

WEBINAR STUDENT PRESENCE SYSTEM BASED ON REGIONAL CONVOLUTIONAL NEURAL NETWORK USING FACE RECOGNITION

Akbar Trisnamulya Putra*¹, Koredianto Usman², Sofia Saidah³

^{1,2,3}Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Indonesia
¹akbartrisnamulya@gmail.com, ²korediantousman@telkomuniversity.ac.id,
³sofiasaidahsfi@telkomuniversity.ac.id

(Naskah masuk: 05 Februari 2021, diterima untuk diterbitkan: 23 Maret 2021)

Abstract

World health organization announce Covid-19 as a pandemic so On March 15th 2020, the social distancing has been established with working, learning, and praying from home. Webinar is one of the solutions so those activities still can be done face to face and conference-based. With webinar, users can interact each other in an online meeting from home. Student presence is part of a webinar. The purpose of this research is to design an accurate student presence with a face recognition system using R-CNN method. The object of this research is a human face with sufficient light, medium, and the face must be facing the camera. This research proposed for a webinar student presence system is using face recognition with Regional Convolutional Neural Network (R-CNN). With object detection and several scenarios used in this method, the webinar student presence system using R-CNN will be more accurate than the methods that have ever been used before. This research has done four scenarios to obtain the best parameters like 45 of total layers, test data of the whole dataset percentage as 10%, RMSProp as model optimizer, and 0.0001 learning rate. With those parameters, it have resulted the best system performance including 99.6% accuracy, 1×10^{-4} loss, 100% precision, 99% recall, and 99.5% F1 Score.

Keywords: deep learning, face recognition, neural network, R-CNN

SISTEM PENCATATAN KEHADIRAN MAHASISWA PADA WEBINAR BERBASIS REGIONAL CONVOLUTIONAL NEURAL NETWORK MENGGUNAKAN PENGENALAN CITRA WAJAH

Abstrak

Organisasi kesehatan dunia mengumumkan Covid-19 sebagai pandemi sehingga pada tanggal 15 Maret 2020, ditetapkan masa *social distancing* dengan cara bekerja, belajar, dan beribadah dilaksanakan di rumah. Webinar merupakan salah satu cara agar kegiatan tersebut bisa dilakukan secara tatap muka dan bersifat *conference* sehingga pengguna berinteraksi selayaknya bertemu langsung walaupun dilaksanakan di rumah masing-masing. Salah satu aktifitas pada webinar adalah pencatatan kehadiran peserta. Penelitian ini bertujuan merancang sistem pencatatan kehadiran mahasiswa pada webinar yang baik dan akurat dengan pengenalan wajah menggunakan metode R-CNN. Objek yang diteliti adalah wajah manusia dengan kondisi cahaya yang cukup, media yang memadai, serta wajah menghadap ke depan kamera. Penelitian yang diusulkan untuk sistem pencatatan kehadiran mahasiswa pada webinar adalah dengan menggunakan metode pengenalan wajah berbasis *Regional Convolutional Neural Network* (R-CNN). Dengan proses deteksi objek dan beberapa skenario yang dilakukan pada metode ini, sistem pencatatan mahasiswa pada webinar ini lebih akurat dibandingkan dengan metode yang telah digunakan sebelumnya. Penelitian ini menggunakan empat skenario untuk mendapatkan parameter terbaik yaitu jumlah *layer* sebanyak 45, persentase data uji terhadap dataset sebesar 10%, RMSProp sebagai *optimizer* modelnya, dan *learning rate* sebesar 0.0001. Dengan hal-hal tersebut dapat dihasilkan performansi terbaik pada sistem yaitu memiliki akurasi sebesar 99.6%, *loss* sebesar 1×10^{-4} , presisi sebesar 100%, 99% *recall*, dan 99.5% F1 Score.

Kata kunci: deep learning, jaringan syaraf tiruan, pengenalan wajah, R-CNN

1. PENDAHULUAN

Organisasi kesehatan dunia mengumumkan Covid-19 sebagai pandemi sehingga tanggal 15 Maret 2020, Presiden Joko Widodo meminta masyarakat

agar melakukan *social distancing* dengan cara bekerja, belajar, dan beribadah dilaksanakan di rumah. Agar kegiatan-kegiatan tersebut bisa tetap dilaksanakan, daring menjadi salah satu solusi yaitu

menggunakan *video conference* (webinar). Webinar adalah *platform* tatap muka yang bersifat *conference* sehingga pengguna bisa berinteraksi selayaknya bertemu langsung [1].

Salah satu aktifitas yang dilakukan pada webinar adalah pencatatan kehadiran yang umumnya menggunakan Google Form yang dimodifikasi. Untuk meminimalisir kecurangan, cara yang dilakukan adalah penyedia mengatur waktu pengisian Google Form. Dengan cara ini, pengisi Google Form akan otomatis tertolak jika pengisian dilakukan di luar batas waktu yang telah ditentukan [2]. Maka dari itu dibutuhkan penelitian Sistem Pencatatan Mahasiswa pada Webinar Berbasis Regional Convolutional Neural Network (R- CNN) Menggunakan Pengenalan Citra Wajah.

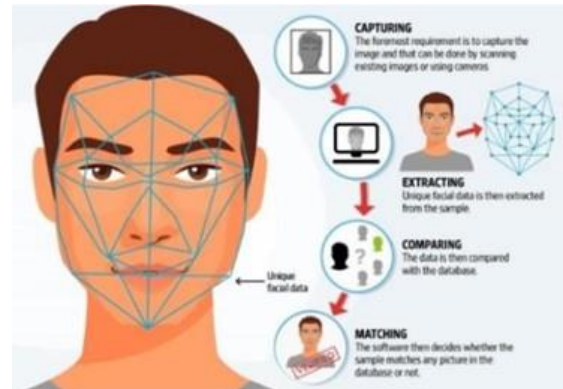
Saat ini penelitian pengenalan wajah sudah banyak dilakukan dilihat dari manfaat wajah yang digunakan dan dikolaborasi dengan IoT saat ini. Salah satu metode penelitian yang pernah digunakan dalam pengenalan wajah adalah *Convolutional Neural Networks* (CNN). Penelitian ini dilakukan menggunakan dataset sebanyak 126 gambar. Metode ini dapat mengurangi parameter bebas dan deformasi citra input sehingga dapat mendeteksi wajah manusia secara *real-time* dengan tingkat akurasi sebesar 87,48% [3]. Kemudian terdapat sistem pengenalan wajah menggunakan metode *Triangle Face* yaitu dengan dataset sebanyak 23 sampel dari 5 orang, metode ini mengukur jarak fitur wajah yang diperlukan sehingga membentuk pola segitiga. Hasilnya dengan jarak 30 cm, didapatkan akurasi tertinggi yaitu 82,6% [4]. Terdapat pula penelitian pengenalan wajah dengan metode Viola-Jones dan algoritma *Principal Component Analysis* (PCA) yang menggunakan 200 *dataset*. Metode Viola-Jones menggabungkan 4 kunci utama yaitu *Haar Like Feature*, *Integral Image*, *Adaboost Learning*, dan *Cascade Classifier* sedangkan algoritma PCA digunakan untuk mereduksi dimensi sekumpulan atau ruang citra agar koordinat baru dapat menggambarkan model yang khas dari kumpulan tersebut. Hasilnya adalah tingkat akurasi akhir sebesar 95% [5]. Dari penjelasan tersebut, diusulkan penelitian Sistem Pencatatan Mahasiswa pada Webinar Berbasis *Regional Convolutional Neural Network* (R-CNN) Menggunakan Pengenalan Citra Wajah yang dikembangkan dengan performansi yang lebih baik dari metode sebelumnya.

2. METODE PENELITIAN

2.1. Sistem Pengenalan Wajah

Sistem pengenalan wajah merupakan identifikasi suatu individu dengan memanfaatkan wajah berbasis biometric [5]. Biometrik adalah cara identifikasi dan verifikasi individu berdasarkan karakteristik fisik atau perilakunya [6]. Secara umum, sistem pengenalan wajah ini dibagi menjadi dua, yaitu *feature-based* dan *image-based*. *Feature based*

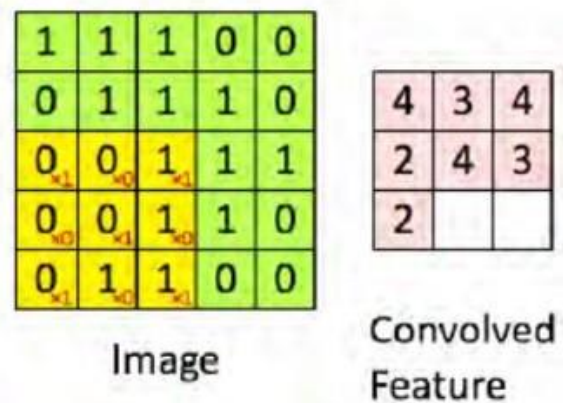
merupakan sistem pengenalan wajah yang memanfaatkan fitur yang diekstraksi dari komponen citra wajah seperti hidung, mulut, dan sebagainya. Sedangkan *image-based* merupakan sistem pengenalan wajah yang memanfaatkan informasi dari tiap piksel citra dan direpresentasikan dalam metode tertentu [7]. Contoh sistem pengenalan wajah dapat ditunjukkan oleh gambar 1.



Gambar 1. Contoh Sistem Pengenalan Wajah [8]

2.2. Konvolusi

Konvolusi adalah fungsi yang diaplikasikan pada output dari fungsi lain secara berulang. Dalam pengolahan citra, konvolusi dapat ditunjukkan pada gambar 2.



Gambar 2. Operasi Konvolusi [9]

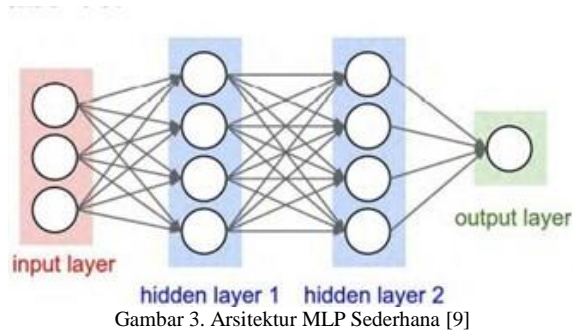
Pada gambar 2. sebelah kiri, filter atau dapat disebut juga sebagai kernel ditunjukkan dengan warna kuning dan citra keseluruhan berwarna hijau. Kernel ini bergerak dari sudut kiri atas hingga ke kanan bawah dengan pergerakan secara horizontal. Kernel ini mengaplikasikan fungsinya sesuai posisi sebelum bergerak, begitu juga saat mulai bergerak ke kotak hijau selanjutnya sehingga hasilnya seperti pada gambar di sebelah kanannya [9].

2.3. Convolutional Neural Network (CNN)

Neural Networks (NN) adalah variasi dari *Multilayer Perceptron* (MLP) yang terinspirasi dari

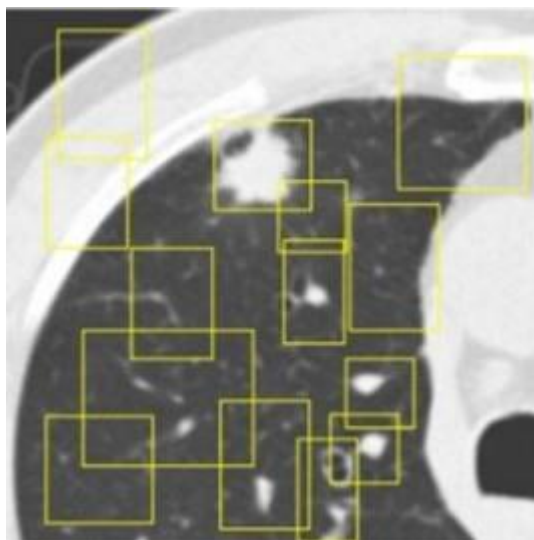
jaringan syaraf manusia. Maka CNN merupakan NN yang menggunakan konvolusi sebagai operasinya [3].

Cara kerja CNN mirip dengan MLP, yaitu pada CNN setiap neuronnya dipresentasikan dalam bentuk dua dimensi, sedangkan MLP hanya satu dimensi [9]. Secara umum, jenis *layer* pada CNN dibedakan menjadi dua yaitu *layer* ekstraksi fitur gambar dan *layer* klasifikasi [3].



2.4. Regional Convolutional Neural Network (R-CNN)

R-CNN adalah sebuah metode untuk mendeteksi objek, dengan menggunakan CNN untuk klasifikasi beberapa region dalam sebuah citra [10] seperti ditunjukkan oleh gambar 4. Dalam proses deteksi dengan metode R-CNN, dibentuk 2000 persegi dalam sebuah citra yang disebut region proposal [11]. Dalam masing- masing region proposal ini digunakan metode CNN [10].

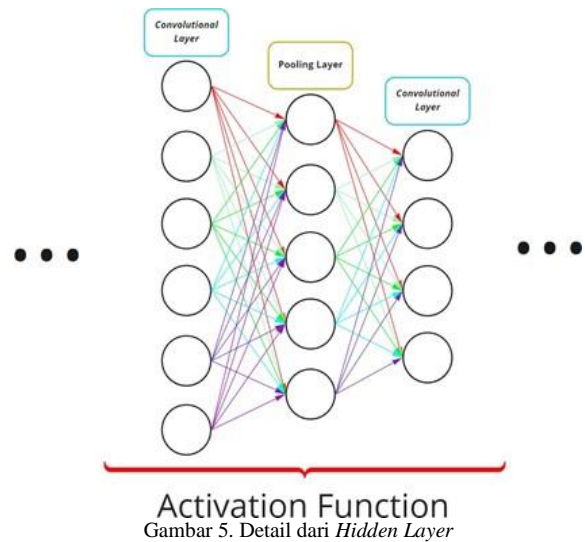


Gambar 4. Region dari R-CNN [10]

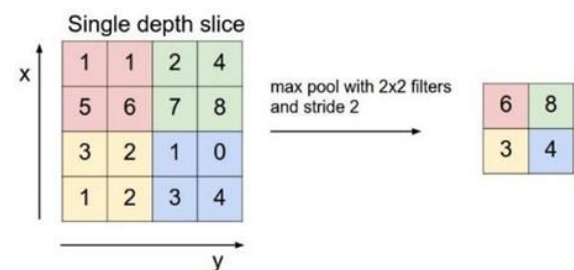
2.5. Layer Ekstraksi Fitur Gambar

Layer ini berada di awal arsitektur serta terdiri dari *layer* konvolusi dan *pooling layer*. Setiap neuron diberlakukan fungsi aktivasi dan posisi dari tiap jenis *layer* ini bergantian. *Layer* ini menerima input dari citra secara langsung dan diolah agar menghasilkan keluaran berupa vektor yang akan diolah lagi pada *layer* selanjutnya.

Dari gambar 5, terlihat bahwa di dalam *hidden layer* terdapat *convolutional layer* dan *pooling layer* yang saling bergantian dengan fungsi aktivasi pada tiap neuronnya [12]. Di *convolutional layer* ini dilakukan operasi konvolusi pada tiap region dengan melakukan metode *sliding window* menggunakan filter atau kernel dengan ukuran dan nilai tertentu. Pada *pooling layer* dilakukan operasi *subsampling* agar dapat mereduksi ukuran citra [13].



Operasi yang digunakan adalah *max pooling* seperti pada gambar 6.



Gambar 6. Operasi *Max Pooling* pada *Pooling Layer* [14]

Pada gambar 6, dapat dilihat bahwa *max pooling* menggunakan *2x2 filter* dengan *stride 2* sehingga setiap irisan dari input akan dipilih nilai terbesar dan hasilnya seperti pada gambar di sebelah kanan [14]. Tiap *layer* ini memiliki neuron yang harus diaktifkan ataupun tidak diaktifkan oleh fungsi aktivasi [15]. Fungsi aktivasi yang digunakan ialah *softmax* yang dapat digunakan untuk klasifikasi lebih dari dua kelas karena dalam penelitian ini mendeteksi hingga 10 wajah. Persamaan dari *softmax* adalah sebagai berikut:

$$f_j = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (1)$$

Pada persamaan 1, f_j merupakan hasil fungsi dari setiap elemen ke- j . Z adalah hipotesis yang diberikan oleh model agar dapat diklasifikasikan oleh fungsi

softmax. Fungsi ini memberikan hasil berupa interpretasi probabilistik yang memiliki rentang 0 hingga 1 sehingga bisa mendeteksi lebih dari 2 kelas [14].

2.6. Layer Klasifikasi

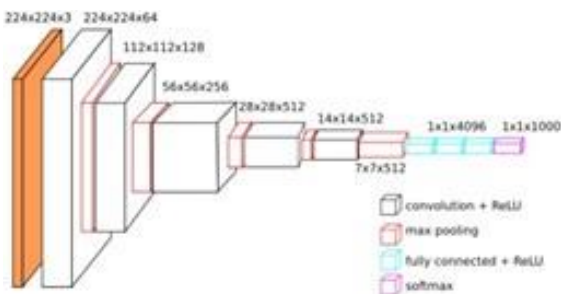
Layer ini tersusun atas beberapa layer yang tiap layer-nya saling terhubung secara penuh (*fully connected*) satu sama lain. Layer ini memiliki keluaran yang sama seperti jenis sebelumnya, yaitu berupa vektor yang kemudian ditransformasikan seperti *multi-NN* dengan beberapa tambahan *hidden layer*. Hasil keluaran berupa *scoring* kelas untuk klasifikasi [3].

2.7. Optimizer

Optimizer adalah algoritma untuk memperbaiki bobot dan bias pada NN dengan cara memperkecil jarak antara output dari jaringan terhadap target [16]. Dalam prosesnya, *optimizer* dibantu oleh *learning rate* untuk menghitung nilai koreksi bobot pada saat latih data dengan rentang nilai *learning rate* dari nol (0) hingga satu (1). Semakin besar nilai *learning rate* maka semakin cepat proses latih data, tetapi konsekuensinya adalah *optimizer* menjadi kurang baik untuk tingkat ketelitian saat proses tersebut [17]. Beberapa *optimizer* yang umum untuk digunakan yaitu *Adaptive Momentum (ADAM)*, *Root Mean Square Propagation (RMSProp)*, dan *Stochastic Gradients Descent (SGD)*.

ADAM adalah gabungan dari *Optimizer RMSProp* dan *Momentum* dengan beberapa keunggulan antara lain efisien dalam komputasi, hemat penyimpanan dan fleksibel pada masalah optimisasi *non-convex* di bidang pembelajaran mesin [20]. SGD adalah algoritma yang digunakan untuk memperbarui bobot dan bias. Tujuan *optimizer* ini adalah untuk menemukan parameter yang dapat meminimalkan *loss*. RMSProp adalah *optimizer* yang memanfaatkan besarnya *gradient* terbaru untuk melakukan normalisasi *gradient* sehingga terjaga di atas *gradient Root Mean Square*, dan disebut RMS. Keunggulan RMSProp salah satunya dapat menangani tujuan stokastik dengan baik [18].

2.8. Model VGG16



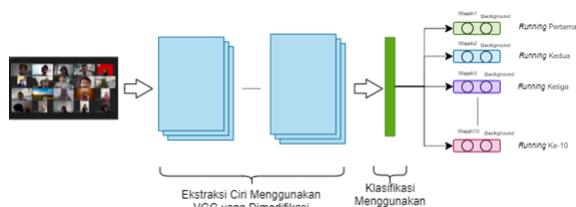
Gambar 7. Arsitektur VGG16 dengan 16 Layer

VGG16 adalah *very deep convnet model* yang dibuat oleh *Visual Geometry Group (VGG)*, Universitas Oxford. Model ini memiliki 16 layer yang terdiri dari *convolutional layer*, *max pooling layer*, dan *fully-connected layer*. Pada *convolutional layer* dan *fully-connected layer*, berlaku fungsi aktivasi *Rectified Linear Unit (ReLU)*, sedangkan pada output layer menggunakan fungsi aktivasi *softmax*. Arsitektur dari model ini dapat diilustrasikan seperti pada gambar 7 [19].

3. ALUR KERJA PENELITIAN

3.1. Desain Sistem

Desain sistem secara umum dapat diilustrasikan dengan menggunakan diagram blok seperti pada gambar 8.

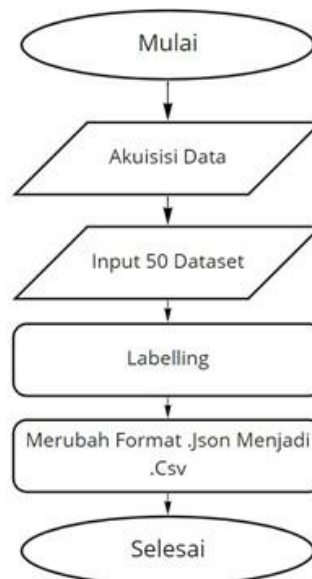


Gambar 8. Desain Sistem Secara Umum

Desain sistem pada penelitian ini secara garis besar terbagi menjadi tiga bagian yaitu *preprocessing*, proses utama, dan *postprocessing*.

3.1.1 Pre-Processing

Pada tahap ini, persiapan dilakukan agar dapat mendukung inti program dari penelitian ini. Terdapat dua bagian utama dari tahap *preprocessing*, yaitu pengolahan dataset dan penyesuaian program. Diagram alir pengolahan dataset dapat diilustrasikan seperti pada gambar 9.



Gambar 9. Tahap Pengolahan Dataset

Akuisisi dataset merupakan tahap awal dalam pengolahan dataset. Tujuannya adalah mendapatkan dataset yang baik sehingga mempermudah melakukan tahap-tahap selanjutnya dari pengolahan dataset. Dalam melakukan akuisisi dataset, semua peserta webinar menyalakan kamera sehingga bagian wajah sisi depan dari ujung atas dahi hingga dagu serta kedua mata terlihat dengan rentang jarak wajah terhadap kamera sekitar 50 cm dan dibantu kondisi cukup cahaya dengan total cahaya yang terpancar minimal 500 lumen dalam ruangan seluas 3.6 m² seperti pada gambar 10.



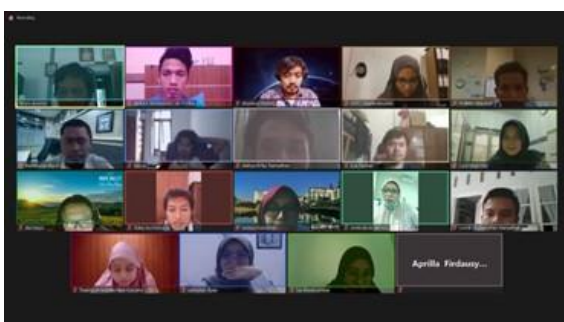
Gambar 10. Jarak Wajah dan Layar 50 cm

Gambar 10 menunjukkan contoh dari akuisisi dataset dengan jarak wajah terhadap kamera sejauh ± 50 cm dan dibantu dengan kondisi cahaya yang terpancar sebesar 500 lumen. Hasil dari akuisisi dataset dapat ditunjukkan oleh gambar 11.



Gambar 11. Contoh Wajah dengan Cahaya Terpancar 500 Lumen

Pada gambar 11 dapat dilihat bahwa hasil dari contoh akuisisi dataset ini menunjukkan bagian wajah sisi depan dari ujung atas dahi hingga dagu serta kedua mata terlihat dengan baik. Penelitian ini mengakuisisi dataset sebanyak 50 citra yang terdapat hingga 20 wajah berbeda pada tiap dataset. Contoh dataset yang didapat dapat ditunjukkan oleh gambar 12.



Gambar 12. Contoh Labelling pada 19 Wajah dalam 1 Dataset

Pada gambar 12, terlihat *bounding box* sebagai label pada setiap koordinat wajah. Tujuan dari penggunaan *bounding box* adalah memudahkan program mendapatkan informasi dari setiap wajah untuk dikenali. Keluaran dari *labeling* ini merupakan *bounding box ground truth* yang berbentuk *.json dan diubah menjadi *.csv serta penyesuaian nama agar mempermudah dalam melakukan pemanggilan tiap label terhadap citra pada program. Semua data ini disimpan pada satu *directory* yang sama pada Google Drive, kemudian dataset dan *bounding box ground truth* dibuatkan *folder* masing-masing agar memudahkan program untuk menghubungkan dengan data yang diolah.

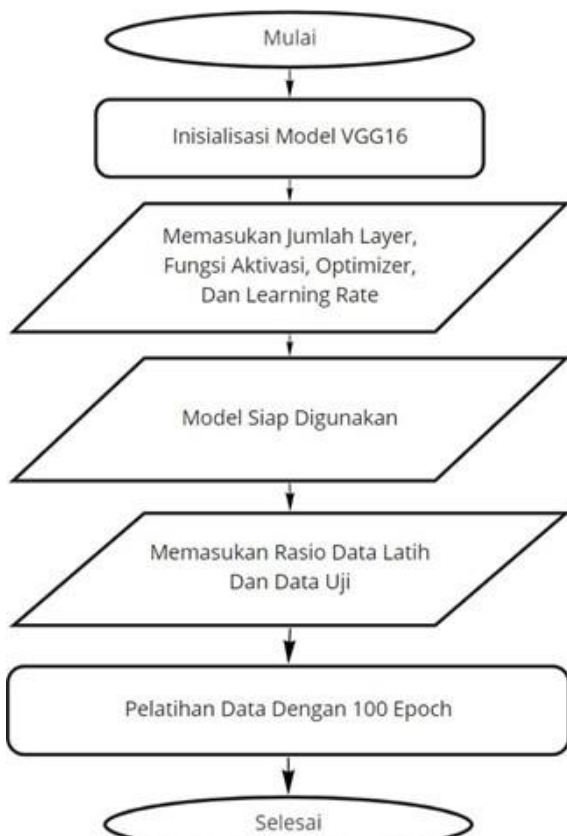
Saat memulai merancang program pada Google Colab, hal pertama yang harus dilakukan ialah menyesuaikan program dengan data yang sudah disiapkan seperti pada diagram alir gambar 13.



Gambar 13. Tahap Penyesuaian Program

3.1.2 Proses Utama

Setelah dataset diolah dan program telah disesuaikan, maka tahap ini yang merupakan inti dari program dapat dijalankan seperti pada diagram alir pada gambar 14.



Gambar 14. Tahap R-CNN

Inisialisasi model VGG dari R-CNN ini merupakan tahap awal karena model ini harus dihubungkan dari pustaka ke program. Setelah terhubung, barulah model bisa dimodifikasi sesuai kebutuhan. *Hyperparameter* yang diatur ialah jumlah *layer*, jenis *optimizer*, dan *learning rate*. Struktur model akan ditampilkan jika telah selesai mengatur parameter-parameter tersebut. Setelah model ditampilkan, persentase data uji terhadap dataset dapat diatur sesuai kebutuhan. Setelah hal tersebut diatur, maka model VGG yang telah dimodifikasi dapat dijalankan dengan metode R-CNN. Dataset yang telah diolah dijadikan input pada model tersebut kemudian dibagi menjadi 2000 region. Tiap regionnya dilakukan fungsi CNN dengan *convolutional layer* dan *pooling layer* disusun bergantian. *Pooling layer* yang digunakan adalah *max pooling* serta fungsi aktivasi yang digunakan adalah ReLU pada *convolutional layer* dan *fully-connected layer* sedangkan pada outputnya digunakan fungsi aktivasi *softmax*. Pelatihan data ini dilakukan dengan *epoch* sebesar 100.

3.1.3 Postprocessing

Setelah berhasil melakukan semua tahap sebelumnya, pada tahap ini hasil performansi diukur dengan melakukan kalkulasi pada nilai akurasi, *loss*, presisi, *recall*, dan *F1 Score* serta melakukan pencatatan kehadiran dalam bentuk .Csv.

3.2. Parameter Performansi

3.2.1 Intersection over Union (IoU)

IoU adalah ukuran kemiripan dan diversitas antara citra hasil prediksi model dengan citra target. Semakin besar nilai IoU maka semakin baik prediksi model terhadap target. Untuk melakukan kalkulasi IoU, mengetahui arti *True Positive*, *True Negative*, *False Positive*, dan *False Negative* dapat membantu kalkulasi menjadi lebih mudah.

True Positive (TP) adalah saat model melakukan prediksi bahwa data tersebut positif dan yang sebenarnya terjadi adalah positif. *True Negative* (TN) adalah saat model melakukan prediksi bahwa data tersebut negatif dan yang sebenarnya terjadi adalah negatif. *False Positive* (FP) adalah saat model melakukan prediksi bahwa data tersebut positif dan yang sebenarnya terjadi adalah negatif. *False Negative* (FN) adalah saat model melakukan prediksi bahwa data tersebut negatif dan yang sebenarnya terjadi adalah positif. IoU dapat dibuat dengan persamaan (2).

$$IoU = \frac{TP}{TP + FP + FN} \quad (2)$$

3.2.2 Akurasi

Akurasi merupakan perbandingan antara keseluruhan *true* baik TP maupun TN dengan keseluruhan data. Persamaan dari akurasi dapat dilihat pada persamaan (3)[20][21]:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

Dari persamaan 3, dapat disimpulkan bahwa semakin besar nilai akurasi maka semakin baik model dalam melakukan prediksi pada objek yang diinginkan.

3.2.3 Presisi

Presisi adalah rasio dari TP dengan jumlah data yang diprediksi positif. Persamaan dari presisi dapat dilihat pada persamaan (4)

$$Presisi = \frac{TP}{TP + FP} \quad (4)$$

Dari persamaan 4, dapat disimpulkan bahwa semakin besar nilai presisi maka semakin baik model dalam melakukan prediksi positif dengan kesalahan prediksi positif yang rendah.

3.2.4 Recall

Recall adalah rasio dari TP dengan jumlah data yang sebenarnya positif. Persamaan dari *recall* dapat dilihat pada persamaan (5).

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

Dari persamaan 5, dapat disimpulkan bahwa semakin besar nilai *recall*, maka semakin baik model dalam melakukan prediksi positif dengan kesalahan prediksi negatif yang rendah.

3.2.5 F1 Score

F1 Score adalah *harmonic mean* dari presisi dan *recall*. Persamaan dari F1 Score dapat dilihat pada persamaan (6).

$$\frac{1}{F1\ Score} = \frac{1}{2} \left(\frac{1}{Presisi} + \frac{1}{Recall} \right) \quad (6)$$

Dari persamaan 6, dapat disimpulkan bahwa semakin besar nilai F1 Score, maka semakin baik model dalam memiliki nilai akurasi dan *recall*.

3.2.6 Loss

Loss adalah rasio dari seluruh *false* baik FP dan FN dengan seluruh data. Persamaan dari *loss* dapat dilihat pada persamaan (7).

$$Loss = \frac{FP + FN}{TP + TN + FP + FN} \quad (7)$$

Dari persamaan 7, dapat disimpulkan bahwa semakin besar nilai *loss*, maka semakin buruk model dalam melakukan prediksi terhadap objek yang diinginkan.

3.2.7 Pencatatan Kehadiran

Setelah program berhasil melakukan pengenalan pada tiap wajah, maka tahap selanjutnya adalah melakukan pencatatan kehadiran. Tahap ini dilakukan dengan penyesuaian wajah yang sudah dilatih dengan ada atau tidaknya wajah pada data uji. Jika suatu wajah pada data uji dikenali oleh program maka wajah tersebut dianggap hadir. Data kehadiran ini dimasukkan kedalam sebuah file berbentuk .Csv.

4. HASIL DAN PEMBAHASAN

Perangkat yang digunakan untuk menunjang penelitian ini adalah sebagai berikut:

Aspek	Spesifikasi
Perangkat Lunak	Google Colab dan Google Drive
CPU	Intel(R) Core(TM) i5-8300H CPU @ 2.30 GHz (8 CPUs)
GPU	Geforce GTX 1050 4GB
RAM	16 GB 2400MHz

Berdasarkan tabel 1, digunakan Google Colab yang merupakan *executable document* berbasis *browser* agar mendapatkan performa yang lebih baik

daripada hanya mengandalkan perangkat keras saja. Media penyimpanan yang digunakan adalah Google Drive berbasis *browser*. Browser yang digunakan adalah Google Chrome dan semua perangkat lunak yang dimanfaatkan menggunakan satu penyedia yaitu Google agar dapat terintegrasi satu sama lain.

Pada sisi perangkat keras, ada tiga hal yang paling berpengaruh terhadap performa pengolahan data pada penelitian ini yaitu CPU, GPU, dan RAM. Kombinasi dari beberapa perangkat lunak dan perangkat keras ini dapat membantu performa pengolahan data.

4.1. Pengujian Sistem

Pengujian sistem ini bertujuan untuk mengetahui hasil performansi pada sistem yang dibuat yaitu akurasi, *loss*, *presisi*, *recall*, dan F1 Score. Pengujian ini dilakukan menggunakan VGG sebagai arsitektur *layer* dari metode R-CNN dan dimodifikasi pada jumlah *layer* dengan 100 *epoch*. Penelitian ini dapat mengenali 10 wajah yang memiliki masing-masing *script code* sehingga hasilnya akan bervariasi. Maka dari itu, ditampilkan hasil pengujian sistem dengan parameter rata-rata dari 10 data terbaik pada masing-masing parameter.

4.2. Pengujian Sistem

Pada tahap ini pengujian dilakukan sebanyak empat skenario dengan masing-masingnya mencari parameter terbaik terhadap performansi sistem. Empat skenario yang akan diuji antara lain :

1. Mencari jumlah *layer* terbaik menggunakan bantuan parameter-parameter yang sudah ditentukan.
2. Mencari persentase data uji terhadap dataset terbaik menggunakan *layer* terbaik yang telah didapat pada skenario sebelumnya.
3. Mencari *optimizer* terbaik dengan menggunakan hasil terbaik dari dua skenario sebelumnya.
4. Mencari *learning rate* terbaik dengan menggunakan hasil terbaik dari tiga skenario sebelumnya.

4.3. Skenario Pertama

Pada skenario pertama, pengujian dilakukan untuk mencari jumlah *layer* terbaik dengan beberapa parameter yang telah ditentukan yaitu persentase data uji terhadap dataset sebesar 10%, *optimizer* menggunakan RMSprop, dan *learning rate* sebesar 0.0001. Hasil dari pengujian skenario pertama ini dapat dilihat pada tabel 2.

Jumlah Layer	Akurasi (%)	Loss	Presisi (%)	Recall (%)	F1 Score (%)
15	97.8	2.2	96	100	97.7
30	99.4	0.6	100	98.2	99
45	99.6	0.4	100	99	99.5

Berdasarkan tabel 2, dapat disimpulkan bahwa jumlah *layer* terbaik terhadap performansi sistem yaitu sebanyak 45.

4.4. Skenario Kedua

Pada skenario kedua, pengujian dilakukan untuk mencari persentase data uji terhadap dataset terbaik dengan beberapa parameter yang telah ditentukan yaitu *optimizer* menggunakan RMSprop dan *learning rate* sebesar 0.0001 serta parameter terbaik pada skenario sebelumnya yaitu jumlah *layer* sebanyak 45. Hasil dari pengujian skenario kedua ini dapat dilihat pada tabel 3.

Tabel 3. Hasil Pengujian Persentase Data Uji Terhadap Dataset

Persentase Data Uji	Akurasi (%)	Loss	Presisi (%)	Recall (%)	F1 Score (%)
10%	99.6	0.4	100	99	99.5
20%	98.6	1.4	94.5	99	96
25%	99.2	0.8	98.6	100	99.2

Berdasarkan tabel 3, dapat disimpulkan bahwa persentase data uji terhadap dataset terbaik terhadap performansi sistem yaitu sebesar 10%.

4.5. Skenario Ketiga

Pada skenario ketiga, pengujian dilakukan untuk mencari *optimizer* terbaik dengan parameter yang telah ditentukan yaitu *learning rate* sebesar 0.0001 sedangkan parameter terbaik pada skenario sebelumnya yaitu jumlah *layer* sebanyak 45 serta persentase data uji terhadap dataset sebesar 10%. Hasil pengujian skenario ketiga ditunjukkan tabel 4.

Tabel 4. Hasil Pengujian *Optimizer*

Jenis <i>Optimizer</i>	Akurasi (%)	Loss	Presisi (%)	Recall (%)	F1 Score (%)
RMSProp	99.6	0.4	100	99	99.5
ADAM	95.3	4.7	100	89.7	93.1
SGD	98.1	1.9	97.6	93.2	94.5

Berdasarkan tabel 4, dapat disimpulkan bahwa jenis *optimizer* terbaik terhadap performansi sistem yaitu RMSProp.

4.6. Skenario Keempat

Pada skenario keempat pengujian dilakukan untuk mencari *learning rate* terbaik dengan parameter terbaik dari skenario sebelumnya yaitu jumlah *layer* sebanyak 45, persentase data uji terhadap dataset sebesar 10%, dan *optimizer* dengan jenis RMSProp. Hasil dari pengujian skenario keempat ini dapat dilihat pada tabel 5.

Tabel 5. Hasil Pengujian *Learning Rate*

<i>Learning Rate</i>	Akurasi (%)	Loss	Presisi (%)	Recall (%)	F1 Score (%)
0.0001	99.6	0.4	100	99	99.5
0.0002	98.8	1.2	99	93.1	100
0.0004	99.3	0.7	99.5	95.2	98.6

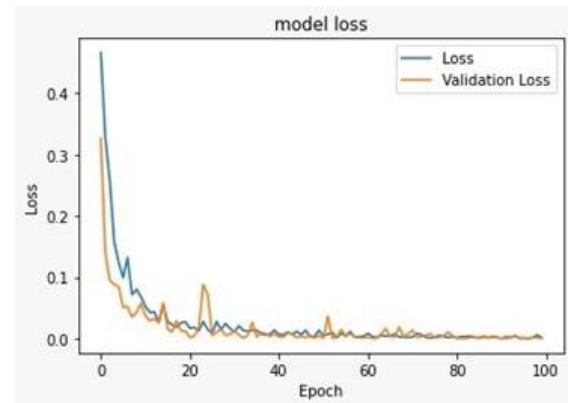
Berdasarkan tabel 5, dapat disimpulkan bahwa *learning rate* terbaik terhadap performansi sistem yaitu sebesar 0.0001.

4.7. Pengujian dengan Skenario Terbaik

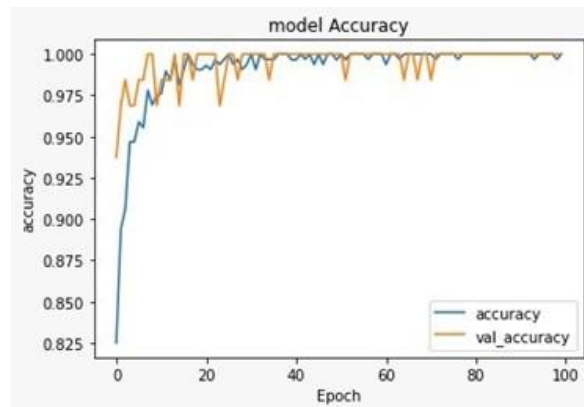
Pengujian tahap ini menggunakan parameter-parameter terbaik yang didapatkan oleh hasil dari empat skenario sebelumnya. Dengan menggunakan parameter terbaik yang sudah didapatkan, maka pengujian dengan skenario terbaik bisa dilakukan yaitu dengan menggunakan jumlah *layer* sebanyak 45, persentase data uji terhadap dataset 10%, *optimizer* dengan jenis RMSProp, dan *learning rate* sebesar 0.0001. Hasil dari pengujian skenario terbaik ini dapat dilihat pada tabel 6 serta grafik pada gambar 15 dan 16.

Tabel 6. Hasil Pengujian dengan Parameter Terbaik

Presisi (%)	Recall (%)	F1 Score (%)	Akurasi (%)	Loss
100	99	99.5	99.6	1 x 10 ⁻⁴



Gambar 15. Grafik Loss



Gambar 16. Grafik Akurasi

4.8. Pencatatan Kehadiran

Setelah melakukan pengujian sistem dan mendapatkan performansi terbaik, maka pencatatan kehadiran menggunakan pengenalan wajah dapat dilakukan dengan baik. Contoh 10 wajah yang dikenali dan dicatat dalam 1 dataset dapat ditunjukkan oleh gambar 17.



Gambar 17. Contoh Gambar dengan 10 Wajah

Pada gambar 17, terdapat 10 wajah yang sudah dilatih dan dicatat dengan menggunakan parameter-parameter terbaik. 10 wajah tersebut dimasukkan dengan nama aditya_rifky, luthfi_muhammad, akbar_trisnamulya_putra, annisa_humairani, tsabita_al_ashhifa, zakiy_burhan, ahzanul_kholish, lyra_vega, samsinar_ilyas, dan robin_sinurat. Hasil dari pencatatan dari wajah yang telah dilatih dapat ditunjukkan oleh gambar 18.

1	aditya_rifky	
2	Hadir	
3	luthfi_muhammad	
4	Hadir	
5	akbar_trisnamulya_putra	
6	Hadir	
7	annisa_humairani	
8	Hadir	
9	tsabita_al_ashhifa	
10	Hadir	
11	zakiy_burhan	
12	Hadir	
13	ahzanul_kholish	
14	Hadir	
15	lyra_vega	
16	Hadir	
17	samsinar_ilyas	
18	Hadir	
19	robin_sinurat	
20	Hadir	

Gambar 18. Hasil Pencatatan Kehadiran

Pada gambar 18, dapat dilihat bahwa semua wajah dapat dikenali dan dicatat kehadirannya sesuai nama yang telah dimasukkan sebelumnya.

5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa Sistem Pencatatan Mahasiswa pada Webinar Berbasis R-CNN Menggunakan Pengenalan Citra Wajah mampu dirancang dengan baik sehingga pencatatan kehadiran bisa dilakukan. Penelitian ini juga mampu untuk menentukan parameter-parameter agar menghasilkan performansi terbaik. Saran pada penelitian ini adalah sebaiknya jumlah dan variasi dari dataset diperbanyak serta kecepatan proses dari program bisa ditingkatkan lagi.

DAFTAR PUSTAKA

- [1] E. Yulianto, P. D. Cahyani, and S. Silvianita, "Perbandingan Kehadiran Sosial dalam Pembelajaran Daring Menggunakan Whatsapp groupdan Webinar Zoom Berdasarkan Sudut Pandang Pembelajar Pada Masa Pandemic COVID-19," *J. Ris. Teknol. Dan Inov. Pendidik.*, vol. 3, no. 2, pp. 331–341, 2020.
- [2] H. Hasan, "Optimalisasi Google Form Dalam Pembelajaran Bahasa Arab di Era Pandemi Covid-19," *Pros. Konf. Nas. Bhs. Arab*, vol. 6, no. 6, pp. 181–192, 2020.
- [3] M. dan B. S. Zufar, "Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time," *SAINS DAN SENI ITS*, vol. 5, no. 2, pp. 2337–3520, 2016.
- [4] P. P. Arhandi, U. D. Rosiani, A. Prasetyawati, and P. Choirina, "Sistem Pengenalan Wajah Untuk Keamanan Folder Menggunakan Metode Triangle Face," *J. Inform. Polinema*, vol. 4, no. 4, pp. 268–273, 2018.
- [5] Y. dan Ida Nurhaida, "Rancang Bangun Aplikasi Pengenalan Wajah menggunakan Metode Viola-Jones dan Algoritma PCA," *Telekomun. dan Komput.*, vol. 8, no. 3, pp. 196–204, 2018.
- [6] N. D. Miranda, L. Novamizanti, and S. Rizal, "Convolutional Neural Network Pada Klasifikasi Sidik Jari Menggunakan Resnet-50," *J. Tek. Inform.*, vol. 1, no. 2, pp. 61–68, 2020.
- [7] C. Lesmana, R. Lim, and L. W. Santoso, "Implementasi Face Recognition menggunakan Raspberry pi untuk akses Ruang Pribadi," *J. Infra*, vol. 7, no. 1, pp. 63–66, 2019.
- [8] E. Indra, M. D. Batubara, M. Yasir, S. Chau, and others, "Desain dan Implementasi Sistem Absensi Mahasiswa Berdasarkan Fitur Pengenalan Wajah dengan Menggunakan Metode Haar-Like Feature: Sistem Informasi," *J. Teknol. Dan Ilmu Komput. Prima*, vol. 2, no. 2, pp. 11–18, 2019.
- [9] I. Putra, "Klasifikasi citra menggunakan convolutional neural network (CNN) pada caltech 101," Institut Teknologi Sepuluh Nopember, 2016.
- [10] S. Kido, Y. Hirano, and N. Hashimoto, "Detection and classification of lung abnormalities by use of convolutional neural network (CNN) and regions with CNN features (R-CNN)," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, 2018, pp. 1–4, doi: 10.1109/IWAIT.2018.8369798.
- [11] J. R. Kala, D. M. Kre, A. N. Gnassou, J. R. K. Kala, Y. M. A. Akpablin, and T. Coulibaly, "Assets management on electrical grid using Faster-RCNN," *Ann. Oper. Res.*, pp. 1–14,

- 2020.
- [12] A. K. Nandi and H. Ahmed, "Deep Learning," in *Condition Monitoring with Vibration Signals: Compressive Sampling and Learning Algorithms for Rotating Machines*, Wiley-IEEE Press, pp. 279–305, 2019.
- [13] M. A. Pangestu and H. Bunyamin, "Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model," *J. Tek. Inform. dan Sist. Inf.*, vol. 4, no. 2, pp. 341–348, 2018.
- [14] S. Ilahiyah and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," *JUSTINDO (Jurnal Sist. dan Teknol. Inf. Indones.)*, vol. 3, no. 2, pp. 49–56, 2018.
- [15] J. Julpan, E. B. Nababan, and M. Zarlis, "Analisis Fungsi Aktivasi Sigmoid Biner dan Sigmoid Bipolar dalam Algoritma Backpropagation pada Prediksi Kemampuan Siswa," *J. Teknovasi J. Tek. dan Inov.*, vol. 2, no. 1, pp. 103–116, 2018.
- [16] A. Nasuha, T. A. Sardjono, and M. H. Purnomo, "Pengenalan Viseme Dinamis Bahasa Indonesia Menggunakan Convolutional Neural Network," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 7, no. 3, pp. 258–265, 2018.
- [17] R. D. Nurfitra and S. T. Gunawan Ariyanto, "Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari," Universitas Muhammadiyah Surakarta, 2018.
- [18] O. N. Putri and others, "Implementasi Metode CNN dalam Klasifikasi Gambar Jamur pada Analisis Image Processing (Studi Kasus: Gambar Jamur dengan Genus Agaricus dan Amanita)," 2020.
- [19] Y. Gultom, A. M. Arymurthy, and R. J. Masikome, "Batik classification using deep convolutional network transfer learning," *J. Ilmu Komput. dan Inf.*, vol. 11, no. 2, pp. 59–66, 2018.
- [20] Y. I. Kurniawan, and T. I. Barokah, "Klasifikasi Penentuan Pengajuan Kartu Kredit Menggunakan K-Nearest Neighbor," *Jurnal Ilmiah Matrik*, vol. 22, no. 1, pp 73-82, 2020.
- [21] A. A. Rahman, and Y. I. Kurniawan, "Aplikasi Klasifikasi Penerima Kartu Indonesia Sehat Menggunakan Algoritma Naïve Bayes Classifier," *Jurnal Teknologi dan Manajemen Informatika*, vol. 4, no. 1, 2018.