

'Klasifikasi Gerakan Cuci Tangan Berbasis *Convolutional Neural Network (CNN)*

Habibul Rahman Qalbi, Eko Mulyanto Yuniarmo, dan Reza Fuad Rachmadi
Departemen Teknik Komputer, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: ekomulyanto@ee.its.ac.id

Abstrak—Cuci tangan merupakan langkah awal dalam menjaga kesehatan tubuh. Dengan mencuci tangan kita dapat mencegah penyebaran penyakit. Akan tetapi, masih banyak masyarakat yang tidak sadar akan tata cara mencuci tangan yang baik, sehingga tidak bersih sepenuhnya. Pemanfaatan teknologi *Deep Learning* dapat menjadi solusi untuk mengetahui apakah masyarakat telah mencuci tangan dengan benar. Menggunakan kamera sebagai input yang kemudian di proses menggunakan algoritma *Convolutional Neural Network (CNN)* kita dapat mengklasifikasikan gerakan - gerakan yang dilakukan pengguna saat mencuci tangan. Metode yang digunakan dalam penelitian ini menghasilkan tingkat akurasi 99% pada fase *Training*. Pada pengujiannya, sistem ini menghasilkan tingkat akurasi prediksi tertinggi sebesar 12/12 gerakan (100%) dengan tingkat inkonsistensi sebesar 2/12 prediksi (16,67%). Harapannya penelitian ini dapat membantu dalam memantau dan memastikan apakah masyarakat mencuci tangan dengan benar, khususnya di tempat umum dimana tingkat penyebaran penyakit cukup tinggi.

Kata Kunci—*CNN*, Cuci Tangan, *EfficientNet*, Klasifikasi, *Moving Average*.

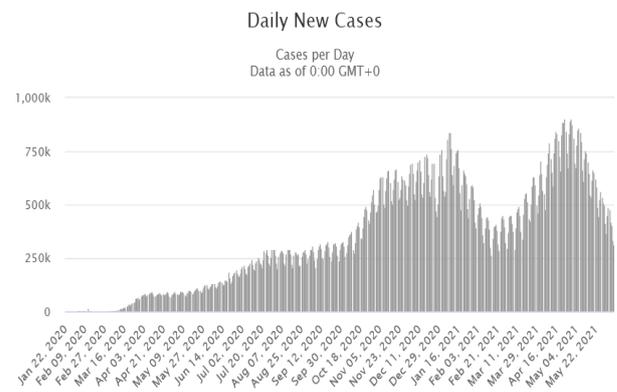
I. PENDAHULUAN

MENJAGA kebersihan merupakan salah satu kunci untuk menjaga kesehatan tubuh. Tubuh yang bersih dapat menghindarkan kita dari berbagai penyakit terutama yang berasal dari lingkungan di sekitar. Kita sering kali menyentuh benda - benda di sekitar seperti gadget, komputer, gagang pintu, meja, lemari, dan lain sebagainya. Akan tetapi, banyak yang tidak sadar bahwa benda-benda tersebut sering kali menjadi sarang bagi bakteri & virus yang dapat menyebabkan penyakit.

Ketika menyentuh benda-benda tersebut, kuman-kuman yang ada akan berpindah dan menyebar di kulit tangan, kemudian ketika menyentuh orang lain, maka kuman penyebab penyakit itu akan berpindah dan menyebar ke tubuh orang tersebut. Pada akhirnya ketika menyentuh bagian - bagian di tubuh seperti mata, hidung dan mulut, maka kuman tersebut akan masuk ke tubuh kita dan menyebabkan penyakit. Oleh karena itu mencuci tangan merupakan hal yang paling utama dalam menjaga kesehatan tubuh dan mencegah penyebaran penyakit.

Pada saat penelitian ini berlangsung, 2020 - 2021, sedang marak terjadinya pandemi virus *SARS-CoV-2 (Severe Acute Respiratory Syndrome - Corona Virus - 2)* atau yang biasa disebut *COVID-19 (Corona Virus Disease - 2019)*. Virus ini menyerang sistem pernafasan yang menyebabkan pengidapnya kesulitan bernafas, hingga kematian. Adapun jumlah kasus kenaikan harian dari varian jenis virus ini diperlihatkan pada Gambar 1.

Virus ini sangat mudah untuk menyebar terutama melalui *droplets* (percikan air) yang terjadi ketika bersin, batuk, dan bicara sekalipun. *Droplets* ini akan menempel ke benda -



Gambar 1. Kasus harian COVID-19.

benda dan tubuh kita. Hanya dengan menyentuh benda atau bagian tubuh yang terkontaminasi kemudian tanpa sadar menyentuh mata, hi-dung ataupun mulut, maka virus tersebut akan masuk ke tubuh dan merusak sel paru-paru yang akhirnya menyebabkan gangguan pernafasan bahkan kematian [1].

Walaupun kebiasaan cuci tangan meningkat selama pandemi COVID-19, banyak orang yang masih *asal-asalan* ketika mencuci tangan lantaran terburu-buru maupun hal lainnya. Menurut laman hallosehat.com, masih banyak kesalahan dalam mencuci tangan yang sering dilakukan oleh masyarakat, di antaranya waktu mencuci tangan yang terlalu cepat dan kebiasaan masyarakat yang hanya menggosok telapak tangan saja ketika sedang mencuci tangan. Akan tetapi, tidak memungkinkan untuk dilakukannya pemantauan 7x24 jam oleh manusia terhadap pengunjung di tempat umum, seperti *mall* dan tempat wisata lainnya demi memastikan pengunjung telah mencuci tangan dengan benar, terutama di masa pandemi COVID-19 ini yang mana masyarakat dihimbau untuk menjaga jarak untuk mencegah penularan.

Pada 2007, D. F. Llorca *et al* mencoba sebuah metode untuk melacak (*tracking*) gerakan tangan saat seseorang sedang mencuci tangan [2]. menggunakan *Particle Filter (PF)* yang dikombinasikan dengan *k-means Based Clustering*, *outputnya* adalah sebuah *elliptical* model yang merupakan hasil segmentasi dari kedua tangan. Hasil segmentasi tersebut kemudian dimasukkan ke dalam sebuah algoritma supervisi yang mengukur jumlah objek yang berhasil dilacak (*tracking*) dan kualitas dari hasil tracking itu sendiri. Performa dari penggunaan PF pada model ini kemudian dibandingkan dengan penggunaan *Kalman Filter (KF)*. Dari perbandingan ini ditemukan bahwa PF dapat mendeteksi objek tangan yang bergerak dengan cepat, sedangkan KF hanya dapat mendeteksi objek tangan yang bergerak lambat.

Pada 2011 D. F. Llorca *et al* kembali menyempurnakan metode yang mereka buat sebelumnya dan membuat sebuah

Tabel 1.
Isi dataset Kaggle

No.	Label	Jumlah
1	<i>Step_1</i>	25
2	<i>Step_2_Left</i>	25
3	<i>Step_2_Right</i>	25
4	<i>Step_3</i>	25
5	<i>Step_4_Left</i>	25
6	<i>Step_4_Right</i>	25
7	<i>Step_5_Left</i>	25
8	<i>Step_5_Right</i>	25
9	<i>Step_6_Left</i>	25
10	<i>Step_6_Right</i>	25
11	<i>Step_7_Left</i>	25
12	<i>Step_7_Right</i>	25

sistem berbasis citra (*vision-based*) untuk penilaian kualitas cuci tangan yang kini disebut sebagai *project SureWash* [3].

Sistem ini mengimplementasikan *single multi-modal particle filter (PF)* dan *k-means-based clustering* yang mereka buat sebelumnya pada 2007 [2]. kemudian diklasifikasikan dengan *Multi-class SVM*. Sistem tersebut menunjukkan tingkat akurasi setara ahli kesehatan.

Pada 2011, S. Ameling *et al* membuat sebuah sistem pendeteksi kualitas cuci tangan dengan *realtime feedback*. Sistem ini menggunakan metode *Skin Filtering* dan *ROI Pose detection* untuk mengekstrak bagian tangan dari keseluruhan citra. Dengan metode tersebut, sistem ini berhasil mencapai tingkat akurasi hingga 84,66% [4]. Pada 2017, D. Nunez-Fernandez dan B. Kwolek membuat sebuah sistem *Hand Posture Recognition* menggunakan *Convolutional Neural Network*. Sistem ini memanfaatkan dan membandingkan penggunaan *Skin Color Hand Segmentation* dengan penggunaan *Gabor Filter*. Ditemukan bahwa penggunaan *Gabor Filter* memberikan akurasi yang lebih baik (0.992) dibandingkan dengan penggunaan *Skin Color Hand Detection* (0.950) [5].

Pada tahun 2020, H. F. Hanafi *et al* menciptakan sebuah aplikasi berbasis *Augmented Reality* untuk memberikan petunjuk dan meningkatkan efektivitas mencuci tangan demi melawan wabah virus Corona. Sistem ini memanfaatkan *AR View*, *AR Hand Marker* dan *AR Preset Tracker* untuk mendeteksi sekaligus menunjukkan cara cuci tangan yang baik dan benar di *smartphone* pengguna. akan tetapi Aplikasi ini masih cukup sulit untuk digunakan oleh awam, sehingga masih dibutuhkan pengembangan lebih lanjut [6].

Di tahun yang sama, C. Wang *et al* melakukan studi instrumen penggunaan *Myo Armband* pada 20 partisipan mengenai tingkat akurasi deteksi kualitas cuci tangan. Ditemukan bahwa penggunaan Armband pada lengan bawah maupun penggunaan dua Armband sekaligus dapat meningkatkan akurasi pendeteksian [7].

Masih di Tahun 2020, E. Prakasa dan B. Sugiarto dari LIPI Bandung menciptakan sistem analisa video untuk mengevaluasi kelengkapan gerakan cuci tangan. Menggunakan CNN sederhana dan dengan implementasi *Binary Segmentation*, mereka melakukan klasifikasi pada video gerakan tangan dengan latar terkendali seperti *green screen*. Dari pengujian yang dilakukan, didapat akurasi sistem hingga maksimum 98.81% dengan *average precision* mencapai 98.89% [8].

Di tahun 2020, perusahaan teknologi asal jepang, Fujitsu, membuat sebuah teknologi berbasis AI (*Artificial*

Intelligence) untuk mengklasifikasikan gerakan cuci tangan. Sistem ini berbasis dari teknologi AI kamera CCTV mereka yaitu *Actlyzer behavioral analysis technology* yang dapat mendeteksi adanya pencurian. sistem ini dikembangkan lebih lanjut dengan menggunakan *High Durability Learning*.

Teknologi AI *Learning* milik Fujitsu sendiri. dengan cara tersebut, sistem ini berhasil mencapai tingkat akurasi klasifikasi hingga 95% pada 6 langkah cuci tangan yang benar dan tingkat akurasi penilaian dari 6 gerakan tersebut sebesar 90%.

II. DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini merupakan salah satu bentuk penerapan dari disiplin ilmu *Deep Learning* yang bertujuan untuk mengklasifikasikan gerakan cuci tangan menggunakan *Convolutional Neural Network (CNN)*

A. Pengumpulan Dataset Cuci Tangan

Dataset yang digunakan pada sistem klasifikasi ini berasal dari *Kaggle Hand Wash Dataset*. Sampel dari dataset ini dapat dilihat pada Gambar 2. Dataset ini berisi potongan-potongan video gerakan cuci tangan seperti yang tertera pada Tabel 1.

Pada dataset ini, masih terdapat kesalahan pelabelan pada video gerakan yang mana beberapa video gerakan ditempatkan pada folder yang tidak seharusnya. Terdapat pula kontaminasi gerakan berbeda pada beberapa video yang dapat menyebabkan kesalahan pada fase *training*. Untuk itu, dataset ini perlu diperbaiki terlebih dahulu sebelum digunakan.

Pada dataset ini, masing-masing label terdiri dari 25 gerakan yang terbagi atas 5 latar wastafel berbeda. Hal ini dirasa kurang mencukupi untuk menggeneralisasi model dalam mengklasifikasikan gerakan cuci tangan. Oleh karena itu, penulis melakukan penambahan dataset pribadi dengan latar berbeda.

Pengambilan dataset yang pertama yaitu menggunakan kamera *Webcam M-TECH WB500*, Sebuah webcam USB eksternal dengan resolusi *FullHD 1080p*. *Setup* pengambilan dan sampel video menggunakan webcam ini dapat dilihat pada Gambar 2 dan Gambar 3.

Pengambilan dataset pribadi yang kedua adalah dengan menggunakan Kamera *DSLR Canon EOS 100D* dengan lensa *Canon EFs 18-55 F/3.5 IS STM*. *Setup* pengambilan video dari DSLR ini dapat dilihat pada Gambar 4 dan Gambar 5.

Pada *Machine Learning*, sebelum melakukan proses algoritma, dataset dibagi menjadi tiga *subset* seperti pada Gambar 6. Pada umumnya, pembagian dataset memiliki komposisi sebagai berikut:

1. 80% Training Set, 10% Validation Set, dan 10% Test Set.
2. 70% Training Set, 20% Validation Set, dan 10% Test Set.
3. 80% TrainVal (80% Training + 20% Validation) Set, dan 10% Evaluation / Test Set.

Masing-masing pembagian dari dataset ini dapat dijelaskan sebagai berikut:

1) Training Set

Training Set merupakan bagian dari dataset yang digunakan pada proses *training* untuk melatih model dari sebuah algoritma *Machine Learning*.



Gambar 2. Setup pengambilan dengan webcam.



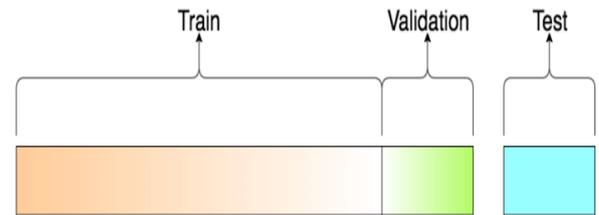
Gambar 3. Sampel hasil video webcam.



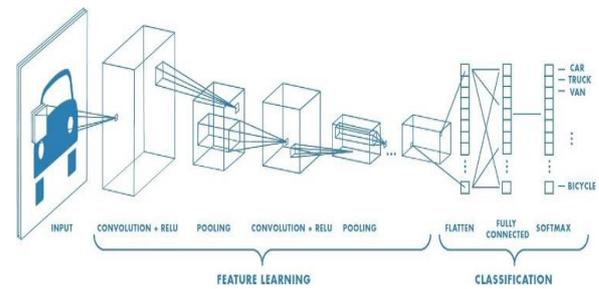
Gambar 4. Setup pengambilan dengan DSLR.



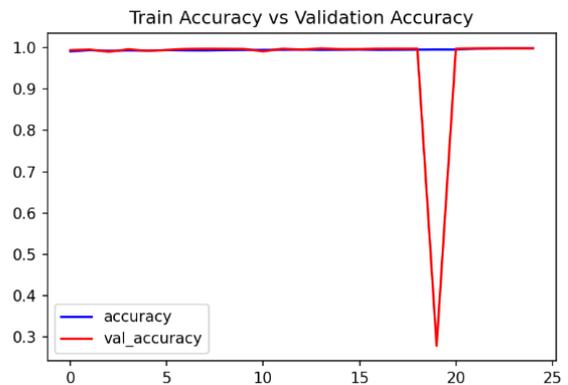
Gambar 5. Sampel hasil video DSLR.



Gambar 6. Visualisasi pembagian dataset.



Gambar 7. Visualisasi arsitektur CNN.



Gambar 8. Grafik nilai akurasi pada proses *training* dan *validation*.

2) *Validation Set*

Validation Set digunakan untuk melakukan validasi dan mencegah *overfitting*.

3) *Test atau Evaluation Set*

Test Set merupakan bagian dari dataset yang digunakan pada proses testing untuk menguji sebuah model.

B. *Frame Extraction*

Pada dasarnya, sebuah video adalah sekumpulan gambar (*Frame*) yang di satukan dan di tampilkan secara berurutan dalam satuan waktu yang biasa disebut *Frame Rate* dengan satuan *Frame-Per-Second (FPS)*. Misalnya sebuah video dengan *Frame Rate 30 FPS*, maka video tersebut sebenarnya adalah urutan gambar yang ditampilkan secara bergantian dengan kecepatan 30 gambar per detik. Semakin tinggi *FPS* video, semakin halus gerakan yang dihasilkan pada video tersebut.

Frame Extraction adalah proses ekstraksi *frame* dari

sebuah video menjadi sekumpulan gambar. Ini diperlukan karena CNN tidak mampu melakukan training secara langsung pada sebuah video. CNN Hanya dapat melakukan training pada setiap satuan gambar untuk dapat mengenali objek pada gambar tersebut.

C. *Pembuatan Arsitektur CNN*

Convolutional Neural Network (CNN) merupakan salah satu jenis *Neural Network* yang biasa digunakan untuk mengolah data dua dimensi. CNN biasa digunakan untuk mendeteksi dan mengenali suatu objek pada sebuah citra. Secara garis besar, CNN hampir sama dengan *Neural Network* lainnya. Hal yang membedakannya terletak pada arsitekturnya. CNN memiliki sebuah proses untuk mengekstraksi fitur dari sebuah gambar. Proses tersebut terdiri dari dua bagian, yaitu *Convolutional Layer* dan *Pooling Layer* seperti pada Gambar 7.

Tabel 2. Pembagian dataset

No.	Label	Total Video	n-Frame		
			Training	Validation	Test
1	Step_1	25	3840	960	1200
2	Step_2_Left	25	3840	960	1200
3	Step_2_Right	25	3840	960	1200
4	Step_3	25	3840	960	1200
5	Step_4_Left	25	3840	960	1200
6	Step_4_Right	25	3840	960	1200
7	Step_5_Left	25	3840	960	1200
8	Step_5_Right	25	3840	960	1200
9	Step_6_Left	25	3840	960	1200
10	Step_6_Right	25	3840	960	1200
11	Step_7_Left	25	3840	960	1200
12	Step_7_Right	25	3840	960	1200

Tabel 3. Spesifikasi arsitektur CNN yang digunakan

Layer (type)	Output Shape	Parameter
InputLayer	(None, 224, 224, 3)	0
Image_augmentation	(None, 224, 224, 3)	0
efficientnetb0	(None, 7, 7, 1280)	4049571
global_average_pooling_2d	(None, 1280)	0
batch_normalization	(None, 1280)	5120
dropout	(None, 1280)	0
dense	(None, 12)	15372

D. Proses Training

Sebelum melakukan proses *training*, perlu dilakukan penentuan ukuran *batch size*, *iteration* dan *epoch*.

1) Batch size

Batch size mendefinisikan banyaknya sampel yang akan disebarakan ke *neural network* dalam sekali iterasi. Pada proses ini, *batch size* yang digunakan sebesar 12.

2) Iteration

Iteration merupakan jumlah batch yang diperlukan untuk menyelesaikan satu *epoch*.

3) Epoch

Epoch adalah ketika seluruh *dataset* sudah melalui proses training pada *neural network* sampai dikembalikan ke awal (*Backpropagation*) untuk sekali putaran. Beberapa *epoch* diperlukan pada proses training agar mendapatkan kesalahan (*loss*) sekecil mungkin. Pada proses ini, *epoch* yang digunakan sebanyak 25 kali.

E. Proses Validation

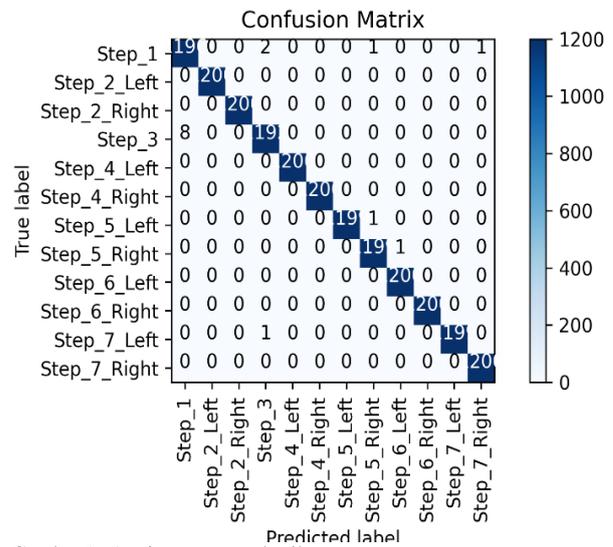
Proses *Validation* berjalan bersamaan dengan proses *training*. Pada setiap *epoch*, proses *training* dan *validation* dilakukan secara berurutan. Proses ini dilakukan untuk melihat perkembangan model dan mencegah *overfitting*. *Dataset* diproses ke *neural network* untuk menghitung *loss* tanpa melakukan *update bias* dan *weight*.

F. Proses Testing

Setelah proses *validation* dilakukan, langkah selanjutnya adalah proses *testing*. Proses ini merupakan sebuah proses untuk memprediksi hasil dari apa yang diinginkan melalui arsitektur CNN yang dibuat.

G. Moving Average

Setelah model berhasil mengklasifikasikan frame pada Video, masing masing frame pada video tersebut kemudian dimasukkan kedalam *Double-ended Queue (Deque)* sesuai



Gambar 9. Confusion matrix hasil testing.



Gambar 10. Sampel standard FOV.

urutannya dengan jumlah antrian (*Window Size*) yang ditentukan. Setelah mencapai *Window Size* yang ditentukan, hasil klasifikasi pada *Deque*, kemudian akan di ambil rata-ratanya. Hasil rata rata tersebut kemudian akan menjadi label prediksi baru, menimpa hasil prediksi sebelumnya, pada setiap frame tersebut. Setelah label prediksi baru di cantumkan, maka *Deque* akan kembali dikosongkan dan siap menerima antrian selanjutnya

H. Proses Output Video

Setelah mendapatkan label dari *Moving Average*, urutan urutan frame tersebut akan kembali dimasukkan kedalam sebuah array. Masing-masing label prediksi yang sudah didapatkan akan dituliskan pada masing masing frame. Akhirnya frame-frame tersebut disatukan kembali menjadi sebuah video output

III. PENGUJIAN DAN ANALISA

A. Pembagian Dataset

Total keseluruhan dataset yang didapatkan berjumlah 516 video atau 49 video pada setiap kelas. Agar tidak terjadi *overfitting* pada salah satu kelas, setiap video dibatasi durasinya kurang lebih ± 10 Detik. Masing-masing kelas kemudian akan di ekstrak sebanyak 6000 frame yang diambil secara acak. Pembagian dataset pada artikel ini menggunakan komposisi 80% *TrainVal* (80% *Training* & 20% *Validation*) *Set* dan 20% *Test Set*. Detail Pembagian dataset pada artikel ini dapat dilihat pada Tabel 2.

Tabel 4.
Hasil pengujian berdasarkan *window size*

Predicted Label	Predicted Label(s)		
	1 Second	2 Second	3 Second
Step 1	Step 1, Step 5 Right	Step 1	Step 1
Step 2 Left	Step 2 Left	Step 2 Left	Step 2 Left
Step 2 Right	Step 2 Right	Step 2 Right	Step 2 Right
Step 3	Step 3, Step 1	Step 3, Step 1	Step 3, Step 1
Step 4 Left	Step 4 Left	Step 4 Left	Step 4 Left
Step 4 Right	Step 4 Right	Step 4 Right	Step 4 Right
Step 5 Left	Step 5 Left	Step 5 Left	Step 5 Left
Step 5 Right	Step 5 Right	Step 5 Right	Step 5 Right
Step 6 Left	Step 6 Left	Step 6 Left	Step 6 Left
Step 6 Right	Step 6 Right	Step 6 Right	Step 6 Right
Step 7 Left	Step 7 Left	Step 7 Left	Step 7 Left
Step 7 Right	Step 7 Right, Step 5 Right	Step 7 Right, Step 5 Right	Step 7 Right, Step 5 Right
Correct / Predicted	12 / 12	12 / 12	12 / 12
Inconsistent	3 / 12	2 / 12	2 / 12

Tabel 5.
Hasil pengujian *standard FOV*

True Label	Predicted Label(s)
	Standard FOV
Step 1	Step 1
Step 2 Left	Step 2 Left
Step 2 Right	Step 2 Right
Step 3	Step 3, Step 1
Step 4 Left	Step 4 Left
Step 4 Right	Step 4 Right
Step 5 Left	Step 5 Left
Step 5 Right	Step 5 Right
Step 6 Left	Step 6 Left
Step 6 Right	Step 6 Right
Step 7 Left	Step 7 Left
Step 7 Right	Step 7 Right, Step 5 Right
Correct / Predicted	12 / 12
Inconsistent	2 / 12

B. Arsitektur CNN

Pada artikel, digunakan *EfficientNet-B0* dengan *weight NoisyStudent* sebagai base model. *EfficientNet* merupakan sebuah *pretrained model* yang diciptakan menggunakan algoritma *AutoML-MNAS*, sebuah algoritma yang ditujukan untuk mencari model *neural network* terbaik untuk menyelesaikan suatu masalah klasifikasi (dalam hal ini *ImageNet*). Ditemukanlah *EfficientNet-B0* yang pada dasarnya merupakan *MobileNetV2* dengan penambahan *Squeeze-and-Excite Block* didalamnya. *EfficientNet-B0* kemudian dikembangkan menjadi model yang lebih besar (varian B1 s.d B7) menggunakan *Compound Scaling*, sebuah metode yang melakukan scaling pada *Depth*, *Width* dan *Resolution* dari model sekaligus [9].

EfficientNet pada sistem ini diatur dengan konfigurasi *Trainable* pada *Layer* ke 21 dan seterusnya, kecuali pada lapisan *Batch Normalization*. Diaplikasikan pula *Image Augmentation* sebelum memasuki *EfficientNet*, di antaranya *Random Rotation (0.15)*, *Random Translation (0.1, 0.1)*, *Random Flip* dan *Random Contrast (0.1)*. Detail arsitektur dari model ini dapat dilihat pada Tabel 3.

C. Hasil Training dan Validation

Proses *training* dan *validation* dilakukan dengan jumlah *epoch* sebanyak 25 dan *batch size* sebesar 12. Dari hasil



Gambar 11. Sampel *wide FOV*.



Gambar 12. Sampel *shallow FOV*.



Gambar 13. Sampel video sudut 30°.

training, didapatkan tingkat akurasi training sebesar 99% dan tingkat akurasi *validation* sebesar 99%. Terlihat pula bahwa ada penurunan drastis pada rentang *epoch* 17-20 seperti yang ditunjukkan pada Gambar 8.

D. Hasil Testing

Menggunakan *Test Dataset*, Sistem di Evaluasi performanya. Pada tahap ini, didapatkan hasil akurasi sebesar 99%. Gambar 9 merupakan *Confusion Matrix* dari hasil *testing*.

E. Pengujian berdasarkan Window Size

Pengujian berdasarkan *Window Size* dilakukan untuk mengetahui *Windows Size* (dinyatakan dalam *second*) yang paling optimal dalam penggunaan sistem ini. Pada pengujian ini, di uji *Window Size* 1s, 2s dan 3s. Pada pengujian ini didapat *Window Size* 1s memberikan inkonsistensi paling tinggi sedangkan *Window Size* 3s tidak memberikan hasil yang lebih baik dibandingkan *Window Size* 2s. Hasil pengujian ini dapat dilihat pada Tabel 4.

F. Pengujian Berdasarkan FOV

Pengujian Berdasarkan *Field of View*, sebagai hasil dari jarak kamera terhadap objek, dilakukan untuk mengetahui jarak kamera yang tepat terhadap objek. Pada pengujian ini, sistem diuji menggunakan 3 FOV berbeda, yakni *Standard*

Tabel 6.
Hasil pengujian *wide FOV*

True Label	Predicted Label(s) <i>Wide FOV</i>
Step 1	Step 1
Step 2 Left	Step 2 Left
Step 2 Right	Step 2 Right
Step 3	Step 3
Step 4 Left	Step 4 Left, Step 5 Right
Step 4 Right	Step 4 Right
Step 5 Left	Step 5 Left
Step 5 Right	Step 5 Right
Step 6 Left	Step 6 Left
Step 6 Right	Step 6 Right, Step 4 Right
Step 7 Left	Step 7 Left
Step 7 Right	Step 7 Right, Step 5 Right
<i>Correct / Predicted</i>	<i>12 / 12</i>
<i>Inconsistent</i>	<i>3 / 12</i>

Tabel 7.
Hasil pengujian *shallow FOV*

True Label	Predicted Label(s) <i>Shallow FOV</i>
Step 1	Step 1
Step 2 Left	Step 2 Left
Step 2 Right	Step 2 Right
Step 3	Step 3
Step 4 Left	Step 4 Left
Step 4 Right	Step 4 Right
Step 5 Left	Step 5 Left, Step 7 Left
Step 5 Right	Step 5 Right
Step 6 Left	Step 6 Left
Step 6 Right	Step 6 Right, Step 4 Right
Step 7 Left	Step 7 Left
Step 7 Right	Step 7 Right
<i>Correct / Predicted</i>	<i>12 / 12</i>
<i>Inconsistent</i>	<i>2 / 12</i>



Gambar 14. Sampel pengujian latar *metal sink*.

(Gambar 10), *Wide* (Gambar 11) dan *Shallow FOV* (Gambar 12). Pada pengujian ini terlihat bahwa *Wide FOV* memberikan inkonsistensi tertinggi (3/12) dibandingkan *Standard* dan *Shallow FOV* (2/12). Hasil Pengujian ini dapat dilihat pada Tabel 5, Tabel 6 dan Tabel 7.

G. Pengujian Berdasarkan Sudut

Pengujian berdasarkan sudut dimaksudkan untuk mengetahui performa sistem pada kondisi pengambilan dengan sudut yang tidak seharusnya. Pada pengujian ini, didapat bahwa pengambilan video pada sudut 30° memberikan akurasi yang lebih rendah dari pada video cuci tangan dengan sudut normal (video *Standard FOV*). Hasil Pengujian ini dapat dilihat pada Tabel 8 dan Gambar 13.

H. Pengujian Berdasarkan Latar

Pada Pengujian ini, sistem akan diuji performanya terhadap video dengan latar berbeda dari dataset yang ada.

Tabel 8.
Hasil pengujian sudut

True Label	Predicted Label(s) 30° Angle
Step 1	Step 1, Step 7 Right, Step 5 Right, Step 2 Left, Step 2 Right, Step 3, Step 7 Left
Step 2 Left	Step 2 Left
Step 2 Right	Step 2 Right, Step 2 Left
Step 3	Step 3
Step 4 Left	Step 4 Left, Step 5 Right
Step 4 Right	Step 4 Right
Step 5 Left	- Not included in the video -
Step 5 Right	- Not included in the video -
Step 6 Left	Step 6 Left, Step 5 Left, Step 4 Left, Step 7 Left
Step 6 Right	Step 6 Right, Step 2 Left
Step 7 Left	Step 7 Left
Step 7 Right	Step 7 Right, Step 7 Left, Step 5 Right, Step 2 Left
<i>Correct / Predicted</i>	<i>10/10</i>
<i>Inconsistent</i>	<i>6 / 10</i>

Tabel 9.
Hasil pengujian latar

True Label	Predicted Label(s) <i>Metal Sink</i>
Step 1	Step 1, Step 7 Left
Step 2 Left	Step 2 Left
Step 2 Right	Step 2 Right
Step 3	Step 7 left
Step 4 Left	Step 6 Left
Step 4 Right	Step 4 Right
Step 5 Left	Step 5 Left
Step 5 Right	Step 5 Right
Step 6 Left	Step 6 Left
Step 6 Right	Step 6 Right, Step 6 Left
Step 7 Left	Step 7 Left
Step 7 Right	Step 7 Right, Step 5 Right
<i>Correct / Predicted</i>	<i>10 / 12</i>
<i>Inconsistent</i>	<i>3 / 10</i>

Terlihat bahwa klasifikasi pada video *Metal Sink*, yakni video dengan latar wastafel berbahan *Stainless Steel*, memberikan akurasi yang jauh lebih rendah, yakni 6/10 (60%), dari video dengan latar wastafel keramik (video *Standard FOV*), yakni 12/12 (100%). Hasil Pengujian ini dapat dilihat pada Tabel 9 dan Gambar 14.

IV. KESIMPULAN

Pada penelitian ini, Metode yang digunakan dapat menghasilkan tingkat akurasi klasifikasi yang cukup baik yakni 12/12 (100%) dengan inkonsistensi 2/12 (16,67%) pada setup dan latar yang sesuai. Penggunaan *EfficientNet-B0* dengan *Weight NoisyStudent* mampu menghasilkan model dengan akurasi di atas 90% pada fase evaluasi *training* dengan kurang dari 20 *epoch*. Window Size terbaik untuk digunakan pada sistem ini adalah 2s (*Second*), Penggunaan pada video dengan sudut 30° akan meningkatkan inkonsistensi hingga ke 6/10 (60%) dari hasil prediksi akibat tidak terlihatnya bagian tangan, oleh karena itu diperlukan penempatan kamera yang sesuai. Penggunaan sistem pada video dengan latar yang berbeda dari *dataset* menyebabkan penurunan akurasi ke 10/12 (83,33%), ini menandakan perlunya penambahan dataset yang lebih beragam. Penempatan kamera yang terlalu jauh juga akan meningkatkan inkonsistensi dari sistem ini, ini menandakan jarak kamera terhadap objek sangat berpengaruh terhadap performa sistem.

DAFTAR PUSTAKA

- [1] WHO, "Coronavirus Disease (Covid-19) Pandemic," *World Health Organization*, 2020. who.int/emergencies/diseases/novel-coronavirus-2019.
- [2] I. Parra, D. Fernandez, M. A. Sotelo, M. Marron, M. Gavilan, and G. Lacey, "Tracking using Particle and Kalman Filters in Hand Washing Quality Assessment System," in *2007 IEEE International Symposium on Intelligent Signal Processing*, 2007, pp. 1–6.
- [3] D. Fernandez Llorca, I. Parra, M. Angel Sotelo, and G. Lacey, "A vision-based system for automatic hand washing quality assessment," *Mach. Vis. Appl.*, vol. 22, no. 2, pp. 219–234, 2011.
- [4] S. Ameling *et al.*, "A vision-based system for handwashing quality assessment with realtime feedback," 2011.
- [5] D. N. Fernández and B. Kwolek, "Hand Posture Recognition using Convolutional Neural Network," in *Iberoamerican Congress on Pattern Recognition*, 2017, pp. 441–449.
- [6] H. F. Hanafi, M. H. Abd Wahab, K.-T. Wong, A. Z. Selamat, M. H. M. Adnan, and F. H. Naning, "Mobile augmented reality Hand wash (MARHw): Mobile application to guide community to ameliorate handwashing effectiveness to oppose COVID-19 disease," *Int. J. Integr. Eng.*, vol. 12, no. 5, pp. 217–223, 2020.
- [7] C. Wang, Z. Sarsenbayeva, X. Chen, T. Dingler, J. Goncalves, and V. Kostakos, "Accurate measurement of handwash quality using sensor armbands: Instrument validation study," *JMIR mHealth uHealth*, vol. 8, no. 3, p. e17001, 2020.
- [8] E. Prakasa and B. Sugiarto, "Video Analysis on Handwashing Movement for the Completeness Evaluation," in *2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, 2020, pp. 296–301.
- [9] M. Tan and Q. Le, "Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114.