

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



**Universitas
Telkom**

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, 13 Agustus 2025

Menyetujui

Pembimbing I,

Pembimbing II,

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

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, 13 Agustus 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. Masih sedikit sistem OCR yang menggunakan dataset kata Aksara Sunda yang berupa gabungan antara ngalagena dan swara dengan rarangkén. 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 0,069 dan nilai *Word Error Rate* (WER) sebesar 0,087.

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 only a few of OCR using 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 0.069 and a Word Error Rate (WER) value of 0.087.

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, 13 Agustus 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 tugas akhir 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 tugas akhir 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, masukan 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 Dr. Tjokorda Agung Budi Wirayuda, 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 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	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 <i>Loss</i>	9
2.6. Stratified K-Fold Cross Validation	9
2.7. Confusion Matrix	9
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)	15
3.2. Preprocessing Data	17

3.3.	Evaluasi Sistem	20
3.3.1.	Evaluasi Model Ensemble	20
3.3.2.	Evaluasi Model CTC	21
BAB 4	HASIL PERCOBAAN DAN ANALISIS	22
4.1.	Skenario Model Ensemble	22
4.1.1.	Model Karakter dan Model Kata	22
4.1.2.	Model <i>Ensemble</i>	30
4.2.	Skenario Model CTC	32
4.3.	Analisis Akhir	36
BAB 5	KESIMPULAN DAN SARAN	38
5.1.	Kesimpulan	38
5.2.	Saran	38
DAFTAR PUSTAKA		39
LAMPIRAN		43

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[15].....	7
Gambar 2.4. Perbandingan Arsitektur EfficientNetB0[17] dengan EfficientNetB1[18].....	8
Gambar 2.5. Stratified K-fold cross validation[30]	9
Gambar 3.1. Diagram alir model <i>ensemble</i>	12
Gambar 3.2. Implementasi EfficientNetB0 sebagai model dasar dari model <i>ensemble</i>	13
Gambar 3.3. Diagram Alir dari pelatihan model yang digunakan untuk <i>Ensemble Model</i>	14
Gambar 3.4. Implementasi EfficientNetB0 sebagai model dasar dari model CTC.....	15
Gambar 3.5. Diagram alir pelatihan model CTC	16
Gambar 3.6. Proses pengumpulan dataset kata	17
Gambar 3.7. Contoh dataset kata bumi, syukur, abah, dan ti	17
Gambar 3.8. <i>Label encoding</i> dan <i>sequence labelling</i>	19
Gambar 4.1. Grafik akurasi dan <i>loss</i> dari pelatihan model karakter	25
Gambar 4.2. Grafik akurasi dan <i>loss</i> dari pelatihan model kata	27
Gambar 4.3. Perbandingan F1 score dari model akurasi tertinggi dan <i>loss</i> terendah untuk model karakter	29
Gambar 4.4. Perbandingan F1 score dari model akurasi tertinggi dan <i>loss</i> terendah untuk model kata	29
Gambar 4.5. Grafik CTC <i>Loss</i> pada pelatihan 5 fold	34
Gambar 4.6. Perbandingan hasil evaluasi CER dan WER setiap fold	35

DAFTAR TABEL

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.....	22
Tabel 4.2. Konfigurasi EfficientNetB0 sebagai base model dari model <i>ensemble</i>	23
Tabel 4.3. Perbandingan <i>epochs</i> , akurasi, dan <i>loss</i> pada pelatihan setiap fold untuk kedua model	28
Tabel 4.4. Perbandingan hasil evaluasi nilai α sebagai nilai <i>ensemble weight</i>	30
Tabel 4.5. Perbandingan nilai α terhadap gambar téh	31
Tabel 4.6. Hasil pengaturan <i>hyperparameter</i> untuk model CTC	32
Tabel 4.7. Konfigurasi model CTC yang dilatih.....	32
Tabel 6.1. Kemungkinan kombinasi ngalagena dengan rarangkén untuk satu karakter ngalagena	44

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. Hal ini menunjukkan masih sedikit penelitian yang berfokus pada klasifikasi kata untuk OCR Aksara Sunda. Salah satu dari sedikitnya penelitian yang ada tentang OCR Aksara Sunda untuk klasifikasi kata dilakukan oleh Gerhana Yana, et al[6] yang menggunakan metode *Template Matching Algorithm* dan *Feature Extraction Algorithm*. Performa model pada penelitian ini untuk klasifikasi kata dari tulisan tangan secara spesifik mencapai akurasi dengan nilai 75% dari 30 sample kata. Oleh karena itu, diperlukan penelitian lebih lanjut pada pengklasifikasian kata sebagai langkah pengembangan berikutnya dalam OCR khususnya tulisan Aksara Sunda.

1.2. Rumusan Masalah

Permasalahan yang diangkat pada penelitian ini adalah belum adanya sistem *Optical Character Recognition* (OCR) pada tingkat kata. Kriteria yang menjadi fokus dalam permasalahan ini adalah menghasilkan model klasifikasi kata tulisan tangan Aksara Sunda tanpa menggunakan model segmentasi. Rumusan masalah diatas dapat dirinci menjadi sebagai berikut:

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

1.3. Tujuan dan Manfaat

Tujuan dari penelitian ini yaitu sebagai berikut:

1. Mengimplementasikan EfficientNet sebagai model dasar pada *Optical Character Recognition* untuk mengenali pola kata tulisan tangan Aksara Sunda.
2. Menganalisis performa EfficientNet 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. Dataset kata yang dikumpulkan berasal dari kumpulan kalimat bahasa sunda secara acak. Dataset kata terdiri dari maksimal 3 karakter ngalagena atau swara dengan rarangkén
2. EfficientNet yang digunakan adalah EfficientNetB0.
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.

BAB 2 TINJAUAN PUSTAKA

2.1. Optical Character Recognition













Menurut Batra P, et al[7], *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[8], 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. Sementara itu, aksara swara pada Aksara Sunda berjumlah 7 karakter seperti ditunjukkan pada tabel 2.2.

Tabel 2.1. Aksara Sunda ngalagena.

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

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. Kemungkinan dari satu karakter ngalagena secara maksimum dapat dilihat pada tabel 6.1 di lampiran. Sementara itu, contoh kalimat dapat dilihat seperti pada gambar 2.1.

Gambar 2.1. Lorem ipsum ditulis dalam Aksara Sunda

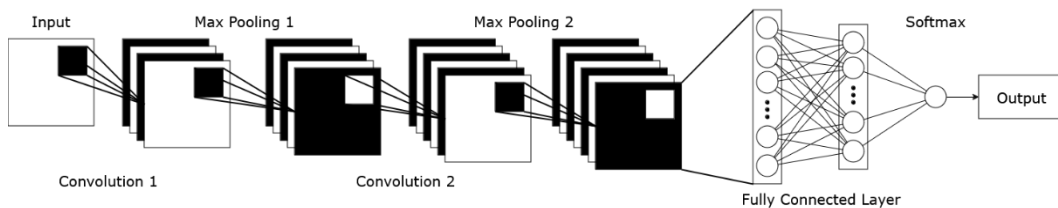
Namun, dari sekian banyak sumber dataset yang ada secara *online*, dicontohkan dari situs Roboflow[9], [10], [11] dan situs Mendeley[12], tidak ada data mengenai gabungan dari setiap karakter ngalagena dengan masing-masing karakter rarangkén seperti dicontohkan pada gambar 2.2. Maka dibutuhkan pembuatan 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)[13]. Viktor Pomazan *et al*[14] 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*[15].

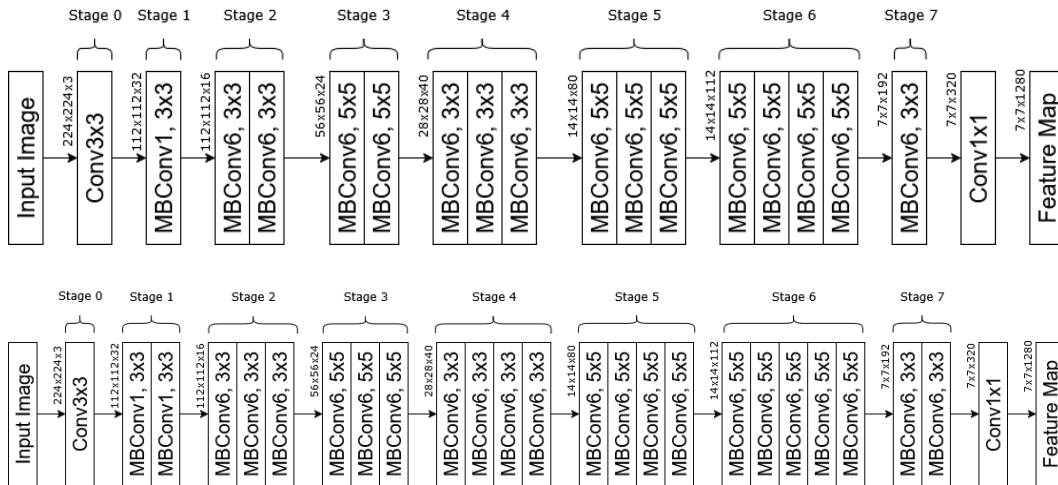


Gambar 2.3. Contoh struktur CNN[15]

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[16], EfficientNet, khususnya EfficientNetB0, merupakan model dengan ukuran 29MB menjadikannya 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. Gambar

2.4 menunjukkan contoh peningkatan tersebut dengan contoh EfficientNetB0 dan EfficientNetB1. Hal tersebut menjadikan EfficientNet menjadi salah satu pilihan model yang adaptif mengikuti jumlah sumber daya yang ada.



Gambar 2.4. Perbandingan Arsitektur EfficientNetB0[17] dengan EfficientNetB1[18]

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[19]. 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[20], [21], dan Optical Character Recognition [22], [23]. Beberapa penelitian juga menyarankan gabungan CNN dengan RNN dengan tujuan saling melengkapi[24].

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[23].

2.4.2. Bidirectional Long-Short Term Memory

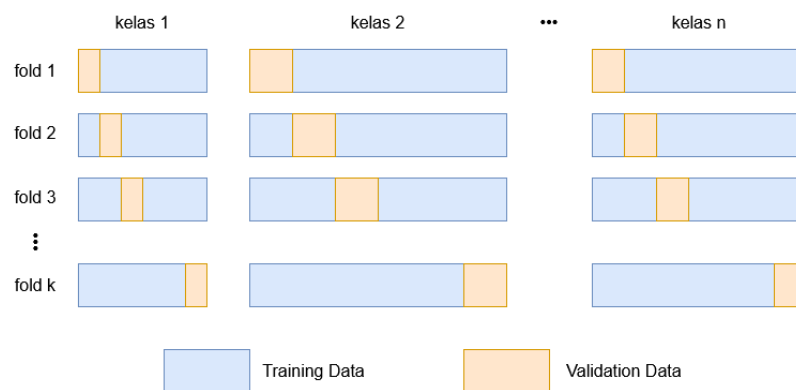
Bidirectional Long-Short Term Memory (BiLSTM) merupakan bentuk pengembangan lainnya dari RNN dan LSTM. Berbeda dengan kedua metode tersebut yang hanya bisa memproses data sekuensial dalam satu arah, BiLSTM dirancang agar bisa memproses data sekuensial dalam dua arah[25]. Metode ini digunakan dalam beberapa penelitian mengenai OCR untuk bahasa arab[26].

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[27]. CTC Loss akan mengevaluasi data input sekuensial dengan menambahkan token “blank” sebagai pengganti data yang berulang dalam sekuensial tersebut[28].

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[29]. Hal ini memastikan model melihat semua kelas walaupun ada perbedaan jumlah data setiap kelas.



Gambar 2.5. Stratified K-fold cross validation[30]

2.7. Confusion Matrix

Confusion matrix merupakan metrik yang umum digunakan dalam pengevaluasian *machine learning* tipe klasifikasi[31]. 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[23]. Hal yang diukur adalah kesalahan penambahan, kesalahan penghapusan, dan kesalahan substitusi baik itu dalam tingkat karakter ataupun kata.

BAB 3 PERANCANGAN SISTEM

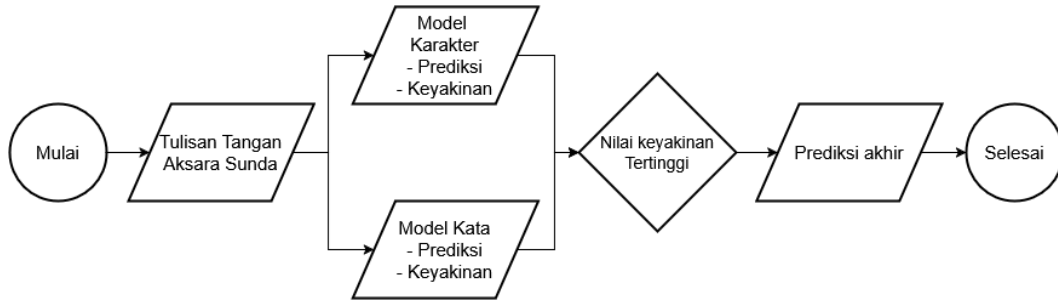
3.1. Desain Perancangan Sistem

Penelitian ini bertujuan untuk membangun sistem OCR yang mengenali kata dalam tulisan tangan Aksara Sunda dengan model berbasis EfficientNet. Pada proses penelitian ini, EfficientNetB0 dipilih sebagai model dasar dari model yang akan diimplementasikan. Arsitektur ini memiliki kemampuan komputasi yang cukup memadai untuk pelatihan model transfer learning dengan dataset berukuran kecil hingga sedang[32].

Arsitektur MobileNetV2 tidak dipilih karena berdasarkan penelitian oleh Permana, *et al*[33], EfficientNetB0 lebih unggul daripada MobileNetV2. Sementara itu, EfficientNetV2 tidak dipilih dengan alasan ukuran dataset yang masih tergolong sedang. Hal ini menjadikan keunggulan utama dari EfficientNetV2 dibanding EfficientNetB0, yaitu waktu pelatihan yang lebih singkat dan parameter yang lebih efisien tidak terlalu signifikan dalam penelitian ini[34]. Model akan diimplementasikan menjadi dua model yang berbeda, yaitu klasifikasi menggunakan *ensemble learning* dan klasifikasi menggunakan fungsi *Connectionist Temporal Classification Loss*.

3.1.1. Model Ensemble

Model pertama yang diimplementasikan adalah 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 model *ensemble* dihasilkan melalui *confidence (weighted) voting* dengan cara membandingkan nilai *confidence (weighted)* dari kedua model.



Gambar 3.1. Diagram alir model *ensemble*

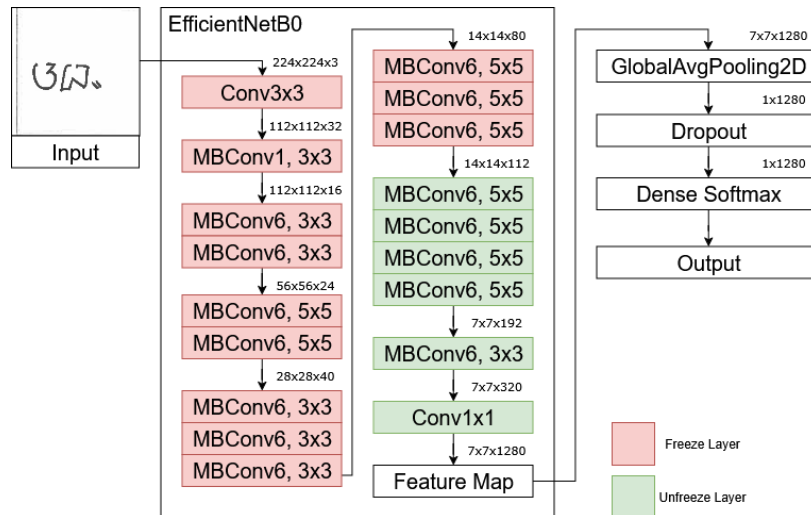
Nilai *confidence (weighted)* masing-masing model didapatkan dengan melakukan perhitungan menggunakan nilai α sebagai *ensemble weight*. Nilai α merepresentasikan bobot tingkat kepercayaan terhadap masing-masing model. Perhitungan *confidence (weighted)* dapat dilihat sebagai berikut.

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

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

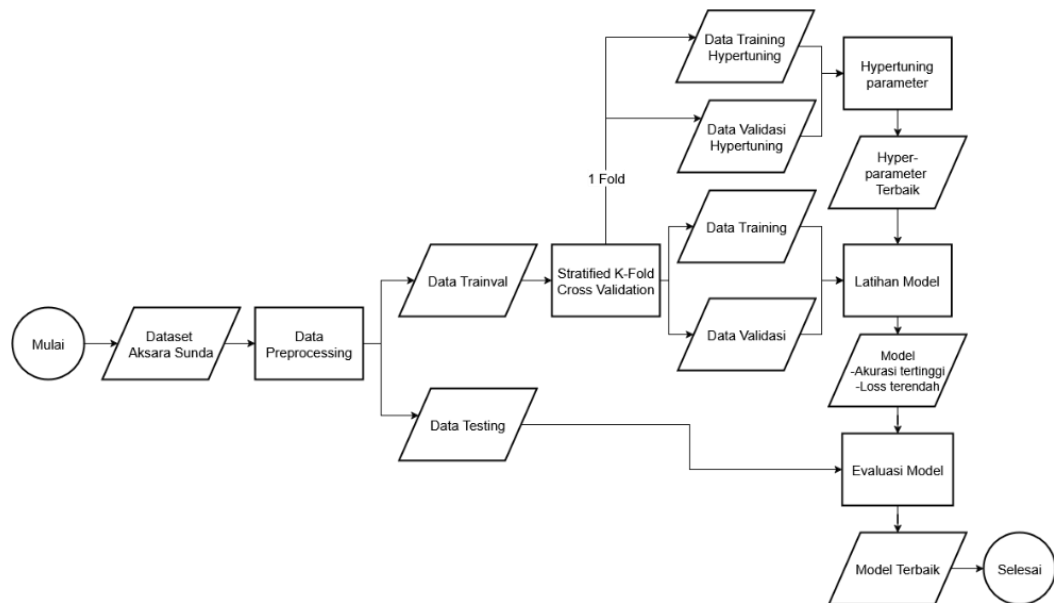
Nilai α yang digunakan dalam rumus (3.1) dan (3.2) akan dites dengan rentang 0,4; 0,5; dan 0,6. Rentang ini dipilih dengan mempertimbangkan pengaruh model sebagai dasar dari model *ensemble*. Nilai $\alpha = 0,4$ memberikan bobot lebih kepada model karakter, nilai $\alpha = 0,6$ memberikan bobot lebih kepada model kata, sementara itu nilai $\alpha = 0,5$ memberikan bobot yang sama antara dua model.

Pada model *ensemble* ini, model EfficientNetB0 yang digunakan sebagai model dasar diimplementasikan seperti terlihat 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 digunakan dalam beberapa penelitian agar mencapai performa yang lebih baik agar model bisa melakukan *fine-tuning* terhadap dataset khusus, dalam penelitian ini tulisan Aksara Sunda yang berbeda dengan tulisan latin[35], [36]. Layer *dropout* digunakan agar model tidak mengalami *overfitting* sebagai salah satu *hyperparameter* yang digunakan.



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

Berdasarkan gambar 3.2 yang menggambarkan arsitektur model *ensemble*, proses pelatihan model karakter dan model kata ditunjukkan pada gambar 3.3. Proses pelatihan tersebut terdiri dari dua tahap, yaitu *hypertuning parameter* dan pelatihan secara menyeluruh. Tahap *hypertuning parameter* dilakukan untuk mengetahui dua parameter terbaik yang akan dipakai dalam pelatihan secara menyeluruh, yaitu *learning rate* dan *dropout*. Rentang nilai *learning rate* yang dipakai adalah dari 0,001; 0,0001; dan 0,00005, sedangkan rentang nilai *dropout* adalah 0,2; 0,3; dan 0,4[37], [38]. Perbedaan antara tahap *hypertuning parameter* dengan pelatihan menyeluruh adalah jumlah *epochs*, yaitu 5 *epochs* untuk tahap *hypertuning parameter* sedangkan 30 *epochs* untuk tahap latihan menyeluruh.



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

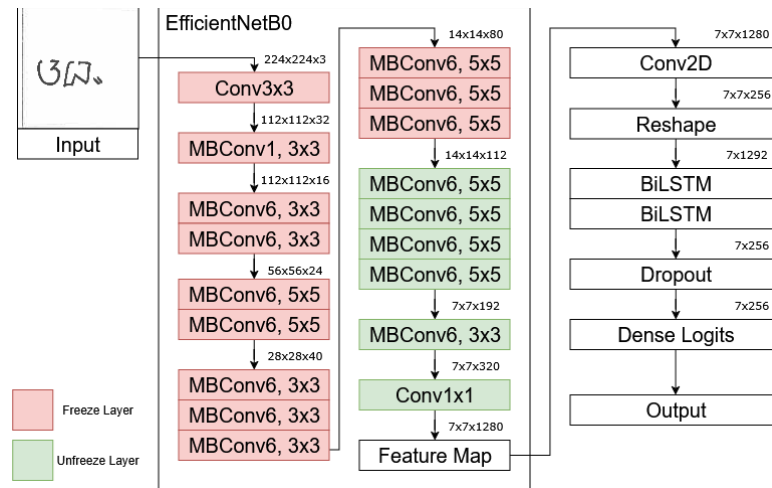
Ditunjukkan pada gambar 3.3 proses pelatihan dimulai dengan proses pembagian dataset. Proses pembagian dataset dilakukan dengan ratio 80:20 untuk data trainval dan data testing. Proses *Stratified K-Fold Cross Validation* dilakukan dua kali, yaitu untuk proses *Hypertuning Parameter* dan pelatihan model. 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 dan dipilih sebagai model dasar pada model *ensemble*.

Tahap evaluasi dari masing-masing model menggunakan metrik *F1-score (weighted)*. Hal ini dilakukan dengan tujuan untuk memberikan gambaran komprehensif terhadap performa model terhadap masing-masing dataset. Pemilihan *F1-score (weighted)* dibandingkan dengan metrik akurasi dilakukan karena metrik akurasi cenderung bias terhadap kelas mayoritas. Hal tersebut dilakukan dengan pertimbangan bahwa dataset kata yang memiliki distribusi

kata antar kelas yang masih timpang. Sehingga penggunaan metrik ini diyakini lebih tepat dan representatif.

3.1.2. Model Connectionist Temporal Classification (CTC)

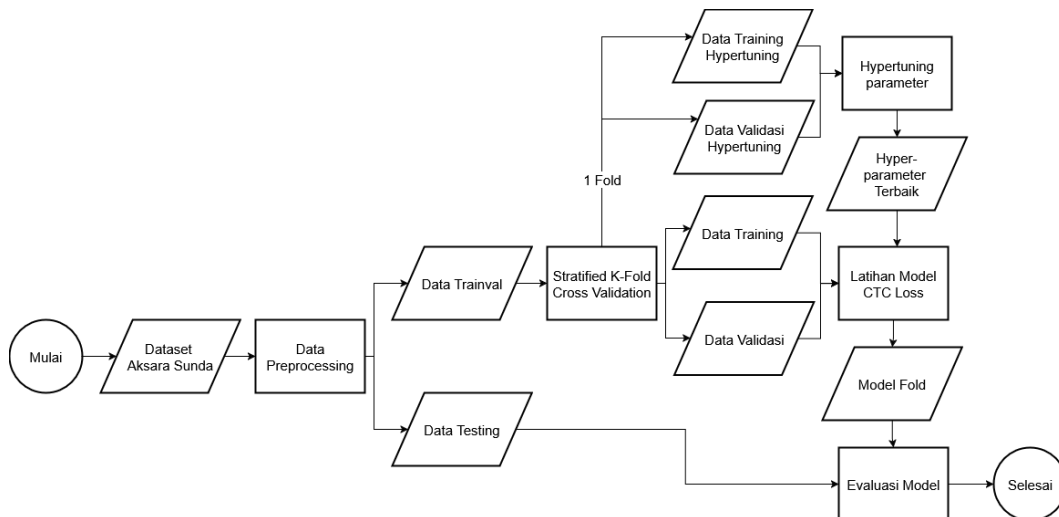
Model kedua yang diimplementasikan adalah Model Connectionist Temporal Classification (CTC). Pada penelitian ini, implementasi dari model ini dilakukan secara sederhana karena tidak menjadi fokus utama pembahasan. Akibatnya penetapan parameter yang digunakan yaitu *dropout* dan *learning rate* tidak dilakukan secara rinci maupun mendalam. Model CTC ini menggunakan EfficientNetB0 sebagai ekstraktor fitur dari input gambar seperti ditunjukkan oleh gambar 3.4.



Gambar 3.4. Implementasi EfficientNetB0 sebagai model dasar dari 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 sequence* 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.

Pelatihan model CTC kali ini dilakukan menggunakan *hyperparameter* yang sama dengan model ensemble, yaitu *learning rate* dan *dropout*

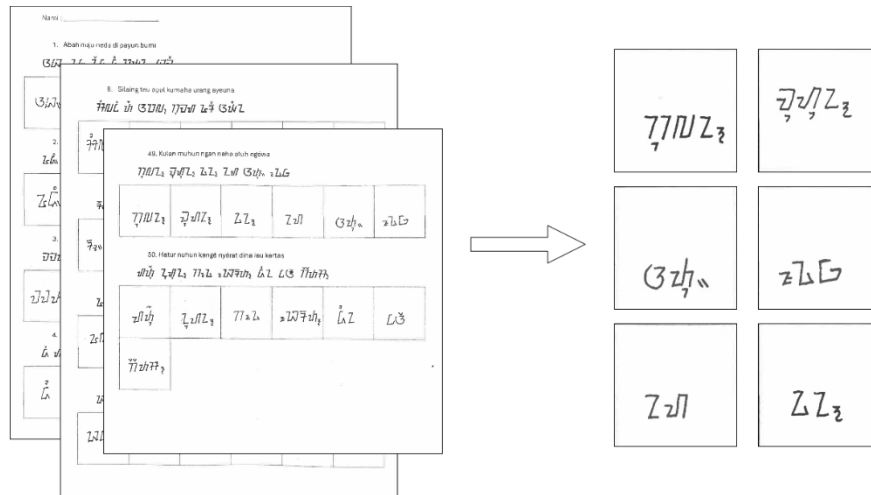


Gambar 3.5. Diagram alir pelatihan model CTC

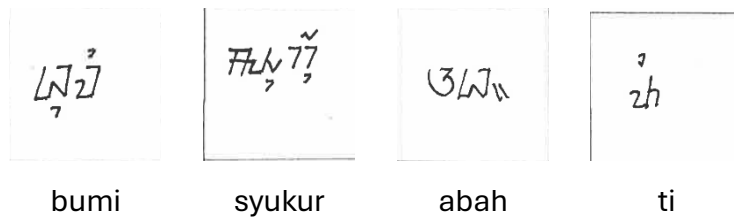
Gambar 3.5 merupakan diagram alir pelatihan model CTC. Dataset yang dipakai melewati *preprocessing* data tambahan dibandingkan model *ensemble* terlebih dahulu yaitu tahap *label encoding* dan *sequence labelling*. Proses *Stratified K-Fold Cross Validation* dilakukan dua kali, yaitu untuk proses *Hypertuning Parameter* dan pelatihan model. Pada proses *Hypertuning Parameter* menggunakan 1 fold, sementara untuk pelatihan model menggunakan 5 fold. *Hypertuning parameter* ini yang digunakan adalah *learning rate* dan *dropout*. Rentang nilai *learning rate* yang dipakai adalah dari 0,001; 0,0001; dan 0,00005, sedangkan rentang nilai *dropout* adalah 0,2; 0,3; dan 0,4[37], [38]. Proses pelatihan model dilakukan setelah mendapatkan *hyperparameter* terbaik. 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[12]. 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. Proses pengumpulan dataset kata ditunjukkan oleh gambar 3.6 sementara contoh dari dataset kata yang dikumpulkan ditunjukkan oleh gambar 3.7.



Gambar 3.6. Proses pengumpulan dataset kata



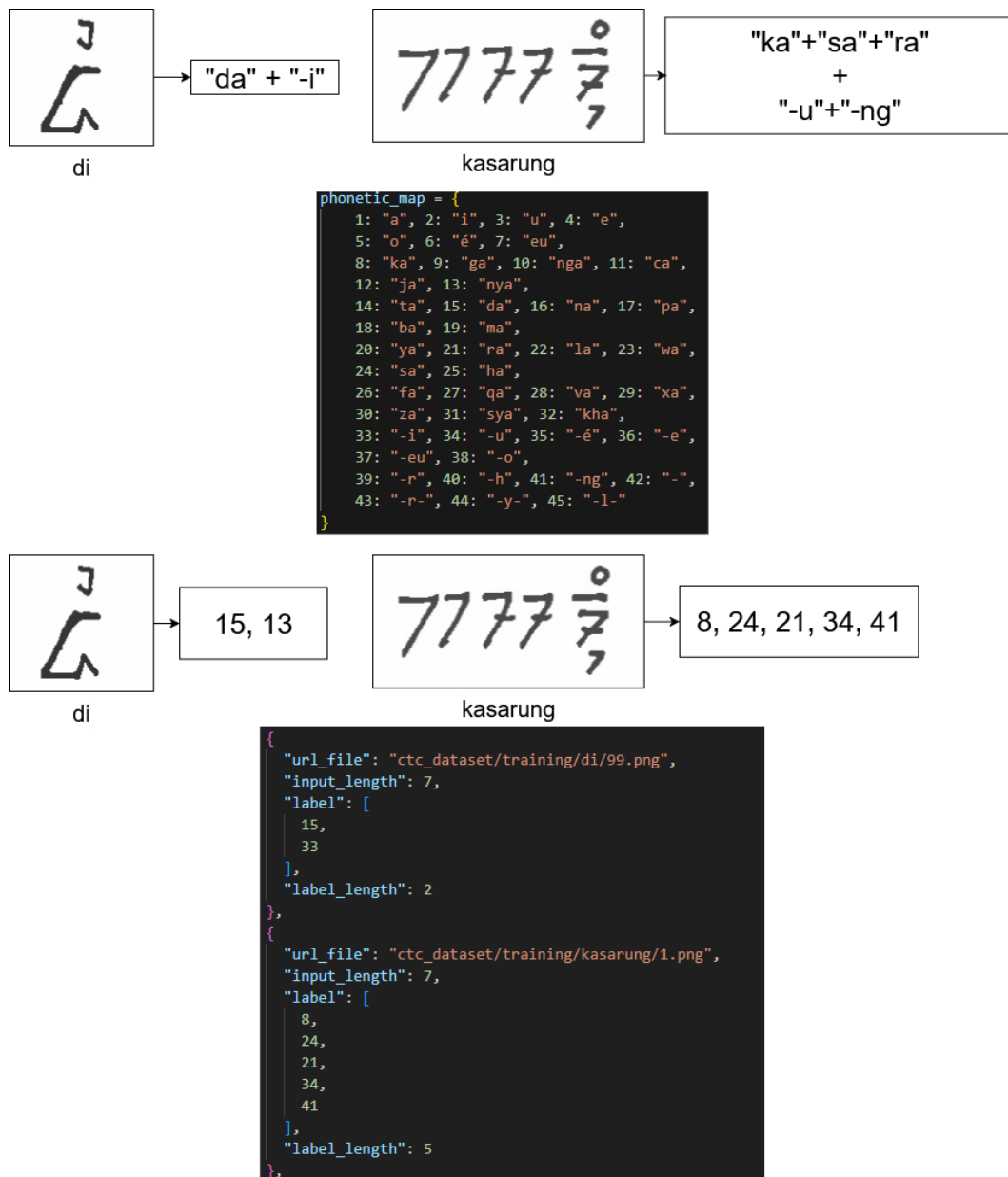
Gambar 3.7. Contoh dataset kata bumi, syukur, abah, dan ti

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. Kalimat dan kata Bahasa Sunda yang dikumpulkan sebagai dataset baru dipilih secara acak, agar mensimulasikan penggunaan Bahasa Sunda dalam

situasi nyata. 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. Kedua proses tersebut ditunjukkan oleh gambar 3.8.



Gambar 3.8. Label encoding dan sequence labelling

Gambar 3.8 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 *hypertuning parameter* sehingga pelatihan model. Masing-masing model terbaik akan dipilih menggunakan metrik *F1-score* sebagai dasar model *ensemble*. Setelah itu, model *ensemble* akan dievaluasi untuk pemilihan *ensemble weight* sehingga menghasilkan model *ensemble* akhir.

4.1.1. Model Karakter dan Model Kata

Hyperparameter yang digunakan pada proses *hypertuning parameter* 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 menampilkan hasil dari tahap *hypertuning parameter* pada masing-masing model.

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

Percobaan	Dropout		Learning rate		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 *hypertuning parameter* 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. Setelah diketahui *hyperparameter* terbaik untuk masing-masing model, maka berlanjut ke pelatihan menyeluruh dengan konfigurasi masing-masing model terlihat pada tabel 4.2.

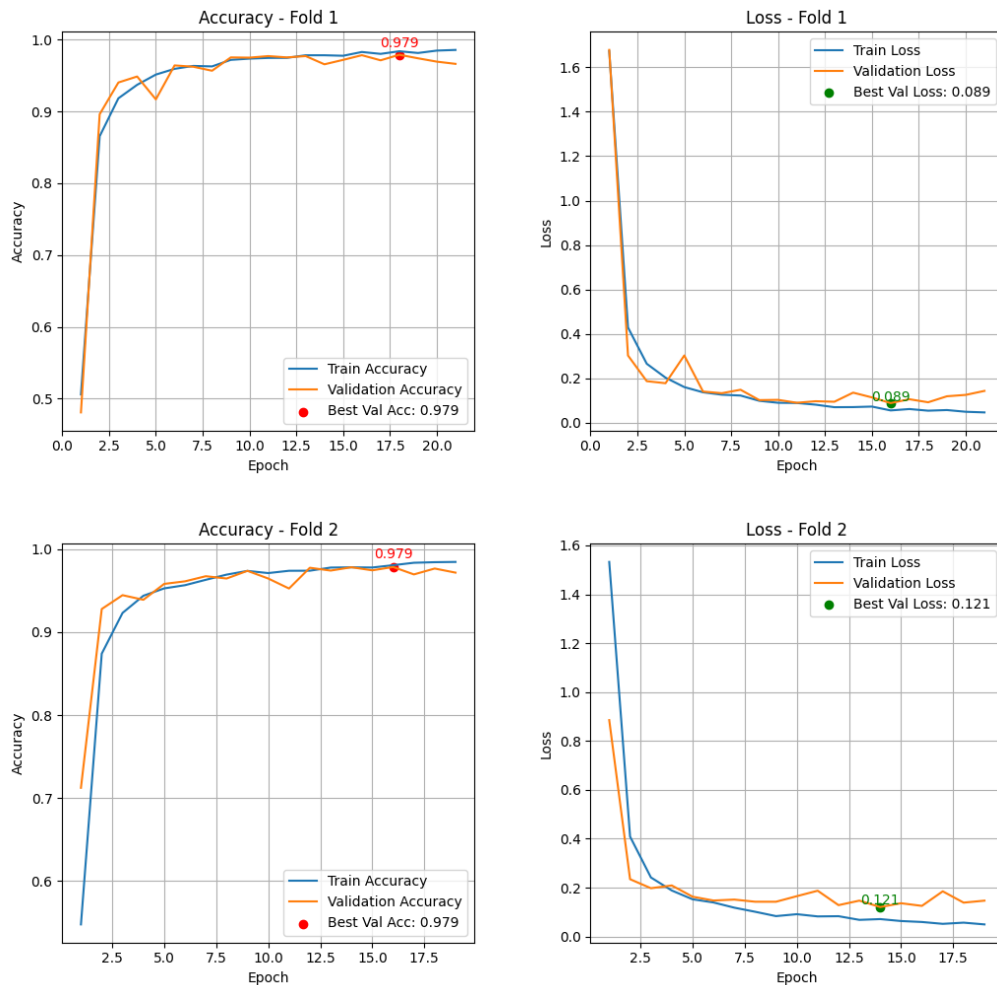
Tabel 4.2. Konfigurasi EfficientNetB0 sebagai base model dari model *ensemble*

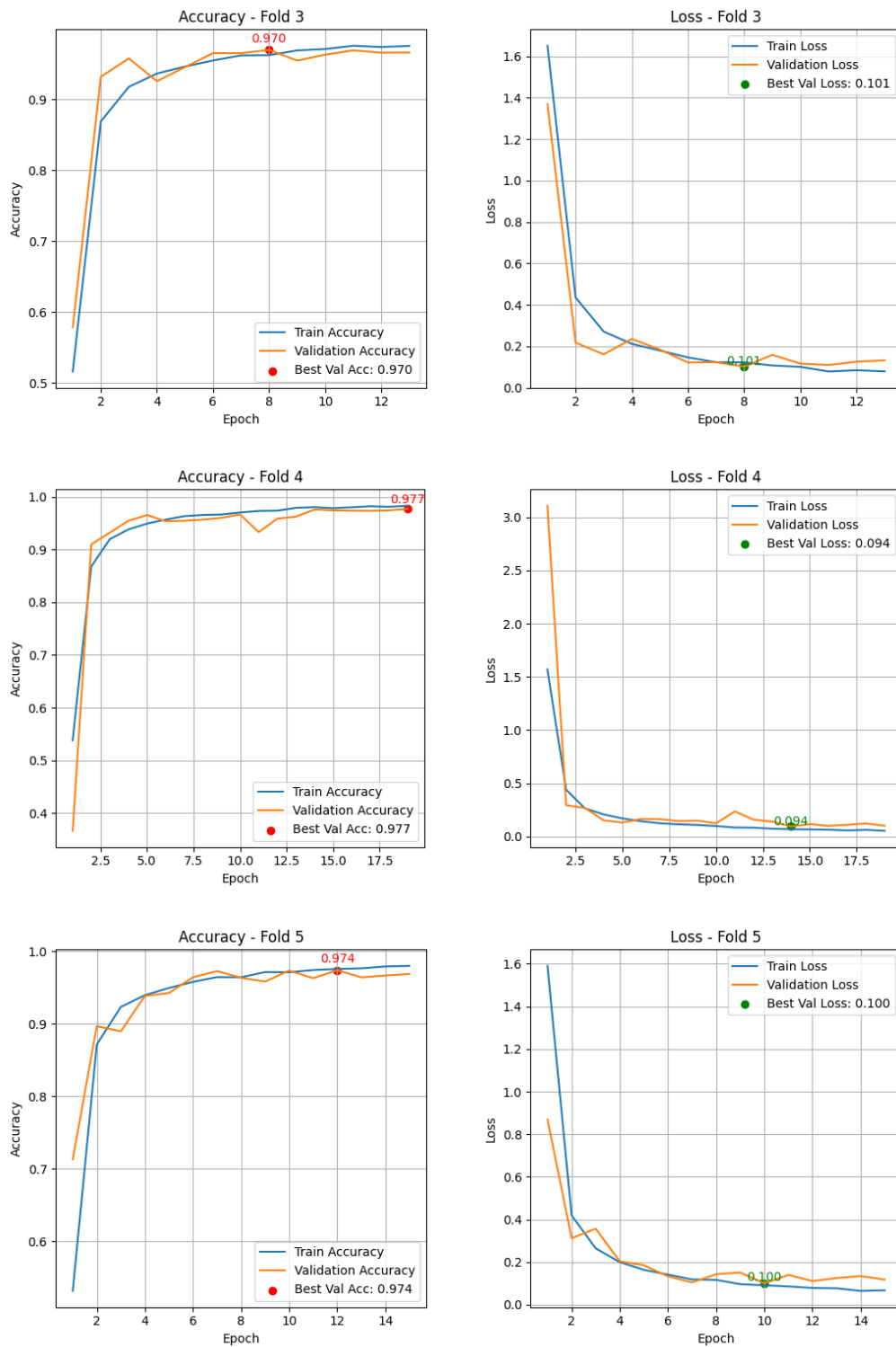
Hyperparameter	Model Karakter	Model Kata
Input size	224 x 224 pixel	224 x 224 pixel
Batch size	32	32
<i>Epochs</i>	30	30
Learning rate	0,001	0,001
Dropout	0,4	0,2
Unfreeze layer	Layer 163 – 240	Layer 163 – 240
Stratified <i>K-fold cross validation</i>	Fold = 5 fold	Fold = 5 fold

Pelatihan dilakukan dengan maksimal 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. Evaluasi akurasi dan *loss* dari pelatihan model karakter diperlihatkan pada gambar 4.1.

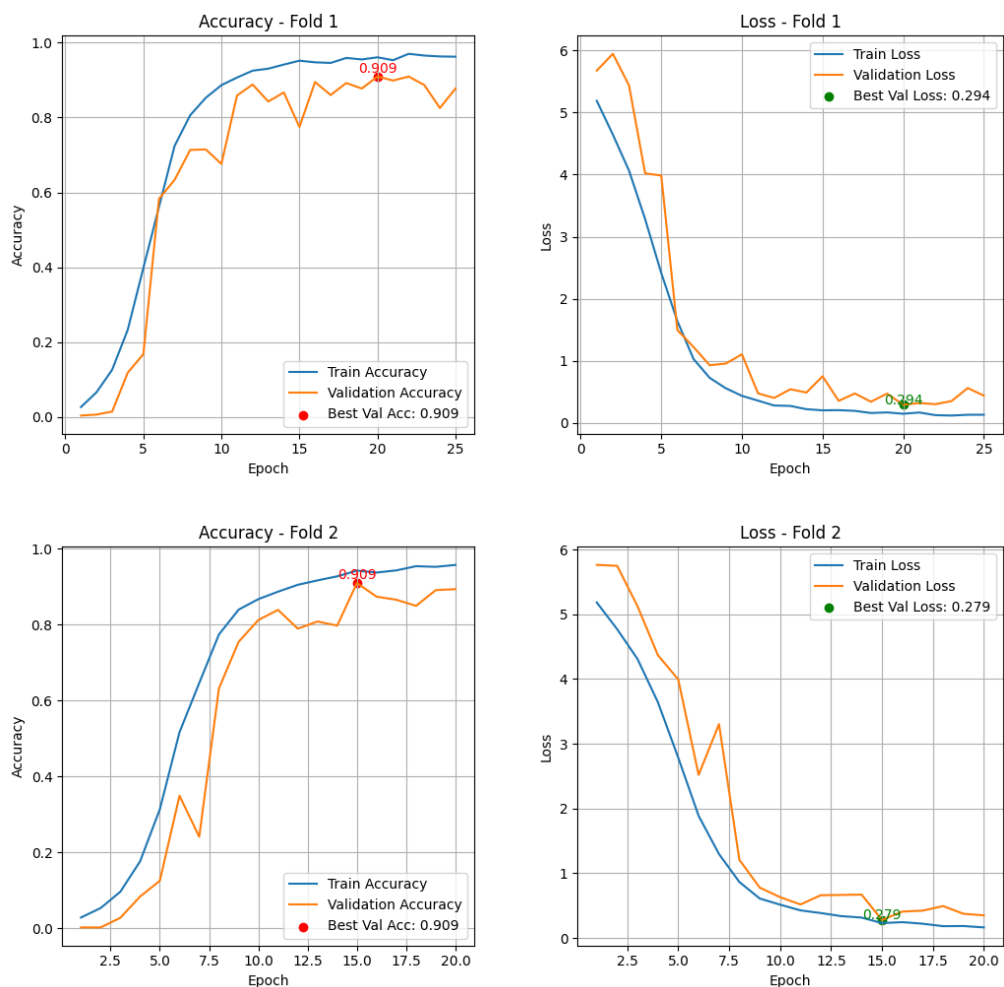


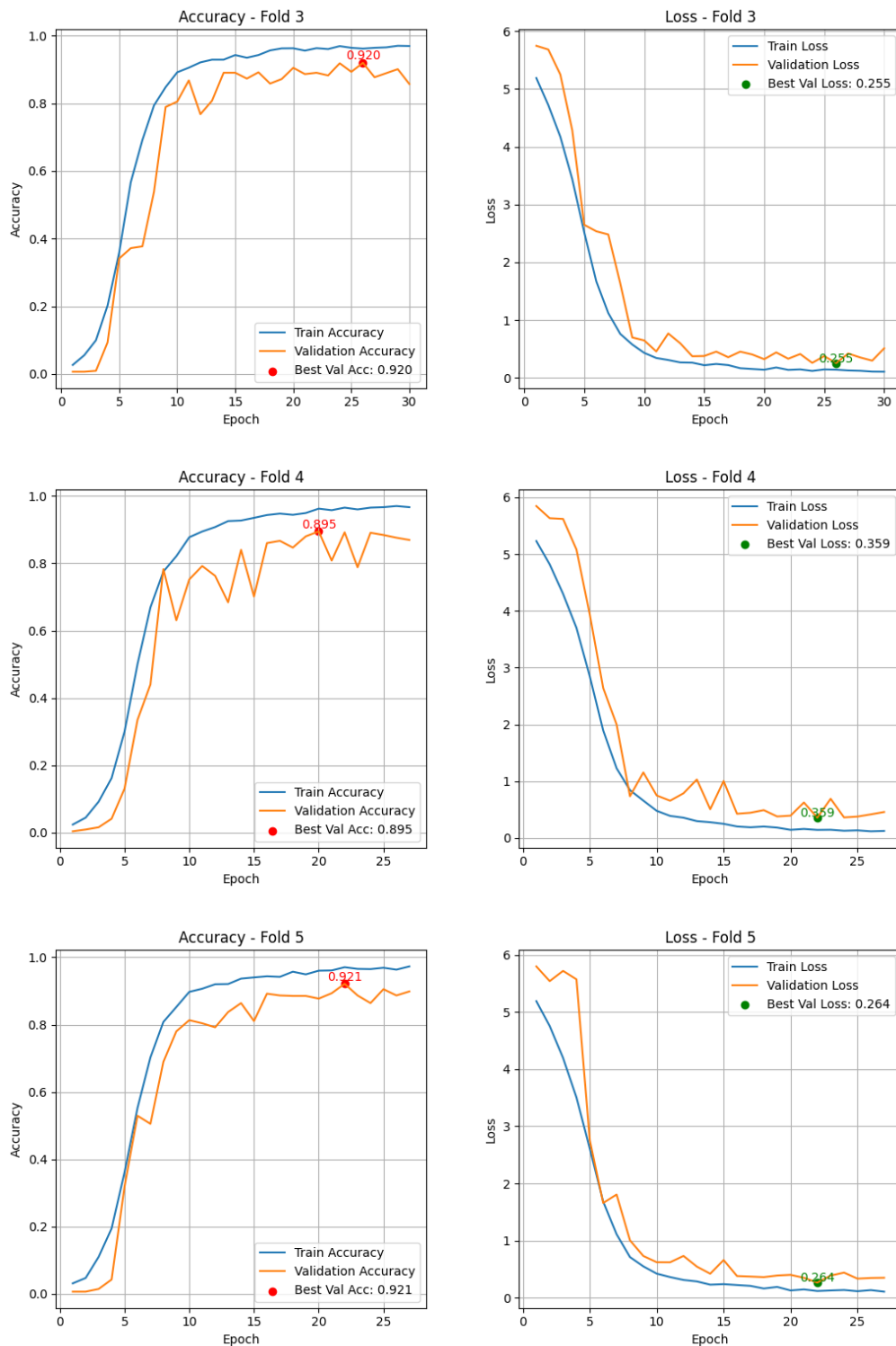


Gambar 4.1. Grafik akurasi dan loss dari pelatihan model karakter

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. Sementara itu, grafik evaluasi akurasi dan *loss* untuk pelatihan model kata diperlihatkan oleh gambar 4.2.





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 seperti perbandingan *epochs*, akurasi dan *loss* pada pelatihan setiap fold dirangkum seperti terlihat di tabel 4.3.

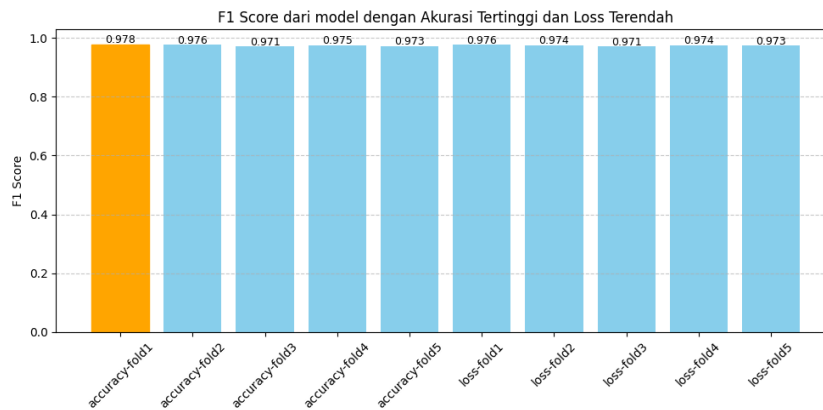
Tabel 4.3. Perbandingan *epochs*, akurasi, dan *loss* pada pelatihan setiap fold untuk kedua model

		Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Model	<i>Epochs</i>	21	19	13	19	15
Karakter	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	<i>Epochs</i>	25	20	30	27	27
Kata	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 menunjukkan perbedaan kompleksitas dataset digunakan oleh kedua model. Dataset kata memiliki 220 kelas dengan jumlah rata-rata per-kelas memiliki 15 data. Hal ini menunjukkan level kompleksitas yang berbeda jika dibandingkan dataset karakter yang hanya memiliki 32 kelas dengan rata-rata per-kelas memiliki 50 data.

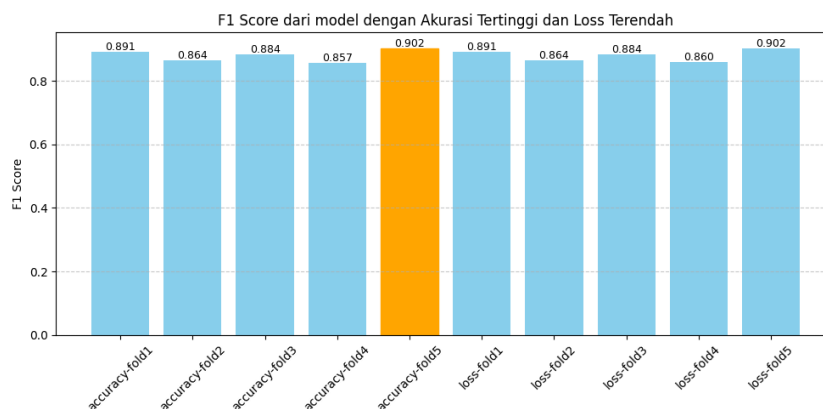
Pemilihan model terbaik yang akan digunakan sebagai model *ensemble* dilakukan melalui perbandingan nilai *F1-score* yang dihasilkan setiap *fold*.

Dikarenakan adanya fungsi *early stopping* pada proses pelatihan model untuk metrik akurasi tertinggi dan *loss* terendah, maka setiap *fold* dari pelatihan menghasilkan 2 model yang bisa dievaluasi. Perbandingan nilai *F1-score* untuk pemilihan model karakter ditampilkan oleh gambar 4.3 dan perbandingan nilai *F1-score* untuk pemilihan model kata ditampilkan oleh gambar 4.4.



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.




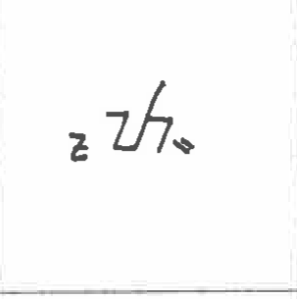
Tabel 4.4. Perbandingan hasil evaluasi nilai α sebagai nilai *ensemble weight*

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</i> (weighted)	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 rangkén 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

Pada proses pelatihan model CTC, melalui proses *hypertuning parameter* dengan parameter *dropout* dan *learning rate*. 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 0,0005. Tabel 4.6 menampilkan hasil dari tahap pada masing-masing model.

Tabel 4.6. Hasil pengaturan *hyperparameter* untuk model CTC

Percobaan	<i>Dropout</i>	<i>Learning rate</i>	<i>Loss</i>
1	0,2	0,001	4,820
2	0,3	0,001	4,682
3	0,4	0,001	4,616
4	0,2	0,0001	2,094
5	0,3	0,0001	2,211
6	0,4	0,0001	2,422
7	0,2	0,0005	3,661
8	0,3	0,0005	3,466

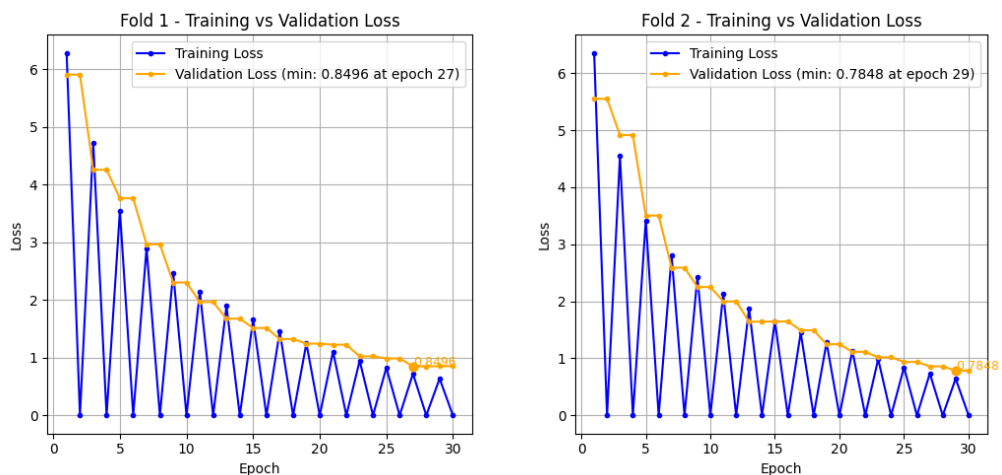
Tabel 4.6 memperlihatkan hasil dari proses *hypertuning parameter* selama 5 *epochs* pada model CTC. Parameter terbaik adalah *dropout* dengan nilai 0,2 dan *learning rate* dengan nilai 0,0001 dengan nilai loss terendah sebesar 2,094. Hasil tersebut mengindikasikan bahwa model CTC membutuhkan nilai *learning rate* yang cukup stabil dan membutuhkan *dropout* terkecil untuk menghindari underfitting. Setelah mengetahui nilai parameter terbaik dilakukan pelatihan model CTC menggunakan konfigurasi terlihat pada tabel 4.7.

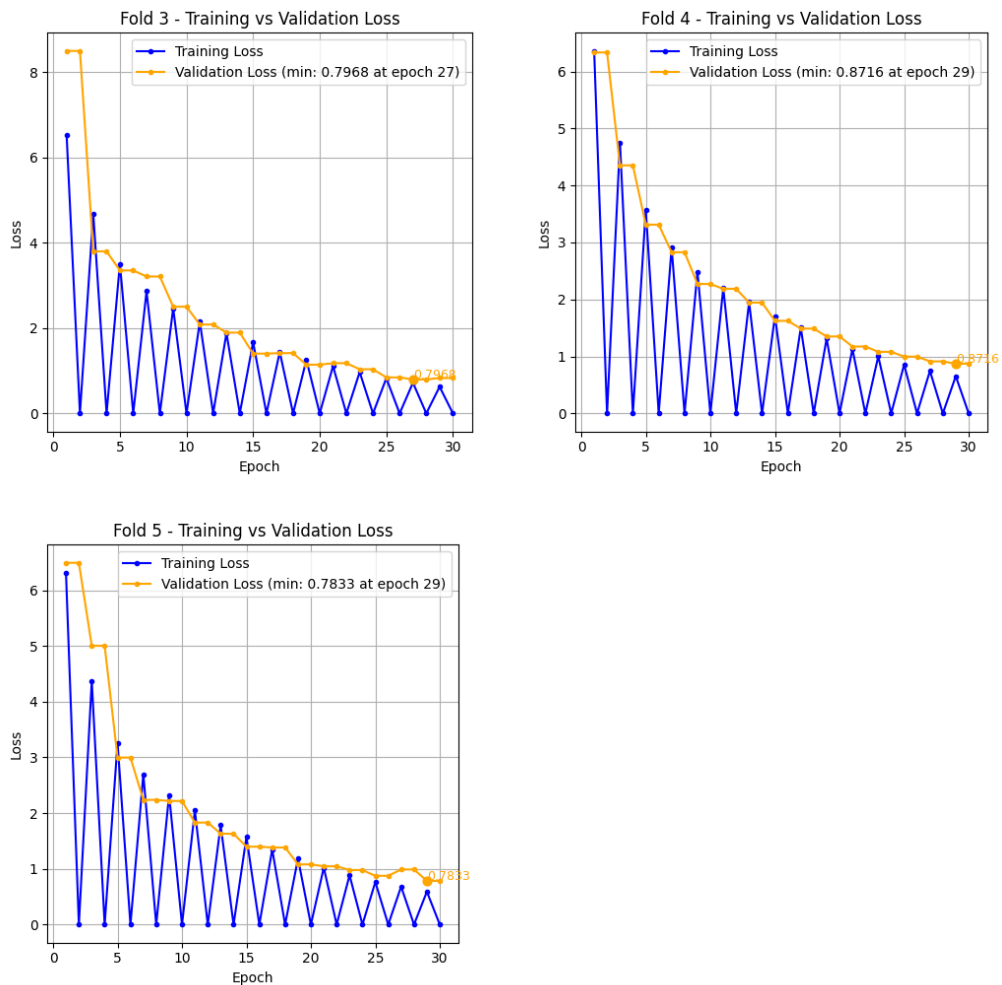
Tabel 4.7. Konfigurasi model CTC yang dilatih

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

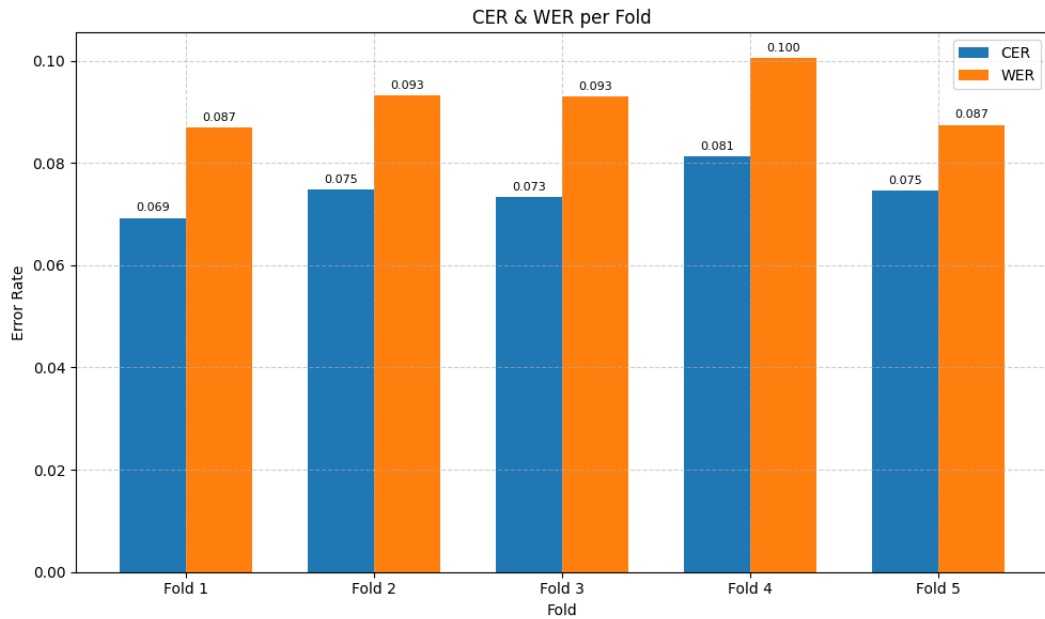
Konfigurasi model CTC diatas kurang lebih sama dengan konfigurasi pelatihan model *ensemble*. Perbedaan utama ada pada evaluasi performa model CTC menggunakan fungsi *CTC loss* dibandingkan *confusion matrix* untuk model *ensemble*. Skenario pelatihan model CTC menggunakan metode *Stratified K-fold Cross 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-5 dengan nilai 0,783. Sedangkan CTC *loss* tertinggi didapatkan pada fold ke-4 dengan nilai 0,871. Hal ini menunjukkan performa pelatihan yang cukup merata di semua fold. 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-4 dengan nilai CER sebesar 0,081 dan nilai WER sebesar 0,100. Model CTC terbaik dihasilkan oleh fold ke-3 dengan nilai CER sebesar 0,069 dan nilai WER sebesar 0,087. Rata-rata nilai CER adalah sebesar 0,074 dan rata-rata nilai WER adalah sebesar 0,092.

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.

Perbedaan nilai *dropout* pada kedua model tersebut terjadi karena adanya perbedaan dataset yang digunakan. Dataset karakter memiliki distribusi data yang lebih merata dan relatif mencukupi pada 32 kelas yang ada. Sebaliknya, dataset kata terdiri dari 220 kelas dengan rata-rata sekitar 15 data per kelas. Oleh karena itu, model karakter memiliki *dropout* yang lebih tinggi untuk mencegah *overfitting*, sedangkan model kata memiliki *dropout* lebih rendah agar model dapat memanfaatkan seluruh informasi yang tersedia untuk mengimbangi keterbatasan jumlah data.

Kedua model tersebut 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. Selisih dari nilai *F1-score*, presisi dan sensitifitas ke nilai akurasi disebabkan oleh ketimpangan antara dataset karakter dengan dataset kata. Dengan penambahan data secara merata dan jumlah data yang mencukupi di setiap kelas dataset kata, diharapkan nilai *F1-score*, nilai presisi dan nilai sensitifitas akan meningkat menjadi lebih baik.

Pada skenario CTC loss, proses *hypertuning parameter* kombinasi parameter yang menghasilkan *loss* terendah adalah *dropout* sebesar 0,2 dan *learning rate* sebesar 0,0001. Evaluasi model CTC menunjukkan performa yang baik dengan rata-rata nilai *Character Error Rate* (CER) adalah sebesar 0,074 dan rata-rata nilai *Word Error Rate* (WER) adalah sebesar 0,092. Model CTC yang

terbaik mencapai nilai CER sebesar 0,069 dan nilai WER sebesar 0,087. Sedangkan Model CTC dengan performa terburuk menghasilkan nilai CER sebesar 0,081 dan nilai WER sebesar 0,100. Nilai CER dan WER yang relatif rendah ini besar kemungkinan karena evaluasi dilakukan menggunakan dataset pada tingkat karakter dan kata, bukan pada tingkat kalimat. Untuk memperoleh hasil evaluasi yang lebih representatif, disarankan untuk menggunakan dataset tulisan tangan aksara sunda berbasis kalimat.

BAB 5 KESIMPULAN DAN SARAN

5.1. Kesimpulan

Penelitian ini bertujuan untuk mengimplementasikan Optical Character Recognition (OCR) tingkat kata 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 terbaik berhasil mencapai nilai *Character Error Rate* (CER) sebesar 0,069 dan nilai *Word Error Rate* (WER) sebesar 0,087. 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 dengan penelitian lebih mendetail.
3. Pengujian metode CTC *loss* dilakukan dengan dataset berbasis kalimat agar evaluasi model lebih representatif.
4. 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. Accessed: Nov. 25, 2024. [Online]. Available: 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, and D. Fiman Ihsan, “Klasifikasi Aksara Sunda Dengan Metode Machine Learning Klasik,” 2024.
- [4] M. A. Prameswari, M. Dwi Sulistiyo, and A. F. Ihsan, “Classification of Handwritten Sundanese Script via Transfer Learning on CNN-Based Architectures,” in *Proceedings - 2023 3rd International Conference on Electronic and Electrical Engineering and Intelligent System: Responsible Technology for Sustainable Humanity, ICE3IS 2023*, 2023, pp. 401–406. doi: 10.1109/ICE3IS59323.2023.10335382.
- [5] R. R. Akram, M. Dwi Sulistiyo, A. Firman Ihsan, P. Eko Yunanto, D. Richasdy, and M. Arzaki, “Exploring ReLU Activation Functions in CNN for Handwritten Sundanese Script Recognition,” in *2024 International Conference on Data Science and Its Applications, ICoDSA 2024*, 2024, pp. 92–98. doi: 10.1109/ICoDSA62899.2024.10652217.
- [6] Y. A. Gerhana, M. Farid Padilah, and A. Rialdy Atmadja, “Comparison of Template Matching Algorithm and Feature Extraction Algorithm in Sundanese Script Transliteration Application using Optical Character Recognition,” *Jurnal Online Informatika*, vol. 5, no. 1, pp. 73–80, Jul. 2020, doi: 10.15575/JOIN.V5I1.580.
- [7] P. Batra, N. Phalnikar, D. Kurmi, J. Tembhurne, P. Sahare, and 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, pp. 447–455, Jan. 2024, doi: 10.1007/S41870-023-01610-2/TABLES/6.
- [8] I. Baidillah *et al.*, “Direktori Aksara Sunda untuk Unicode,” 2008.
- [9] aksarasunda, “Aksara Sunda Computer Vision Project,” Roboflow. Accessed: Dec. 02, 2024. [Online]. Available: <https://universe.roboflow.com/aksarasunda/aksara-sunda-eayhq>
- [10] Teknik Informatika, “Aksara Sunda Computer Vision Project,” Roboflow. Accessed: Dec. 05, 2024. [Online]. Available: <https://universe.roboflow.com/teknik-informatika/aksara-sunda-4lcpy>
- [11] A. Angelina, “Aksara Sunda Computer Vision Project,” Roboflow. Accessed: Dec. 05, 2024. [Online]. Available: <https://universe.roboflow.com/agatha-angelina-hnexv/aksara-sunda-xvbwp>







































































- [12] A. F. Ihsan, "Indonesian Local Script Characters," 2024, *Mendeley Data*. doi: 10.17632/VFJ32BPJSF.1.
- [13] S. INGOLE, "An Extensive Guide to Convolution Neural Network- (2023)," Medium. Accessed: Dec. 08, 2024. [Online]. Available: <https://medium.com/@singole/an-extensive-guide-to-convolution-neural-network-2023-84872b16bd78>
- [14] V. Pomazan, I. Tvoroshenko, and V. Gorokhovatskyi, "Handwritten character recognition models based on convolutional neural networks," 2023, Accessed: Dec. 08, 2024. [Online]. Available: <https://openarchive.nure.ua/entities/publication/09998e00-8e88-4642-b299-791755e6ea08>
- [15] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artif Intell Rev*, vol. 57, no. 4, p. 99, Mar. 2024, doi: 10.1007/s10462-024-10721-6.
- [16] Keras Team, "Keras Applications." Accessed: Nov. 28, 2024. [Online]. Available: <https://keras.io/api/applications/>
- [17] J. Riza, S. Barman, S. Ridita, Z. Mahmud, and A. Bhattacharya, "PhytoCare : A hybrid approach for identifying Rice, Potato and Corn diseases," 2024.
- [18] Y. Jie *et al.*, "Combined Multi-Layer Feature Fusion and Edge Detection Method for Distributed Photovoltaic Power Station Identification," *Energies (Basel)*, vol. 13, p. 6742, Jan. 2020, doi: 10.3390/en13246742.
- [19] R. Dodda, S. B. Reddy, A. C. Naik, and 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, pp. 107–11, Jun. 2024, doi: 10.32377/CVRJST2617.
- [20] X. Liu and W. Wang, "Deep Time Series Forecasting Models: A Comprehensive Survey," *Mathematics 2024, Vol. 12, Page 1504*, vol. 12, no. 10, p. 1504, May 2024, doi: 10.3390/MATH12101504.
- [21] S. M. Vatanchi, H. Etemadfard, M. F. Maghrebi, and 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.
- [22] B. Imane, · Ammour Alae, · Khaissidi Ghizlane, and M. Mrabti, "Enhancing Arabic handwritten word recognition: a CNN-BiLSTM-CTC architecture with attention mechanism and adaptive augmentation," *Discover Applied Sciences*, vol. 7, p. 460, 123AD, doi: 10.1007/s42452-025-06952-z.
- [23] R. Najam and S. Faizullah, "Analysis of Recent Deep Learning Techniques for Arabic Handwritten-Text OCR and Post-OCR















































Correction,” Jul. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/app13137568.

- [24] P. Verma and 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.
- [25] X. H. Le, D. H. Nguyen, S. Jung, M. Yeon, and G. Lee, “Comparison of Deep Learning Techniques for River Streamflow Forecasting,” *IEEE Access*, vol. 9, pp. 71805–71820, 2021, doi: 10.1109/ACCESS.2021.3077703.
- [26] L. Mosbah, I. Moalla, T. M. Hamdani, B. Neji, T. Beyrouthy, and A. M. Alimi, “ADOCRNet: A Deep Learning OCR for Arabic Documents Recognition,” *IEEE Access*, vol. 12, pp. 55620–55631, 2024, doi: 10.1109/ACCESS.2024.3379530.
- [27] D. Vaibhav Khatavkar, D. Makarand Velankar, and S. Petkar, “Segmentation-free Connectionist Temporal Classification loss based OCR Model for Text Captcha Classification,” Feb. 2024, Accessed: Jun. 18, 2025. [Online]. Available: <https://arxiv.org/pdf/2402.05417>
- [28] K. M. M. Chit and L. L. Lin, “Exploring CTC Based End-To-End Techniques for Myanmar Speech Recognition,” 2021, doi: 10.1007/978-3-030-68154-8_87.
- [29] S. Szeghalmy and 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, p. 2333, Feb. 2023, doi: 10.3390/S23042333.
- [30] 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.
- [31] D. A. Neu, J. Lahann, and P. Fettke, “A systematic literature review on state-of-the-art deep learning methods for process prediction,” *Artif Intell Rev*, vol. 55, no. 2, pp. 801–827, Feb. 2022, doi: 10.1007/S10462-021-09960-8/FIGURES/6.
- [32] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10691–10700, May 2019, Accessed: Dec. 08, 2024. [Online]. Available: <https://arxiv.org/abs/1905.11946v5>
- [33] G. S. Permana, S. Sa'idah, and R. Purnamasari, “Deteksi Aksara Sunda Menggunakan Metode CNN Arsitektur EfficientNet,” 2023.

- [34] M. Tan and Q. V Le, "EfficientNetV2: Smaller Models and Faster Training," 2021, Accessed: Aug. 13, 2025. [Online]. Available: <https://github.com/google/>
- [35] A. R. Philip, "Fine-Tuning Depth Analysis: Identifying the Sweet Spot for Maximum Accuracy in CNNs," *International Journal of Scientific Research in Computer Science and Engineering*, vol. 13, no. 3, pp. 48–63, 2025, doi: 10.26438/ijsrcse.v13i3.703.
- [36] H. Ali, N. Shifa, R. Benlamri, A. A. Farooque, and R. Yaqub, "A fine tuned EfficientNet-B0 convolutional neural network for accurate and efficient classification of apple leaf diseases," *Sci Rep*, vol. 15, no. 1, pp. 1–26, Dec. 2025, doi: 10.1038/S41598-025-04479-2;SUBJMETA=114,1305,1564,631;KWRD=IMAGE+PROCESSING,MACHINE+LEARNING.
- [37] Y. Li, D. Chen, T. Tang, and X. Shen, "HTR-VT: Handwritten text recognition with vision transformer," *Pattern Recognit*, vol. 158, p. 110967, Feb. 2025, doi: 10.1016/J.PATCOG.2024.110967.
- [38] A. Shetty and S. Sharma, "Ensemble deep learning model for optical character recognition," *Multimed Tools Appl*, vol. 83, no. 4, pp. 11411–11431, Jan. 2023, doi: 10.1007/S11042-023-16018-0.

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	 Kzur	 Kror	 Krér	 Krer	 Kreur
 Krah	 Krih	 Kruh	 Kroh	 Kréh	 Kreh	 Kreuh
 Kya	 Kyi	 Kyu	 Kyo	 Kyé	 Kye	 Kyeu
 Kyang	 Kying	 Kyung	 Kyong	 Kyéng	 Kyeng	 Kyeung
 Kyar	 Kyir	 Kyur	 Kyor	 Kyér	 Kyer	 Kyeur
 Kyah	 Kyih	 Kyuh	 Kyoh	 Kyéh	 Kye	 Kyeuh
 Ky	 Kl	 Kr	 K			

Tabel 6.1. Kemungkinan kombinasi ngalagena dengan rarangkén untuk satu karakter ngalagena