

## PENGEMBANGAN SISTEM IDENTIFIKASI ALAT PELINDUNG DIRI (APD) MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)

Tevin Abdi Kurnia<sup>1</sup>, Joko Endrasmono<sup>2</sup>, Ryan Yudha Adhitya<sup>3</sup>

<sup>123</sup>Program Studi Teknik Otomasi, Jurusan Teknik Kelistrikan Kapal

Politeknik Perkapalan Negeri Surabaya

Jalan Teknik Kimia, Kampus ITS Sukolilo, Surabaya 60111

E-mail: [tevinpratama@student.ppns.ac.id](mailto:tevinpratama@student.ppns.ac.id)

### Abstrak

Dalam penelitian kali ini, akan dikembangkan Sistem Identifikasi Alat Pelindung Diri (APD). Pada penelitian sebelumnya APD yang dideteksi hanya APD area kepala berupa *Safety Helmet, Earmuff, Safety Glasses* dan *Safety Mask*. Kenyataannya Alat Pelindung Diri juga berupa *Safety Gloves* dan *Safety Shoes*. Maka dari itu pada penelitian kali ini sistem mampu menambahkan deteksi pada area lain berupa deteksi *Safety Gloves* dan *Safety Shoes*. Penelitian ini didasari oleh kurangnya kesadaran dan kedisiplinan pekerja mengenai penggunaan alat pelindung diri. Berdasarkan data Badan Penyelenggara Jaminan Sosial Kesehatan (BPJS), sepanjang tahun 2018 terjadi 157.313 kasus kecelakaan kerja. *Input* dari sistem ini berupa *capture* pekerja dan Sensor Proximity Induktif untuk mendeteksi seputar *safety*. Identifikasi sistem ini menggunakan pengolahan citra. Pengolahan citra akan dilakukan dengan metode *Convolutional Neural Network*. Penelitian ini dapat menggantikan peranan manusia sebagai pengawas kelengkapan penggunaan APD. *Output* CNN dibagi menjadi 2 kategori yaitu lengkap dan tidak lengkap. Akurasi dari CNN yang digunakan mencapai 93,75%. Untuk pengujian kondisi berbeda dengan dataset akurasi mencapai 87,5%

**Kata Kunci:** Convolutional Neural Network, Pengolahan Citra, Alat Pelindung Diri.

### 1. PENDAHULUAN

#### 1.1 Latar Belakang

Keselamatan pekerja merupakan hal yang sangat penting dalam dunia industri. Kesehatan dan keselamatan kerja atau yang bisa disebut K3 mutlak harus dipatuhi oleh semua industri. Salah satu hal yang wajib diperhatikan adalah alat pelindung diri. Alat pelindung diri sangat penting untuk diperhatikan karena alat pelindung diri berfungsi untuk mengurangi resiko terhadap *hazard* yang ada di industri. Tidak menggunakan Alat Pelindung Diri (APD) merupakan salah satu faktor terjadinya kecelakaan kerja. Faktor kurangnya kedisiplinan penggunaan Alat Pelindung Diri (APD) oleh para pekerja salah satu diakibatkan oleh kurangnya pengawasan kelengkapan alat pelindung diri oleh perusahaan terhadap karyawannya. Berdasarkan data BPJS Ketenagakerjaan (2018), terdapat 157.313 kasus kecelakaan kerja. Oleh karena itu, dibutuhkan suatu alat yang dapat mendeteksi dan memastikan kelengkapan alat pelindung diri yang digunakan oleh karyawan. Pada penelitian sebelumnya, target sasaran deteksi sistem hanya pada area kepala saja. Namun, menurut Kementerian Tenaga Kerja dan

Transmigrasi (2010) juga disebutkan pelindung tangan dan pelindung kaki. Maka dari itu penelitian kali ini bertujuan mengembangkan sistem dari penelitian sebelumnya sehingga pengenalan sistem kepada Alat Pelindung Diri (APD) dapat menyeluruh termasuk *Safety Gloves* dan *Safety Shoes*. Penelitian kali ini yaitu “Pengembangan Sistem Identifikasi Kelengkapan Alat Pelindung Diri (APD) Menggunakan Metode Convolutional Neural Network (CNN)”. Sistem ini akan menggunakan metode *Convolutional Neural Network* (CNN). CNN merupakan salah satu perhitungan *neural network* yang digunakan pada pengolahan citra. Pengolahan citra digunakan pada sistem ini sebagai proses identifikasi kelengkapan alat pelindung diri yang digunakan. Dari hasil proses 2 pengolahan citra akan didapatkan 2 kategori yaitu alat pelindung diri yang digunakan pekerja dalam kategori lengkap dan kategori tidak lengkap. Kategori kelengkapan tersebut akan mempengaruhi aktuator.

#### 1.2. Rumusan Masalah

Berikut adalah rumusan masalah :

1. Bagaimana pengaplikasian Sistem Identifikasi Kelengkapan Alat Pelindung Diri (APD) Menggunakan Metode *Convolutional Neural Network* (CNN)?
2. Bagaimana tingkat akurasi yang didapatkan dari hasil klasifikasi kelengkapan alat pelindung diri menggunakan CNN?
3. Bagaimana cara mendapatkan nilai akurasi pelatihan yang optimal?

### 1.3 Tujuan

Tujuan yang ingin dicapai, yaitu, mampu mengaplikasikan metode *Convolutional Neural Network* pada sistem identifikasi kelengkapan alat pelindung diri., mampu mengetahui tingkat akurasi yang didapatkan dari hasil klasifikasi kelengkapan alat pelindung diri menggunakan CNN, dan mampu mendapatkan nilai akurasi pelatihan yang optimal.

## 2. METODOLOGI

### 2.1 Alat Pelindung Diri (APD)

Menurut Kementerian Tenaga Kerja dan Transmigrasi (2010) Alat Pelindung Diri (APD) adalah suatu alat yang mempunyai kemampuan untuk melindungi seseorang yang fungsinya mengisolasi sebagian atau seluruh tubuh dari potensi bahaya di tempat kerja.

Menurut Kementerian Tenaga Kerja dan Transmigrasi (2010) Alat Pelindung Diri (APD) berupa :

- a. pelindung kepala.
- b. pelindung mata dan muka.
- c. pelindung telinga.
- d. pelindung pernapasan beserta perlengkapannya.
- e. pelindung tangan.
- f. pelindung kaki.

### 2.2 Model RGB

Model warna *RGB* merupakan model warna yang paling umum digunakan pada pengolahan citra. *RGB* adalah adalah suatu model warna yang terdiri dari 3 buah warna yaitu : *Red*, *Green*, dan *Blue* yang kemudian ditambahkan berbagai komposisi untuk menghasilkan warna baru.

### 2.3 Model Warna *Grayscale*

Citra *grayscale* adalah citra yang hanya memiliki warna tingkat keabuan. Citra *grayscale* digunakan untuk mempermudah pengambilan informasi ketimbang citra berwarna.

### 2.4 Neural Network

*Neural Network* dapat dikatakan sebagai replika dari sistem syaraf yang terdapat pada sistem otak manusia. *Neural network* merupakan bagian dari sistem kecerdasan buatan yang digunakan untuk memproses informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya (Norvig, 1995).

Umumnya, terdapat 3 lapisan pada *Neural Network*, lapisan tersebut yaitu :

#### 1. Input layer

*Input layer* berisi *neuron-neuron* yang masing-masing menyimpan sebuah nilai masukan.

#### 2. Hidden layer

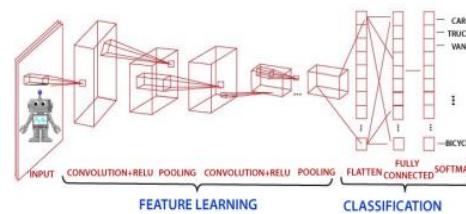
Semua proses pada fase pelatihan dan fase pengenalan dijalankan di lapisan ini. Lapisan ini tidak pernah muncul sehingga dinamakan *hidden layer*.

#### 3. Output layer

*Output layer* berfungsi untuk menampilkan hasil perhitungan sistem oleh fungsi aktivasi pada lapisan *hidden layer* berdasarkan *Input* yang diterima.

### 2.5 Convolutional Neural Network

*Convolutional Neural Network* (CNN) merupakan salah satu jenis *neural network* yang biasa digunakan pada *data image*. Secara garis besar 12 CNN tidak berbeda jauh dengan *neural network* biasa.. *Convolutional neural network* sendiri adalah perpaduan antara konvolusi citra untuk proses ekstraksi *feature*, dan *neural network* untuk klasifikasi mendekteksi dan mengenali objek pada sebuah *image*. Berikut adalah jaringan arsitektur *Convolutional Neural Network*:



Gambar 2.1 Arsitektur Convolutional Neural Network (javapoint.com, 2020)

#### 1. Convolutional Layer

*Convolutional Layer* melakukan operasi konvolusi pada output dari layer sebelumnya.

## 2. ReLu Layer

*ReLU* atau *rectified linear unit layer*, diibaratkan thresholding atau aktivasi jaringan syaraf tiruan. *ReLU Layer* bertujuan menjaga hasil citra proses konvolusi berada pada domain definit positif.

## 3. Subsampling Layer

Proses mereduksi ukuran sebuah data citra. Dalam pengolahan citra. Tujuan *subsampling* untuk meningkatkan invariansi posisi.

## 4. Fully Connected Layer

Layer yang biasanya digunakan dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data.

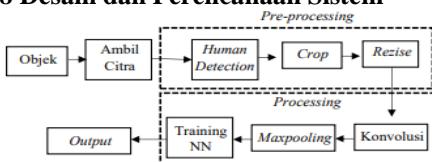
## 2.6 Webcam

*Webcam* pada dasarnya adalah sebuah kamera digital yang terhubung ke komputer, yang berfungsi untuk mengambil citra yang akan diolah oleh computer. Disini *webcam* digunakan untuk mendeteksi kelengkapan Alat Pelindung Diri yang dikenakan.

## 2.7 Solenoid Door Lock

*Solenoid Door Lock* merupakan solenoid otomatis yang difungsikan untuk pengunci pintu. Sistem kerja *Solenoid Door Lock* ini adalah *Normally Close* (NC). Disini *Solenoid Door Lock* berfungsi untuk mengunci pintu apabila pekerja terdeteksi tidak menggunakan Alat Pelindung Diri (APD) yang lengkap.

## 2.8 Desain dan Perencanaan Sistem



Gambar 2.2 Gambaran Umum Image Processing

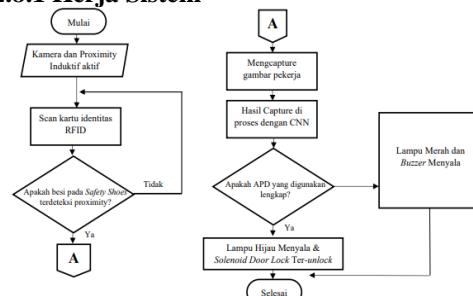
Sebelum tahap Proses CNN, citra akan dilakukan proses penyesuaian dengan dataset. Mula-mula dengan proses *Human Detection* yang berfungsi menentukan batas pixel terluar yang terdeteksi. Selanjutnya *crop*. *Input image* akan di-*crop* menyesuaikan batas yang ditentukan pada proses *human detection*. Teknik *cropping* bertujuan untuk memfokuskan objek yang diklasifikasi. (Putri, 2018).

Kemudian *rezise*, yang berfungsi merubah ukuran sebuah gambar. Dapat dirubah lebih

besar atau lebih kecil dari ukuran sebelumnya. (Hibatullah,2019).

Karena proses *cropping* merubah pixel *input image*, perlu penyesuaian pixel ulang agar ukuran pixel *input image* sesuai dengan dataset. Selanjutnya tahap *processing*. Dimana, *Input* citra akan masuk tahap konvolusi. Proses konvolusi membuat *array* baru atau *Input* baru. Setelah itu proses *maxpooling*, yang berfungsi untuk mempercepat proses komputasi pada program dan menghilangkan *noise* pada *Input* citra yang diambil. Setelah itu proses inti, yaitu proses *Training* dengan *Neural Network* (NN). Pada proses ini akan dihasilkan bobot yang digunakan untuk klasifikasi *Output*.

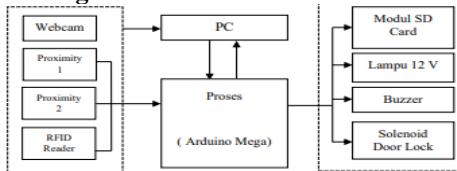
## 2.8.1 Kerja Sistem



Gambar 2.3 Flowchart alur kerja sistem

Pada Gambar 2.3, tahap awal pada sistem deteksi ini adalah mempersiapkan *Data Training* dan *library* untuk pemrograman. Kemudian sensor proximity induktif mendeteksi adanya logam pada *Safety Shoes*. Setelah itu pekerja absen *scan* kartu *RFID Reader* untuk proses absensi record. Setelah data yang dibaca *RFID Reader* cocok, maka kamera akan mengambil gambar pekerja. Setelah itu masuk tahap Algortima CNN memproses data sehingga dapat mengklasifikasikan kelengkapan alat pelindung diri seperti *Safety Helmet*, *Safety Glasses*, *Safety Mask*, *Safety Earmuff*, *Safety Gloves*, *Safety Shoes*. Ini adalah tahap inti sistem pendekripsi kelengkapan alat pelindung diri. Jika pekerja menggunakan APD lengkap, maka indikator lampu hijau 12V DC akan menyala dan *solenoid door lock* membuka, dan pekerja dipersilahkan masuk. Tetapi, jika APD yang digunakan pekerja masuk kategori tidak lengkap maka indikator lampu merah 12V DC dan alarm akan menyala sebagai tanda pekerja tidak boleh masuk.

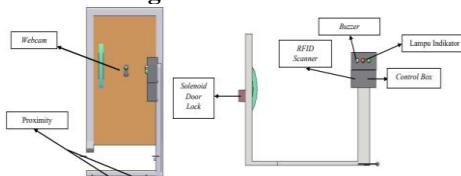
### 2.8.2 Diagram Blok Sistem



Gambar 2.4 Rancangan Desain Kerja Hardware

Pada Gambar 2.4, dijelaskan bahwa sistem ini mempunyai 4 input yaitu kamera webcam, RFID dan sensor *proximity* untuk pendekripsi metal yang terdapat pada *safety shoes*. Dan terdapat 4 buah aktuator yaitu Buzzer untuk alarm, lampu merah dan hijau 12V DC untuk indikator klasifikasi kelengkapan alat pelindung diri yang digunakan pekerja serta *Solenoid Door Lock* untuk mengunci pintu apabila APD pekerja tidak lengkap.

### 2.8.3 Perancangan Hardware



Gambar 2.5 Rancangan Hardware

Gambar 2.5 merupakan rancangan *Hardware*, dimana pekerja akan melakukan scan kartu RFID, *Scanner RFID* berada di balik *control box*, lalu sensor *proximity* induktif akan mendekripsi besi yang terdapat pada *safety shoes*, *webcam* akan men-capture pekerja untuk mendekripsi kelengkapan APD. Apabila kelengkapan alat pelindung diri sudah sesuai standar maka lampu hijau 12V DC menyala dan *solenoid door lock* akan membuka menandakan pekerja diperbolehkan masuk. Namun jika tidak sesuai standar, alarm dan lampu merah 12V DC akan aktif sebagai tanda peringatan.

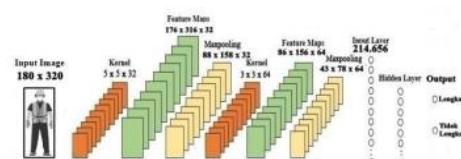
### 2.9 Arsitektur CNN

Setelah penyesuaian dataset, *input image* akan diproses dengan menggunakan Metode CNN. Arsitektur metode *Convolutional Neural Network* terdapat pada Gambar 2.6 dengan proses *pre-processing* terdiri dari 2 proses, pembentukan dataset dan konversi dataset menjadi 1 layer. Kemudian proses ke 2 adalah *processing* yang terdiri dari konvolusi citra, *maxpooling*, *training NN*, dan *softmax*. Untuk proses yang terakhir adalah *classifying* yang mempunyai 1 proses yaitu penentuan *output*.

Arsitektur digunakan untuk dasar pembuatan program CNN. Pada arsitektur ini, terdapat *input image* yang memiliki resolusi sebesar 180 x 320. *Input image* ini akan masuk ke proses konvolusi. Proses ini melibatkan kernel atau filter yang berukuran 5 x 5 x 32 yang artinya filter ini mempunyai ukuran pixel 5 x 5 sebanyak 32. Setelah diproses melalui program Python, proses ini menghasilkan *input image* baru yang dinamakan *feature maps*. *Feature maps* mempunyai resolusi lebih kecil dibandingkan *image* sebelumnya tetapi mempunyai *depth* yang bertambah. Resolusi yang dihasilkan sebesar 176 x 316 x 32.

Hasil proses akan masuk ke tahap *maxpooling*. Yaitu suatu proses dimana citra akan diambil nilai terbesarnya dengan ketentuan pixel tertentu. Pada tahap ini pixel yang digunakan sebesar 2 x 2. Jadi, setiap 2 x 2 pixel pada *feature maps* akan diambil nilai terbesarnya. Hasilnya, resolusi pada citra sebelumnya akan berubah menjadi 88 x 158 x 32.

Citra yang dihasilkan setelah melewati proses konvolusi sebanyak 2 kali mempunyai resolusi sebesar 43 x 78 x 64. Setelah itu, masuk tahap *flatten*. Yang merupakan proses dimana *input image* yang didapatkan akan diubah menjadi 1 dimensi untuk di *training* dengan *neural network* (NN). Hasil dari *flatten* sebesar 214.656 x 1. Nilai ini yang akan menjadi vektor input pada *neural network*. Sebelum *training* perlu adanya proses analisis *Optimizer*. Analisis ini berfungsi untuk menentukan *Optimizer* yang akan digunakan. Dan yang di tes adalah *Adamax Optimizer*.



Gambar 2.6 Arsitektur CNN

*Output* dari CNN berupa 2 kategori klasifikasi yaitu, lengkap, dan tidak lengkap.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Hasil Hardware

Bahan yang digunakan untuk rangka utama adalah *hollow aluminium* ukuran 75 mm x 25 mm x 2 mm. Pintu berjenis PVC berdimensi 70 cm x 190 cm sebagai akses masuk. Dan juga digunakan untuk meletakkan *webcam* dan *solenoid door lock*. *RFID reader* dan komponen lain diletakkan di dalam *control box* yang berjarak 200 cm dari kamera dan tinggi dari lantai 100 cm. Sedangkan jarak antara kamera dan sensor *proximity* induktif sekitar

220 cm. Gambar hasil alatnya dan peletakan kamera serta solenoid door lock dapat dilihat di Gambar 3.1



Gambar 3.1 Hasil Alat yang digunakan

### 3.2 Pengujian Webcam

Kamera yang digunakan logitech C525. PC akan membaca dan menampilkan video secara *real-time*. Pengecekan kamera dapat dilihat dari indikator kamera. Terdapat indikator LED kuning. Indikator akan menyala jika program meminta kamera untuk hidup. Gambar 3.2 berikut ini merupakan gambar indikator LED ketika kamera mati dan hidup.



Gambar 3.2 Indikator LED Ketika Mati dan Hidup

### 3.3 Pengujian RFID MFRC522

Pengujian RFID MFRC522 dilakukan dengan menghubungkan Arduino dan diprogram ketika kartu RFID didekatkan ke arah *receiver* RFID. Maka *Serial* akan menampilkan code dari *tag* kartu yang terdeteksi di Arduino IDE.

Pengujian dilakukan 5 kali dengan jarak yang berbeda - beda. Pada Jarak 0 – 3 cm menunjukkan bahwa RFID dapat mendeteksi objek. Tapi pada jarak lebih dari 3 cm, RFID tidak mampu mendeteksi kartu RFID.

Tabel 3.1 Pengujian RFID MFRC522

No.	Pengujian	Jarak Pengujian	Kondisi
1.	1	0	Terdeteksi
2.	2	1	Terdeteksi
3.	3	2	Terdeteksi
4.	4	3	Terdeteksi
5.	5	4	Tidak Terdeteksi

### 3.4 Pengujian SD Card

Modul *SD Card* digunakan untuk menghubungkan sistem dengan *SD Card*. *SD Card* berfungsi sebagai penyimpanan data absensi. Data tersebut disimpan dalam bentuk .txt. data yang disimpan pada *SD card* berupa *Tag UID*, Kelengkapan dan waktu *tap* RFID.

### 3.5 Pengujian RTC DS1307

Modul RTC DS1307 digunakan sebagai pewaktu sistem. Pengujian dilakukan untuk memastikan keakuratan RTC DS1307 dibanding waktu sebenarnya. Dari pengujian didapati bahwa waktu yang ditunjukkan RTC DS1307 lebih cepat 1 detik dari waktu sebenarnya. Tabel 3.2 berikut merupakan hasil uji dari RTC DS1307.

Tabel 3.2. Pengetesan RTC DS1307

No.	RTCDS1307	Jam Sebenarnya	Selisih / error
1.	23 : 36 : 44	23 : 36 : 43	1 detik
2.	23 : 36 : 45	23 : 36 : 44	1 detik
3.	23 : 36 : 46	23 : 36 : 45	1 detik
4.	23 : 36 : 47	23 : 36 : 46	1 detik
5.	23 : 36 : 48	23 : 36 : 47	1 detik

### 3.6 Pengujian Proximity Induktif

Proximity induktif digunakan untuk mendeteksi besi pada sepatu *safety*. Terdapat indikator apabila terdeteksi besi / metal. Jika sensor tidak mendeteksi besi maka indikator akan mati. Sensor hanya mendeteksi besi, jika benda non-metal didekatkan sensor tidak akan terjadi apa-apa. Gambar 3.3 adalah hasil pengujian.



Gambar 3.3. Pengujian logam, dan non-logam.

### 3.7 Pembuatan Dataset

Sebelum dilakukan proses *training*, yang perlu disiapkan sebelumnya adalah dataset. Terdapat 2 *class* dalam dataset tersebut. Dataset berjumlah 930 setiap kategorinya. Total dataset berjumlah 1860 gambar. Beberapa sampel dataset untuk kategori lengkap dan tidak lengkap pada Tabel 3.3

Tabel 3.3 Dataset

Data Training	Kategori
	Lengkap
	Tidak Lengkap

### 3.8 Penyesuaian Konversi Citra

Dataset harus melalui proses *human detection* terlebih dahulu untuk menormalisasi perbedaan tinggi badan dari manusia. Proses deteksi manusia dibantu oleh *library Histogram of Oriented Gradient* yang kemudian akan dihasilkan nilai x, y, w dan h untuk objek manusia. Dikarenakan metode *Convolutional Neural Network* hanya dapat menggunakan gambar dengan 1 layer. Maka perlu dilakukan konversi gambar RGB menjadi citra 1 layer.



Gambar 3.4 Proses Human Detection dan Konversi Citra

### 3.9 Konvolusi dan Maxpooling

Arsitektur *image* akan diaplikasikan dengan karnel 5x5 dan menghasilkan *feature maps* 1. Tahap selanjutnya dilakukan proses *maxpooling* 2x2 dengan pergeseran 2x2. Lalu konvolusi kedua menggunakan karnel 3x3. Dan selanjutnya *maxpooling* dengan ukuran yang sama dengan *maxpooling* sebelumnya. Sample dataset yang digunakan untuk tiap kategori ada pada tabel 3.4 dan 3.5

Tabel 3.4. Dataset kategori lengkap

No.	Gambar Asli (180 x 320)	Feature Maps 1 (176 x 316)	Maxpooling 1 (88 x 158)	Feature Maps 2 (86 x 156)	Maxpooling 2 (43 x 78)
1					

Tabel 3.5 Sample Dataset kategori tidak lengkap

No.	Gambar Asli (180 x 320)	Feature Maps 1 (176 x 316)	Maxpooling 1 (88 x 158)	Feature Maps 2 (86 x 156)	Maxpooling 2 (43 x 78)
1					

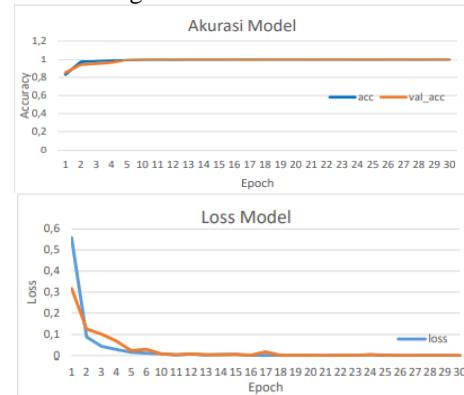
### 3.10 Training

Proses training dilakukan ketika proses konvolusi selesai. Menggunakan aplikasi spyder dengan *library* keras.io. Training dilakukan dengan jumlah *hidden layer* pada *hidden neuron* 256 67 dan fungsi aktivasi *ReLU*. Dilakukan untuk mencari *optimizer* yang cocok.

#### 3.10.1 Training Adamax Optimizer

Jumlah epoch = 30

Durasi learning = 189 menit



Gambar 3.5 Grafik Akurasi dan Loss Adamax

Didapatkan akurasi sebesar 1.000. Akurasi meningkat dan pada epoch ke 11 mulai mendekati 100% dan pada epoch ke 16 akurasi sudah 1.000.

Loss dari 0,55 menurun. Pada epoch 10 mulai mendekati 0.

Tabel 3.6. Hasil Training Adamax

	Hasil
Acc. Train	1.000
Loss Train	3.1290e-04
Val. Acc	1.000
Val. Loss	1.6871e-04

### 3.11 Pengujian Realtime

Pengujian *Realtime* untuk mengetahui akurasi real model hasil training sebelumnya. Bobot yang didapat dari hasil training diaplikasikan pada plan. Berikut beberapa sampel hasil uji *realtime* dengan kategori lengkap dan tidak lengkap pada Tabel 3.7. dari 96 pengujian yang dilakukan.

Tabel 3.7 Pengujian Realtime

	lengkap	['lengkap']	Benar
	tidak lengkap	['tidaklengkap']	Benar

Dari hasil pengujian, sistem mendeteksi 90 dari 96 kali uji *realtime*. Pengujian dilakukan 3 kali setiap probabilitas penggunaan APD dari 2 kategori kelengkapan. Akurasi dari hasil uji *realtime* mencapai 93,75% dengan total 6 kali kesalahan. Kesalahan deteksi diakibatkan sistem kesulitan untuk membedakan 1 item yang tidak dikenakan terutama untuk item yang

kecil seperti kacamata dan *earmuff*. Tabel 3.8. Menunjukkan hasil pengetesan *realtime*.

Tabel 3.8 Akurasi Realtime

No.	Kategori	Uji	Benar	Salah
1.	Lengkap	3	3	0
2.	Tidak Lengkap	93	87	6
	Total	96	90	6

### 3.12 Pengujian Beda Kondisi

Pengujian sistem menggunakan 3 APD dengan warna berbeda yaitu *safety helmet*, *safety gloves*, *safety earmuff*, *background* yang berbeda warna, lokasi pengujian berbeda dengan pengambilan dataset. Pada dataset, *background* menggunakan warna hijau, *safety helmet* putih, *earmuff* kuning dan *safety gloves* merah. Sedangkan uji kali ini, *background* yang digunakan berwarna kuning, *safety helmet* merah, *safety earmuff* merah, serta *safety gloves* biru. Beberapa sampel uji kondisi yang benar dan salah deteksi dapat dilihat pada Tabel 3.9

Tabel 3.9 Uji Beda Kondisi

Gambar Capture	Target	Deteksi	Keterangan
	Lengkap	['lengkap']	Benar
	Tidak Lengkap	['tidaklengkap']	Benar

Dari hasil pengujian, sistem memprediksi 35 benar dari 40 pengujian dengan akurasi 87,5%. Terdapat beberapa kesalahan deteksi ketika diuji dengan kondisi yang berbeda dari dataset namun kesalahan deteksi tersebut tidak terlalu signifikan. Hal tersebut menandakan bahwa metode *Convolutional Neural Network* mampu memprediksi cukup baik walaupun kondisi uji berbeda dengan kondisi saat pengambilan dataset.

## 4. KESIMPULAN

Dari hasil pengujian yang telah dilakukan. Dapat disimpulkan bahwa:

- Penelitian ini menggunakan metode *Convolution Neural Network* dengan 2 kategori output lengkap dan tidak lengkap. Dataset yang digunakan berjumlah 930 gambar tiap kategori sehingga Total 1860 gambar. Filter 5x5 yang berjumlah 32 digunakan

pada konvolusi pertama. Lalu proses *maxpooling* 2x2 dan berlanjut ke filter 3x3 digunakan pada konvolusi kedua dengan jumlah 64. Setelah itu kembali di *maxpooling* 2x2 dan dilakukan proses *flatten* menghasilkan vektor input 214.656. Proses training dengan *Neural Network* dengan 1 *hidden layer* dan jumlah *hidden neuron* sebanyak 256.

- Akurasi dari CNN yang digunakan sebesar 93,75%. Error deteksi diakibatkan sistem sulit mendeteksi kategori tidak lengkap yang kurang 1 item terutama kacamata *safety*, *earmuff* dan *gloves*. Untuk pengujian berbeda kondisi dengan dataset akurasi sistem mencapai 87,5%.
- Penelitian ini menggunakan *adamax optimizer*. Dengan menggunakan *adamax* didapatkan akurasi sebesar 1, akurasi validasi 1, *Loss* 3.1290e-04 dan *Val. Loss* 1.6871e-04

## 5. DAFTAR PUSTAKA

- Hibatullah, A., Maliki, I. and Bandung, J. D. (2019) ‘Penerapan Metode Convolutional Neural Network Pada Pengenalan Pola Citra Sandi Rumput’. Kemennakertrans (2010) ‘Peraturan Menteri Tenaga Kerja dan Transmigrasi Republik Indonesia’, Peraturan Menteri, pp. 1–69
- Putri, R. K. S. (2018) ‘IMPLEMENTASI DEEP LEARNING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI GAMBAR’, Society, pp. 14–18.
- Nazar, A., 2018. Prototype Sistem Sorting Packaging Rokok Dengan Metode Convolution Neural Network (CNN). Skripsi, pp. 9-13
- Pradana, R. D.W., 2019. Rancang Bangun Sistem Identifikasi Kelengkapan Alat Pelindung Diri (APD) Menggunakan Metode Convolutional Neural Network (CNN).
- Norvig, S. J. R. a. P., 1995. Artificial Intelligence A Modern Approach. 3rd ed. New Jersey: Prentice Hall.
- I Wayan Suartika E. P, A. Y. W. d. R. S., 2016. Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101. JURNAL TEKNIK ITS, 5(1), pp. 66-69.
- Stanford, 2016. An Introduction to Convolutional Neural Network. [Online]

Available at: <http://scarlet.stanford.edu/>

[Accessed 22 01 2020].

J. T. Springenberg, A. D. T. B. a. M. R., 2015.

Striving For Simplicity: The All  
Convolutional Net. ICLR 2015.

Triasanti, D., 2001. Konsep Dasar Python.

Sinau Arduino, 2018. Mengenal Arduino

Software (IDE). [Online] Available at:  
<https://www.sinauarduino.com/>  
[Accessed 22 01 2020].

Therzian Richard Perkasa, H. W. P. S., 2014.

Rancang Bangun Pendekripsi Gerak  
Menggunakan Metode Image  
Subtraction Pada Single Board  
Computer (SBC).