



Universitas
Muhammadiyah
Sorong



2024

MACHINE LEARNING KLASIFIKASI MATA UANG RUPIAH
MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK



Disusun Oleh:

Haldi Alwijais Rumadedey
Arlince Worotmasom Bame
Martina Mirino

**MACHINE LEARNING KLASIFIKASI MATA UANG RUPIAH
MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL
NETWORK**



Disusun Oleh:

KELOMPOK 1

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH SORONG
TAHUN 2024**

LEMBAR PERSETUJUAN

MACHINE LEARNING KLASIFIKASI MATA UANG RUPIAH MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Nilai UTS dan UAS
Mata Kuliah Algoritma dan Pemrograman 2
Pada Prodi Informatika Fakultas Teknik
Universitas Muhammadiyah Sorong**

**Disusun Oleh:
KELOMPOK 1**



Sorong, 25 September 2024

Menyetujui dan Mengetahui

Menyetujui

Dosen Pengganti Mata Kuliah

Ketua Kelompok 1

Fajar R. B Putra, S.Kom., M.Kom.

NIDN: 1428099501

Haldi Alwijais Rumadedey

NIM: 202355202081

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat, rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Besar dengan judul "*Machine Learning Klasifikasi Mata Uang Rupiah Menggunakan Algoritma Convolutioal Neural Network*". Adapun Tugas Besar ini disusun sebagai salah satu syarat untuk memperoleh nilai UTS dan UAS Mata Kuliah Algortima dan Pemograman 2, Program Studi Teknik Informatika, Fakultas Teknik, UNAMIN. Tentunya tidak lupa yang kami hormati kepada:

1. Bapak Dr. H. Muhammad Ali, M.M., M.H. Selaku Rektor Universitas Muhammadiyah Sorong
2. Bapak Ir. Hendrik Pristianto, ST., M.T., IPM. selaku Dekan Fakultas Teknik
3. Bapak Ir. Rendra Soekarta, S.Kom., M.T., IPP. selaku Kaprodi Teknik Informatika
4. Teman-teman dan juga sahabat-sahabatku.

Penulis menyadari bahwa penyusunan Tugas Besar ini masih banyak terdapat kekurangan, maka dari itu kelompok mengharapkan kritikan dan saran yang bersifat membangun.

Sorong, 28 April 2024

KELOMPOK 1

DAFTAR ISI

LEMBAR PERSETUJUAN.....	ii
KATA PENGANTAR	iii
DAFTAR ISI.....	iv
DAFTAR TABEL.....	vi
DAFTAR GAMBAR	vii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan.....	2
1.4. Batasan Masalah.....	3
BAB II LANDASAN TEORI	4
2.1. State Of The Art	4
2.2. Literatur Terkait	14
2.2.1. Uang Kertas Rupiah	14
2.2.2. Machine Learning.....	14
2.2.3. Convolutional Neural Network (CNN)	14
2.2.4. VGG16	15
2.2.5. NASNetMobile.....	15
2.2.6. Xception	16
2.2.7. MobileNetV2.....	16
2.2.8. Python.....	17
2.2.9. Jupyter Notebook	17
2.2.10. HTML	17
2.2.11. CSS.....	18
2.2.12. Pengertian FlowCharts	18

2.2.13.	Metode pengujian sistem.....	19
2.2.14.	Blackbox Testing	19
2.2.15.	Usability Testing	20
BAB III ANALISA DATA DAN PEMBAHASAN.....		21
3.1.	Analisa Data	21
3.1.1.	Percobaan A: Hasil pembentukan model CNN MobileNetV2.....	23
3.1.2.	Percobaan B : Hasil pembentukan model CNN VGG16.....	30
3.1.3.	Percobaan C : Hasil pembentukan model CNN Xception.....	37
3.1.4.	Percobaan D: Hasil pembentukan model CNN NASNetMobile.....	44
3.2.	Flowchart sistem.....	51
3.3.	Hasil Klasifikasi	53
3.4.	Tampilan Interface.....	57
3.5.	Pengujian	61
BAB IV PENUTUP		69
3.1.	Kesimpulan.....	69
3.2.	Saran.....	70
DAFTAR PUSTAKA		71
LAMPIRAN 1		76
LAMPIRAN 2		77
LAMPIRAN 3		78
FORM PENGISIAN PENGERJAAN TUGAS BESAR		79

DAFTAR TABEL

Tabel 1 Perbandingan peneliti penulis	12
Tabel 2 Flowchart	18
Tabel 3 Dataset.....	21
Tabel 4 Komposisi model	22
Tabel 5 Tabel Hyperparameter Percobaan A	25
Tabel 6 Tabel Hyperparameter Percobaan B	32
Tabel 7 Tabel Hyperparameter Percobaan C	39
Tabel 8 Tabel Hyperparameter Percobaan D	46
Tabel 9 Pengujian Blackbox Testing	62
Tabel 10 Pengisian pengerjaan tugas besar.....	79

DAFTAR GAMBAR

Gambar 1 <i>state of the art</i>	4
Gambar 2 Flowchart pembentukan model CNN MobileNetV2	23
Gambar 3 Hasil deteksi model CNN <i>MobileNetv2</i>	24
Gambar 4 Model Summary Percobaan A	25
Gambar 5 Percobaan A dilakukan dengan 25 <i>epoch</i> atau 25 kali iterasi	27
Gambar 6 Grafik Percobaan A	28
Gambar 7 <i>Confusion Matrix</i> Percobaan A	29
Gambar 8 Flowchart pembentukan model CNN VGG16.....	30
Gambar 9 Hasil deteksi model CNN <i>VGG16</i>	31
Gambar 10 <i>Model Summary</i> Percobaan B	32
Gambar 11 Percobaan B dilakukan dengan 25 <i>epoch</i> atau 25 kali iterasi	34
Gambar 12 Grafik Percobaan B	35
Gambar 13 Confusion Matriks Percobaan B	36
Gambar 14 pembentukan model CNN Xception	37
Gambar 15 Hasil deteksi model CNN <i>Xception</i>	38
Gambar 16 Model Summary Percobaan C.....	39
Gambar 17 Percobaan C dilakukan dengan 25 <i>epoch</i> atau 25 kali iterasi	41
Gambar 18 Grafik Percobaan C	42
Gambar 19 Confusion Matriks Percobaan C	43
Gambar 20 Flowchart pembentukan model CNN <i>NASNetMobile</i>	44
Gambar 21 Hasil deteksi model CNN <i>NASNetMobile</i>	45
Gambar 22 Model Summary Percobaan D	46
Gambar 23 Percobaan D dilakukan dengan 25 <i>epoch</i> atau 25 kali iterasi	48
Gambar 24 Grafik Percobaan D	49
Gambar 25 Confusion Matriks Percobaan D	50
Gambar 26 Flowchart Sistem.....	51
Gambar 27 Hasil klasifikasi Rp 1000	53
Gambar 28 Hasil klasifikasi Rp 2000	54

Gambar 29 Hasil klasifikasi Rp 5000	54
Gambar 30 Hasil klasifikasi Rp 10.000	55
Gambar 31 Hasil klasifikasi Rp 20.000	55
Gambar 32 Hasil klasifikasi Rp 50.000	56
Gambar 33 Hasil klasifikasi Rp 100.000	56
Gambar 34 Halaman <i>CNN Model</i>	57
Gambar 35 <i>Chart</i>	58
Gambar 36 <i>Confusion Matrix</i>	58
Gambar 37 Image Original.....	59
Gambar 38 <i>Image RGB</i>	59
Gambar 39 <i>Image Classification</i>	60
Gambar 40 Pilih File	60
Gambar 41 Submit	61
Gambar 42 Presentase Pertanyaan 1	64
Gambar 43 Presentase Pertanyaan 2	65
Gambar 44 Presentase Pertanyaan 3	66
Gambar 45 Presentase Pertanyaan 4	66
Gambar 46 Presentase Pertanyaan 5	67
Gambar 47 Presentase Pertanyaan 6	68

BAB I

PENDAHULUAN

1.1. Latar Belakang

Di Indonesia, Rupiah (IDR) merupakan mata uang yang sah untuk dipakai pada proses jual beli. menurut penelitian (Aprillia dkk. 2024). Uang kertas rupiah memiliki berbagai pecahan seperti “RP 1.000, RP 2.000, RP 5.000, RP 10.000, RP 20.000, RP 50.000, RP 75.000, dan RP 100.000”. Namun terkadang masyarakat salah dalam mengenali nilai mata uang rupiah, maka program yang dapat mengidentifikasi nilai mata uang rupiah berdasarkan gambar uang kertas akan cukup membantu. Tujuannya untuk mencegah terjadinya kesalahan dalam mengenali nilai mata uang rupiah.

Berdasarkan penelitian (Desi Handayani, Syifa Nurulia, dan Udin Saripudin 2024), Konsep uang telah berkembang dari waktu ke waktu dengan perkembangan peradaban dan kebutuhan manusia. Uang dapat didefinisikan sebagaimana fungsinya, yaitu sebagai alat tukar, dan sebagai standar pembayaran.

Machine learning atau pembelajaran mesin Berdasarkan penelitian, (Gomathy dkk., 2021) Pembelajaran Mesin adalah domain yang menggunakan data masa lalu untuk memprediksi. Pembelajaran Mesin adalah pemahaman tentang sistem komputer tempat model Pembelajaran Mesin belajar dari data dan pengalaman.

Alat identifikasi nilai mata uang rupiah masih belum terjangkau khususnya bagi pedagang kecil dikarenakan harganya, maka program yang dapat

mengidentifikasi nilai mata uang rupiah yang gratis akan sangat membantu. Agar kedepannya kesalahan dalam mengenali nilai mata uang rupiah dapat dihindari.

Laporan dengan judul “*Machine Learning Identifikasi Mata Uang Rupiah Menggunakan Algoritma Convolutional Neural Network*”, merupakan sebuah penelitian yang bertujuan untuk mengembangkan program yang dapat mengidentifikasi nilai mata uang rupiah berdasarkan gambar uang kertas dengan algoritma *Convolutional Neural Network* (CNN).

Dalam penelitian ini, Terdapat berbagai pecahan uang kertas rupiah yang memiliki ciri khasnya masing-masing seperti nominal, gambar tokoh pahlawan, dan warna. Selain berdasarkan ciri tersebut, pecahan uang rupiah dapat di identifikasi menggunakan algoritma *Convolutional Neural Network* (CNN) dengan 4 jenis model, yaitu *MobileNetV2*, *VGG16*, *NasNetMobile*, dan *Xception*.

1.2. Rumusan Masalah

Berdasarkan Latar Belakang di atas maka dapat dirumuskan beberapa masalah, Rumusan masalah dalam penelitian ini adalah:

1. Bagaimana mengembangkan progam yang efektif dalam mengidentifikasi nilai mata uang rupiah?
2. Bagaimana mengembangkan program identifikasi nilai mata uang rupiah yang terjangkau oleh semua orang dan akurat menggunakan algoritma CNN?

1.3. Tujuan

Berdasarkan rumusan masalah yang ditentukan di atas maka tujuan dari penelitian ini adalah:

1. Mengembangkan sistem yang dapat mengidentifikasi nilai mata uang rupiah berdasarkan gambar dari uang kertas.
2. Mengembangkan sebuah sistem identifikasi yang terjangkau dan akurat untuk para pedagang kecil.

1.4. Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

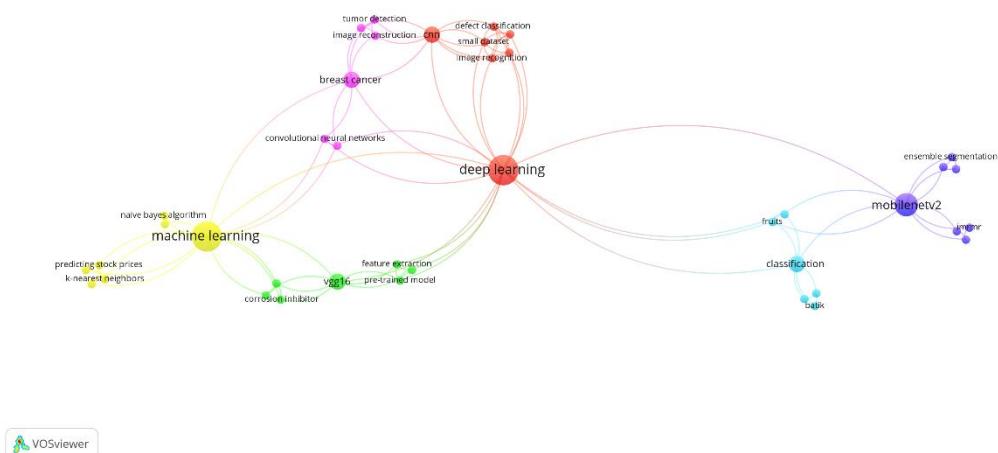
1. Dataset yang digunakan berdasarkan data studi kasus yang di dapat melalui website Kaggle.
2. Program ini hanya berfokus pada identifikasi nilai mata uang rupiah, bukan keaslian uang rupiah.
3. Uang rupiah yang digunakan adalah Emisi 2022
4. Program ini hanya berbasis Website.

BAB II

LANDASAN TEORI

2.1. State Of The Art

State of the art diambil dari beberapa penelitian yang telah dilakukan sebelumnya sebagai panduan serta menjadi acuan perbandingan dalam penelitian yang akan dilakukan. *State of the art* dalam penelitian ini dapat dilihat pada Gambar 1 di bawah:



Gambar 1 *state of the art*

State of the art diatas diambil dari 20 jurnal penelitian terdahulu yang berkaitan dengan program penulis, 20 jurnal tersebut terdiri dari 10 jurnal nasional dan 10 jurnal internasional.

2.2. Studi Literatur

Studi literatur adalah teknik pengumpulan data atau cara untuk menyelesaikan persoalan dengan menelusuri sumber-sumber tulisan yang pernah dibuat sebelumnya. Dengan kata lain, istilah studi literatur ini juga sangat familiar dengan sebutan studi pustaka. Dalam hal ini penulis mengutip beberapa jurnal yang dijadikan acuan sebagai sumber untuk membuat sebuah aplikasi “*Machine Learning Identifikasi Mata Uang Rupiah Menggunakan Algoritma Convolutional Neural Network*” yang telah dibuat. Berikut beberapa jurnal yang berkaitan dengan penelitian yang akan dilakukan:

1. Jurnal nasional: “*Kajian Ilmiah Informatika dan Komputer Deteksi Nominal Mata Uang Rupiah Menggunakan Metode Convolutional Neural Network dan Feedforward Neural Network*” berdasarkan penelitian (Aprillia dkk. 2024), Di Indonesia, Rupiah (IDR) merupakan mata uang yang sah untuk dipakai pada proses jual beli.
2. Jurnal nasional: “*Deteksi Nominal Rupiah Uang Kertas Berdasarkan Citra Warna Menggunakan Segmentasi K-Means Clustering dan Klasifikasi Random Forest*” berdasarkan penelitian (Antara dkk. 2023), Uang adalah sebuah benda yang digunakan oleh masyarakat sebagai alat tukar dan pembayaran barang yang sah, yang dikeluarkan oleh lembaga negara dengan bentuk dan gambar tertentu.
3. Jurnal nasional: “*Relevansi Konsep Uang Perspektif Ibnu Miskawaih di Era Modernisasi*” berdasarkan penelitian (Desi Handayani dkk. 2024), Konsep uang telah berkembang dari waktu ke waktu dengan perkembangan

peradaban dan kebutuhan manusia. Uang dapat didefinisikan sebagaimana fungsinya, yaitu sebagai alat tukar, dan sebagai standar pembayaran.

4. Jurnal nasional: “***Identifikasi Nilai Mata Uang Kertas Rupiah Dengan Metode Ekstrasi Ciri Local Binary Pattern Dan Metode Klasifikasi Naïve Bayes***” berdasarkan penelitian (Mardha dkk. 2022), Terdapat berbagai pecahan uang kertas rupiah yang memiliki ciri khasnya masing-masing seperti nominal, gambar tokoh pahlawan, dan warna.
5. Jurnal nasional: “***Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN***” berdasarkan penelitian (Sri Rahmadhani & Lysbetti Marpaung, 2023), *Deep Learning* merupakan salah satu bagian dalam Machine Learning yang menggunakan algoritma, yang dibuat berdasarkan pada hukum matematik, yang bekerja layaknya otak manusia. Salah satu metode *Deep Learning* yang digunakan dalam *Image Processing* adalah *Convolutional Neural Network (CNN)*.
6. Jurnal nasional: “***Python : Dasar Dan Pemrograman Berorientasi Objek Tahita Media Group***” berdasarkan penelitian (Rahman dkk., 2023) *Python* merupakan salah satu perangkat lunak yang sedang populer saat ini. Dengan Python, kita memiliki kemampuan untuk melakukan analisis data, menjalankan perhitungan data statistik yang kompleks atau memakan waktu, membuat visualisasi data, mengimplementasikan algoritma machine learning, dan juga dapat digunakan untuk manipulasi data serta menyelesaikan berbagai tugas matematika lainnya. Keunggulan Python terletak pada kemampuannya menghasilkan hasil yang lebih akurat dan efisien dibandingkan dengan metode manual.

7. Jurnal nasional: “**Pendektesi Nominal Uang Pada gambar menggunakan convolutional Neural Network : Integrasi Metode Pra-Pemrosesan Citra Dan Klasifikasi Berbasis CNN**” berdasarkan penelitian (Malik Ibrahim, Rahmadewi, dan Nurpulaela 2023a) Dalam penelitian ini akan dirancang sebuah system pendektesian uang kertas pada sebuah citra menggunakan arsitektur *deep learning Convolutional Neural Network (CNN)* dengan integrasi pra-pemrosesan citra menggunakan teknik anotasi *user-based manual labeling code program python*.
8. Jurnal nasional: “**Pembelajaran Mendalam Pengklasifikasi Ekspresi Wajah Manusia dengan Model Arsitektur Xception pada Metode Convolutional Neural Network**” berdasarkan penelitian (Musa dkk. 2023) Model arsitektur *Xception* merupakan model arsitektur berbasis *Convolutional Neural Networks* yang dapat menghilangkan lapisan yang terhubung sepenuhnya (*fully-connected layer*) dan jumlah parameter yang digunakan jauh berkurang pada lapisan konvolusional yang ada.
9. Jurnal nasional: “**Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning**” berdasarkan penelitian (Eka dkk. 2023a), Arsitektur ini menggunakan *depth-wise separable convolutions* sebagai pengganti dari modul *Inception*. *Xception* terdiri dari 3 flow, yaitu *entry, flow, middle flow, dan exit flow*. Data yang digunakan sebagai input nantinya akan masuk melalui *entry flow* terlebih dahulu kemudian melalui *middle flow* sebanyak 8 kali kemudian menuju ke *exit flow*.
10. Jurnal nasional: “**Perubahan Topik Media Tentang Program Keluarga Berencana Pascarebranding Bkkbn: Pendekatan Pemodelan Topik**

Digital” berdasarkan penelitian (Dwi Fajar Maulana dkk., 2021), *Jupyter Notebook* merupakan perangkat lunak berbasis kode terbuka yang digunakan menggunakan browser yang berfungsi sebagai *notebook lab* virtual untuk mendukung alur kerja, kode, data, dan visualisasi.

11. Jurnal internasional: “*A convolutional neural network -VGG16 method for corrosion inhibition of 304SS in sulfuric acid solution by timoho leaf extract*” berdasarkan penelitian (Gapsari dkk. 2024), *Convolutional Neural Network (CNN)-VGG16* merupakan salah satu pendekatan pembelajaran mesin yang digunakan untuk mengklasifikasikan dan memprediksi perubahan fisik dalam gelembung gas hidrogen.
12. Jurnal internasional: “*Deep CNN Model based on VGG16 for Breast Cancer Classification*” berdasarkan penelitian (Albashish, Ryalat, dan Almansour t.t.) Dalam penelitian ini, model transfer learning berdasarkan *Visual Geometry Group* dengan arsitektur *deepmodel 16-layer (VGG16)* digunakan untuk mengekstrak fitur tingkat tinggi dari dataset *histopathological images benchmark BreakHis*.
13. Jurnal internasional: “*Particle Swarm Optimization Algorithm for Hyperparameter Convolutional Neural Network and Transfer Learning VGG16 Model*” berdasarkan penelitian (Winiarti dkk. 2024), Diperoleh akurasi tertinggi saat mengklasifikasikan gambar motif batik menggunakan model pembelajaran transfer *PSOCNN VGG16*, yaitu sebesar 83%. Tingkat akurasi yang meningkat jika dibandingkan dengan model CNN biasanya menunjukkan bahwa penggunaan *transfer learning* memberikan pengaruh yang baik terhadap tingkat akurasi yang diperoleh.

14. Jurnal internasional: “*Breast Cancer Detection Using Deep Learning*” berdasarkan penelitian Spandana dkk. (2023), Arsitektur jaringan saraf dalam yang disebut *NASNetMobile* dibuat untuk digunakan dalam aplikasi visi tertanam dan seluler.
15. Jurnal internasional: “*Classifying Surface Fault in Steel Strips Using a Customized NasNet-Mobile CNN and Small Dataset*” berdasarkan penelitian (Khebli dkk., 2024), *Neural Architecture Search (NAS)* merupakan teknologi yang mengotomatiskan pengembangan arsitektur jaringan saraf untuk mendapatkan hasil terbaik pada suatu proyek tertentu.
16. Jurnal internasional: “*Detection and Classification of Breast cancer from Ultrasound Images using NASNet Model*” berdasarkan penelitian (Sathishkumar dkk., 2024.), *Neural Architecture Search (NASNet)* adalah sebuah pendekatan dalam menciptakan Jaringan Saraf Konvolusional (*CNN*) dengan arsitektur seluler yang dioptimalkan untuk perangkat tertanam dan seluler. Ini merupakan hasil dari proses *Neural Architecture Search (NAS)*, di mana algoritma secara otomatis mengeksplorasi dan menemukan arsitektur jaringan saraf yang paling efektif berdasarkan kriteria dan batasan yang ditentukan, mengeliminasi kebutuhan akan spesialis manusia untuk merancang arsitektur tersebut.
17. Jurnal internasional: “*Robust Zero Watermarking Algorithm for Medical Images Based on Improved NasNet-Mobile and DCT*” berdasarkan penelitian (Dong dkk. 2023), *NasNet-Mobile* adalah arsitektur jaringan saraf ringan untuk mencapai tugas pengenalan gambar berperforma tinggi dan latensi rendah. Dikembangkan oleh tim Google Otak dan berdasarkan

teknologi *Neural Architecture Search (NAS)*, jaringan *NasNet-Mobile* arsitektur bertujuan untuk menjaga akurasi sekaligus mengurangi sumber daya komputasi secara signifikan kebutuhan dan konsumsi daya.

18. Jurnal internasional: “***UML Modeling and Black Box Testing Methods in the School Payment Information System***” berdasarkan penelitian .(Rambe dkk., 2020), Rekayasa perangkat lunak *Blackbox testing* adalah metode pengujian yang berurusan dengan hasil internal yang tidak diketahui. Penguji melihat perangkat lunak sebagai “*Blackbox*” yang tidak perlu menunjukkan isinya, tetapi tunduk pada pengujian eksternal. Dalam jenis pengujian kotak hitam ini, perangkat lunak akan dijalankan dan kemudian mencoba untuk menguji apakah memenuhi kebutuhan pengguna yang telah ditentukan di awal tanpa membongkar daftar program.
19. Jurnal internasional: “***MobileNetV2 Ensemble Segmentation for Mandibular on Panoramic Radiography***” berdasarkan penelitian (Nafi’iyah dkk. 2023) Pembelajaran mendalam *MobileNetV2* memiliki jumlah yang kecil parameter, tetapi hasil kinerjanya bagus. Arsitektur *MobileNetV2* dapat digunakan sebagai *autoencoder* untuk klasifikasi, segmentasi, atau deteksi.
20. Jurnal internasional: “***Key methodological considerations for usability testing of electronic patient-reported outcome (ePRO) systems***” berdasarkan penelitian .(Aiyegbusi, 2020), *MobileNetV2 Usability testing* adalah teknik yang melibatkan pengujian sistem, produk atau situs web dengan peserta yang diambil dari populasi target. *Usability testing* dapat membantu pengembang *ePRO* dalam evaluasi antarmuka pengguna *ePRO*.

Kompleksitas sistem *ePRO*; tahap pengembangan; metrik untuk mengukur; dan penggunaan skenario, moderator, dan ukuran sampel yang sesuai adalah masalah metodologis utama yang perlu dipertimbangkan ketika merencanakan *usability testing*

Tabel 1 Perbandingan peneliti penulis

NO	FITUR	PENELITIAN																								
		Kelompok 1		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20			
1	Uang rupiah	✓	✓	(Aprililia dkk. 2024)	✓	(Antara dkk. 2023)	✓	(Desi Handayani dkk. 2024)	✓	(Mardha dkk. 2022)	(Sri Rahmaadhani., dkk 2023),	(Rahman dkk., 2023)	(Malik Ibrahim dkk., 2023)	(Musa dkk., 2023)	(Eka dkk., 2023)	(Dwi Fajar Maulana dkk., 2021)	(Gapsari dkk., 2024)	(Albashish dkk., 2021)	(Winiarti dkk., 2024),	(Spandana dkk., 2023)	(Khebli dkk., 2024)	(Sathishkumar dkk., 2024)	(Manning et al., n.d.)	(Rambe dkk., 2020)	(Nafi'iyah dkk., 2023)	(Aiyegbusi., 2020)
2	Klasifikasi	✓																								
3	Evaluasi	✓																								
METODE																										
1	CNN VGG16	✓													✓	✓	✓	✓								
2	CNN NASNetMobile	✓															✓	✓	✓	✓						

3	CNN Xception	✓								✓	✓	✓												
4	CNN MobileNetV2	✓						✓															✓	✓
5	Blackbox Testing	✓																					✓	
6	Usability testing	✓																						✓
NO	TOOLS																							
1	Jupyter notebook	✓																						
2	python	✓							✓															

Keterangan :

- 1) PP : Kelompok 1
- 2) P1-P20 : Peneliti Terkait.

2.2. Literatur Terkait

2.2.1. Uang Kertas Rupiah

Uang merupakan isntrumen perekonomian yang berfungsi sebagai alat tukar atau alat bayar dalam suatu wilayah tertentu. Berdasarkan Peraturan Pemerintah (PP) Nomor: 06 tahun 2019, uang kertas dicetak oleh Lembaga Peruri sesuai pesanan dari Bank Indonesia. Uang kertas rupiah terbuat dari serat kapas yang dicetak dengan ciri khas tertentu setiap mata uang pecahannya. Bank Indonesia telah mengeluarkan 8(delapan) pecahan Uang Rupiah kertas yaitu pecahan Rp1.000, Rp2.000, Rp5.000, Rp10.000, Rp20.000, Rp50.000, Rp75.000 dan Rp100.000 (Mardha dkk. 2022)

2.2.2. *Machine Learning*

Machine Learning adalah domain yang menggunakan data masa lalu untuk memprediksi. *Machine Learning* adalah pemahaman tentang sistem komputer di mana *Machine Learning* belajar dari data dan pengalaman. Algoritma pembelajaran mesin Algoritma *machine learning* memiliki dua fase: 1) Pelatihan & 2) Pengujian. (Gomathy dkk., 2021)

2.2.3. *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) adalah jenis arsitektur jaringan saraf yang memiliki keahlian khusus dalam pengolahan citra. Dengan lapisan konvolusi dan pooling, CNN mampu mengenali pola, tekstur, dan fitur yang signifikan dalam citra. Keunggulan utama CNN adalah kemampuannya untuk melakukan ekstraksi fitur secara otomatis dari

data pelatihan, sehingga mengurangi ketergantungan pada fitur manual dan memungkinkan pembelajaran yang lebih mendalam. (Malik Ibrahim, Rahmadewi, dan Nurpulaela 2023b)

2.2.4. VGG16

VGG-16 merupakan model *Convolutional Neural Network*. VGG-16 adalah penyempurnaan dari *AlexNet*. Model *AlexNet* menggunakan ukuran kernel yang besar. Sedangkan VGG-16 menggunakan Kernel 3x3. Citra akan melalui tumpukan dari *Convolutional Layer*, dimana filter *convolutional layer* dan *3 fully connected layer*. Network ini adalah jaringan yang cukup besar dan memiliki sekitar 138 juta parameter. (Hartati dkk., 2022)

2.2.5. NASNetMobile

Arsitektur *NASNetMobile* merupakan sebuah arsitektur yang memiliki jumlah parameter yang lebih sedikit jika dibandingkan dengan arsitektur *MobileNet*. Ide awal arsitektur ini bermula dari penggunaan *framework Neural Architecture Search (NAS)* sebagai metode search untuk mencari arsitektur convolutional terbaik pada dataset kecil. Kemudian dengan adanya kontribusi desain *search space* baru yang dinamakan *NAsNet search space*, arsitektur tersebut ditransfer ke dataset yang lebih besar. Dari *NasNet search space* ditemukan arsitektur terbaik yang kemudian dinamakan *NASNet*. Pada arsitektur *NASNet* terdapat dua *convolutional cell* yang disebut *Normal Cell* dan *Reduction Cell*. *Normal Cell* berfungsi mengembalikan *feature map* dengan dimensi yang sama, sedangkan *Reduction Cell* berfungsi mengembalikan *feature map* di mana

tinggi dan lebar peta fitur dikurangi dengan faktor dua. (Sahro Winanto, Rozikin, dan Jamaludin 2023)

2.2.6. *Xception*

Xception merupakan arsitektur CNN yang diadaptasi dari arsitektur *Inception*. *Xception* menggunakan hipotesis dari *Inception* namun ditingkatkan lagi ke level yang ekstrim, hal ini merupakan asal dari penamaan *Xception* atau *Extreme Inception*. Arsitektur ini menggunakan *depth-wise separable convolutions* sebagai pengganti dari modul *Inception*. *Xception* terdiri dari 3 *flow*, yaitu *entry flow*, *middle flow*, dan *exit flow*. Data yang digunakan sebagai input nantinya akan masuk melalui *entry flow* terlebih dahulu kemudian melalui *middle flow* sebanyak 8 kali kemudian menuju ke *exit flow*. (Eka dkk. 2023b)

2.2.7. *MobileNetV2*

MobileNetV2 adalah salah satu arsitektur *convolutional neural network (CNN)* berbasis ponsel yang dapat digunakan untuk mengatasi kebutuhan akan *computing resource* berlebih. *MobileNetV2* merupakan penyempurnaan dari arsitektur *MobileNet*. Arsitektur *MobileNet* dan arsitektur CNN pada umumnya memiliki perbedaan pada penggunaan lapisan atau *convolution layer*. *Convolution layer* pada *MobileNetV2* menggunakan ketebalan filter yang sesuai dengan ketebalan dari input image. *MobileNetV2* menggunakan *depthwise convolution*, *pointwise convolution*, *linear bottleneck* dan *shortcut connections* antar *bottlenecks*.

(Hikmatia dan Zul 2021)

2.2.8. *Python*

Python merupakan salah satu perangkat lunak yang sedang populer saat ini. Dengan Python, kita memiliki kemampuan untuk melakukan analisis data, menjalankan perhitungan data statistik yang kompleks atau memakan waktu, membuat visualisasi data, mengimplementasikan algoritma machine learning, dan juga dapat digunakan untuk manipulasi data serta menyelesaikan berbagai tugas matematika lainnya. Keunggulan Python terletak pada kemampuannya menghasilkan hasil yang lebih akurat dan efisien dibandingkan dengan metode manual. (Rahman dkk., 2023)

2.2.9. *Jupyter Notebook*

Jupyter Notebook merupakan perangkat lunak berbasis kode terbuka yang digunakan menggunakan browser yang berfungsi sebagai *notebook lab* virtual untuk mendukung alur kerja, kode, data, dan visualisasi. (Dwi Fajar Maulana dkk., 2021)

2.2.10. *HTML*

Hypertext Markup Language (HTML). Disebut *Hypertext* karena di dalam *script* HTML kita bisa membuat sebuah teks menjadi link yang bisa menavigasikan user ke halaman lain dengan meng-klik teks tersebut. Teks yang ber-link inilah yang disebut *Hypertext* karena hakikat sebuah website adalah dokumen yang mengandung banyak link untuk menghubungkan satu dokumen dengan yang lainnya. Disebut *Markup Language* karena *script* HTML menggunakan tanda (dalam bahasa inggris disebut “*Mark*”) untuk menandai bagian-bagian dari teks agar teks itu memiliki tampilan/fungsi tertentu. Dalam praktiknya tanda atau mark disebut dengan istilah “*tag*”

.HTML itu sendiri merupakan bahasa dasar pembuatan web. HTML menggunakan tanda atau mark, untuk menandai bagian-bagian dari text. HTML disebut sebagai bahasa dasar, karena dalam membuat web, jika hanya menggunakan HTML maka tampilan web terasa hambar. (Saputra dkk. 2023)

2.2.11. CSS

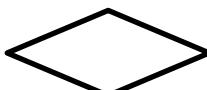
Cascading Style Sheet (CSS) merupakan aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemograman. Sama halnya *styles* dalam aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa style, misalnya heading, subbab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file) (Saputra dkk. 2023)

2.2.12. Pengertian *FlowCharts*

Flowchart adalah representasi dalam bentuk diagram dari alur pemecahan sebuah masalah atau algoritma. *Flowchart* merupakan representasi dari algoritma yang statefull, karena setiap langkah diwakili oleh satu bentuk diagram. *Flowchart* merupakan salah satu bentuk diagram yang paling sederhana dalam menuliskan algoritma, oleh karena itu tidak semua algoritma dapat dituliskan dalam bentuk *flowchart*.

Tabel 2 *Flowchart*

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program

	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/pemberian harga awal
	PROCESS	Proses perhitungan/proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Sumber: (Zalukhu dkk. 2023)

2.2.13. Metode pengujian sistem

Pengujian sistem adalah pengujian berdasar spesifikasi atau kebutuhan perangkat lunak. Pengujian ini biasanya dilakukan berdasarkan spesifikasi yang dianalisa secara informal dan manual. Metode pengujian sistem yang kami gunakan pada program ini adalah *blackbox testing* dan *usability testing*.

2.2.14. Blackbox Testing

Rekayasa perangkat lunak *Blackbox testing* adalah metode pengujian yang berurusan dengan hasil internal yang tidak diketahui.

Penguji melihat perangkat lunak sebagai "Blackbox" yang tidak perlu menunjukkan isinya, tetapi tunduk pada pengujian eksternal. Dalam jenis pengujian *Blackbox* ini, perangkat lunak akan dijalankan dan kemudian mencoba untuk menguji apakah memenuhi kebutuhan pengguna yang telah ditentukan di awal tanpa membongkar daftar program.(Rambe dkk. 2020)

2.2.15. *Usability Testing*

Usability testing adalah teknik yang melibatkan pengujian sistem, produk atau situs web dengan peserta yang diambil dari populasi target. *Usability testing* dapat membantu pengembang *ePRO* dalam evaluasi antarmuka pengguna *ePRO*. Kompleksitas sistem *ePRO*; tahap pengembangan; metrik untuk mengukur; dan penggunaan skenario, moderator, dan ukuran sampel yang sesuai adalah masalah metodologis utama yang perlu dipertimbangkan ketika merencanakan *usability testing*.(Aiyegbusi 2020)

BAB III

ANALISA DATA DAN PEMBAHASAN

3.1. Analisa Data

Dataset terdiri dari 1.050 citra uang kertas, masing-masing 150 citra dari 7 denominasi: Rp1.000, Rp2.000, Rp5.000, Rp10.000, Rp20.000, Rp50.000, dan Rp100.000.

Tabel 3 Dataset

NO	CITRA	LABEL/CLASS	JUMLAH
1		Rp1.000	150
2		Rp2.000	150
3		Rp5.000	150
4		Rp10.000	150
5		Rp20.000	150

6		Rp50.000	150
7		Rp100.000	150

Dari tabel diatas terdapat 150 citra pada tiap kelas, jumlah totalnya adalah 1.050 citra. citra dibagi menjadi data *train*, *validation*, dan *test* dengan komposisi di setiap percobaan dengan komposisi seperti berikut :

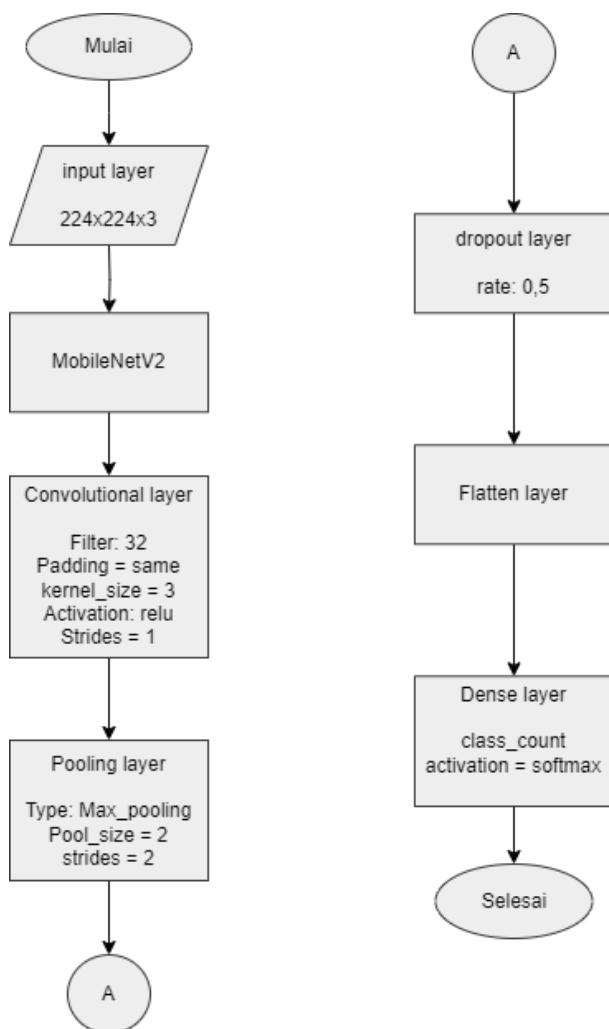
Tabel 4 Komposisi model

No	Nama	Training	Validation	test
1	Percobaan A	80%	10%	10%
2	Percobaan B	80%	10%	10%
3	Percobaan C	80%	10%	10%
4	Percobaan D	80%	10%	10%

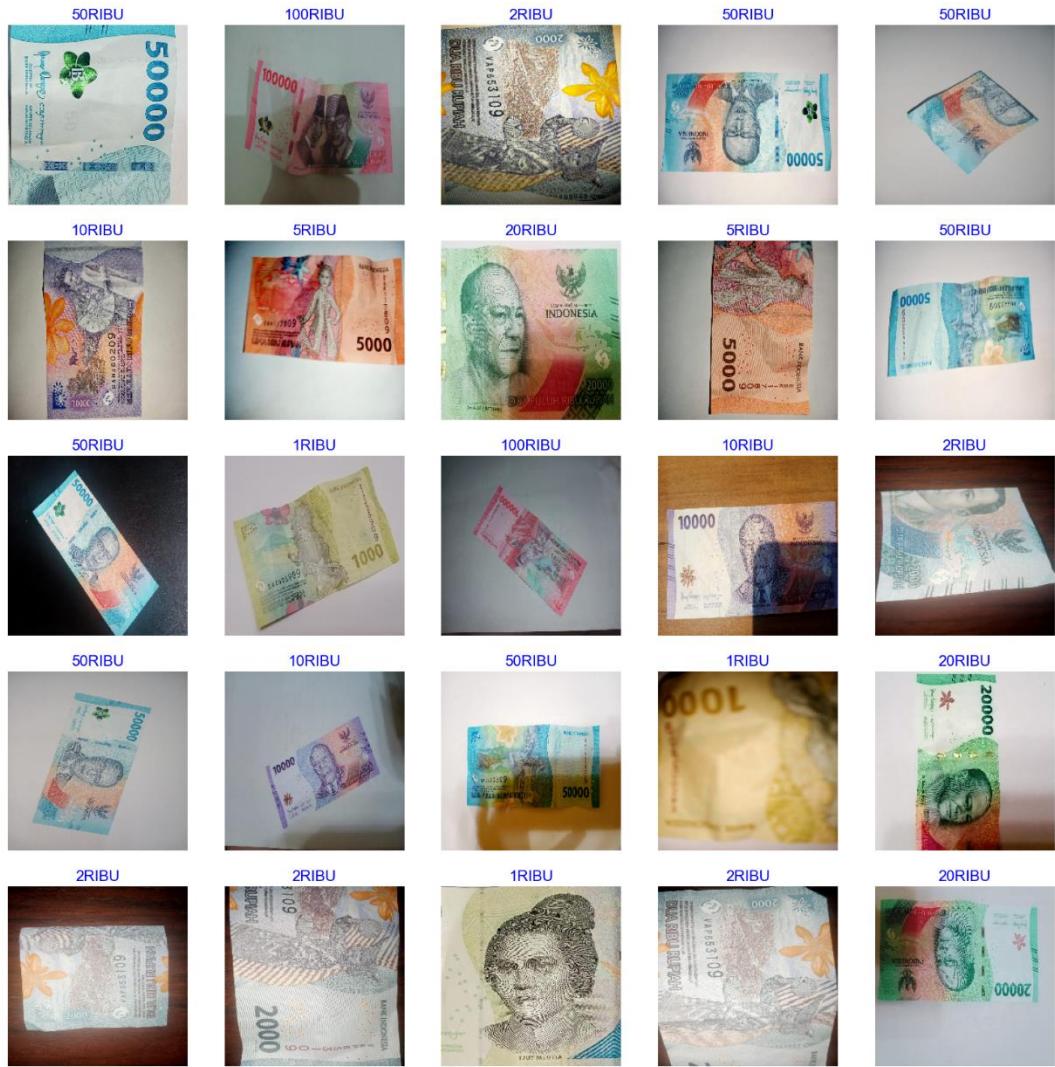
Hasil Pembentukan Model Pada tahapan ini, data diolah menggunakan metode klasifikasi dengan algoritma *Convolutional Neural Network* (CNN) dengan arsitektur yang berbeda pada setiap percobaan.

3.1.1. Percobaan A: Hasil pembentukan model CNN *MobileNetV2*

Hasil dari percobaan A dengan inputan 224x224x3 maka dibentuklah arsitektur dengan total 6 layer. Dilakukan 1 kali tahapan convolution dan pooling, kemudian masuk ke flatten layer untuk mengubahnya menjadi matriks satu dimensi sehingga bisa menjadi inputan pada *fully connected layer*. Adapun tahapan dari arsitektur seperti gambar



Gambar 2 Flowchart pembentukan model CNN MobileNetV2



Gambar 3 Hasil deteksi model CNN *MobileNetv2*

Gambar 3 diatas menunjukkan berbagai uang kertas dengan nilai nominal berbeda dalam mata uang rupiah (IDR), yang telah diidentifikasi oleh model CNN *MobileNetv2*. Setiap gambar diberi label sesuai dengan nilai nominalnya, termasuk 1RIBU, 2RIBU, 5RIBU, 10RIBU, 20RIBU, 50RIBU, dan 100RIBU. Uang kertas ditampilkan dalam berbagai posisi dan sudut pandang, menunjukkan variasi visual yang berbeda-beda. Model CNN *MobileNetV2* telah berhasil mendeteksi dan mengklasifikasikan nilai nominal

dari setiap uang kertas berdasarkan fitur visual yang dikenali dari masing-masing uang kertas.

```

Model: "sequential"
...


| Layer (type)                      | Output Shape | Param #     |
|-----------------------------------|--------------|-------------|
| mobilenetv2_1.00_224 (Functional) | ?            | 2,257,984   |
| conv2d (Conv2D)                   | ?            | 0 (unbuilt) |
| max_pooling2d (MaxPooling2D)      | ?            | 0 (unbuilt) |
| dropout (Dropout)                 | ?            | 0 (unbuilt) |
| flatten (Flatten)                 | ?            | 0 (unbuilt) |
| dense (Dense)                     | ?            | 0 (unbuilt) |


...
Total params: 2,257,984 (8.61 MB)
...
Trainable params: 0 (0.00 B)
...
Non-trainable params: 2,257,984 (8.61 MB)

```

Gambar 4 Model Summary Percobaan A

Arsitektur pada gambar 4 diatas dibuat dengan hyperparameter sebagai berikut, proses trial dan error pada nilai hyperparameter membantu menemukan arsitektur dengan akurasi terbaik. Pada percobaan A hyperparameter-nya sebagai berikut:

Tabel 5 Tabel Hyperparameter Percobaan A

Hyperparameter	
Split data	80% Train, 10% valid, 10% Test
Bach size	64
Layers	6

Epoch	25
Optimizer	Adam
Learning rate	0.001

Pada tabel diatas data dibagi menjadi 80% Train, 10% Valid, 10% Test dimana menggunakan batch size atau sampel data yang disebarluaskan ke *Neural Network* sebanyak 64, yang berarti jika kita memiliki 1.050 dataset maka algoritma CNN akan menggunakan 64 sampel data pertama dari 1.050 data yang kita miliki lalu disebarluaskan atau training oleh *Neural Network* sampai selesai kemudian mengambil kembali 64 sampel selanjutnya dari 1.050 data, begitu seterusnya. Sedangkan *layers* 6 artinya arsitektur terdiri dari 6 *layer* yaitu *layer konvolusi*, *layer maxpooling*, *layer flatten*, dan *layer fully connected*. *Epoch* 25 berarti proses *training* dilakukan sebanyak 25 kali iterasi atau 25 kali perulangan. Berikutnya ada *Optimizer Adam* mengatur learning rate berdasarkan besaran nilai rata-rata dari *weight*, dan yang terakhir kita menggunakan *learning rate* 0.001 dimana semakin kecil suatu *learning rate* maka model akan mempelajari data *training* semakin detail.

Percobaan A dilakukan dengan 25 *epoch* atau 25 kali iterasi, pelatihan model ini menghasilkan akurasi 89,52% dengan *loss*, *accuracy* seperti pada gambar berikut :

```

Epoch 1/25
c:\Users\USER\anaconda3\envs\cnn\lib\site-packages\keras\src\trainers\data_adapters\py_dataset_adapter.py:121: UserWarning: self._warn_if_super_not_called()
14/14    44s 2s/step - accuracy: 0.2104 - loss: 4.1383 - val_accuracy: 0.2571 - val_loss: 1.8042
Epoch 2/25
14/14    33s 2s/step - accuracy: 0.3093 - loss: 1.7258 - val_accuracy: 0.3714 - val_loss: 1.5991
Epoch 3/25
14/14    32s 2s/step - accuracy: 0.4248 - loss: 1.4462 - val_accuracy: 0.5333 - val_loss: 1.3346
Epoch 4/25
14/14    32s 2s/step - accuracy: 0.5215 - loss: 1.2788 - val_accuracy: 0.6095 - val_loss: 1.1986
Epoch 5/25
14/14    32s 2s/step - accuracy: 0.6787 - loss: 0.9638 - val_accuracy: 0.6762 - val_loss: 0.9559
Epoch 6/25
14/14    33s 2s/step - accuracy: 0.6938 - loss: 0.8556 - val_accuracy: 0.6952 - val_loss: 0.8943
Epoch 7/25
14/14    33s 2s/step - accuracy: 0.7673 - loss: 0.6311 - val_accuracy: 0.7429 - val_loss: 0.7636
Epoch 8/25
14/14    33s 2s/step - accuracy: 0.8467 - loss: 0.4515 - val_accuracy: 0.7714 - val_loss: 0.7143
Epoch 9/25
14/14    33s 2s/step - accuracy: 0.8507 - loss: 0.4703 - val_accuracy: 0.7714 - val_loss: 0.6214
Epoch 10/25
14/14    33s 2s/step - accuracy: 0.8585 - loss: 0.3959 - val_accuracy: 0.8190 - val_loss: 0.5716
Epoch 11/25
14/14    32s 2s/step - accuracy: 0.9067 - loss: 0.3000 - val_accuracy: 0.8381 - val_loss: 0.4989
Epoch 12/25
14/14    32s 2s/step - accuracy: 0.9070 - loss: 0.2409 - val_accuracy: 0.8286 - val_loss: 0.5650
Epoch 13/25
14/14    32s 2s/step - accuracy: 0.9345 - loss: 0.2210 - val_accuracy: 0.8762 - val_loss: 0.4607

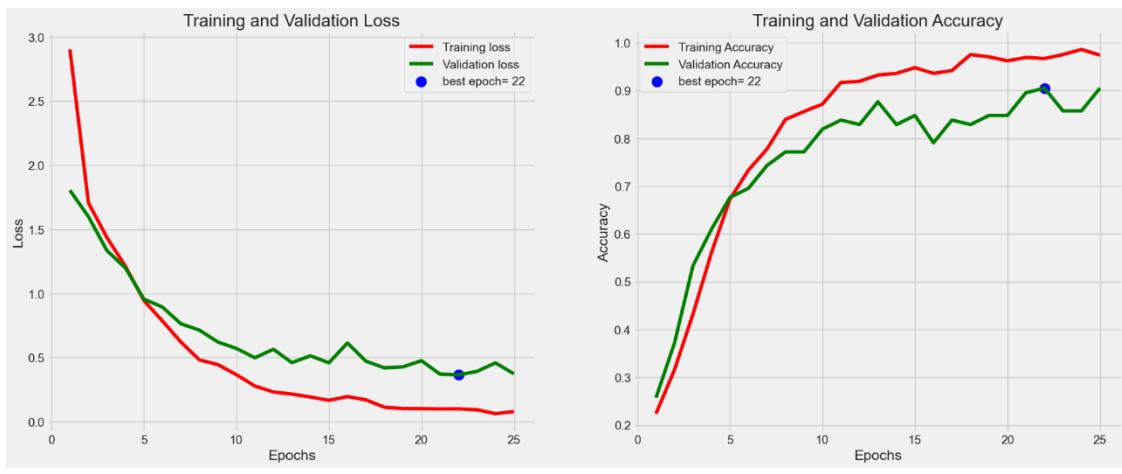
Epoch 14/25
14/14    33s 2s/step - accuracy: 0.9431 - loss: 0.1709 - val_accuracy: 0.8286 - val_loss: 0.5145
Epoch 15/25
14/14    33s 2s/step - accuracy: 0.9552 - loss: 0.1703 - val_accuracy: 0.8476 - val_loss: 0.4597
Epoch 16/25
14/14    32s 2s/step - accuracy: 0.9591 - loss: 0.1402 - val_accuracy: 0.7905 - val_loss: 0.6141
Epoch 17/25
14/14    34s 2s/step - accuracy: 0.9358 - loss: 0.1842 - val_accuracy: 0.8381 - val_loss: 0.4725
Epoch 18/25
14/14    34s 2s/step - accuracy: 0.9690 - loss: 0.1227 - val_accuracy: 0.8286 - val_loss: 0.4204
Epoch 19/25
14/14    33s 2s/step - accuracy: 0.9772 - loss: 0.0874 - val_accuracy: 0.8476 - val_loss: 0.4278
Epoch 20/25
14/14    33s 2s/step - accuracy: 0.9541 - loss: 0.1119 - val_accuracy: 0.8476 - val_loss: 0.4762
Epoch 21/25
14/14    32s 2s/step - accuracy: 0.9738 - loss: 0.0958 - val_accuracy: 0.8952 - val_loss: 0.3720
Epoch 22/25
14/14    32s 2s/step - accuracy: 0.9714 - loss: 0.0998 - val_accuracy: 0.9048 - val_loss: 0.3650
Epoch 23/25
14/14    32s 2s/step - accuracy: 0.9800 - loss: 0.0821 - val_accuracy: 0.8571 - val_loss: 0.3927
Epoch 24/25
14/14    32s 2s/step - accuracy: 0.9871 - loss: 0.0707 - val_accuracy: 0.8571 - val_loss: 0.4591
Epoch 25/25
14/14    32s 2s/step - accuracy: 0.9730 - loss: 0.0755 - val_accuracy: 0.9048 - val_loss: 0.3740

```

Gambar 5 Percobaan A dilakukan dengan 25 epoch atau 25 kali iterasi

Nilai pada gambar akan menghasilkan grafik seperti pada gambar

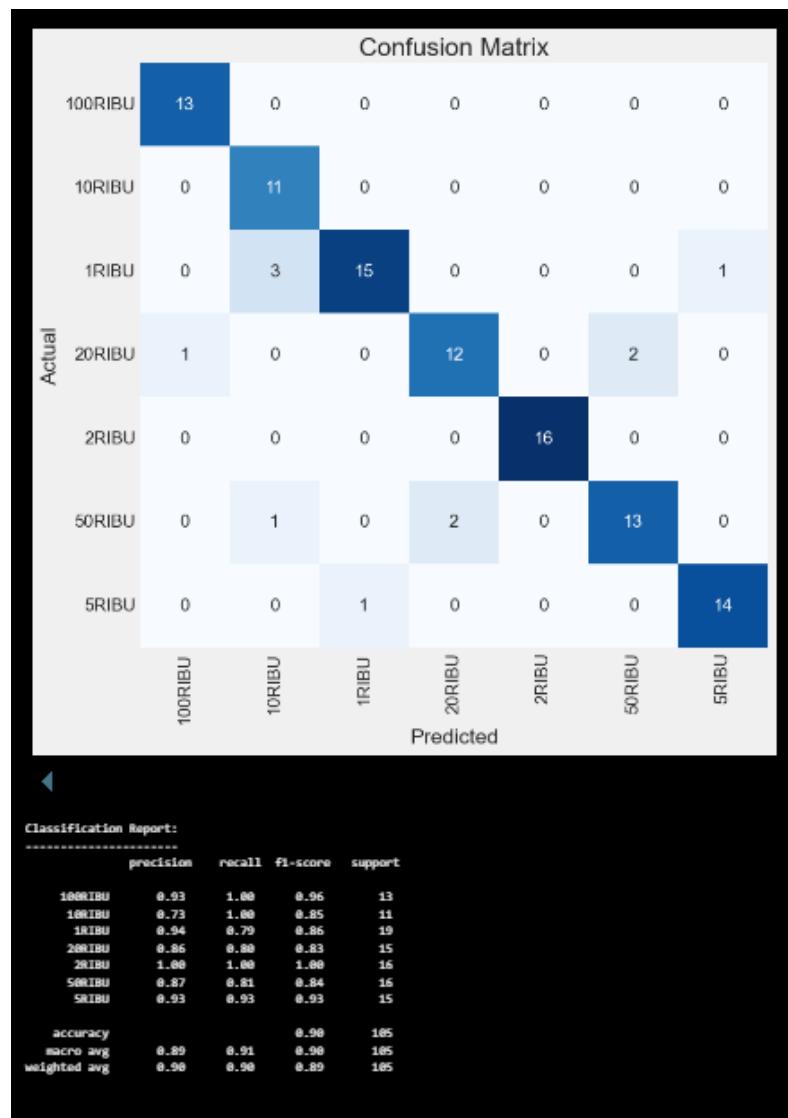
berikut:



Gambar 6 Grafik Percobaan A

Gambar diatas menunjukan model mempelajarai data *training* secara perlahan dan mendetail ini bisa dilihat dari bentuk kurva yang mana untuk *training* dan *validation loss* perlahan turun yang mengukur seberapa bagus performa yang dihasilkan model pembelajaran dalam melakukan prediksi terhadap target dan untuk *training* dan *validation accuracy* perlahan naik yang menunjukan performa akurasinya semakin meningkat.

Untuk memastikan model perlu dilakukan pelatihan ulang atau tidak dapat dilakukan pengujian model dengan menggunakan confusion matrix yang mana output dari pengujian ini sendiri adalah nilai akurasi, presisi, recall dan hasil f1. Berikut hasil pengujian model disetiap percobaan sebagai berikut:



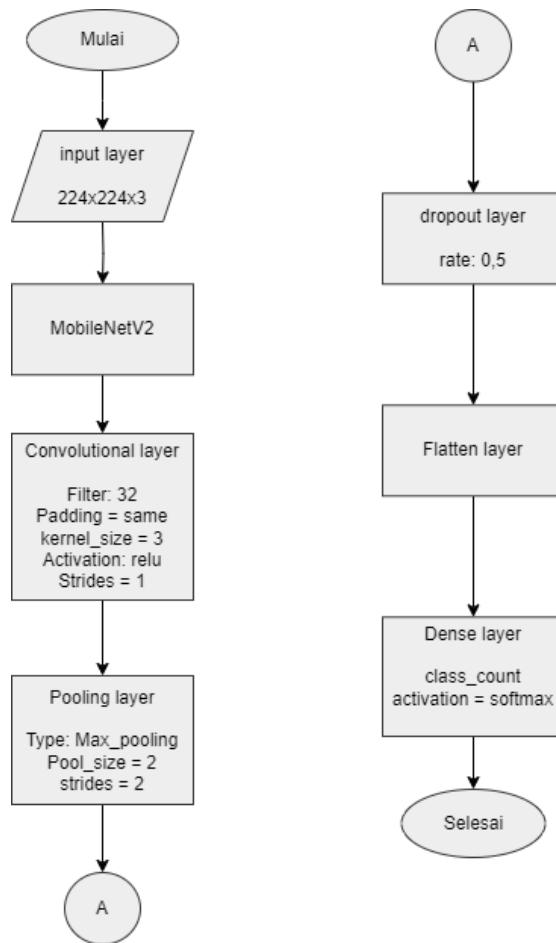
Gambar 7 *Confusion Matrix* Percobaan A

Confusion matrix menunjukkan performa model dalam memprediksi kelas-kelas, dengan hasil yang cukup baik secara keseluruhan. model berhasil mengklasifikasikan sebagian besar dengan benar, menghasilkan akurasi sebesar 0.89. Precision dan recall juga menunjukkan hasil yang memadai, terutama pada kelas 100RIBU (1.00), 2RIBU (0.94), dan 1RIBU (0.88). Namun, terdapat beberapa kesalahan klasifikasi pada kelas 10RIBU dan 50RIBU. Nilai F1-Score rata-rata menunjukkan model seimbang dalam hal precision dan recall,

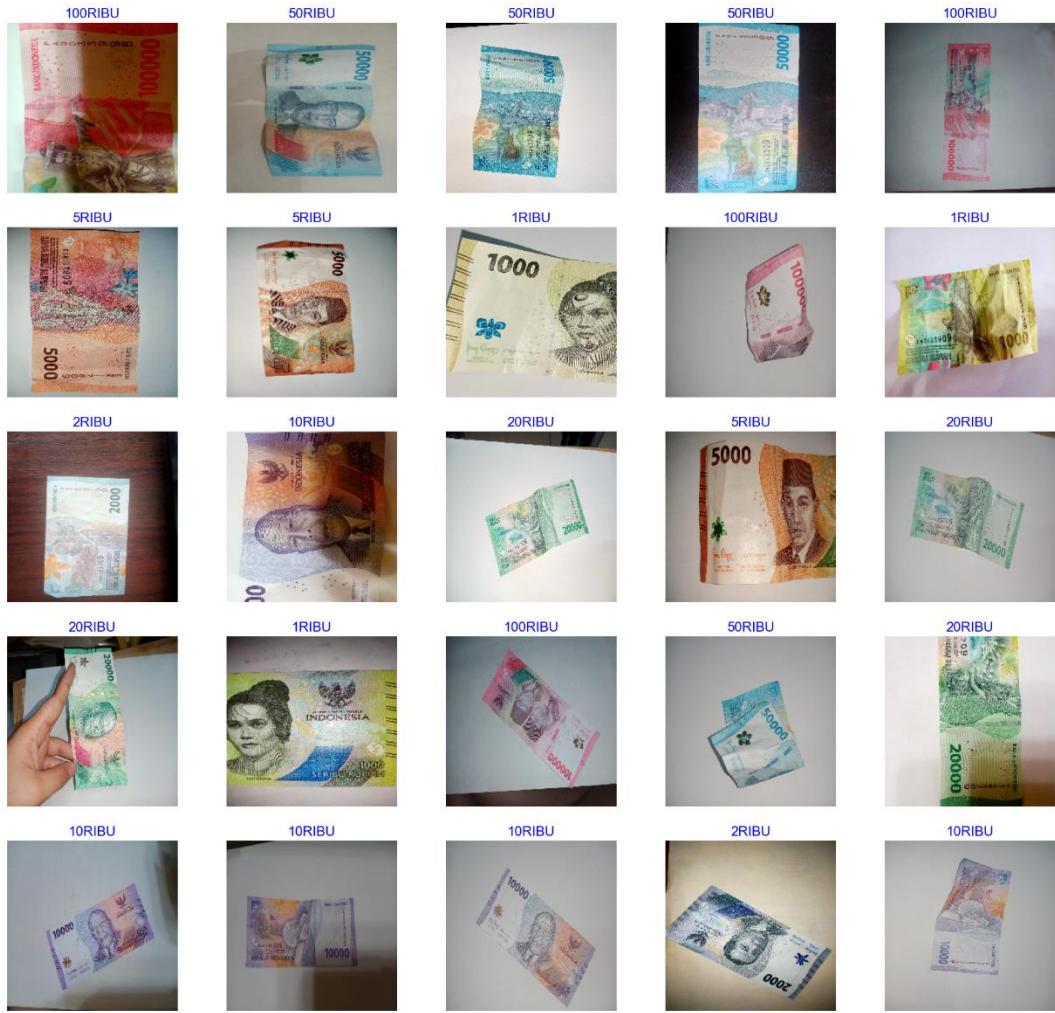
meskipun masih ada ruang untuk perbaikan terutama pada kelas dengan hasil yang kurang optimal. Secara keseluruhan, model menunjukkan kinerja yang baik dan dapat diandalkan dalam mengklasifikasikan sampel ke dalam kelas yang benar.

3.1.2. Percobaan B : Hasil pembentukan model CNN VGG16

Hasil dari percobaan B dengan inputan 224x224x3 maka dibentuklah arsitektur dengan total 6 layer. Dilakukan 1 kali tahapan convolution dan pooling, kemudian masuk ke *flatten layer* untuk mengubahnya menjadi matriks satu dimensi sehingga bisa menjadi inputan pada *fully connected layer*. Adapun tahapan dari arsitektur seperti gambar.



Gambar 8 Flowchart pembentukan model CNN VGG16



Gambar 9 Hasil deteksi model CNN *VGG16*

Gambar 9 diatas menunjukkan berbagai uang kertas dengan nilai nominal berbeda dalam mata uang rupiah (IDR), yang telah diidentifikasi oleh model CNN *VGG16*. Setiap gambar diberi label sesuai dengan nilai nominalnya, termasuk 1RIBU, 2RIBU, 5RIBU, 10RIBU, 20RIBU, 50RIBU, dan 100RIBU. Uang kertas ditampilkan dalam berbagai posisi dan sudut pandang, menunjukkan variasi visual yang berbeda-beda. Model CNN *VGG16* telah berhasil mendeteksi dan mengklasifikasikan nilai nominal dari setiap uang kertas berdasarkan fitur visual yang dikenali dari masing-masing uang kertas.

```

Model: "sequential_1"
...


| Layer (type)                   | Output Shape | Param #     |
|--------------------------------|--------------|-------------|
| vgg16 (Functional)             | ?            | 14,714,688  |
| conv2d_1 (Conv2D)              | ?            | 0 (unbuilt) |
| max_pooling2d_1 (MaxPooling2D) | ?            | 0 (unbuilt) |
| dropout_1 (Dropout)            | ?            | 0 (unbuilt) |
| flatten_1 (Flatten)            | ?            | 0 (unbuilt) |
| dense_1 (Dense)                | ?            | 0 (unbuilt) |


...
Total params: 14,714,688 (56.13 MB)
...
Trainable params: 0 (0.00 B)
...
Non-trainable params: 14,714,688 (56.13 MB)

```

Gambar 10 Model Summary Percobaan B

Arsitektur pada gambar 7 diatas dibuat dengan *Hyperparameter* sebagai berikut peroses *trial* dan *eror* pada nilai *Hyperparameter* membantu menemukan arsitektur dan akurasi terbaik pada percobaan B *Hyperparameter*-nya sebagai berikut :

Tabel 6 Tabel *Hyperparameter* Percobaan B

Hyperparameter	
Split data	80% Train, 10% valid, 10% Test
Bach size	64

Layers	6
Epoch	25
Optimizer	Adam
Learning rate	0.001

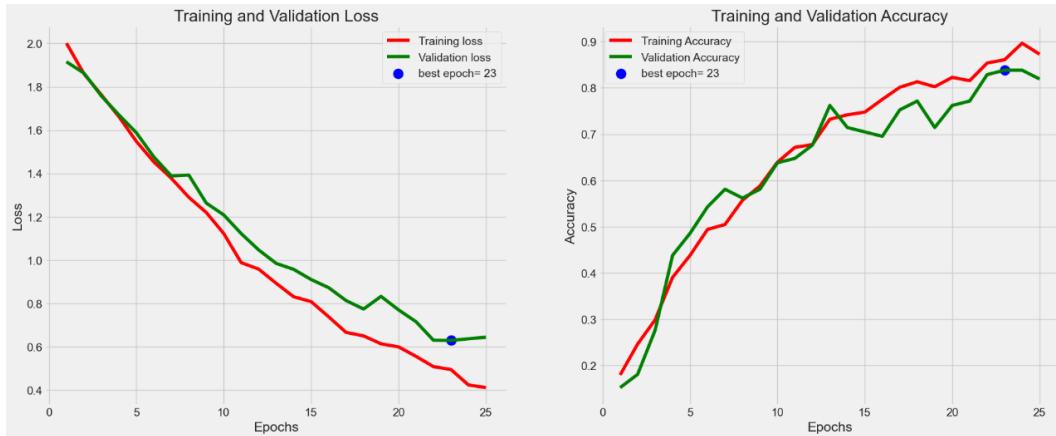
Pada tabel diatas data dibagi menjadi 80% Train, 10% Valid, 10% Test dimana menggunakan batch size atau sampel data yang disebarluaskan ke Neural Network sebanyak 64, yang berarti jika kita memiliki 1.050 dataset maka algoritma CNN akan menggunakan 64 sampel data pertama dari 1.050 data yang kita miliki lalu disebarluaskan atau training oleh Neural Network sampai selesai kemudian mengambil kembali 64 sampel selanjutnya dari 1.050 data, begitu seterusnya. Sedangkan layers 6 artinya arsitektur terdiri dari 6 layer yaitu *layer konvolusi*, *layer maxpooling*, *layer konvolusi*, *layer maxpooling*, *layer konvolusi*, *layer maxpooling*, *layer konvolusi*, *layer maxpooling*, *layer flatten*, dan *layer fully connected*. Epoch 25 berarti proses training dilakukan sebanyak 25 kali iterasi atau 25 kali perulangan. Berikutnya ada Optimizer Adam mengatur learning rate berdasarkan besaran nilai rata-rata dari weight, dan yang terakhir kita menggunakan learning rate 0.001 dimana semakin kecil suatu learning rate maka model akan mempelajari data training semakin detail.

Percobaan B dilakukan dengan 25 epoch atau 25 kali iterasi, pelatihan model ini menghasilkan akurasi 87,61% dengan loss, accuracy seperti pada gambar berikut:

14/14	183s	12s/step - accuracy: 0.1684 - loss: 2.0975 - val_accuracy: 0.1524 - val_loss: 1.9153
Epoch 2/25		
14/14	246s	17s/step - accuracy: 0.2241 - loss: 1.8916 - val_accuracy: 0.1810 - val_loss: 1.8628
Epoch 3/25		
14/14	245s	17s/step - accuracy: 0.2916 - loss: 1.7765 - val_accuracy: 0.2762 - val_loss: 1.7573
Epoch 4/25		
14/14	245s	17s/step - accuracy: 0.3642 - loss: 1.6883 - val_accuracy: 0.4381 - val_loss: 1.6701
Epoch 5/25		
14/14	241s	17s/step - accuracy: 0.4249 - loss: 1.5638 - val_accuracy: 0.4857 - val_loss: 1.5873
Epoch 6/25		
14/14	196s	13s/step - accuracy: 0.5025 - loss: 1.4557 - val_accuracy: 0.5429 - val_loss: 1.4758
Epoch 7/25		
14/14	142s	10s/step - accuracy: 0.5015 - loss: 1.4004 - val_accuracy: 0.5810 - val_loss: 1.3879
Epoch 8/25		
14/14	149s	10s/step - accuracy: 0.5652 - loss: 1.2961 - val_accuracy: 0.5619 - val_loss: 1.3921
Epoch 9/25		
14/14	144s	10s/step - accuracy: 0.5765 - loss: 1.2235 - val_accuracy: 0.5810 - val_loss: 1.2626
Epoch 10/25		
14/14	143s	10s/step - accuracy: 0.6243 - loss: 1.1571 - val_accuracy: 0.6381 - val_loss: 1.2078
Epoch 11/25		
14/14	144s	10s/step - accuracy: 0.6589 - loss: 0.9965 - val_accuracy: 0.6476 - val_loss: 1.1215
Epoch 12/25		
14/14	145s	10s/step - accuracy: 0.6831 - loss: 0.9473 - val_accuracy: 0.6762 - val_loss: 1.0460
Epoch 13/25		
14/14	144s	10s/step - accuracy: 0.7395 - loss: 0.9165 - val_accuracy: 0.7619 - val_loss: 0.9839
Epoch 14/25		
14/14	145s	10s/step - accuracy: 0.7472 - loss: 0.8238 - val_accuracy: 0.7143 - val_loss: 0.9563
Epoch 15/25		
14/14	146s	10s/step - accuracy: 0.7381 - loss: 0.8134 - val_accuracy: 0.7048 - val_loss: 0.9094
Epoch 16/25		
14/14	143s	10s/step - accuracy: 0.7660 - loss: 0.7403 - val_accuracy: 0.6952 - val_loss: 0.8718
Epoch 17/25		
14/14	145s	10s/step - accuracy: 0.8070 - loss: 0.6440 - val_accuracy: 0.7524 - val_loss: 0.8122
Epoch 18/25		
14/14	144s	10s/step - accuracy: 0.8305 - loss: 0.6287 - val_accuracy: 0.7714 - val_loss: 0.7730
Epoch 19/25		
14/14	144s	10s/step - accuracy: 0.8127 - loss: 0.6019 - val_accuracy: 0.7143 - val_loss: 0.8315
Epoch 20/25		
14/14	143s	10s/step - accuracy: 0.8189 - loss: 0.6273 - val_accuracy: 0.7619 - val_loss: 0.7693
Epoch 21/25		
14/14	147s	10s/step - accuracy: 0.8296 - loss: 0.5378 - val_accuracy: 0.7714 - val_loss: 0.7146
Epoch 22/25		
14/14	144s	10s/step - accuracy: 0.8470 - loss: 0.5052 - val_accuracy: 0.8286 - val_loss: 0.6289
Epoch 23/25		
14/14	145s	10s/step - accuracy: 0.8600 - loss: 0.5035 - val_accuracy: 0.8381 - val_loss: 0.6278
Epoch 24/25		
14/14	145s	10s/step - accuracy: 0.8846 - loss: 0.4573 - val_accuracy: 0.8381 - val_loss: 0.6357
Epoch 25/25		
14/14	144s	10s/step - accuracy: 0.8660 - loss: 0.4216 - val_accuracy: 0.8190 - val_loss: 0.6430

Gambar 11 Percobaan B dilakukan dengan 25 epoch atau 25 kali iterasi

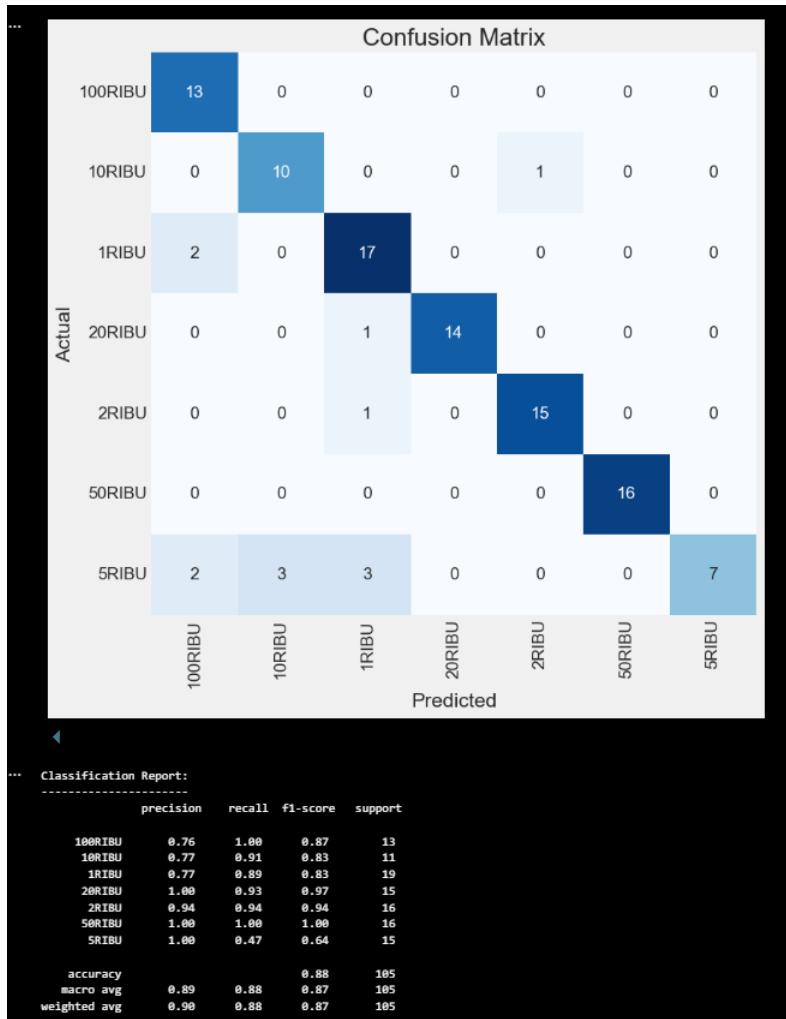
Nilai pada gambar 8 akan menghasilkan grafik seperti pada gambar berikut :



Gambar 12 Grafik Percobaan B

Gambar diatas menunjukan model mempelajarai data training secara perlahan dan mendetail ini bisa dilihat dari bentuk kurva yang mana untuk training dan validation loss perlahan turun yang mengukur seberapa bagus performa yang dihasilkan model pembelajaran dalam melakukan prediksi terhadap target dan untuk training dan validation accuracy perlahan naik yang menunjukan performa akurasinya semakin meningkat.

Untuk memastikan model perlu dilakukan pelatihan ulang atau tidak dapat dilakukan pengujian model dengan menggunakan confusion matrix yang mana output dari pengujian ini sendiri adalah nilai akurasi, presisi, recall dan hasil f1. Berikut hasil pengujian model disetiap percobaan sebagai berikut:



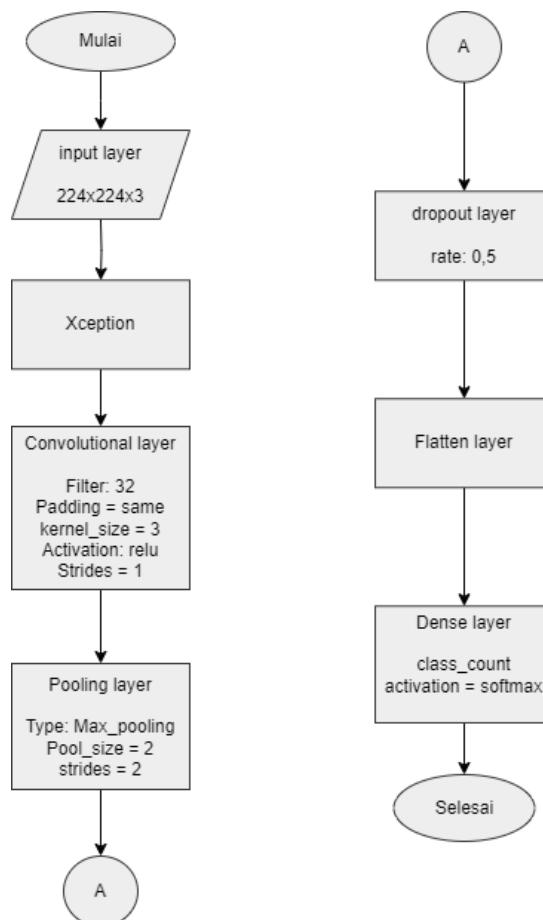
Gambar 13 Confusion Matriks Percobaan B

Confusion matrix ini menunjukkan performa model dalam mengklasifikasikan sampel ke dalam beberapa kelas, dengan akurasi keseluruhan sebesar 0.88. Model berhasil mengklasifikasikan sebagian besar dengan benar, namun terdapat beberapa kesalahan klasifikasi terutama pada kelas 10RIBU, 1RIBU, dan 5RIBU. Kelas 100RIBU dan 50RIBU memiliki precision dan recall yang tinggi (masing-masing 0.76 dan 1.00), menunjukkan ketepatan prediksi yang baik. Kelas 2RIBU dan 5RIBU memiliki precision yang baik, tetapi kelas 5RIBU menunjukkan penurunan signifikan pada recall (0.47).

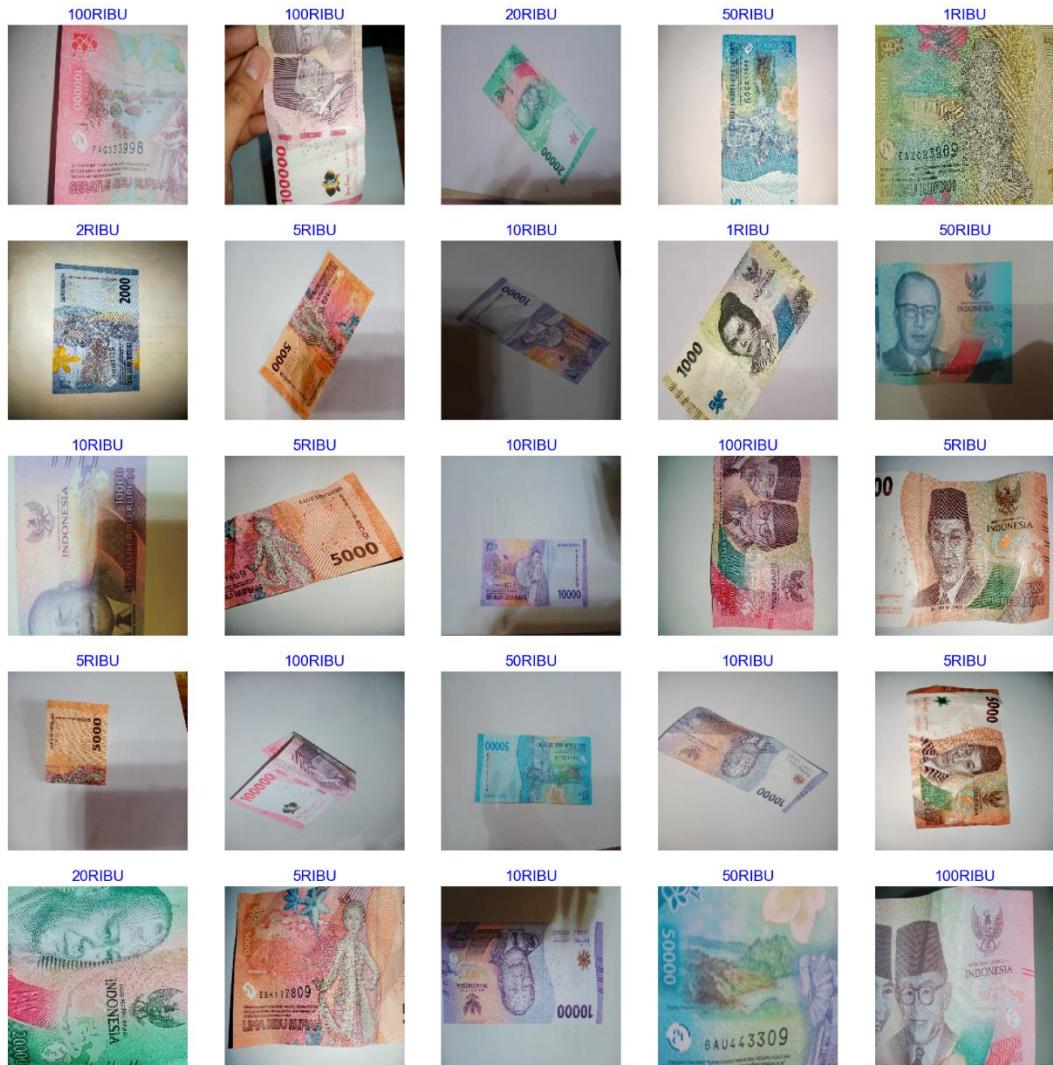
Nilai F1-Score rata-rata menunjukkan keseimbangan antara precision dan recall, meskipun ada ruang untuk perbaikan terutama pada kelas-kelas yang memiliki performa lebih rendah. Secara keseluruhan, model cukup andal namun memerlukan peningkatan lebih lanjut untuk mengatasi kesalahan pada beberapa kelas.

3.1.3. Percobaan C : Hasil pembentukan model CNN Xception

Hasil dari percobaan C dengan inputan 224x224x3 maka dibentuklah arsitektur dengan total 6 layer. Dilakukan 1 kali tahapan convolution dan pooling, kemudian masuk ke flatten layer untuk mengubahnya menjadi matriks satu dimensi sehingga bisa menjadi inputan pada *fully connected layer*. Adapun tahapan dari arsitektur seperti gambar.



Gambar 14 pembentukan model CNN Xception



Gambar 15 Hasil deteksi model CNN *Xception*

Gambar 15 diatas menunjukkan berbagai uang kertas dengan nilai nominal berbeda dalam mata uang rupiah (IDR), yang telah diidentifikasi oleh model CNN *Xception*. Setiap gambar diberi label sesuai dengan nilai nominalnya, termasuk 1RIBU, 2RIBU, 5RIBU, 10RIBU, 20RIBU, 50RIBU, dan 100RIBU. Uang kertas ditampilkan dalam berbagai posisi dan sudut pandang, menunjukkan variasi visual yang berbeda-beda. Model CNN *Xception* telah berhasil mendeteksi dan mengklasifikasikan nilai nominal dari setiap uang kertas berdasarkan fitur visual yang dikenali dari masing-masing uang kertas.

```

...
Model: "sequential"
...



| Layer (type)                 | Output Shape | Param #     |
|------------------------------|--------------|-------------|
| xception (Functional)        | ?            | 20,861,480  |
| conv2d_4 (Conv2D)            | ?            | 0 (unbuilt) |
| max_pooling2d (MaxPooling2D) | ?            | 0 (unbuilt) |
| dropout (Dropout)            | ?            | 0 (unbuilt) |
| flatten (Flatten)            | ?            | 0 (unbuilt) |
| dense (Dense)                | ?            | 0 (unbuilt) |



...
Total params: 20,861,480 (79.58 MB)

...
Trainable params: 0 (0.00 B)

...
Non-trainable params: 20,861,480 (79.58 MB)

```

Gambar 16 Model Summary Percobaan C

Arsitektur pada gambar diatas di buat dengan Hyperparameter sebagai peroses trial dan eror pada nilai Hyperparameter membantu menemukan arsitektur dan akurasi terbaik. pada percobaan C Hyperparameter-nya sebagai berikut :

Tabel 7 Tabel Hyperparameter Percobaan C

Hyperparameter	
Split data	80% Train, 10%valid, 10% Test
Bach size	64
Layers	6
Epoch	25

Optimizer	Adam
Learning rate	0.001

Pada tabel diatas data dibagi menjadi 80% Train, 10% Valid, 10% Test dimana menggunakan batch size atau sampel data yang disebarluaskan ke Neural Network sebanyak 64, yang berarti jika kita memiliki 1.050 dataset maka algoritma CNN akan menggunakan 64 sampel data pertama dari 1.050 data yang kita miliki lalu disebarluaskan atau training oleh Neural Network sampai selesai kemudian mengambil kembali 64 sampel selanjutnya dari 1.050 data, begitu seterusnya. Sedangkan layers 6 artinya arsitektur terdiri dari 6 layer yaitu layer konvolusi, layer maxpooling, layer konvolusi, layer maxpooling, layer konvolusi, layer maxpooling, layer konvolusi, layer maxpooling, layer flatten, dan layer fully connected. Epoch 25 berarti proses training dilakukan sebanyak 25 kali iterasi atau 25 kali perulangan. Berikutnya ada Optimizer Adam mengatur learning rate berdasarkan besaran nilai rata-rata dari weight, dan yang terakhir kita menggunakan learning rate 0.001 dimana semakin kecil suatu learning rate maka model akan mempelajari data training semakin detail.

Percobaan C dilakukan dengan 25 *epoch* atau 25 kali iterasi, pelatihan model ini menghasilkan akurasi 85,71% dengan *loss, accuracy* seperti pada gambar berikut:

```

14/14    - - - - - 82s 5s/step - accuracy: 0.2109 - loss: 2.1645 - val_accuracy: 0.3048 - val_loss: 1.7863
Epoch 2/25
14/14    71s 5s/step - accuracy: 0.3271 - loss: 1.6994 - val_accuracy: 0.4286 - val_loss: 1.5721
Epoch 3/25
14/14    70s 4s/step - accuracy: 0.4564 - loss: 1.4767 - val_accuracy: 0.4762 - val_loss: 1.5463
Epoch 4/25
14/14    71s 5s/step - accuracy: 0.5355 - loss: 1.3152 - val_accuracy: 0.5429 - val_loss: 1.3268
Epoch 5/25
14/14    70s 4s/step - accuracy: 0.5810 - loss: 1.1337 - val_accuracy: 0.6286 - val_loss: 1.1686
Epoch 6/25
14/14    71s 4s/step - accuracy: 0.6421 - loss: 0.9671 - val_accuracy: 0.6381 - val_loss: 1.0181
Epoch 7/25
14/14    72s 5s/step - accuracy: 0.7298 - loss: 0.8122 - val_accuracy: 0.6571 - val_loss: 0.9669
Epoch 8/25
14/14    70s 4s/step - accuracy: 0.7781 - loss: 0.6498 - val_accuracy: 0.6857 - val_loss: 0.8764
Epoch 9/25
14/14    70s 5s/step - accuracy: 0.8068 - loss: 0.5405 - val_accuracy: 0.6667 - val_loss: 0.8962
Epoch 10/25
14/14    71s 5s/step - accuracy: 0.8415 - loss: 0.4921 - val_accuracy: 0.7333 - val_loss: 0.7915
Epoch 11/25
14/14    71s 5s/step - accuracy: 0.8718 - loss: 0.4181 - val_accuracy: 0.7333 - val_loss: 0.9118
Epoch 12/25
14/14    70s 5s/step - accuracy: 0.8631 - loss: 0.4557 - val_accuracy: 0.7429 - val_loss: 0.8105
Epoch 13/25
14/14    72s 5s/step - accuracy: 0.8901 - loss: 0.3164 - val_accuracy: 0.7714 - val_loss: 0.8229

```

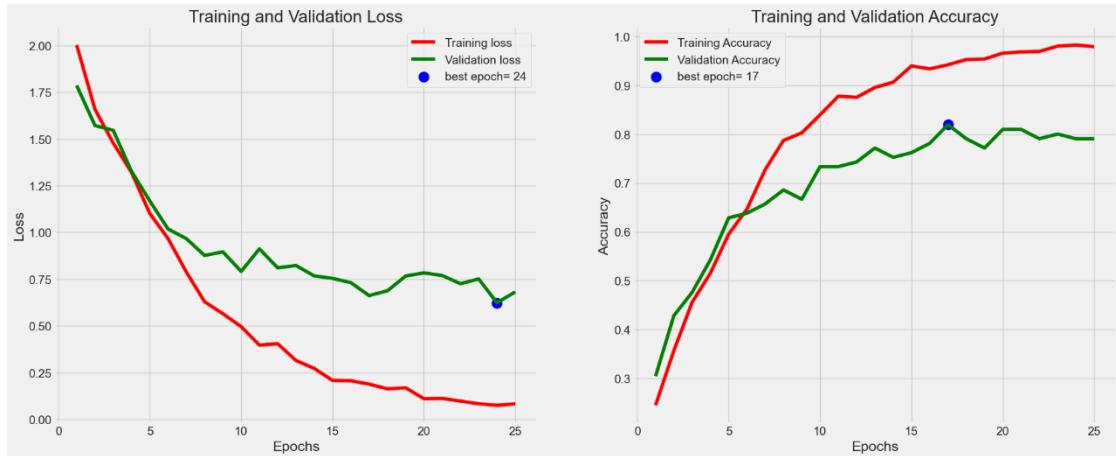
```

Epoch 14/25
14/14    71s 5s/step - accuracy: 0.9022 - loss: 0.2627 - val_accuracy: 0.7524 - val_loss: 0.7673
Epoch 15/25
14/14    71s 5s/step - accuracy: 0.9374 - loss: 0.2171 - val_accuracy: 0.7619 - val_loss: 0.7540
Epoch 16/25
14/14    72s 5s/step - accuracy: 0.9328 - loss: 0.2040 - val_accuracy: 0.7810 - val_loss: 0.7315
Epoch 17/25
14/14    71s 5s/step - accuracy: 0.9404 - loss: 0.2024 - val_accuracy: 0.8190 - val_loss: 0.6622
Epoch 18/25
14/14    71s 5s/step - accuracy: 0.9468 - loss: 0.1780 - val_accuracy: 0.7905 - val_loss: 0.6874
Epoch 19/25
14/14    73s 5s/step - accuracy: 0.9516 - loss: 0.1796 - val_accuracy: 0.7714 - val_loss: 0.7666
Epoch 20/25
14/14    71s 4s/step - accuracy: 0.9735 - loss: 0.1055 - val_accuracy: 0.8095 - val_loss: 0.7834
Epoch 21/25
14/14    71s 5s/step - accuracy: 0.9714 - loss: 0.1007 - val_accuracy: 0.8095 - val_loss: 0.7695
Epoch 22/25
14/14    70s 5s/step - accuracy: 0.9643 - loss: 0.0969 - val_accuracy: 0.7905 - val_loss: 0.7257
Epoch 23/25
14/14    70s 4s/step - accuracy: 0.9774 - loss: 0.0952 - val_accuracy: 0.8000 - val_loss: 0.7509
Epoch 24/25
14/14    70s 5s/step - accuracy: 0.9822 - loss: 0.0699 - val_accuracy: 0.7905 - val_loss: 0.6236
Epoch 25/25
14/14    72s 5s/step - accuracy: 0.9800 - loss: 0.0809 - val_accuracy: 0.7905 - val_loss: 0.6815

```

Gambar 17 Percobaan C dilakukan dengan 25 epoch atau 25 kali iterasi

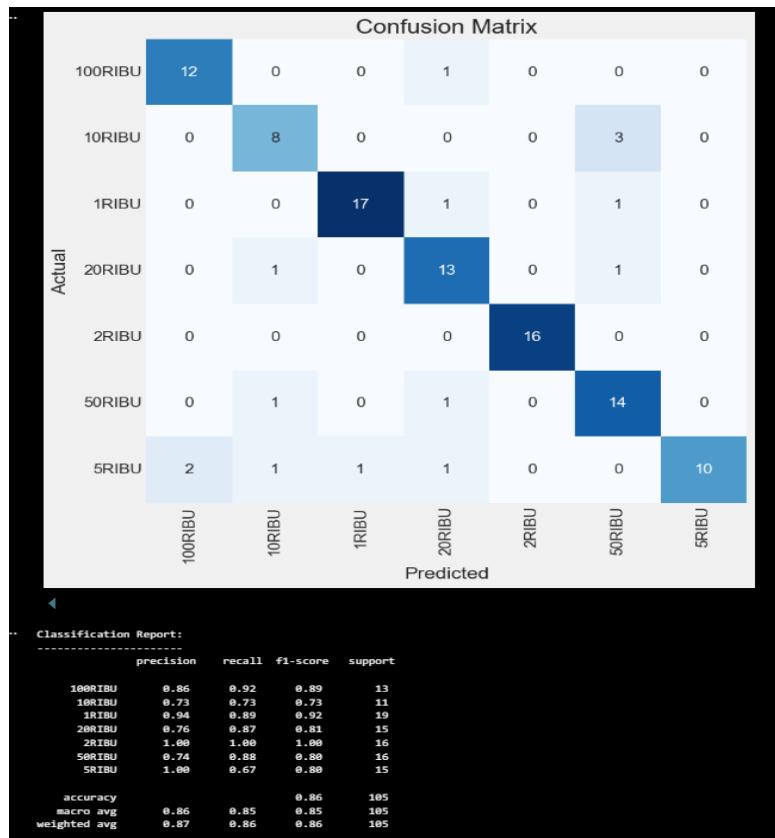
Nilai pada tabel akan menghasilkan grafik seperti pada gambar berikut :



Gambar 18 Grafik Percobaan C

Gambar diatas menunjukan model mempelajarai data training secara perlahan dan mendetail ini bisa dilihat dari bentuk kurva yang mana untuk training dan validation loss perlahan turun yang mengukur seberapa bagus performa yang dihasilkan model pembelajaran dalam melakukan prediksi terhadap target dan untuk training dan validation accuracy perlahan naik yang menunjukan performa akurasinya semakin meningkat.

Untuk memastikan model perlu dilakukan pelatihan ulang atau tidak dapat dilakukan pengujian model dengan menggunakan confusion matrix yang mana output dari pengujian ini sendiri adalah nilai akurasi, presisi, recall dan hasil f1. Berikut hasil pengujian model disetiap percobaan sebagai berikut:



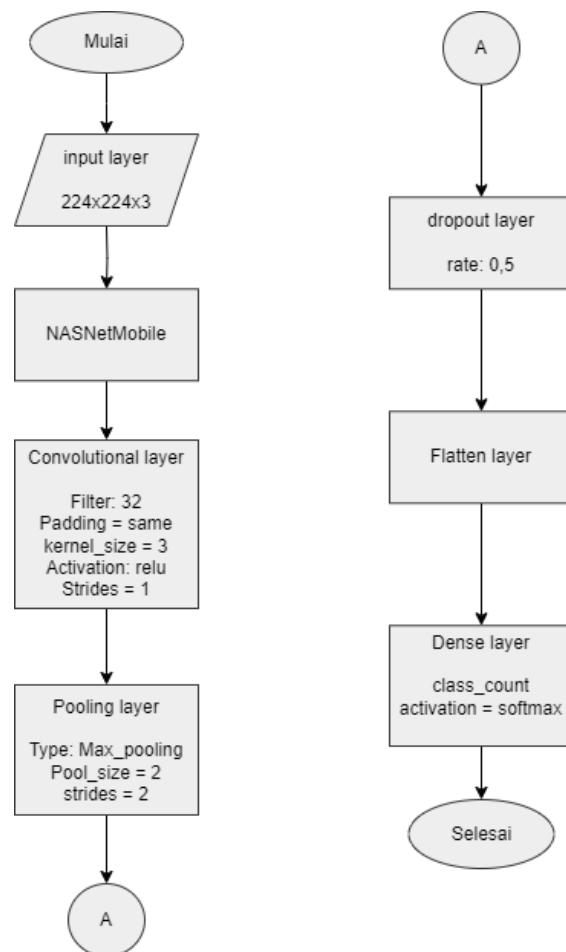
Gambar 19 Confusion Matriks Percobaan C

Confusion matrix ini menunjukkan performa model klasifikasi dengan akurasi keseluruhan sebesar 0.86. model berhasil mengklasifikasikan sebagian besar dengan benar, namun ada beberapa kesalahan klasifikasi terutama pada kelas 10RIBU, 20RIBU, dan 50RIBU. Kelas 100RIBU dan 1RIBU memiliki precision dan recall yang tinggi, masing-masing 0.86 dan 0.94, menunjukkan ketepatan dan kemampuan model dalam mengidentifikasi sampel-sampel di kelas tersebut. Kelas 10RIBU memiliki precision, recall, dan F1-score yang lebih rendah (0.73), menunjukkan tantangan dalam mengklasifikasikan sampel di kelas ini. Nilai F1-Score rata-rata menunjukkan keseimbangan antara precision dan recall, dengan weighted average precision sebesar 0.87 dan recall sebesar 0.86, menunjukkan performa model yang cukup baik secara

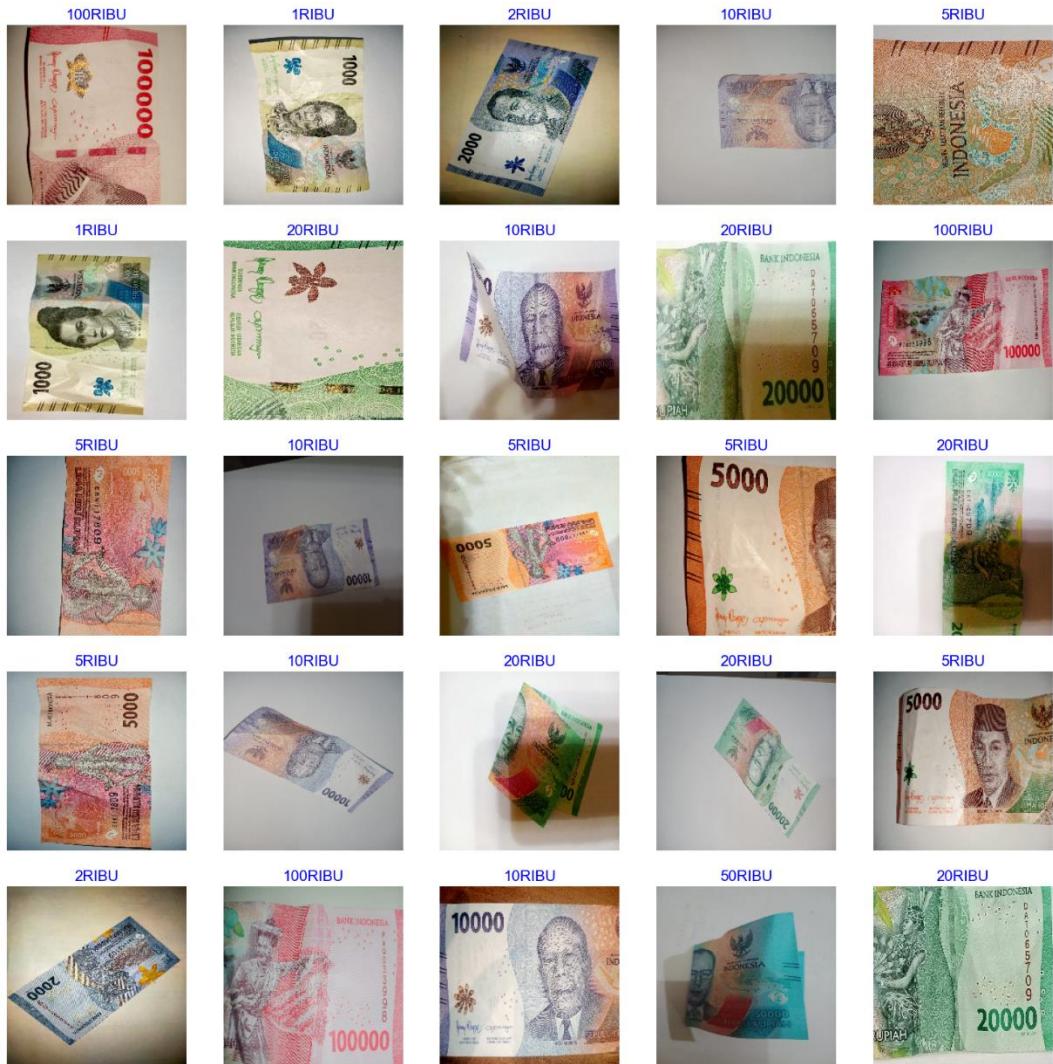
keseluruhan. Meskipun demikian, ada ruang untuk perbaikan terutama pada kelas yang menunjukkan hasil yang kurang optimal, seperti 10RIBU dan 50RIBU. Secara keseluruhan, model cukup andal dalam klasifikasinya.

3.1.4. Percobaan D: Hasil pembentukan model CNN *NASNetMobile*

Hasil dari percobaan D dengan inputan 224x224x3 maka dibentuklah arsitektur dengan total 6 layer. Dilakukan 1 kali tahapan convolution dan pooling, kemudian masuk ke flatten layer untuk mengubahnya menjadi matriks satu dimensi sehingga bisa menjadi inputan pada *fully connected layer*. Adapun tahapan dari arsitektur seperti gambar.



Gambar 20 Flowchart pembentukan model CNN NASNetMobile



Gambar 21 Hasil deteksi model CNN *NASNetMobile*

Gambar 15 diatas menunjukkan berbagai uang kertas dengan nilai nominal berbeda dalam mata uang rupiah (IDR), yang telah diidentifikasi oleh model CNN *NASNetMobile*. Setiap gambar diberi label sesuai dengan nilai nominalnya, termasuk 1RIBU, 2RIBU, 5RIBU, 10RIBU, 20RIBU, 50RIBU, dan 100RIBU. Uang kertas ditampilkan dalam berbagai posisi dan sudut pandang, menunjukkan variasi visual yang berbeda-beda. Model CNN *NASNetMobile* telah berhasil mendekripsi dan mengklasifikasikan nilai nominal

dari setiap uang kertas berdasarkan fitur visual yang dikenali dari masing-masing uang kertas.

```
...
Model: "sequential"
...


| Layer (type)                 | Output Shape | Param #     |
|------------------------------|--------------|-------------|
| NASNet (Functional)          | ?            | 4,269,716   |
| conv2d (Conv2D)              | ?            | 0 (unbuilt) |
| max_pooling2d (MaxPooling2D) | ?            | 0 (unbuilt) |
| dropout (Dropout)            | ?            | 0 (unbuilt) |
| flatten (Flatten)            | ?            | 0 (unbuilt) |
| dense (Dense)                | ?            | 0 (unbuilt) |


...
Total params: 4,269,716 (16.29 MB)
...
Trainable params: 0 (0.00 B)
...
Non-trainable params: 4,269,716 (16.29 MB)
```

Gambar 22 Model Summary Percobaan D

Arsitektur pada gambar diatas di buat dengan Hyperparameter sebagai berikut peroses trial dan eror pada nilai Hyperparameter membantu menemukan arsitektur dan akurasi terbaik .pada percobaan D Hyperparameter-nya sebagai berikut :

Tabel 8 Tabel Hyperparameter Percobaan D

Hyperparameter	
Split data	80% Train, 10% valid, 10% Test
Bach size	64

Layers	6
Epoch	25
Optimizer	Adam
Learning rate	0.001

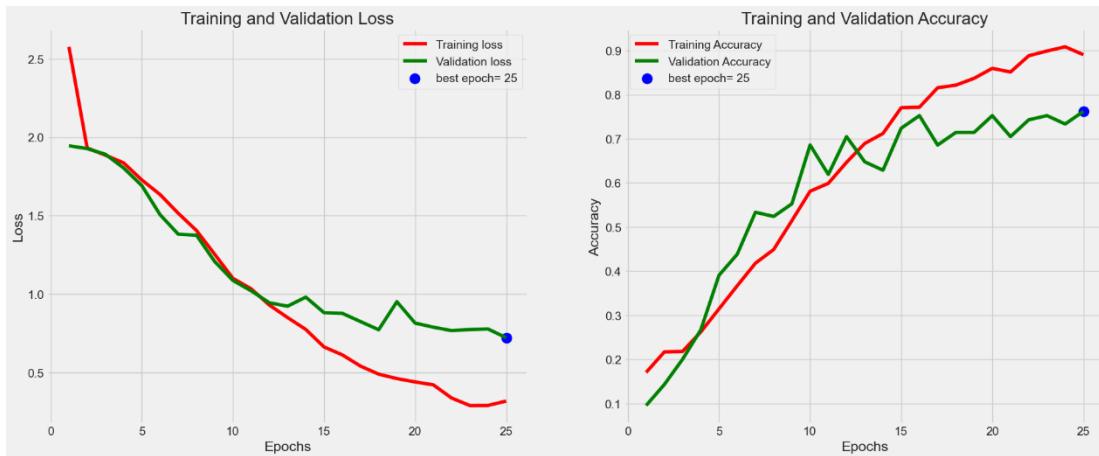
Pada tabel diatas data dibagi menjadi 80% Train, 10% Valid, 10% Test dimana menggunakan batch size atau sampel data yang disebarluaskan ke *Neural Network* sebanyak 64, yang berarti jika kita memiliki 1.050 dataset maka algoritma CNN akan menggunakan 64 sampel data pertama dari 1.050 data yang kita miliki lalu disebarluaskan atau *training* oleh *Neural Network* sampai selesai kemudian mengambil kembali 64 sampel selanjutnya dari 1.050 data, begitu seterusnya. Sedangkan *layers* 6 artinya arsitektur terdiri dari 6 *layer* yaitu *layer konvolusi*, *layer maxpooling*, *layer konvolusi*, *layer maxpooling*, *layer konvolusi*, *layer maxpooling*, *layer konvolusi*, *layer maxpooling*, *layer flatten*, dan *layer fully connected*. *Epoch* 25 berarti proses *training* dilakukan sebanyak 25 kali iterasi atau 25 kali perulangan. Berikutnya ada *Optimizer Adam* mengatur *learning rate* berdasarkan besaran nilai rata-rata dari *weight*, dan yang terakhir kita menggunakan *learning rate* 0.001 dimana semakin kecil suatu *learning rate* maka model akan mempelajari data training semakin detail.

Percobaan D dilakukan dengan 25 *epoch* atau 25 kali iterasi, pelatihan model ini menghasilkan akurasi 84,76% dengan *loss*, *accuracy* seperti pada gambar berikut:

Epoch 1/25
14/14 37s 2s/step - accuracy: 0.8934 - loss: 0.3068 - val_accuracy: 0.7619 - val_loss: 0.7525
Epoch 2/25
14/14 36s 2s/step - accuracy: 0.8988 - loss: 0.2773 - val_accuracy: 0.7714 - val_loss: 0.6790
Epoch 3/25
14/14 38s 2s/step - accuracy: 0.8982 - loss: 0.3040 - val_accuracy: 0.7810 - val_loss: 0.6763
Epoch 4/25
14/14 38s 2s/step - accuracy: 0.9395 - loss: 0.2005 - val_accuracy: 0.8000 - val_loss: 0.7464
Epoch 5/25
14/14 37s 2s/step - accuracy: 0.9169 - loss: 0.2151 - val_accuracy: 0.7714 - val_loss: 0.7558
Epoch 6/25
14/14 38s 2s/step - accuracy: 0.9402 - loss: 0.2113 - val_accuracy: 0.7333 - val_loss: 0.7267
Epoch 7/25
14/14 37s 2s/step - accuracy: 0.9555 - loss: 0.1654 - val_accuracy: 0.7524 - val_loss: 0.8547
Epoch 8/25
14/14 38s 2s/step - accuracy: 0.9431 - loss: 0.1737 - val_accuracy: 0.7714 - val_loss: 0.7517
Epoch 9/25
14/14 38s 2s/step - accuracy: 0.9526 - loss: 0.1370 - val_accuracy: 0.7619 - val_loss: 0.6332
Epoch 10/25
14/14 37s 2s/step - accuracy: 0.9528 - loss: 0.1564 - val_accuracy: 0.7905 - val_loss: 0.7763
Epoch 11/25
14/14 38s 2s/step - accuracy: 0.9257 - loss: 0.2179 - val_accuracy: 0.7714 - val_loss: 0.8032
Epoch 12/25
14/14 37s 2s/step - accuracy: 0.9589 - loss: 0.1508 - val_accuracy: 0.7333 - val_loss: 0.8071
Epoch 13/25
14/14 37s 2s/step - accuracy: 0.9450 - loss: 0.1837 - val_accuracy: 0.7714 - val_loss: 0.7501
Epoch 14/25
14/14 38s 2s/step - accuracy: 0.9611 - loss: 0.1197 - val_accuracy: 0.7714 - val_loss: 0.7896
Epoch 15/25
14/14 47s 3s/step - accuracy: 0.9381 - loss: 0.1564 - val_accuracy: 0.7714 - val_loss: 0.7978
Epoch 16/25
14/14 89s 5s/step - accuracy: 0.9673 - loss: 0.1164 - val_accuracy: 0.7524 - val_loss: 0.7718
Epoch 17/25
14/14 89s 5s/step - accuracy: 0.9630 - loss: 0.1265 - val_accuracy: 0.7619 - val_loss: 0.8896
Epoch 18/25
14/14 93s 5s/step - accuracy: 0.9769 - loss: 0.1045 - val_accuracy: 0.7810 - val_loss: 0.8790
Epoch 19/25
14/14 88s 5s/step - accuracy: 0.9761 - loss: 0.0758 - val_accuracy: 0.7714 - val_loss: 0.7648
Epoch 20/25
14/14 67s 3s/step - accuracy: 0.9643 - loss: 0.1085 - val_accuracy: 0.8000 - val_loss: 0.7465
Epoch 21/25
14/14 38s 2s/step - accuracy: 0.9731 - loss: 0.0935 - val_accuracy: 0.8286 - val_loss: 0.7120
Epoch 22/25
14/14 37s 2s/step - accuracy: 0.9750 - loss: 0.0713 - val_accuracy: 0.8000 - val_loss: 0.8198
Epoch 23/25
14/14 36s 2s/step - accuracy: 0.9723 - loss: 0.0843 - val_accuracy: 0.8095 - val_loss: 0.7050
Epoch 24/25
14/14 37s 2s/step - accuracy: 0.9752 - loss: 0.0684 - val_accuracy: 0.7524 - val_loss: 0.9133
Epoch 25/25
14/14 37s 2s/step - accuracy: 0.9788 - loss: 0.0671 - val_accuracy: 0.7810 - val_loss: 0.8586

Gambar 23 Percobaan D dilakukan dengan 25 epoch atau 25 kali iterasi

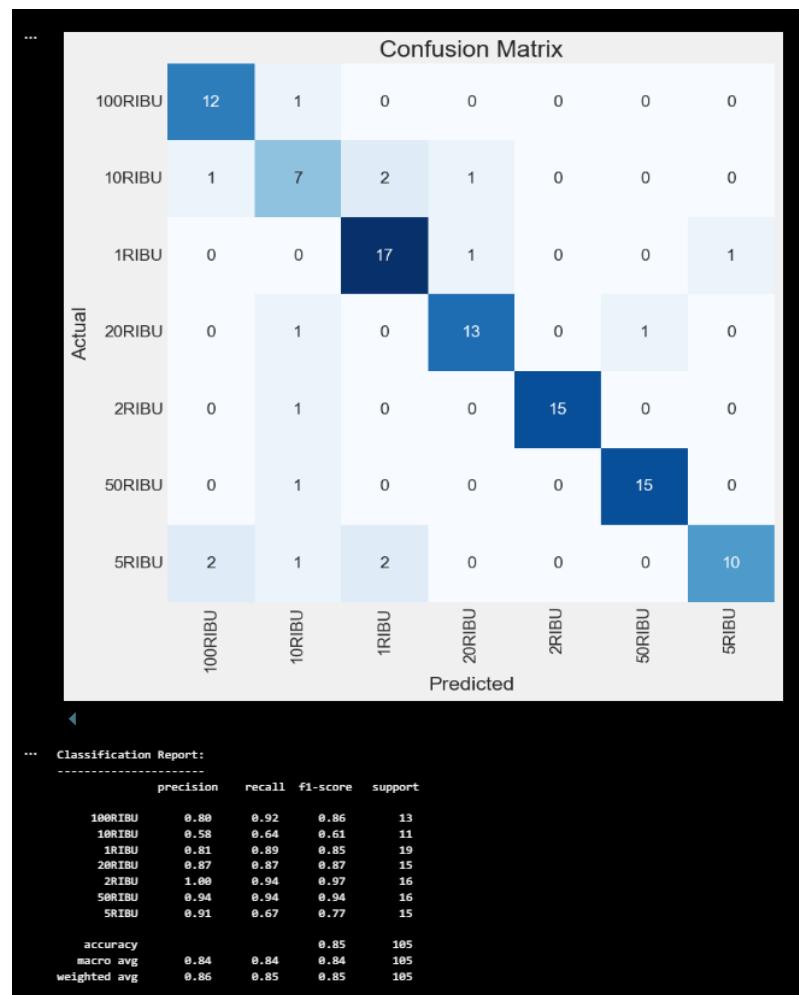
Nilai pada tabel akan menghasilkan grafik seperti pada gambar berikut :



Gambar 24 Grafik Percobaan D

Gambar diatas menunjukan model mempelajarai data *training* secara perlahan dan mendetail ini bisa dilihat dari bentuk kurva yang mana untuk *training* dan *validation loss* perlahan turun yang mengukur seberapa bagus performa yang dihasilkan model pembelajaran dalam melakukan prediksi terhadap target dan untuk training dan validation accuracy perlahan naik yang menunjukan performa akurasinya semakin meningkat.

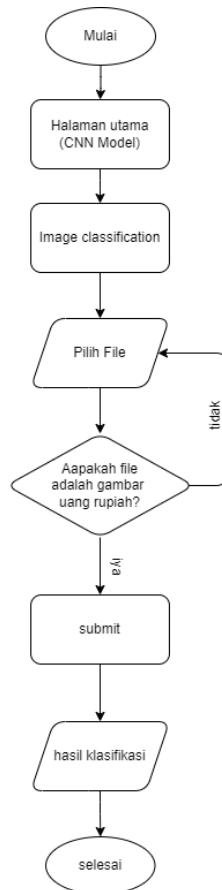
Untuk memastikan model perlu dilakukan pelatihan ulang atau tidak dapat dilakukan pengujian model dengan menggunakan confusion matrix yang mana output dari pengujian ini sendiri adalah nilai akurasi, presisi, recall dan hasil f1. Berikut hasil pengujian model disetiap percobaan sebagai berikut:

Gambar 25 *Confusion Matriks* Percobaan D

Hasil *confusion matrix* menunjukkan bahwa model ini memiliki performa yang cukup baik dengan 85% akurasi, namun ada beberapa kelas yang masih memerlukan peningkatan, seperti `10RIBU` dan `5RIBU` yang memiliki *precision* masing-masing 0.58 dan 0.91 serta recall 0.64 dan 0.67. Secara keseluruhan, model menunjukkan *precision*, *recall*, dan *f1-score* rata-rata berbobot sebesar 0.86, 0.85, dan 0.85. Kelas-kelas seperti `100RIBU`, `1RIBU`, `20RIBU`, `2RIBU`, dan `50RIBU` menunjukkan performa yang tinggi dengan

f1-score di atas 0.77, sementara kelas `2RIBU` memiliki performa tertinggi dengan *f1-score* 0.97.

3.2. Flowchart sistem



Gambar 26 *Flowchart* Sistem

Flowchart sistem diatas menunjukkan alur proses untuk melakukan klasifikasi gambar menggunakan model *Convolutional Neural Network* (CNN). Berikut adalah penjelasan setiap langkah dalam *flowchart* tersebut:

- 1) Mulai:

Proses dimulai dari sini

2) Halaman utama (CNN Model):

Pengguna mengakses halaman utama atau antarmuka dari aplikasi model CNN.

3) *Image classification:*

Pengguna menuju ke halaman klasifikasi gambar dari aplikasi.

4) Pilih File:

Pengguna diminta untuk memilih file yang akan diklasifikasikan.

5) Apakah file adalah gambar uang rupiah?:

Ini adalah titik keputusan di mana pengguna memeriksa apakah file yang dipilih adalah gambar uang rupiah.

- Jika Ya, lanjut ke langkah berikutnya.
- Jika Tidak, kembali ke langkah "Pilih File" untuk memilih file yang berbeda.

6) *Submit:*

Pengguna mengirimkan file yang telah dipilih untuk diklasifikasikan oleh model CNN.

7) Hasil klasifikasi:

Hasil dari klasifikasi ditampilkan.

8) Selesai:

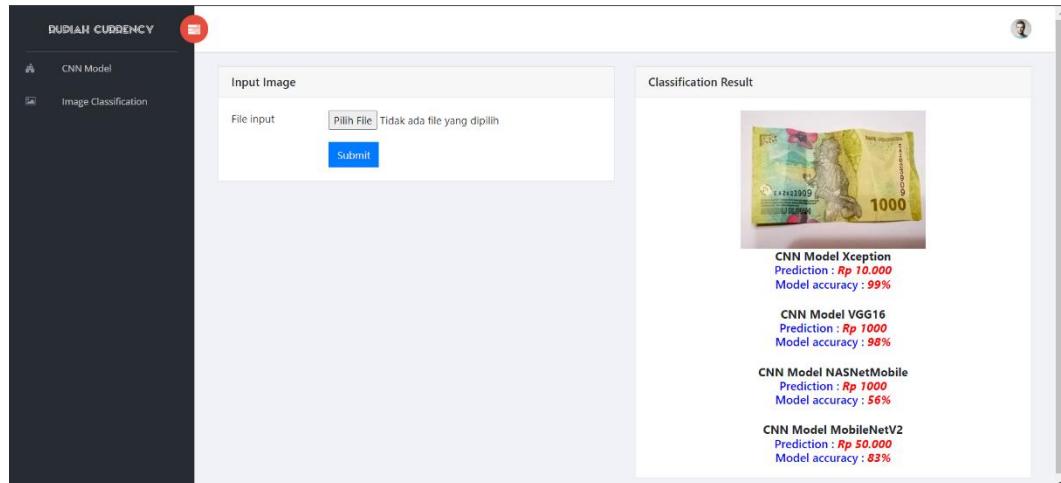
Proses berakhir di sini.

3.3. Hasil Klasifikasi

Pada tahap ini diberikan hasil klasifikasi menggunakan 4 base model CNN dengan citra dari setiap kelas. Berikut hasil klasifikasi dari masing-masing kelas:

- 1) Rp1000

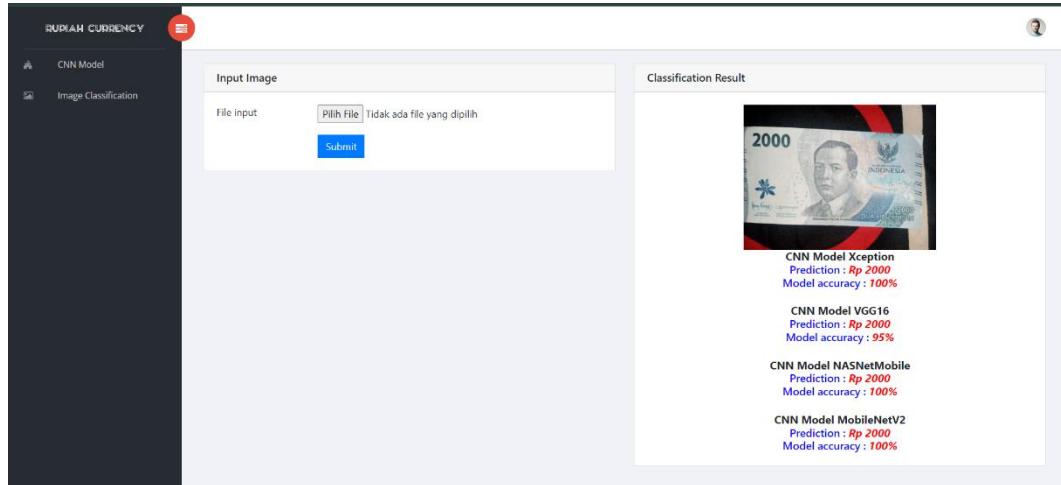
Halaman ini menampilkan hasil klasifikasi dari citra Rp1000, Berikut interface pada hasil klasifikasi Rp1000 sebagai berikut:



Gambar 27 Hasil klasifikasi Rp 1000

- 2) Rp2000

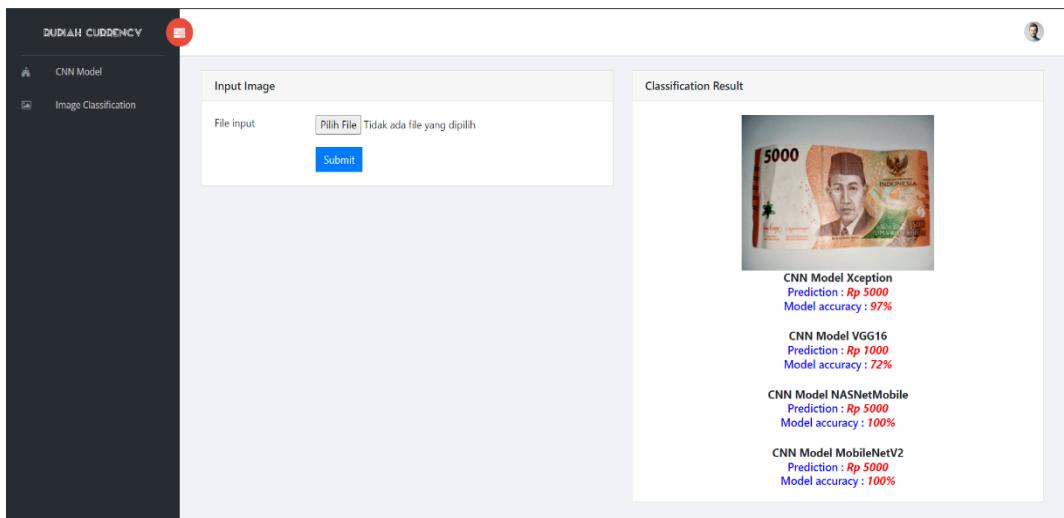
Halaman ini menampilkan hasil klasifikasi dari citra Rp2000, Berikut interface pada hasil klasifikasi Rp2000 sebagai berikut:



Gambar 28 Hasil klasifikasi Rp 2000

3) Rp5000

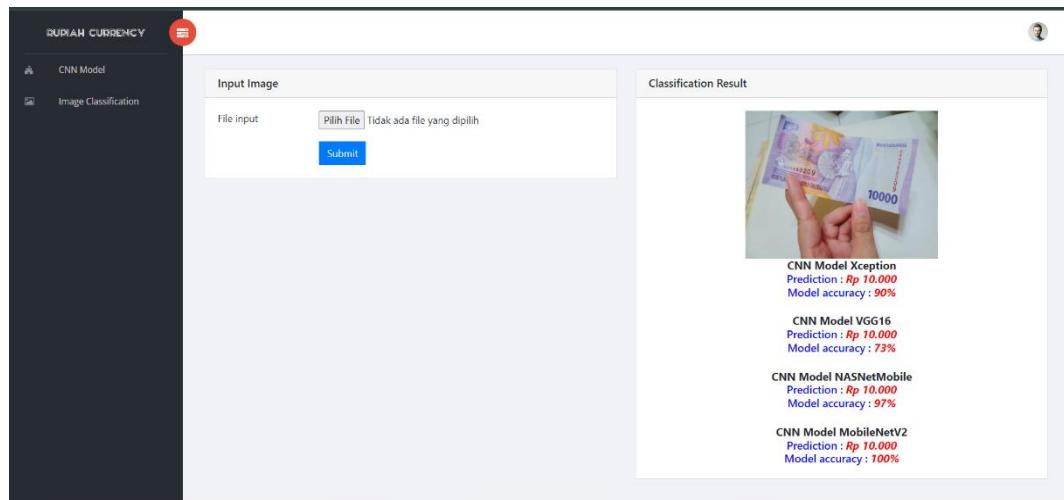
Halaman ini menampilkan hasil klasifikasi dari citra Rp5000, Berikut interface pada hasil klasifikasi Rp5000 sebagai berikut:



Gambar 29 Hasil klasifikasi Rp 5000

4) Rp10.000

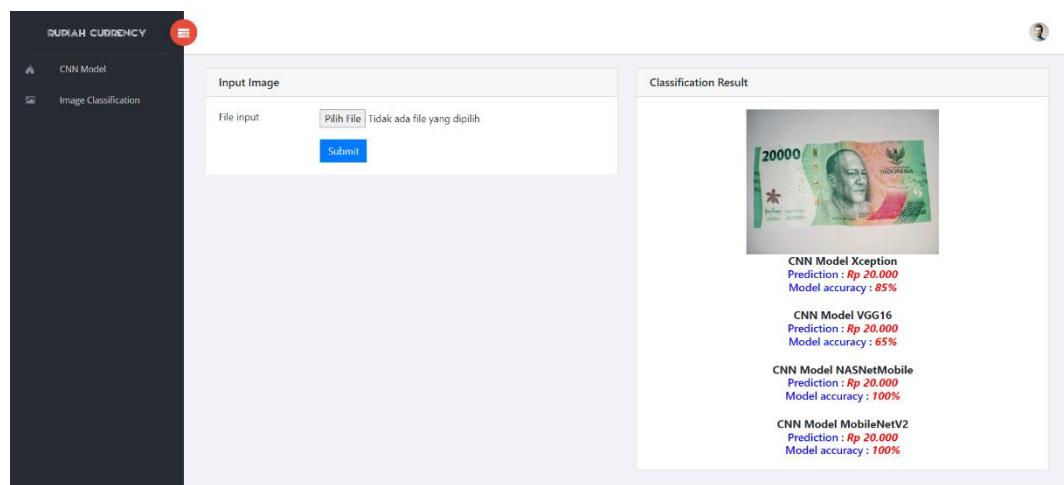
Halaman ini menampilkan hasil klasifikasi dari citra Rp10.000, Berikut interface pada hasil klasifikasi Rp10.000 sebagai berikut:



Gambar 30 Hasil klasifikasi Rp 10.000

5) Rp20.000

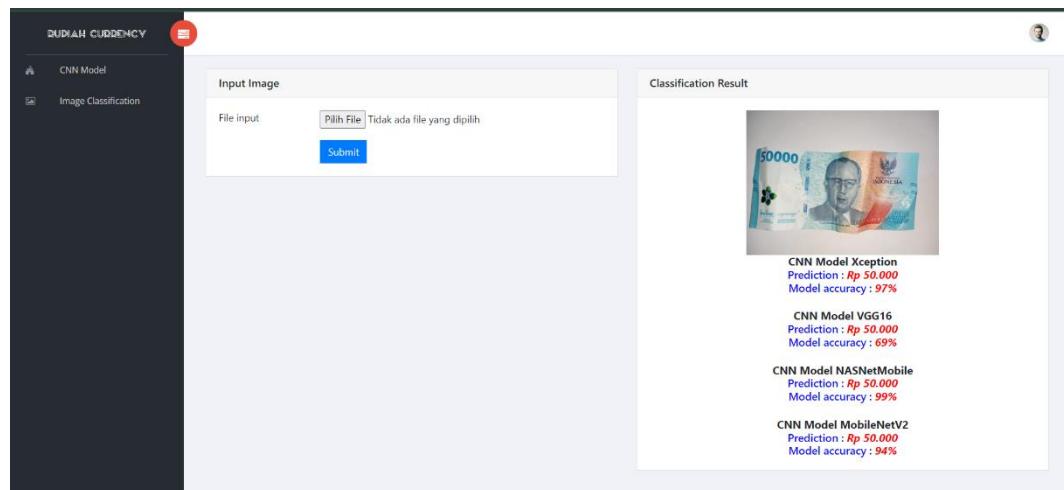
Halaman ini menampilkan hasil klasifikasi dari citra Rp20.000, Berikut interface pada hasil klasifikasi Rp20.000 sebagai berikut:



Gambar 31 Hasil klasifikasi Rp 20.000

6) Rp50.000

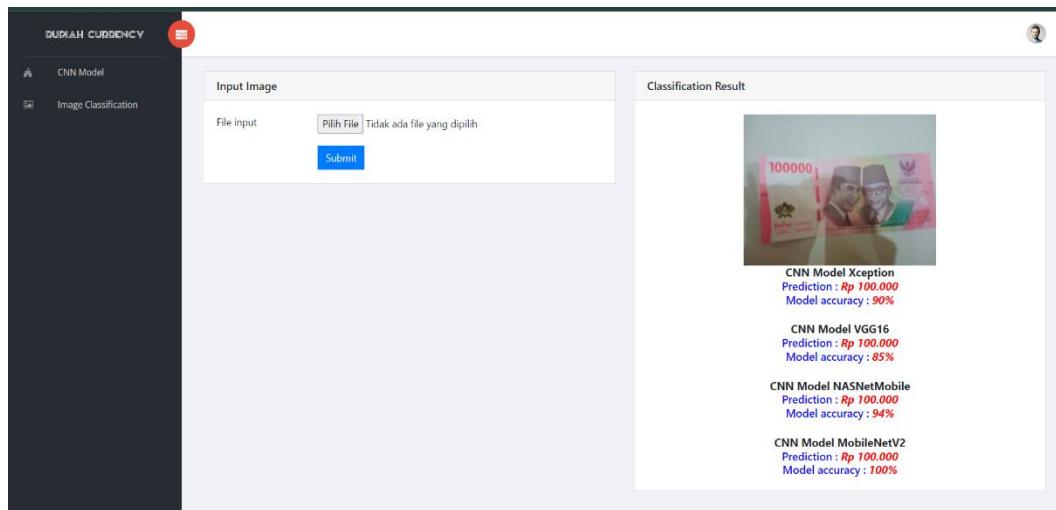
Halaman ini menampilkan hasil klasifikasi dari citra Rp50.000, Berikut interface pada hasil klasifikasi Rp50.000 sebagai berikut:



Gambar 32 Hasil klasifikasi Rp 50.000

7) Rp100.000

Halaman ini menampilkan hasil klasifikasi dari citra Rp100.000, Berikut interface pada hasil klasifikasi Rp100.000 sebagai berikut:



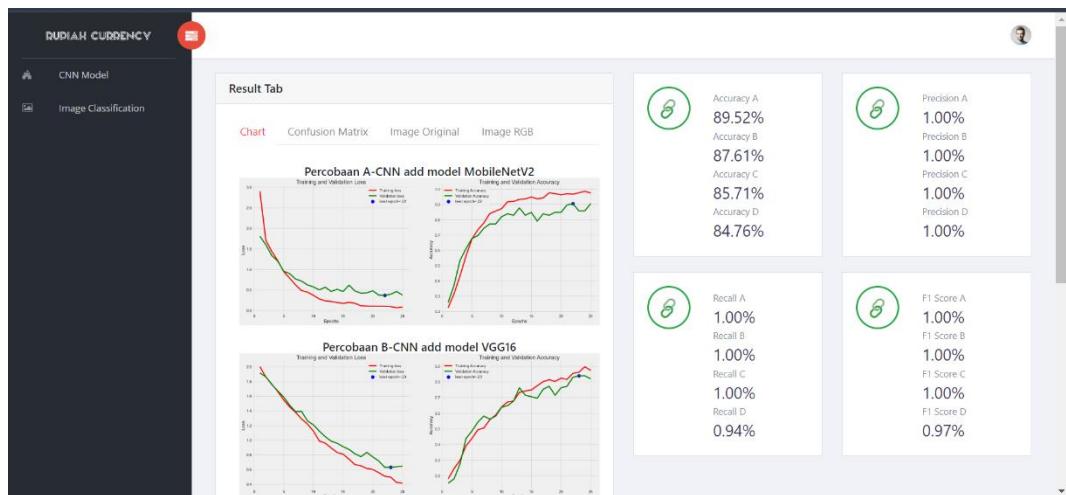
Gambar 33 Hasil klasifikasi Rp 100.000

3.4. Tampilan *Interface*

Pada tahap ini tampilan *interface* dari 2 halaman atau *page* akan diberikan, mulai dari halaman utama yaitu *CNN Model* kemudian halaman klasifikasi yaitu *Image Classification*.

1) *CNN Model*

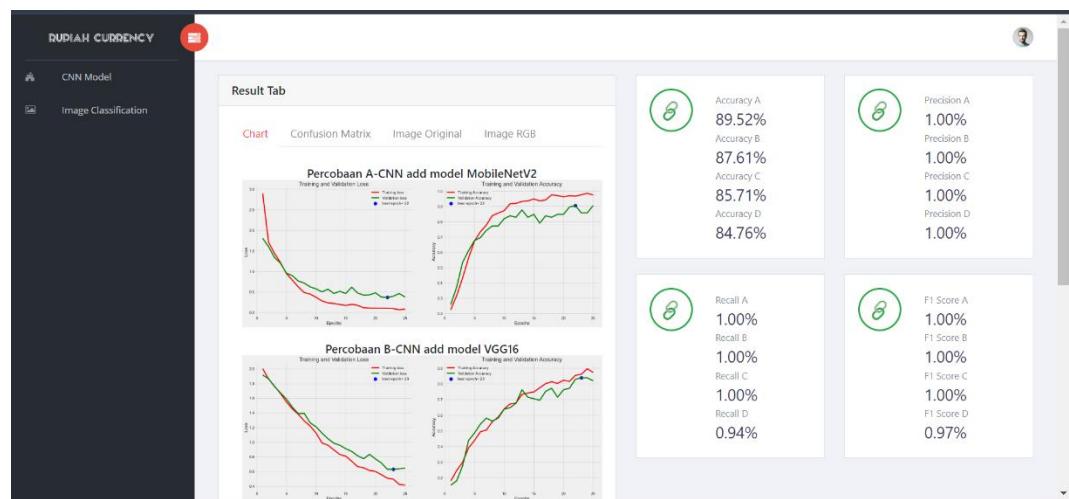
CNN Model merupakan halaman utama yang akan muncul ketika user memulai program, Berikut halaman *CNN Model* :



Gambar 34 Halaman *CNN Model*

2) *Chart*

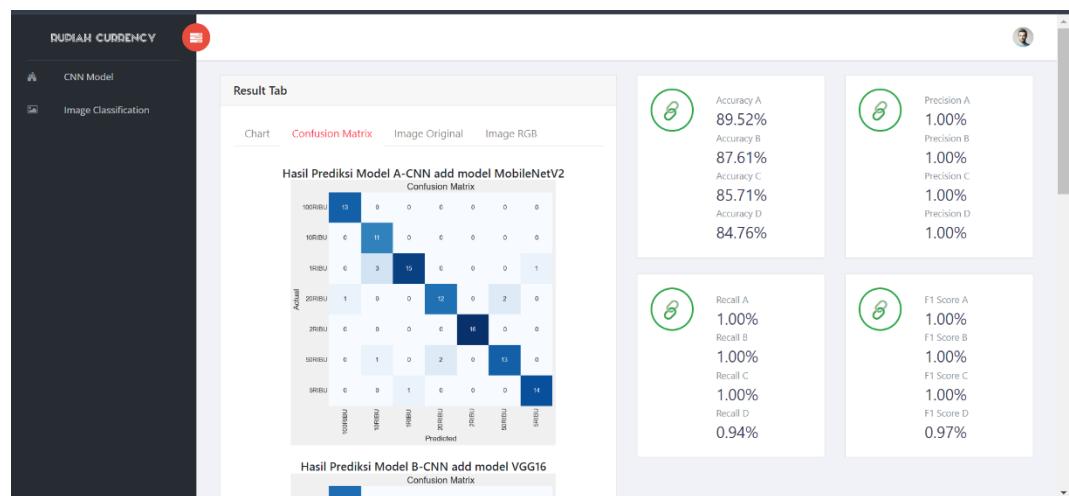
Chart merupakan tab untuk melihat hasil grafik dari 4 base model CNN yang telah dibangun, *tab Chart* sendiri akan otomatis tertampilkan ketika user memasuki halaman utama yaitu CNN Model Berikut adalah tampilan dari tab *Chart*:



Gambar 35 *Chart*

3) Confusion Matrix

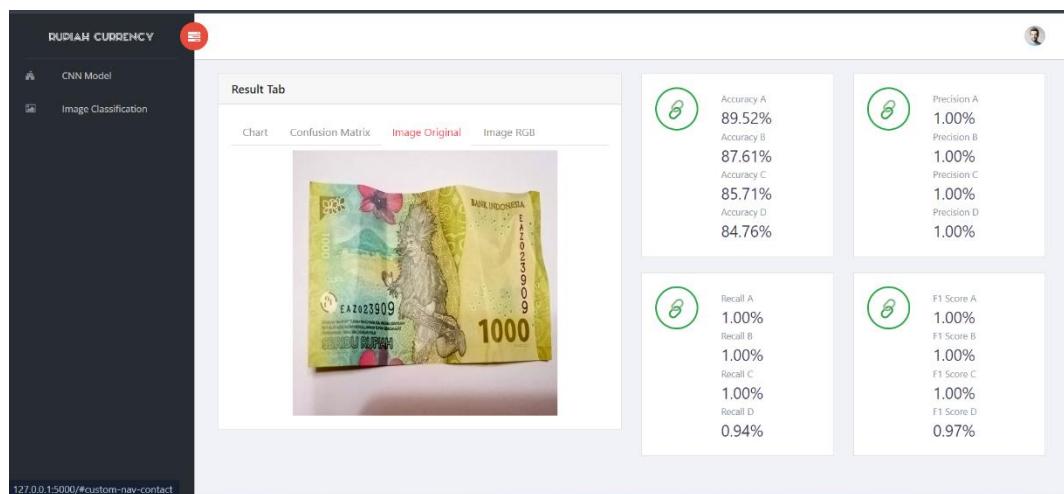
Confusion Matrix merupakan tab untuk menampilkan hasil *Confusion Matrix* dari 4 base model CNN, Berikut tampilan dari tab *Confusion Matrix*:



Gambar 36 *Confusion Matrix*

4) *Image Original*

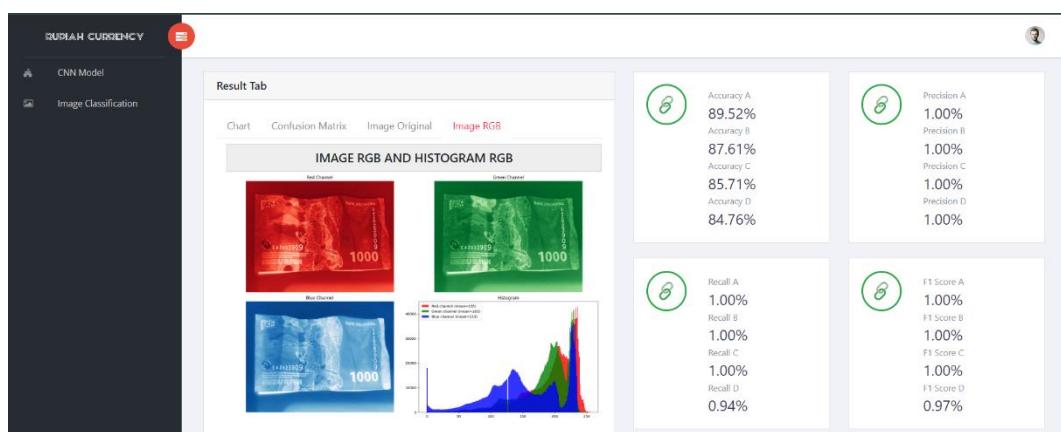
Image original merupakan *tab* untuk menampilkan gambar dari riwayat klasifikasi *user* pada halaman *Image Classification*. Berikut tampilan dari *tab Image Original*:



Gambar 37 *Image Original*

5) *Image RGB*

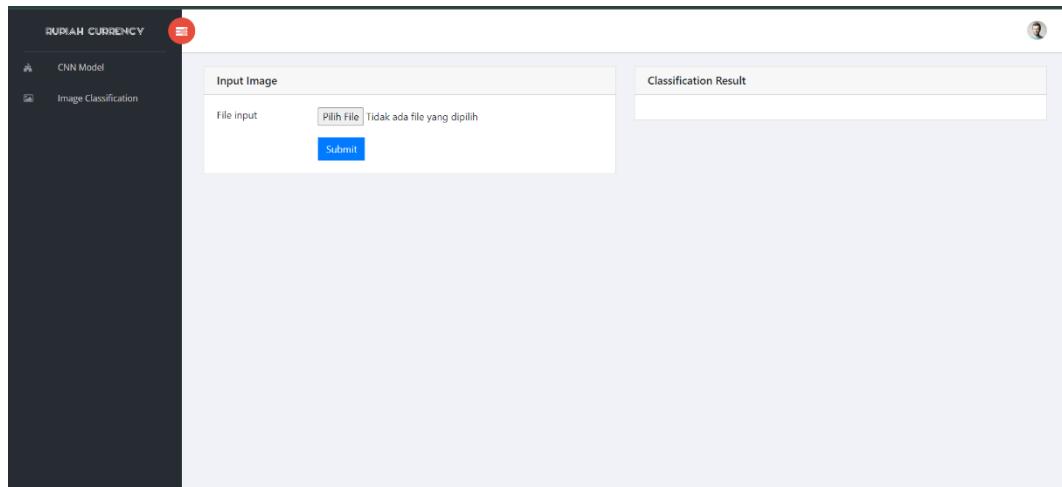
Image RGB merupakan *tab* untuk menampilkan gambar *RGB* yang didapat dari riwayat klasifikasi *user* pada halaman *Image Classification* kemudian gambar tersebut di *convert* menjadi gambar *RGB*. Berikut tampilan dari *tab Image RGB*:



Gambar 38 *Image RGB*

6) Image Classification

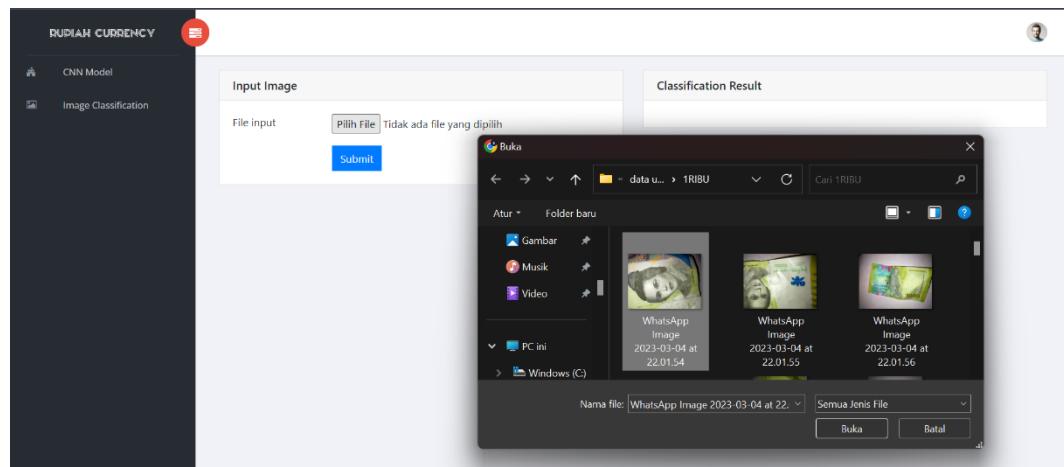
Image Classification merupakan halaman klasifikasi, dimana pada halaman ini *user* dapat melakukan klasifikasi terhadap mata uang rupiah. Berikut adalah tampilan dari halaman *Image Classification*:



Gambar 39 *Image Classification*

7) Pilih File

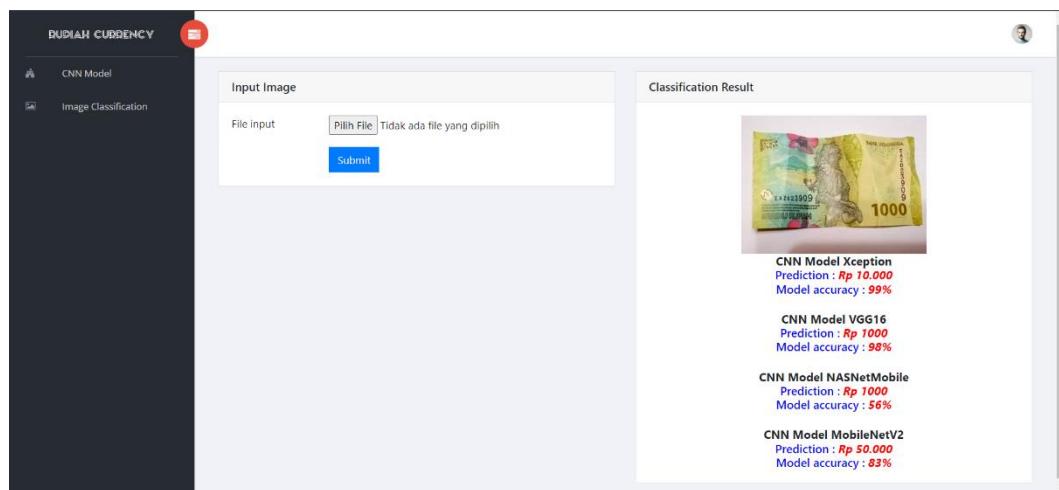
Pilih File merupakan tombol yang berfungsi untuk memilih gambar uang rupiah yang ingin digunakan sebagai objek yang diklasifikasi. Berikut tampilan setelah tombol Pilih File di klik:



Gambar 40 Pilih File

8) Submit

Submit merupakan tombol untuk mengirim gambar yang telah dipilih sebelumnya ke sistem untuk diklasifikasi oleh 4 base model CNN yang telah dibangun, sehingga akan menghasilkan output yang berbeda dari setiap base model CNN mulai dari accuracy dan result. Berikut tampilan setelah tombol submit di klik:



Gambar 41 *Submit*

3.5. Pengujian

3.5.1. *Blackbox Testing*

Blackbox testing adalah metode pengujian yang berurusan dengan hasil internal yang tidak diketahui. Penguji melihat perangkat lunak sebagai "*Blackbox*" yang tidak perlu menunjukkan isinya, tetapi tunduk pada pengujian eksternal. Dalam jenis pengujian *Blackbox* ini, perangkat lunak akan dijalankan dan kemudian mencoba untuk menguji apakah memenuhi kebutuhan pengguna yang telah ditentukan di awal tanpa membongkar daftar program.

Dibawah ini merupakan hasil pengujian pada aplikasi menggunakan metode *Blackbox testing*:

Tabel 9 Pengujian *Blackbox Testing*

ID	Deskripsi Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
T01	Menampilkan halaman utama <i>CNN Model</i>	Klik <i>CNN Model</i>	halaman utama <i>CNN Model</i> tertampil	Sesuai Harapan	[√] <i>Valid</i> [] <i>Invalid</i>
T02	Melihat grafik model cnn	Klik tab <i>Chart</i>	Menampilkan grafik model CNN	Sesuai harapan	[√] <i>Valid</i> [] <i>Invalid</i>
T03	Melihat hasil <i>confusion matrix model CNN</i>	Klik tab <i>Confusion Matrix</i>	Menampilkan hasil <i>confusion matrix model CNN</i>	Sesuai harapan	[√] <i>Valid</i> [] <i>Invalid</i>
T04	Melihat gambar <i>original</i>	Klik tab <i>Image Original</i>	Menampilkan gambar <i>original</i>	Sesuai harapan	[√] <i>Valid</i> [] <i>Invalid</i>
T05	Melihat gambar <i>RGB</i>	Klik tab <i>Image RGB</i>	Menampilkan gambar <i>RGB</i>	Sesuai harapan	[√] <i>Valid</i> [] <i>Invalid</i>
T06	Menampilkan halaman	Klik menu	Halaman <i>Image</i>	Sesuai harapan	[√] <i>Valid</i>

	klasifikasi gambar	<i>Image Classification</i>	<i>Classification</i> tertampil		<input type="checkbox"/> <i>Invalid</i>
T07	Memilih file gambar untuk diklasifikasikan	Klik tombol Pilih File	File gambar dapat dipilih	Sesuai harapan	<input checked="" type="checkbox"/> <i>Valid</i> <input type="checkbox"/> <i>Invalid</i>
T08	Menampilkan hasil klasifikasi gambar	Klik tombol <i>Submit</i>	Hasil klasifikasi gambar tertampil dengan prediksi model	Sesuai harapan	<input checked="" type="checkbox"/> <i>Valid</i> <input type="checkbox"/> <i>Invalid</i>

Dari tabel diatas, pengujian aplikasi dilakukan dengan menggunakan metode *Blackbox Testing* mendapatkan hasil yang Baik.

3.5.2. *Usability Testing*

Di bawah ini adalah hasil dari kuisioner yang telah diisi oleh 15 responden dengan. Data ini diperoleh melalui *Google Form*. Berdasarkan kuisioner yang diperoleh dari responden. Berikut ringkasan *usability testing* yang telah dilakukan:

1) Nama Responden

- Sahril Kamarey
- Yefta kambu
- CELINE JEKOLINE TAKAMOKAN
- Levina H.J Mirino
- Mei Merlince Gimlah
- Elis Antoh
- Elius kalolik
- Gabreliya Parihala

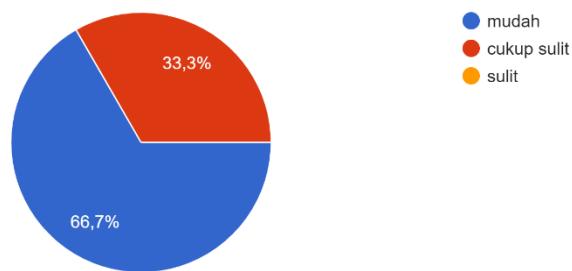
- Marsel
- Indra Pandu Bahari
- Elisabeth
- Fitri Febry Iriyanti
- Nurul
- Cintia dwi pusrita
- Petrus F.X Frabuku

2) Jurusan Responden

- Teknik Industri
- Teknik
- PGSD
- Teknik sipil
- Ilmu pemerintahan
- Akuntansi
- Ilmu komputer
- Ilmu pemerintahan
- Bahasa inggris
- Teknik Sipil
- Kesehatan masyarakat
- Pendidikan Teknologi informasi
- Fakultas hukum
- Hukum
- Ekonomi pembangunan

3) Apakah Anda mengalami kesulitan saat menggunakan program?

Apakah Anda mengalami kesulitan saat menggunakan program?
15 jawaban

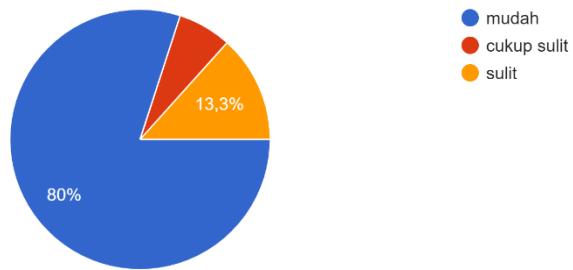


Gambar 42 Presentase Pertanyaan 1

Mayoritas responden merasa penggunaan program ini mudah (66.7%), namun ada sekelompok kecil yang merasa cukup sulit (33.3%), namun tidak ada yang merasa sulit.

- 4) Seberapa mudah Anda menemukan fitur untuk menuju ke halaman klasifikasi?

Seberapa mudah Anda menemukan fitur untuk menuju ke halaman klasifikasi?
15 jawaban



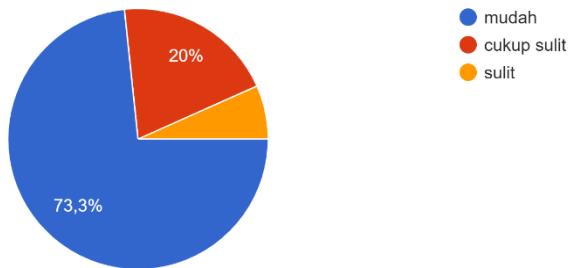
Gambar 43 Presentase Pertanyaan 2

- Mudah: 80%
- Cukup Sulit: 13.3%
- Sulit: 6.7%

Sebagian besar responden menemukan fitur untuk menuju halaman klasifikasi dengan mudah (80%).

- 5) Seberapa mudah Anda menemukan fitur untuk mengunggah gambar?

Seberapa mudah Anda menemukan fitur untuk mengunggah gambar?
15 jawaban



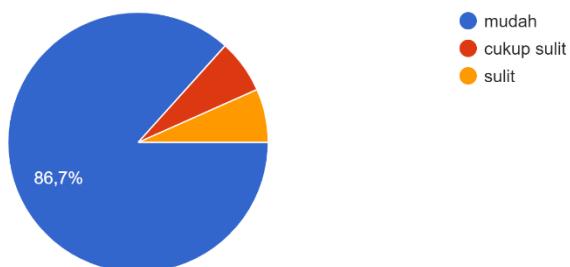
Gambar 44 Presentase Pertanyaan 3

- Mudah: 73,3%
- Cukup Sulit: 20%
- Sulit: 6,7%

Sebagian besar responden merasa mudah menemukan fitur untuk mengunggah gambar (73.3%), meskipun ada beberapa yang merasa cukup sulit (20%).

6) Apakah Anda mengalami kesulitan saat mengunggah gambar?

Apakah Anda mengalami kesulitan saat mengunggah gambar?
15 jawaban



Gambar 45 Presentase Pertanyaan 4

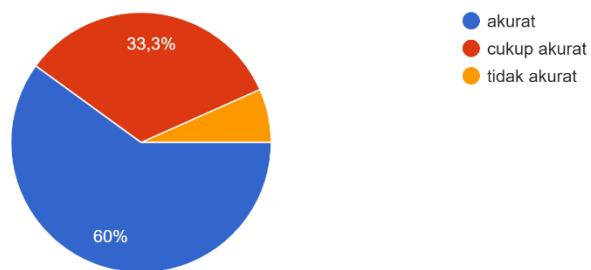
- Mudah: 86.7%
- Cukup Sulit: 6.7%
- Sulit: 6.7%

Sebagian besar responden merasa mudah dalam proses mengunggah gambar (86.7%).

7) Seberapa akurat hasil deteksi yang ditampilkan?

Seberapa akurat hasil deteksi yang ditampilkan?

15 jawaban



Gambar 46 Presentase Pertanyaan 5

- Akurat: 60%
- Cukup Akurat: 33.3%
- Tidak Akurat: 6.7%

Sebagian besar responden merasa hasil deteksi cukup akurat (60%), dan ada yang merasa hanya cukup akurat (33.3%), meskipun ada yang merasa tidak akurat(6.7).

8) Bagaimana penilaian keseluruhan Anda terhadap program ini?



Gambar 47 Presentase Pertanyaan 6

- Sangat Puas: 66.7%
- Cukup Puas: 26.7%
- Tidak Puas: 6.7%

Sebagian besar responden sangat puas dengan program ini (66.7%).

Secara umum, program ini mendapatkan penilaian positif dari sebagian besar responden. Sebagian besar responden merasa program ini mudah digunakan, fitur-fitur mudah ditemukan, dan mereka puas dengan hasil deteksi yang akurat. Namun, ada beberapa area yang perlu diperbaiki, terutama terkait kemudahan menemukan fitur dan meningkatkan akurasi hasil deteksi.

BAB IV

PENUTUP

3.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut:

- a) Mayoritas responden merasa bahwa program ini mudah digunakan. Sebagian besar responden menemukan fitur untuk menuju halaman klasifikasi dengan mudah (80%) dan menemukan fitur untuk mengunggah gambar dengan mudah (73.3%). Selain itu, proses mengunggah gambar juga dirasa mudah oleh 86.7% responden.
- b) Sebagian besar responden merasa hasil deteksi program ini akurat (60%), dengan 33.3% responden merasa hasilnya cukup akurat. Namun, ada juga 6.7% yang merasa hasil deteksi tidak akurat.
- c) Program ini mendapatkan penilaian positif dari sebagian besar responden dengan 66.7% merasa sangat puas dan 26.7% merasa cukup puas. Hanya 6.7% responden yang merasa tidak puas dengan program ini.

3.2. Saran

Berdasarkan kesimpulan di atas, beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut adalah:

- a) Perlu adanya perbaikan dan penyederhanaan antarmuka pengguna agar fitur-fitur lebih mudah ditemukan oleh semua pengguna. Hal ini penting untuk mengurangi jumlah responden yang merasa cukup sulit menemukan fitur tertentu.
- b) Meskipun sebagian besar responden merasa akurasi deteksi sudah cukup baik, namun masih ada yang merasa hasilnya tidak akurat. Oleh karena itu, perlu dilakukan peningkatan pada algoritma deteksi agar hasil yang diberikan lebih akurat.
- c) Untuk membantu pengguna yang merasa kesulitan dalam menggunakan program, dapat disediakan pelatihan atau panduan penggunaan yang lebih detail. Hal ini dapat berupa video tutorial atau dokumentasi yang mudah diakses oleh pengguna.
- d) Melakukan iterasi dan perbaikan berdasarkan feedback dari pengguna secara berkala. Mengadakan survei kepuasan pengguna setelah setiap update atau perbaikan dapat membantu mengidentifikasi area yang masih memerlukan peningkatan lebih lanjut.

DAFTAR PUSTAKA

- Aini, Nur, Wiwin Handoko, dan Rizky Nurhaliza. 2024. "JUTSI: Jurnal Teknologi dan Sistem Informasi PREDIKSI PENERIMAAN BANTUAN PIP PADA SMKS AL-FURQON BATUBARA DENGAN METODE NAÏVE BAYES." 4(1). doi: 10.33330/jutsi.v4i1.2971.
- Aiyegbusi, Olalekan Lee. 2020. "Key methodological considerations for usability testing of electronic patient-reported outcome (ePRO) systems." *Quality of Life Research* 29(2):325–33.
- Akbar, Stefano, Muhammad Syaiful Qisam, dan Gladyns Anandita Yasmin. t.t. *Identifikasi Kanker Paru-Paru Menggunakan Metode Ekualisasi Histogram dan LBP AFILIASI*.
- AKGÜN, Devrim. 2021. "A TensorFlow implementation of Local Binary Patterns Transform." *MANAS Journal of Engineering* 9(1):15–21. doi: 10.51354/mjen.822630.
- Albashish, Dheeb, Mohammad Hashem Ryalat, dan Nedaa Ahmad Almansour. t.t. *XXX-X-XXXX-XXXX-X/XX/\$XX.00 ©20XX IEEE Deep CNN Model based on VGG16 for Breast Cancer Classification*.
- Antara, Alecia Maharani Ektya, Syafrina Aulia Sari, Nita Riswanti, Dhestyara Alivia Amin, Vebi Verdila, dan Amin Padmo Azam Masa. 2023. "Deteksi Nominal Rupiah Uang Kertas Berdasarkan Citra Warna Menggunakan Segmentasi K-Means Clustering dan Klasifikasi Random Forest." *Kreatif Teknologi dan Sistem Informasi (KRETISI)* 1(1):34–39. doi: 10.30872/kretisi.v1i1.776.
- Aprillia, Dede, Tatang Rohana, Tohirin Al Mudzakir, dan Deden Wahiddin. 2024. "KLIK: Kajian Ilmiah Informatika dan Komputer Deteksi Nominal Mata Uang Rupiah Menggunakan Metode Convolutional Neural Network dan Feedforward Neural Network." *Media Online* 4(4). doi: 10.30865/klik.v4i4.1711.
- Chen, Hong, Songhua Hu, Rui Hua, dan Xiuju Zhao. 2021. "Improved naive Bayes classification algorithm for traffic risk management." *Eurasip Journal on Advances in Signal Processing* 2021(1). doi: 10.1186/s13634-021-00742-6.
- Desi Handayani, Syifa Nurulia, dan Udin Saripudin. 2024. "Relevansi Konsep Uang Perspektif Ibnu Miskawaih di Era Modernisasi." *Al-Kharaj: Jurnal Ekonomi, Keuangan & Bisnis Syariah* 6(5). doi: 10.47467/alkharaj.v6i5.1829.

- Dong, Fangchun, Jingbing Li, Uzair Aslam Bhatti, Jing Liu, Yen Wei Chen, dan Dekai Li. 2023. "Robust Zero Watermarking Algorithm for Medical Images Based on Improved NasNet-Mobile and DCT." *Electronics (Switzerland)* 12(16). doi: 10.3390/electronics12163444.
- Dwi Fajar Maulana Perwakilan Badan Kependudukan dan Keluarga Berencana Nasional Provinsi Papua Jl Abepura Kotaraja, Yudhi, dan Kota Jayapura. 2021. *PERUBAHAN TOPIK MEDIA TENTANG PROGRAM KELUARGA BERENCANA PASCAREBRANDING BKKBN: PENDEKATAN PEMODELAN TOPIK DIGITAL*. Vol. 6.
- Eka, Alvin, Putra #1, Mohammad Farid, Naufal #2, Vincentius Riandaru, Prasetyo #3, Jl Raya Kalirungkut, Kali Rungkut, Kec Rungkut, Kota Sby, dan Jawa Timur. 2023a. "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning."
- Eka, Alvin, Putra #1, Mohammad Farid, Naufal #2, Vincentius Riandaru, Prasetyo #3, Jl Raya Kalirungkut, Kali Rungkut, Kec Rungkut, Kota Sby, dan Jawa Timur. 2023b. "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning."
- Faridj, Miftah, dan Abadi Informatika. t.t. *PENYUSUNAN LAMA STUDI MAHASISWA MENGGUNAKAN ATURAN K-NEAREST NEIGHBOR*. Vol. 2.
- Gapsari, Femiana, Fitri Utaminingrum, Chin Wei Lai, Khairul Anam, Abdul M. Sulaiman, Muhamad F. Haidar, Tobias S. Julian, dan Eno E. Ebenso. 2024. "A convolutional neural network -VGG16 method for corrosion inhibition of 304SS in sulfuric acid solution by timoho leaf extract." *Journal of Materials Research and Technology* 30:1116–27. doi: 10.1016/j.jmrt.2024.03.156.
- Gomathy, C. K., dan A. Rohith Naidu Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya. 2021. "THE PREDICTION OF DISEASE USING MACHINE LEARNING." *International Journal of Scientific Research in Engineering and Management*.
- Hartati, Ery, Kata Kunci, dan Kupu Kupu. t.t. *MDP STUDENT CONFERENCE (MSC) 2022 Klasifikasi Spesies Kupu Kupu Menggunakan Metode Convolutional Neural Network*.
- Hikmatia, Nasha, dan Muhammad Ihsan Zul. 2021. *Jurnal Politeknik Caltex Riau*. Vol. 7.

- Khudhair, Zaid Nidhal, Ahmed Nidhal Khdiar, Nidhal K. El Abbadi, Farhan Mohamed, Tanzila Saba, Faten S. Alamri, dan Amjad Rehman. 2023. "Color to Grayscale Image Conversion Based on Singular Value Decomposition." *IEEE Access* 11:54629–38. doi: 10.1109/ACCESS.2023.3279734.
- Kim, Taeheung, dan Jong Seok Lee. 2022. "Exponential Loss Minimization for Learning Weighted Naive Bayes Classifiers." *IEEE Access* 10:22724–36. doi: 10.1109/ACCESS.2022.3155231.
- Malik Ibrahim, Muhamad, Reni Rahmadewi, dan Lela Nurpulaela. 2023a. *PENDETEKSIAN NOMINAL UANG PADA GAMBAR MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK: INTEGRASI METODE PRA-PEMROSESAN CITRA DAN KLASIFIKASI BERBASIS CNN*. Vol. 7.
- Malik Ibrahim, Muhamad, Reni Rahmadewi, dan Lela Nurpulaela. 2023b. *PENDETEKSIAN NOMINAL UANG PADA GAMBAR MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK: INTEGRASI METODE PRA-PEMROSESAN CITRA DAN KLASIFIKASI BERBASIS CNN*. Vol. 7.
- Mardha, Fitria Adyati, Salwa Ziada Salsabiila, Siti Khaalishah Sayid, dan Weni Ariska. 2022. *IDENTIFIKASI NILAI MATA UANG KERTAS RUPIAH DENGAN METODE EKSTRASI CIRI LOCAL BINARY PATTERN DAN METODE KLASIFIKASI NAÏVE BAYES*.
- Musa, Purnawarman, Wahid Khairul Anam, Saiful Bahri Musa, Witari Aryunani, Remi Senjaya, dan Puji Sularsih. 2023. "Pembelajaran Mendalam Pengklasifikasi Ekspresi Wajah Manusia dengan Model Arsitektur Xception pada Metode Convolutional Neural Network." *Rekayasa* 16(1):65–73. doi: 10.21107/rekayasa.v16i1.16974.
- Nafi'iyah, Nur, Chastine Fatichah, Darlis Herumurti, Eha Renwi Astuti, dan Ramadhan Hardani Putra. 2023. "MobileNetV2 Ensemble Segmentation for Mandibular on Panoramic Radiography." *International Journal of Intelligent Engineering and Systems* 16(2):546–60. doi: 10.22266/ijies2023.0430.45.
- Nti, Isaac Kofi, Owusu Nyarko-Boateng, dan Justice Aning. 2021. "Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation." *International Journal of Information Technology and Computer Science* 13(6):61–71. doi: 10.5815/ijitcs.2021.06.05.
- Nur Cahyo, Dwi, Hani Zulfia Zahro, dan Nurlaily Vendyansyah. 2023. *PENGENALAN EKSPRESI MIKRO WAJAH DENGAN EKSTRAKSI FITUR PADA KOMPONEN WAJAH MENGGUNAKAN METODE LOCAL BINARY PATTERN HISTOGRAM*. Vol. 7.

- Nurainun, Nurainun, Elin Haerani, Fadhilah Syafria, dan Lola Oktavia. 2023. “Penerapan Algoritma Naïve Bayes Classifier Dalam Klasifikasi Status Gizi Balita dengan Pengujian K-Fold Cross Validation.” *Journal of Computer System and Informatics (JoSYC)* 4(3):578–86. doi: 10.47065/josyc.v4i3.3414.
- Perez, Joann Galopo, dan Eugene S. Perez. 2021. “Predicting Student Program Completion Using Naïve Bayes Classification Algorithm.” *International Journal of Modern Education and Computer Science* 13(3):57–67. doi: 10.5815/IJMECS.2021.03.05.
- Rahman, Sayuti, Arnes Sembiring, Dodi Siregar, Husnul Khair, Gusti Prahmana, Ratih Puspadiini, dan Muhammad Zen. t.t. *PYTHON: DASAR DAN PEMROGRAMAN BERORIENTASI OBJEK TAHTA MEDIA GROUP*.
- Rambe, Bhakti Helvi, Rahmadani Pane, Deci Irmayani, Marnis Nasution, Ibnu Rasyid Munthe, Fakultas Ekonomi, dan Dan Bisnis. 2020. *UML Modeling and Black Box Testing Methods in the School Payment Information System*. Vol. 4.
- Sahro Winanto, Tawang, Chaerur Rozikin, dan Asep Jamaludin. 2023. *Analisa Performa Arsitektur Transfer Learning Untuk Mengidentifikasi Penyakit Daun Pada Tanaman Pangan*. Vol. 7.
- Saputra, Arif Azani, Abram George Stephen Pakpahan, Ahmad Kurtubi, Abdullah Amiruddin, Bagas Fridaniarta, Erdo Yusuf Wicaksono, Hasbi Saputra, Muhammad Yudha Arsyia Putra, dan Rahma Yuni Azahra. 2023. “PELATIHAN DAN PEMBUATAN WEBSITE MENGGUNAKAN HTML DAN CSS.” *Beujroh: Jurnal Pemberdayaan dan Pengabdian pada Masyarakat* 1(1):119–25. doi: 10.61579/beujroh.v1i1.41.
- Sathishkumar, R., B. Vinothini, N. Rajasri, dan M. Govindarajan. t.t. *International Journal of Computing and Digital Systems Detection and Classification of Breast cancer from Ultrasound Images using NASNet Model*.
- Spandana, Ponnathota, Sathiya Narayanan Sekar, dan Badrish Vasu. t.t. *BREAST CANCER DETECTION USING DEEP LEARNING*.
- Sri Rahmadhani, Ummi, dan Noveri Lysbetti Marpaung. 2023. “Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN.” 8(2).
- Tembusai, Zoelkarnain Rinanda, Herman Mawengkang, dan Muhammad Zarlis. 2021. “K-Nearest Neighbor with K-Fold Cross Validation and Analytic Hierarchy Process on Data Classification.” *International Journal of Advances in Data and Information Systems* 2(1). doi: 10.25008/ijadis.v2i1.1204.

Vierisyah, Aldo, dan Ricky Maulana Fajri. t.t. *KLASIFIKASI KANKER PARU PARU MENGGUNAKAN CNN DENGAN 5 ARSITEKTUR.*

Wickramasinghe, Indika, dan Harsha Kalutarage. 2021. “Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation.” *Soft Computing* 25(3):2277–93. doi: 10.1007/s00500-020-05297-6.

Winiarti, Sri, Ilyas Faisal, Universitas Ahmad Dahlan Yogyakarta Kampus, dan Uad Ringroad Selatan. 2024. “Particle Swarm Optimization Algorithm for Hyperparameter Convolutional Neural Network and Transfer Learning VGG16 Model Murinto.” *Journal of Computer Science, Information Technology and Telecommunication Engineering (JCoSITTE)* 5(1):474–80. doi: 10.30596/jcositte.v5i1.16680.

Zalukhu, Agustinus, Swingly Purba, Dedi Darma, Agustinus Zalukhu¹, Swingly Purba², Dedi Darma³, Mahasiswa Teknik Informatika, dan Fakultas Teknologi Industri. 2023. “PERANGKAT LUNAK APLIKASI PEMBELAJARAN FLOWCHART.” *Jurnal Teknologi Informasi dan Industri* 4(1).

LAMPIRAN 1

Lampiran 1 Evaluasi Pengerjaan Tugas Besar

Kelompok 1 B:

1. HALDI : BAB 3, BAB 4, Aplikasi Web.
2. MARTINA : BAB 2.
3. ARLINCE : BAB 1.
4. DESYANA : -

Mengetahui Dosen Pengganti Mata Kuliah
Mata Kuliah Algoritma Pemrograman 2

FAJAR R. B PUTRA, S.Kom., M.Kom.

LAMPIRAN 2

Lampiran 2 Dokumentasi Kelompok

Kelompok 1 B:



LAMPIRAN 3

Lampiran 3 Berisi link GitHub

Kelompok 1 B:

Berikut link GitHub dari program ini:

<https://github.com/haldi-jr20/MACHINE-LEARNING-KLASIFIKASI-MATA-UANG-RUPIAH-MENGGUNAKAN-ALGORITMA-CONVOLUTIONAL-NEURAL-NETWORK>

FORM PENGISIAN PENGERJAAN TUGAS BESAR

Jenis tugas: *Machine Learning* Klasifikasi Mata Uang Rupiah Menggunakan Algoritma *Convolutional Neural Network* (CNN)

Tabel 10 Pengisian pengerajan tugas besar

NO	Hari/Tanggal	Kegiatan	Paraf
1	12/06/2024	Membuat BAB I	
2	14/06/2024	penambahan 20 jurnal BAB II	
3	19/06/2024	Melanjutkan BAB II	
4	20/06/2024	Membuat aplikasi web	
4	22/06/2024	Membuat sub bab Analisa data BAB III	
5	23/06/2024	Membuat flowchart sistem BAB III	
6	25/06/2024	Membuat hasil klasifikasi BAB III	
7	01/07/2024	Membuat implementasi sistemBAB III	
8	03/07/2024	Membuat pengujian sistem BAB III	
9	05/07/2024	Menguji sistem dengan usability testing BAB III	
10	06/07/2024	Menambahkan hasil usability testing BAB III	
11	08/07/2024	Membuat kesimpulan dan saran BAB IV	
12	09/07/2024	Menyelesaikan BAB IV	
13	17/07/2024	Upload project ke GitHub	