

SKRIPSI

**VERIFIKASI KEASLIAN TUGAS SEKOLAH BERDASARKAN  
ANALISIS TULISAN TANGAN MENGGUNAKAN  
KOMBINASI *CONVOLUTIONAL NEURAL*  
*NETWORK* DAN *SIAMESE NETWORK***

DISUSUN OLEH

INDAH RAHAYU

2109020092



**UMSU**  
Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

**MEDAN**

**2025**

**SKRIPSI**

**VERIFIKASI KEASLIAN TUGAS SEKOLAH BERDASARKAN  
ANALISIS TULISAN TANGAN MENGGUNAKAN  
KOMBINASI *CONVOLUTIONAL NEURAL*  
*NETWORK* DAN *SIAMESE NETWORK***

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana  
Komputer (S.Kom) dalam Program Studi Teknologi Informasi Pada  
Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas  
Muhammadiyah Sumatera Utara**

**INDAH RAHAYU**

**NPM. 2109020092**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA  
MEDAN**

**2025**

**LEMBAR PENGESAHAN**

Judul Skripsi : VERIFIKASI KEASLIAN TUGAS SEKOLAH  
BERDASARKAN ANALISIS TULISAN TANGAN  
MENGUNAKAN KOMBINASI CONVOLUTIONAL  
NEURAL NETWORK DAN SIAMESE NETWORK

Nama Mahasiswa : INDAH RAHAYU

NPM : 2109020002

Program Studi : TEKNOLOGI INFORMASI

Menyetujui  
Komisi Pembimbing



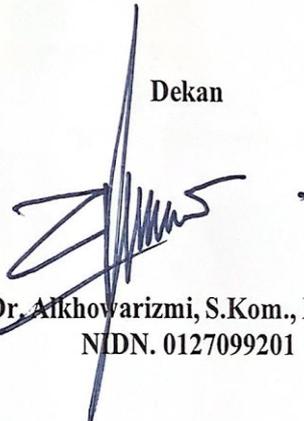
(Fatma Sari Hutagalung, S.Kom., M.Kom)  
NIDN. 0117019301

Ketua Program Studi



(Fatma Sari Hutagalung, S.Kom., M.Kom)  
NIDN. 0117019301

Dekan



(Dr. Alkhowarizmi, S.Kom., M.Kom)  
NIDN. 0127099201

**PERNYATAAN ORISINALITAS**

**VERIFIKASI KEASLIAN TUGAS SEKOLAH BERDASARKAN  
ANALISIS TULISAN TANGAN MENGGUNAKAN  
KOMBINASI *CONVOLUTIONAL NEURAL  
NETWORK* DAN *SIAMESE NETWORK***

**SKRIPSI**

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 21 April 2025

Yang membuat pernyataan



Indah Rahayu  
NPM. 2109020092

**PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK  
KEPENTINGAN AKADEMIS**

Sebagai civitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Indah Rahayu  
NPM : 2109020092  
Program Studi : Teknologi Informasi  
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

VERIFIKASI KEASLIAN TUGAS SEKOLAH BERDASARKAN ANALISIS  
TULISAN TANGAN MENGGUNAKAN KOMBINASI CONVOLUTIONAL  
NEURAL NETWORK DAN SIAMESE NETWORK

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Mei 2025

Yang membuat pernyataan



Indah Rahayu

NPM. 2109020092

## RIWAYAT HIDUP

### DATA PRIBADI

Nama Lengkap : Indah Rahayu  
Tempat dan Tanggal Lahir : Sidomulyo, 02 November 2003  
Alamat Rumah : Dusun VI Sidomulyo  
Telepon/Faks/HP : 089668909889  
E-mail : [indahrahayu022003@gmail.com](mailto:indahrahayu022003@gmail.com)  
Instansi Tempat Kerja : -  
Alamat Kantor : -

### DATA PENDIDIKAN

SD : SD Negeri No.057764 Sukaramai TAMAT : 2015  
SMP : SMP Negeri Satu Atap Sei Lapan TAMAT : 2018  
SMA : SMK Swasta Sri Langkat Tanjung Pura TAMAT : 2020

## KATA PENGANTAR



Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini. Laporan skripsi ini disusun untuk memenuhi salah satu syarat dalam memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara.

Ucapan terima kasih yang tulus Penulis sampaikan kepada kedua orang tua tercinta, **Budiono** dan **Jiyem** atas segala doa, dukungan moril dan materil, serta semangat yang tak pernah henti diberikan kepada Penulis. Tanpa mereka, penyusunan skripsi ini tidak akan berjalan sebagaimana mestinya. Penulis juga menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, Penulis menyampaikan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Agussani, MAP, selaku Rektor Universitas Muhammadiyah Sumatera Utara.
2. Bapak Prof. Dr. Muhammad Arifin, S.H., M.Hum, Bapak Prof. Dr. Akrim, M.Pd, dan Bapak Assoc. Prof. Dr. Rudianto, S.Sos., M.Si, selaku Wakil Rektor I, II, dan III Universitas Muhammadiyah Sumatera Utara.
3. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom, selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi.
4. Bapak Halim Maulana, S.T., M.Kom, selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi
5. Bapak Dr. Lutfi Basit, S.Sos., M.I.Kom., selaku Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi.
6. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom. Ketua Program Studi Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU sekaligus sebagai dosen pembimbing yang telah membimbing,

memberikan arahan, dan motivasi kepada penulis selama proses penyusunan skripsi ini.

7. Bapak Mhd Basri, S.Si, M.Kom., Sekretaris Program Studi Teknologi Informasi
8. Saudara dan saudari kandung penulis yang terus dan selalu mendukung penulis serta memberikan doa baik kepada penulis.
9. Teman-teman sekelas Penulis di kelas A2 Siang Program Studi Teknologi Informasi yang telah menjadi bagian penting dalam perjalanan akademik Penulis.
10. BTS, boy group Korea Selatan yang telah menjadi sumber semangat dan inspirasi bagi Penulis. Terutama melalui karya-karya dan pesan positif yang disampaikan, yang senantiasa menemani dan menguatkan Penulis dalam menjalani proses perkuliahan dan penyusunan skripsi ini.

Semoga segala kebaikan, dukungan, dan bantuan yang telah diberikan kepada penulis menjadi amal jariyah dan mendapatkan balasan yang setimpal dari Allah SWT. Penulis berharap skripsi ini dapat memberikan manfaat dan menjadi langkah awal untuk terus berkembang serta berkontribusi di bidang ilmu pengetahuan dan teknologi.

Akhir kata, Penulis memohon maaf apabila terdapat kekurangan dalam penulisan skripsi ini. Segala kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa yang akan datang.

Medan, 29 April 2025

Penulis



Indah Rahayu

**VERIFIKASI KEASLIAN TUGAS SEKOLAH BERDASARKAN  
ANALISIS TULISAN TANGAN MENGGUNAKAN  
KOMBINASI CONVOLUTIONAL NEURAL  
NETWORK DAN SIAMESE NETWORK**

**ABSTRAK**

Keaslian tugas siswa merupakan aspek penting dalam proses evaluasi pembelajaran yang objektif. Namun, dalam praktiknya, tidak sedikit ditemukan kasus di mana tugas siswa dikerjakan oleh pihak lain, seperti orang tua atau anggota keluarga, sehingga hasilnya tidak mencerminkan kemampuan asli siswa. Penelitian ini bertujuan untuk mengembangkan sistem verifikasi keaslian tugas sekolah berdasarkan analisis tulisan tangan dengan memanfaatkan teknologi pengolahan citra digital dan *Machine Learning*. Sistem ini menggunakan kombinasi algoritma *Convolutional Neural Network* (CNN) untuk ekstraksi fitur tulisan tangan dan *Siamese Network* (SN) untuk membandingkan tingkat kemiripan antara tulisan siswa. Dataset yang digunakan terdiri dari gambar tulisan tangan siswa kelas VI SD Negeri 060868 Medan Timur dan tulisan tangan bukan siswa sebagai data pembanding. Gambar dikonversi ke skala abu-abu, diubah ukurannya menjadi 128x128 piksel, dan diproses melalui CNN untuk menghasilkan vektor fitur berdimensi satu. Vektor fitur ini kemudian dibandingkan menggunakan *Siamese Network* dengan metode *Euclidean Distance*. Jika skor kemiripan  $\geq 0.75$ , sistem menyimpulkan tulisan tangan tersebut cocok dan asli. Sistem ini dibangun menggunakan Python, TensorFlow/Keras, OpenCV untuk pemrosesan dan pelatihan model, serta Express.js dan Flutter untuk pengembangan aplikasi web dan mobile. Hasil implementasi menunjukkan bahwa sistem dapat menjalankan proses verifikasi secara otomatis dan membantu guru melakukan penilaian keaslian tugas dengan lebih objektif, efisien, dan terstruktur. Sistem ini diharapkan dapat mendukung transparansi dan integritas akademik serta menjadi solusi inovatif dalam dunia pendidikan.

**Kata Kunci:** tulisan tangan, verifikasi tugas sekolah, CNN, *Siamese Network*, keaslian tulisan

**VERIFIKASI KEASLIAN TUGAS SEKOLAH BERDASARKAN  
ANALISIS TULISAN TANGAN MENGGUNAKAN  
KOMBINASI CONVOLUTIONAL NEURAL  
NETWORK DAN SIAMESE NETWORK**

**ABSTRACT**

The authenticity of student assignments plays a crucial role in ensuring objective assessment in the educational process. However, in practice, many cases have been found where assignments are completed by other parties, such as parents or family members, resulting in work that does not reflect the student's actual abilities. This research aims to develop a system for verifying the authenticity of school assignments based on handwriting analysis by utilizing digital image processing and Machine Learning technology. The system employs a combination of Convolutional Neural Network (CNN) for handwriting feature extraction and Siamese Network (SN) to compare the similarity between handwriting samples. The dataset consists of handwriting images from 6th-grade students of SD Negeri 060868 Medan Timur and non-student handwriting samples as comparison data. Images are converted to grayscale, resized to 128x128 pixels, and processed through CNN to generate one-dimensional feature vectors. These vectors are then compared using a Siamese Network with the Euclidean Distance method. If the similarity score is  $\geq 0.75$ , the system determines that the handwriting is authentic. The system is developed using Python, TensorFlow/Keras, and OpenCV for model training and processing, as well as Express.js and Flutter for web and mobile application development. The implementation results show that the system can automatically perform verification and assist teachers in assessing assignment authenticity more objectively, efficiently, and systematically. This system is expected to promote transparency, academic integrity, and serve as an innovative solution in the field of education.

**Keywords:** handwriting, assignment verification, CNN, Siamese Network, handwriting authenticity

## DAFTAR ISI

<b>KATA PENGANTAR</b> .....	Error! Bookmark not defined.
<b>ABSTRAK</b> .....	<b>ix</b>
<b>ABSTRACT</b> .....	<b>x</b>
<b>DAFTAR ISI</b> .....	<b>xi</b>
<b>DAFTAR GAMBAR</b> .....	<b>xiv</b>
<b>DAFTAR TABEL</b> .....	<b>xv</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
<b>1.1 Latar Belakang Masalah</b> .....	<b>1</b>
<b>1.2 Rumusan Masalah</b> .....	<b>3</b>
<b>1.3 Batasan Masalah</b> .....	<b>3</b>
<b>1.4 Tujuan Penelitian</b> .....	<b>4</b>
<b>1.5 Manfaat Penelitian</b> .....	<b>5</b>
<b>BAB II LANDASAN TEORI</b> .....	<b>7</b>
<b>2.1 Convolutional Neural Network (CNN)</b> .....	<b>7</b>
<b>2.2 Siamese Network (SN)</b> .....	<b>7</b>
<b>2.3 Penelitian Terdahulu</b> .....	<b>8</b>
<b>2.4 Database</b> .....	<b>8</b>
2.4.1 MySQL .....	<b>9</b>
<b>2.5 Bahasa Pemrograman</b> .....	<b>10</b>
2.5.1 Python .....	<b>10</b>
2.5.2 JavaScript (Node.js).....	<b>12</b>
<b>2.6 Framework Back-End</b> .....	<b>12</b>
2.6.1 Flask.....	<b>13</b>
2.6.2 Express.js .....	<b>13</b>
<b>2.7 Framework Mobile</b> .....	<b>14</b>
2.7.1 Flutter .....	<b>15</b>
<b>2.8 Teknologi Machine Learning dan Pengolahan Citra</b> .....	<b>15</b>
2.8.1 TensorFlow/Keras .....	<b>16</b>
2.8.2 OpenCV .....	<b>16</b>
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>18</b>
<b>3.1 Pendekatan Penelitian</b> .....	<b>18</b>
3.1.1 Model pengembangan perangkat lunak.....	<b>18</b>

3.1.2 Pendekatan teknikal.....	19
<b>3.2 Analisis Kebutuhan Sistem.....</b>	<b>20</b>
3.2.1 Diagram <i>use case</i> .....	20
3.2.2 Diagram <i>sequence</i> .....	21
<b>3.3 Desain Sistem .....</b>	<b>22</b>
3.3.1 Diagram ERD ( <i>Entity-Relationship Diagram</i> ).....	22
3.3.2 <i>Flowchart</i> sistem.....	24
3.3.3 <i>Flowchart</i> pengguna.....	24
3.3.4 <i>Flowchart</i> algoritma.....	25
3.3.5 <i>Wireframe app mobile</i> .....	26
<b>3.4 Pengumpulan Data .....</b>	<b>27</b>
3.4.1 Data tulis tangan siswa.....	28
3.4.2 Data tulis tangan bukan siswa .....	29
3.4.3 Karakteristik <i>dataset</i> .....	29
3.4.4 Pengolahan data awal.....	30
<b>3.5 Pengembangan Model <i>Machine Learning</i>.....</b>	<b>30</b>
3.5.1 Tujuan pengembangan model .....	30
3.5.2 Arsitektur model.....	31
3.5.3 Dataset pelatihan dan pengujian.....	32
3.5.4 <i>Framework</i> dan <i>tools</i> yang digunakan .....	32
3.5.5 Proses pelatihan model.....	33
3.5.6 Output model.....	33
3.5.7 Evaluasi dan <i>Tuning Hyperparameter</i> .....	34
<b>3.6 Perhitungan dalam <i>Convolutional Neural Network (CNN)</i> .....</b>	<b>34</b>
3.6.1 Hasil Pra-Pemrosesan Gambar.....	37
3.6.2 Ekstraksi fitur dengan CNN .....	39
3.6.3 Analisis Vektor Fitur dari <i>Fully Connected Layer</i> .....	42
<b>3.7 Perhitungan dalam <i>Siamese Network (SN)</i>.....</b>	<b>44</b>
<b>3.8 Implementasi <i>Siamese Network</i> dalam Verifikasi Tulis Tangan .....</b>	<b>45</b>
3.8.1 Arsitektur <i>Siamese Network</i> .....	46
3.8.2 Proses Implementasi <i>Siamese Network</i> .....	47
<b>3.9 Implementasi Sistem .....</b>	<b>48</b>
3.9.1 Lingkungan implementasi .....	48
3.9.2 Integrasi komponen .....	48

3.9.3 Proses <i>deployment</i> .....	49
3.9.4 Validasi dan pengujian awal .....	50
3.9.5 Dokumentasi.....	50
<b>3.10 Pengujian Sistem .....</b>	<b>51</b>
3.10.1 Metodologi pengujian .....	51
<b>3.11 Tempat dan Waktu Penelitian .....</b>	<b>53</b>
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>55</b>
<b>4.1 Pendahuluan .....</b>	<b>55</b>
<b>4.2 Hasil Implementasi Sistem .....</b>	<b>56</b>
4.2.1 Halaman <i>login</i> .....	56
4.2.2 Halaman <i>dashboard</i> .....	57
4.2.3 Menu Utama.....	57
4.2.4 Halaman siswa.....	58
4.2.5 Halaman sample tulisan .....	59
4.2.6 Halaman tugas siswa .....	59
4.2.7 Implementasi teknis proses verifikasi .....	60
4.2.8 Halaman histori .....	63
<b>4.3 Dataset dan Pelatihan Model .....</b>	<b>64</b>
4.3.1 Dataset.....	64
4.3.2 Pelatihan Model.....	65
<b>4.4 Pengujian Sistem .....</b>	<b>67</b>
4.4.1 Pengujian <i>Blackbox</i> .....	67
<b>4.5 Evaluasi Sistem.....</b>	<b>70</b>
4.5.1 Kelebihan Sistem.....	70
4.5.2 Kekurangan Sistem .....	71
<b>4.6 Pembahasan .....</b>	<b>71</b>
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>73</b>
<b>5.1 Kesimpulan .....</b>	<b>73</b>
<b>5.2 Saran.....</b>	<b>73</b>
<b>DAFTAR PUSTAKA.....</b>	<b>75</b>

## DAFTAR GAMBAR

Gambar 3.1 Model <i>Waterfall</i> .....	18
Gambar 3.2 Diagram <i>Usecase</i> .....	21
Gambar 3.3 Diagram <i>Sequence</i> .....	21
Gambar 3.4 Diagram ERD ( <i>Entity-Relationship Diagram</i> ) .....	22
Gambar 3.6 <i>Flowchart</i> Sistem .....	24
Gambar 3.7 <i>Flowchart</i> Pengguna .....	25
Gambar 3.8 <i>Flowchart</i> Algoritma .....	25
Gambar 3.9 <i>Wireframe</i> <i>Splash</i> dan <i>Login</i> .....	26
Gambar 3.10 <i>Wireframe</i> Halaman Utama dan Menu .....	26
Gambar 3.11 <i>Wireframe</i> Halaman Daftar dan <i>Form Input</i> Siswa .....	27
Gambar 3.12 <i>Wireframe</i> Daftar dan <i>Form Input</i> Sample .....	27
Gambar 3.13 Struktur CNN Sumber: Gunawan & Setiawan (2022).....	31
Gambar 3.14 Strider (CS231n) .....	35
Gambar 3.15 <i>Pooling Layer</i> .....	36
Gambar 3.16 Tulis tangan asli siswa .....	38
Gambar 3.17 Hasil grayscale .....	38
Gambar 3.18 Hasil Resize.....	39
Gambar 3.19 Hasil <i>Convolutional Layer</i> .....	41
Gambar 3.20 Arsitektur <i>Siamese Network</i> .....	46
Gambar 4.1 Halaman <i>Login</i> .....	56
Gambar 4.2 Halaman <i>Dashboard</i> .....	57
Gambar 4.2 Menu Utama .....	58
Gambar 4.3 Halaman Siswa.....	58
Gambar 4.4 Halaman Sample Tulisan .....	59
Gambar 4.5 Halaman Tugas Siswa.....	60
Gambar 4.6 Kode Memuat Model.....	61
Gambar 4.7 Kode Pra-pemrosesan gambar .....	61
Gambar 4.8 Kode Membandingkan Sample dengan Tugas.....	62
Gambar 4.9 Kode perhitungan dan evaluasi skor .....	62
Gambar 4.10 Halaman Histori.....	63
Gambar 4.11 Grafik Akurasi Pelatihan dan Validasi Model .....	67

## DAFTAR TABEL

<b>Tabel 3.1 Contoh Data tulis tangan siswa .....</b>	<b>28</b>
<b>Tabel 3.2 Contoh Data bukan tulis tangan siswa .....</b>	<b>29</b>
<b>Tabel 3.3 Hasil vektor fitur 1D .....</b>	<b>41</b>
<b>Tabel 3.4 Waktu Penelitian .....</b>	<b>53</b>
<b>Tabel 4.1 Pengujian Halaman <i>Login</i> .....</b>	<b>68</b>
<b>Tabel 4.2 Pengujian Halaman <i>Dashboard</i> .....</b>	<b>68</b>
<b>Tabel 4.3 Pengujian Halaman Siswa .....</b>	<b>69</b>
<b>Tabel 4.4 Pengujian Halaman Sample Tulisan Tangan .....</b>	<b>69</b>
<b>Tabel 4.5 Pengujian Halaman <i>Upload</i> Tugas .....</b>	<b>70</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Pada perkembangan zaman yang semakin canggih dengan teknologi mempermudah akses informasi yang sangat penting dalam beberapa bidang termasuk dunia pendidikan. Salah satu tantangan utama dalam dunia pendidikan saat ini adalah memastikan keaslian tugas para siswa, terutama dalam tugas rumah yang diberikan sebagai bentuk evaluasi pemahaman. Keaslian tugas menjadi faktor penting dalam menilai perkembangan akademik para siswa secara objektif dari guru, namun dalam praktiknya, tantangan ini semakin rumit.

Di SD Negeri 060868 Medan Timur, khususnya pada siswa kelas 6, muncul kekhawatiran mengenai validitas tugas yang dikumpulkan. Tidak jarang ditemukan indikasi bahwa tugas yang diserahkan siswa bukan sepenuhnya hasil kerja mereka sendiri. Orang tua atau anggota keluarga sering kali terlibat dalam pengerjaan tugas, sehingga hasilnya kurang mencerminkan kemampuan individu siswa. Akibatnya, guru menghadapi kesulitan dalam menilai pemahaman dan keterampilan siswa secara objektif, yang berpotensi mempengaruhi efektivitas proses pembelajaran.

Untuk mengatasi permasalahan ini, pemanfaatan teknologi verifikasi berbasis analisis tulisan tangan menawarkan solusi inovatif. Teknologi ini memanfaatkan pengolahan citra digital dan kecerdasan buatan (artificial intelligence) untuk membedakan tulisan tangan asli siswa dengan tulisan yang diduga dikerjakan oleh pihak lain. Dalam penelitian ini, sistem verifikasi keaslian tugas dikembangkan menggunakan kombinasi *Convolutional Neural Network*

(CNN) dan *Siamese Network* (SN). CNN merupakan model jaringan saraf tiruan yang dirancang untuk mengenali dan mengekstraksi fitur spesifik dari tulisan tangan, seperti bentuk huruf, tekanan, dan gaya penulisan. Sementara itu, SN digunakan untuk membandingkan dua sampel tulisan dan mengukur tingkat kemiripan di antara keduanya.

Penerapan *Convolutional Neural Network* (CNN) dan *Siamese Network* (SN) dalam sistem verifikasi tulisan tangan dilakukan melalui beberapa tahap utama. Tahap pertama adalah preprocessing gambar, di mana gambar tulisan tangan dikonversi ke skala abu-abu untuk menyederhanakan data serta distandarisasi ukurannya, misalnya menjadi 224 x 224 piksel. Selain itu, teknik augmentasi data, seperti rotasi dan perubahan pencahayaan, diterapkan untuk meningkatkan ketahanan model terhadap variasi tulisan tangan yang berbeda.

Setelah proses preprocessing selesai, tahap berikutnya adalah ekstraksi fitur menggunakan CNN. CNN berperan dalam menganalisis elemen-elemen khas dalam tulisan tangan, seperti bentuk huruf, pola goresan, dan tekanan pena. Model CNN yang digunakan dapat berupa VGG16 atau ResNet, yang memanfaatkan pendekatan transfer learning untuk meningkatkan akurasi, terutama jika dataset yang digunakan relatif kecil.

Selanjutnya, hasil ekstraksi fitur ini digunakan dalam proses perbandingan menggunakan *Siamese Network*. Tulisan tangan terbaru dibandingkan dengan sampel tulisan siswa yang telah tersimpan dalam *Database*, di mana *Siamese Network* menghitung tingkat kesamaan (*similarity score*) antara kedua gambar. Jika skor kesamaan melebihi ambang batas tertentu, misalnya  $\geq 75\%$ , maka sistem menganggap tugas tersebut asli dan dikerjakan oleh siswa yang bersangkutan.

Sebaliknya, jika skor berada di bawah ambang batas, tugas tersebut dicurigai tidak dikerjakan oleh siswa.

Sebagai tahap akhir, keputusan verifikasi dicatat dalam sistem dan dilaporkan kepada guru untuk evaluasi lebih lanjut. Dengan menerapkan metode ini, sistem dapat secara otomatis mengidentifikasi keaslian tugas, mengurangi kemungkinan kecurangan dalam pengerjaan tugas, serta membantu guru dalam melakukan penilaian yang lebih adil dan objektif.

Dengan menggabungkan kedua pendekatan ini, sistem yang dikembangkan diharapkan dapat mengidentifikasi pola tulisan tangan dengan akurasi tinggi, sehingga memungkinkan guru untuk melakukan verifikasi keaslian tugas secara lebih efektif dan objektif. Implementasi teknologi ini tidak hanya mendukung proses evaluasi akademik yang lebih adil tetapi juga mendorong siswa untuk lebih mandiri dalam mengerjakan tugas mereka. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi positif dalam meningkatkan kualitas pendidikan melalui pemanfaatan teknologi cerdas.

## **1.2 Rumusan Masalah**

Bagaimana merancang sistem verifikasi keaslian tugas sekolah berbasis analisis tulisan tangan menggunakan kombinasi *Convolutional Neural Network* (CNN) dan *Siamese Network* (SN)?

## **1.3 Batasan Masalah**

1. Penelitian ini hanya diterapkan pada siswa kelas 6 di SD Negeri 060868 Medan Timur, sehingga hasil yang diperoleh akan difokuskan pada kelompok usia dan tingkat pendidikan tertentu sesuai dengan konteks pembelajaran di sekolah tersebut.

2. Tulisan tangan asli siswa diperoleh dari catatan dan tugas sekolah yang telah dikerjakan secara mandiri, Tulisan tangan yang bukan berasal dari siswa dikumpulkan dari berbagai sumber untuk membantu model dalam membedakan pola tulisan yang bukan milik siswa.
3. Verifikasi tulisan tangan dilakukan dengan membandingkan tulisan terbaru siswa dengan sampel tulisan tangan yang telah dikumpulkan sebelumnya. Analisis dilakukan menggunakan kombinasi *Convolutional Neural Network* (CNN) dan *Siamese Network* (SN) untuk mengidentifikasi tingkat kemiripan tulisan tangan siswa.
4. Sistem yang dikembangkan hanya bertujuan untuk menganalisis keaslian tulisan tangan dan tidak mempertimbangkan isi atau kualitas tugas yang dikerjakan siswa. Oleh karena itu, aspek penilaian terhadap isi tugas tetap menjadi tanggung jawab guru.
5. Penelitian ini diimplementasikan menggunakan bahasa pemrograman Python untuk implementasi model *Machine Learning* menggunakan TanserFlow/Keras dan OpenCV, JavaScript (Node.js) untuk pengelolaan server dan API menggunakan Express.js, dan untuk pengembangan aplikasi mobile WebView menggunakan Flutter.

#### **1.4 Tujuan Penelitian**

Penelitian ini bertujuan untuk:

1. Mengembangkan sistem berbasis pengolahan citra digital dan *Machine Learning* untuk memverifikasi keaslian tugas siswa berdasarkan analisis dan perbandingan tulisan tangan.

2. Menyediakan alat bantu bagi guru dalam memeriksa keaslian tugas siswa secara lebih efisien dan akurat, sehingga mendukung penilaian yang lebih objektif.
3. Mengoptimalkan pemanfaatan teknologi dalam dunia pendidikan, khususnya dalam aspek evaluasi keaslian tugas, guna meningkatkan transparansi dan keadilan dalam proses penilaian.

### **1.5 Manfaat Penelitian**

Hasil penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Bagi Guru dan Sekolah
  - a. Mempermudah dalam memantau dan memverifikasi keaslian tugas siswa, sehingga guru dapat melakukan penilaian yang lebih objektif dan memantau perkembangan tulisan tangan siswa.
  - b. Membantu guru dalam mengidentifikasi pola kecurangan akademik, sehingga dapat diterapkan strategi pembelajaran yang lebih efektif.
2. Bagi Siswa
  - a. Meningkatkan kesadaran akan pentingnya integritas akademik, serta menanamkan nilai-nilai kejujuran dan tanggung jawab dalam mengerjakan tugas sekolah.
  - b. Memberikan pemahaman tentang dampak negatif keterlibatan berlebihan orang tua dalam tugas siswa, yang dapat menghambat perkembangan kemandirian dan pemahaman siswa.
3. Bagi Dunia Pendidikan

- a. Menunjukkan bagaimana teknologi *Machine Learning* dapat diterapkan dalam dunia pendidikan untuk mendukung sistem evaluasi yang lebih transparan dan efektif.
- b. Menjadi referensi bagi pengembangan teknologi serupa dalam bidang verifikasi tulisan tangan atau keaslian dokumen lainnya di sektor pendidikan maupun industri lainnya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 *Convolutional Neural Network (CNN)***

*Convolutional Neural Network (CNN)* adalah arsitektur deep learning yang terinspirasi dari cara makhluk hidup memproses penglihatan. Pada tahun 1959, Hubel & Wiesel menemukan bahwa sel di korteks visual hewan mendeteksi cahaya dalam bidang reseptif. Berdasarkan temuan ini, Kunihiko Fukushima mengembangkan neocognitron pada 1980, yang menjadi cikal bakal CNN.

*Convolutional Neural Network (CNN)* adalah arsitektur jaringan saraf tiruan yang dirancang khusus untuk mengenali pola dalam data visual, seperti gambar atau video. CNN bekerja dengan menggunakan operasi konvolusi untuk mengekstrak fitur dari gambar secara otomatis, tanpa perlu melakukan ekstraksi fitur secara manual seperti pada metode tradisional (Gu, Wang, & Kuen, 2017).

#### **2.2 *Siamese Network (SN)***

*Siamese Network (SN)* adalah jenis jaringan saraf tiruan yang digunakan untuk membandingkan dua *input* dan menentukan seberapa mirip kedua *input* tersebut. SN bekerja dengan dua subnet yang memiliki bobot yang sama (*shared weights*) dan menggunakan metrik jarak untuk menentukan kesamaan.

*Network Siamese* banyak digunakan dalam pelacakan visual karena keseimbangan antara akurasi dan kecepatan. Namun, tracker Siamese masih menggunakan backbone dangkal seperti AlexNet, sehingga belum sepenuhnya memanfaatkan kemampuan jaringan saraf dalam yang lebih modern (Zhang & Peng, 2019).

### 2.3 Penelitian Terdahulu

Beberapa penelitian sebelumnya telah membuktikan efektivitas metode CNN dan *Siamese Network* dalam pengenalan tulisan tangan serta tugas-tugas klasifikasi lainnya:

1. Smith et al. (2021) mengembangkan model berbasis CNN untuk mengenali tulisan tangan dalam dataset MNIST dan mendapatkan akurasi tinggi dalam klasifikasi karakter tulisan tangan.
2. Zhang et al. (2022) menerapkan *Siamese Network* untuk verifikasi tanda tangan dalam sistem keamanan, yang menunjukkan bahwa model ini mampu membedakan tanda tangan asli dan palsu dengan tingkat akurasi tinggi.
3. Prasetyo et al. (2023) melakukan penelitian mengenai pengenalan tulisan tangan siswa menggunakan CNN dan berhasil meningkatkan akurasi dalam mendeteksi tulisan dengan berbagai variasi gaya tulisan.

### 2.4 Database

*Database* berperan penting dalam menyimpan, mengelola, dan mengambil data secara terstruktur dalam pengembangan perangkat lunak. DBMS memungkinkan pengelolaan informasi secara efisien. Dalam proyek ini, *Database* digunakan untuk menyimpan data tugas, hasil verifikasi, serta informasi pengguna dan siswa.

Menurut Rosa, DBMS adalah aplikasi untuk menyimpan, mengelola, dan menampilkan data (Rosa & Shalauddin, 2016). Sementara itu, menurut Abdulloh, *Database* adalah perangkat lunak yang menyediakan layanan bagi pengguna

untuk pembuatan, pengelolaan, dan pengaturan akses ke *Database* (Abdulloh, 2018).

Pengelolaan data yang baik diperlukan untuk menjaga konsistensi, keandalan, dan akses cepat. Oleh karena itu, pemilihan DBMS yang tepat sangat krusial agar aplikasi berjalan optimal.

#### 2.4.1 MySQL

MySQL adalah RDBMS yang banyak digunakan dalam pengembangan aplikasi web dan perangkat lunak, menggunakan SQL untuk mengelola data dalam tabel yang terhubung. Menurut Rusli, MySQL berfungsi mengelola struktur *Database* dalam proses pembuatan dan pengelolannya (Rusli, Ahmar, & Rahman, 2019). Agustini (2016) menyebutnya sebagai *Database* multiuser berbasis SQL, sementara Sulistiati mendefinisikannya sebagai server pengelolaan *Database*. Secara umum, MySQL berperan dalam mengatur, membuat, dan mengolah data dalam *Database* (Sulistiati et al., 2019).

##### 1. Peran MySQL dalam Proyek ini

Dalam proyek ini, MySQL berfungsi sebagai penyimpanan data yang diperlukan untuk proses verifikasi tulisan tangan, meliputi:

- a. Data Pengguna: Menyimpan informasi akun guru yang terdaftar.
- b. Data Tugas Siswa: Menyimpan tugas yang diunggah guru untuk dianalisis.
- c. Data Verifikasi: Menyimpan hasil analisis tulisan tangan untuk menentukan keaslian tugas siswa.

MySQL memungkinkan penyimpanan data yang terstruktur, akses cepat, serta pengelolaan data dalam jumlah besar, mendukung efisiensi sistem verifikasi.

## **2.5 Bahasa Pemrograman**

Bahasa pemrograman adalah alat utama dalam pengembangan perangkat lunak untuk merancang logika dan fungsionalitas aplikasi. Dalam penelitian ini, digunakan Python untuk pengolahan data dan *Machine Learning*, serta JavaScript (Node.js) untuk membangun server back-end dan API.

Menurut Abdul Kadir, program ditulis mengikuti aturan bahasa pemrograman, yang berfungsi seperti bahasa manusia tetapi dirancang agar dipahami komputer (Kadir, 2020). Hanief menambahkan bahwa bahasa pemrograman adalah seperangkat instruksi standar yang digunakan untuk memberikan perintah kepada komputer, dengan aturan sintaksis dan semantis tertentu (Hanief & Jepriana, 2020).

### **2.5.1 Python**

Python adalah bahasa pemrograman tingkat tinggi yang populer di bidang pengolahan data, *Machine Learning*, dan kecerdasan buatan. Bahasa ini dikenal karena sintaksisnya yang sederhana serta kemampuannya menyelesaikan tugas kompleks secara efisien.

Menurut Lutz, Python adalah bahasa open-source yang dirancang untuk meningkatkan kualitas perangkat lunak, produktivitas pengembang, serta integrasi komponen (Lutz, 2010). Agus menambahkan bahwa Python merupakan bahasa dinamis tingkat tinggi dengan interpreter yang langsung mengonversi kode sumber ke kode mesin saat dijalankan (Suharto, 2023). Diciptakan oleh Guido van Rossum pada 1990, Python kini menjadi pilihan

utama bagi banyak perusahaan dan pengembang (Romzi & Kurniawan, 2020).

1. Keunggulan Python
  - a. Pengolahan Data: Dilengkapi pustaka seperti NumPy, Pandas, dan Matplotlib.
  - b. *Machine Learning*: Mendukung pustaka seperti TensorFlow, Keras, dan Scikit-learn untuk pengembangan model CNN dan Siamese Network.
  - c. Pengolahan Citra: Memanfaatkan OpenCV dan Pillow untuk pra-pemrosesan gambar tulisan tangan.
  - d. Kompatibilitas: Mudah diintegrasikan dengan MySQL dan API berbasis teknologi lain.
2. Penggunaan Python dalam Penelitian
  - a. Pengolahan Citra: Menggunakan OpenCV dan Pillow untuk menyiapkan gambar sebelum analisis.
  - b. Model *Machine Learning*: Digunakan dalam pengembangan CNN dan *Siamese Network* untuk verifikasi tulisan tangan.
  - c. Integrasi dengan *Backend*: Flask digunakan untuk membangun API yang menghubungkan aplikasi dengan model *Machine Learning*.

Python memainkan peran kunci dalam pengolahan data dan pengembangan algoritma deep learning dalam proyek ini.

### 2.5.2 JavaScript (Node.js)

JavaScript adalah bahasa pemrograman yang umum digunakan untuk aplikasi web interaktif. Dalam proyek ini, JavaScript dipadukan dengan Node.js untuk membangun server back-end dan API. Node.js memungkinkan eksekusi JavaScript di sisi server, meningkatkan efisiensi dan mendukung interaksi real-time.

Menurut Rismon, JavaScript adalah bahasa skrip dinamis yang menambahkan interaktivitas pada halaman HTML, memungkinkan web berinteraksi dengan pengguna dan merespons berbagai peristiwa (Sianipar, 2017). Selain itu, JavaScript juga digunakan dalam pengembangan aplikasi mobile, desktop, dan game berbasis web (Firdaus & Ratnasari, 2022).

Dalam penelitian ini, Node.js digunakan untuk:

1. Membangun API: Dengan Express.js, digunakan untuk menghubungkan antarmuka pengguna dengan *back-end*.
2. Pengelolaan Permintaan: Menangani komunikasi antara front-end dan back-end, serta mengirimkan hasil verifikasi tugas.
3. Autentikasi dan Keamanan: Menyediakan akses aman bagi admin dan guru dalam sistem.

Node.js memberikan kecepatan dan skalabilitas tinggi, menjadikannya pilihan ideal untuk server dalam proyek verifikasi tulisan tangan ini.

### 2.6 Framework Back-End

*Framework back-end* digunakan untuk membangun logika *server*, API, dan manajemen data di sisi *server*, menghubungkan *front-end*, *Database*, dan sistem *Machine Learning*. Dalam penelitian "Verifikasi Keaslian Tugas Sekolah Berdasarkan Analisis Tulisan Tangan", dua *framework* utama yang digunakan

adalah Flask (Python) dan Express.js (Node.js) karena kehandalannya dalam menangani kebutuhan sistem.

Back-end development berfokus pada sisi logis aplikasi, mencakup bahasa pemrograman, arsitektur inti, dan logika, yang dijalankan di server komputer (Ghimire, 2010).

### 2.6.1 Flask

Flask adalah framework web mikro berbasis Python yang mempermudah pengembangan aplikasi web dan API dengan struktur sederhana dan fleksibel. Framework ini hanya menyediakan fungsi inti, memungkinkan pengembang menyesuaikan fitur sesuai kebutuhan (Grinberg, 2018; Bachtiar, Paniran, & Suksmadana, 2024). Flask digunakan dalam penelitian ini untuk:

1. API untuk Model *Machine Learning*: Menghubungkan aplikasi dengan model CNN dan *Siamese Network* untuk verifikasi tulisan tangan.
2. Integrasi dengan Python dan TensorFlow: Mengelola komunikasi antara model *Machine Learning* dan antarmuka pengguna.
3. Logika Verifikasi: Menangani pengolahan *input* gambar, pra-pemrosesan data, dan penyimpanan hasil verifikasi di *Database*.

Flask dipilih karena kompatibilitasnya dengan Python serta kemampuannya dalam integrasi *Machine Learning*.

### 2.6.2 Express.js

Express.js adalah *framework* web minimalis untuk Node.js yang digunakan dalam pengembangan *server* dan API. *Framework* ini populer karena kemudahannya serta kompatibilitasnya dengan Node.js, yang bersifat

*non-blocking*, memungkinkan penanganan banyak permintaan secara simultan (Pradigi, Harlina, & Solehatin, 2022).

Express.js digunakan untuk membangun REST API, yang berperan sebagai penghubung antara front-end dan back-end guna meningkatkan kinerja sistem (Lazuardy & Anggraini, 2022). Dalam penelitian ini, Express.js digunakan untuk:

1. Membangun API Server: Menangani permintaan dari aplikasi web dan *mobile* serta mengelola alur data antara *front-end*, *Database*, dan Flask.
2. Autentikasi: Memastikan hanya pengguna terotorisasi yang dapat mengakses fitur sistem.
3. Integrasi MySQL: Mengelola penyimpanan dan pengambilan data terkait tugas dan hasil verifikasi.
4. Pengelolaan Permintaan dan Respons: Meneruskan permintaan ke Flask jika diperlukan dan mengirimkan hasil verifikasi kembali ke aplikasi.

Express.js memberikan fondasi kuat untuk *server back-end* yang efisien, sekaligus mendukung integrasi yang mudah dengan berbagai komponen sistem lainnya.

## **2.7 Framework Mobile**

*Framework mobile* digunakan untuk mengembangkan aplikasi lintas platform yang dapat berjalan di berbagai perangkat, seperti Android dan iOS. Dalam proyek ini, Flutter dipilih sebagai *framework* untuk membangun aplikasi mobile, yang akan dikembangkan setelah versi website selesai dan terhosting.

### 2.7.1 Flutter

Flutter adalah *framework* UI lintas *platform* dari Google yang memungkinkan pengembangan aplikasi Android dan iOS dengan satu basis kode. Menggunakan bahasa pemrograman Dart, Flutter menawarkan performa mendekati native, berkat mesin rendering mandiri yang tidak bergantung pada komponen bawaan perangkat (OEM) (Wu, 2018).

Penggunaan Flutter dalam penelitian ini

- a. Integrasi Domain Website: Menggunakan WebView untuk menampilkan versi website dalam aplikasi *mobile*, menjaga konsistensi fitur.
- b. Kemudahan Pengembangan Lintas *Platform*: Memungkinkan pembuatan aplikasi Android dan iOS dengan satu kode, menghemat waktu dan sumber daya.
- c. Fitur Tambahan: Memudahkan penambahan fitur interaktif di masa depan yang tidak tersedia di versi website.

Flutter memastikan aplikasi mobile dalam penelitian ini tetap efisien, serbaguna, dan memberikan pengalaman pengguna yang optimal.

### 2.8 Teknologi *Machine Learning* dan Pengolahan Citra

Dalam proyek ini, teknologi *Machine Learning* dan pengolahan citra memainkan peran kunci dalam proses verifikasi keaslian tulisan tangan. Dengan memanfaatkan teknologi ini, sistem dapat menganalisis pola tulisan tangan secara mendalam, membandingkan kemiripan, dan memberikan hasil yang akurat terkait keaslian tugas siswa. Dua alat utama yang digunakan adalah TensorFlow/Keras untuk pengembangan model *Machine Learning* dan OpenCV untuk pra-pemrosesan data citra.

### 2.8.1 TensorFlow/Keras

TensorFlow adalah pustaka *open-source* dari Google untuk membangun dan melatih model *Machine Learning* dan *deep learning*. Keras, sebagai API tingkat tinggi di atas TensorFlow, menyediakan antarmuka yang lebih sederhana untuk mendefinisikan dan melatih model. Penggunaan TensorFlow/Keras pada penelitian ini:

- a. Pengembangan Model CNN: CNN digunakan untuk mengekstraksi fitur tulisan tangan, seperti bentuk huruf dan tekanan.
- b. Implementasi *Siamese Network*: Membandingkan dua gambar tulisan tangan untuk mengukur kemiripan, membantu verifikasi keaslian tugas siswa.
- c. Pelatihan Model: Model dilatih menggunakan dataset tulisan tangan yang telah dikumpulkan untuk meningkatkan akurasi.
- d. Inferensi *Real-Time*: Model yang telah dilatih digunakan untuk memberikan hasil verifikasi secara cepat dan efisien.

Dengan TensorFlow/Keras, proyek ini dapat mengembangkan model *deep learning* yang handal untuk analisis pola tulisan tangan.

### 2.8.2 OpenCV

OpenCV (*Open-Source Computer Vision Library*) adalah pustaka *open-source* untuk pengolahan citra dan pengenalan pola. Pustaka ini menyediakan berbagai fungsi manipulasi dan analisis gambar, ideal untuk pra-pemrosesan data citra dalam penelitian ini. Penggunaan OpenCV dalam proyek ini

1. Pra-Pemrosesan Gambar:

- a. Konversi ke *grayscale* untuk mengurangi dimensi data.
  - b. Normalisasi ukuran gambar untuk konsistensi analisis.
  - c. Pembersihan *noise* untuk menghilangkan gangguan seperti latar belakang.
2. Segmentasi Karakter: Memisahkan karakter individu dalam tulisan tangan jika diperlukan.
  3. Augmentasi Data: Melakukan rotasi, *flipping*, atau penyesuaian skala untuk meningkatkan variasi dataset.

Dengan menggunakan OpenCV, data tulisan tangan dapat diproses dengan baik sebelum dimasukkan ke dalam model TensorFlow/Keras, memastikan kualitas data yang optimal untuk analisis.

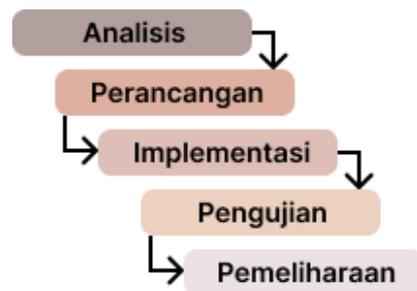
## BAB III METODOLOGI PENELITIAN

### 3.1 Pendekatan Penelitian

Pendekatan penelitian dalam proyek ini bersifat terapan, dengan tujuan utama mengembangkan sebuah sistem teknologi yang dapat membantu guru dalam memverifikasi keaslian tugas siswa berdasarkan analisis tulisan tangan. Penelitian ini memadukan elemen pengembangan perangkat lunak, pengolahan data, dan implementasi algoritma *Machine Learning* untuk menciptakan solusi inovatif di bidang pendidikan.

#### 3.1.1 Model pengembangan perangkat lunak

Model pengembangan perangkat lunak yang digunakan dalam proyek ini adalah model *Waterfall*. Model ini dipilih karena pendekatan bertahapnya yang sistematis dan terstruktur, sehingga cocok untuk pengembangan proyek dengan lingkup yang telah didefinisikan secara jelas. Tahapan dalam model ini meliputi:



**Gambar 3.1** Model *Waterfall*

1. Analisis Kebutuhan: Mengidentifikasi kebutuhan sistem berdasarkan permasalahan yang dihadapi guru dalam memverifikasi keaslian tugas siswa.

2. Perancangan Sistem: Membuat desain sistem, termasuk diagram ERD, *use case*, dan *Flowchart* sistem, untuk memvisualisasikan fungsi dan alur sistem.
3. Implementasi: Pengembangan aplikasi difokuskan pada platform mobile dengan menggunakan Flutter sebagai framework utama. Aplikasi ini akan menampilkan sistem berbasis web melalui WebView, yang memungkinkan akses langsung ke fitur-fitur verifikasi tulisan tangan tanpa perlu membangun ulang seluruh sistem untuk mobile. *Backend* akan menggunakan Flask dan Express.js.
4. Pengujian: Melakukan pengujian fungsi sistem dan model *Machine Learning* untuk memastikan sistem berjalan sesuai dengan yang diharapkan.
5. Pemeliharaan: Mencakup perbaikan dan peningkatan sistem berdasarkan umpan balik dari guru sebagai pengguna.

### 3.1.2 Pendekatan teknikal

1. Pendekatan Algoritmik: Sistem ini memanfaatkan algoritma *Convolutional Neural Network* (CNN) dan *Siamese Network* untuk mengenali pola tulisan tangan dan membandingkannya dengan data referensi yang tersimpan.
2. Pendekatan Pra-Pemrosesan Citra: Data tulisan tangan diproses menggunakan OpenCV untuk memastikan kualitas gambar yang optimal sebelum dianalisis oleh model.
3. Pendekatan Sistematis untuk Guru: Mengingat bahwa aplikasi ini dirancang khusus untuk guru, pendekatan yang digunakan berfokus

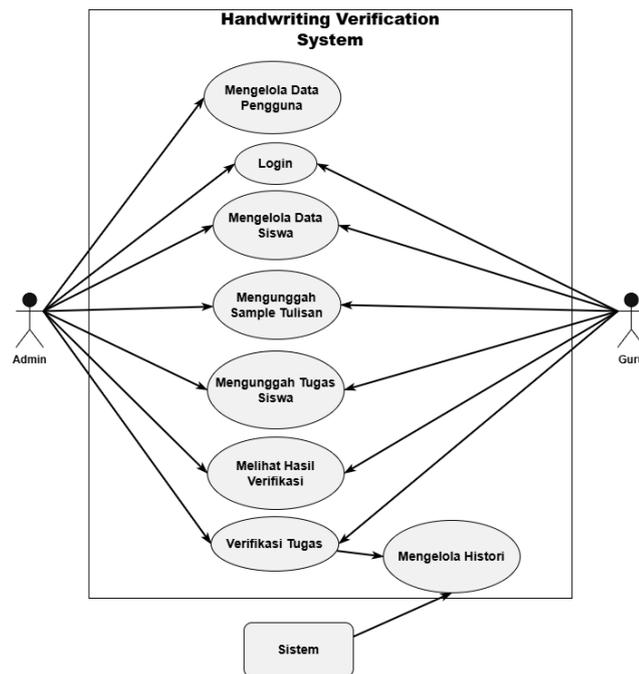
pada antarmuka yang sederhana dan intuitif, serta fitur yang relevan dengan kebutuhan guru.

### **3.2 Analisis Kebutuhan Sistem**

Analisis kebutuhan sistem bertujuan untuk mengidentifikasi dan mendefinisikan elemen-elemen penting yang harus ada dalam sistem agar dapat memenuhi tujuan dan fungsi yang diharapkan. Dalam proyek ini, kebutuhan sistem dibagi menjadi dua kategori utama: kebutuhan perangkat keras dan kebutuhan perangkat lunak.

#### **3.2.1 Diagram *use case***

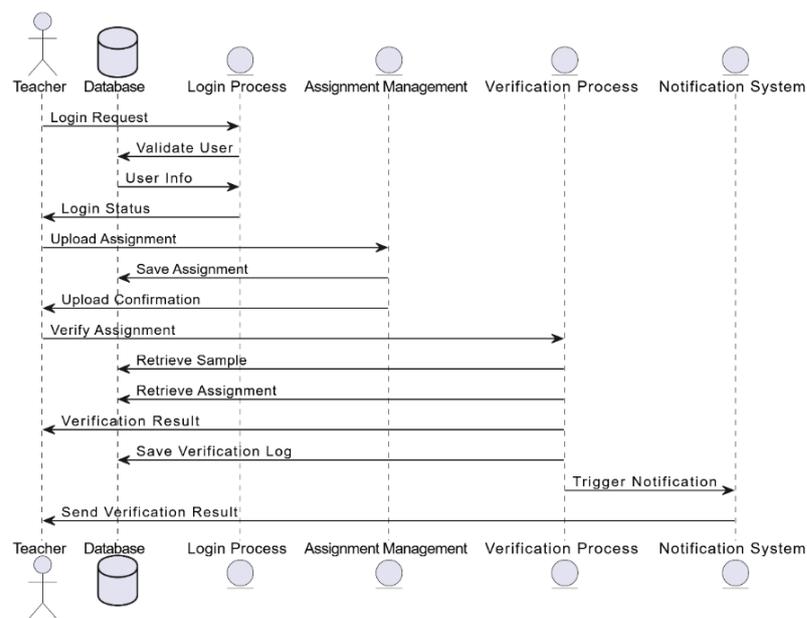
Diagram *use case* sistem Verifikasi Keaslian Tugas menggambarkan peran Guru, Admin, dan Sistem dalam proses verifikasi tugas siswa. Guru dapat *login*, mengelola data siswa, mengunggah sampel tulisan, mengunggah tugas siswa, melakukan verifikasi tugas, serta melihat hasil verifikasi. Admin memiliki akses untuk *login*, mengelola data siswa, dan mengunggah sampel tulisan. Sistem secara otomatis mengelola notifikasi dan menerima hasil dari proses verifikasi tugas, kemudian mengirimkan notifikasi kepada pengguna terkait. Berikut diagram *use case*.



Gambar 3.2 Diagram *Usecase*

### 3.2.2 Diagram *sequence*

Diagram *sequence* di bawah ini menggambarkan alur interaksi antara pengguna utama, yaitu guru, dengan sistem yang mencakup proses login, manajemen tugas, verifikasi tugas, dan pemberitahuan hasil verifikasi.



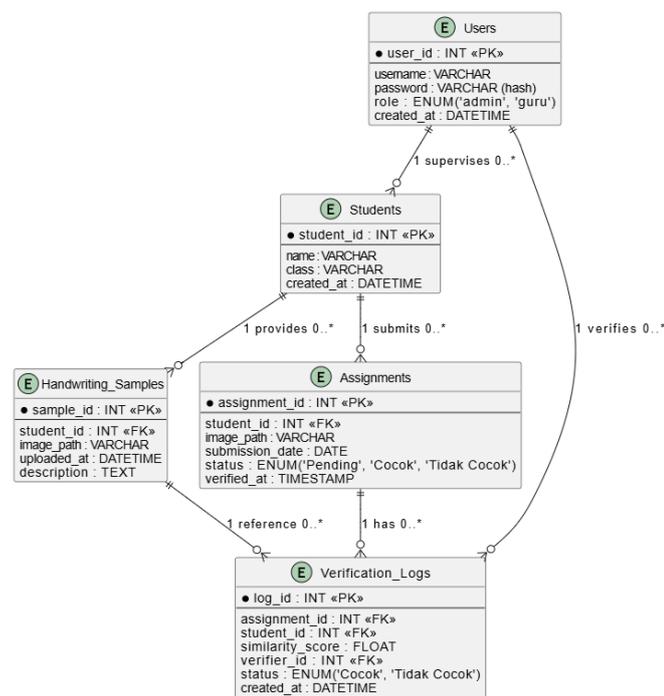
Gambar 3.3 Diagram *Sequence*

### 3.3 Desain Sistem

Bagian ini menjelaskan desain sistem yang digunakan dalam pengembangan aplikasi verifikasi keaslian tugas berbasis tulisan tangan. Desain sistem meliputi Diagram ERD (*Entity-Relationship Diagram*), *Flowchart* Sistem, dan *Flowchart* Pengguna, yang memberikan gambaran alur kerja dan relasi antar komponen dalam sistem.

#### 3.3.1 Diagram ERD (*Entity-Relationship Diagram*)

ERD dibawah ini menggambarkan sistem verifikasi tulisan tangan siswa menggunakan *Machine Learning* (ML). Alurnya dimulai dari pengguna mengunggah contoh tulisan tangan siswa, lalu sistem memverifikasi tugas siswa yang dikumpulkan, dan akhirnya guru melihat hasil verifikasi.



**Gambar 3.4** Diagram ERD (*Entity-Relationship Diagram*)

Berikut adalah penjelasan dari setiap entitas yang ada:

1. *Users*:

a. Menyimpan data pengguna (guru dan admin), termasuk kredensial login dan peran pengguna.

b. Kolom kunci utama: `user_id`.

2. *Students:*

a. Menyimpan data siswa, termasuk nama dan kelas.

b. Kolom kunci utama: `student_id`.

3. *Handwriting Sample:*

a. Menyimpan sampel tulisan tangan siswa untuk proses verifikasi.

b. Kolom kunci utama: `sample_id`.

c. Kolom kunci tamu: `student_id`.

4. *Assignments:*

a. Menyimpan tugas siswa yang diunggah untuk diverifikasi.

b. Kolom kunci utama: `assignment_id`.

c. Kolom kunci tamu: `student_id`.

5. *Verification logs:*

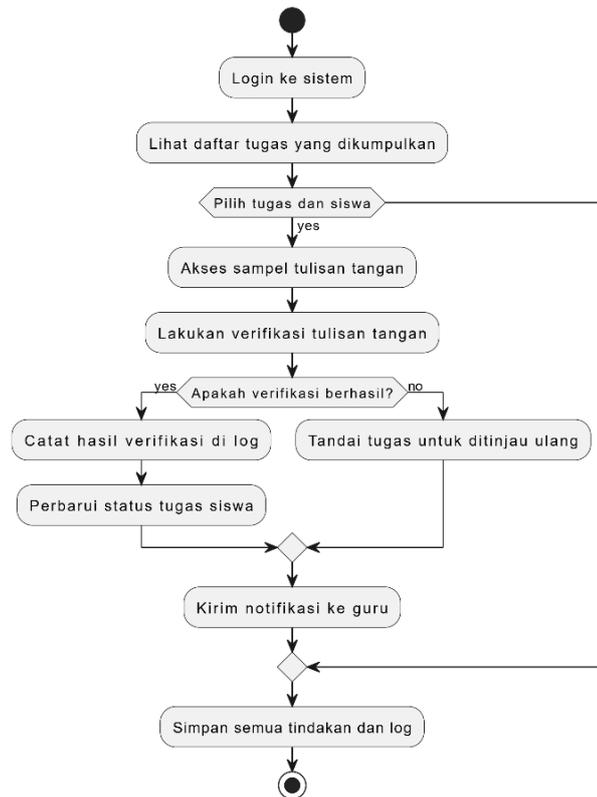
a. Mencatat hasil verifikasi tulisan tangan siswa, termasuk skor kecocokan dan status verifikasi.

b. Kolom kunci utama: `log_id`.

c. Kolom kunci tamu: `assignment_id`, `student_id`, dan `verifier_id`.

### 3.3.2 Flowchart sistem

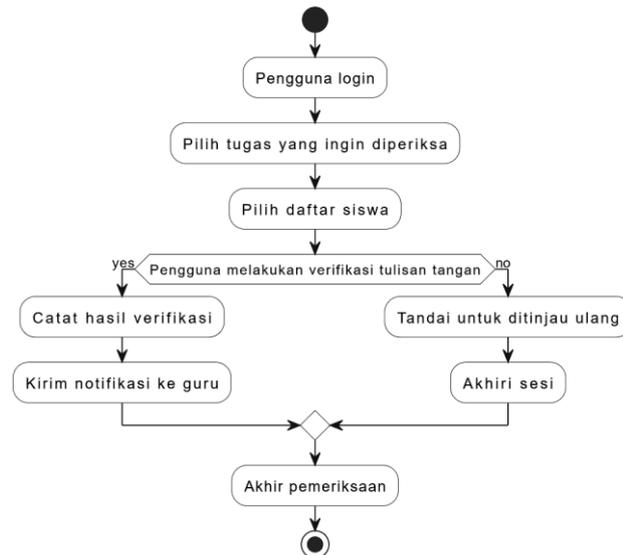
Flowchart sistem dapat dilihat pada gambar berikut:



Gambar 3.6 Flowchart Sistem

### 3.3.3 Flowchart pengguna

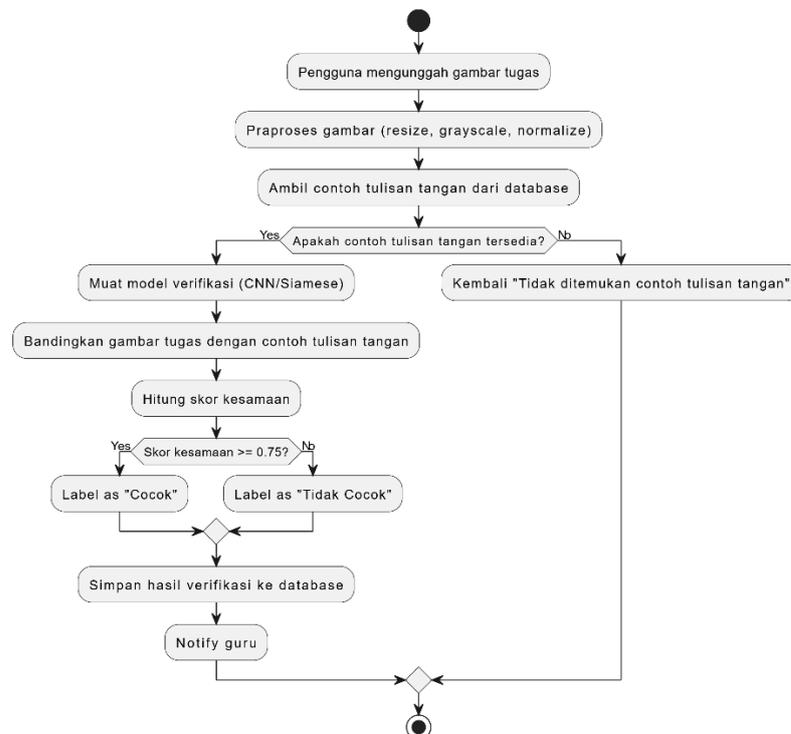
Flowchart pengguna dapat dilihat pada gambar berikut:



Gambar 3.7 Flowchart Pengguna

### 3.3.4 Flowchart algoritma

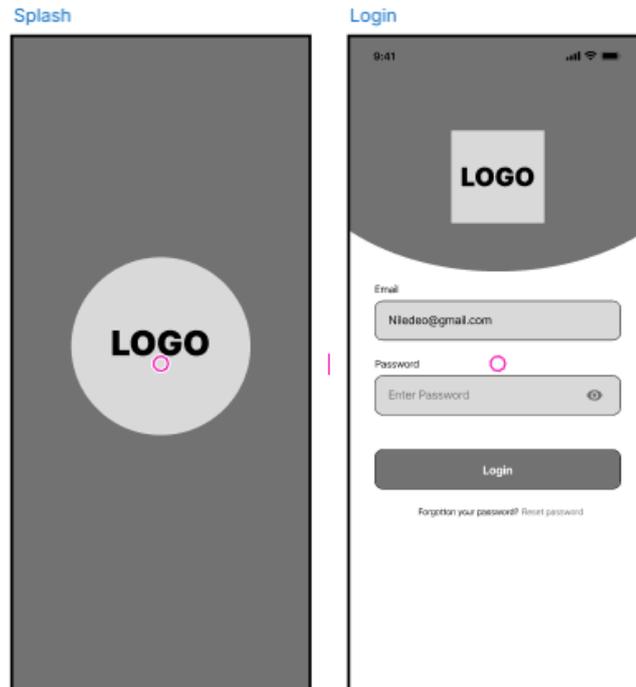
Flowchart algoritma langkah-langkah logika pemrosesan sistem verifikasi tulisan tangan menggunakan CNN dan Siamese Network pada gambar berikut:



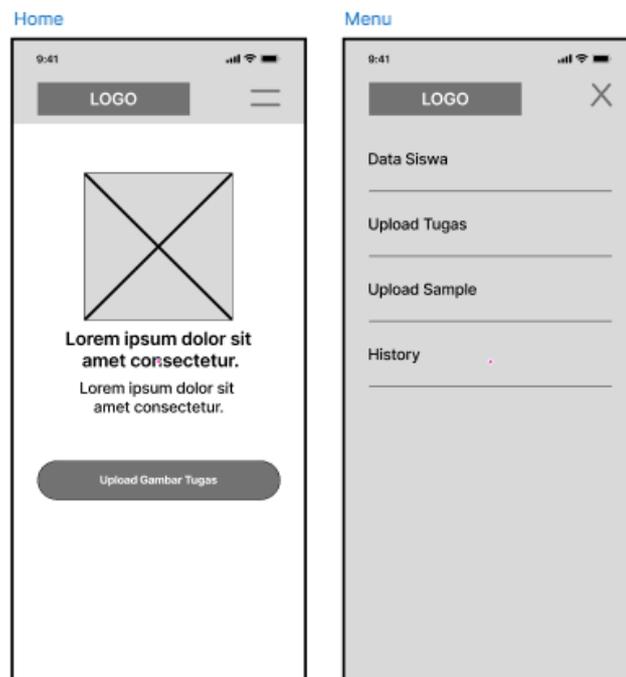
Gambar 3.8 Flowchart Algoritma

### 3.3.5 Wireframe app mobile

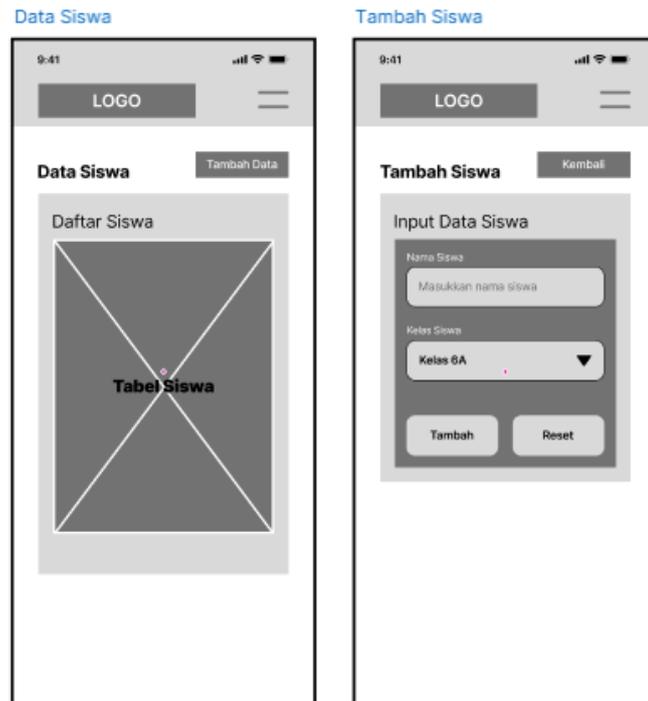
*Wireframe* adalah sebuah representasi visual yang sederhana dari tata letak dan struktur aplikasi ini, berikut dibawah ini adalah *Wireframe* aplikasi ini:



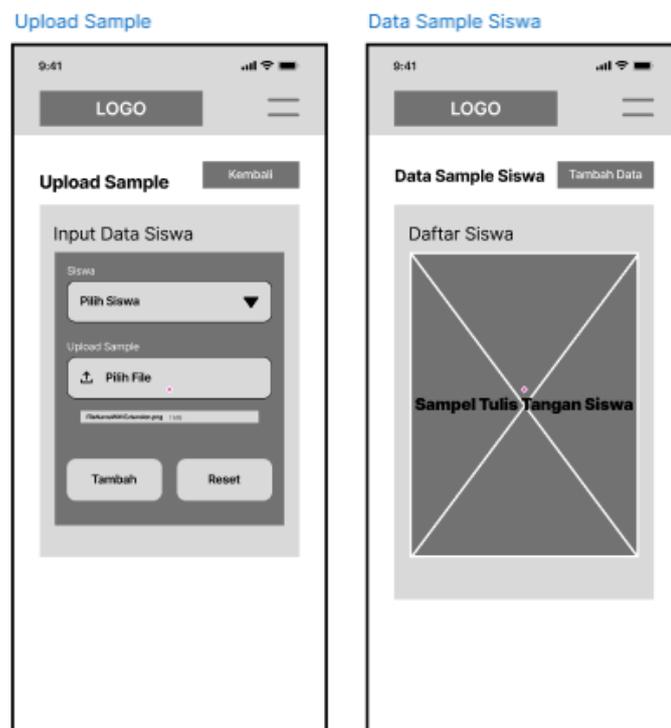
**Gambar 3.9** *Wireframe Splash dan Login*



**Gambar 3.10** *Wireframe Halaman Utama dan Menu*



**Gambar 3.11** Wireframe Halaman Daftar dan *Form Input* Siswa



**Gambar 3.12** Wireframe Daftar dan *Form Input* Sample

### 3.4 Pengumpulan Data

Pengumpulan data merupakan langkah penting dalam penelitian ini karena data yang diperoleh akan digunakan sebagai dasar untuk melatih dan menguji

model *Machine Learning* yang dikembangkan. Data yang dikumpulkan mencakup berbagai sampel tulisan tangan yang relevan untuk melatih dan menguji kemampuan model dalam mengenali keaslian tugas. Data ini terdiri dari dua kategori utama:

### 3.4.1 Data tulis tangan siswa

Data ini merupakan sampel tulisan tangan asli siswa kelas 6 SD Negeri 060868 Medan Timur. Sampel diambil dari catatan pelajaran, tugas harian, dan pekerjaan rumah yang telah dikumpulkan sebelumnya. Digunakan sebagai referensi utama untuk verifikasi keaslian tugas yang diunggah ke sistem.

1. Sampel dikumpulkan dengan izin siswa dan wali kelas.
2. Setiap sampel diberi label sesuai dengan identitas siswa (misalnya, nama dan kelas) untuk memastikan keterkaitan dengan data siswa dalam *Database*.
3. Gambar tulisan tangan di-*scan* atau difoto dalam resolusi tinggi untuk menjaga kualitas data.
4. Data akan di simpan kedalam folder genuine dan akan digunakan untuk pelatihan model.

**Tabel 3.1 Contoh Data tulis tangan siswa**

Label	Tipe
Student-01-Sample1.png	Genuine
Student-01-Sample2.png	Genuine
Student-01-Sample3.png	Genuine
Student-02-Sample1.png	Genuine
Student-02-Sample2.png	Genuine

### 3.4.2 Data tulis tangan bukan siswa

Sampel tulisan tangan dari individu yang bukan siswa, dikumpulkan dari berbagai sumber untuk memperkaya variasi dataset dan membantu model dalam mengenali tulisan tangan yang tidak relevan dengan siswa.

Sumber data meliputi:

1. Tulisan tangan teman, keluarga, dan relawan lainnya.
2. Tulisan tangan acak yang dikumpulkan dari buku, dokumen, atau kertas catatan yang tersedia di lingkungan sekitar.

Tujuan digunakan sebagai data pembandingan untuk membantu model mendeteksi perbedaan pola tulisan tangan, sehingga dapat membedakan antara tulisan asli siswa dan tulisan pihak lain.

**Tabel 3.2 Contoh Data bukan tulis tangan siswa**

<b>Label</b>	<b>Tipe</b>
forged_sample1.png	Forgery
forged_sample2.png	Forgery
forged_sample3.png	Forgery
forged_sample4.png	Forgery
forged_sample5.png	Forgery

### 3.4.3 Karakteristik *dataset*

*Dataset* mencakup total lebih dari masukan jumlah jika sudah ada sampel tulisan tangan, dengan distribusi antara tulisan siswa dan bukan siswa yang seimbang. Semua gambar tulisan tangan disimpan dalam format digital (JPG/PNG) dengan resolusi minimal 300 DPI untuk memastikan detail tulisan terlihat jelas.

1. Tulisan tangan berupa teks singkat, paragraf, atau jawaban soal.
2. Berbagai gaya penulisan seperti tulisan miring, tegak, atau campuran.

#### **3.4.4 Pengolahan data awal**

Setelah data terkumpul, langkah-langkah pengolahan data awal dilakukan untuk mempersiapkan data sebelum dimasukkan ke dalam proses pelatihan model *Machine Learning*. Langkah-langkah ini meliputi:

1. Konversi ke *Grayscale*: Semua gambar diubah menjadi format *grayscale* untuk mengurangi kompleksitas data.
2. Normalisasi Resolusi: Resolusi gambar disesuaikan agar seragam, misalnya 128x128 piksel.
3. Augmentasi Data: Teknik augmentasi seperti rotasi, *flipping*, dan *scaling* diterapkan untuk meningkatkan variasi dataset.

Dengan prosedur ini, data yang digunakan dalam penelitian memiliki kualitas yang memadai untuk mendukung pengembangan model *Machine Learning*.

### **3.5 Pengembangan Model *Machine Learning***

Pengembangan model *Machine Learning* merupakan inti dari penelitian ini. Model yang dikembangkan bertujuan untuk mengenali tulisan tangan siswa dan memverifikasi keaslian tugas berdasarkan pola tulisan yang telah dilatih sebelumnya.

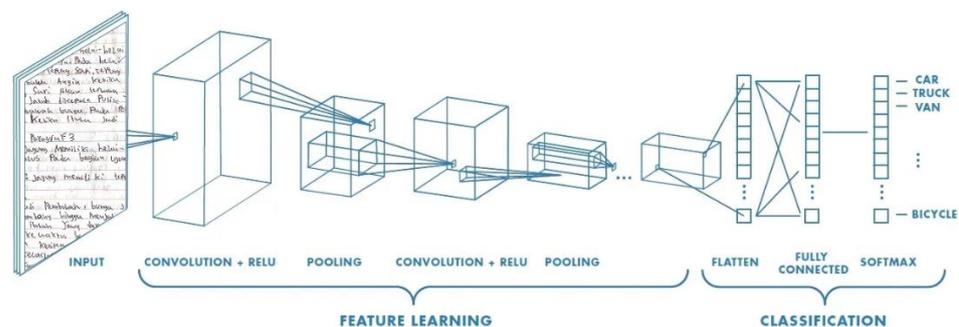
#### **3.5.1 Tujuan pengembangan model**

Model *Machine Learning* yang dikembangkan memiliki beberapa tujuan utama:

1. Menganalisis dan membandingkan pola tulisan tangan siswa.
2. Mengidentifikasi kesesuaian antara tugas yang diunggah dengan sampel tulisan tangan siswa yang telah tersimpan dalam *Database*.
3. Memberikan hasil berupa tingkat kesesuaian (*similarity score*) dan status verifikasi.

### 3.5.2 Arsitektur model

Model yang digunakan dalam penelitian ini berbasis deep learning dengan pendekatan *Convolutional Neural Network* (CNN). CNN dipilih karena kemampuannya dalam menangani data visual, seperti gambar tulisan tangan. Arsitektur model mencakup:



**Gambar 3.13** Struktur CNN

Sumber: Gunawan & Setiawan (2022)

1. *Input Layer*, gambar tulisan tangan siswa dalam format grayscale (128x128 piksel).
2. *Convolutional Layers*, *Layer* ini mengekstraksi fitur-fitur utama tulisan tangan, seperti pola garis, lengkungan, dan tekstur.
3. *Pooling Layers*, *Layer* ini mengurangi dimensi data untuk mempercepat proses komputasi tanpa kehilangan informasi penting.

4. *Fully Connected Layers*, *Layer* ini menghubungkan hasil ekstraksi fitur ke *Layer* output untuk menghasilkan prediksi akhir.
5. *Output Layer*, menghasilkan *similarity score* (dalam rentang 0-1) yang menunjukkan tingkat kesesuaian antara tulisan tangan tugas dan sampel.

### 3.5.3 Dataset pelatihan dan pengujian

Dataset yang digunakan untuk pengembangan model berasal dari tulisan tangan siswa kelas 6 SD Negeri 060868 Medan Timur. Dataset ini dibagi menjadi:

1. Training Set (80%): Untuk melatih model mengenali pola tulisan tangan.
2. Validation Set (10%): Untuk memvalidasi performa model selama proses pelatihan.
3. Testing Set (10%): Untuk menguji akurasi model setelah pelatihan selesai.

Augmentasi data dilakukan pada dataset untuk meningkatkan variasi pola tulisan tangan, seperti:

1. Rotasi
2. Penskalaan (*scaling*)
3. *Flipping* horizontal dan vertikal

### 3.5.4 Framework dan tools yang digunakan

Pengembangan model dilakukan menggunakan *framework* dan tools berikut:

1. *TensorFlow/Keras*, digunakan untuk membangun, melatih, dan menguji arsitektur CNN.
2. *OpenCV*, digunakan untuk *preprocessing* gambar, seperti konversi ke grayscale, normalisasi ukuran, dan augmentasi data.
3. Python, bahasa pemrograman utama untuk mengimplementasikan algoritma dan integrasi dengan sistem.

### 3.5.5 Proses pelatihan model

Proses pelatihan model meliputi langkah-langkah berikut:

1. Preprocessing Data
  - a. Semua gambar dikonversi ke format *grayscale* dan diubah ukurannya menjadi 128x128 piksel.
  - b. Data dinormalisasi agar nilai piksel berada dalam rentang 0-1.
2. Pelatihan Model
  - a. Model dilatih menggunakan dataset training dengan algoritma optimasi Adam Optimizer dan fungsi loss Binary *Cross entropy*.
3. Validasi Model
  - a. Performa model dievaluasi menggunakan dataset validasi pada setiap *epoch* untuk mencegah *overfitting*.
4. Pengujian Model
  - a. Setelah pelatihan selesai, model diuji menggunakan dataset testing untuk mengukur akurasi, presisi, *recall*, dan F1-*score*.

### 3.5.6 Output model

Model yang telah dilatih menghasilkan *output* berupa:

1. Similarity Score: Nilai yang menunjukkan tingkat kesesuaian tulisan tangan antara tugas dan sampel.
2. Status Verifikasi: Kategori hasil verifikasi, seperti "Cocok" atau "Tidak Cocok."

Dengan pendekatan ini, model *Machine Learning* yang dikembangkan diharapkan mampu memberikan hasil verifikasi yang akurat dan konsisten, mendukung tujuan utama dari sistem ini.

### **3.5.7 Evaluasi dan *Tuning Hyperparameter***

Evaluasi model dilakukan untuk memastikan performa yang optimal.

Beberapa hyperparameter yang diuji dan disesuaikan meliputi:

1. Learning Rate: Mengontrol kecepatan pembaruan bobot model.
2. Jumlah *Layer* CNN: Menentukan kompleksitas arsitektur.
3. Batch Size: Mengatur jumlah sampel yang digunakan dalam setiap iterasi pelatihan.
4. Epochs: Menentukan jumlah siklus pelatihan.

## **3.6 Perhitungan dalam *Convolutional Neural Network* (CNN)**

*Convolutional Neural Network* (CNN) digunakan dalam penelitian ini untuk mengekstraksi fitur tulisan tangan siswa sebelum dilakukan perbandingan menggunakan Siamese Network. Proses utama dalam CNN meliputi *Convolutional Layer*, *Pooling Layer*, dan *Fully Connected Layer*, yang masing-masing memiliki perhitungan sebagai berikut:

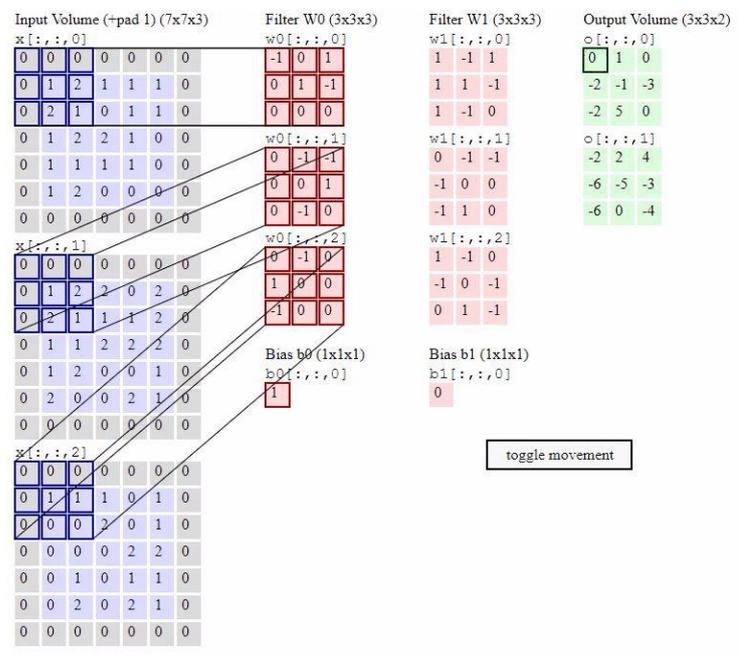
### *a. Convolutional Layer*

Lapisan konvolusi menerapkan operasi konvolusi untuk mengekstraksi fitur dari tulisan tangan. Perhitungan dilakukan dengan rumus:

$$\mathbf{O} = (\mathbf{I} - \mathbf{K}) + 1$$

Di mana:

- $\mathbf{O}$ : Ukuran *output* setelah konvolusi
- $\mathbf{I}$ : Ukuran gambar *input* (tinggi/ lebar)
- $\mathbf{K}$ : Ukuran *filter*



**Gambar 3.14** Strider (CS231n)

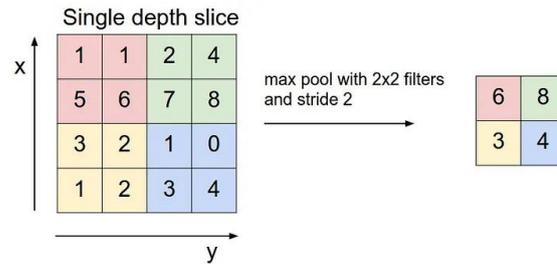
Jika gambar masukan berukuran 5×5 piksel dan *filter* yang digunakan adalah 3×3 dengan *stride* 1, maka ukuran keluaran dihitung sebagai berikut:

$$\mathbf{O} = (5 - 3) + 1 = 3$$

Sehingga, hasil keluaran dari lapisan konvolusi memiliki ukuran 3×3.

#### b. Pooling Layer

*Pooling* digunakan untuk mengurangi dimensi keluaran dari *Convolutional Layer* tanpa kehilangan informasi penting.



**Gambar 3.15** *Pooling Layer*

Dalam penelitian ini, digunakan *Max Pooling*, yang mengambil nilai maksimum dari setiap area tertentu dalam gambar. Sebagai contoh, jika diterapkan *Max Pooling*  $2 \times 2$  pada matriks berikut:

$$\begin{bmatrix} 1 & 3 & 2 & 4 \\ 5 & 6 & 8 & 7 \\ 9 & 10 & 12 & 11 \\ 13 & 14 & 16 & 15 \end{bmatrix}$$

Maka, hasil setelah *Max Pooling* akan menjadi:

$$\begin{bmatrix} 6 & 8 \\ 14 & 16 \end{bmatrix}$$

Dimensi data berkurang dari  $4 \times 4$  menjadi  $2 \times 2$ , sehingga mempercepat proses komputasi.

c. *Fully Connected Layer*

Lapisan ini mengubah keluaran CNN menjadi vektor fitur untuk klasifikasi akhir. Perhitungan dilakukan dengan dot product antara bobot dan *input*:

$$y = (W \cdot X) + b$$

Di mana:

- W** : bobot neuron
- X** : *input* dari lapisan sebelumnya
- b** : bias

Sebagai contoh, jika *input*  $X = [0.5, 1.0]$ , bobot  $W = [0.2, 0.8]$ , dan bias  $b = 0.1$ :

$$y = (0.5 \times 0.2) + (1.0 \times 0.8) + 0.1$$

$$y = 0.1 + 0.8 + 0.1 = 1.0$$

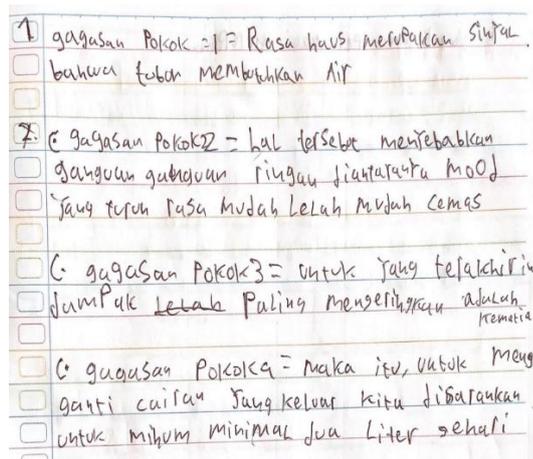
Hasil ini kemudian diterapkan ke fungsi aktivasi, seperti ReLU atau Softmax, untuk mendapatkan output akhir.

### 3.6.1 Hasil Pra-Pemrosesan Gambar

Sebelum gambar tulisan tangan siswa diproses oleh *Convolutional Neural Network* (CNN), diperlukan tahap pra-pemrosesan untuk memastikan bahwa data yang digunakan sesuai dengan format yang dibutuhkan oleh model. Pra-pemrosesan ini bertujuan untuk menelaraskan dimensi gambar, mengurangi *noise*, dan menyesuaikan skala warna agar lebih optimal dalam ekstraksi fitur. Tahapan yang dilakukan dalam pra-pemrosesan gambar adalah sebagai berikut:

1. Gambar asli tulis tangan siswa

Gambar asli merupakan hasil pemindaian atau pemotretan tulisan tangan siswa. Pada tahap ini, gambar masih dalam format aslinya dengan resolusi dan warna yang belum diubah.

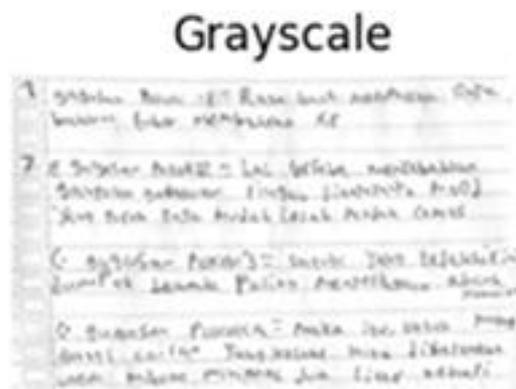


**Gambar 3.16** Tulis tangan asli siswa

## 2. Konversi ke *grayscale*

Langkah pertama dalam pra-pemrosesan adalah mengonversi gambar ke grayscale untuk menghilangkan informasi warna yang tidak diperlukan dalam analisis tulisan tangan. CNN umumnya lebih efektif dalam mengekstraksi fitur dari gambar hitam putih dibandingkan dengan gambar berwarna.

- a. Setiap piksel dalam gambar grayscale memiliki nilai intensitas antara 0 (hitam) hingga 255 (putih).
- b. Konversi ini membantu dalam menekankan pola tulisan tanpa gangguan warna latar belakang.



**Gambar 3.17** Hasil grayscale

## 3. *Resize* ke 128x 128 piksel

Setelah dikonversi ke grayscale, gambar diubah ukurannya menjadi  $128 \times 128$  piksel.

- a. Tujuan utama dari proses ini adalah untuk menyamakan ukuran *input* agar dapat diproses secara konsisten oleh model CNN.
- b. CNN memerlukan ukuran *input* yang seragam untuk memudahkan ekstraksi fitur dan mengurangi beban komputasi.



Gambar 3.18 Hasil Resize

### 3.6.2 Ekstraksi fitur dengan CNN

Setelah tahap pra-pemrosesan selesai, gambar tulisan tangan diproses melalui beberapa lapisan *Convolutional Neural Network* (CNN) untuk mengekstraksi fitur penting. Proses ekstraksi ini terdiri dari beberapa tahap sebagai berikut:

#### 1. Hasil Perhitungan Lapisan CNN

Pada tahap ini, gambar tulisan tangan diproses melalui sejumlah lapisan CNN, mulai dari *Convolutional Layer* hingga *Pooling Layer*. Setiap *Layer* memiliki bentuk keluaran (output shape) dan

nilai-nilai fitur yang berbeda, yang dihasilkan dari hasil konvolusi dan *pooling* terhadap gambar. Contoh hasil perhitungan untuk gambar siswa “Afifah-05.png” adalah sebagai berikut:

a. *Layer: Conv1*

*Output Shape:* (1, 126, 126, 16)

Contoh Feature Map (filter ke-0):

$$\begin{bmatrix} [0.094 & 0.095 & 0.092 \cdots 0.097 & 0.097 & 0.097] \\ [0.098 & 0.094 & 0.095 \cdots 0.097 & 0.097 & 0.097] \\ [0.087 & 0.073 & 0.116 \cdots 0.097 & 0.097 & 0.097] \end{bmatrix}$$

b. *Layer: Pool1*

*Output Shape:* (1, 63, 63, 16)

Contoh Feature Maps (filter ke-0):

$$\begin{bmatrix} [0.098 & 0.102 & 0.104 \cdots 0.097 & 0.097 & 0.097] \\ [0.128 & 0.149 & 0.102 \cdots 0.097 & 0.097 & 0.097] \end{bmatrix}$$

c. *Layer: Conv2*

*Output Shape:* (1, 61, 61, 32)

Contoh Feature Maps (filter ke-0):

$$\begin{bmatrix} [0.025 & 0.023 & 0.042 \cdots 0.037 & 0.046 & 0.038] \\ [0.083 & 0.042 & 0.052 \cdots 0.037 & 0.041 & 0.038] \end{bmatrix}$$

d. *Layer: Pool2*

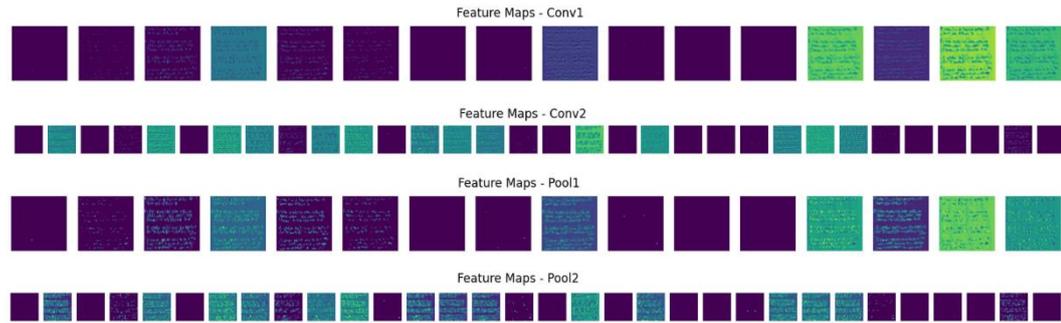
*Output Shape:* (1, 30, 30, 32)

Contoh Feature Maps (filter ke-0):

$$\begin{bmatrix} [0.083 & 0.052 & 0.044 \cdots 0.046] \\ [0.09 & 0.051 & 0.063 \cdots 0.045] \end{bmatrix}$$

## 2. Visualisasi *Feature Maps*

Untuk mempermudah pemahaman terhadap proses ekstraksi fitur, hasil aktivasi dari *Layer Conv1*, *Conv2*, *Pool1*, dan *Pool2* divisualisasikan dalam bentuk heatmap. Gambar-gambar berikut menunjukkan bagaimana CNN menangkap pola tulisan tangan dari gambar yang telah diproses.



**Gambar 3.19** Hasil *Convolutional Layer*

### 3. Hasil Ekstraksi Fitur 1D (*Fully Connected Layer*)

Setelah melalui beberapa lapisan konvolusi dan *pooling*, data yang semula berbentuk gambar diubah menjadi vektor fitur 1D yang merepresentasikan karakteristik tulisan tangan siswa dalam bentuk numerik.

- a. Setiap nilai dalam vektor fitur ini mewakili pola yang telah dipelajari oleh CNN dari tulisan tangan siswa.
- b. Vektor fitur ini digunakan dalam proses perbandingan tulisan siswa dengan tulisan sebelumnya untuk menentukan apakah tulisan tersebut asli atau ditulis oleh orang lain.

Misalnya, jika vektor fitur yang diekstraksi dari dua tulisan berbeda memiliki perbedaan kecil (dengan *cosine similarity* mendekati 1), maka tulisan dapat dianggap berasal dari siswa yang sama.

**Tabel 3.3** Hasil vektor fitur 1D

Indeks	Nilai	Indeks	Nilai	Indeks	Nilai	Indeks	Nilai
1	7.3126e-04	2	4.2615e-01	3	0.0000e+00	4	0.0000e+00
5	1.8811e-01	6	3.4448e-01	7	9.0793e-02	8	1.6498e-01
9	1.5039e-02	10	0.0000e+00	11	3.4448e-01	12	9.0793e-02

13	1.6498e-01	14	1.5039e-02	15	0.0000e+00	16	3.4448e-01
17	9.0793e-02	18	1.6498e-01	19	1.5039e-02	20	0.0000e+00
21	3.4448e-01	22	9.0793e-02	23	1.6498e-01	24	1.5039e-02
25	0.0000e+00	26	6.7814e-02	27	1.3547e-02	28	1.8623e-01
29	4.2493e-02	30	0.0000e+00	31	0.0000e+00	32	0.0000e+00
33	1.2640e-01	34	8.1493e-02	35	0.0000e+00	36	0.0000e+00
37	0.0000e+00	38	0.0000e+00	39	3.5408e-01	40	1.9182e-02
41	1.6472e-01	42	0.0000e+00	43	0.0000e+00	44	1.6255e-01
45	0.0000e+00	46	0.0000e+00	47	1.1614e-01	48	4.5026e-01
49	3.3839e-01	50	0.0000e+00	51	2.9820e-01	52	2.0186e-01
53	0.0000e+00	54	0.0000e+00	55	0.0000e+00	56	1.8069e-01
57	1.0904e-01	58	0.0000e+00	59	5.0056e-02	60	0.0000e+00
61	2.1611e-01	62	0.0000e+00	63	0.0000e+00	64	3.9370e-04
65	0.0000e+00	66	0.0000e+00	67	1.3598e-01	68	0.0000e+00
69	1.2085e-03	70	0.0000e+00	71	1.9767e-01	72	3.9307e-01
73	0.0000e+00	74	0.0000e+00				

### 3.6.3 Analisis Vektor Fitur dari *Fully Connected Layer*

Vektor fitur dari *Fully Connected Layer* berisi 74 nilai yang menunjukkan representasi numerik dari *input* setelah melewati beberapa lapisan konvolusi dan *pooling* dalam CNN. Setiap nilai dalam vektor ini mencerminkan aktivasi neuron pada *Layer* terakhir sebelum output diklasifikasikan.

## 1. Pola umum pada vektor fitur

### a. Nilai antara 0 hingga 0.45

Sebagian besar nilai berkisar antara 0 hingga 0.45, yang menunjukkan bahwa banyak neuron dalam *Layer* ini memiliki aktivasi sedang hingga tinggi, tetapi tidak mencapai maksimum.

### b. Banyaknya nilai 0 (0.0000e+00)

Terlihat bahwa ada beberapa nilai nol dalam vektor fitur, yang mengindikasikan bahwa beberapa neuron dalam *Fully Connected Layer* tidak diaktifkan sama sekali. Ini bisa disebabkan oleh fungsi aktivasi ReLU (*Rectified Linear Unit*), yang mengubah semua nilai negatif menjadi nol dan Neuron yang tidak cukup terstimulasi dari fitur sebelumnya.

### c. Pola berulang dalam fitur

Ada beberapa segmen dalam vektor yang tampaknya mengulang pola tertentu, seperti (3.4447e-01, 9.0793e-02, 1.6498e-01, 1.5039e-02, 0.0000e+00) yang muncul beberapa kali. Hal ini bisa terjadi karena struktur tulisan tangan yang mirip atau CNN mendeteksi pola tertentu yang dianggap penting untuk klasifikasi.

## 2. Interpretasi berdasarkan CNN

### a. Aktivasi tinggi (>0.3) pada beberapa indeks

Nilai yang cukup tinggi (misalnya 4.2615e-01, 3.4447e-01, 3.9307e-01) menunjukkan bahwa fitur yang terdeteksi pada

area tertentu dalam tulisan tangan lebih relevan dibanding yang lain.

- b. Aktivasi rendah ( $\sim 0.01 - 0.1$ ) pada beberapa indeks  
Beberapa nilai berada dalam rentang rendah, yang menunjukkan bahwa fitur yang ditemukan pada bagian tersebut kurang signifikan untuk klasifikasi.
- c. Banyaknya nol menunjukkan sparsity dalam fitur  
Ini menandakan bahwa hanya sebagian kecil dari fitur yang benar-benar signifikan.

### 3. Kesimpulan dari vektor fitur

- a. Vektor fitur ini adalah representasi tulisan tangan siswa dalam ruang dimensi tinggi.
- b. Aktivasi tinggi menunjukkan fitur dominan dalam tulisan, sedangkan nol menunjukkan bagian yang kurang signifikan.
- c. Beberapa pola berulang dapat menunjukkan struktur tulisan yang memiliki pola yang dapat dikenali oleh CNN.
- d. Jika fitur yang diekstrak cukup diskriminatif, model seharusnya bisa membedakan antara tulisan siswa yang satu dengan yang lain.

### 3.7 Perhitungan dalam *Siamese Network* (SN)

Setelah CNN mengekstraksi fitur tulisan tangan, *Siamese Network* membandingkan tulisan tangan siswa dengan tugas yang dikumpulkan. Perhitungan utama dalam SN menggunakan *Euclidean Distance*, yang dirumuskan sebagai berikut:

$$d(X_1, X_2) = \sqrt{\sum (X_1 - X_2)^2}$$

Di mana:

- $X_1$  : vektor fitur dari tulis tangan siswa dalam *Database*
- $X_2$  : vektor fitur dari tulisan tangan pada tugas yang di unggah
- $d$  : nilai jarak antara kedua vektor (semakin kecil, semakin mirip)

Sebagai contoh Misalkan dua vektor fitur yang dihasilkan dari gambar tulisan tangan adalah:

$$X_1 = [0.2, 0.5, 0.7]$$

$$X_2 = [0.3, 0.4, 0.6]$$

Maka, jarak *Euclidean* dihitung sebagai berikut:

$$\begin{aligned} d(X_1, X_2) &= \sqrt{(0.2 - 0.3)^2 + (0.5 - 0.4)^2 + (0.7 - 0.6)^2} \\ &= \sqrt{(0.1)^2 + (-0.1)^2 + (0.1)^2} \\ &= \sqrt{0.01 + 0.01 + 0.01} = \sqrt{0.03} \approx 0.173 \end{aligned}$$

- Jika  $d \geq 0.75$  → Tulis tangan dianggap cocok.
- Jika  $d \geq 0.75$  → Tulis tangan dianggap berbeda.

Nilai ambang batas ini ditentukan berdasarkan pelatihan model, di mana model akan belajar untuk membedakan antara tulisan asli siswa dan tulisan tangan bukan siswa.

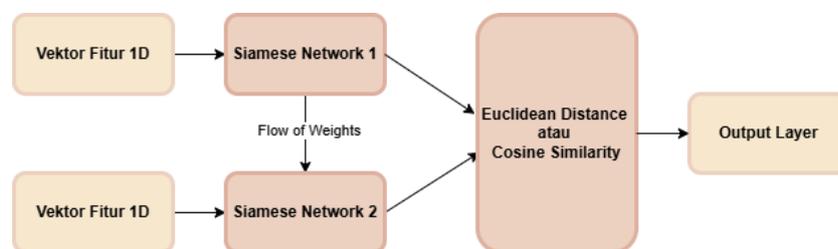
### 3.8 Implementasi *Siamese Network* dalam Verifikasi Tulis Tangan

*Siamese Network* adalah arsitektur jaringan saraf tiruan yang dirancang untuk membandingkan dua *input* dan menentukan tingkat kesamaan di antara keduanya. Dalam penelitian ini, *Siamese Network* digunakan untuk membandingkan vektor fitur 1D dari dua gambar tulisan tangan dan menentukan apakah keduanya berasal dari individu yang sama atau tidak.

Arsitektur *Siamese Network* terdiri dari dua jaringan kembar (*twin networks*) yang memiliki parameter bobot yang sama dan bertujuan untuk mengekstrak fitur dari setiap *input*. Hasil ekstraksi fitur ini kemudian dibandingkan menggunakan fungsi jarak untuk memperoleh nilai kesamaan.

### 3.8.1 Arsitektur Siamese Network

Arsitektur Jaringan Siam terdiri dari dua jaringan saraf konvolusional yang berjalan secara paralel pada dua gambar yang berbeda. Arsitektur *Siamese Network* dalam penelitian ini terdiri dari tiga komponen utama:



**Gambar 3.20** Arsitektur *Siamese Network*

#### 1. Dua jaringan CNN kembar

Setiap jaringan menerima satu gambar tulisan tangan yang telah melalui tahap pra-pemrosesan. Proses ekstraksi fitur dilakukan dengan lapisan konvolusi, *pooling*, dan *Fully Connected Layer*, menghasilkan vektor fitur 1D. Output dari kedua jaringan adalah dua vektor fitur dengan dimensi yang sama.

#### 2. Lapisan Perbandingan (*Similarity Calculation Layer*)

Kedua vektor fitur dibandingkan dengan menggunakan fungsi jarak. Fungsi jarak yang umum digunakan adalah *Euclidean Distance* atau *Cosine Similarity*, yang mengukur tingkat kesamaan antara dua vektor fitur. Jika nilai jarak rendah, maka tulisan kemungkinan besar berasal dari individu yang

sama. Jika nilai jarak tinggi, maka tulisan berasal dari individu yang berbeda.

### 3. Lapisan Keputusan (*Output Layer*)

Hasil perhitungan fungsi jarak diolah melalui fungsi aktivasi sigmoid, yang mengubah nilai kesamaan menjadi probabilitas antara 0 dan 1. Jika nilai probabilitas lebih dari threshold tertentu ( $\geq 0.75$ ), maka kedua tulisan dianggap berasal dari individu yang sama, sedangkan jika kurang dari threshold ( $\leq 0.75$ ), maka dianggap berasal dari individu yang berbeda.

## 3.8.2 Proses Implementasi *Siamese Network*

### 1. *Training Model*

Dataset terdiri dari pasangan tulisan tangan, di mana ada positive pair (dua gambar dari siswa yang sama) dan negative pair (dua gambar dari siswa berbeda). Model dilatih menggunakan *Contrastive Loss* atau *Triplet Loss*, yang bertujuan untuk mengurangi jarak antara pasangan tulisan yang sama dan meningkatkan jarak antara pasangan tulisan yang berbeda.

### 2. Evaluasi Model

Setelah model selesai dilatih, kinerjanya diuji menggunakan akurasi verifikasi dengan dataset baru. Model dinilai berdasarkan kemampuannya membedakan tulisan asli siswa dengan tulisan dari individu lain.

### 3. Pengujian pada Data nyata

Pada tahap pengujian, model menerima dua gambar tulisan tangan yang telah melalui proses ekstraksi fitur menggunakan CNN. *Siamese Network* menghitung tingkat kesamaan dan memberikan keputusan apakah tulisan tersebut berasal dari individu yang sama atau tidak.

### **3.9 Implementasi Sistem**

Implementasi sistem adalah tahap di mana desain dan perencanaan sistem diterapkan ke dalam bentuk yang dapat digunakan oleh pengguna. Pada tahap ini, sistem dibuat sesuai dengan arsitektur yang telah dirancang untuk memastikan fungsionalitas berjalan dengan baik. Implementasi sistem ini terdiri dari beberapa aspek penting.

#### **3.9.1 Lingkungan implementasi**

Sistem dikembangkan menggunakan teknologi yang sudah direncanakan sebelumnya. Berikut adalah detail lingkungan implementasi:

1. *Back-end: Framework* Flask (Python) untuk mengelola logika bisnis dan komunikasi dengan *Database*.
2. *Database:* MySQL untuk menyimpan data pengguna, tugas, sampel tulisan tangan, log verifikasi, dan notifikasi.
3. *Mobile Framework:* Flutter digunakan untuk mengembangkan aplikasi mobile berbasis WebView yang menampilkan konten website.

#### **3.9.2 Integrasi komponen**

Sistem ini terdiri dari beberapa komponen utama yang diintegrasikan untuk memastikan alur kerja sistem berjalan lancar:

1. Antarmuka Login

a. Guru dapat login ke sistem dengan memasukkan kredensial. Validasi dilakukan di sisi server menggunakan data dari *Database*.

## 2. Manajemen Tugas

a. Guru dapat mengunggah tugas siswa, melihat daftar tugas yang telah dikumpulkan, dan mengakses detail tugas tertentu.

b. Data tugas disimpan di *Database* dengan status awal "Pending" sebelum diverifikasi.

## 3. Proses Verifikasi

a. Sistem melakukan verifikasi tulisan tangan menggunakan model *Machine Learning* yang telah dikembangkan.

b. Model menerima gambar tulisan tangan siswa dan membandingkannya dengan sampel yang relevan untuk menentukan kecocokan.

c. Hasil verifikasi dicatat dalam *Database* dan log verifikasi.

## 4. Pengelolaan Notifikasi

a. Sistem mengirim notifikasi kepada guru dan siswa setelah proses verifikasi selesai.

b. Notifikasi berisi informasi tentang status tugas (Cocok/Tidak Cocok) dan tindakan lebih lanjut jika diperlukan.

### 3.9.3 Proses *deployment*

Sistem di-deploy pada server lokal dan/atau cloud untuk memastikan aksesibilitas yang optimal. Langkah-langkah *deployment* meliputi:

1. *Hosting website*, Front-end dan back-end di-deploy menggunakan layanan hosting seperti AWS, DigitalOcean, atau platform lokal yang disediakan sekolah.
2. *Database management*, Database MySQL dikelola pada server yang sama atau server terpisah untuk menjaga keamanan dan skalabilitas.
3. *Aplikasi mobile*, Aplikasi mobile dikembangkan menggunakan Flutter, dengan mengintegrasikan WebView untuk menampilkan antarmuka website dalam aplikasi.

#### **3.9.4 Validasi dan pengujian awal**

Setelah implementasi selesai, validasi dilakukan untuk memastikan bahwa setiap fitur berjalan sesuai dengan kebutuhan pengguna. Pengujian awal mencakup:

1. Pengujian Login: Memastikan hanya guru yang dapat mengakses sistem.
2. Pengujian Unggahan: Memastikan tugas dapat diunggah dengan benar.
3. Pengujian Verifikasi: Memastikan model *Machine Learning* dapat memberikan hasil yang akurat.
4. Pengujian Notifikasi: Memastikan notifikasi dikirimkan ke guru sesuai dengan hasil verifikasi.

#### **3.9.5 Dokumentasi**

Seluruh proses implementasi didokumentasikan dengan rinci, termasuk struktur direktori proyek, konfigurasi server, langkah-langkah

*deployment*, dan panduan penggunaan sistem. Dokumentasi ini berguna untuk pemeliharaan sistem di masa mendatang.

### **3.10 Pengujian Sistem**

Pengujian sistem dilakukan untuk memastikan bahwa sistem yang dikembangkan dapat berfungsi sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Pada tahap ini, metodologi pengujian diterapkan untuk memverifikasi dan memvalidasi setiap komponen sistem, baik secara individu maupun secara keseluruhan.

#### **3.10.1 Metodologi pengujian**

Metodologi pengujian yang digunakan dalam proyek ini meliputi pengujian unit, pengujian integrasi, dan pengujian sistem secara menyeluruh. Berikut adalah penjelasan setiap metode pengujian:

##### **1. Pengujian unit**

Pengujian ini bertujuan untuk memverifikasi bahwa setiap unit atau komponen sistem berfungsi dengan benar. Unit yang diuji meliputi:

- a. Fungsi login dan validasi pengguna.
- b. Fungsi unggah tugas siswa.
- c. Fungsi pemrosesan data tulisan tangan.
- d. Fungsi pengiriman notifikasi.

Metode pengujian dilakukan dengan menggunakan data uji untuk setiap unit secara terpisah.

##### **2. Pengujian integrasi**

Pengujian ini dilakukan untuk memastikan bahwa integrasi antar komponen berjalan dengan baik dan tidak terjadi konflik.

Pengujian dilakukan pada:

- a. Integrasi antara back-end dan *Database* untuk penyimpanan data tugas dan hasil verifikasi.
- b. Integrasi antara front-end dan back-end untuk pengambilan dan pengiriman data.
- c. Integrasi antara model *Machine Learning* dengan sistem verifikasi.

Metode data simulasi digunakan untuk menguji komunikasi antar komponen.

### 3. Pengujian sistem

Pengujian sistem secara menyeluruh dilakukan untuk memastikan bahwa sistem bekerja sesuai dengan spesifikasi yang telah ditentukan. Aspek yang diuji meliputi:

- a. Fungsionalitas utama sistem, seperti login, unggah tugas, verifikasi tulisan tangan, dan pengiriman notifikasi.
- b. Kemampuan sistem menangani berbagai skenario penggunaan, termasuk skenario normal, ekstrem, dan skenario kesalahan.

Metode data dari pengguna sebenarnya digunakan untuk pengujian sistem secara keseluruhan.

### 4. Pengujian antarmuka

Pengujian ini dilakukan untuk memverifikasi bahwa antarmuka sistem, baik pada website maupun aplikasi mobile,

dapat digunakan dengan baik oleh pengguna. Aspek yang diuji meliputi:

- a. Konsistensi tampilan antarmuka.
- b. Kemudahan navigasi.
- c. Responsivitas antarmuka pada berbagai perangkat.

Metode: Pengujian dilakukan dengan melibatkan pengguna akhir (guru) dalam simulasi penggunaan.

### 3.11 Tempat dan Waktu Penelitian

Penelitian ini dilakukan di SD Negeri 060868 Medan Timur, yang menjadi lokasi utama dalam pengumpulan data tulisan tangan siswa sebagai bagian dari proses pengembangan sistem verifikasi keaslian tugas. Data yang dikumpulkan mencakup sampel tulisan tangan siswa kelas 6, yang digunakan sebagai dataset untuk melatih dan menguji model *Machine Learning*.

Penelitian ini dilaksanakan dalam beberapa tahapan selama periode 2 Desember 2024 hingga 20 Desember 2024, dengan rincian sebagai berikut:

**Tabel 3.4 Waktu Penelitian**

No	Kegiatan	Desember	Januari	Februari	Maret	April
1	Menyusun Proposal	■	■	■	■	■
2	Seminar Proposal			■	■	■
3	Perbaikan Proposal				■	■
4	Pelaksanaan Penelitian			■	■	■
5	Pengolahan dan Penyusunan Laporan Hasil Penelitian			■	■	■
6	Seminar Hasil Penelitian Sidang					■

7	Perbaikan Hasil Penelitian																	
---	-------------------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Pendahuluan

Bab ini menyajikan hasil implementasi sistem verifikasi keaslian tugas sekolah berdasarkan analisis tulisan tangan. Sistem ini dibangun dengan memanfaatkan teknologi *Machine Learning*, khususnya menggunakan kombinasi algoritma *Convolutional Neural Network* (CNN) dan *Siamese Network* (SN). Proses implementasi mencakup tahapan pra-pemrosesan gambar, ekstraksi fitur menggunakan CNN, serta penghitungan tingkat kesamaan menggunakan *Siamese Network* untuk menentukan apakah dua gambar tulisan tangan berasal dari individu yang sama.

Algoritma CNN berperan dalam mengekstraksi fitur visual dari gambar tulisan tangan siswa. Melalui beberapa tahapan seperti konvolusi, *pooling*, dan *Fully Connected Layer*, CNN mampu mengubah gambar menjadi representasi vektor fitur berdimensi satu (1D feature vector) yang merepresentasikan ciri khas tulisan tangan tersebut.

Selanjutnya, vektor fitur dari dua gambar tulisan tangan dibandingkan menggunakan Siamese Network. Algoritma ini mengukur tingkat kemiripan antara dua vektor fitur dengan menggunakan metode *Euclidean Distance*. Hasil perhitungan ini kemudian digunakan untuk memutuskan apakah dua gambar tulisan tangan berasal dari orang yang sama atau tidak, berdasarkan ambang batas kesamaan yang telah ditentukan sebelumnya.

Dengan implementasi sistem ini, diharapkan dapat membantu pihak sekolah dalam melakukan verifikasi keaslian tugas siswa, khususnya untuk mengurangi

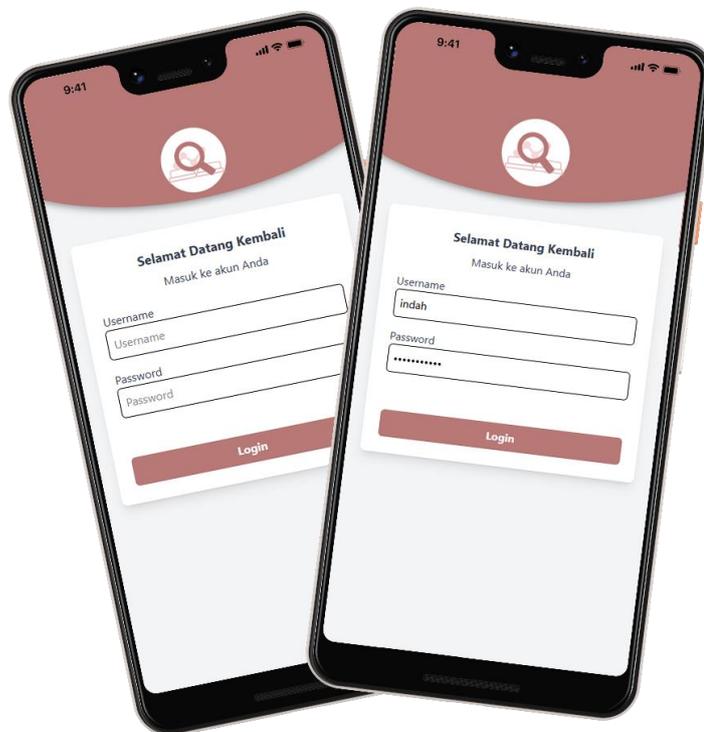
kecurangan seperti peniruan tulisan tangan oleh pihak lain. Adapun penjelasan lebih lanjut mengenai tampilan antarmuka sistem, hasil pengujian, dan analisis performa sistem akan dibahas pada subbab-subbab berikutnya.

## 4.2 Hasil Implementasi Sistem

### 4.2.1 Halaman *login*

Halaman login berfungsi sebagai akses masuk bagi guru yang melakukan verifikasi tugas. Pengguna diminta memasukkan username dan password yang diberikan oleh admin. Jika data login salah, sistem menampilkan notifikasi kesalahan. Setelah berhasil login, pengguna diarahkan ke halaman utama sistem.

Berikut tampilan halaman login pada sistem verifikasi keaslian tugas siswa:



**Gambar 4.1** Halaman *Login*

#### 4.2.2 Halaman *dashboard*

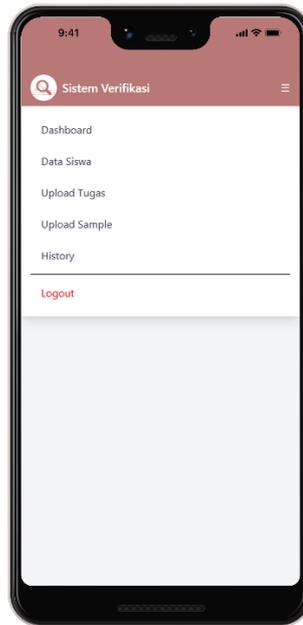
Setelah berhasil login, pengguna akan diarahkan ke halaman dashboard utama. Halaman ini merupakan titik awal interaksi pengguna dengan sistem dan dirancang dengan tampilan yang sederhana dan informatif. Tampilan dashboard menyajikan Teks ucapan selamat datang, penjelasan singkat mengenai fungsi sistem, dan tombol “Upload Tugas” berfungsi untuk mengarahkan pengguna ke halaman unggah gambar tugas siswa.



**Gambar 4.2** Halaman *Dashboard*

#### 4.2.3 Menu Utama

Setiap halaman, terdapat ikon menu navigasi (hamburger menu) yang terletak di bagian kanan atas layar. Ketika diklik, menu ini akan menampilkan beberapa opsi navigasi yang dapat diakses oleh guru, seperti gambar dibawah:

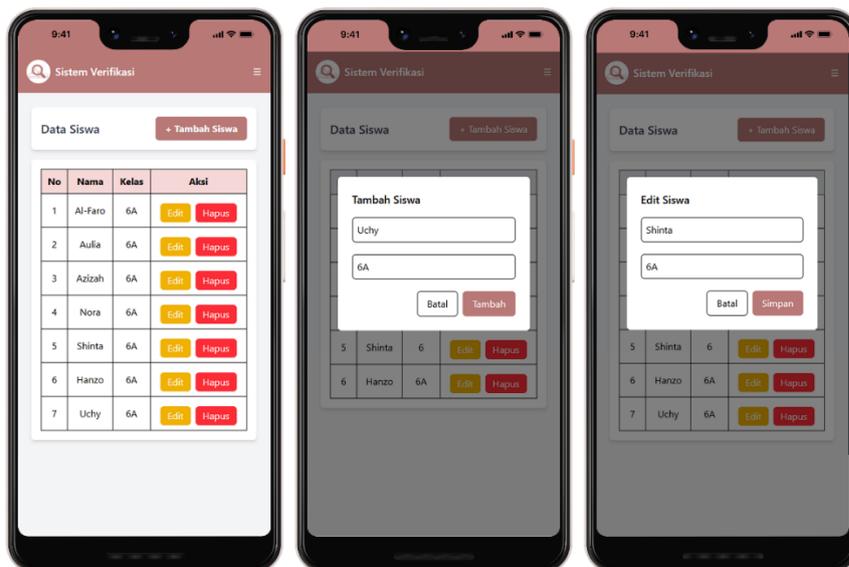


**Gambar 4.2** Menu Utama

#### 4.2.4 Halaman siswa

Halaman Siswa digunakan untuk mengelola data siswa yang terlibat dalam proses verifikasi tulisan tangan, termasuk pengelompokan hasil unggahan tugas.

Terdapat tombol Tambah Siswa untuk membuka form *input* nama dan kelas. Data siswa ditampilkan dalam tabel, lengkap dengan tombol Edit untuk mengubah data dan Delete untuk menghapus siswa dari sistem.



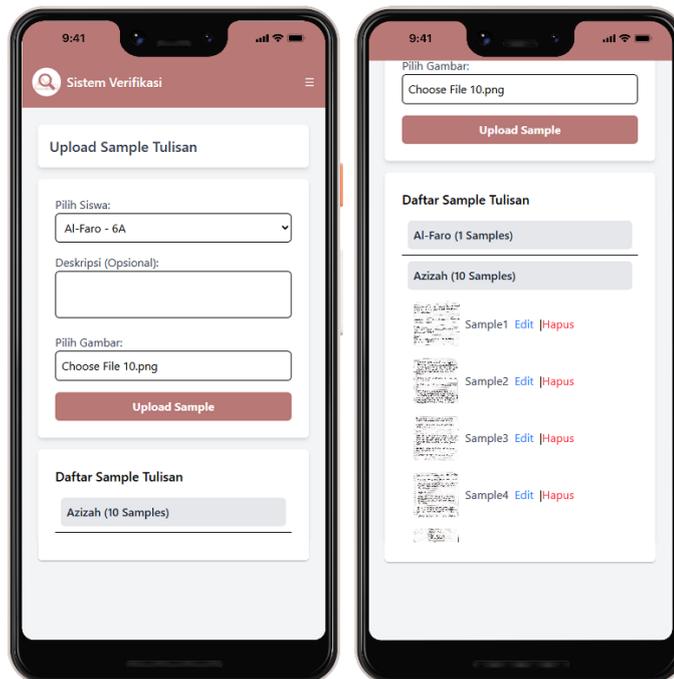
**Gambar 4.3** Halaman Siswa

#### 4.2.5 Halaman sample tulisan

Halaman Sample Tulisan digunakan untuk mengunggah dan mengelola contoh tulisan tangan siswa yang akan dijadikan acuan dalam proses verifikasi menggunakan model Siamese Network.

Guru dapat mengunggah sample melalui form yang terdiri dari dropdown nama siswa dan unggahan gambar (.jpg/.png). Data yang dikirim akan disimpan dan dikaitkan dengan nama siswa.

Di bawah form, terdapat daftar siswa yang sudah memiliki sample. Nama siswa dapat diklik untuk menampilkan gambar-gambar sample yang telah diunggah. Setiap gambar disertai tombol Edit untuk memperbarui dan Delete untuk menghapus sample dari sistem.



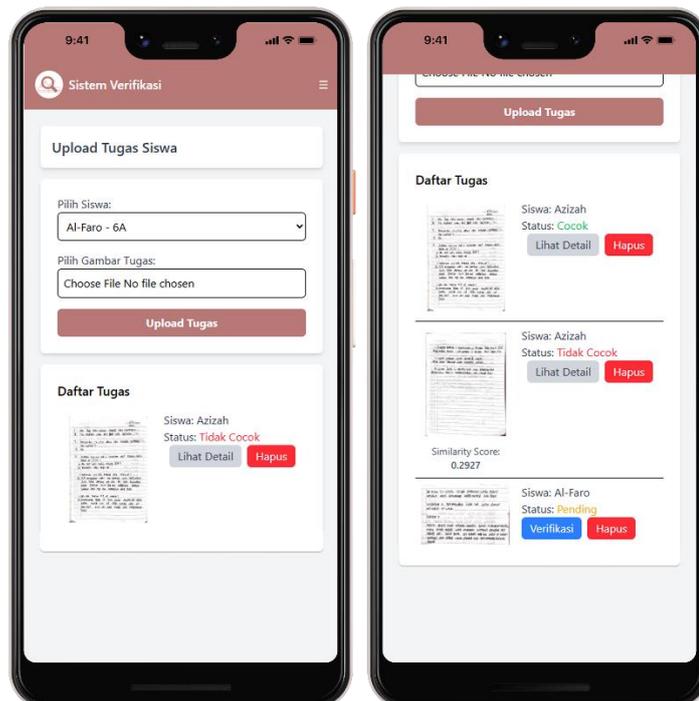
Gambar 4.4 Halaman Sample Tulisan

#### 4.2.6 Halaman tugas siswa

Halaman Upload Tugas adalah komponen utama sistem verifikasi, tempat guru mengunggah tugas siswa untuk dibandingkan dengan sample tulisan tangan menggunakan model Siamese Network.

Halaman ini terdiri dari Form Upload Tugas dan Daftar Tugas. Form berisi dropdown Nama Siswa (yang sudah memiliki sample) dan Upload Gambar Tugas. Daftar tugas menampilkan nama siswa, gambar tugas, status verifikasi (Pending, Cocok, atau Tidak Cocok), serta tombol aksi. Jika status masih Pending, tersedia tombol Verifikasi untuk memulai proses analisis. Setelah verifikasi, status berubah dan tombol Lihat Detail muncul untuk menampilkan nilai Similarity Score. Tersedia juga tombol Delete untuk menghapus tugas.

Sistem hanya menerima unggahan jika siswa sudah memiliki sample tulisan tangan. Jika belum, akan muncul pesan error. Jika syarat terpenuhi, tugas disimpan dan langsung ditampilkan di daftar.



**Gambar 4.5** Halaman Tugas Siswa

#### 4.2.7 Implementasi teknis proses verifikasi

Saat tombol Verifikasi diklik oleh pengguna pada daftar tugas, sistem secara otomatis menjalankan proses verifikasi keaslian tulisan tangan

menggunakan model *Siamese Network* yang telah dilatih sebelumnya dan disimpan dalam format file `.h5`. Proses ini melibatkan langkah-langkah sebagai berikut:

### 1. Memuat model Siamese network

```
def load_siamese_model(build_model_fn):
    """Muat model Siamese dari bobot yang telah dilatih"""
    global model
    if model is None:
        model = build_model_fn()
        model.load_weights(MODEL_PATH)
        print("✅ Model Siamese berhasil dimuat!")
```

**Gambar 4.6** Kode Memuat Model

Model dimuat satu kali melalui fungsi `load_siamese_model()`, yang memanggil arsitektur model dari fungsi `build_model_fn()` dan mengimpor bobot terlatih dari file `.h5`. Model disiapkan di backend agar siap digunakan untuk prediksi kapan pun diperlukan.

### 2. Pra-pemrosesan gambar

```
def preprocess_image(image_path):
    if not os.path.exists(image_path):
        raise FileNotFoundError(f"Gambar tidak ditemukan: {image_path}")

    img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    if img is None:
        raise ValueError(f"File gambar tidak valid atau rusak: {image_path}")

    img = cv2.resize(img, (IMG_SIZE, IMG_SIZE))
    img = img.astype("float32") / 255.0
    return np.expand_dims(img, axis=[0, -1])
```

**Gambar 4.7** Kode Pra-pemrosesan gambar

Gambar tugas yang diunggah oleh guru dikonversi menjadi grayscale, diubah ukurannya menjadi 128x128 piksel, dan dinormalisasi agar sesuai dengan *input* model. Proses ini dilakukan oleh fungsi `preprocess_image()`.

### 3. Perbandingan dengan sample tulisan asli

```

# Ambil sampel tulisan asli siswa dari tabel HandwritingSamples
samples = HandwritingSample.query.filter_by(student_id=student_id).all()
if not samples:
    return jsonify({"message": "X Tidak ada sampel tulisan untuk siswa ini!"}), 400

# Bandingkan gambar tugas dengan semua sampel tulisan siswa
uploaded_img = preprocess_image(filepath)
similarity_scores = []

for sample in samples:
    sample_img = preprocess_image(sample.image_path)
    prediction = model.predict([uploaded_img, sample_img])[0][0]
    similarity_scores.append(prediction)

```

**Gambar 4.8** Kode Membandingkan Sample dengan Tugas

Sistem mengambil semua gambar sample tulisan tangan milik siswa dari basis data. Setiap sample dibandingkan satu per satu dengan gambar tugas menggunakan model Siamese Network. CNN dalam model akan mengekstraksi fitur dari masing-masing gambar, Dua vektor fitur hasil CNN dibandingkan menggunakan *Euclidean Distance* untuk menghasilkan skor kemiripan.

### 4. Perhitungan dan evaluasi skor kemiripan

```

# Hitung skor kemiripan rata-rata
avg_similarity = np.mean(similarity_scores)
status_str = "Cocok" if avg_similarity >= 0.75 else "Tidak Cocok"
status_emun = AssignmentStatus.COCOK if status_str == "Cocok" else AssignmentStatus.TIDAK_COCOK

```

**Gambar 4.9** Kode perhitungan dan evaluasi skor

Skor prediksi dari setiap sample disimpan dalam daftar. Nilai-nilai tersebut dirata-ratakan untuk mendapatkan nilai similarity akhir. Jika nilai rata-rata  $\geq 0.75$ , maka sistem menyimpulkan tulisan tangan Cocok; jika kurang, maka dianggap Tidak Cocok.

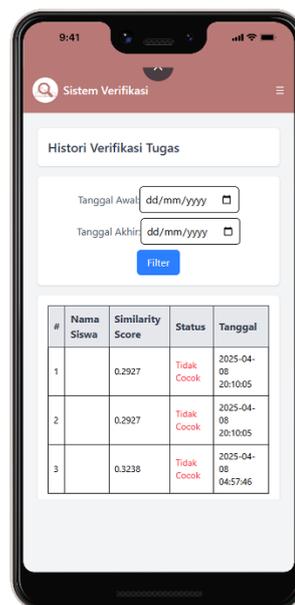
### 5. Penyimpanan hasil ke *Database*

Sistem menyimpan hasil verifikasi, termasuk similarity score dan status akhir, ke dalam tabel VerificationLog. Informasi ini

kemudian ditampilkan kepada pengguna sebagai status verifikasi dan tombol “Lihat Detail”.

#### 4.2.8 Halaman histori

Halaman Histori digunakan untuk menampilkan seluruh tugas siswa yang telah berhasil diverifikasi oleh sistem, baik yang dinyatakan “Cocok” maupun “Tidak Cocok”. Halaman ini membantu guru dalam memantau riwayat verifikasi secara keseluruhan dan memberikan akses cepat terhadap hasil verifikasi sebelumnya. Pada halaman ini, setiap entri histori berisi informasi seperti nama siswa, status verifikasi (Cocok atau Tidak Cocok), nilai similarity score.



**Gambar 4.10** Halaman Histori

Guru dapat menggunakan halaman ini untuk melakukan peninjauan ulang, pencatatan laporan, atau sekadar memantau konsistensi tugas siswa dari waktu ke waktu. Data histori yang tersimpan secara otomatis juga berfungsi sebagai arsip digital untuk keperluan dokumentasi sekolah.

## 4.3 Dataset dan Pelatihan Model

### 4.3.1 Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari kumpulan gambar tulisan tangan yang dikategorikan menjadi dua kelompok utama, yaitu tulisan tangan asli siswa dan tulisan tangan bukan siswa. Dataset ini menjadi komponen utama dalam proses pelatihan dan pengujian model verifikasi keaslian tulisan tangan berbasis CNN dan Siamese Network. Sumber data diperoleh dari:

1. Tulisan tangan asli siswa kelas VI SD Negeri 060868 Medan Timur, yang dikumpulkan dari buku catatan harian dan tugas-tugas siswa.
2. Tulisan tangan bukan siswa, yang berasal dari berbagai individu di luar lingkungan sekolah. Data ini digunakan sebagai pembanding dalam proses pelatihan model, khususnya untuk membentuk pasangan data negatif (negative pair).

Seluruh data dikumpulkan dalam format gambar (.jpg atau .png). Sebelum digunakan, gambar-gambar tersebut diproses melalui tahapan pra-pemrosesan, seperti konversi ke *grayscale* dan perubahan ukuran gambar menjadi  $128 \times 128$  piksel, untuk menyesuaikan dengan *input Layer* dari model CNN. Dataset digunakan dalam dua konteks utama:

1. Sample Tulisan Tangan: digunakan sebagai referensi dalam proses verifikasi; ditautkan ke identitas siswa.
2. Tugas yang Diverifikasi: digunakan sebagai data uji yang akan dibandingkan dengan sample menggunakan model Siamese Network.

Dalam proses pelatihan *Siamese Network*, dataset dikombinasikan dalam bentuk pasangan gambar:

1. Positive Pair: pasangan dua gambar dari siswa yang sama (sample dan tugas).
2. Negative Pair: pasangan gambar dari siswa berbeda atau dari bukan siswa.

#### 4.3.2 Pelatihan Model

Pelatihan model dalam penelitian ini telah berhasil diselesaikan dengan membangun dan mengimplementasikan dua arsitektur utama, yaitu *Convolutional Neural Network* (CNN) untuk ekstraksi fitur dan *Siamese Network* (SN) untuk mengukur tingkat kemiripan antara dua gambar tulisan tangan.

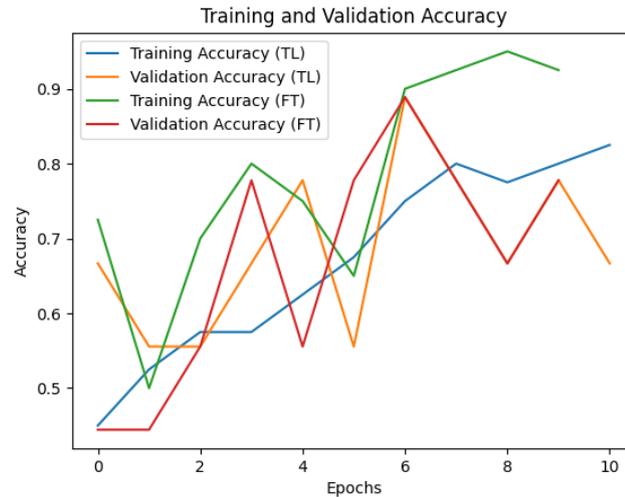
Model dilatih menggunakan dataset tulisan tangan asli siswa kelas VI SD Negeri 060868 Medan Timur serta tulisan tangan bukan siswa yang dikumpulkan dari berbagai sumber sebagai data pembanding. Seluruh gambar telah melalui proses pra-pemrosesan berupa konversi ke grayscale dan resize ke ukuran 128×128 piksel agar sesuai dengan struktur *input* jaringan.

Proses pelatihan dilakukan menggunakan framework TensorFlow dan Keras, dengan pendekatan sebagai berikut:

1. Arsitektur CNN terdiri dari dua lapisan konvolusi (Conv2D), dua lapisan *pooling* (MaxPooling2D), dan satu lapisan fully connected (Dense) yang menghasilkan vektor fitur berdimensi satu (1D feature vector) sebagai representasi tulisan tangan.

2. Vektor fitur dari dua gambar (tulisan baru dan sample siswa) digunakan sebagai *input* oleh *Siamese Network* untuk melakukan perbandingan.
3. Jarak antar dua vektor fitur dihitung menggunakan fungsi *Euclidean Distance*, yang memberikan nilai kemiripan dalam rentang 0 hingga 1.
4. Pasangan data pelatihan dibentuk dalam dua kategori:
  - a. Positive pair: dua tulisan dari siswa yang sama.
  - b. Negative pair: tulisan dari siswa berbeda atau bukan siswa.
5. Model dilatih menggunakan fungsi loss Contrastive Loss, yang secara progresif menyesuaikan bobot agar pasangan positif memiliki jarak yang lebih kecil dibandingkan pasangan negatif.
6. Pelatihan dilakukan selama 20 epoch dengan batch size 32 dan validasi 20% dari dataset digunakan untuk mengukur performa generalisasi model.

Setelah proses pelatihan, model disimpan dalam format .h5 dan digunakan dalam sistem verifikasi untuk melakukan prediksi secara real-time. Hasil evaluasi menunjukkan bahwa model mampu membedakan tulisan tangan siswa secara akurat dengan nilai similarity score yang konsisten, dan ambang batas  $\geq 0.75$  digunakan untuk menyatakan tulisan "Cocok". berikut menunjukkan kurva akurasi pelatihan dan validasi dari dua pendekatan yang digunakan, yaitu Transfer Learning (TL) dan Fine-Tuning (FT).



**Gambar 4.11** Grafik Akurasi Pelatihan dan Validasi Model

Berdasarkan grafik tersebut, dapat dilihat bahwa pendekatan Fine-Tuning (FT) menghasilkan akurasi yang lebih tinggi dan stabil dibandingkan Transfer Learning (TL), baik pada data pelatihan maupun data validasi. Akurasi pelatihan pada FT mencapai lebih dari 95% pada epoch ke-8, dan akurasi validasinya juga cukup tinggi dengan rata-rata di atas 85%.

Sementara itu, pendekatan TL menunjukkan peningkatan yang lebih lambat dan akurasi validasi yang cenderung fluktuatif. Hal ini mengindikasikan bahwa model dengan pendekatan Fine-Tuning lebih mampu menangkap pola tulisan tangan yang khas dari siswa dan membedakannya dari tulisan tangan non-siswa secara konsisten.

## 4.4 Pengujian Sistem

### 4.4.1 Pengujian Blackbox

Pengujian blackbox dilakukan untuk mengevaluasi fungsionalitas sistem tanpa melihat struktur kode program. Pengujian ini bertujuan untuk memastikan bahwa seluruh fitur yang diakses oleh pengguna bekerja sesuai dengan yang diharapkan. Pengujian dilakukan dengan memberikan *input*

tertentu pada sistem, kemudian memeriksa output-nya, tanpa mengetahui bagaimana sistem memproses *input* tersebut di dalam.

Pengujian dilakukan pada semua halaman dalam sistem, yaitu halaman login, dashboard, siswa, sample tulisan, dan upload tugas. Setiap skenario diuji dengan kondisi *input* yang valid maupun tidak valid, untuk mengamati bagaimana sistem menangani setiap kasus. Hasil pengujian dirangkum dalam bentuk tabel.

**Tabel 4.1** Pengujian Halaman Login  
**Pengujian Blackbox Halaman Login**

No	Skenario Pengujian	<i>Input</i>	Output yang Diharapkan	Status
1	Username, Password tidak diisi tetapi klik tombol login	Username, password (Kosong)	Pesan Error (Username dan password tidak boleh kosong)	Lulus
2	Mengetikkan Username atau Password yang salah	Username (kyda) Password (2801)	Pesan Error (Login gagal! Periksa username dan password.)	Lulus
3	Mengetikkan Username, password dengan benar	Username (indah) Password (password123)	Berhasil login dan diarahkan ke halaman dashboard	Lulus

**Tabel 4.2** Pengujian Halaman Dashboard  
**Pengujian Blackbox Halaman Dashboard**

No	Skenario Pengujian	<i>Input</i>	Output yang Diharapkan	Status
1	Akses ke halaman Dashboard	Pengguna berhasil login	Tampilan ucapan selamat datang dan tombol upload	Lulus

2	Klik tombol hamburger menu	Klik ikon menu navigasi	Muncul daftar menu navigasi	Lulus
---	----------------------------	-------------------------	-----------------------------	-------

**Tabel 4.3** Pengujian Halaman Siswa  
**Pengujian Blackbox Halaman Siswa**

No	Skenario Pengujian	Input	Output yang Diharapkan	Status
1	Tambah siswa baru	Nama & kelas valid	Data siswa muncul dalam tabel	Lulus
2	Tambah siswa dengan field kosong	Nama atau kelas kosong	Muncul pesan kesalahan	Lulus
3	Edit data siswa	Perubahan data valid	Data siswa diperbarui	Lulus
4	Hapus siswa	Klik tombol hapus	Data siswa terhapus dari daftar	Lulus

**Tabel 4.4** Pengujian Halaman Sample Tulisan Tangan  
**Pengujian Blackbox Halaman Sample Tulisan Tangan**

No	Skenario Pengujian	Input	Output yang Diharapkan	Status
1	Upload sample berhasil	Nama siswa & gambar valid	Gambar sample muncul di daftar siswa	Lulus
2	Upload sample tanpa gambar	Field file kosong	Muncul pesan error "gambar diperlukan"	Lulus
3	Lihat daftar sample	Klik nama siswa di daftar	Gambar sample muncul (expand)	Lulus
4	Edit sample tulisan	Ganti gambar valid	Sample diperbarui	Lulus
5	Hapus sample tulisan	Klik tombol hapus	Sample terhapus dari sistem	Lulus

**Tabel 4.5** Pengujian Halaman Upload Tugas  
**Pengujian Blackbox Halaman Upload Tugas**

No	Skenario Pengujian	Input	Output yang Diharapkan	Status
1	Upload tugas berhasil	Nama siswa dengan sample + gambar	Tugas muncul di daftar, status Pending, dan tombol Verifikasi	Lulus
2	Upload tugas tanpa sample	Nama siswa tanpa sample	Muncul pesan error “tidak ada sample”	Lulus
3	Verifikasi tugas (Cocok)	Tugas mirip sample ( $\geq 0.75$ )	Status berubah ke Cocok, skor ditampilkan	Lulus
4	Verifikasi tugas (Tidak Cocok)	Tugas berbeda sample ( $< 0.75$ )	Status berubah ke Tidak Cocok, skor ditampilkan	Lulus
5	Lihat detail tugas	Klik tombol “Lihat Detail”	Muncul similarity score	Lulus
6	Hapus tugas	Klik tombol hapus	Tugas dihapus dari daftar	Lulus

## 4.5 Evaluasi Sistem

### 4.5.1 Kelebihan Sistem

Berikut adalah kelebihan dari sistem verifikasi tugas siswa berbasis tulisan tangan yang telah dikembangkan:

1. Efisiensi Waktu: Proses verifikasi hanya membutuhkan waktu beberapa detik setelah unggah tugas.
2. Objektivitas Tinggi: Menggunakan model *Machine Learning* untuk membandingkan pola tulisan, sehingga mengurangi bias manusia.

3. Pengelolaan Data Terpusat: Riwayat verifikasi, skor kemiripan, dan data siswa tercatat dengan baik dalam sistem.
4. Kemudahan Penggunaan: Antarmuka yang sederhana memudahkan guru dalam melakukan semua proses, mulai dari *input* data siswa hingga melihat hasil verifikasi.

#### 4.5.2 Kekurangan Sistem

Meskipun sistem memberikan solusi yang inovatif, terdapat beberapa keterbatasan yang perlu diperhatikan:

1. Ketergantungan pada Kualitas Dataset: Jika sampel tulisan tidak representatif atau jumlahnya terbatas, maka akurasi model bisa menurun.
2. Tampilan Antarmuka Masih Sederhana: Beberapa pengguna merasa tampilan UI masih perlu ditingkatkan agar lebih informatif dan menarik.
3. Belum Diuji pada Skala Besar: Sistem belum diuji dalam skala sekolah yang lebih luas atau dengan ratusan data siswa secara bersamaan.
4. Kemungkinan Kesalahan Prediksi: Meskipun akurat, model masih memiliki potensi menghasilkan *false positive/negative* jika tulisan sangat mirip atau sangat berbeda dari sample.

#### 4.6 Pembahasan

Dari hasil implementasi dan pengujian, sistem dapat mengidentifikasi kemiripan tulisan tangan menggunakan CNN dan Siamese Network. CNN digunakan untuk mengekstraksi vektor fitur dari gambar tulisan tangan. Vektor fitur ini kemudian dibandingkan menggunakan *Siamese Network* untuk

menghitung jarak (similarity) antara tulisan siswa sebelumnya dengan tugas yang diunggah.

Contoh hasil vektor fitur 1D pada *Fully Connected Layer* menunjukkan representasi numerik dari pola tulisan siswa. Nilai-nilai ini menunjukkan aktivasi neuron berdasarkan fitur penting dari tulisan tangan.

Dengan pendekatan ini, sistem tidak hanya mengotomatisasi proses verifikasi tetapi juga memberikan dasar objektif untuk menentukan apakah tulisan dalam tugas benar berasal dari siswa tersebut.

Model memiliki potensi untuk dikembangkan lebih lanjut dengan melibatkan teknik augmentasi data, penyesuaian arsitektur CNN/SN, dan penerapan evaluasi metrik yang lebih kompleks seperti ROC, precision-recall, dan F1-score.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Penelitian ini berhasil mengembangkan sebuah sistem verifikasi keaslian tugas sekolah berdasarkan analisis tulisan tangan menggunakan kombinasi algoritma *Convolutional Neural Network* (CNN) dan *Siamese Network* (SN). Sistem ini ditujukan untuk membantu guru dalam memverifikasi apakah sebuah tugas benar-benar dikerjakan oleh siswa yang bersangkutan.

Sistem yang dibangun mampu melakukan proses upload sample tulisan tangan, upload tugas siswa, serta verifikasi otomatis berdasarkan tingkat kemiripan tulisan. Hasil pengujian menunjukkan bahwa sistem berfungsi dengan baik sesuai dengan skenario yang dirancang dan diterima dengan baik oleh guru sebagai pengguna akhir.

Melalui implementasi ini, sistem memberikan solusi yang efisien, objektif, dan terstruktur dalam menilai keaslian tugas siswa, serta berpotensi untuk dikembangkan lebih lanjut dalam skala yang lebih besar.

#### **5.2 Saran**

Berdasarkan penelitian yang telah dilakukan, beberapa saran dapat diberikan untuk pengembangan lebih lanjut dan peningkatan sistem verifikasi ini:

1. Perlu dilakukan pengujian dan pelatihan model lebih lanjut dengan jumlah data tulisan tangan yang lebih besar dan lebih bervariasi untuk meningkatkan akurasi model.
2. Sistem dapat dikembangkan lebih lanjut pada sisi antarmuka pengguna agar lebih interaktif dan responsif.

3. Diperlukan integrasi login multi-role, agar sistem dapat digunakan tidak hanya oleh guru, tetapi juga admin sekolah atau wali kelas.
4. Implementasi sistem pada skala yang lebih luas, misalnya lintas kelas atau sekolah lain, dapat dilakukan untuk menguji stabilitas dan efektivitas sistem.
5. Perlu dilakukan evaluasi jangka panjang untuk mengetahui dampak penerapan sistem ini terhadap nilai-nilai kejujuran dan tanggung jawab siswa dalam mengerjakan tugas sekolah.

## DAFTAR PUSTAKA

- Abdulloh, R. (2018). 7 in 1 pemrograman web untuk pemula (Edisi Pertama, hlm. 103). Elex Media Komputindo.
- Agustini. (2016). MySQL multiuser *Database* (pp. 1–23).
- Bachtiar, D. H., Paniran, P., & Suksmadana, I. M. B. (2024, Juni). Perancangan back-end API pada aplikasi mobile Fruityfit menggunakan framework Express JS. *Mars Jurnal Teknik Mesin, Industri, Elektro dan Ilmu Komputer*, 2(3), 107–117.
- Firdaus, M. R. A., & Ratnasari, C. I. (2022, Agustus). Pengembangan Telegram bot sebagai solusi pengolahan data kolam pada budi daya udang untuk Jala Tech. *JIKO (Jurnal Informatika dan Komputer)*, 5(2), 99–107.
- Ghimire, D. (2010, May 5). Comparative study on Python web frameworks: Flask and Django (Tesis, 37 halaman).
- Grinberg, M. (2018, Maret). *Flask web development* (2nd ed., 316 hlm.). Sebastopol, CA: O'Reilly Media, Inc.
- Gu, J., Wang, Z., & Kuen, J. (2017, October 19). Recent advances in convolutional neural networks. Nanyang Technological University.
- Gunawan, D., & Setiawan, H. (2022, Desember). *Convolutional Neural Network* dalam analisis citra medis. *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, 2(2), xx–xx.
- Hanief, S., & Jepriana, I. W. (2020). Konsep algoritme dan aplikasinya dalam bahasa pemrograman C++ (hlm. 14). Yogyakarta: ANDI.
- Kadir, A. (2020). *Logika pemrograman Java* (hlm. 22). Jakarta: PT Alex Media Komputindo.
- Lutz, M. (2010). *Programming Python* (4th ed.). Sebastopol, CA: O'Reilly Media, Inc.
- Pradigi, C. H., Harlina, T., & Solehatin. (2022, Juni). Implementasi Express JS untuk membangun REST API Website STIKOM PGRI Banyuwangi. *JIKOM (Jurnal Informatika dan Komputer)*, 9(2), 118–122.
- Prasetyo, D., Wibowo, R., & Hidayat, S. (2023). Handwriting-based student identity verification using deep learning. In *Proceedings of the IEEE International Conference on Artificial Intelligence and Education (ICAIE)* (pp. 210–215). Jakarta, Indonesia.

- Romzi, M., & Kurniawan, B. (2020, Desember). Pembelajaran pemrograman Python dengan pendekatan logika algoritma. Vol. 3(2), xx–xx.
- Rosa, A. S., & Shalauddin, M. (2016). Rekayasa perangkat lunak terstruktur dan berorientasi objek (hlm. 44).
- Rusli, M. S., Ahmar, A. S., & Rahman, A. (2019). Pemrograman website dengan PHP-MySQL untuk pemula. Yayasan Ahmar Cendekia Indonesia.
- Sianipar, R. H. (2017, Desember). Dasar pemrograman JavaScript: Langkah demi langkah. Yogyakarta, Indonesia: ANDI.
- Suharto, A. (2023, Februari). Fundamental bahasa pemrograman Python. Yogyakarta, Indonesia: Eureka Media Aksara.
- Sulistiati, T., Anika, F., Murti, W., & Aristyanti, R. (2019). Membangun website CV. Marisa Kavling Baturaja menggunakan PHP dan MySQL. JTIM: Jurnal Teknik Informatika Mahakarya, 2(1), 47–53.
- Smith, Y., Johnson, A., & Lee, B. (2021). Handwritten character recognition using convolutional neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5), 1234–1245.
- Wu, W. (2018, Maret). React Native vs Flutter, cross-platform mobile application frameworks (28 hlm.).
- Zhang, X., Wu, L., & Li, T. (2022). Signature verification using Siamese neural networks for secure authentication. *IEEE Transactions on Information Forensics and Security*, 17(8), 6543–6555.
- Zhang, Z., & Peng, H. (2019). Deeper and wider Siamese networks for real-time visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 4591–4600).