



InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600



Available online at : <http://bit.ly/InfoTekJar>

Pengenalan Rambu Lalu-lintas menggunakan Convolutional Neural Network (Studi Kasus: Rambu Lalu-lintas Indonesia)

Mutaqin Akbar

Universitas Mercu Buana Yogyakarta, Jln. Jembatan Merah No.84C, Condongcatur, Depok, Sleman, Daerah Istimewa Yogyakarta, Indonesia 55283

KEYWORDS

Convolutional neural network, rambu lalu-lintas, traffic sign recognition

CORRESPONDENCE

E-mail: mutaqin@mercubuana-yogya.ac.id

A B S T R A K

Salah satu teknologi yang digunakan untuk mengenali rambu lalu-lintas dengan pemanfaatan pengolahan citra adalah *traffic sign recognition* (TSR) atau pengenalan rambu lalu-lintas. TSR dapat diaplikasikan pada *driver assistance systems* (DAS), *advance driver assistance systems* (ADAS), *autonomous driving systems* (ADS), pemahaman suasana perkotaan, keamanan jalan raya, dan pemantauan rambu untuk kepentingan perawatan. Artikel ini menyajikan perbaruan dari pengenalan rambu lalu-lintas di Indonesia menggunakan *convolutional neural network* (CNN). *Dataset* yang digunakan sejumlah 2050 citra rambu lalu-lintas, yang terdiri dari 10 macam rambu lalu-lintas. Model CNN yang digunakan terdiri dari 3 lapisan konvolusi, 3 lapisan penggabungan (*maxpool*) dan 1 lapisan *fully-connected*. *Stochastic gradient descent* (SGD) digunakan sebagai algoritma pelatihan. Skenario terbaik yang diperoleh yaitu menggunakan 32 filter pada lapisan konvolusi pertama, 32 filter pada lapisan konvolusi kedua dan 32 filter pada lapisan konvolusi ketiga, dimana pada tahap pelatihan menggunakan 1750 data citra latih, 20 *epoch* dan laju pelatihan 0,005 didapatkan nilai galat 0,0025 dan nilai akurasi 100%. Dan pada tahap pengujian menggunakan 300 data citra uji, didapatkan nilai galat 0,0159 dan nilai akurasi sebesar 99,33%.

PENDAHULUAN

Pemerintah melalui Peraturan Menteri Perhubungan Nomor 13 tahun 2014, telah mengatur mengenai rambu lalu-lintas di Indonesia. Rambu lalu-lintas itu sendiri adalah bagian perlengkapan jalan yang berupa lambang, huruf, angka, kalimat, dan/atau perpaduan yang berfungsi sebagai peringatan, larangan, perintah, atau petunjuk bagi pengguna jalan. Berdasarkan data dari Dinas Perhubungan, terdapat lebih dari 300 rambu lalu-lintas yang ada di Indonesia, terdiri dari beberapa jenis rambu lalu-lintas sesuai kebutuhan dan fungsinya, antara lain rambu peringatan, rambu larangan, rambu perintah, rambu anjuran dan rambu petunjuk.

Salah satu teknologi yang digunakan untuk mengenali rambu lalu-lintas dengan pemanfaatan pengolahan citra adalah *traffic sign recognition* (TSR). TSR telah diaplikasikan pada *driver assistance systems* (DAS) [1], *advance driver assistance systems* (ADAS) [2], [3], *autonomous driving systems* (ADS) [4], pemahaman suasana perkotaan, keamanan jalan raya, dan pemantauan rambu untuk kepentingan perawatan. Keluaran dari TSR dapat menjadi masukan, baik untuk DAS, ADAS, ADS, maupun sistem lainnya, sehingga akurasi keluaran dari TSR akan

sangat mempengaruhi keputusan yang diambil oleh sistem-sistem tersebut. Kesalahan pengambilan keputusan tentunya akan berdampak fatal dan yang paling buruk adalah terjadinya kecelakaan lalu-lintas yang menimbulkan korban jiwa.

Dalam pengenalan rambu lalu-lintas di Indonesia, beberapa pendekatan atau metode telah dilakukan mulai dari algoritma sederhana dan secara komputasi ringan seperti deteksi bentuk dan perhitungan jarak [5]–[7], klasterisasi [8], sampai dengan klasifikasi dengan algoritma yang lebih berat secara komputasi dan lebih akurat seperti *convolutional neural network* (CNN) [9], [10].

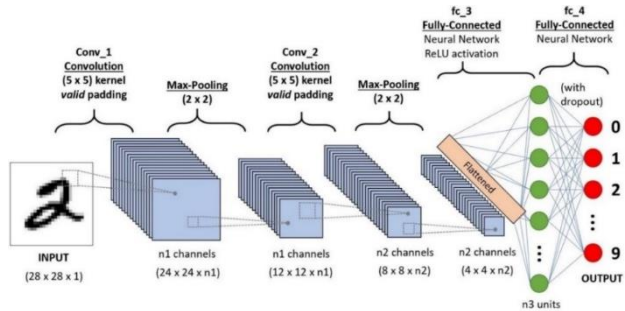
Pendekatan CNN yang telah dilakukan sebelumnya telah mampu mendapatkan nilai akurasi yang sangat baik [10]. Namun, masih terdapat ruang untuk peningkatan mengingat pendekatan CNN tersebut masih menggunakan arsitektur yang sederhana. Artikel ini mengusulkan perbaikan arsitektur CNN sebelumnya sehingga mampu menaikkan nilai akurasi pengenalan rambu lalu-lintas di Indonesia.

Convolutional Neural Network

CNN dicetuskan oleh Yann LeCun dkk pada tahun 1989 [11], merupakan salah satu algoritma pelatihan terbaik untuk

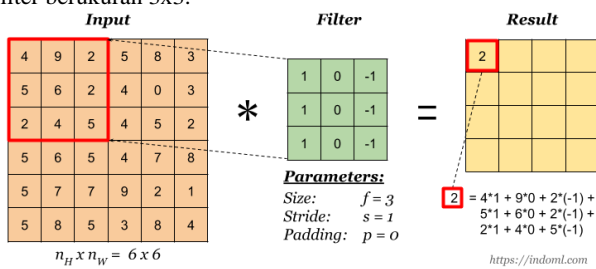
memahami muatan sebuah citra dan sudah menunjukkan kinerja yang baik pada segmentasi citra, klasifikasi citra, deteksi citra dan tugas-tugas sejenisnya [12]. CNN terdiri dari 3 lapisan utama (selain lapisan masukan dan keluaran) yaitu lapisan konvolusi, lapisan pooling dan lapisan fully-connected seperti dapat dilihat pada Gambar 1.

CNN termasuk kedalam jaringan saraf tiruan dengan pelatihan yang disupervisi, dimana membutuhkan sekumpulan data latih yang terdiri dari masukan dan target yang terkait dengan setiap masukan. Jaringan saraf tiruan menggunakan target untuk menentukan seberapa baik pelatihannya, dan sebagai panduan penyesuaian pada nilai bobot untuk mengurangi galat secara keseluruhan [13].



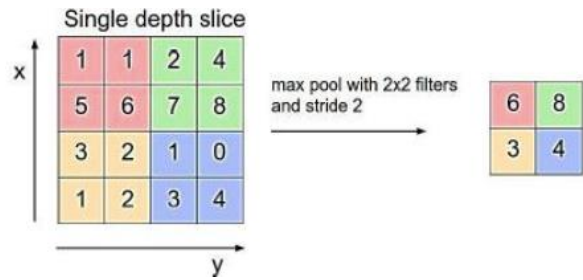
Gambar 1. Contoh arsitektur CNN pada pengenalan angka

Lapisan pertama dari CNN adalah lapisan konvolusi. Konvolusi adalah cara matematis untuk menggabungkan dua sinyal untuk membentuk sinyal ketiga [14]. Sinyal pertama sering disebut sinyal masukan. Sinyal kedua mempunyai perbedaan nama bergantung pada pengaplikasiannya. Pada CNN, sinyal tersebut disebut filter *kernel* atau sering disebut dengan filter saja. Sedangkan sinyal ketiga disebut sinyal keluaran. Masukan pada lapisan konvolusi biasanya merupakan data citra aras keabuan. Jumlah filter yang digunakan pada lapisan konvolusi bermacam-macam, mulai dari 8, 16, 32, dan seterusnya. Filter berukuran 3x3 dengan setiap filter diinisiasi dengan nilai acak. Gambar 2 menjelaskan ilustrasi proses konvolusi pada sebuah masukan dan filter berukuran 3x3.



Gambar 2. Proses konvolusi

Matriks keluaran dari lapisan konvolusi selanjutnya masuk kedalam lapisan penggabungan. Tujuan dari lapisan penggabungan adalah mencapai invariansi spasial dengan mengurangi resolusi matriks keluaran dari lapisan konvolusi. Lapisan pooling mengurangi resolusi matriks dengan meringkas neuron dari ketetangaan spasial yang kecil [15]. Operasi yang digunakan pada lapisan ini adalah penggabungan dengan mempertimbangkan nilai maksimal (*maxpool*), dimana diterapkan fungsi jendela 2x2 untuk menghitung nilai maksimal dari ketetangaan piksel seukuran fungsi jendela tersebut. Keluaran dari lapisan ini yaitu matriks berukuran setengah dari ukuran matriks masukan-nya.



Gambar 3. Proses penggabungan

Selanjutnya matriks keluaran dari lapisan penggabungan harus diubah menjadi sebuah vektor, sebelum masuk kedalam lapisan *fully-connected*. Proses tersebut dinamakan proses perataan (*flatten*).

Sinyal Masukan Jaringan dan Fungsi Aktivasi

Vektor hasil proses perataan diteruskan ke lapisan *fully-connected* dimana akan dilakukan proses klasifikasi. Fungsi aktivasi yang digunakan dalam hal ini adalah fungsi aktivasi *softmax*. Fungsi aktivasi *softmax* menyediakan cara untuk memprediksi distribusi probabilitas dari kelas-kelas yang ada [16]. Berikut merupakan rumus fungsi aktivasi *softmax*:

$$f(y_i) = \frac{e^{y_i}}{\sum_{j=1}^n e^{y_j}} \tag{1}$$

dimana,

$$y = (\sum_{i=1}^n x_i w_i) + b \tag{2}$$

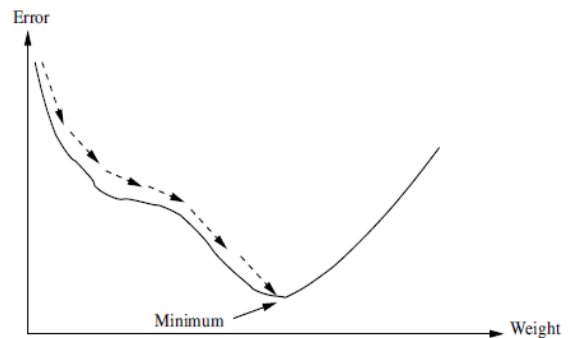
Perhitungan Galat

Setelah melalui proses prediksi tentunya akan dicari galat dari prediksi tersebut berdasar pada target yang akan dituju menggunakan fungsi galat *cross-entropy*.

$$L = -\log(p_c) \tag{3}$$

Dimana p_c merupakan prediksi probabilitas untuk kelas yang benar yakni kelas c . Dengan demikian semakin kecil galat, maka semakin baik pengenalannya.

Penurunan Gradien



Gambar 4. Ilustrasi penurunan gradien

Pelatihan CNN dilakukan dengan melakukan pengaturan terhadap parameter untuk meminimalisir fungsi galat. Pengaturan dan penyesuaian bobot bergantung pada pengaturan bobot dengan fungsi galat terendah (penurunan gradien) dan arah perubahan kemiringan selama penurunan gradien. Pada penelitian ini akan digunakan *stochastic gradient descent* (SGD) dimana memiliki kelebihan dari penurunan gradien total yakni konvergen lebih cepat dan dapat keluar dari *local minimum* [17].

METODE

Data Penelitian

Dataset citra rambu lalu-lintas yang digunakan pada penelitian ini adalah sejumlah 10 macam dan terdiri dari berbagai bentuk seperti lingkaran, belah ketupat dan oktagon, juga merupakan campuran dari rambu peringatan, rambu larangan, rambu perintah, rambu anjuran dan rambu petunjuk.

Pengambilan data citra terbagi menjadi dua cara yaitu pengambilan secara langsung dengan menggunakan kamera dan pengambilan secara tidak langsung melalui aplikasi peta. Data citra diambil dari berbagai sudut pandang, juga diambil pada siang dan malam hari. Hasil pengambilan citra berupa citra berwarna, yang kemudian dipotong pada *region of interest* (ROI) dengan ukuran piksel citra 128x128. Jumlah data citra yang digunakan yaitu 2050 data citra yang dibagi menjadi masing-masing 1750 data citra latih dan 300 data citra uji. Contoh *dataset* citra yang digunakan dapat dilihat pada Tabel 1.

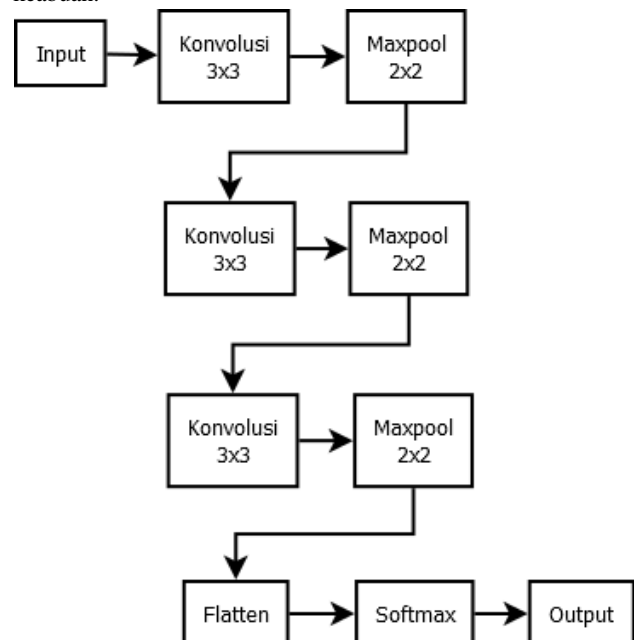
Tabel 1. Contoh *dataset* citra rambu lalu-lintas

No	Contoh 1	Contoh 2	Contoh 3
1			
2			
3			
4			
5			
6			

No	Contoh 1	Contoh 2	Contoh 3
7			
8			
9			
10			

Arsitektur CNN

Data citra yang digunakan nantinya akan dilakukan praproses sebelum masuk kedalam proses konvolusi yaitu akan dilakukan pengaturan ukuran citra menjadi citra berukuran piksel 28x28. Kemudian dilakukan proses perubahan ruang warna menjadi aras keabuan.



Gambar 5. Arsitektur CNN yang digunakan

Pada Gambar 5, dapat diamati bahwa sebelum proses perataan terdapat 3 lapisan konvolusi dan 3 lapisan penggabungan

Tabel 2. Hasil pelatihan untuk setiap skenario

epoch	Skenario 1		Skenario 2		Skenario 3		Skenario 4	
	Galat	Akurasi (%)	Galat	Akurasi (%)	Galat	Akurasi (%)	Galat	Akurasi (%)
1	2,0677	26,97	1,9684	31,14	2,0157	29,2	2,1461	23,14
2	0,8985	74,23	0,7035	80,69	0,7788	77,77	1,0849	68,11
3	0,323	91,49	0,2875	92,51	0,2903	92,63	0,3452	91,26
4	0,1875	95,14	0,1876	94,63	0,1799	94,86	0,198	94,86
5	0,1293	96,51	0,1209	96,8	0,1144	97,37	0,1253	96,51
6	0,0844	97,37	0,0963	97,54	0,0966	97,14	0,0761	97,77
7	0,0728	97,83	0,0503	98,34	0,0704	97,83	0,0552	98,46
8	0,0572	98,4	0,0376	99,31	0,0542	98,57	0,033	99,43
9	0,0324	99,37	0,0196	99,89	0,0319	99,37	0,0239	99,77
10	0,0218	99,66	0,0133	99,89	0,0181	99,77	0,0134	100
11	0,0188	99,66	0,0117	99,89	0,0136	99,89	0,0091	99,94
12	0,0135	99,77	0,0078	99,89	0,0085	100	0,007	99,94
13	0,0105	99,94	0,0047	100	0,0068	100	0,0047	100
14	0,0058	100	0,0037	100	0,0055	100	0,0038	100
15	0,0048	100	0,003	100	0,0045	100	0,0033	100
16	0,0039	100	0,0027	100	0,004	100	0,0031	100
17	0,0034	100	0,0024	100	0,0034	100	0,0025	100
18	0,003	100	0,0021	100	0,0031	100	0,0023	100
19	0,0027	100	0,002	100	0,0028	100	0,0021	100
20	0,0025	100	0,0018	100	0,0025	100	0,0019	100

menggunakan *maxpool*. Jumlah filter yang digunakan pada setiap lapisan konvolusi dibagi menjadi 4 yaitu:

- Skenario 1: 32 filter pada lapisan konvolusi pertama, 32 filter pada lapisan konvolusi kedua dan 16 filter pada lapisan konvolusi ketiga
- Skenario 2: 48 filter pada lapisan konvolusi pertama, 32 filter pada lapisan konvolusi kedua dan 16 filter pada lapisan konvolusi ketiga
- Skenario 3: 32 filter pada lapisan konvolusi pertama, 32 filter pada lapisan konvolusi kedua dan 32 filter pada lapisan konvolusi ketiga
- Skenario 4: 48 filter pada lapisan konvolusi pertama, 32 filter pada lapisan konvolusi kedua dan 32 filter pada lapisan konvolusi ketiga

Kemudian setelah melalui proses flattening, akan masuk ke lapisan fully-connected dan diklasifikasi berdasarkan fungsi aktivasi softmax seperti pada rumus (1) dan (2). Hasil dari klasifikasi menggunakan fungsi aktivasi softmax adalah probabilitas untuk setiap kelasnya. Perhitungan galat menggunakan rumus (3) dilakukan terhadap hasil klasifikasi dan selanjutnya digunakan pada tahap pelatihan. Selanjutnya adalah tahap pelatihan dimana dilakukan proses runut mundur mulai dari hasil sampai ke citra awal dengan parameter-parameter yang akan dibahas pada Bagian 4.

Model yang sudah dilatih akan diuji dengan data citra uji yang merupakan data baru yang bukan merupakan bagian dari data citra latih.

HASIL DAN PEMBAHASAN

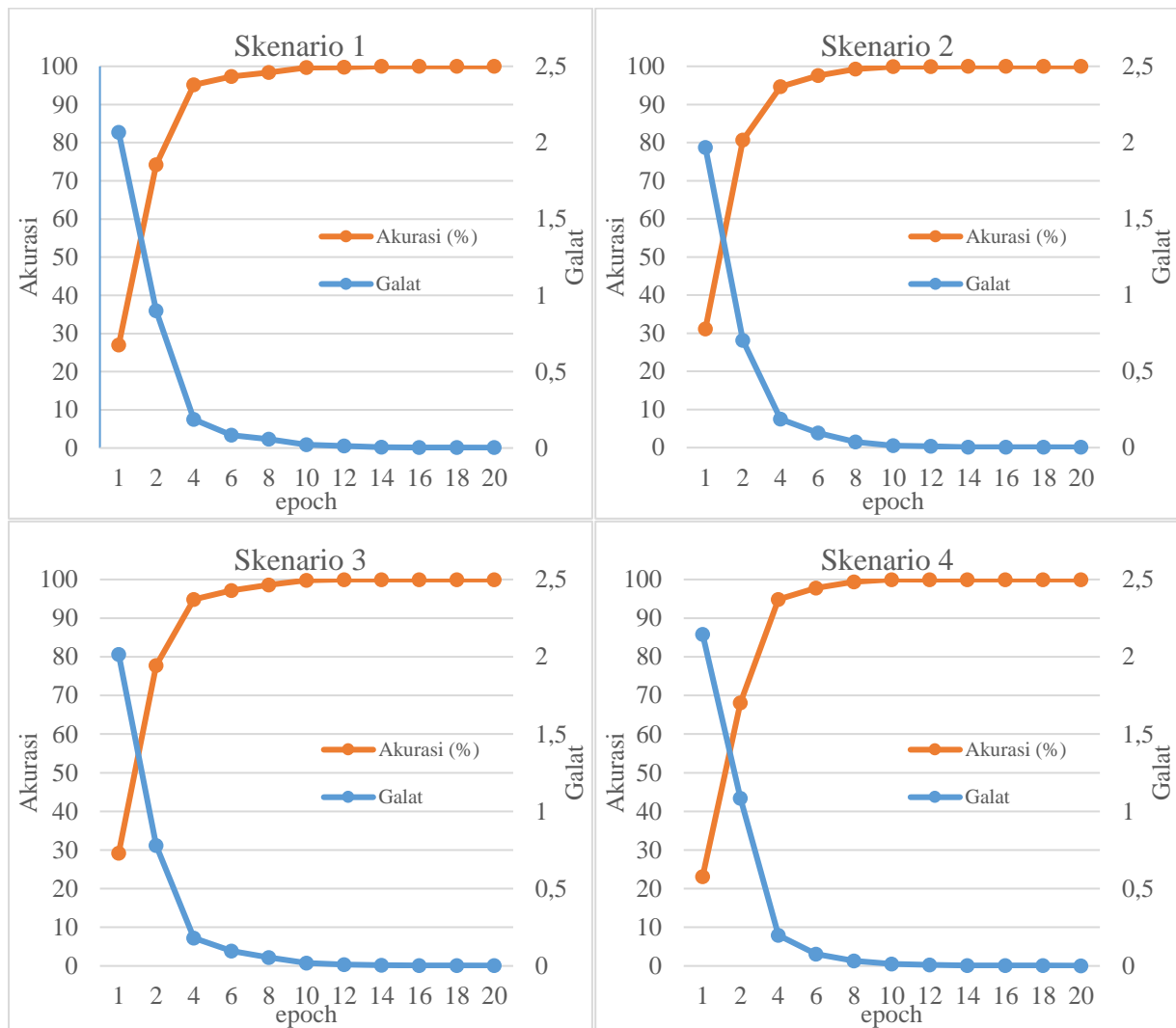
Parameter pelatihan yang digunakan adalah laju pelatihan dan *epoch* masing-masing yaitu 0,005 dan 20. Parameter yang akan dilatih sendiri terdiri dari filter konvolusi, bobot dan bias lapisan fully-connected. Pada bagian ini akan dibagi menjadi 2 yaitu bagian pelatihan dan bagian pengujian. Program dibuat menggunakan bahasa pemrograman *python* dan pustaka *tensorflow*.

Pelatihan

Berdasarkan pada Tabel 2, pada tahap pelatihan menggunakan skenario 1 dapat dilihat nilai galat stabil turun dari 2,0677 pada *epoch* ke-1 dan 0,0025 pada *epoch* ke-20. Nilai akurasi pada skenario ini juga stabil naik dan mampu mencapai nilai akurasi 100% mulai dari *epoch* ke-14 sampai dengan *epoch* ke-20.

Pada skenario 2 nilai galat terlihat stabil turun dan mencapai nilai 0,0018 pada *epoch* ke-20, lebih baik 0,0007 dari skenario 1. Dan nilai akurasi mencapai nilai 100% mulai dari *epoch* ke-13. Sedangkan pada skenario 3 nilai galat pada *epoch* ke-20 yaitu 0,0025, lebih buruk dari skenario 2. Tetapi, nilai akurasi pada skenario 3 mencapai nilai 100% mulai dari *epoch* ke-12.

Pada skenario terakhir yaitu skenario 4, nilai galat terlihat stabil turun hingga mencapai 0,0019 pada *epoch* ke-20. Dan nilai akurasi terlihat terjadi fluktuasi yaitu pada *epoch* ke-11 dan 12,



Gambar 6. Grafik galat dan akurasi untuk setiap skenario

Tabel 3. Hasil pengujian untuk setiap skenario

Skenario 1		Skenario 2		Skenario 3		Skenario 4	
Galat	Akurasi (%)	Galat	Akurasi (%)	Galat	Akurasi (%)	Galat	Akurasi (%)
0,052	99	0,0251	99,33	0,0159	99,33	0,0165	99,33

walaupun demikian nilai galat mulai dari *epoch* ke-10 sampai *epoch* ke-14 tidak terjadi fluktuasi. Grafik nilai galat dan akurasi dapat dilihat pada Gambar 6.

Program dijalankan pada *graphical processing unit* (GPU) untuk mempercepat proses pelatihan untuk setiap skenario. Dengan menggunakan GPU, waktu diperlukan untuk melatih model pada setiap skenario yaitu 80 detik.

Pengujian

Berdasarkan pada Tabel 3, nilai galat terkecil dari keempat skenario terdapat pada skenario 3 yaitu 0,0159. Sedangkan nilai akurasi yang didapatkan oleh semua skenario sudah sangat baik yaitu lebih dari sama dengan 99%, dimana nilai akurasi terbaik adalah 99,33% pada skenario 2, 3 dan 4. Jika menimbang nilai galat terkecil dan nilai akurasi terbesar, maka skenario 3 lebih baik dari keempat skenario.

KESIMPULAN

CNN digunakan pada pengenalan rambu lalu-lintas, khususnya rambu lalu-lintas di Indonesia. Model CNN dilatih menggunakan 1750 data citra latih dan diuji menggunakan 300 data citra uji. Berdasarkan pengujian yang telah dilakukan, skenario yang mampu menghasilkan nilai galat terkecil dan nilai akurasi terbesar pada tahap pengujian adalah skenario 3 yaitu 32 filter pada lapisan konvolusi pertama, 32 filter pada lapisan konvolusi kedua dan 32 filter pada lapisan konvolusi ketiga. Nilai galat dan nilai akurasi yang diperoleh dari skenario tersebut yaitu masing-masing 0,0159 dan 99,33%.

DAFTAR PUSTAKA

- [1] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three Decades of Driver Assistance Systems: Review and Future Perspectives," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 4, pp. 6–22, 2014, doi: 10.1109/ITS.2014.2336271.
- [2] V. K. Kukkala, J. Tunnell, S. Pasricha, and T. Bradley, "Advanced Driver-Assistance Systems: A Path Toward Autonomous Vehicles," *IEEE Consum. Electron. Mag.*, vol. 7, no. 5, pp. 18–25, Sep. 2018, doi: 10.1109/MCE.2018.2828440.
- [3] A. Ziebinski, R. Cupek, D. Grzechca, and L. Chruszczyk, "Review of advanced driver assistance systems (ADAS)," Thessaloniki, Greece, 2017, p. 120002. doi: 10.1063/1.5012394.
- [4] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, Jun. 2011, pp. 163–168. doi: 10.1109/IVS.2011.5940562.
- [5] O. Rumiiris Sitanggang, H. Fitriyah, and F. Utaminigrum, "Sistem Deteksi dan Pengenalan Jenis Rambu Lalu Lintas Menggunakan Metode Shape Detection Pada Raspberry Pi," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 2, no. 12, pp. 6108–6117, Dec. 2018.
- [6] T. Oddy Chrisdwianto, H. Fitriyah, and E. Rosana Widarsari, "Perancangan Sistem Deteksi dan Pengenalan Rambu Peringatan Menggunakan Metode Template Matching," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 2, no. 3, pp. 1265–1274, Mar. 2018.
- [7] G. Romadhon and Murinto, "Aplikasi pengenalan citra rambu lalu lintas berbentuk lingkaran menggunakan metode jarak city-block," *J. Sarj. Tek. Inform.*, vol. 2, no. 2, pp. 286–294, Jun. 2014.
- [8] C. Rahmad, I. Fauziah Rahmah, and R. Andrie Asmara, "Deteksi dan pengenalan rambu lalu lintas di indonesia menggunakan RGBN dan Gabor," in *SENTRINOV*, 2017, vol. 3, pp. T113–22.
- [9] A. Triyadi and F. Utaminigrum, "Pengembangan Sistem Rekognisi Rambu Kecepatan Menggunakan Circle Hough Transform dan Convolutional Neural Network Berbasis Raspberry Pi," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 2, no. 1, pp. 56–64, Jan. 2020.
- [10] M. Akbar, "Traffic sign recognition using convolutional neural networks," *J. Teknol. Dan Sist. Komput.*, vol. 9, no. 2, pp. 120–125, Apr. 2021, doi: 10.14710/jtsiskom.2021.13959.
- [11] Y. Le Cun *et al.*, "Handwritten Digit Recognition: Applications of Neural Net Chips and Automatic Learning," in *Neurocomputing*, F. F. Soulié and J. Héroult, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 303–318. doi: 10.1007/978-3-642-76153-9_35.
- [12] A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020, doi: 10.1007/s10462-020-09825-6.
- [13] A. P. Engelbrecht, *Computational intelligence: an introduction*, 2nd ed. Chichester, England ; Hoboken, NJ: John Wiley & Sons, 2007.
- [14] S. W. Smith, *The scientist and engineer's guide to digital signal processing*. San Diego, Calif.: California Technical Pub., 1999.
- [15] D. Hutchison *et al.*, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," in *Artificial Neural Networks – ICANN 2010*, vol. 6354, K. Diamantaras, W. Duch, and L. S. Iliadis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–101. doi: 10.1007/978-3-642-15825-4_10.
- [16] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," *ArXiv181103378 Cs*, Nov. 2018, Accessed: Nov. 17, 2021. [Online]. Available: <http://arxiv.org/abs/1811.03378>
- [17] B. Zhou, C. Han, and T. Guo, "Convergence of Stochastic Gradient Descent in Deep Neural Network," *Acta Math. Appl. Sin. Engl. Ser.*, vol. 37, no. 1, pp. 126–136, Jan. 2021, doi: 10.1007/s10255-021-0991-2.

NOMENCLATURE

f	fungsi aktivasi
x	masukan
w	bobot
b	bias
y	sinyal masukan jaringan ke fungsi aktivasi

AUTHOR(S) BIOGRAPHY

**Mutaqin Akbar**

Meraih gelar Magister Teknik Informatika dari Universitas Atma Jaya Yogyakarta pada tahun 2018. Saat ini, dia bekerja sebagai tenaga pendidik dan peneliti pada Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta. Minat penelitiannya meliputi jaringan saraf tiruan, *machine learning* dan komputasi paralel.