

Optimasi Optical Character Recognition Tulisan Tangan Aksara Sunda Menggunakan Model Berbasis EfficientNet

Tugas Akhir

diajukan untuk memenuhi salah satu syarat memperoleh gelar sarjana

dari Program Studi S1 Informatika

Fakultas Informatika

Universitas Telkom

103012380509

Gid Achmad Ahlul Fadli



Program Studi Sarjana Informatika

Fakultas Informatika

Universitas Telkom

Bandung

2025

LEMBAR PENGESAHAN

Optimasi Optical Character Recognition Tulisan Tangan Aksara Sunda Menggunakan Model Berbasis EfficientNet

Optimizing Handwritten Aksara Sunda Optical Character Recognition using EfficientNet-based Model

103012380509

Gid Achmad Ahlul Fadli

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagai syarat
memperoleh gelar pada Program Studi Sarjana Informatika

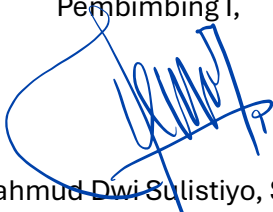
Fakultas Informatika

Universitas Telkom

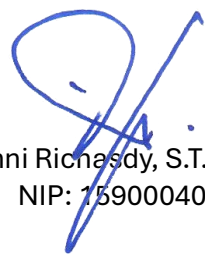
Bandung, 15 Juni 2025

Menyetujui

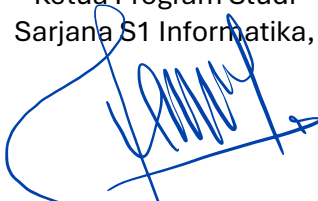
Pembimbing I,


Dr. Mahmud Dwi Sulistiyo, S.T., M.T
NIP: 13880017

Pembimbing II,


Donni Richasdy, S.T., M.T.
NIP: 15900040

Ketua Program Studi
Sarjana S1 Informatika,


Dr. Mahmud Dwi Sulistiyo, S.T., M.T
NIP: 13880017

LEMBAR ORISINALITAS

Dengan ini saya, Gid Achmad Ahlul Fadli, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul “Optimasi Optical Character Recognition Tulisan Tangan Aksara Sunda Menggunakan Model Berbasis EfficientNet” berserta dengan seluruh isinya merupakan hasil karya saya sendiri, dengan tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang berlaku dengan masyarakat keilmuan, serta produk dari tugas akhir ini bukan merupakan hasil dari *Generative AI*. Saya siap menggugung risiko/sanksi yang diberikan jika di kemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam Laporan Tugas Akhir, atau jika ada klaim dari pihak lain terhadap keaslian karya.

Bandung, 15 Juni 2025

Yang menyatakan

Gid Achmad Ahlul Fadli

NIM 103012380509

ABSTRAK

Aksara daerah merupakan sistem ortografi hasil masyarakat daerah yang meliputi aksara dan sistem pengaksaraan untuk menuliskan bahasa daerah, salah satunya adalah Aksara Sunda. Berbagai penelitian sebagai usaha digitasi dalam pelestarian Aksara Sunda dengan berbagai metode dalam *machine learning* dengan fokus *Optical Recognition Character* (OCR) terutama *Convolutional Neural Network* (CNN) sudah dilakukan. Namun, data yang ada pada saat ini terbatas pada karakter yang menjadi bagian swara, ngalagena, rarangkén dan angka. Belum ada data yang berupa gabungan ngalagena dan swara dengan rarangkén menjadi dataset kata. Dengan penambahan dataset kata tersebut dapat menghasilkan dua model, yaitu model *ensemble* dan model *Connectionist Temporal Classification* (CTC). Model *ensemble* dengan *ensemble weight* sebesar 0,5 berhasil mencapai akurasi sebesar 0,912, presisi sebesar 0,889, sensitifitas sebesar 0,701, dan *F1-score* sebesar 0,755. Model CTC berhasil mencapai nilai *Character Error Rate* (CER) sebesar 0,082 dan nilai *Word Error Rate* (WER) sebesar 0,102.

Kata Kunci: *Optical Character Recognition*, Aksara Sunda, *Convolutional Neural Network*, *Recurrent Neural Network*, *Connectionist Temporal Classification*, *EfficientNet*.

ABSTRACT

Aksara is an orthographic system resulting from regional communities which includes characters and pronunciation systems for writing regional languages, one of which is Sundanese script. Various studies as an effort to digitize in the preservation of Sundanese script with various methods in machine learning with a focus on Optical Recognition Character (OCR) especially Convolutional Neural Network (CNN) have been carried out. However, the current data is limited to characters that are part of swara, ngalagena, rarangkén and numbers. There is no data in the form of a combination of ngalagena and swara with rarangkén into a word dataset. With the addition of the word dataset, it can produce two models, namely the ensemble model and the Connectionist Temporal Classification (CTC) model. The ensemble model with an ensemble weight of 0.5 achieved an accuracy of 0.912, precision of 0.889, sensitivity of 0.701, and F1-score of 0.755. While, the CTC model successfully achieved a Character Error Rate (CER) value of 0.082 and a Word Error Rate (WER) value of 0.102.

Keywords: *Optical Character Recognition, Aksara Sunda, Convolutional Neural Network, Recurrent Neural Network, Connectionist Temporal Classification, EfficientNet.*

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Optimasi Optical Character Recognition Tulisan Tangan Aksara Sunda Menggunakan Model Berbasis EfficientNet”. Tugas akhir ini bertujuan untuk mengeksplorasi penerapan EfficientNetB0 dalam mendukung proses digitalisasi Aksara Sunda.

Selama proses penyusunan, penulis mendapat banyak dukungan dari berbagai pihak yang sangat membantu hingga tugas akhir ini dapat diselesaikan tepat waktu dengan hasil yang memuaskan. Untuk itu, penulis menyampaikan terima kasih yang sebesar-besarnya kepada kedua orang tua dan keluarga atas kasih sayang dan dukungan yang tiada henti, kepada dosen pembimbing atas arahan serta masukan yang berharga, serta kepada teman-teman dan semua pihak yang telah berkontribusi dalam penyusunan tugas akhir ini.

Penulis menyadari bahwa penulisan tugas akhir ini masih memiliki kekurangan di berbagai aspek. Oleh karena itu, kritik dan saran sangat diharapkan sebagai bahan evaluasi untuk perbaikan ke depannya. Semoga tugas akhir ini dapat memberikan manfaat bagi para pembaca serta turut mendukung pengembangan ilmu pengetahuan, khususnya di *bidang Optical Character Recognition*.

Bandung, 1 Juli 2025

Gid Achmad Ahlul Fadli

UCAPAN TERIMA KASIH

Dengan rasa syukur yang mendalam, dengan selesainya tugas akhir ini penulis berterima kasih sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa, yang telah memberikan berbagai macam rahmat-Nya kepada penulis sehingga penulis dalam menyelesaikan laporan magang ini dengan tepat waktu.
2. Ibu Ika Dartika dan Bapak Achmad Ruhyana, sebagai orang tua penulis yang selalu memberikan berbagai macam dukungan yang tidak dapat diukur dengan apapun kepada penulis dalam penyelesaian laporan magang ini.
3. Keluarga besar penulis yang memberikan berbagai macam dukungan yang berharga bagi penulis.
4. Bapak Dr. Mahmud Dwi Sulistiyo, S.T., M.T, selaku dosen pembimbing pertama, yang selalu memberikan arahan dan inspirasi selama penulisan tugas akhir.
5. Bapak Donni Richasdy, S.T., M.T., selaku dosen pembimbing kedua, yang selalu memberikan arahan dan masukan serta sabar dalam membimbing penulis selama penulisan tugas akhir.
6. Bapak Erda Guslinar Perdana S.T M.T, selaku dosen wali yang senantiasa membimbing penulis sejak menempuh pendidikan ekstensi Sarjana.
7. Semua teman terdekat penulis yang tidak bisa disebutkan satu-persatu.
8. Berbagai layanan dari Telkom University termasuk yang membantu penulis dalam menempuh pendidikan ekstensi Sarjana.

DAFTAR ISI

LEMBAR PENGESAHAN.....	ii
LEMBAR ORISINALITAS.....	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
UCAPAN TERIMA KASIH	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan dan Manfaat.....	3
1.4. Batasan Masalah	3
1.5. Metode Penelitian	3
1.6. Jadwal Pelaksanaan	4
BAB 2 TINJAUAN PUSTAKA	5
2.1. Optical Character Recognition	5
2.2. Aksara Sunda	5
2.3. Convolutional Neural Network	7
2.3.1. EfficientNet	7
2.4. Recurrent Neural Network	8
2.4.1. Long Short-Term Memory	8
2.4.2. Bidirectional Long-Short Term Memory	9
2.5. Connectionist Temporal Classification Loss	9
2.6. Stratified K-Fold Cross Validation.....	9
2.7. Confusion Matrix	10
2.8. Character Error Rate dan Word Error Rate	10
BAB 3 PERANCANGAN SISTEM	11
3.1. Desain Perancangan Sistem.....	11
3.1.1. Model Ensemble	11
3.1.2. Model Connectionist Temporal Classification (CTC)	13

3.2.	Preprocessing Data	14
3.3.	Evaluasi Sistem	17
3.3.1.	Evaluasi Model Ensemble	17
3.3.2.	Evaluasi Model CTC.....	18
BAB 4	HASIL PERCOBAAN DAN ANALISIS	19
4.1.	Skenario Model Ensemble.....	19
4.1.1.	Model Karakter dan Model Kata.....	19
4.1.2.	Model <i>Ensemble</i>	27
4.2.	Skenario Model CTC	28
4.3.	Analisis Akhir	31
BAB 5	KESIMPULAN DAN SARAN	32
5.1.	Kesimpulan.....	32
5.2.	Saran	32
	DAFTAR PUSTAKA.....	33
	LAMPIRAN.....	36

DAFTAR GAMBAR

Gambar 2.1 Lorem ipsum ditulis dalam Aksara Sunda	6
Gambar 2.2. Contoh perubahan ngalagena dengan rarangkén.....	7
Gambar 2.3. Contoh struktur CNN[14].....	7
Gambar 2.4 Perbandingan Arsitektur EfficientNetB0[16] dengan EfficientNetB1[17].....	8
Gambar 2.5. <i>Stratified K-fold cross validation</i> [29]	9
Gambar 3.1. Diagram alir model <i>ensemble</i>	11
Gambar 3.2. Implementasi EfficientNetB0 sebagai model dasar dari model <i>ensemble</i>	12
Gambar 3.3 Diagram Alir dari pelatihan model yang digunakan untuk <i>Ensemble Model</i>	12
Gambar 3.4 Implementasi model CTC	13
Gambar 3.5 Diagram alir pelatihan model CTC	14
Gambar 3.6 Proses pengumpulan dataset kata dan contoh dataset.....	15
Gambar 3.7 <i>Label encoding</i> dan <i>sequence labelling</i>	16
Gambar 4.1 Grafik akurasi dan <i>loss</i> dari pelatihan model karakter	22
Gambar 4.2 Grafik akurasi dan <i>loss</i> dari pelatihan model kata	24
Gambar 4.3 Perbandingan F1 score dari model akurasi tertinggi dan <i>loss</i> terendah untuk model karakter	25
Gambar 4.4 Perbandingan F1 score dari model akurasi tertinggi dan <i>loss</i> terendah untuk model kata	26
Gambar 4.5 Grafik CTC <i>Loss</i> pada pelatihan 5 fold	29
Gambar 4.6 Perbandingan hasil evaluasi CER dan WER setiap fold	30

DAFTAR TABEL

Tabel 1.1. Jadwal Pelaksanaan Tugas Akhir.....	4
Tabel 2.1. Aksara Sunda ngalagena.....	5
Tabel 2.2 Aksara Sunda Swara.....	6
Tabel 4.1 Hasil pengaturan <i>hyperparameter</i> untuk model karakter dan model kata.....	19
Tabel 4.2 Konfigurasi EfficientNetB0 sebagai base model dari model <i>ensemble</i>	20
Tabel 4.3 Perbandingan <i>epochs</i> , akurasi, dan <i>loss</i> pada pelatihan setiap fold untuk kedua model	25
Tabel 4.4 Perbandingan hasil evaluasi nilai α sebagai nilai <i>ensemble weight</i>	27
Tabel 4.5 Perbandingan nilai α terhadap gambar téh.....	27
Tabel 4.6 Konfigurasi model CTC yang dilatih.....	28
Tabel 6.1 Kemungkinan kombinasi ngalagena dengan rarangkén untuk satu karakter ngalagena.....	37

BAB 1 PENDAHULUAN

1.1. Latar Belakang

Indonesia merupakan negara yang kaya akan budaya dan tradisi yang unik. Menurut Statistik Kebudayaan tahun 2023 yang dikeluarkan oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, jumlah warisan budaya tak benda di Indonesia adalah sebanyak 1.941[1]. Dari sekian banyak jumlah warisan budaya tak benda tersebut, aksara daerah merupakan salah satu budaya sastra yang beraneka ragam karena berasal dari berbagai suku yang ada di Indonesia. Aksara daerah merupakan sistem ortografi hasil masyarakat daerah yang meliputi aksara dan sistem pengaksaraan untuk menuliskan bahasa daerah[2]. Aksara Sunda termasuk ke dalam kategori Aksara dan Naskah Ka Ga Nga yang diakui sebagai warisan budaya tak benda pada tahun 2013[1].

Upaya pelestarian aksara daerah sudah marak dilakukan oleh berbagai lapisan masyarakat. Salah satu upaya melestarikan Aksara Sunda, Pemerintah daerah provinsi Jawa Barat (Jabar) mengeluarkan Peraturan Daerah (Perda) Provinsi Jabar nomor 5 tahun 2003, lalu dirubah oleh Perda Provinsi Jabar nomor 14 tahun 2014 tentang Pemeliharaan Bahasa, Sastra, dan Aksara Daerah[2]. Upaya lainnya adalah dari kalangan civitas akademik, khususnya bidang informasi dan teknologi, yang sudah banyak melakukan penelitian untuk melestarikan aksara daerah menggunakan media digital.

Banyak penelitian dengan fokus *Optical Character Recognition* (OCR) melalui metode *Convolutional Neural Network* (CNN) sudah dilakukan sebagai bentuk proses digitasi Aksara Sunda. Penelitian sebelumnya oleh Hayati K, et al[3] dilakukan eksplorasi metode CNN klasik untuk OCR Aksara Sunda. Fitur pada Aksara Sunda diekstraksi menggunakan *Histogram of Oriented Gradients* (HOG). Penelitian ini membandingkan metode *Support Vector*

Machine (SVM), K-Nearest Neighbors (k-NN), Decision Tree, Naïve Bayes, Random Forest, dan Artificial Neural Network (ANN) dalam proses OCR. Penelitian lainnya oleh Prameswari M, *et al*[4] dengan fokus *transfer learning* berbasis CNN membandingkan antara arsitektur ResNet-50, arsitektur VGG-19, dan arsitektur MobileNet. Penelitian ini dilakukan menggunakan 2.560 *training data* dan 640 *validation data*. Penelitian lainnya dilakukan oleh Akram R, *et al*[5] yang bertujuan untuk mencari *activation function* paling efektif dalam CNN untuk OCR Aksara Sunda. Penelitian ini membandingkan beberapa *activation function* dari *Rectified Linear Unit (ReLU)*, yaitu *Leaky ReLU*, *Randomized Leaky ReLU (RLReLU)*, dan *Optimized Leaky ReLU (OLReLU)*. Penelitian ini menunjukkan bahwa OLReLU merupakan *activation function* terbaik dengan nilai akurasi 0.987.

Dilihat dari penelitian-penelitian diatas, penelitian yang sudah dilakukan masih berfokus pada klasifikasi karakter. Selain itu, belum adanya penelitian tentang Sehingga, diperlukan penelitian lebih lanjut yang berfokus pada pendeteksian kata sebagai tahap pengembangan lanjutan.

1.2. Rumusan Masalah

Masalah yang akan dibahas dalam penelitian ini adalah bagaimana cara untuk menghasilkan model klasifikasi untuk mengenali kata pada Aksara Sunda. Kriteria yang menjadi fokus dalam permasalahan ini adalah menghasilkan model untuk klasifikasi kata Aksara Sunda tanpa menggunakan model untuk segmentasi. Rumusan masalah diatas dapat diurai lagi menjadi sebagai berikut:

1. Bagaimana penerapan model berbasis EfficientNet untuk klasifikasi dalam Optical Character Recognition tanpa segmentasi untuk tulisan tangan Aksara Sunda?
2. Bagaimana perbandingan performa yang dihasilkan dari penerapan model berbasis EfficientNet?

1.3. Tujuan dan Manfaat

Tujuan dari penelitian ini yaitu sebagai berikut:

1. Mengimplementasikan EfficientNetB0 sebagai model dasar untuk Optical Character Recognition untuk mengenali pola kata Aksara Sunda.
2. Menganalisis performa EfficientNetB0 sebagai model dasar untuk Optical Character Recognition untuk mengenali pola kata Aksara Sunda.

1.4. Batasan Masalah

Batasan masalah pada penelitian Tugas Akhir ini yaitu:

1. Penelitian ini menggunakan dua jenis dataset.
 - a. Dataset karakter yang terdiri dari ngalagena dan swara. Terdiri dari 32 kelas.
 - b. Dataset kata yang terdiri dari gabungan antara ngalagena dan swara dengan rarangkén. Terdiri dari 220 kelas.
2. Maksimal pelatihan model dilakukan selama 30 *epochs*.
3. Tidak ada model segmentasi yang digunakan dalam penelitian ini

1.5. Metode Penelitian

Metode dalam penelitian Tugas Akhir ini terdiri dari beberapa tahap yaitu sebagai berikut:

1. Kajian studi

Mencari materi yang bersangkutan, seperti Aksara Sunda dan tata cara penulisannya, *deep learning*, *Convolutional Neural Network (CNN)*, *Recurrent Neural Network (RNN)*, arsitektur EfficientNet dan arsitektur CNN ataupun RNN lainnya.

2. Pengumpulan data

Menggunakan data dengan sumber Direktori Aksara Sunda untuk Unicode, ditambah dengan data kata yang ditulis tangan dalam Aksara Sunda yang dikumpulkan.

3. Perancangan sistem

Sistem yang dirancang pada penelitian ini berupa sistem pendeteksian pola kata Aksara Sunda dengan model dasar berbasis EfficientNet.

4. Implementasi sistem

Implementasi rancangan sistem pada penelitian ini akan menggunakan bahasa pemrograman *python* dibantu dengan *library-library python* yang ada.

5. Evaluasi model

Evaluasi ini akan dilakukan dengan parameter acuan berupa akurasi, presisi, *recall* dan *F1-score*, disertai dengan evaluasi *Word Error Rate* (WER) dan *Character Error Rate* (CER).

6. Pengambilan kesimpulan

Pengambilan kesimpulan penelitian ini dilakukan berdasarkan hasil dari pengujian sistem dan bertujuan untuk menjawab permasalahan yang diusulkan.

1.6. Jadwal Pelaksanaan

Tabel 1.1. Jadwal Pelaksanaan Tugas Akhir.

No.	Deskripsi Tahapan	Bulan 1	Bulan 2	Bulan 3	Bulan 4	Bulan 5	Bulan 6
1	Studi Literatur						
2	Pengumpulan Data						
3	Perancangan Sistem						
4	Implementasi Sistem						
5	Evaluasi Model						
6	Penyusunan Laporan/Buku TA						

BAB 2 TINJAUAN PUSTAKA

2.1. Optical Character Recognition













Menurut Batra P, et al[6], *Optical Character Recognition* (OCR) adalah ranah penelitian yang aktif mencari cara untuk menghasilkan sebuah sistem otomatis yang dapat mengekstrak dan memproses teks dari gambar. Tujuan dari OCR adalah untuk menghasilkan teks yang dapat disunting dalam bentuk digital sebagai hasil dari gambar yang telah dipindai.














Beberapa metode yang dilakukan pada OCR adalah *Convolutional Neural Networks* (CNN), *Recurrent Neural Networks* (RNN), *Transformers* dan masih banyak yang lainnya. Beberapa teknologi OCR yang populer dan lumrah digunakan karena sifat *open-source* contohnya adalah *Tesseract OCR*, *EasyOCR*, dan *docTR*.

2.2. Aksara Sunda

Dataset Aksara Sunda yang akan digunakan pada penelitian ini merupakan hasil standarisasi dari buku Direktori Aksara Sunda untuk Unicode[7], sebagai upaya standarisasi Aksara Sunda dalam pelestarian salah satu budaya suku Sunda. Aksara Sunda terdiri dari swara, ngalagena, angka, dan rarangkén. Aksara Sunda ngalagena berjumlah 25 karakter dengan pemabagian 23 karakter dasar dengan tambahan karakter serapan sebanyak 2 karakter yaitu *sya* dan *kha* ditunjukkan pada tabel 2.1.








Tabel 2.1. Aksara Sunda ngalagena.

 ka	 ga	 nga	 ca	 ja	 nya
 ta	 da	 na	 pa	 ba	 ma

 ya	 ra	 la	 wa	 sa	 Ha
 fa	 va	 qa	 xa	 za	
 kha	 sya				

Sementara itu, aksara swara pada Aksara Sunda berjumlah 7 karakter seperti ditunjukkan pada tabel 2.2.

Tabel 2.2 Aksara Sunda Swara.

						
a	é	i	o	u	e	eu

Pada penulisan kata hingga kalimat Aksara Sunda yang memiliki peran yang sangat penting adalah gabungan ngalagena dan rarangkén seperti dicontohkan pada gambar 2.1.



Gambar 2.1 Lorem ipsum ditulis dalam Aksara Sunda

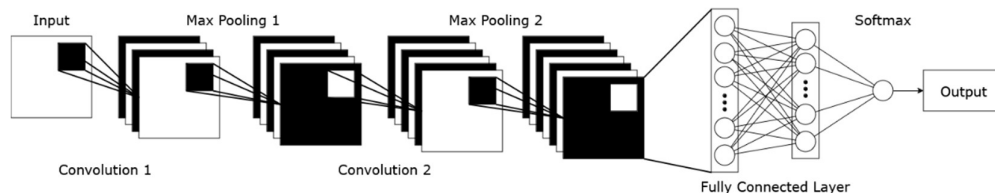
Namun, dari sekian banyak sumber dataset yang ada secara *online*, dicontohkan dari situs Roboflow[8], [9], [10] dan situs Mendeley[11], tidak ada data mengenai gabungan dari setiap karakter ngalagena dengan masing-masing karakter rarangkén seperti dicontohkan pada gambar 3.2. Maka dibutuhkan pembuatan dataset baru yang mengandung gabungan dari setiap ngalagena dengan rarangkén yang ada. Kemungkinan kombinasi dari satu ngalagena dengan semua rarangkén dilampirkan di tabel 6.1.



Gambar 2.2. Contoh perubahan ngalagena dengan rarangkén

2.3. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu tipe dari *Artificial Neural Networks* (ANN)[12]. Viktor Pomazan *et al*[13] mengungkapkan bahwa CNN sendiri telah terbukti menjadi salah satu metode yang efektif dalam kegiatan *Optical Character Recognition* (OCR). Struktur dari CNN pada umumnya dapat dilihat pada gambar 2.3. Dalam perkembangannya banyak model berbasis CNN telah dikembangkan, contoh yang populer diantaranya adalah *Visual Geometry Group* (VGG), *GoogleNet*, *Residual Network* (ResNet), *Squeeze and Excitation Network* (SENet) dan *MobileNet*[14].

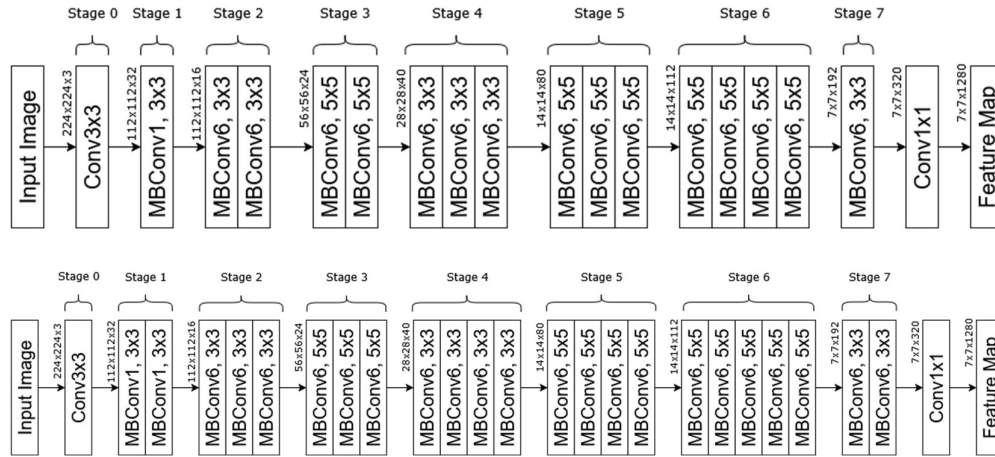


Gambar 2.3. Contoh struktur CNN[14]

2.3.1. EfficientNet

Model *Convolutional Neural Network* (CNN) yang telah banyak berkembang menjadikan ada banyak pilihan menarik dibandingkan model-model populer yang telah disebutkan. Seperti dilansir pada situs web keras.io[15], EfficientNet, khususnya EfficientNetB0, merupakan model dengan ukuran 29MB menjadikanya model terkecil setelah MobileNet dan MobileNetV2 dengan ukuran 16MB dan 14MB. Poin menarik dari EfficientNet adalah mudahnya melakukan *fine-tuning* dalam skalabilitas model yang digunakan. Skalabilitas tersebut dimulai dari B0 sampai dengan B7, dilakukan dengan menambahkan satu blok proses pada setiap satu tahap peningkatan. Hal

tersebut menjadikan EfficientNet menjadi salah satu pilihan model yang adaptif mengikuti jumlah sumber daya yang ada.



Gambar 2.4 Perbandingan Arsitektur EfficientNetB0[16] dengan EfficientNetB1[17]

2.4. Recurrent Neural Network

Recurrent Neural Network (RNN) adalah salah satu tipe ANN yang dirancang secara khusus untuk menangani data secara sekuensial dengan bantuan timestep[18]. Hal tersebut menjadikan RNN sebagai metode yang populer dalam mengolah data yang memiliki pola data secara sekuensial, baik dalam bentuk teks, audio, dan *time series*. Beberapa contoh penerapan RNN adalah pendeteksian perubahan cuaca[19], [20], dan Optical Character Recognition [21], [22]. Beberapa penelitian juga menyarankan gabungan CNN dengan RNN dengan tujuan saling melengkapi[23].

2.4.1. Long Short-Term Memory

Long Short-Term Memory (LSTM) merupakan pengembangan dari RNN untuk mengatasi masalah memori RNN yang masih tergolong kecil dengan menambahkan satu blok memori. Meskipun hal ini menangani masalah *vanishing gradient* pada RNN dengan sekuens data kecil, tapi LSTM masih bisa mengalami masalah *exploding gradient* jika sekuens data terlalu besar[22].

2.4.2. Bidirectional Long-Short Term Memory

Bidirectional Long-Short Term Memory (BiLSTM) merupakan bentuk pengembangan lainnya dari RNN dan LSTM. Berbeda dengan kedua metode tersebut yang hanya bisa memproses data sekuensial dalam satu arah, BiLSTM dirancang agar bisa memproses data sekuensial dalam dua arah[24]. Metode ini digunakan dalam beberapa penelitian mengenai OCR untuk bahasa arab[25].

2.5. Connectionist Temporal Classification Loss

Connectionist Temporal Classification (CTC) Loss adalah teknik evaluasi yang digunakan sebagai pengganti dalam pengolahan data sekuensial dengan panjang input dan target output yang tidak selaras[26]. CTC Loss akan mengevaluasi data input sekuensial dengan menambahkan token “blank” sebagai pengganti data yang berulang dalam sekuensial tersebut[27].

2.6. Stratified K-Fold Cross Validation

Stratified K-Fold Cross Validation adalah variasi pengembangan dari *K-fold cross validation* biasa yang memungkinkan data terbagi dengan persentase yang sama untuk masing-masing kelas[28]. Hal ini memastikan model melihat semua kelas walaupun ada perbedaan jumlah data setiap kelas.



Gambar 2.5. Stratified K-fold cross validation[29]

2.7. Confusion Matrix

Confusion matrix merupakan metrik yang umum digunakan dalam pengevaluasian *machine learning* tipe klasifikasi[30]. Metrik-metrik yang dihasilkan berupa akurasi (accuracy), presisi (precision), sensitifitas (recall), dan nilai F1 (F1-score). Akurasi menunjukkan keberhasilan sistem mendeteksi dan mengklasifikasi data dengan benar. Sensitifitas (*recall*) menunjukkan keberhasilan sistem mendeteksi data dengan akurat. Presisi merupakan nilai yang menunjukkan keberhasilan sistem dalam menentukan prediksi terhadap data positif yang tepat dibandingkan dengan total keseluruhan prediksi. *F1-score* menunjukkan *harmonic mean* dari sensitifitas dan presisi.

2.8. Character Error Rate dan Word Error Rate

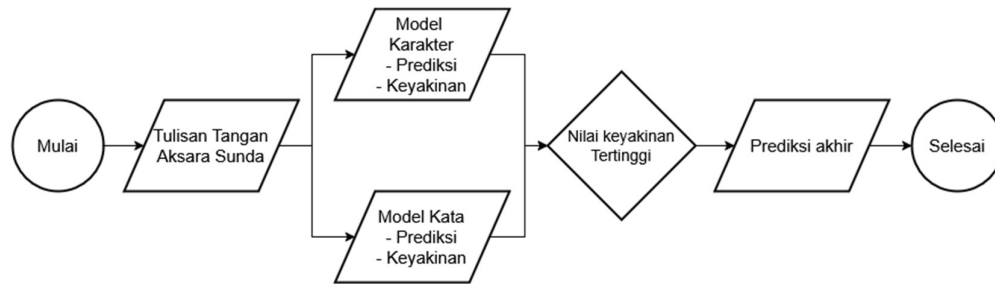
Character Error Rate (CER) dan *Word Error Rate* (WER) adalah metrik yang bagus digunakan untuk mengukur akurasi OCR terutama yang berbasis kata. Kedua metrik ini menggunakan *levenshtein distance*, yang mengukur jumlah minimum perubahan karakter/kata agar dua string menjadi sama baik dalam tingkat karakter atau kata[22]. Hal yang diukur adalah kesalahan penambahan, kesalahan penghapusan, dan kesalahan substitusi baik itu dalam tingkat karakter ataupun kata.

BAB 3 PERANCANGAN SISTEM

3.1. Desain Perancangan Sistem

Penelitian ini bertujuan untuk membangun sistem OCR yang mengenali kata dalam tulisan tangan Aksara Sunda dengan model berbasis EfficientNet. Pada proses prenelitian ini, EfficientNetB0 diimplementasikan dalam dua model yang berbeda, yaitu klasifikasi menggunakan *ensemble learning* dan klasifikasi menggunakan fungsi *Connectionist Temporal Classification Loss*.

3.1.1. Model Ensemble

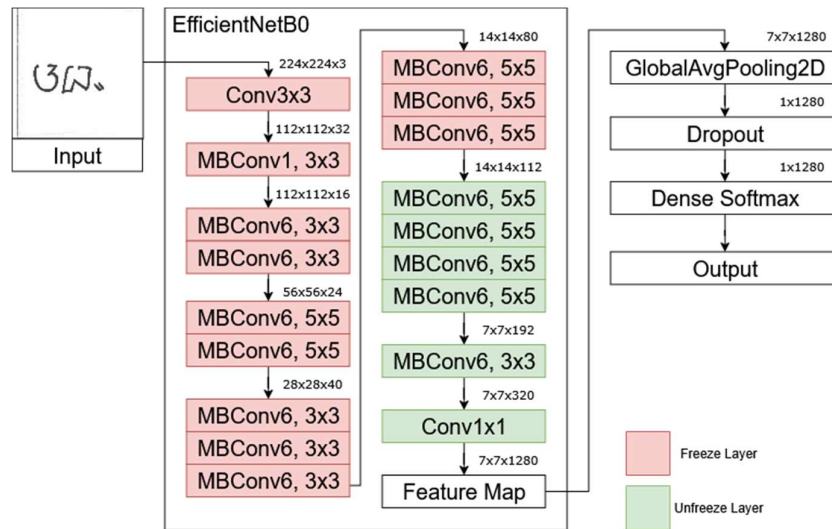


Gambar 3.1. Diagram alir model *ensemble*

Model yang digunakan dalam model *ensemble* berjumlah dua model klasifikasi berbasis EfficientNetB0 seperti terlihat pada gambar 3.1, yaitu model 1 untuk klasifikasi karakter, dan model 2 untuk klasifikasi kata. Dengan begitu, dataset yang digunakan dalam pelatihan model ini terbagi menjadi dua dataset, yaitu dataset karakter dan dataset kata. Prediksi akhir dihasilkan dengan cara membandingkan nilai *confidence (weighted)* dari kedua model. Nilai *confidence (weighted)* didapatkan dengan melakukan perhitungan menggunakan nilai α sebagai *ensemble weight*. Perhitungan *confidence (weighted)* dapat dilihat sebagai berikut.

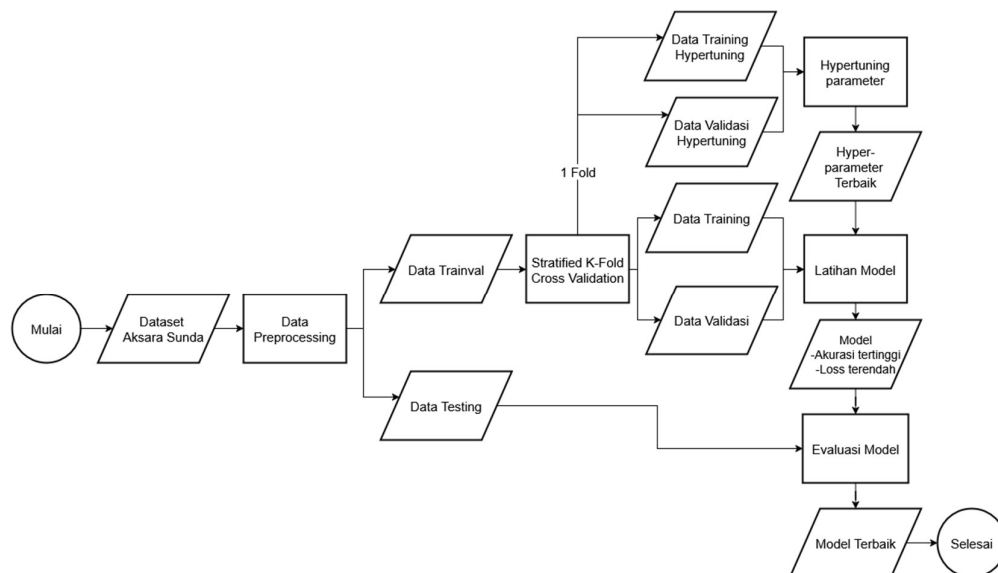
$$confidence (weighted)_{karakter} = \alpha * confidence_{karakter} \quad (3.1)$$

$$confidence (weighted)_{kata} = (1 - \alpha) * confidence_{kata} \quad (3.2)$$



Gambar 3.2. Implementasi EfficientNetB0 sebagai model dasar dari model *ensemble*

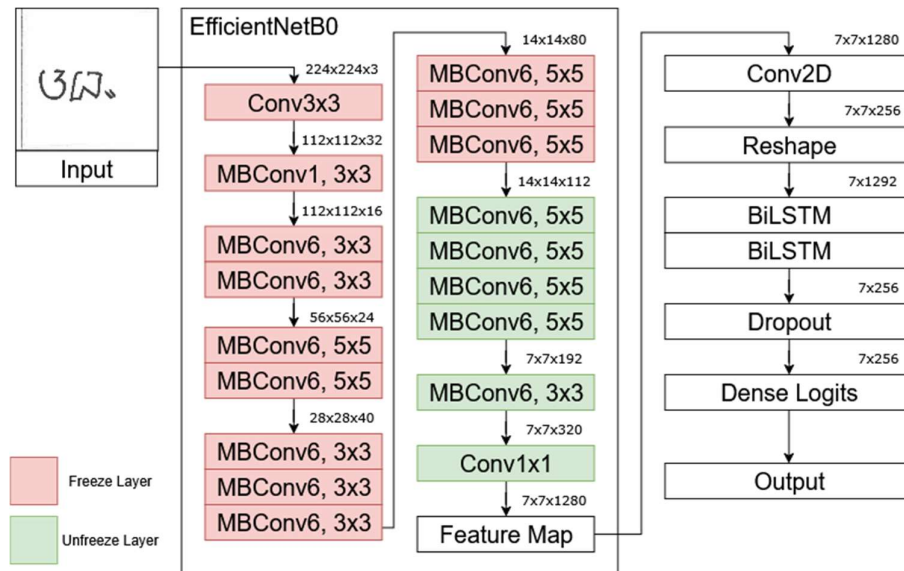
Model EfficientNetB0 yang digunakan sebagai model dasar dalam penelitian ini diimplementasikan seperti pada gambar 3.2. Gambar Aksara Sunda diterima oleh input layer pada EfficientNetB0 yang menghasilkan output berupa *feature map*. EfficientNetB0 dimodifikasi dengan proses *unfreeze layer* dari layer 163 sampai layer terakhir. Proses ini dilakukan agar model bisa melakukan *fine-tuning* terhadap dataset tulisan Aksara Sunda yang berbeda dengan tulisan latin. Layer *dropout* digunakan agar model tidak mengalami *overfitting*.



Gambar 3.3 Diagram Alir dari pelatihan model yang digunakan untuk *Ensemble Model*

Gambar 3.3 merupakan diagram alir yang menunjukkan proses pelatihan model yang digunakan dalam proses pelatihan model *ensemble*. Pembagian dataset dilakukan dengan ratio 80:20 untuk data trainval dan data testing. Proses *Stratified K-Fold Cross Validation* dilakukan dua kali, yaitu untuk proses *Hypertuning Parameter* dan pelatihan model. Pada proses *Hypertuning Parameter* menggunakan 1 fold, sementara untuk pelatihan model menggunakan 5 fold. Proses pelatihan model dilakukan setelah mendapatkan *hyperparameter* terbaik. *Stratified K-Fold Cross Validation* membagi data trainval dengan ratio 80:20 untuk data training dan data validation. Pada pelatihan diterapkan *early stopping* dengan dua *model checkpoint* yaitu untuk hasil akurasi tertinggi dan hasil *loss* terendah. Maka dari itu setiap fold menghasilkan 2 model, sehingga menghasilkan total 10 model yang bisa dievaluasi.

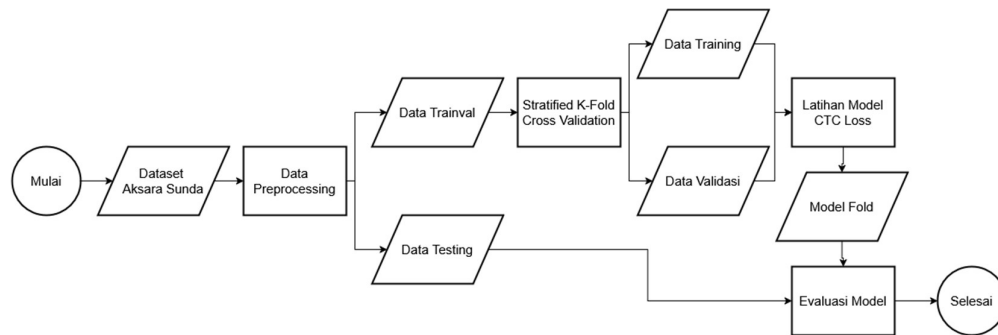
3.1.2. Model Connectionist Temporal Classification (CTC)



Gambar 3.4 Implementasi model CTC

Gambar 3.4 merupakan implementasi model CTC dengan menggunakan *EfficientNetB0* sebagai ekstraktor fitur dari input gambar. Selanjutnya, fitur gambar diproses melalui layer *conv2D* dan *reshape* agar menjadi data sekuensial yang memiliki *timestep*. Data ini selanjutnya diproses oleh dua

layer *Bidirectional Long Short Term-Memory* (BiLSTM) untuk menghasilkan *logits*. *Logits* merupakan representasi prediksi karakter pada setiap langkah waktu. *Logits* inilah yang akan dievaluasi oleh fungsi *CTC Loss* untuk menyelaraskan antara hasil prediksi dan label target.

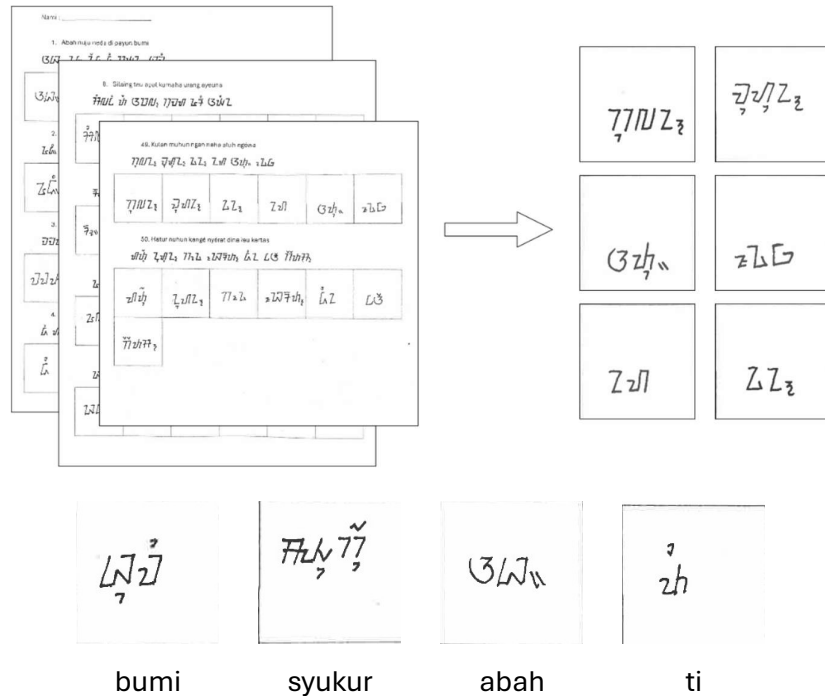


Gambar 3.5 Diagram alir pelatihan model CTC

Gambar 3.5 merupakan diagram alir pelatihan model CTC. Dataset yang dipakai melewati *preprocessing* data tambahan dibandingkan model *ensemble* terlebih dahulu yaitu tahap *label encoding* dan *sequence labelling*. Dataset akan dibagi menjadi data trainval dan data testing dengan ratio 80:20. Data trainval kemudian menggunakan metode *stratified K-fold cross validation* dibagi lagi menjadi 5 fold dengan ratio data training dan data validation sebesar 80:20 untuk masing-masing fold. Pelatihan model ini dilakukan dengan menggunakan *CTC Loss* sebagai fungsi evaluasi per-fold. *CTC Loss* akan mengevaluasi output *dense layer* per timestep dengan menjumlahkan probabilitas semua kemungkinan urutan prediksi terhadap label target.

3.2. Preprocessing Data

Dataset yang digunakan dalam penelitian ini terbagi menjadi dua, yaitu dataset karakter dan dataset kata. Dataset karakter menggunakan dataset yang berasal dari situs Mendeley[11]. Sementara itu, dataset kata menggunakan dataset yang dikumpulkan sendiri. Dataset kata terdiri dari 220 kelas dengan rata-rata terdapat 15 data untuk masing-masing kelas. Jumlah data terkecil dalam satu kelas adalah 12 data dan jumlah terbanyak dalam satu kelas adalah 90 data.

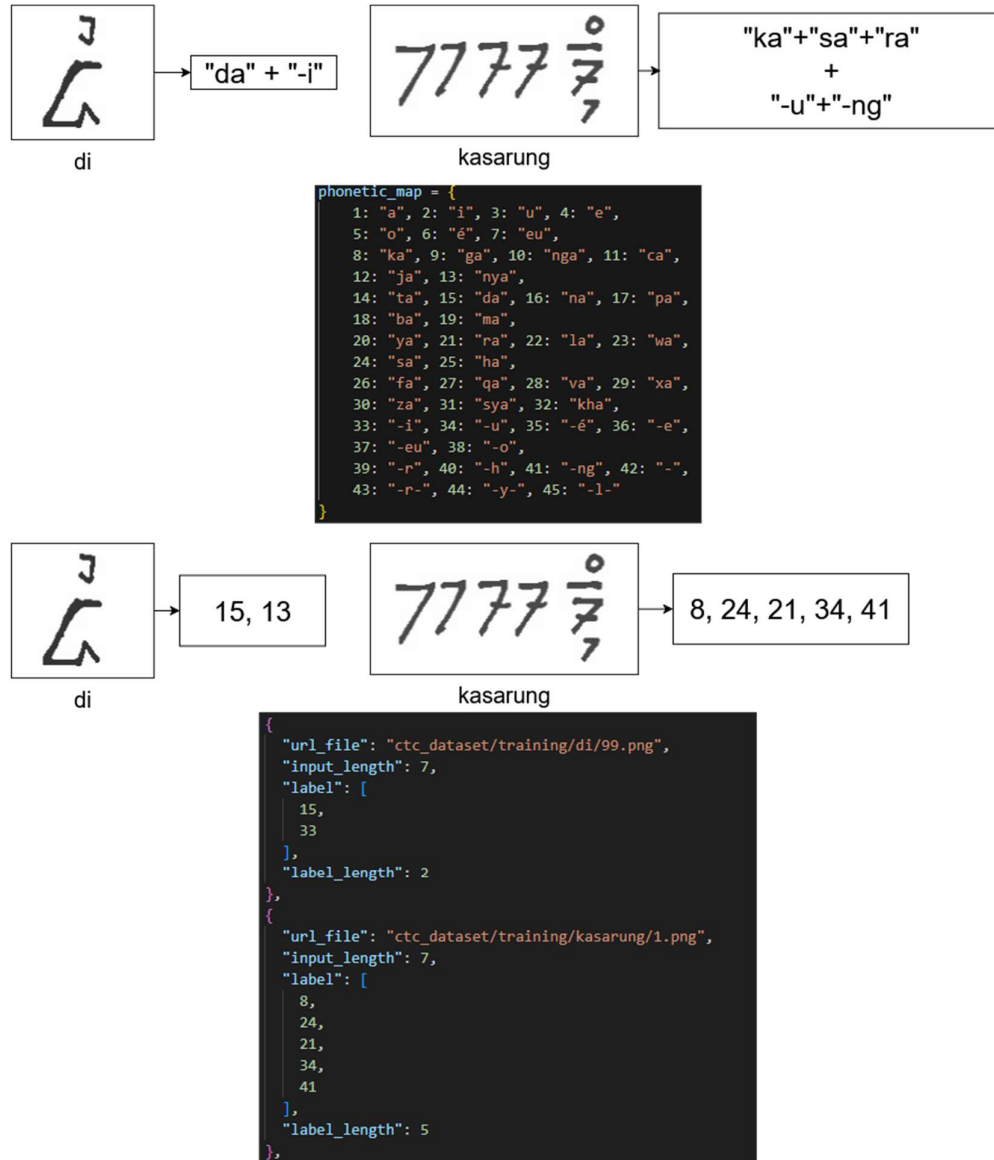


Gambar 3.6 Proses pengumpulan dataset kata dan contoh dataset

Proses pengumpulan dataset kata dimulai dengan penyusunan formulir dengan contoh kalimat dalam Bahasa Sunda, contoh Aksara Sunda, dan kolom yang akan diisi oleh tulisan tangan Aksara Sunda sebanyak 50 kalimat. Formulir yang dibagikan dan diisi tulisan Aksara Sunda oleh responden kemudian dipindai (*scan*) ke dalam format *PDF*. Selanjutnya dilakukan penentuan koordinat dari masing-masing *bounding box* kata yang ada pada formulir. Lalu, pemotongan gambar (*cropping*) dilakukan secara otomatis menggunakan skrip *python* berdasarkan koordinat yang sudah ditentukan. Hasil pemotongan kemudian diklasifikasikan ke dalam kelas masing-masing sesuai kata yang ditulis.

Jumlah responden dalam pengumpulan data ini adalah sebanyak 15 orang. Setelah melewati tahap *screening*, ada beberapa tulisan yang tidak masuk kriteria. Hal ini menjadikan beberapa kelas memiliki jumlah data minimal 12 sampel. Di sisi lain, karena terdapat kata yang berulang dalam 50 kalimat yang disediakan, maka terdapat satu kelas yang mencapai maksimal data sebanyak 132 data dan dua kelas dengan jumlah data 90 data.

Proses *data preprocessing* lainnya yang dilakukan adalah proses *label encoding* dan *sequence labeling*. Proses tersebut dilakukan agar dataset dapat diproses untuk model CTC.



Gambar 3.7 Label encoding dan sequence labelling

Gambar 3.7 menunjukkan *label encoding* dan *sequence labelling* dengan menggunakan acuan tata tulis Aksara Sunda. Dengan menggunakan label encoding tersebut maka data yang ada dipetakan melalui proses *sequence labelling* agar bisa diproses melalui RNN.

3.3. Evaluasi Sistem

Dikarenakan terdapat dua skenario berbeda pada penelitian ini, maka evaluasi sistem pun dilakukan dalam bentuk yang berbeda untuk masing-masing skenario.

3.3.1. Evaluasi Model Ensemble

Evaluasi model *ensemble* akan menggunakan metrik dari *confussion matrix*, yaitu akurasi, sensitifitas, presisi, dan *F1-score*. Berikut adalah rumus akurasi, sensitifitas, presisi, dan *F1-score*.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.3)$$

$$Sensitifitas = \frac{TP}{TP + FN} \quad (3.4)$$

$$Presisi = \frac{TP}{TP + FP} \quad (3.5)$$

$$F1 - score = \frac{2(sensitifitas \times presisi)}{sensitifitas + presisi} \quad (3.6)$$

dimana:

TP = Jumlah data positif yang diklasifikasikan sebagai data positif.

FP = Jumlah data negatif yang diklasifikasikan sebagai data positif.

TN = Jumlah data negatif yang diklasifikasikan sebagai data negatif.

FN = Jumlah data positif yang diklasifikasikan sebagai data negatif.

3.3.2. Evaluasi Model CTC

Evaluasi model CTC akan menggunakan metrik CER dan WER. Dengan rumus sebagai berikut:

$$CER = \frac{CI + CD + CS}{CC + CD + CS} \quad (3.7)$$

dimana:

CI = Jumlah karakter yang ditambahkan.

CD = Jumlah karakter yang dihapus.

CS = Jumlah karakter yang disubstitusi.

CC = Jumlah karakter yang benar.

Berikut adalah rumus untuk menghitung WER.

$$WER = \frac{WI + WD + WS}{WC + WD + WS} \quad (3.8)$$

dimana:

WI = Jumlah kata yang ditambahkan.

WD = Jumlah kata yang dihapus.

WS = Jumlah kata yang disubstitusi.

WC = Jumlah kata yang benar.

BAB 4 HASIL PERCOBAAN DAN ANALISIS

4.1. Skenario Model Ensemble

Model *ensemble* dimulai dengan proses pelatihan model karakter dan model kata. Proses pelatihan masing-masing model meliputi proses *hyperparameter tuning* sehingga pelatihan model.

4.1.1. Model Karakter dan Model Kata

Hyperparameter yang digunakan pada proses *hyperparameter tuning* adalah *dropout* dan *learning rate*. Proses ini dilakukan dengan 5 *epochs* sebanyak 8 kali untuk masing-masing model. Rentang nilai parameter *dropout* yang dipakai adalah 0,2; 0,3; dan 0,4. Sementara untuk parameter *learning rate* adalah 0,001; 0,0001; dan 0,00005.

Tabel 4.1 Hasil pengaturan *hyperparameter* untuk model karakter dan model kata

Percobaan	<i>Dropout</i>		<i>Learning rate</i>		Akurasi	
	Model Karakter	Model Kata	Model Karakter	Model Kata	Model Karakter	Model Kata
1	0,2	0,2	0,001	0,001	0,959	0,496
2	0,3	0,3	0,001	0,001	0,945	0,284
3	0,4	0,4	0,001	0,001	0,966	0,153
4	0,2	0,2	0,0001	0,0001	0,880	0,005
5	0,3	0,3	0,0001	0,0001	0,861	0,004
6	0,2	0,2	0,00005	0,00005	0,774	0,004
7	0,3	0,3	0,00005	0,00005	0,760	0,004
8	0,4	0,4	0,00005	0,00005	0,805	0,004

Tabel 4.1 memperlihatkan hasil dari proses *hyperparameter tuning* pada kedua model. Model karakter mendapatkan hasil terbaik dalam 5 *epochs* dengan nilai *dropout* sebesar 0,4 dan nilai *learning rate* sebesar 0,001 dengan

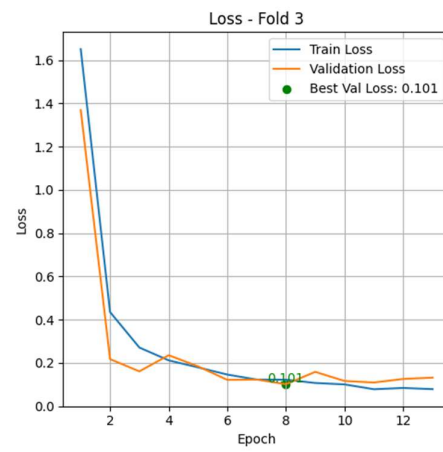
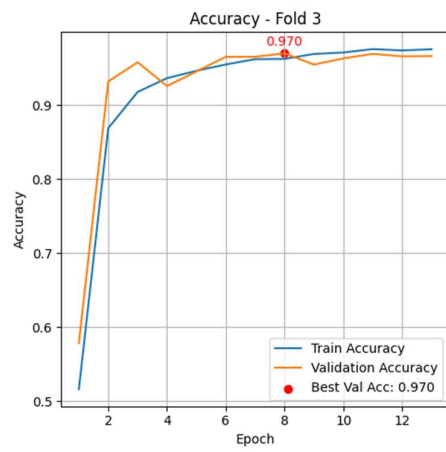
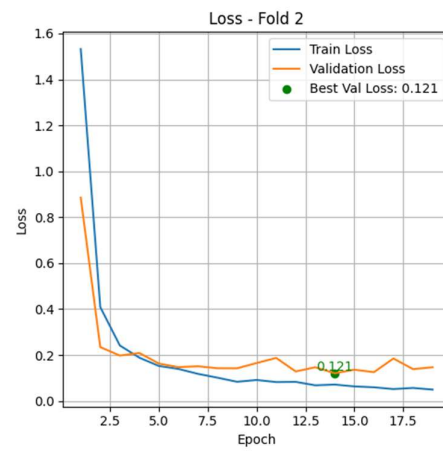
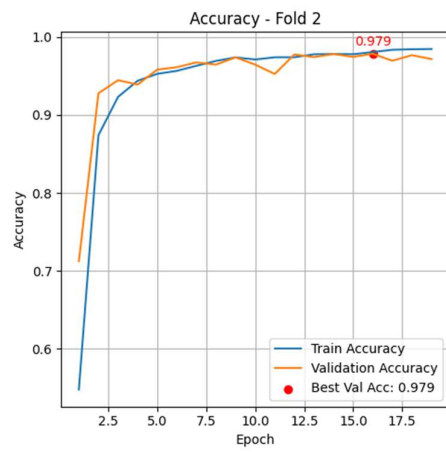
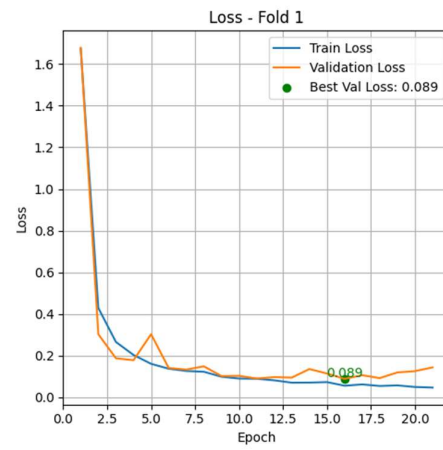
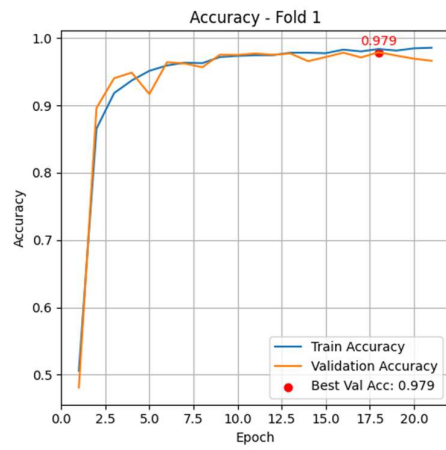
akurasi 0,966. Sementara itu, model kata mendapatkan hasil terbaik dalam 5 *epochs* dengan nilai *dropout* sebesar 0,2 dan nilai *learning rate* sebesar 0,001 dengan nilai akurasi 0,496. Perbedaan parameter terdapat pada nilai *dropout* yang bertugas untuk menjaga model agar tidak terjadi *overfitting*. Model karakter dengan dataset kelas yang lebih kecil dan merata memerlukan nilai *dropout* lebih besar agar menjaga model tidak *overfit* dan tetap general. Sedangkan, model kata dengan dataset kelas yang banyak (220 kelas) dan jumlah data setiap kelas sedikit cenderung sensitif akan kehilangan informasi sehingga memerlukan *dropout* lebih kecil.

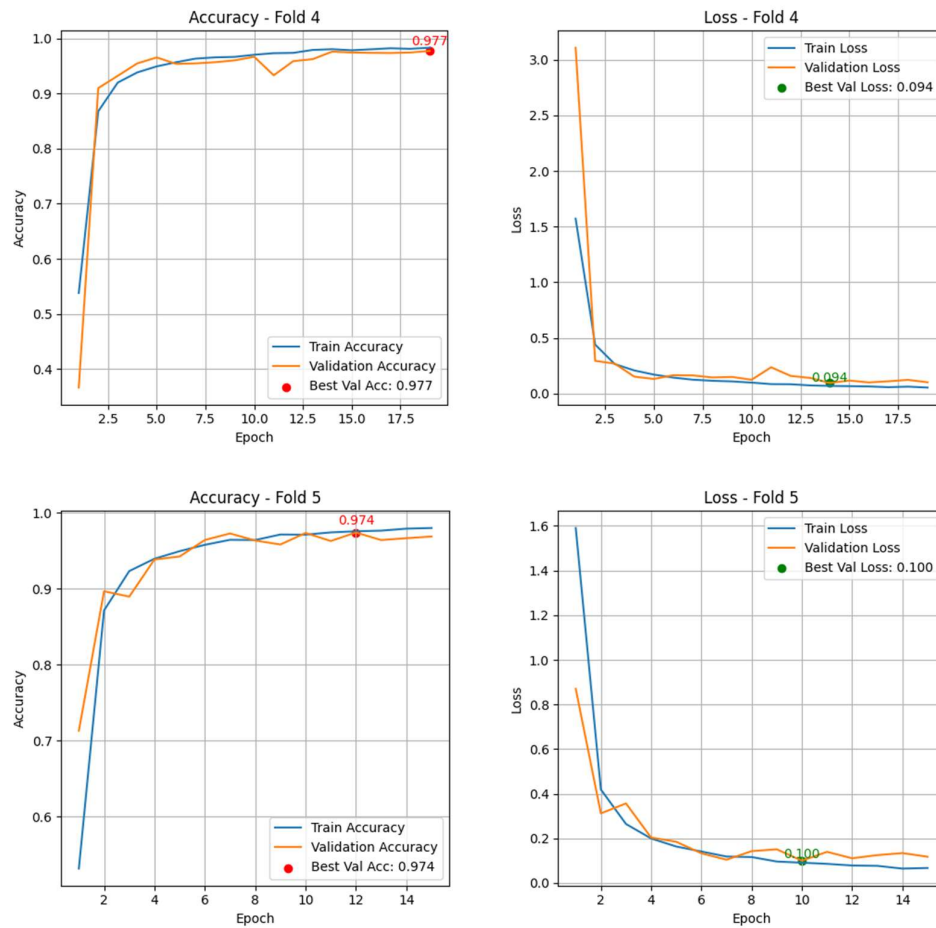
Tabel 4.2 Konfigurasi EfficientNetB0 sebagai base model dari model *ensemble*

Hyperparameter	Model Karakter	Model Kata
Input size	224 x 224 pixel	224 x 224 pixel
Batch size	32	32
<i>Epochs</i>	30	30
Learning rate	0,001	0,001
Dropout	0,4	0,2
Unfreeze layer	Layer 163 – 240	Layer 163 – 240
Stratified <i>K-fold cross validation</i>	Fold = 5 fold	Fold = 5 fold

Selanjutnya model dilatih dengan konfigurasi EfficientNetB0 seperti pada tabel 4.2. Pelatihan dilakukan dengan 30 *epochs*. Namun begitu, diberikan fungsi *early stopping* dengan nilai *patience* 5 *epochs* yang mengakibatkan pelatihan model berhenti jika nilai akurasi validasi tidak meningkat atau nilai *loss* akurasi tidak menurun. Pada pelatihan ini juga menggunakan dua *checkpoint* model untuk masing-masing fold, yaitu model dengan akurasi tertinggi dan model dengan *loss* terendah.

Untuk grafik akurasi dan *loss* masing-masing fold dari kedua model yang dilatih terlihat seperti berikut.



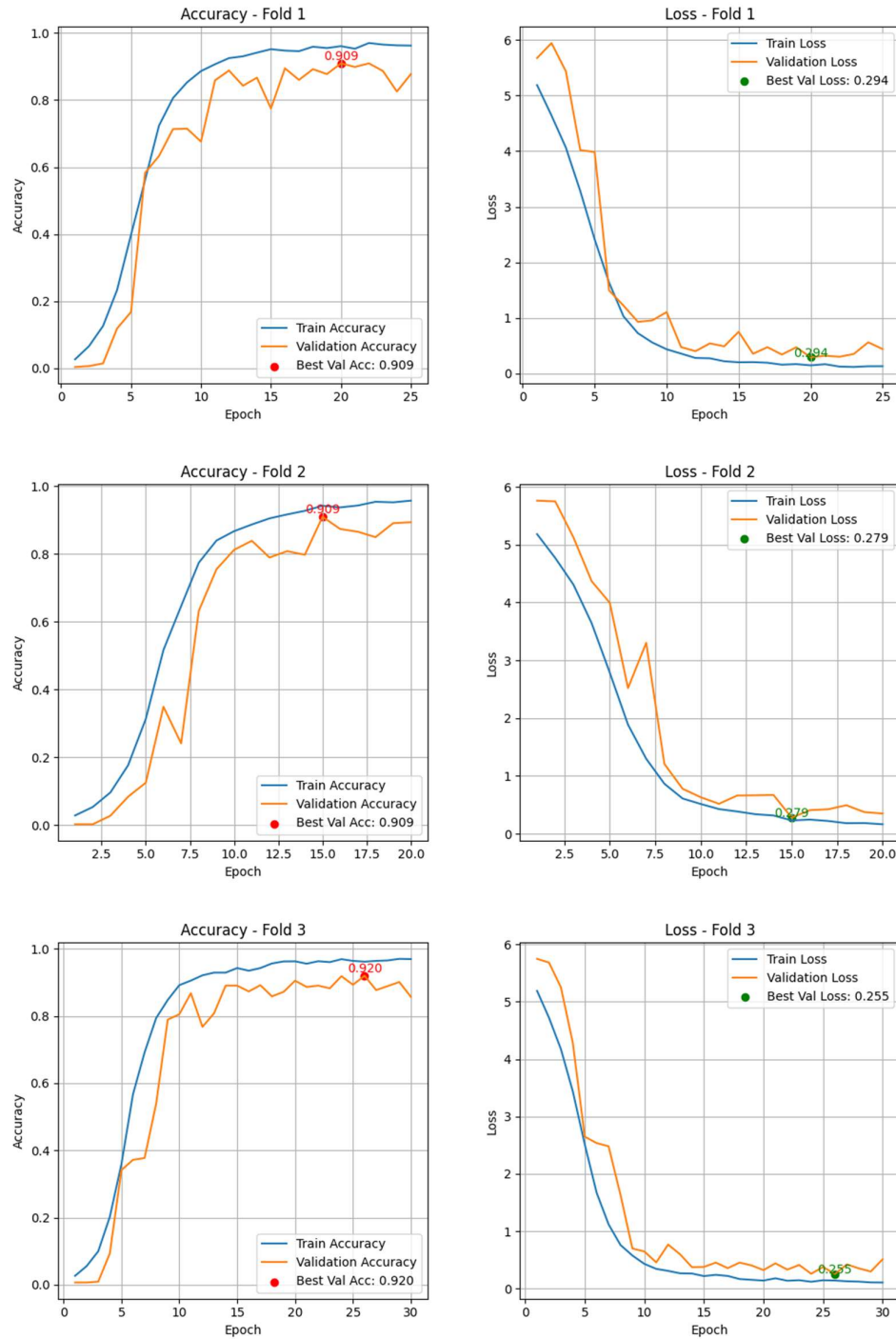


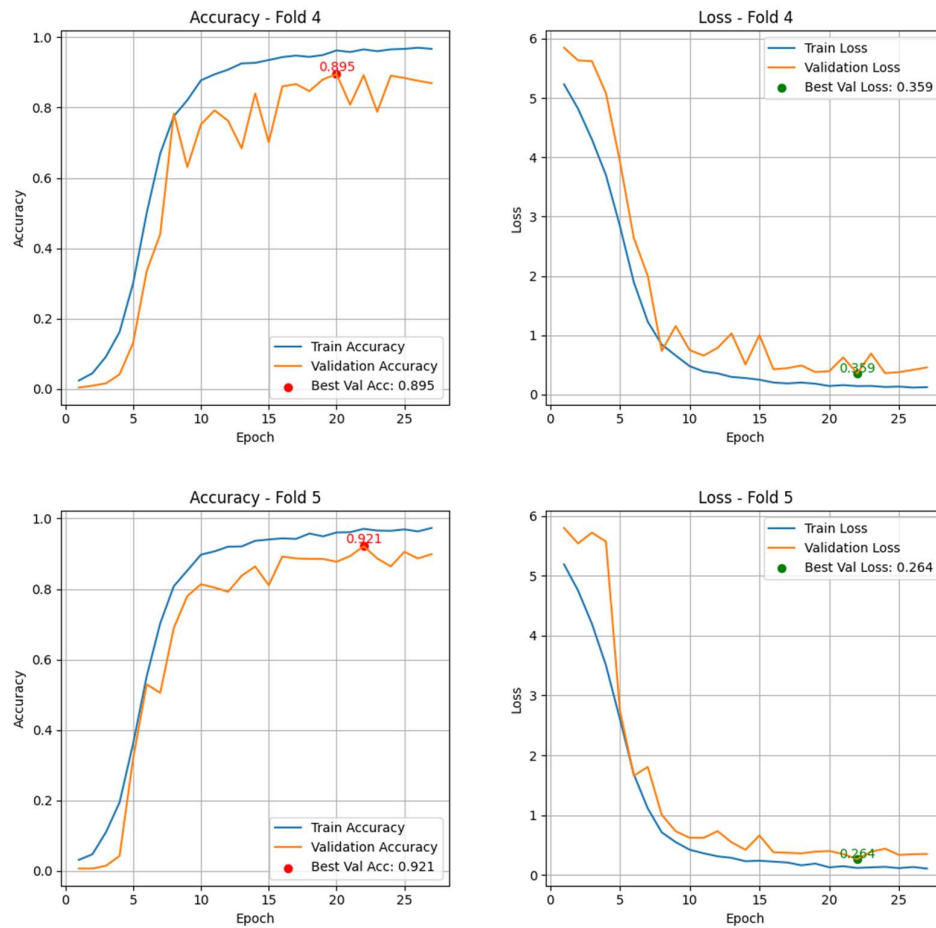
Gambar 4.1 Grafik akurasi dan loss dari pelatihan model karakter

Dari gambar 4.1 menunjukkan perbedaan *epochs* yang diperlukan agar masing-masing fold mencapai nilai akurasi tertinggi dan nilai *loss* terendah. Perbedaan ini disebabkan oleh fungsi *early stopping* yang menghentikan pelatihan model jika dalam 5 *epochs* nilai yang dievaluasi tidak meningkat. Fold 1 menghasilkan nilai akurasi 0,979 dan nilai *loss* 0,089 dengan *early stopping* dipicu oleh *checkpoint* pada metrik *loss* di *epochs* ke-21. Fold 2 menghasilkan nilai akurasi 0,979 dan nilai *loss* 0,121 dengan *early stopping* dipicu oleh *checkpoint* pada metrik *loss* di *epochs* ke-19. Fold 3 menghasilkan nilai akurasi 0,970 dan nilai *loss* 0,101 dengan *early stopping* dipicu oleh kedua metrik *checkpoint* di *epochs* ke-13. Fold 4 menghasilkan nilai akurasi 0,977 dan nilai *loss* 0,094 dengan *early stopping* dipicu oleh *checkpoint* pada metrik *loss* di *epochs* ke-19. Fold 5 menghasilkan nilai akurasi 0,974 dan nilai

loss 0,100 dengan *early stopping* dipicu oleh *checkpoint* pada metrik *loss* di *epochs* ke-15.

Grafik akurasi dan *loss* untuk pelatihan model kata adalah sebagai berikut.





Gambar 4.2 Grafik akurasi dan loss dari pelatihan model kata

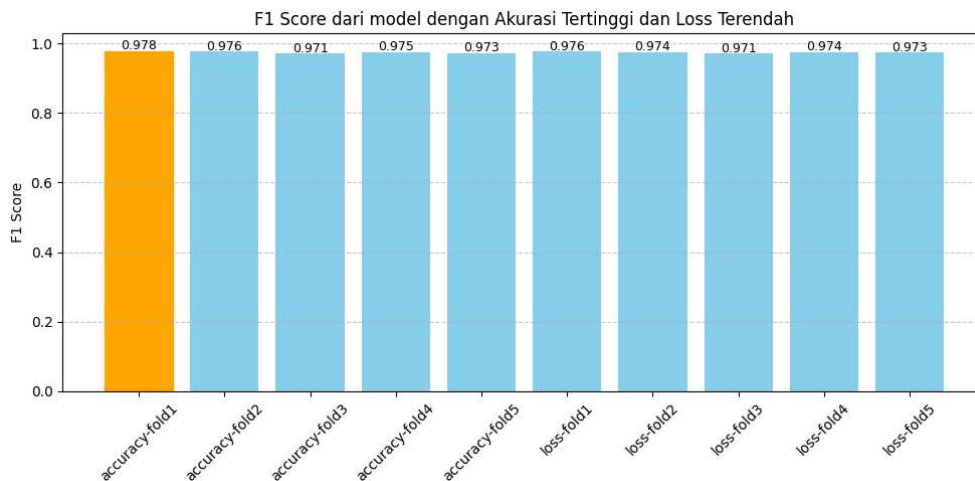
Gambar 4.2 menunjukkan pelatihan model kata membutuhkan *epochs* lebih banyak daripada model karakter. Bahkan, ada satu fold yang mencapai 30 *epochs* penuh. Fold 1 menghasilkan nilai akurasi 0,909 dan nilai *loss* 0,294 dengan *early stopping* dipicu oleh kedua metrik *checkpoint* di *epochs* ke-25. Fold 2 menghasilkan nilai akurasi 0,909 dan nilai *loss* 0,279 dengan *early stopping* dipicu oleh kedua metrik *checkpoint* di *epochs* ke-20. Fold 3 menghasilkan nilai akurasi 0,920 dan nilai *loss* 0,255 dengan full 30 *epochs*. Fold 4 menghasilkan nilai akurasi 0,895 dan nilai *loss* 0,359 dengan *early stopping* dipicu oleh *checkpoint* pada metrik *loss* di *epochs* ke-27. Fold 5 menghasilkan nilai akurasi 0,921 dan nilai *loss* 0,264 dengan *early stopping* dipicu oleh *checkpoint* pada metrik *loss* di *epochs* ke-27.

Dari pelatihan diatas, data pelatihan kedua model dapat dirangkum sebagai berikut.

Tabel 4.3 Perbandingan *epochs*, akurasi, dan *loss* pada pelatihan setiap fold untuk kedua model

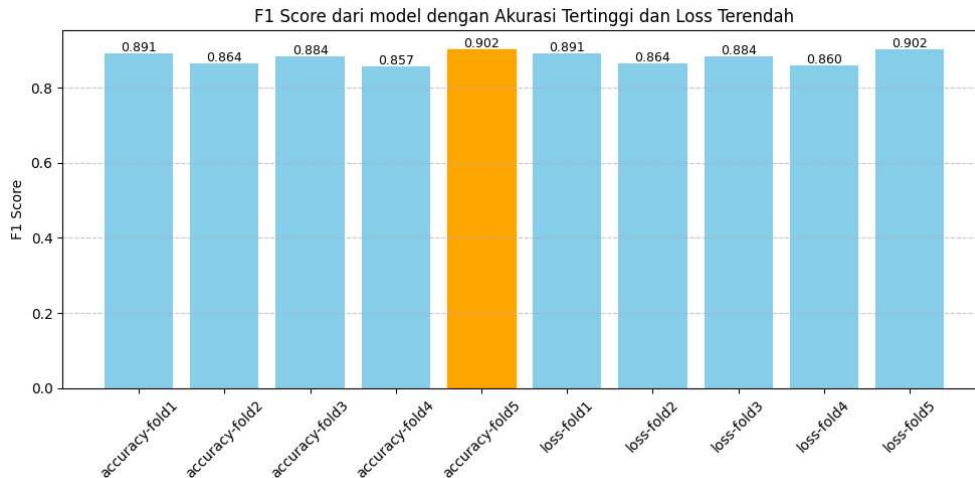
		Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Model Karakter	<i>Epochs</i>	21	19	13	19	15
	Akurasi	0,978	0,978	0,970	0,975	0,974
	<i>Loss</i>	0,088	0,120	0,101	0,094	0,099
Model Kata	<i>Epochs</i>	25	20	30	27	27
	Akurasi	0,909	0,909	0,920	0,895	0,921
	<i>Loss</i>	0,294	0,279	0,255	0,359	0,264

Menurut tabel 4.3 rata-rata *epochs* pelatihan model karakter berjalan selama 17,4 *epochs*. Sementara itu, rata-rata *epochs* pelatihan model kata berjalan selama 25,8 *epochs*. Dari data pelatihan diatas menunjukan perbedaan kompleksitas dataset digunakan oleh kedua model. Dataset kata memiliki 220 kelas dengan jumlah rata-rata per-kelas memilki 15 data. Hal ini menunjukan level kompleksitas yang berbeda jika dibandingkan dataset karakter yang hanya memiliki 32 kelas dengan rata-rata per-kelas memiliki 50 data.



Gambar 4.3 Perbandingan F1 score dari model akurasi tertinggi dan *loss* terendah untuk model karakter

Gambar 4.3 menunjukkan perbandingan nilai F1 score dari model yang dihasilkan setiap fold pelatihan model karakter. Nilai tertinggi dari evaluasi dihasilkan oleh model akurasi pelatihan tertinggi fold ke-1 sebesar 0,978. Sementara itu, nilai terendah dihasilkan oleh kedua model fold ke-3 sebesar 0,971. Secara keseluruhan, rata-rata F1 score evaluasi model karakter adalah sebesar 0,9741.



Gambar 4.4 Perbandingan F1 score dari model akurasi tertinggi dan loss terendah untuk model kata

Gambar 4.4 menunjukkan perbandingan nilai F1 score dari model yang dihasilkan setiap fold pelatihan model kata. Nilai tertinggi dari evaluasi dihasilkan oleh kedua model fold ke-5 sebesar 0,902. Sementara itu, nilai terendah dihasilkan oleh model akurasi pelatihan tertinggi model fold ke-4 sebesar 0,857. Secara keseluruhan, rata-rata F1 score evaluasi model kata adalah sebesar 0,8799.

Dari perbandingan nilai F1-score diatas, model karakter yang dipilih untuk digunakan dalam model *ensemble* adalah model dengan akurasi tertinggi di fold ke-1. Sementara itu, model kata yang dipilih untuk digunakan dalam model *ensemble* adalah model dengan akurasi tertinggi di fold ke-5.

4.1.2. Model *Ensemble*


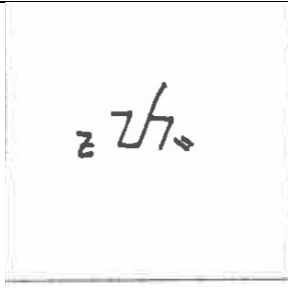
Kedua model terbaik dari proses pelatihan sebelumnya digabungkan menjadi satu model dengan metode *confidence* voting dengan menggunakan nilai α sebagai *ensemble weight*. Untuk mengetahui nilai α terbaik dilakukan evaluasi dengan hasil terlihat di tabel 4.4.



Tabel 4.4 Perbandingan hasil evaluasi nilai α sebagai nilai *ensemble weight*

Nilai α	0,4	0,5	0,6
Akurasi	0,597	0,912	0,843
F1 – score (<i>macro</i>)	0,795	0,755	0,380
F1 – score (<i>weighted</i>)	0,676	0,910	0,819
Presisi	0,855	0,889	0,571
Sensitifitas	0,827	0,701	0,319

Data perbandingan nilai α diatas menunjukkan nilai 0,4 dan nilai 0,5 yang bisa menjadi pilihan terbaik. Nilai 0,5 memberikan evaluasi akurasi, presisi dan nilai F1 (*weighted*) paling tinggi, sementara nilai 0,4 memberikan evaluasi nilai F1 (*macro*) dan *recall* paling tinggi. Perbedaan hasil evaluasi ini disebabkan ada perbedaan jumlah kelas dengan data per-kelas untuk dataset karakter dan dataset kata.

Tabel 4.5 Perbandingan nilai α terhadap gambar téh

Nilai α	0,4		0,5	
Gambar (Input)				
Model	Karakter	Kata	Karakter	Kata
Prediksi model	ta	téh	ta	téh
<i>Confidence</i> (model)	0,826	0,999	0,826	0,999

<i>Confidence (weighted)</i>	0,495	0,399	0,413	0,499
Prediksi Akhir	ta		téh	
Ilustrasi prediksi dalam Aksara Sunda				

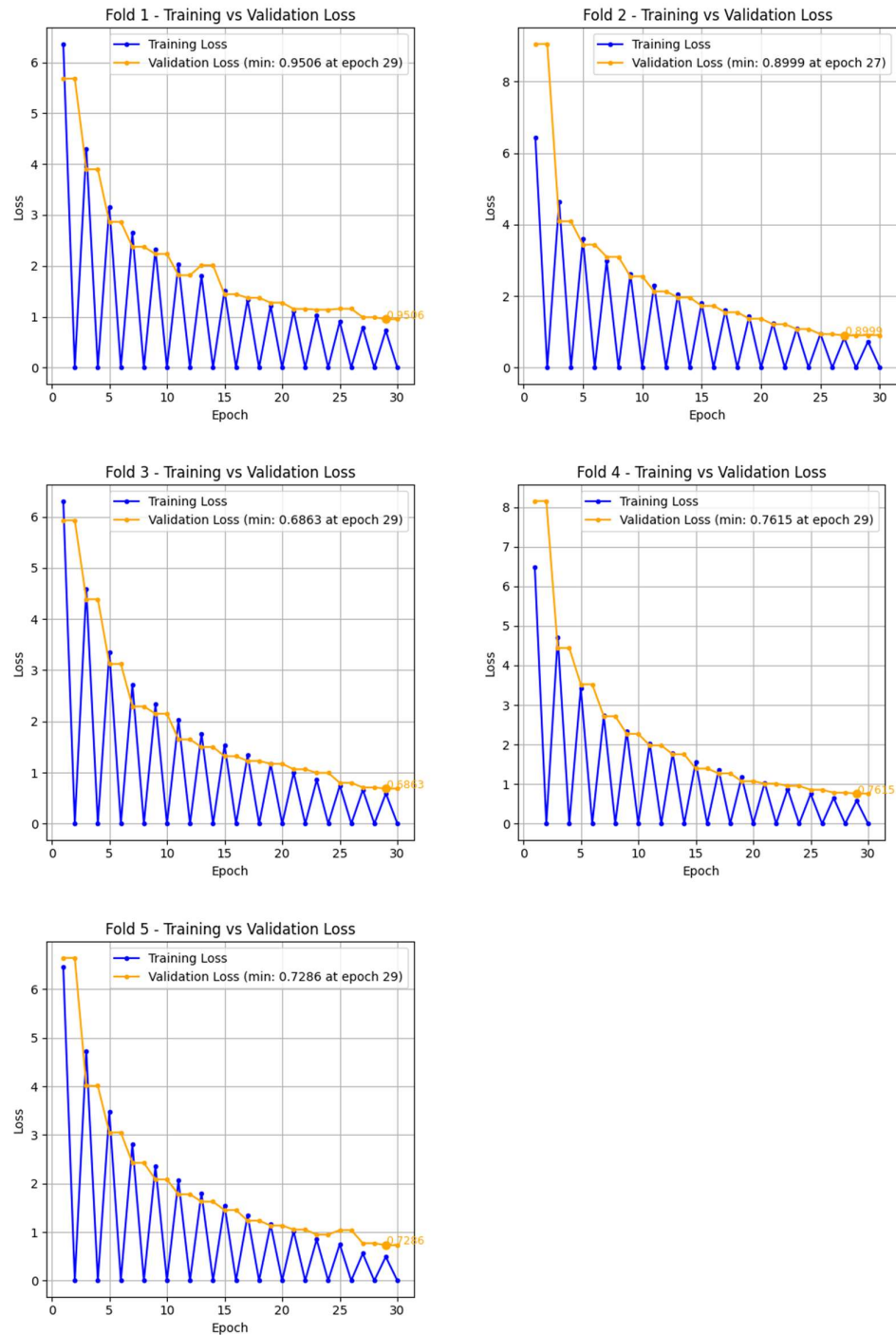
Tabel 4.5 memberikan ilustrasi kenapa nilai α bernilai 0,4 menghasilkan performa akurasi yang buruk. Kasus ini terjadi karena adanya penambahan satu atau dua rarangkén menjadi satu kata yang menurunkan nilai *confidence* dari model karakter. Namun, penurunan tersebut tidak cukup signifikan untuk mengurangi nilai akhir *confidence (weighted)* model karakter agar lebih rendah daripada nilai *confidence (weighted)* dari model kata. Akibatnya, prediksi akhir tetap mengacu pada prediksi model karakter, padahal dalam kasus tersebut prediksi yang lebih diandalkan adalah prediksi dari model kata.

4.2. Skenario Model CTC

Tabel 4.6 Konfigurasi model CTC yang dilatih

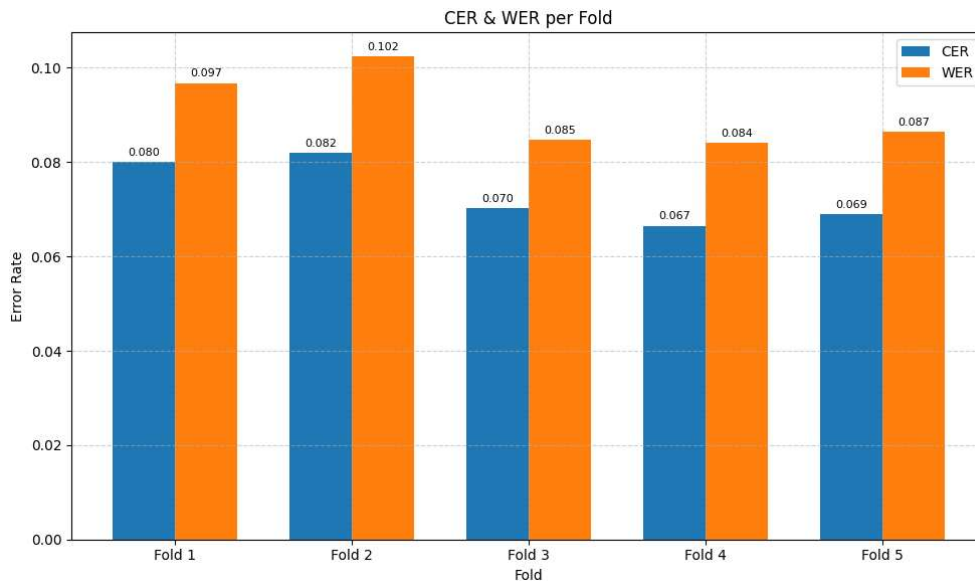
Hyperparameter	Nilai
Input shape	224, 224, 3
CNN Model	EfficientNetB0 <ul style="list-style-type: none"> - Unfreeze dari layer 163 – 240 - 240 layer
RNN layer	BiLSTM – 2 layer
<i>Epochs</i> maksimal	30
Dropout	0,2
Learning rate	0,0001

Skenario pelatihan model CTC menggunakan metode *Stratified K-fold Cross Validation* sebanyak 5 fold dengan konfigurasi di tabel 4.6. Evaluasi model yang digunakan dalam pelatihan model CTC adalah fungsi CTC loss.



Gambar 4.5 Grafik CTC Loss pada pelatihan 5 fold

Gambar 4.5 menunjukkan grafik CTC *loss* pada pelatihan model CTC. CTC *loss* terendah didapatkan pada fold ke-3 dengan nilai 0,686. Sedangkan CTC *loss* tertinggi didapatkan pada fold ke-1 dengan nilai 0,950. Dari grafik diatas terlihat bahwa meskipun ditetapkan *early stopping*, proses pelatihan model CTC menyentuh 30 *epochs* pada setiap fold. Hal tersebut menunjukkan masih ada kemungkinan performa model membaik jika *epochs* maksimal ditingkatkan melebihi dari 30.



Gambar 4.6 Perbandingan hasil evaluasi CER dan WER setiap fold

Dalam penelitian ini, evaluasi dilakukan pada tingkat kata, dan tidak mempertimbangkan kalimat. Gambar 4.6 menunjukkan nilai metrik WER lebih tinggi daripada metrik CER pada setiap fold. Hal ini disebabkan jika 1 huruf saja salah, maka sudah pasti satu kata dianggap salah dalam evaluasi WER. Setiap fold menghasilkan CER dan WER yang cukup konsisten dengan variasi yang tidak terlalu besar. Model CTC dengan performa terburuk dihasilkan oleh fold ke-2 dengan nilai CER sebesar 0,082 dan nilai WER sebesar 0,102. Model CTC terbaik dihasilkan oleh fold ke-4 dengan nilai CER sebesar 0,067 dan nilai WER sebesar 0,084. Rata-rata nilai CER adalah sebesar 0,074 dan rata-rata nilai WER adalah sebesar 0,091.

4.3. Analisis Akhir

Model karakter dilatih dengan *hyperparameter* berupa *dropout* sebesar 0,4 dan *learning rate* sebesar 0,001. Nilai *F1-score* rata-rata hasil pelatihan model karakter adalah 0,974 dengan model terbaik menunjukkan nilai *F1-score* sebesar 0,978.

Sementara itu, model kata dilatih dengan *hyperparameter dropout* sebesar 0,2 dan *learning rate* sebesar 0,001. Nilai *F1-score* rata-rata hasil pelatihan model kata adalah 0,880 dengan model terbaik menunjukkan nilai *F1-score* sebesar 0,902.

Kedua model tersebut kemudian digabung menjadi model *ensemble* dengan nilai *ensemble weight* sebesar 0,5. Evaluasi model menghasilkan akurasi sebesar 0,912, *F1-score* sebesar 0,755, presisi sebesar 0,889 dan sensitifitas sebesar 0,701.

Selain itu, model CTC juga menunjukkan performa yang baik dengan rata-rata nilai CER adalah sebesar 0,074 dan rata-rata nilai WER adalah sebesar 0,091. Model CTC yang terbaik nilai CER 0,082 dan nilai WER 0,102.

BAB 5 KESIMPULAN DAN SARAN

5.1. Kesimpulan

Penelitian ini bertujuan untuk mengimplementasikan Optical Character Recognition (OCR) tulisan tangan Aksara Sunda menggunakan model berbasis EfficientNet. Model OCR yang dikembangkan menggunakan EfficientNetB0 sebagai model dasar dengan dua dataset, yaitu dataset karakter dan dataset kata. Penelitian ini menghasilkan dua pendekatan model yaitu model *ensemble* dan model *Connectionist Temporal Classification* (CTC). Model *ensemble* dengan *ensemble weight* sebesar 0,5 berhasil mencapai akurasi sebesar 0,912, presisi sebesar 0,889, sensitifitas sebesar 0,701, dan *F1-score* sebesar 0,755. Model CTC berhasil mencapai nilai *Character Error Rate* (CER) sebesar 0,082 dan nilai *Word Error Rate* (WER) sebesar 0,102. Hasil ini menunjukkan penambahan dataset kata sebagai dataset baru membuka peluang pengembangan lebih lanjut dalam sistem OCR Aksara Sunda. Selain itu, performa model CTC yang cukup kompetitif memperlihatkan potensi besar dari penerapan metode ini dalam sistem OCR, tidak hanya Aksara Sunda tapi juga aksara nusantara lainnya.

5.2. Saran

Pada penelitian ini terdapat beberapa hal yang dapat ditingkatkan lagi.

1. Dataset kata yang harus lebih dikembangkan lagi, baik dari sisi kuantitas kelas yang ada dengan kuantitas data per-kelas.
2. Implementasi metode CTC *Loss* tidak dilakukan pengujian lebih mendetail. Disarankan agar mencoba metode CTC *loss* sebagai klasifikasi sekuensial.
3. Penelitian bisa dilakukan menggunakan arsitektur CNN dan metode-metode lainnya.

DAFTAR PUSTAKA







































































- [1] K. P. dan K. R. I. Direktorat Jenderal Kebudayaan, *Statistik Kebudayaan 2023*, vol. 8. 2023. Diakses: 25 November 2024. [Daring]. Tersedia pada: https://budbas.data.kemdikbud.go.id/statistik/isi_55797b4c-197d-4108-a450-0e37cfffef80.pdf
- [2] Pemerintah Daerah Jawa Barat, “Perda no 14 Tahun 2014”.
- [3] K. Hayati, D. Dwi Sulistiyo, dan D. Fiman Ihsan, “Klasifikasi Aksara Sunda Dengan Metode Machine Learning Klasik,” 2024.
- [4] M. A. Prameswari, M. Dwi Sulistiyo, dan A. F. Ihsan, “Classification of Handwritten Sundanese Script via Transfer Learning on CNN-Based Architectures,” dalam *Proceedings - 2023 3rd International Conference on Electronic and Electrical Engineering and Intelligent System: Responsible Technology for Sustainable Humanity, ICE3IS 2023*, 2023, hlm. 401–406. doi: 10.1109/ICE3IS59323.2023.10335382.
- [5] R. R. Akram, M. Dwi Sulistiyo, A. Firman Ihsan, P. Eko Yunanto, D. Richasdy, dan M. Arzaki, “Exploring ReLU Activation Functions in CNN for Handwritten Sundanese Script Recognition,” dalam *2024 International Conference on Data Science and Its Applications, ICoDSA 2024*, 2024, hlm. 92–98. doi: 10.1109/ICoDSA62899.2024.10652217.
- [6] P. Batra, N. Phalnikar, D. Kurmi, J. Tembhurne, P. Sahare, dan T. Diwan, “OCR-MRD: performance analysis of different optical character recognition engines for medical report digitization,” *International Journal of Information Technology (Singapore)*, vol. 16, no. 1, hlm. 447–455, Jan 2024, doi: 10.1007/S41870-023-01610-2/TABLES/6.
- [7] I. Baidillah *dkk.*, “Direktori Aksara Sunda untuk Unicode,” 2008.
- [8] aksarasunda, “Aksara Sunda Computer Vision Project,” Roboflow. Diakses: 2 Desember 2024. [Daring]. Tersedia pada: <https://universe.roboflow.com/aksarasunda/aksara-sunda-eayhq>
- [9] Teknik Informatika, “Aksara Sunda Computer Vision Project,” Roboflow. Diakses: 5 Desember 2024. [Daring]. Tersedia pada: <https://universe.roboflow.com/teknik-informatika/aksara-sunda-4lcpy>
- [10] A. Angelina, “Aksara Sunda Computer Vision Project,” Roboflow. Diakses: 5 Desember 2024. [Daring]. Tersedia pada: <https://universe.roboflow.com/agatha-angelina-hnexv/aksara-sunda-xvbwp>
- [11] A. F. Ihsan, “Indonesian Local Script Characters,” 2024, *Mendeley Data*. doi: 10.17632/VFJ32BPJSF.1.
- [12] S. INGOLE, “An Extensive Guide to Convolution Neural Network- (2023),” Medium. Diakses: 8 Desember 2024. [Daring]. Tersedia pada:









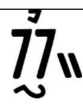




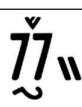

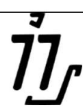
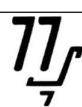



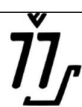









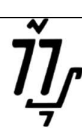

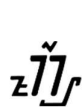



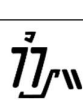
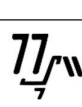


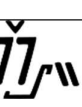
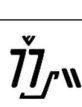




<https://medium.com/@singole/an-extensive-guide-to-convolution-neural-network-2023-84872b16bd78>

- [13] V. Pomazan, I. Tvoroshenko, dan V. Gorokhovatskyi, "Handwritten character recognition models based on convolutional neural networks," 2023, Diakses: 8 Desember 2024. [Daring]. Tersedia pada: <https://openarchive.nure.ua/entities/publication/09998e00-8e88-4642-b299-791755e6ea08>
- [14] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, dan M. Parmar, "A review of convolutional neural networks in computer vision," *Artif Intell Rev*, vol. 57, no. 4, hlm. 99, Mar 2024, doi: 10.1007/s10462-024-10721-6.
- [15] Keras Team, "Keras Applications." Diakses: 28 November 2024. [Daring]. Tersedia pada: <https://keras.io/api/applications/>
- [16] J. Riza, S. Barman, S. Ridita, Z. Mahmud, dan A. Bhattacharya, "PhytoCare : A hybrid approach for identifying Rice, Potato and Corn diseases," 2024.
- [17] Y. Jie dkk., "Combined Multi-Layer Feature Fusion and Edge Detection Method for Distributed Photovoltaic Power Station Identification," *Energies (Basel)*, vol. 13, hlm. 6742, Jan 2020, doi: 10.3390/en13246742.
- [18] R. Dodda, S. B. Reddy, A. C. Naik, dan V. Gaddam, "A Study on Handwritten Text Recognition Classification using Diverse Deep Learning Techniques and Computation of CTC Loss," *CVR Journal of Science & Technology*, vol. 26, no. 1, hlm. 107–11, Jun 2024, doi: 10.32377/CVRJST2617.
- [19] X. Liu dan W. Wang, "Deep Time Series Forecasting Models: A Comprehensive Survey," *Mathematics 2024, Vol. 12, Page 1504*, vol. 12, no. 10, hlm. 1504, Mei 2024, doi: 10.3390/MATH12101504.
- [20] S. M. Vatanchi, H. Etemadfard, M. F. Maghrebi, dan R. Shad, "A Comparative Study on Forecasting of Long-term Daily Streamflow using ANN, ANFIS, BiLSTM, and CNN-GRU-LSTM," 2022, doi: 10.21203/rs.3.rs-1443377/v1.
- [21] B. Imane, · Ammour Alae, · Khaissidi Ghizlane, dan M. Mrabti, "Enhancing Arabic handwritten word recognition: a CNN-BiLSTM-CTC architecture with attention mechanism and adaptive augmentation," *Discover Applied Sciences*, vol. 7, hlm. 460, 123M, doi: 10.1007/s42452-025-06952-z.
- [22] R. Najam dan S. Faizullah, "Analysis of Recent Deep Learning Techniques for Arabic Handwritten-Text OCR and Post-OCR Correction," 1 Juli 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/app13137568.

- [23] P. Verma dan G. Foomani, "Improvement in OCR Technologies in Postal Industry Using CNN-RNN Architecture: Literature Review," *Int J Mach Learn Comput*, vol. 12, no. 5, Sep 2022, doi: 10.18178/ijmlc.2022.12.5.1095.
- [24] X. H. Le, D. H. Nguyen, S. Jung, M. Yeon, dan G. Lee, "Comparison of Deep Learning Techniques for River Streamflow Forecasting," *IEEE Access*, vol. 9, hlm. 71805–71820, 2021, doi: 10.1109/ACCESS.2021.3077703.
- [25] L. Mosbah, I. Moalla, T. M. Hamdani, B. Neji, T. Beyrouthy, dan A. M. Alimi, "ADOCRNet: A Deep Learning OCR for Arabic Documents Recognition," *IEEE Access*, vol. 12, hlm. 55620–55631, 2024, doi: 10.1109/ACCESS.2024.3379530.
- [26] D. Vaibhav Khatavkar, D. Makarand Velankar, dan S. Petkar, "Segmentation-free Connectionist Temporal Classification loss based OCR Model for Text Captcha Classification," Feb 2024, Diakses: 18 Juni 2025. [Daring]. Tersedia pada: <https://arxiv.org/pdf/2402.05417>
- [27] K. M. M. Chit dan L. L. Lin, "Exploring CTC Based End-To-End Techniques for Myanmar Speech Recognition," 2021, doi: 10.1007/978-3-030-68154-8_87.
- [28] S. Szeghalmy dan A. Fazekas, "A Comparative Study of the Use of Stratified Cross-Validation and Distribution-Balanced Stratified Cross-Validation in Imbalanced Learning," *Sensors 2023, Vol. 23, Page 2333*, vol. 23, no. 4, hlm. 2333, Feb 2023, doi: 10.3390/S23042333.
- [29] X. Duan, "Automatic identification of conodont species using fine-grained convolutional neural networks," *Front Earth Sci (Lausanne)*, vol. 10, Jan 2023, doi: 10.3389/FEART.2022.1046327.
- [30] D. A. Neu, J. Lahann, dan P. Fettke, "A systematic literature review on state-of-the-art deep learning methods for process prediction," *Artif Intell Rev*, vol. 55, no. 2, hlm. 801–827, Feb 2022, doi: 10.1007/S10462-021-09960-8/FIGURES/6.

LAMPIRAN

 Ka	 Ki	 Ku	 Ko	 Ké	 Ke	 Keu
 Kang	 King	 Kung	 Kong	 Kéng	 Keng	 Keung
 Kah	 Kih	 Kuh	 Koh	 Kéh	 Keh	 Keuh
 Kar	 Kir	 Kur	 Kor	 Kér	 Ker	 Keur
 Kla	 Kli	 Klu	 Klo	 Klé	 Kle	 Kleu
 Klang	 Kling	 Klung	 Klong	 Kléng	 Kleng	 Kleung
 Klah	 Klih	 Kluh	 Kloh	 Kléh	 Kleh	 Kleuh
 Klar	 Klir	 Klur	 Klor	 Klér	 Kler	 Kleur
 Kra	 Kri	 Kru	 Kro	 Kré	 Kre	 Kreu
 Krang	 Kring	 Krung	 Krong	 Kréng	 Kreng	 Kreung

 Krar	 Krir	 Kru ₇ r	 Kror	 Krér	 Krer	 Kreur
 Krah	 Krih	 Kru ₇ h	 Kro _z h	 Kréh	 Kreh	 Kreuh
 Kya	 Kyi	 Kyu ₇	 Kyo _z	 Kyé	 Kye	 Kyeu
 Kyang	 Kying	 Kyung ₇	 Kyong _z	 Kyéng	 Kyeng	 Kyeung
 Kyar	 Kyir	 Kyur ₇	 Kyor _z	 Kyér	 Kyer	 Kyeur
 Kyah	 Kyih	 Kyu ₇ h	 Kyo _z h	 Kyéh	 Kyeh	 Kyeuh
 Ky	 Kl	 Kr	 K			

Tabel 6.1 Kemungkinan kombinasi ngalagena dengan rarangkén untuk satu karakter ngalagena