**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Bahasa Jepang memiliki tiga sistem penulisan yang digunakan sehari – hari, yaitu hiragana, katakana, dan Kanji. Diantara ketiga sistem tersebut, kanji memiliki peran yang penting dalam tata bahasa jepang karena digunakan untuk merepresentasikan makna kata secara lebih tepat dan efisien. Secara keseluruhan terdapat puluhan ribu karakter kanji, namun hanya ada 2.136 kanji yang dikategorikan sebagai Jōyō Kanji, yakni daftar kanji standar yang digunakan dalam kehidupan sehari-hari.

Tata cara penulisan kanji tersusun atas goresan–goresan dasar yang disebut *stroke*. Jumlah dan urutan penulisan stroke harus diperhatikan dengan saksama agar dapat dipahami dengan benar. Beberapa kanji sederhana hanya terdiri atas dua hingga lima *stroke*, sementara kanji kompleks seperti *ryū* yang berarti naga dapat mencapai lebih dari 15 *stroke*. Semakin banyak jumlah *stroke* pada suatu kanji, semakin sulit pula kanji tersebut untuk diingat, mengingat banyaknya kanji yang mirip, para pelajar kanji seringkali salah membedakan maknanya [1]. Kompleksitas ini menjadikan kanji sebagai tantangan besar, baik dalam proses pembelajaran maupun dalam aplikasi komputasi seperti pengenalan tulisan tangan [2], [3], [4], [5].

Untuk mengatasi permasalahan dalam pembelajaran kanji, berbagai penelitian telah dilakukan sebelumnya, beberapa di antaranya meliputi pengenalan tulisan tangan karakter kanji menggunakan Ensemble of Convolutional Neural Network [5] dan penggabungan image processing dan deep learning untuk pengorganisasian ulang buku-–buku Jepang kuno [6]. Berbeda dengan penelitian – penelitian tersebut, penelitian ini berfokus pada pengembangan sebuah model *Artificial Intelligence* (AI) yang mampu mengukur tingkat kemiripan visual antarkarakter kanji. Dengan demikian, model dapat dengan mudah mengidentifikasi karakter kanji lain yang memiliki bentuk serupa dengan suatu karakter kanji tertentu.

Convolutional Neural Network (CNN) dipilih sebagai solusi utama dalam penelitian ini. Dalam beberapa tahun terakhir, perkembangan AI, khususnya di bidang *computer vision*, telah memberikan kontribusi signifikan dalam pengenalan karakter kanji [7]. Model-model CNN telah digunakan secara luas untuk mengenali berbagai karakter bahasa dalam berbagai aplikasi, mulai dari pengenalan tulisan tangan hingga pelabelan otomatis [8], [9]. *Computer vision* berfokus pada pemrosesan dan analisis citra, sehingga memungkinkan komputer memahami struktur stroke pada karakter kanji.

Dataset yang digunakan dalam penelitian ini adalah KanjiVG yang berisi ribuan kanji dari JIS level satu dan dua dengan format *Scalable Vector Graphics* (SVG) dengan jumlah data setiap kanji yang relatif terbatas. KanjiVG berfokus pada *stroke* dari setiap karakter sehingga sesuai untuk penelitian yang menekankan pada struktur goresan. Namun, dataset ini memiliki kekurangan, yaitu jumlah data yang tidak banyak. Untuk mengatasi keterbatasan jumlah data, penelitian ini menggunakan konsep *Few-Shot Learning* (FSL) agar model dapat berjalan sesuai dengan keinginan. Selain itu, *Siamese Neural Network* (SNN) yang berbasis CNN juga digunakan karena dapat mendukung model untuk mencari *similarity* pada kanji. SNN berisi dua *neural network* yang sejenis dan saling berbagi bobot yang sama. Dengan berbagi bobot, kedua jaringan ini mampu menghasilkan representasi vektor yang konsisten, sehingga cocok untuk perhitungan nilai kesamaan (*similarity score*) antara dua input [10].

Penelitian ini menggunakan python sebagai bahasa pemrograman utama karena python memiliki banyak *library* yang mendukung pembelajaran mesin, khususnya di bidang *computer vision*, seperti Pandas, NumPy, PyTorch, dan Matplotlib. Platform Google Colab digunakan sebagai *notebook* untuk mengeksekusi kode, karena menyediakan akses GPU dan *environtmet* yang nyaman, sehingga cocok untuk menjalankan model AI yang memiliki beban komputasi berat, seperti dataset citra.

Algoritma yang digunakan untuk SNN adalah EfficientNet-B0 karena arsitektur ini dikenal memiliki kinerja bagus dan beban komputasi yang tidak terlalu berat. Berdasarkan studi komparatif yang telah dilakukan, EfficientNet-B0 secara konsisten menunjukkan akurasi yang kompetitif dengan jumlah parameter yang jauh lebih sedikit dibandingkan arsitektur konvensional seperti *ResNet50* [11].

* 1. **Rumusan Masalah**

Berdasarkan latar belakang masalah yang sudah dipaparkan, maka dapat dirumuskan beberapa masalah yang perlu diperhatikan:

1. Bagaimana model dapat membedakan pola-pola *stroke* yang ada pada kanji?
2. Mengapa diperlukan penerapan FSL dan arsitektur SNN untuk mengukur kemiripan visual antar karakter kanji di dataset kanjiVG?
3. Bagaimana tingkat akurasi EfficientNet-B0 dengan arsitektur SNN untuk pencarian kemiripan visual pada kanji?
   1. **Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah:

1. Mengetahui bagaimana model mampu mengidentifikasi perbedaan pola *stroke* pada kanji.
2. Mengembangkan sistem berbasis kecerdasan buatan yang mampu membantu proses pencarian karakter kanji dengan mempertimbangkan tingkat kemiripan visual.
3. Mengetahui tingkat akurasi EfficientNet-B0 untuk mencari kemiripan visual pada karakter kanji.
   1. **Manfaat Penelitian**

Penelitian ini bermanfaat untuk pengembangan metode pencarian kemiripan visual antar karakter kanji menggunakan konsep FSL dengan basis SNN menggunakan algoritma EfficientNet-B0 yang tidak hanya berkontribusi pada pengembangan ilmu *computer vision* tetapi juga dapat membantu para pelajar kanji dalam membedakan serta mempelajari kanji secara lebih mudah.

* 1. **Batasan Penelitian**

Penelitian ini memiliki beberapa batasan, diantaranya:

1. Dataset yang digunakan berasal dari KanjiVG, dengan format citra (hasil konversi dari SVG ke PNG).
2. Fokus penelitian adalah pada kemiripan visual karakter kanji berdasarkan struktur *stroke*, bukan pada aspek semantik atau pelafalan (kunyomi/onyomi).
3. Arsitektur yang digunakan adalah SNN dengan model EfficientNet-B0 sebagai dasarnya.
4. Penelitian dilakukan menggunakan Google Colab dan bahasa pemrograman Python.
   1. **Sistematika Penulisan**

Penulisan Skripsi ini dibagi dalam 5 bab, adapun isi dari masing-masing bab dapat diuraikan secara singkat sebagai berikut:

BAB I: PENDAHULUAN

Bab ini berisi uraian mengenai latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, serta sistematika penulisan skripsi.

BAB II: TINJAUAN PUSTAKA

Bab ini memuat kajian teori dan penelitian terdahulu yang relevan, meliputi sistem penulisan kanji, konsep computer vision, CNN, FSL, SNN, serta arsitektur *EfficientNet*-B0 yang digunakan sebagai backbone.

BAB III: METODOLOGI PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan, termasuk deskripsi dataset KanjiVG, tahap pengolahan data, arsitektur model SNN dengan backbone *EfficientNet*-B0, skema pelatihan menggunakan FSL, serta rancangan evaluasi model.

BAB IV: HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil eksperimen yang dilakukan, mulai dari preprocessing dataset, hasil pelatihan model, evaluasi performa menggunakan metrik akurasi maupun top-k retrieval, hingga analisis efektivitas penggunaan SNN berbasis *EfficientNet*-B0 pada pencarian kemiripan visual kanji.

BAB V: PENUTUP

Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian serta saran untuk pengembangan penelitian lebih lanjut, baik dari sisi metode, dataset, maupun aplikasi praktis di bidang pembelajaran kanji dan computer vision.

# **BAB II**

# **TINJAUAN PUSTAKA**

1. **Kanji**

Karakter kanji adalah sistem penulisan digunakan dalam bahasa Jepang yang berasal dari karakter Tionghoa, oleh karena itu kanji memiliki dua cara pelafalan, yaitu kun`yomi (cara baca Jepang) dan on`yomi (cara baca Tionghoa). Kanji tidak hanya merepresentasikan bunyi, melainkan juga unsur makna dan struktur visual yang kompleks [12].

Setiap karakter kanji terdiri dari goresan (*stroke*) dan radikal (komponen dasar) yang mempengaruhi bentuk visualnya. Struktur ini menyebabkan variasi visual yang beragam antar karakter, tetapi berbeda dalam makna atau pelafalan [13]. Karena itu, pengenalan atau pencarian karakter kanji berdasarkan kemiripan visual menjadi tantangan yang menarik dalam *computer vision* dan *deep learning*.

Dalam lingkup penggunaan resmi dan pendidikan di Jepang, karakter kanji diklasifikasikan ke dalam daftar yang diatur pemerintah Jepang. Salah satu daftar yang paling dikenal dan sering dijadikan acuan adalah Jōyō kanji. Daftar ini berisi karakter kanji yang dianggap sebagai karakter yang sering digunakan dalam kegiatan sehari-hari dan wajib diketahui oleh penutur bahasa Jepang dalam konteks literasi umum. Saat ini, jumlah Jōyō kanji adalah 2.136 karakter, setelah revisi tahun 2010. 1.026 karakter diajarkan pada jenjang *sekolah dasar* (kelas 1–6), sedangkan sisanya diajarkan pada jenjang menengah (kelas 7–9) dan sekolah menengah atas [12]. Selain Jōyō kanji, ada pula karakter Jinmeiyō kanji yang digunakan untuk nama pribadi.

**2.1.1. Standar Kode dan Representasi Digital Kanji**

Agar karakter kanji dapat diproses secara digital (misalnya dalam sistem komputer atau basis data), diperlukan standar pengkodean karakter. Beberapa standar yang dipakai hingga sekarang adalah Unicode dan *Japanese Industrial Standards* (JIS). Unicode sudah menjadi standar internasional untuk merepresentasikan berbagai karakter, termasuk kanji. Sedangkan JIS adalah standar nasional Jepang untuk representasi huruf dan simbol Jepang secara digital. JIS memiliki beberapa jenis, seperti JIS X 0208 dan JIS X 0213 [14].

1. **KanjiVG Dataset**

KanjiVG merupakan dataset berbasis vektor yang berisi informasi struktur dan urutan goresan (stroke order) dari karakter kanji dalam format Scalable Vector Graphics (SVG). Dataset ini dikembangkan oleh Ulrich Apel sebagai bagian dari proyek *KanjiVG Project* pada tahun 2010 dan dipublikasikan secara terbuka di GitHub di bawah lisensi Creative Commons Attribution-ShareAlike (CC BY-SA)

Setiap file dalam KanjiVG merepresentasikan satu karakter kanji yang diberi nama sesuai dengan kode Unicode-nya. Misalnya, karakter *mizu* yang artinya air disimpan dalam 6c34.svg. File SVG tersebut berisi elemen-elemen <path> dan <g> yang mendefinisikan setiap goresan serta kelompok komponennya. Dengan demikian, KanjiVG tidak hanya menyimpan bentuk akhir suatu kanji, tetapi juga struktur hierarkis dan urutan pembentukannya.

KanjiVG terdiri dari sekitar 6.353 karakter kanji, yang mencakup Seluruh jōyō kanji (2.136 karakter), jinmeiyō kanji, JIS X 0208 Level 1 dan Level 2. KanjiVG juga mencakup beberapa kanji dari standar JIS X 0212/0213, serta sejumlah kecil karakter langka atau varian bentuk historis. Dengan cakupan tersebut, KanjiVG dapat dianggap mewakili hampir seluruh kanji yang digunakan dalam penulisan bahasa Jepang modern.

1. *Computer Vision*

Perkembangan bidang Computer Vision berawal pada dekade 1960-an, ketika penelitian mulai berfokus pada bagaimana komputer dapat mengekstraksi informasi tiga dimensi dari representasi dua dimensi sebuah objek. Pada tahap awal, penelitian di bidang ini menitikberatkan pada proses pengambilan informasi penting dari citra digital melalui pemodelan komputasional. Secara garis besar, Computer Vision memiliki dua orientasi utama. Dari sisi rekayasa, bidang ini bertujuan mengembangkan sistem otonom yang mampu melakukan tugas-tugas visual layaknya penglihatan manusia. Sementara itu, dari sisi biologis, pendekatan komputasional digunakan untuk meniru mekanisme penglihatan manusia, seperti dalam proses deteksi atau diagnosis kondisi medis melalui citra tubuh [15].

Dalam konteks teknis, Computer Vision telah diterapkan di berbagai sektor, termasuk bidang pendidikan, di mana sistem pengenalan wajah dimanfaatkan untuk mencatat kehadiran secara otomatis melalui kamera. Teknologi ini juga berkembang pesat dalam sistem kendaraan udara tanpa awak (Unmanned Aerial Vehicle atau UAV) yang menggunakan penglihatan komputer untuk melakukan pemantauan situasi secara otomatis tanpa kendali langsung manusia.  
Sementara dalam ranah biomedis, Computer Vision berperan besar dalam pencitraan medis (medical imaging), sebuah bidang yang mengalami perkembangan pesat karena kemampuannya membantu proses diagnosis dan peningkatan mutu pelayanan kesehatan [15].

Seiring perkembangan teknologi, **Computer Vision** menjadi salah satu bidang utama dalam **Artificial Intelligence (AI)** yang berfokus pada kemampuan sistem komputer untuk menafsirkan, menganalisis, dan memahami informasi visual dari dunia nyata dalam bentuk gambar maupun video. Teknologi ini memungkinkan mesin untuk meniru persepsi visual manusia melalui proses pengenalan objek, analisis citra, pelacakan gerakan, hingga interpretasi konteks visual secara otomatis [16].  
Meskipun demikian, baik penglihatan manusia maupun computer vision sama-sama berupaya menafsirkan data multidimensi secara spasial, namun sistem komputer belum sepenuhnya mampu menandingi kompleksitas dan kepekaan penglihatan manusia karena keterbatasan algoritma, variabel, dan tingkat akurasi. Tantangan utama dalam pengembangannya meliputi efektivitas algoritma, ketepatan hasil analisis, serta kemampuan sistem untuk beradaptasi secara akurat, tangguh, dan skalabel dalam berbagai kondisi. Berbagai penelitian juga telah memperluas penerapan computer vision ke berbagai bidang, seperti bantuan bagi penyandang disabilitas visual, otomasi industri, penginderaan jauh, robotika, hingga komunikasi manusia berbasis komputer, dengan efektivitas sistem yang sangat bergantung pada arsitektur dan rancangan aplikasinya [17].

Dengan demikian, *Computer Vision* dapat dipandang sebagai disiplin ilmu yang berperan penting dalam menjembatani data visual mentah menjadi **representasi bermakna** yang mendukung pengambilan keputusan otomatis dalam berbagai aplikasi — mulai dari kesehatan, transportasi, hingga pengenalan karakter Kanji dalam konteks penelitian ini.

1. ***Deep Learning***

Deep Learning merupakan cabang dari Machine Learning yang menggunakan jaringan saraf tiruan berlapis (deep neural networks) untuk mengekstrak fitur otomatis dari data mentah, seperti gambar atau video, dan mengaplikasikannya pada tugas-tugas seperti klasifikasi, deteksi objek, segmentasi, dan restorasi citra; CNN (Convolutional Neural Network) misalnya, telah digunakan secara luas dalam visi komputer karena kemampuannya dalam menangkap fitur visual dari lapisan rendah ke lapisan tinggi [18].

Secara struktural, deep learning terdiri dari beberapa lapisan tersembunyi (hidden layers) di antara lapisan input dan output, yang memungkinkan pemodelan hubungan non-linier yang kompleks dalam data; metode seperti forward propagation dan backpropagation digunakan untuk melatih model dan mengoptimalkan bobot antar lapisan agar kesalahan prediksi berkurang [19].

Kebutuhan data yang besar dan berkualitas menjadi faktor krusial pada deep learning. Tanpa dataset yang cukup besar, model-model ini sering kali mengalami overfitting atau generalisasi yang buruk; teknik augmentasi data (data augmentation) dan penggunaan model pralatih (pretrained models) sering diterapkan untuk mengatasi keterbatasan data [20].

Selain itu, tantangan utama dalam pengembangan deep learning meliputi computational cost yang tinggi, kebutuhan terhadap perangkat keras yang kuat (GPU/TPU), kebutuhan untuk arsitektur yang efisien agar model bisa digunakan pada perangkat dengan sumber daya terbatas [21], dan robustness, yaitu bagaimana model tetap mampu melakukan prediksi yang akurat ketika kondisi input berubah — misalnya adanya noise, pencahayaan berbeda, kondisi occlusion atau gangguan adversarial [22].

Berbagai penelitian terkini telah memperluas aplikasi deep learning dalam computer vision, mencakup deteksi objek, segmentasi semantik, pengolahan citra medis, dan deteksi serta klasifikasi dalam data point cloud 3D; aplikasi-aplikasi ini menunjukkan bahwa deep learning telah menjadi tulang punggung banyak sistem visi komputer modern [23].

1. ***Convolutional Neural Network* (CNN)**

Convolutional Neural Network (CNN) adalah jenis jaringan saraf tiruan dalam deep learning yang dirancang khusus untuk mengolah data yang memiliki struktur grid seperti gambar. CNN mampu mengekstrak fitur visual secara otomatis melalui operasi konvolusi (convolution), di mana filter (kernel) digeser pada citra input untuk mendeteksi pola lokal seperti tepi, garis, tekstur, atau titik-perubahan intensitas. Lapisan-lapisan konvolusi ini diikuti oleh operasi pooling atau downsampling untuk mengurangi dimensi spasial sambil mempertahankan informasi penting, kemudian diakhiri dengan lapisan fully connected atau output untuk melakukan klasifikasi atau prediksi berdasarkan fitur-fitur yang telah dipelajari, di mana CNN membangun hirarki fitur mulai dari lapisan awal yang mengenali pola sederhana hingga lapisan yang lebih dalam yang menangkap fitur abstrak [24].

**2.5.1. Evolusi & Arsitektur CNN Terkini**

Seiring berjalannya riset, arsitektur-CNN menjadi semakin kompleks dan beragam. Misalnya, peneliti menggabungkan modul-modul baru, merancang blok arsitektur yang efisien (seperti residual blocks, dense connectivity, dan blok-blok modul lainnya) agar model menjadi lebih dalam, lebih lebar, namun tetap mengusahakan agar dapat dilatih dengan biaya komputasi yang lebih terkendali [24]. Selain itu, penelitian terbaru juga membahas cara baru dalam memproses gambar resolusi rendah atau objek kecil, salah satunya dengan mengganti pooling atau strided convolution yang biasa digunakan dengan modul yang menjaga detil spasial lebih baik agar tidak kehilangan informasi penting [25].

**2.5.2. Kelebihan dan Tantangan CNN**

CNN memiliki sejumlah keunggulan yang menjadikannya salah satu arsitektur paling dominan dalam bidang *deep learning* dan *computer vision*:

1. Ekstraksi fitur otomatis: CNN dapat belajar dan mengenali pola visual tanpa memerlukan fitur manual yang rumit, sehingga sangat cocok untuk tugas-citra kompleks.
2. Invarian terhadap transformasi lokal: Karena operasi konvolusi dan pooling, CNN relatif tangguh terhadap translasi, rotasi ringan, dan variasi kecil lainnya dalam data gambar [26].
3. Performanya sangat baik dalam berbagai tugas seperti klasifikasi gambar, deteksi objek, segmentasi semantik, dan prediksi video[24].

Meskipun memiliki banyak keunggulan, CNN juga menghadapi sejumlah tantangan yang perlu diperhatikan dalam pengembangannya

1. Overfitting dan kapasitas model: Semakin dalam atau kompleks arsitektur CNN, semakin besar kemungkinan terjadi overfitting jika data latih tidak cukup atau kurang bervariasi.
2. Kebutuhan komputasi dan sumber daya: Pelatihan CNN besar biasanya memerlukan GPU/TPU dan sumber daya memori yang besar. Ini menjadi kendala khususnya pada aplikasi yang dijalankan pada perangkat dengan keterbatasan hardware/edge devices [24].
3. Penurunan performa pada gambar resolusi rendah atau objek kecil: Beberapa desain standar CNN (misalnya penggunaan pooling atau convolusi yang “strided”) bisa kehilangan detil penting, yang berdampak buruk pada deteksi objek kecil atau fitur-fitur halus. Penelitian seperti “No More Strided Convolutions or Pooling” mengatasi hal ini dengan modul alternatif yang mempertahankan informasi spasial lebih baik [25].
4. ***Few-Shot learning* (FSL)**

Few-Shot Learning adalah pendekatan dalam machine learning / deep learning yang bertujuan agar model dapat belajar dan beradaptasi pada kelas atau tugas baru dengan hanya **beberapa contoh** (biasanya sangat sedikit) sebagai data pelatihan. FSL sering dipandang sebagai metode yang jembatan antara kebutuhan data besar pada deep learning dan kondisi nyata dimana data yang tersedia terbatas .

1. One-Shot Learning: suatu kasus di mana hanya satu contoh per kelas yang tersedia.
2. Zero-Shot Learning: menangani kelas yang tidak pernah muncul dalam data pelatihan (melalui atribut atau semantik lain).
   * 1. **Metode Utama dalam FSL**
3. Meta-Learning: model dilatih dengan banyak “task” kecil sehingga bisa cepat adaptasi ke task baru yang hanya memiliki sedikit data. Contohnya episodic training dan pembelajaran cara belajar.
4. Metric-Learning / Embedding-based Methods: model membuat representasi fitur (embedding) di mana contoh-contoh dari kelas yang sama lebih dekat, dan contoh dari kelas berbeda lebih jauh; menggunakan prototipe, jarak-jarak (misalnya Euclidean), atau fungsi jarak khusus.
5. Data Augmentation dan Transfer Learning: memperluas data latih dengan augmentasi, atau menggunakan model pra-latih (pretrained) dari domain lain dan disesuaikan ke domain baru dengan sedikit data.
6. Optimisasi dan Regularisasi: teknik untuk mencegah overfitting ketika data sangat terbatas; misalnya variational autoencoders untuk generative augmentation, regularization, dropout, dan lain-lain.
   * 1. **Kelebihan dan Tantangan FSL**

Few-Shot Learning (FSL) memiliki sejumlah keunggulan yang menjadikannya pendekatan menarik dalam kondisi keterbatasan data. Beberapa kelebihannya antara lain:

* Efisiensi data: model tidak memerlukan dataset besar, sangat membantu ketika label sulit atau mahal diperoleh.
* Adaptabilitas: model FSL mampu cepat beradaptasi ke kelas atau tugas baru setelah pelatihan meta atau pra-pelatihan.
* Generalitas: beberapa metode FSL menunjukkan kemampuan untuk mentransfer pengetahuan antar domain, memperluas penggunaan model ke konteks yang berbeda dari pelatihan asal.

Meskipun menawarkan beberapa kelebihan, FSL juga menghadapi sejumlah tantangan yang membatasi kinerjanya dalam praktik. Beberapa tantangan utama yang sering ditemui meliputiOverfitting: model mudah mengalami overfitting karena jumlah contoh pelatihan yang sangat sedikit.

* Representasi yang kurang kuat: embedding atau fitur yang dipelajari mungkin tidak menangkap variasi penting karena keterbatasan data.
* Perbedaan domain (domain shift): tantangan ketika data tugas baru berbeda distribusinya dari data pelatihan (sumber), yang bisa menurunkan performa.
* Ketidakstabilan dalam performa: kinerja model FSL bisa sangat dipengaruhi oleh pemilihan contoh pelatihan (sampling), ukuran variasi antar kelas, dan kesiapan model pra-latih.

1. *Siamese Neural Network* (SNN)
2. *Triplet Network*
   1. Arsitektur Triplet Network
   2. Triplet Loss
   3. Triplet Sampling
3. *EfficientNet*-B0
4. *Visual Similarity*
5. *Image Retrieval*
6. Kajian Pustaka

**BAB III**

**METODOLOGI PENELITIAN**

* 1. Metode Penelitian => jenis penelitian (eksperimen), pendekatan (kuantitatif), dan tujuan umum (menguji performa *Siamese Neural Network* dalam pencarian kemiripan visual Kanji).
  2. Instrumen Penelitian
     1. Perangkat Keras => (spesifikasi CPU, GPU, RAM, OS.)
     2. Perangkat Lunak => (Python, PyTorch, library, IDE, dll.)
     3. Dataset Penelitian => (jelaskan *KanjiVG*, format data, jumlah sampel, alasan pemilihan dataset.)
  3. Objek Penelitian => Objek penelitian berupa sistem retrieval berbasis kemiripan visual karakter Kanji, di mana sistem dirancang untuk menerima satu citra karakter Kanji sebagai *query* dan menampilkan karakter lain yang secara visual mirip berdasarkan hasil embedding dari model SNN.  
     Komponen utama yang dianalisis meliputi: Dataset KanjiVG sebagai sumber data visual., Model *SNN* dengan backbone *EfficientNet-B0*, Mekanisme perhitungan jarak antar embedding (misal *Cosine distance*).
  4. Metode Pengumpulan Data => (Pengumpulan dataset KanjiVG dari sumber terbuka (GitHub / repositori publik) dan Validasi data (mis. pemeriksaan file rusak atau duplikat))
  5. Pra-pemrosesan Data => (Konversi SVG → PNG, *Resizing*, *grayscale*, *normalization*, dan *augmentation*, Pembagian data (train/test)), tambahkan diagram alur preprocessing sederhana agar mudah dipahami penguji.
  6. Desain Sistem dan Arsitektur Model => Arsitektur *Siamese Network, Backbone EfficientNet-B0,* Fungsi loss (Triplet Loss), Mekanisme perhitungan jarak antar embedding, Diagram arsitektur model.
  7. Pelatihan Model => Konfigurasi training (batch size, epoch, optimizer, learning rate, loss function), Tools yang digunakan.
  8. Evaluasi Model => Metrik evaluasi (*Recall@K*, *Precision*, *Accuracy, hasil retrieval),* Penjelasan logika pengujian (query vs gallery).

[1] N. T. Danh, “Sustanaible Methods of Improving Kanji Learning Skills for Japanese Language Learners at Basic Level at FPT University,” in *E3S Web of Conferences*, EDP Sciences, Jul. 2021. doi: 10.1051/e3sconf/202129505031.

[2] V. Rusyn, A. Boichuk, and L. Mochurad, “Cross-Language Transfer-Learning Approach via a Pretrained Preact ResNet-18 Architecture for Improving Kanji Recognition Accuracy and Enhancing a Number of Recognizable Kanji,” *Applied Sciences (Switzerland)*, vol. 15, no. 9, May 2025, doi: 10.3390/app15094894.

[3] D. Kurniadi, A. Mulyani, and N. Asih, “Comparative Analysis of Machine Learning Algorithms for Kanji Character Recognition Using HOG Features,” Jul. 24, 2025. doi: 10.20944/preprints202507.2038.v1.

[4] R. J. Rodríguez, B. Abstract, R. Badas, and R. José, “Kanji Recognition with AI,” 2024.

[5] A. I. Solis, J. Zarkovacki, J. Ly, and A. Atyabi, “Recognition of Handwritten Japanese Characters Using Ensemble of Convolutional Neural Networks,” Jun. 2023, [Online]. Available: http://arxiv.org/abs/2306.03954

[6] B. Lyu, H. Li, A. Tanaka, and L. Meng, “The early Japanese books reorganization by combining image processing and deep learning,” *CAAI Trans Intell Technol*, vol. 7, no. 4, pp. 627–643, Dec. 2022, doi: 10.1049/cit2.12104.

[7] P. Annisa, Z. A. Gultom, and Y. Sary, “Analysis and Implementation of CNN in Real- time Classification and Translation of Kanji Characters,” *Sinkron*, vol. 9, no. 1, pp. 296–305, Jan. 2024, doi: 10.33395/sinkron.v9i1.13176.

[8] S. Pandkar, K. Sabane, S. Rathod, P. Bansode, and S. Deore, “Handwritten Japanese Kanji Character Recognition using different Pruning Algorithm.”

[9] M. Assran *et al.*, “Masked Siamese Networks for Label-Efficient Learning,” Apr. 2022, [Online]. Available: http://arxiv.org/abs/2204.07141

[10] Y. Li, C. L. P. Chen, and T. Zhang, “A Survey on Siamese Network: Methodologies, Applications, and Opportunities,” *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 6, pp. 994–1014, Dec. 2022, doi: 10.1109/TAI.2022.3207112.

[11] D. Hartanto and R. Herawati, “COMPARATIVE ANALYSIS OF EFFICIENTNET AND RESNET MODELS IN THE CLASSIFICATION OF SKIN CANCER.”

[12] S. Otsuka and T. Murai, “Cognitive underpinnings of multidimensional Japanese literacy and its impact on higher-level language skills,” *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-81909-x.

[13] Y. Sakurai, Y. Uchiyama, A. Takeda, and Y. Terao, “On-Reading (Chinese-Style Pronunciation) Predominance Over Kun-Reading (Native Japanese Pronunciation) in Japanese Semantic Dementia,” *Front Hum Neurosci*, vol. 15, Aug. 2021, doi: 10.3389/fnhum.2021.700181.

[14] J. Breen and V. H. Mair, “SINO-PLATONIC PAPERS Kanji and the Computer: A Brief History of Japanese Character Set Standards.” [Online]. Available: www.sino-platonic.org

[15] A. A. Khan, A. A. Laghari, and S. A. Awan, “Machine Learning in Computer Vision: A Review,” *EAI Endorsed Transactions on Scalable Information Systems*, vol. 8, no. 32, pp. 1–11, 2021, doi: 10.4108/eai.21-4-2021.169418.

[16] M. Gao, G. Zou, Y. Li, and X. Guo, “Recent Advances in Computer Vision: Technologies and Applications,” Jul. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/electronics13142734.

[17] “View of Deep Learning in Computer Vision\_ A Critical Review”.

[18] M. H. M. Noor and A. O. Ige, “A Survey on State-of-the-art Deep Learning Applications and Challenges,” Jul. 2025, doi: 10.1016/j.engappai.2025.111225.

[19] D. Dakshayani Himabindu and S. Praveen Kumar, “A Survey on Computer Vision Architectures for Large Scale Image Classification using Deep Learning.” [Online]. Available: www.ijacsa.thesai.org

[20] N. E. Khalifa, M. Loey, and S. Mirjalili, “A comprehensive survey of recent trends in deep learning for digital images augmentation,” *Artif Intell Rev*, vol. 55, no. 3, pp. 2351–2377, Mar. 2022, doi: 10.1007/s10462-021-10066-4.

[21] P. Mittal, “A comprehensive survey of deep learning-based lightweight object detection models for edge devices,” *Artif Intell Rev*, vol. 57, no. 9, Sep. 2024, doi: 10.1007/s10462-024-10877-1.

[22] J. Liu and Y. Jin, “A comprehensive survey of robust deep learning in computer vision,” Nov. 01, 2023, *KeAi Communications Co.* doi: 10.1016/j.jai.2023.10.002.

[23] P. K. Vinodkumar, D. Karabulut, E. Avots, C. Ozcinar, and G. Anbarjafari, “A Survey on Deep Learning Based Segmentation, Detection and Classification for 3D Point Clouds,” Apr. 01, 2023, *MDPI*. doi: 10.3390/e25040635.

[24] 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, Apr. 2024, doi: 10.1007/s10462-024-10721-6.

[25] R. Sunkara and T. Luo, “No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects,” Aug. 2022, [Online]. Available: http://arxiv.org/abs/2208.03641

[26] M. M. Taye, “Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions,” Mar. 01, 2023, *MDPI*. doi: 10.3390/computation11030052.