

# Klasifikasi Uang Rupiah Kertas Tidak Layak Edar Menggunakan *CNN Xception Transfer Learning* Berbasis *Website*

Muhammad Albani<sup>1</sup>, Rahmat Rizal Andhi<sup>2</sup>,

<sup>1,2</sup>Teknik Informatika, Fakultas Teknik, Universitas Riau, Pekanbaru, Riau, Indonesia  
Email: [muhammad.albani5688@student.unri.ac.id](mailto:muhammad.albani5688@student.unri.ac.id)<sup>1</sup>, [rahmat.rizal@lecturer.unri.ac.id](mailto:rahmat.rizal@lecturer.unri.ac.id)<sup>2</sup>

**Abstract** - Rupiah banknotes are the main means of payment used by the public, but the lack of public knowledge regarding their maintenance and appropriateness characteristics causes damage to the Rupiah currency. Bank Indonesia is trying to overcome this problem with the "Love Proudly Understand the Rupiah" campaign, but it will be difficult to reach the entire community with this education alone. Therefore, a system was developed "Classification of Rupiah Currency Unfit for Circulation using Website-based CNN time which has high accuracy for image classification, producing an accurate model with a short training time. Using a dataset of 14 classes of 2016 emission Rupiah currency, including 7 eligible and non-eligible denominations. Final results show 99.22% accuracy for training, 96.19 % for validation, and 93.57% for testing, in addition to developing deep learning methods, this model will be implemented on the website, which aims to make it easier and help the public to find out the suitability of the Rupiah banknotes they have.

**Keywords** - Indonesian Rupiah Banknotes, Unfit for Circulation, Xception, Transfer Learning.

**Intisari** – Uang kertas Rupiah adalah alat pembayaran utama yang dipakai oleh masyarakat, tetapi minimnya pengetahuan masyarakat mengenai perawatan dan ciri kelayakannya menyebabkan kerusakan Uang Rupiah. Bank Indonesia mencoba mengatasi masalah ini dengan kampanye "Cinta Bangsa Paham Rupiah," namun akan sulit mencapai seluruh masyarakat hanya dengan edukasi itu saja. Oleh karena itu, dikembangkan sistem "Klasifikasi Uang Rupiah Tidak Layak Edar dengan CNN Xception Transfer Learning berbasis *Website*". Metode ini memadukan *deep learning* yakni *Convolutional Neural Network* bermodel Xception dengan metode *Transfer Learning*, pemilihan Xception dipilih karena mempunyai arsitektur yang kompleks namun efisien dalam penggunaan waktunya dan memiliki akurasi yang tinggi untuk melakukan klasifikasi gambar, menghasilkan model akurat dengan waktu pelatihan singkat. Menggunakan dataset 14 kelas uang Rupiah emisi 2016, termasuk 7 pecahan yang layak dan tidak. Hasil akhir menunjukkan akurasi 99,22% untuk pelatihan, 96,19% untuk validasi, dan 93,57% untuk pengujian, disamping untuk mengembangkan metode *deep learning* menggunakan CNN Xception dengan *Transfer Learning*, dengan model hasil penelitian ini akan diimplementasikan kedalam *website*, hal ini bertujuan untuk memudahkan dan membantu masyarakat dalam mengetahui kelayakan uang kertas Rupiah yang mereka punya.

**Kata Kunci** – Uang Rupiah Kertas, Tidak Layak Edar, Xception, Transfer Learning.

## I. PENDAHULUAN

Uang merupakan salah satu alat pembayaran yang diakui oleh negara, tiap-tiap negara memiliki mata uangnya tersendiri, pada umumnya uang yang digunakan di tiap negara ini berbentuk Uang Kartal yang meliputi dua jenis yakni Uang Kertas dan Uang Logam, dengan besaran, nominal dan keunikannya tersendiri. Di Indonesia sendiri banyak ditemui uang kartal berjenis kertas berdasarkan publikasi yang dirilis oleh Bank Indonesia pada bulan September 2023 tercatat uang kartal yang beredar di Masyarakat sebanyak 4.671,0 triliun Uang Kartal. Tidak dapat dihindari pula, banyaknya jumlah beredarnya Uang Rupiah Kertas di Masyarakat, berbanding lurus dengan permasalahan yang dihadapi dalam menjaga kualitasnya.

Suatu fenomena yang umum terjadi dalam masyarakat adalah ketidakpedulian Masyarakat dalam mengenali dan menjaga uang rupiah mereka. Oleh karena itu sudah menjadi rahasia umum ketika uang kertas mengalami kerusakan, seperti sobek, terbakar, terendam air, atau dimakan oleh ngengat, tidak sedikit juga mereka yang menolak bertransaksi dengan uang rupiah seperti itu, walaupun diketahui juga banyak uang yang tidak layak edar seperti itu masih disimpan oleh Masyarakat [1]. Uang Tidak Layak Edar menurut Peraturan Bank Indonesia Nomor 6/14/PBI/2004 Uang Tidak Layak Edar adalah Uang lusuh, Uang cacat, Uang rusak, dan Uang yang telah dicabut dan ditarik dari peredaran.

Kerusakan uang rupiah ini dipandang sangat penting dikarenakan pada tahun 2022, Bank Indonesia melakukan pemusnahan uang rupiah tidak layak edar dengan nilai bilyet sebesar 6.669.096.396 Uang Rupiah Kertas [2], yang sebelumnya pada tahun 2021 mengalami penurunan diangka 5.557.312.287 Uang Kertas Rupiah [3]. Nominal bilyet sebesar itu berasal dari Penarikan Uang Tidak Layak Edar (UTLE) yang ditukarkan oleh masyarakat ke Bank Indonesia melalui Kas Keliling, Kas Titipan dan Kas Bank Indonesia. Pemusnahan uang Rupiah yang diatur ini bertujuan untuk memastikan bahwa uang yang beredar di masyarakat memenuhi standar kualitas yang ditetapkan. Hal ini merupakan bagian dari inisiatif kebijakan *clean money policy*, yang bertujuan untuk menghapus Uang Tidak Layak Edar (UTLE) dari peredaran dan menggantinya dengan uang berkualitas [4].

Sejalan dengan itu, Bank Indonesia bersama dengan pihak terkait kini sedang gencar-gencarnya melakukan kampanye Cinta, Bangga dan Paham Rupiah, dalam kampanye yang sedang dijalankan oleh Bank Indonesia ini salah satunya adalah Cinta Rupiah dengan menjaga dan merawat uang rupiah. Tidak hanya itu saja, Bank Indonesia juga mengeluarkan kampanye 5J (5 Jangan) yakni kampanye yang berfungsi sebagai edukasi kepada masyarakat dalam menjaga rupiahnya dengan cara Jangan Dilipat, Jangan Dicoret, Jangan Displar, Jangan Diremas. Kampanye ini diselenggarakan dengan maksud untuk meningkatkan literasi masyarakat untuk menjaga dan mengenali kelayakan dari uang rupiahnya.

Berangkat dari masalah yang telah diuraikan sebelumnya, diperlukanlah sebuah solusi untuk mengatasi ketidaktahuan dan kurangnya kepedulian masyarakat dalam menjaga dan mengetahui kelayakan uang rupiahnya. Salah satunya dengan menguatkan kampanye Cinta, Bangga dan Paham Rupiah tadi dengan sebuah sistem yang dapat mendeteksi kelayakan Uang Rupiah Kertas di tengah-tengah masyarakat. Dalam pendeteksian objek berupa gambar belakangan ini teknologi *deep learning* banyak dipakai, karena konsep teknologi ini memungkinkan sistem dapat menginterpretasikan data dengan meniru mekanisme kerja otak dan syaraf manusia [5]. Untuk mempermudah penerapan *deep learning* dibutuhkan *transfer learning* yang merupakan metode menggunakan jaringan saraf yang sudah dilatih sebelumnya lalu mengurangi jumlah parameter dengan cara mengambil beberapa bagian dari model yang sudah dilatih untuk digunakan dalam mengenali model baru [6].

Model yang sudah dilatih sebelumnya atau *pre-trained model* yang dipakai pada penelitian ini ialah Xception, pemilihan model ini dikarenakan Arsitektur Xception dikenal sebagai struktur yang mampu memahami objek pelatihan secara efisien dan efektif, karena *depthwise separable convolutions* membantu mengurangi jumlah parameter serta perhitungan yang terlibat dalam operasi konvolusi, sambil meningkatkan efektivitas representasi [7][8].

Selanjutnya pelatihan yang dilakukan akan menghasilkan sebuah model yang nantinya akan *diload* kedalam *website* yang dirancang menggunakan *microframework* dari *python* yaitu Flask. Flask digunakan untuk pembuatan antarmuka pengguna yang akan terkoneksi dengan sistem pengenalan citra menggunakan Keras [8].

Berdasarkan uraian sebelumnya, maka penelitian ini bertujuan untuk mengimplementasikan *deep learning* menggunakan metode *transfer learning* untuk mengklasifikasi Uang Rupiah Kertas Tidak Layak Edar berbasis *website*. Terdapat jenis kelas yang akan dijadikan data latih pada penelitian ini yang terdiri atas 7 pecahan didalamnya yakni

Rp. 1.000,-, Rp. 2.000,-, Rp.5.000,-, Rp. 10.000,-, Rp.20.000,-, Rp.50.000,-, Rp.100.000,- yang dibagi atas 2 kondisi kelayakan yakni Layak Edar (LE) dan Tidak Layak Edar (TLE). Seluruh pecahan dimasukkan agar deteksi kelayakan uang bisa diketahui diseluruh pecahan. Arsitektur Xception dipilih menjadi *pre-trained model* sebagai pengimplementasian terhadap metode *transfer learning*. Penelitian ini nantinya akan menghasilkan model yang mempelajari data masukan menggunakan metode *transfer learning* dan dapat digunakan untuk mengidentifikasi kelayakan Uang Rupiah Kertas Emisi 2016 bagi masyarakat yang masih membingungkan seperti apa uang yang layak dan tidak layak edar menurut peraturan yang berlaku.

## II. SIGNIFIKANSI STUDI

Pada signifikansi studi ini akan dijelaskan Studi Literatur terkait berupa Penelitian Terdahulu, *Dataset*, *Pre-Processing Data*, *Flask*, *Transfer Learning* dan Simulasi Pelatihan.

### A. Studi Literatur

Penelitian Daniel Martomanggolo Wonohadidjojo, dengan judul penelitian “Perbandingan *Convolutional Neural Network* pada *Transfer Learning Method* untuk Mengklasifikasikan Sel Darah Putih” menggunakan *transfer learning* dengan arsitektur CNN *VGGNet*, *ResNet-50*, *GoogleNet*, dan *AlexNet*, dengan menggunakan 70% data latih dan 30% data uji, penelitian ini berhasil mengklasifikasikan 4 kelas dari darah putih dengan Dengan keakuratan tertinggi pada arsitektur *ResNet-50* dengan hasil 94,17% serta *GoogleNet*, *VGGNet-16*, dan *AlexNet* masing-masing memiliki accuracy 93,33%, 91,67% dan 90,00%. [9]

Penelitian Alexander dkk. berjudul “Implementasi *Transfer Learning* Pada *Convolutional Neural Network* Untuk Diagnosis Covid-19 dan *Pneumonia*. Pada Citra *X-Ray*” menggunakan metode *transfer learning* dengan Xception untuk mendeteksi paru-paru terkena covid-19 dan *Pneumonia* dengan *dataset* yang terdiri dari 537 data *x-ray* manusia normal dan 565 data citra *x-ray* dada pasien penderita COVID-19, dengan pembagian 80% data pelatihan dan 20% data pengujian, penelitian ini menghasilkan sebuah sistem identifikasi berbasis *website* dengan akurasi sebesar 97,5% dengan sensitifitas sebesar 100% dan spesifisitas sebesar 95%. [10]

Penelitian Darmatasia dengan judul “Deteksi Penggunaan Masker Menggunakan Xception *Transfer Learning*” menggunakan Xception *Transfer Learning* untuk mendeteksi penggunaan masker yang benar dan tidak benar, dengan dataset berupa citra penggunaan masker yang benar, tidak menggunakan masker dan menggunakan masker tetapi tidak benar dengan pembagian 80% data pelatihan dan 20% data pengujian dengan hasil akhir akurasi sebesar 97%. [11]

Penelitian Muftah Afrizal Pangestu, Hendra Bunyamin pada berjudul “Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan *Pre-Trained CNN Model*” berhasil membuat sistem pendeteksi Ras Anjing berbasis *website* dengan menggunakan 3 *Pre-Trained CNN Model* yakni Xception, *VGG16* dan *Resnet50*. Pada penelitian ini menghasilkan keakuratan tertinggi pada arsitektur CNN berjenis Xception dan memiliki tingkat kalkulasi keakuratan 82,9%, *VGG16* memiliki 79,75% dan *ResNet50* dengan tingkat keakuratan 77,85%. [8]

Muhammad Afif Amanullah Fawwaz dkk. berjudul “Klasifikasi Ras pada Kucing menggunakan Algoritma *Convolutional Neural Network* (CNN)” dengan menggunakan *dataset* sebanyak 2393 gambar yang dibagi menjadi 2 jenis data yakni 1068 untuk data latih dan 528 untuk data validasi, dengan menggunakan arsitektur *VGG16*, *InceptionV3*, *ResNet50*, Xception dan penambahan *fine tuning* berhasil membuat sistem yang dapat mengidentifikasi ras kucing dengan akurasi tertinggi didapatkan oleh Xception sebesar 93,75%. [12]

Berdasarkan kajian pustaka di atas, maka dapat dideskripsikan beberapa perbandingan dalam bentuk Tabel I dibawah ini:

TABEL I  
PERBANDINGAN PENELITIAN TERKAIT

| No | Ref  | Masalah  | Metode   | Hasil   |
|----|------|--|--|---|
| 1  | [9]  | Tantangan para dokter dalam menganalisa jumlah sel darah putih yang ada dalam sel darah lengkap karena kategori sel darah putih mencakup sel-sel yang dapat sangat bervariasi satu sama lain   | <i>Transfer Learning, ResNet-50, GoogleNet, VGGNet-16, dan AlexNet</i> | Terbuat sebuah sistem yang dapat mengklasifikasikan Sel Darah Putih dengan keakuratan tertinggi pada arsitektur <i>ResNet-50</i> dengan hasil 94,17%  |
| 2  | [10] | Pada citra <i>x-ray</i> dada pasien COVID-19, terdapat <i>Ground Glass Opacity (GGO)</i> yakni rongga udara di paru-paru terisi oleh semacam cairan, disisi lain, citra <i>x-ray</i> dada pasien <i>Pneumonia</i> kebanyakan masih didapati batas-batas pada paru-paru dan juga sedikitnya GGO atau bahkan tidak ada GGO sama sekali pada citra <i>x-ray</i> | <i>Transfer Learning, Xception</i>                                     | Terbentuk sebuah sistem berbasis <i>website</i> yang dapat membaca Citra <i>X-Ray</i> Paru-paru, serta didapatkan akurasi sebesar 97,5%.  |
| 3  | [11] | Penerapan protokol kesehatan khususnya terkait penggunaan alat pelindung diri berupa masker adalah banyaknya masyarakat yang belum menyadari pentingnya penggunaan masker tersebut.  | <i>Transfer Learning, Xception</i>                                     | Xception telah digunakan untuk mendeteksi penggunaan masker dengan performa akurasi secara keseluruhan mencapai 97%   |
| 4  | [8]  | Banyaknya ras anjing di dunia membuat orang bertanya-tanya apakah jenis dari ras anjing yang dilihatnya baik secara langsung maupun tidak langsung (di dunia maya/ internet), agar mereka tidak kebingungan dan tak menerka jenis anjingnya.   | <i>Transfer Learning, Xception, VGG16, ResNet50</i>                    | Sistem berbasis <i>website</i> yang dapat mengidentifikasi ras anjing dengan Menggunakan <i>Pre-Trained CNN Model</i> dengan akurasi tertinggi untuk Xception 82,9%, <i>VGG16</i> 79,75 dan <i>ResNet50</i> 77,85 |
| 5  | [12] | Tidak semua pemilik kucing mengetahui ras kucingnya. Perawatan kucing seharusnya dilakukan dengan perawatan khusus untuk menjamin kesehatan kucing, disesuaikan dengan ras yang dimiliki kucing tersebut.  | <i>Transfer Learning, VGG16, InceptionV3, ResNet50, Xception</i>       | dapat mengklasifikasi Ras Kucing dengan CNN Arsitektur <i>VGG16, InceptionV3, ResNet50</i> dan Xception. Akurasi untuk tiap modelnya yaitu 60.85%, 84.94%, 71.39%, dan 93.75%.                                    |

**B. Dataset**

Penelitian ini menggunakan data berupa gambar atau citra yang dibagi kedalam 14 jenis yang terdiri dari 7 pecahan Uang Rupiah Kertas Emisi 2016 Rp. 1.000,-, Rp. 2.000,-, Rp.5.000,-, Rp. 10.000,-, Rp.20.000,-, Rp.50.000,-, Rp.100.000,- yang tidak layak edar dan layak edar. Pengumpulan data dalam penelitian ini dilakukan dengan dua cara yakni pengambilan data gambar uang layak dan tidak layak edar dengan cara memotret secara langsung. Selain mengumpulkan data dengan memotret objek secara langsung pengumpulan juga dilakukan secara tidak langsung dengan pengumpulan data gambar melalui internet. Setelah proses pengumpulan data telah diselesaikan, didapatkan sebanyak 2100 data gambar yang terdiri dari 700 data uang layak edar dan 1400 data uang tidak layak edar yang dibagi dalam 7 kelas untuk masing-masing kondisi kelayakan uang. Setelah itu data tersebut akan dibagi menjadi data latih dan data uji dengan persentase 80% data latih dan 20% data uji. Detail yang digunakan pada penelitian ini ditunjukkan seperti pada Tabel II.

TABEL III  
DETAIL PENGGUNAAN DATASET

| No | Nama Kelas | Jumlah | Data Latih | Data Uji |
|----|------------|--------|------------|----------|
|----|------------|--------|------------|----------|

|    |                             |     |     |    |
|----|-----------------------------|-----|-----|----|
| 1  | Rp.1.000 Layak Edar         | 100 | 80  | 20 |
| 2  | Rp.1.000 Tidak Layak Edar   | 200 | 160 | 40 |
| 3  | Rp.2.000 Layak Edar         | 100 | 80  | 20 |
| 4  | Rp.2.000 Tidak Layak Edar   | 200 | 160 | 40 |
| 5  | Rp.5.000 Layak Edar         | 100 | 80  | 20 |
| 6  | Rp.5.000 Tidak Layak Edar   | 200 | 160 | 40 |
| 7  | Rp.10.000 Layak Edar        | 100 | 80  | 20 |
| 8  | Rp.10.000 Tidak Layak Edar  | 200 | 160 | 40 |
| 9  | Rp.20.000 Layak Edar        | 100 | 80  | 20 |
| 10 | Rp.20.000 Tidak Layak Edar  | 200 | 160 | 40 |
| 11 | Rp.50.000 Layak Edar        | 100 | 80  | 20 |
| 12 | Rp.50.000 Tidak Layak Edar  | 200 | 160 | 40 |
| 13 | Rp.100.000 Layak Edar       | 100 | 80  | 20 |
| 14 | Rp.100.000 Tidak Layak Edar | 200 | 160 | 40 |

Adapun tampilan dari *dataset* yang digunakan pada penelitian ini disajikan di gambar 1.



Gambar 2. Cuplikan tampilan *dataset* penelitian

### C. Pre-Processing

Pengumpulan data telah diselesaikan, selanjutnya data akan masuk kepada tahap *Pre-Processing* data. *Pre-Processing* data adalah langkah awal dalam memproses data dengan tujuan menghilangkan gangguan data (noise) atau data yang tidak konsisten [13]. Tujuan Data Preprocessing adalah mengubah data awal menjadi data yang berkualitas sehingga dapat diolah pada langkah selanjutnya [14].

Terdapat dua skema *pre-processing* dalam penelitian ini yakni menggunakan augmentasi data dengan *pre-processing input* tidak menggunakan augmentasi data dan menggunakan augmentasi data dengan *data generator*. Pada skema pertama dengan tidak melakukan augmentasi pada data, dikarenakan data yang dipakai pada penelitian ini tidak terlalu bervariasi, adapun Langkah dari *pre-processing* skema kedua ini diawali dengan mengiterasi data dalam direktori yang telah dideklarasikan sebelumnya dimana direktori ini tempat untuk menyimpan dataset dengan *looping os.walk* ini akan berfungsi untuk mengambil data yang ada. Selanjutnya akan dilakukan *load* data menggunakan *load\_img* yang merupakan fungsi dari *keras / tensorflow*, lalu data akan dikunci dalam skema warna RGB dan mengubah ukuran target *height* dan *weight*nya sesuai dengan ukuran *height* dan *weight* yang telah dijabarkan sebelumnya, selanjutnya gambar dimuat dengan *array* bertipe *float32* dengan menggunakan *\_img = img\_to\_array(\_img)* lalu gambar akan diproses dengan fitur *preprocess\_input* lalu menyimpan data kedalam *array x\_train* lalu mengambil label yang ada pada direktori dengan dan di indeks dengan *label\_code* lalu dilabeli dan disimpan dalam pelabelan kela dan dilakukan *one-hot encoding* lalu disimpan kedalam *y\_train*, setelah itu data dibagi kedalam kelompok data latih dan data validasi dengan besaran masing-masing sebesar 80% dan 20% Kode lengkap untuk *pre-processing* skema kedua ini dapat dilihat pada gambar 2.

```

for dirname, _, filenames in os.walk(data_dir):
    for filename in filenames:
        _path = os.path.join(dirname, filename)
        _img = load_img(_path, color_mode="rgb", target_size=(IMG_HEIGHT, IMG_WIDTH))
        _img = img_to_array(_img)
        _img = preprocess_input(_img) # Apply Xception-specific preprocessing
        X_train.append(_img)
        _label = dirname.split('/')[-1]
        _index = label_code[_label]
        y = [0] * CATEGORIES
        y[_index] = 1
        y_train.append(y)
        i += 1
    
```

Gambar 2. Kode *Pre-processing* pada skema kedua

Untuk skema kedua masih sama dengan skema yang pertama hanya saja didahului oleh augmentasi data dengan *data generator* terlebih dahulu skema kedua ini data akan masuk kedalam sistem (*diload*) setelah itu akan di *rescale* menjadi 1./255, dengan penggunaan rotasi 20 derajat secara acak, lalu akan ditambahkan pergeseran data secara horizontal dan vertikal sebesar 20% dengan menggunakan *width\_shift\_range=0.2* dan *height\_shift\_range=0.2* lalu akan di *flip* secara horizontal dan akhirnya data akan dibagi menjadi 20% untuk validasi dan 80% untuk pelatihan, lalu selanjutnya akan disimpan dengan *looping os.walk* seperti skema pertama. ringkasan pada *augmentasi data* skema kedua ini akan disajikan dalam tabel III.

TABEL III  
DETAIL PENGGUNAAN DATASET

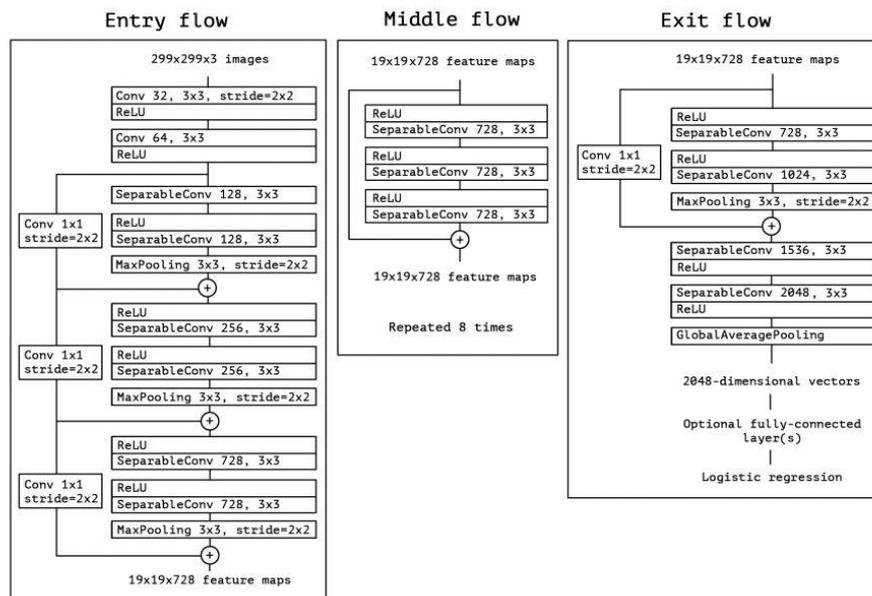
| Konfigurasi        | Nilai  |
|--------------------|--------|
| rescale            | 1./255 |
| rotation_range     | 20     |
| width_shift_range  | 0,2    |
| height_shift_range | 0,2    |
| horizontal_flip    | True   |

D. *Flask*

*Flask* adalah sebuah *web framework* yang ditulis dalam bahasa *Python* dan termasuk dalam kategori *microframework*. *Flask* berfungsi sebagai kerangka kerja untuk pengembangan aplikasi *web* [8]. Penelitian ini menggunakan *flask* dikarenakan akan dikembangkan implementasi *website* untuk pengembangan *user interface* nya, lalu akan dihubungkan dengan pemrosesan *image recognition* yang ada pada *Keras*, kedua hal ini sama-sama ditulis atau di kode dalam bahasa pemrograman *python*, dengan demikian diharapkan proses integrasi antara tampilan atau *user interface* dalam penelitian ini dapat berjalan dengan efisien.

E. *Xception*

Pada penelitian ini akan digunakan *Xception* sebagai *pre-trained* model, alasan *Xception* dipakai dalam penelitian ini karena arsitekturnya yang kompleks yang terdiri dari 3 bagian yakni *entry flow*, *middle flow* dan *exit flow*. Pada *entry flow* akan diterima citra inputan model, selanjutnya pada *middle flow* citra akan menerapkan *Separable Convolution*, yang merupakan varian dari *Depthwise Separable Convolution*, dengan fungsi aktivasi *ReLU* pada setiap layer konvolusi dan fully connected. Pada tahap akhir, menggunakan metode *Logistic Regression*. Pada bagian *Exit Flow*, model *Xception* dimodifikasi untuk menerapkan *Transfer Learning* dengan pembuatan custom *head model*. Modifikasi ini melibatkan penghapusan *head model default* *Xception* dan penggantian dengan *custom head model* [9]. Gambar 3 menunjukkan arsitektur *Xception*.



Gambar 3. Arsitektur Xception

F. Transfer Learning

Transfer learning adalah metode yang memanfaatkan arsitektur jaringan yang sudah ada, kemudian mengubah dan mengupdate parameter-parameter di dalamnya. Dengan transfer learning, arsitektur jaringan yang telah dimodifikasi digunakan untuk memecahkan tugas yang berbeda. CNN yang digunakan dalam transfer learning telah melakukan pembelajaran dengan dataset lain, sehingga tidak diperlukan proses pembelajaran dari awal. Dengan kata lain, arsitektur jaringan telah mengidentifikasi fitur-fitur seperti tekstur, bentuk, dan warna sebagai hasil dari pembelajaran sebelumnya [15].

G. Optimizer

Optimizer bertujuan untuk mencari bobot optimal, mengurangi kesalahan, dan meningkatkan akurasi. Selama proses pelatihan, parameter (bobot) dapat disesuaikan untuk mencoba meminimalkan fungsi kerugian, sehingga prediksi menjadi lebih akurat [16]. Optimizer yang digunakan dalam penelitian ini adalah optimizer SGD alasan pemilihan optimizer ini karena, optimizer SGD inipada saat yang sama, memperbarui parameter-parameternya untuk mengurangi error rate dengan menghilangkan redundansi.

H. Simulasi Pelatihan

Setelah membuat perancangan terhadap arsitektur model yang ingin digunakan, tahapan selanjutnya yakni membuat simulasi pelatihan. Pada pelatihan ini akan ada 2 skema pelatihan pada model. Model akan dilatih dengan dua macam pre-processing yakni dengan augmentasi dan tidak menggunakan augmentasi.

Dalam proses pelatihan gambar nantinya akan diubah menjadi 299x299x3, lalu akan dimasukkan model Xception sebagai base model. Selanjutnya akan ditambah lapisan pooling dengan Global Average Pooling 2D, lalu ditambahkan lapisan dropout 0,5 dan lapisan Dense dengan 256 unit neuron dan fungsi aktivasi ReLU, setelah itu ditambahkan lapisan terakhir yakni Dense 14 yang telah disesuaikan dengan jumlah kelas, bersamaan dengan itu ditambahkan juga fungsi aktivasi softmax untuk menghasilkan probabilitas kelas, lalu semua lapisan akan digabungkan menjadi model yang menghasilkan probabilitas prediksi kelas sebagai output. Kode untuk proses pelatihan dapat dilihat pada gambar 4 berikut ini.

```
# Create a new model
inputs = tf.keras.Input(shape=(IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS))
x = base_model(inputs)
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dropout(0.5)(x)
x = tf.keras.layers.Dense(256, activation='relu')(x)

# Pass the extracted features through the remaining layers
outputs = tf.keras.layers.Dense(14, activation='softmax')(x)

# Create the final model
model = tf.keras.Model(inputs=inputs, outputs=outputs)
```

Gambar 4. Kode pembuatan model baru

Setelah dibuat model baru dengan skema yang telah terlihat pada Gambar 4, selanjutnya model akan dilatih dengan *tuning hyperparameter*, pada pelatihan ini dilakukan dengan menggunakan *Epoch* yang berbeda yakni 10, 20 dan 40 dengan *batch\_size* 32, adapun skenario pelatihan yang diterapkan pada penelitian ini dapat dilihat pada Tabel IV berikut ini.

TABEL IV  
SKENARIO PELATIHAN

| Parameter             | Skenario 1     |     |     | Skenario 2 |     |     |
|-----------------------|----------------|-----|-----|------------|-----|-----|
| <i>Pre-Processing</i> | Non-Augmentasi |     |     | Augmentasi |     |     |
| <i>Epoch</i>          | 10             | 20  | 40  | 10         | 20  | 40  |
| <i>Optimizer</i>      | SGD            | SGD | SGD | SGD        | SGD | SGD |
| <i>Batch_size</i>     | 32             | 32  | 32  | 32         | 32  | 32  |

### III. HASIL DAN PEMBAHASAN

#### A. Pelatihan Model

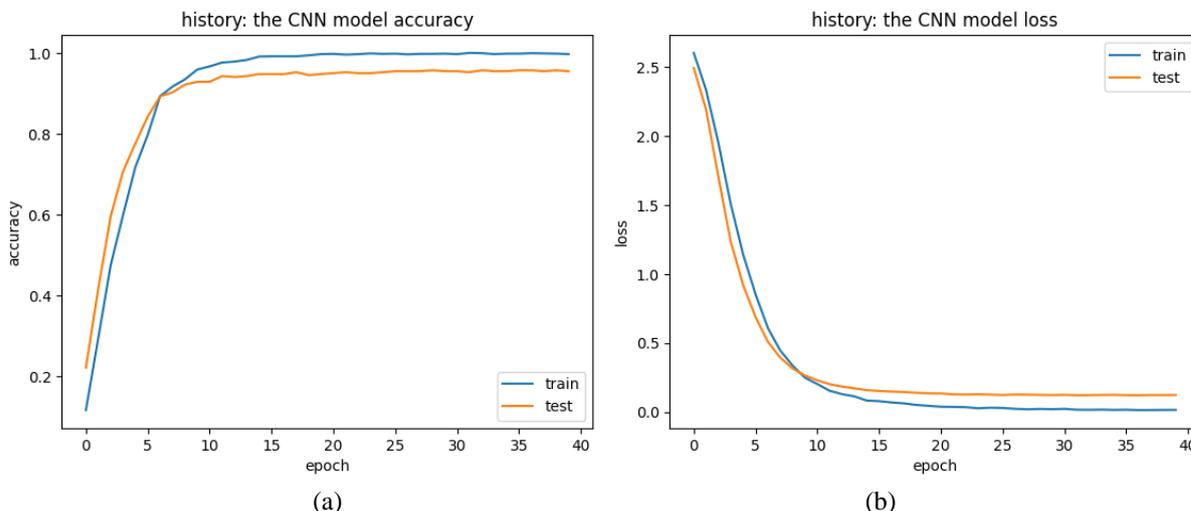
Pada penelitian ini terdapat dua skenario yang akan diterapkan pada pelatihan, walaupun hanya berbeda pada saat tahap *pre-processing* data saja, hal itu diterapkan dalam penelitian ini agar mendapatkan hasil berupa model yang terbaik dalam pelatihan ini. Selanjutnya pada tahap pelatihan dimulai dengan skenario pertama yakni dilakukan *pre-processing input* dari Xception pada saat *pre-processing*, lalu dilatih dengan *Epoch* yang berbeda yaitu 10, 20, dan 40. Hasil yang diperoleh pada skenario pelatihan pertama ini dapat dilihat pada Tabel V.

TABEL V  
HASIL SKENARIO PELATIHAN I

| Percobaan | <i>Epoch</i> | Akurasi |            | Selisih | Loss   |            |
|-----------|--------------|---------|------------|---------|--------|------------|
|           |              | Train   | Validation |         | Train  | Validation |
| 1         | 10           | 0,9773  | 0,8834     | 0,0939  | 0,2242 | 0,4433     |
| 2         | 20           | 0,9958  | 0,9238     | 0,072   | 0,0447 | 0,9238     |
| 3         | 40           | 0,9970  | 0,9548     | 0,0422  | 0,0148 | 0,1231     |

Setelah dilakukan pelatihan dengan menggunakan skenario pertama, hasil terbaik didapatkan oleh percobaan ketiga, pada percobaan tersebut model dilatih menggunakan 40 *Epoch* dengan tanpa adanya proses augmentasi sebelumnya pada tahap *pre-processing* data,

alasan dipilihnya percobaan ketiga sebagai percobaan dengan hasil terbaik pada skenario pertama ini adalah dikarenakan selisih antara nilai akurasi pada *train* dan *validation* cukup rendah yakni di angka 0,0422, serta diiringi dengan nilai *loss* yang sudah menurun dari *Epoch* sebelumnya. Grafik hasil percobaan kedua dapat dilihat pada Gambar 5.



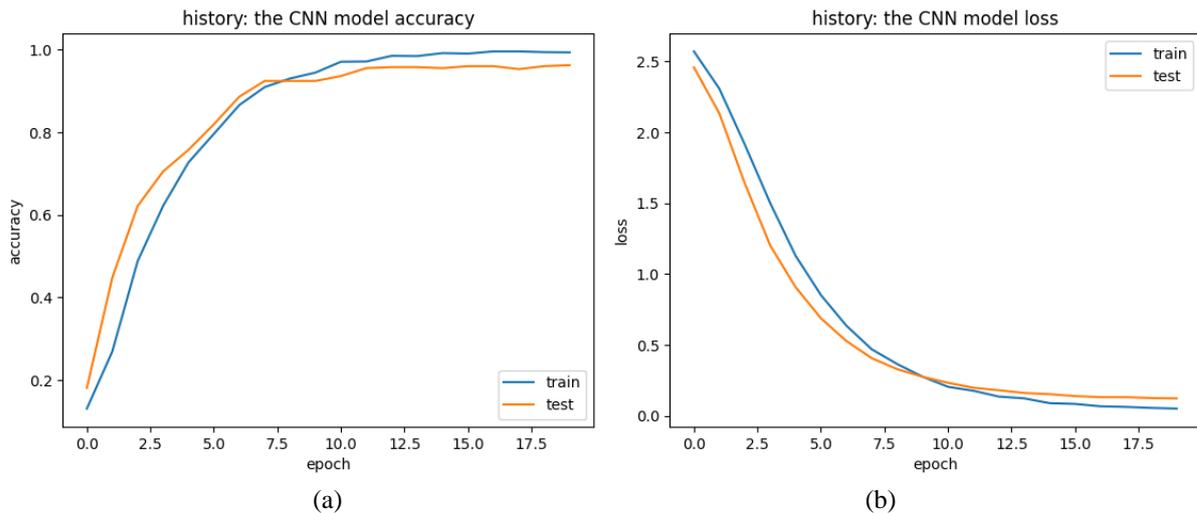
Gambar 5. (a) Grafik Accuracy (b) Grafik Loss pada percobaan 3 dari skenario pertama

Pada percobaan ketiga ini, sesuai dengan gambar grafik diatas, model dilatih dengan menambahkan augmentasi data dalam proses pre-processing dan model mendapatkan akurasi diatas 80% pada *Epoch* ke-7 untuk *training* dan *Epoch* ke-6 untuk *validation*. Hasil akurasi akhir pada proses *training* percobaan ketiga dengan hanya menggunakan *pre-processing input* Xception menghasilkan nilai akurasi sebesar 0,997 untuk *training* dan 0,9548 untuk *validation*, terdapat selisih sebesar 0,0422, dari percobaan tersebut terdapat peningkatan akurasi namun selisihnya masih lumayan besar. Selanjutnya, pada Tabel VI disajikan hasil pengujian memakai skenario kedua.

TABEL VI  
HASIL SKENARIO PELATIHAN 2

| Percobaan | Epoch | Akurasi       |               | Selisih      | Loss          |               |
|-----------|-------|---------------|---------------|--------------|---------------|---------------|
|           |       | Train         | Validation    |              | Train         | Validation    |
| 1         | 10    | 0,9507        | 0,8801        | 0,0706       | 0,1140        | 0,3140        |
| 2         | 20    | <b>0,9929</b> | <b>0,9619</b> | <b>0,031</b> | <b>0,0504</b> | <b>0,1226</b> |
| 3         | 40    | 0,9976        | 0,9500        | 0,0476       | 0,0144        | 0,1611        |

Setelah dilakukan pelatihan dengan menggunakan skenario kedua, hasil terbaik didapatkan oleh percobaan kedua, pada percobaan tersebut model dilatih menggunakan 20 *Epoch* tanpa adanya proses augmentasi sebelumnya, alasan dipilihnya percobaan kedua sebagai percobaan dengan hasil terbaik pada skenario pertama ini adalah dikarenakan selisih antara nilai akurasi pada train dan validasi cukup rendah yakni di angka 0,031. Grafik hasil percobaan kedua dapat dilihat pada Gambar 6.



Gambar 6. (a) Grafik Accuracy (b) Grafik Loss pada percobaan 2 dari skenario kedua

Pada percobaan kedua ini dapat dilihat pada gambar 6 proses pelatihan berjalan dengan baik dibuktikan dengan kelandaian Grafik hasil pelatihan dimana pada pelatihan kedua ini. didapatkan akurasi diatas 80% pada *Epoch* ke-7 pada training dan *Epoch* ke-6 untuk validation. Hasil akhir untuk akurasi yang didapatkan pada pelatihan percobaan kedua ini adalah 0.9929 untuk *training* dan 0,9619 untuk *validation*, dengan selisih sebesar 0,031, terdapat peningkatan akurasi dari percobaan pertama yang hanya mendapatkan nilai akurasi akhir sebesar 0.9507 untuk *training* dan 0.8801 untuk *validatio*, dengan demikian hasil dari percobaan ini pada skenario kedua lebih baik dari percobaan ketiga dari skenario pertama dan dapat dilanjutkan kepada tahap pengujian.

### B. Hasil Pengujian

Setelah didapatkan model yang terbaik hasil pelatihan dari dua skenario sebelumnya, setelah itu model masuk kedalam tahap pelatihan pada tahapan ini terdapat 420 data gambar yang terdiri dari 20 gambar untuk setiap kelas Uang Layak Edar dan 40 untuk kelas Uang Tidak Layak Edar. Pengujian dilakukan dengan mengupload gambar kemudian model di *load* kedalam sistem dan digunakan fungsi prediksi pada tensorflow untuk memprediksi gambar tersebut telah sesuai dengan kelas dalam model tersebut. Adapun hasil pengujian dapat dilihat pada confusion matrix yang disajikan pada gambar 7.

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0  | 19 | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |
| 1  | 1  | 38 | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |
| 2  | 0  | 2  | 18 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |
| 3  | 0  | 0  | 2  | 36 | 0  | 0  | 0  | 0  | 1  | 0  | 1  | 0  | 0  |    |
| 4  | 0  | 0  | 0  | 0  | 18 | 1  | 0  | 0  | 0  | 0  | 1  | 0  | 0  |    |
| 5  | 0  | 0  | 0  | 0  | 1  | 38 | 0  | 0  | 0  | 0  | 0  | 0  | 1  |    |
| 6  | 0  | 0  | 0  | 0  | 0  | 2  | 17 | 0  | 0  | 0  | 1  | 0  | 0  |    |
| 7  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 38 | 1  | 0  | 0  | 0  | 0  |    |
| 8  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 18 | 1  | 0  | 1  | 1  |    |
| 9  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 1  | 37 | 1  | 0  | 0  |    |
| 10 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 20 | 0  | 0  |    |
| 11 | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 39 | 0  |    |
| 12 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 18 | 2  |    |
| 13 | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 38 |    |
|    | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 |

Keterangan:

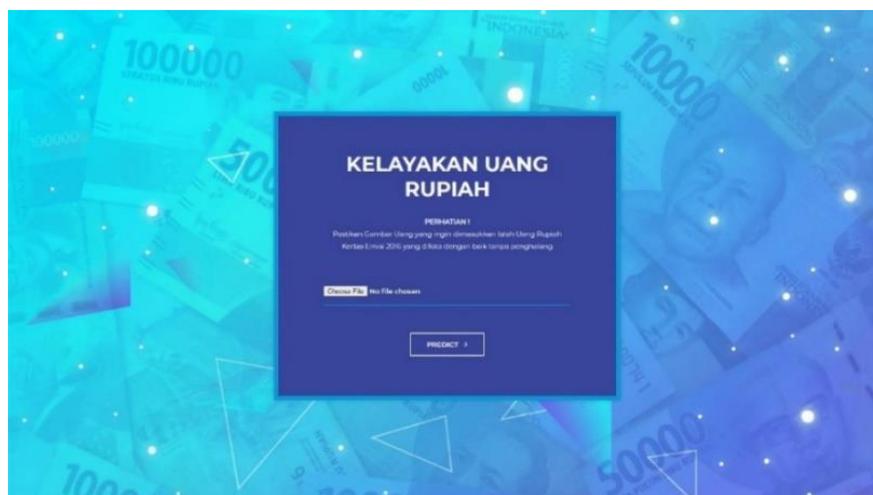
- 0 : Rp.1.000 Layak Edar
- 1 : Rp.1.000 Tidak Layak Edar
- 2 : Rp.2.000 Layak Edar
- 3 : Rp.2.000 Tidak Layak Edar
- 4 : Rp.5.000 Layak Edar
- 5 : Rp.5.000 Tidak Layak Edar
- 6 : Rp.10.000 Layak Edar
- 7 : Rp.10.000 Tidak Layak Edar
- 8 : Rp.20.000 Layak Edar
- 9 : Rp.20.000 Tidak Layak Edar
- 10 : Rp.50.000 Layak Edar
- 11 : Rp.50.000 Tidak Layak Edar
- 12 : Rp.100.000 Tidak Layak Edar
- 13 : Rp.100.000 Layak Edar

Gambar 7. Confusion Matrix

Terlihat pada Gambar 7, ialah *confusion matrix* dari tahap pengujian model pada penelitian ini, Adapun hasil akhir akurasi dari pengujian yang telah dilakukan terhadap model ialah dengan jumlah prediksi data benar sebanyak 394 dari 420 data uji yakni 93,75.

C. Implementasi Website

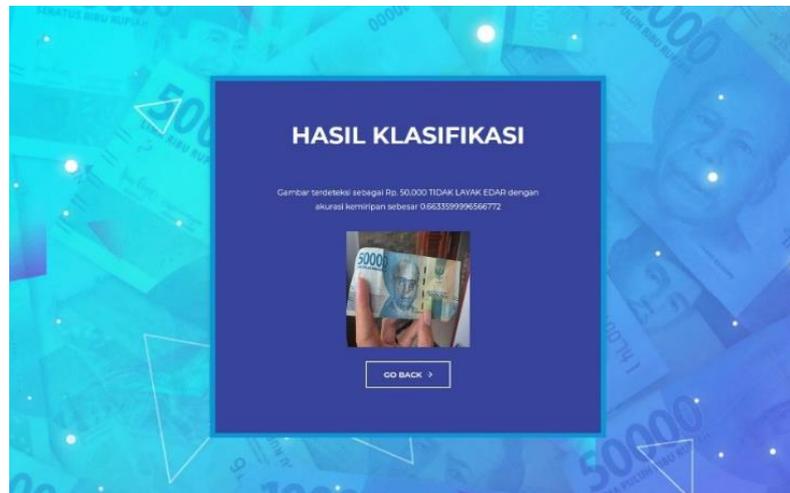
Implementasi website penelitian ini menggunakan Flask yang merupakan fitur *microframework* dari python yang memungkinkan pengguna untuk membuat website. Sesuai dengan kerangka implementasi sistem, yang telah dibahas pada bab sebelumnya, implementasi akan diawali dengan tampilan awal atau beranda, pada tampilan ini, pengguna dapat menemui fitur unggah atau upload yang berguna untuk memasukkan citra rupiah yang dimilikinya. Tampilan awal atau beranda dapat dilihat pada Gambar 8.



Gambar 8. Halaman Awal Website

Setelah pengguna menekan tombol *Choose File* pengguna akan diarahkan kepada Kotak Dialog yang berguna untuk melakukan pemilihan gambar atau citra dari Uang Rupiah yang

ingin mereka klasifikasi dengan website ini. apabila gambar telah berhasil dipilih dan diupload kedalam website, akan tampak nama *file* yang sebelumnya dipilih tadi pada bagian *Choose File* munculnya nama *file* pada bilah tersebut menandakan *file* berhasil diunggah dan bisa untuk diprediksi. Untuk tahap selanjutnya pengguna dapat menekan tombol *PREDICT* > untuk memprediksi atau mengklasifikasikan citra yang diunggah sebelumnya. Gambar 9 menampilkan tampilan *website* setelah data telah di prediksi.



Gambar 9. Halaman Awal Website

Setelah pengguna menekan tombol Predict > yang ada pada halaman sebelumnya, sistem akan melakukan prediksi dan pengklasifikasian terhadap data yang telah dimasukkan, tidak membutuhkan waktu lama hasil prediksi terhadap citra yang telah dimasukkan akan muncul pada tampilan hasil, diiringi dengan seberapa besar tingkat kemiripan terhadap model yang telah diujikan.

#### IV. KESIMPULAN

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa metode *transfer learning* dengan *pre-trained* model Xception dinilai baik untuk mengklasifikasikan Uang Kertas Rupiah Tidak Layak Edar dan Layak Edar tahun Emisi 2016. Hal ini ditunjukkan oleh akurasi pengujian model sebesar 93,75%. Hasil penelitian ini dapat digunakan secara langsung untuk memudahkan masyarakat mengetahui kelayakan uang kertasnya, dikarenakan sistem ini dikembangkan menjadi aplikasi *website* yang dapat diakses oleh masyarakat secara luas.

Namun, kekurangan dari penelitian ini adalah perlunya untuk meningkatkan akurasi *validation* dan *testing* dari proses pelatihan maupun pengujian, dengan cara memperluas variasi data yang digunakan dan *pre-processing* dengan augmentasi data yang baik. Saran untuk peneliti selanjutnya ialah dapat menggunakan *pre-trained* model lainnya seperti MobileNet, Inception, ResNet agar mendapatkan metode terbaik dalam mengklasifikasikan Uang Rupiah Tidak Layak Edar.

#### REFERENSI

- [1] Fitriyah, F., & Ma'arif, M. M. Analisis Jasa Penukaran Uang Rusak Keliling Perspektif Hukum Ekonomi Islam (Damaged Money Exchange Services on the Perspective of Islamic Law). *Al-Hikmah*, Vol. 7, No. 2: pp. 163–165, 2019.
- [2] Peraturan Bank Indonesia. Jumlah Dan Nilai Nominal Uang Rupiah Yang Dimusnahkan Tahun 2022. Nomor 1 Tahun 2023. Bank Indonesia, 2023.

- [3] Peraturan Bank Indonesia. Jumlah Dan Nilai Nominal Uang Rupiah Yang Dimusnahkan Tahun 2022. 24/1/PBI/2022. Bank Indonesia, 2022.
- [4] Indonesia Bank, (2021), *Laporan Tahunan Bank Indonesia Tahun 2021*, Bank Indonesia, Jakarta.
- [5] R. Patel and A. Chaware, "Transfer Learning with Fine-Tuned MobileNetV2 for Diabetic Retinopathy," *2020 International Conference for Emerging Technology (INCET)*, Belgaum, India, 2020, pp. 1-4, 2020.
- [6] Abas, Mohamad Aqib Haqmi. VGG16 for plant image classification with transfer learning and data augmentation. *International Journal of Engineering and Technology (UAE)*, Vol. 7, No. 4, pp. 90-94.
- [7] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 1800-1807.
- [8] Pangestu, M. A., & Bunyamin, H. . Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model. *Jurnal Teknik Informatika Dan Sistem Informasi*. Vol. 4, No. 2, pp. 337-334, 2018.
- [9] Wonohadidjojo, D. M. Perbandingan Convolutional Neural Network pada Transfer Learning Method untuk Mengklasifikasikan Sel Darah Putih. *Ultimatics : Jurnal Teknik Informatika*, Vol. 13, No. 1, pp. 51-57, 2021.
- [10] Ramadhan, M., Iskandar Mulyana, D., Betty Yel, M., & Tinggi Ilmu Komputer Cipta Karya Informatika Jl Raden, S. Optimasi Algoritma CNN Menggunakan Metode Transfer Learning Untuk Klasifikasi Citra X-Ray Paru-Paru Pneumonia Dan Non-Pneumonia. *Jurnal Teknik Informatika Kaputama (JTJK)*, Vol. 6, No. 2, pp. 670-679, 2022.
- [11] Darmasita. Deteksi Penggunaan Masker Menggunakan Xception Transfer Learning. *JURNAL INSTEK (Informatika Sains Dan Teknologi)*, Vol. 5, No. 2, pp. 279-288, 2020.
- [12] Alghifari, F., & Juardi, D. Penerapan Data Mining Pada Penjualan Makanan Dan Minuman Menggunakan Metode Algoritma Naïve Bayes. *Jurnal Ilmiah Informatika*, Vol. 9, No. 2, pp. 75-81, 2021.
- [13] Shevira, S., Suarjaya, I. M. A. D., & Buana, P. W. Pengaruh Kombinasi dan Urutan Pre-Processing pada Tweets Bahasa Indonesia. *JITTER : Jurnal Ilmiah Teknologi Dan Komputer*, Vol. 3, No. 2, pp. 1074-1081, 2022.
- [14] Eben Haezer, E. H. Y. K., & Setiyawati, N. Pembangunan Aplikasi Virtual Inventory System (VIS) Berbasis Web Menggunakan Flask Framework. *Jurnal Mnemonic*, Vol. 5, No. 2, pp. 128-135, 2022.
- [15] Ramadhan, M., Iskandar Mulyana, D., Betty Yel, M. Optimasi Algoritma CNN Menggunakan Metode Transfer Learning Untuk Klasifikasi Citra X-Ray Paru-Paru Pneumonia dan Non-Pneumonia. *Jurnal Teknik Informatika Kaputama (JTJK)*, Vol. 6, No. 2, pp. 670-679, 2022.
- [16] Micheal, Micheal, and Ery Hartati. "Klasifikasi Spesies Kupu Kupu Menggunakan Metode Convolutional Neural Network." *MDP Student Conference*. Vol. 1, No. 1, pp. 569-277, 2022.

#### UCAPAN TERIMA KASIH

Terima kasih yang tulus kepada semua pihak yang memberi dukungan dengan penuh serta kerjasama yang sangat berperan penting dalam kesuksesan pelaksanaan penelitian ini.