Deteksi Serangan *Spoofing* Wajah Menggunakan *Convolutional Neural Network*

http://dx.doi.org/10.28932/jutisi.v7i3.4001

Riwayat Artikel
Received: 21 September 2021 | *Final Revision*: 25 November 2021 | *Accepted*: 26 November 2021

Raden Budiarto Hadiprakoso✉ #1, I Komang Setia Buana*2

Rekayasa Kriptografi, Politeknik Siber dan Sandi Negara
Jl. H. USA, Ciseeng, Bogor
1.raden.budiarto@poltekssn.ac.id
2 komang.setia@poltekssn.ac.id

Abstract — **Facial recognition-based biometric authentication is increasingly frequently employed. However, a facial recognition system should not only recognize an individual's face, but it should also be capable of detecting spoofing attempts using printed faces or digital photographs. There are now various methods for detecting spoofing, including blinking, lip movement, and head tilt detection. However, this approach has limitations when dealing with dynamic video spoofing assaults. On the other hand, these types of motion detection systems can diminish user comfort. As a result, this article presents a method for identifying facial spoofing attacks through Convolutional Neural Networks (CNN). The anti-spoofing technique is intended to be used in conjunction with deep learning models without using extra tools or equipment. The CNN classification *dataset* used in this study was obtained from the NUAA photo imposter and CASIA v2. The collection contains numerous examples of facial spoofing, including those created with posters, masks, and smartphones. In the pre-processing stage, image augmentation is carried out with brightness adjustments and other filters so that the model adapts to various environmental conditions. We evaluate the number of epochs, optimizer types, and the learning rate during the testing process. The test results show that the proposed model achieves an accuracy value of 91.23% and an F1 score of 92.01%.**

Keywords— **convolution neural network; deep learning; face spoofing detection; face recognition.**

I. PENDAHULUAN

Biometrik merupakan salah satu metode autentikasi yang mulai populer digunakan pada belakangan ini. Salah satu penerapannya adalah teknologi pengenalan wajah yang akhir-akhir ini mulai banyak digunakan karena faktor kemudahan dan kenyamanan. Berbagai macam gawai seperti *smartphone*, tablet, dan laptop sudah mulai menerapkan teknologi pengenalan wajah ini untuk otentikasi. Program pengenalan wajah ini beroperasi dengan cara menangkap gambar wajah seseorang melalui kamera, kemudian wajah tersebut diproses dengan algoritma tertentu untuk memastikan apakah wajah tersebut sesuai dengan *database* atau tidak [1]. Bagaimana pun, ada kelemahan dalam strategi pengenalan wajah, yang dikenal dengan serangan *spoofing*. Mengandalkan sebuah sistem pengenalan wajah tidak akan dapat membedakan antara wajah sebenarnya dan serangan *spoofing* seperti penggunaan topeng wajah, video, atau foto. Dengan demikian, kelemahan ini menciptakan celah bagi seseorang untuk menipu perangkat. Terlebih wajah seseorang jauh lebih mudah didapat dibandingkan biometrik lainnya seperti sidik jari. Wajah seseorang bisa dengan mudah didapatkan melalui foto profil seseorang di media sosial [2].

Serangan *spoofing* wajah dapat dikategorikan menjadi serangan statis dan serangan dinamis. Serangan statis berarti serangan yang bersifat diam dan tidak bergerak. Sebaliknya serangan dinamis berarti serangan wajah *spoofing* yang dapat bergerak. Serangan statis dapat dibagi lagi menjadi serangan statis 2 dimensi (2D) maupun 3 Dimensi (3D). Serangan *spoofing* statis 2D dapat menggunakan foto wajah seseorang. Serangan 3D statis dapat menggunakan cetakan wajah, kosmetik wajah atau topeng wajah. Sementara itu serangan dinamis pada umumnya menggunakan pemutaran video atau beberapa foto yang dianimasikan [3].

Serangan *spoofing* wajah dapat dilakukan dengan cara menampilkan wajah menggunakan gadget seperti menggunakan foto atau video pada *smartphone* atau tablet. Upaya serangan seperti ini menciptakan tekstur wajah berkualitas rendah dan mudah dideteksi dengan menilai kualitas gambar dan varian HSV (*Hue Saturation Value*). Reproduksi warna (gamut) media layar, seperti video atau foto, mungkin akan sangat terbatas dibandingkan dengan wajah aslinya [4]. Selain itu, wajah yang

direpresentasikan dapat berupa variasi warna lokal. Gamut warna tergantung pada media tampilan, dan variasi kroma piksel terdekat dapat dijelaskan dengan memeriksa fitur warna saluran kroma. Juga dapat diperiksa model warna mana yang menyajikan representasi tekstur mikro paling berharga dengan mengekstraksi informasi LBP (*Local Binary Pattern*) dari berbagai *channel* warna. Cara ini diperlukan untuk menyelidiki *spoofing* di daerah dengan pencahayaan yang baik, bagaimanapun pada tempat dengan keadaan cahaya yang minim akan mempengaruhi akurasi secara signifikan [5].

Gambar wajah asli dan palsu memiliki pola tekstur yang berbeda. Hal ini terjadi karena proses rekonstruksi wajah dari foto kamera mengakibatkan penurunan kualitas ekspresi wajah dan kesenjangan antara reflektifitas [6]. Beberapa penelitian sebelumnya telah mencoba menangkap perbedaan tersebut menggunakan karakteristik tekstur warna yang direkayasa, seperti variasi RGB (*Red Green Blue*) atau LBP [7]. Penelitian serupa lainnya juga telah menggunakan algoritma klasifikasi seperti *Support Vector Machine* (SVM) atau *K-Nearest Neighbor* (K-NN) [8] untuk membedakan wajah asli dengan wajah *spoof*. Adapun kelemahan sistem analisis tekstur ini adalah ketergantungan pada kondisi cahaya ruangan. Pada kondisi ruangan tertentu seperti ruangan gelap, tekstur wajah awal yang menggunakan tiruan akan sulit dibedakan.

Convolutional neural network (CNN) merupakan salah satu arsitektur jaringan saraf tiruan yang paling umum diterapkan untuk menganalisis citra visual. CNN dapat solusi tambahan yang dapat membantu *anti-spoofing* [9]. Pada dasarnya mungkin melatih CNN untuk mengenali mana foto asli dan mana yang palsu namun terdapat satu masalah yakni tidak ada serangkaian fitur yang konsisten yang akan dilihat dan dipahami oleh CNN. Seluruh model ini mengandalkan asumsi bahwa sistem akan mendeteksi apa yang disediakan pada *dataset*. Oleh karena itu, penting untuk memanfaatkan kombinasi metode pendeteksian tanda-tanda kehidupan, seperti kedipan atau gerakan bibir dengan metode analisis CNN. Pada penelitian ini cakupannya dibatasi dengan hanya menerapkan deteksi kedipan dan deteksi gerakan bibir untuk deteksi keaktifan wajah karena kedua tanda ini adalah yang paling umum dan mudah dideteksi.

Berangkat dari permasalahan yang telah dijabarkan, makalah ini mengusulkan metode deteksi CNN untuk membedakan antara wajah *spoof* dan asli. Konsep yang paling penting di sini adalah pendekatan ini lebih tahan terhadap perubahan kondisi pencahayaan dan berbagai metode serangan *spoof*. Kontribusi signifikan dari makalah ini antara lain: (1) Pendekatan yang diusulkan akurat dan nyaman karena menggunakan CNN dan *deep transfer learning* untuk mempelajari tanda-tanda karakteristik wajah asli dan tiruan tanpa perlu melakukan gerakan tambahan. (2) Model yang dibuat pada makalah ini dapat diimplementasikan tanpa memerlukan peralatan perangkat keras atau alat tambahan. (3) Skema anti-*spoofing* wajah yang diusulkan dapat menangani serangan *spoofing* yang berbeda (foto, video, atau topeng) dalam skenario dalam ruangan atau luar ruangan serta baik dicetak maupun digital.

II. METODE

Selama ini telah dikembangkan berbagai pendekatan untuk deteksi *spoofing* wajah. Deteksi kedipan mata adalah salah satu pendekatan deteksi keaktifan wajah yang sangat presisi. Berkedip adalah cara mudah dan alami untuk menentukan apakah suatu wajah asli atau tidak. Mata akan tetap tertutup selama sekitar 200-350 mili detik pada saat kedipan. Rata-rata orang berkedip sekitar 5-10 kali dalam satu menit [10]. Dengan demikian kita dapat menggunakan implementasi deteksi kedipan mata untuk melakukan analisis *spoofing* wajah. Bagaimanapun kelemahan pendekatan ini ada pada serangan *spoofing* dinamis yang menggunakan video atau animasi. Keberadaan teknologi saat ini memudahkan untuk menyerang tayangan ulang rekaman video menggunakan gadget seperti *smartphone* atau tablet, sehingga mengandalkan deteksi mata kedip saja tidak cukup.

Metode lainnya untuk mendeteksi wajah *spoofing* adalah pendekatan tantangan dan respons. Metode ini merupakan teknik anti-*spoofing* yang cukup andal. Pendekatan ini menggunakan tindakan unik yang disebut sebagai tantangan (*challenge*) [11]. Sebagai contoh tantangan seperti perintah untuk menengok ke kiri atau ke kanan, mengangguk atau menggelengkan kepala. Perangkat yang digunakan akan berfungsi untuk mengkonfirmasi tantangan yang terjadi selama proses verifikasi. Sebuah model verifikasi dibangun berdasarkan pada kumpulan tantangan untuk mengkonfirmasi identitas seseorang. Namun demikian, sekalipun dapat berhasil, prosedur ini memerlukan *input* ekstra dan dapat secara signifikan mempengaruhi pengalaman pengguna. Bayangkan jika setiap kali *login* pengguna diminta untuk menggelengkan kepala atau mengangguk., tentunya hal ini akan secara signifikan mempengaruhi kenyamanan pengguna.

Salah satu cara lainnya untuk mendeteksi wajah *spoof* adalah dengan deteksi gerakan. Pendekatan ini bermaksud untuk mengenali tanda-tanda atau aktivitas kehidupan melalui evaluasi gerakan wajah individu. Gerakan inilah yang membedakan orang dengan benda mati seperti foto. Salah satu teknik pendeteksi gerakan yang paling banyak digunakan adalah pergeseran ekspresi wajah, dan gerakan bibir seperti senyuman. Metode evaluasi berbasis gerakan biasanya cukup untuk mencegah serangan representasi tidak aktif seperti foto-*spoofing* tetapi akhirnya menjadi tidak efektif ketika bekerja dengan serangan *rendering* dinamis seperti video [12].

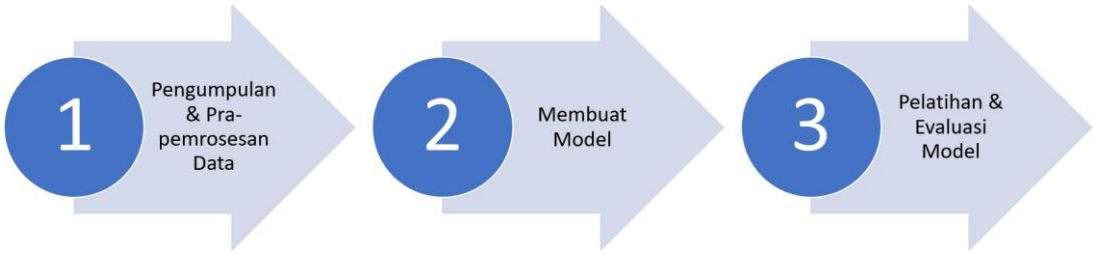
Pendekatan lainnya dalam mendeteksi wajah *spoof* dapat menggunakan perangkat keras tambahan. Kamera 3D atau *photoplethysmography* bisa menjadi pendekatan anti-*spoofing* yang dapat diandalkan. Menggunakan alat ini kedalaman piksel tertentu. Hal ini akan menawarkan presisi tinggi terhadap serangan *spoofing* wajah 3D karena dapat membedakan antara wajah dan objek datar. Bagaimana pun terlepas dari akses ke kamera yang mungkin sulit dijangkau, tidak banyak

pelanggan yang memilikinya di komputer mereka, dan tidak cocok untuk aplikasi di perangkat seluler seperti *smartphone*. Penggunaan alat seperti ini juga berarti biaya tambahan untuk otentikasi [13].

Dibandingkan dengan pendekatan berbasis perangkat lunak, metode pendekatan perangkat keras menggunakan sensor atau alat bantu untuk mengolah citra. Melalui alat bantu ini akan membuat perbedaan antara wajah asli dan serangan *spoofing* secara signifikan sehingga dapat dibedakan. Karena perbedaan antara wajah asli dan serangan *spoofing*, teknik pendekatan menggunakan multispektral [14], inframerah [15], dan foto *plethysmography* [16] dapat digunakan dalam deteksi *spoofing* yang memiliki akurasi tinggi. Namun, selain upaya untuk mendapatkan alatnya, perangkat ini juga memiliki proses instalasi perangkat keras yang relatif rumit. Oleh karena itu, pendekatan ini sulit untuk dimanfaatkan secara luas. Selain merepresentasikan data, perangkat keras berbasis sensor kepadatan juga digunakan untuk pendeteksian tanda kehidupan seperti pada kamera *time-of-flight* [17]. Proses ini secara efisien dapat mengatasi serangan 2D tetapi tidak serangan 3D. Makalah [18] menyediakan sistem deteksi keaktifan dengan menggunakan kamera medan cahaya, yang dapat mendeteksi berbagai serangan *spoofing* yang berbeda. Namun, peralatan pencitraan medan cahaya ini memiliki biaya mahal, dan hasil penemuan sangat dipengaruhi oleh cahaya. Secara umum kelemahan metode pendeteksian berbasis perangkat keras adalah harga peralatan yang mahal atau sulit diperoleh dan memerlukan proses instalasi tambahan. Oleh karena itu, metode ini tidak dapat diterapkan secara luas.

Dalam penelitian ini, model *deep learning* diterapkan untuk mendeteksi serangan *spoofing* wajah berdasarkan arsitektur CNN. Skema cara kerja model yang diusulkan cukup sederhana. *Input* akan melewati modul sensor *input* seperti *webcam* atau kamera depan pada *smartphone*. Jika gambar terdeteksi maka, *input* akan diteruskan untuk diproses ke modul CNN *classifier* apakah wajah itu palsu atau asli. *Input* dinyatakan sebagai wajah asli jika melewati memiliki nilai klasifikasi sama atau mendekati angka 1. Metodologi yang diusulkan mencakup beberapa langkah umum yang terdiri dari (1) pengumpulan dan pra-pemrosesan data, (2) membuat model (3) pelatihan dan evaluasi model. Gambaran alur penelitian yang dilakukan diilustrasikan pada Gambar 1.

Metode deteksi serangan *anti-spoofing* berbasis perangkat lunak memiliki biaya rendah dan presisi tinggi telah berkembang pesat dalam beberapa tahun terakhir. Beberapa contoh pendekatan aplikasi ini mengharuskan pengguna untuk berkedip [19], menggerakkan bibir mereka [20], atau melakukan tantangan sesuai dengan instruksi [21]. Secara umum pendekatan ini yang dapat secara efektif merespons serangan *spoofing*, tetapi juga menciptakan pengalaman pengguna yang buruk. Pada sisi lain pendekatan ini juga tidak efektif terhadap serangan dinamis seperti pemutaran video. Para peneliti mulai menyelidiki sistem analisis berdasarkan fitur kostum seperti pada makalah [22] untuk mengatasi masalah ini. Meskipun pendekatan ini dapat berfungsi dengan baik dalam *dataset* yang digunakan, pendekatan ini tidak cocok untuk aplikasi dunia nyata. Dengan semakin banyaknya *dataset* publik, studi untuk deteksi *liveness* telah banyak dilakukan [23], serta akurasi deteksi diperbaiki terus menerus.



Gambar 1. Tahap-tahap penelitian

A. Pengumpulan & Pra-pemrosesan Data

Tahap awal penelitian yang dilakukan adalah mengumpulkan data. ini dilakukan melalui *dataset* yang disediakan oleh beberapa peneliti sebelumnya [12] [15]. *Dataset* yang dikumpulkan adalah gambar dari berbagai jenis serangan *spoof* seperti poster, topeng, video, dan foto. *Dataset* juga memiliki variasi pencahayaan ruangan, kualitas resolusi gambar, dan latar belakang yang berbeda. Data ini akan membantu memastikan bahwa model tersebut tahan terhadap berbagai jenis serangan *spoofing*. *Dataset* yang digunakan pada penelitian ini ditunjukkan pada Tabel 1.

TABEL 1
DATASET YANG DIGUNAKAN

Dataset	Jumlah wajah asli	Jumlah spoof wajah
Casia V2	1701	3274
NUAA Photo Imposter	3345	1845
Total	5046	5119

Setelah *dataset* dikumpulkan, tahap selanjutnya adalah pra-pemrosesan data. Pada tahap pra-pemrosesan data dilakukan dengan berbagai cara sebagai berikut:

1. Mengubah ukuran gambar.
2. Mengurangi *noise* gambar.
3. Augmentasi gambar
4. Membagi *dataset*.

Pertama, ukuran gambar akan disesuaikan agar semua inputan menjadi seragam. Gambar diubah dengan ukuran 100x100 px. Hal ini diperlukan agar model dapat dengan lancar saat dilatih pada model arsitektur nantinya. Berikut akan mengurangi *noise* pada gambar dengan menggunakan filter *gaussian blur*. Hal ini dilakukan untuk meningkatkan akurasi model sehingga dapat memprediksi gambar lebih akurat.

Augmentasi gambar merupakan sebuah teknik yang berguna untuk memperbanyak data latih untuk model tanpa perlu mencari data yang baru. Augmentasi gambar pada prinsipnya adalah duplikasikan gambar yang telah ada dengan beberapa variasi sehingga data menjadi lebih banyak. Proses augmentasi gambar dilakukan dengan cara *zooming* +/- 10% dan rotasi +/- 10%, menyesuaikan *brightness* +/-10%, serta rotasi gambar -/+10%. Melalui cara ini model yang akan dilatih lebih sesuai dengan skenario dunia nyata dan dapat menyesuai berbagai perubahan kondisi yang ada. Terakhir *dataset* akan digunakan dengan rasio 80% untuk melatih model dan sisanya 20% digunakan untuk menguji program.

B. Membuat Model

Model arsitektur yang digunakan pada penelitian ini adalah Convolutional Neural Network (CNN). Model CNN digagas pertama kali oleh Yann LeCun dkk., pada tahun 1998 dalam sebuah makalah berjudul "*Gradient-Based Learning Applied to Document Recognition*" [24]. LeCun mengusulkan sebuah versi awal CNN yang bernama LeNet yang dapat mengenali karakter tulisan tangan. Pada waktu itu LeNet hanya dapat bekerja dengan baik pada gambar resolusi rendah.

Basis data yang digunakan LeCun adalah MNIST *handwritten digits*, terdiri dari angka tulisan tangan digital 0 hingga 9 dengan labelnya. *Dataset* MNIST dikenal sampai saat ini dan sudah banyak digunakan oleh para peneliti untuk melatih model *machine learning*. Sejak ditemukannya LeNet, para peneliti terus melakukan penelitian untuk mengembangkan model CNN. Hingga pada tahun 2012, Alex Krizhevsky memperkenalkan AlexNet [25], versi pengolah citra lebih baik dari LeNet yang memenangkan perlombaan terkenal ImageNet. AlexNet ini merupakan cikal bakal deep learning CNN.

Convolutional Neural Network (CNN) adalah algoritma *deep learning* yang dapat mengambil gambar *input*, menetapkan *importance* (bobot dan bias yang dapat dipelajari) untuk berbagai aspek atau objek pada gambar sehingga dapat membedakan satu dari yang lain. Hal tersebut merupakan salah satu mengapa model CNN populer digunakan dalam bidang pengolahan citra atau *computer vision*. Pra-pemrosesan yang diperlukan dalam CNN jauh lebih rendah dibandingkan dengan algoritma klasifikasi lainnya. Pada akhirnya hal tersebut akan memuat proses klasifikasi objek gambar akan lebih cepat. Sementara itu dalam metode *machine learning* primitif, filter direkayasa secara manual, dengan pelatihan yang memadai, adapun CNN telah memiliki kemampuan untuk mempelajari filter atau karakteristik ini.

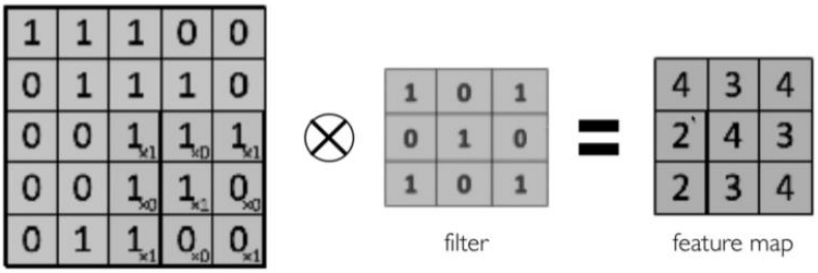
Sebuah gambar pada dasarnya tidak lain adalah matriks nilai piksel. Sebelum CNN diperkenalkan pemrosesan gambar kerap dilakukan dengan *flatten* gambar (misalnya gambar ukuran 3x3px menjadi vektor 9x1) kemudian memasukkannya ke *Multi-Level Perceptron* (MLP) untuk tujuan klasifikasi. Dalam kasus klasifikasi gambar biner yang sangat sederhana, metode ini mungkin menunjukkan skor presisi rata-rata saat melakukan prediksi kelas tetapi akan memiliki sedikit atau tidak ada akurasi ketika datang ke gambar kompleks yang memiliki ketergantungan piksel secara keseluruhan.

CNN bekerja dengan cara menangkap dependensi spasial dan temporal dalam gambar melalui penerapan filter yang relevan. Arsitektur melakukan *fitting* yang lebih baik ke *dataset* gambar karena pengurangan jumlah parameter yang terlibat dan bobot yang dapat digunakan kembali. Dengan kata lain, jaringan dapat dilatih untuk memahami kerumitan gambar dengan lebih baik. *Convolutional* layer dapat mengenali atribut pada objek menggunakan filter. Filter pada dasarnya hanyalah sebuah matriks yang berisi angka-angka. Sebagai contoh dari penerapan filter dapat dilihat pada gambar 2.

Gambar 2. Contoh penerapan filter

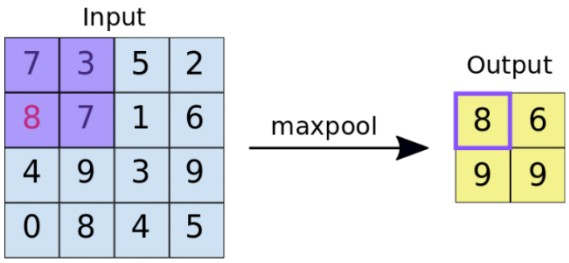
Pada sebuah gambar 2 aplikasi dari filter yang berbeda menghasilkan gambar yang berbeda. Melalui filter seperti pada gambar yang paling kanan, dapat mendeteksi garis-garis yang bisa menunjukkan apakah seseorang merupakan kuda atau manusia berdasarkan bentuk garisnya. Filter tersebut bekerja dengan cara mengalikan piksel pada gambar dengan matriks pada filter sehingga dapat menampilkan deteksi tepi pada sebuah gambar di sisi paling kanan.

Proses berikut adalah konvolusi yang merupakan proses yang mengaplikasikan filter pada gambar. Pada proses konvolusi ada perkalian matriks terhadap filter dan area pada gambar. Tujuan dari proses konvolusi adalah untuk mengekstrak fitur tingkat tinggi seperti tepi, dari gambar. CNN tidak perlu dibatasi hanya pada satu lapisan (layer) *convolutional*. Secara konvensional, CNN layer pertama bertanggung jawab untuk menangkap fitur tingkat rendah seperti tepi, warna, orientasi gradien. Adapun layer berikutnya arsitektur akan menyesuaikan dengan fitur tingkat tinggi. Proses konvolusi sebuah *input* gambar, filter, dan hasil dari proses konvolusi terhadap gambar (*feature map*) ditunjukkan pada gambar 3.



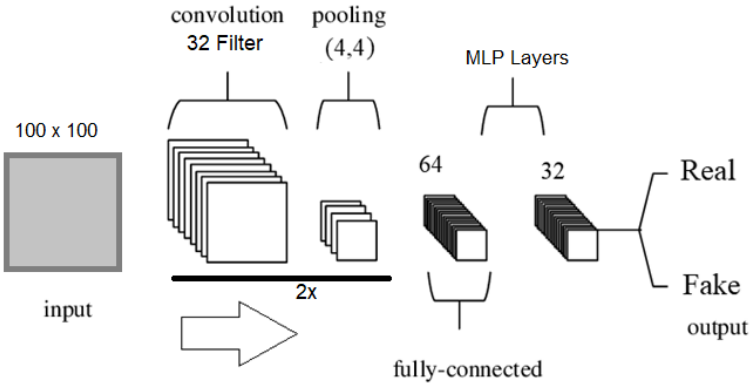
Gambar 3. Proses konvolusi gambar

Pada sebuah jaringan saraf tiruan, setelah proses konvolusi dilakukan pada gambar masukan, akan dilakukan proses *pooling*. *Pooling* adalah proses untuk mengurangi resolusi gambar dengan tetap mempertahankan informasi pada gambar. Pendekatan *pooling* yang dilakukan pada penelitian ini adalah *max pooling*. Pada *max pooling* (max *pool*) di antara setiap area dengan luas piksel tertentu, akan diambil satu buah piksel dengan nilai tertinggi. Hasilnya akan menjadi gambar baru. Ketika *maxpool* (2,2) dilakukan pada gambar berukuran 4x4 piksel maka akan menghasilkan dengan ukuran 2x2 piksel dengan nilai tertinggi seperti yang diilustrasikan pada gambar 4.



Gambar 4. Proses *maxpool*

Terakhir, hasil *max pooling* dapat dimasukkan ke dalam sebuah *hidden* layer MLP. Layer MLP ini bekerja layaknya sebagai *neural network* yang dapat memberikan bobot dan bias dalam mengklasifikasikan sebuah *input*an gambar. Pada sebuah arsitektur CNN dapat menggunakan beberapa lapis konvolusi dan *max pool* sebelum mulai memasukkannya ke layer MLP. Dengan beberapa lapis proses konvolusi, makin detail fitur yang dapat dikenali dari gambar. Pada proses konvolusi pertama dapat mendeteksi wajah dari seorang manusia. Lalu pada proses konvolusi kedua, wajah hasil konvolusi pertama dapat dideteksi fitur yang lebih detail seperti hidung, mata, dan telinganya sehingga, model makin pintar membedakan wajah setiap orang. Adapun penggunaan *max pool* berulang kali dapat mempercepat proses pelatihan model karena citra yang diolah memiliki lebih sedikit piksel. Model yang diusulkan pada penelitian ini seperti yang ditunjukkan pada gambar 5.



Gambar 5. Arsitektur model CNN yang diusulkan

Pada gambar 5 model menerima *input*an gambar dengan ukuran 100x100 px yang merupakan hasil *output* dari tahap pra pemrosesan data. *Input* ini diteruskan ke layer CNN yang memiliki 32 filter. Kemudian diteruskan ke layer *maxpool* dengan ukuran *pool* 4x4. Pada model yang diusulkan terdapat duplikat layer CNN dan *maxpool* ini sehingga proses dilakukan secara berulang sebanyak dua kali. Setelah itu data akan diproses pada layer MLP. Terdapat 2 layer MLP pada layer pertama memiliki 64 neuron sedangkan layer kedua memiliki 32 neuron. Semua neuron tersebut saling terkoneksi satu sama lainnya. Terakhir model diaktifkan dengan fungsi *sigmoid* sehingga menghasilkan *output* biner 0 atau 1 yang merupakan representasi dari gambar wajah asli atau *spoofing*.

C. Pelatihan & Evaluasi Model

Setelah dibuat model akan dilatih dengan *dataset* yang telah dipra-proses sebelumnya. Proses pelatihan model menggunakan parameter awal 300 jumlah *epoch*, *optimizer* adam dan *learning rate* 0,01. Parameter ini nanti akan dievaluasi kembali untuk menemukan parameter model dengan hasil terbaik. Sebagai metrik untuk dasar evaluasi model digunakan akurasi, presisi, pemanggilan ulang serta skor F1. Persamaan yang digunakan untuk menghitung nilai akurasi, presisi, pemanggilan, dan skor F1 ditunjukkan pada persamaan 1 sampai 4.

Akurasi = TP+TN/TP+FP+FN+TN (1)
Presisi = TP/TP+FP (2)
Pemanggilan ulang = TP/TP+FN (3)
Skor F1 = 2*(Pemanggilan ulang * Presisi) / (Pemanggilan ulang + Presisi) (4)
Di mana: TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), FN (*False Negative*)

Selama tahap penelitian ini, perangkat lunak yang digunakan untuk pengumpulan data hingga evaluasi model adalah bahasa pemrograman Python, IDE Jupyter Notebook, serta *library* TensorFlow, dan Keras. *Setting* lingkungan pengembangan model dilakukan pada sebuah *Workstation* dengan platform Windows 10, Intel Core I7 11050H, Memori DDR-4 32GB, dan kartu grafis Quadro T-2000 GPU 8GB.

III. HASIL DAN PEMBAHASAN

Pada proses pengumpulan data, diperoleh 10.165 gambar yang terdiri dari 5046 gambar wajah asli dan 5.119 wajah *spoof*. Gambar-gambar ini memiliki berbagai resolusi yang berbeda, serta latar belakang dan kondisi pencahayaan. Sebelum *dataset* gambar ini digunakan untuk melatih model, gambar ini akan dipra-proses terlebih dahulu secara bertahap yang dijelaskan di bagian metodologi. *Salah* tahap pra-pemrosesan yang penting adalah augmentasi gambar. Pada proses tersebut gambar diolah menggunakan library Tensorflow dan Keras dengan memanfaatkan modul ‘image preprocessing’ sehingga dapat menghasilkan *output* seperti pada gambar 6.

p-ISSN : 2443-2210
e-ISSN : 2443-2229

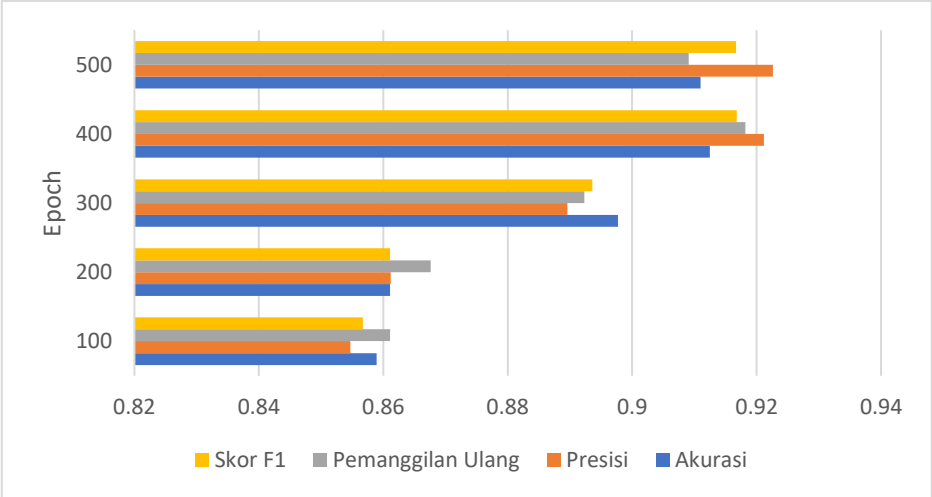
Gambar 6. Hasil augmentasi gambar

Gambar 6 menunjukkan contoh *dataset* gambar awal sebelum augmentasi gambar (gambar paling kiri). Kemudian di augmentasi dengan cara *zoom out* 10% dan *brighhness* -10% sehingga menghasilkan seperti gambar di tengah. Adapun gambar paling kanan menunjukkan hasil augmentasi gambar dengan *flip* horizontal dan *zoom in* 10%. Setelah melalui tahap pra-pemrosesan, *dataset* gambar digunakan untuk melatih dan menguji model. Hasil dari pelatihan model ditunjukkan pada tabel 2.

TABEL 2
HASIL PELATIHAN MODEL

Dataset	Akurasi	Presisi	Pemanggilan Ulang	Skor F1
Data Latih	0.9472	0.9343	0.9508	0.9424
Data Uji	0.8977	0.8896	0.8923	0.8909

Seperti dapat dilihat pada tabel 2 model dilatih dengan berbagai jumlah *epoch* yang disesuaikan Setelah model dilatih dengan *dataset* yang ada, akurasi terbaiknya mencapai 94,72%, presisi 93,43%, dan pemanggilan ulang 95,08% pada data pengujian. Sedangkan pada validasi data didapatkan akurasi 89,77%, presisi 88,96%, dan pemanggilan ulang 89,23%. Hasil ini menunjukkan bahwa model sudah cukup fit berdasarkan hasil pengujian pada data uji. Hal itu dengan nilai pengujian yang lebih rendah ketimbang hasil pelatihan model. Demikian pula akurasi dan nilai skor F1 yang menunjukkan hasil yang cukup baik, namun masih memungkinkan untuk ditingkatkan mengingat selisih hasil pelatihan dan pengujian cukup signifikan. Untuk proses evaluasi dalam mendapatkan nilai yang lebih baik, pertama kali dilakukan proses memantau nilai pada jumlah *epoch*. Hasil dari pengujian ini ditampilkan pada gambar 7.



Gambar 7. Hasil pengujian jumlah *epoch*

Hasil pengujian pada gambar 7 menunjukkan bahwa nilai akurasi dan skor F1 terbaik didapat dengan menggunakan jumlah *epoch* 400 dengan akurasi 91,25% dan skor F1 91,68%. Hasil ini pada dasarnya tidak berbeda secara signifikan

dengan hasil akurasi dan skor F1 pada jumlah *epoch* 500. Jumlah *epoch* yang lebih banyak berarti model lebih lama dilatih dengan harapan dapat meningkatkan akurasi keseluruhan. Bagaimanapun pada titik jumlah *epoch* tertentu nilai akurasi dan skor F1 tidak dapat ditingkatkan lagi. Terlalu banyak jumlah *epoch* akan memperlambat waktu pelatihan model. Pada sisi jumlah *epoch* yang terlalu banyak dapat membuat model menjadi *overfit*. Gejala model *overfit* ditunjukkan ketika model sangat baik memprediksi data latih namun secara signifikan menjadi kurang baik ketika dicoba pada data uji.

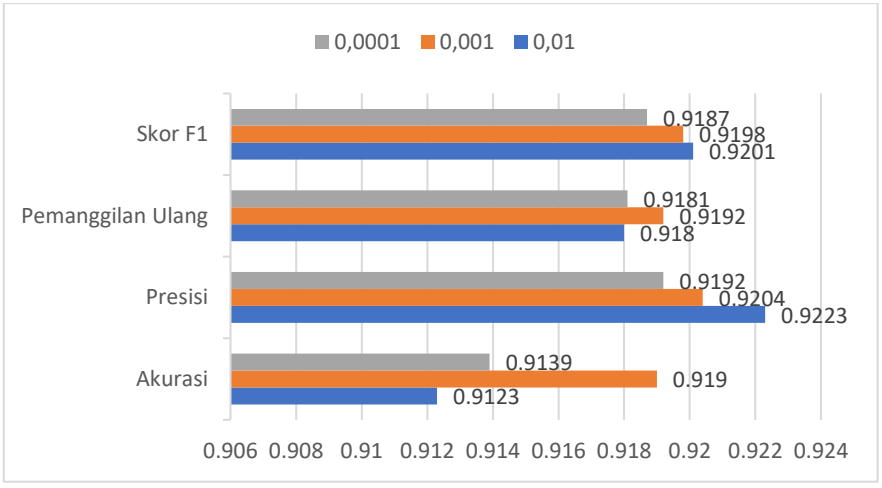
Pada langkah berikutnya model diuji untuk mengetahui tipe *optimizer* yang paling menghasilkan hasil yang optimal. Terdapat tiga tipe *optimizer* yang diuji yakni Adam, RMS Prop dan SGD. Pengujian akan dilakukan dengan menggunakan data uji dengan parameter pelatihan 400 jumlah *epoch*. Hasil pengujian ditampilkan pada tabel 3.

TABEL 3
HASIL PENGUJIAN TIPE OPTIMIZER

Tipe Optimizer	Akurasi	Presisi	Pemanggilan Ulang	Skor F1
Adam	0.9125	0.9212	0.9182	0.9168
RMS Prop	0.9123	0.9223	0.9180	0.9201
SGD	0.8189	0.8216	0.8222	0.8219

Pada tabel 3 menunjukkan hasil akurasi dan skor F1 terbaik didapatkan dengan tipe *optimizer* RMS Prop. Hasil ini tidak berbeda jauh dengan tipe *optimizer* Adam namun unggul signifikan dengan *optimizer* SGD. *Optimizer* digunakan untuk mengurangi perbedaan antara *output* yang diprediksi dan *output* sebenarnya dengan mencari nilai *optimal* pada pembobotan jaringan syaraf tiruan. Selain itu perlu dipastikan bahwa algoritma dapat digeneralisasi dengan baik. Hal ini akan membantu membuat prediksi yang lebih baik untuk data yang tidak terlihat sebelumnya. SGD bekerja dengan prinsip kerja momentum. Momentum membantu mempercepat *gradient descent* ketika perbedaan nilai antara *output* yang diprediksi dan *output* sebenarnya. Untuk memperbarui bobot dibutuhkan gradien langkah saat ini serta gradien langkah waktu sebelumnya. Ini membantu kita bergerak lebih cepat menuju konvergensi. Hal ini mengakibat model dapat cepat belajar namun hasilnya mungkin kurang akurat. Pada sisi lain *optimizer* RMSProp akan disesuaikan secara otomatis dan memilih tingkat pembelajaran yang berbeda untuk setiap parameter. Hal ini menyebabkan tipe *optimizer* ini akan terus menyesuaikan bobot dari jaringan syaraf tiruan untuk mendapatkan hasil yang terbaik.

Pada tahap pengujian selanjutnya akan diuji parameter *learning rat*e. Nilai *learning rate* yang digunakan adalah 0,01 (nilai *default*); 0,001 dan 0,0001. Pengujian menggunakan data uji dan parameter epoch 400 dan tipe *optimizer* RMSProp. Hasil pengujian ditampilkan pada gambar 8.



Gambar 8. Hasil Pengujian *Learning Rate*

Eksperimen selanjutnya yang dilakukan adalah pengujian parameter *learning rate* yang ditampilkan pada gambar 8. Berdasarkan hasil pengujian menunjukkan parameter *learning rate* hasilnya tidak menunjukkan perbedaan hasil yang signifikan. Parameter *learning rate* menunjukkan seberapa banyak model belajar untuk menyesuaikan bobotnya agar hasil prediksi sebisa mungkin sesuai dengan label sebenarnya. Menggunakan *learning rate* yang lebih sedikit diharapkan dapat mempercepat meminimalkan *loss* serta meningkatkan akurasi. Bagaimanapun pada saat yang sama *learning rate* yang lebih

kecil akan memperlambat waktu pelatihan model untuk mencapai hasil yang optimal. Bagaimanapun berdasarkan hasil pengujian tidak menunjukkan perubahan yang signifikan sekalipun *learning rate* sudah diturunkan. Hal ini kemungkinan diakibatkan model sudah mencapai jumlah *epoch* yang optimal. Jumlah *epoch* dan *learning rate* sangat berkaitan erat. Dapat diibaratkan *epoch* adalah frekuensi belajar sedangkan *learning rate* adalah seberapa lama waktu belajar. Jika sudah menemukan frekuensi belajar yang pas, maka lama waktu belajar sudah lagi tidak menjadi hal yang signifikan.

IV. SIMPULAN

Seiring banyaknya perangkat yang menggunakan otentikasi biometrik pengenalan wajah, kebutuhan akan anti-*spoof* wajah adalah suatu kebutuhan mutlak. Makalah ini mengusulkan pendekatan anti-*spoofing* wajah menggunakan metode *deep learning* berdasarkan arsitektur *convolution neural network*. Berdasarkan hasil pengujian dapat disimpulkan bahwa model yang diusulkan cukup andal ketika mendeteksi serangan *spoof* wajah. Hasil evaluasi model menunjukkan model dengan parameter jumlah *epoch* 400, *learning rate* 0,01 dan tipe *optimizer* RMSProp memberikan nilai hasil terbaik dengan akurasi 91,23% dan skor F1 92,01%. Penelitian selanjutnya dapat difokuskan untuk menemukan kinerja anti-*spoofing* yang lebih baik sehingga dapat bekerja lebih cepat dan efisien dengan menggunakan teknik pemrograman paralel.

DAFTAR PUSTAKA

[1] P. Kavitha and K.Vijaya, "A Study on Spoofing Face Detection System," International Journal of Pure and Applied Mathematics, vol. 117, no. 22, pp. 205-208, 2017.
[2] F. Sthevanie and R. kurniawan, "Spoofing detection on facial images recognition using LBP and GLCM combination," International Conference on Data and Information Science, Bandung, 2018.
[3] D. Wen, "Face Spoof Detection with Image Distortion Analysis," IEEE Biometrics Compendium, vol. 10, no. 4, pp. 746 - 761, 2015.
[4] Schiffman, H.R., "Sensation and Perception. An Integrated Approach", New York: John Wiley and Sons, Inc., 2001
[5] B. Zinelabidine, K. Jukka and H. Abdenour, "Face Anti-spoofing based on Color Texture Analysis," 22nd IEEE International Conference on Image Processing (ICIP), Chicago, 2015.
[6] Y. A. Rahman, M. Liu and L. M. Po, "Deep learning for face anti-spoofing: An end-to-end approach," Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), Hong Kong, 2017.
[7] E. Alexey, "Algorithm for optimization of Viola–Jones object detection framework parameters," Journal of Physics Conference Series, no. 1, p. 945, 2018.
[8] Z. Boulkenafet, J. Komulainen and A. Hadid, "Face Anti-Spoofing Based on Color Texture Analysis," Machine Vision Research, vol. 1, 2015.
[9] R. Hasan, H. Mahmud and X. Y. Li, "Face Anti-Spoofing Using Texture-Based Techniques and Filtering Methods," Journal of Physics: Conference Series, 2019.
[10] I. B. Kusuma, A. Kartika, T. A. Budi, K. N. Ramadhani and F. Sthevanie, "Image Spoofing Detection Using Local Binary Pattern and Local Binary Pattern Variance," International Journal on Information and Communication Technology (IJoICT), vol. 4, no. 2, pp. 11-18, 2018.
[11] K. Larbi, W. Ouarda, H. Drira, B. B. Amor and C. B. Amar, "DeepColorFASD: Face Anti Spoofing Solution Using a Multi Channeled Color Spaces CNN," International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, 2018.
[12] A. Anjos, J. Komulainen, S. Marcel, A. Hadid and M. Pietik, "Face Anti-spoofing: Visual Approach," Handbook of Biometric Anti-Spoofing: Trusted Biometrics under Spoofing Attacks, London, Springer, 2014, pp. 65-82.
[13] G. Pan, L. Sun and S. Lao, "Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera," IEEE International Conference on Computer Vision, Rio de Janeiro, 2017.
[14] S. Parveen, S. Mumtazah, M. Hanafi and W. Azizun, "Face anti-spoofing methods," Current Science, vol. 108, no. 8, pp. 1491-1500, 2015.
[15] Y. Liu, J. Stehouwer, A. Jourabloo and X. Liu, "Deep Tree Learning for Zero-Shot Face Anti-Spoofing," The IEEE Conference on Computer Vision and Pattern Recognition, California, 2019.
[16] U. Hasanah, Mayangsari and P. Lintang, "Perbandingan Metode SVM, FUZZY-KNN, dan BDT-SVM Untuk Klasifikasi Detak Jantung," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 3, no. 1, pp. 201-210, 2016.
[17] E.Marasco and A. Ross. "A survey on antispoofing schemes for fingerprint recognition systems", ACM Computing Surveys (CSUR), 47 (2), 2015.
[18] L. Dora, S. Agrawal, R. Panda, A. Abraham. "An evolutionary single Gabor kernel based filter approach to face recognition", Engineering Applications of Artificial Intelligence, 62 (1), pp 286-301, 2017.
[19] L. Feng, Lai-Man Po, Y. Li, X. Xu, F. Yuan, T. Chun-Ho Cheung, Kwok- Wai Cheung. "Integration of image quality and motion cues for face anti-spoofing: A neural network approach", JVCIR, 38, pp 451-460, 2016.
[20] Wu, B., Pan, M., Zhang, Y.: "A review of face anti-spoo_ng and its applications in china". In: International Conference on Harmony Search Algorithm. pp. 35{43), Springer, 2019.
[21] Yuan, S., Liang, D., Shi, L., Zhao, X., Wu, J., Li, G., Qiu, L.: "Recent progress on distributed structural health monitoring research at nuaa". Journal of Intelligent Material systems and structures 19(3), 2008
[22] Sachinsdate, "lip-movement-net" [Online]. Available: https://github.com/sachinsdate/lip-movement-net/ [Access on 10 July 2020].
[23] M. A. Abuzneid and A. Mahmood, "Enhanced Human Face Recognition Using LBPH Descriptor, Multi-KNN, and Back-Propagation Neural Network," in IEEE Access, vol. 6, pp. 20641-20651, 2018, doi: 10.1109/ACCESS.2018.2825310.
[24] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.
[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. 2017. "ImageNet classification with deep convolutional neural networks". In Communications of the ACM Vol. 60 No.6 (June 2017), 84–90. DOI: https://doi.org/10.1145/3065386