

# DETEKSI OTOMATIS NOMINAL UANG KERTAS RUPIAH UNTUK TUNANETRA MENGGUNAKAN ALGORITMA ARSITEKTUR SSD MOBILENETV3

Ario Prima<sup>1\*</sup>, Dian Budhi Santoso<sup>1</sup>, Lela Nurpulaela<sup>1</sup>

<sup>1</sup>Department of Electrical Engineering, Universitas Singaperbangsa Karawang, Indonesia

## ARTICLE INFO

### History of the article:

Received July 13, 2023

Revised July 28, 2023

Accepted August 13, 2023

Published August 15, 2023

### Keywords:

artificial intelligence

convolutional neural network

deep learning

mobilenetv3

single shot multibox detector

## ABSTRACT

Money is a legal tender in the buying and selling transactions of society. However, visually impaired people face visual limitations that affect their ability to recognize the value of Rupiah banknotes. Although Rupiah banknotes already have a blind code feature, this is often ineffective due to the condition of the banknotes. In this study, we use Artificial Intelligence technology, especially deep learning, to help visually impaired people recognize the value of Rupiah banknotes more easily. Our system is built using Convolutional Neural Network (CNN) technique with MobileNetV3 architecture and Single Shot Multibox Detector (SSD) algorithm. The results show that our system is able to operate well under various lighting conditions, including daytime and nighttime. Under sufficient lighting conditions, the system achieves an accuracy of between 80% to 95%. However, we acknowledge that in low-light or nighttime conditions, our system has problems detecting banknotes. Thus, this research contributes to the effort of improving the accessibility of visually impaired people in recognizing Rupiah banknotes, although we also recognize the need for more attention to address low lighting conditions in future development.

*This is an open access article under the CC BY-ND license.*



## Kata Kunci :

artificial intelligence

convolutional neural network

deep learning

mobilenetv3

single shot multibox detector

## ABSTRAK

Uang merupakan alat pembayaran sah dalam transaksi jual beli masyarakat. Namun, para penyandang tunanetra menghadapi keterbatasan penglihatan yang mempengaruhi kemampuan mereka dalam mengenali nilai uang kertas Rupiah. Meskipun uang kertas Rupiah sudah memiliki fitur blind code, hal ini sering tidak efektif karena kondisi uang yang tidak selalu terjaga dengan baik. Dalam penelitian ini, kami menggunakan teknologi Artificial Intelligence, terutama deep learning, untuk membantu penyandang tunanetra mengenali nominal uang kertas Rupiah dengan lebih mudah. Sistem kami dibangun dengan menggunakan teknik Convolutional Neural Network (CNN) dengan arsitektur MobileNetV3 dan algoritma Single Shot Multibox Detector (SSD). Hasil penelitian menunjukkan bahwa sistem yang kami kembangkan mampu beroperasi secara baik dalam berbagai kondisi pencahayaan, termasuk siang dan malam hari. Pada kondisi pencahayaan memadai, sistem mencapai akurasi antara 80% hingga 95%. Namun, kami mengakui bahwa pada kondisi pencahayaan redup atau malam hari, sistem kami mengalami kendala dalam mendeteksi nominal uang. Dengan demikian, penelitian ini berkontribusi pada usaha meningkatkan aksesibilitas tunanetra dalam mengenali nominal uang kertas Rupiah, walaupun kami juga menyadari perlunya perhatian lebih untuk mengatasi kondisi pencahayaan rendah dalam pengembangan selanjutnya.

**Correspondece:**

Ario Prima,  
Department of Electrical Engineering,  
Universitas Singaperbangsa Karawang,  
Indonesia  
Email :  
ario.prima19051@student.unsika.ac.id

**PENDAHULUAN**

Seiring perkembangan dan kemajuan teknologi akhir-akhir ini telah memberikan dampak yang cukup signifikan terhadap berbagai aspek kehidupan manusia. Hal ini termasuk ke dalam bidang *Artificial Intelligence* (kecerdasan buatan), yang telah mengubah cara kerja, pola interaksi, serta gaya hidup masyarakat secara kesem eluruhan. Penggunaan teknologi berbasis kecerdasan buatan ini tidak hanya memberikan dampak positif bagi kebanyakan orang, tetapi juga memiliki potensi untuk memberikan manfaat yang signifikan bagi para penyandang tunanetra.

Uang adalah alat tukar jual beli yang sah untuk melakukan transaksi jual beli barang atau jasa. Namun, dalam hal ini menjadi terbatas bagi para penyandang tunanetra untuk melakukan transaksi jual beli barang atau jasa, karena para penyandang tunanetra memiliki keterbatasan akan kemampuan penglihatannya, sehingga sulit untuk dapat mengenali nominal uang, terutama pada uang kertas. Meskipun uang rupiah saat ini sudah tersedia fitur *blind code* (kode untuk tunanetra) yang berupa garis timbul pada sisi samping uang kertas [1], akan tetapi aksesibilitas ini akan terasa kurang efektif, hal ini karena nominal uang tidak selalu baru yang beredar di masyarakat.

Berbagai penelitian mengenai sistem pendeteksi nominal uang kertas rupiah sebelumnya sudah pernah dilakukan. Salah satunya yaitu Implementasi Deep Learning menggunakan Metode Convolutional Neural Network dan Algoritma Yolo dalam sistem pendeteksi nominal uang kertas rupiah [2]. Pada penelitian ini berfokus pada Metode CNN sebagai *feature extractor* dan Yolo sebagai algoritma dalam mendeteksi objek. Pada Yolo hanya ada 1 ukuran feature map yang menggunakan *feature map 7x7*, sehingga per sel pada *grid* harus bisa mendeteksi dari yang kecil hingga yang besar, karena memiliki nilai *range* prediksi yang besar sehingga susah untuk dioptimasi[3]. Berbeda dengan SSD yang digunakan dalam penelitian ini SSD memiliki berbagai macam ukuran *feature* yang berbeda-beda yang mana *feature* yang lebih besar untuk objek kecil dan makin kecil *feature* makin besar objeknya, sehingga proses hanya sekali saja sehingga menjadi cepat[4].

Salah satu metode yang digunakan dalam membuat sistem berbasis *artificial intelligence* ini adalah dengan menggunakan metode *Deep Learning*. Kelebihan *deep learning* dibandingkan dengan *machine learning*, karena pada *deep learning* menggunakan jaringan syaraf yang kompleks yang terdiri dari beberapa lapisan untuk menganalisis pola dan hubungan yang rumit [5]. Berbeda dengan *machine learning* yang memiliki jumlah lapisan yang sedikit sehingga tidak cocok untuk melakukan latihan model dengan gambar yang rumit dan unik [6].

*Deep learning* saat telah membawa terobosan dalam berbagai bidang *artificial intelligence*, hal ini karena *deep learning* dapat menentukan struktur yang begitu rumit di dalam sebuah dataset yang sangat besar dengan menggunakan algoritma *backpropagation*. *Backpropagation* memiliki tujuan untuk memodifikasi bobot untuk melatih jaringan *neural* untuk memetakan *input arbiter* ke *output* yang benar [7].

*Artificial Intelligence* (AI) atau biasa dikenal dengan istilah kecerdasan buatan adalah suatu metode yang berusaha untuk menanamkan kecerdasan buatan layaknya manusia kepada mesin [8]. Melakukan analisis penalaran untuk mengambil suatu kesimpulan atau keputusan penerjemahan dari satu bahasa ke bahasa lain [9]. Teknologi *deep learning* ini adalah salah satu teknologi yang paling populer untuk mengenali suatu keinginan atau objek yang memiliki tingkat keakuratan lebih tinggi dibandingkan dengan metode mesin sebelumnya [10].

*Convolutional Neural Network* (CNN) merupakan salah satu jenis algoritma *deep learning* yang memiliki kemampuan dalam menerima *input* berupa gambar atau citra yang mana hal ini bertujuan agar mesin dapat untuk belajar dan mengenali gambar sehingga dapat membedakan suatu gambar dengan gambar lainnya [11].

*MobileNetv3* merupakan suatu arsitektur jaringan neural yang dikembangkan khusus untuk penggunaan pada perangkat mobile atau perangkat dengan sumber daya terbatas. Arsitektur ini dirancang dengan tujuan untuk mengoptimalkan kinerja model dan efisiensi komputasi sehingga dapat dijalankan dengan

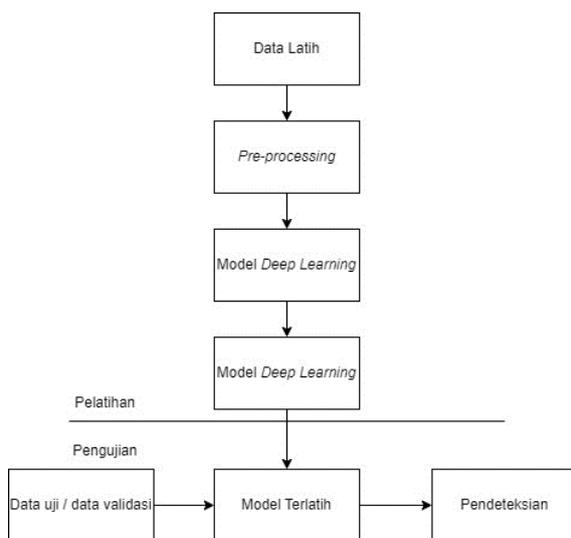
cepat dan hemat energi pada perangkat dengan daya terbatas. *Mobilenetv3* terdiri dari dua model, yaitu *Mobilenetv3 small* dan *Mobilenetv3 large*[12].

SSD adalah pendeteksi yang cukup cepat dan dapat digunakan pada real-time video[13]. *Single Shot Multibox Detector* (SSD) termasuk ke dalam kerangka kerja satu langkah sama seperti YOLO atau *RetinaNet*[14]. *Mobiledets* adalah sebuah model *object detection* untuk perangkat *mobile* yang mana memiliki kelebihan pada latensi akan tetapi menurunkan sedikit nilai akurasi. Adapun model yang menggunakannya seperti pada CPU, Edge TPU, dan Edge GPU. Pada penelitian [15].

Berdasarkan uraian permasalahan maka diperlukanlah sebuah sistem yang dapat membantu para penyandang tunanetra agar dapat mengetahui nominal uang rupiah sehingga para penyandang tunanetra dapat dengan mudah untuk mengenali nominal uang rupiah dalam transaksi jual beli barang atau jasa. Pada penelitian ini, sistem ini dibuat dengan algoritma *Convolutional Neural Network* (CNN) menggunakan arsitektur *MobileNetv3* dan deteksi objek dengan algoritma *Single Shot Multibox Detector* (SSD) dengan menggunakan Google Collaboratory atau yang biasa dikenal dengan sebutan Google Collab sebagai *virtual machine*, untuk pelatihan modelnya menggunakan *library* TensorFlow.

**METODE PENELITIAN**

Perancangan sistem yang akan dibuat dapat diilustrasikan dalam diagram blok seperti berikut berikut.

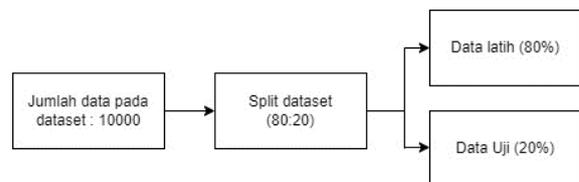


Gambar 1. Diagram Blok sistem

Penelitian ini dilakukan dalam tahap siklus yang saling berkaitan, dimulai dengan melakukan studi pustaka yaitu mempelajari mengenai konsep pemrograman dan cara kerja dari setiap komponen yang akan digunakan dalam penelitian. Setelah itu, tahap selanjutnya adalah melakukan perancangan yaitu membuat sistem perancangan perangkat lunak. Perangkat lunak yaitu membuat *script code* dari sistem deteksi nominal uang kertas rupiah sehingga akan didapatkan *output* yang sesuai dengan hasil yang diharapkan. Tahap selanjutnya merupakan pengujian terhadap perancangan sistem yang telah dibuat, yang akan menghasilkan data pendukung untuk menyelesaikan atau menjawab permasalahan yang diangkat. Setiap data yang diperoleh dari pengujian sistem akan diolah dan dianalisis mengenai keberhaslilan dan tingkat akurasi. Tahap terakhir adalah menarik kesimpulan terhadap hasil yang didapat.

**Dataset**

Pada tahap ini pengambilan gambar untuk pembuatan dataset menggunakan format file .jpg dengan nominal uang kertas (1000, 2000, 5000, 10000, 20000, 50000, 100000) emisi 2016 dan emisi 2022. Ukuran gambar pada dataset di *resize* sehingga gambar pada file memiliki ukuran 7-14 KB. Jumlah gambar pada dataset terbagi menjadi tiga folder, yaitu folder *train* yang berisi jumlah gambar sebanyak 6114, folder *test* yang terdiri dari 282 gambar, dan terakhir folder *valid* yang terdiri dari 639 gambar.



Gambar 2. Ilustrasi split dataset

berikut ini adalah gambar dari uang kertas rupiah tersebut:



Gambar 3. Uang kertas emisi 2022



Gambar 4. Uang kertas emisi 2016

Setelah melakukan pengambilan gambar ini dan membaginya, maka tahap selanjutnya adalah melakukan anotasi atau pelabelan pada gambar uang rupiah.

**Anotasi Gambar dan struktur folder**

Anotasi gambar merujuk pada sebuah proses dalam menambahkan metadata atau informasi tambahan kepada gambar. Metadata ini digunakan untuk memberikan penjelasan tentang objek, lokasi, kejadian, atau atribut lain yang terdapat pada gambar. Dalam hal ini anotasi gambar dilakukan dengan cara memberikan atau membuat *box detection* pada suatu objek berupa gambar dan memberikan penamaan pada setiap gambar sesuai dengan kelas masing-masing. Berikut ini adalah gambar dari anotasi gambar untuk pembuatan dataset:



Gambar 5. Anotasi Gambar

Hasil dari anotasi gambar tersebut yaitu berupa file XML berupa sebuah kode *script* yang terlihat seperti pada gambar dibawah ini:

```
<annotation>
  <folder />
  <filename>1000_1_jpg.rf.28bbaf8dc9d61653daadd91e1cf9e795.jpg</filename>
  <path>1000_1_jpg.rf.28bbaf8dc9d61653daadd91e1cf9e795.jpg</path>
  <source>
    <database>roboflow.ai</database>
  </source>
  <size>
    <width>300</width>
    <height>300</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>Seribu</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <occluded>0</occluded>
    <bndbox>
      <xmin>91</xmin>
      <xmax>301</xmax>
      <ymin>99</ymin>
      <ymax>219</ymax>
    </bndbox>
  </object>
</annotation>
```

Gambar 6. Kode *script* file XML

Lalu setelah melakukan anotasi ini maka hasil dari folder pada pembuatan dataset gambar adalah sebagai berikut:

test	21/05/2023 14.06	File folder
train	22/06/2023 13.15	File folder
valid	22/06/2023 13.17	File folder

Gambar 7. Folder gambar dataset

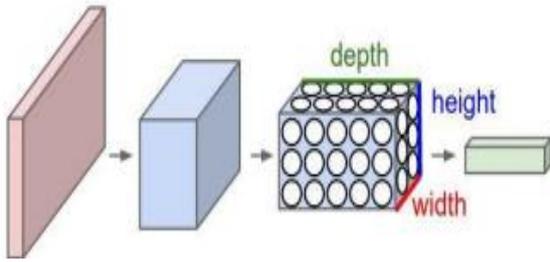
Berdasarkan masing-masing folder, maka akan ada masing-masing gambar dan kode *script* file xml sesuai nominal uang kertas, yang terlihat seperti pada gambar berikut ini:

1000_1_jpg.rf.28bbaf8dc9d61653daadd91...	25/05/2023 09.18	JPG File	10 KB
1000_1_jpg.rf.28bbaf8dc9d61653daadd91...	26/05/2023 12.39	Microsoft Edge H...	1 KB
1000_2_jpg.rf.eb552f0b00f18db0cfece416...	25/05/2023 09.18	JPG File	10 KB
1000_2_jpg.rf.eb552f0b00f18db0cfece416...	26/05/2023 12.39	Microsoft Edge H...	1 KB
1000_3_jpg.rf.3201c8019be31a9f91015358...	25/05/2023 09.18	JPG File	10 KB
1000_3_jpg.rf.3201c8019be31a9f91015358...	26/05/2023 12.39	Microsoft Edge H...	1 KB

Gambar 8. File jpg dan file xml

**Convolutional Neural Network (CNN)**

CNN merupakan jaringan syaraf yang diciptakan bertujuan untuk memproses data yang memiliki struktur kotak, seperti pada data citra dua dimensi. CNN memiliki lapisan konvolusi yang mana merupakan lapisan utama yang memiliki peran penting untuk digunakan. Lalu, ada juga lapisan lainnya yaitu *pooling layer* yang merupakan untuk mengambil nilai pada bagian lapisan pixel dalam citra. Berikut ini adalah gambaran dari struktur CNN:



Gambar 9. Struktur CNN untuk citra

Berdasarkan gambar di atas dapat diketahui bahwa setiap lapisan memiliki susunan neural 3 dimensi, yaitu lebar, tinggi, dan kedalaman.

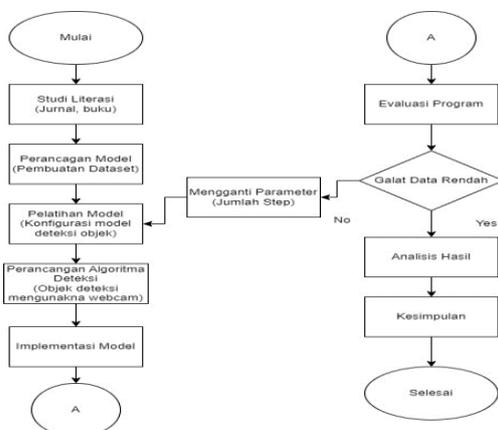
Lapisan konvolusi pada umumnya terdiri dari operasi yaitu operasi konvolusi, *pooling*, dan fungsi aktivasi, berikut ini adalah rumus dari persamaan operasi konvolusi

$$(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a) * w(t - a) \quad (1)$$

Operasi konvolusi yang digunakan pada fungsi  $x(t)$  dengan nilai bobot  $w(t)$ ,  $x*w$ . Pada persamaan tersebut,  $s(t)$  merupakan fungsi dari hasil operasi konvolusi, dengan  $t$  merupakan variable dari fungsi dan  $a$  merupakan sebuah konstanta.

**Desain Sistem**

Desain sistem deteksi nominal uang kertas rupiah yang digunakan dalam penelitian ini terdiri dari perancangan sistem perangkat lunak (*software*). Berikut ini adalah diagram alir dari perancangan dan pelatihan model pada penelitian ini:

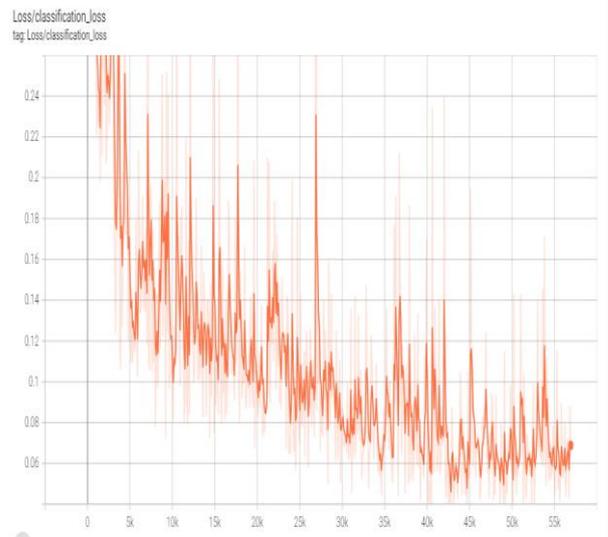


Gambar 10. Diagram alir perancangan sistem

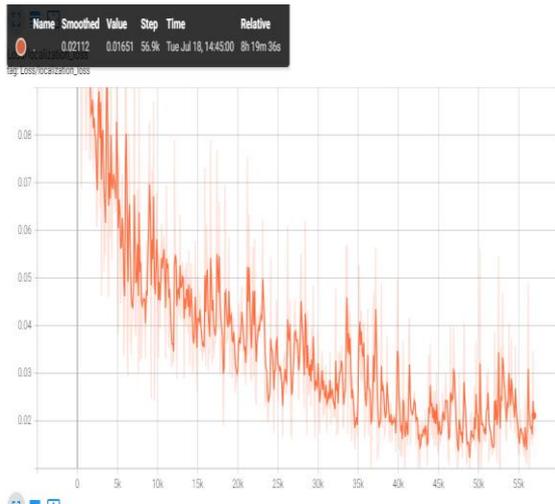
Berdasarkan gambar 10 diagram alir di atas studi literasi dalam melakukan penelitian ini berdasarkan dari sumber jurnal dan buku, perancangan model dari membuat dataset, melakukan anotasi dan augmentasi dataset. Setelah itu, tahap selanjutnya adalah pelatihan model yang pada tahap ini mengatur konfigurasi pipeline, jumlah *step*, membuat *tfrecord*, membuat *labelmap*, dan menjalankan program untuk melatih model. Setelah pelatihan model selesai akan ada sebuah folder yang terbentuk yang berisikan file *.ckpt* yang mana hasil dari pelatihan tersebut berupa *checkpoint*. Tahap selanjutnya yaitu implementasi model yang mana hasil deteksi modelnya akan dilakukan menggunakan kamera *webcam* yang mana menggunakan *library* *OpenCV* untuk mengambil gambar dari *webcam*. Lalu setelah itu pada tahap evaluasi program yang mana hasil dari implementasi berupa koordinat *bounding box*, label kelas objek, dan hasil nilai akurasi model dalam *input gambar*.

**HASIL DAN PEMBAHASAN**

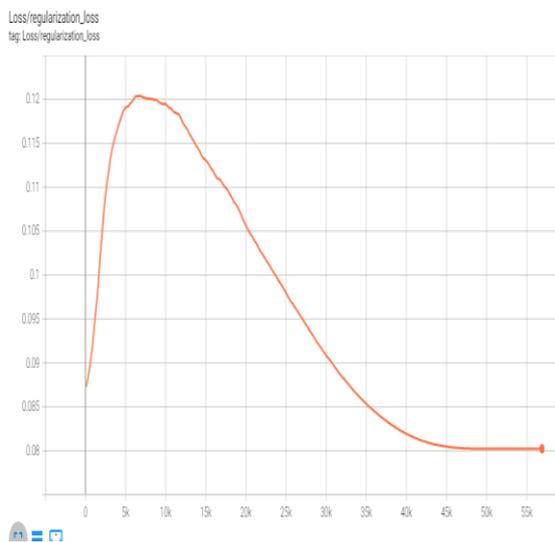
Setelah dilakukan proses pelatihan menggunakan arsitektur *SSD MobileNetv3* akan menghasilkan folder training yang mana akan di generate, sehingga model dapat digunakan untuk melakukan proses klasifikasi dan deteksi. Saat proses pelatihan model ada beberapa tahap yang dilalui seperti *initializing data*, *resizing image*, dan konversi gambar ke *NumPy array*. Berikut ini adalah hasil dari evaluasi pengujian model dengan nilai total *step* sebanyak 56900:



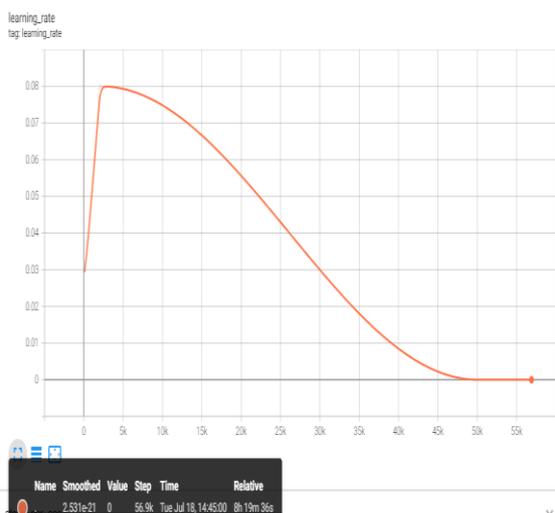
Gambar 11. Grafik Classification Loss



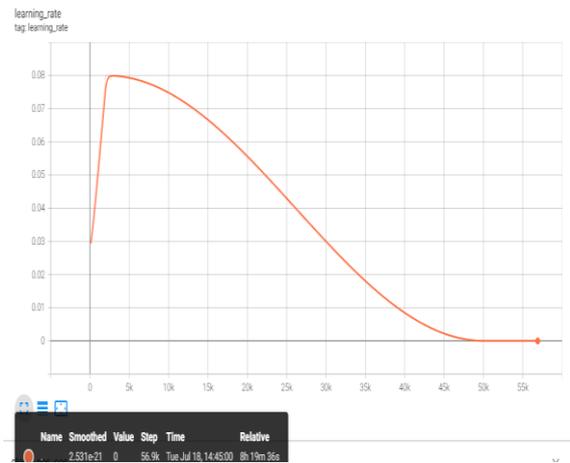
Gambar 12. Grafik Localization loss



Gambar 13. Grafik Regularization Loss



Gambar 14. Grafik Total loss



Gambar 15. Grafik Learning Rate

Dalam analisis yang mendalam terhadap gambar grafik di atas, terlihat secara nyata bahwa pelatihan model dilakukan dengan tingkat keseriusan dan keahlian yang tinggi, menghasilkan performa yang mengesankan dan memberikan implikasi yang signifikan terhadap kemajuan dalam bidang kecerdasan buatan. Grafik yang menggambarkan perubahan nilai loss pada komponen utama, yaitu classification loss, localization loss, regularization loss, dan total loss, memberikan gambaran yang jelas tentang perkembangan dan konvergensi model sepanjang proses pelatihan. Dalam kurva yang dihasilkan, terlihat secara visual bahwa nilai loss dari setiap komponen secara bertahap menurun hingga mencapai tingkat yang sangat rendah, mengindikasikan kemampuan adaptasi dan pembelajaran model yang kuat.

Melalui pemerhatian mendalam terhadap grafik, dapat diketahui bahwa perubahan nilai loss ini terjadi secara konsisten dan stabil seiring dengan meningkatnya jumlah iterasi pelatihan. Setiap langkah pelatihan, yang berjumlah 56900, memberikan kontribusi signifikan terhadap peningkatan performa dan penurunan nilai loss. Fenomena ini menggambarkan keahlian yang luar biasa dalam merancang dan mengimplementasikan algoritma pelatihan yang efektif, yang memungkinkan model untuk mengenali dan memahami objek dengan tingkat keakuratan dan generalisasi yang tinggi. Selain itu, grafik juga mengungkapkan dinamika yang menarik dalam perjalanan pelatihan model. Dapat dilihat bahwa model telah mencapai tingkat penyesuaian parameter yang optimal dan berhasil mengurangi overfitting, yang tecermin dari penurunan bertahap dan berkelanjutan dalam nilai loss. Berikut ini adalah tabel hasil dari grafik dalam perhitungan per-1000 steps

Tabel 1. Data kesalahan dan learning rate model setiap 10.000 langkah pelatihan

Jumlah Step	Classification	Loss		Total	Learning Rate
		Localization	Regularization		
1000	0,2469	0,07672	0,1964	0,4365	0,056
2000	0,2068	0,07521	0,1078	0,3899	0,08
3000	0,1624	0,08244	0,1148	0,3593	0,07988
4000	0,1865	0,08234	0,1171	0,5798	0,07966
5000	0,1276	0,06367	0,1192	0,386	0,0792
6000	0,2131	0,1175	0,1202	0,4508	0,0788
7000	0,1178	0,055	0,1202	0,5495	0,07712
8000	0,07813	0,03377	0,1201	0,2931	0,07701
9000	0,1729	0,08274	0,1198	0,3574	0,0769
10000	0,09565	0,04645	0,1195	0,2618	0,0758
...	...	...	...	...	...
56900	0,06837	0,01651	0,08023	0,1651	0,0

Berdasarkan hasil dalam table dapat disimpulkan bahwa model melakukan pelatihan dengan baik, hal ini karena pada table nilai dari total loss cenderung menurun setiap 1000 step. Keberhasilan ini menunjukkan kemampuan model dalam menangkap dan merepresentasikan fitur-fitur penting dari objek yang diberikan, sehingga mampu menghasilkan prediksi yang akurat dan konsisten. Dengan demikian, grafik yang tergambar memberikan bukti konkret akan keberhasilan dan kualitas pelatihan model. Hasil yang impresif ini memberikan landasan yang kuat untuk implementasi model dalam berbagai aplikasi kecerdasan buatan yang membutuhkan deteksi objek yang akurat dan andal. Selain itu, kesuksesan ini juga memperlihatkan dedikasi dan ketekunan para ahli di bidang ini, yang secara komprehensif dan detail mampu mengembangkan model yang superior dan mengoptimalkan proses pelatihan secara efisien. Selain itu berikut ini adalah hasil dari percobaan sistem deteksi:

Tabel 2. Hasil percobaan deteksi siang hari di dalam ruangan

No	Input	Output visual	Keterangan
1	Seribu		Uang terdeteksi dengan akurasi 88%

2	Dua ribu		Uang Terdeteksi dengan akurasi 93%
3	Lima ribu		Uang terdeteksi dengan akurasi 90%
4	Sepuluh ribu		Uang terdeteksi dengan akurasi 91%
5	Dua puluh ribu		Uang terdeteksi dengan akurasi 91%
6	Lima puluh ribu		Uang terdeteksi dengan akurasi 92%
7	Seratus ribu		Uang terdeteksi dengan akurasi 94%

Tabel 3. Hasil percobaan deteksi malan hari di luar ruangan

No	Input	Output visual	Keterangan
1	Seribu		Tidak dapat mendeteksi uang
2	Dua ribu		Tidak dapat mendeteksi uang
3	Lima ribu		Tidak dapat mendeteksi uang
4	Sepuluh ribu		Tidak dapat mendeteksi uang

5	Dua puluh ribu		Uang terdeteksi dengan akurasi 71%
6	Lima puluh ribu		Tidak dapat mendeteksi uang
7	Seratus ribu		Tidak dapat mendeteksi uang

Berdasarkan hasil pada tabel 1 dan 2, dapat dilihat bahwa nilai akurasi dari model saat melakukan deteksi objek berbeda-beda hal ini terjadi pada beberapa faktor, seperti faktor cahaya, resolusi gambar, resolusi latar belakang, posisi uang kertas, dan juga kualitas uang kertas rupiah yang di input karena kondisi uang rupiah dapat berbeda-beda. Rata – rata hasil dari akurasi sistem yang dilakukan pada siang hari adalah sebagai berikut:

$$\text{Rata – rata} = \frac{\text{jumlah total akurasi}}{\text{jumlah sample}} \quad (2)$$

$$\text{Rata – rata} = 91,28\%$$

Berdasarkan pada hasil dari table pengujian dilakukan terhadap nominal uang kertas rupiah mulai dari nominal Rp. 1000, Rp. 2000, Rp. 5000, Rp. 10000, Rp. 20000, Rp. 50000, Rp. 100000. Pada hasil percobaan sistem dapat melakukan deteksi dengan baik untuk setiap nominal uang kertas rupiah pada waktu siang hari karena memiliki tingkat pencahayaan yang cukup. Namun, sistem kesulitan untuk mendeteksi nominal uang kertas di malam hari, karena pada malam hari cahaya redup dan dari hasil percobaan hanya uang Rp.20000 yang dapat terbaca oleh sistem.

## KESIMPULAN

Kesimpulan dari hasil pengujian ini adalah bahwa model menggunakan SSD MobileNetv3 menunjukkan kinerja yang baik dalam mendeteksi objek, terutama dalam kondisi siang hari dengan akurasi antara 80% - 95%. Model juga berhasil mendeteksi nominal uang. Namun, model ini mengalami kesulitan dalam mendeteksi objek pada malam hari dengan pencahayaan minim. Meskipun demikian, rata-rata akurasi sistem sebesar 91,28% adalah hasil yang memuaskan

dalam deteksi objek. Penting untuk diperhatikan bahwa kualitas kamera memiliki pengaruh signifikan terhadap performa model, khususnya saat beroperasi pada malam hari dengan pencahayaan rendah. Diperlukan kamera yang mampu menangkap gambar dengan baik dalam kondisi pencahayaan rendah seperti di malam hari untuk meningkatkan performa model secara keseluruhan. Dalam penelitian selanjutnya, dapat direkomendasikan beberapa hal. Pertama, perlu dilakukan penyempurnaan terhadap model agar lebih baik dalam mendeteksi objek pada kondisi minim pencahayaan. Kedua, penelitian lebih lanjut dapat memfokuskan pada pengembangan algoritma khusus yang memungkinkan deteksi objek yang lebih baik di malam hari. Terakhir, perlu dilakukan eksperimen lebih lanjut dengan berbagai jenis kamera untuk memahami lebih lanjut bagaimana kualitas kamera mempengaruhi performa model. Dengan demikian, penelitian selanjutnya dapat memberikan kontribusi yang lebih besar dalam pengembangan teknologi deteksi objek dalam berbagai kondisi pencahayaan.

## REFERENSI

- [1] O. E. Pamungkas *et al.*, "Classification of Rupiah to Help Blind with The Convolutional Neural Network Method," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 2, pp. 259–268, 2022.
- [2] K. M. Azhar, I. Santoso, and Y. A. A. Soetrisno, "IMPLEMENTASI DEEP LEARNING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DAN ALGORITMA YOLO DALAM SISTEM PENDETEKSI UANG KERTAS RUPIAH BAGI PENYANDANG LOW VISION," *Transient: Jurnal Ilmiah Teknik Elektro*, vol. 10, no. 3, pp. 502–509, 2021.
- [3] L. Tan, T. Huangfu, L. Wu, and W. Chen, "Comparison of YOLO v3, faster R-CNN, and SSD for real-time pill identification," 2021.
- [4] J. Kim, J.-Y. Sung, and S. Park, "Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition," in *2020 IEEE international conference on consumer electronics-Asia (ICCE-Asia)*, IEEE, 2020, pp. 1–4.
- [5] D. Jakhar and I. Kaur, "Artificial intelligence, machine learning and deep learning: definitions and differences," *Clin Exp Dermatol*, vol. 45, no. 1, pp. 131–132, 2020.
- [6] M. R. Dobbelaere, P. P. Plehiers, R. Van de Vijver, C. V. Stevens, and K. M. Van Geem, "Machine learning in chemical engineering:

- strengths, weaknesses, opportunities, and threats,” *Engineering*, vol. 7, no. 9, pp. 1201–1211, 2021.
- [7] L. G. Wright *et al.*, “Deep physical neural networks trained with backpropagation,” *Nature*, vol. 601, no. 7894, pp. 549–555, 2022.
- [8] Y. Jiang, X. Li, H. Luo, S. Yin, and O. Kaynak, “Quo vadis artificial intelligence?,” *Discover Artificial Intelligence*, vol. 2, no. 1, p. 4, 2022.
- [9] I. Ahmed, G. Jeon, and F. Piccialli, “From artificial intelligence to explainable artificial intelligence in industry 4.0: a survey on what, how, and where,” *IEEE Trans Industr Inform*, vol. 18, no. 8, pp. 5031–5042, 2022.
- [10] Y. Ghasemi, H. Jeong, S. H. Choi, K.-B. Park, and J. Y. Lee, “Deep learning-based object detection in augmented reality: A systematic review,” *Comput Ind*, vol. 139, p. 103661, 2022.
- [11] L. Huang, G. Liu, Y. Wang, H. Yuan, and T. Chen, “Fire detection in video surveillances using convolutional neural networks and wavelet transform,” *Eng Appl Artif Intell*, vol. 110, p. 104737, 2022.
- [12] A. Howard *et al.*, “Searching for mobilenetv3,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.
- [13] C. G. W. Pramana, D. C. Khrisne, and N. P. Sastra, “Rancang Bangun Object Detection Pada Robot Soccer Menggunakan Metode Single Shot Multibox Detector (SSD MobileNetV2),” *Jurnal SPEKTRUM Vol*, vol. 8, no. 2, 2021.
- [14] Z.-Q. Zhao, P. Zheng, S. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE Trans Neural Netw Learn Syst*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [15] Y. Xiong *et al.*, “MobileDets: Searching for object detection architectures for mobile accelerators,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3825–3834.