

# PENGEMBANGAN APLIKASI *MOBILE* UNTUK MENGESTIMASI RENTANG USIA BERDASARKAN FOTO WAJAH DAN *FINE TUNING PRETRAINED CNN*

Muhammad Fadli Gunardi  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Jalan Ganesha 10, Bandung, Indonesia  
m.fadli.gunardi@gmail.com

**Abstrak**—Saat ini, teknologi *computer vision* semakin banyak digunakan untuk membantu kehidupan manusia. Salah satunya, pengaplikasian *computer vision* pada pengenalan wajah untuk mengestimasi usia seseorang. Pendeteksian usia seseorang yang diimplementasikan pada tugas akhir ini menggunakan suatu model pretrained CNN yang telah di fine tuning dengan input berupa foto wajah. Foto wajah tersebut oleh sistem akan dikelompokkan ke rentang usia tertentu. Pengelompokan ini ditujukan untuk mengetahui kelompok usia mereka. Namun, dalam mendeteksi rentang usia seseorang secara akurat dibutuhkan pemilihan dataset dan model pretrained CNN yang tepat. Oleh karena itu, pada tugas akhir ini dilakukan suatu eksperimen model pretrained CNN guna memilih model pretrained yang memiliki nilai akurasi yang tinggi dalam mengestimasi rentang usia. Selanjutnya, berdasarkan studi literatur yang telah dilakukan, pengaplikasian sistem deteksi usia seseorang yang ada saat ini juga masih kurang praktis. Beberapa sistem deteksi usia yang telah dikembangkan masih membutuhkan laptop atau komputer dalam menjalankannya. Selain itu, beberapa tampilan antarmuka sistem deteksi usia masih sangat sederhana, bahkan masih menggunakan command line tanpa suatu GUI. Dengan demikian, pengembangan aplikasi mobile untuk deteksi rentang usia kemudian dipilih pada tugas akhir ini dengan menekankan tampilan antarmuka yang menarik dan pengalaman pengguna yang menyenangkan.

Berdasarkan hasil eksperimen model pretrained CNN, model MobileNet meraih nilai akurasi tertinggi dengan memperoleh nilai 86,17% dari 597 data uji. Dengan demikian, model MobileNet dipilih untuk kemudian diintegrasikan pada aplikasi mobile. Selanjutnya, berdasarkan hasil pengujian aplikasi mobile, sistem sudah dapat memenuhi seluruh kebutuhan fungsional dan non fungsional yang telah didefinisikan sebelumnya. Akan tetapi, sistem saat ini belum dapat digunakan secara offline.

**Kata kunci**—aplikasi *mobile*, pretrained CNN, MobileNet

## I. PENDAHULUAN

### A. Latar Belakang

*Computer vision* merupakan salah satu penerapan *artificial intelligence* yang melibatkan komputer dalam mengatasi berbagai permasalahan dengan cara mengekstrak informasi dari gambar ataupun video. Pada saat ini, teknologi *computer vision* sudah banyak dikembangkan dalam berbagai bidang, contohnya pada bidang otomotif, pengenalan wajah, kesehatan, manufaktur, pertambangan, dll. Misal, pada bidang otomotif *computer vision* dapat diterapkan dalam pendeteksian jarak mobil depan dan pendeteksian batas lajur yang berguna untuk mendukung penerapan kendaraan semi *autonomous*. Selain itu, pada bidang pengenalan wajah *computer vision* dapat diterapkan untuk mengetahui jenis kelamin seseorang, usia, nama, dll. Pada bidang kesehatan *computer vision* dapat diterapkan untuk CT-Scan dan MRI untuk mendiagnosis sesuatu yang terjadi pada pasien. Dengan demikian, *computer vision* sangat membantu dan meringankan beban pekerjaan manusia pada kehidupan sehari-hari di berbagai bidang. Oleh karena itu, dengan keunggulan yang ditawarkan teknologi *computer vision* sangatlah menarik untuk dikembangkan.

Salah satunya pada bidang pengenalan wajah, pengenalan wajah bekerja dengan mencocokkan citra wajah manusia dengan *database* wajah (Srinivas, Tippavajhala, et al, 2022). Pada saat ini, pengenalan wajah semakin banyak diaplikasikan dalam sistem pengenalan biometrik, pengelompokan suatu citra, sistem keamanan, dan interaksi manusia dengan komputer. Dalam melakukan proses pengenalan wajah diperlukan proses pendeteksian wajah terlebih dahulu untuk mendeteksi apakah dalam suatu citra mengandung unsur wajah atau tidak. Setelah sistem mendeteksi adanya unsur wajah pada citra, perlu diterapkan *image preprocessing* untuk citra-citra yang memiliki pencahayaan yang rendah, blur, atau kontras yang rendah begitupun sebaliknya. Proses ini ditujukan agar model yang digunakan dalam pengenalan wajah dapat bekerja untuk mengenali wajah tersebut secara akurat.

Salah satu pengaplikasian pengenalan wajah yang ada saat ini adalah pendeteksian usia seseorang. Kehadiran sistem deteksi usia ini disebabkan oleh manusia yang masih sangat sulit memprediksi usia seseorang hanya dengan melihat wajah seseorang. Manusia hanya mampu mengenali wajah tersebut ke dalam kelompok bayi, balita, anak-anak, remaja, dewasa, atau lansia. Hal tersebut disebabkan karena manusia jarang mengetahui dan mengingat usia seseorang sehingga tidak mempunyai ingatan atas usia seseorang yang cukup banyak. Lain halnya dengan suatu *artificial intelligence* yang diciptakan untuk mengestimasi usia seseorang, berbekal dengan pembelajaran *deep learning* dari suatu *dataset* yang berisi puluhan ribu atau bahkan jumlahnya menyentuh jutaan citra yang telah dilabeli usianya. Tentu, model tersebut dapat lebih akurat dalam mengestimasi usia seseorang dibandingkan manusia. Selain itu, pengaplikasian *computer vision* untuk mengestimasi usia seseorang juga mempunyai beberapa manfaat, diantaranya adalah untuk verifikasi usia seseorang dalam melakukan proses registrasi suatu akun, kebutuhan forensik (Gita, Ilevina, Novita, et al, 2020), dan sebagai sarana pembantu manusia dalam mengetahui usia seseorang.

Dalam melakukan estimasi usia seseorang secara akurat dibutuhkan model CNN yang memiliki nilai akurasi yang tinggi. Berdasarkan studi literatur yang telah dilakukan, didapatkan bahwa model *pretrained CNN* berupa arsitektur VGGnet atau VGG16 memperoleh nilai akurasi tertinggi dalam mengestimasi usia seseorang, hal ini dibuktikan dalam penelitian yang dilakukan (Daniel, 2018) model VGG16 yang telah dilakukan proses *fine tune* dan modifikasi *layer* dapat memperoleh nilai akurasi sebesar 99,6% dari 2.801 *test data* yang berasal dari *dataset* LFW yang telah dilakukan proses *face alignment* dan augmentasi citra dengan teknik penambahan noise, penerapan blur, dan penajaman citra terlebih dahulu. Selain itu, pada penelitian (Anand, et al, 2017) VGG16 yang telah dilakukan proses *fine tune* bahkan memperoleh nilai akurasi 64% dengan *dataset* WIKI yang memiliki kondisi pencahayaan citra yang buruk dan arah pandang wajah yang tidak teratur. Selanjutnya, pada penelitian (Barkana, et al, 2017) didapatkan bahwa model *pretrained VGG16* yang dilakukan proses modifikasi *layer* pada dua *hidden layer* terakhir memperoleh nilai akurasi 59,9% dengan *dataset* *Adience*, sedangkan dengan proses yang sama model *pretrained GoogLeNet* memperoleh nilai akurasi 45,07%. Dengan demikian, pada tugas akhir ini perlu ditentukan model *pretrained* dan *dataset* yang tepat untuk digunakan pada aplikasi estimasi usia ini.

Selanjutnya, berdasarkan studi literatur yang telah dilakukan, didapatkan juga bahwa implementasi model CNN dalam mengestimasi usia masih belum praktis dalam penggunaannya. Hal ini ditandai dengan beberapa model CNN yang telah dikembangkan saat ini masih membutuhkan laptop atau komputer untuk menjalankannya sehingga tidak praktis untuk digunakan dalam mengestimasi usia seseorang dalam kehidupan sehari-hari. Selain itu, tampilan antarmuka hasil pengembangan model CNN untuk mengestimasi usia masih sangat sederhana, bahkan mayoritas masih menggunakan *command line* via *code editor* tanpa menggunakan GUI. Oleh

karena itu, dibutuhkan suatu aplikasi dengan tampilan antarmuka yang menarik untuk meningkatkan pengalaman pengguna.

Dalam hal kepraktisan, ponsel pintar merupakan hal yang tepat untuk menjawab permasalahan itu karena saat ini penggunaan ponsel pintar sudah merupakan hal yang biasa digunakan masyarakat pada kehidupan sehari-hari. Berbagai kalangan pun sudah memiliki ponsel pintar, mulai dari remaja hingga lansia. Dengan demikian, melihat potensi penggunaan ponsel pintar yang cukup besar untuk digunakan dalam kehidupan sehari-hari, maka pada tugas akhir ini dikembangkan suatu aplikasi mobile dalam mengestimasi usia seseorang menggunakan *pretrained CNN*.

## B. Rumusan Masalah

Rumusan masalah yang dirumuskan dalam pengerjaan tugas akhir ini adalah sebagai berikut

1. Model *pretrained CNN* apa yang memiliki nilai akurasi tinggi dalam mengestimasi usia seseorang?
1. Bagaimana pengaplikasian model *pretrained CNN* pada aplikasi *mobile* dapat membantu dan mempermudah manusia dalam mengestimasi rentang usia seseorang?

## C. Tujuan

Adapun tujuan dari pengerjaan tugas akhir ini adalah sebagai berikut.

1. Memilih model *pretrained CNN* yang memiliki nilai akurasi yang tinggi.
2. Melakukan proses *fine tune* model *pretrained CNN*.
3. Mengukur dan mengevaluasi kinerja model *pretrained CNN* untuk mendapatkan nilai akurasi yang tinggi.
4. Membangun perangkat lunak berbentuk aplikasi mobile dengan mengintegrasikan model *pretrained CNN*.
5. Mengevaluasi perangkat lunak yang telah dibangun dari sisi kebutuhan fungsional dan non fungsional.

## D. Batasan Masalah

Batasan masalah dalam pengerjaan tugas akhir ini adalah sebagai berikut.

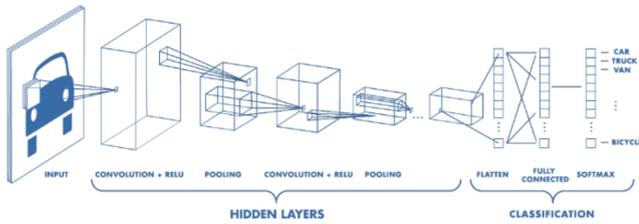
1. Model *pretrained CNN* yang dibandingkan adalah VGG16, ResNet50, MobileNet.
2. *Dataset* yang digunakan dalam pelatihan model CNN sudah tersedia.
3. Citra pada *dataset* sudah melalui proses penggandaan atau augmentasi berupa penambahan blur, penajaman, modifikasi pencahayaan, dan rotasi.
4. Pengguna memiliki akses internet dalam menggunakan aplikasi *mobile*.

## II. DASAR TEORI

### A. Convolutional Neural Network (CNN)

*Convolutional neural network*, juga dikenal sebagai CNN atau ConvNet, adalah kelas *neural network* yang mengkhususkan diri dalam pemrosesan data yang memiliki topologi seperti grid atau gambar (Mishra, Mayank, 2020).

Selain hal tersebut, CNN juga dapat mengenali informasi prediktif, seperti teks dan potongan suara. CNN termasuk ke dalam jenis *deep neural network* karena kedalaman jaringan yang tinggi. Maksud dari kedalaman jaringan yang tinggi ini adalah CNN memiliki berbagai macam *layer*, diantaranya adalah *convolution layer*, fungsi *activation layer*, *pooling layer*, dan *fully connected layer*. lapisan-lapisan tersebut disusun sedemikian rupa sehingga dapat mendeteksi pola yang lebih sederhana terlebih dahulu (garis, titik, kurva, dll) yang kemudian lanjut ke pola yang lebih kompleks, seperti wajah dan objek. Berikut merupakan ilustrasi dari arsitektur CNN.



Gambar Arsitektur CNN (Mishra, M., 2020)

CNN memiliki beberapa *layer*, diantaranya adalah sebagai berikut

### 1. Convolution Layer

*Convolution Layer* atau yang biasa disebut dengan filter konvolusi/lapisan konvolusi merupakan proses utama yang mendasari pembentukan *convolutional neural network*. Pada lapisan ini, dilakukan perkalian produk titik antara dua matriks, di mana satu matriks adalah himpunan parameter yang dapat dipelajari atau dikenal sebagai kernel dan matriks lainnya adalah bagian terbatas dari bidang reseptif

### 2. Pooling Layer

Proses yang terjadi pada *Pooling Layer* kurang lebih sama dengan convolutional layer, *Pooling Layer* bertanggung jawab untuk mengurangi ukuran spasial dari *convolved feature* yang telah melalui proses konvolusi. Proses ini dilakukan untuk mengurangi daya komputasi yang diperlukan untuk memproses data melalui pengurangan dimensi (Saha, 2018). Selain itu, pooling layer juga berguna untuk mengekstraksi fitur dominan berupa invarian rotasi dan posisi sehingga dapat mempertahankan proses pelatihan model secara efektif (Saha, 2018).

### 3. Fully Connected Layer

*Fully Connected Layer* merupakan sebuah lapisan dimana semua *neuron* aktivasi dari lapisan sebelumnya terhubung dengan *neuron* lapisan selanjutnya. Proses yang terjadi pada lapisan ini adalah perkalian matriks biasa dengan fungsi aktivasi yang pada umumnya adalah *Softmax* dan diikuti dengan bias. *Fully Connected Layer* berguna untuk membantu memetakan representasi antara input dan output.

Selain itu, CNN juga memiliki fungsi aktivasi yang berfungsi untuk menentukan apakah *neuron* tersebut harus “aktif” atau tidak berdasarkan dari *weighted sum* dari input (Sena, 2017). Secara umum terdapat 2 jenis fungsi aktivasi, linear dan nonlinear. Beberapa jenis fungsi aktivasi yang sering digunakan pada *deep learning* adalah fungsi aktivasi nonlinear, seperti *Sigmoid*, *Tanh*, *ReLU*, dan *Softmax* (Khan dkk., 2018). Hal tersebut disebabkan oleh operasi konvolusi yang merupakan operasi linear, sedangkan gambar jauh dari kata linier. Oleh karena itu, lapisan nonlinear sering ditempatkan langsung setelah lapisan konvolusi untuk menghasilkan *non-linearity* ke peta aktivasi.

### B. Hyperparameter

*Hyperparameter* merupakan variabel yang menentukan struktur jaringan dan variabel yang menentukan bagaimana suatu jaringan dilatih (Radhakrishnan, Pranoy, 2017). *hyperparameter* perlu ditentukan sebelum *training* untuk mengoptimalkan bobot dan bias. Terdapat 2 jenis *hyperparameter*, yaitu *hyperparameter* terkait struktur jaringan dan *hyperparameter* terkait algoritma training data. Contoh dari *hyperparameter* terkait struktur jaringan adalah jumlah *hidden layer*, jumlah *neuron*, dan jenis fungsi aktivasinya, sedangkan *hyperparameter* terkait training model adalah *learning rate*, *epoch*, dan *batch size*.

### C. Transfer Learning

*Transfer learning* adalah subbidang dari pembelajaran mesin dan kecerdasan buatan yang bertujuan untuk menerapkan pengetahuan yang diperoleh dari suatu *task* ke *task* yang berbeda tetapi serupa (Yalçın, G., 2021). Misalnya, pengetahuan yang didapat suatu model saat belajar mengklasifikasikan teks Wikipedia dapat digunakan untuk mengatasi masalah klasifikasi teks hukum, sedangkan contoh yang salah adalah menggunakan pengetahuan yang didapat saat belajar mengklasifikasikan mobil untuk mengenali burung di langit (Yalçın, G., 2021). Oleh karena itu, metode *transfer learning* hanya dapat diterapkan pada pembelajaran yang jenisnya serupa.

Dalam *transfer learning*, pertama-tama dilakukan pelatihan neural *network* pada kumpulan dataset awal, kemudian menggunakan atau mentransfer kembali fitur yang telah dipelajari oleh model neural network ke *target network* untuk dilatih pada *dataset* yang akan dipilih (Brownlee, J., 2019). *Transfer learning* hanya bekerja apabila fitur yang dipelajari oleh model deep learning dari tugas pertama bersifat umum (Brownlee, J., 2019). Namun demikian, *transfer learning* merupakan metode yang sangat menarik dan memberikan banyak keuntungan dalam menyelesaikan masalah yang membutuhkan *deep learning* karena mengingat waktu yang diperlukan sangat besar dalam membangun dan melatih model *deep learning* pada suatu *dataset* yang besar (Brownlee, J., 2019).

### D. Fine Tuning

Fine tuning adalah proses yang menggunakan model yang telah dilatih untuk satu tugas yang telah diberikan sebelumnya untuk kemudian dilakukan tuning atau men-tweak model

tersebut untuk membuatnya dapat melakukan tugas kedua yang serupa. Dengan kata lain, fine tuning adalah cara menerapkan atau memanfaatkan pembelajaran transfer. Proses fine tuning dilakukan untuk memperoleh nilai akurasi yang lebih tinggi dan menghasilkan output yang sesuai dengan task yang diinginkan (Yalçın, G., 2021).

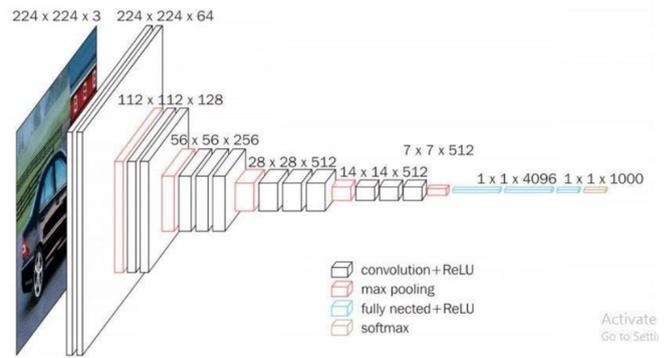
Secara umum, proses *fine tuning* dilakukan dengan membuang lapisan atas pada *neural network* karena lapisan tersebut mewakili fitur khusus masalah. Selanjutnya, dilakukan penambahan lapisan khusus yang disesuaikan dengan *output* yang kita inginkan. Misalnya, model yang dilatih dengan *ImageNet* dapat mengklasifikasikan hingga 1000 objek. Jika kita mencoba untuk mengklasifikasikan digit tulisan tangan (misalnya, klasifikasi MNIST), mungkin lebih baik diakhiri dengan lapisan terakhir dengan hanya 10 *neuron*. Terakhir, setelah dilakukan penambahan lapisan khusus ke model *pretrained*, dapat dilakukan konfigurasi tambahan dengan *special loss functions* dan *optimizers* untuk mendapatkan hasil yang maksimal (Yalçın, G., 2021).

### E. Pretrained Model

Model *pretrained* adalah model yang dibuat dan dilatih oleh orang lain untuk memecahkan masalah yang mirip dengan masalah kita. Model *pretrained* biasanya telah dilatih dengan dataset yang sangat besar, seperti ImageNet atau Wikipedia Corpus. Dalam *pretrained* model, kita dapat memanfaatkan pengetahuan yang telah dipelajari oleh suatu model yang meliputi *features*, *weights*, dan *biases*. Dalam menggunakan pretrained model, kita dapat menggunakan *library*, seperti TensorFlow, Keras, PyTorch, dll. Beberapa contoh dari Pretrained model yang akan dieksperimenkan pada tugas akhir ini adalah VGG16, ResNet50, MobileNet. Alasan pemilihan model-model tersebut adalah ketiganya telah dilatih sebelumnya dengan *dataset* ImageNet yang memiliki 14 juta gambar dan lebih dari 20.000 kategori sehingga task untuk mengklasifikasikan usia masih tergolong serupa dengan pelatihan model sebelumnya. Model-model tersebut akan dibahas secara lebih detail di bawah ini.

#### 1. VGG16

VGG16 adalah model CNN yang secara umum memiliki 13 lapisan konvolusi, 5 lapisan *max pooling*, dan 3 *dense layer* sehingga total memiliki 21 layer (Rohini, G., 2021). VGG16 dibangun dan dilatih oleh Karen Simonyan dan Andrew Zisserman di Universitas Oxford pada tahun 2014. Model ini telah dilatih dengan lebih dari 14 juta gambar yang termasuk dari 22.000 kategori pada *database* ImageNet. Model ini dapat mengklasifikasikan hingga 1000 kategori, serta telah dilatih pada gambar berwarna berukuran 224x224 piksel (Yalçın, G., 2021). Secara umum ilustrasi VGG16 dapat dilihat pada gambar 1 di bawah ini.



Gambar 1 Ilustrasi arsitektur VGG16 (Rohini, G., 2021)

#### 2. ResNet50

ResNet50 adalah model CNN yang memiliki kedalaman sebanyak 50 lapisan. ResNet50 dibuat dan dilatih oleh Microsoft pada tahun 2015. Model ini telah dilatih dengan lebih dari 1 juta gambar dari *database* ImageNet. Sama seperti VGG16, model ini dapat mengklasifikasikan hingga 1000 kategori, serta telah dilatih pada gambar berwarna berukuran 224x224 piksel (Yalçın, G., 2021). ResNet sangat menarik untuk dieksperimenkan karena dengan tingginya jumlah *layer* yang dimiliki suatu model pretrained terdapat beberapa permasalahan yang terjadi antara lain. Secara umum ilustrasi ResNet dapat dilihat pada gambar 2 di bawah ini pada kolom yang bernama “50-layer”.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112x112	7x7, 64, stride 2				
conv2.x	56x56	3x3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3.x	28x28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4.x	14x14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5.x	7x7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1x1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

Gambar 2 Ilustrasi arsitektur ResNet (Kaushik, A., 2020)

#### 3. MobileNet

MobileNet merupakan salah satu model *pretrained* CNN yang dibangun oleh Google dan sangat cocok untuk digunakan pada perangkat *mobile* karena ukuran model yang sangat kecil dan tingkat *latency* yang rendah. MobileNet pun sama dengan model VGG16 dan ResNet, telah di-*train* terlebih dahulu dengan dataset ImageNet. Meskipun arsitektur dasar MobileNet kecil dan secara komputasi tidak terlalu berat, namun MobileNet dapat memperoleh nilai akurasi yang cukup tinggi. Hal tersebut disebabkan oleh penggunaan dua *hyperparameter* global berupa *depthwise separable convolutions* untuk menggantikan *layer* konvolusi pada umumnya sehingga efektif untuk mengurangi biaya komputasi

lebih lanjut (Singh, N., 2020). Secara umum ilustrasi MobileNet dapat dilihat pada gambar 3 di bawah ini.

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5 × Conv dw / s1	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 512	14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Gambar 3 Detail arsitektur MobileNet (PA, S., 2020).

#### F. Dataset UTK Face

*Dataset UTKFace* merupakan *dataset* wajah berskala besar dengan rentang usia yang panjang, yaitu berkisar antara 0 hingga 116 tahun. *Dataset* ini terdiri dari 23.708 gambar wajah dengan label berupa usia, jenis kelamin, dan etnis sehingga cocok untuk berbagai jenis *task* klasifikasi. Gambar-gambar pada *dataset* ini pun memiliki variasi besar dalam pose, ekspresi wajah, pencahayaan, resolusi, dll. Dengan demikian *dataset* ini dapat digunakan pada berbagai tugas, misal deteksi wajah, estimasi usia, perkembangan atau regresi usia, lokalisasi landmark, dll (Susanqq, 2017).

### III. ANALISIS DAN PERANCANGAN SOLUSI ESTIMASI RENTANG USIA

#### A. Analisis Masalah

Mengacu pada bab I, terdapat beberapa masalah yang akan diselesaikan pada tugas akhir ini, diantaranya adalah sebagai berikut.

1. Nilai akurasi dari model *pretrained* CNN nilainya beragam

Berdasarkan pada penelitian-penelitian yang telah dilakukan sebelumnya, didapatkan bahwa nilai akurasi yang diperoleh dari suatu *pretrained* model yang diujikan pada *dataset* yang berbeda akan memperoleh nilai yang sangat berbeda. Hal ini tentu akan menjadi suatu permasalahan dalam tugas akhir ini karena *dataset* yang digunakan tentu akan berbeda dengan *dataset* yang telah digunakan pada beberapa penelitian-penelitian sebelumnya.

2. Pengaplikasian sistem deteksi usia seseorang masih kurang praktis

Berdasarkan studi literatur yang telah dilakukan, didapatkan bahwa pengaplikasian sistem deteksi usia masih kurang praktis karena masih membutuhkan

perangkat laptop atau komputer dalam menjalankannya. Penggunaan perangkat laptop juga membatasi penggunaan perangkat lunak ini dalam kehidupan sehari-hari sehingga kurang praktis.

3. Kurangnya pengalaman pengguna pada perangkat lunak estimasi usia yang sudah ada

Berdasarkan studi literatur yang telah dilakukan, didapatkan juga bahwa tampilan antarmuka sistem dalam mengestimasi usia masih sangat sederhana, bahkan bahkan mayoritas masih menggunakan command line via code editor tanpa menggunakan GUI. Oleh karena itu, pengalaman yang pengguna rasakan masih sangat kurang sehingga jarang digunakan dalam kehidupan sehari-hari

#### B. Analisis Solusi

Berdasarkan studi literatur yang telah dilakukan dan penggunaan metode *crisp-dm* untuk menyelesaikan masalah-masalah utama yang telah diuraikan pada subbab III.1, didapatkan beberapa solusi atas permasalahan tersebut.

1. Eksperimen *fine tuning* model *pretrained* CNN.
2. Pengklasifikasian *output* dari model *pretrained* ke suatu rentang usia.
3. Pengembangan aplikasi *mobile* yang memiliki desain antarmuka yang menarik.

Metodologi CRISP-DM atau *Cross-Industry Standard Process for Data Mining* dilakukan untuk menyelesaikan masalah pada *data mining* dan *data modeling* sesuai dengan kebutuhan yang telah ditentukan. metodologi ini dimulai dari *business understanding*, yaitu mengetahui kebutuhan obyektif yang ingin dicapai untuk kemudian dirumuskan ke beberapa strategi untuk mencapai tujuan tersebut. Dalam persoalan estimasi usia ini tujuannya adalah mengestimasi usia seseorang secara akurat sehingga dibutuhkan *dataset* yang memiliki jumlah data yang cukup banyak dan memiliki persebaran usia yang luas, serta gambar pada *dataset* tersebut harus mengakomodir berbagai macam kondisi pencahayaan, pose, blur, dan resolusi. Oleh karena itu, dipilihlah *dataset UTK Face* karena *dataset* tersebut sudah memenuhi berbagai persyaratan diatas.

Selanjutnya, saat ini kebutuhan untuk mengklasifikasikan seseorang ke suatu rentang usia semakin potensial. Hal ini dapat dimanfaatkan pada sistem verifikasi usia atau dimanfaatkan untuk mengetahui persebaran kelompok usia di suatu tempat. Selain itu, dengan mengelompokkan usia ke suatu rentang usia dapat meminimalkan *error* yang dihasilkan model *pretrained* atas hasil prediksi yang dilakukan. Dengan demikian, dilakukanlah poin 2 di atas pada tugas akhir ini. Selanjutnya, untuk memastikan model *pretrained* mana yang terbaik untuk diintegrasikan ke dalam aplikasi *mobile*, diperlukan suatu eksperimen. Hal ini dilakukan karena berdasarkan analisis masalah pada bab III ini dan studi literatur yang telah dilakukan, nilai akurasi dari suatu model *pretrained* beragam dan bergantung pada *dataset* dan arsitektur yang digunakan. Oleh karena itu, dilakukan solusi pada poin 1 pada tugas akhir ini.

Selanjutnya tahapan pada *crisp-dm* yang kedua adalah *data understanding*, *data understanding* adalah proses untuk

mengidentifikasi masalah-masalah yang akan terjadi pada *dataset* yang digunakan. Pada *dataset UTK Face* masalah yang kemungkinan terjadi adalah persebaran jumlah usia yang tidak merata, misal usia 20-30 cenderung lebih banyak daripada usia 10-20. Permasalahan ini akan diselesaikan dengan penyesuaian rentang usia yang ada sehingga distribusi datanya lebih merata. Untuk selengkapnya dapat dilihat secara lebih detail pada bab IV.

Tahapan selanjutnya pada *crisp-dm* adalah *data preparation*. *Data preparation* dilakukan untuk menyelesaikan masalah pada suatu *dataset*. Masalah yang terjadi pada *dataset UTK Face* adalah *noise data* atau *data outlier* yang sangat tinggi karena persebaran usia pada *dataset* tersebut sangat lebar, yaitu 0 hingga 116 tahun. Untuk menyelesaikan masalah ini data yang diluar kategori rentang usia yang telah ditentukan akan dibuang sehingga tidak mempengaruhi proses *fine tuning* yang dilakukan pada model *pretrained*.

Tahapan selanjutnya pada metodologi *crisp-dm* adalah *modeling*. *Modeling* adalah suatu tahapan yang menggunakan suatu model ML atau statistika untuk mendapatkan insight dari suatu data. Dalam kasus ini *modeling* yang dilakukan menggunakan model *pretrained* dengan mengklasifikasikan ke suatu rentang usia. Penggunaan model *pretrained*, seperti detail *fine tuning* dan hasil klasifikasi secara lebih detail akan dibahas pada bab IV. Untuk tahapan-tahapan selanjutnya, seperti *evaluation* dan *deployment* akan dijelaskan pada bab IV dan subbab III.4.

Selain dari metodologi *crisp-dm* yang digunakan, dibutuhkan juga solusi atas permasalahan pada analisis masalah poin 2 dan 3. Untuk menyelesaikan masalah-masalah tersebut akan dibangun suatu aplikasi *mobile* yang akan didesain semenarik mungkin dan sesimpel mungkin agar dapat meningkatkan *user experience* yang dirasakan pengguna. Pengimplementasian sistem deteksi usia juga dilakukan pada suatu perangkat *mobile*. Hal ini dilakukan agar penggunaan sistem ini dapat praktis dan dapat digunakan pada kehidupan sehari-hari dimanapun *user* berada. Dengan demikian berdasarkan analisis solusi di atas, tugas akhir ini akan dirangkum ke dalam 5 tahap implementasi solusi, yaitu sebagai berikut.

#### 1. *Data preprocessing*

*Data preprocessing* dilakukan pada *dataset UTK Face* untuk menangani persebaran jumlah data yang tidak merata dan menangani *noise data* atau *outlier data* karena *dataset UTK Face* memiliki rentang usia yang sangat lebar, yaitu 0 hingga 116 tahun. Pada tahap ini data pada *dataset* akan dikelompokkan ke suatu rentang usia yang mendekati ideal sehingga persebaran data antar kelas pada *dataset* lebih merata. Proses ini akan dibahas secara lebih detail pada bab selanjutnya.

#### 2. Proses *fine tune* model-model *pretrained CNN*

Pada proses ini akan dilakukan *pretrained* dari model-model CNN yang akan dipilih. Model-model tersebut adalah VGG16, ResNet50, MobileNet. Proses *fine tune* yang dilakukan pada ketiga model tersebut dilakukan dengan *dataset UTK Face* guna

memilih model *pretrained* terbaik, yaitu model yang memiliki nilai akurasi tertinggi. Proses *fine tune* juga perlu dilakukan karena model-model *pretrained* tersebut karena model-model tersebut belum pernah dilakukan *training* dengan *dataset* yang akan digunakan sehingga *output* yang dihasilkan belum sesuai dengan yang diinginkan. Proses *fine tuning* model-model *pretrained* juga dilakukan untuk meningkatkan nilai akurasi yang diperoleh.

#### 3. Evaluasi dan pemilihan model *pretrained CNN*

Selanjutnya, dilakukan evaluasi dari proses *fine tune* ketiga model tersebut. Hal ini ditujukan untuk mengukur kinerja model-model *pretrained CNN* yang telah di *fine tune* untuk kemudian menentukan apakah nilai akurasi yang didapatkan dari hasil testing model-model *pretrained* dengan data uji sudah cukup tinggi atau belum. Proses ini cukup penting untuk menjamin keakuratan yang didapat model dalam mengestimasi rentang usia seseorang nantinya. Pada tahap ini juga akan dilakukan pemilihan model *pretrained CNN* yang akan digunakan pada aplikasi *mobile* nantinya. pemilihan tersebut didasari oleh perbandingan nilai akurasi dari model-model yang telah di-*fine tune*.

#### 4. Pengembangan aplikasi *mobile*

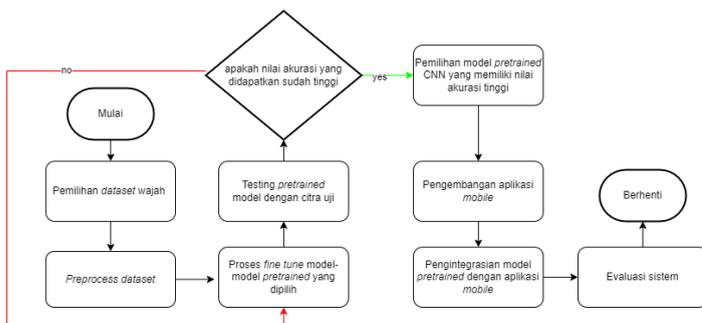
Selanjutnya, apabila model *pretrained CNN* yang telah di *fine tune* telah memiliki nilai akurasi yang cukup dalam mengestimasi rentang usia seseorang, maka akan dilanjutkan dengan tahap pengembangan aplikasi *mobile*. Pengembangan aplikasi *mobile* dilakukan dengan menerapkan desain antarmuka yang menarik dan fitur-fitur yang mendukung dalam mengestimasi rentang usia seseorang pada kehidupan sehari-hari. Penerapan desain antarmuka yang menarik dilakukan untuk menyelesaikan masalah utama yang ketiga, yaitu kurangnya pengalaman pengguna yang dirasakan.

#### 5. Pengintegrasian model *pretrained CNN*

Tahapan terakhir setelah pengembangan aplikasi *mobile* selesai adalah pengintegrasian model *pretrained CNN* yang telah dilakukan proses *fine tune* ke dalam sebuah aplikasi *mobile* yang sudah jadi untuk kemudian dapat dilakukan testing dan evaluasi.

### C. Gambaran Solusi

Untuk menggambarkan tahapan implementasi solusi atas analisis solusi yang telah dilakukan dibuat suatu diagram alir. Diagram alir tersebut dapat dilihat pada gambar 4 di bawah ini.



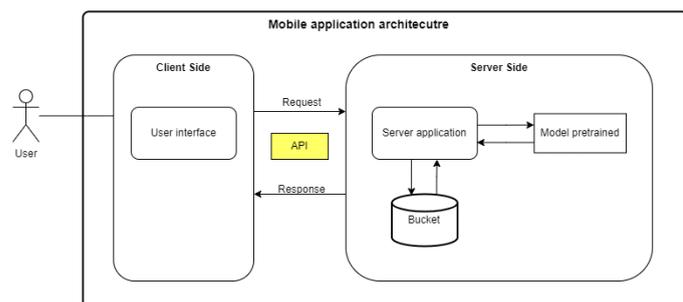
Gambar 4 Flowchart sistem

#### D. Perancangan Aplikasi Mobile

Aplikasi *mobile* yang dibangun akan mendukung penggunaan pada *platform* Android dan ios. Hal tersebut dilakukan agar penggunaan aplikasi *mobile* ini diharapkan dapat digunakan pada seluruh lapisan masyarakat Indonesia. Untuk memenuhi kebutuhan tersebut, pengembangan aplikasi *mobile* dapat dilakukan secara *native* atau *cross platform*. Pada tugas akhir ini, pengembangan aplikasi *mobile* akan menggunakan teknologi *cross platform* karena dapat menghasilkan produk akhir lebih cepat. Hal tersebut disebabkan teknologi *cross platform* yang memungkinkan penggunaan satu basis kode untuk diimplementasikan pada lebih dari satu *platform*.

Teknologi *cross platform* yang akan digunakan pada tugas akhir ini adalah Flutter. Pemilihan ini didasarkan oleh beberapa kelebihan yang ditawarkan Flutter dibandingkan teknologi *cross platform* lainnya, seperti React Native. Salah satu kelebihan Flutter adalah banyaknya komponen *widget* yang telah disediakan dan dapat digunakan langsung dalam pengembangannya. Selain itu, Flutter tidak membutuhkan *framework* CSS tambahan karena *styling* dapat dilakukan langsung pada komponen-komponen *widget* mereka.

Secara umum arsitektur aplikasi yang akan dirancang dapat dilihat pada gambar 5, dimana *user* akan akan berinteraksi dengan sistem melalui antarmuka yang telah dirancang.



Gambar 5 Arsitektur aplikasi

fungsional dan non fungsional sistem ditunjukkan pada tabel 1 dan 2.

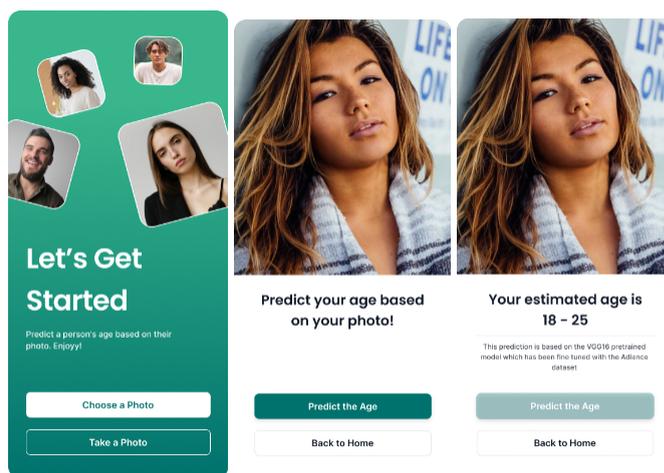
Tabel 1 Kebutuhan fungsional

ID	Kebutuhan Fungsional
F-01	Sistem dapat menampilkan rentang usia berdasarkan gambar <i>input</i>
F-02	Sistem dapat mendeteksi suatu wajah berdasarkan gambar <i>input</i>
F-03	Sistem dapat melakukan <i>cropping</i> bagian wajah dari gambar <i>input</i>
F-04	Sistem dapat menerima gambar <i>input</i> dari kamera HP
F-05	Sistem dapat menerima gambar <i>input</i> dari galeri HP

Tabel 2 Kebutuhan non fungsional

ID	Parameter	Kebutuhan Non Fungsional
NF-01	<i>Usability</i>	Sistem dapat dengan mudah digunakan dan dipahami

Adapun tampilan antarmuka aplikasi ini akan dibagi menjadi 2 tampilan utama, yaitu halaman *landing page* dan halaman *result page*. Seluruh tampilan antarmuka menggunakan *color pallete* dan *layouting* yang didesain semenarik mungkin untuk mendukung pengalaman pengguna yang lebih baik. Rancangan desain tampilan antarmuka dapat dilihat secara lebih jelas pada gambar 6.



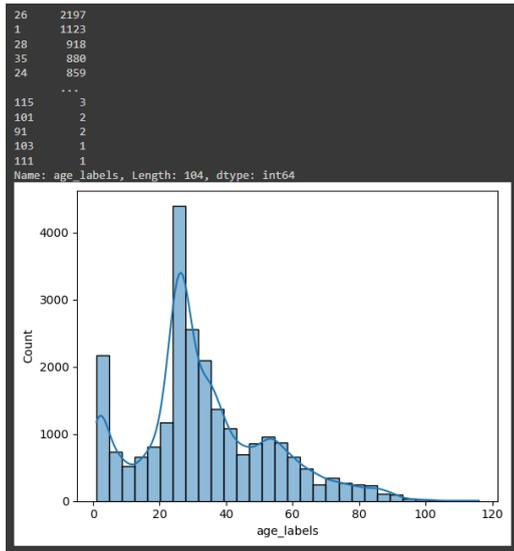
Gambar 6 Desain tampilan antarmuka

Selain itu, untuk menentukan fitur-fitur yang akan dibangun akan didefinisikan terlebih dahulu kebutuhan-kebutuhan fungsional dan non fungsional sistem. Adapun kebutuhan

#### IV. EKSPERIMEN PEMILIHAN MODEL *PRETRAINED* CNN DAN HASIL PENGUJIAN

##### A. *Preprocess dataset* dan pengklasifikasian usia

*Preprocess dataset* dilakukan untuk memastikan data yang digunakan pada model pretrained CNN sudah sesuai sehingga tidak berdampak pada learning process dari suatu model pretrained. *preprocess dataset*. Pada eksperimen ini, pertama-tama dilakukan visualisasi terlebih dahulu terkait jumlah distribusi data pada dataset sebelum dapat dilakukan *preprocess*. Jumlah distribusi data pada dataset yang dapat dilihat pada gambar 7 di bawah ini.



Gambar 7 Grafik distribusi jumlah data pada *dataset* UTK Face

Berdasarkan gambar 7 dapat dilihat bahwasanya persebaran jumlah data pada usia tertentu tidak seimbang dan bahkan perbedaan jumlah label usia dari yang paling banyak hingga yang paling sedikit jumlahnya sangat signifikan hingga 1 sampai 2 kali lipat. Hal tersebut perlu diperbaiki agar tidak terjadi risiko dimana model *pretrained* lebih mengenali usia tertentu dibanding usia-usia lainnya. Selanjutnya, dilakukan pemilihan rentang usia yang tepat dengan menimbang kelompok umur yang diakui oleh Departemen Kesehatan RI (2009) dan ketersediaan data pada *dataset*. Menurut Departemen Kesehatan RI, kelompok usia dikelompokkan ke dalam 9 kategori antara lain adalah sebagai berikut.

- a. masa balita (usia 0-5 tahun)
- b. masa kanak-kanak (usia 5-11 tahun)
- c. masa remaja awal (usia 12-16 tahun)
- d. masa remaja akhir (usia 17-25 tahun)
- e. masa dewasa awal (usia 26-35 tahun)
- f. masa dewasa akhir (usia 36-45 tahun)
- g. masa lansia awal (usia 46-55 tahun)
- h. masa lansia akhir (usia 56-65 tahun)
- i. masa manula (usia 65 tahun ke atas)

Namun, setelah melakukan eksperimen dengan pengkategorian tersebut hasil akurasi yang diperoleh model hanya memperoleh masih tergolong rendah sehingga masih belum cukup akurat. Hal tersebut disebabkan oleh jumlah distribusi data antar kelas yang sangat berbeda. Proses yang dilakukan pada eksperimen tersebut adalah membandingkan nilai akurasi kebenaran pengklasifikasian hasil testing model pretrained VGG16, ResNet, dan MobileNet dengan data uji. Tentu, ketiga model tersebut telah di-*fine tuning* terlebih dahulu dengan beberapa *hyperparameter* dan *additional layer* untuk menyesuaikan dengan *output* yang diinginkan. Hasil pengujian pengklasifikasian ke suatu rentang usia menurut Departemen Kesehatan dapat dilihat pada tabel 3 di bawah ini.

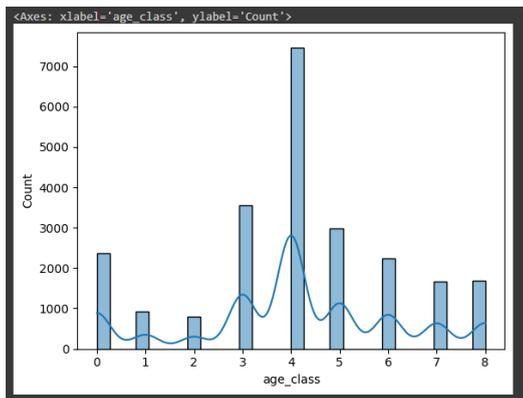
Tabel 3 Hasil pengujian sebelum data *preprocess*

Model <i>Pretrained</i>	Nilai akurasi sebelum data <i>preprocessing</i>
VGG16	53,43%
ResNet	40,48%
MobileNet	53,11%

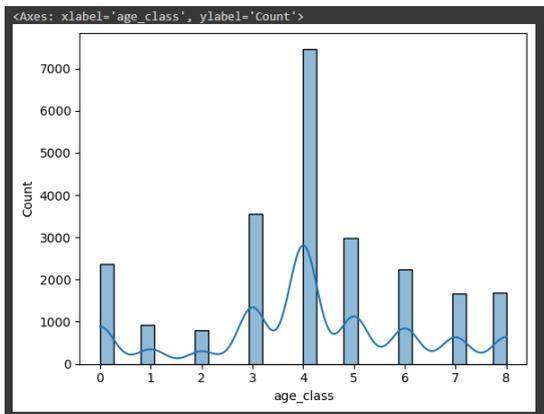
Dengan demikian, dilakukanlah data *preprocessing* untuk mengklasifikasikan hasil yang didapatkan model pretrained ke suatu rentang usia baru yang didefinisikan berdasarkan rentang usia menurut Departemen Kesehatan. Namun, tetap memperhatikan distribusi data di antara rentang-rentang usia tersebut. Adapun rentang usia yang dirumuskan setelah melihat distribusi data adalah sebagai berikut.

- a. masa bayi (usia 0-2 tahun)
- b. masa balita (usia 4-6 tahun)
- c. masa kanak-kanak (usia 8-12 tahun)
- d. masa remaja (usia 15-20 tahun)
- e. masa dewasa awal (usia 26-32 tahun)
- f. masa dewasa akhir (usia 38-43 tahun)
- g. masa lansia awal (usia 48-53 tahun)
- h. masa manula (usia 65 tahun ke atas)

Adapun perbedaan distribusi jumlah data antar kelas dengan jenis pengkategorian menurut Departemen Kesehatan RI dan pengkategorian usulan dapat dilihat pada gambar 8 dan 9 di bawah ini.



Gambar 8 Distribusi kelas rentang usia berdasarkan Departemen Kesehatan RI pada dataset



Gambar 9 Distribusi kelas rentang usia usulan pada dataset

Selanjutnya, setelah dataset dikelompokkan ke dalam rentang usia tersebut, dilakukan proses *shuffling* karena dataset UTK Face cenderung memiliki pola berurutan, seperti usia muda terlebih dahulu baru usia tua. Hal tersebut, tentu dapat mempengaruhi *learning process* dari model *pretrained* karena cenderung mengenali kelompok usia tertentu terlebih dahulu

#### B. Proses *fine tuning* model-model *pretrained*

Proses *fine tune* pada dilakukan pada model *pretrained* VGG16, ResNet50, dan MobileNet. Konfigurasi jumlah data *training* dan *testing* adalah 80% : 20%. Konfigurasi split tersebut dilakukan karena dataset mengalami penurunan jumlah data setelah dilakukan proses *preprocessing*. Konfigurasi selengkapnya terkait jumlah data training dan testing dapat dilihat sebagai berikut.

- Jumlah data *training* = 11.930
- Jumlah data *validation* = 2.386
- Jumlah data *testing* = 597

Selanjutnya, proses *fine tune* dilakukan dengan *hyperparameter* yang sama, yaitu:

- Learning rate* = 0.001
- Batch size* = 64

#### c. *Epoch* = 25

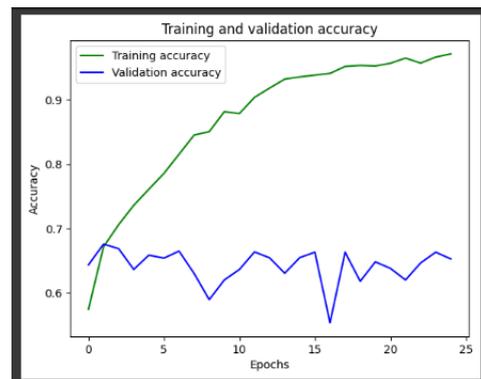
*Epoch* yang digunakan pada proses *fine tuning* kali ini hanya 25 karena keterbatasan daya komputasi yang dimiliki. Ketiga model *pretrained* CNN tersebut menggunakan *output dense layer* sebanyak 4. Hal itu disebabkan butuhnya penyesuaian *output* yang dihasilkan dari *pretrained* model yang telah di *train* sebelumnya dengan *output* yang digunakan pada tugas akhir ini, yaitu klasifikasi ke 8 rentang usia. Adapun spesifikasi dari keempat layer tersebut adalah sebagai berikut.

- Dense(1024, activation='ReLU')
- Dense(1024, activation='ReLU')
- Dense(512, activation='ReLU')
- Dense(8, activation='Softmax')

Penggunaan *dense layer* sebesar 1024 *neuron* pada dua layer tambahan pertama adalah untuk memaksimalkan hasil konvolusi dalam mengenali fitur dari proses-proses sebelumnya. Selanjutnya, penggunaan *dense layer* akan diturunkan secara perlahan sesuai dengan size dari *output layer*. *Output layer* yang digunakan pada proses ini adalah fungsi aktivasi *softmax* dengan 8 kelas *output*.

#### 1) Hasil dan analisis proses *fine tuning* model VGG16

Hasil proses *fine tune* pada model VGG16 memperoleh nilai akurasi pada data validasi sebesar 67,56%. Selain itu, proses training ini juga mengalami *overfitting*. Hal tersebut dapat terjadi karena kurangnya jumlah data *training* yang tersedia. Selengkapnya dapat dilihat pada gambar 10 di bawah ini.

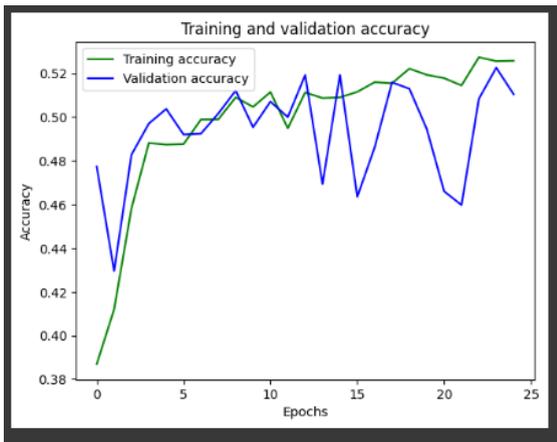


Gambar 10 Grafik nilai akurasi training dan validasi pada VGG16

#### 2) Hasil dan analisis proses *fine tuning* model ResNet50

Hasil proses *fine tune* pada model ResNet50 memperoleh nilai akurasi sebesar 52,25%. Nilai akurasi yang didapatkan belum cukup maksimal karena proses *fine tuning* hanya berlangsung sebanyak 25 epoch, sedangkan ResNet50 memiliki *hidden layer* sebanyak 50 sehingga membutuhkan training dalam jumlah *epoch* yang besar. Hal tersebut diperkuat dengan gambar 11 yang menunjukkan bahwa graf nilai akurasi *training* baru mencapai sekitar angka 52% saat mencapai *epoch* 25. Hal tersebut menandakan bahwasanya bukan tidak mungkin bila model ResNet50 dapat mencapai

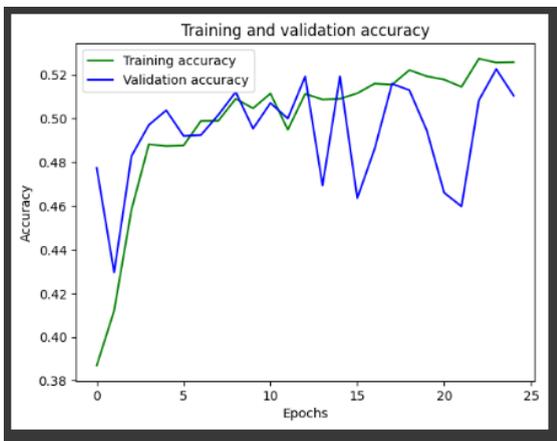
nilai akurasi yang lebih tinggi dari model VGG16 atau MobileNet bilamana mendapatkan jumlah *training* yang tepat.



Gambar 11 Grafik nilai akurasi training dan validasi pada ResNet50

### 3) Hasil dan analisis proses *fine tuning* model MobileNet

Hasil dari proses *fine tuning* model MobileNet hampir sama persis dengan hasil VGG16, yaitu mendapatkan nilai akurasi sebesar 67,22%. Namun, proses *training* pada model MobileNet juga mengalami *overfitting* sama seperti VGG16. Hal ini menandakan bahwasanya model *pretrained* dengan jumlah *layer* sedikit cenderung lebih baik dengan jumlah data *training* yang sedikit dan jumlah *epoch* yang sedikit. Untuk grafik nilai akurasi training dan validasi selengkapnya dapat dilihat pada gambar 12 di bawah ini.



Gambar 12 Grafik nilai akurasi training dan validasi pada MobileNet

### B. Pengujian model *pretrained* dengan data uji

Dari proses eksperimen *fine tuning* ketiga model *pretrained* tersebut, didapatkan bahwa model VGG16 dan MobileNet

memiliki nilai akurasi tertinggi pada data validasi. Oleh karena itu, kedua model tersebut dilakukan pengujian dengan data uji. Perlu dicatat juga bahwasanya pengujian ini dilakukan setelah melakukan proses *preprocessing dataset* dengan mengklasifikasikan ke rentang usia yang dirumuskan pada sub bab IV.A. Hasil pengujian ketiga model *pretrained* yang telah di-*fine tune* terhadap data uji dapat dilihat pada tabel 4 di bawah ini. Pada tabel tersebut terlihat model MobileNet mendapatkan nilai akurasi sebesar 86,17% yang mana lebih tinggi dibandingkan model VGG16 yang mendapatkan nilai akurasi sebesar 71,54%, sedangkan model ResNet mendapatkan nilai akurasi terendah, yaitu hanya 59,8%. Model ResNet mendapatkan nilai akurasi yang rendah karena proses *fine tune* yang dilakukan hanya menggunakan nilai *epoch* yang kecil sehingga belum dapat memaksimalkan *training* pada model ResNet yang memiliki kedalaman *layer* yang cukup tinggi.

Tabel 4 Hasil pengujian setelah data *preprocess* dan *fine tuning* model *pretrained*

Model <i>Pretrained</i>	Nilai akurasi sesudah data <i>preprocessing</i>
VGG16	71,54%
ResNet	59,8%
MobileNet	86,17%

## IX. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian dan eksperimen, didapatkan berbagai kesimpulan sebagai berikut.

1. Berdasarkan eksperimen model *pretrained* di bab IV, didapatkan model MobileNet memperoleh nilai akurasi tertinggi dengan memperoleh nilai kebenaran sebesar 86,17% dari 597 data uji.
2. Pengaplikasian model *pretrained* CNN yang telah di-*fine tune* terlebih dahulu dengan *dataset* UTK Face dapat mempermudah memprediksi rentang usia seseorang.
3. Hasil prediksi rentang usia yang dihasilkan aplikasi *mobile* telah berhasil mengklasifikasikan usia seseorang tepat atau mendekati ke suatu rentang usia.

Adapun saran yang didapatkan atas pengerjaan tugas akhir ini adalah sebagai berikut.

1. Eksplorasi metode penanganan *overfitting* yang disebabkan oleh kurangnya jumlah *training* data pada proses *fine tuning* model *pretrained*, seperti proses Augmentasi data dan metode *K-fold cross validation*.
2. Mencari *server* yang memiliki kekuatan komputasi cukup kuat untuk melakukan proses *fine tune* model-model *pretrained* CNN dengan kedalaman *layer* yang cukup tinggi.

### 3. Penglementasian algoritma *reset allocation memory* pada *client application* untuk support fitur *offline*.

#### REFERENCES

- [1] A. Anand, R. D. Labati, A. Genovese, E. Muñoz, V. Piuri and F. Scotti, "Age estimation based on face images and pre-trained convolutional neural networks," 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 2017, pp. 1-7, doi: 10.1109/SSCI.2017.8285381.
- [2] H. Liu, J. Lu, J. Feng, and J. Zhou, "Group-aware deep feature learning for facial age estimation," Pattern Recognition, vol. 66, no. Supplement C, pp. 82–94, June 2017.
- [3] E. Eiding, R. Enbar, and T. Hassner, "Age and gender estimation of unfiltered faces," IEEE Trans. on Information Forensics and Security, vol. 9, no. 12, pp. 2170–2179, December 2014.
- [4] G. Levi and T. Hassner, "Age and gender classification using Convolutional Neural Networks," in Proc. of the 2015 IEEE Int. Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015, pp. 34–42.
- [5] J. Huang, B. Li, J. Zhu, and J. Chen, "Age classification with Deep Learning face representation," Multimedia Tools and Applications, vol. 76, no. 19, pp. 20 231–20 247, October 2017.
- [6] Omkar M. Parkhi, Andrea Vedaldi and Andrew Zisserman. Deep Face Recognition. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, Proceedings of the British Machine Vision Conference (BMVC), pages 41.1-41.12. BMVA Press, September 2015.
- [7] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems.
- [8] Daniel, V. (2018). Pengenalan Wajah Menggunakan Convolutional Neural Network. Tesis, Institut Teknologi Sepuluh November, Surabaya: Departemen Informatika.
- [9] Zein, A. (2020). Memprediksi Usia dan Jenis Kelamin Menggunakan Convolutional Neural Network. STMIK Eresha, Tangerang Selatan: Program Studi Teknik Informatika.
- [10] Nugroho, Setyo & Harjoko, Agus. (2005). Penerapan Jaringan Syaraf Tiruan untuk Mendeteksi Posisi Wajah Manusia pada Citra Digital.
- [11] Ciresan, D.; Meier, U.; Schmidhuber, J. (2012). "Multi-column deep neural networks for image classification". 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3642–3649.
- [12] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey (2012). "ImageNet Classification with Deep Convolutional Neural Networks". NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada. Archived from the original on 2017-01-10. Retrieved 2017-05-24.
- [13] ELKarazle, K., Raman, V., & Then, P. (2022). Facial age estimation using Machine Learning Techniques: An overview. Big Data and Cognitive Computing, 6(4), 128. <https://doi.org/10.3390/bdcc6040128>.
- [14] Russell, J. (2017, May 25). Google's AlphaGo AI wins three-match series against the World's Best Go player. TechCrunch. Retrieved December 18, 2022, from <https://techcrunch.com/2017/05/24/alphago-beats-planets-best-human-go-player-ke-jie/amp/>
- [15] Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. SN COMPUT. SCI. 2, 420 (2021).
- [16] LeCun, Yann; Bengio, Yoshua; Hinton, Geoffrey (2015). "Deep Learning". Nature. 521 (7553): 436–444.
- [17] Pai, A. (2020, February 20). CNN vs. RNN vs. Ann-analyzing 3 types of neural networks in deep learning. Medium. Retrieved December 18, 2022, from <https://medium.com/analytics-vidhya/cnn-vs-rnn-vs-ann-analyzing-3-types-of-neural-networks-in-deep-learning-f3fa1249589d>
- [18] Mishra, M. (2020, September 2). Convolutional Neural Networks, explained. Medium. Retrieved December 18, 2022, from <https://towardsdatascience.com/convolutional-neural-networks-explained-d9cc5188c4939>
- [19] Saha, S. (2022, November 16). A comprehensive guide to Convolutional Neural Networks-the eli5 way. Medium. Retrieved December 18, 2022, from <https://medium.com/towards-data-science/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [20] Sena, S. (2018, March 19). Pengenalan Deep Learning Part 1 : Neural network. Medium. Retrieved December 18, 2022, from <https://medium.com/@samuelsena/pengenalan-deep-learning-8fbb7d8028ac>
- [21] Khan, S., Rahmani, H., Afaq, S., Shah, A., & Bennamoun, M. (2018). A Guide to Convolutional Neural Networks for Computer Vision. (G. Medioni & S. Dickinson, Eds.). California: Morgan & Claypool Publishers series.
- [22] Radhakrishnan, P. (2017, October 18). What are hyperparameters ? and how to tune the hyperparameters in a deep neural network? Medium. Retrieved December 18, 2022, from <https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a>
- [23] Yağın, O. G. (2021, February 2). 4 pre-trained CNN models to use for computer vision with transfer learning. Medium". <https://towardsdatascience.com/4-pre-trained-cnn-models-to-use-for-computer-vision-with-transfer-learning-885cb1b2dfc>
- [24] Brownlee, J. (2019, September 16). A gentle introduction to transfer learning for Deep learning. MachineLearningMastery.com. <https://machinelearningmastery.com/transfer-learning-for-deep-learning/>
- [25] Rohini, G. (2021, September 23). Everything you need to know about VGG16. Medium. <https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>
- [26] Elhamraoui, Z. (2020, May 16). INCEPTIONRESNETV2 simple introduction. Medium. <https://medium.com/@zahraelhamraoui1997/inceptionresnetv2-simple-introduction-9a2000edcddb6>
- [27] Yan, Chenjing & Lang, Congyan & Wang, Tao & Du, Xuetao & Zhang, Chen. (2014). Age Estimation Based on Convolutional Neural Network. Lect. Notes Comput. Sci.. 8879. 211-220. 10.1007/978-3-319-13168-9\_22.
- [28] Zhang, Z.; Song, Y.; Qi, H. Age Progression/Regression by Conditional Adversarial Autoencoder. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
- [29] Aswini K. Samantaray, Amol D. (2021). "Rahulkar, in Image Processing for Automated Diagnosis of Cardiac Diseases".
- [30] Lin C-T, Li D-L, Lai J-H, Han M-F, Chang J-Y. (2012). "Automatic Age Estimation System for Face Images. International Journal of Advanced Robotic Systems". doi:10.5772/52862.
- [31] Pietikäinen, M. (2010). Local binary patterns. Scholarpedia. [http://www.scholarpedia.org/article/Local\\_Binary\\_Patterns](http://www.scholarpedia.org/article/Local_Binary_Patterns)
- [32] susanqq. (2017). Utkface. <https://susanqq.github.io/UTKFace/>
- [33] Singh, N. (2020). MobileNet V1 architecture. OpenGenus IQ: Computing Expertise & Legacy. <https://iq.opengenus.org/mobilenet-v1-architecture/>
- [34] PA, S. (2020). An overview on MobileNet: An Efficient Mobile Vision CNN. Medium. <https://medium.com/@godeep48/an-overview-on-mobilenet-an-efficient-mobile-vision-cnn-f301141db94d>
- [35] Mukherjee, S. (2022). The annotated resnet-50. Medium. <https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758>
- [36] Kaushik, A. (2020). Understanding Resnet50 architecture. OpenGenus IQ: Computing Expertise & Legacy. <https://iq.opengenus.org/resnet50-architecture/>
- [37] Sarkar, A. (2021). Understanding depthwise separable convolutions and the efficiency of MobileNets. Medium. <https://towardsdatascience.com/understanding-depthwise-separable-convolutions-and-the-efficiency-of-mobilenets-6de3d6b62503>

