

# Pengenalan Karakter Optikal Aksara Jawa Menggunakan *Connectionist Temporal Classification*

Ridwan Hidayatullah<sup>1</sup>, Wahyu Tjahjo Saputro<sup>2</sup>, Dewi Chirzah<sup>3\*</sup>

<sup>1,2,3</sup> Teknologi Informasi, Universitas Muhammadiyah Purworejo, 54111, Indonesia

24.ridwanhidayatullah.9b@gmail.com, wahjusaputro@umpwr.ac.id, dewichirzah@umpwr.ac.id

## Abstrak

Aksara Jawa memiliki sejarah panjang dan penting di Pulau Jawa. Sampai saat ini aksara Jawa banyak digunakan untuk obyek penelitian. Salah satunya menggunakan aplikasi OCR. Penelitian ini bertujuan mengembangkan model OCR untuk aksara Jawa menggunakan Metode CTC (*Connectionist Temporal Classification*). Pengumpulan *dataset* diambil dari Wikipedia, kemudian diubah ke dalam aksara Jawa menggunakan aplikasi transliterasi. Selanjutnya teks aksara Jawa diubah menjadi gambar dan pelatihat menggunakan Metode CTC. Penelitian ini memilih Metode CTC karena mampu mengatasi masalah pelabelan urutan yang tidak sejajar, dengan memprediksi urutan karakter langsung dari input gambar. *Dataset* akhir terdiri dari 25.038 buah teks yang diubah ke dalam gambar dengan resolusi acak dan seragam. Model seragam menunjukkan kinerja lebih baik dari pada model acak dalam pengenalan gambar aksara Jawa, dengan *mean edit distance*, *train loss*, dan *validation loss* yang lebih rendah. Model seragam efektif dan baik untuk data gambar seragam, sementara model acak lebih fleksibel untuk data berbagai ukuran gambar. Peningkatan ukuran *dataset* meningkatkan performa kedua model.

**Kata kunci:** *Optical Character Recognition, Connectionist Temporal Classification, Aksara Jawa*

## Abstract

*Javanese script has a long and important history in Java. Until now, Javanese script is widely used for research objects. One of them is using OCR application. This research aims to develop an OCR model for Javanese script using the CTC (Connectionist Temporal Classification) method. Dataset collection is taken from Wikipedia, then converted into Javanese script using transliteration application. Furthermore, the Javanese script text is converted into images and trained using the CTC Method. This research chose the CTC Method because it is able to overcome the problem of misaligned sequence labeling, by predicting the character sequence directly from the input image. The final dataset consists of 25,038 pieces of text converted into images with random and uniform resolution. The uniform model shows better performance than the random model in Javanese script image recognition, with lower mean edit distance, train loss, and validation loss. The uniform model is effective and good for uniform image data, while the random model is more flexible for data of various image sizes. Increasing the dataset size improves the performance of both models.*

**Keywords:** *Optical Character Recognition, Connectionist Temporal Classification, Javanese Script*

## 1. PENDAHULUAN

Aksara Jawa, yang dikenal sebagai Hanacaraka, merupakan hasil evolusi panjang dari aksara Brahmi di India, dengan berbagai pengaruh dan modifikasi sepanjang sejarahnya. Aksara ini terinspirasi dari aksara Pallawa India Selatan dan menjadi induk berbagai aksara di Nusantara. Aksara Jawa banyak digunakan dalam sastra dan tulisan sehari-hari selama berabad-abad, dan kini menjadi bagian penting dalam identitas dan budaya masyarakat Jawa (Fakhrudin et al., 2019; Mahakarti et al., 2023; Standard, n.d.). Aksara Jawa di Indonesia memiliki nilai sangat tinggi,

keunikan Aksara menjadi salah satu hal untuk dilestarikan (Hanindria & Hendry, 2022; Mahakarti et al., 2023).

Unicode adalah standar internasional yang digunakan untuk karakter teks dari berbagai bahasa, termasuk aksara Jawa (Standard, n.d.), (Lindenberg, 2022). Ini memungkinkan teks untuk ditulis dan ditampilkan secara konsisten di berbagai *platform*, perangkat, dan aplikasi yang mendukung Unicode (Standard, n.d.). Blok Unicode untuk Aksara Jawa adalah U+A980–U+A9DF, yang terdiri dari 91 kode titik yang mencakup 53 huruf, 19 tanda baca, 10 angka,

dan 9 vokal (Standard, n.d.). Seiring dengan perkembangan zaman, penggunaan Unicode telah menjadi standar dalam representasi teks aksara Jawa di dunia (Lindenberg, 2022).

Salah satu hambatan utama membaca dan memahami literatur beraksara Jawa adalah kesulitan mentransliterasi gambar aksara Jawa menjadi teks latin secara manual. Aplikasi *Optical Character Recognition* yang populer merupakan teknologi untuk mengonversi gambar teks. Seperti dokumen yang dipindai atau foto teks, menjadi teks yang dapat diedit secara digital (Mulyanto et al., 2021). Proses OCR melibatkan pengenalan pola karakter dalam gambar, mengidentifikasi karakter tersebut, dan mengonversi menjadi teks yang dapat disunting di komputer (Khatavkar et al., 2024; Mulyanto et al., 2021; Nisha et al., 2024).

Klasifikasi menjadi metode yang digunakan untuk mengubah aksara jawa ke dalam bentuk teks. Seperti yang telah dilakukan oleh (Hanindria & Hendry, 2022; Mulyanto et al., 2021) menggunakan metode *Convolutional Neural Network* (CNN) untuk mengklasifikasikan aksara jawa. Sedangkan (Mulyanto et al., 2021) mengimplementasikan CNN untuk pengenalan aksara lampung berbasis OCR. Kemudian OCR juga telah diimplementasikan oleh (Khatavkar et al., 2024) mengklasifikasi teks captcha.

Banyak penelitian Model CTC digunakan untuk data tidak tersegmentasi (Graves et al., 2006), sequence-to-sequence (B. R. Lu et al., 2017), teks captcha ((Khatavkar et al., 2024), sensor lingkungan (Mäs et al., 2018), kombinasi pola string (Zhan et al., 2017), metode efisiensi meori (S. Lu et al., 2019), pola suara (Zenkel et al., 2017), dan data label sequensial (Hou et al., 2020). Selain itu, CTC bisa digunakan untuk mengompresi urutan masukan sesuai dengan karakteristik fonetik dari data berupa suara (Gaido et al., 2021).

Hasil dari identifikasi penelitian ini aksara Jawa memiliki kemiripan jenis data dengan penelitian sebelumnya. Implementasi metode OCR untuk pengenalan karakter optikal Aksara Jawa dapat diimplementasikan berdasarkan dataset berupa gambar Aksara Jawa. Sedangkan metode pembelannya dapat menggunakan CTC.

## 2. METODE

OCR merupakan metode yang mampu menerjemahkan image character untuk diubah menjadi

bentuk teks, dengan menyesuaikan pola per blok dengan pola yang sudah tersimpan dalam database (Kusnanto et al., 2022). OCR sebagai bagian dari *image processing* merupakan teknik menerjemahkan karakter teks yang ditulis tangan, diketik, atau dicetak menjadi teks yang dikodekan mesin. OCR dapat diimplementasikan dalam berbagai hal seperti Pengenalan Pola Berbasis OCR untuk Pengambilan Data Bursa Saham (Okta et al., 2021). Deteksi Plat Nomor Kendaraan Pada Sistem Parkir (Kusnanto et al., 2022), *image processing* untuk audio output (Ganesh, 2019).

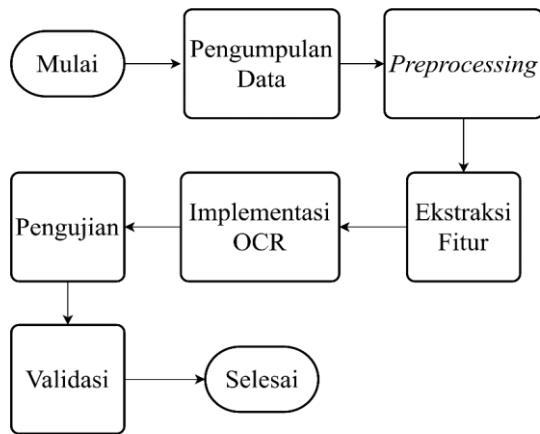
CTC adalah kriteria pelatihan yang dirancang untuk masalah pelabelan sekuens di mana penyelarasan antara masukan dan label target tidak diketahui. Salah satu langkah kuncinya adalah menambahkan simbol kosong ke kosakata target (Chao et al., 2020). Kemudian (Feng et al., 2019) mengimplementasikan CTC untuk pengenalan karakter optik Tiongkok pada dataset tidak seimbang. Dalam penelitiannya dihasilkan bahwa CTC memiliki performa yang baik untuk dataset tidak seimbang. Sedangkan (Susetianingtiyas et al., 2023) menggunakan model RNN dan CTC untuk mengenali tulisan tangan sambung offline dengan mengatasi kesulitan pengenalan pada kesamaan bentuk dan pola huruf untuk dapat digunakan pada sistem pengenalan tulisan tangan.

CTC secara eksplisit digunakan untuk melatih masalah sekuensial, penyelarasan monotonik antara fitur ucapan dan transkripsi karakter (Wigington dkk., 2019) (Karita dkk., 2019). Menurut (Salazar dkk., 2019), CTC telah matang sebagai pendekatan transduksi sekuens yang bebas penyelarasan dan non-autoregresif, baik dengan sendirinya maupun dalam berbagai kerangka kerja multitugas dan decode. Maka, pada penelitian ini mengimplementasikan CTC untuk melakukan pelabelan.

Seperti pada penelitian (Khatavkar et al., 2024) yang mengimplementasikan CTC dan OCR untuk mengklasifikasikan text Captcha. Dalam penelitiannya, diperoleh akurasi sebesar 99,80%. Metode CTC dan OCR diimplementasikan pada pengenalan karakter optikal aksara Jawa diterapkan dalam penelitian ini sebagaimana Gambar 1.

Terdapat beberapa tahapan sistematis dalam penelitian ini seperti pada Gambar 1 dalam pengembangan sistem pengenalan karakter optikal aksara Jawa. Proses dimulai dengan pengumpulan data, diikuti oleh langkah *prepro-*

cessing untuk mempersiapkan data. Selanjutnya, ekstraksi fitur dilakukan untuk mengidentifikasi elemen penting dari data. Implementasi OCR mengikuti dengan pelatihan model menggunakan metode CTC. Setelah model dibangun, tahap pengujian dilakukan untuk mengevaluasi kinerja, diakhiri dengan validasi untuk memastikan keandalan dan efektivitas model.



Gambar 1 Bagan alir tahapan penelitian

Pengumpulan Data adalah langkah awal dalam penelitian yang melibatkan pengumpulan gambar aksara Jawa dari satu jenis font tertentu. Proses ini dilakukan melalui observasi langsung terhadap teks digital, manuskrip, atau dokumen fisik untuk memperoleh dataset yang diperlukan. Dengan mengumpulkan data dari sumber yang konsisten, penelitian dapat memastikan bahwa model yang dikembangkan akan dilatih dengan data yang sesuai dan berkualitas.

*Preprocessing* mencakup langkah-langkah untuk mempersiapkan data sebelum digunakan dalam pelatihan model. Ini termasuk normalisasi gambar dengan mengubah ukuran dan konversi ke format *grayscale* dan menyamakan ukuran gambar yang berbeda. Selain itu, *dataset* dibagi menjadi data pelatihan, validasi, dan pengujian untuk memastikan model dapat dilatih dan dievaluasi secara efektif.

Ekstraksi Fitur melibatkan identifikasi dan penarikan fitur penting dari gambar aksara yang akan digunakan dalam pelatihan model. Teknik seperti *Convolutional Neural Networks* (CNN) digunakan untuk mengekstrak pola-pola karakteristik dari gambar aksara Jawa, yang akan membantu model dalam mengenali karakter dengan lebih akurat. Fitur-fitur ini menjadi input utama bagi model OCR untuk proses selanjutnya.

Implementasi OCR adalah tahap di mana model OCR dibangun dan dilatih menggunakan teknik CTC. Model yang dikembangkan dilatih dengan dataset yang telah diproses dan fitur yang diekstrak untuk mengenali karakter aksara Jawa. Proses ini melibatkan pengaturan parameter pelatihan, pemilihan *optimizer*, dan evaluasi model selama pelatihan untuk memastikan kinerja yang optimal.

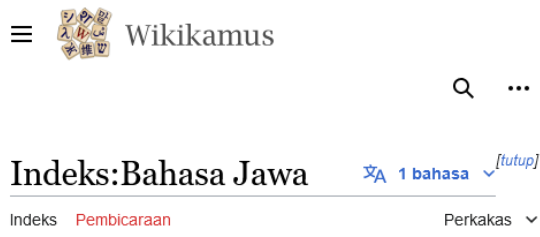
Pengujian dilakukan untuk mengevaluasi kinerja model pada data yang belum pernah dilihat sebelumnya. Data pengujian digunakan untuk mengukur akurasi dan efektivitas model dalam mengenali aksara Jawa. Hasil dari pengujian memberikan gambaran tentang kemampuan model dalam menghadapi data nyata dan mengidentifikasi area yang mungkin perlu diperbaiki.

Validasi melibatkan pemeriksaan model menggunakan data validasi untuk memastikan bahwa model tidak hanya berfungsi dengan baik pada data pengujian tetapi juga dapat diandalkan dalam situasi nyata. Validasi memastikan bahwa model mampu menggeneralisasi dengan baik di luar data pelatihan dan pengujian, serta menilai konsistensi dan keberhasilan model dalam pengenalan aksara Jawa secara keseluruhan.

### 3. HASIL DAN PEMBAHASAN

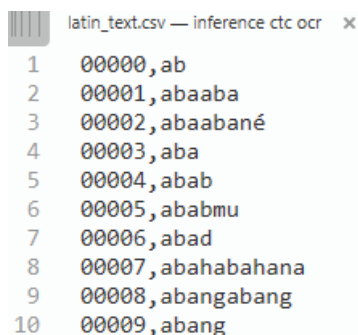
Dalam penelitian ini, pengumpulan dataset teks aksara Jawa dilakukan melalui sumber daring yaitu Wiktionary Bahasa Jawa, yang dapat diakses melalui [https://id.wiktionary.org/wiki/Indeks:Bahasa\\_Jawa](https://id.wiktionary.org/wiki/Indeks:Bahasa_Jawa) seperti yang ditunjukkan pada Gambar 2 Teks yang memiliki warna biru memiliki arti bahwa teks tersebut dapat diakses halamannya sedangkan bagian berwarna merah menandakan belum ada halaman untuk teks tersebut. Teks perlu diolah dengan menghapus tanda baca yang tidak perlu agar memudahkan dalam perubahan ke dalam aksara Jawa.

Proses awal pengumpulan berhasil mengidentifikasi dan mengunduh sebanyak 24.644 kosakata bahasa Jawa. Setelah itu, dilakukan penyaringan untuk memastikan kualitas dataset. Kosakata yang memiliki panjang lebih dari 15 huruf dihapus untuk menjaga konsistensi dan relevansi data, sehingga jumlah kosakata bahasa Jawa yang tersisa menjadi 24.038.

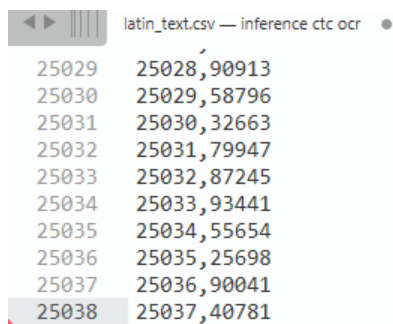


Gambar 2 Kosakata bahasa Jawa

Selanjutnya, untuk memperkaya dataset dan meningkatkan variasi data, ditambahkan 1.000 nomor acak. Sehingga jumlah kosakata dalam dataset final yang digunakan untuk penelitian ini adalah 25.038 kosakata. Dataset yang telah dipersiapkan ini dapat dilihat pada gambar 3 dan gambar 4. Gambar 3 menunjukkan indeks, sedangkan pada Gambar 4 merupakan teks Bahasa Jawa yang dipisahkan dengan tanda koma untuk membedakan antara indeks dan teksnya.



Gambar 3. dataset teks yang telah diolah



Gambar 4. dataset teks yang telah diolah

Pembuatan dataset gambar aksara Jawa dilakukan dengan menggunakan aplikasi transliterasi yang bersumber dari <https://github.com/bennylin/transliterasijawa> dan font yang digunakan adalah NotoSansJavanese-Regular.ttf.

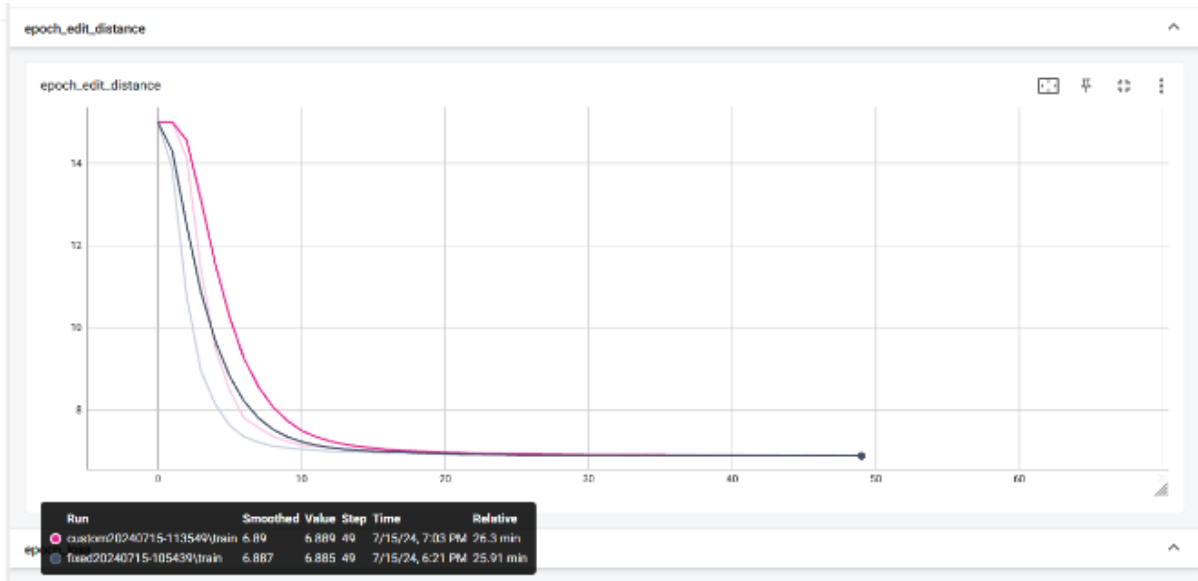
Dalam proses ini, dihasilkan dua tipe *dataset* gambar berdasarkan ukurannya: sama dan acak. Untuk tipe gambar dengan ukuran acak, menggunakan ukuran *font* bervariasi antara 10pt hingga 16pt. Hal ini menghasilkan gambar dengan dimensi yang bervariasi pula. Ukuran panjang dan lebar terbesar dari gambar dalam dataset ini adalah 262x42 piksel, sementara ukuran terkecilnya adalah 18x30 piksel. Rata-rata ukuran gambar dalam dataset acak adalah sekitar 91x36 piksel. Variasi ukuran ini bertujuan untuk memberikan keragaman pada data sehingga model dapat lebih adaptif terhadap berbagai bentuk dan ukuran teks aksara Jawa.

Sedangkan untuk tipe gambar dengan ukuran yang sama, setiap gambar dibuat dengan ukuran keseluruhan tetap sebesar 256x32 piksel. Ukuran ini dipilih untuk menjaga konsistensi dan keragaman data selama proses pelatihan model OCR.

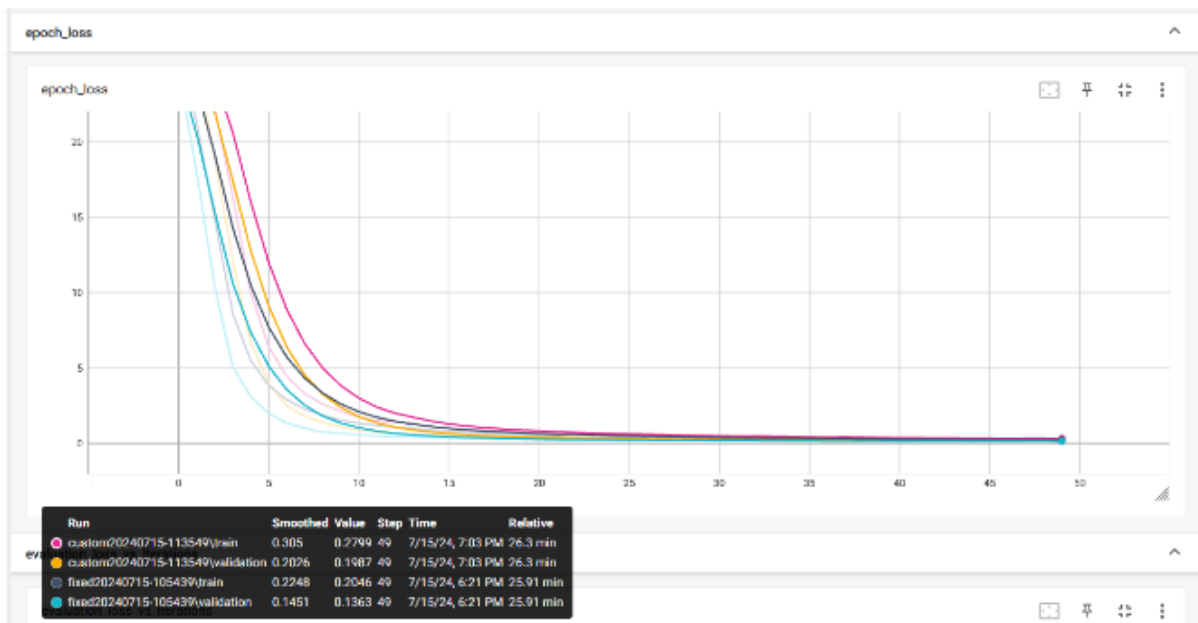
Hasil dari pelatihan dua model yang dibuat pada Gambar 5 menunjukkan perbedaan yang cukup kecil dalam nilai *mean edit distance* setelah *epoch* kesepuluh jarak garis semakin mengecil dan terlihat menyatu di-*epoch* terakhir. Model menggunakan gambar ukuran berbeda dengan nama *file* ‘modelaksaracustom.h5’ mencapai nilai *mean edit distance* sebesar 6.89, dengan nilai *train loss* sebesar 0.2799 dan *validation loss* sebesar 0.1987. Sementara itu, model menggunakan gambar ukuran seragam dengan nama *file* ‘modelaksarafixed.h5’ mencatat *mean edit distance* sedikit lebih rendah yaitu 6.887, dengan *train loss* sebesar 0.2046 dan *validation loss* sebesar 0.1363.

Dapat dilihat pada Gambar 6 nilai *loss* kedua model semakin mengecil. Hal ini menunjukkan bahwa model yang dilatih menunjukkan kemampuan yang baik dalam mengenali dan menerjemahkan teks aksara Jawa, meskipun model dengan nama *file* ‘modelaksarafixed.h5’ menunjukkan performa yang sedikit lebih baik dalam hal *train* dan *validation loss*. Hasil ini menunjukkan bahwa penggunaan *dataset* dengan variasi ukuran (seperti pada dataset acak) tidak secara signifikan mempengaruhi kinerja model, yang tetap konsisten dalam menghadapi variasi tersebut.

Dua model OCR aksara Jawa telah dievaluasi dengan hasil yang berbeda. Model pertama, dilatih dengan gambar berukuran acak, berhasil mengenali 973 gambar dan gagal pada 27 gambar.



Gambar 5. Grafik nilai *mean edit distance*



Gambar 6. Grafik nilai *loss*

Namun, saat diuji dengan gambar berukuran seragam 256x32 piksel, model ini berhasil mengenali 294 gambar dan gagal pada 706 gambar. Di sisi lain, model kedua, yang dilatih dengan gambar berukuran seragam 256x32 piksel, gagal sepenuhnya (0 keberhasilan, 1000 kegagalan) saat diuji dengan gambar berukuran acak. Namun, saat diuji dengan gambar berukuran seragam 256x32 piksel, model ini menunjukkan bahwa 972 gambar berhasil dikenali dan 28 gambar yang gagal.

Pengujian kedua dilakukan menggunakan variasi jumlah *dataset* aksara Jawa, 20%, 40%, 60%, 80%, dan 100%. Selain itu, setiap pengujian juga dilakukan dengan jumlah epoch berbeda, yaitu 10, 20, 30, 40, dan 50 *epoch*. Pengujian diterapkan pada dua jenis dataset: pertama, dataset dengan gambar ukuran acak, hasilnya dicatat dalam tabel 4.1 dan kedua, dataset dengan gambar ukuran seragam, yang hasilnya dicatat dalam tabel 4.2. Penilaian dilakukan untuk membandingkan performa model pada setiap variasi ukuran dataset dan *epoch* yang digunakan.

Hasil pengujian model acak yang ditampilkan pada tabel 1 menunjukkan tren peningkatan kinerja yang konsisten ketika jumlah data yang digunakan bertambah dan jumlah *epoch* ditingkatkan.

Tabel 1. Pengujian model acak

Epoch		Jumlah Dataset				
		20%	40%	60%	80%	100%
10	Mean edit distance	14.99	14.86	13.62	13.33	8.08
	Train loss	1575.05	1553.56	1185.33	1246.38	304.54
	Validation loss	1545.46	1534.97	1065.78	1151.06	224.77
20	Mean edit distance	14.83	8.84	7.84	7.89	7.12
	Train loss	1529.84	395.60	195.75	245.81	70.30
	Validation loss	1504.63	322.46	156.42	184.45	53.36
30	Mean edit distance	12.68	7.45	7.21	7.26	7.00
	Train loss	986.75	127.18	85.65	95.53	45.38
	Validation loss	942.19	116.23	64.61	69.74	33.30
40	Mean edit distance	9.42	7.20	7.09	7.15	6.97
	Train loss	460.06	75.51	55.46	58.00	30.82
	Validation loss	423.62	63.27	44.53	43.76	22.92
50	Mean edit distance	8.27	7.10	7.10	7.08	6.96
	Train loss	230.05	52.60	39.93	42.56	25.90
	Validation loss	232.15	52.93	36.77	34.64	22.66

Tabel 2. Pengujian model seragam

Epoch		Jumlah Dataset				
		20%	40%	60%	80%	100%
10	Mean edit distance	14.36	11.49	9.08	8.17	7.70
	Train loss	1276.89	878.73	430.84	267.06	183.48
	Validation loss	1210.82	783.68	333.21	191.48	129.17
20	Mean edit distance	11.26	8.44	7.35	7.15	7.05
	Train loss	718.69	296.99	110.19	74.61	62.24
	Validation loss	664.46	259.15	83.22	47.90	37.21
30	Mean edit distance	8.89	7.50	7.17	7.02	6.97
	Train loss	314.91	145.81	64.98	41.50	38.00
	Validation loss	291.44	120.83	52.57	29.74	26.71
40	Mean edit distance	8.10	7.22	7.04	7.01	6.96
	Train loss	182.19	96.85	42.83	27.87	28.60
	Validation loss	166.19	71.42	35.77	20.94	22.36
50	Mean edit distance	7.72	7.19	7.03	7.00	6.94
	Train loss	114.01	64.29	32.06	19.82	22.48
	Validation loss	109.03	57.42	28.94	17.52	20.43

Pada setiap tahap pengujian, *mean edit distance* terus menurun, menandakan bahwa model semakin akurat dalam mengenali data. Selain itu, *train loss* dan *validation loss* juga mengalami penurunan yang signifikan, terutama ketika menggunakan 100% *dataset*. Misalnya, pada *epoch* ke-50, *mean edit distance* turun menjadi 6,96, dengan *train loss* dan *validation loss* masing-

masing berkurang menjadi 25,90 dan 22,66. Penurunan ini menunjukkan bahwa model menjadi lebih efektif dalam mempelajari pola dari data, sehingga kesalahan prediksi berkurang seiring dengan semakin banyaknya data dan peningkatan jumlah *epoch*. Dengan demikian, model acak menunjukkan kinerja yang lebih baik ketika dilatih lebih lama dan dengan lebih banyak data.

Pengujian model seragam yang ditampilkan pada Tabel 2 menunjukkan perbaikan kinerja yang konsisten dengan bertambahnya jumlah data dan *epoch*. Pada setiap tahap pengujian, *mean edit distance* menurun, menandakan peningkatan akurasi model dalam mengenali pola data. Pada *epoch* ke-10, *mean edit distance* turun dari 14,36 pada dataset 20% menjadi 7,70 pada dataset 100%. Penurunan ini diikuti oleh penurunan yang signifikan pada *train loss* dan *validation loss*, yang masing-masing mencapai 183,48 dan 129,17 pada dataset 100%.

Seiring bertambahnya *epoch*, *mean edit distance* terus menurun, meskipun dengan kecepatan yang lebih lambat. Pada *epoch* ke-20, nilai *mean edit distance* mencapai 7,05 pada dataset 100%, sementara *train loss* dan *validation loss* turun menjadi 62,24 dan 37,21. Tren ini berlanjut hingga *epoch* ke-50, di mana *mean edit distance* mencapai 6,94, dan *train loss* serta *validation loss* masing-masing turun menjadi 22,48 dan 20,43 pada dataset 100%.

Secara keseluruhan, model seragam menunjukkan kinerja yang semakin baik dengan lebih banyak data dan pelatihan yang lebih lama. Penurunan yang konsisten dalam *mean edit distance*, *train loss*, dan *validation loss* mengindikasikan bahwa model menjadi lebih efisien dan akurat dalam mempelajari dan mengenali data dengan bertambahnya *epoch* dan ukuran *dataset*.

#### 4. KESIMPULAN

Penelitian ini berhasil mengumpulkan 24.644 kosakata, yang setelah dilakukan *cleaning* menjadi 24.038 kosakata. Ditambahkan 1.000 angka acak sehingga totalnya menjadi 25.038 kosakata. Kemudian dikonversi menjadi gambar berukuran 256x32 piksel untuk model seragam dan model acak dengan ukuran font antara 10 hingga 16 menggunakan *font* Noto Sans Javanese.

#### DAFTAR PUSTAKA

Chao, L., Chen, J., & Chu, W. (2020). Variational Connectionist Temporal Classification. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12373 LNCS, 460–476. [https://doi.org/10.1007/978-3-030-58604-1\\_28](https://doi.org/10.1007/978-3-030-58604-1_28)

Hasil evaluasi setelah 50 *epoch* menunjukkan bahwa model seragam mencapai nilai *mean edit distance* 6.887, *train loss* 0.2046, dan *validation loss* 0.1363. Model acak memperoleh *mean edit distance* 6.89, *train loss* 0.2799, dan *validation loss* 0.1987. Model seragam menunjukkan peningkatan sekitar 0.043% dalam nilai *mean edit distance*, 27.0% dalam *train loss*, dan 31.4% dalam *validation loss* dibandingkan dengan model acak.

Dua model OCR untuk aksara Jawa telah dievaluasi. Model acak mengenali 973 gambar dan gagal pada 27 gambar dengan gambar acak. Sedangkan pada gambar seragam hanya mengenali 294 gambar dan gagal pada 706 gambar. Model seragam gagal sepenuhnya pada gambar acak, namun pada gambar seragam dapat mengenali 972 gambar dan hanya 28 gambar yang gagal dikenali. Model yang dilatih dengan gambar berukuran seragam lebih efektif untuk gambar dengan ukuran yang sama, sementara model yang dilatih dengan gambar berukuran acak lebih fleksibel dalam mengenali gambar dengan berbagai ukuran.

Pengujian dua model dengan ukuran *dataset* yang berbeda menunjukkan bahwa model seragam secara konsisten memiliki kinerja yang lebih baik dibandingkan model acak, terutama pada ukuran *dataset* yang lebih besar. Model seragam berhasil mencapai *mean edit distance*, *train loss*, dan *validation loss* yang lebih rendah pada setiap tahap pengujian, menunjukkan kemampuan yang lebih baik dalam mempelajari pola data dan menghasilkan prediksi yang lebih akurat. Selain itu, peningkatan ukuran *dataset* berpengaruh positif terhadap kedua model, dengan penurunan yang lebih signifikan pada nilai *train loss*, nilai *validation loss* dan *mean edit distance*, mengindikasikan bahwa lebih banyak data membantu model dalam mencapai performa yang lebih optimal.

Fakhruddin, D., Sachari, A., & Haswanto, N. (2019). Pengembangan Desain Informasi dan Pembelajaran Aksara Jawa Melalui Media Website. *ANDHARUPA: Jurnal Desain Komunikasi Visual & Multimedia*, 5(01), 1–23. <https://doi.org/10.33633/andharupa.v5i01.1990>

Feng, X., Yao, H., & Zhang, S. (2019). Focal CTC Loss for Chinese Optical Character

- Recognition on Unbalanced Datasets. *Complexity*, 2019. <https://doi.org/10.1155/2019/9345861>
- Gaido, M., Cettolo, M., Negri, M., & Turchi, M. (2021). CTC-Based Compression for Direct Speech Translation. *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, 690–696. <https://doi.org/10.18653/v1/2021.eacl-main.57>
- Ganesh, A. S. (2019). OCR based Image Processing with Audio Output for Visually Challenged People. *International Journal for Research in Applied Science and Engineering Technology*, 7(3), 599–604. <https://doi.org/10.22214/ijraset.2019.3104>
- Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. *ACM International Conference Proceeding Series*, 148, 369–376. <https://doi.org/10.1145/1143844.1143891>
- Hanindria, I. S., & Hendry. (2022). Pengklasifikasian Aksara Jawa Metode Convolutional Neural Network. *Jurnal Teknik Informatika Dan Sitem Informasi*, 9(3), 2727–2737. <https://jurnal.mdp.ac.id/index.php/jatisi/article/view/2177>
- Hou, Y., Kong, Q., & Li, S. (2020). Audio Tagging With Connectionist Temporal Classification Model Using Sequentially Labelled Data. *Lecture Notes in Electrical Engineering*, 516, 955–964. [https://doi.org/10.1007/978-981-13-6504-1\\_114](https://doi.org/10.1007/978-981-13-6504-1_114)
- Khatavkar, V., Velankar, M., & Petkar, S. (2024). Segmentation-Free Connectionist Temporal Classification Loss Based OCR Model for Text Captcha Classification. *ArXiv Preprint ArXiv:2402.05417*.
- Kusnantoro, Rohana, T., & Sulistya Kusumaningrum, D. (2022). Implementasi Metode Tesseract Optical Character Recognition untuk Deteksi Plat Nomor Kendaraan Pada Sistem Parkir. *Scientific Student Journal for Information, Technology and Science*, 3(1), 59–67.
- Lindenberg, N. (2022). *Implementing Javanese*. 1–14. <https://www.unicode.org/notes/tn47/UTN47-Implementing-Javanese-1.pdf>
- Lu, B. R., Shyu, F., Chen, Y. N., Lee, H. Y., & Lee, L. S. (2017). Order-Preserving Abstractive Summarization for Spoken Content Based on Connectionist Temporal Classification. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2017-Augus*, 2899–2903. <https://doi.org/10.21437/Interspeech.2017-862>
- Lu, S., Lu, J., Lin, J., & Wang, Z. (2019). A Hardware-Oriented and Memory-Efficient Method for CTC Decoding. *IEEE Access*, 7, 120681–120694. <https://doi.org/10.1109/ACCESS.2019.2937680>
- Mahakarti, D. C., Sulanjari, B., & Werdiningsih, Y. K. (2023). Kemampuan Menulis Paragraf Aksara Jawa Menggunakan Metode Cycle Learning Kelas XI SMKN 1 Sayung Demak Tahun Ajaran 2022. *JISABDA: Jurnal Ilmiah Sastra Dan Bahasa Daerah, Serta Pengajarannya*, 4(2), 28–36. <https://doi.org/10.26877/jisabda.v4i2.15106>
- Mäs, S., Henzen, D., Bernard, L., Müller, M., Jirka, S., & Senner, I. (2018). Generic Schema Descriptions for Comma-Separated Values Files of Environmental Data. *Agile*, 1–5.
- Mulyanto, A., Susanti, E., Rossi, F., Wajiran, W., & Borman, R. I. (2021). Penerapan Convolutional Neural Network pada Pengenalan Aksara Lampung Berbasis Optical Character Recognition. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 7(1), 52. <https://doi.org/10.26418/jp.v7i1.44133>
- Nisha, K., Wahyuni, T., & Hayat, M. A. M. (2024). Arus Jurnal Sains dan Teknologi Pemeriksaan KTP Menggunakan Optical



- Character Recognition dan Pengenalan Background serta Komponen KTP. *Arus Jurnal Sains Teknologi (AJST)*, 2(2), 490–495.
- Okta, M. D. U., Aulia, S., & Burhanuddin, B. (2021). Pengenalan Pola Berbasis OCR untuk Pengambilan Data Bursa Saham. *Jurnal Rekayasa ElektriKa*, 17(2), 100–106. <https://doi.org/10.17529/jre.v17i2.19656>
- Standard, T. U. (n.d.). *Japanese*.
- Susetianingtias, D. T., Arianty, R., Rodiah, R., & Patriya, E. (2023). *Pembentukan Model Recirrent Neural Network dan Connectionist Temporal Classification Pada Pengenalan Kata Tulisan Tangan Offline*. 3(2), 161–171.
- Zenkel, T., Sanabria, R., Metze, F., Niehues, J., Sperber, M., Stüker, S., & Waibel, A. (2017). Comparison of Decoding Strategies for CTC Acoustic Models. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2017-Augus*, 513–517. <https://doi.org/10.21437/Interspeech.2017-1683>
- Zhan, H., Wang, Q., & Lu, Y. (2017). Handwritten Digit String Recognition by Combination of Residual Network and RNN-CTC. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10639 LNCS, 583–591. [https://doi.org/10.1007/978-3-319-70136-3\\_62](https://doi.org/10.1007/978-3-319-70136-3_62)