

## Deteksi Kecacatan Permukaan Rel Menggunakan Metode Deep Learning Neural Network

**Teguh Arifianto<sup>1\*</sup>, Sunaryo<sup>2</sup>, Sunardi<sup>3</sup>, Akhwan<sup>4</sup>**

<sup>1234</sup> Teknologi Elektro Perkeretaapian, Politeknik Perkeretaapian Madiun, Indonesia

\*teguh@ppi.ac.id

### Abstrak

Rel adalah sebuah konstruksi dalam satu kesatuan yang terbuat dari baja, beton, dan bahan konstruksi lain di atas atau di bawah tanah tergantung dari arah dan areanya. Kondisi permukaan rel tidak boleh ada kecacatan agar perjalanan kereta api menjadi aman dan nyaman. Metode penelitian ini membahas tentang rancangan penelitian. Rancangan penelitian ini terdapat empat proses yaitu proses akuisi data, proses konversi warna citra RGB menjadi HSI, proses filtering menggunakan gabor filter, dan proses klasifikasi menggunakan deep learning neural network. Tujuan dari penelitian ini yaitu membangun sistem untuk mendeteksi cacat permukaan rel kereta api secara visual yaitu teknik pengolahan citra (image processing). Kegiatan ini dilaksanakan di Laboratorium Stasiun Politeknik Perkeretaapian Indonesia Madiun. Berdasarkan penelitian yang telah dilakukan, maka didapatkan kesimpulan bahwa citra dengan ukuran 32x32 piksel menghasilkan nilai akurasi tertinggi pada nilai epoch 90 menggunakan jenis gambar citra filtering gabor filter. Semakin banyak epoch yang digunakan, maka hasilnya akan lebih baik dan mampu membuat model semakin baik. Hasil akurasi yang diperoleh sebesar 0,8041 atau 80,41% untuk akurasi training dan akurasi testing sebesar 0,79 atau 79%.

**Kata kunci :** Deteksi Kecacatan, Rel Kereta Api, Deep Learning Neural Network

### Abstract

Rail is a construction in one unit made of steel, concrete, and other construction materials above or below the ground depending on the direction and area. The condition of the rail surface must not have any defects so that train travel is safe and comfortable. This research method discusses the research design. There are four processes in this research design, namely the data acquisition process, the RGB image color conversion process to HSI, the filtering process using the gabor filter, and the classification process using deep learning neural networks. The purpose of this study is to build a system to visually detect defects in the surface of the railroad tracks, namely image processing techniques. This activity was carried out at the Madiun Indonesian Railways Polytechnic Station Laboratory. Based on the research that has been done, it can be concluded that an image with a size of 32x32 pixels produces the highest accuracy value at epoch 90 using the gabor filter image type. The more epochs used, the better the results will be and the better the model can be made. The accuracy results obtained were 0.8041 or 80.41% for training accuracy and testing accuracy of 0.79 or 79%.

**Keywords :** *Disability Detection, Railways, Deep Learning Neural Networks*

### 1. Pendahuluan

Politeknik Perkeretaapian Indonesia Madiun adalah salah satu perguruan tinggi negeri yang berada di Kementerian Perhubungan yang menyelenggarakan program pendidikan vokasi di bidang perkeretaapian [1] dan dilengkapi dengan

fasilitas untuk mendukung program pendidikan tersebut yaitu Laboratorium Stasiun. Salah satu fasilitas dari laboratorium stasiun yaitu sebuah kereta api inspeksi dan rel yang mengelilingi laboratorium dan workshop Politeknik Perkeretaapian Indonesia Madiun. Rel yang

sering digunakan terus menerus akan mengalami kecacatan pada permukaan rel sehingga mengakibatkan rel tidak dapat berfungsi kembali. Teknologi pengukuran kecacatan pada permukaan rel kereta api sudah banyak dikembangkan.

Kereta api berjalan dengan roda besi sehingga dibutuhkan jalan khusus agar dapat berjalan dengan aman dan baik. Untuk itulah dibuat jalan rel kereta api dengan menggunakan permukaan baja sehingga memiliki karakteristik yang berbeda dengan jalan aspal. Konstruksi rel lebih rumit dan melibatkan komponen yang banyak. Jalan rel kereta api harus dibangun sangat kokoh karena kereta api yang melintas memiliki beban yang sangat berat dan dilewati berulang kali oleh rangkaian kereta api. Konstruksi rel dibuat sebaik mungkin agar mampu menahan beban berat dari rangkaian kereta api yang berjalan di atasnya.

Teknik yang digunakan pada penelitian ini menggunakan teknik pengolahan citra. Terdapat penelitian sebelumnya yang menggunakan deteksi tepi untuk menggambarkan kondisi permukaan rel yaitu metode paling baik menggunakan disk 15 [2]. Permukaan rel yang cacat juga diteliti menggunakan parallel lines laser dengan metode disk 12 karena gambar rel berada di ruang terbuka sehingga efek cahaya lebih lebar [3]. Deteksi kerusakan rel juga dapat menggunakan machine vision [4]. Dalam penelitian tersebut, citra RGB dirubah menjadi

citra HSI dan menggunakan gaussian untuk menghilangkan noise pada citra. Terdapat juga mendeteksi kerusakan menggunakan metode Crack Width Transform (CWT) [5]. Rencana penelitian ini pengembangan dari penelitian sebelumnya yaitu metode deep learning neural network [6]. Hasil dari penelitian ini menerapkan metode deep learning neural network, akan terdeteksi kecacatan pada permukaan rel dan segera untuk diperbaiki kecacatan permukaan rel tersebut.

Penelitian ini bertujuan untuk membangun sistem untuk mendeteksi cacat permukaan rel kereta api secara visual yaitu teknik pengolahan citra (image processing). Pada penelitian ini memiliki judul "Deteksi Kecacatan Permukaan Rel Menggunakan Metode Deep Learning Neural Network".

## 2. Tinjauan Pustaka

### 2.1. Penelitian Terkait

Menurut [7] Kereta api adalah salah satu jenis transportasi darat yang tersedia di masyarakat. Angkutan kereta api adalah jenis angkutan yang bergerak di atas rel. Rel mempunyai fungsi untuk pijakan menggelindingnya roda kereta api dan untuk meneruskan beban dari roda ke bantalan rel. Rel ditumpu oleh bantalan sehingga rel merupakan batang yang ditumpu oleh penumpu. Batang rel terbuat dari besi atau baja yang memiliki tekanan tinggi, mengandung mangan,

karbon, dan silikon. Batang rel memiliki panjang hanya 5-15 meter setiap segmen. Tipe rel yang digunakan di Indonesia yaitu tipe R25, R33, R42, dan R54. Arti dari R25 yaitu batang rel memiliki berat rata-rata 25 kg/m [7].

Perbedaan tipe batang rel dapat mempengaruhi berbagai hal yaitu besarnya tekanan maksimum pada *axle load* yang sanggup diterima rel pada saat kereta api melintas, kecepatan laju kereta yang diperbolehkan untuk melewati rel.

## 2.2. Landasan Teori

### 1. Citra

Citra yaitu kombinasi anatara titik, garis, bidang, dan warna untuk menciptakan suatu imitasi dari suatu objek dan bisa berwujud gambar dua dan tiga dimensi [8][9]. Suatu citra didefinisikan dengan fungsi  $f(x,y)$  yang memiliki ukuran  $M$  baris dan  $N$  kolom. Setiap elemen matrik memiliki posisi koordinat  $x$  dan  $y$  sebagai koordinat spasial. Amplitudo  $f$  di titik koordinat  $(x,y)$  dinamakan intensitas tingkat keabuan dari titik citra pada titik tersebut. Apabila nilai  $x$ ,  $y$ , dan  $f$  secara keseluruhan dan mempunyai nilai diskrit, maka dapat dikatakan bahwa citra tersebut yaitu citra

digital. Setiap elemen matrik citra disebut piksel. Nilai dari intensitas bentuknya adalah mulai dari 0 (minimum) sampai 255 (maksimum). Persamaan (2.1) dan (2.2) menunjukkan posisi koordinat citra digital.

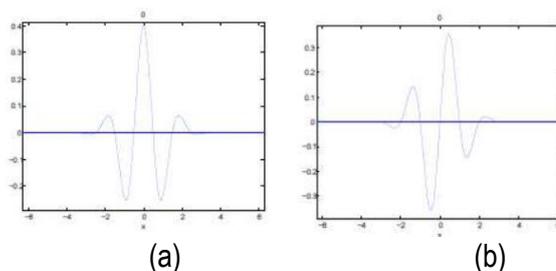
$$0 \leq f \leq \infty \dots\dots\dots (2.1)$$

$$f(x, y) = i(x, y).r(x, y) \dots\dots\dots (2.2)$$

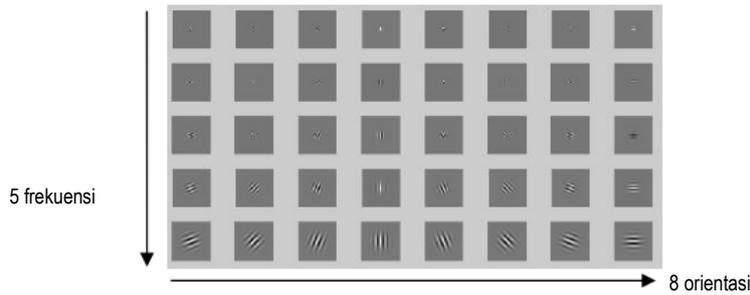
### 2. Proses *preprocessing*

Proses *preprocessing* merupakan suatu tahap pemrosesan data agar dapat digunakan untuk tahap berikutnya.. Proses *preprocessing* memerlukan tahapan untuk kelancaran pada proses berikutnya antara lain menghilangkan *noise*, perbaikan citra, peningkatan kualitas citra, dan *transformasi* [10]. Penelitian ini menggunakan perbaikan citra dengan metode *gabor filter*.

*Gabor filter* diperkenalkan oleh Dennis Gabor pada tahun 1946. Dimensi *gabor filter* pada 1D didefinisikan sebagai perkalian gelombang *cos/sin* dengan jendela *gaussian*. Frekuensi yang digunakan ada lima, yaitu  $u = 0, 1, 2, 3, 4$  dan sudut orientasi ada delapan yaitu  $\theta = 0, 1, 2, 3, 4, 5, 6, 7$  sehingga menghasilkan 40 *gabor response* (gambar 5).



Gambar 1 Gabor cosine (a) dan gabor sine (b)



Gambar 2 Gabor kernel

Ekstraksi ciri mengukur besaran kuantitatif ciri setiap piksel citra. Ekstraksi ciri juga dapat dipilih dari ciri energi yaitu mencari nilai rata tekstur dari *magnitude response*.

$$output = \sqrt{imaginer^2 + real^2} \dots\dots\dots(2.3)$$

$$e(x) = \frac{1}{MN} \sum_{i=1}^m \sum_{j=1}^n |x(m, n)|^2 \dots\dots\dots(2.4)$$

*M* = panjang citra

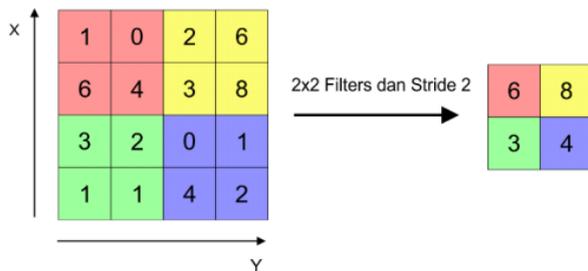
*N* = lebar citra

2.3. Tahapan Penelitian

Metode *deep learning neural network*

*Deep learning* adalah sebuah metode *learning* yang memanfaatkan *multiple non-linier transformation* dan dapat dipandang sebagai gabungan dari *machine learning* dengan *artificial*

*neural network* [11]. *Convolution neural network* digunakan untuk menganalisis gambar secara visual, mendeteksi, dan mengenali objek pada citra. Komponen utama dari *convolution neural network* yaitu *convolution layer*, *pooling layer*, *fully connected layer*, dan *dropout*. *Convolution layer* merupakan lapisan yang utama dan terpenting dalam metode *convolution neural network* [12]. Hasil dari *convolution layer* berupa citra baru yang menunjukkan fitur dari citra *input* (gambar 2.7). *Pooling layer* bekerja di setiap susunan *feature maps* dan mengurangi ukurannya. Bentuk *pooling layer* menggunakan filter berukuran 2x2 yang diaplikasikan dengan langkah sebanyak dua dan beroperasi pada setiap irisan dari *input* (gambar 5).



Gambar 5 Pooling layer [13]

*Feature map* yang dihasilkan pada tahap sebelumnya berbentuk array secara multidimensional sehingga akan melalui proses

*flatten*. Proses *flatten* menghasilkan sebuah vektor yang akan digunakan untuk *input* dari *fully connected layer* [14].

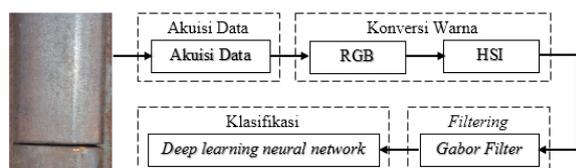
*Dropout* merupakan sebuah proses untuk mencegah terjadinya *overfitting* dan dapat mempercepat proses *learning* [15]. *Overfitting* merupakan kondisi dimana hampir semua data mencapai presentase yang baik setelah melalui proses *training* tetapi terjadi ketidaksesuaian pada proses prediksi

### 3. Metode Penelitian

Metode penelitian ini membahas tentang rancangan penelitian. Rancangan penelitian ini terdapat empat proses yaitu proses akuisi data, proses konversi warna citra RGB menjadi HSI, proses *filtering* menggunakan *gabor filter*, dan proses klasifikasi menggunakan *deep learning neural network*.

#### 3.1. Gambaran umum sistem deteksi kecacatan permukaan rel

Penelitian ini diawali dengan proses akuisi data. Selanjutnya dilakukan konversi warna citra RGB menjadi HSI. Setelah dikonversi citra menjadi HSI, dilanjutkan dengan proses *filtering*. Dalam hal ini proses *filtering* menggunakan *gabor filter*. Tahap terakhir yaitu proses klasifikasi. Proses klasifikasi dalam penelitian ini menggunakan *deep learning neural network*.



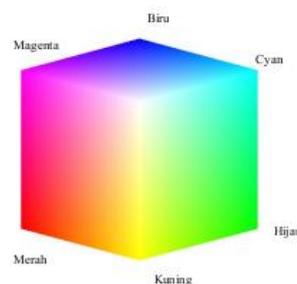
Gambar 8 Desain sistem penelitian

#### 3.2. Proses akuisi data

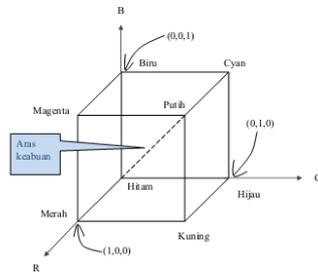
Akuisi data adalah tahap pertama untuk mendapatkan citra digital dengan tujuan untuk mengambil data dari analog permukaan rel menjadi citra permukaan rel menggunakan kamera tanpa merubah fokus kamera dengan resolusi 8 mega piksel [16]. Data yang telah diperoleh akan diolah untuk proses konversi citra. Data rel ini diambil di Laboratorium Politeknik Perkeretaapian Indonesia Madiun

#### 3.3. Proses konversi warna

Citra yang didapatkan dari proses akuisi citra akan dikonversikan menjadi citra RGB atau *true colour*. Model ruang warna menghasilkan ciri khas untuk mewakili sebuah warna. Citra RGB memiliki tiga komponen warna dasar yaitu *red* (merah), *green* (hijau), dan *blue* (biru) [17]. Warna putih berada di ujung kubus yang berseberangan sedangkan warna hitam terdapat pada titik asal (gambar 10).



Gambar 10 Kubus warna tiga dimensi



Gambar 11 Skema ruang warna RGB

HSI merupakan salah satu ruang warna yang merepresentasikan warna yang dilihat oleh mata manusia. H berasal dari kata *hue*, S berasal dari kata *saturation*, dan I berasal dari kata *intensity*. Model HSI dapat dilihat pada gambar 3.4 yang memperlihatkan ruang warna HSI. Konversi dari RGB ke HSI dapat dilakukan dengan persamaan sebagai berikut:

$$H = \begin{cases} 0 & , \text{jika } B \leq G \\ 360 - \theta & , \text{jika } B > G \end{cases} \dots\dots\dots(3.1)$$

Pada persamaan (3.1), *H* menyatakan *hue*. Adapun  $\theta$  diperoleh melalui persamaan berikut:

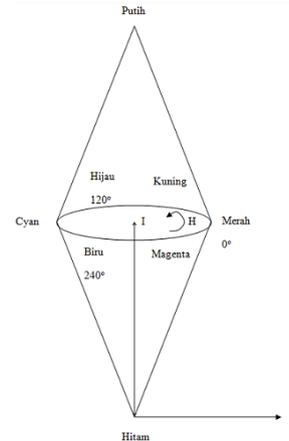
$$\theta = \cos^{-1} \left\{ \frac{1/2[R(R-G)+(R-B)]}{[(R-G)^2(R-B)(G-B)]^{1/2}} \right\} \dots\dots\dots(3.2)$$

Selanjutnya, *S* menyatakan *saturation* dihitung dengan persamaan:

$$S = 1 - \frac{3}{(R+G+B)} [\min(R, G, B)] \dots\dots\dots(3.3)$$

dan komponen *I* menyatakan *intensity* diperoleh dengan menggunakan persamaan:

$$I = \frac{1}{3} (R + G + B) \dots\dots\dots(3.4)$$



Gambar 12 Model HSI

Konversi yang sangat penting dari ruang warna RGB ke HSI adalah konversi dari kubus satuan sistem dengan koordinat *cartesian* pada ruang warna RGB berbasis koordinat kutub silinder dengan model *bipyramid*. Pada proses konversi RGB ke HSI dapat diuraikan menjadi beberapa langkah-langkah berikut:

- lepaskan komponen *Intensitas* (*I*) dari warna RGB ruang angkasa;
- mengurangi ruang warna RGB tiga dimensi menjadi ruang dua dimensi;
- dekomposisikan *chroma* menjadi komponen *Hue* (*H*) dan komponen *Kejenuhan* (*S*); dan
- menghitung komponen *H* dengan titik rumus vektor.

### 3.4. Proses Filtering

Proses *filtering* ini merupakan sebuah proses untuk membuat citra yang kualitasnya turun akibat adanya tambahan *noise* agar menjadi mirip dengan keadaan aslinya. Rencana penelitian ini menggunakan *gabor filter* [18].



Proses selanjutnya yaitu proses *backpropagation*. Hasil akhir dari proses ini menghasilkan akurasi dari klasifikasi yang dilakukan, data yang gagal diklasifikasikan, dan bentuk *network* yang terbentuk dari proses *feed forward*. Dengan bobot dan bias yang baru proses *feed forward* diterapkan kemudian menghasilkan lapisan *output*. Lapisan *output* sudah *fully connected* dengan label yang telah disediakan. Hasil dari *fully connected* diperoleh data yang gagal dan berhasil diklasifikasi

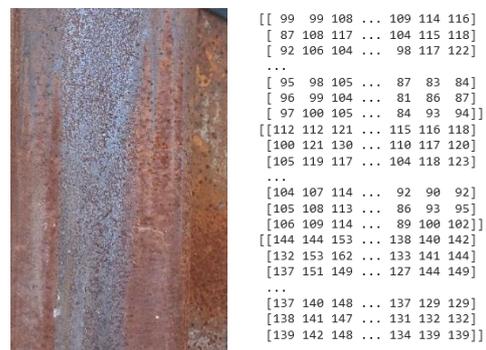
#### 4. Hasil dan Pembahasan

##### Analisis Sistem

Aplikasi ini dibuat menggunakan bahasa pemrograman *python* dan *library OpenCV* dengan *input*-nya adalah citra digital. Citra diperoleh dari pengambilan dengan menggunakan kamera tanpa merubah fokus kamera dengan resolusi 8 mega piksel. Jumlah citra yang digunakan dalam penelitian ini sebanyak 100 citra. Untuk selanjutnya dalam proses klasifikasi menggunakan *software cascade training GUI* sehingga menghasilkan *file* yang memiliki format *.xml* yang isinya berupa fitur dari citra rel. Di dalam tahap akhir akan dilakukan klasifikasi pencocokan citra dan akan dibedakan antara citra rel yang mengalami kecacatan pada permukaan dan tidak mengalami kecacatan pada permukaan rel.

##### Citra RGB

Proses pertama berupa memasukkan citra rel yang akan diujicoba. Dalam penelitian ini penulis menggunakan citra RGB. Citra RGB atau biasa disebut gambar berwarna adalah jenis citra yang memiliki kumpulan piksel warna (red, green, dan blue) dan memiliki nilai dari 0 (hitam) sampai 255 (putih).



(a) (b)  
Gambar 14 Citra (a) dan piksel RGB (b)

##### Konversi Citra RGB ke HSI

Sistem warna citra HSI paling mendekati dengan cara kerja mata manusia. Nilai H terletak antara 0°-360° terhadap *axis* warna merah. Sudut ini menggambarkan warna murni yang mengalami penipisan oleh cahaya putih[20].

Intensitas merupakan warna yang menjelaskan sebuah warna selain *hue* dan *saturation*. Nilai *I=0* menyatakan warna hitam. Intensitas ini sangat cocok dalam menginterpretasikan tingkat warna monokromatis sehingga dengan warna *grayscale* dapat diukur dan mudah dalam menginterpretasikan. Langkah mengubah ruang warna RGB menjadi HSI yaitu sebagai berikut:

$$H = \begin{cases} 0 & , \text{jika } B \leq G \\ 360 - \theta & , \text{jika } B > G \end{cases} \dots\dots\dots(4.1)$$

Pada persamaan (4.1),  $H$  menyatakan *hue*.

Adapun  $\theta$  diperoleh melalui persamaan berikut:

$$\theta = \cos^{-1} \left\{ \frac{1/2[R(R-G)+(R-B)]}{[(R-G)^2(R-B)(G-B)]^{1/2}} \right\} \dots\dots\dots(4.2)$$

Selanjutnya,  $S$  menyatakan *saturation* dihitung

dengan persamaan:

$$S = 1 - \frac{3}{(R+G+B)} [\min(R, G, B)] \dots\dots\dots(4.3)$$

dan komponen  $I$  menyatakan *intensity* diperoleh

dengan menggunakan persamaan:

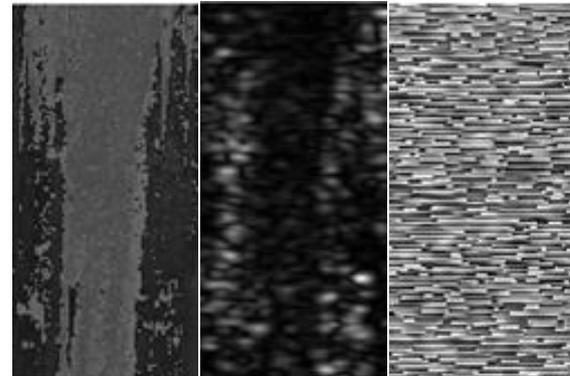
$$I = \frac{1}{3} (R + G + B) \dots\dots\dots(4.4)$$



Gambar 15 Hasil citra HSI

#### Gabor Filter

*Gabor filter* yang digunakan dalam penelitian ini memiliki 5 frekuensi spasial, 8 orientasi, dan menggunakan ukuran *window* 32x32. *Gabor filter* akan mengembalikan dua buah matriks *gabor* dalam bentuk vektor (bilangan riil dan imajiner). Hasil dari keduanya akan dirata-rata menggunakan *numpy.mean()* dan dicatat dalam sebuah *file* untuk diolah secara statistika dan diuji signifikasinya dengan menggunakan *linier regression*. Hasil dari *gabor filter* dapat dilihat pada gambar 16



(a) (b) (c)

Gambar 16. Hasil gabor original (a), gabor magnitude (b), dan gabor phase (c)

#### Data Set Permukaan Rel Kereta Api

Data yang digunakan dalam penelitian ini merupakan data permukaan rel kereta api yang diambil di Laboratorium Politeknik Perkeretaapian Indonesia Madiun. Peneliti menggunakan format citra *\*.jpg*. Data yang diambil sebanyak 100 citra dan data tersebut telah cukup untuk melakukan penelitian mengenai deteksi kecacatan permukaan rel kereta api. Semakin banyak *dataset* yang digunakan maka citra akan lebih meningkatkan akurasi semakin akurat. *Dataset* yang digunakan diambil dari sudut pandang objek yang berbeda-beda agar algoritma ini mampu mengenali objek yang lebih banyak.

#### Preprocessing Dataset

Dari hasil proses *filtering* menggunakan *gabor filter*, penelitian melakukan *preprocessing dataset* yakni melakukan *resize* data untuk menyamakan unsur dari citra yang akan dianalisa dan memperjelas fitur citra. Peneliti melakukan *resize* dari ukuran asli yang besar serta ukurannya

berbeda-beda diubah menjadi sebesar 32x32 piksel dengan kualitas gambar 80%. Ukuran citra yang digunakan yaitu nilai pangkat dua, nilai yang lebih kecil dari 32 adalah 16, sedangkan yang lebih besar dari 32 yaitu 64. Penggunaan *dataset* dengan ukuran 64x64 menjadikan proses *training* menjadi lambat sehingga peneliti menggunakan ukuran *dataset* sebesar 32x32 piksel. Ketika *input* citra berbeda, maka tidak ada pengaruh terhadap hasil klasifikasi.

#### Skenario Dataset Training dan Testing

Pada penelitian ini *dataset* citra dijadikan ke dalam satu folder yang bernama DataRel yang berisi 100

Tabel 1. Skenario pembagian data

Skenario 70% : 30%		Skenario 80% : 20%		Skenario 90% : 10%	
Data <i>training</i>	Data <i>Testing</i>	Data <i>training</i>	Data <i>Testing</i>	Data <i>training</i>	Data <i>Testing</i>
70	30	80	20	90	10

#### Perancangan Convolution Neural Network

Pada tahap ini, peneliti akan melakukan percobaan untuk membandingkan model yang menggunakan *convolution neural network* untuk memperoleh hasil klasifikasi terbaik. Peneliti akan membandingkan nilai *epoch* dan membandingkan

Tabel 2. Perbandingan nilai *epoch*

Data	Epoch 50	Epoch 70	Epoch 90
	Akurasi	Akurasi	Akurasi
<i>Training</i>	0,6457	0,7425	0,8456
<i>Los training</i>	0,4562764	0,3256146	0,3053561
<i>Testing</i>	0,73	0,78	0,79
<i>Los testing</i>	0,6535672	0,8645536	1,2467834

Pada tabel 2. hasil yang diperoleh dari perbandingan *epoch* 50, 70, dan 90 menunjukkan

data citra. Data rel diberi nama kemudian diikuti dengan nomor dimulai dari 1 sampai 100 yang menunjukkan barisan keberapa data tersebut. *Dataset* yang telah tersedia selanjutnya akan ditentukan jumlah datanya yang akan dibagi menjadi data *training* dan *testing*. Peneliti menentukan setiap rel menggunakan data citra sebanyak 100 dimana dengan 3 perbandingan skenario data *training* yaitu 90%, 80%, dan 70%. Berikut tabel skenario pembagian data sebagai berikut:

jumlah data *training* dan data *set*. Nilai *epoch* yang akan dibandingkan pada percobaan ini yaitu *epoch* 50, 70, dan 90. Berdasarkan hasil *training* diperoleh nilai perbandingan nilai akurasi dari masing-masing *epoch*

bahwa semakin besar *epoch*, maka semakin baik hasil akurasi yang diperoleh. Tingkat akurasi

tertinggi pada nilai *epoch* 90. Semakin banyak *epoch* yang digunakan, maka hasilnya akan lebih baik dan mampu membuat model semakin baik.

### Perbandingan Jumlah Data Training dan Data Set

Setelah membandingkan nilai *epoch*, selanjutnya akan membandingkan jumlah data training dan

.Tabel 3. Perbandingan jumlah data *training* dan data set

Data	Skenario 70% : 30%	Skenario 80% : 20%	Skenario 90% : 10%
<i>Training</i>	0,7842	0,8041	0,7642
<i>Los training</i>	0,3213656	0,3042141	0,3432110
<i>Testing</i>	0,7	0,79	0,6
<i>Los testing</i>	0,9345551	1,2536532	1,29445632

Berdasarkan tabel 3. dapat

diketahui bahwa nilai akurasi tertinggi diperoleh dengan menggunakan perbandingan 80%:20% dengan data train 80 dan data set 20. Hasil akurasi yang diperoleh sebesar 0,8041 atau 80,41% untuk akurasi training dan akurasi testing sebesar 0,79 atau 79%.

## 5. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka didapatkan kesimpulan bahwa citra dengan adalah menambahkan data permukaan rel untuk memperoleh akurasi yang lebih optimal dan menambahkan beberapa proses *preprocessing* guna mendapatkan arsitektur yang lebih baik

## 6. Daftar Pustaka

[1] Permenhub, "Peraturan Menteri Perhubungan Republik Indonesia Nomor 18 Tahun 2020 tentang Pengendalian

data set sesuai dengan tabel 3 dengan arsitektur yang lebih optimal sebelumnya. Hasil dari perbandingan jumlah data training dan data set dapat dilihat pada tabel di bawah ini

ukuran 32x32 piksel menghasilkan nilai akurasi tertinggi pada nilai *epoch* 90 menggunakan jenis gambar citra *filtering gabor filter*. Semakin banyak *epoch* yang digunakan, maka hasilnya akan lebih baik dan mampu membuat model semakin baik. Hasil akurasi yang diperoleh sebesar 0,8041 atau 80,41% untuk akurasi *training* dan akurasi *testing* sebesar 0,79 atau 79%. Saran untuk penelitian selanjutnya

Transportasi Dalam Rangka Pencegahan Penyebaran Coronavirus Disease 2019 (COVID-19)," vol. Nomor 9, no. Pedoman Pembatasan Sosial Berskala Besar dalam Rangka Percepatan Penanganan Corona Virus Disease 2019 (COVID-19). Jakarta, pp. 1–71, 2020.

- [2] Budiono and Suwarsono, "Inspection on Rail Quality by using Image Processing Method," *J. Energy, Mech. Mater. Manuf. Eng.*, vol. 4, no. 2, pp. 111–118, 2019.
- [3] Suwarsono, Budiono, K. Hadi, E. Satiawan,

- and M. Prabaswara4, "Karakterisasi dan modifikasi pembakaran sekunder pada kompor biomassa rendah emisi," *Semin. Nas. Teknol. dan Rekayasa*, no. 2017, pp. 61–64, 2018.
- [4] X. Zhou, P. Lu, Z. Zheng, D. Tolliver, and A. Keramati, "Accident Prediction Accuracy Assessment for Highway-Rail Grade Crossings Using Random Forest Algorithm Compared with Decision Tree," *Reliab. Eng. Syst. Saf.*, 2020, doi: 10.1016/j.ress.2020.106931.
- [5] H. Cho, H. J. Yoon, and J. Y. Jung, "Image-Based Crack Detection Using Crack Width Transform (CWT) Algorithm," *IEEE Access*, 2018, doi: 10.1109/ACCESS.2018.2875889.
- [6] R. Thendral and A. Ranjeeth, "Computer vision system for railway track crack detection using deep learning neural network," 2021. doi: 10.1109/ICSPC51351.2021.9451771.
- [7] T. Hadi, S. Sugiharto, S. Supriyo, and ..., "Rancang Bangun Komponen Pendorong Alat Pasang Pendrol Untuk Uji Penambat Rel," *Bangun Rekaprima ...*, vol. 04, pp. 36–44, 2018.
- [8] H. Edha, S. H. Sitorus, and U. Ristian, "Penerapan Metode Transformasi Ruang Warna Hue Saturation Intensity (HSI) Untuk Mendeteksi Kematangan Buah Mangga Harum Manis Hendryanto," *J. Komput. dan Apl.*, vol. 8, no. 1, pp. 1–10, 2020.
- [9] Prasetyo and Eko, *Data mining mengolah data menjadi informasi menggunakan matlab*. Yogyakarta: Penerbit Andi, 2019.
- [10] Lubis and D. Darmawan, "Perbandingan Metode Contrast Stretching Dan Metode Retinex Untuk Peningkatan Citra Digital," Universitas Islam Negeri Sumatera Utara Medan, 2021.
- [11] Primartha, *Belajar Machine Learning Teori dan Praktik*. Bandung: Penerbit Informatika, 2018.
- [12] A. Peryanto, A. Yudhana, and R. Umar, "Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network," *Format J. Ilm. Tek. Inform.*, 2020, doi: 10.22441/format.2019.v8.i2.007.
- [13] A. Santoso and G. Ariyanto, "Implementasi Deep Learning berbasis Keras untuk Pengenalan Wajah," *Emit. J. Tek. Elektro*, 2018, doi: 10.23917/emitor.v18i01.6235.
- [14] P. A. Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network ( Cnn ) Pada Ekspresi Manusia," *Algor*, vol. 2, no. 1, pp. 12–21, 2020.
- [15] H. Abhirawa, M. S. Jondri, and A. Arifianto, "Pengenalan Wajah Menggunakan Convolutional Neural Network Face Recognition Using Convolutional Neural Network," *e-Proceeding Eng.*, vol. 4, no. 3, pp. 4907–4916, 2017.
- [16] Suwarsono, "Railroad Surface Defect Inspection Using Parallel Lines Laser," *Pros. SENTRA (Seminar Teknol. dan Rekayasa)*, vol. 4, 2019.
- [17] A. R. Putri, "Pengolahan Citra Dengan Menggunakan Web Cam Pada Kendaraan Bergerak Di Jalan Raya," *JIPi (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, 2016, doi: 10.29100/jipi.v1i01.18.
- [18] N. Afiyat, "Analisis Restorasi Citra Kabur Algoritma Wiener Menggunakan Indeks Kualitas Citra," *NJCA (Nusantara J. Comput. Its Appl.*, 2017, doi: 10.36564/njca.v2i1.27.
- [19] M. Wasil, H. Harianto, and F. Fathurrahman, "Pengaruh epoch pada akurasi menggunakan convolutional neural network untuk klasifikasi fashion dan furniture," *Info J. Inform. dan Teknol.*, vol. 5, no. 1, pp. 53–61, 2022.
- [20] A. M. Nur and F. Farhurrahman, "Identifikasi Kematangan Buah Mentimun Berbasis Citra Digital Menggunakan Jaringan Syaraf Tiruan Backpropagation," *Infotek J. Inform. dan Teknol.*, vol. 2, no. 1, pp. 27–33, 2019.