PENDETEKSIAN CITRA DEEPFAKE WAJAH DI SMARTPHONE MENGGUNAKAN MOBILENETV3-SMALL DAN LBP

Matthew Patrick [1)] Chairisni Lubis [2)] Agus Budi Dharmawan [3)]

[1) 2) 3)] Teknik Informatika, FTI, Universitas Tarumanaraga
Jl. Letjen S Parman no 1, Jakarta 11440 Indonesia
email : matthew.535200018@stu.untar.ac.id[1)], chairisnil@fti.untar.ac.id[2)], agusd@fti.untar.ac.id[3)]

ABSTRACT

Citra *DeepFake*, yang dihasilkan menggunakan algoritma kecerdasan buatan seperti *GAN*, menjadi ancaman besar di dalam dunia digital saat ini karena mudah disebar dan dapat menyebabkan misinformasi. Studi ini berfokus pada desain dan penerapan sistem yang menggunakan *model-model MobileNetV3-Small*. Dilakukan perbandingan atas dua varian *model*, satu dengan *input* citra *RGB* dan yang lain dengan *input* citra *grayscale* hasil dari proses *Local Binary Pattern*. Hasilnya menunjukkan bahwa *model MobileNetV3-Small* dengan *input RGB* mempunyai tingkat akurasi yang lebih tinggi sebesar 88.23%, melebihi *model* yang menggunakan *input* citra *grayscale* hasil dari *proses Local Binary Pattern*, yang mencapai tingkat akurasi sebesar 72.63%. *Model MobileNetV3-Small* dengan *input RGB* yang menunjukkan kinerja superior, diintegrasikan ke dalam aplikasi *smartphone* untuk pendeteksian citra wajah *DeepFake* yang efisien.

Key words

CNN, DeepFake, Local Binary Pattern, MobileNetV3-Small, Smartphone

1. Pendahuluan

DeepFake adalah salah satu jenis *Artificial Intelligence* yang digunakan untuk membuat citra palsu yang sulit dibedakan dengan citra yang asli [1] . *AI Algorithm Generative Adversarial Networks (GANs)* biasa digunakan untuk menciptakan citra palsu karena hasil dari citra tersebut sulit dibedakan dengan yang aslinya [2].

Citra *DeepFake* mempunyai potensi bahaya yang serius di internet karena citra *DeepFake* dapat digunakan untuk menyebar informasi palsu yang dapat merusak reputasi orang yang wajahnya digunakan di citra tersebut. Citra *DeepFake* juga dapat dibuat dengan mudah di jaman sekarang yang dapat menyebabkan peningkatan dalam penggunaan citra *DeepFake* dalam penyebaran *hoax*.

Pada penelitian-penelitian sebelumnya, citra *DeepFake* dapat dikenali menggunakan keluarga *model EfficientNet* [3], dan lainnya. Tetapi, di studi ini akan digunakan *model MobileNetV3-Small* karena *model* tersebut berasal dari keluarga *MobileNets* yang diciptakan untuk dijalankan di *low powered devices* [4].

Tujuan dari studi ini adalah untuk melakukan perbandingan atas dua *variant model* yang menggunakan *model MobileNetV3-Small*. Perbedaan dari kedua *model* tersebut adalah jenis *input* yang digunakan. *Model* pertama akan menggunakan *input RGB image* dan yang kedua hanya akan menggunakan *grayscale image* yang di *generate* menggunakan proses *Local Binary Pattern*. Kedua *model* akan dibandingkan dan *model* terbaik akan diimplementasikan menjadi *smartphone application* yang dapat berguna sebagai aplikasi *early detection* yang dapat mendeteksi citra *deepfake* wajah.

2. Dasar Teori

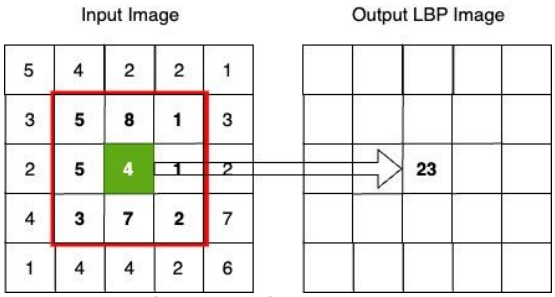
2.1 *DeepFake*

DeepFake merupakan media palsu yang diciptakan menggunakan *Artificial Intelligence* yang dapat berupa bentuk citra, video, maupun audio [1]. *DeepFake* dapat membuat citra atau video palsu di mana dengan menggunakan *AI*, sebuah wajah orang dari suatu media dapat dipindahkan ke media lainnya. Terdapat sisi positif dari teknologi ini. Contohnya dengan teknologi *DeepFake* dapat dibuat *movies* dengan *actor/actress* yang sudah meninggal. Tetapi terdapat juga sisi *negative* dari teknologi ini. Contohnya, teknologi *DeepFake* pernah digunakan untuk memalsukan kejadian yang terjadi, penyebaran *fake news*, perlakuan *blackmail*, dan lainnya [5].

2.2 *Local Binary Pattern*

Local Binary Pattern adalah salah satu teknik yang biasa digunakan untuk mengekstrak fitur dari citra [6]. *Local Binary* Pattern melakukan ekstraksi fitur dengan cara melakukan perbandingan nilai pada *pixel* di dalam lingkaran kecil yang dilakukan di setiap *pixel* di sebuah citra. Dalam sebuah citra, *Local Binary Pattern* dapat melakukan identifikasi pola intensitas dengan melakukan konversi nilai *pixel* ke dalam *binary code* dari

perbandingannya dengan nilai di *pixel* pusat. *Binary code* yang dihasilkan akan dikonversi menjadi nilai hasil *Local Binary Pattern* dan akan digunakan sebagai *texture representation* dari citra tersebut. Berikut merupakan contoh representasi *input image* yang dikonversi menjadi *LBP image*.



Gambar 1 - *Local Binary Pattern*

2.3 *Convolutional Neural Network*

Convolutional Neural Network adalah *deep neural network* yang digunakan untuk mengenali informasi yang terdapat di dalam sebuah citra yang dapat digunakan untuk melakukan analisis [7]. *Convolutional Neural Network* dapat melakukan identifikasi *object* dengan akurat karena *Convolutional Neural Network* dapat mengumpulkan *spatial feature* yang merupakan susunan dan hubungan antara *pixel-pixel* dari sebuah citra. Itulah kenapa *Convolutional Neural Network* cocok digunakan untuk memproses citra [8].

2.4 *MobileNetV3*

MobileNetV3 adalah sebuah *model neural network* berbasis *CNN* yang dikembangkan untuk melakukan tugas pengenalan pada citra yang diciptakan oleh Andrew Howard, Mark Sandler, Grace Chu, dan lainnya [9]. *MobileNetV3* merupakan generasi terbaru yang berasal dari keluarga *MobileNets* yang diciptakan untuk dapat berjalan dengan baik di *low-powered devices* seperti *smartphones*, dan *IoT d*evices [4]. *MobileNetV3* dikembangkan menjadi dua jenis *variant*. Yang pertama adalah *MobileNetV3-Large* dan yang kedua adalah *MobileNetV3-Small*. Perbedaan dari kedua *variant* tersebut adalah *variant Large* mempunyai *layer* yang lebih banyak dan kompleks. *Variant* tersebut juga mempunyai *parameter* yang lebih banyak. Artinya, akan diperlukan daya komputasi yang lebih besar untuk menggunakan *variant Large* dibandingkan dengan *variant Small*. Berikut merupakan spesifikasi dari kedua jenis *variant MobileNetV3*.

Input	Operator	exp size	#*out*	SE	NL	*s*
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Gambar 2 - Spesifikasi *Variant Large*

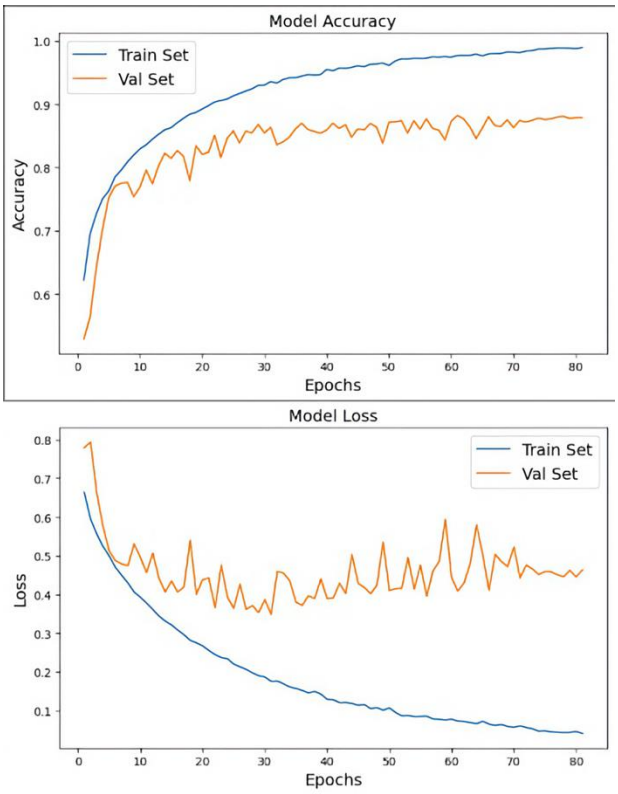
Input	Operator	exp size	#*out*	SE	NL	*s*
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

Gambar 3 - Spesifikasi *Variant Small*

3. Hasil Percobaan

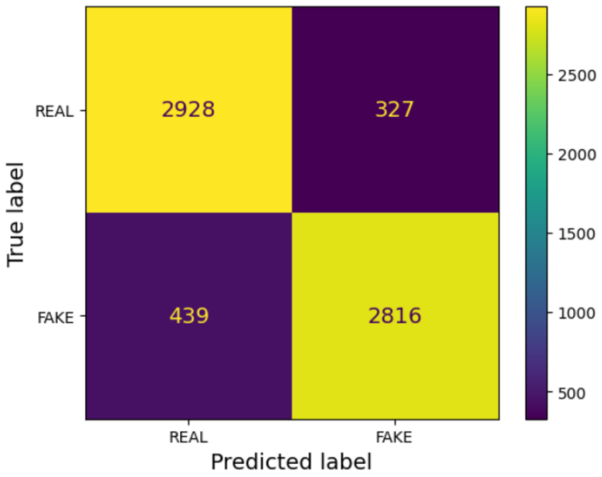
3.1 *Model* Dengan *Input* Citra *RGB*

Model ini merupakan *model MobileNetV3-Small* yang menggunakan *input* citra *RGB*. Dalam melakukan *training* di *model* ini, digunakan *Early Stop feature* dengan *patience* 20 dan *ReduceLROnPlateau feature* dengan *patience* 10, *Initial LR* 0.001 dan *Minimum LR* 0.00001. *Early Stop* digunakan untuk memastikan *model* akan berhenti dalam proses *training* jika *model* sudah tidak dapat meningkat lagi akurasinya, dan *ReduceLROnPlateau* digunakan untuk melakukan *fine-tuning* yang dapat meningkatkan kinerja *model*. Cara kerjanya adalah dengan mengurangi *learning rate training* saat *model* sudah mendekati atau saat *model* sudah berhenti dalam peningkatan kinerjanya.



Gambar 4 - *Graph Training Model Input RGB*

Graph training di atas menunjukkan bahwa *model* dapat mempelajari data set *deepfake* yang digunakan, dan *model* tersebut dapat melakukan klasifikasi terhadap data set yang digunakan.



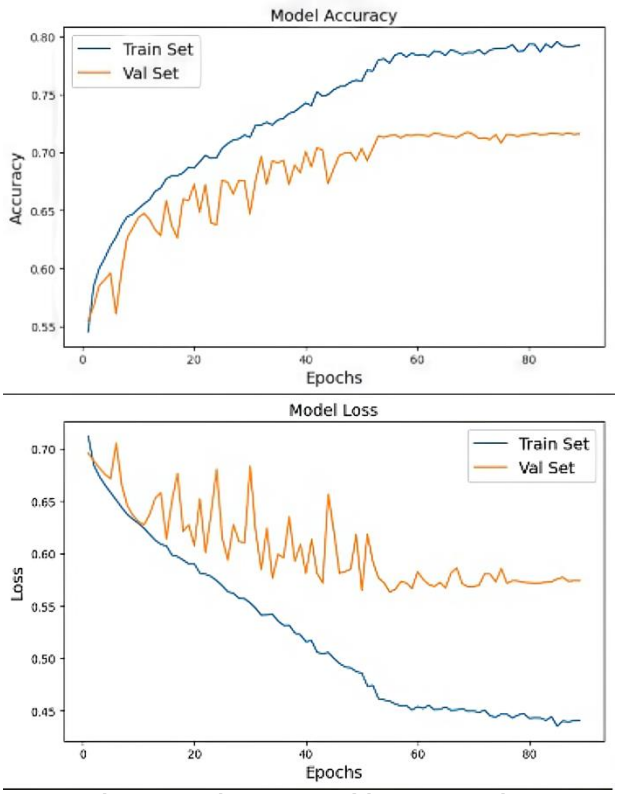
Gambar 5 - *Graph Confusion Matrix Model Input RGB*

Confusion matrix di atas menunjukkan bahwa pengujian *model* terhadap data set khusus *test* menghasilkan *model* yang mempunyai akurasi sebesar 88.23% yang mempunyai arti bahwa *model* dapat melakukan prediksi benar dengan kemampuan yang sangat baik. Didapatkan juga *Precision* sebesar 89.6% yang mempunyai arti bahwa saat *model* mendeteksi sebuah citra adalah *deepfake*, kemungkinan besar prediksi tersebut merupakan prediksi yang benar. Didapatkan juga *Recall* sebesar 86.51% yang mempunyai arti bahwa *model* dapat melakukan pendeteksian terhadap sebagian besar citra *deepfake* di data set tes yang digunakan. Terakhir, didapatkan *F1 Score* sebesar 88.03% yang merupakan skor hasil dari nilai *precision* dan *recall* sebagai skor evaluasi yang seimbang.

Dari *confusion matrix* di atas, terdapat nilai *False Positive* sebesar 327 yang merupakan nilai yang lebih rendah dibandingkan dengan nilai *False Negative* yang merupakan 439. Ini merupakan hal yang kurang optimal karena dalam pendeteksian *DeepFake*, jika nilai *False Negative* lebih tinggi dibandingkan dengan nilai *False Positive*. Artinya *model* akan mempunyai kemungkinan yang lebih tinggi dalam mendeteksi citra *DeepFake* sebagai citra *real* yang dapat memberikan efek *a false sense of security* karena pengguna akan merasa aman bahwa citra tersebut di prediksi sebagai citra asli walaupun sebenarnya merupakan citra *deepfake*. Hal ini tidak menjadi masalah jika nilai *False Positive* lebih tinggi dari nilai *False Negative* karena artinya jika *model* salah memprediksi citra *real* sebagai citra *DeepFake*, pengguna akan lebih berhati-hati dalam mempercayai citra yang diuji tersebut.

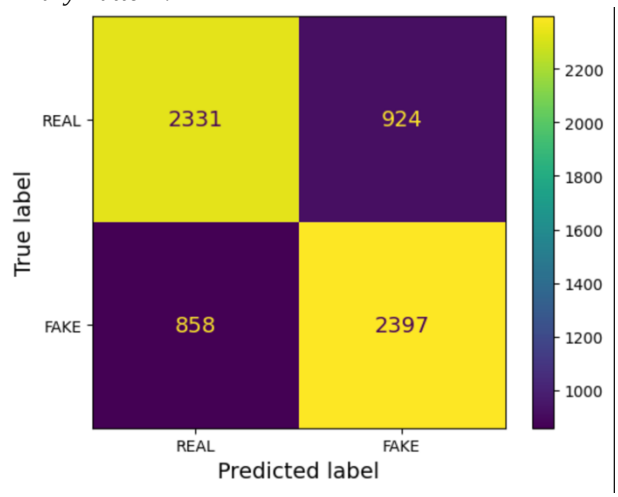
3.2 *Model* Dengan *Input* Citra *Grayscale LBP*

Model ini merupakan *model* yang sama dengan *model* sebelumnya. Digunakan juga *Early Stopping feature*, dan *ReduceLROnPlateau feature* dengan *parameter* yang sama. Hal ini dilakukan supaya dapat diperoleh *model* yang dapat dilakukan perbandingan secara adil. Alasan utama tidaknya digunakan *static value* buat jumlah *epochs* dan *learning rate* adalah proses *training* dapat berjalan dengan kecepatan yang berbeda dan bervariasi tergantung dari jenis arsitektur model, dan juga jenis dari data set yang di kasus ini merupakan citra *grayscale* yang dihasilkan oleh proses *Local Binary Pattern*. Artinya *model* dapat mempunyai kemampuan dan kecepatan mempelajari data yang berbeda dengan *model* yang pertama. Itulah kenapa dengan menggunakan fitur dan *parameter* di *Early Stop* dan *ReduceLROnPlateau* yang sama, dapat diperoleh *model* yang dapat dibandingkan dengan standar yang sama tetapi mempunyai kinerja yang paling baik.



Gambar 6 - *Graph Training Model Input Grayscale LBP*

Graph di atas menunjukkan bahwa *model* dapat menggunakan *input* citra *grayscale* yang dihasilkan oleh *Local Binary Pattern* pada data set *training*. Tetapi, terlihat bahwa *model* ini mempunyai akurasi yang jauh lebih rendah dibandingkan dengan *model* yang menggunakan citra *input RGB*. Ini mungkin terjadi karena citra *grayscale* yang dihasilkan oleh *Local Binary Pattern* mempunyai informasi yang lebih sedikit dibandingkan dengan citra *RGB* yang tidak dilakukan proses *Local Binary Pattern*.



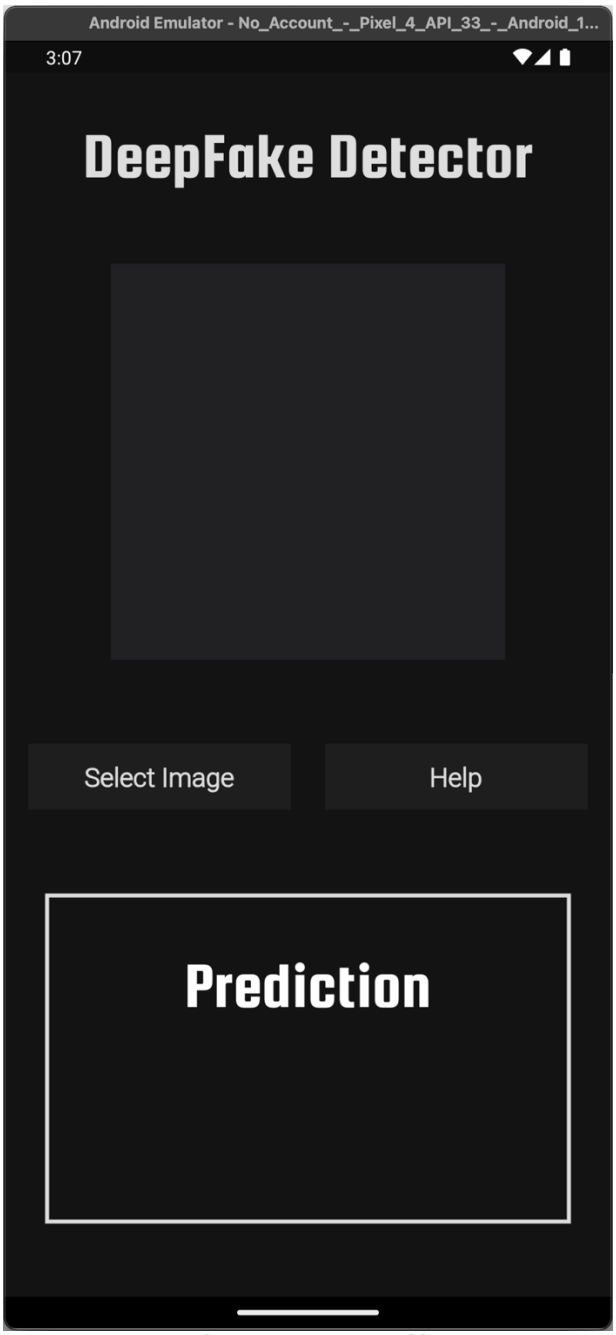
Gambar 7 - *Graph Confusion Matrix Model Input Grayscale LBP*

Confusion Matrix di atas menunjukkan bahwa pengujian *model* terhadap data set khusus *test* menghasilkan *model* yang mempunyai akurasi sebesar 72.63% yang mempunyai arti bahwa *model* dapat melakukan prediksi benar dengan kemampuan yang cukup baik. Didapatkan juga *Precision* sebesar 72.18% yang mempunyai arti bahwa saat *model* mendeteksi sebuah citra adalah *deepfake*, terdapat kemungkinan yang cukup baik bahwa prediksi tersebut merupakan prediksi yang benar. Didapatkan juga *Recall* sebesar 73.64% yang mempunyai arti bahwa *model* dapat melakukan pendeteksian terhadap cukup banyak citra *deepfake* di data set tes yang digunakan. Terakhir, didapatkan juga *F1 Score* sebesar 72.90% yang merupakan skor yang cukup baik yang merupakan skor hasil dari nilai *precision* dan *recall* sebagai skor evaluasi yang seimbang.

Dari *confusion matrix* di atas, terhadap nilai *False Positive* yang lebih tinggi dari nilai *False Negative*. Artinya, pengguna akan mendapatkan hasil prediksi salah lebih banyak yang menunjukkan bahwa citra *real* yang dites merupakan citra *deepfake* dibandingkan sebaliknya. Hal ini lebih optimal di konteks *DeepFake detection* karena pengguna akan hati-hati dan pengguna akan melakukan verifikasi lebih lanjut atas citra yang telah dites.

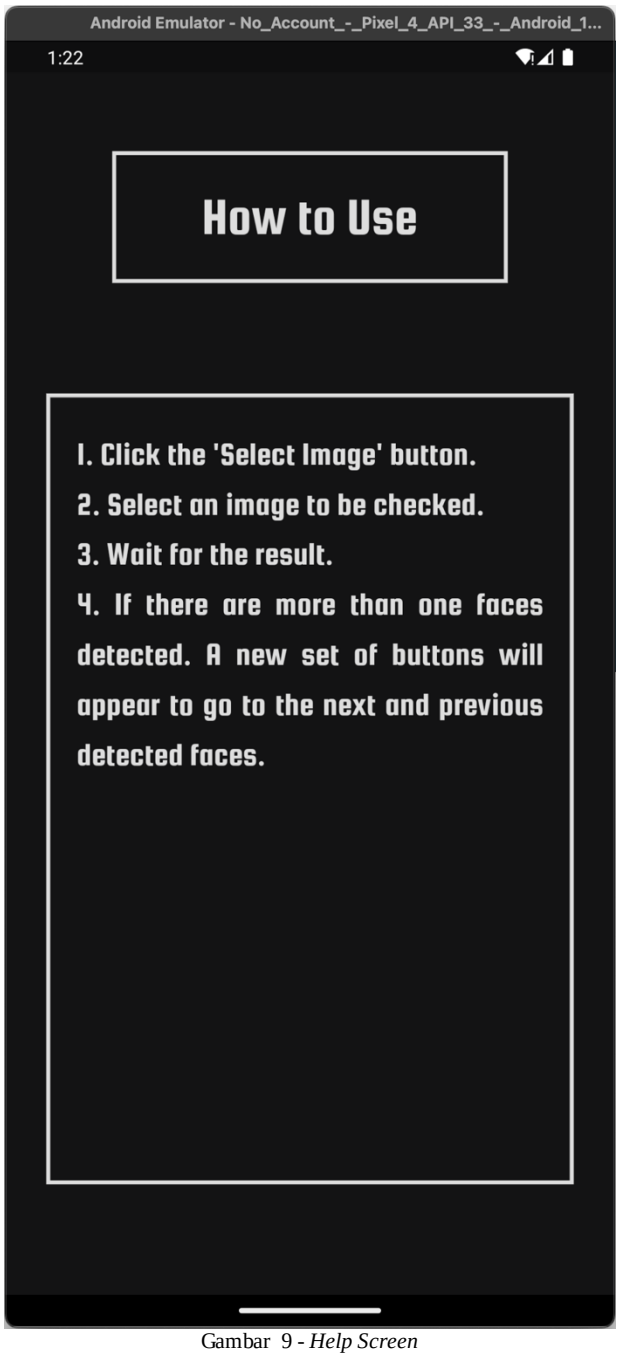
3.3 Aplikasi *Smartphone*

Dalam pembuatan aplikasi, digunakan *Flutter* dengan *package Tensorflow-Lite* supaya dapat digunakan *model* terbaik yang merupakan *model* dengan *input RGB* untuk membuat aplikasi pendeteksi *deepfake* wajah yang akurat. Karena citra yang dites merupakan citra wajah yang merupakan citra yang jenisnya sensitif. Aplikasi akan berjalan secara *local* atau *offline* biar terjaga keamanan datanya. Berikut merupakan beberapa *screenshot* dari aplikasi yang dirancang.



Gambar 8 - *GUI* Utama Aplikasi

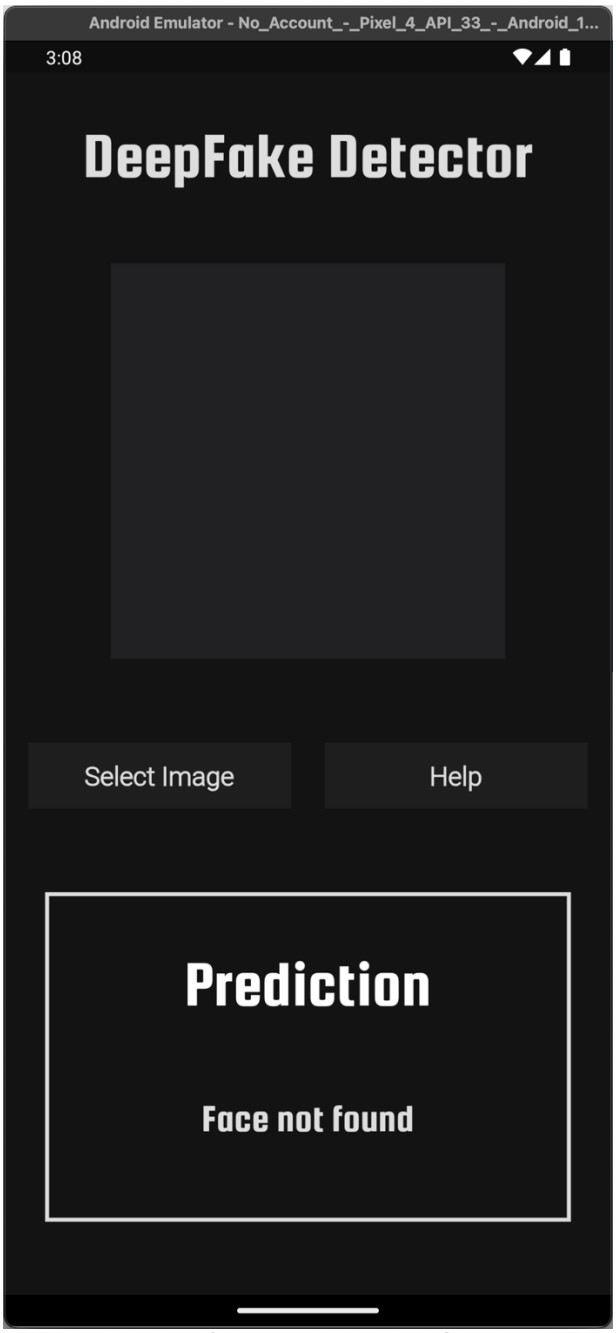
Di atas dapat terlihat *screen* utama yang akan ditunjukkan kepada pengguna aplikasi. Di *screen* utama akan ditunjukkan kotak yang akan menunjukkan wajah yang terdeteksi. Terdapat juga dua *buttons* yang mempunyai fungsi untuk memilih citra yang akan di tes, dan *button Help* digunakan untuk menunjukkan *screen Help* yang dapat membantu pengguna untuk dapat menggunakan aplikasi ini dengan baik.



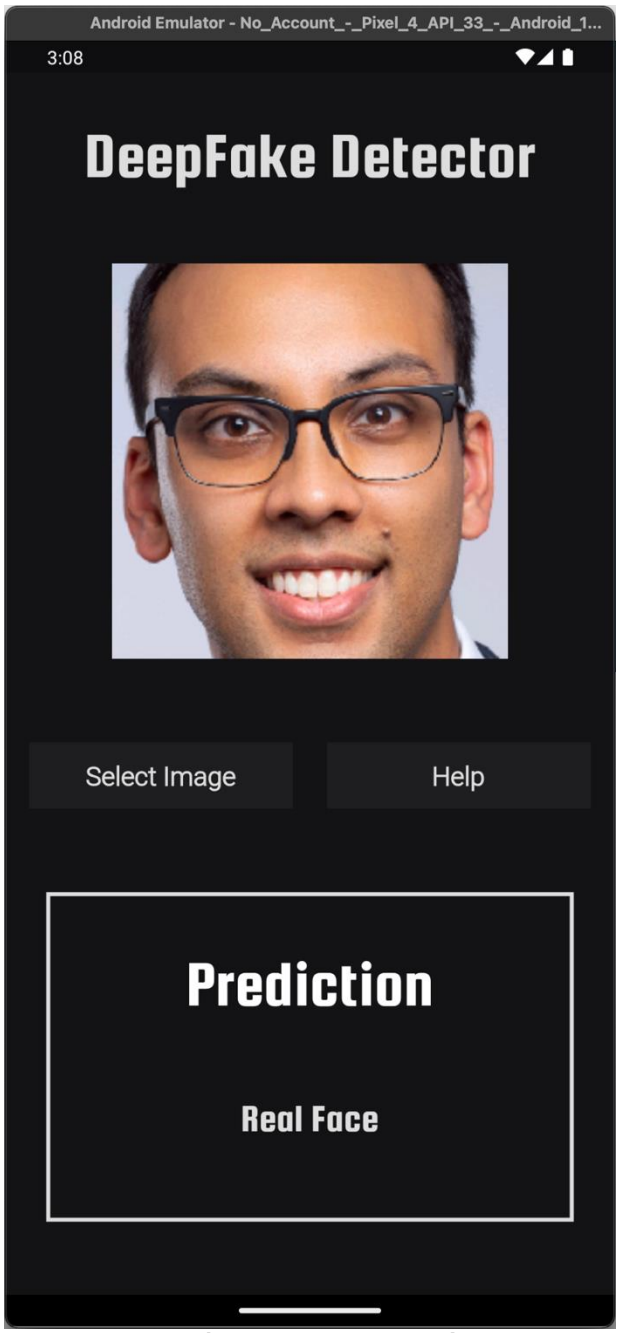
Gambar 9 - *Help Screen*

Di gambar 9, dapat terlihat *Help Screen* yang akan ditunjukkan ke pengguna aplikasi saat pengguna memilih *button "Help"* di *screen* utama.

Untuk memulai pengujian citra. Pengguna akan memilih *button "Select Image".* Setelah pengguna memilih citra yang akan diuji, aplikasi akan memproses citra yang dipilih dengan melakukan *face recognition, cropping and resizing faces,* dan akhirnya akan dilakukan prediksi terhadap wajah yang terdapat di citra yang dipilih. Jika saat dilakukan pengujian tidak dapat terdeteksi wajah, akan ditunjukkan *screen* seperti berikut.



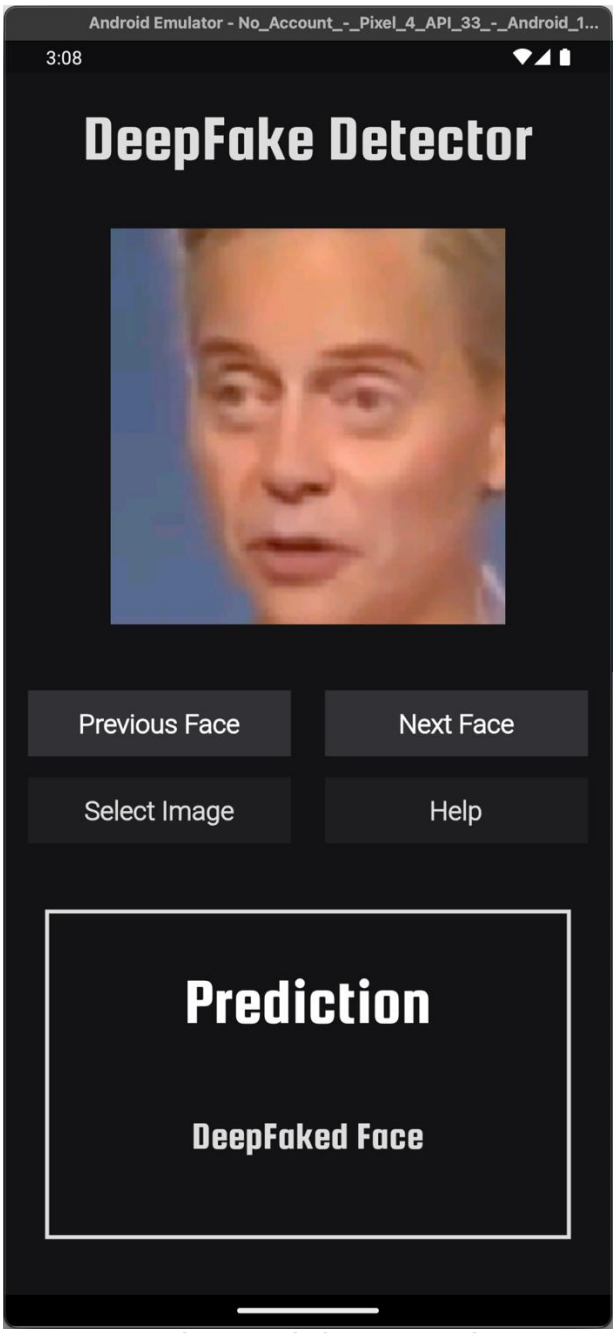
Gambar 10 - *No Face Detected*



Gambar 11 - *One Face Detected*

Di gambar 10 dapat terlihat kalimat "*Face not found*" di *box* prediksi yang artinya adalah tidak ada wajah yang terdeteksi di citra yang dipilih. Jika terdapat wajah yang terdeteksi, aplikasi akan menunjukkan wajah yang dites dengan hasil dari prediksi *model* tersebut

Di gambar 11 dapat terlihat wajah yang diuji dan hasil dari pengujian tersebut. Jika dalam citra yang dipilih terdapat lebih dari satu wajah. Aplikasi akan menunjukkan dua *buttons* baru supaya pengguna dapat memilih wajah yang ingin diuji.



Gambar 12 - *Multiple Faces Detected*

Seperti yang terlihat di gambar 12, saat terdapat lebih dari satu wajah yang terdeteksi. Aplikasi akan menunjukkan *button "Previous Face"* dan *"Next Face"* yang dapat digunakan pengguna untuk memilih wajah yang ingin diuji.

4. Kesimpulan

Dari hasil pengujian yang dilakukan dalam rancangan sistem dan aplikasi *DeepFake Detector* menggunakan dua *model* berbasis *MobileNetV3-Small*. Sistem yang dirancang mampu membedakan citra *real* dan citra *fake*. Terdapat juga beberapa kesimpulan seperti:

1. *Model MobileNetV3-Small* dengan citra *input RGB* dapat melakukan pendeteksian citra *DeepFake* wajah dengan tingkat akurasi sebesar 88.23%, dan *precision* sebesar 89.6% pada pengujian terhadap data set *test* yang berasal dari data set yang sudah dipisahkan di tahap awal.

2. *Model* dengan *input RGB* menghasilkan *model* dengan kinerja yang lebih baik dibandingkan dengan *model* yang hanya menggunakan citra *grayscale* hasil dari proses *Local Binary Pattern*. Hal ini mungkin terjadi karena beberapa alasan, seperti perbedaannya jumlah *channel* dalam *input* (3 *channels RGB* dibandingkan dengan 1 *channel grayscale*), dan juga karena proses *Local Binary Pattern* yang mungkin tidak dapat melakukan ekstraksi fitur penting dengan baik.

Setelah dilakukan perancangan aplikasi dan *model-model* tersebut. Saran yang dapat diberikan adalah untuk menggunakan data set dengan jumlah yang lebih banyak dengan kualitas yang jauh lebih baik dan beragam untuk mendapatkan tingkat akurasi yang lebih baik.

REFERENSI

[1] A. Tirta, "Apa Itu Deepfake Dan Cara Mendeteksinya?," bpti.uhamka.ac.id, 9 July 2022. [Online]. Available: https://bpti.uhamka.ac.id/sharing/apa-itu-deepfake-dan-cara-mendeteksinya/. [Diakses 30 August 2023].

[2] S. ADEE, "What Are Deepfakes and How Are They Created?," IEEE Spectrum, 29 April 2020. [Online]. Available: https://spectrum.ieee.org/what-is-deepfake. [Diakses 10 September 2023].

[3] A. A. Pokroy dan A. D. Egorov, "EfficientNets for deepfake detection: Comparison of pretrained models," *2021 IEEE conference of russian young researchers in electrical and electronic engineering (ElConRus)*, pp. 598--600, 2021.

[4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto dan H. Adam, *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*, 2017.

[5] M. Albahar dan J. Almalki, "Deepfakes: Threats and countermeasures systematic review," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 22, p. 3242–3250, 2019.

[6] E. Prakasa, "Texture feature extraction by using local binary pattern," *INKOM Journal*, vol. 9, no. 2, pp. 45-48, 2016.

[7] P. A. Nugroho, I. Fenriana dan R. Arijanto, "IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA EKSPRESI MANUSIA," *ALGOR*, vol. 2, no. 1, pp. 12-20, 2020.

[8] T. T. Platform, "CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning," 30

April 2021. [Online]. Available: https://www.thetechplatform.com/post/cnn-vs-rnn-vs-ann-analyzing-3-types-of-neural-networks-in-deep-learning. [Diakses 9 October 2023].

[9] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le dan H. Adam, *Searching for MobileNetV3*, 2019.

Matthew Patrick, merupakan mahasiswa angkatan 2020 semester akhir yang berasal dari program studi Teknik Informatika, di Fakultas Teknologi Informasi Universitas Tarumanagara.

Dra. Chairisni Lubis, M.Kom., memperoleh gelar Dra. Dan gelar M.Kom dari Universitas Indonesia. Saat ini sebagai Dosen program studi Teknik Informatika, di Universitas Tarumanagara.

Agus Budi Dharmawan S.kom, M.T., M.sc., memperoleh gelar M.T dari ITS Surabaya. Kemudian memperoleh gelar M.Sc dari Elektronic Engenering FH Darmstad. Saat ini sebagai Dosen program studi Teknik Informatika, di Universitas Tarumanagara.