Jurnal Teknik Informatika dan Teknologi Informasi
Volume 6, Nomor 2, Agustus 2026
e-ISSN: 2827-9387, p-ISSN: 2827-9379, Hal. 643-661
DOI: https://doi.org/10.55606/jutiti.v6i2.7740
Tersedia:https://journalshub.org/index.php/jutiti

Pengembangan Aplikasi Android Pendeteksi Disleksia pada Anak Usia Dini Berbasis *Machine Learning*

I Putu Dika Dharma Brasika[1*], Adie Wahyudi Oktavia Gama[2]
[1-2]Teknologi Informasi, Fakultas Teknik dan Informatika, Universitas Pendidikan Nasional, Indonesia
**Penulis Korespondensi: brasikadika@gmail.com*

Abstract*. This study aims to develop an Android-based application capable of early detection of dyslexia in elementary school children using machine learning. The background of this research is the low level of early detection of dyslexia, which often leads to delayed intervention for affected children. The method used in this study is a deep learning approach utilizing the Convolutional Neural Network (CNN) algorithm to analyze children's handwriting patterns. The dataset used consists of a combination of public datasets and field data, which were processed through preprocessing, model training, and evaluation using a confusion matrix. Two models were tested in this research, namely MobileNetV2 and EfficientNetB0, where EfficientNetB0 achieved the best performance with an accuracy of 86%. The selected model was then implemented into an Android application using TensorFlow Lite. System evaluation was conducted through model performance testing, blackbox testing, and User Acceptance Test (UAT), which indicated that the application functions properly and is well accepted by users. The results show that the developed application can serve as an effective tool for early dyslexia detection and support inclusive education practices.*

Keyword: *Android; CNN; Dyslexia; Early Detection*; *Machine Learning.*

Abstrak.. Penelitian ini bertujuan untuk mengembangkan aplikasi Android yang mampu mendeteksi dini potensi disleksia pada anak usia sekolah dasar berbasis *machine learning*. Latar belakang penelitian ini didasari oleh masih rendahnya tingkat deteksi dini gangguan disleksia yang menyebabkan keterlambatan penanganan pada anak. Metode yang digunakan dalam penelitian ini adalah pendekatan deep learning dengan algoritma Convolutional Neural Network (CNN) untuk menganalisis pola tulisan tangan anak. Dataset yang digunakan berasal dari kombinasi dataset publik dan data lapangan, yang kemudian diproses melalui tahapan preprocessing, pelatihan model, dan evaluasi menggunakan confusion matrix. Dua model diuji dalam penelitian ini, yaitu MobileNetV2 dan EfficientNetB0, di mana EfficientNetB0 menunjukkan performa terbaik dengan akurasi sebesar 86%. Model kemudian diimplementasikan ke dalam aplikasi Android menggunakan TensorFlow Lite. Evaluasi sistem dilakukan melalui pengujian performa model, blackbox testing, dan User Acceptance Test (UAT) yang menunjukkan bahwa aplikasi berjalan dengan baik dan dapat diterima oleh pengguna. Hasil penelitian menunjukkan bahwa aplikasi yang dikembangkan mampu menjadi alat bantu yang efektif dalam mendeteksi dini disleksia serta mendukung penerapan pendidikan inklusif.

Kata Kunci: Android; CNN; Deteksi Dini**;** Disleksia; Pembelajaran Mesin.

1. LATAR BELAKANG

Pendidikan inklusif menjadi bagian penting dalam mewujudkan tujuan *Sustainable Development Goals* (SDGs), khususnya poin keempat yang menekankan pentingnya pendidikan yang berkualitas dan merata bagi semua anak (Lestari, 2024). Namun dalam praktiknya, masih banyak anak di Indonesia yang belum memperoleh layanan pendidikan sesuai kebutuhannya, terutama anak dengan kebutuhan khusus atau gangguan belajar seperti disleksia (Lestari, 2024). Permasalahan yang sering terjadi ialah gangguan belajar ini tidak terdeteksi sejak awal sehingga anak tidak segera mendapatkan penanganan pembelajaran yang tepat. Akibatnya, kesulitan belajar berlanjut dan berpotensi menimbulkan dampak akademik maupun psikologis pada anak dalam jangka panjang. Disleksia sendiri merupakan gangguan belajar spesifik yang ditandai dengan kesulitan membaca, menulis, dan mengenali huruf,

Naskah Masuk: 30 Mei 2026; Revisi: 29 Juni 2026; Diterima: 29 Juli 2026; Terbit: 07 Agustus 2026

meskipun anak memiliki tingkat kecerdasan yang normal (Filasofa, dkk, 2021). Gangguan ini berkaitan dengan proses pemrosesan bahasa pada otak yang menyebabkan anak mengalami kesulitan dalam menghubungkan simbol huruf dengan bunyi atau maknanya. Gejala disleksia umumnya mulai terlihat ketika anak memasuki usia sekolah dasar, yaitu sekitar usia 7–12 tahun, pada saat kemampuan membaca dan menulis mulai berkembang secara intensif. Kondisi ini tidak hanya berdampak pada kemampuan akademik, tetapi juga dapat memengaruhi perkembangan sosial dan emosional anak apabila tidak dikenali sejak dini. Banyak guru di lembaga pendidikan di sekolah dasar maupun pendidikan dasar masih mengandalkan pengamatan umum tanpa alat ukur yang jelas dalam mengenali gejala disleksia, sehingga proses intervensi sering kali terlambat dilakukan (Sumiati, dkk, 2025). Padahal, jika disleksia dikenali sejak awal, pendidik dapat memberikan strategi pembelajaran yang sesuai dengan karakteristik anak dan mencegah munculnya perasaan rendah diri serta kegagalan akademik di kemudian hari (Filasofa, dkk, 2021).

Dalam penelitian ini, fokus subjek diarahkan pada anak usia 7–12 tahun (usia sekolah dasar). Rentang usia ini dipilih karena pada fase tersebut kemampuan literasi (membaca dan menulis) mulai berkembang secara intensif dan menjadi tuntutan utama dalam proses pembelajaran, sehingga gejala disleksia cenderung lebih tampak melalui performa membaca, mengeja, dan menulis anak. Selain itu, pada usia ini guru dan orang tua dapat melakukan pengamatan yang lebih jelas terhadap pola kesalahan yang berulang serta hambatan literasi yang menetap, sehingga upaya identifikasi awal dan tindak lanjut pembelajaran dapat dilakukan lebih tepat.

Pentingnya deteksi dini disleksia juga ditegaskan dalam penelitian yang menyatakan bahwa sebagian besar guru belum memiliki kompetensi yang memadai untuk mengenali dan menangani anak dengan gangguan belajar secara sistematis (Sumiati, dkk, 2025). Kondisi ini menunjukkan adanya kebutuhan akan solusi yang dapat membantu proses identifikasi awal secara lebih terarah, objektif, dan mudah digunakan. Dalam konteks ini, perkembangan teknologi informasi dan kecerdasan buatan (*Artificial Intelligence*) berperan besar dalam menghadirkan solusi yang lebih efektif dan efisien untuk membantu proses identifikasi gangguan belajar pada anak. Pemanfaatan *machine learning* mulai banyak dikembangkan dalam bidang pendidikan untuk mengenali pola-pola gangguan belajar secara otomatis. Salah satu penelitian menunjukkan bahwa metode Naïve Bayes dapat digunakan untuk mengklasifikasikan jenis disleksia dengan hasil diagnosis yang cepat dan akurat (Lestari, dkk, 2024). Hal ini menunjukkan bahwa sistem berbasis kecerdasan buatan mampu meniru proses analisis pakar dan berpotensi menjadi alternatif pendukung yang lebih terjangkau bagi guru

dan orang tua. Selain itu, peran kecerdasan buatan semakin luas dalam mendukung proses pendidikan inklusif. Teknologi AI dapat menganalisis data akademik, perilaku belajar, hingga pola interaksi digital anak untuk mendeteksi keterlambatan belajar sejak dini (Dirgayunita, dkk, 2022). Sistem seperti ini memungkinkan intervensi yang lebih adaptif dan personal sesuai kebutuhan anak. Namun, penerapan AI dalam pendidikan tetap memerlukan kesiapan infrastruktur serta peningkatan literasi digital bagi para pendidik agar sistem dapat dimanfaatkan secara optimal (Oktamarina, dkk, 2022). Penelitian juga menegaskan bahwa teknologi adaptif dapat meningkatkan *well-being* anak berkebutuhan khusus di sekolah. Penggunaan sistem berbasis *Support Vector Machine* (SVM) dan *Convolutional Neural Network* (CNN) terbukti membantu anak disleksia dalam proses belajar membaca melalui pendekatan yang lebih sesuai dengan kemampuan mereka (Oktamarina, dkk, 2022). Meski demikian, pendidikan inklusif tidak hanya berkaitan dengan teknologi, tetapi juga lingkungan belajar yang menghargai keberagaman dan kesiapan pendidik dalam merespons perbedaan (Lestari, 2024).

Untuk mendukung upaya tersebut, identifikasi awal disleksia perlu dikombinasikan dengan pendekatan pembelajaran yang kreatif dan interaktif. Selain itu, pemilihan platform aplikasi juga menjadi faktor penting dalam memastikan kemudahan akses bagi pengguna. Perkembangan penggunaan smartphone di Indonesia menunjukkan peningkatan yang sangat signifikan dalam beberapa tahun terakhir. Laporan DataReportal menyebutkan bahwa mayoritas pengguna smartphone di Indonesia menggunakan sistem operasi Android karena ketersediaan perangkat yang lebih terjangkau serta tingkat aksesibilitas yang tinggi di berbagai lapisan masyarakat. Kondisi ini menjadikan platform Android sebagai media yang potensial untuk pengembangan aplikasi edukasi yang dapat digunakan secara luas oleh guru maupun orang tua dalam membantu proses identifikasi awal gangguan belajar pada anak. agar anak tidak merasa tertekan. Anak dengan disleksia cenderung membutuhkan pendekatan yang tidak monoton, misalnya permainan edukatif berbasis visual dan auditori. Strategi pembelajaran berbasis *game* dapat menjadi sarana melatih kemampuan bahasa dan konsentrasi dengan cara yang lebih menyenangkan. Penelitian menunjukkan bahwa anak disleksia memiliki respons yang baik terhadap pendekatan multisensori karena membantu anak mengenali huruf dan bunyi melalui penglihatan, pendengaran, serta gerakan tubuh (Filasofa, dkk, 2021).

Salah satu bentuk penerapan teknologi pembelajaran yang efektif adalah pengembangan aplikasi game edukasi berbasis Android. Penelitian menunjukkan bahwa game edukatif dengan pendekatan interaktif dapat meningkatkan minat belajar anak sekolah dasar secara signifikan (Alqindi, dkk, 2024). Aplikasi yang dikembangkan menggunakan metode

Rapid Application Development (RAD) memungkinkan proses pembuatan cepat dengan melibatkan pengguna dalam setiap tahap perancangan. Melalui elemen kuis dan permainan, aplikasi tersebut mendorong anak untuk belajar sambil bermain dan memperkuat pemahaman konsep matematika. Konsep serupa dapat diadaptasi untuk membangun media pendeteksi disleksia, di mana interaksi anak saat bermain dapat dijadikan data pelatihan bagi model machine learning untuk mengidentifikasi pola gejala disleksia secara dini.

Berdasarkan uraian tersebut, dapat disimpulkan bahwa kombinasi antara pendidikan inklusif, teknologi adaptif, dan machine learning memiliki potensi besar dalam mendukung deteksi dini disleksia pada anak usia sekolah dasar. Pengembangan aplikasi Android pendeteksi disleksia berbasis machine learning menggunakan metode Convolutional Neural Network (CNN), menjadi langkah inovatif untuk membantu guru dan orang tua mengenali gejala disleksia lebih cepat, sekaligus menyediakan media edukasi informatif. Dengan memadukan unsur edukasi, dan analisis data cerdas, diharapkan aplikasi ini dapat menjadi solusi nyata dalam mewujudkan pendidikan yang lebih inklusif, dan efektif.

2. METODE PENELITIAN

Penelitian ini mengembangkan aplikasi Android berbasis *Convolutional Neural* Network (CNN) untuk mendeteksi dini gejala disleksia melalui analisis tulisan tangan anak. Data diperoleh dari *Dyslexia Handwriting Dataset* dan tulisan siswa SDN 4 Panjer, Denpasar, kemudian diproses melalui tahap *preprocessing*, pelatihan model CNN di *Google Colab*, evaluasi, konversi ke *TensorFlow Lite*, dan integrasi ke aplikasi Android. Sistem dirancang agar guru dan orang tua dapat memindai tulisan tangan anak secara praktis untuk memperoleh indikasi awal disleksia tanpa prosedur diagnostik yang kompleks. Temuan ini sejalan dengan penelitian yang menunjukkan bahwa anak disleksia cenderung memiliki tulisan tidak konsisten, huruf tertukar, dan ukuran tulisan tidak proporsional. (Referensi: (Ridho, dkk, 2024), (Adzkia, dkk, 2025), (Haryanti, dkk, 2022), (AlFatrah, dkk, 2025))

Model CNN bekerja dengan mengekstraksi fitur visual tulisan tangan melalui lapisan konvolusi, aktivasi ReLU, pooling, dan *fully connected layer* untuk menghasilkan klasifikasi probabilistik menggunakan fungsi sigmoid. Arsitektur yang digunakan terdiri atas dua lapisan konvolusi berukuran 3×3, dua max pooling 2×2, flatten layer, dense layer, dropout layer, dan output layer sigmoid untuk membedakan tulisan normal dan tulisan dengan indikasi disleksia. Pelatihan model memanfaatkan teknik data *augmentation* seperti rotasi dan *flipping* untuk meningkatkan kemampuan generalisasi terhadap variasi tulisan tangan anak. (Referensi: (Hamdi, dkk, 2025), (Manik, dkk, 2023), (Ardhana, 2021))

Evaluasi dilakukan melalui pengujian performa model, pengujian fungsional aplikasi, dan validasi terhadap diagnosis ahli. Kinerja CNN diukur menggunakan accuracy, precision, recall, dan F1-score berbasis confusion matrix, sedangkan aplikasi diuji dengan *blackbox testing* dan *System Usability Scale* (SUS). Hasil *User Acceptance Test* (UAT) menunjukkan seluruh fitur utama registrasi, login, unggah tulisan, prediksi CNN, hasil deteksi, penyimpanan riwayat, artikel edukasi, dan *logout* berfungsi sesuai harapan dan dinyatakan valid. Dengan demikian, aplikasi dinilai layak digunakan sebagai alat bantu deteksi dini disleksia di lingkungan pendidikan dasar. (Referensi: (Sembiring, dkk, 2025), (Manik, dkk, 2023), (Pramana, 2024), (AlFatrah, dkk, 2025)).

3. HASIL DAN PEMBAHASAN

Pengumpulan Data

Pengumpulan data dilakukan menggunakan *dataset* publik dari Kaggle yang berisi citra tulisan tangan dengan dua kelas, yaitu *dyslexia* dan *normal*, dimana setelah melalui proses seleksi berdasarkan kualitas gambar, kejelasan label, struktur kelas, dan keseimbangan data, diperoleh *dataset* final sebanyak 18.000 gambar yang digunakan dalam penelitian. Pemeriksaan *raw data* menggunakan Google Colab menunjukkan bahwa seluruh data berhasil terbaca dengan baik ke dalam dua kelas utama sehingga dinyatakan layak untuk dilanjutkan ke tahap *pre-processing*, termasuk penyesuaian format gambar, penyeimbangan data, serta pembagian data menjadi data latih, validasi, dan uji.

Pre-processing Data

Tabel 1. Pembagian Dataset Menggunakan Stratified Split 80:10:10.

No	Subset	Jumlah Data	Kelas Dyslexia	Kelas Normal	Keterangan
1	Train (80%)	14.400	7.200	7.200	Data untuk pelatihan model
2	Validation (10%)	1.800	900	900	Data untuk validasi model
3	Test (10%)	1.800	900	900	Data untuk evaluasi akhir
Total	-	18.000	9.000	9.000	Dataset final

Sumber : Diolah oleh peneliti

Berdasarkan Tabel 1, dapat diketahui bahwa dataset final berjumlah 18.000 gambar dengan distribusi kelas yang seimbang, yaitu 9.000 gambar kelas dyslexia dan 9.000 gambar kelas normal. Pembagian data dilakukan secara stratified agar proporsi setiap kelas tetap terjaga pada data latih, data validasi, dan data uji.

Selain penyesuaian ukuran gambar, dilakukan pula augmentasi data untuk meningkatkan variasi citra tulisan tangan. Pada MobileNetV2, augmentasi yang digunakan meliputi RandomFlip, RandomRotation, dan RandomZoom. Sementara itu, pada

EfficientNetB0 digunakan augmentasi yang lebih komprehensif, yaitu RandomRotation, RandomZoom, RandomTranslation, dan RandomContrast. Perbedaan strategi augmentasi ini digunakan untuk melihat pengaruh variasi data terhadap kemampuan model dalam mengenali pola tulisan tangan dyslexia dan normal.

Pemuatan Dataset

Tahapan pemuatan dataset dilakukan di lingkungan Google Colab yang digunakan sebagai platform pelatihan model. Dataset yang telah disimpan pada Google Drive dimuat menggunakan TensorFlow melalui fungsi image_dataset_from_directory. Fungsi ini digunakan karena dapat membaca gambar secara langsung dari struktur folder serta melakukan inferensi label berdasarkan nama folder kelas.

Dataset dimuat dengan ukuran gambar 224×224 piksel, menggunakan batch size sebesar 32, dan label berbentuk biner karena penelitian ini hanya memiliki dua kelas, yaitu dyslexia dan normal. Proses pemuatan dataset menghasilkan tiga subset utama, yaitu data latih, data validasi, dan data uji.

Kode Program 1. Pemuatan Dataset.

```
# Memuat dataset menggunakan image_dataset_from_directory
train_ds = tf.keras.utils.image_dataset_from_directory(
   train_dir,
   labels="inferred",
   label_mode="binary",
   image_size=(224, 224),
   batch_size=32,
   shuffle=True,
   seed=42
)
# Output train: Found 14400 files belonging to 2 classes.
# Output validation: Found 1800 files belonging to 2 classes.
# Output test: Found 1800 files belonging to 2 classes.
# Class names: ['dyslexia', 'normal']
```

Sumber : Diolah oleh peneliti

Setelah dataset berhasil dimuat, dilakukan optimasi pipeline data menggunakan tf.data.AUTOTUNE untuk mempercepat proses pemuatan data secara paralel selama pelatihan. Proses ini memastikan bahwa data siap tersedia saat model membutuhkannya, sehingga dapat mengurangi waktu tunggu antara setiap epoch pelatihan.

Augmentasi Data

Augmentasi data dilakukan untuk meningkatkan variasi data latih sehingga model lebih mampu melakukan generalisasi terhadap data baru yang belum pernah dilihat sebelumnya.

Pada model MobileNetV2, augmentasi yang diterapkan meliputi RandomFlip, RandomRotation, dan RandomZoom. Sementara itu, pada model EfficientNetB0 digunakan augmentasi yang lebih komprehensif, yaitu RandomRotation, RandomZoom, RandomTranslation, dan RandomContrast.

Perbedaan strategi augmentasi digunakan untuk melihat pengaruh variasi data terhadap kemampuan model dalam mengenali pola tulisan tangan dyslexia dan normal. Augmentasi juga membantu mengurangi risiko overfitting karena model tidak hanya mempelajari bentuk gambar yang sama, tetapi juga variasi kecil dari citra tulisan tangan.

Kode Program 2. Augmentasi Data.

```
# Augmentasi Data untuk EfficientNetB0
data_augmentation = tf.keras.Sequential([
   layers.RandomRotation(0.05),
   layers.RandomZoom(0.10),
   layers.RandomTranslation(0.05,0.05),
   layers.RandomContrast(0.10),
], name="data_augmentation")
```

Sumber : Diolah oleh peneliti

Normalisasi Piksel

Normalisasi piksel dilakukan menggunakan fungsi preprocess_input yang disediakan oleh masing-masing arsitektur. Pada MobileNetV2, nilai piksel dinormalisasi menggunakan fungsi mobilenet_v2 preprocess_input, sedangkan pada EfficientNetB0 digunakan fungsi efficientnet.preprocess_input. Normalisasi ini penting karena kedua model menggunakan bobot pra-pelatihan dari ImageNet, sehingga format input perlu disesuaikan dengan karakteristik data yang digunakan pada proses pra-pelatihan.

Dengan adanya normalisasi, representasi nilai piksel menjadi lebih konsisten dan sesuai dengan kebutuhan model. Tahap ini membantu model dalam mempelajari fitur visual dari citra tulisan tangan secara lebih stabil, sehingga proses pelatihan dapat berjalan lebih optimal.

Pelatihan Model

Pelatihan model dilakukan menggunakan pendekatan *transfer learning* dengan dua arsitektur, yaitu *MobileNetV2* dan *EfficientNetB0*, yang dilatih menggunakan *dataset* yang sama sebanyak 18.000 gambar untuk memastikan perbandingan kinerja yang objektif. *MobileNetV2* digunakan sebagai model *baseline* karena ringan dan sesuai untuk perangkat bergerak, sedangkan *EfficientNetB0* digunakan sebagai model pembanding karena memiliki kemampuan representasi fitur yang lebih baik; seluruh proses pelatihan dilakukan di Google Colab dengan dukungan GPU. Pelatihan terdiri atas tahap *feature extraction* dengan membekukan *base model* pra-latih dari *ImageNet* dan tahap *fine-tuning* dengan membuka

sebagian *layer* akhir agar model dapat beradaptasi dengan karakteristik citra tulisan tangan *dyslexia* dan *normal.*

Pelatihan Model MobileNetV2

MobileNetV2 digunakan sebagai model baseline dalam penelitian ini karena memiliki ukuran yang ringan, efisien, dan sesuai untuk kebutuhan implementasi pada perangkat mobile. Model ini menggunakan bobot pra-pelatihan dari ImageNet dengan konfigurasi `include_top=False`, sehingga layer klasifikasi bawaan tidak digunakan dan diganti dengan classification head yang disesuaikan untuk klasifikasi biner, yaitu kelas dyslexia dan normal.

Arsitektur MobileNetV2 dalam penelitian ini terdiri dari input gambar berukuran 224×224 piksel dengan tiga kanal warna RGB. Data input terlebih dahulu melalui proses augmentasi, kemudian diproses menggunakan fungsi `preprocess_input` milik MobileNetV2. Selanjutnya, citra diteruskan ke base model MobileNetV2 dengan bobot ImageNet. Hasil ekstraksi fitur kemudian diproses menggunakan `GlobalAveragePooling2D`, `Dropout(0.3)`, dan `Dense(1, activation='sigmoid')`. Fungsi sigmoid digunakan karena model hanya menghasilkan satu nilai probabilitas untuk menentukan kelas dyslexia atau normal.

Kode Program 3. Arsitektur Model MobileNetV2.

```
# Membangun model MobileNetV2
base_model = MobileNetV2(input_shape=(224,224,3),
                include_top=False, weights='imagenet')
base_model.trainable = False  # Fase 1: freeze
inputs = layers.Input(shape=(224, 224, 3))
x = data_augmentation(inputs)
x = preprocess_input(x)
x = base_model(x, training=False)
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dropout(0.3)(x)
outputs = layers.Dense(1, activation='sigmoid')
(x) model = tf.keras.Model(inputs, outputs)
```

Sumber : Diolah oleh peneliti

Model MobileNetV2 dikompilasi menggunakan optimizer Adam dengan learning rate 1e-3, fungsi loss `binary_crossentropy`, serta metrik evaluasi berupa accuracy, precision, dan recall. Pelatihan dilakukan dalam dua tahap, yaitu feature extraction dan fine-tuning. Pada tahap feature extraction, base model berada dalam kondisi frozen dan hanya classification head yang dilatih selama maksimal 15 epoch. Selanjutnya, pada tahap fine-tuning, sebagian layer atas dari base model dibuka kembali dan dilatih menggunakan learning rate yang lebih kecil, yaitu 1e-5, selama maksimal 10 epoch.

Strategi tersebut digunakan agar model dapat memanfaatkan fitur umum dari ImageNet pada tahap awal, kemudian menyesuaikan sebagian fitur tingkat tinggi terhadap karakteristik dataset tulisan tangan dyslexia dan normal pada tahap fine-tuning. Berdasarkan laporan eksperimen, MobileNetV2 memiliki total parameter sebanyak 2.259.265 atau sekitar 8,62 MB, dengan trainable parameter pada fase frozen sebanyak 1.281 parameter. Model ini tergolong ringan dan cepat, tetapi performanya masih digunakan sebagai baseline pembanding terhadap EfficientNetB0.

Pelatihan Model EfficientNetB0

EfficientNetB0 digunakan sebagai model pembanding sekaligus kandidat model utama karena memiliki kemampuan representasi fitur yang lebih kuat dibandingkan MobileNetV2. Arsitektur EfficientNetB0 dirancang dengan keseimbangan antara kedalaman jaringan, lebar jaringan, dan resolusi input, sehingga mampu mempelajari pola visual dengan lebih baik. Dalam penelitian ini, EfficientNetB0 menggunakan bobot pra-pelatihan dari ImageNet dengan konfigurasi `include_top=False`, sehingga layer klasifikasi bawaan diganti dengan classification head yang disesuaikan untuk klasifikasi biner.

Classification head pada EfficientNetB0 dibuat lebih kompleks dibandingkan MobileNetV2. Struktur yang digunakan terdiri dari `GlobalAveragePooling2D`, `BatchNormalization`, `Dense(256, activation='relu')`, `Dropout(0.4)`, `Dense(128, activation='relu')`, `Dropout(0.2)`, dan `Dense(1, activation='sigmoid')`. Penambahan BatchNormalization dan dua dense layer bertujuan untuk meningkatkan kapasitas model dalam mempelajari fitur tulisan tangan, sedangkan dropout digunakan untuk mengurangi risiko overfitting.

Kode Program 4. Setup Fine-Tuning EfficientNetB0.

```
# Arsitektur EfficientNetB0
base_model = EfficientNetB0(
        input_shape=(224, 224, 3),
        include_top=False,
        weights="imagenet"
)
base_model.trainable = False # Fase 1: freeze 238 layer
inputs = layers.Input(shape=(224, 224, 3))
x = data_augmentation(inputs)
x = efficientnet.preprocess_input(x)
x = base_model(x, training=False)
x = layers.GlobalAveragePooling2D()(x)
x = layers.BatchNormalization()(x)
x = layers.Dense(256, activation="relu")(x)
```

```
x = layers.Dropout(0.4)(x)
x = layers.Dense(128, activation="relu")(x)
x = layers.Dropout(0.2)(x)
outputs = layers.Dense(1, activation="sigmoid")(x)
model = tf.keras.Model(inputs, outputs)
```

Sumber : Diolah oleh peneliti

Pelatihan EfficientNetB0 dilakukan dalam dua tahap. Pada tahap pertama, yaitu feature extraction, seluruh layer pada base model dibekukan dan hanya classification head yang dilatih menggunakan optimizer Adam dengan learning rate 3e-4 selama maksimal 15 epoch. Pada tahap kedua, yaitu fine-tuning, sebanyak 38 layer terakhir dari base model dibuka kembali, yaitu mulai layer ke-200 hingga layer terakhir dari total 238 layer. Fine-tuning dilakukan menggunakan learning rate yang lebih kecil, yaitu 1e-5, selama maksimal 30 epoch.

Kode Program 5. Setup Fine-Tuning EfficientNetB0.

```
# Setup Fine-Tuning EfficientNetB0
base_model.trainable = True
for layer in base_model.layers[:200]:
   layer.trainable = False  # Freeze 200 layer awal
for layer in base_model.layers[200:]:
   layer.trainable = True   # Buka 38 layer terakhir
# Kompilasi ulang dengan lr lebih kecil
model.compile(optimizer=Adam(learning_rate=1e-5),
          loss='binary_crossentropy'),
                metrics=["accuracy", Precision(), Recall()] )
```

Sumber : Diolah oleh peneliti

Strategi fine-tuning pada EfficientNetB0 dibuat lebih konservatif dibandingkan MobileNetV2 karena hanya membuka 38 layer terakhir. Hal ini bertujuan agar model tetap mempertahankan fitur umum dari ImageNet, tetapi tetap mampu menyesuaikan fitur tingkat tinggi terhadap karakteristik citra tulisan tangan dyslexia dan normal. Berdasarkan laporan eksperimen, EfficientNetB0 memiliki total parameter sebanyak 4.415.652 atau sekitar 16,84 MB, dengan trainable parameter pada fase frozen sebanyak 363.521 parameter. Meskipun ukuran model lebih besar dibandingkan MobileNetV2, EfficientNetB0 memberikan kapasitas belajar yang lebih kuat dan menjadi kandidat model utama untuk integrasi ke aplikasi Android.Selama proses pelatihan, digunakan beberapa callback untuk menjaga stabilitas proses training, yaitu EarlyStopping, ModelCheckpoint, dan ReduceLROnPlateau. EarlyStopping digunakan untuk menghentikan pelatihan ketika performa validasi tidak mengalami peningkatan, ModelCheckpoint digunakan untuk menyimpan model terbaik secara otomatis, sedangkan ReduceLROnPlateau digunakan untuk menurunkan learning rate secara adaptif ketika performa validasi mulai stagnan.

Hasil Evaluasi Model

Evaluasi model dilakukan menggunakan *test set* sebanyak 1.800 gambar yang terdiri dari 900 gambar kelas *dyslexia* dan 900 gambar kelas *normal*, dimana data tersebut tidak digunakan pada tahap pelatihan maupun validasi sehingga mampu menggambarkan kinerja model terhadap data baru. Kinerja model diukur menggunakan metrik *loss*, *accuracy*, *precision*, *recall*, dan *F1-score* untuk mengevaluasi tingkat akurasi, ketepatan prediksi, kemampuan identifikasi kelas, serta keseimbangan antara *precision* dan *recall*.

Hasil Model MobileNetV2

Tabel 2. Hasil Evaluasi Model MobileNetV2.

No	Metrik Evaluasi	Nilai	No	Metrik Evaluasi
1	Loss	0,5600	1	Loss
2	Accuracy	70,1%	2	Accuracy
3	Precision	71,9%	3	Precision
4	Recall	66,1%	4	Recall
5	F1-score Macro	0,70	5	F1-score Macro

Sumber : Diolah oleh peneliti

Berdasarkan Tabel 2, MobileNetV2 menghasilkan performa yang cukup baik sebagai baseline, tetapi masih memiliki keterbatasan dalam mengenali pola tulisan tangan secara konsisten. Nilai recall sebesar 66,1% menunjukkan bahwa masih terdapat sejumlah data yang belum berhasil diklasifikasikan dengan tepat. Oleh karena itu, model ini digunakan sebagai pembanding terhadap EfficientNetB0 yang memiliki arsitektur lebih kuat.

Hasil Model EfficientNetB0

Tabel 3. Hasil Evaluasi Model EfficientNetB0.

No	**Metrik Evaluasi**	**Nilai**	**No**	**Metrik Evaluasi**
1	Loss	0,1383	1	Loss
2	Accuracy	94,22%	2	Accuracy
3	Precision	95,96%	3	Precision
4	Recall	92,33%	4	Recall
5	F1-score Macro	0,94	5	F1-score Macro

Sumber : Diolah oleh peneliti

Berdasarkan Tabel 3, EfficientNetB0 menunjukkan performa yang jauh lebih baik dibandingkan MobileNetV2. Nilai accuracy sebesar 94,22% menunjukkan bahwa model mampu mengklasifikasikan sebagian besar data uji dengan benar. Selain itu, nilai recall sebesar 92,33% menunjukkan bahwa model memiliki kemampuan yang baik dalam mendeteksi data yang termasuk kelas dyslexia maupun normal. Dengan hasil tersebut, EfficientNetB0 dipilih sebagai model utama untuk dikonversi ke TensorFlow Lite dan diintegrasikan ke aplikasi Android.

Confusion Matrix Model

Confusion Matrix MobileNetV2

Tabel 4. Confusion Matrix Model MobileNetV2.

Aktual / Prediksi	Prediksi: Dyslexia	Prediksi: Normal
Aktual: Dyslexia	666	234
Aktual: Normal	306	594

Sumber : Diolah oleh peneliti

Berdasarkan Tabel 4, MobileNetV2 masih memiliki jumlah kesalahan yang cukup tinggi, yaitu 540 kesalahan dari 1.800 data uji. Kesalahan tersebut terdiri atas 234 data dyslexia yang salah diklasifikasikan sebagai normal dan 306 data normal yang salah diklasifikasikan sebagai dyslexia. Kondisi ini menunjukkan bahwa MobileNetV2 masih belum cukup stabil dalam membedakan pola tulisan tangan dyslexia dan normal.

Confusion Matrix EfficientNetB0

Tabel 5. Confusion Matrix Model EfficientNetB0.

Aktual / Prediksi	Prediksi: Dyslexia	Prediksi: Normal
Aktual: Dyslexia	864	36
Aktual: Normal	72	828

Sumber : Diolah oleh peneliti

Berdasarkan Tabel 5, EfficientNetB0 menunjukkan performa yang jauh lebih baik dibandingkan MobileNetV2. Jumlah kesalahan model hanya sebanyak 108 data dari 1.800 data uji. Nilai false negative pada kelas dyslexia juga rendah, yaitu 36 data, sehingga model lebih mampu meminimalkan kasus dyslexia yang terlewat. Hal ini menjadikan EfficientNetB0 lebih layak digunakan sebagai model utama dalam sistem deteksi dini disleksia.

Analisis Perbandingan Model

Tabel 6. Perbandingan Hasil Evaluasi Model MobileNetV2 dan EfficientNetB0.

No	Metrik Evaluasi	MobileNetV2	EfficientNetB0
1	Accuracy	70,1%	94,22%
2	Loss	0,5600	0,1383
3	Precision	71,9%	95,96%
4	Recall	66,1%	92,33%
5	F1-score Macro	0,70	0,94
6	Total Kesalahan Prediksi	540 data	108 data

Sumber : Diolah oleh penelit

Berdasarkan Tabel 6, EfficientNetB0 memiliki peningkatan performa yang signifikan dibandingkan MobileNetV2. Selisih accuracy antara kedua model mencapai 24,12 poin persentase. Selain itu, jumlah kesalahan prediksi EfficientNetB0 hanya 108 data dari 1.800 data uji, sedangkan MobileNetV2 memiliki 540 kesalahan prediksi. Hal ini menunjukkan bahwa EfficientNetB0 lebih stabil dalam mengenali pola tulisan tangan pada kelas dyslexia maupun normal.

Dari sisi arsitektur, MobileNetV2 memiliki keunggulan pada ukuran model yang lebih ringan dan efisiensi komputasi yang baik, sehingga cocok digunakan sebagai baseline untuk perangkat mobile. Namun, pada eksperimen ini performa MobileNetV2 masih belum optimal karena nilai recall dan precision masih lebih rendah dibandingkan EfficientNetB0. Sebaliknya, EfficientNetB0 memiliki kapasitas representasi fitur yang lebih kuat melalui keseimbangan kedalaman jaringan, lebar jaringan, dan resolusi input. Hal ini membuat EfficientNetB0 lebih mampu membedakan pola visual pada citra tulisan tangan.

Berdasarkan hasil perbandingan tersebut, EfficientNetB0 dipilih sebagai model utama dalam penelitian ini. Model ini dipilih karena memiliki accuracy, precision, recall, dan F1-score yang lebih tinggi dibandingkan MobileNetV2. Selain itu, EfficientNetB0 juga menghasilkan jumlah kesalahan prediksi yang lebih rendah, sehingga lebih layak digunakan sebagai model yang akan dikonversi ke format TensorFlow Lite dan diintegrasikan ke dalam aplikasi Android pendeteksi disleksia.

Konversi Model ke TensorFlow Lite

Tabel 7. Hasil Konversi Model EfficientNetB0 ke TensorFlow Lite.

No	Format Model	Ukuran Model	Keterangan
1	Keras (.keras)	±16,84 MB	Digunakan untuk training dan evaluasi
2	TFLite Float32	±16,68 MB	Digunakan untuk deployment dengan akurasi optimal
3	TFLite Quantized INT8	±8,40 MB	Digunakan untuk deployment mobile yang lebih hemat memori

Sumber : Diolah oleh peneliti

Berdasarkan Tabel 7, model TensorFlow Lite Quantized INT8 memiliki ukuran yang lebih kecil dibandingkan model Keras dan TFLite Float32. Pengurangan ukuran model ini bermanfaat untuk implementasi pada perangkat Android karena dapat menghemat ruang penyimpanan dan mempercepat proses inferensi. Meskipun demikian, pemilihan format model tetap perlu mempertimbangkan keseimbangan antara ukuran model, kecepatan inferensi, dan akurasi hasil prediksi.

Model TensorFlow Lite yang digunakan memiliki input berupa citra RGB berukuran 224×224 piksel dengan bentuk input (1, 224, 224, 3). Output model berupa satu nilai sigmoid dengan bentuk output (1, 1). Nilai output tersebut digunakan sebagai dasar klasifikasi, di mana nilai lebih dari 0,5 dikategorikan sebagai dyslexia, sedangkan nilai kurang dari atau sama dengan 0,5 dikategorikan sebagai normal.

Integrasi Model ke Aplikasi Android

Setelah model *EfficientNetB0* dikonversi ke format *TensorFlow Lite*, model diintegrasikan ke dalam aplikasi Android agar proses deteksi dapat dilakukan secara langsung

pada perangkat pengguna tanpa memerlukan server eksternal, sehingga lebih cepat, ringan, dan menjaga privasi data. Model ditempatkan pada folder *assets* dan dijalankan menggunakan *TensorFlow Lite Interpreter*, dengan masukan berupa citra tulisan tangan yang telah melalui proses *resize* menjadi 224×224 piksel, konversi RGB, dan normalisasi data. Proses klasifikasi dimulai dari pengambilan gambar oleh pengguna, dilanjutkan dengan *pre-processing*, inferensi menggunakan model, dan penentuan hasil berdasarkan *threshold* 0,5, di mana nilai di atas 0,5 diklasifikasikan sebagai *dyslexia* dan nilai di bawah atau sama dengan 0,5 diklasifikasikan sebagai *normal*.

Kode Program 6. Konversi Model ke TFLite

```
private Interpreter tflite;
private void loadModel() {
   try {
      MappedByteBuffer modelBuffer =
      FileUtil.loadMappedFile(this, "dyslexia_model.tflite");
      tflite = new Interpreter(modelBuffer);
   } catch (IOException e) {
      e.printStackTrace();
   }
}
private float[][] runInference(float[][][][] inputImage) {
   float[][] output = new float(Lestari, 2024)(Lestari, 2024);
   tflite.run(inputImage, output);
   return output;
}
```

Sumber : Diolah oleh peneliti

Kode Program 6 menunjukkan proses pemanggilan model TensorFlow Lite pada aplikasi Android. Fungsi loadModel() digunakan untuk membaca file model dyslexia_model.tflite dari folder assets dan memuatnya ke dalam TensorFlow Lite Interpreter. Selanjutnya, fungsi runInference() digunakan untuk mengirimkan input gambar ke model dan menerima output prediksi berupa nilai sigmoid.

Output dari model kemudian digunakan untuk menentukan hasil klasifikasi. Nilai output lebih dari 0,5 dikategorikan sebagai dyslexia, sedangkan nilai kurang dari atau sama dengan 0,5 dikategorikan sebagai normal. Hasil klasifikasi tersebut kemudian ditampilkan pada antarmuka aplikasi sebagai informasi awal bagi pengguna. Dengan adanya integrasi ini, aplikasi Android dapat menjalankan proses deteksi dini disleksia secara langsung pada perangkat mobile.

Pengujian Blackbox Testing dan User Acceptance Test (UAT)

Tabel 8. Kategori Penilaian System Usability Scale (SUS).

No	Fitur Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
1	Register	Pengguna mengisi nama, email, dan password lalu menekan tombol Daftar	Akun berhasil dibuat dan pengguna diarahkan ke halaman login	Akun terdaftar, halaman login tampil	Valid
2	Login	Pengguna memasukkan email dan password yang valid lalu menekan tombol Masuk	Pengguna berhasil masuk ke halaman utama aplikasi	Halaman utama berhasil ditampilkan	Valid
3	Upload Tulisan Tangan	Pengguna memilih foto tulisan tangan anak dari galeri atau mengambil foto melalui kamera	Gambar tulisan tangan berhasil ditampilkan pada layar dan siap untuk diproses	Gambar tampil pada area preview	Valid
4	Prediksi Model CNN	Pengguna menekan tombol Periksa setelah gambar diunggah	Sistem memproses gambar menggunakan model EfficientNetB0 TFLite dan menghasilkan prediksi kelas	Proses inferensi berjalan dan prediksi dihasilkan	Valid
5	Tampil Hasil Deteksi	Sistem selesai melakukan inferensi terhadap gambar tulisan tangan	Hasil deteksi tampil berupa label Normal atau Indikasi Disleksia disertai nilai probabilitas	Label dan probabilitas berhasil ditampilkan	Valid
6	Rekomendasi Tindak Lanjut	Hasil deteksi menampilkan label Indikasi Disleksia	Sistem menampilkan rekomendasi tindak lanjut berupa saran konsultasi dengan tenaga ahli atau latihan membaca	Rekomendasi tindak lanjut tampil sesuai hasil deteksi	Valid
7	Simpan Riwayat	Setelah hasil deteksi ditampilkan, sistem menyimpan hasil secara otomatis	Data hasil deteksi tersimpan pada menu Riwayat dan dapat diakses kembali	Data tersimpan pada menu History	Valid
8	Riwayat Pemeriksaan	Pengguna membuka menu Riwayat pada navigasi utama	Daftar riwayat pemeriksaan sebelumnya ditampilkan beserta tanggal dan hasil deteksi	Riwayat pemeriksaan tampil dengan lengkap	Valid
9	Artikel Edukasi	Pengguna membuka menu Artikel pada navigasi utama	Daftar artikel edukatif mengenai disleksia dan cara penanganannya ditampilkan dan dapat dibaca	Artikel berhasil ditampilkan dan dapat dibaca	Valid
10	Logout	Pengguna menekan tombol Keluar pada halaman profil	Sesi pengguna diakhiri dan sistem kembali menampilkan halaman login	Sistem kembali ke halaman login	Valid

Sumber : Diolah oleh peneliti

Berdasarkan hasil *Blackbox Testing* pada Tabel 8, seluruh sepuluh skenario pengujian yang mencakup fitur utama aplikasi menunjukkan status valid, sehingga seluruh fungsi aplikasi telah berjalan sesuai dengan keluaran yang diharapkan berdasarkan pengujian oleh 50 responden dari SDN 4 Panjer. Fitur autentikasi pengguna yang meliputi registrasi dan *login* juga berfungsi dengan baik, termasuk validasi email yang telah terdaftar, pengisian data pengguna, serta pengalihan ke halaman utama setelah proses registrasi berhasil dilakukan.

Evaluasi Usabilitas dengan System Usability Scale (SUS)

Setelah *Blackbox Testing* selesai dilaksanakan, dilakukan evaluasi usabilitas menggunakan metode *System Usability Scale* (SUS) yang melibatkan 50 penguji, terdiri atas 10 guru dan 40 orang tua atau wali murid, untuk mengukur tingkat penerimaan dan kenyamanan pengguna terhadap sistem berdasarkan pengalaman penggunaan secara langsung. Instrumen SUS terdiri dari 10 pernyataan dengan skala penilaian 1–5 yang digunakan untuk menilai aspek kemudahan penggunaan, fungsionalitas, konsistensi antarmuka, kepercayaan diri pengguna, dan kepuasan terhadap sistem, dengan skor akhir berada pada rentang 0–100. Interpretasi skor SUS dilakukan menggunakan kategori *adjective scale* yang menjadi acuan dalam menentukan tingkat usabilitas sistem yang dikembangkan.

Tabel 9. Kategori Penilaian System Usability Scale (SUS).

No	Rentang Skor SUS	Kategori
1	85 – 100	Excellent (Best Imaginable)
2	70 – 84	Good
3	50 – 69	OK (Fair)
4	35 – 49	Poor
5	0 – 34	Awful (Worst Imaginable)

Sumber : Diolah oleh peneliti

Skor SUS dihitung per responden menggunakan rumus standar, kemudian dirata-ratakan untuk mendapatkan nilai akhir. Skor rata-rata yang diperoleh selanjutnya diinterpretasikan berdasarkan kategori adjective scale SUS. Rekapitulasi hasil perhitungan skor SUS dari 50 penguji disajikan pada Tabel 10.

Tabel 10. Rekapitulasi Hasil Evaluasi System Usability Scale (SUS).

No	Kelompok Penguji	Jumlah Penguji	Skor SUS Rata-rata	Interpretasi
1	Guru SDN 4 Panjer	10	86,25	Good
2	Orang Tua / Wali Murid	40	85,50	Good
3	Keseluruhan (50 penguji)	50	85,75	Good
	Rata-rata SUS Keseluruhan	50	85,75	Good

Sumber : Diolah oleh peneliti

Berdasarkan Tabel 10, hasil evaluasi System Usability Scale (SUS) menunjukkan bahwa seluruh kelompok penguji memberikan skor di atas 85. Kelompok guru SDN 4 Panjer (*usability*) memperoleh skor SUS rata-rata sebesar 86,25, yang termasuk dalam kategori Good

berdasarkan adjective scale SUS. Kelompok orang tua dan wali murid memberikan skor rata-rata sebesar 85,50, juga masuk dalam kategori Good, yang mengindikasikan bahwa antarmuka aplikasi mudah dipahami oleh pengguna non-teknis.

Secara keseluruhan, skor SUS rata-rata dari 50 penguji adalah 85,75, yang berada dalam kategori Good berdasarkan adjective scale SUS. Nilai ini berada di atas ambang batas 70 yang umum dijadikan standar minimum kelayakan usabilitas, dan mendekati kategori Excellent (di atas 85). Hal ini menunjukkan bahwa sistem diterima dengan baik oleh pengguna dan layak digunakan sebagai alat bantu deteksi dini disleksia di lingkungan pendidikan.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa pengembangan aplikasi Android pendeteksi disleksia pada anak usia dini berbasis *machine learning* berhasil diimplementasikan menggunakan metode *Convolutional Neural Network* (CNN). Model terbaik yang diperoleh, yaitu EfficientNetB0, menunjukkan kinerja yang optimal dengan tingkat akurasi sebesar 86% dalam mendeteksi pola tulisan tangan anak. Aplikasi yang dikembangkan mampu melakukan klasifikasi secara otomatis dan efisien ke dalam kategori normal maupun indikasi disleksia. Selain itu, seluruh fitur utama aplikasi, meliputi pemindaian tulisan, hasil deteksi, riwayat, serta rekomendasi tindak lanjut, telah berfungsi dengan baik berdasarkan hasil pengujian sistem. Hasil evaluasi menggunakan *User Acceptance Testing* menunjukkan bahwa sistem memperoleh tingkat usabilitas yang baik dari pengguna berdasarkan evaluasi System Usability Scale (SUS), dengan skor rata-rata keseluruhan sebesar 85,75 yang masuk dalam kategori Good. Seluruh fitur utama aplikasi dinyatakan Valid dalam pengujian Blackbox Testing yang dilakukan oleh 50 penguji dari SDN 4 Panjer. Hal ini menunjukkan bahwa sistem tidak hanya berfungsi secara teknis, tetapi juga mampu memberikan pengalaman penggunaan yang positif dan memenuhi kebutuhan pengguna. Dengan demikian, aplikasi ini memiliki potensi sebagai alat bantu deteksi dini disleksia yang praktis, cepat, dan objektif bagi guru maupun orang tua.

DAFTAR REFERENSI

Adzkia, S. A., & Arifin, T. (2025). Perbandingan arsitektur CNN berbasis *transfer learning* untuk klasifikasi pada BreastMNIST. *Jurnal Nasional Teknologi dan Sistem Informasi, 11*(2), 192–200. https://doi.org/10.25077/TEKNOSI.v11i2.2025.192-200

Alfaruqi, R., Alfarisi, S., & Afrizal, T. (2022). Implementasi Firebase Cloud Storage pada aplikasi *e-commerce* Toko Ktoys berbasis Android. *Jurnal Rekayasa Komputasi Terapan, 2*(3). https://doi.org/10.30998/jrkt.v2i03.7970

AlFatrah, M. I., Sudaryanto, H., & Rimbawa, H. A. D. (2025). Implementasi teknologi MediaPipe menggunakan metode CNN berbasis website untuk pengamanan VVIP dalam mobil. *SITEKNIK: Sistem Informasi, Teknik dan Teknologi Terapan, 2*(3), 260–371.

Alqindi, A. N., Harahap, Y. M., Stompul, T. G., & Purba, Y. A. (2024). Disleksia dalam pendidikan: Tantangan yang dihadapi dan strategi pembelajaran yang mendukung. *Jurnal Intelek dan Cendikiawan Nusantara, 1*(6), 10140–10147.

Apriani, E., & Pratiwi, N. (2025). Perancangan sistem pengenalan tulisan tangan pada jawaban esai menggunakan metode CNN-LSTM berbasis Android. *METIK Jurnal, 9*(2), 385–396.

Ardhana, V. Y. P. (2021). Pengujian usability aplikasi Halodoc menggunakan metode *System Usability Scale* (SUS). *Jurnal Kesehatan Qamarul Huda, 9*(2), 132–136. https://doi.org/10.37824/jkqh.v9i2.2021.311

Astuti, Y. P., Subhiyakto, E. R., Wardatunizza, I., & Kartikadarma, E. (2023). Implementasi algoritma *Convolutional Neural Network* (CNN) untuk klasifikasi jenis tanah berbasis Android. *Jurnal Informatika: Jurnal Pengembangan IT, 8*(3), 220–225. https://doi.org/10.30591/jpit.v8i3.5026

Azmi, A. F., & Voutama, A. (2024). Prediksi *churn* nasabah bank menggunakan klasifikasi Random Forest dan Decision Tree dengan evaluasi *confusion matrix*. *Komputa: Jurnal Ilmiah Komputer dan Informatika, 13*(1), 111–119. https://doi.org/10.34010/komputa.v13i1.12639

Dewi, F. K. S., Adithama, S. P., & Suhardi, A. T. (2023). Pengujian aplikasi Doctor to Doctor menggunakan metode *black box testing*. *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi, 3*(1), 61–72. https://doi.org/10.24002/konstelasi.v3i1.7046

Dirgayunita, A., Dheasari, A. E., & Masyhuri, M. (2022). Identifikasi kesulitan belajar disleksia anak usia dini. *Al-Athfal: Jurnal Pendidikan Anak, 3*(1), 36–54. https://doi.org/10.46773/al-athfal.v3i1.426

Dwijantie, J. S. (2025). Pendekatan *deep learning* dalam pembelajaran PAUD: Membangun pemahaman mendalam bagi anak usia dini. *Edukasiana: Jurnal Inovasi Pendidikan, 4*(3), 1238–1246. https://doi.org/10.56916/ejip.v4i3.1666

Filasofa, L. M. K., & Miswati, M. (2021). Perkembangan kognitif anak usia dini penyandang disleksia: Studi kasus pada lembaga pendidikan di Indonesia. *Journal of Early Childhood and Character Education, 1*(1), 53–72. https://doi.org/10.21580/joecce.v1i1.6615

Hamdi, N., Junidar, J., Serianti, P., & Nargaza, J. (2025). Implementasi *machine learning* dalam meningkatkan akurasi sistem pakar diagnosa penyakit jantung berbasis Android. *Journal of Informatics and Computer Science, 11*(1), 75–81.

Haryanti, N., Muhibbudin, M., & Junaris, I. (2022). Analisis kesulitan belajar siswa (disleksia dan disgrafia) di masa pandemi COVID-19. *Journal of Instructional and Development Researches, 2*(1), 7–16. https://doi.org/10.53621/jider.v2i1.60

Lestari, D. A., Khasanah, M. N., Ambarita, Y. S., & Mustika, D. (2024). Hakikat pendidikan inklusif anak usia dini. *Catha: Journal of Creative and Innovative Research, 1*(3), 25–29.

Lestari, S. A. M., Pardede, A. M. H., & Simanjuntak, M. (2024). Prediksi disleksia pada anak menggunakan metode Naive Bayes. *Jurnal Kajian dan Penelitian Umum, 2*(5), 37–51. https://doi.org/10.47861/jkpu-nalanda.v2i5.1287

Manik, J., Faizah, N. M., & Ginting, W. (2023). Pengembangan aplikasi game edukasi matematika SD 02 Pagi Pondok Labu berbasis Android dengan metode *Rapid Application Development* menggunakan Adobe Flash CS6. *Design Journal, 1*(1), 99–108. https://doi.org/10.58477/dj.v1i1.62

Nabila, N. A., & Rachman, I. F. (2025). Peran AI dalam mendeteksi dan mengintervensi keterlambatan belajar siswa di pendidikan dasar. *Jurnal Kesehatan dan Kebidanan Nusantara, 3*(2), 36–43. https://doi.org/10.69688/jkn.v2i2.110

Oktamarina, L., et al. (2022). Gangguan gejala disleksia pada anak usia dini. *Jurnal Multidisipliner Bharasumba, 1*(2), 104–118. https://doi.org/10.62668/bharasumba.v1i02.189

Pramana, A. L. (2024). Analisis perbandingan evaluasi metode *deep learning* pada klasifikasi jenis kendaraan. *Journal of Computer Science and Visual Communication Design, 9*(1), 451–465. https://doi.org/10.55732/jikdiskomvis.v9i1.1252

Ridho, A. N., Mellyka, G., Saputra, F., & Masa, A. P. A. (2024). Implementasi metode *Convolutional Neural Network* (CNN) untuk klasifikasi gambar mobil dan motor menggunakan Keras. *Jurnal Smart Technology dan Otomasi, 1*(1), 54–64.

Sembiring, M. T., Azhari, I. F., & Wigatra, R. (2025). Perancangan aplikasi latihan soal ujian berbasis Android menggunakan Android Studio pada SMP Negeri 2 Pematangsiantar. *(JRSIKOM) Jurnal Riset Sistem Informasi dan Aplikasi Komputer, 1*(1), 5–11.

Sitorus, H. (2022). Implementasi *deep learning* mendeteksi pengguna masker berbasis framework TensorFlow dengan metode *Convolutional Neural Network*. *SENTRI: Jurnal Riset Ilmiah, 1*(3), 862–874. https://doi.org/10.55681/sentri.v1i3.298

Suarjaya, I. M. A. D., Wiratama, B. A., & Wirdiani, A. (2024). Sistem pengenalan huruf Braille menggunakan metode *deep learning* berbasis website. *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi, 5*(3), 93–99. https://doi.org/10.62527/jitsi.5.3.244

Sumiati, U., Hidayah, T. R., Suryani, A. I., Aliyani, N., & Anggraeni, I. (2025). Deteksi dini dan penanganan disleksia: Studi kasus pada anak usia dini di SPS TAAM Nurul Hidayah. *Jurnal Anak Bangsa, 4*(1), 15–23.

Syahroni, I., Rofiqoh, W., & Latipah, E. (2021). Ciri-ciri disleksia pada anak usia dini. *Jurnal Buah Hati, 8*(1), 62–77. https://doi.org/10.46244/buahhati.v8i1.1326