

# Deteksi Nominal Mata Uang Rupiah Menggunakan Metode Convolutional Neural Network dan Feedforward Neural Network

Dede Aprillia<sup>1,\*</sup>, Tatang Rohana, Tohirin Al Mudzakir, Deden Wahiddin

Fakultas Ilmu Komputer, Teknik Informatika, Universitas Buana Perjuangan Karawang, Karawang, Indonesia

Email: <sup>1,\*</sup>if19.dedeaprillia@mhs.ubpkarawang.ac.id, <sup>2</sup>tatang.rohana@ubpkarawang.ac.id, <sup>3</sup>tohirin@ubpkarawang.ac.id,

<sup>4</sup>deden.wahiddin@ubpkarawang.ac.id

Email Penulis Korespondensi: if19.dedeaprillia@mhs.ubpkarawang.ac.id

**Abstrak**—Penelitian ini bertujuan untuk mengembangkan sistem deteksi nominal mata uang Rupiah tahun emisi 2022 menggunakan metode Convolutional Neural Network (CNN) dan Feedforward Neural Network (FNN), khususnya dalam konteks penerapan bagi vending machine. Penelitian ini menjelajahi potensi teknologi computer vision untuk memfasilitasi pengenalan uang kertas Rupiah dan memberikan kontribusi pada pengembangan vending machine. Dataset yang digunakan mencakup variasi kondisi pencahayaan, orientasi, dan posisi uang kertas, sehingga melibatkan proses augmentasi dan preprocessing yang beragam. Hasil evaluasi model mencakup akurasi deteksi nominal pada berbagai kondisi, mempertimbangkan keberhasilan sistem untuk mendukung performa dari vending machine. Penelitian ini diharapkan dapat memberikan kontribusi pada pengembangan teknologi yang lebih komprehensif dan memperluas penerapan CNN dan FNN dalam konteks deteksi mata uang. Pada penelitian ini, metode CNN menghasilkan akurasi terbaik sebesar 100% untuk pengujian dengan kondisi terang, lalu pada kondisi cukup cahaya menghasilkan akurasi sebesar 96,43%. Sedangkan pengujian dengan kondisi gelap mendapatkan hasil yang cukup rendah hanya 78,56%. Lalu metode FNN menghasilkan akurasi yang sama sebesar 53,57% pada kondisi cahaya terang, cukup cahaya, dan kurang cahaya.

**Kata Kunci:** Jaringan Saraf Konvolusional; Visi Komputer; Mata Uang; Deteksi; Rupiah

**Abstract**—This research aims to develop a nominal detection system for the Rupiah currency for the 2022 emission year using the Convolutional Neural Network (CNN) and Feedforward Neural Network (FNN) methods, especially in the context of applications for vending machines. This research explores the potential of computer vision technology to facilitate the introduction of Rupiah banknotes and contribute to the development of vending machines. The dataset used includes variations in lighting conditions, orientation, and position of banknotes, thus involving various augmentation and preprocessing processes. The model evaluation results include nominal detection accuracy in various conditions, considering the success of the system to support the performance of the vending machine. This research is expected to contribute to the development of more comprehensive technology and expand the application of CNN and FNN in the context of currency detection. In this research, the CNN method produced the best accuracy of 100% for testing in bright conditions, then in sufficient light conditions it produced an accuracy of 96.43%. Meanwhile, testing in dark conditions got quite low results, only 78.56%. Then the FNN method produces the same accuracy of 53.57% in bright light, sufficient light and low light conditions.

**Keywords:** Convolutional Neural Network; Computer Vision; Currency; Detection; Rupiah

## 1. PENDAHULUAN

Di Indonesia, Rupiah (IDR) merupakan mata uang yang sah untuk dipakai pada proses jual beli. Pada tahun 2022, Pemerintah Republik Indonesia memperkenalkan uang kertas Rupiah baru dengan berbagai denominasi, yang mencakup perubahan desain dan fitur keamanan. Kendati perubahan ini bertujuan meningkatkan keamanan dan mencegah pemalsuan, hal tersebut juga menimbulkan tantangan bagi vending machine yang harus dapat secara akurat dan efisien mendeteksi dan memproses uang kertas yang baru [1].

Pentingnya kemampuan vending machine untuk mendeteksi nominal mata uang Rupiah dengan tepat muncul karena vending machine menjadi salah satu alat transaksi otomatis yang populer dalam berbagai sektor, termasuk di tempat-tempat umum, perkantoran, dan pusat perbelanjaan. Seiring dengan berkembangnya teknologi dan meningkatnya penggunaan uang kertas Rupiah yang baru, keberlanjutan operasi vending machine menjadi tergantung pada kemampuannya untuk beradaptasi dengan perubahan tersebut [2].

Machine learning memberikan kontribusi besar dalam mendeteksi mata uang pada vending machine dengan kemampuannya mengenali pola visual dan karakteristik unik pada uang kertas, termasuk adaptasi cepat terhadap perubahan desain dan denominasi baru. Hal ini meningkatkan akurasi deteksi, mencegah kesalahan, dan mengoptimalkan pengalaman pengguna dengan transaksi yang lebih efisien. Dengan memanfaatkan teknologi ini, vending machine menjadi lebih cerdas, responsif, dan dapat menjawab tantangan perubahan mata uang dengan lebih efektif [3].

Penelitian untuk deteksi nominal mata uang kertas Rupiah telah dilakukan oleh [4], penelitian tersebut menggunakan metode CNN dengan teknik labeling anotasi, namun hasil rata-rata pengujian menunjukkan akurasi yang sangat rendah hanya sebesar 57.5%. Selanjutnya, penelitian telah dilakukan oleh [5], hasil akurasi sebesar 100% pada model menunjukkan hasil yang baik, namun pengujian hanya dilakukan dengan 3 nominal mata uang yaitu Rp.50.000, Rp.75.000, dan Rp.100.000, lalu pengujian juga hanya dilakukan dengan 1 kondisi, yaitu kondisi terang. Sehingga hal tersebut perlu dilakukan pengujian lebih lanjut untuk menguji validitas dari model yang telah dilatih. Lalu pada 2020 penelitian dilakukan oleh [6], penelitian tersebut hanya menggunakan ekstraksi fitur warna Red, Green, Blue (RGB) sebagai metode yang dipakai, akurasi yang didapat sebesar 93,7%. Walaupun hasil tersebut lumayan tinggi, tetapi tingkat akurasi akan menjadi sangat rendah apabila dites pada kondisi kurang cahaya. Selanjutnya penelitian untuk deteksi nominal mata uang juga pernah dilakukan oleh [7], dimana peneliti menggunakan neural network dengan hasil akurasi

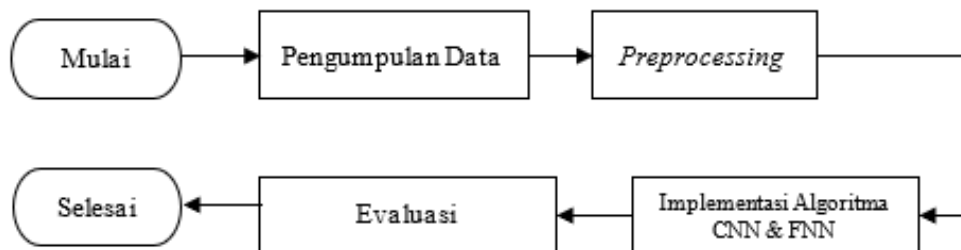
sebesar 97,6%, namun pada penelitian tersebut learning rate yang digunakan pada neural network terlalu tinggi yaitu sebesar 0,8. Hal tersebut dapat mengakibatkan model machine learning mengalami overfitting.

Penelitian ini bertujuan untuk membandingkan sistem deteksi nominal mata uang Rupiah menggunakan Convolutional Neural Network (CNN) dan Feedforward Neural Network (FNN) sebagai solusi untuk mendukung vending machine dalam mengenali nilai mata uang Rupiah tahun emisi 2022. Dengan implementasi sistem ini, diharapkan bahwa vending machine dapat beroperasi dengan performa tinggi dan memperoleh efisiensi kecepatan untuk mengenali mata uang pada saat ada transaksi. Pada penelitian ini, arsitektur CNN yang digunakan yaitu VGGNet dengan menggunakan optimizer Adam untuk mengoptimalkan proses pelatihan model dengan mengurangi nilai fungsi kerugian (loss function) pada data latih.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Penelitian ini dilakukan dengan menjalankan beberapa tahapan yaitu pengumpulan data, preprocessing, implementasi algoritma CNN & FNN, dan yang terakhir evaluasi. Alur tahapan tersebut ditunjukkan pada gambar 1.



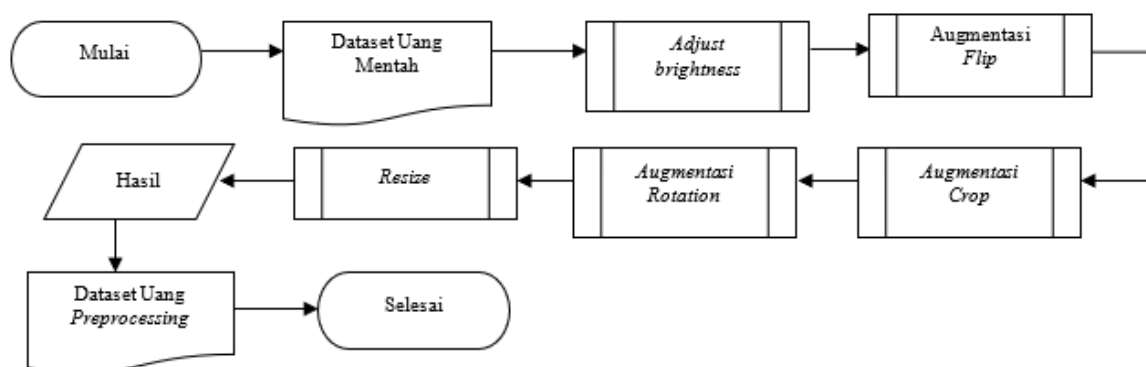
**Gambar 1.** Tahapan Penelitian.

### 2.2 Pengumpulan Data

Tahap pengumpulan data menjadi langkah yang sangat krusial dalam suatu penelitian, dimana data yang dikumpulkan menjadi bahan utama yang menjadi inti dari fokus penelitian [8]. Pada penelitian ini, data yang digunakan adalah mata uang kertas Indonesia tahun emisi 2022 dengan pecahan Rp. 1.000, Rp. 2.000, Rp. 5000, Rp. 10.000, Rp. 20.000, Rp. 50.000, dan Rp. 100.000. Dataset terdiri dari kedua sisi mata uang, yaitu bagian depan dan bagian belakang.

### 2.3 Preprocessing

Tahap preprocessing merupakan tahapan awal untuk mempersiapkan data agar model yang dibuat bisa lebih baik pada saat Training maupun testing [9]. Pada penelitian ini dilakukan adjust brightness, berbagai teknik augmentasi, serta resize gambar seperti yang diperlihatkan pada alur preprocessing gambar 2.



**Gambar 2.** Alur Preprocessing

Pada gambar 2, langkah awal yaitu menerima dataset uang mentah (raw data). Selanjutnya, dilakukan penyesuaian kecerahan (adjust brightness), augmentasi dengan flip, crop, dan rotasi untuk meningkatkan variasi data. Proses ini diikuti dengan resizing untuk memastikan konsistensi ukuran. Hasil dari langkah-langkah ini adalah dataset uang yang telah melalui serangkaian transformasi preprocessing. Terakhir menyimpan dataset tersebut sebagai output dari proses preprocessing.

### 2.4 Implementasi Algoritma

#### 2.4.1 Implementasi Algoritma CNN

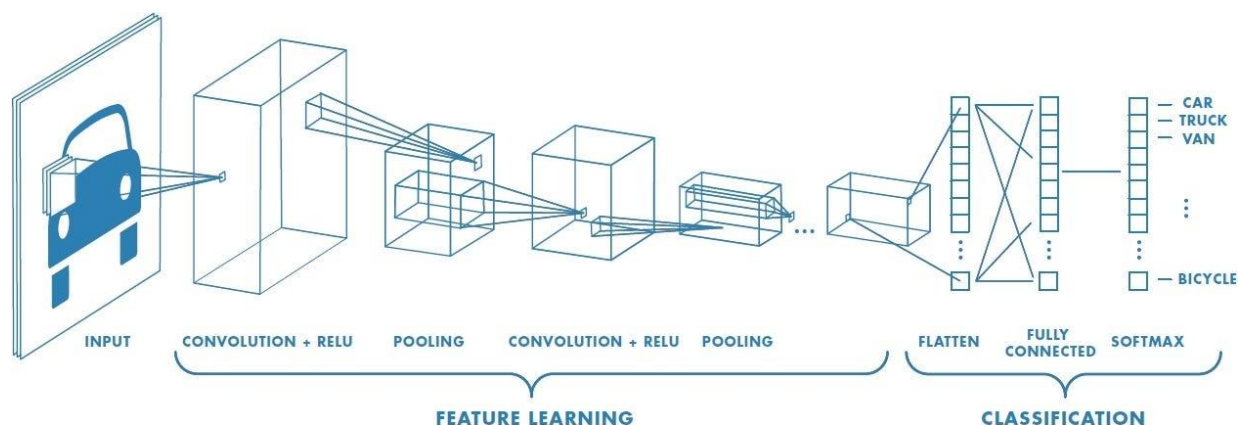
Setelah melewati tahap preprocessing, dataset bertambah jumlahnya dari 14 data citra menjadi 1120 data citra. Pada tahap implementasi algoritma CNN ini terdiri dari proses Modeling dan Training model. Convolutional Neural Network (CNN) telah terbukti menjadi alat yang efektif dalam berbagai aplikasi pengolahan gambar, termasuk pengenalan objek dan pengenalan pola. CNN dapat dilatih untuk mengenali dan membedakan berbagai nilai nominal mata uang dengan tingkat akurasi yang tinggi, sehingga dapat menjadi alat bantu yang sangat berguna bagi individu dengan disabilitas visual [4].

Dalam ranah deep learning, Convolutional Neural Network (CNN/ConvNet) merupakan bagian integral dari deep neural network, suatu tipe jaringan saraf tiruan yang umumnya digunakan untuk mengenali dan memproses gambar [10]. Algoritma ini dirancang khusus untuk mengolah data piksel dan citra visual. CNN memiliki neuron yang dibentuk untuk meniru fungsi lobus frontal, terutama bagian visual cortex pada otak manusia dan hewan. Visual cortex bertanggung jawab atas pemrosesan informasi dalam bentuk rangsangan visual, sehingga membuat CNN sangat efektif dalam tugas pemrosesan gambar dibandingkan dengan algoritma neural network serupa [11].

Lapisan-lapisan dalam arsitektur CNN terdiri dari susunan neuron dengan tiga dimensi yang melibatkan lebar, tinggi, dan kedalaman. Dimensi lebar dan tinggi menunjukkan ukuran dari lapisan tersebut, sedangkan dimensi kedalaman mengacu pada jumlah lapisan yang ada. Sebuah jaringan CNN dapat terdiri dari puluhan hingga ratusan lapisan, dan setiap lapisan tersebut bertugas untuk mempelajari berbagai fitur dalam gambar. Ketika mengolah citra, proses ini diterapkan pada setiap citra pelatihan dengan resolusi yang berbeda, dan hasil keluaran dari setiap citra yang telah diolah digunakan sebagai input untuk lapisan berikutnya dalam jaringan. Selama proses pengolahan citra, fitur-fitur awal yang dapat berupa informasi sederhana seperti kecerahan dan tepi dapat meningkat dalam kompleksitasnya seiring dengan meningkatnya kedalaman lapisan, hingga akhirnya fitur-fitur tersebut menjadi unik dan mampu mendeteksi objek berdasarkan ketebalan lapisan yang bersangkutan [12].

Secara umum, dalam CNN, terdapat dua tipe lapisan yang memainkan peran khusus. Lapisan pertama disebut sebagai lapisan feature extraction, dan terletak di awal arsitektur. Lapisan ini terdiri dari beberapa lapisan, dengan setiap lapisan yang terhubung secara lokal ke daerah-daerah tertentu pada lapisan sebelumnya [13]. Lapisan jenis pertama ini dikenal sebagai convolutional layer, sementara lapisan kedua disebut sebagai pooling layer. Setiap lapisan ini menerapkan fungsi aktivasi bergantian antara jenis pertama dan jenis kedua. Lapisan ekstraksi fitur ini menerima input gambar secara langsung dan mengolahnya hingga menghasilkan keluaran berupa vektor, yang selanjutnya digunakan pada lapisan berikutnya [14].

Sementara itu, lapisan kedua adalah lapisan klasifikasi, yang terdiri dari beberapa lapisan dan setiap lapisan terhubung sepenuhnya dengan lapisan lainnya. Layer ini menerima input berupa vektor hasil keluaran dari lapisan ekstraksi fitur gambar. Input ini kemudian mengalami transformasi yang melibatkan Multi Neural Networks dengan tambahan beberapa hidden layer. Hasil keluaran dari lapisan ini adalah tingkat akurasi kelas yang digunakan untuk tujuan klasifikasi [15].



**Gambar 3.** Ilustrasi CNN.

Berdasarkan ilustrasi CNN pada gambar 3, tahapan pertama dalam struktur CNN adalah langkah konvolusi yang mengoperasikan kernel dengan ukuran tertentu. Jumlah kernel yang diterapkan bervariasi tergantung pada banyaknya fitur yang dihasilkan. Selanjutnya, tahapan berikutnya adalah mencapai fungsi aktivasi, yang umumnya memanfaatkan fungsi aktivasi ReLU (Rectifier Linear Unit). Setelah melewati proses aktivasi, langkah selanjutnya adalah melalui proses penggabungan (pooling). Proses ini diulang beberapa kali hingga terbentuk peta fitur yang memadai untuk diteruskan ke jaringan saraf terhubung penuh (fully connected neural network). Dari jaringan terhubung penuh (fully connected layer) ini, menghasilkan keluaran kelas yang sesuai [16].

#### **2.4.2 Implementasi Algoritma FNN**

Pada tahap implementasi algoritma FNN ini terdiri dari proses Modeling dan Training model. Feedforward Neural Network adalah kerangka kerja yang sangat serbaguna untuk mengembangkan model non-linier dalam tugas seperti regresi, reduksi data, dan pemodelan sistem dinamis non-linier. Model Feedforward Neural Network terdiri dari banyak neuron yang kompleks, yang saling berinteraksi secara rumit dalam lapisan-lapisan yang terstruktur. Feedforward Neural Network memiliki kapasitas untuk mengolah volume data yang besar dan menghasilkan prediksi yang sangat akurat [17].

Metode FNN beroperasi dengan menerima vektor input dan menghasilkan output melalui proses pengolahan input melalui unsur-unsur yang berinteraksi satu sama lain. Proses ini terstruktur dalam beberapa lapisan, di mana input diproses secara berurutan dari satu lapisan ke lapisan berikutnya [18]. Terdapat beberapa faktor yang dapat memengaruhi tingkat akurasi dari metode Feedforward Neural Network, termasuk struktur jaringan, metode pelatihan atau algoritma yang digunakan, dan fungsi aktivasi [19]. Persamaan untuk model FNN dengan p input, satu lapisan tersembunyi dengan m neuron dapat dirumuskan sebagai berikut:

$$f(x_t, v, w) = g_2\left\{\sum_{j=0}^m v_j g_1\left[\sum_{i=0}^p w_{ji} x_{it}\right]\right\} \quad (1)$$

Dimana w sebagai vektor bobot yang menghubungkan lapisan input dan lapisan tersembunyi, dan v sebagai vektor bobot yang menghubungkan lapisan tersembunyi dengan lapisan output (respons). Fungsi aktivasi  $g_1(\cdot)$  dan  $g_2(\cdot)$  digunakan untuk mengubah data pada masing-masing lapisan. Selain itu,  $W_{ji}$  dan  $V_j$  adalah bobot yang diestimasi melalui proses backpropagation [17].

### 2.5 Evaluasi

Tahap evaluasi ditujukan untuk memperoleh hasil akurasi yang telah dilakukan pelatihan pada data Training [20]. Dalam tahap ini, data testing yang digunakan berjumlah 84 data. Model yang telah dilatih dilakukan pengujian menggunakan data testing dengan berbagai kondisi cahaya yaitu terang, cukup cahaya, dan kurang cahaya sebanyak 4 kali pengujian pada tiap kondisi. Selanjutnya hasil akurasi 4 kali pengujian dibandingkan untuk menyimpulkan hasil akurasi pengujian terbaik.

## 3. HASIL DAN PEMBAHASAN

Bab ini merupakan penjabaran dan analisis dari langkah-langkah yang telah diuraikan dalam metode penelitian.

### 3.1 Pengumpulan Data

Dataset yang digunakan pada penelitian ini merupakan data primer yang diambil menggunakan kamera handphone Iphone X. Proses pengambilan gambar ini dilakukan pada tanggal 4 Mei 2023 dengan mengambil gambar sebanyak 14 gambar. Data yang telah di ambil merupakan kumpulan foto uang kertas Rupiah tahun emisi 2022 bagian depan dan bagian belakang. Gambar 2 merupakan sampel data yang digunakan pada penelitian ini.



**Gambar 2.** (a) (b) Sampel Data Bagian Depan dan Bagian Belakang.

### 3.2 Preprocessing

Pada tahap ini, dataset dilakukan preprocessing untuk memperbanyak jumlah dataset agar meningkatkan performa dari model yang akan dilatih. Berikut merupakan tahapan preprocessing yang dilakukan pada penelitian ini.

#### 3.2.1 Adjust Brightness

Adjust brightness adalah salah satu teknik pemrosesan citra yang digunakan untuk mengubah tingkat kecerahan (brightness) dari sebuah citra digital [21]. Teknik ini memungkinkan pengguna untuk membuat citra menjadi lebih terang atau lebih gelap, yang dapat berguna dalam berbagai konteks, seperti perbaikan kualitas citra, pemrosesan gambar medis, atau penyempurnaan citra sebelum penggunaan dalam aplikasi pengenalan objek atau deteksi pola. Pada penelitian ini, data dilakukan adjust brightness dengan membuat citra lebih terang 25% dari citra normalnya. Berikut merupakan sampel data pada tahap ini yang dilampirkan pada gambar 3.



**Gambar 3.** (a)(b) Sampel Data Sebelum dan Sesudah Adjust Brightness.



### 3.2.2 Augmentasi Flip

Tahap ini terdiri dari horizontal flip dan vertical flip. Dalam augmentasi flip horizontal, citra diputar secara horizontal (dibolak-balik) sehingga objek yang muncul di sisi kiri akan muncul di sisi kanan, dan sebaliknya. Sedangkan pada flip vertikal, citra diputar secara vertikal (dari atas ke bawah) [22]. Hasil dari tahap ini dapat dilihat pada gambar 4.



**Gambar 4.** (a)(b)(c)(d) Sampel Augmentasi Flip.

### 3.2.3 Augmentasi Crop

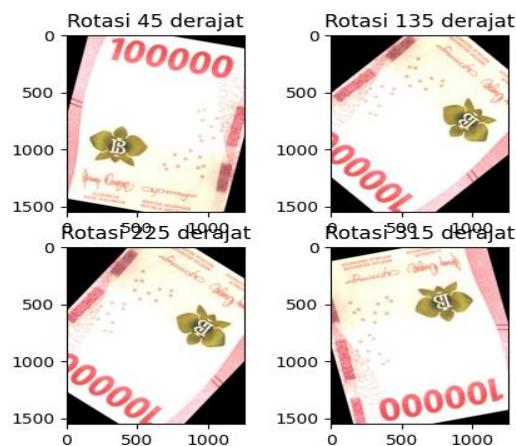
Pada augmentasi crop, setiap gambar dipotong menjadi 3 bagian. Augmentasi crop berguna untuk menghasilkan citra-citra yang berbeda dengan fokus pada area yang berbeda dari gambar asli [23]. Sampel hasil pada tahap ini dapat dilihat pada Gambar 5.



**Gambar 5.** Sampel Hasil Augmentasi Crop.

### 3.2.4 Augmentasi Rotation

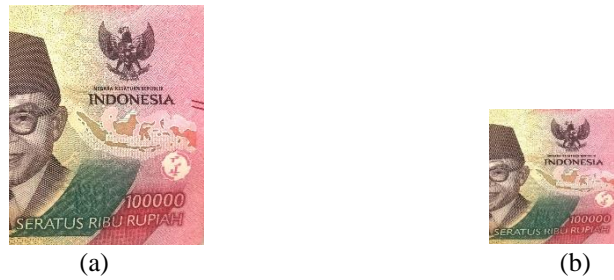
Dalam augmentasi rotasi, citra diputar pada derajat tertentu searah jarum jam atau berlawanan arah jarum jam [24]. Pada penelitian ini setiap gambar dilakukan rotasi sebanyak 4 kali yaitu  $45^\circ$ ,  $135^\circ$ ,  $225^\circ$ , dan  $315^\circ$ . Tahap ini digunakan untuk melatih model agar dapat mengenali objek dalam berbagai orientasi. Berikut merupakan sampel hasil dari Rotation pada gambar 6.



**Gambar 6.** Sampel Hasil Augmentasi Rotation.

### 3.2.5 Augmentasi Resize

Tahapan resize digunakan untuk mengubah ukuran dengan mengurangi (downsampling) atau memperbesar (upsampling) dimensinya [23]. Pada penelitian ini dilakukan downsampling (mengurangi) dimensi gambar menjadi skala  $256 \times 256$  piksel. Proses resizing dapat mempercepat waktu running model karena gambar yang lebih kecil memerlukan waktu komputasi yang lebih sedikit untuk diproses oleh model. Hasil dari tahap ini dapat dilihat pada gambar 7.



Gambar 7. (a)(b) Sampel Hasil Sebelum dan Sesudah Resize.

### 3.3 Implementasi Algoritma CNN

Setelah melewati tahap preprocessing, dataset bertambah jumlahnya dari 14 data citra menjadi 1120 data citra. Pada tahap implementasi algoritma CNN ini terdiri dari Modeling dan Training.

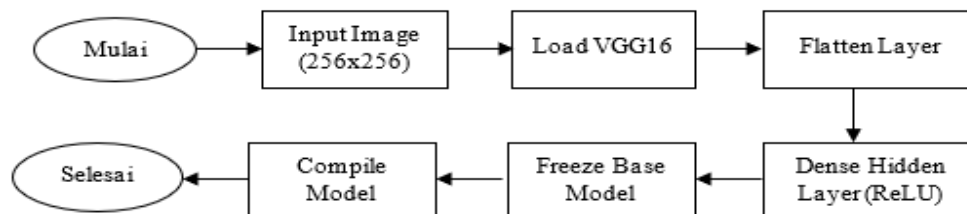
#### 3.3.1 Modelling

Model pada penelitian ini menggunakan salah satu arsitektur CNN yaitu VGGNet yang memiliki 16 lapisan atau biasa disebut VGG16. Arsitektur VGG16 memiliki konfigurasi 13 lapisan konvolusi dan 3 lapisan fully connected di akhir. Langkah pertama, model VGG16 menggunakan parameter “weights='imagenet'” untuk mengindikasikan model telah dilatih dengan dataset besar berbagai jenis gambar oleh library Keras.

Selanjutnya menambahkan lapisan tambahan (fully connected layer) ke model ini. Lapisan yang pertama yaitu Flatten, Lapisan ini membuat data dari model VGG16 menjadi bentuk yang lebih sederhana yaitu mengubah tensor tiga dimensi menjadi vektor satu dimensi. Selanjutnya, sebuah lapisan Dense (fully connected) dengan 128 neuron dan aktivasi ReLU ditambahkan. Lapisan ini bertindak sebagai lapisan tersembunyi untuk memproses fitur-fitur yang diekstraksi dari model VGG16 yang membantu kita mengenali mata uang dengan memahami fitur-fitur penting dalam gambar tersebut.

Lapisan-lapisan di dalam base model (model VGG16) dibekukan (freeze) dengan mengatur atribut “trainable = False”. Hal ini berarti lapisan-lapisan tersebut tidak akan diperbarui selama pelatihan model. Ini dilakukan untuk mempertahankan representasi-fitur yang telah dipelajari oleh VGG16.

Selanjutnya model akhir dikompilasi dengan menggunakan optimizer 'adam', fungsi loss menggunakan 'categorical\_crossentropy' yang cocok untuk masalah klasifikasi multikelas, dan metrik 'accuracy' untuk mengukur kinerja model selama pelatihan. Alur pembuatan model yang telah dibuat dapat dilihat pada gambar 8.



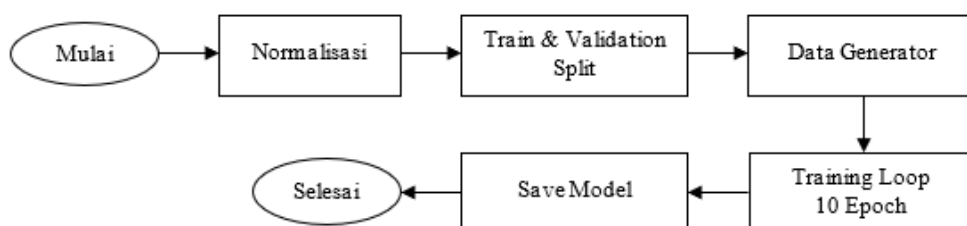
Gambar 8. Alur Modelling CNN.

#### 3.3.2 Training

Langkah awal pada tahap Training yaitu mengubah intensitas piksel citra ke dalam rentang 0-1 menggunakan Library Keras. Setelah itu dilakukan split dataset menjadi 2 bagian yaitu Data Training berjumlah 896 citra dan Data Validasi berjumlah 224 citra, dengan perbandingan 80:20.

Selanjutnya, data generator dibuat untuk Training dan validasi dengan ketentuan citra 256 x 256 piksel, ukuran batch sebanyak 32 citra per batch. Setelah persiapan data, proses pelatihan dilakukan sebanyak 10 epoch yaitu 10 kali perulangan dimana setiap 1 epoch selesai, performa model Training akan diukur dengan data validasi.

Terakhir, model hasil pelatihan disimpan dalam format H5. Hal ini memungkinkan untuk menggunakan model Training ini di masa depan tanpa perlu melatih ulang. Berikut merupakan Gambar 9, alur training model yang telah dibuat.



Gambar 9. Alur Model Training CNN.

Setelah dilakukan pelatihan sebanyak 10 epoch. Model menunjukkan performa akurasi yang meningkat di setiap epoch, dengan akurasi validasi akhir sebesar 84,38% dengan rata-rata waktu komputasi selama 927,8 detik atau sekitar 15 menit untuk setiap epoch. Hasil model Training dilampirkan pada tabel 1.

**Tabel 1.** Hasil Model Training 10 Epoch.

Epoch	Waktu (Detik)	Akurasi	Loss	Akurasi Validasi	Loss Validasi
1	924	0.3181	1.8554	0.2902	1.7688
2	943	0.5212	1.3725	0.3170	1.6094
3	911	0.7746	1.0015	0.5446	1.3497
4	931	0.9375	0.5844	0.6250	1.1155
5	914	0.9754	0.3351	0.7232	0.8740
6	935	0.9933	0.1889	0.7455	0.7971
7	931	1.0000	0.1137	0.7634	0.7244
8	931	1.0000	0.0710	0.7946	0.6249
9	930	1.0000	0.0444	0.8482	0.5261
10	928	1.0000	0.0321	0.8438	0.5590

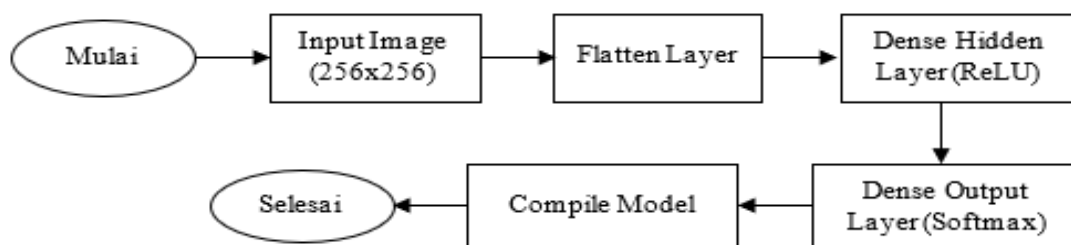
### 3.4 Implementasi Algoritma FNN

Setelah melewati tahap preprocessing, dataset bertambah jumlahnya dari 14 data citra menjadi 1120 data citra. Pada tahap implementasi algoritma FNN ini terdiri dari Modeling dan Training.

#### 3.3.1 Modelling

Pada modelling FNN ini library yang digunakan yaitu Keras. Model FNN adalah jenis jaringan saraf tiruan yang secara berurutan menggabungkan lapisan-lapisan untuk pemrosesan data.

Dalam penelitian ini, model dibuat dengan tiga lapisan utama: lapisan Flatten, lapisan Dense dengan fungsi aktivasi ReLU, dan lapisan Dense terakhir dengan fungsi aktivasi softmax. Lapisan Flatten digunakan untuk mengubah data gambar tiga dimensi (256x256 piksel dengan tiga saluran warna) menjadi vektor satu dimensi. Lapisan Dense tersembunyi memiliki 128 neuron dan bertugas untuk mempelajari pola dalam data. Lapisan Dense terakhir adalah lapisan output yang menghasilkan probabilitas kelas. Dalam kompilasi model, kita menentukan optimizer, fungsi kerugian, dan metrik akurasi untuk proses pelatihan. Alur pembuatan model yang telah dibuat dapat dilihat pada gambar 10.



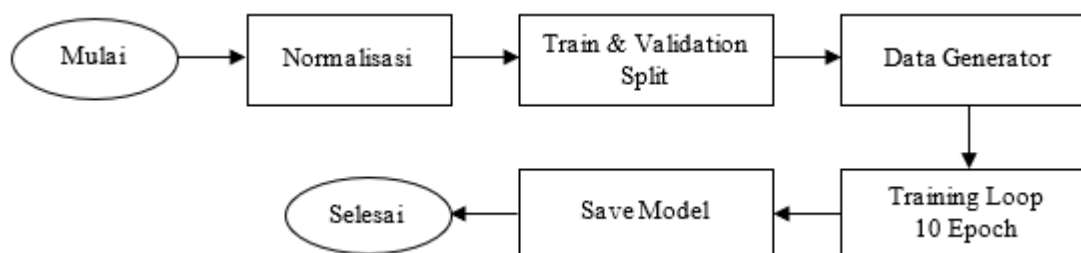
**Gambar 10.** Alur Modelling FNN.

#### 3.3.2 Training

Langkah awal pada tahap Training yaitu melakukan normalisasi/mengubah intensitas piksel citra ke dalam rentang 0-1 menggunakan Library Keras. Setelah itu dilakukan split dataset menjadi 2 bagian yaitu Data Training berjumlah 896 citra dan Data Validasi berjumlah 224 citra, dengan perbandingan 80:20.

Selanjutnya, data generator dibuat untuk Training dan validasi dengan ketentuan citra 256 x 256 piksel dan ukuran batch sebanyak 32 citra per batch. Setelah persiapan data, proses pelatihan dilakukan sebanyak 10 epoch, dimana setiap 1 epoch selesai, performa model Training akan diukur dengan data validasi.

Terakhir, model hasil pelatihan disimpan dalam format H5. Hal ini memungkinkan untuk menggunakan model Training ini di masa depan tanpa perlu mengulang proses training. Berikut merupakan gambar 11, alur training model yang telah dibuat.



**Gambar 11.** Alur Training Model FNN.

Setelah dilakukan pelatihan sebanyak 10 epoch. Model menunjukkan performa akurasi yang meningkat hampir di setiap epoch, dengan akurasi validasi terbaik sebesar 51,34% dengan rata-rata waktu komputasi selama 23,1 detik untuk setiap epoch. Hasil model Training dilampirkan pada tabel 2.

**Tabel 2.** Hasil Model Training FNN 10 Epoch.

Epoch	Waktu (Detik)	Akurasi	Loss	Akurasi Validasi	Loss Validasi
1	21	0.1574	78.5225	0.1429	18.4869
2	19	0.26	8.8339	0.3438	3.9868
3	34	0.433	2.9257	0.4464	2.5147
4	23	0.558	1.3976	0.2411	3.6092
5	23	0.5056	2.3248	0.3304	3.335
6	24	0.5022	2.5031	0.2723	4.9826
7	21	0.5078	2.4648	0.3884	2.9925
8	22	0.6328	1.3023	0.4196	2.7091
9	21	0.7042	0.8857	0.5134	1.9867
10	23	0.6496	1.4588	0.4018	3.9081

### 3.5 Evaluasi

Setelah dilakukan Training, model dilakukan evaluasi dengan diuji menggunakan data baru untuk dibandingkan hasil akurasinya. Data testing pada penelitian ini berjumlah 84 citra yang terdiri dari bagian depan dan bagian belakang untuk setiap nominal mata uang. Dimana 28 citra uji dengan kondisi terang (25% lebih terang dari citra normal), 28 citra uji dengan kondisi cukup cahaya (cahaya normal), dan 28 citra dengan kondisi kurang cahaya (25% lebih gelap dari citra normal). Tahap pengujian ini dilakukan sebanyak 4 kali untuk membandingkan pengujian dengan kondisi terang, cukup cahaya dan kurang cahaya. Berikut merupakan sampel dari data uji pada gambar 12.



**Gambar 12.** (a) (b)(c) Sampel Dataset Testing.

Langkah dalam proses testing adalah sebagai berikut. Pertama, model Training yang telah disimpan dalam file H5 dimuat menggunakan `tf.keras.models.load_model`. Label kelas dari generator saat training juga diambil untuk membantu menafsirkan hasil prediksi. Dalam sebuah loop for, setiap gambar uji dimuat menggunakan `load_img`, diubah menjadi array dengan `img_to_array`, dan diubah bentuk sesuai dengan format input model. Kemudian, dilakukan normalisasi intensitas piksel dengan membagi array dengan 255.0 untuk mengubah gambar ke dalam rentang 0-1. Setelah persiapan gambar uji, model melakukan prediksi menggunakan fungsi `model.predict`.

**Tabel 3.** Hasil Pengujian CNN.

Pecahan (Rp)	Hasil uji berdasarkan kondisi cahaya														
	Terang				Akurasi (%)	Cukup Cahaya				Akurasi (%)	Kurang Cahaya				Akurasi (%)
	1	2	3	4		1	2	3	4		1	2	3	4	
1000	B	B	B	B	100	B	B	B	B	100	B	B	B	B	100
2000	B	B	B	B	100	B	B	B	B	100	B	B	B	B	100
5000	B	B	B	B	100	B	B	B	B	100	B	B	B	B	100
10000	B	B	B	B	100	B	S	B	B	75	S	S	S	S	0
20000	B	B	B	B	100	B	B	B	B	100	B	S	B	B	75
50000	B	B	B	B	100	B	B	B	B	100	S	B	B	B	75
100000	B	B	B	B	100	B	B	B	B	100	B	B	S	B	100
Akurasi (%)	100					96.43					78.57				

Berdasarkan hasil pengujian pada Tabel 3 yang terdiri dari 4 kali percobaan, hasil akurasi tertinggi sebesar 100% didapatkan pada pengujian dengan kondisi terang. Lalu pada pengujian dengan kondisi cukup cahaya mendapatkan hasil akurasi yang cukup baik sebesar 96,43%. Sedangkan pada pengujian dengan kondisi kurang cahaya menghasilkan akurasi terendah hanya 78,57%.

**Tabel 4.** Hasil Pengujian FNN.

Pecahan (Rp)	Hasil uji berdasarkan kondisi cahaya															
	Terang				Akurasi (%)	Cukup Cahaya				Akurasi (%)	Kurang Cahaya				Akurasi (%)	
	1	2	3	4		1	2	3	4		1	2	3	4		



<b>1000</b>	S	B	B	S	50	S	B	B	S	50	S	B	B	S	50
<b>2000</b>	B	B	B	B	100	B	B	B	B	100	B	B	B	B	100
<b>5000</b>	B	B	B	B	100	B	B	B	B	100	B	B	B	B	100
<b>10000</b>	S	S	B	S	25	S	S	B	S	25	S	S	B	S	25
<b>20000</b>	B	S	B	B	75	B	S	B	B	75	B	S	B	B	75
<b>50000</b>	S	S	B	S	25	S	S	B	S	25	S	S	B	S	25
<b>100000</b>	S	S	S	S	0	S	S	S	S	0	S	S	S	S	0
<b>Akurasi (%)</b>	<b>53.57</b>					<b>53.57</b>					<b>53.57</b>				

Berdasarkan hasil pengujian pada Tabel 4 yang terdiri dari 4 kali percobaan, hasil akurasi dengan berbagai kondisi cahaya menunjukkan hasil akurasi yang sama sebesar 53,57%. Walaupun hasil akurasi menggunakan 10 epoch sangat rendah, hal ini menunjukkan bahwa metode FNN dapat digunakan dengan berbagai macam kondisi cahaya tanpa mengurangi akurasinya.

## 4. KESIMPULAN

Penelitian ini menggunakan metode CNN dan FNN untuk mendeteksi nominal mata uang Rupiah tahun emisi 2022. Pada metode tersebut, dataset dilakukan proses preprocessing yang selanjutnya data dibagi menjadi dua bagian yaitu 80% data Training berjumlah 896 data dan 20% data validasi berjumlah 224 data. Selanjutnya dilakukan pengujian pada data testing menggunakan model yang telah dilatih sebelumnya. Pada metode CNN ini hasil tertinggi dari tingkat akurasi, mencapai 100%, diperoleh saat pengujian dalam kondisi cahaya terang. Sementara itu, pada pengujian dengan kondisi cukup cahaya, didapatkan hasil akurasi yang sangat baik sebesar 96,43%. Namun, saat melakukan uji pada kondisi kurang cahaya, didapatkan hasil akurasi terendah, hanya mencapai 78,57%. Sedangkan metode FNN mendapatkan hasil akurasi sangat rendah hanya sebesar 53,57% jika menggunakan 10 epoch, tetapi metode FNN dapat digunakan pada berbagai kondisi cahaya tanpa mengurangi keakuratan dari model. Pada penelitian ini memiliki keterbatasan dalam jumlah dataset hanya berjumlah 14 data, setelah dilakukan berbagai teknik preprocessing, dataset bertambah menjadi 1120 data. Lalu pada penelitian ini dibatasi hanya menggunakan 10 epoch untuk mempersingkat waktu komputasi, karena untuk running dataset citra yang menggunakan warna membutuhkan waktu komputasi yang cukup lama untuk setiap epoch-nya. Dalam penelitian selanjutnya penulis menyarankan untuk melakukan penambahan jumlah dataset, berdasarkan pengujian yang telah dilakukan peneliti, menambahkan dataset dan memperbanyak epoch pada saat Training model bisa meningkatkan hasil akurasi. Selanjutnya, pada pengujian dengan kondisi kurang cahaya, saat melakukan pengujian menggunakan nominal Rp.10.000 hasil pengujiannya salah semua (0%), menurut penulis hal ini perlu dilakukan pengecekan kembali pada dataset apakah perlu ditambahkan dataset dengan nominal 10000 dengan kondisi kurang cahaya atau perlu dilakukan dengan teknik yang lain.

## REFERENCES

- [1] I. Putri, F. Ajiani, B. Dinda, P. Ayu, A. Rahmatyar, and W. D. Isasih, "Edukasi Mata Uang Baru Emisi 2022 dan Menghindari Mata Uang Palsu," *JILPI J. Ilm. Pengabd. dan Inov.*, vol. 1, no. 4, pp. 865–874, 2023.
- [2] N. Ratnasri and T. Sharmilan, "Vending Machine Technologies: A Review Article," *Int. J. Sci. Basic Appl. Res.*, vol. 58, pp. 1–7, 2021.
- [3] A. P. Pujiputra, H. Kusuma, and T. A. Sardjono, "Ultraviolet Rupiah Currency Image Recognition using Gabor Wavelet," *Int. Semin. Intell. Technol. Its Appl.*, pp. 299–303, 2018, doi: 10.1109/ISITIA.2018.8711296.
- [4] M. M. Ibrahim, R. Rahmadewi, L. Nurpulaela, F. Teknik, and U. S. Karawang, "PENDETEKSIAN NOMINAL UANG PADA GAMBAR MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK : INTEGRASI METODE PRA- PEMROSESAN CITRA DAN KLASIFIKASI BERBASIS CNN," vol. 7, no. 2, pp. 1395–1400, 2023.
- [5] O. Ery Pamungkas *et al.*, "Classification of Rupiah to Help Blind with The Convolutional Neural," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 158, pp. 259–268, 2022.
- [6] A. R. Pratama, M. Mustajib, and A. Nugroho, "Deteksi Citra Uang Kertas dengan Fitur RGB Menggunakan K-Nearest Neighbor," *J. Eksplora Inform.*, vol. 9, no. 2, pp. 163–172, 2020, doi: 10.30864/eksplora.v9i2.336.
- [7] L. Kurniawati, K. Sumantri, Risandriya, and H. Wijanarko, "Pendeteksi Nominal Uang Kertas," *J. Appl. Electr. Eng.*, vol. 3, no. December, pp. 2–6, 2019.
- [8] I. Handayani, I. J. Dewanto, and D. Andriani, "Pemanfaatan RinfoForm Sebagai Media Pengumpulan Data Kinerja Dosen," *Technomedia J.*, vol. 2, no. 2, pp. 14–28, 2018, doi: 10.33050/tmj.v2i2.321.
- [9] H. Alamsyah, Y. Cahyana, and A. R. Pratama, "Deteksi Fake Review Menggunakan Metode Support Vector Machine dan Naïve Bayes Di Tokopedia," 2023.
- [10] R. Randel and R. Cornet, *Informatics for Health: Connected Citizen-Led Wellness and Population Health*. Amsterdam: IOS Press, 2017.
- [11] P. A. Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network ( Cnn ) Pada Ekspresi Manusia," *Algor*, vol. 2, no. 1, pp. 12–21, 2020.
- [12] S. Ilahiyah and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.)*, vol. 3, no. 2, pp. 49–56, 2018.
- [13] R. Venkatesan and B. Li, *Convolutional Neural Networks in Visual Computing*. Boca Raton: CRC Press, 2017.
- [14] M. R. Alwanda, R. P. K. Ramadhan, and D. Alamsyah, "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," *J. Algoritm.*, vol. 1, no. 1, pp. 45–56, 2020, doi: 10.35957/algoritme.v1i1.434.

- [15] F. F. Maulana and N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," *J. Informatics Comput. Sci.*, vol. 1, no. 02, pp. 104–108, 2020, doi: 10.26740/jinacs.v1n02.p104-108.
- [16] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, 2018, doi: 10.1016/j.patcog.2017.10.013.
- [17] G. Armingier and D. Enache, "Statistical Models and Artificial Neural Networks," *Wuppertal*, vol. 1, pp. 1–13, 1996.
- [18] Suhartono, "Feedforward Neural Networks untuk pemodelan runtun waktu," *Univ. Gadjah Mada*, 2007.
- [19] G. P. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," *Eur. J. Oper. Res.*, vol. 160, no. 2, pp. 501–514, 2005, doi: 10.1016/j.ejor.2003.08.037.
- [20] R. Yati, T. Rohana, and A. R. Pratama, "Klasifikasi Jenis Mangga Menggunakan Algoritma Convolutional Neural Network," vol. 7, pp. 1265–1275, 2023, doi: 10.30865/mib.v7i3.6445.
- [21] D. Yin, J. Shlens, and J. Gilmer, "A Fourier Perspective on Model Robustness in Computer Vision," no. NeurIPS, 2019.
- [22] Z. Hussain, F. Gimenez, D. Yi, and D. Rubin, "Differential Data Augmentation Techniques for Medical Imaging Classification Tasks," pp. 979–984, 2018.
- [23] L. Huang, W. Pan, Y. O. U. Zhang, Y. Wu, and S. Member, "Data Augmentation for Deep Learning-Based Radio Modulation Classification," *IEEE Access*, vol. 8, pp. 1498–1506, 2020, doi: 10.1109/ACCESS.2019.2960775.
- [24] J. Sanjaya and M. Ayub, "Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop , Rotate , dan Mixup," vol. 6, pp. 311–323, 2020.