

HYPERPARAMETER TUNING EPOCH DALAM MENINGKATKAN AKURASI DATA LATIH DAN DATA VALIDASI PADA CITRA PENGENDARA

Indra Griha Tofik Isa^{1*} dan Beni Junedi²

¹ Jurusan Manajemen Informatika, Politeknik Negeri Sriwijaya
Jl. Srijaya Negara, Bukit Lama, Palembang 30139.

² Jurusan Pendidikan Matematika, Fakultas FKIP, Universitas Bina Bangsa
JL Raya Serang - Jakarta, Cipocok Jaya, Serang 42124.

*Email: indra_isa_mi@polsri.ac.id

Abstrak

Epoch merupakan salah satu parameter penting dalam pengolahan citra untuk melatih data latih, data validasi ataupun data uji yang menghasilkan nilai akurasi tertentu. Di dalam penelitian ini dilakukan hyperparameter tuning terhadap nilai epoch untuk melihat optimasi nilai akurasi yang dihasilkan. Adapun objek yang dicermati adalah kondisi pengendara kendaraan roda empat yang diklasifikasikan pada posisi (1) Kondisi Ideal berkendara, dan (2) Kondisi tidak ideal berkendara. Data gambar yang diamati sejumlah Sehingga rumusan masalah dari penelitian ini adalah bagaimana mengukur nilai Epoch dalam menghasilkan akurasi prediksi pengendara kendaraan roda empat. Objek yang diamati adalah 5416 citra yang bersumber dari dataset citra amazonaws.com yang sudah tersedia secara publik. Tujuan dari penelitian ini adalah mengukur nilai akurasi terbaik dengan melakukan hyperparameter tuning pada Epoch, sehingga menghasilkan nilai akurasi tertinggi. Tahapan penelitian dilakukan dengan (1) Integrasi Dataset; (2) Image Augmentation; (3) Preprocessing Data; (4) Pemodelan CNN; (5) Hyperparameter Tuning; (6) Penentuan Nilai Akurasi. Dari hasil hyperparameter tuning Epoch, nilai 15 memiliki hasil akurasi yang tertinggi yakni 92,83% untuk akurasi data latih dan 86,04% untuk akurasi data validasi.

Kata kunci: *Hyperparameter Tuning, Epoch, Image Processing, CNN*

1. PENDAHULUAN

Teknologi informasi telah memberikan dampak pada beragam dan banyaknya data yang dihasilkan, terlebih pada era revolusi industri 4.0 dimana data merupakan sebuah aset yang memberikan nilai tambah bagi pengguna (Astini, 2019). Revolusi industri 4.0 salah satunya ditandai dengan adanya teknologi yang memanfaatkan data-data tersebut menjadi pengetahuan baru, melalui kecerdasan buatan, teknologi *internet of things*, teknologi big data dan teknologi lainnya yang memanfaatkan data-data tersebut. Khususnya dalam kecerdasan buatan, dimana kecerdasan manusia dimodelkan atau disimulasikan pada perangkat dan mesin (Santoso, 2021). Diantara aspek kecerdasan buatan salah satunya adalah *Computer Vision* yang berarti kemampuan sebuah perangkat, dalam hal ini adalah komputer, untuk melihat serta mengartikan data visual menjadi sebuah parameter untuk diolah menjadi output tertentu. Tentunya di dalam *Computer Vision* diperlukan serangkaian teknik dan pengolahan citra visual tersebut, yang disebut dengan *image processing* atau pengolahan citra.

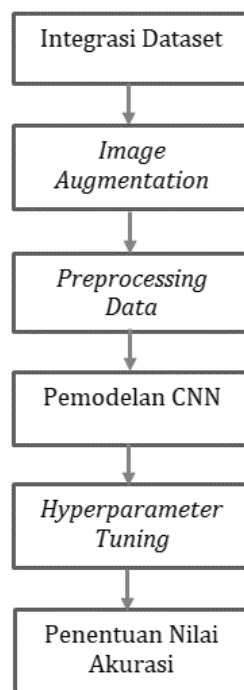
Teknologi pengolahan citra saat ini sudah berkembang dan banyak diimplementasikan di dalam perangkat teknologi, diantaranya adalah pengenalan citra sidik jari, dimana dilakukan implementasi *Convolutional Neural Network* untuk proses klasifikasi citra sidik jari yang menghasilkan output akurasi data pelatihan hingga mencapai 100% (Nurfita and Ariyanto, 2018). Teknologi citra klasifikasi kucing jenis Genus *Panthera* dengan pendekatan *computer vision* melalui bahasa pemrograman python dengan arsitektur CNN 1 *input layer*, 5 *convolution layer*, dan 2 *fully connected layer*, dataset training sebanyak 3840 citra yang menghasilkan akurasi sebesar 92,1% (Anwar and Riminarsih, 2019). Sedangkan implementasi pengolahan citra lainnya digunakan untuk klasifikasi kanker kulit melalui arsitektur VGG-16, yang menghasilkan akurasi sebesar 99,70% (S. R. Reynaldi, J. Apri and S. A. Wahyu, 2022) *Convolutional Neural Network* atau CNN merupakan salah satu algoritma yang digunakan dalam pengolahan citra, dimana CNN sangat mirip dengan jaringan saraf normal yang dapat divisualisasikan sebagai kumpulan neuron yang disusun sebagai grafik asiklik (Bhairnallykar *et al.*, 2020). Perbedaan

utama dari jaringan saraf adalah bahwa neuron lapisan tersembunyi hanya terhubung ke subset neuron di lapisan sebelumnya (Saputra and Ibadillah, 2019). Karena konektivitas yang jarang ini, ia mampu mempelajari fitur secara implisit. Secara umum CNN terdiri dari empat komponen yaitu: (1) Convolution layer (2) ReLU layer (Activation Function) (3) Pooling layer (4) Fully connected layer. CNN merupakan model yang mendapat perhatian karena kemampuan klasifikasinya berdasarkan informasi kontekstual

Pada penelitian yang dilakukan mengimplementasikan algoritma CNN untuk mengukur tingkat akurasi dari prediksi yang dihasilkan. Dalam mengoptimalkan algoritma CNN perlu dilakukan *hyperparameter tuning* untuk beberapa aspek, diantaranya adalah penyesuaian nilai Epoch untuk mengukur akurasi terbaik dari model yang diamati. Adapun objek yang dicermati adalah kondisi pengemudi kendaraan roda empat yang diklasifikasikan pada posisi (1) Kondisi Ideal berkendara, dan (2) Kondisi tidak ideal berkendara. Data gambar yang diamati sejumlah Sehingga rumusan masalah dari penelitian ini adalah bagaimana mengukur nilai Epoch dalam menghasilkan akurasi prediksi pengemudi kendaraan roda empat. Adapun untuk objek yang diamati adalah 5416 citra yang bersumber dari dataset citra amazonaws.com yang sudah tersedia secara publik. Tujuan dari penelitian ini adalah mengukur nilai akurasi terbaik dengan melakukan *hyperparameter tuning* pada Epoch, sehingga menghasilkan nilai akurasi tertinggi. Adapun nilai Epoch yang diukur adalah 3, 5, 10, 12 dan 15.

2. METODOLOGI

Penelitian dilakukan dengan objek gambar pengemudi kendaraan roda empat sebanyak 5416 citra. Perangkat lunak yang digunakan secara *online* melalui Google Colab dengan bahasa pemrograman python. Adapun metodologi penelitian ini dapat dilihat pada gambar 1 berikut :



Gambar 1. Metodologi Penelitian

Penelitian yang dilakukan dimulai dengan tahapan integrasi dataset melalui pengunduhan dataset citra posisi pengemudi kendaraan roda empat yang diambil dari dataset publik dari amazonaws.com. Dataset yang sudah diunduh, selanjutnya diklasifikasikan menjadi data latih dan data uji. Adapun jumlah data latih sebanyak 4015 yang terdiri dari data latih berlabel 0 yang masuk kelompok kondisi pengemudi

tidak ideal dan data latih berlabel 1 yang masuk kelompok kondisi pengendara ideal. Sedangkan jumlah data uji sebanyak 1401.

Tahapan berikutnya adalah *image augmented* dengan memperhatikan kondisi gambar secara keseluruhan, dimana melihat kondisi keseragaman zoom gambar (*rescale*), kondisi kemiringan (*rotation_range*), kondisi kemencengan (*shear_range*). *Preprocessing Data* dilakukan untuk menentukan berapa jumlah batch gambar, target ukuran gambar yang diuji, serta membagi data uji kedalam data uji dan data validasi.

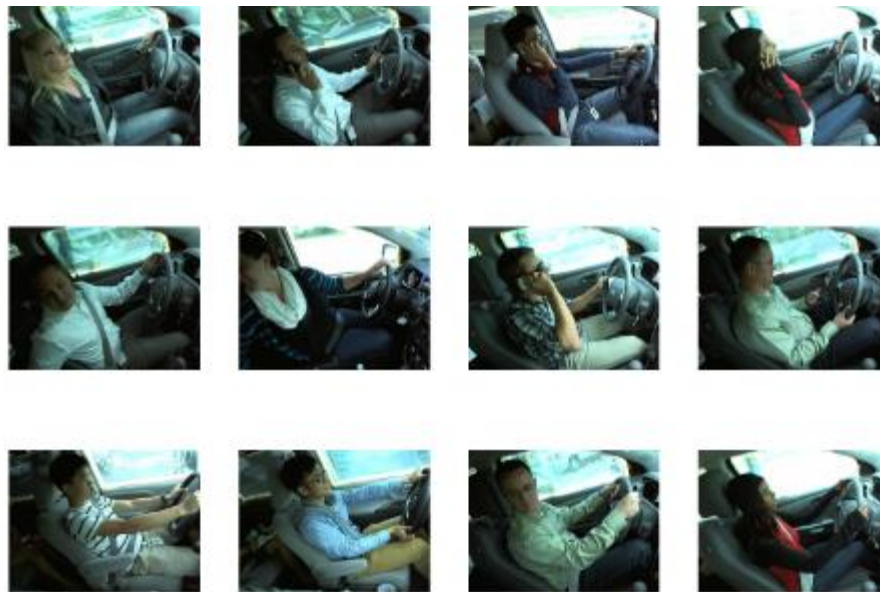
Pemodelan CNN dengan mengimplementasikan *convolution*, *relu*, dan *pooling*. *Convolution* dilakukan untuk menyederhanakan gambar melalui perhitungan matriks sehingga memudahkan sistem untuk membaca dataset dari citra diamati. Kemudian dilakukan metode *relu* yang digunakan untuk memberikan nilai 0 bagi angka-angka yang menghasilkan nilai negatif pada tahapan *convolution* (Maulana and Rochmawati, 2020). Sedangkan *pooling* dilakukan untuk menentukan nilai tertinggi dari berdasarkan hasil perhitungan matriks (Rasyid, Tahir and Syafaruddin, 2019)

Tahapan berikutnya adalah melakukan *hyperparameter tuning* pada epoch. Epoch dalam *machine learning* berarti satu lintasan lengkap dari kumpulan data pelatihan melalui algoritma. Nilai epoch ini merupakan *hyperparameter* penting untuk algoritma. Ini menentukan jumlah epoch atau lintasan lengkap dari seluruh dataset pelatihan yang melewati proses pelatihan atau pembelajaran algoritma (Setiawan, Candra and Suharyanto, 2019)

3. HASIL DAN PEMBAHASAN

3.1. Integrasi Dataset

Sebagaimana yang telah diuraikan dalam bab metodologi bahwa data yang diambil merupakan citra pengendara kendaraan roda empat dengan membagi kedalam data latih dan data uji. Tangkapan layar representasi data yang diamati menggunakan library `matplotlib.pyplot` dan `matplotlib.image`, divisualisasikan pada gambar 2 berikut :



Gambar 2. Tangkapan Layar Citra yang Diamati

Dari citra gambar di atas dapat dilihat dengan beragam posisi, diantaranya adalah posisi pengendara ideal (kedua tangan memegang kendali dan tatapan fokus ke depan) dan posisi pengendara tidak ideal yakni posisi pengendara dengan tangan memegang telepon selular, posisi pengendara dengan tangan

kanan mengambil benda ke belakang, posisi pengendara sedang membaca pesan pada telepon selular, posisi pengendara sedang melihat ke samping.

3.2. Image Augmentation

Image Augmentation dilakukan untuk meningkatkan akurasi dari pemodelan yang akan dibangun, agar model dapat memberikan toleransi untuk beberapa kondisi citra yang bersifat *outlier* namun masih dalam konteks domain citra yang diamati. Sebelum dilakukan pengaturan *image augmentation*, terlebih dahulu mengamati kondisi data citra pengendara kendaraan motor dengan melihat (1) Ukuran objek pengendara (kondisi zoom atau tidak); (2) Bagaimana kemiringan dari objek, apakah garis horizontal yang dihasilkan rata atau tidak; (3) kemencengan dari citra pengendara kendaraan. Sehingga terdapat 4 parameter yang diatur pada *Image Augmentation* yang menggunakan library *ImageDataGenerator* dari *tensorflow*, seperti yang ditunjukkan pada tabel 1 berikut :

Tabel 1. Parameter *Image Augmentation*

No	Parameter	Nilai
1	Rescale	1/255
2	Rotation Range	20
3	Zoom Range	0.2
4	Shear Range	0.2

Berdasarkan Tabel 1 di atas dapat dilihat bahwa nilai *rotation range* adalah 20, yang mengindikasikan bahwa model akan memberikan toleransi pada citra yang diuji atau dilatih jika ada kemiringan sebesar 20° ke kanan ataupun ke kiri. Nilai *zoom range* 0,2 yang artinya memberikan toleransi untuk citra dengan kondisi objek zoom in hingga 20%. Sedangkan *Shear Range* memiliki nilai 0,2 yang artinya memberikan toleransi untuk citra dengan kemencengan hingga 20%.

3.3. Preprocessing Data dan Modeling

1. Dalam tahapan *preprocessing data* dilakukan dengan membagi data latih ke dalam 2 bagian, yakni data latih dan data validasi dimana masing-masing terdiri dari 2 class, yakni class dengan label bernilai 0 dan label bernilai 1. Persentasi untuk pembagian data latih adalah 75% yakni sebanyak 3012 data, sedangkan persentasi untuk pembagian data validasi adalah 25% yakni sebanyak 1003 data.
2. *Modelling* dilakukan dengan membangun pemodelan CNN melalui *Convolution*, Relu dan *Pooling*. Proses *Convolution* dilakukan sebanyak 4 kali dengan *Convolution 1* senilai 64, *Convolution 2* senilai 32, *convolution 3* senilai 16 dan *convolution 4* senilai 8. Dari masing-masing *convolution* tersebut dilakukan proses relu yang bertujuan untuk memberikan nilai 0 bagi proses *convolution* yang menghasilkan nilai negatif. Serta *pooling* untuk memilih nilai tertinggi dari hasil *convolution* dengan menggunakan pola matriks (2 x 2). Gambar 3 berikut menunjukkan ringkasan dari model yang diimplementasikan:

3.4. Hyperparameter Tuning

Dalam tahapan ini dilakukan parameter *tuning* yang bertujuan untuk mengukur sejauh mana tingkat akurasi dari model yang diimplementasikan. Selain itu juga merupakan bagian dari melatih data latih dan data validasi sehingga mencapai titik akurasi yang optimal. Terdapat parameter yang bernama **Epoch** dimana selanjutnya parameter tersebut diatur untuk melatih berapa kali data latih dan data validasi di *training* untuk menghasilkan pemodelan dengan akurasi yang optimum. Di dalam penelitian ini, nilai epoch yang diberikan adalah 3, 5, 10, 12 dan 15. Adapun hasil dari *hyperparameter tuning* terhadap nilai epoch terdapat pada Tabel 2 :

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 127, 127, 64)	0
conv2d_1 (Conv2D)	(None, 125, 125, 32)	18464
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 32)	0
conv2d_2 (Conv2D)	(None, 60, 60, 16)	4624
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 16)	0
conv2d_3 (Conv2D)	(None, 28, 28, 8)	1168
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 8)	0
dropout (Dropout)	(None, 14, 14, 8)	0
flatten (Flatten)	(None, 1568)	0
dense (Dense)	(None, 32)	50288
dense_1 (Dense)	(None, 16)	528
dense_2 (Dense)	(None, 1)	17

Total params: 76,793
 Trainable params: 76,793
 Non-trainable params: 0

Gambar 3. Model Summary Pemodelan CNN yang Diimplementasikan

Tabel 2. Parameter Image Augmentation

No	Nilai Epoch	Hasil Akurasi
1	Nilai 3	0.2065
2	Nilai 5	0.4102
3	Nilai 10	0.7844
4	Nilai 12	0.9005
5	Nilai 15	0.9283

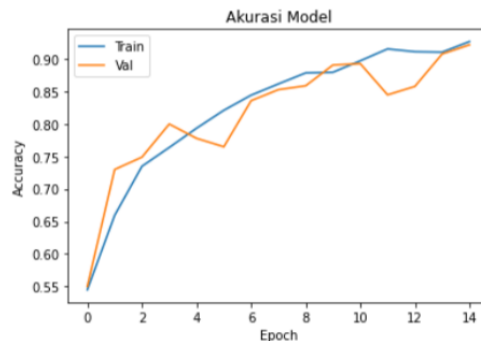
Pada Tabel 2 nampak terlihat bahwa dengan *hyperparameter tuning* untuk epoch dengan nilai 3 menghasilkan nilai akurasi 0,2065 yang mengindikasikan bahwa tingkat akurasi sebesar 20.65% dengan proses *training* bagi data latih dan data validasi dilakukan sebanyak 3 kali. Seiring dengan perubahan dan penambahan nilai epoch, berbanding lurus dengan nilai akurasi akhir yang dihasilkan. Dimana untuk nilai epoch 15 menghasilkan nilai akurasi akhir sebesar 0,9283 atau 92,83% dan nilai akurasi akhir data validasi sebesar 0,8604 atau 86,04%. Secara detail proses *training* melalui nilai epoch sebesar 15 dapat dilihat pada Gambar 4:

```

Epoch 1/15
47/47 [=====] - 82s 15/step - loss: 0.6909 - accuracy: 0.5332 - val_loss: 0.6805 - val_accuracy: 0.5025
Epoch 2/15
47/47 [=====] - 66s 15/step - loss: 0.6461 - accuracy: 0.6521 - val_loss: 0.5563 - val_accuracy: 0.7178
Epoch 3/15
47/47 [=====] - 66s 15/step - loss: 0.5538 - accuracy: 0.7361 - val_loss: 0.4916 - val_accuracy: 0.7567
Epoch 4/15
47/47 [=====] - 67s 15/step - loss: 0.5165 - accuracy: 0.7410 - val_loss: 0.4630 - val_accuracy: 0.7747
Epoch 5/15
47/47 [=====] - 66s 15/step - loss: 0.4493 - accuracy: 0.7832 - val_loss: 0.5134 - val_accuracy: 0.7288
Epoch 6/15
47/47 [=====] - 66s 15/step - loss: 0.3917 - accuracy: 0.8144 - val_loss: 0.4144 - val_accuracy: 0.8265
Epoch 7/15
47/47 [=====] - 66s 15/step - loss: 0.3402 - accuracy: 0.8499 - val_loss: 0.4887 - val_accuracy: 0.7747
Epoch 8/15
47/47 [=====] - 66s 15/step - loss: 0.3336 - accuracy: 0.8622 - val_loss: 0.3567 - val_accuracy: 0.8345
Epoch 9/15
47/47 [=====] - 66s 15/step - loss: 0.2992 - accuracy: 0.8748 - val_loss: 0.3266 - val_accuracy: 0.8465
Epoch 10/15
47/47 [=====] - 66s 15/step - loss: 0.2692 - accuracy: 0.8861 - val_loss: 0.3229 - val_accuracy: 0.8674
Epoch 11/15
47/47 [=====] - 66s 15/step - loss: 0.2404 - accuracy: 0.8994 - val_loss: 0.3626 - val_accuracy: 0.8644
Epoch 12/15
47/47 [=====] - 66s 15/step - loss: 0.2433 - accuracy: 0.9057 - val_loss: 0.3379 - val_accuracy: 0.8495
Epoch 13/15
47/47 [=====] - 66s 15/step - loss: 0.2174 - accuracy: 0.9143 - val_loss: 0.2745 - val_accuracy: 0.8833
Epoch 14/15
47/47 [=====] - 66s 15/step - loss: 0.2123 - accuracy: 0.9180 - val_loss: 0.2375 - val_accuracy: 0.9063
Epoch 15/15
47/47 [=====] - 66s 15/step - loss: 0.1988 - accuracy: 0.9283 - val_loss: 0.3486 - val_accuracy: 0.8604
    
```

Gambar 4. Detail Proses Training data latih dan data validasi melalui hyperparameter tuning Epoch bernilai 15

Jika direpresentasikan dalam grafik, selama 15 kali epoch terjadi peningkatan nilai akurasi secara eksponensial, baik untuk data latih maupun data validasi. Gambar 5 berikut menunjukkan peningkatan akurasi epoch 1 hingga epoch 15 untuk data latih dan data validasi:



Gambar 5. Nilai Akurasi Data Latih dan Data Validasi melalui Epoch 15

4. KESIMPULAN

Hyperparameter tuning merupakan salah satu teknik yang digunakan untuk mengoptimalkan hasil output yang diharapkan, salah satunya adalah nilai akurasi dari pemodelan yang diimplementasikan. Dalam penelitian ini dilakukan *hyperparameter tuning* terhadap nilai Epoch, yang terdiri dari nilai 3, 5, 10, 12 dan 15. Dari hasil *hyperparameter tuning* Epoch, nilai 15 memiliki hasil akurasi yang tertinggi yakni 92,83% untuk akurasi data latih dan 86,04% untuk akurasi data validasi.

Adapun saran dalam penelitian ini adalah nilai epoch dapat ditingkatkan kembali untuk menghasilkan akurasi maksimal dengan memperhatikan kenaikan eksponensial antara data latih dan data validasi sehingga model tetap dalam kondisi ideal.

DAFTAR PUSTAKA

- Anwar, G.A. and Rimirasih, D. (2019) 'Klasifikasi Citra Genus Panthera Menggunakan Metode Convolutional Neural Network (Cnn)', *Jurnal Ilmiah Informatika Komputer*, 24(3), pp. 220–228. doi:10.35760/ik.2019.v24i3.2364.
- Astini, N.K.S. (2019) 'PENTINGNYA LITERASI TEKNOLOGI INFORMASI DAN KOMUNIKASI BAGI GURU SEKOLAH DASAR UNTUK MENYIAPKAN GENERASI MILENIAL', in *Prosiding Seminar Nasional Dharma Acarya ke-1. Tantangan dan Peluang Dunia Pendidikan di Era 4.0*. Denpasar, pp. 113–120.
- Bhairnallykar, P.S. *et al.* (2020) 'Convolutional Neural Network (CNN) for Image Detection', pp. 1239–1243.
- Maulana, F.F. and Rochmawati, N. (2020) 'Klasifikasi Citra Buah Menggunakan Convolutional Neural Network', *Journal of Informatics and Computer Science (JINACS)*, 1(02), pp. 104–108. doi:10.26740/jinacs.v1n02.p104-108.
- Nurfita, R.D. and Ariyanto, G. (2018) 'Implementasi Deep Learning berbasis Tensorflow untuk Pengenalan Sidik Jari', *Emitor: Jurnal Teknik Elektro*, 18(1), pp. 22–27. doi:10.23917/emitor.v18i01.6236.
- Rasyid, M.R., Tahir, Z. and Syafaruddin, N. (2019) 'Digital Image Processing for Detecting Industrial Machine Work Failure with Quantization Vector Learning Method', *Journal Pekommas*, 4(2), p. 131. doi:10.30818/jpkm.2019.2040203.
- S. R. Reynaldi, J. Apri and S. A. Wahyu (2022) 'Klasifikasi Penyakit Kanker Kulit Menggunakan Metode Convolutional Neural Network', *Data Institut Teknologi Telkom Purwokerto*, 2(1), pp. 52–57.
- Santoso, M.H. (2021) 'Application of Association Rule Method Using Apriori Algorithm to Find Sales Patterns Case Study of Indomaret Tanjung Anom', *Brilliance: Research of Artificial Intelligence*,

- 1(2), pp. 54–66. doi:10.47709/brilliance.v1i2.1228.
- Saputra, D.E. and Ibadillah, A.F. (2019) ‘Pengolahan Citra Digital Dalam Penentuan Panen Jamur Tiram’, *Jurnal Teknik Elektro dan Komputer TRIAC*, 6(1), pp. 2–6. doi:10.21107/triac.v6i1.4356.
- Setiawan, D., Candra, J.E. and Suharyanto, C.E. (2019) ‘Perancangan Sistem Pengontrol Keamanan Rumah dengan Smart CCTV Menggunakan Arduino Berbasis Telegram’, *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, 4(1), pp. 185–190. doi:10.30743/infotekjar.v4i1.1598.