

Klasifikasi Karakter Jepang (Hiragana dan Katakana) Menggunakan Convolutional Neural Network (CNN)

Febriyan Biopsa Minanda¹

¹Program Studi Informatika, Universitas Atma Jaya Yogyakarta

¹biopza27@gmail.com

Abstrak

Jepang memiliki budaya yang berpengaruh di Indonesia melalui *anime*, *manga*, dan kuliner, sehingga meningkatkan minat masyarakat untuk mempelajari bahasa Jepang. Namun, pembelajaran bahasa Jepang masih memiliki kendala, terutama dalam mengenali huruf *hiragana* dan *katakana* yang memiliki bentuk berbeda dan sulit diingat. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem klasifikasi karakter *hiragana* dan *katakana* menggunakan metode *custom Convolutional Neural Network* (CNN). *Dataset* diperoleh secara mandiri melalui proses konversi teks karakter Jepang menjadi citra menggunakan 45 variasi *font* bahasa Jepang. Selanjutnya, dilakukan augmentasi data berupa erosi, dilasi, dan *Gaussian blur* untuk meningkatkan keragaman citra, sehingga diperoleh total 46.000 gambar dari 92 kelas, yang terdiri atas 46 karakter *hiragana* dan 46 karakter *katakana*. Tahapan penelitian meliputi pengumpulan data, *preprocessing* citra, pembagian *dataset*, perancangan arsitektur CNN, pelatihan model, evaluasi model, implementasi sistem berbasis *Streamlit*, serta pengujian *usability* menggunakan metode *System Usability Scale* (SUS). Hasil penelitian menunjukkan bahwa model CNN mampu mencapai akurasi sebesar 97,47% dengan *validation accuracy* tertinggi sebesar 98,00%. Sistem yang dikembangkan dapat melakukan prediksi karakter melalui dua fitur utama, yaitu *upload* gambar dan *canvas* secara *real-time*. Hasil pengujian *usability* terhadap 18 responden memperoleh nilai rata-rata SUS sebesar 68,61 yang termasuk kategori *marginal high*, *Grade C*, dan berada pada rentang OK, sehingga sistem dinilai layak sebagai media pendukung pembelajaran karakter Jepang secara interaktif, mandiri, dan mudah diakses.

Kata kunci: klasifikasi citra, *deep learning*, *convolutional neural network*, *hiragana*, *katakana*

1. Pendahuluan

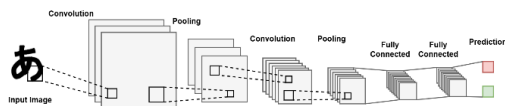
Jepang termasuk salah satu negara di wilayah Asia yang terkenal dengan kebudayaan unik dan memikat. Pengaruh budaya Jepang telah menyebar ke berbagai negara, termasuk Indonesia, melalui *anime*, *manga*, serta kuliner khas seperti *sushi*, *ramen*, dan *takoyaki*, sehingga meningkatkan minat masyarakat untuk mempelajari bahasa Jepang. Kondisi tersebut menyebabkan bahasa Jepang menjadi salah satu bahasa asing yang banyak menarik minat pelajar serta mahasiswa di Indonesia (Setiadi dkk., 2024). Hal ini sejalan dengan penelitian yang menyatakan bahwa “pembelajaran bahasa Jepang memiliki pengaruh signifikan terhadap minat individu dalam bidang pendidikan maupun karier” (Hidayanti dkk., 2024).

Bahasa Jepang mempunyai mekanisme penulisan yang tidak sama dengan *alfabet Latin* sebab tersusun atas beberapa bentuk aksara, yakni *hiragana*, *katakana*, dan *kanji*. Pada proses pembelajaran tingkat dasar, huruf *hiragana* dan *katakana* merupakan materi awal yang wajib dipahami karena berfungsi sebagai landasan untuk membaca dan menulis bahasa Jepang (Rasyiq dkk., 2023). Masing-masing jenis huruf tersebut memiliki 46 karakter

dasar yang perlu dipahami oleh pembelajar. Namun demikian, banyak pembelajar mengalami kesulitan dalam mengenali dan mengingat karakter tersebut karena bentuknya yang berbeda dengan huruf Latin (Purnamasari dkk., 2023).

Kesulitan dalam mempelajari huruf *hiragana* dan *katakana* juga dipengaruhi oleh adanya kemiripan bentuk antar karakter serta kompleksitas dalam penulisannya. Kondisi ini menyebabkan rendahnya tingkat penguasaan huruf pada pembelajar sehingga berdampak pada kemampuan membaca dan menulis bahasa Jepang. Penelitian sebelumnya menunjukkan bahwa “penguasaan huruf *hiragana* pada siswa masih belum optimal dan memerlukan metode pembelajaran yang lebih efektif” (Mutiar dkk., 2025). Oleh karena itu, diperlukan pendekatan yang lebih inovatif untuk membantu proses pengenalan karakter secara lebih efektif.

Bersamaan dengan kemajuan teknologi, penerapan kecerdasan buatan pada bidang pengolahan citra menjadi salah satu alternatif penyelesaian terhadap permasalahan tersebut. Pengolahan citra digital adalah kegiatan mengubah sebuah gambar ke bentuk berbeda dengan mutu yang lebih tinggi



Gambar 1. Contoh Arsitektur CNN

melalui penerapan algoritma maupun metode tertentu. Bidang tersebut mempelajari pembentukan, pemrosesan, serta analisis citra untuk menghasilkan informasi yang dapat dipahami oleh manusia maupun *computer* (Ardi Wijaya dkk., 2024). Tujuan dari pengolahan citra meliputi:

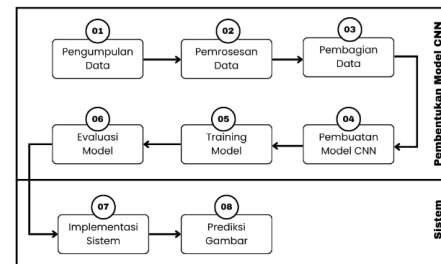
1. Meningkatkan kualitas citra agar lebih mudah diinterpretasikan,
2. Mentransformasikan citra ke dalam bentuk lain sesuai kebutuhan,
3. Berperan sebagai tahap awal dalam proses komputerisasi.

Salah satu pendekatan yang umum diterapkan adalah *Convolutional Neural Network* (CNN), yang mampu mengidentifikasi pola visual secara otomatis dengan ketepatan tinggi. CNN telah menunjukkan efektivitasnya dalam berbagai studi pengenalan tulisan tangan, termasuk aksara *hiragana* Jepang dengan akurasi yang sangat baik (Willyanto dkk., 2021). Selain itu, penelitian lain juga menunjukkan bahwa “CNN mampu memberikan performa yang sangat baik dalam pengenalan aksara berbasis citra digital” (Handoko dkk., 2024).

Convolutional Neural Network (CNN) adalah arsitektur pada *deep learning* yang dikembangkan dari *Multilayer Perceptron* (MLP) dan dirancang berdasarkan konsep jaringan saraf biologis manusia. CNN tersusun atas lapisan konvolusi, *pooling*, dan *fully connected* yang beroperasi secara bertahap untuk memperoleh karakteristik dari data terutama citra. Masing masing lapisan berperan dalam mengenali pola lokal maupun pola global. Metode tersebut telah menunjukkan efektivitas pada beragam aplikasi visi komputer seperti pengelompokan gambar dan penberdasarkan objek (Nurmaini dkk., 2021). Contoh susunan arsitektur CNN ditampilkan pada Gambar 1.

Arsitektur CNN dibangun oleh *neuron-neuron* yang terkoneksi menggunakan bobot serta pada umumnya mencakup tiga lapisan pokok, yakni konvolusi, *pooling*, dan *fully connected*. Lapisan konvolusi bertugas memperoleh fitur utama seperti pola maupun batas objek dari masukan, sementara *pooling* dimanfaatkan untuk mereduksi ukuran data sehingga proses menjadi lebih efisien. Berikutnya, lapisan *fully connected* bertugas menghasilkan keluaran akhir dengan memanfaatkan fitur yang telah dipelajari sebelumnya (Nurmaini dkk., 2021).

Beberapa penelitian terdahulu telah mengkaji pengenalan karakter Jepang dan pengembangan media pembelajaran bahasa Jepang. Rasyiq dkk. (2023) dan Setiadi dkk. (2024) mengembangkan aplikasi



Gambar 2. Alur Penelitian

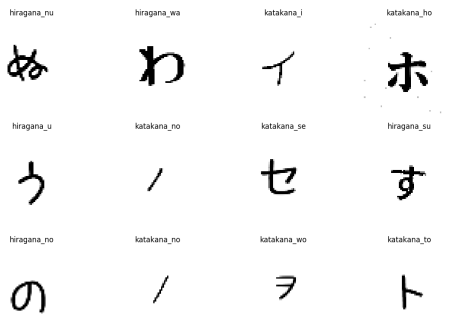
pembelajaran *hiragana* dan *katakana* berbasis *Android*, tetapi belum berfokus pada klasifikasi karakter secara otomatis menggunakan *deep learning*. Willyanto dkk. (2021) menerapkan CNN berarsitektur VGG-16 untuk identifikasi tulisan tangan *hiragana*, sedangkan Nurhidayat dkk. (2025) menerapkan CNN pada pengenalan aksara *kanji*. Berdasarkan celah tersebut, kontribusi penelitian ini terletak pada pengembangan model *custom CNN* untuk mengklasifikasikan karakter dasar *hiragana* dan *katakana* sekaligus, serta mengimplementasikannya dalam sistem berbasis web yang dapat digunakan secara interaktif melalui fitur *upload* gambar dan *canvas*.

Atas dasar permasalahan tersebut, penelitian ini diarahkan untuk merancang sistem klasifikasi karakter Jepang, terutama *hiragana* dan *katakana*, dengan metode *custom Convolutional Neural Network* (CNN). Model yang dirancang akan menjalani proses pelatihan menggunakan *dataset* citra karakter Jepang guna menghasilkan sistem klasifikasi yang presisi serta mampu membedakan karakter dengan bentuk yang serupa.

Selain itu, hasil dari model akan diimplementasikan dalam sebuah *platform* berbasis *web* menggunakan *Streamlit* untuk mempermudah pengguna dalam melakukan pengujian secara langsung. Kehadiran sistem tersebut diharapkan mampu mendukung kegiatan belajar bahasa Jepang sekaligus berkontribusi terhadap kemajuan teknologi identifikasi karakter yang memanfaatkan *deep learning*.

2. Metode

Tahapan penggunaan model CNN pada penelitian ini mencakup sejumlah langkah, mulai dari penghimpunan *dataset* karakter *hiragana* dan *katakana*, kemudian pemrosesan awal data, penyusunan arsitektur CNN, proses pelatihan, evaluasi model, sampai penerapan sistem. Setiap tahapan tersebut dilakukan secara berurutan untuk menghasilkan model klasifikasi yang optimal. Rangkaian penelitian ini disajikan pada Gambar 2.



Gambar 3. Dataset Karakter Jepang

2.1 Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan sendiri dengan mengonversi teks karakter *hiragana* dan *katakana* ke dalam bentuk citra memakai beragam *font* bahasa Jepang. *Font* yang digunakan diperoleh dari *website* “Free Japanese Font” dengan contoh *font* seperti *NotoSansCJKp*, *Keinanm-POPJP*, *HolidayMDJP*, *Idamoji*, dan *Kiloji*, serta total sebanyak 45 jenis *font* yang berbeda. Tahapan ini dimaksudkan untuk menciptakan keragaman bentuk karakter sehingga kemampuan model dalam menggeneralisasi pola dapat meningkat.

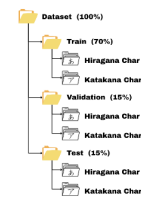
Setelah citra berhasil dibuat, dilakukan proses *data augmentation* guna menambah keberagaman *dataset*. Metode *augmentation* yang diterapkan mencakup operasi morfologis berupa erosi dan dilasi, disertai pemberian efek *Gaussian Blur* guna merepresentasikan beragam kondisi citra. Langkah ini dilakukan agar model lebih tangguh dalam menghadapi *noise* dan perbedaan bentuk tulisan. Contoh *dataset* karakter Jepang yang digunakan dalam penelitian ini ditampilkan pada Gambar 3.

Hasil proses tersebut menghasilkan masing-masing 500 citra untuk setiap kelas, yang terdiri atas 46 karakter *hiragana* dan 46 karakter *katakana*. Dengan demikian, total *dataset* yang digunakan dalam penelitian ini adalah 46.000 citra dari 92 kelas.

2.2 Pemrosesan Data

Proses pengolahan data dilaksanakan guna menyiapkan citra sebelum memasuki tahap pelatihan model. Pada tahapan ini, semua citra diatur menjadi ukuran 64×64 piksel supaya mempunyai dimensi yang konsisten dan memenuhi kebutuhan masukan model CNN. Keseragaman ukuran tersebut dilakukan untuk menyederhanakan komputasi sekaligus mengoptimalkan efisiensi pelatihan model.

Berikutnya, nilai piksel pada citra dinormalisasi dengan menyesuaikan rentang nilainya ke skala 0 sampai 1. Tahap ini dilakukan untuk mempercepat tercapainya konvergensi ketika pelatihan berlangsung serta mempertahankan kestabilan model saat mempelajari pola data. Selain itu, citra juga disesuaikan dengan menambahkan dimensi kanal (*channel*) se-



Gambar 4. Struktur Pembagian Dataset

hingga data dapat diproses dengan baik oleh arsitektur CNN yang membutuhkan format *input* tertentu.

2.3 Pembagian Data

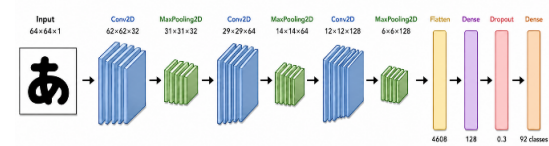
Langkah berikutnya ialah melakukan pemisahan *dataset* untuk kebutuhan pelatihan serta penilaian model. *Dataset* tersebut dikelompokkan menjadi tiga bagian, yaitu data *training*, *validation*, dan *testing* dengan rasio 70:15:15. Data *training* dimanfaatkan untuk membentuk kemampuan model dalam mengenali pola data, sementara data *validation* digunakan guna menilai performa model selama pelatihan berlangsung serta menghindari kondisi *overfitting*. Di sisi lain, data *testing* dimanfaatkan untuk mengevaluasi performa final model pada data yang belum pernah diproses sebelumnya. Struktur pembagian *dataset* ditampilkan pada Gambar 4.

Alokasi *dataset* dilakukan dengan perbandingan yang sesuai sehingga masing-masing bagian mempunyai persebaran data yang merata. Dengan pembagian ini, model diharapkan mampu memiliki kemampuan generalisasi yang baik serta menghasilkan prediksi yang akurat ketika dihadapkan pada data baru.

2.4 Pembuatan Model CNN

Perancangan arsitektur *Convolutional Neural Network* (CNN) pada penelitian ini dilakukan melalui penyusunan sejumlah lapisan yang saling terintegrasi untuk mengekstraksi fitur citra dan melakukan proses klasifikasi karakter. Arsitektur yang digunakan merupakan *custom CNN* yang dibangun secara mandiri sesuai kebutuhan penelitian. Model yang digunakan merupakan arsitektur *custom CNN* yang dibangun dan dilatih dari awal tanpa menggunakan arsitektur maupun bobot *pretrained* seperti *VGG16*, *ResNet*, atau *MobileNet*. Arsitektur yang diterapkan mencakup gabungan lapisan konvolusi, *pooling*, *flatten*, *fully connected*, serta *dropout*. Gambar 5 menampilkan rancangan *custom* arsitektur CNN yang digunakan dalam penelitian.

Sebagaimana ditampilkan pada Gambar 5, citra masukan berupa gambar $64 \times 64 \times 1$ diproses melalui tiga lapisan *Conv2D* dengan jumlah filter berturut-turut sebanyak 32, 64, dan 128 menggunakan kernel 3×3 serta fungsi aktivasi *ReLU*. Setiap lapisan konvolusi diikuti oleh *MaxPooling2D* berukuran 2×2 untuk mengurangi dimensi *feature map* dan beban komputasi. Proses tersebut menghasilkan perubahan uku-



Gambar 5. Custom Arsitektur CNN

Tabel 1. Arsitektur Model CNN

Type	Neuron Size	Activation Function	Output
Conv 2D	32 filter (3×3)	ReLU	62×62×32
Max Pooling 2D	Pool size (2×2)	–	31×31×32
Conv 2D	64 filter (3×3)	ReLU	29×29×64
Max Pooling 2D	Pool size (2×2)	–	14×14×64
Conv 2D	128 filter (3×3)	ReLU	12×12×128
Max Pooling 2D	Pool size (2×2)	–	6×6×128
Flatten	–	–	4608
Dense	128 neuron	ReLU	128
Dropout	0.3	–	128
Dense (Output)	92 class	Softmax	92

ran *feature map* secara bertahap.

Setelah proses ekstraksi fitur selesai, *feature map* berukuran $6 \times 6 \times 128$ diubah menjadi vektor satu dimensi melalui lapisan *Flatten* sehingga menghasilkan 4.608 fitur. Vektor tersebut kemudian diteruskan ke lapisan *Dense* yang terdiri atas 128 *neuron* dengan fungsi aktivasi *ReLU* untuk mempelajari hubungan antarfitur yang telah diperoleh. Selanjutnya, lapisan *Dropout* dengan nilai 0,3 digunakan untuk menonaktifkan 30% *neuron* secara acak selama pelatihan guna mengurangi risiko *overfitting*. Pada bagian akhir, digunakan lapisan *Dense output* dengan 92 *neuron* dan fungsi aktivasi *Softmax* untuk menghasilkan probabilitas klasifikasi pada 92 kelas karakter *hiragana* dan *katakana*. Rincian setiap lapisan dalam *custom* arsitektur CNN tersebut disajikan pada Tabel 1.

Mengacu pada Gambar 5, rincian susunan dan perubahan ukuran data pada setiap lapisan *custom* CNN ditampilkan pada Tabel 1. Tabel tersebut memperjelas alur pemrosesan citra mulai dari *input* berukuran $64 \times 64 \times 1$ hingga menghasilkan *feature map* berukuran $6 \times 6 \times 128$. Perubahan ukuran *output* pada setiap tahap menunjukkan bahwa dimensi spasial citra berkurang secara bertahap, sedangkan jumlah kanal fitur meningkat dari 32 menjadi 64 dan 128. Dengan demikian, Tabel 1 berfungsi sebagai penjelasan numerik terhadap visualisasi arsitektur

pada Gambar 5.

Selain menunjukkan ukuran *output* setiap lapisan, Tabel 1 juga merinci perubahan data setelah tahap ekstraksi fitur. *Feature map* berukuran $6 \times 6 \times 128$ menghasilkan 4.608 nilai setelah melalui lapisan *Flatten*, kemudian diteruskan menjadi 128 representasi pada lapisan *Dense*. Tahap akhir terdiri atas 92 *neuron* keluaran yang disesuaikan dengan jumlah kelas pada *dataset*, yaitu 46 karakter *hiragana* dan 46 karakter *katakana*. Rincian tersebut memperlihatkan alur transformasi data secara lengkap dari citra masukan hingga menghasilkan keluaran klasifikasi.

Fungsi aktivasi adalah elemen krusial pada jaringan saraf yang berperan menetapkan aktif atau tidaknya *neuron* sesuai masukan yang diperoleh. Fungsi tersebut menghadirkan karakteristik nonlinier pada model sehingga dapat mengenali struktur data yang rumit. Apabila fungsi aktivasi tidak digunakan, model akan bersifat linier saja dan memiliki kemampuan terbatas dalam pemrosesan citra (Zhou dkk., 2024). Berikut adalah sejumlah fungsi aktivasi yang digunakan pada pengolahan citra:

1. *ReLU*: dimanfaatkan untuk mengonversi nilai bernilai negatif menjadi nol serta menjaga nilai positif tetap ada sehingga proses pelatihan berlangsung lebih cepat dan risiko *vanishing gradient* berkurang.
2. *Softmax*: diterapkan pada lapisan keluaran guna mengonversi nilai menjadi probabilitas pada klasifikasi multikelas.

2.5 Training Model

Pelatihan model dilaksanakan menggunakan metode optimasi *Adam* yang dipilih atas kemampuannya mempercepat pencapaian konvergensi serta mempertahankan kestabilan selama pelatihan berlangsung. Nilai *loss* digunakan sebagai dasar penyesuaian bobot selama pelatihan sehingga ketepatan prediksi model terus meningkat. Penurunan nilai *loss* menunjukkan peningkatan kemampuan model dalam menghasilkan prediksi (Zhou dkk., 2024).

Optimasi pada jaringan saraf adalah tahapan penyesuaian bobot model guna meningkatkan ketepatan hasil prediksi. Salah satu teknik optimasi yang populer ialah *Adam* (*Adaptive Moment Estimation*) yang memadukan kelebihan *Momentum* dan *RMSProp* untuk mempercepat proses konvergensi. *Adam* beroperasi melalui penyesuaian *learning rate* secara adaptif pada masing-masing parameter sehingga pelatihan berlangsung lebih stabil dan hemat sumber daya (Nie dkk., 2025).

Fungsi *loss* yang diterapkan berupa *categorical crossentropy* karena cocok digunakan pada kasus klasifikasi multi kelas, termasuk identifikasi karakter *hiragana* dan *katakana*. Fungsi ini bekerja dengan membandingkan distribusi probabilitas hasil prediksi dengan label sebenarnya, sehingga mampu mem-

berikan penalti yang lebih besar terhadap prediksi yang salah. Penggunaan *categorical crossentropy* terbukti efektif dalam meningkatkan performa model klasifikasi berbasis CNN (Wu dkk., 2024). Sementara metrik evaluasinya adalah *accuracy* untuk menilai akurasi prediksi model. Model menjalani pelatihan selama 20 *epoch* dengan *batch size* 32, sehingga tahapan pembelajaran berlangsung optimal dan mudah dipantau.

Selain itu, dalam proses pelatihan juga diterapkan teknik *callback* berupa *Early Stopping* yang didasarkan pada nilai *validation loss*. Metode ini dirancang untuk mengakhiri pelatihan secara otomatis saat model tidak menunjukkan kemajuan kinerja pada data validasi, sehingga risiko *overfitting* dapat diminimalkan. Dengan pendekatan ini, diharapkan model yang dihasilkan memiliki performa yang optimal serta mampu melakukan generalisasi dengan baik terhadap data baru.

3. Hasil dan Pembahasan

Bagian ini membahas evaluasi kinerja model CNN dan implementasi sistem yang dikembangkan. Evaluasi dilakukan menggunakan *accuracy* dan *loss* untuk mengukur performa model. Implementasi sistem digunakan untuk menguji model dalam aplikasi pada platform *Streamlit*.

3.1 Evaluasi Model

Evaluasi model merupakan tahap penting dalam proses pelatihan untuk mengukur kinerja model dalam melakukan prediksi. Pada penelitian ini digunakan metrik *accuracy* dan *loss*, di mana *accuracy* mengukur tingkat ketepatan prediksi, sedangkan *loss* mengukur selisih antara hasil prediksi dengan nilai sebenarnya. *Loss function* berperan penting dalam proses pembelajaran karena digunakan untuk meminimalkan kesalahan model selama *training*.

Secara matematis, *accuracy* dapat dihitung sebagai:

$$Accuracy = \frac{\text{Correctly Classified Samples}}{\text{Total Number of Samples}} \quad (1)$$

Sedangkan *loss* menggunakan *categorical crossentropy*:

$$Loss = - \sum_{i=1}^n y_i \log(\hat{y}_i) \quad (2)$$

Dimana y_i adalah label sebenarnya dan $\hat{y}_i$ adalah probabilitas prediksi. Fungsi tersebut umum digunakan pada klasifikasi multi-kelas karena mampu mengukur perbedaan distribusi probabilitas secara efektif.

Proses pelatihan model dilakukan hingga mencapai *epoch* ke-15, meskipun sebelumnya ditetapkan hingga 20 *epoch*. Kondisi tersebut terjadi akibat penggunaan teknik *early stopping* yang mengakhiri proses pelatihan secara otomatis ketika *validation loss* tidak

Tabel 2. Hasil Evaluasi Model

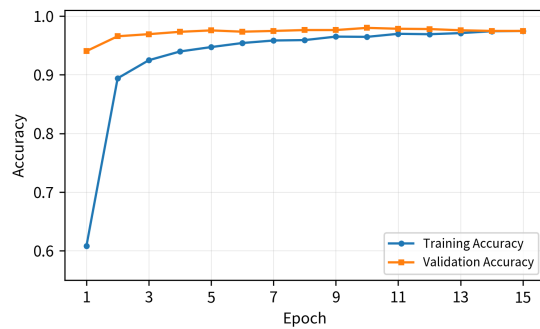
Epoch	Acc	Loss	Val Acc	Val Loss
1	0.6086	15.161	0.9404	0.2086
2	0.8941	0.3284	0.9657	0.1130
3	0.9249	0.2230	0.9693	0.0933
4	0.9398	0.1742	0.9733	0.0819
5	0.9473	0.1507	0.9757	0.0800
6	0.9542	0.1298	0.9735	0.0781
7	0.9585	0.1163	0.9748	0.0698
8	0.9592	0.1082	0.9763	0.0752
9	0.9651	0.0973	0.9763	0.0704
10	0.9647	0.0961	0.9800	0.0640
11	0.9699	0.0805	0.9785	0.0663
12	0.9692	0.0817	0.9780	0.0601
13	0.9711	0.0759	0.9759	0.0624
14	0.9743	0.0691	0.9748	0.0740
15	0.9747	0.0697	0.9748	0.0730

mengalami perbaikan. Oleh sebab itu, model yang diperoleh menjadi model paling optimal berdasarkan kinerja validasi. Temuan evaluasi tersaji pada Tabel 2.

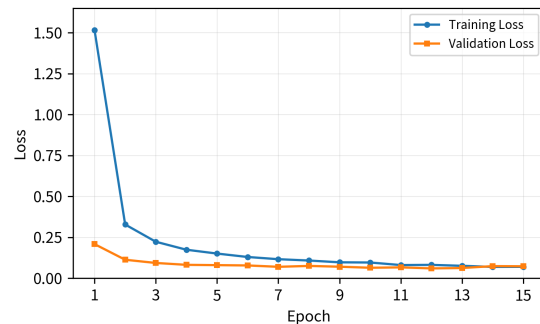
Mengacu pada Tabel 2, kinerja model menunjukkan perkembangan yang sangat besar pada *epoch* awal, ditandai oleh peningkatan nilai *accuracy* dari 0.6086 hingga melampaui 0.97 pada *epoch* ke-15. Nilai *loss* turut menurun secara berkelanjutan dari 15.161 hingga sekitar 0.0697, yang menandakan bahwa model semakin efektif dalam menekan kesalahan prediksi. Di samping itu, nilai *validation accuracy* relatif tetap berada di atas 0.97, dengan capaian maksimum sebesar 0.9800 pada *epoch* ke-10. Temuan tersebut mengindikasikan bahwa model mempunyai kapasitas generalisasi yang kuat serta tidak menghadapi *overfitting* yang berarti selama pelatihan berlangsung. Perkembangan nilai *accuracy* selama proses pelatihan divisualisasikan pada Gambar 6.

Jika dibandingkan dengan penelitian terdahulu, capaian *accuracy* sebesar 97,47% pada penelitian ini menunjukkan performa yang kompetitif. Willyanto dkk. (2021) memperoleh akurasi 95,6% pada identifikasi tulisan tangan *hiragana* menggunakan CNN berarsitektur *VGG-16*, sedangkan Nurhidayat dkk. (2025) melaporkan akurasi 95,95% pada pendeteksian tulisan *kanji* Jepang menggunakan CNN. Perbedaan capaian tersebut dapat dipengaruhi oleh karakteristik *dataset*, jumlah kelas, proses augmentasi, dan kompleksitas bentuk karakter yang dikenali. Pada penelitian ini, penggunaan 46.000 citra dari karakter *hiragana* dan *katakana* dengan pembagian data *training*, *validation*, dan *testing* yang terpisah membantu model mempelajari variasi bentuk karakter secara lebih stabil.

Hasil ini juga sejalan dengan Handoko dkk. (2024) yang menunjukkan bahwa CNN mampu memberikan kinerja baik dalam pengenalan aksara berba-



Gambar 6. Evaluasi Accuracy Model



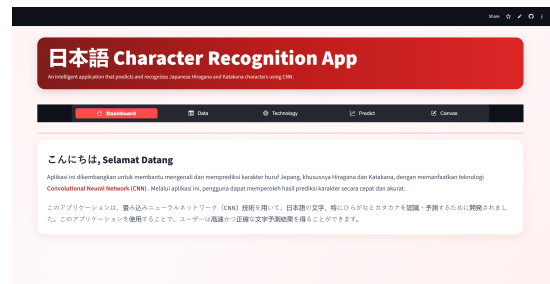
Gambar 7. Evaluasi Loss Model

sis citra. Stabilitasnya *validation accuracy* di atas 0,97 menunjukkan bahwa model tidak hanya mampu menghafal pola pada data *training*, tetapi juga memiliki kemampuan generalisasi terhadap data validasi. Penggunaan *dropout* dan *early stopping* turut berperan dalam mengurangi risiko *overfitting*, sehingga model tetap stabil meskipun proses pelatihan dihentikan sebelum mencapai *epoch* maksimum.

Grafik pada Gambar 6 menunjukkan perkembangan nilai *accuracy* selama proses pelatihan. Terlihat bahwa *accuracy* pada data *training* meningkat secara bertahap seiring bertambahnya *epoch* hingga mendekati nilai maksimal. Sementara itu, nilai *validation accuracy* cenderung stabil dan berada pada kisaran tinggi, yang mengindikasikan bahwa model mampu mempertahankan performa yang baik pada data yang tidak dilatih. Kestabilan antara kedua kurva tersebut menunjukkan bahwa model tidak mengalami *overfitting* secara signifikan.

Model yang selesai dilatih kemudian disimpan dalam bentuk berkas (.h5) dan dimanfaatkan pada tahap penerapan sistem berbasis *web* melalui *platform Streamlit*. Penyimpanan model dalam format ini memungkinkan proses pemanggilan dan penggunaan model dilakukan secara efisien tanpa perlu melakukan pelatihan ulang. Selain itu, model yang telah tersimpan dapat dengan mudah diintegrasikan ke dalam aplikasi untuk melakukan prediksi secara *real-time*.

Grafik pada Gambar 7 menunjukkan berkurangnya nilai *loss* pada data *training* maupun *validation*. Pada fase awal pelatihan, nilai *loss* berada pada tingkat yang tinggi, kemudian turun tajam pada beberapa *epoch* awal dan selanjutnya berangsur menurun hingga mencapai keadaan stabil. Pola tersebut mengindikasikan bahwa model dapat men-



Gambar 8. Halaman Awal Sistem

genali karakteristik data secara efektif serta sedikit demi sedikit menekan kesalahan prediksi. Nilai *validation loss* yang cukup rendah dan tetap stabil turut menunjukkan bahwa model mampu mempertahankan performa yang seragam pada data baru.

Penurunan nilai *loss* pada data *training* dan *validation* memperkuat interpretasi bahwa proses optimasi berjalan efektif. Temuan ini mendukung uraian Wu dkk. (2024) bahwa *categorical crossentropy* sesuai digunakan pada klasifikasi multi-kelas karena mampu memberikan penalti terhadap prediksi yang salah. Selain itu, penggunaan fungsi aktivasi *ReLU* dan *Softmax* dalam arsitektur CNN juga mendukung proses pembelajaran fitur citra secara bertahap sebagaimana dijelaskan oleh Zhou dkk. (2024). Dengan demikian, kombinasi arsitektur CNN, *optimizer Adam*, fungsi aktivasi, dan *loss function* yang digunakan pada penelitian ini dapat dinilai relevan untuk kasus klasifikasi karakter *hiragana* dan *katakana*.

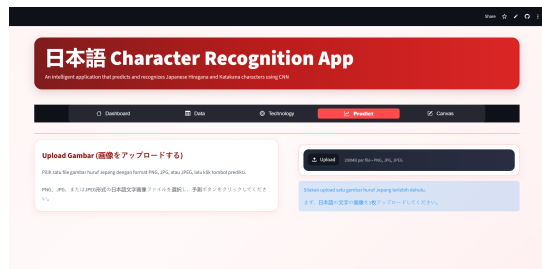
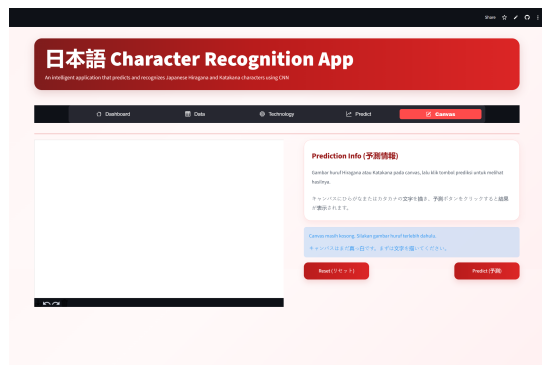
3.2 Implementasi Sistem

Penerapan sistem dalam penelitian ini dilaksanakan dengan menggunakan *platform Streamlit* sebagai sarana antarmuka berbasis web. *Platform* ini dipilih karena mampu menampilkan model secara interaktif dan mudah digunakan. Dengan demikian, pengguna dapat melakukan pengujian model secara langsung melalui tampilan yang sederhana dan responsif.

Pada Gambar 8 ditampilkan halaman awal sistem yang berisi informasi umum mengenai aplikasi pengenalan karakter *hiragana* dan *katakana*. Sistem turut dibekali menu navigasi yang membantu pengguna mengakses beragam fitur yang terdapat pada sistem.

Selain itu, sistem menyediakan fitur untuk melakukan prediksi karakter dengan dua metode, yaitu melalui unggahan gambar maupun secara langsung menggunakan fitur *canvas*. Pengguna dapat memilih metode sesuai kebutuhan, baik dengan menggunakan citra yang telah tersedia maupun dengan menggambar karakter secara langsung. Melalui fitur tersebut, pengguna dapat melakukan pengujian secara *real-time* sehingga hasil prediksi dapat diperoleh dengan cepat dan interaktif.

Berdasarkan Gambar 9, terlihat halaman yang digunakan untuk melakukan prediksi melalui proses

Gambar 9. Halaman *Upload Gambar*Gambar 10. Halaman Gambar di *Canvas*Gambar 11. Hasil Prediksi *Upload Gambar*

upload citra. Pada sistem tersebut, pengguna dapat menentukan berkas gambar yang akan dianalisis, kemudian sistem mengolah citra tersebut memakai model CNN yang sudah melalui pelatihan. Luaran prediksi kemudian ditampilkan pada gambar setelah tahapan klasifikasi berakhir.

Pada Gambar 10 diperlihatkan halaman *canvas* yang memungkinkan pengguna untuk menggambar karakter secara langsung. Melalui tampilan pada gambar tersebut, sistem akan menangkap hasil goresan pengguna sebagai *input* untuk diproses oleh model CNN. Hasil prediksi kemudian ditampilkan berdasarkan gambar yang telah dibuat secara *real-time* oleh pengguna.

3.3 Prediksi Gambar

Proses prediksi dilakukan untuk menguji kemampuan model CNN dalam mengenali karakter *hiragana* dan *katakana* berdasarkan *input* yang diberikan. Pengujian dilakukan menggunakan dua metode, yaitu melalui gambar yang diunggah serta melalui gambar yang dibuat secara langsung pada *canvas*. Hasil prediksi ditampilkan dalam bentuk label karakter beserta informasi tipe huruf yang dikenali oleh sistem.

Gambar 12. Hasil Prediksi Gambar di *Canvas*

Pada Gambar 11 ditampilkan hasil prediksi dari citra yang diunggah oleh pengguna. Gambar tersebut menunjukkan bahwa sistem mampu mengenali karakter dengan cukup akurat, seperti huruf *hiragana* “を” yang berhasil diprediksi dengan benar. Selain itu, sistem juga menampilkan informasi jenis huruf serta contoh karakter yang serupa sebagai referensi tambahan bagi pengguna. Kondisi ini mengindikasikan bahwa model CNN yang digunakan mempunyai kapabilitas tinggi untuk mengelompokkan karakter berdasarkan citra yang diberikan.

Pada Gambar 12 diperlihatkan hasil prediksi berdasarkan *input* yang dibuat secara langsung melalui fitur *canvas*. Gambar tersebut menunjukkan bahwa sistem mampu mengenali karakter *katakana* “タ” meskipun dibuat secara manual oleh pengguna. Proses tersebut mengindikasikan bahwa model tidak terbatas pada citra statis, melainkan dapat mengidentifikasi pola tulisan yang digambar secara *real time*. Dengan demikian, fitur ini membuktikan bahwa sistem memiliki fleksibilitas yang tinggi dalam menerima berbagai jenis *input* untuk proses klasifikasi.

Sesudah sistem klasifikasi karakter Jepang berhasil diterapkan pada platform *Streamlit*, langkah berikutnya berupa pengujian kegunaan atau *usability testing* dengan metode *System Usability Scale* (SUS). Pengujian ini dimaksudkan untuk mengukur tingkat kemudahan penggunaan sistem, kejelasan antarmuka, serta kenyamanan pengguna ketika melakukan prediksi karakter *hiragana* dan *katakana* melalui fitur *upload* gambar maupun *canvas*. Pertanyaan yang digunakan dalam pengujian *usability* disajikan pada Tabel 3.

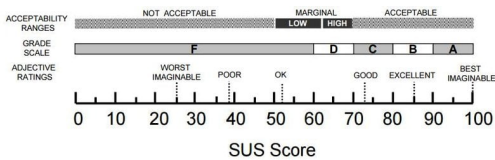
Berdasarkan Tabel 3, pengujian *usability* menggunakan metode *System Usability Scale* (SUS) memanfaatkan 10 item pertanyaan yang dinilai melalui skala Likert 1 sampai 5. Pada skala tersebut, nilai 1 menggambarkan tingkat persetujuan paling rendah, sedangkan nilai 5 menunjukkan tingkat persetujuan paling tinggi (Alfansuri dkk., 2024). Selanjutnya, skor SUS yang diperoleh ditafsirkan berdasarkan *SUS Score Percentile Rank*, yakni *Grade A* jika skor $\geq 80,3$, *Grade B* jika skor ≥ 74 dan $< 80,3$, *Grade C* jika skor ≥ 68 dan < 74 , *Grade D* jika skor ≥ 51 dan < 68 , serta *Grade F* jika skor < 51 (Putra dkk., 2025). Klasifikasi interpretasi tersebut disajikan pada Gambar 13.

Pengujian *usability* dilaksanakan melalui distribusi kuesioner *System Usability Scale* (SUS)

Tabel 3. Pertanyaan Pengujian *Usability*

No	Pertanyaan	Skala
1	Saya merasa akan menggunakan sistem klasifikasi karakter Jepang ini apabila diperlukan.	1–5
2	Saya merasa sistem klasifikasi karakter hiragana dan katakana ini terlalu rumit untuk digunakan.	1–5
3	Saya merasa sistem ini mudah digunakan untuk mengenali karakter hiragana dan katakana.	1–5
4	Saya merasa membutuhkan bantuan teknis dari orang lain untuk dapat menggunakan sistem klasifikasi ini.	1–5
5	Saya merasa fitur pada sistem, seperti upload gambar dan canvas, sudah terintegrasi dengan baik.	1–5
6	Saya merasa tampilan dan fungsi pada sistem klasifikasi karakter Jepang ini belum konsisten.	1–5
7	Saya merasa sebagian besar pengguna dapat mempelajari cara menggunakan sistem ini dengan cepat.	1–5
8	Saya merasa sistem klasifikasi karakter hiragana dan katakana ini merepotkan untuk digunakan.	1–5
9	Saya merasa percaya diri saat menggunakan sistem ini untuk melakukan prediksi karakter Jepang.	1–5
10	Saya merasa perlu mempelajari banyak hal terlebih dahulu sebelum dapat menggunakan sistem klasifikasi karakter Jepang ini.	1–5

Sumber: diadaptasi dari (Putra dkk., 2025)



Gambar 13. SUS Score Percentile Rank

kepada pengguna sistem klasifikasi. Berdasarkan hasil distribusi kuesioner tersebut, sebanyak 18 responden memberikan evaluasi terhadap tingkat kegunaan sistem. Hasil evaluasi dari 18 responden tersebut ditampilkan pada Tabel 4.

Merujuk pada Tabel 4, kalkulasi *System Usability Scale* (SUS) yang melibatkan 18 responden menghasilkan skor rerata 68,61. Berdasarkan penafsiran *SUS Score* pada Gambar 13, skor tersebut tergolong *marginal high*, masuk ke Grade C, serta menempati kisaran evaluasi OK atau hampir mencapai *good*. Hasil ini menunjukkan bahwa sistem klasifikasi karakter Jepang yang dikembangkan sudah cukup dapat digunakan oleh pengguna, namun masih memerlukan beberapa perbaikan agar tingkat kegunaannya dapat meningkat ke kategori *acceptable* dan penilaian *good*.

Skor SUS sebesar 68,61 menunjukkan bahwa sistem berada pada tingkat kegunaan yang cukup, tetapi belum termasuk kategori sangat baik. Inter-

Tabel 4. Hasil Penilaian *Usability*

Responden	Jumlah Skor	Nilai SUS
1	28	70
2	27	67,5
3	29	72,5
4	26	65
5	26	65
6	27	67,5
7	27	67,5
8	30	75
9	28	70
10	26	65
11	29	72,5
12	27	67,5
13	28	70
14	25	62,5
15	30	75
16	27	67,5
17	29	72,5
18	25	62,5
Rata-rata	27,44	68,61

pretasi ini sejalan dengan Putra dkk. (2025) yang menempatkan skor 68 sampai kurang dari 74 pada Grade C, serta mendukung Alfansuri dkk. (2024) bahwa pengujian SUS dapat digunakan untuk mengidentifikasi aspek antarmuka yang perlu diperbaiki. Berdasarkan hasil tersebut, sistem sudah layak digunakan sebagai media pendukung pembelajaran, namun peningkatan kualitas tampilan, kejelasan instruksi penggunaan, keterbacaan hasil prediksi, dan respons fitur *canvas* masih perlu dilakukan agar pengalaman pengguna meningkat.

4. Kesimpulan dan Saran

Mengacu pada temuan penelitian, model *custom Convolutional Neural Network* (CNN) yang dirancang berhasil mengidentifikasi karakter *hiragana* dan *katakana* dengan tingkat kinerja yang sangat optimal. Data penelitian dikumpulkan secara mandiri dengan mengubah teks menjadi citra menggunakan 45 jenis *font* bahasa Jepang, kemudian dilakukan augmentasi sehingga diperoleh total 46.000 gambar. Proses pelatihan menunjukkan peningkatan *accuracy* hingga mencapai 97,47% dengan nilai *validation accuracy* tertinggi sebesar 98,00% pada *epoch* ke-10. Di samping itu, nilai *loss* terus berkurang secara konsisten hingga menyentuh angka 0,0697, yang mengindikasikan bahwa model dapat menekan tingkat kekeliruan prediksi secara efektif. Penerapan sistem pada *platform Streamlit* turut sukses menyajikan keluaran prediksi secara interaktif melalui fasilitas *upload* gambar dan *canvas*.

Hasil evaluasi *usability* melalui metode *System Usability Scale* (SUS) yang melibatkan 18 re-

sponden menghasilkan skor rerata 68,61. Mengacu pada penafsiran *SUS Score*, skor tersebut tergolong kategori *marginal high*, masuk *Grade C*, dan menempati kisaran evaluasi OK. Kondisi tersebut mengindikasikan bahwa sistem dapat dimanfaatkan oleh pengguna sebagai sarana penunjang pembelajaran bahasa Jepang serta identifikasi karakter berbasis *deep learning*. Meskipun demikian, studi ini masih menyimpan sejumlah keterbatasan, yakni *dataset* yang dimanfaatkan masih dihasilkan secara otomatis melalui beragam variasi *font*, belum sepenuhnya terintegrasi dengan data tulisan tangan asli pengguna, serta belum mencakup karakter dengan tanda *dakuten* dan *handakuten*. Selain itu, arsitektur model CNN yang digunakan masih terbatas sehingga masih berpeluang untuk dikembangkan dengan arsitektur yang lebih kompleks agar kemampuan klasifikasi menjadi lebih optimal.

Daftar Pustaka

- Alfansuri, H. M., Putra, P. S., & Zunaidi, R. A. (2024): *Innovative Design of ITTS Mart Application with Design Thinking & System Usability Scale Method*, Sinkron, 8(3), 1369–1383. <https://doi.org/10.33395/sinkron.v8i3.13631>
- Ardi Wijaya, Daryono, B. R., Rozali Toyib, & Yovi Apriadiansyah. (2024): *Analisis Pengenalan Pola Pada Citra Digital Untuk Prediksi Berat Buah Sawit*, Decode: Jurnal Pendidikan Teknologi Informasi, 4(3), 713–724. <https://doi.org/10.51454/decode.v4i3.481>
- Handoko, A. A., Rosid, M. A., & Indahyanti, U. (2024): *Implementasi Convolutional Neural Network (CNN) Untuk Pengenalan Tulisan Tangan Aksara Bima*, Smatika Jurnal, 14(01), 96–110. <https://doi.org/10.32664/smatika.v14i01.1196>
- Hidayanti, S. A., Sholikhah, S. N. I., Abdurrahman, I. R., Feryanto, N. J., Saragih, B. C., Rachman, I. fadhila, Dinillah, H. F., & Lestari, H. R. (2024): *Analisis Pengaruh Pembelajaran Bahasa Jepang Terhadap Minat Mahasiswa Bidang Informatika Untuk Bekerja Di Jepang*, Jurnal Insan Pendidikan dan Sosial Humaniora, 2(3), 190–200. <https://doi.org/10.59581/jipsoshum-widyakarya.v2i3.3340>
- Mutiara, K. M. S., Yeni, & Sadyana, I. W. (2025): *Penerapan Model Pembelajaran Kooperatif Numbered Heads Together (NHT) Berbantuan Media Kartu Papan Interaktif untuk Meningkatkan Penguasaan Huruf Hiragana Siswa kelas X PH 1 SMK Pariwisata Banyuwatis*, Pustaka: Jurnal Ilmu-Ilmu Budaya, 25(2), 204–209. <https://doi.org/10.24843/pjiib.2025.v25.i02.p06>
- Nie, Y., Chen, Y., Guo, J., Li, S., Xiao, Y., Gong, W., & Lan, R. (2025): *An improved CNN model in image classification application on water turbidity*, Scientific Reports, 15(1). <https://doi.org/10.1038/s41598-025-93521-4>
- Nurhidayat, A., Satriatama, F., Mayanti, E., & Samsinar, R. (2025): *Pendeteksi Tulisan Bahasa Kanji Jepang dengan Metode Convolutional Neural Network (CNN)*, Prosiding Seminar Nasional Hukum, Bisnis, Sains Dan Teknologi, 5, 314–321. <https://www.ojs.uib.ac.id/HUBISINTEK/article/view/4762>
- Nurmaini, S., Darmawahyuni, A., Sapitri, A. I., Rachmatullah, M. N., Firdaus, & Tutuko, B. (2021): *Pengenalan Deep Learning dan Implementasinya*, Unsri Press.
- Purnamasari, I. A. P. A., Suartini, N. N., & Mardani, D. M. S. (2023): *Peningkatan Penguasaan Huruf Hiragana dan Katakana pada Siswa Kelas XI IBB di SMA Negeri 1 Singaraja Berbantuan Media Aplikasi Memory Hint dan Obenkyo*, Jurnal Penelitian Mahasiswa Indonesia, 3(2), 469–475. <https://doi.org/10.36663/jpmi.v3i2.564>
- Putra, F. A. A., Waluyo, B., Faturrohman, R., Purwoprasetyo, W. D., & Setiawan, I. (2025): *Analisis Usability Testing Menggunakan Metode System Usability Scale terhadap Kepuasan Pengguna Website Kemahasiswaan Universitas Amikom Purwokerto*, Uranus: Jurnal Ilmiah Teknik Elektro, Sains dan Informatika, 3(1), 121–130. <https://doi.org/10.61132/uranus.v3i1.673>
- Rasyiq, F. R., Marsiani, E. S., & Ali, H. N. (2023): *Aplikasi Pembelajaran Bahasa Jepang Dasar (Hiragana dan Katakana) untuk Pelajar Berbasis Android*, Jurnal Rekayasa Komputasi Terapan, (2), 2776–5873. <https://doi.org/10.30998/jrkt.v3i02.8598>
- Setiadi, I., Widiyanti, S., & Afratofani, A. A. (2024): *Aplikasi Pengelanaan Huruf Hiragana dan Katakana Jepang Berbasis Android Menggunakan Kodular*, Jurnal Penelitian Sistem Informasi, 2, 20–35. <https://doi.org/10.54066/jpsi.v2i2.2027>
- Tensorflow. (2026): *TensorFlow: An end-to-end opensource platform for machine learning.*, <https://www.tensorflow.org/>
- Willyanto, A., Alamsyah, D., & Irsyad, H. (2021): *Identifikasi Tulisan Tangan Aksara Jepang Hiragana Menggunakan Metode CNN Arsitektur VGG-16*, Jurnal Algoritme, 2(1), 1–11. <https://doi.org/10.35957/algoritme.v2i1.1450>
- Wu, M., Zhou, J., Peng, Y., Wang, S., & Zhang, Y. (2024): *Deep Learning for Image Classification: A Review* (hlm. 352–362). https://doi.org/10.1007/978-981-97-1335-6_31
- Zhou, M., Jie, W., Tang, F., Zhang, S., Mao, Q., Liu, C., & Hao, Y. (2024): *Deep learning algorithms for classification and detection of recurrent aphthous ulcerations using oral clinical photographic images*, Journal of Dental Sciences, 19(1), 254–260. <https://doi.org/10.1016/j.jds.2023.04.022>

Halaman ini sengaja dikosongkan