

Optimisasi Hyperparameter CNN Menggunakan Random Search Untuk Deteksi COVID-19 Dari Citra X-Ray Dada

Malik Dwi Yoni Fordana¹, Naim Rochmawati²

^{1,2}Jurusan Teknik Informatika/Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya

¹malik.17051204024@mhs.unesa.ac.id

²Naimrochmawati@unesa.ac.id

Abstrak— *Corona Virus Disease* merupakan penyakit yang memiliki resiko penularan sangat tinggi sehingga mengakibatkan pandemi di seluruh dunia. Berbagai metode pencegahan dilakukan untuk menekan bertambahnya jumlah kasus akibat virus ini. Berbagai metode digunakan untuk mencegah penyebaran virus COVID-19, salah satunya dengan melakukan deteksi awal pada orang yang terindikasi mengalami gejala-gejala terpapar virus COVID-19. Alternatif lain yang bisa dipertimbangkan untuk mendeteksi COVID-19 pada pasien, yaitu menggunakan pendekatan *deep learning* dengan pencitraan CT dan X-Ray. Metode *deep learning* yang digunakan yaitu Convolutional Neural Network. Namun permasalahan yang sering dihadapi saat menggunakan CNN adalah kurang efisiennya nilai parameter yang digunakan pada setiap *filter* atau *layer*. Sehingga hasil yang didapatkan kurang optimal baik dari nilai akurasi maupun efisiensi dari model CNN. Proses optimisasi menjadi salah satu solusi yaitu dengan menerapkan algoritma *Random Search* untuk mencari kombinasi dari keempat nilai *hyperparameter* model CNN. Keempat *hyperparameter* tersebut terdiri dari *convolutional layer*, *kernel size*, *fully connected layer*, dan *learning rate*. Hasil perhitungan menunjukkan bahwa setelah dilakukan optimisasi menggunakan *Random Search* selama kurang lebih 65 menit didapatkan 50 model terbaik dengan nilai akurasi terbesar yaitu 95.38% dan akurasi terendah yaitu 88%.

Kata Kunci— *Deep Learning*, Optimisasi, *Random Search*. CNN, COVID-19.

I. PENDAHULUAN

Penyakit Coronavirus 2019 (COVID-19) merupakan penyakit yang sangat menular dan sudah menyebar dengan cepat secara global. Penyakit ini menginfeksi hampir semua negara dengan jumlah kasus positif dan kematian yang terus bertambah tiap waktu [1]. Virus Corona pertama kali diidentifikasi di Wuhan, China pada Desember 2019 [10], [13]. Virus ini ditemukan dengan mudah menular dari orang ke orang mengakibatkan pandemi yang mempengaruhi berbagai sektor kehidupan seperti kesehatan, keuangan, hingga mental health. Semakin bertambah hari, terdapat ratusan jumlah kasus kematian akibat virus ini dan ribuan jumlah korban terinfeksi dari berbagai negara [2], [12], dan hingga pada tulisan ini dibuat (10 Juni 2021) jumlah kasus kematian akibat COVID-19 mencapai sekitar 3.8 juta lebih [3].

Beberapa strategi dilakukan guna menekan laju pertambahan penderita yang relatif sangat cepat seperti dengan melakukan pelacakan asal mula penderita, isolasi kota/negara atau melakukan tes masal. Dengan demikian angka kematian dapat diturunkan [4]. Saat ini metode yang digunakan untuk mendeteksi penyakit COVID-19 adalah

Reverse Transcriptase-Polymerase Chain Reaction (RT-PCR) dan *swab*. Namun tes uji RT-PCR ini memakan waktu dan biaya yang tidak sedikit [5].

Alternatif lain yang bisa dipertimbangkan adalah dengan pendekatan *deep learning* menggunakan pencitraan CT dan X-Ray dada [11]. Saat ini *deep learning* terbukti sebagai metode yang ampuh dalam pendeteksian COVID-19 yang ditunjukkan oleh gambar X-RAY dan CT. Beberapa sistem dikembangkan menggunakan model yang sudah dilatih dan jaringan saraf yang disesuaikan dengan gambar X-Ray dan CT sebagai gambar input [16]. Dikarenakan COVID-19 menyerang sel *epitel* pada saluran pernapasan, maka sangat dimungkinkan jika mendeteksi COVID-19 menggunakan citra X-Ray atau CT. Citra X-Ray maupun CT biasanya digunakan medis untuk mendiagnosis penyakit seperti *pneumonia*, radang paru-paru, *abscesses*, dan sebagainya [8].

Beberapa penelitian dengan menggunakan pencitraan CT dan X-Ray pernah dilakukan sebelumnya untuk mendeteksi penyakit COVID-19 yaitu Afrizal Zein melakukan pendeteksian COVID-19 menggunakan CNN dengan *binary classification* (Positif dan Negatif). Akurasi yang didapatkan mencapai 92%. Namun dalam model tersebut dataset yang diperoleh hanya sebatas 50 citra X-ray [7]. Penelitian lain yang berkaitan dengan deteksi COVID-19 adalah dari Dilbag Singh, et al., yang membandingkan antara tiga metode dalam mendeteksi penyakit COVID-19 yaitu CNN, ANN, ANFIS. Dalam penelitiannya dataset yang berupa gambar CT pasien dibagi kedalam *training* dan *testing* dalam beberapa skema rasio yang kemudian dibandingkan diantara ketiga metode yang digunakan. Hasil akhir yang didapat yaitu nilai *accuracy*, *sensitivity*, dan *specificity* CNN lebih besar berturut-turut 0.978%, 0.826%, dan 0.682% dari ANN dan ANFIS [5].

Penelitian yang dilakukan oleh Yuli Sun Hariyani, dkk., menggunakan arsitektur ResNet50 dan metode *transfer learning* dalam pengujiannya [9]. Data gambar X-Ray pasien yang didapat dibagi menjadi 60% untuk data latih, 20% untuk data validasi, dan 20% untuk data tes. Nilai akurasi yang didapat dari penelitian tersebut juga sangat besar yaitu mencapai 97%. Sementara itu, Prabira Kumar Sethy & Santi Kumari Behera membandingkan 11 arsitektur CNN untuk mendeteksi penyakit COVID-19 [8]. Pada model CNN tersebut dikembangkan dengan menggunakan *transfer learning*. Diantara 11 arsitektur CNN, ResNet50 memberikan hasil akhir yang lebih bagus daripada 10 arsitektur lainnya. Nilai *accuracy* yang didapat sebesar 95.38%, *sensitivity* 97.29%, dan *specificity* sebesar 93.47%.

Soumya, R. et al. juga membandingkan antara model CNN yang sudah ada seperti AlexNet, VGG-16, MobileNet-V2, Squeezenet, ResNet-34, dan Inception-V3. Dalam penelitiannya digunakan dataset dari dua sumber yang berbeda. Untuk dataset pasien yang dinyatakan positif Corona berasal dari Cohen JP sejumlah 203 gambar. Sementara untuk pasien normal sebanyak 203 gambar yang didapatkan dari GitHub. Hasil yang didapatkan bahwa Resnet mempunyai nilai akurasi terbesar daripada model yang lain yaitu sebesar 98.33% [15].

Irmak, E. mengembangkan model CNN untuk *binary classification* [COVID-19(+) dan COVID-19(-)] dan *multiclassification* [COVID-19(+), normal dan *Pneumonia*] dengan melakukan tuning hyperparameter menggunakan *Grid Search*. Model CNN yang dibangun menggunakan 12 layer menghasilkan nilai akurasi sebesar 98.92% pada *binary classification*. Sementara pada 14 layer menghasilkan nilai akurasi sebesar 98.27% pada *multi classification* [6]. *Grid Search* sendiri merupakan metode tradisional untuk mencari nilai parameter terbaik dengan range yang sudah ditentukan. Namun algoritma ini akan memakan waktu yang lama saat eksekusi dikarenakan algoritma ini bekerja dengan mengkombinasikan semua *range* parameter yang sudah ditentukan sebelumnya. Berbeda dengan algoritma *Random Search*. Algoritma *Random Search* akan melakukan pencarian parameter secara acak. Pencarian acak dapat mengungguli pencarian *grid* karena hanya membutuhkan waktu yang lebih sedikit dalam pengeksekusiannya [14].

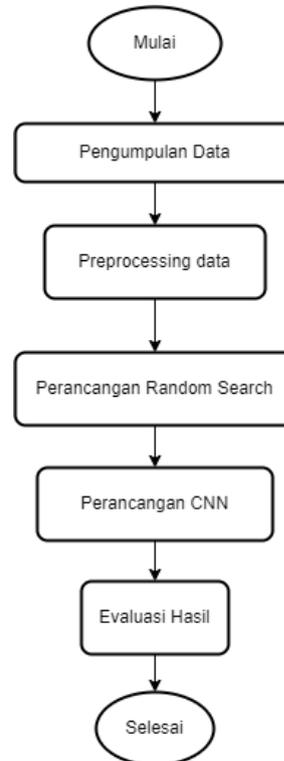
Dari beberapa penelitian terkait yang sudah dilakukan sebelumnya dapat diketahui bahwa penggunaan CNN untuk mendeteksi penyakit COVID-19 pada citra CT atau X-Ray menghasilkan nilai akurasi yang bagus. Fokus pada tulisan ini adalah untuk mengembangkan sebuah model CNN untuk mendeteksi penyakit COVID-19 dari citra X-Ray pasien menggunakan *deep learning*. CNN digunakan untuk mengklasifikasi apakah citra pasien terdeteksi COVID-19(+) atau COVID-19(-). Sedangkan *hyperparameter* dari CNN akan dilakukan optimisasi menggunakan algoritma *Random Search*. Evaluasi akan digunakan untuk mengetahui performa dari model CNN terbaik yang berupa nilai *accuracy* dan *error* setelah dilakukan optimisasi *hyperparameter*.

II. METODE PENELITIAN

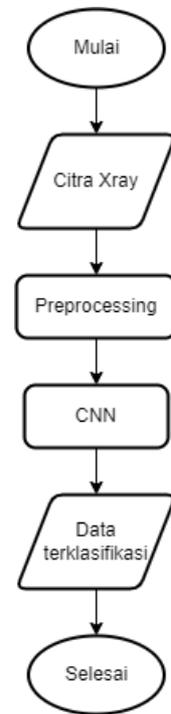
Metode penelitian merupakan langkah yang dilakukan oleh seorang peneliti untuk mengumpulkan data yang diperlukan dalam sebuah penelitian. Dalam penelitian ini dimulai dari perumusan masalah, penentuan tujuan dan batasan masalah, studi literatur, pengumpulan *dataset*, perancangan *Random*

Search, perancangan model CNN, pengujian performa dan evaluasi. Beberapa tahapan dalam penelitian ini dapat dilihat pada Gbr 1a.

Pada penelitian ini juga dilakukan perancangan aplikasi berbasis web dimana pada aplikasi ini akan berfungsi mengetahui apakah gambar X-Ray masukan dari pengguna merupakan gambar Xray yang terinfeksi COVID atau bukan. Proses pendeteksian gambar dapat dilihat pada Gbr 1b.



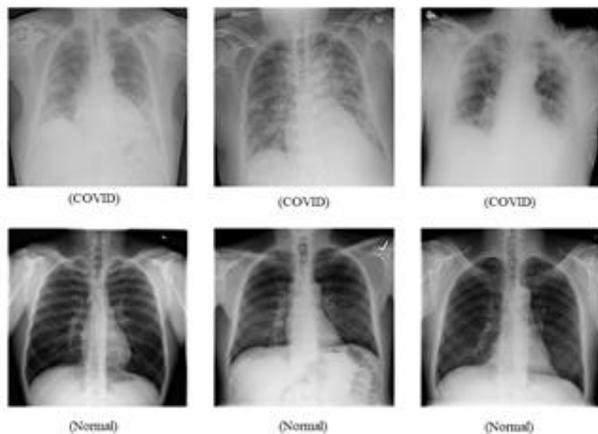
Gbr. 1a Diagram Alir Penelitian



Gbr. 1b Proses Deteksi gambar X-Ray Pada Aplikasi

A. Pengumpulan Data

Dataset yang digunakan pada penelitian ini merupakan *dataset* yang bersifat *public* yang didapat dari situs kaggle. *Dataset* ini berisi citra X-Ray dengan 3 label kelas masing-masing COVID berjumlah 3616 gambar, Normal sejumlah 10192 gambar, dan *Pneumonia* sejumlah 1345 gambar. Semua gambar memiliki format PNG dan ukuran yang sama yaitu 299×299 piksel. Namun pada penelitian ini *dataset* yang digunakan hanya dua kelas yaitu COVID dan Normal masing-masing sejumlah 2000 gambar. Beberapa contoh gambar pada *dataset* masing-masing kelas ditunjukkan pada Gbr. 2.



Gbr. 2 Beberapa gambar X-Ray pada dataset

B. Preprocessing Data

Proses preprocessing merupakan proses menyiapkan data sebelum diolah. Pada penelitian ini data yang diperoleh akan dilakukan pelabelan dan penurunan resolusi.

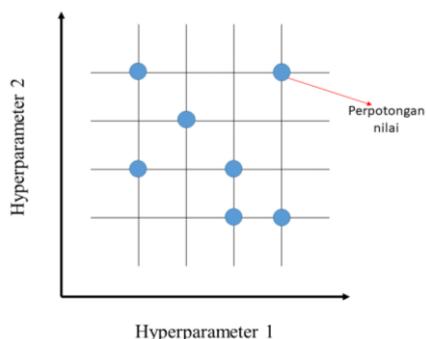
1) *Pelabelan Data*: Pelabelan gambar dilakukan untuk memberi label pada dataset gambar sesuai kelasnya yaitu COVID dan Normal. Pada penelitian ini pelabelan dilakukan secara manual dengan menggunakan nama folder yang tertera pada dataset yang sudah diperoleh.

2) *Penurunan Resolusi*: Penurunan resolusi atau biasa disebut dengan *resize* adalah yang paling umum digunakan dalam model klasifikasi. Alasan utama dilakukan penurunan resolusi yaitu diperlukan input dalam ukuran sama pada CNN, keterbatasan memori pada proses pelatihan CNN dengan resolusi tinggi sehingga akan memperlambat proses pembelajaran dan mempengaruhi hasil keakuratan model. Data gambar X-Ray pada penelitian ini berukuran 299x299 piksel akan dilakukan proses *resize* sehingga menjadi 150x150 piksel.

Dataset yang telah dilakukan preprocessing maka selanjutnya digunakan sebagai nilai parameter untuk proses *Random Search* dimana dalam algoritma tersebut kemudian digunakan sebagai lapisan input model CNN.

C. Perancangan Random Search

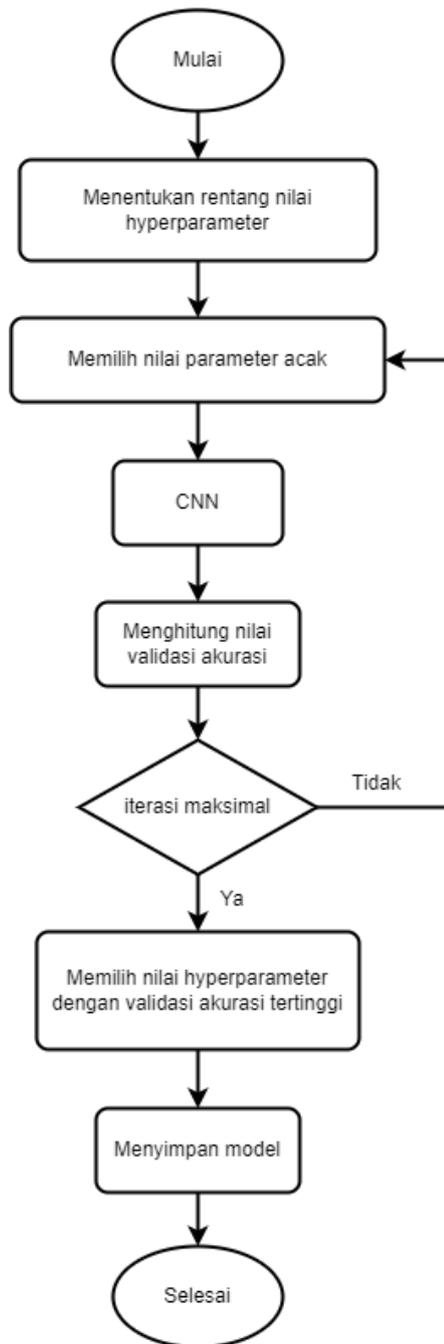
Algoritma *Random Search* pada penelitian ini akan melakukan pencarian terhadap semua kombinasi *hyperparameter* yang sudah ditentukan secara acak sehingga bisa mengurangi waktu yang diperlukan untuk proses pembelajaran CNN. Ilustrasi dari algoritma *Random Search*



Gbr. 3 Ilustrasi Pengambilan Nilai Random Search

dapat dilihat pada Gbr. 3.

Random Search hanya akan memilih beberapa *point* dari keseluruhan secara *random*. Proses ini sangat berguna apabila kita mempunyai beberapa *hyperparameter* yang akan dilakukan optimisasi. Meskipun *point* yang terpilih bisa jadi bukan merupakan *point* terbaik, tetapi hal itu sudah bisa untuk membangun model CNN dengan performa yang cukup bagus. Alur kerja dari algoritma *Random Search* ditunjukkan pada Gbr. 4.



Gbr. 4 Diagram Alir Random Search

Setiap layer CNN terdapat beberapa parameter penting yang akan dilakukan optimisasi menggunakan *Random Search*. Proses optimisasi ini akan dilakukan sebanyak 50 kali untuk mendapatkan kombinasi *hyperparameter* yang memiliki nilai *validation accuracy* tertinggi. Besarnya nilai tiap-tiap *hyperparameter* ditunjukkan pada tabel 1.

TABEL I
 TABEL HYPERPARAMETER

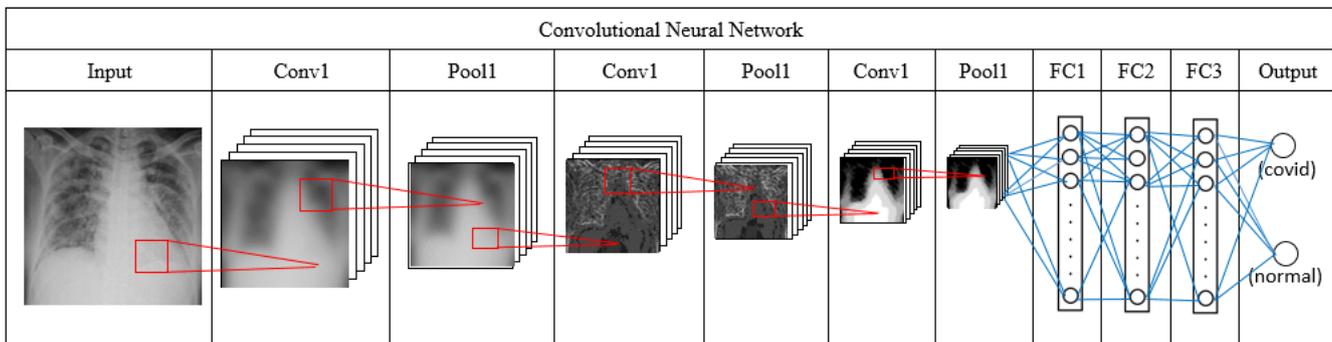
Layer	Fitur
Convolution layer(3)	Filter [32-256]
Kernel size	Kernel [3, 5]
Fully connected layer 1	Filter [32-512]
Fully connected layer 2	Filter [32-256]
Fully connected layer 3	Filter [32-128]
Learning rate	[0.01, 0.001, 0.0001]

D. Convolutional Neural Network (CNN)

Proses setelah *tuning* menggunakan *Random Search* akan didapatkan hasil berupa nilai-nilai *hyperparameter* yang akan digunakan untuk membangun model klasifikasi CNN. CNN sendiri merupakan salah satu bentuk dari *feed forward network* yang sering digunakan untuk model dengan input berupa data gambar atau video. Arsitektur CNN pada umumnya terdiri dari dua bagian besar yaitu *Feature Extraction Layer* dan *Fully-Connected Layer*.

Dalam *Feature Extraction Layer* gambar input akan di-*encoding* menghasilkan angka yang merepresentasikan gambar. Bagian ini terdiri dari *Convolutional Layer* dan *Pooling Layer*. Sedangkan *Fully Connected Layer* semua nilai yang dihasilkan pada *neuron* sebelumnya akan dihubungkan ke lapisan selanjutnya.

Perancangan dilakukan menggunakan platform *google collaboratory*. *Dataset* yang sudah diperoleh sebelumnya akan diupload ke dalam *server google drive* agar lebih mudah untuk proses input ke dalam model CNN. Desain arsitektur CNN ditunjukkan pada Gbr. 5.



Gbr. 5 Arsitektur Convolutional Neural Network

Pada arsitektur CNN yang penulis rancang terdiri dari tiga lapisan konvolusi (*Convolutional layer*) dan Lapisan *Fully Connected Layer* berupa tiga lapisan *Hidden layer*. Proses pertama dimulai dari masukan berupa gambar X-Ray.

Tahap selanjutnya adalah tahap dimana gambar akan melalui proses konvolusi dengan menggerakkan sebuah kernel konvolusi (*filter*) berukuran tertentu ke sebuah gambar, hasil dari proses konvolusi berupa informasi representatif baru dari hasil perkalian bagian gambar masukan dengan *filter* yang digunakan. Selanjutnya informasi representatif tersebut dilakukan *reshape (flatten)* agar menjadi sebuah *vector* supaya bisa digunakan sebagai input dari *fully-connected layer*.

Pada tahap *fully connected layer* semua *neuron* dari proses sebelumnya akan dihubungkan ke *neuron* lapisan selanjutnya. Setelah melalui *fully connected layer* gambar masukan akan dilakukan proses klasifikasi menjadi dua kelas menggunakan

fungsi aktivasi *sigmoid*. Proses training dalam tahap ini dilakukan sebanyak 50 *epoch*.

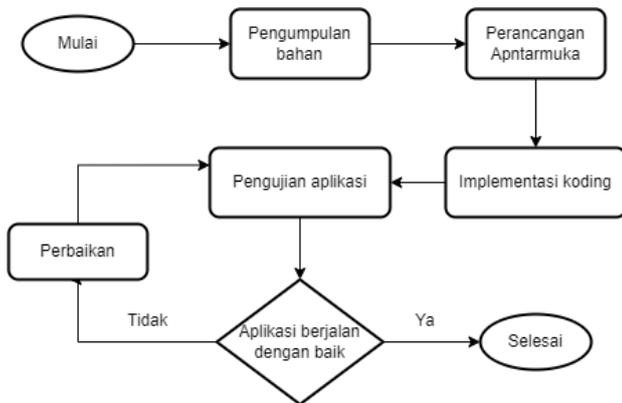
E. Evaluasi

Setelah proses pembelajaran dengan data *hyperparameter* yang telah dilakukan optimisasi, langkah selanjutnya adalah melakukan evaluasi terhadap data tes. Data tes merupakan data yang digunakan untuk proses evaluasi yaitu sebanyak 20% dari jumlah *dataset* keseluruhan. Hasil dari proses ini adalah nilai *accuracy* dan *loss*.

F. Perancangan Aplikasi

Proses perancangan aplikasi dilakukan setelah kita mendapatkan model CNN yang sudah optimal. Model akan disimpan kedalam format h5 sehingga nantinya bisa langsung dilakukan *import* ke dalam aplikasi dan melakukan klasifikasi terhadap input citra. Pada penelitian ini *framework* yang

digunakan adalah Flask. Flask adalah salah satu *framework* atau kerangka yang menghubungkan Python dengan aplikasi web. Proses yang akan dilakukan pada perancangan aplikasi ditunjukkan pada Gbr 6.



Gbr. 6 Diagram Alir Perancangan Aplikasi

III. HASIL DAN PEMBAHASAN

Hasil dan pembahasan berisi beberapa informasi yaitu hasil dari tahap perancangan Random Search dimana berisi nilai-nilai hyperparameter yang telah dilakukan optimasi. Hasil dari tahap *training* CNN dimana akan berisi hasil *training* dari beberapa *epoch* yang sudah dilakukan. Tahap evaluasi berisi tentang grafik akurasi dan loss selama *training* berlangsung. Tahap terakhir yaitu tampilan aplikasi deteksi COVID dari citra X-Ray.

A. Optimisasi Hyperparameter

Proses optimisasi menghasilkan kombinasi nilai-nilai *hyperparameter* dengan nilai akurasi terbaik. Pada tahapan ini *Random Search* akan mengambil satu kombinasi terbaik berdasarkan nilai validasi akurasi yang didapatkan dari proses *searching*. Adapun pencarian nilai *hyperparameter* dilakukan sebanyak 50 kali dengan memakan waktu sekitar 65 menit. Hasil dari optimisasi *hyperparameter* menggunakan *Random Search* ditunjukkan pada tabel 2.

TABEL II

HYPERPARAMETER TERBAIK

No	Conv1	Kernel1	Conv2	Kernel2	Conv3	Kernel3	FC1	FC2	FC3	Learning rate	Validasi akurasi	Validasi Loss
1	48	5	256	5	160	5	240	128	96	0.001	0.92188	0.20874
2	224	3	48	3	240	5	272	240	112	0.001	0.9125	0.20941
3	240	5	192	5	96	3	64	48	96	0.001	0.9125	0.20331
4	96	5	160	5	224	3	144	144	64	0.001	0.90469	0.23765
5	192	3	176	5	96	5	96	192	64	0.001	0.89844	0.25893
...
46	224	5	96	3	112	5	144	176	64	0.01	0.51719	0.69294
47	64	5	144	5	176	5	272	192	64	0.01	0.51719	0.69291
48	192	5	176	5	96	3	208	160	48	0.01	0.51719	0.69283
49	192	3	144	5	240	5	224	208	48	0.01	0.51719	0.69274
50	64	3	48	3	224	5	416	128	96	0.01	0.51719	0.69299

B. Training CNN

Nilai hyperparameter yang sudah didapat kemudian dilakukan proses *re-train* dengan epoch sebanyak 50. Hasil proses *re-train* dalam 10 epoch terakhir untuk hyperparameter dengan nilai validasi akurasi tertinggi dapat dilihat pada tabel 3.

TABEL III

PROSES TRAINING 10 EPOCH TERAKHIR MODEL TERBAIK

Epoch	Akurasi training	Loss training	Akurasi validasi	Loss validasi
41	0.9718	0.0872	0.9093	0.2198
42	0.9671	0.0920	0.9281	0.1837

43	0.9726	0.0808	0.9234	0.1795
44	0.9742	0.0739	0.9359	0.1750
45	0.9746	0.0765	0.9250	0.1997
46	0.9753	0.0737	0.9328	0.1785
47	0.9796	0.0664	0.9296	0.1842
48	0.9757	0.0701	0.9171	0.2321
49	0.9773	0.0718	0.9328	0.2022
50	0.9808	0.0643	0.9265	0.1942

Hyperparameter dengan nilai validasi akurasi tertinggi memberikan hasil yang baik dimana nilai akurasi *training* selama 10 epoch terakhir hampir mendekati sempurna berkisar antara 96%-98% dan nilai *loss* yang diperoleh sangat rendah yaitu 6%-8%. Detail arsitektur untuk model CNN

dengan nilai validasi akurasi tertinggi dapat dilihat pada tabel 4.

TABEL IV
DETAIL ARSITEKTUR CNN AKURASI TERTINGGI

Layer	Fitur	Parameter Pembelajaran
Conv2D_1	146 × 146 × 48	3648
MaxPooling2D_1	48 × 48 × 48	0
Conv2D_2	44 × 44 × 256	307456
MaxPooling2D_2	14 × 14 × 256	0
Conv2D_3	10 × 10 × 160	1024160
MaxPooling2D_3	3 × 3 × 160	0
Fully connected layer(Dense_1)	240	345840
Fully connected layer(Dense_2)	128	30848
Fully connected layer(Dense_3)	96	12384
Binary Crossentropy	1	97

Hasil proses re-train dalam 10 epoch terakhir untuk hyperparameter dengan nilai validasi terendah dapat dilihat pada tabel 5.

TABEL V
PROSES TRAINING 10 EPOCH TERAKHIR MODEL TERENDAH

Epoch	Akurasi training	Loss training	Akurasi validasi	Loss validasi
41	0.9605	0.1028	0.8969	0.3879
42	0.9648	0.0918	0.9031	0.3653
43	0.9703	0.0824	0.9047	0.3298
44	0.9648	0.0925	0.8953	0.3366
45	0.9676	0.0844	0.9078	0.3696
46	0.9555	0.1065	0.9047	0.3314
47	0.9406	0.1535	0.8875	0.3255
48	0.9633	0.0989	0.8891	0.3445
49	0.9645	0.0886	0.8969	0.4159
50	0.9652	0.0795	0.9031	0.4221

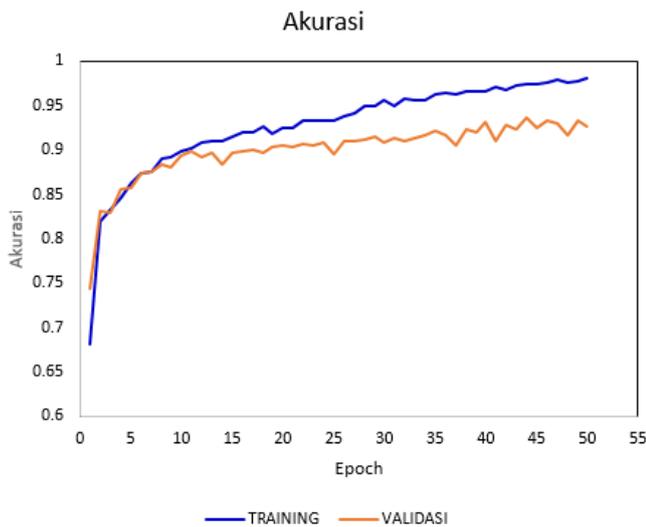
Pada hyperparameter dengan nilai validasi akurasi terendah memberikan hasil yang kurang baik. Nilai akurasi yang didapat selama 10 epoch terakhir cukup besar antara 95%-97% namun pada nilai loss validasi sangat besar mencapai 42%. Detail arsitektur untuk model CNN dengan nilai validasi akurasi paling rendah dapat dilihat pada tabel 6.

TABEL IV
HASIL OPTIMISASI PARAMETER ARSITEKTUR CNN AKURASI PALING RENDAH

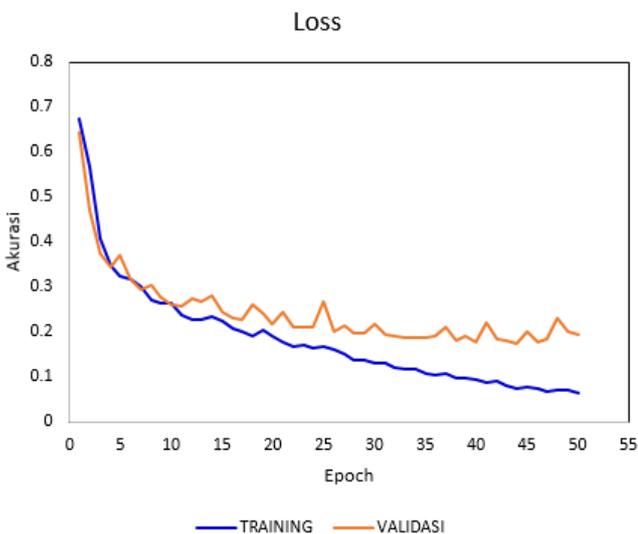
Layer	Fitur	Parameter Pembelajaran
Conv2D_1	148 × 148 × 64	1792
MaxPooling2D_1	49 × 49 × 64	0
Conv2D_2	47 × 47 × 48	27696
MaxPooling2D_2	15 × 15 × 48	0

Conv2D_3	11 × 11 × 224	269024
MaxPooling2D_3	3 × 3 × 224	0
Fully connected layer(Dense_1)	416	839072
Fully connected layer(Dense_2)	128	53376
Fully connected layer(Dense_3)	96	12384
Binary Crossentropy	1	97

Hasil yang diperoleh dari proses training pada model CNN dengan akurasi tertinggi dan terendah menunjukkan hasil yang berbeda. Model CNN dengan akurasi tertinggi memiliki performa yang baik dapat dilihat pada nilai akurasi yang terus bertambah hingga mendekati sempurna selama proses *training* sebanyak 50 epoch dan nilai loss yang terus berkurang hingga menjadi sangat kecil begitu juga dengan proses selama validasi. Model CNN terbaik dapat mengklasifikasikan gambar X-Ray dengan baik. Grafik nilai akurasi selama proses *training* dan validasi dapat dilihat pada Gbr 7a. Grafik nilai *loss* selama proses *training* dan validasi dapat dilihat pada Gbr 7b.

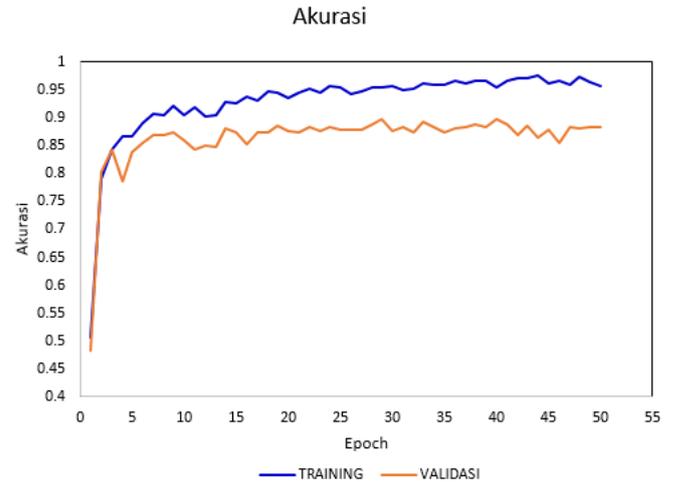


Gbr. 7a Grafik Akurasi Training dan Validasi Model Terbaik

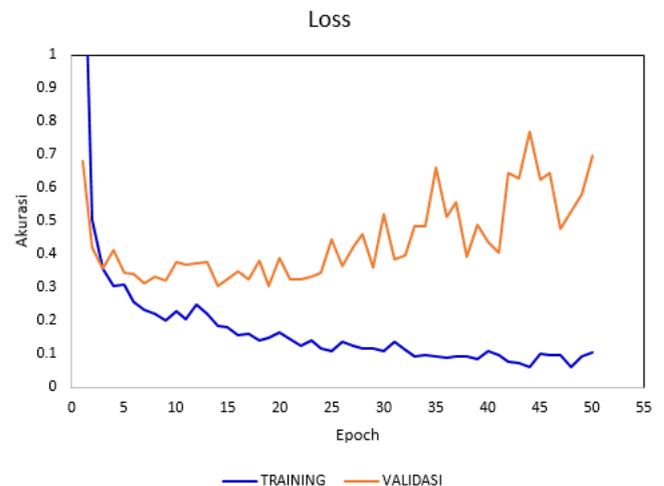


Gbr. 7b Grafik Loss Training dan Validasi Model Terbaik

Model CNN dengan akurasi paling rendah mempunyai akurasi yang cukup bagus pada proses *training*. Namun pada proses validasi mempunyai nilai yang lebih rendah dan nilai *loss* validasi yang cukup besar. Model CNN terendah kurang dapat mengklasifikasikan gambar X-Ray dengan baik. Grafik nilai akurasi selama proses *training* dan validasi dapat dilihat pada Gbr 8a. Grafik nilai *loss* selama proses *training* dan validasi dapat dilihat pada Gbr 8b.



Gbr. 8a Grafik Akurasi Training dan Validasi Model Terendah



Gbr. 8b Grafik Loss Training dan Validasi Model Terbaik

C. Evaluasi

Tahap evaluasi model CNN dengan nilai akurasi tertinggi dan terendah akan dilakukan evaluasi terhadap data tes. Data tes yang digunakan untuk proses evaluasi sebesar 20% dari keseluruhan. Model CNN terbaik dilakukan evaluasi dan mendapatkan nilai akurasi sebesar 95.38% dan nilai *loss* sebesar 47.08%. Model CNN terendah dilakukan evaluasi dan menghasilkan nilai akurasi sebesar 88.00% dan nilai *loss* sebesar 56.65%

D. Implementasi Aplikasi

Proses pengembangan aplikasi dilakukan menggunakan *framework* Flask. Aplikasi ini dijalankan pada server lokal dan dapat diakses dengan *url* <http://127.0.0.1:5000/>. Tampilan awal aplikasi web untuk Klasifikasi Citra X-Ray Dada ditunjukkan pada Gbr. 9.



Gbr. 9 Tampilan Utama Aplikasi

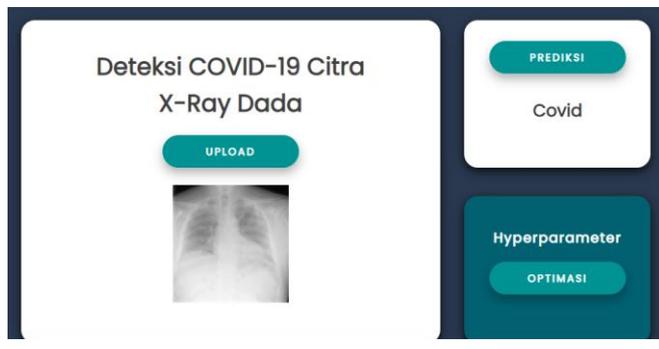
Tiga fitur utama pada aplikasi ini sebagai berikut:

1) *Upload*: Fungsi untuk memasukkan gambar input (X-Ray) yang berada di penyimpanan komputer. Gambar inputan tersebut akan ditampilkan pada layar.

2) *Output*: Fungsi untuk melakukan proses klasifikasi gambar inputan apakah termasuk kedalam kelas COVID atau Normal kemudian menampilkan hasil yang diperoleh.

3) *Optimasi*: Fungsi untuk menampilkan tabel nilai-nilai dari hyperparameter dengan hasil akurasi terbaik.

Tampilan proses klasifikasi gambar X-Ray untuk klasifikasi apakah termasuk kedalam kelas COVID atau Normal dapat dilihat pada Gbr 10.



Gbr. 10 Tampilan Proses Klasifikasi Terhadap Citra X-Ray

Fungsi optimasi pada tombol di dalam aplikasi dapat dilihat pada Gbr. 11 dimana terdapat jenis-jenis *hyperparameter* dan juga nilai setiap parameter.

Hyperparameter	Nilai
Conv2d Layer 1	48
Maxpool 1	5
Conv2d Layer 2	256
Maxpool 2	5
Conv2d Layer 3	160
Maxpool 3	5
Fully Connected 1	240
Fully Connected 2	128
Fully Connected 3	96
Learning rate	0.001

Gbr. 11 Hyperparameter Optimal pada Aplikasi

IV. KESIMPULAN

Penelitian yang dilakukan oleh Irmak, dkk. menggunakan CNN dengan menerapkan optimasi *hyperparameter* menggunakan *Grid Search* dimana pada arsitektur yang dibangun terdiri dari 12 layer dan menghasilkan nilai akurasi sebesar 98.92%. Pada penelitian yang penulis lakukan berjudul Optimisasi *Hyperparameter* CNN Menggunakan *Random Search* Untuk Deteksi COVID-19 Dari Citra X-Ray Dada, menggunakan 10 jenis layer CNN pada arsitektur modelnya dan mendapatkan hasil akurasi tertinggi sebesar 95.38% dan akurasi terendah sebesar 88%. Akurasi yang didapatkan pada penelitian ini lebih rendah daripada penelitian sebelumnya namun pada model yang didapat pada penelitian ini lebih menghemat waktu untuk melakukan komputasi. *Grid Search* akan melakukan perhitungan terhadap semua kombinasi *hyperparameter* yang ada sehingga akan memakan waktu yang lebih lama dalam mendapatkan model dengan akurasi terbaik. Sementara pada penelitian yang penulis rancang yaitu menggunakan *Random Search* dimana akan mengambil beberapa kombinasi *hyperparameter* saja sehingga akan membutuhkan waktu lebih sedikit untuk mendapatkan model dengan akurasi tertinggi dimana dalam mendapatkan model dengan akurasi terbaik sebesar 95.38% diperlukan waktu kurang lebih 65 menit.

Hasil akurasi dari model tertinggi sebesar 95.38% dan nilai *loss* yaitu 47.08%. Nilai validasi akurasi dan *loss* validasi selama proses training mendapatkan hasil yang bagus. Nilai validasi akurasi yang dihasilkan sebesar 92.18% sedangkan nilai *loss* validasi sebesar 20.87% yang menunjukkan bahwa model CNN yang dibangun bisa melakukan klasifikasi gambar X-Ray Normal dan COVID dengan baik.

V. SARAN

Proses optimisasi model CNN pada penelitian ini melibatkan empat *hyperparameter* akan tetapi hal ini tidak mengurangi kemungkinan pada penelitian selanjutnya untuk menambah jenis *hyperparameter* beserta jumlah perulangan yang digunakan.

Proses pembuatan model CNN diharapkan pada penelitian selanjutnya untuk menambah jumlah kelas data, jumlah *dataset*, dan jenis *layer* agar tidak terjadi *overfitting*. Proses pengembangan aplikasi bisa ditambahkan beberapa nilai *hyperparameter* teratas yang memiliki nilai akurasi tertinggi.

VI. UCAPAN TERIMA KASIH

Ucapan puji syukur saya panjatkan kepada Allah SWT Tuhan semesta alam yang senantiasa memberikan pertolongan untuk menyelesaikan penelitian ini. Ucapan terima kasih saya persembahkan kepada orang tua yang selalu memberikan dukungan untuk menyelesaikan penelitian ini. Saya ucapkan terima kasih kepada dosen pembimbing yang selalu memberi masukan dan arahan hingga penelitian ini berakhir. Tidak lupa terima kasih kepada teman-teman yang selalu memberi dukungan berupa semangat maupun saran kepada saya selama proses penelitian ini.

REFERENSI

- [1] Akram, T., Attique, M., Gul, S. et al. A novel framework for rapid diagnosis of COVID-19 on computed tomography scans. *Pattern Analysis and Applications* (2021).
- [2] Turkoglu, Muammer. "COVIDetectioNet: COVID-19 diagnosis system based on X-ray images using features selected from pre-learned deep features ensemble." *Applied Intelligence* (2021): 1 - 14.
- [3] Worldometer Coronavirus.
<https://www.worldometers.info/coronavirus/>. Diakses 10 Juni 2021
- [4] Wang, D., Hu, B., Hu, C., Zhu, F., Liu, X., Zhang, J., ... Peng, Z. (2020). Clinical Characteristics of 138 Hospitalized Patients with 2019 Novel Coronavirus-Infected Pneumonia in Wuhan, China. *JAMA - Journal of the American Medical Association*, 323(11), 1061–1069
- [5] Singh, D., Kumar, V., Vaishali, & Kaur, M. (2020). Classification of COVID-19 patients from chest CT images using multi-objective differential evolution-based convolutional neural networks. *European journal of clinical microbiology & infectious diseases* : official publication of the European Society of Clinical Microbiology, 39(7), 1379–1389. <https://doi.org/10.1007/s10096-020-03901-z>
- [6] Irmak E. Implementation of convolutional neural network approach for COVID-19 disease detection. *Physiol Genomics*. 2020 Dec 1;52(12):590-601. doi: 10.1152/physiolgenomics.00084.2020. Epub 2020 Oct 23. PMID: 33094700; PMCID: PMC7774002.
- [7] Zein, A. (2020). Pendeteksian Virus Corona Dalam Gambar X-Ray Menggunakan Algoritma Artificial Intelligence Dengan Deep Learning Python. *Teknik Informatika STMIK Eresha*.
- [8] Sethy, P.K.; Behera, S.K. Detection of Coronavirus Disease (COVID-19) Based on Deep Features. *Preprints 2020*, 2020030300 (doi: 10.20944/preprints202003.0300.v1).
- [9] Hariyani, Y.S., Hadiyoso, S., & Siadari, T.S. (2020). Deteksi Penyakit Covid-19 Berdasarkan Citra X-Ray Menggunakan Deep Residual Network.
- [10] Ji M, Yuan L, Shen W, Lv J, Li Y, Li M, Lu X, Hu L, Dong W (2020). Characteristics of disease progress in patients with coronavirus disease 2019 in Wuhan, China. *Epidemiology and Infection* 148, e94, 1–10. <https://doi.org/10.1017/S0950268820000977>
- [11] Al-Waisy, A.S., Al-Fahdawi, S., Mohammed, M.A. et al. COVID-CheXNet: hybrid deep learning framework for identifying COVID-19 virus in chest X-rays images. *Soft Comput* (2020). <https://doi.org/10.1007/s00500-020-05424-3>
- [12] Gavin, B., Lyne, J., & McNicholas, F. (2020). Mental health and the COVID-19 pandemic. *Irish Journal of Psychological Medicine*, 37(3), 156-158. doi:10.1017/ipm.2020.72
- [13] Zhou, Z., Zhou, H., Lin, X., SU, Z., Zhao, L., & Chen, X. (2020). Outbreak of COVID-19 in a family, Wenzhou, China. *Epidemiology and Infection*, 148, E103. doi:10.1017/S0950268820001089
- [14] Liashchynskiy, P., & Liashchynskiy, P. (2019). Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. *ArXiv*, abs/1912.06059.
- [15] Soumya Ranjan Nayak, Deepak Ranjan Nayak, Utkarsh Sinha, Vaibhav Arora, Ram Bilas Pachori, Application of deep learning techniques for detection of COVID-19 cases using chest X-ray images: A comprehensive study, *Biomedical Signal Processing and Control*, Volume 64, 2021, 102365, ISSN 1746-8094, <https://doi.org/10.1016/j.bspc.2020.102365>.
- [16] Santoshi Kumari, Ediga Ranjith, Abhishek Gujjar, Siranjeevi Narasimman, H S Aadil Sha Zeelani, Comparative Analysis of Deep Learning Models For COVID-19 Detection, *Global Transitions Proceedings*, 2021, ISSN 2666-285X, <https://doi.org/10.1016/j.gtp.2021.08.030>.