

# **Aplikasi Deteksi Katarak (Eye Care) Menggunakan Model Convolutional Neural Network Berbasis Mobile**

**Rico Apriliansyah<sup>\*1</sup>, Khoiriya Latifah<sup>2</sup>**

<sup>1,2</sup> Program Studi Informatika, Universitas PGRI Semarang, Kota Semarang

\*Email: [ricoapriliansyah@gmail.com](mailto:ricoapriliansyah@gmail.com)

## **Abstract**

Visual impairment is still a major problem in Indonesia. Most of the visual impairment is caused by cataracts. Based on the results of the Rapid Assessment of Avoidable Blindness (RAAB) survey by the Indonesian Ophthalmologist Association and Balitbangkes in 15 provinces, the blindness rate in Indonesia reached 3 per cent. Of this figure, cataract is the highest cause, which is around 81 per cent. Cataracts generally occur in the elderly aged over 40-50 years and above, but actually everyone can get cataracts caused by things in the environment such as being bumped while working or playing, accidents, excessive sun exposure, smoking and others. This cataract disorder is also often unconsciously attacking the eye as it grows older slowly. If not treated properly and quickly, it can experience blindness. Therefore we plan to create an application that can detect cataract disease in the eye. so that this application can reduce the risk of blindness by helping to detect eye problems at an early stage to provide proper treatment before the problem becomes severe. The method used is deep learning with a CNN model. The research results show that the accuracy of the model for detecting cataracts is 93%, precision 95%, recall 90% and f1 score 92%.

Keywords: CNN, Cataract Disease, Deep Learning

## **Abstrak**

Gangguan penglihatan masih menjadi masalah utama di Indonesia. Sebagian besar gangguan penglihatan disebabkan oleh katarak. Berdasarkan hasil survei Rapid Assessment of Preventionable Blindness (RAAB) yang dilakukan Persatuan Dokter Spesialis Mata Indonesia dan Balitbangkes di 15 provinsi, angka kebutaan di Indonesia mencapai 3 persen. Dari angka tersebut, katarak menjadi penyebab tertinggi, yakni sekitar 81 persen. Katarak umumnya terjadi pada lansia berusia di atas 40-50 tahun ke atas, namun sebenarnya setiap orang bisa terkena katarak disebabkan oleh hal-hal di lingkungan seperti terbentur saat bekerja atau bermain, kecelakaan, paparan sinar matahari berlebihan, merokok dan lain-lain. Kelainan katarak ini juga seringkali tanpa disadari menyerang mata seiring bertambahnya usia secara perlahan. Jika tidak ditangani dengan baik dan cepat, dapat mengalami kebutaan. Oleh karena itu kami berencana untuk membuat sebuah aplikasi yang dapat mendeteksi penyakit katarak pada mata. sehingga aplikasi ini dapat mengurangi resiko kebutaan dengan membantu mendeteksi gangguan mata secara dini untuk memberikan pengobatan yang tepat sebelum masalah menjadi parah. Metode yang digunakan adalah deep learning dengan model CNN. Hasil penelitian menunjukkan bahwa akurasi model untuk mendeteksi penyakit katarak yaitu 93%, presisi 95%, recall 90% dan f1 score 92%.

Kata Kunci: CNN, Deep Learning, Penyakit Katarak

## **1. Pendahuluan**

Masalah kesehatan mata, khususnya katarak, merupakan masalah kesehatan yang signifikan di seluruh dunia. Katarak adalah suatu kondisi dimana lensa mata menjadi keruh sehingga mengakibatkan penglihatan menjadi kabur bahkan kehilangan penglihatan jika tidak segera ditangani. Menurut Organisasi Kesehatan Dunia (WHO), katarak adalah penyebab utama kebutaan di seluruh dunia, yang dapat dicegah dan diobati jika terdeteksi sejak dini[1].

Kekhawatiran bagi individu dengan katarak. Penyakit mata katarak merupakan penyebab utama kebutaan di Indonesia. Berdasarkan hasil survei Rapid Assessment of Preventionable Blindness (RAAB) pada tahun 2014 hingga 2016 diketahui angka kebutaan mencapai 3%, dengan katarak menjadi penyebab kebutaan tertinggi (81%).

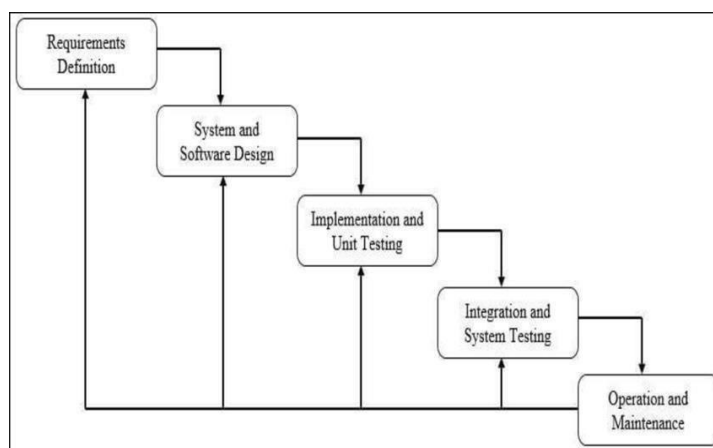
Meskipun katarak dapat diobati, deteksi dini memainkan peran penting dalam menghindari komplikasi serius. Sayangnya, banyak orang yang tidak menyadari gejala katarak pada tahap awal, dan upaya pemantauan kesehatan mata seringkali dibatasi oleh akses terhadap fasilitas medis atau kurangnya pengetahuan[2].

*Deep learning* merupakan salah satu cabang dari pembelajaran mesin yang menggunakan algoritma pemodelan abstrak tingkat tinggi untuk data. Algoritma-algoritma ini melibatkan serangkaian fungsi transformasi nonlinier yang disusun dalam berbagai level dan kedalaman.[3] Salah satu contoh implementasinya adalah Convolutional Neural Network (CNN), yang sering digunakan untuk analisis citra visual. CNN terdiri dari berbagai lapisan seperti lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Dalam penelitian ini, CNN digunakan untuk mengklasifikasikan citra mata guna mendeteksi penyakit katarak[4].

"Eye Care: Aplikasi Deteksi Mata Katarak" adalah aplikasi inovatif yang dirancang untuk membantu masyarakat mendeteksi potensi risiko katarak sejak dini. Aplikasi ini menggabungkan teknologi kecerdasan buatan dan analisis gambar menggunakan *deep learning image analysis* dengan model Convolutional Neural Network (CNN) untuk menilai kemungkinan katarak berdasarkan foto mata pengguna.

## 2. Metode

Sistem aplikasi dalam penelitian ini dikembangkan menggunakan Metode Waterfall (Air Terjun), suatu pendekatan pengembangan perangkat lunak yang sistematis dan berurutan. Metode Waterfall memerlukan penyelesaian serangkaian fase secara berurutan, di mana tiap fase tergantung pada penyelesaian fase sebelumnya.



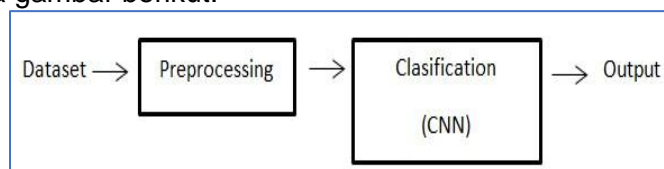
Gambar 1. Metode Waterfall

1. Analisis Kebutuhan (Requirement): Memahami dan mencatat dengan detail apa yang pengguna butuhkan dari sistem, sehingga kita tahu persis apa yang harus dikembangkan.
2. Perancangan Desain (Design System): Merancang bagaimana sistem akan terstruktur dan beroperasi berdasarkan kebutuhan yang telah diketahui, seperti merencanakan bagaimana semua bagian akan saling berhubungan.
3. Implementasi (Implementation): Membuat sistem dengan menuliskan kode-kode program berdasarkan rancangan yang telah dibuat, seperti membangun sebuah bangunan berdasarkan blueprint-nya.
4. Pengujian (Testing): Menguji sistem secara menyeluruh untuk memastikan bahwa semuanya berfungsi sesuai dengan yang diinginkan, seperti memeriksa semua fitur dan memastikan tidak ada kesalahan.

5. Pemeliharaan (Maintenance): Merawat sistem dengan memperbaiki masalah yang mungkin muncul, meningkatkan keamanan, dan melakukan penyesuaian kecil setelah sistem digunakan oleh pengguna, mirip dengan merawat mobil agar tetap berjalan lancar.

Metode Waterfall terkadang dianggap terlalu kaku dan kurang fleksibel dikarenakan persyaratan akan diubah atau diperbarui seiring berjalannya waktu.[5] Walaupun demikian kami tetap menggunakan metode waterfall karena menurut kami cocok dengan proyeknya.

Adapun metode yang digunakan untuk membuat model yaitu menggunakan CNN untuk dapat mengklasifikasi penyakit mata. Metode CNN yang dibuat dalam penelitian ini dapat di gambarkan seperti pada gambar berikut:



Gambar 2. Metode CNN [4]

### 3. Hasil dan Pembahasan

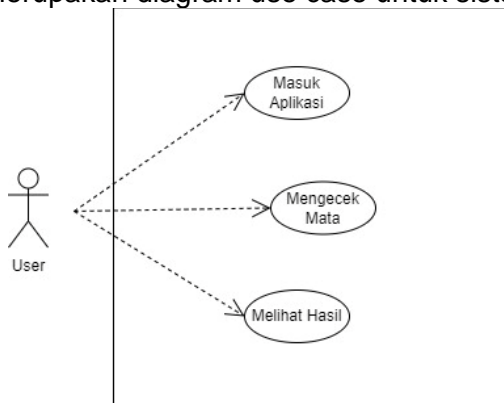
#### 1) Analisis Kebutuhan

Analisis kebutuhan yang dilakukan meliputi identifikasi kebutuhan pengguna, terutama pasien yang sulit mengakses fasilitas medis. Aplikasi harus mudah digunakan oleh semua kelompok umur, terutama orang tua. Fungsionalitas utama adalah analisis gambar mata menggunakan kecerdasan buatan untuk deteksi katarak dengan akurasi tinggi dan juga mencari dataset supaya memudahkan dalam pengerjaan model. Kebutuhan nonfungsional mencakup keamanan data, keandalan, dan kinerja cepat.

#### 2) Desain Sistem

##### a) Use Case Diagram

Diagram Use Case merupakan representasi fungsional dalam sebuah sistem yang menjelaskan interaksi antara aktor dan sistem mengenai alur kerja proses yang telah dikembangkan. Berikut merupakan diagram use case untuk sistem aplikasi yang dibuat:

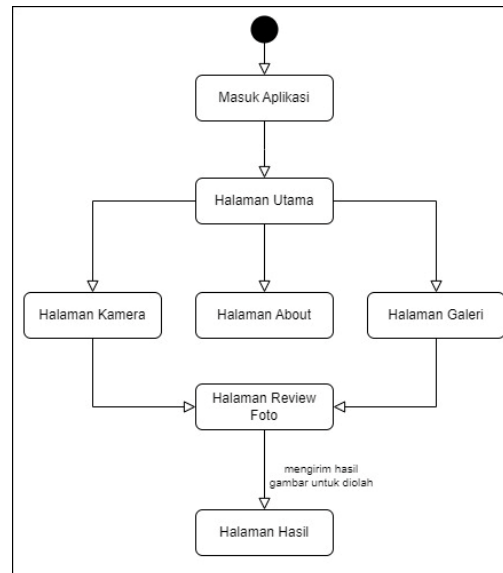


Gambar 2. Diagram Use Case

Pada gambar diatas merupakan Use Case diagram untuk aplikasi yang menunjukkan bahwa user bisa masuk untuk mengakses aplikasi setelah di download tanpa login dan bisa langsung mengecek mata pengguna sehingga bisa lebih mudah digunakan semua usia.

##### b) Activity Diagram

Diagram Activity merupakan representasi visual dari urutan kerja yang mencakup aktivitas, tindakan, serta kemungkinan pilihan atau pengulangan. Dalam UML, diagram aktivitas digunakan untuk menggambarkan aliran kerja baik dalam konteks komputer maupun dalam konteks aktivitas organisasional. Berikut merupakan diagram activity untuk sistem yang dibuat:



Gambar 3. Diagram Activity

User masuk aplikasi kemudian di halaman utama user bisa memilih mau mengecek mata lewat fitur kamera atau lewat galeri, setelah itu user ditampilkan hasil dari foto yang dipilih untuk dicek terlebih dahulu sudah jelas atau belum, jika sudah jelas bisa langsung dikirim. Sistem mengirimkan hasil gambar ke backend melewati API untuk diolah dengan memprediksi menggunakan model yang sudah kita buat lalu jika proses pengolahan sudah selesai sistem menampilkan hasil dari olahan tersebut.

### 3) Implementasi

Implementasi pembuatan pemodelan untuk deteksi katarak menggunakan deep learning dengan model CNN sebagai berikut:

#### a) Dataset

Dataset dalam penelitian ini merupakan dataset sekunder yang diperoleh dari Kaggle (<https://www.kaggle.com/datasets/nandanp6/cataract-image-dataset>). Dataset tersebut terdiri dari dua path yaitu test dan train yang didalamnya terdiri dari dua kelas yakni katarak dan normal. Dimana dalam path test, data katarak terdiri dari 61 file gambar dan data normal terdiri dari 60 file gambar; sedangkan dalam path train, data katarak terdiri dari 245 file gambar dan data normal terdiri dari 246 file gambar.[6]

#### b) Preprocessing

Preprocessing adalah tahap pengolahan data atau citra sebelum data atau citra tersebut dimasukkan ke dalam algoritma Convolutional Neural Network (CNN). Proses ini dapat mencakup beberapa metode seperti mengubah citra RGB menjadi greyscale, melakukan binerisasi gambar, dan memotong gambar.[7] Tujuan dari preprocessing antara lain untuk mengurangi atau menghilangkan noise, meningkatkan kejelasan fitur pada data atau citra, menyesuaikan ukuran data atau citra, serta mengonversi data atau citra asli agar sesuai dengan kebutuhan yang ditentukan.[3]

```

def predict_model():
    # Preprocessing
    training_dir = os.path.join("data", "train")
    image_size = (55, 94, 3)
    target_size = (55, 94)

    # Data Augmentation
    train_datagen = ImageDataGenerator(
        rescale=1./255,
        rotation_range=10,
        horizontal_flip=True,
        zoom_range=0.5
    )
    train_generator = train_datagen.flow_from_directory(
        training_dir,
        target_size=target_size,
        class_mode='binary'
    )
  
```

Gambar 4. Preprocessing

Gambar diatas merupakan proses preprocessing yang dilakukan untuk menyiapkan data agar data yang akan ditrain adalah data yang lengkap/bersih. Data augmentation (train\_datagen) bagian ini membuat data tambahan dengan mengubah dataset yang ada berdasarkan parameter berikut train generator:

- 1) rescale (menormalisasi dataset dengan nilai dari 0 sampai 1),
- 2) rotation\_range : merotasi gambar sebanyak 10 derajat ke kanan/kiri,
- 3) horizontal flip = flip gambar secara horizontal aktif,
- 4) zoom\_range = gambar di-zoom sebanyak 0.5 tujuannya adalah menambah data (jika dataset dirasa kurang banyak/kurang bervariasi).

Setelah parameter disimpan dalam variable (train\_datagen), parameter kemudian dipakai oleh dataset (training\_dir) dengan ukuran sesuai dengan target size dan klasifikasi binari (klasifikasi 2 kelas).

#### c) CNN/Pembuatan Model

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.optimizers.experimental import RMSprop

model = Sequential([
    # Convolutional Neural Network (CNN) for feature extraction
    Conv2D(32, (3, 3), activation='relu', input_shape=image_size),
    MaxPooling2D(2, 2),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Flatten(),
    # Dense Neural Network (DNN) for classification
    Dense(512, activation='relu'),
    Dense(1, activation='sigmoid')
])
```

Gambar 5. CNN Model

Pada gambar diatas merupakan code pembuatan model. Bagian ini adalah arsitektur dari model, setiap data akan masuk melalui model Sequential ini untuk di train dengan model CNN dan MaxPolling2D. Setelahnya, data akan di Flatten agar dapat diproses pada Dense Layer dengan activation sigmoid agar kelasnya dapat menjadi binary classification.

```
model.compile(
    loss = 'binary_crossentropy',
    optimizer=RMSprop(learning_rate=0.001),
    metrics=['accuracy']
)

from keras.callbacks import EarlyStopping

early_stop = EarlyStopping(monitor='accuracy', patience=5)

history = model.fit(
    train_generator,
    epochs=30,
    callbacks=[early_stop]
)
```

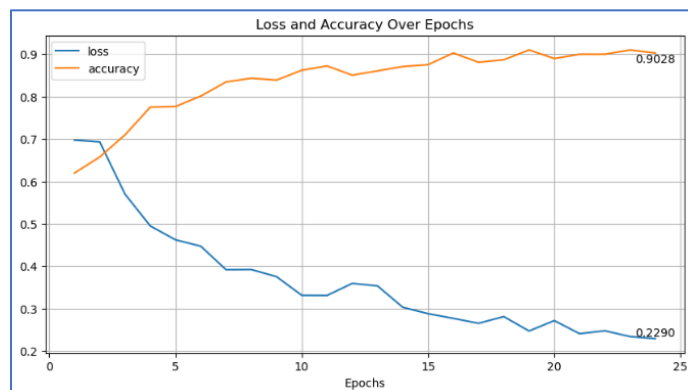
Gambar 6. Compile dan Callback

```
Epoch 1/30
1/22 [.....] - ETA: 36s - loss: 0.6914 - accuracy: 0.5938
WARNING:tensorflow:From /usr/local/lib/python3.10/site-packages/tensorflow/python/ops/image_ops_impl.py:1295: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
warnings.warn(
Epoch 2/30
1/22 [.....] - 7s 290ms/step - loss: 0.6974 - accuracy: 0.6197
Epoch 3/30
1/22 [.....] - 7s 310ms/step - loss: 0.6930 - accuracy: 0.6575
Epoch 4/30
1/22 [.....] - 7s 320ms/step - loss: 0.4947 - accuracy: 0.7750
Epoch 5/30
1/22 [.....] - 7s 310ms/step - loss: 0.4621 - accuracy: 0.7765
Epoch 6/30
1/22 [.....] - 7s 300ms/step - loss: 0.4470 - accuracy: 0.8012
Epoch 7/30
1/22 [.....] - 8s 341ms/step - loss: 0.3916 - accuracy: 0.8345
Epoch 8/30
1/22 [.....] - 8s 325ms/step - loss: 0.3920 - accuracy: 0.8433
Epoch 9/30
1/22 [.....] - 7s 338ms/step - loss: 0.3751 - accuracy: 0.8389
Epoch 10/30
1/22 [.....] - 8s 370ms/step - loss: 0.3312 - accuracy: 0.8621
Epoch 11/30
1/22 [.....] - 7s 312ms/step - loss: 0.3308 - accuracy: 0.8723
Epoch 12/30
1/22 [.....] - 7s 313ms/step - loss: 0.3593 - accuracy: 0.8595
Epoch 13/30
1/22 [.....] - 7s 318ms/step - loss: 0.3533 - accuracy: 0.8607
...
Epoch 23/30
1/22 [.....] - 7s 339ms/step - loss: 0.2339 - accuracy: 0.9100
Epoch 24/30
1/22 [.....] - 7s 327ms/step - loss: 0.2290 - accuracy: 0.9028
Output is truncated. View on a scrollable element or open in a text editor. Adjust cell output settings.
```

Gambar 7. Epoch

Setelah model dibuat, model kemudian di-Compile dengan memilih parameter yang ingin digunakan untuk menghasilkan model yang akurat. Membuat callback `early_stop` untuk mengentikan training jika tidak ada perubahan yang signifikan pada akurasi model pada 5 epochs `model_fit` untuk melatih model berdasarkan arsitek model dan `model_compile` `train_model` adalah nama folder/path yang akan di-train dengan jumlah epochs sebanyak dan menerapkan callback yang sudah dibuat.

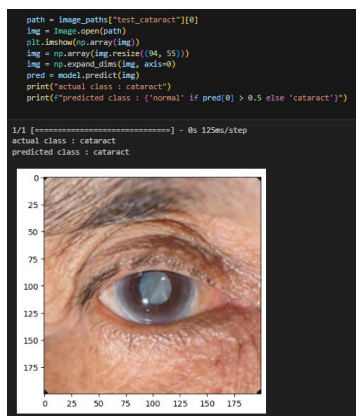
#### d) Training Model



Gambar 8. Loss and Accuracy

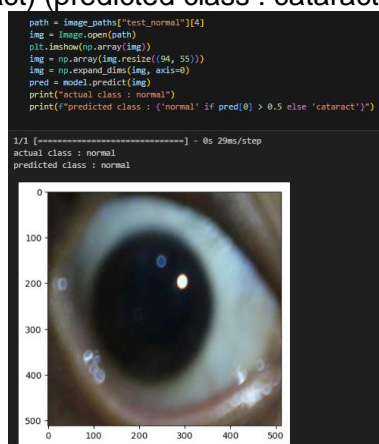
Semua hasil train (accuracy dan loss) disimpan dalam variable history. Lalu data accuracy dan loss diambil untuk membuat plot di atas. Dari grafik diatas bisa dilihat kalau semakin tinggi epochs, accuracy semakin naik (hingga 90%) dan loss semakin menurun (hingga 0.229). Dari data tersebut dapat disimpulkan bahwa model berjalan dengan akurat dalam memprediksi model.

#### e) Predict



Gambar 9. Predict Cataract

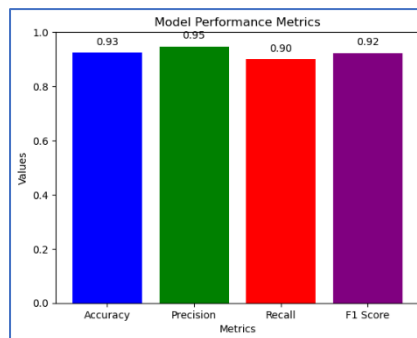
Pada gambar diatas merupakan langkah dalam memprediksi salah satu gambar dengan hasil actual (class : cataract) (predicted class : cataract).



Gambar 10. Predict Normal

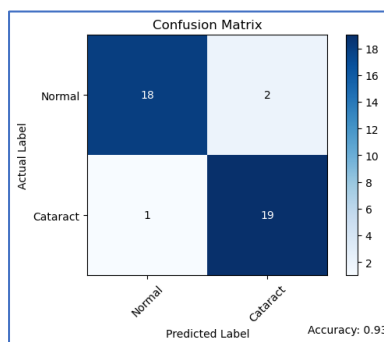
Pada gambar diatas merupakan langkah dalam memprediksi gambar dengan hasil actual (class : normal) (predicted class : normal). Kami juga menambahkan kode untuk jika nilai prediksi > 0.5 maka hasil prediksi akan normal, selain itu katarak.

f) Testing dan Hasil



Gambar 11. Model Performance

Dari tabel balok di atas, nilai keempat performance metrics tersebut di atas 90% dimana accuracy 93%, precision 95%, recall 90%, dan f1 score 92%. Sehingga dapat dikatakan bahwa hasil prediksi kita akurat dan lebih baik dalam memprediksi dataset baru.

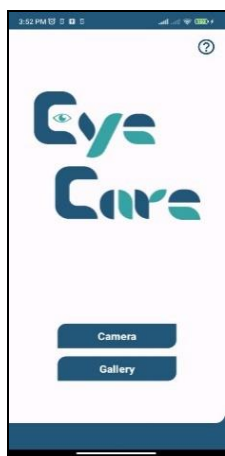


Gambar 12. Confusion Matrix

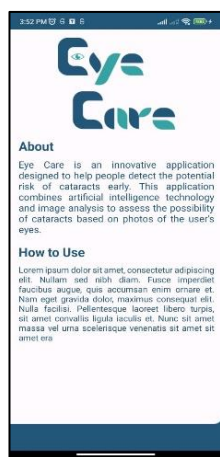
Confusion Matrix dibuat untuk melihat hasil prediksi, data normal yang diprediksi katarak sebanyak 2 gambar dan data katarak yang diprediksi normal sebanyak 1 gambar.

g) Aplikasi

Hasil implementasi dari model yang dibuat sampai bisa dibuat untuk memprediksi kami langsung konfigurasi model ke aplikasi menggunakan API upload dan predict yang dibuat menggunakan Flask dan Express.js. Tampilan aplikasi bisa dilihat pada gambar berikut:



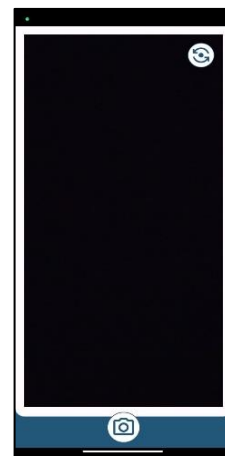
Gambar 13. Home



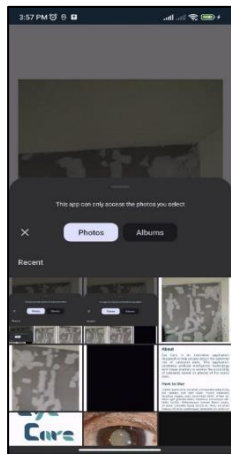
Gambar 14. About



Gambar 15. Depan



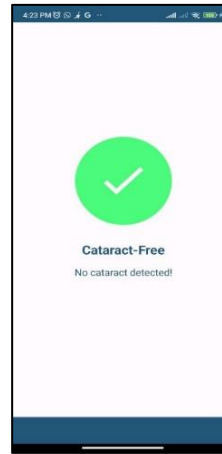
Gambar 16. Belakang



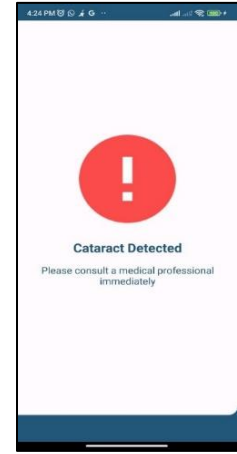
Gambar 13. Galeri



Gambar 14. Review



Gambar 15. Normal



Gambar 16. Katarak

#### 4. Kesimpulan

Berdasarkan hasil penelitian artikel ini membahas pengembangan aplikasi mobile "Eye Care" yang menggunakan analisis gambar menggunakan deep learning image analysis dengan model Convolutional Neural Network (CNN) untuk mendeteksi katarak dari foto mata pengguna. Aplikasi ini bertujuan untuk membantu mengurangi kebutaan di Indonesia, yang sebagian besar disebabkan oleh katarak. Dengan metode pengembangan Waterfall dan pengembangan model menggunakan metode CNN, aplikasi ini dirancang agar mudah digunakan oleh semua usia dan mampu mendeteksi katarak secara akurat sejak dini, sehingga memungkinkan penanganan lebih cepat dan mengurangi risiko kebutaan. Hasil penelitian menunjukkan bahwa akurasi model untuk mendeteksi penyakit katarak yaitu 93%, presisi 95%, recall 90% dan f1 score 92%

#### 5. Referensi

- [1] Verdy and Ery Hartati, "KLASIFIKASI PENYAKIT MATA MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK MODEL RESNET-50," *JRSIT*, vol. 1, no. 3, pp. 199–206, Feb. 2024, doi: 10.59407/jrsit.v1i3.529.
- [2] H. Fahmi, "Sistem Pakar Mendiagnosa Penyakit Mata Katarak Dengan Metode Certainty Factor Berbasis Web," *M*, vol. 11, no. 1, p. 27, Oct. 2019, doi: 10.18860/mat.v11i1.7673.
- [3] M. Aarsal, B. Agus Wardijono, and D. Anggraini, "Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN," *TEKNOSI*, vol. 6, no. 1, pp. 55–63, Jun. 2020, doi: 10.25077/TEKNOSI.v6i1.2020.55-63.
- [4] F. N. Cahya, N. Hardi, D. Riana, and S. Hadiyanti, "Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)," *SISTEMASI*, vol. 10, no. 3, p. 618, Sep. 2021, doi: 10.32520/stmsi.v10i3.1248.
- [5] M. A. N. Rachmad and N. D. Saputro, "SISTEM INFORMASI MANAJEMEN PARIWISATA BERBASIS WEB DI KABUPATEN KENDAL," vol. 2023, 2023.
- [6] M. Zahir and R. A. Saputra, "DETEKSI PENYAKIT RETINOPATI DIABETES MENGGUNAKAN CITRA MATA DENGAN IMPLEMENTASI DEEP LEARNING CNN," vol. 18, 2024.
- [7] D. H. Firdaus, B. Imran, L. D. Bakti, and E. Suryadi, "KLASIFIKASI PENYAKIT KATARAK PADA MATA MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN) BERBASIS WEB".