



UNIVERSITAS
Dinamika

**KLASIFIKASI UANG KERTAS RUPIAH BERDASARKAN ANGKA
NOMINAL SECARA *REALTIME* MENGGUNAKAN METODE *SINGLE
SHOT DETECTOR***



TUGAS AKHIR

Program Studi

S1 TEKNIK KOMPUTER

UNIVERSITAS
Dinamika

Oleh:

L'DIVA ANANDA CATYANINGGA

18410200054

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2023

**KLASIFIKASI UANG KERTAS RUPIAH BERDASARKAN ANGKA
NOMINAL SECARA *REALTIME* MENGGUNAKAN METODE *SINGLE
SHOT DETECTOR***

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Teknik



UNIVERSITAS
Dinamika

Disusun Oleh:

Nama : L'diva Ananda Catyaningga

NIM : 18410200054

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2023

TUGAS AKHIR

KLASIFIKASI UANG KERTAS RUPIAH BERDASARKAN ANGKA NOMINAL SECARA *REALTIME* MENGGUNAKAN METODE *SINGLE SHOT DETECTOR*

Dipersiapkan dan disusun oleh:

L'diva Ananda Catyaningga

NIM: 18410200054

Telah diperiksa, dibahas, dan disetujui oleh Dewan Pembahas

Pada: 8 Agustus 2023

Susunan Dewan Pembahas

Pembimbing:

- I. Heri Pratikno, M.T., MTCNA., MTCRE.
NIDN 0716117302
- II. Weny Indah Kusumawati, S.Kom., M.MT
NIDN 0721047201


Digitally signed by Heri Pratikno, M.T.
DN: cn=Heri Pratikno, M.T.,
o=Universitas Dinamika, ou=51 Teknik
Komputer,
email=heri@dinamika.ac.id, c=ID
Date: 2023.08.10 12:15:32 +0700'
Adobe Acrobat version: 11.0.23
Universitas
Dinamika
2023.08.10
11:11:43 +0700'

Pembahas:

- I. Pauladie Susanto, S.Kom., M.T.
NIDN 0729047501


Universitas Dinamika
2023.08.10 16:14:38
+0700'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar sarjana



Digitally signed by
Universitas Dinamika
Date: 2023.08.16
08:35:28 +0700'

Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

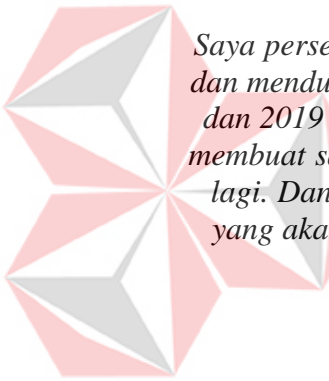
Dekan Fakultas Teknologi dan Informatika
UNIVERSITAS DINAMIKA



UNIVERSITAS
"Pain make peoples change"
-L-
Dinamika

Atas berkat Allah SWT,

Saya persembahkan tugas akhir ini kepada ayah dan ibu saya yang telah berdoa dan mendukung saya yang tiada habisnya. Bersama teman-teman Angkatan 2018 dan 2019 terima kasih banyak atas bantuan doa dan segala dukungannya yang membuat saya lebih semangat hidup dan belajar menjadi pribadi yang lebih baik lagi. Dan juga saya ingin mempersembahkan tugas akhir ini kepada siapapun yang akan mempelajari dan mengembangkan ilmu tentang kecerdasan buatan untuk menuju kearah yang lebih baik bagi manusia.



SURAT PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya:

Nama : L'diva Ananda Catyaningga
NIM : 18410200054
Program Studi : S1 Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Laporan Tugas Akhir
Judul Karya : **KLASIFIKASI UANG KERTAS RUPIAH
BERDASARKAN ANGKA NOMINAL SECARA
REALTIME MENGGUNAKAN METODE SINGLES
SHOT DETECTOR**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau sebagai pemilik pencipta dan Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 26 Juni 2023

Yang menyatakan




L'diva Ananda Catyaningga
NIM : 18410200054

ABSTRAK

Uang kertas menjadi alat berharga bagi masyarakat untuk alat pembayaran yang sah di kehidupan sehari-hari. Dalam terjadinya transaksi atau perdagangan, uang kertas memiliki masing-masing nilai yang telah tertera pada setiap uang kertas. Dengan itu masyarakat telah mampu membedakan nilai di setiap uang kertas. Di era ini tak luput juga uang kertas dapat dikenali dengan menggunakan mesin contohnya mesin ATM (*Automated Teller Machine*) tetapi hanya mengenali dua nilai uang kertas saja yaitu lima puluh ribu rupiah dan seratus ribu rupiah. Berdasarkan hal tersebut maka dibutuhkan suatu sistem yang cara kerjanya seperti otak manusia. Sistem itu harus dapat mengenali dan mengklasifikasikan nominal uang kertas secara cepat dan tepat seperti yang dilakukan oleh manusia. Proses pengenalan uang kertas tersebut dilihat dari gambar permukaannya atau pengenalan citra (*image recognition*). Sistem klasifikasi uang kertas dengan metode *Single Shot Detector* secara *realtime*. Berdasarkan penelitian-penelitian yang telah ada hasil dari metode SSD yang didapatkan akurasi nya tinggi dan akurat untuk mendeteksi objek, oleh karena itu penulis ingin menggunakan metode ini dalam penelitiannya. *Dataset* yang digunakan pun uang kertas mata uang rupiah (IDR) yang terdiri dari 7 subjek: Rp. 1.000, Rp. 2.000, Rp. 5.000, Rp. 10.000, Rp. 20.000, Rp. 50.000, Rp. 100.000. Pengujian dilakukan sebanyak 30 kali dengan 3 jarak yang berbeda dengan kriteria pengujian akurasi, *frame per second* (FPS) yang dihasilkan, dan performa sistem untuk mendeteksi nominal uang yang tertangkap kamera. Dari hasil pengujian didapatkan hasil yang terbaik untuk performa pendeteksian pada uang Rp. 50.000, dengan akurasi tertinggi 94.5% dan hasil performa kurang baik pada uang Rp. 1.000,- dan Rp. 10.000,-. Rata-rata total *frame per second* (FPS) yang dihasilkan dari ketiga jarak selama pengujian adalah 8-11 FPS. Hasil pengujian dipengaruhi 3 faktor yaitu warna mata uang, intensitas cahaya saat pengambilan sample gambar, dan kualitas kamera yang digunakan pada saat pengambilan sampel gambar mata uang.

Kata kunci: Uang Kertas Rupiah, Klasifikasi, *Single Shot Detector*, *Image Recognition*.

KATA PENGANTAR

Puji syukur saya panjatkan pada Tuhan Yang Maha Esa atas segala rahmat dan berkah yang telah diberikan-Nya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul “Klasifikasi Uang Kertas Rupiah Berdasarkan Angka Nominal Secara *Realtime* Menggunakan Metode *Single Shot Detector*”. Dalam perjalanan menyelesaikan pengerjaan laporan Tugas Akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Allah SWT, karena dengan rahmat-Nya penulis dapat menyelesaikan laporan Tugas Akhir ini.
2. Orang tua dan seluruh keluarga yang telah memberikan dorongan dan dukungan baik secara moril maupun materiil, sehingga penulis dapat menempuh dan menyelesaikan laporan Tugas Akhir ini.
3. Ibu Tri Sagirani, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
4. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
5. Bapak Pauladie Susanto, S.Kom., M.T., selaku Dosen Pembahas. Penulis mengucapkan terimakasih atas bimbingan yang diberikan dan kesempatan serta tuntunan baik secara lisan maupun tertulis sehingga penulis dapat menyelesaikan Tugas Akhir.
6. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku dosen pembimbing I yang telah memberikan dukungan berupa motivasi, wawasan, dan saran bagi penulis selama pelaksanaan pengerjaan tugas akhir dan dalam pembuatan laporan Tugas Akhir
7. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku dosen pembimbing II yang banyak memberikan masukan dan solusi agar Tugas Akhir ini menjadi lebih baik dan penulis dapat menyelesaikan tugas akhir ini.
8. Seluruh Dosen pengajar Program Studi S1 Teknik Komputer Universitas Dinamika yang telah memberikan ilmu, dan juga bimbingan yang berharga dari semester 1 hingga sampai saat ini.

9. Seluruh rekan–rekan S1 Teknik Komputer angkatan 2018 dan 2019 yang telah memberikan dukungan dan semangatnya untuk membantu penulis menyelesaikan laporan Tugas Akhir ini.
10. Dan Seluruh pihak yang tidak dapat disebutkan satu per satu yang telah memberikan dukungan serta bantuan dalam segala bentuk yang akhirnya terselesaikannya laporan Tugas Akhir ini.

Penulis berharap semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan laporan ini banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan saran dan kritik untuk memperbaiki kekurangan dan berusaha untuk lebih baik lagi.

Surabaya, 8 Agustus 2023

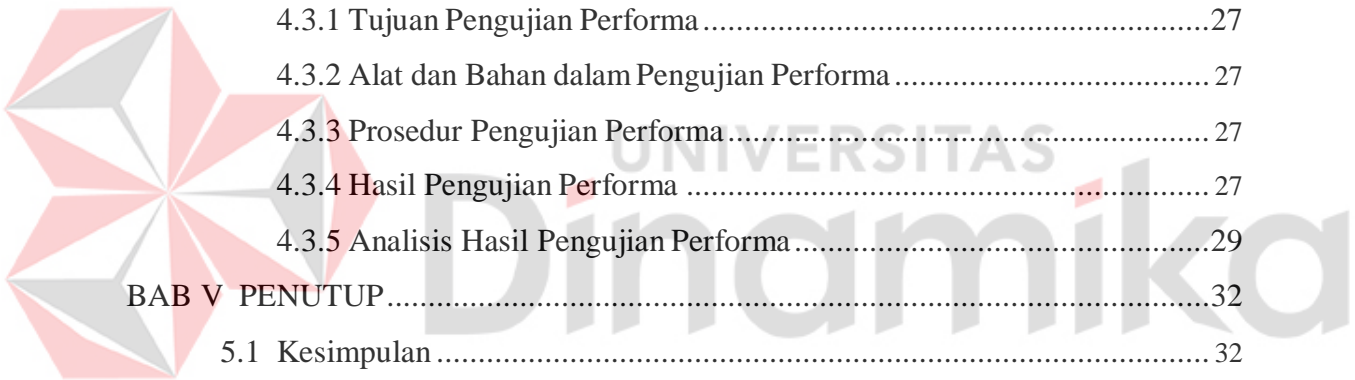


UNIVERSITAS
Dinamika Penulis

DAFTAR ISI

ABSTRAK.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
BAB II LANDASAN TEORI.....	4
2.1 Uang Kertas.....	4
2.2 <i>Deep Learning</i>	4
2.3 Python.....	5
2.4 Anaconda.....	5
2.5 OpenCV.....	5
2.6 <i>Object Detection</i>	6
2.7 <i>Single Shot Detector (SSD)</i>	6
2.8 Nvidia CUDA.....	7
BAB III METODOLOGI PENELITIAN.....	8
3.1 <i>Setting Virtual Environment</i>	8
3.2 Instalasi Cuda dan CUDNN.....	8
3.3 <i>Dataset</i>	8
3.4 <i>LabelImg</i>	10
3.5 <i>Training</i>	12
3.6 <i>Testing</i>	15
3.7 Spesifikasi Hardware.....	17

BAB IV HASIL DAN PEMBAHASAN	18
4.1 <i>Training dataset</i> 7 nominal uang Rupiah	18
4.1.1 Alat dan Bahan dalam proses <i>training</i>	18
4.1.2 Prosedur <i>training dataset</i> model SSD	18
4.1.3 Hasil <i>training dataset</i> model SSD	18
4.2 Pengujian Akurasi dan FPS	20
4.2.1 Tujuan Pengujian Akurasi dan FPS	20
4.2.2 Alat dan Bahan dalam Pengujian Akurasi dan FPS	20
4.2.3 Prosedur Pengujian Akurasi dan FPS	21
4.2.4 Hasil Pengujian Akurasi dan FPS	21
4.2.5 Analisis Hasil Pengujian Akurasi dan FPS	24
4.3 Pengujian Performa	27
4.3.1 Tujuan Pengujian Performa	27
4.3.2 Alat dan Bahan dalam Pengujian Performa	27
4.3.3 Prosedur Pengujian Performa	27
4.3.4 Hasil Pengujian Performa	27
4.3.5 Analisis Hasil Pengujian Performa	29
BAB V PENUTUP	32
5.1 Kesimpulan	32
5.2 Saran	33
DAFTAR PUSTAKA	34
LAMPIRAN	35
BIODATA PENULIS	41



DAFTAR GAMBAR

Gambar 2.1 Uang Kertas.....	4
Gambar 2.2 OpenCV	5
Gambar 2.3 Arsitektur SSD-MobileNetV2.....	7
Gambar 2.4 Logo Nvidia CUDA	7
Gambar 3.1 Salah satu contoh pengambilan citra untuk <i>dataset</i>	9
Gambar 3.2 Hasil <i>crop</i> dari rekaman video untuk digunakan <i>dataset</i>	11
Gambar 3.3 Proses melabel <i>dataset</i> yang diprediksi.....	12
Gambar 3.4 Gambar <i>dataset</i> beserta data labeling nya dengan format .xml	13
Gambar 3.5 Isi dari data label di .xml	15
Gambar 3.6 Alur proses <i>training dataset</i>	15
Gambar 3.7 <i>Dataset</i> diinputkan ke dalam folder <i>enviromtment</i>	16
Gambar 3.8 <i>Generate</i> data .xml menjadi .csv pada <i>anaconda prompt</i>	14
Gambar 3.9 Proses <i>training dataset</i>	15
Gambar 3.10 Hasil <i>save</i> model <i>training</i> dalam bentuk .ckpt.....	12
Gambar 3.11 Alur proses <i>testing</i>	13
Gambar 3.12 Berhasil mengaktifkan <i>enviromtment</i> yang berisi data model <i>training</i> .	16
Gambar 4.1 Grafik <i>Learning Rate Training</i>	19
Gambar 4.2 Grafik <i>Loss Training</i>	19

DAFTAR TABEL

Tabel 4.1 Hasil pengujian akurasi dan FPS jarak 20 cm	21
Tabel 4.2 Hasil pengujian akurasi dan FPS jarak 60 cm	28
Tabel 4.3 Hasil pengujian akurasi dan FPS jarak 100 cm	23
Tabel 4.4 Hasil pengujian performa jarak 20 cm	28
Tabel 4.5 Hasil pengujian performa jarak 60 cm	28
Tabel 4.6 Hasil pengujian performa jarak 100 cm	28



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

Lampiran 1 Program untuk Test.....	35
------------------------------------	----



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Uang kertas menjadi alat berharga bagi masyarakat untuk alat pembayaran yang sah di kehidupan sehari-hari. Dalam terjadinya transaksi atau perdagangan, uang kertas memiliki masing-masing nilai yang telah tertera pada setiap uang kertas. Diketahui uang kertas rupiah yang sering yang digunakan ada 7 nilai yaitu seribu rupiah, dua ribu rupiah, lima ribu rupiah, sepuluh ribu rupiah, dua puluh ribu rupiah, lima puluh ribu rupiah, dan seratus ribu rupiah. Dengan itu masyarakat telah mampu membedakan nilai di setiap uang kertas. Di era ini tak luput juga uang kertas dapat dikenali dengan menggunakan mesin contohnya mesin ATM (*Automated Teller Machine*) tetapi hanya mengenali dua nilai uang kertas saja yaitu lima puluh ribu rupiah dan seratus ribu rupiah.

Berdasarkan hal tersebut maka dibutuhkan suatu sistem yang cara kerjanya seperti otak manusia. Sistem itu harus dapat mengenali dan mengklasifikasikan nominal uang kertas secara cepat dan tepat seperti yang dilakukan oleh manusia. Sistem tersebut akan digunakan pada mesin otomatis. Jika mesin menerima salah satu pecahan uang kertas, mesin itu dapat langsung mengenali nominal dari uang tersebut. Sehingga sistem ini dapat dikembangkan dan digunakan sebagai alat bantu penjualan yang melibatkan mesin. Proses pengenalan uang kertas tersebut dilihat dari gambar permukaannya atau pengenalan citra (*image recogniton*).

Adapun hasil karya yang terkait tentang klasifikasi nominal uang kertas yang dilakukan oleh (Izah, 2018), dengan menggunakan metode CNN-MXNet. Hasil penelitian tersebut adalah sistem dapat mengklasifikasi nominal uang kertas yang sudah lusuh. Dengan jumlah data 700 citra yang dibagi dua *dataset*, yaitu 560 data yang telah di *training* dan 140 data untuk test. Diperoleh akurasi sebesar 100% untuk data train dan 93.57% untuk data test. Kekurangan dari penelitian ialah alat masih berupa sistem jadi tidak dapat digunakan secara *realtime*.

Berdasarkan penelitian sebelumnya yang dilakukan oleh (Priadana &

Murdiyanto, 2019) berjudul “Metode SURF dan FLANN untuk Identifikasi Nominal Uang Kertas Rupiah Tahun Emisi 2016 pada Variasi Rotasi”. Sistem yang dibuat mampu mengidentifikasi uang kertas rupiah dengan dua metode yaitu metode SURF untuk ekstraksi ciri dan metode FLANN untuk *feature matching*. Dengan cara merotasi objek dengan berbagai sudut : 0° , 90° , 180° , 270° tingkat akurasi identifikasi uang yang diperoleh sebesar 100%. Kekurangan dari penelitian ini adalah hasil penelitian hanyalah sebuah sistem, belum bisa diimplementasikan secara *realtime*.

Lalu pada penelitian sebelumnya yang dilakukan diluar negeri oleh (Zhang, 2018) yang berjudul “Currency Recognition Using Deep Learning” ini menggunakan berbagai metode. Untuk *training* peneliti menggunakan metode CNN dengan hasil akurasi mencapai 96.6%, untuk pengenalan mata uang nya menggunakan dua metode untuk sebagai perbandingan saja yaitu MobileNet dan Faster R-CNN. Hasil yang dicapai dari MobileNet lebih akurat daripada Faster R-CNN. Proyek ini hanya bisa mengenali mata uang *New Zealand Dollar* (NZD).

Pada Tugas Akhir ini, penulis ingin membuat sistem klasifikasi uang kertas dengan metode *Single Shot Detector* secara *realtime*. Berdasarkan penelitian-penelitian yang telah ada hasil dari metode SSD yang didapatkan akurasi nya tinggi dan akurat untuk mendeteksi objek, oleh karena itu penulis ingin menggunakan metode ini dalam penelitiannya. *Dataset* yang digunakan pun uang kertas mata uang rupiah (IDR) yang terdiri dari 7 subjek: Rp. 1.000, Rp. 2.000, Rp. 5.000, Rp. 10.000, Rp. 20.000, Rp. 50.000, Rp. 100.000.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan masalah pada Tugas Akhir ini sebagai berikut:

1. Bagaimana performa klasifikasi nominal uang kertas menggunakan metode SSD secara *realtime* melalui webcam?
2. Berapa besar FPS pada proses klasifikasi nominal uang kertas menggunakan metode SSD?

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, pembahasan masalah dibatasi pada beberapa hal berikut:

1. Tidak bisa mendeteksi uang logam.
2. Alat hanya bisa mendeteksi satu uang kertas dalam satu waktu bersamaan.
3. Pencahayaan ruangan merata.
4. Tidak dapat mendeteksi uang palsu.
5. Hanya dapat mendeteksi uang tahun emisi 2016.

1.4 Tujuan

Berdasarkan latar belakang dan rumusan masalah diatas, mendapatkan tujuan pada tugas akhir ini sebagai berikut:

1. Dapat mengukur performa klasifikasi nominal uang kertas menggunakan metode SSD secara *realtime* melalui webcam.
2. Mengetahui besar FPS pada proses klasifikasi nominal uang kertas menggunakan metode SSD.

1.5 Manfaat

Adapun dari Tugas Akhir ini dapat diperoleh manfaat sebagai berikut:

1. Untuk menambah pengetahuan dan penerapan mengenai klasifikasi uang kertas berdasarkan nominal menggunakan metode *Single Shot Detector*.
2. Menjadi referensi bagi yang ingin melanjutkan penelitian mengenai sistem klasifikasi uang kertas rupiah.
3. Dapat menjadi solusi untuk layanan transaksi di area publik.
4. Penelitian ini memberikan kontribusi berupa sistem klasifikasi uang kertas rupiah berdasarkan nominal menggunakan metode *Single Shot Detector* secara *realtime*.

BAB II

LANDASAN TEORI

2.1 Uang Kertas

Menurut Kamus Besar Bahasa Indonesia, Uang adalah alat penukar atau standar pengukur nilai (kesatuan hitungan) yang sah, yang dikeluarkan oleh pemerintah suatu negara berupa uang kertas, emas, perak, atau uang lain yang dicetak dengan bentuk dan gambar tertentu. Uang kertas merupakan uang yang tercetak dari kertas dengan gambar dan nominal tertentu dan merupakan alat pembayaran yang sah. Uang kertas mempunyai nilai karena nominalnya. Oleh karena itu, uang kertas hanya memiliki dua macam nilai, yaitu nilai nominal dan nilai tukar (Wariant, 2017).



Gambar 2.1 Uang Kertas
(Sumber: (Putri, 2022))

2.2 Deep Learning

Deep Learning merupakan bagian dari *machine learning* yang menggunakan jaringan syaraf tiruan untuk penerapan permasalahan yang memakai *dataset* yang berjumlah besar (Setiawan, 2021). Teknik ini memberikan arsitektur yang akurat untuk *Supervised Learning*. Dengan menggunakan *deep learning*, sebuah komputer dapat mempelajari dan mengklasifikasikan tugas-tugasnya melalui data yang dikumpulkan. Data tersebut bisa berupa gambar, teks, hingga suara. Bahkan, tingkat akurasi pun lebih tinggi dalam mengolah data-data berjumlah besar.

2.3 Python

Python yakni bahasa pemrograman yang bersifat *open source* dan mudah dipahami. Meskipun tidak sukar dimengerti dikarenakan bahasa nya mendekati bahasa manusia, kinerja python terbilang lebih lambat ketika dijalankan (*run*) tidak seperti bahasa pemrograman yang lain (Muhammad, 2021). Sebab python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks dan juga bila terjadi kesalahan penulisan lebih mudah dimengerti dan diperbaiki oleh manusia.

2.4 Anaconda

Anaconda adalah platform ilmu data gratis menggunakan bahasa pemrograman python atau R yang bersifat *open source* pada mesin lokal. Anaconda sering digunakan untuk *deep learning* bahkan tidak jarang para *data scientist* menggunakan platform ini. Dengan versi paket yang dikelola oleh conda sistem manajemen paket, Anaconda menyediakan banyak *package* dan *library* untuk diinstall dengan keperluan membuat proyek *deep learning* bahkan *data scientist*. Beberapa dari *library* nya ialah Numpy, matplotlib, pandas, scipy, dan masih banyak lagi.

2.5 OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah sebuah library open source untuk membuat suatu program citra digital. Di dalam OpenCV mempunyai banyak fitur seperti pengenalan wajah, pengenalan objek dan berbagai jenis model AI (*Artificial Intelligence*) (Sidharta, 2017).



Gambar 2.2 *OpenCV*
(Sumber: Wikipedia, 2019)

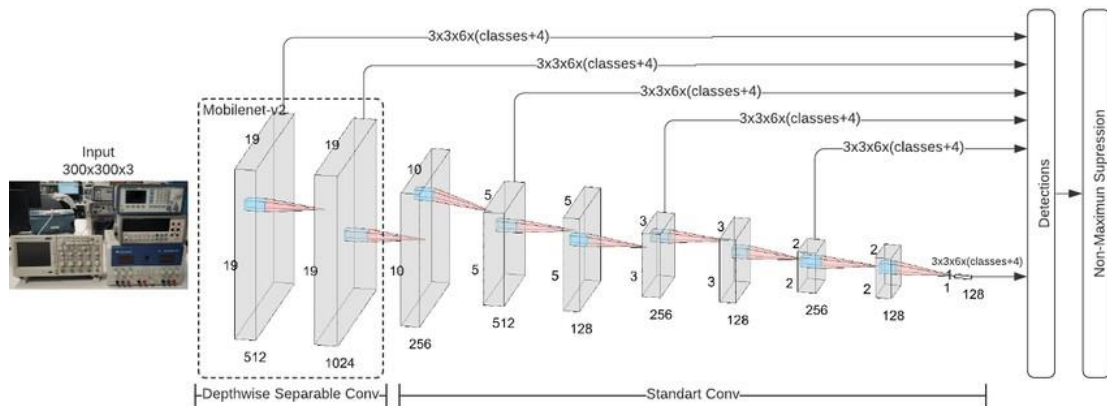
2.6 *Object Detection*

Object Detection ialah teknik klasifikasi citra dan juga mengidentifikasi suatu objek dari sejumlah *class* yang telah ditentukan dengan memberi *bounding box* ketika dapat menemukan *class* yang telah berhasil di prediksi. Teknik ini sering juga digunakan pada proyek yang berbasis *realtime* seperti deteksi benda, deteksi kendaraan, deteksi wajah, dan lain-lain (Alfi, 2020).

2.7 *Single Shot Detector (SSD)*

Single Shot Detector adalah arsitektur *Neural Network* yang dikembangkan oleh tim *Google Research* pada tahun 2016 sebagai melengkapi kebutuhan model yang dapat digunakan untuk deteksi secara *realtime* pada perangkat tanpa perbedaan akurasi yang signifikan, SSD menangkap satu bidikan tunggal untuk mendeteksi banyak objek dalam citra dengan menggunakan *bounding box* dan klasifikasi secara bersamaan (Hidayatullah, 2022). SSD dibangun agar berdiri sendiri tidak bergantung pada jaringan dasar, sehingga dapat dijalankan diatas hampir semua *neural network* lain, termasuk MobileNet.

Pada penelitian ini menggunakan model arsitektur SSD-MobileNetV2, yang dimana model ini terdiri dari 2 jenis *Neural Network* yaitu *Single Shot Detector (SSD)* dan MobileNetV2. MobileNet adalah arsitektur *Neural Network* yang ringan dan bisa digunakan dalam komputasi rendah contohnya *mobile application*. Arsitektur *Single Shot Detector (SSD)* menerapkan MobileNet sebagai *base network* lalu dilanjutkan beberapa *convolution layer* seperti pada Gambar 2.3 dengan *Fully-Connected*, *Maxpool*, dan *softmax* dihapus.



Gambar 2.3 Arsitektur SSD-MobileNetV2
(Sumber: researchgate.net, 2022)

Jadi *output* dari *convolution layer* diakhir milik MobileNetV2 digunakan sebagai penggantinya. Model SSD-MobileNetV2 menyatukan output dari 6 tingkat resolusi dan melakukan perhitungan hingga ribuan *bounding box*, kemudian di bagian akhir melakukan filter *bounding box* dengan menggunakan *Non-Maximum Supression*.

2.8 Nvidia CUDA



Gambar 2.4 Logo Nvidia CUDA
(Sumber: dignited.com, 2022)

CUDA yakni sebuah tambahan utilisasi dari Nvidia untuk menjalankan suatu program yang eksekusi nya membutuhkan GPU buatan Nvidia. Sering juga digunakan untuk jaringan syaraf pada *deep learning* yang membutuhkan kartu grafis yang tinggi (Kurnia, 2013).

BAB III METODOLOGI PENELITIAN

3.1 *Setting Virtual Environment*

Dalam Tugas Akhir ini sebelum memasuki proses selanjutnya perlu dilakukan instalasi dan pengaturan *environment* pada anaconda. Perlunya proses ini karena proyek objek deteksi ini membutuhkan *library* khusus yang ada pada anaconda. Dengan memasang *Python 3.7* sebagai bahasa programnya dan *library* yang diperlukan pada terminal pip dan Anaconda.

3.2 Instalasi Cuda dan CUDNN

Menginstal Cuda dan CUDNN ini hanya khusus pengguna *device* yang memakai Nvidia *graphic card*. Perlunya penginstalan ini dilakukan untuk proses *training* data dengan menggunakan *Graphic Processing Unit* (GPU) yang bisa mempersingkat proses *training*. Dengan adanya GPU yang memadai waktu proses *training* data dapat dipercepat, yang umumnya membutuhkan 15 hingga 20 jam *training* menjadi hanya 3 sampai 5 jam proses *training*.

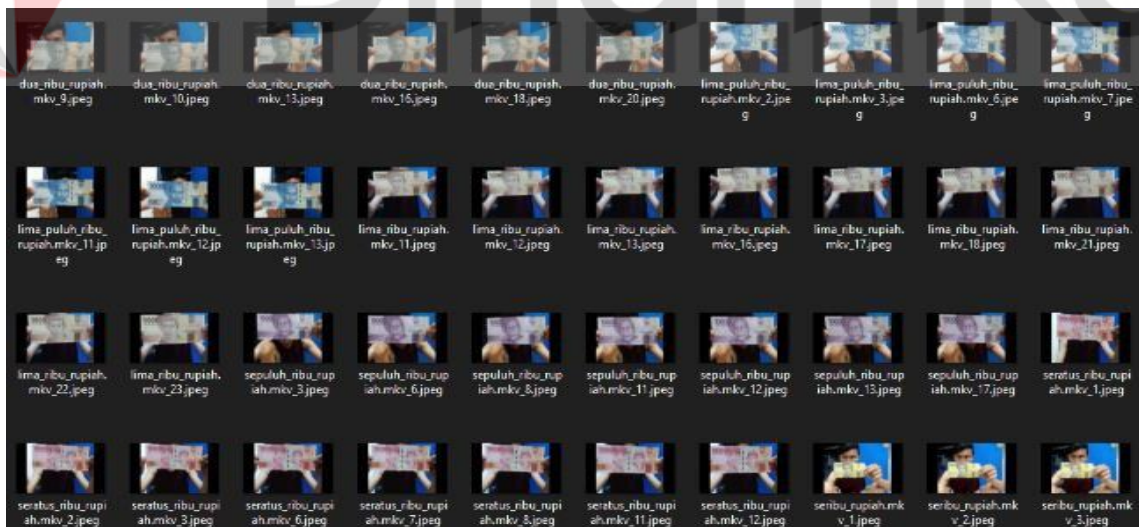
3.3 *Dataset*

Dataset yang dipakai pada Tugas Akhir ini adalah *dataset* yang telah dibuat dan dikumpulkan oleh penulis sendiri menggunakan *webcam*. Dengan spesifikasi *webcam* yaitu mempunyai resolusi maksimal *full HD* 1080p di 30 FPS dan 720p di 60 FPS, lalu untuk resolusi foto nya 3 *megapixel*, dan memiliki fitur autofokus juga. Untuk *dataset* berisi citra tujuh macam uang kertas yang sedang dipegang terdiri dari ‘Rp. 1.000’, ‘Rp. 2.000’, ‘Rp. 5.000’, ‘Rp. 10.000’, ‘Rp. 20.000’, ‘Rp. 50.000’, ‘Rp. 100.000’. *Dataset* yang telah dikumpulkan berjumlah 1400 citra dengan setiap nominalnya berisi 200 citra.



Gambar 3.1 Salah satu contoh pengambilan citra untuk *dataset*

Pengambilan citra dilakukan seperti Gambar 3.1 yaitu dengan cara merekam video penulis menggunakan *webcam* yang sedang memegang subjek uang dengan menggerakkan uang yang dipegang ke kiri ataupun ke kanan dan juga dimiringkan setelah itu hasil di *crop* atau dipotong disetiap detiknya untuk mendapatkan berbagai macam posisi uang yang dipegang.

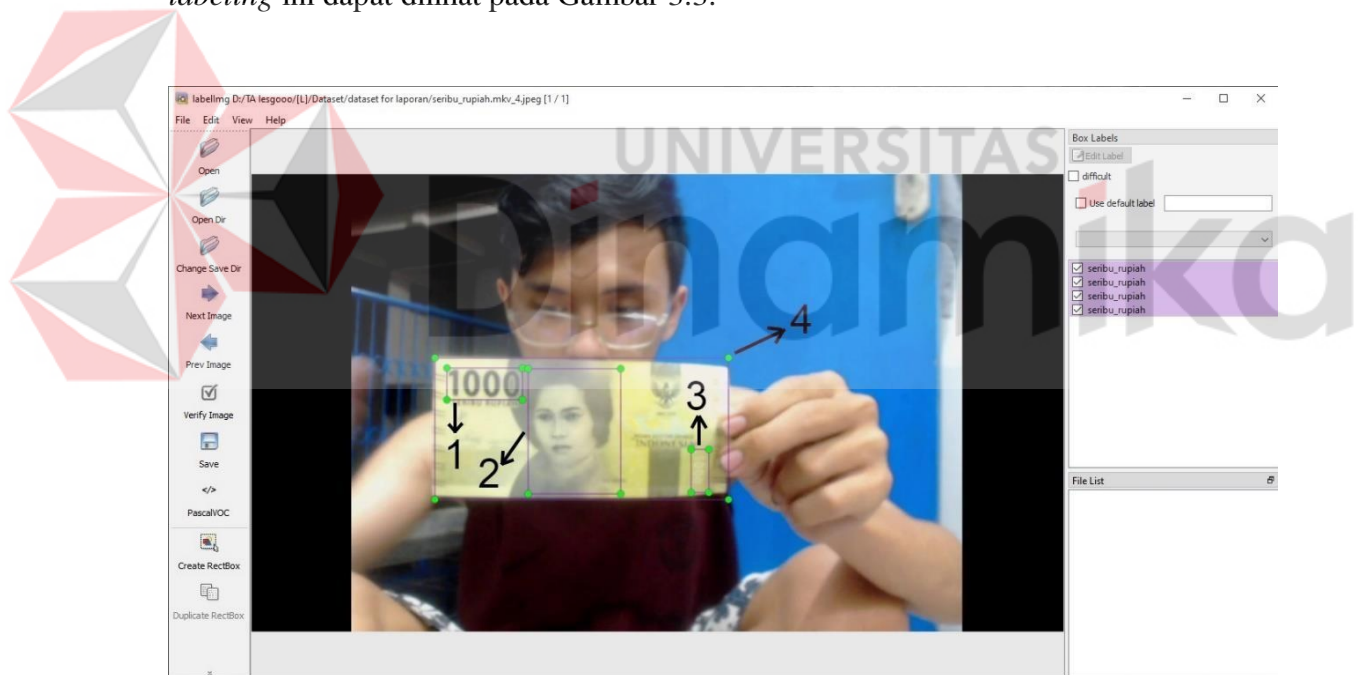


Gambar 3.2 Hasil *crop* dari rekaman video untuk digunakan *dataset*

Pada Gambar 3.2 diatas adalah hasil yang didapatkan dari *crop* rekaman videonya sebanyak 1.400 gambar. Seluruh *dataset* nantinya dibagi menjadi 2 bagian dengan komposisi 80% untuk *training*, 20% untuk *test*. Pembagiannya yaitu *training* sebanyak 1.120 gambar dan untuk *test* sebanyak 280 gambar.

3.4 LabelImg

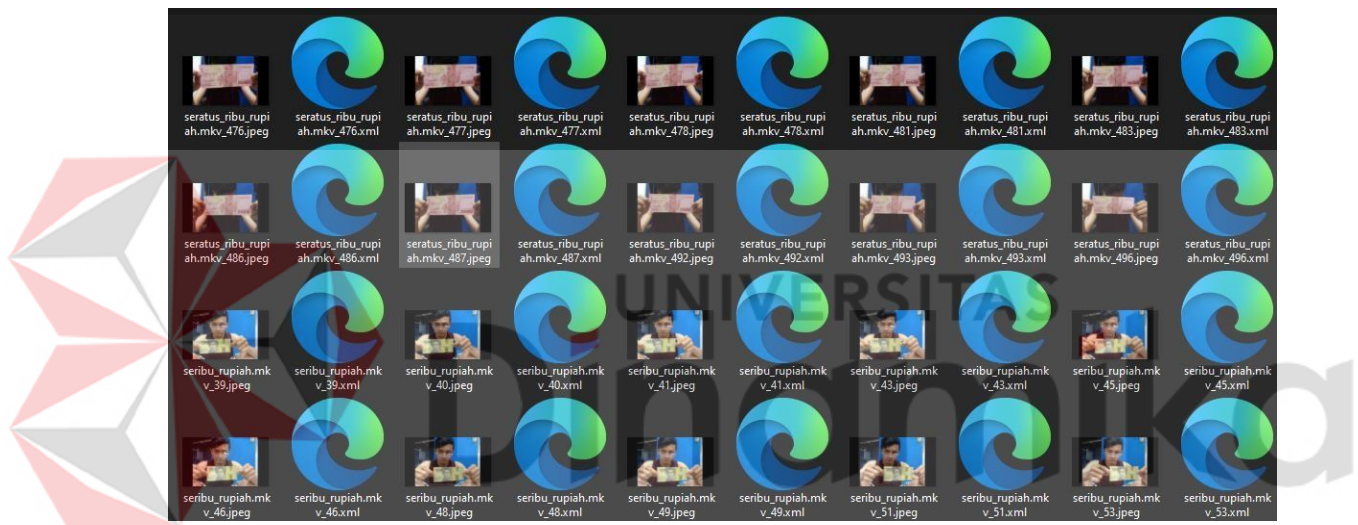
Lalu setelah proses pembuatan *dataset*, gambar-gambar tersebut berlanjut ke proses bernama *labeling*. Proses ini perlu dilakukan untuk melabel setiap subjek sehingga model *training* dapat mengetahui subjek yang diprediksi. Dengan menggunakan software *LabelImg* dari python, gambar *dataset* dilabeli dengan format .xml dan otomatis tersimpan pada folder *training* dan *test*. Sebagai contoh proses *labeling* ini dapat dilihat pada Gambar 3.3.



Gambar 3.3 Proses melabel *dataset* yang diprediksi
(Keterangan : 1 = Nominal Horizontal, 2 = Gambar tengah, 3 = Nominal Vertikal, 4 = Full)

Dataset yang terkumpul sebanyak 1.400 gambar oleh penulis diberi label prediksi secara manual satu per satu, agar hasil model *training* yang diperoleh memiliki akurasi bagus, saat proses *labeling* dianjurkan memberi label sesuai yang diperlukan,

tidak kebesaran maupun kelebaran, seperti pada Gambar 3.3. Penulis membagi subjeknya setiap 1 nominal uang menjadi 4 seperti di Gambar 3.3, yaitu Full (memprediksi keseluruhan lembar uang), Nominal Horizontal (memprediksi nominal uang yang tertulis horizontal pada lembar uang), Nominal Vertikal (memprediksi nominal uang yang tertulis vertikal pada lembar uang), Gambar Tengah (memprediksi gambar yang ada di tengah lembar uang). Setelah memberi label prediksi sebanyak 1.400 gambar selanjutnya hasil dari proses *labeling* didapatkan dengan format file .xml dan dijadikan satu folder dengan citra aslinya seperti pada Gambar 3.4.



Gambar 3.4 Gambar *dataset* beserta data labelingnya dengan format .xml

Untuk isi dari file .xml tersebut dapat dilihat pada Gambar 3.5, terdapat koordinat lokasi label yang dibuat oleh penulis dan juga terdapat panjang dan lebar ukuran gambar *dataset*, nama-nama class yang digunakan, serta lokasi file .xml tersebut.

```

<annotation>
  <folder>seratus_ribu_rupiah</folder>
  <filename>seratus_ribu_rupiah.mkv_476.jpeg</filename>
  <path>C:\Users\lenovo\Desktop\noan\file Kuliah\My Project\TA Diva\RealTimeObjectDetection-main\Dataset Images\seratus_ribu_rupiah\seratus_ribu_rupiah.mkv_476.jpeg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1280</width>
    <height>720</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>seratus_ribu_rupiah</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>242</xmin>
      <ymin>112</ymin>
      <xmax>1067</xmax>
      <ymax>480</ymax>
    </bndbox>
  </object>
  <object>
    <name>seratus_ribu_rupiah</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>819</xmin>
      <ymin>343</ymin>
      <xmax>1015</xmax>
      <ymax>429</ymax>
    </bndbox>
  </object>
  <object>
    <name>seratus_ribu_rupiah</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>498</xmin>
      <ymin>141</ymin>
      <xmax>831</xmax>
      <ymax>464</ymax>
    </bndbox>
  </object>

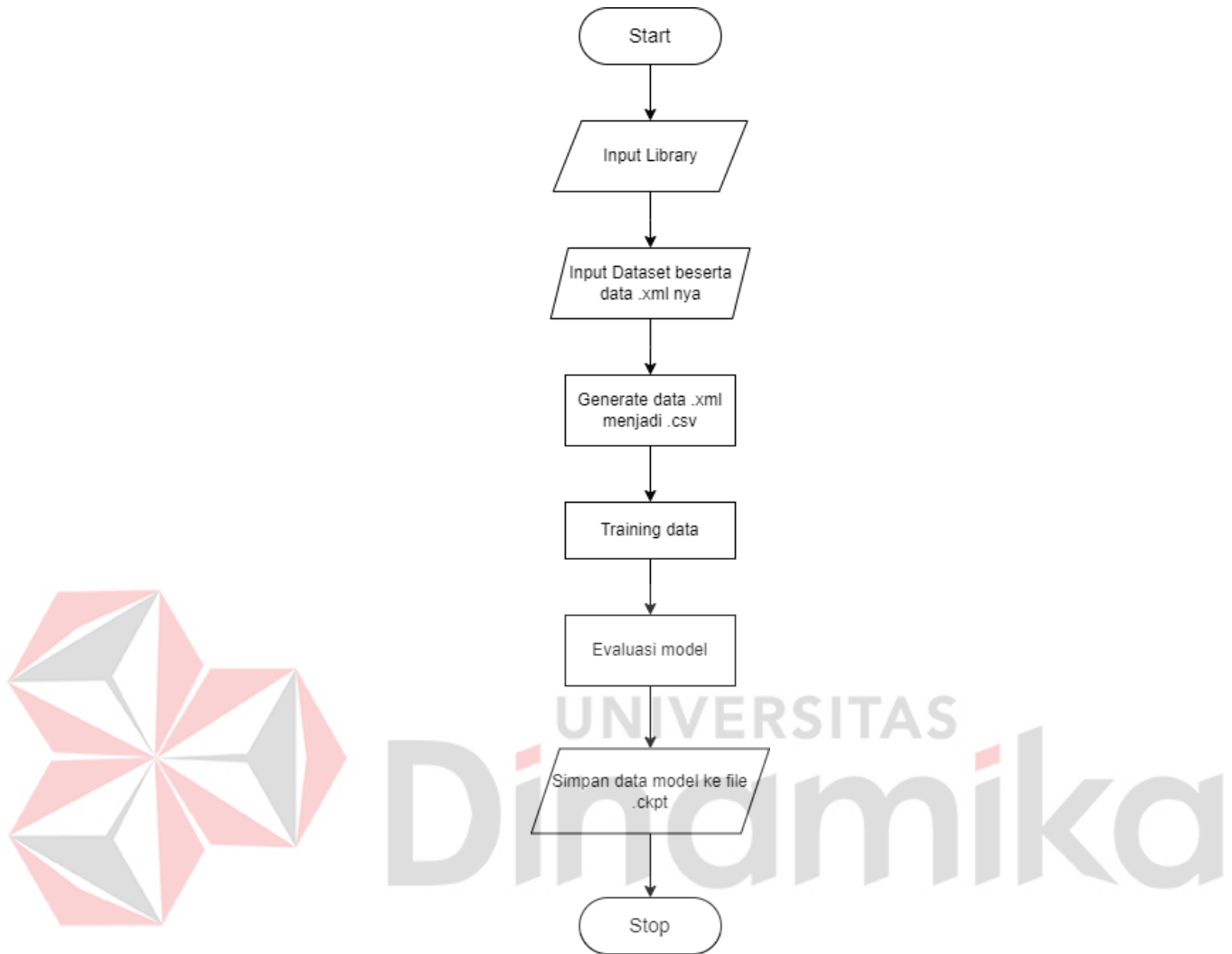
```

Gambar 3.5 Isi dari data label di .xml

3.5 Training

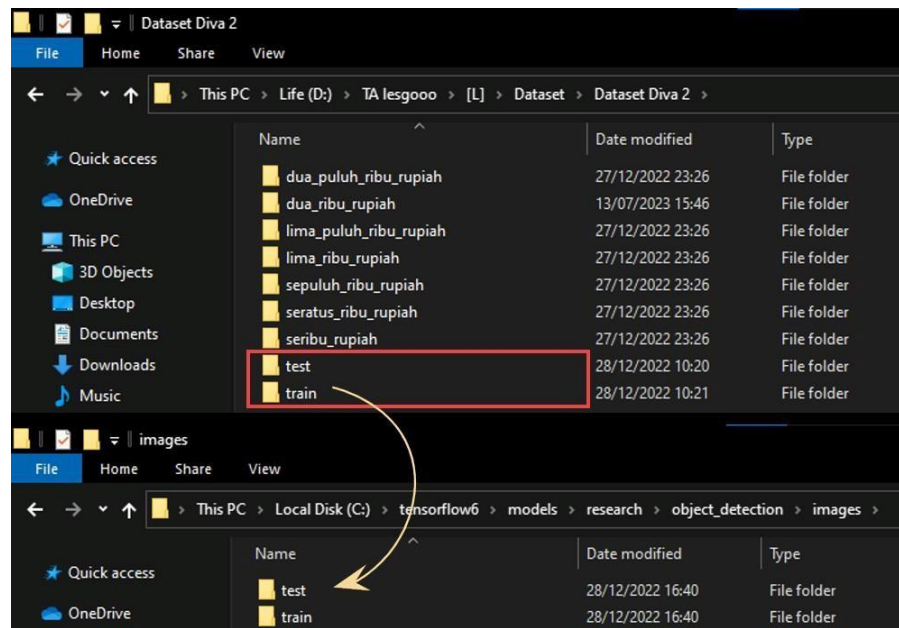
Pada Gambar 3.6 menunjukkan alur proses *training dataset* yang sedang diuji pada proyek Tugas Akhir ini. Dengan awal dari *Start* lalu melakukan *input library* atau *import library* pada *anaconda prompt* yang digunakan seperti *anaconda protobuf*, *pillow*, *lxml*, *Cython*, *contextlib2*, *jupyter*, *matplotlib*, *pandas*, *opencv-python*. Setelah itu memberikan *input* gambar *dataset* yang akan *ditraining* dan sudah dibagi menjadi 80% data *train* 20% data *test* ke dalam folder *enviromtment* yang digunakan seperti pada Gambar 3.7. Di dalam folder ini juga terdapat *dataset* yang telah diberi label dengan menggunakan *LabelImg* untuk memberikan koordinat posisi dari citra yang *ditraining* dengan hasil format berupa .xml.

Dataset yang sudah berupa .xml otomatis di generate oleh sistem menjadi format .csv dengan menggunakan *anaconda prompt* seperti pada Gambar 3.8. Karena *enviromtment* dari Anaconda hanya bisa menerima data berupa .csv, jadi *dataset* perlu dikonversi terlebih dahulu untuk yang berformat .xml.

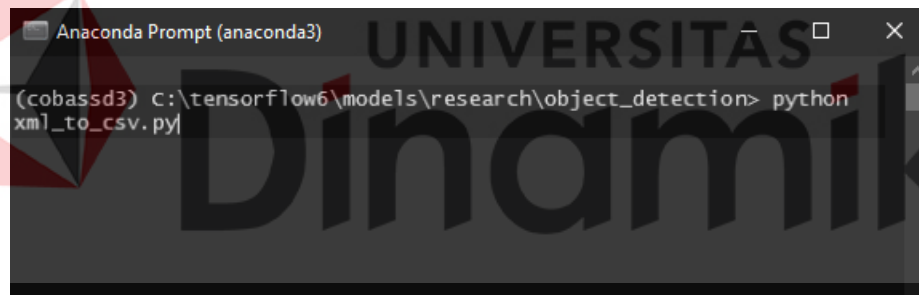


Gambar 3.6 Alur proses *training dataset*

Kemudian memasuki proses *training* dengan model SSD, proses *training* ini dilakukan sebanyak 10.000 step. Pada Gambar 3.9 di setiap step menunjukkan *loss*, *loss* tersebut untuk mengetahui apakah model yang sedang *training* ini berlatih dengan optimal atau tidak. Lalu lanjut ke proses evaluasi model, diproses ini tujuannya untuk mengevaluasi hasil model *training* yang disimpan pada model. Untuk melihat evaluasi model perlu menggunakan tensorboard, yang berisi hasil total *loss* saat *training* dan *learning rate* saat proses *training*.



Gambar 3.7 Dataset diinputkan ke dalam folder *environment*



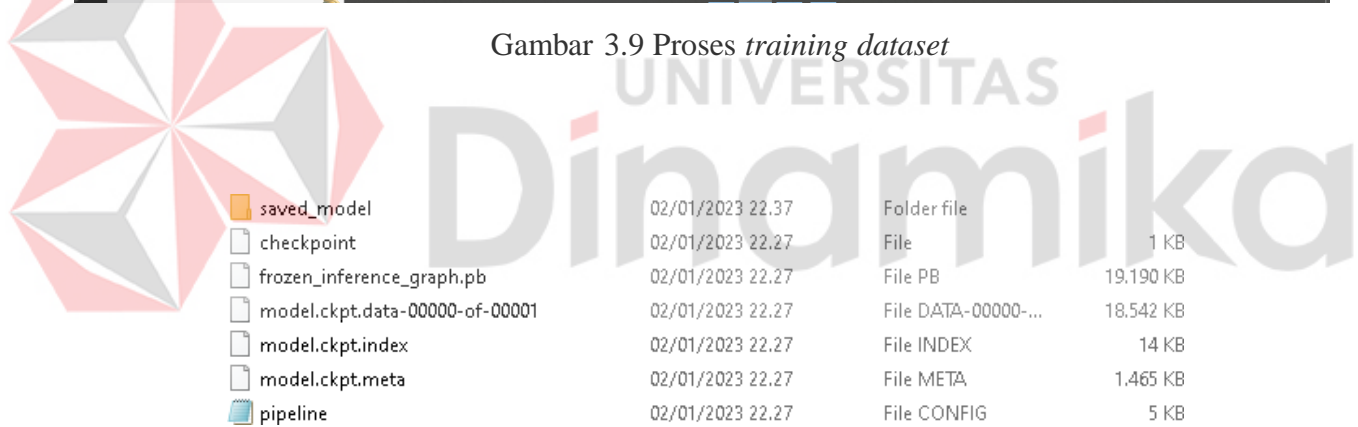
Gambar 3.8 Generate data .xml menjadi .csv pada *anaconda prompt*

Dengan akhir dari proses *training*, hasil model otomatis tersimpan berupa format .ckpt file dan beberapa file lainnya di folder *training* di dalam *environment* yang digunakan seperti pada Gambar 3.10. Penggunaan file .ckpt pada model *training* objek deteksi *realtime* sering digunakan karena dapat melakukan *tuning* atau mengoptimalkan lagi model *training* dan juga hasil akurasi dapat dimaksimalkan lagi.

```

Anaconda Prompt (anaconda3) - tensorboard --logdir=training
INFO:tensorflow:global step 5543: loss = 1.8202 (0.801 sec/step)
I0102 21:11:28.212754 9880 learning.py:512] global step 5543: loss = 1.8202 (0.801 sec/step)
INFO:tensorflow:global step 5544: loss = 1.7927 (0.796 sec/step)
I0102 21:11:29.009622 9880 learning.py:512] global step 5544: loss = 1.7927 (0.796 sec/step)
INFO:tensorflow:global step 5545: loss = 2.1200 (0.797 sec/step)
I0102 21:11:29.808365 9880 learning.py:512] global step 5545: loss = 2.1200 (0.797 sec/step)
INFO:tensorflow:global step 5546: loss = 1.8809 (0.760 sec/step)
I0102 21:11:30.570359 9880 learning.py:512] global step 5546: loss = 1.8809 (0.760 sec/step)
INFO:tensorflow:global step 5547: loss = 2.3724 (0.779 sec/step)
I0102 21:11:31.350859 9880 learning.py:512] global step 5547: loss = 2.3724 (0.779 sec/step)
INFO:tensorflow:global step 5548: loss = 2.6955 (0.796 sec/step)
I0102 21:11:32.148727 9880 learning.py:512] global step 5548: loss = 2.6955 (0.796 sec/step)
INFO:tensorflow:global step 5549: loss = 1.7268 (0.798 sec/step)
I0102 21:11:32.947591 9880 learning.py:512] global step 5549: loss = 1.7268 (0.798 sec/step)
INFO:tensorflow:global step 5550: loss = 1.7371 (0.796 sec/step)
I0102 21:11:33.745456 9880 learning.py:512] global step 5550: loss = 1.7371 (0.796 sec/step)
INFO:tensorflow:global step 5551: loss = 2.2366 (0.787 sec/step)
I0102 21:11:34.535340 9880 learning.py:512] global step 5551: loss = 2.2366 (0.787 sec/step)
INFO:tensorflow:global step 5552: loss = 2.2771 (0.792 sec/step)
I0102 21:11:35.329216 9880 learning.py:512] global step 5552: loss = 2.2771 (0.792 sec/step)
INFO:tensorflow:global step 5553: loss = 1.9788 (0.785 sec/step)
I0102 21:11:36.116109 9880 learning.py:512] global step 5553: loss = 1.9788 (0.785 sec/step)
INFO:tensorflow:global step 5554: loss = 1.9411 (0.795 sec/step)
I0102 21:11:36.911989 9880 learning.py:512] global step 5554: loss = 1.9411 (0.795 sec/step)
INFO:tensorflow:global step 5555: loss = 2.0329 (0.805 sec/step)
I0102 21:11:37.710921 9880 learning.py:512] global step 5555: loss = 2.0329 (0.805 sec/step)
INFO:tensorflow:global step 5556: loss = 1.8343 (0.775 sec/step)
I0102 21:11:38.496738 9880 learning.py:512] global step 5556: loss = 1.8343 (0.775 sec/step)
INFO:tensorflow:global step 5557: loss = 1.9145 (0.784 sec/step)
I0102 21:11:39.281639 9880 learning.py:512] global step 5557: loss = 1.9145 (0.784 sec/step)
INFO:tensorflow:global step 5558: loss = 2.0356 (0.790 sec/step)
I0102 21:11:40.074517 9880 learning.py:512] global step 5558: loss = 2.0356 (0.790 sec/step)
INFO:tensorflow:global step 5559: loss = 2.3209 (0.762 sec/step)
I0102 21:11:40.837474 9880 learning.py:512] global step 5559: loss = 2.3209 (0.762 sec/step)
INFO:tensorflow:global step 5560: loss = 2.0636 (0.813 sec/step)
I0102 21:11:41.652291 9880 learning.py:512] global step 5560: loss = 2.0636 (0.813 sec/step)
INFO:tensorflow:global step 5561: loss = 1.8084 (0.792 sec/step)
I0102 21:11:42.447198 9880 learning.py:512] global step 5561: loss = 1.8084 (0.792 sec/step)
INFO:tensorflow:global step 5562: loss = 2.0619 (0.793 sec/step)
I0102 21:11:43.242040 9880 learning.py:512] global step 5562: loss = 2.0619 (0.793 sec/step)
INFO:tensorflow:global step 5563: loss = 1.9431 (0.777 sec/step)
I0102 21:11:44.019958 9880 learning.py:512] global step 5563: loss = 1.9431 (0.777 sec/step)
INFO:tensorflow:global step 5564: loss = 2.0876 (0.784 sec/step)
I0102 21:11:44.806851 9880 learning.py:512] global step 5564: loss = 2.0876 (0.784 sec/step)

```

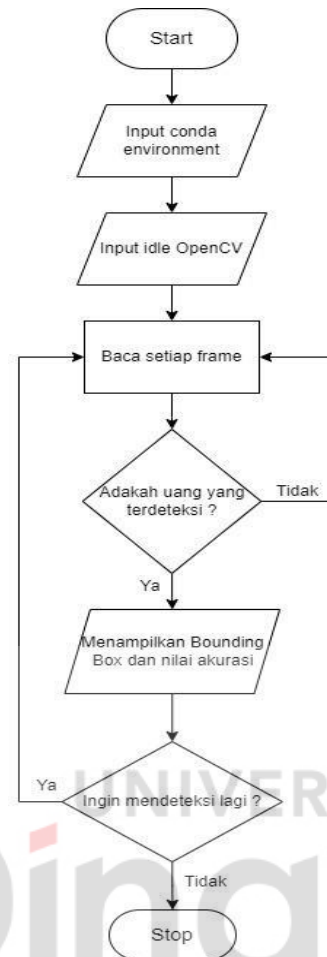
Gambar 3.9 Proses *training dataset*


saved_model	02/01/2023 22:37	Folder file	
checkpoint	02/01/2023 22:27	File	1 KB
frozen_inference_graph.pb	02/01/2023 22:27	File PB	19.190 KB
model.ckpt.data-00000-of-00001	02/01/2023 22:27	File DATA-00000-...	18.542 KB
model.ckpt.index	02/01/2023 22:27	File INDEX	14 KB
model.ckpt.meta	02/01/2023 22:27	File META	1.465 KB
pipeline	02/01/2023 22:27	File CONFIG	5 KB

Gambar 3.10 Hasil *save model training* dalam bentuk *.ckpt*

3.6 Testing

Setelah model *training* diperoleh, proses mendeteksi citra secara *realtime* dapat dijalankan dengan menggunakan file *.ckpt* yang telah disimpan pada model *training* diatas.



Gambar 3.11 Alur proses *testing*

Proses *testing* secara *realtime* bisa dilihat di Gambar 3.11. Pada langkah awal dimulai dari memasukkan *conda environment* yang didalamnya ada file *.ckpt* dari *save model training*. Penulis menggunakan *environment* bernama “cobassd3”, lalu diaktifkan pada *anaconda prompt* seperti pada Gambar 3.12

```

Anaconda Prompt (anaconda3)
(base) C:\Users\L>conda activate cobassd3
(cobassd3) C:\Users\L>
  
```

Gambar 3.12 Berhasil mengaktifkan *environment* yang berisi data model *training*

Lalu lanjut ke membuka idle OpenCV tetap melalui *anaconda prompt* untuk mendapatkan akses *webcam* yang digunakan mendeteksi objek. Setelah idle muncul buka file “object_detection_webcam.py” pada idle tersebut. Ketika idle dari “object_detection_webcam.py” sudah terbuka maka lanjut *run module* tersebut. Jika *webcam* telah terbuka dan berjalan, pembacaan frame pun dilakukan seiring program deteksi berjalan dan menampilkan *frame per second* (FPS) pada layar deteksi. Bila terdapat objek uang yang telah di *training* terdeteksi, maka *bounding box* dan nilai akurasi pun muncul dilayar sebagai output. *Bounding box* terbentuk sesuai koordinat lokasi yang sudah ditentukan saat proses *labeling* pada citra dan sekaligus melakukan klasifikasi *class* nominal uang nya beserta menampilkan nilai akurasi dalam bentuk persen. Ketika tidak ada objek uang yang terdeteksi pada *webcam*, program pendeteksian tetap berjalan dan menampilkan FPS pada layar. Namun bila tidak ingin mendeteksi uang lagi, maka tekan tombol “Q” pada keyboard memberhentikan program deteksi dan juga mematikan *webcam*.

3.7 Spesifikasi Hardware

Berikut adalah hardware-hardware yang digunakan saat proses *training* maupun *testing* beserta spesifikasi nya.

- *Training*

Pada proses *training* penulis menggunakan laptop dengan spesifikasi menggunakan prosesor Intel core i3-4005U dual core 1,7Ghz dan memiliki kartu grafis Nvidia Geforce GT 930M dengan serta VRAM 2GB lalu memori RAM sebesar 8GB.

- *Testing*

Proses *testing* dilakukan dengan menggunakan *webcam* berspesifikasi mempunyai resolusi video Full HD 1080p dengan 30 *frame per second* serta memiliki resolusi foto yaitu 3 *megapixel*. Dan juga menyediakan fitur autofocus. Untuk pencahayaan saat *testing*, penulis menggunakan LED *ring light* berdiameter 18 inci dengan daya sekitar 55 *watt*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 *Training dataset 7 nominal uang Rupiah*

Sebelum menjalankan sistem deteksi secara *realtime*, diperlukan adanya *training dataset* untuk memperoleh model *training*, sehingga sistem dapat mengetahui objek yang ingin dideteksi sesuai model yang telah ada.

4.1.1 *Alat dan Bahan dalam proses training*

Alat dan bahan yang digunakan dalam proses ini adalah laptop dengan spesifikasi yang telah disebut pada sub-bab 3.7. Lalu *dataset* sebagai model dalam penelitian ini menggunakan citra uang kertas rupiah dengan tahun emisi 2016.

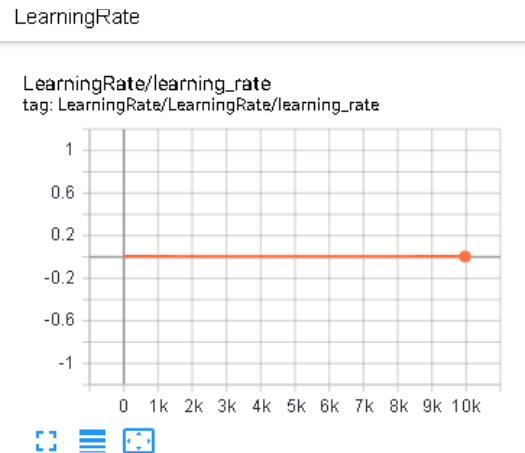
4.1.2 *Prosedur training dataset model SSD*

Beberapa langkah *training* adalah sebagai berikut:

1. Memberi *input dataset* yang telah dikumpulkan.
2. Memberikan *setting* konfigurasi berupa *training* sebesar 10.000 step.
3. Hasil dari *training* disimpan dalam bentuk format *.ckpt*

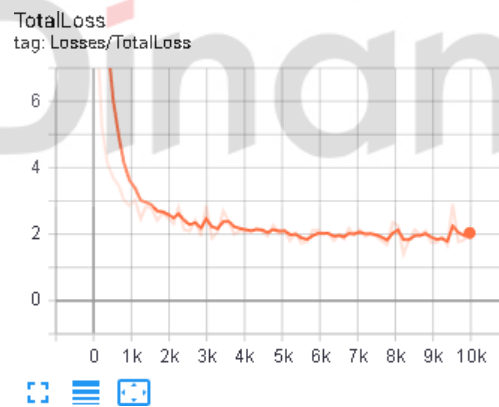
4.1.3 *Hasil training dataset model SSD*

Setelah *training* sudah berhenti berjalan, adapun grafik *learning rate* dan *loss* yang telah disimpan. Dengan menggunakan *framework Tensorboard* parameter grafik sebagaimana tampak pada Gambar 4.1 dibawah sebagai berikut.



Gambar 4.1 Grafik *Learning Rate Training*

Arti dari *Learning Rate* pada hasil *training* untuk menunjukkan adanya kendala disaat proses *training* berjalan. Pada Gambar 4.1 menampilkan bahwa garis grafik stabil lurus pada angka 0, berarti proses *training* di awal step hingga akhir step dilakukan secara optimal.



Gambar 4.2 Grafik *Loss Training*

Untuk grafik *loss* ditunjukkan pada Gambar 4.2, di grafik tersebut menampilkan bahwa adanya nilai *loss* yang besar hingga melebihi nilai 6 pada step awalnya. Pada step ke 1.000 menunjukkan bahwa nilai *loss* terjadi penurunan drastis, berarti proses *training* berjalan dengan baik. Pada saat step pertengahan hingga akhir terjadi kenaikan dan penurunan nilai angka *loss*. Hingga pada saat *training* step terakhir

nilai *loss* menunjukkan di nilai 2. Jadi semakin kecil nilai angka *loss*, maka model *dataset* yang sedang di *training* semakin bagus dan dapat digunakan sebagai model untuk digunakan pada *testing*. Jika grafik *loss* pada *training* semakin tidak ada penurunan maka hasil *training* nya tidak dapat digunakan sebagai model untuk *testing* dan juga memungkinkan bahwa saat pengambilan *dataset* pun tidak optimal dikarenakan mungkin pencahayaan kurang maksimal, *dataset* kurang banyak, lalu kualitas kamera yang digunakan untuk pengambilan citra kurang bagus.

4.2 Pengujian Akurasi dan FPS

Pada pengujian program klasifikasi nominal uang ini meliputi seberapa besar akurasi dan FPS setiap subjek dalam 1 nominal lembar uang dalam pendeteksian uang dalam jarak 20 cm, 60 cm, 100cm dengan akurasi berbentuk nilai persentase menggunakan *webcam*. Untuk pengujian akurasi dan FPS ini dilakukan dengan 7 nominal uang kertas rupiah yang meliputi: Rp. 1.000, Rp. 2.000, Rp. 5.000, Rp. 10.000, Rp. 20.000, Rp. 50.000, dan Rp. 100.000. Di setiap nominalnya dibagi 4 subjek seperti pada Gambar 3.3.

4.2.1 Tujuan Pengujian Akurasi dan FPS

Pengujian ini bertujuan untuk mengetahui hasil akurasi beserta rata-ratanya dalam mendeteksi seluruh nominal uang kertas rupiah dengan dibagi 4 subjek tiap nominalnya yaitu *Full*, Gambar Tengah, Nominal Horizontal, dan Nominal Vertikal yang dapat dilihat contohnya pada Gambar 3.3.

4.2.2 Alat dan Bahan dalam Pengujian Akurasi dan FPS

Alat dan bahan yang digunakan dalam pengujian Akurasi dan FPS ini yaitu *webcam* yang sama persis digunakan saat pengambilan *dataset*. Spesifikasi *webcam* yang digunakan memiliki resolusi *full HD* 1080p di 30 FPS untuk resolusi foto nya 3 *megapixel*, dan fitur autofokus juga. Untuk bahan pastinya yaitu 7 uang kertas rupiah dengan nominal yang berbeda-beda.

4.2.3 Prosedur Pengujian Akurasi dan FPS

Prosedur pengujian untuk Akurasi dan FPS adalah sebagai berikut:

1. *Webcam* terhubung ke komputer dahulu.
2. Program pendeteksi di jalankan.
3. 7 uang kertas rupiah diperlihatkan pada *webcam*.
4. Saat pengujian berlangsung pencahayaan pada subjek nya dianjurkan optimal.

4.2.4 Hasil Pengujian Akurasi dan FPS

Pengujian ini dilakukan secara *realtime* dengan cara memegang objek / uang dengan masing-masing jarak 20 cm, 60 cm, dan 100 cm sebanyak 10 kali percobaan.

Berikut data hasil pengujian dari akurasi dan FPS:

1. Pengujian pada jarak 20 cm :

Tabel 4.1 Hasil pengujian akurasi dan FPS jarak 20 cm

No.	Subjek	Percobaan Ke-										Rata - Rata	
		1	2	3	4	5	6	7	8	9	10		
1.	Rp. 1.000	Full	94	72	94	80	84	85	78	79	73	80	81,9
		Gambar Tengah	76	74	84	84	72	80	74	81	79	83	78,7
		Nominal Horizontal	70	83	71	89	61	87	74	83	79	82	77,9
		Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	0
		FPS	8,5	8,84	8,58	8,6	8,47	8,55	8,57	8,51	8,56	8,45	8,56
2.	Rp. 2.000	Full	68	73	78	71	78	77	79	75	78	73	75
		Gambar Tengah	90	74	80	75	86	87	80	88	83	79	82,2
		Nominal Horizontal	61	73	71	83	74	78	73	69	79	80	74,1
		Nominal Vertikal	74	70	63	75	71	76	73	68	75	77	72,2
		FPS	8,3	8,37	8,5	8,01	8,31	8,03	8,51	8,56	8,01	8	8,26
3.	Rp. 5.000	Full	74	71	79	80	72	75	79	80	76	77	76,3
		Gambar Tengah	79	84	72	77	68	69	74	82	75	79	75,9
		Nominal Horizontal	78	60	73	79	67	63	73	78	68	72	71,1
		Nominal Vertikal	80	74	71	75	76	77	78	81	79	74	76,5
		FPS	8,56	8,06	8,59	8,59	8,4	8,11	8,47	8,5	8,57	8,6	8,45
4.	Rp. 10.000	Full	78	73	76	71	79	75	73	71	79	77	75,2
		Gambar Tengah	61	72	75	80	77	79	81	76	80	78	75,9
		Nominal Horizontal	68	75	71	79	68	71	74	69	77	73	72,5

No.	Subjek	Percobaan Ke-										Rata - Rata			
		1	2	3	4	5	6	7	8	9	10				
4.	Rp. 10.000	Nominal Horizontal	0	0	0	0	0	0	0	0	0	0	0		
		Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	0		
		FPS	10	10,1	10	10,02	10,1	10	10	10	10	10	10,01		
		Full	75	71	79	78	75	80	73	82	81	79	77,3		
		Gambar Tengah	0	0	0	0	0	0	0	0	0	0	0		
		Nominal Horizontal	66	63	61	64	68	60	63	66	62	70	64,3		
		Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	0		
		FPS	9,91	10	10	10	10	9,99	10	10	10	10	9,99		
		Full	88	81	86	83	80	83	81	89	84	88	84,3		
		Gambar Tengah	65	63	68	0	61	72	69	0	62	68	52,8		
		Nominal Horizontal	71	77	0	73	0	74	72	79	0	77	52,3		
		5.	Rp. 20.000	Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	
FPS	10			10	10	10	10	9,98	10	10	10,1	9,98	10,01		
Full	96			91	97	90	89	95	98	93	97	99	94,5		
Gambar Tengah	90			88	81	83	85	92	87	89	94	88	87,7		
Nominal Horizontal	68			69	61	66	64	68	63	70	69	68	66,6		
Nominal Vertikal	63			68	65	66	62	63	69	63	66	67	65,2		
FPS	10			9,95	10	9,98	10	9,99	10	9,92	10	10	9,99		
Full	93			95	90	91	87	98	94	89	99	92	92,8		
Gambar Tengah	0			0	0	0	0	0	0	0	0	0	0		
Nominal Horizontal	65			0	67	0	64	65	62	0	0	61	38,4		
Nominal Vertikal	0			0	0	0	0	0	0	0	0	0	0		
FPS	10			9,98	9,98	10	10	10	10	10,1	10	10	10,01		
6.	Rp. 50.000	Nominal Horizontal	68	69	61	66	64	68	63	70	69	68	66,6		
		Nominal Vertikal	63	68	65	66	62	63	69	63	66	67	65,2		
		FPS	10	9,95	10	9,98	10	9,99	10	9,92	10	10	9,99		
		Full	93	95	90	91	87	98	94	89	99	92	92,8		
		Gambar Tengah	0	0	0	0	0	0	0	0	0	0	0		
		Nominal Horizontal	65	0	67	0	64	65	62	0	0	61	38,4		
		Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	0		
		FPS	10	9,98	9,98	10	10	10	10	10,1	10	10	10,01		
		7.	Rp. 100.000	Nominal Horizontal	65	0	67	0	64	65	62	0	0	61	38,4
				Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	0
				FPS	10	9,98	9,98	10	10	10	10	10,1	10	10	10,01

3. Pengujian pada jarak 100 cm :

Tabel 4.3 Hasil pengujian akurasi dan FPS jarak 100 cm

No.	Subjek	Percobaan Ke-										Rata - Rata	
		1	2	3	4	5	6	7	8	9	10		
1.	Rp. 1.000	Full	0	0	0	0	0	0	0	0	0	0	0
		Gambar Tengah	0	0	0	0	0	0	0	0	0	0	0
		Nominal Horizontal	0	0	0	0	0	0	0	0	0	0	0
		Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	0
		FPS	11,5	11,6	11,3	11,5	11,6	11,7	11,6	11,8	11,5	11,7	11,58

No.	Subjek	Percobaan Ke-										Rata - Rata	
		1	2	3	4	5	6	7	8	9	10		
2.	Rp. 2.000	Full	0	0	0	0	0	0	0	0	0	0	0
		Gambar Tengah	0	0	0	0	0	0	0	0	0	0	0
		Nominal Horizontal	0	0	0	0	0	0	0	0	0	0	0
		Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	0
		FPS	11,8	11,7	11,6	11,8	11,7	11,6	11,8	11,5	11,5	11,6	11,67
3.	Rp. 5.000	Full	0	0	0	0	0	0	0	0	0	0	
		Gambar Tengah	0	0	0	0	0	0	0	0	0	0	
		Nominal Horizontal	0	0	0	0	0	0	0	0	0	0	
		Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	
		FPS	11,5	11,5	11,3	11,6	11,5	11,7	11,5	11,8	11,3	11,3	11,49
4.	Rp. 10.000	Full	0	0	0	0	0	0	0	0	0	0	
		Gambar Tengah	0	0	0	0	0	0	0	0	0	0	
		Nominal Horizontal	0	0	0	0	0	0	0	0	0	0	
		Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	
		FPS	11,6	11,7	11,5	11,6	11,7	11,6	11,6	11,3	11,8	11,5	11,59
5.	Rp. 20.000	Full	0	0	0	0	0	0	0	0	0	0	
		Gambar Tengah	0	0	0	0	0	0	0	0	0	0	
		Nominal Horizontal	0	0	0	0	0	0	0	0	0	0	
		Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	
		FPS	11,7	11,2	11,4	11,5	11,5	11,3	11,4	11,4	11,7	11,7	11,48
6.	Rp. 50.000	Full	0	0	0	0	0	0	0	0	0	0	
		Gambar Tengah	79	77	75	79	71	74	75	81	77	75	76,3
		Nominal Horizontal	0	0	0	0	0	0	0	0	0	0	0
		Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	0
		FPS	8,59	8,58	8,57	8,57	8,56	8,54	8,58	8,6	8,56	8,59	8,57
7.	Rp. 100.000	Full	0	0	0	0	0	0	0	0	0	0	
		Gambar Tengah	0	0	0	0	0	0	0	0	0	0	
		Nominal Horizontal	0	0	0	0	0	0	0	0	0	0	
		Nominal Vertikal	0	0	0	0	0	0	0	0	0	0	
		FPS	11,6	11,7	11,7	11,6	11,9	11,5	11,4	11,6	11,6	11,4	11,59

4.2.5 Analisis Hasil Pengujian Akurasi dan FPS

Untuk mendapatkan nilai rata-rata akurasi dari setiap subjek yaitu dengan cara menambahkan semua akurasi di setiap percobaan lalu dibagi banyaknya percobaan.

$$\text{Rata - rata akurasi subjek} = \frac{\text{Total akurasi setiap percobaan}}{\text{banyak nya percobaan}} \quad (1)$$

Lalu untuk rata-rata FPS didapatkan dengan sebagai berikut:

$$\text{Rata - rata FPS} = \frac{\text{total FPS tiap percobaan}}{\text{banyak nya percobaan}} \quad (2)$$

Pada Tabel 4.1 menunjukkan bahwa di jarak 20 cm semua subjek dapat dideteksi dan mendapatkan akurasi yang baik, namun ada 1 subjek yang tidak terbaca yaitu pada uang nominal Rp. 1.000 bagian subjek Nominal Vertikal. Ini dikarenakan saat pengambilan *dataset* untuk uang kertas Rp. 1.000 pencahayaan kurang optimal. Tetapi selain subjek tersebut semua subjek dapat terbaca dengan baik. Untuk rata-rata akurasi terbaik di semua subjek ada di uang nominal Rp. 50.000 dengan rata-rata akurasi setiap subjeknya sebesar 80% keatas. Diikuti uang nominal Rp. 20.000 dengan rata-rata akurasi setiap subjeknya berada di 77% keatas. Dan pada subjeknya yang “Nominal Vertikal” rata-rata akurasinya menyentuh hingga 92%. Di Tabel 4.1 untuk FPS yang didapat pada jarak 20 cm di setiap percobaan mendapatkan nilai sebesar 8 FPS keatas. Pada nominal uang Rp. 50.000 nilai FPS sempat menunjukkan angka 7 FPS di percobaan 3 sampai 5. Dikarenakan saat pengujian muncul banyak *bounding box* yang terdeteksi pada layar. Rata-rata tertinggi FPS berada di uang nominal Rp. 1.000, Rp. 10.000, dan Rp. 100.000 sebesar 8.56 FPS.

Menurut Tabel 4.2 untuk jarak 60 cm menunjukkan bahwa beberapa subjek pada tiap nominalnya mulai tidak dapat dikenali tetapi banyak juga yang dapat dideteksi. Di jarak ini, Subjek “Full” dapat dikenali di semua uang tapi untuk variabel lain ada yang tidak dapat dideteksi atau memiliki nilai akurasi yang tidak bagus hingga bernilai 0. Sebagai contoh untuk uang Rp. 50.000 di setiap subjeknya dapat dikenali, di subjek “Full” nya pun mendapatkan rata-rata akurasi yang sangat bagus sebesar 94.5% begitupun juga subjek “Gambar Tengah” mendapatkan rata-rata akurasi sebesar 87.7% namun mendapatkan rata-rata akurasi yang kurang optimal sebesar 66.6% untuk “Nominal Horizontal” dan 65.2% untuk “Nominal Vertikal”. Dan juga pada uang nominal Rp. 100.000 di subjek “Full” rata-rata akurasi nya bahkan mendapatkan nilai sebesar 92.8%. Tetapi untuk subjeknya yang “Nominal Horizontal” hanya mendapatkan rata-rata akurasi 38.4% dikarenakan sempat tidak dikenali pada beberapa percobaan, lalu untuk subjek “Gambar Tengah” dan “Nominal Vertikal” tidak dapat

dikenali hingga mendapatkan nilai rata-rata akurasi 0%. Untuk uang kertas yang lain pun juga mendapatkan rata-rata nilai akurasi yang kurang baik terutama di subjek “Gambar Tengah” dan “Nominal Vertikal” kecuali di uang nominal uang Rp. 50.000. Untuk FPS di jarak 60 cm ini didapatkan rata-rata sebesar 9 hingga 10 FPS, karena di jarak ini *bounding box* di setiap variabel terkadang tidak terdeteksi maka FPS jauh lebih lancar jika dibandingkan di jarak 20 cm.

Dari Tabel 4.3 di jarak 100 cm ini hampir semua uang kertas tidak dapat dikenali, tetapi ada satu subjek yang dapat dikenali walaupun akurasi nya yang didapat tidak seberapa baik. Di uang nominal Rp. 50.000 pada subjek “Gambar Tengah” di jarak ini masih dapat dikenali, dengan nilai rata-rata akurasi hanya sebesar 76.3%. Di subjek “Full”, “Nominal Horizontal”, maupun “Nominal Vertikal” pada subjek Rp. 50.000 di jarak ini tidak dapat dideteksi. Namun untuk FPS di jarak ini mendapatkan angka yang besar yaitu mencapai 11 FPS kecuali pada uang nominal Rp. 50.000. Penyebab FPS di jarak ini lebih besar jika dibandingkan jarak-jarak yang sebelumnya adalah karena tidak munculnya *bounding box* pada layar maka FPS dapat berjalan dengan lancar. Karena itu pada uang nominal Rp. 50.000 rata-rata FPS nya sebesar 8.57 FPS, sebab adanya *bounding box* yang muncul pada layar.

Dari ketiga tabel tersebut dapat disimpulkan bahwa di jarak 20 cm rata-rata akurasi pada setiap subjek lumayan bagus dan rata-rata FPS sebesar lebih dari 8 FPS. Di jarak 60 cm rata-rata akurasi pada setiap subjek ada yang cukup bagus dan ada yang tidak terdeteksi / akurasi, untuk rata-rata FPS nya yaitu 10 FPS. Lalu pada jarak 100 cm mempunyai rata-rata FPS sebesar lebih dari 11 keatas, kecuali 1 uang kertas. Untuk rata-rata akurasi yang didapat pada jarak 100 cm sangat buruk, dikarenakan jarak ini adalah jarak terjauh yang bisa dideteksi oleh *webcam* walaupun jarak terjauh yang dapat dideteksi tetapi masih ada 1 subjek yang dapat dideteksi oleh *webcam*.

4.3 Pengujian Performa

Pengujian performa klasifikasi nominal uang ini untuk mengetahui seberapa performa program dalam mendeteksi uang. Dengan menggunakan jarak sebagai variabel, uang kertas dideteksi melewati *webcam* secara *realtime*.

4.3.1 Tujuan Pengujian Performa

Pengujian performa ini dilakukan untuk mengetahui benar atau tidaknya program dalam mendeteksi secara keseluruhan lembar uang kertas yang ditampilkan lewat *webcam* secara *realtime*. Pengujian ini dilakukan sebanyak 10 percobaan.

4.3.2 Alat dan Bahan dalam Pengujian Performa

Alat dan bahan yang digunakan dalam pengujian Akurasi dan FPS ini yaitu webcam yang sama persis digunakan saat pengambilan dataset. Spesifikasi webcam yang digunakan memiliki resolusi full HD 1080p di 30 FPS untuk resolusi foto nya 3 megapixel, dan fitur autofocus juga. Untuk bahan pastinya yaitu 7 uang kertas rupiah dengan nominal yang berbeda-beda.

4.3.3 Prosedur Pengujian Performa

Prosedur pengujian untuk Akurasi dan FPS adalah sebagai berikut:

1. Webcam terhubung ke komputer dahulu.
2. Program pendeteksian dijalankan.
3. 7 uang kertas rupiah diperlihatkan pada webcam.
4. Saat pengujian berlangsung pencahayaan pada subjek nya dianjurkan optimal.

4.3.4 Hasil Pengujian Performa

Berikut hasil data yang didapat saat pengujian performa yang dilakukan sebanyak 10 percobaan.

1. Pengujian pada jarak 20 cm :

Tabel 4.4 Hasil pengujian performa jarak 20 cm

No.	Subjek	Performa	Percobaan Ke-										Jumlah	Akurasi (%)
			1	2	3	4	5	6	7	8	9	10		
1.	Rp. 1.000	Benar		✓	✓		✓	✓	✓			✓	6	60
		Salah	✓			✓				✓	✓		4	
2.	Rp. 2.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100
		Salah											0	
3.	Rp. 5.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100
		Salah											0	
4.	Rp. 10.000	Benar	✓									✓	2	20
		Salah		✓	✓	✓	✓	✓	✓	✓	✓		8	
5.	Rp. 20.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100
		Salah											0	
6.	Rp. 50.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100
		Salah											0	
7.	Rp. 100.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100
		Salah											0	

2. Pengujian pada jarak 60 cm :

Tabel 4.5 Hasil pengujian performa jarak 60 cm

No	Subjek	Performa	Percobaan Ke-										Jumlah	Akurasi (%)
			1	2	3	4	5	6	7	8	9	10		
1.	Rp. 1.000	Benar											0	0
		Salah	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	
2.	Rp. 2.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100
		Salah											0	
3.	Rp. 5.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100
		Salah											0	
4.	Rp. 10.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100
		Salah											0	
5.	Rp. 20.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100
		Salah											0	
6.	Rp. 50.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100
		Salah											0	
7.	Rp. 100.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100
		Salah											0	

3. Pengujian pada jarak 100 cm :

Tabel 4.6 Hasil pengujian performa jarak 100 cm

No	Subjek	Performa	Percobaan Ke-										Jumlah	Akurasi (%)
			1	2	3	4	5	6	7	8	9	10		
1.	Rp. 1.000	Benar											0	0
		Salah	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	
2.	Rp. 2.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100	
		Salah										0		
3.	Rp. 5.000	Benar										0	0	
		Salah	✓	✓	✓	✓	✓	✓	✓	✓	✓	10		
4.	Rp. 10.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100	
		Salah										0		
5.	Rp. 20.000	Benar										0	0	
		Salah	✓	✓	✓	✓	✓	✓	✓	✓	✓	10		
6.	Rp. 50.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100	
		Salah										0		
7.	Rp. 100.000	Benar	✓	✓	✓	✓	✓	✓	✓	✓	✓	10	100	
		Salah										0		

4.3.5 Analisis Hasil Pengujian Performa

Untuk mendapatkan nilai performa akurasi dari setiap subjek yaitu dengan cara menambahkan semua hasil benar di setiap percobaan lalu dibagi banyaknya percobaan lalu dikali 100. Nilai 100 adalah dari 100%.

$$\text{Performa akurasi (\%)} = \frac{\text{Total percobaan seluruh uang} - \text{Hasil salah}}{\text{Total percobaan seluruh uang}} \times 100$$

(3)

Menurut Tabel 4.4 pada pengujian di jarak 20 cm program mendeteksi beberapasubjek dengan benar sesuai apa yang diperlihatkan di *webcam*. Hanya ada 2 subjek yang performa akurasi nya tidak 100% karena di beberapa percobaan mengeluarkan output salah yaitu di subjek Rp. 1.000 dan Rp. 10.000. Saat pengambilan data secara *realtime* pada subjek Rp. 1.000 di percobaan ke-1, 4, 8, dan 9 layar menunjukkan output an berbeda dari yang ditampilkan di *webcam*. Output yang dikeluarkan seharusnya “seribu_rupiah” tetapi program mengeluarkan output yang lain. Dan begitu pula pada subjek Rp. 10.000 hanya 2 percobaan yang benar dalam mendeteksi uang tersebut yaitu di percobaan awal dan akhir, selain percobaan itu program mengeluarkan output yang tidak sesuai yang ditampilkan di *webcam*.

Untuk subjek Rp.2.000, Rp5.000, Rp. 20.000, Rp. 50.000, dan Rp. 100.000 pada setiap percobaan menunjukkan tidak adanya kesalahan mengenali subjek dan mengeluarkan output yang sesuai yang ditunjukkan di *webcam*.

Pada Tabel 4.5 menunjukkan pengujian di jarak 60 cm pun hasilnya hampir mendekati benar semua dalam mengenali subjek dan mengeluarkan output sesuai yang diperlihatkan pada *webcam*. Namun ada 1 subjek yang tidak dapat dikenali pada pengujian di jarak ini yaitu di subjek Rp. 1.000. Saat pengambilan data uji berlangsung, program tidak mengeluarkan output apa yang sesuai ditunjukkan di *webcam* dan terus mengeluarkan output lain selain “seribu_rupiah”. Program tak dapat mengenali dengan benar subjek Rp. 1.000 hingga awal sampai akhir percobaan saat pengujian. Penyebab program tidak dapat mengenali subjek ini karena saat pengujian berlangsung penerangan kurang optimal hingga subjek tidak dapat dikenali oleh program. Untuk subjek selain Rp. 1.000 tidak menunjukkan adanya kesalahan atau *error* satupun dalam mengenali subjek dan mengeluarkan output yang sesuai apa yang ditampilkan pada *webcam* di setiap percobaannya.

Lalu ditunjukkan pada Tabel 4.6 bahwa pengujian di jarak 100 cm ini mulai menunjukkan adanya kesalahan dalam mengenali beberapa subjek, tetapi ada pula juga program dapat mengenali subjek dengan benar dan mengeluarkan output yang benar sesuai diperlihatkan pada *webcam*. Pada subjek Rp. 1.000, Rp. 5.000, dan Rp. 20.000 menunjukkan adanya kesalahan dalam mengenali subjek. Ini dikarenakan jarak 100 cm adalah jarak maksimum program untuk mengenali atau mendeteksi sebuah subjek dan juga faktor dari kualitas *webcam* tersebut. Meskipun begitu program masih dapat mendeteksi dan mengenali dengan benar 4 subjek saat pengujian.

Pada subjek Rp. 2.000, Rp. 10.000, Rp. 50.000, dan Rp. 100.000 yang dapat dikenali dan mengeluarkan output yang sesuai di layar saat subjek ditunjukkan pada *webcam*. Di saat pengujian yang konsisten benar dalam mengenali subjek yaitu uang dengan nominal Rp.50.000. Dikarenakan warna dari subjek Rp.50.000 sangat jelas dan mencolok, maka program dapat mengenali subjek tersebut meskipun di jarak 100 cm. Untuk subjek selain Rp. 50.000, pada saat pengujian ada sempat terlintas mengeluarkan output yang tidak sesuai ditunjukkan pada *webcam* lalu setelah itu

program dapat mengenali dengan benar dan sesuai apa yang diperlihatkan pada *webcam*. Sebab untuk subjek selain Rp. 50.000 kurang optimal dan adanya blur lalu sumber cahayanya kurang menerangi subjek dengan jelas saat pengambilan *dataset*.

Berdasarkan ketiga tabel pengujian performa diatas menunjukkan bahwa pada saat pengujian di jarak 20 cm mendapatkan hasil yang bagus dengan nilai performa akurasi 100%. Dan meskipun di subjek Rp. 1.000 dan Rp. 10.000 mendapatkan performa akurasi yang kurang baik masing-masing yang bernilai 60% dan 20%, tetapi setidaknya program sempat mengenali subjek-subjek tersebut dengan benar pada beberapa percobaan. Lalu di jarak 60 cm menunjukkan hasil performa yang bagus semua kecuali pada subjek Rp. 1.000. Di saat pengujian, subjek Rp. 1.000 di setiap percobaan selalu terus mengeluarkan output yang seharusnya “seribu_rupiah” menjadi output an subjek yang lain. Pada jarak terjauh yaitu 100 cm, program banyak kesalahandalam mengenali di berbagai subjek. Dikarenakan jarak ini memang jarak terjauh yang dapat digapai oleh program dalam mendeteksi dan mengenali uang kertas. Adapun 4 dari 7 subjek yang mendapatkan hasil performa yang bagus. Untuk subjek selain itu program tidak dapat mengenali dengan benar sebab saat pengambilan *dataset* kebanyakan ada yang blur dan pencahayaan kurang jelas di beberapa subjek.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisa pengujian sistem klasifikasi nominal uang pada Bab 4 tersebut, maka dapat disimpulkan sebagai berikut:

1. Klasifikasi uang kertas rupiah berdasarkan angka nominal secara *realtime* menggunakan metode SSD dilakukan dengan jumlah data 1.400 citra, yang dibagi menjadi dua *dataset*, yaitu 1.120 data train dan 280 data test.
2. Sistem hanya dapat mengklasifikasikan uang kertas tahun emisi 2016 yang dibedakan menjadi 7 *class*, yaitu Rp. 1.000, Rp. 2.000, Rp. 5.000, Rp. 10.000, Rp. 20.000, Rp. 50.000, Rp. 100.000.
3. Dari hasil *training* dari model SSD menghasilkan nilai *learning rate* di angka 0 dan untuk nilai total *loss* di angka 2 dengan menggunakan 10.000 *step*. Untuk total *loss* semakin kecil nilainya maka sistem yang mempunyai akurasi yang lebih baik.
4. Hasil pengujian akurasi dibagi 4 subjek di tiap uang kertas yaitu “Full”, “Gambar Tengah”, “Nominal Horizontal”, dan “Nominal Vertikal”. Pada penelitian ini untuk dijarak 20 cm didapatkan hasil lumayan bagus karena seluruh subjek dalam uang kertas dapat dikenali oleh sistem dan rata-rata akurasi subjek yang tertinggi didapatkan nilainya sebesar 93.5%. Pada jarak 60 cm belum bisa dibilang baik dikarenakan adanya kesalahan atau sistem tidak dapat mengenali beberapa subjek dan hanya uang kertas Rp. 50.000 yang memiliki rata-rata akurasi tiap subjeknya bernilai diatas 65% tetapi program tetap dapat mengenali dan mendeteksi dengan benar untuk subjek “Full” pada semua uang kertas. Pada jarak 100 cm sistem hanya mengenali 1 uang kertas yaitu Rp.50.000 di subjek “Gambar Tengah” dengan nilai rata-rata akurasi sebesar 76.3%.
5. Hasil pengujian performa akurasi didapatkan dari total hasil subjek yang benar pada setiap percobaan lalu dibagi banyaknya percobaan setelah itu dibagi 100. Untuk jarak 20 cm didapatkan hasil performa akurasi sebesar 100% pada semua subjek kecuali Rp. 1.000 dan Rp. 10.000 dengan nilai masing-masing 60% dan

20%. Lalu jarak 60 cm hasil yang didapat hampir bagus dikarenakan hanya 1 subjek yang menunjukkan performa akurasi 0% yaitu Rp. 1.000. Dan pada jarak 100 cm hasil pengujian cukup buruk karena performa akurasi yang didapat 0% di 3 subjek, selain ketiga subjek tersebut hasil performa akurasi yang didapat sebesar 100%.

6. Hasil pengujian FPS didapatkan dari jumlah FPS di semua percobaan lalu dibagi dengan banyaknya percobaan. Tinggi rendahnya hasil nilai FPS tergantung dari munculnya *bounding box* di layar. Semakin banyak *bounding box* muncul di layar, semakin rendah nilai FPS yang didapat. Di jarak 20 cm mendapatkan rata-rata FPS tertinggi di angka 8.56 FPS, lalu di jarak 60 rata-rata FPS tertinggi di angka 10 FPS, dan pada jarak 100 cm didapatkan nilai rata-rata tertinggi 11 FPS terkecuali uang nominal Rp. 50.000. Rata-rata *frame per second* nya sebesar 8.57 FPS.

5.2 Saran

Adapun saran untuk pengembangan pada penelitian ini agar lebih baik terdapat beberapa saran berikut:

1. Untuk penelitian selanjutnya bisa menggunakan sinar UV agar dapat mengidentifikasi uang yang digunakan asli atau palsu. Dan menggunakan uang dengan tahun emisi baru.
2. Untuk meningkatkan hasil *training dataset* yang optimal dan mendapatkan total *loss* yang rendah, dapat menambah jumlah citra dan memvariasi citra untuk digunakan *dataset*, menggunakan kamera dengan resolusi yang baik, dengan menggunakan pencahayaan yang maksimal saat pengambilan *dataset*.
3. Menggunakan model *deep learning* selain SSD untuk menghasilkan hasil yang maksimal seperti YOLO (*You Only Look Once*), *Faster RCNN* atau LSTM.
4. Menggunakan *device* yang lebih mumpuni agar proses *training* dapat berjalan lancar, disarankan menggunakan *graphic card* NVIDIA terbaru karena membutuhkan versi CUDA dan CUDNN yang terbaru juga.

DAFTAR PUSTAKA

- Alfi, S. (2020). *Object Detection (Case: Plat Detection) | by Alfi Salim | BISA.AI | Medium*. <https://medium.com/bisa-ai/object-detection-case-plat-detection-7cb5f53682ae>
- Hidayahtullah, A. M. (2022). *SISTEM DETEKSI SIMBOL PADA SIBI (SISTEM ISYARAT BAHASA INDONESIA) SECARA REALTIME MENGGUNAKAN MOBILENET-SSD*. Universitas Dinamika.
- Izah, R. N. (2018). *KLASIFIKASI NOMINAL UANG KERTAS RUPIAH TAHUN EMISI 2017 DENGAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK MENGGUNAKAN MXNET*. Universitas Islam Indonesia, Yogyakarta.
- Kurnia, S. (2013). PEMROSESAN PARALEL PADA NVIDIA CUDA [Universitas Kristen Satya Wicana]. In *Perpustakaan Universitas Kristen Satya Wicana*. https://repository.uksw.edu/bitstream/123456789/3854/1/T1_622007006_Fulltext.pdf
- Muhammad. (2021). Mengajarkan Bahasa Pemrograman Python Di Tingkat SMA Untuk Meningkatkan Manajemen Sumber Daya Manusia. *Jurnal Riset Akuntansi Dan Manajemen Malahayati*, 10(1), 8. <https://doi.org/10.33024/jrm.v10i1.4624>
- Priadana, A., & Murdiyanto, A. W. (2019). Metode SURF dan FLANN untuk Identifikasi Nominal Uang Kertas Rupiah Tahun Emisi 2016 pada Variasi Rotasi. *Jurnal Teknologi Dan Sistem Komputer*, 7(1), 19–24. <https://doi.org/10.14710/jtsiskom.7.1.2019.19-24>
- Putri, V. K. M. (2022). Jenis Uang Kartal dan Giral. In *Kompas.com*. <https://www.kompas.com/skola/read/2022/05/17/090000369/jenis-uang-kartal-dan-giral?page=all>
- Setiawan, R. (2021). Mengenal Deep Learning Lebih Jelas - Dicoding Blog. In *Dicoding Blog*. <https://www.dicoding.com/blog/mengenal-deep-learning/>
- Sidharta, H. A. (2017). Introduction to Open CV | BINUS UNIVERSITY MALANG | Pilihan Universitas Terbaik di Malang. In *Binus.Ac.Id*. <https://binus.ac.id/malang/2017/10/introduction-to-open-cv/>
- Warianto, W. (2017). *Tinjauan Yuridis Penggunaan Mata Uang Asing Sebagai Alat Pembayaran Dalam Transaksi di Wilayah Kedaulatan Negara Kesatuan Republik Indonesia (NKRI)*. Universitas Internasional Batam.
- Zhang, Q. (2018). *Currency Recognition Using Deep Learning*. Auckland University of Technology.