lebih cepat sebesar 6% untuk FPS

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

2. Diperoleh rata-rata jumlah inferensi YOLO 21.4, sementara SSD adalah 25, YOLO menghasilkan jumlah inferensi lebih rendah 15%, menandakan YOLO tidak lebih baik dari SSD dalam menghasilkan inferensi.
3. Diperoleh rata-rata kecepatan inferensi YOLO 10.6, sementara SSD adalah 8.8, YOLO menghasilkan inferensi dengan lebih lambat 16%, menandakan YOLO tidak lebih baik dari SSD dalam kecepatan menghasilkan inferensi.

Tabel 4. 2 Profiling YOLO dan SSD pada Xiaomi POCO M3 Pro 5G Note 4 Pro

PERCOBAAN	YANG DIUJIKAN	ALGORITMA	
		YOLO	SSD
1	FPS	60	60
	Inference Count	23	25
	Average Inference (ms)	3	4
2	FPS	59	59
	Inference Count	20	25
	Average Inference (ms)	2	4
3	FPS	60	60
	Inference Count	22	25
	Average Inference (ms)	3	4
4	FPS	60	59
	Inference Count	22	25
	Average Inference (ms)	4	4
	FPS	60	59



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

PERCOBAAN	YANG DIUJIKAN	ALGORITMA	
		YOLO	SSD
5	Inference Count	24	25
	Average Inference (ms)	3	4

Tabel diatas menunjukan hasil dari pengujian pada POCO M3 Pro 5G Note 4 Pro, antara lain:

1. Diperoleh rata-rata FPS untuk YOLO adalah 59,9 dan SSD diperoleh 59,5. YOLO menghasilkan rerata FPS lebih tinggi senilai 0,4 FPS.
2. Diperoleh rata-rata jumlah inferensi YOLO 22.2, sementara SSD adalah 25, YOLO menghasilkan jumlah inferensi lebih rendah 13%, menandakan YOLO tidak lebih baik dari SSD dalam menghasilkan inferensi, namun sudah lebih baik dari pengujian di Samsung M31.
3. Diperoleh rata-rata kecepatan inferensi YOLO 3, sementara SSD adalah 4, YOLO menghasilkan kecepatan inferensi lebih cepat 25%, sedikit lebih baik daripada hasil di Samsung M31.

**POLITEKNIK  
NEGERI  
JAKARTA**





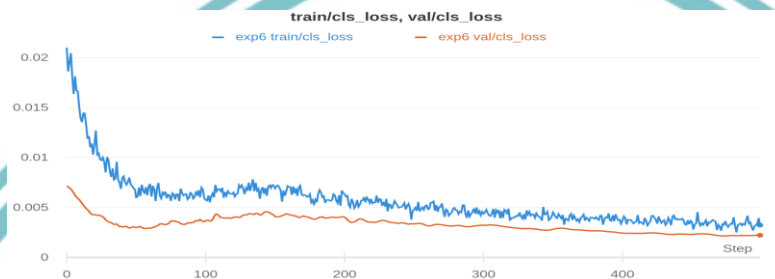
#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

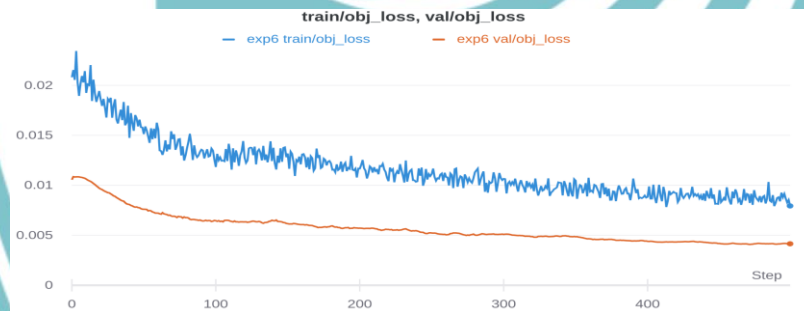
### 4.3 Analisa Data/Evaluasi

#### 4.3.1 Evaluasi Training

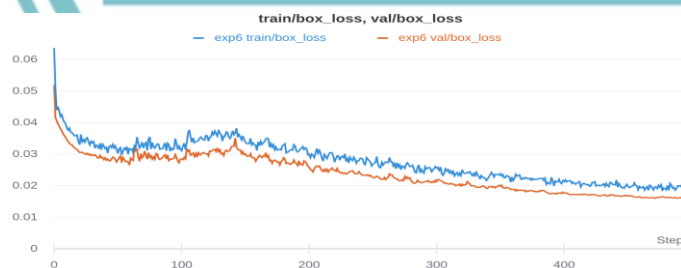
Berdasarkan pengukuran classification (cls), objectness (obj) dan localization (box) ditemukan bahwa hasil validasi cenderung lebih kecil daripada hasil training untuk classification dan objectness, namun nilai loss masih lebih dekat untuk localization.



Gambar 4. 6 Grafik Perbandingan Train dan Validation dari Loss Function Classification



Gambar 4. 7 Grafik Perbandingan Train dan Validation Objectness



Gambar 4. 8 Grafik Perbandingan Train dan Validation Localization



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Ketiga pengukuran *loss function* menunjukkan bahwa loss dari validasi lebih rendah dari saat training, menandakan bahwa data yang terpilih saat validasi lebih mudah.

#### 4.3.2 Evaluasi Performa

Diperoleh rata-rata FPS untuk YOLO dan SSD diperoleh 60. Smartphone uji tidak terpengaruh oleh kedua model untuk FPS. Diperoleh rata-rata jumlah inferensi YOLO 22.2, sementara SSD adalah 25, YOLO menghasilkan jumlah inferensi lebih rendah 13%, menandakan YOLO tidak lebih baik dari SSD dalam menghasilkan inferensi, namun sudah lebih baik dari pengujian di Samsung M31. Diperoleh rata-rata kecepatan inferensi YOLO 3.8, sementara SSD adalah 4, YOLO menghasilkan kecepatan inferensi lebih cepat 5%, sedikit lebih baik daripada hasil di Samsung M31

**POLITEKNIK  
NEGERI  
JAKARTA**



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## BAB V PENUTUP

### 5.1 Kesimpulan

Penelitian ini bertujuan untuk menguji akurasi dan performa dari algoritma deteksi objek YOLO dan SSD. Pengujian akurasi berhasil menunjukkan bahwa kedua model mampu menurunkan loss serta meningkatkan akurasi dengan 500 kali epoch. Hasil dari pengujian performa yang diperoleh menunjukkan bahwa YOLO memiliki FPS dan ms/Inference dan FPS yang sangat baik, lebih besar dari SSD, namun belum menunjukkan hasil yang lebih signifikan dari SSD. SSD yang memiliki jumlah inference jauh lebih tinggi daripada YOLO sehingga SSD menghasilkan inferensi dengan lebih baik.

### 5.2 Saran

Berdasarkan hasil penelitian ini, dapat dilakukan beberapa peningkatan yang bisa diimplementasikan, berikut di antaranya:

1. Terdapat beberapa metrik ukur performa yang belum diambil dalam penelitian ini, seperti penggunaan sumber daya CPU usage dan RAM usage ketika model dijalankan dalam aplikasi.
2. Pengayaan dari variasi *hardware* uji, seperti pemberagaman spesifikasi smartphone uji, iPhone, Arduino atau Raspberry Pi
3. Optimasi penggunaan sumberdaya seperti pemanfaatan *multi-threading* dan *GPU* dapat menghasilkan performa yang lebih cepat untuk masing-masing model.
4. Penilaian label klasifikasi secara langsung dari sisi smartphone dapat dilakukan guna memperoleh pengukuran akurasi serta performa secara bersamaan.
5. Pengembangan aplikasi menggunakan framework Hybrid dapat sedikit mempengaruhi performa saat penerapan, pengembangan aplikasi melalui *android* atau *ios native* dapat dipilih untuk penelitian selanjutnya.





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

**DAFTAR PUSTAKA**

- Andrea, A. (2018) *Tensor Flow Lite Android*. Available at: <https://adiandrea.id/articles/2018-05/tensor-flow-lite-android> (Accessed: 14 June 2021).
- Bresilla, K., Perulli, G.D., Boini, A., Morandi, B., Corelli Grappadelli, L. and Manfrini, L., 2019. Single-shot convolution neural networks for real-time fruit detection within the tree. *Frontiers in plant science*, 10, p.611.
- Huang, G.-B., Zhu, Q.-Y. and Siew, C.-K. (2006) 'Extreme learning machine: theory and applications', *Neurocomputing*, 70(1–3), pp. 489–501.
- Jeong, K. and Moon, H. (2011) 'Object detection using FAST corner detector based on smartphone platforms', in 2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering. IEEE, pp. 111–115.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T., 2014, November. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 675-678).
- Lewis, J. R. and Sauro, J. (2018) 'Item benchmarks for the system usability scale.', *Journal of Usability Studies*, 13(3).
- Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X. and Pietikäinen, M., 2020. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2), pp.261-318.
- Liu, L., Li, H. and Gruteser, M., 2019, August. Edge assisted real-time object detection for mobile augmented reality. In *The 25th Annual International Conference on Mobile Computing and Networking* (pp. 1-16).
- Mahmud, M.S., Zaman, Q.U., Esau, T.J., Chang, Y.K., Price, G.W. and Prithiviraj, B., 2020. Real-time detection of strawberry powdery mildew disease using a mobile machine vision system. *Agronomy*, 10(7), p.1027.
- Manning, C. D., Raghavan, P. and Schütze, H. (2009) 'Probabilistic information



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

retrieval’, Introduction to Information Retrieval, pp. 220–235.

Redmon, J. et al. (2016) ‘You only look once: Unified, real-time object detection’, in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788.

Redmon, J. and Farhadi, A. (2018) ‘Yolov3: An incremental improvement’, arXiv preprint arXiv:1804.02767.

Ren, Y., Zhu, C. and Xiao, S., 2018. Deformable Faster R-Cnn With Aggregating Multi-Layer Features For Partially Occluded Object Detection In Optical Remote Sensing Images. Remote Sensing, 10(9), P.1470.

Rochmawati, Y. and Kusumaningrum, R. (2016) ‘Studi Perbandingan Algoritma Pencarian String dalam Metode Approximate String Matching untuk Identifikasi Kesalahan Pengetikan Teks’, Jurnal Buana Informatika, 7(2), pp. 125–134. doi: 10.24002/jbi.v7i2.491.

Sanchez, S.A., Romero, H.J. and Morales, A.D., 2020, May. A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework. In IOP Conference Series: Materials Science and Engineering (Vol. 844, No. 1, p. 012024). IOP Publishing.

Srivastava, S., Divekar, A.V., Anilkumar, C., Naik, I., Kulkarni, V. and Pattabiraman, V., 2021. Comparative analysis of deep learning image detection algorithms. Journal of Big Data, 8(1), pp.1-27.

Wang, C.-Y., Bochkovskiy, A. and Liao, H.-Y. M. (2020) ‘Scaled-YOLOv4: Scaling Cross Stage Partial Network’, arXiv preprint arXiv:2011.08036.

Whitten, J. L., Bentley, L. D. and Dittman, K. C. (2004) ‘Metode desain dan analisis sistem’, Andi Offset, Yogyakarta.

Willox, M., Vossaert, J. and Naessens, V., 2016, May. Comparing performance parameters of mobile app development strategies. In Proceedings of the International Conference on Mobile Software Engineering and Systems (pp. 38-47).





## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X., 2019. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), pp.3212-3232.





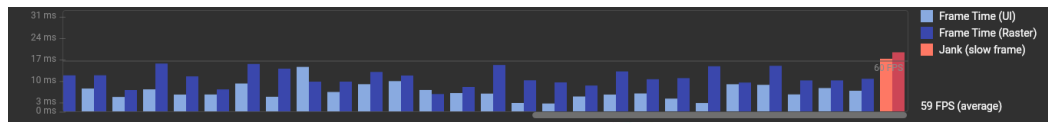


### Hak Cipta :

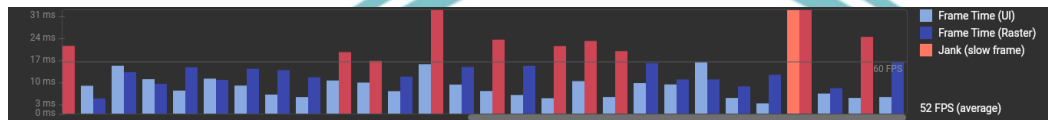
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

## Lampiran 1 Hasil Diagram Pengujian Performa

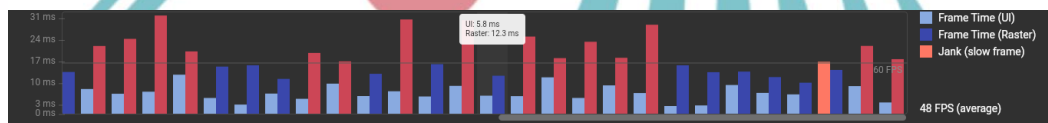
SSD di Samsung M31, dijalankan 5 kali masing-masing selama 5 detik



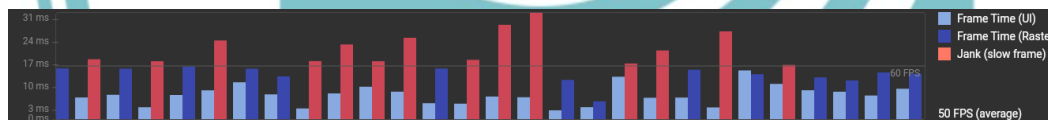
fps 59, jumlah inference 25, avg Inference = 4ms



fps 52, avg Inference = 10ms



fps 48, jumlah inference 25, avg Inference = 9ms



fps 50, jumlah inference 25, avg Inference = 10ms



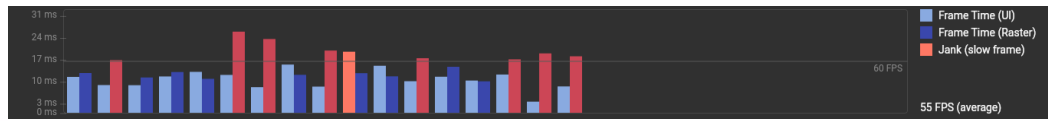
fps 50, jumlah inference 25, avg Inference = 11ms



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

YOLO di Samsung M31, dijalankan 5 kali masing-masing selama 5 detik



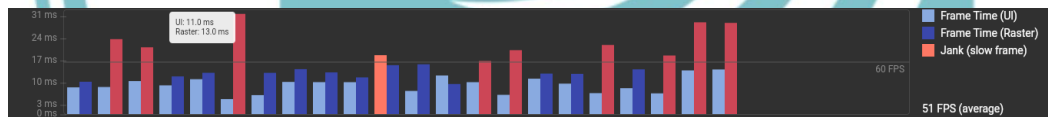
fps 55, jumlah inference 17, avg Inference = 11ms



fps 56, jumlah inference 23, avg Inference = 8ms



fps 58, jumlah inference 22, avg Inference = 9ms



fps 51, jumlah inference 22, avg Inference = 12ms



fps 56, jumlah inference 23, avg Inference = 10ms



### Hak Cipta :

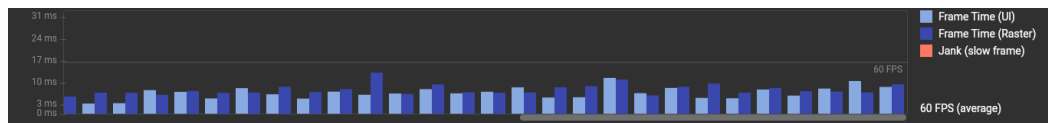
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

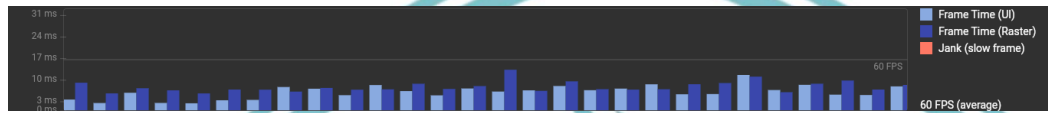
b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

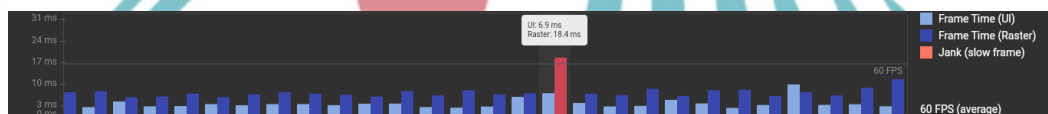
SSD di Xiaomi POCO M3 Pro 5G Note 4 Pro, dijalankan 5 kali masing-masing selama 5 detik



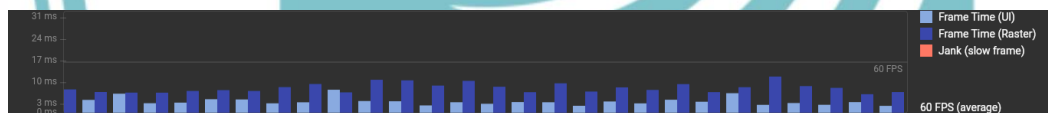
fps 60, jumlah inference 25, avg Inference = 4ms



fps 60, jumlah inference 25, avg Inference = 4ms



fps 60, jumlah inference 25, avg Inference = 4ms



fps 60, jumlah inference 25, avg Inference = 4ms



fps 60, jumlah inference 25, avg Inference = 4ms



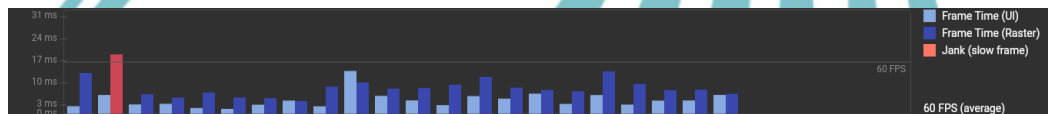


## © Hak Cipta milik Jurusan TIK Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumukan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

YOLO di Xiaomi POCO M3 Pro 5G Note 4 Pro, dijalankan 5 kali masing-masing selama 5 detik



POLITEKNIK  
NEGERI  
JAKARTA



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumpulkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin dari Jurusan TIK Politeknik Negeri Jakarta

*Lampiran 2 Daftar Riwayat Hidup*

**DAFTAR RIWAYAT HIDUP**



Fathur Rahman Haikal

Lahir di Jakarta, 13 Oktober 1999 (4 Rajab 1420 H). Penulis mengenyam pendidikan terakhir di SMAN 62 Jakarta dan lulus pada tahun 2017. Saat ini penulis sedang menempuh pendidikan Diploma IV Program Studi Teknik Informatika Jurusan Teknik Informatika dan Komputer di Politeknik Negeri Jakarta.

**POLITEKNIK  
NEGERI  
JAKARTA**