

**PENERAPAN METODE *SUPPORT VECTOR MACHINE* (SVM)
BERBASIS *FACE TRACKING* UNTUK DETEKSI
KECURANGAN UJIAN ONLINE SECARA *REAL-TIME***

LAPORAN TUGAS AKHIR

Laporan ini disusun guna memenuhi salah satu syarat untuk menyelesaikan
program studi S1 Teknik Informatika pada Fakultas Teknologi Industri Universitas
Islam Sultan Agung Semarang



DISUSUN OLEH :

IRFAN YANUARSYAH

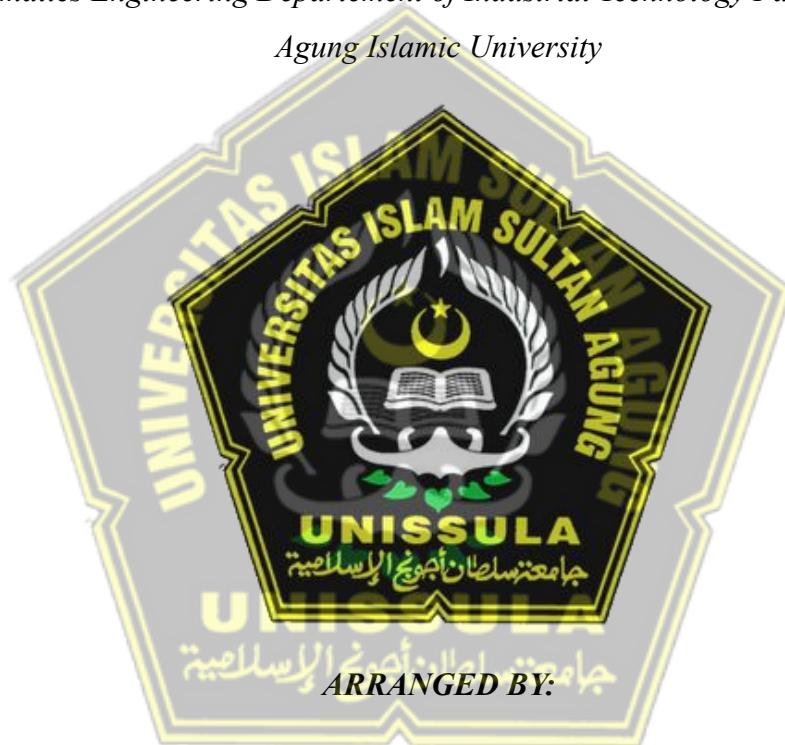
32602100060

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG
2025**

***IMPLEMENTATION OF FACE TRACKING BASED SUPPORT
VECTOR MACHINE (SVM) METHOD FOR REAL-TIME
ONLINE EXAM CHEATING DETECTION***

FINAL PROJECT

*Proposed to complete the requirement to obtain a bachelor's degree (SI) at
Informatics Engineering Departement of Industrial Technology Faculty Sultan
Agung Islamic University*



IRFAN YANUARYAH

32602100060

***MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG
2025***

LEMBAR PENGESAHAN TUGAS AKHIR

LEMBAR PENGESAHAN TUGAS AKHIR

PENERAPAN METODE SUPPORT VECTOR MACHINE (SVM)
BERBASIS FACE TRACKING UNTUK DETEKSI KECURANGAN PADA
UJIAN ONLINE SECARA REAL-TIME

IRFAN YANUARYAH
NIM 32602100060

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 02 September 2025

TIM PENGUJI UJIAN SARJANA :

Moch. Taufik, ST., MIT
NIK. 210604034
(Penguji 1)

Andi Riansyah, ST., M.Kom
NIK. 210616053
(Penguji 2)

Ir. Sri Mulvono, M.Eng
NIK. 210616049
(Pembimbing)

..... 30 September 2025

..... 30 September 2025

..... 30 September 2025

..... 30 September 2025
Semarang

Mengetahui,
Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, ST., MIT
NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Irfan Yanuarsyah

NIM : 32602100060

Judul Tugas Akhir : PENERAPAN METODE SUPPORT VECTOR MACHINE (SVM) BERBASIS FACE TRACKING UNTUK DETEKSI KECURANGAN PADA UJIAN ONLINE SECARA REAL-TIME.

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 20 November 2025

Yang Menyatakan,


SUPPLIER EUPAH
11058ANX174358344
METERAI TEMPEL
Irfan Yanuarsyah

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Irfan Yanuarsyah
NIM : 32602100060
Program Studi : Teknik Informatika
Fakultas : Teknologi Industri
Alamat Asal : Cilacap Jawa Tengah

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Penerapan Metode *Support Vector Machine (SVM)* Berbasis *Face Tracking* untuk Deteksi Kecurangan pada Ujian Online Secara *Real-Time*.

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 20 November 2025

Yang menyatakan,


Irfan Yanuarsyah

KATA PENGANTAR

Puji syukur penulis panjatkan kepada ALLAH SWT, yang telah memberikan rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Penerapan Metode *Support Vector Machine (SVM)* Berbasis *Face Tracking* untuk Deteksi Kecurangan pada Ujian Online Secara *Real-Time*.” ini dengan baik. Dengan penuh rasa hormat, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Rektor UNISSULA Bapak Prof. Dr. H. Gunarto, SH., M.Hum yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T
3. Dosen pembimbing penulis Bapak Ir. Sri Mulyono, M.Eng yang telah memberikan arahan, bimbingan, dan saran yang berarti dalam penyelesaian tugas akhir ini.
4. Seluruh dosen Program Studi Teknik Informatika, Fakultas Teknologi Industri UNISSULA yang telah memberikan ilmunya kepada penulis.
5. Orang tua penulis, Bapak Amir Mahmud dan Ibu Ratna Sulandari serta ketiga adik penulis yang selalu memberikan segala doa, dukungan, dan motivasi dengan penuh kasih sayang sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik.
6. Teman-teman seperjuangan atas kebersamaanya yang telah bekerja keras serta semangat dalam proses penyelesaian tugas akhir ini.

Semarang, 20 November 2025

Penulis

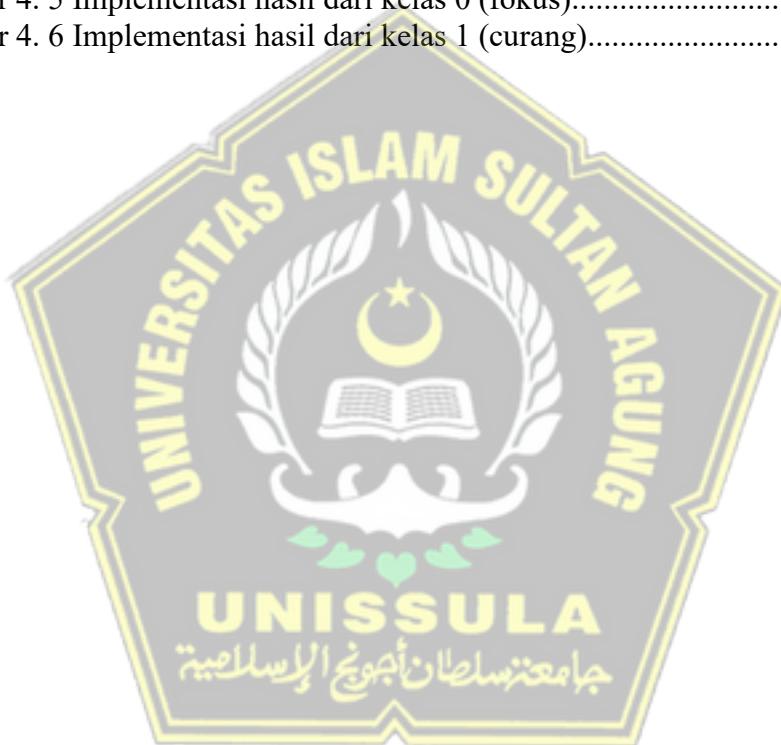
DAFTAR ISI

PENERAPAN METODE SUPPORT VECTOR MACHINE (SVM) BERBASIS FACE TRACKING UNTUK DETEKSI KECURANGAN UJIAN ONLINE SECARA REAL-TIME	i
IMPLEMENTATION OF FACE TRACKING BASED SUPPORT VECTOR MACHINE (SVM) METHOD FOR REAL-TIME ONLINE EXAM CHEATING DETECTION.....	ii
LEMBAR PENGESAHAN TUGAS AKHIR	iii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR	iv
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	x
ABSTRAK	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	6
2.1 Tinjauan Pustaka.....	6
2.2 Dasar Teori	10
2.2.1 Ujian Online.....	10
2.2.2 Kecurangan Dalam Ujian.....	11
2.2.3 Face Tracking	11
2.2.4 Ekstrasi Fitur Visual.....	12
2.2.5 Support Vector Machine.....	13
2.2.6 Real-Time Processing	15
BAB III METODE PENELITIAN	16
3.1 Metode Penelitian.....	16
3.1.1 Studi Literatur	16
3.1.2 Pengumpulan Dataset.....	16
3.1.3 Ekstrasi Fitur Visual.....	17
3.1.4 Preprocessing Data	17
3.1.5 Pelatihan Pengujian Model.....	18

3.1.6	Evaluasi Model.....	19
3.1.7	Implementasi Sistem <i>Realtime</i>	21
3.1.8	<i>Deployment</i> Model.....	22
3.2	Software Yang Digunakan.....	23
3.3	Perancangan Sistem.....	25
3.4	Perancangan <i>User Interface</i>	27
3.4.1	Halaman Awal Sistem.....	27
3.4.2	Tampilan Hasil Sistem	28
BABI IV HASIL DAN ANALISIS PENELITIAN.....		30
4.1	Hasil Pengumpulan Dataset.....	30
4.2	Hasil Ekstrasi Fitur Visual.....	32
4.3	Hasil <i>Preprocessing</i> Data.....	34
4.4	Hasil Pelatihan Model SVM.....	35
4.5	Evaluasi Sistem	37
4.5.1	<i>Confusion Matrix</i>	37
4.5.2	<i>Classification Report</i>	38
4.5.3	Evaluasi <i>Face Tracking</i>	40
4.6	Hasil Implementasi Sistem	40
4.6.1	Implementasi Kelas 0 (Fokus).....	42
4.6.2	Implementasi Kelas 1 (Curang).....	44
BAB V KESIMPULAN DAN SARAN		45
5.1	Kesimpulan.....	45
5.2	Saran	46
DAFTAR PUSTAKA.....		47
LAMPIRAN.....		50

DAFTAR GAMBAR

Gambar 3. 1 Alur Penelitian.....	16
Gambar 3. 2 Proses implementasi sistem real-time	21
Gambar 3. 3 flowcart alur sistem streamlit	25
Gambar 3. 4 Tampilan halaman awal sistem	27
Gambar 3. 5 Tampilan hasil sistem mendeteksi perilaku fokus.....	28
Gambar 3. 6 Tampilan hasil sistem mendeteksi perilaku curang.....	29
Gambar 4. 1 Dataset Fokus (tidak curang)	30
Gambar 4. 2 Dataset Curang	30
Gambar 4. 3 Confusion Matrix	38
Gambar 4. 4 Tampilan antarmuka layar login sistem	41
Gambar 4. 5 Implementasi hasil dari kelas 0 (fokus).....	43
Gambar 4. 6 Implementasi hasil dari kelas 1 (curang).....	44



DAFTAR TABEL

Tabel 4. 1 Komposisi Dataset Final	31
Tabel 4. 2 Pembagian dataset.....	34
Tabel 4. 3 Classification Report.....	39



ABSTRAK

Tingginya pelaksanaan ujian online meningkatkan kemungkinan terjadinya kecurangan akademis, sehingga menimbulkan kebutuhan akan sistem pengawasan otomatis yang tepat untuk menjaga integritas. Studi ini bertujuan untuk menerapkan sistem deteksi penipuan secara langsung dengan menggunakan metode *machine learning Support Vector Machine (SVM)* yang berfokus pada *Face Tracking*. Data yang digunakan merupakan dataset gambar wajah yang dikategorikan ke dalam dua kelas perilaku: ‘Fokus’ dan ‘Curang’. Metode penelitian mencakup pengambilan fitur visual menggunakan *MediaPipe* untuk mengukur rotasi kepala dan pergerakan mulut, serta pemisahan data dengan proporsi 80% data pelatihan dan 20% data pengujian. Kinerja model dinilai dengan menggunakan metrik Akurasi, Presisi, dan *Recall*. Hasil uji menunjukkan bahwa model yang dibuat berhasil memperoleh akurasi 80%, dengan presisi 100% dan recall 60% untuk kategori ‘Curang’. Hasil ini menunjukkan bahwa model SVM sangat efisien dalam mencegah tuduhan yang tidak benar (presisi 100%), yang merupakan persyaratan penting. Hasil dari penelitian ini menunjukkan bahwa metode SVM yang menggunakan *face tracking* dapat dipercaya untuk menciptakan prototipe fungsional bagi sistem pengawasan ujian daring.

Kata Kunci: Deteksi Kecurangan, Ujian Online, Support Vector Machine (SVM), Face Tracking, MediaPipe.

ABSTRACT

The high number of online exams increases the likelihood of academic cheating, thus necessitating a proper automated proctoring system to maintain integrity. This study aims to implement a live fraud detection system using the Support Vector Machine (SVM) machine learning method focusing on face tracking. The data used is a dataset of facial images categorized into two behavioral classes: 'Focused' and 'Cheating'. The research method includes visual feature extraction using MediaPipe to measure head rotation and mouth movement, and data separation with a proportion of 80% training data and 20% testing data. Model performance was assessed using Accuracy, Precision, and Recall metrics. The test results showed that the model successfully achieved 80% accuracy, with 100% precision and 60% recall for the 'Cheating' category. These results indicate that the SVM model is highly efficient in preventing false accusations (100% precision), which is an important requirement. The results of this study indicate that the SVM method using face tracking is reliable for creating a functional prototype for an online exam proctoring system.

Keywords: Fraud Detection, Online Exams, Support Vector Machine (SVM), Face Tracking, MediaPipe.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan pesatnya perkembangan teknologi pada masa kini, manusia menjadi semakin bergantung pada keberadaan teknologi tersebut. Hal ini disebabkan karena teknologi mampu memberikan kemudahan serta meringankan berbagai aktivitas, baik di bidang industri, perkantoran, kesehatan, maupun pendidikan (Jurnal dkk., 2021). Meskipun sejumlah perguruan tinggi maupun sekolah telah menerapkan ujian berbasis komputer (Wong dkk., 2024), kenyataannya baik ujian tulis maupun ujian digital masih menunjukkan tingkat kecurangan yang cukup tinggi. Kondisi ini dapat dipengaruhi oleh berbagai faktor, antara lain rendahnya rasa efikasi diri serta adanya ketakutan siswa dalam menghadapi kemungkinan kegagalan saat mengerjakan ujian. Faktor-faktor tersebut termasuk dalam kategori internal yang bersumber dari kepribadian setiap individu. Namun demikian, terdapat pula faktor eksternal yang turut berperan, salah satunya adalah rendahnya konsistensi pengawas atau proktor dalam melakukan pemantauan selama jalannya ujian (Bimantoro dkk., 2024).

Di sisi lain, rendahnya tingkat keamanan dalam pelaksanaan ujian dapat membuka peluang bagi peserta untuk melakukan tindakan kecurangan, yang pada akhirnya mengancam keutuhan serta validitas hasil evaluasi (Pratama dkk., 2021). Seiring dengan berkembangnya ujian berbasis daring, muncul pula tantangan baru yang berkaitan dengan integritas ujian dan potensi kecurangan akademik (Syarif dkk., 2023). Berbagai bentuk pelanggaran, seperti menyontek maupun memanfaatkan sumber eksternal, berpotensi merusak objektivitas dan integritas proses penilaian yang seharusnya dijalankan secara jujur (Putri Iskandar dkk., 2024a). Oleh karena itu, solusi inovatif diperlukan untuk meningkatkan keamanan ujian online dan memastikan keaslian identitas peserta ujian. Kecurangan ini tentu saja merusak keadilan dalam sistem pendidikan dan meragukan validasi hasil penilaian (Sari & Arvita, 2022).

Kecurangan dalam pelaksanaan ujian online sangat sulit teridentifikasi karena keterbatasan pengawasan langsung dari pengawas maupun dosen. Pengawasan manual memiliki keterbatasan, terutama karena kemampuan manusia tidak memungkinkan untuk memantau seluruh peserta ujian secara bersamaan dengan optimal. Untuk itu, diperlukan sebuah sistem otomatis berbasis teknologi yang mampu mendeteksi perilaku mencurigakan secara *real-time* dengan tingkat akurasi yang tinggi. Salah satu pendekatan yang saat ini banyak dikembangkan adalah pemanfaatan teknologi *face tracking* (pelacakan wajah) sebagai solusi yang potensial. Melalui integrasi kamera komputer, sistem ini dapat secara otomatis mengidentifikasi kehadiran wajah peserta, mendeteksi pergerakan kepala, serta mengenali adanya objek atau individu lain dalam bingkai. Teknologi ini dianggap efektif karena mampu menangkap indikator visual yang sering kali menjadi tanda terjadinya praktik kecurangan selama ujian berlangsung.

Agar sistem pengawasan otomatis tidak sekadar mengandalkan deteksi visual, diperlukan adanya algoritma klasifikasi yang dapat menentukan apakah suatu pola gerakan termasuk dalam kategori kecurangan atau bukan. Salah satu metode yang banyak digunakan untuk klasifikasi biner adalah *Support Vector Machine* (SVM). Metode ini bekerja dengan memisahkan data ke dalam dua kelas berbeda melalui pencarian margin yang optimal, sehingga sangat sesuai untuk diterapkan pada data dengan dimensi yang tinggi. Dalam studi Soni et al. (2024), *Support Vector Machine* berhasil mencapai tingkat akurasi di atas 90% dalam mendeteksi tindakan kecurangan berbasis fitur visual dari video peserta ujian.

1.2 Perumusan Masalah

Berdasarkan latar belakang permasalahan, maka dapat diidentifikasi permasalahan yang akan dibahas sebagai berikut :

1. Bagaimana cara mendeteksi kecurangan dalam ujian online menggunakan teknologi pelacakan wajah?
2. Bagaimana algoritma SVM dapat untuk mengklasifikasikan perilaku peserta sebagai tindakan curang atau tidak curang?
3. Seberapa akurat sistem yang dikembangkan dalam mendeteksi kecurangan ujian online?

1.3 Batasan Masalah

Adapun batasan masalah dari penulisan proposal ini yaitu sebagai berikut :

1. Penelitian ini menggunakan input dari kamera web tunggal secara *real-time* dan diasumsikan beroperasi dalam kondisi pencahayaan dalam ruangan yang memadai.
2. Fokus deteksi hanya pada arah keberadaan wajah di kamera.
3. Sistem hanya mendeteksi indikasi kecurangan, tidak memberikan hukuman atau sanksi.



1.4 Tujuan Penelitian

Adapun tujuan tugas akhir dari penulisan proposal ini yaitu :

1. Membuat sistem *face tracking* yang dapat memantau serta menganalisis gerakan dan posisi wajah peserta selama pelaksanaan ujian daring.
2. Melakukan pengujian dan evaluasi terhadap kinerja sistem deteksi kecurangan dengan menggunakan metrik akurasi, presisi, *recall*, dan *F1-score*, guna memastikan efektivitas sistem dalam mendeteksi kecurangan secara otomatis.
3. Memberikan kontribusi terhadap sistem pengawasan ujian *online* yang lebih cerdas dan efisien dengan mengintegrasikan teknologi *machine learning* dan *computer vision*.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi sebagai alat bantu yang efektif dalam menjaga serta meningkatkan integritas dan kredibilitas pelaksanaan ujian yang dilakukan secara daring. Sistem ini memiliki potensi untuk meningkatkan objektivitas pengawasan, mengurangi risiko kecurangan, serta menjadi alternatif pengawasan yang lebih hemat biaya dibandingkan layanan proctoring komersial yang tersedia di pasaran.

1.6 Sistematika Penulisan

Struktur penulisan yang akan diterapkan dalam pembuatan laporan tugas akhir adalah sebagai berikut :

BAB I : PENDAHULUAN

Pada BAB I menjelaskan tentang latar belakang, pemilihan judul, rumusan masalah, Batasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penelitian

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Pada BAB II memuat tentang penelitian terdahulu dan landasan teori yang berkaitan untuk membantu memahami konsep

algoritma SVM dan *Face Tracking* untuk melengkapi penelitian ini.

BAB III : METODE PENELITIAN

Pada BAB III menjelaskan proses penelitian yang dimulai dari pengumpulan data hingga evaluasi hasil prediksi.

BAB IV : HASIL DAN ANALISIS PENELITIAN

Pada BAB IV berisi tentang pemaparan hasil penelitian yang dimulai dari pembuatan sistem sampai dengan proses implementasi sistem real-time.

BAB V : KESIMPULAN DAN SARAN

Pada BAB V merangkum keseluruhan proses penelitian dari awal sampai akhir.



BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinajauan Pustaka

Pada penelitian yang dilakukan oleh (Septiawan dkk., 2022.) Dalam penelitian ini, penerapan *Support Vector Machine* (SVM) yang dioptimalkan menggunakan *Adaboost*, serta *Decision Trees* yang juga dioptimalkan dengan *Adaboost*, dilakukan untuk membandingkan nilai akurasi yang dihasilkan. Pengujian difokuskan pada metrik akurasi untuk mengevaluasi tingkat ketepatan prediksi dari hasil pengujian. Berdasarkan hasil akurasi, dapat ditentukan algoritma mana yang lebih efektif untuk deteksi sentimen ujaran kebencian. Hasil ini menunjukkan bahwa SVM dan *Decision Tree*, setelah dioptimalkan dengan *Adaboost*, memiliki performa lebih baik dan lebih sesuai digunakan pada dataset yang tersedia.

Pada penelitian yang dilakukan oleh (Yahya dkk., 2024) Sistem berhasil mengklasifikasikan kalimat yang mengandung unsur kebencian maupun yang tidak mengandungnya. Dari pengujian terhadap 100 dokumen tweet, diperoleh nilai *recall* 76%, *precision* 96%, dan akurasi 81%. Kinerja klasifikasi dan tingkat akurasi dipengaruhi oleh jumlah data pelatihan serta ketelitian dalam proses text *preprocessing*. Hasil penelitian ini dapat dijadikan sebagai bahan pembelajaran bagi masyarakat, khususnya pengguna media sosial, dengan saran untuk menambah jumlah data pelatihan serta memperbaiki proses *preprocessing* teks.

Pada penelitian yang dilakukan oleh (Alamsyah dkk., 2022) Penelitian ini menerapkan metode SVM dan Naïve Bayes pada platform Tokopedia. Dataset terlebih dahulu melalui tahap *preprocessing* sebelum dibagi menjadi dua bagian, yaitu 80% untuk data pelatihan sebanyak 621 sampel dan 20% untuk data pengujian sebanyak 266 sampel. Selanjutnya, prediksi dilakukan menggunakan model SVM dan Naïve Bayes yang telah dilatih terhadap dataset baru tanpa label, yang berjumlah 29 sampel. Akhirnya, kedua model diuji pada fase evaluasi dengan menggunakan *confusion matrix*. Pada tahap pengujian menggunakan SVM yang telah dilatih untuk memprediksi dataset baru, diperoleh 8 data berlabel Original

(OR) dan 21 data berlabel *Computer Generated* (CG). Sementara itu, metode Naïve Bayes menghasilkan prediksi 7 data OR dan 22 data CG. Evaluasi kinerja menunjukkan bahwa SVM memiliki akurasi sebesar 94,38%, dengan nilai presisi, *recall*, dan *F1-score* masing-masing 95,57%. Metode Naïve Bayes menghasilkan akurasi sebesar 91,57%, dengan nilai presisi 89,51%, *recall* 98,23%, dan *F1-score* 93,67%. Berdasarkan perbandingan akurasi antara kedua metode, dapat disimpulkan bahwa SVM memiliki performa yang lebih unggul dibandingkan Naïve Bayes.

Pada penelitian yang dilakukan oleh (Vebriani dan Yustanti, 2024) Model klasifikasi berbasis *Support Vector Machine* (SVM) menunjukkan kinerja yang konsisten, dengan peningkatan akurasi pada data pelatihan dan pengujian seiring bertambahnya jumlah fold dalam *cross-validation*. Pada *fold* 3, akurasi pelatihan mencapai 88% dan pengujian 84%; pada *fold* 5 meningkat menjadi 89% dan 85%; sementara pada *fold* 10, akurasi pelatihan 90% dan pengujian 88%, menandakan kemampuan generalisasi model yang baik.. Pelaksanaan deteksi *phishing* dengan *Flask* memperlihatkan pendekatan yang mutakhir dan efektif untuk keamanan di dunia maya, di mana pengujian *black box* menjamin ketepatan deteksi serangan *phishing*, sehingga melindungi pengguna dari risiko siber.

Pada penelitian yang dilakukan oleh (Karinasari dkk., 2023) *Intra Uterine Growth Restriction* (IUGR) adalah kondisi di mana pertumbuhan janin terhambat sebelum kelahiran, sehingga bayi lahir dengan ukuran lebih kecil dari normal dan tidak sesuai dengan pola pertumbuhan janin yang seharusnya. Pemeriksaan menggunakan USG akan menunjukkan ukuran janin yang lebih kecil beserta berat badan yang rendah. Tindakan cepat diperlukan agar bayi dapat lahir secara normal. Melalui percobaan menggunakan metode SVM *Multiclass* dengan empat jenis kernel, diperoleh nilai akurasi sebesar 96%. Diharapkan hasil percobaan ini dapat membantu rumah sakit dalam menangani ibu hamil yang janinnya mengalami kondisi IUGR.

Pada penelitian yang dilakukan oleh (Frenica dan Soim, 2023.) Dalam penelitian ini, berbagai jenis kernel SVM seperti linear, *polynomial*, RBF, dan sigmoid diuji untuk menentukan kernel yang paling tepat dalam mendeteksi kondisi

banjir. Hasil evaluasi menunjukkan bahwa kernel *polynomial* memberikan performa terbaik dibandingkan kernel lainnya, dengan akurasi pelatihan dan pengujian masing-masing mencapai 1,0. Hasil komparasi waktu dan skalabilitas SVM untuk berbagai kernel menunjukkan bahwa kernel *polynomial* memiliki durasi pelatihan dan pengujian paling singkat, masing-masing 0,0012 detik dan 0,0002 detik. Berdasarkan hasil confusion matrix untuk data pelatihan dan pengujian, kernel ini memberikan nilai konstan sebesar 1,0. Nilai recall (sensitivitas), presisi, dan *F1-score* yang dicapai sebesar 1,0 menunjukkan bahwa model mampu membedakan kelas-kelas dalam data secara konsisten dan akurat.

Penelitian yang dilakukan oleh (Andi Saenong dan Rahmat, 2024) Studi ini menunjukkan adanya potensi yang signifikan untuk penerapan dalam sistem parkir cerdas. Dengan penggunaan SVM dan parameter yang telah dioptimalkan, sistem dapat memantau kendaraan secara *real-time*. Meskipun hasil penelitian ini cukup memuaskan, studi lanjutan dapat diarahkan untuk mengeksplorasi metode ekstraksi fitur yang lebih kompleks atau penerapan teknik pembelajaran mesin lainnya guna membandingkan performa masing-masing. Selain itu, evaluasi model pada berbagai kondisi pencahayaan dan sudut pandang kamera yang berbeda dapat memberikan wawasan tambahan untuk meningkatkan ketahanan dan keandalan model. Penelitian ini menunjukkan bahwa model SVM yang telah dioptimalkan mampu memberikan solusi yang efektif dan efisien dalam mendeteksi kendaraan di area parkir, sehingga mendukung perkembangan teknologi parkir cerdas.

Penelitian yang dilakukan oleh (Hendrastuty dkk., 2021a) Penelitian ini melaporkan hasil klasifikasi menggunakan metode *Support Vector Machine* (SVM) yang terbagi ke dalam tiga kategori, yakni kelas netral sebesar 98,34%, kelas negatif sebesar 0,99%, dan kelas positif sebesar 0,66%. Hasil evaluasi menunjukkan bahwa nilai akurasi kernel linear mencapai 98,67%, *precision* 98%, *recall* 99%, dan *F1-Score* 98%.

Penelitian yang dilakukan oleh (Niswa dan Widasari, 2017) Penelitian mengenai deteksi gestur tangan menggunakan metode *Support Vector Machine* (SVM) pada lengan prostetik bionik menghasilkan beberapa temuan dan kesimpulan yang signifikan. Sensor yang terdapat pada perangkat *OYMotion*

terbukti efektif dalam membaca dan merekam sinyal *Electromyography* (EMG) dengan baik. Hal ini terlihat dari variasi nilai voltase yang dihasilkan oleh setiap gerakan yang berada dalam rentang sinyal EMG sesuai dengan tegangan kerja antara 0–3,3V. Selanjutnya, penerapan *Support Vector Machine* (SVM) dengan kernel *Radial Basis Function* (RBF), yang dipadukan dengan filter Gaussian dan ekstraksi fitur Waveform Length, berhasil diimplementasikan dalam sistem ini. Uji coba yang dilakukan pada lima subjek menunjukkan tingkat akurasi mencapai 82%, yang membuktikan bahwa sistem ini dapat memenuhi faktor kehandalan dalam proses klasifikasinya. Rata-rata waktu komputasi yang diperoleh dari pengujian adalah 101 mili detik, mengindikasikan bahwa sistem ini memiliki kinerja yang efisien. Dengan hasil ini, sistem dapat dianggap layak diterapkan pada lengan prostetik bionik, baik dari segi keandalan deteksi gerakan maupun efisiensi waktu pemrosesan.

Penelitian yang dilakukan oleh (Japit dkk., 2024) Metode *Support Vector Machine* (SVM) terbukti sangat efektif dalam mendeteksi anomali pada transaksi e-commerce. Dengan tingkat akurasi 94,5% dan nilai AUC 96,3%, model ini mampu membedakan transaksi normal dan transaksi yang mencurigakan secara efisien. Hal ini menegaskan bahwa SVM merupakan algoritma yang andal untuk menangani dataset transaksi yang kompleks dan beragam. Model ini menunjukkan tingkat presisi sebesar 91,2%, yang mengindikasikan bahwa sebagian besar transaksi yang diklasifikasikan sebagai anomali memang benar-benar termasuk anomali. Sementara itu, *recall* sebesar 89,8% menandakan bahwa model memiliki kinerja yang baik dalam mendeteksi transaksi anomali secara keseluruhan, meskipun masih terdapat beberapa kasus transaksi anomali yang tidak teridentifikasi (*false negative*). Nilai *F1-score* sebesar 90,5% menunjukkan keseimbangan yang baik antara presisi dan *recall*, menegaskan bahwa model mampu mendeteksi anomali secara konsisten dan andal. Proses pra-pemrosesan dan pengambilan fitur memiliki peran krusial dalam memperbaiki kinerja model. Tahapan seperti normalisasi, transformasi data, dan seleksi fitur terbukti mampu mengurangi noise pada data serta memastikan model dapat bekerja secara optimal. Implementasi sistem deteksi anomali berbasis SVM pada platform e-commerce

berperan penting dalam meningkatkan keamanan transaksi serta mencegah kerugian akibat tindakan penipuan. Sistem ini dapat merespons aktivitas mencurigakan dengan cepat dan efisien, sehingga mampu meningkatkan tingkat kepercayaan pengguna terhadap layanan.

2.2 Dasar Teori

2.2.1 Ujian Online

Ujian online merupakan metode penilaian hasil belajar yang dilaksanakan secara daring dengan memanfaatkan teknologi digital sebagai media utama. Selama pelaksanaan, peserta ujian menyelesaikan soal menggunakan perangkat seperti laptop atau komputer yang terhubung ke internet. Walaupun efektif dan serbaguna, sistem ini dapat terpapar pada tindakan penipuan akibat kurangnya pengawasan secara langsung. Ujian daring kini menjadi pilihan yang banyak diminati di perguruan tinggi karena memberikan efisiensi dan fleksibilitas yang tinggi. Ujian dapat disajikan dalam berbagai bentuk soal, seperti pilihan ganda, isian singkat, maupun esai, yang dapat diakses secara daring melalui internet maupun intranet. Hal ini memungkinkan institusi pendidikan menyelenggarakan ujian tanpa harus mengumpulkan peserta pada lokasi fisik tertentu. (Wicaksono dan Yamasari, 2025). Oleh karena itu, diperlukan sistem otomatis untuk mengidentifikasi perilaku mencurigakan selama ujian berlangsung.

Praktik mencontek dapat ditemukan pada berbagai jenjang pendidikan, mulai dari sekolah dasar hingga sekolah menengah atas. Perilaku ini bisa dilakukan secara individu maupun berkelompok, dan tidak terbatas oleh faktor usia. Pada pelaksanaan ujian online, praktik menyontek cenderung lebih sering terjadi karena keterbatasan pengawasan manusia yang tidak optimal. Selain itu, tingkat kecurangan juga dipengaruhi oleh faktor lain, seperti jenis platform ujian online yang digunakan serta minimnya sistem pendukung untuk pengawasan secara virtual (Pangestu dkk., 2024). Melalui perkembangan teknologi, penerapan sistem proctoring dengan teknologi pengenalan wajah adalah langkah maju untuk memastikan integritas ujian. Akan tetapi, transisi ke ujian daring juga menghadirkan tantangan tersendiri, sebab penipuan, identitas ganda, dan minimnya pengawasan dapat menurunkan kepercayaan terhadap hasil ujian. Oleh karena itu,

pengembangan sistem pemantauan yang andal dan efisien menjadi hal yang sangat penting dalam menjaga integritas serta standar akademik. Penelitian ini bertujuan untuk merancang dan membangun sistem pengawasan ujian online dengan memanfaatkan teknologi pengenalan wajah (Putri Iskandar dkk., 2024b).

2.2.2 Kecurangan Dalam Ujian

Kecurangan saat ujian adalah tindakan yang tidak diizinkan demi meraih pencapaian akademik yang tinggi. Kecurangan saat ujian terjadi karena berbagai faktor. Perilaku tersebut dipengaruhi oleh dua kelompok faktor, yaitu sekitar 10% berasal dari aspek internal seperti kontrol diri, efikasi diri akademik, dan prestasi akademik, sedangkan 90% lainnya ditentukan oleh faktor eksternal berupa tingkat hukuman yang diberikan, pengaruh teman sendiri, persepsi terhadap materi maupun pengajar, serta faktor lain di luar diri individu. Faktor eksternal dapat dikatakan memiliki pengaruh besar dalam terjadinya kecurangan akademik (Mushthofa dkk., 2021). Kecurangan dalam ujian online mencakup berbagai tindakan tidak sah yang dilakukan peserta demi memperoleh hasil yang lebih baik. Beberapa bentuk kecurangan yang sering terjadi antara lain :

- Melihat ke luar layar untuk membaca catatan atau berkomunikasi.
- Menggunakan perangkat tambahan seperti ponsel.
- Menghindari pandangan kamera (wajah tidak terdeteksi).
- Adanya orang lain yang membantu menjawab soal.

Deteksi kecurangan secara otomatis dapat dilakukan dengan menganalisis pola visual dan perilaku wajah peserta selama ujian.

2.2.3 Face Tracking

Pelacakan wajah merupakan salah satu metode dalam bidang *computer vision* yang berfungsi untuk mendeteksi sekaligus mengikuti posisi wajah manusia pada citra maupun aliran video. Teknik pelacakan wajah atau *face tracking* adalah sebuah metode dari pelacakan berbasis tanpa *marker* yang digunakan untuk mengidentifikasi posisi wajah. Koordinat wajah yang digunakan untuk mengendalikan perangkat dalam program pelacakan wajah seringkali membantu dalam identifikasi seseorang (Arief dkk., 2025).

Dengan algoritma yang telah banyak dikembangkan, komputer mampu mengidentifikasi wajah melalui tekstur, bentuk, dan faktor lain yang direkam oleh kamera. Metode ini bekerja dengan menetapkan titik-titik pada gerakan wajah manusia yang kemudian dapat dimanfaatkan untuk menghasilkan ilustrasi yang mengikuti pergerakan tersebut. Proses pemantauan wajah tergolong kompleks, mengingat wajah manusia sangat dinamis dan dapat dipengaruhi oleh berbagai faktor seperti pencahayaan, ekspresi, transformasi, pose, maupun perubahan sudut pandang (Ilmiah dan Grafis, 2023). Teknologi ini mampu :

- Mendeteksi keberadaan wajah dalam *frame*.
- Melacak pergerakan kepala dan ekspresi wajah.

Beberapa teknologi yang sering digunakan untuk *face tracking* adalah:

- *Haar Cascade Classifier*.
- *Dlib Facial Landmark Detection*.
- *MediaPipe Face Mesh (Google)*.

Face tracking sangat penting dalam mendeteksi perilaku mencurigakan seperti sering menoleh, melihat ke bawah, atau wajah yang sering hilang dari layar.

2.2.4 Ekstraksi Fitur Visual

Ekstraksi fitur visual merupakan proses mengambil informasi penting dari data video wajah peserta ujian menggunakan teknologi *face tracking*. Proses ini dilakukan untuk mengidentifikasi pola gerakan atau perubahan posisi wajah yang dapat menunjukkan perilaku mencurigakan selama ujian online. Ekstraksi fitur (*feature extraction*) merupakan komponen penting dalam analisis citra, yang berfungsi untuk memperoleh informasi atau karakteristik yang terdapat pada suatu citra. Informasi tersebut kemudian dijadikan acuan dalam membedakan antara citra satu dengan yang lainnya berdasarkan karakteristik unik dari objek. Dalam penelitian ini, fitur yang diekstraksi meliputi warna, bentuk, dan tekstur. Karakteristik fitur yang dihasilkan diharapkan memenuhi kriteria sebagai berikut:

- Fitur yang diekstraksi harus mampu membedakan suatu objek dengan objek lainnya (*discrimination*).
- Proses ekstraksi fitur perlu mempertimbangkan kompleksitas komputasi, sehingga tetap efisien dalam hal waktu pemrosesan dan penggunaan sumber

daya.

- Kompleksitas komputasi yang tinggi tentu akan menjadi beban tersendiri dalam menemukan suatu fitur.
- Tidak terikat (*independence*) dalam arti bersifat invarian terhadap berbagai transformasi (rotasi, penskalaan, pergeseran, dan lain sebagainya).
- Jumlah fitur yang relatif sedikit akan memberikan keuntungan, karena dapat menghemat waktu komputasi serta kebutuhan ruang penyimpanan pada proses berikutnya (Arifin dan Jason Hendryli, 2021).

Fitur-fitur utama yang diekstraksi dari hasil *face tracking* meliputi :

- Deteksi keberadaan wajah sistem akan mendeteksi apakah wajah peserta selalu terlihat oleh kamera. Wajah yang sering hilang dari frame dapat menjadi indikasi adanya tindakan kecurangan.
- Pergerakan kepala pergerakan kepala yang berlebihan (misalnya sering menoleh ke samping atau ke bawah) dapat dikategorikan sebagai aktivitas tidak wajar saat ujian.
- Durasi ketidakterlihatan wajah sistem mencatat berapa lama wajah tidak terdeteksi oleh kamera. Jika melebihi batas waktu tertentu, maka bisa dianggap mencurigakan.

Data hasil ekstraksi fitur tersebut selanjutnya dijadikan sebagai masukan pada algoritma klasifikasi *Support Vector Machine* (SVM). Algoritma ini bertugas menentukan apakah perilaku peserta termasuk dalam kategori wajar (tidak curang) atau justru mencurigakan (curang).

2.2.5 Support Vector Machine

Support Vector Machine (SVM) merupakan salah satu algoritma supervised learning yang dapat digunakan baik untuk klasifikasi maupun regresi. Sebagai metode *machine learning* modern, SVM dimanfaatkan dalam berbagai tugas prediksi, termasuk pengelompokan data dan pemodelan regresi. Pendekatan ini menerapkan prinsip statistik dalam proses pengenalan pola, dengan tujuan menemukan *support vector* yang optimal untuk memisahkan dua kelas berbeda menggunakan margin pemisah terbesar.

Pada awalnya, *Support Vector Machine* (SVM) dirancang untuk menyelesaikan permasalahan klasifikasi dua kelas. Namun, seiring perkembangannya, metode ini dapat diperluas untuk menangani klasifikasi multikelas. Dalam ruang dua dimensi, pemisah antar kelas direpresentasikan oleh sebuah garis; pada ruang tiga dimensi berupa sebuah bidang; sedangkan pada dimensi yang lebih tinggi disebut sebagai *hyperplane*. SVM pertama kali digunakan untuk mengklasifikasikan data menjadi dua kategori. Seiring perkembangannya, SVM bisa diperluas untuk klasifikasi multikelas. Pendekatan yang digunakan pada SVM multikelas berbeda dengan kasus klasifikasi dua kelas. Salah satu strategi yang umum digunakan adalah metode *One-Against-One* (OAO), di mana sejumlah model SVM biner dibangun untuk membandingkan setiap pasangan kelas. Untuk mengelompokkan data ke dalam k kelas, diperlukan kombinasi antar pasangan kelas, sehingga total model SVM biner yang harus dibangun berjumlah $k(k-1)$ atau $k(k-2)$ (Hendrastuty dkk., 2021).

SVM adalah teknik yang populer untuk mengatasi masalah dalam penambangan data dan klasifikasi. Proses klasifikasi pada *Support Vector Machine* (SVM) dilakukan dengan memanfaatkan sebuah fungsi berupa *hyperplane* yang berfungsi memisahkan dua kelas pola. SVM berusaha menemukan *hyperplane* yang paling optimal sehingga kedua kelas tersebut dapat terpisahkan dengan margin maksimal. Secara prinsip, *Support Vector Machine* (SVM) merupakan metode yang dirancang untuk melakukan klasifikasi biner. Namun, penerapannya dapat diperluas pada kasus klasifikasi multikelas dengan cara membagi dataset ke dalam beberapa pasangan kelas sehingga terbentuk sejumlah klasifikasi biner (Tantika dan Kudus, 2022). Algoritma *Support Vector Machine* (SVM) beroperasi dengan mencari sebuah *hyperplane* yang paling optimal untuk memisahkan dua kelas data dalam ruang dengan dimensi tinggi. Karakteristik SVM :

- Cocok untuk data berdimensi tinggi.
- Dapat menggunakan kernel (*linear, RBF, polynomial*).
- Kuat terhadap overfitting dalam ruang fitur terbatas.

Dalam konteks deteksi kecurangan, SVM digunakan untuk membedakan antara data visual yang tergolong “fokus” dan “curang”.

2.2.6 Real-Time Processing

Real-time processing merupakan suatu sistem yang memiliki kemampuan untuk memproses data secara langsung seiring dengan waktu terjadinya. Dalam konteks penelitian ini, pemrosesan *real-time* dimanfaatkan untuk :

- Memantau wajah peserta ujian secara langsung.
- Mengklasifikasi perilaku peserta.
- Memberikan peringatan otomatis ketika terdeteksi aktivitas mencurigakan.

Hal ini penting agar sistem dapat berfungsi sebagai *proctor* otomatis yang aktif selama ujian berlangsung.

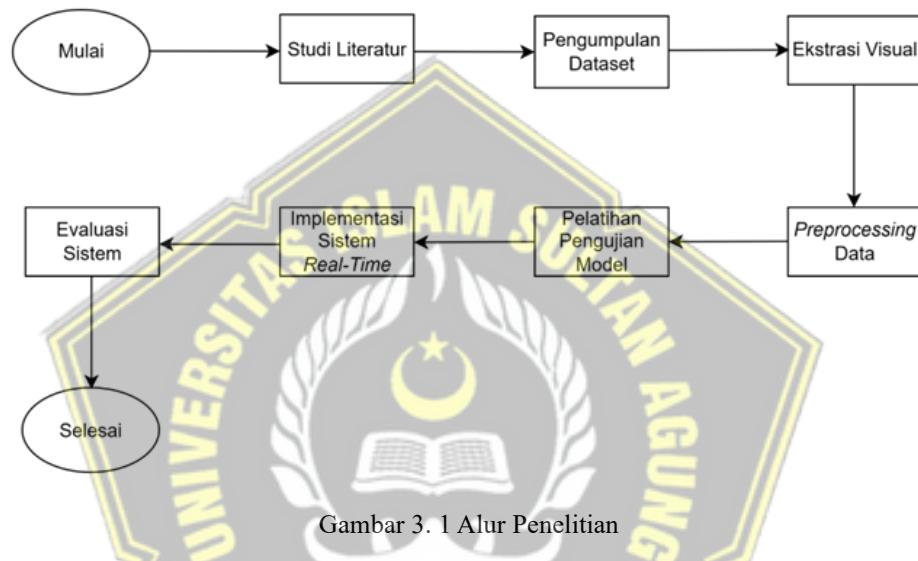


BAB III

METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang diterapkan pada studi ini menggunakan algoritma *Support Vector Machine* (SVM) yang dipadukan dengan model *Face Tracking*. Adapun tahapan penelitian yang dilakukan dapat dijabarkan sebagai berikut :



3.1.1 Studi Literatur

Tahap awal ini berfokus pada pengumpulan informasi dan tinjauan pustaka terhadap penelitian-penelitian sebelumnya yang relevan dengan sistem pengawasan ujian online (*online proctoring*), deteksi wajah, dan metode klasifikasi. Berdasarkan studi ini, algoritma *Support Vector Machine* (SVM) dipilih sebagai metode klasifikasi karena memiliki efektivitas yang tinggi dalam mengolah data dengan dimensi besar. Google *MediaPipe* dipilih sebagai teknologi deteksi wajah karena kinerjanya yang cepat dan akurat untuk aplikasi *real-time*.

3.1.2 Pengumpulan Dataset

Tahap ini bertujuan untuk membangun dataset citra wajah sebagai bahan baku pelatihan model. Proses ini dilakukan menggunakan aplikasi kustom yang mampu merekam dan menyimpan gambar wajah secara otomatis. Data yang dikumpulkan dibagi menjadi dua kelas :

1. Kelas 0 (Fokus) : Berisi gambar-gambar subjek yang sedang menatap layar dan berkonsentrasi, dengan variasi postur natural seperti membaca atau berpikir.
2. Kelas 1 (Curang) : Berisi gambar-gambar subjek yang melakukan berbagai gerakan yang diindikasikan sebagai perilaku curang, seperti menoleh ke samping, melihat ke bawah, dan menggerakkan bibir (berbicara).

3.1.3 Ekstraksi Fitur Visual

Proses ekstraksi fitur dimulai dengan mengiterasi setiap citra wajah yang telah dikumpulkan dalam folder dataset. Untuk setiap citra, model *face mesh* dari pustaka Google *MediaPipe* diaplikasikan untuk mendeteksi dan memetakan lokasi presisi dari 468 titik kunci *landmarks* pada wajah. *Landmarks* ini menyediakan representasi struktural yang kaya dari wajah subjek. Berdasarkan koordinat spasial dari *landmarks* tersebut, tiga fitur numerik utama kemudian dihitung. Fitur pertama adalah rasio posisi hidung horizontal, yang dirancang untuk mengkuantifikasi rotasi wajah pada sumbu *yaw* (menoleh ke kiri atau kanan). Fitur kedua adalah Mouth Aspect Ratio (MAR), yang mengukur tingkat keterbukaan mulut sebagai indikator kuat dari aktivitas berbicara. Fitur ketiga adalah rasio posisi hidung vertikal, yang mengukur rotasi wajah pada sumbu *pitch* (menunduk atau mendongak). Setiap gambar dalam dataset pada akhirnya direduksi menjadi sebuah vektor fitur tunggal yang berisi tiga nilai numerik hasil perhitungan tersebut, secara efektif meringkas postur wajah yang relevan ke dalam format yang ringkas dan terukur.

3.1.4 Preprocessing Data

Setelah proses ekstraksi selesai untuk seluruh gambar, tahap pra-pemrosesan data dimulai. Seluruh vektor fitur yang telah dihasilkan diagregasi menjadi sebuah matriks fitur X, dan label kelas yang bersesuaian (0 untuk Fokus, 1 untuk Curang) disusun menjadi sebuah vektor target y. Langkah penting berikutnya adalah membagi dataset ke dalam dua bagian terpisah, yaitu data latih (training data) dan data uji (testing data). Pembagian ini dilakukan dengan proporsi 80% untuk data latih dan 20% untuk data uji menggunakan fungsi `train_test_split` dari pustaka *Scikit-learn*. Untuk memastikan validitas dan reproducibilitas hasil penelitian, parameter `random_state` digunakan agar proses pengacakan data selalu konsisten.

Selain itu, parameter *stratify* digunakan untuk menjaga keseimbangan proporsi kelas "Fokus" dan "Curang" baik pada data latih maupun data uji, sehingga tetap konsisten dengan distribusi pada dataset asli. Hasil akhir dari proses ini berupa empat himpunan data terpisah (X_{train} , y_{train} , X_{test} , y_{test}) yang telah siap digunakan dalam tahap pelatihan serta pengujian model.

3.1.5 Pelatihan Pengujian Model

Proses pelatihan model dilakukan dengan memanggil metode `.fit()` pada data latih, yang terdiri dari matriks fitur X_{train} dan vektor label y_{train} . Selama proses ini, algoritma SVM secara iteratif bekerja untuk menemukan sebuah batas keputusan optimal (optimal *hyperplane*) di dalam ruang fitur (*feature space*). *Hyperplane* ini adalah bidang yang paling baik dalam memisahkan data dari kelas "Fokus" dengan data dari kelas "Curang", dengan margin atau jarak sebesar mungkin. Setelah proses fitting ini selesai, model dianggap telah mempelajari pola-pola yang membedakan kedua perilaku tersebut dan siap untuk diuji.

Tahap berikutnya adalah pengujian model, yang bertujuan untuk mengevaluasi sejauh mana kemampuan generalisasi model terhadap data yang belum pernah digunakan pada proses pelatihan. Model yang telah dilatih kemudian diaplikasikan untuk melakukan prediksi terhadap matriks fitur X_{test} . Hasil prediksi ini (y_{pred}) kemudian dibandingkan dengan label sebenarnya (y_{test}) untuk mengukur performa model. Evaluasi kinerja dilakukan secara komprehensif menggunakan beberapa metrik standar. *Confusion Matrix* digunakan sebagai dasar untuk melihat rincian prediksi yang benar dan salah (*True Positive*, *True Negative*, *False Positive*, *False Negative*). Dari matriks ini, dihitung metrik turunan seperti Akurasi untuk mengukur persentase prediksi benar secara keseluruhan, Presisi untuk mengukur keandalan prediksi positif (seberapa sering prediksi "Curang" itu benar), dan Recall untuk mengukur sensitivitas model (seberapa banyak kasus "Curang" yang berhasil terdeteksi). Hasil kuantitatif dari metrik-metrik ini akan dianalisis pada Bab 4, dan model yang telah tervalidasi kemudian disimpan sebagai file `svm_model.pkl` untuk diimplementasikan pada sistem real-time.

3.1.6 Evaluasi Model

Setelah model dilatih, kinerjanya harus diuji secara objektif. Pengujian dilakukan dengan memberikan data yang belum pernah dilihat sebelumnya, yaitu matriks fitur X_{test} . Model kemudian menghasilkan prediksi (y_{pred}) untuk setiap sampel dalam X_{test} . Prediksi ini kemudian dibandingkan dengan label yang sebenarnya (y_{test}) untuk mengukur seberapa akurat model tersebut. Parameter evaluasi utama yang digunakan meliputi nilai akurasi, loss (fungsi kerugian), serta analisis lebih mendalam melalui *confusion matrix* dan *classification report*. *Confusion matrix* digunakan untuk melihat distribusi hasil prediksi model terhadap kelas yang sebenarnya, sehingga dapat diketahui sejauh mana kemampuan model dalam mendekripsi kategori “Curang” maupun “Tidak Curang”. Selain itu, *classification report* menghasilkan metrik evaluasi seperti *precision*, *recall*, dan *F1-score* pada masing-masing kelas, yang berguna untuk menganalisis kekuatan serta kelemahan model dalam melakukan klasifikasi pada setiap kategori.

Pengukuran *Accuracy*, *Precision*, *Recall*, dan *F1-Score* dilakukan dengan memanfaatkan metode *confusion matrix*, yang didasarkan pada nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), serta *False Negative* (FN). Persamaan pertama digunakan untuk menghitung permutasi *accuracy*, persamaan kedua untuk menghitung permutasi *precision*, persamaan ketiga untuk menghitung permutasi *recall*, dan persamaan keempat untuk menghitung permutasi *F1-Score* dengan lebih dari dua kelas keluaran.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad (3)$$

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\% \quad (4)$$

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\% \quad (5)$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}} \times 100\% \quad (6)$$

Keterangan :

TP : Jumlah data *actual* yang sebenarnya benar dan diprediksi benar.

TN : Jumlah data *actual* yang sebenarnya salah dan diprediksi salah.

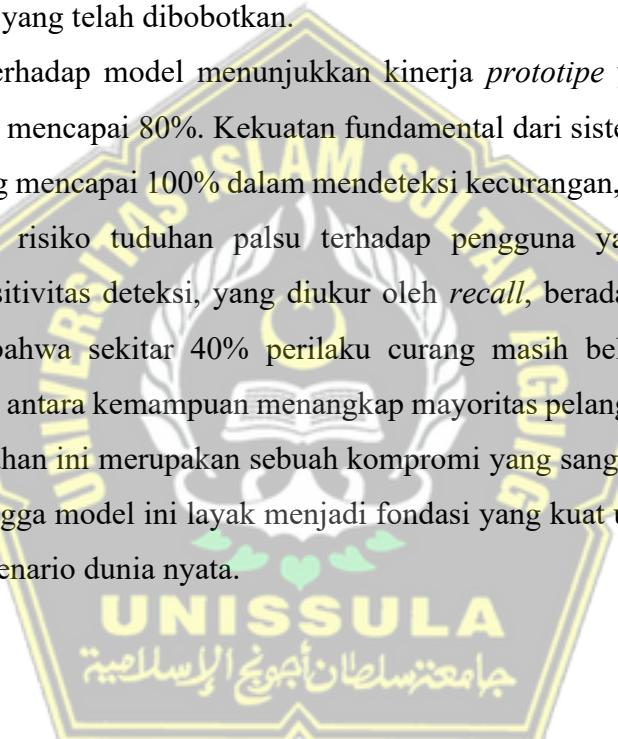
FP : Jumlah data *actual* yang sebenarnya benar dan diprediksi salah.

FN : Jumlah data *actual* yang sebenarnya salah dan diprediksi benar.

Tiga jenis metrik evaluasi yang diperoleh dari confusion matrix dalam menghitung performa prediksi antara lain :

1. *Precision*, yaitu rasio antara jumlah prediksi benar terhadap total hasil prediksi positif maupun negatif.
2. *Recall*, yaitu rasio antara jumlah prediksi benar dengan total nilai aktual pada kelas positif atau negatif.
3. *F1-Score*, yaitu nilai rata-rata harmonis (harmonic mean) dari precision dan recall yang telah dibobotkan.

Evaluasi terhadap model menunjukkan kinerja *prototipe* yang solid, dengan tingkat akurasi mencapai 80%. Kekuatan fundamental dari sistem ini terletak pada presisinya yang mencapai 100% dalam mendeteksi kecurangan, yang secara efektif mengeliminasi risiko tuduhan palsu terhadap pengguna yang jujur. Kendati demikian, sensitivitas deteksi, yang diukur oleh *recall*, berada pada angka 60%, menandakan bahwa sekitar 40% perilaku curang masih belum teridentifikasi. Keseimbangan antara kemampuan menangkap mayoritas pelanggaran dan nihilnya kesalahan tuduhan ini merupakan sebuah kompromi yang sangat efektif dan dapat diterima, sehingga model ini layak menjadi fondasi yang kuat untuk implementasi sistem pada skenario dunia nyata.



3.1.7 Implementasi Sistem *Realtime*

Tahap implementasi sistem real-time merupakan fase di mana model *machine learning* yang telah dilatih dan divalidasi pada tahap sebelumnya akan diintegrasikan ke dalam sebuah aplikasi prototipe yang fungsional. Tujuan utama dari tahap ini adalah merancang sebuah sistem yang mampu menerapkan model klasifikasi SVM pada aliran video secara langsung dari kamera, sekaligus memberikan umpan balik secara instan kepada pengguna. Dengan demikian, alur kerja pemrosesan data dapat berlangsung secara *real-time*.



Gambar 3. 2 Proses implementasi sistem real-time

Flowcart pada gambar 3.2 menggambarkan proses akan dimulai dengan implementasi sistem, yang mencakup pemuatan model SVM (*svm_model.pkl*) yang telah dilatih ke dalam memori dan pengaktifan modul akuisisi video. Setelah sistem siap, proses akan memasuki *loop* deteksi yang kontinu untuk setiap *frame* video yang masuk.

Proses *real-time* dimulai dengan akuisisi Video Input Via Webcam. Untuk setiap *frame* yang ditangkap, sistem akan langsung mendeteksi wajah peserta menggunakan model *face mesh* dari *MediaPipe*. Setelah wajah terdeteksi, tahap

Ekstraksi Fitur Visual dijalankan untuk memetakan 468 *landmarks* dan menghitung tiga fitur numerik utama (rasio posisi hidung, Mouth Aspect Ratio, dan rasio posisi vertikal). Vektor fitur yang dihasilkan kemudian melalui tahap memproses data, di mana data diformat menggunakan *numPy* agar siap untuk dianalisis.

Selanjutnya, vektor fitur tersebut akan di klasifikasi oleh SVM untuk menentukan apakah perilaku tergolong "Fokus" atau "Curang". Hasil prediksi ini akan Muncul sebagai Output pada antarmuka secara multimodal, yaitu berupa umpan balik visual (teks dan warna pada layar) dan umpan balik audio (peringatan suara jika terjadi pelanggaran). Sebagai langkah akhir dalam siklus, pada tahap Data *Output* Disimpan, setiap insiden pelanggaran yang terdeteksi akan dicatat ke dalam sebuah file log eksternal untuk keperluan audit. Seluruh siklus proses ini berjalan secara kontinu untuk setiap *frame* video guna memastikan pemantauan yang berkelanjutan selama sesi ujian.

3.1.8 Deployment Model

Setelah sistem berhasil diimplementasikan dan diuji secara lokal, tahap berikutnya adalah *deployment* model, yaitu proses memublikasikan aplikasi yang telah dikembangkan ke dalam lingkungan daring sehingga dapat diakses secara langsung oleh pengguna melalui peramban web, tanpa perlu melakukan instalasi perangkat lunak di komputer masing-masing. Untuk penelitian ini, proses deployment akan menggunakan platform *Streamlit Community Cloud*, sebuah layanan gratis yang dioptimalkan khusus untuk aplikasi *Streamlit*.

Proses persiapan deployment akan dimulai dengan manajemen dependensi. Seluruh pustaka Python yang digunakan dalam proyek (seperti *streamlit*, *opencv-python-headless*, *mediapipe*, *scikit-learn*, *gTTS*, dan *playsound*) akan didaftarkan dalam sebuah file konfigurasi bernama *requirements.txt*. File ini berfungsi sebagai pedoman bagi server dalam menginstal seluruh perangkat lunak yang diperlukan sehingga aplikasi dapat dijalankan dengan baik .Selanjutnya, keseluruhan kode sumber proyek termasuk script utama *app.py*, file *requirements.txt*, model SVM yang telah dilatih (*svm_model.pkl*).

Proses deployment itu sendiri akan dieksekusi melalui dasbor *Streamlit Community Cloud*. Akun GitHub peneliti akan dihubungkan ke platform tersebut,

kemudian repositori proyek yang relevan akan dipilih. Platform akan secara otomatis membaca file requirements.txt untuk membangun lingkungan virtual dan menginstal semua dependensi yang diperlukan. Setelah konfigurasi selesai, aplikasi akan diluncurkan pada sebuah server dan akan diberikan sebuah URL publik yang unik. Hasil akhir dari tahap ini adalah sebuah aplikasi web fungsional yang dapat diakses secara global, yang memungkinkan proses demonstrasi, pengujian, dan evaluasi oleh pihak lain (seperti dosen pembimbing dan penguji) dapat dilakukan dengan mudah dan efisien.

3.2 Software Yang Digunakan

1. Bahasa Pemrograman
 - a. Phyton

Python dipilih sebagai bahasa utama untuk keseluruhan proses pengembangan, mulai dari pengumpulan data hingga implementasi sistem real-time. Keputusan ini didasarkan pada beberapa keunggulan fundamental yang menjadikan Python sebagai standar industri, Khususnya pada ranah kecerdasan buatan (*Artificial Intelligence* atau AI) dan ilmu data (Data Science). Alasan utama pemilihan Python adalah ekosistem *libraries* (pustaka) yang sangat luas. Penelitian ini sangat bergantung pada fungsionalitas dari berbagai pustaka pihak ketiga yang kompleks, dan Python berfungsi sebagai "lem" atau antarmuka tingkat tinggi (*high-level interface*) yang mengintegrasikan semua teknologi ini ke dalam satu alur kerja yang kohesif. Pustaka-pustaka kunci seperti *OpenCV* (untuk pemrosesan citra dan video), *MediaPipe* (untuk deteksi landmarks wajah), dan *Scikit-learn* (untuk implementasi model SVM dan evaluasi) semuanya memiliki *Application Programming Interface* (API) yang paling stabil dan didukung penuh untuk Python. Hal ini memungkinkan peneliti untuk memanfaatkan kekuatan algoritma-algoritma kompleks yang seringkali ditulis dalam bahasa tingkat rendah (seperti C++) melalui sintaks Python yang sederhana dan mudah dibaca.

2. Perangkat Lunak (Software)

a. *Visual Studio Code*

Digunakan sebagai *Integrated Development Environment* (IDE) atau editor kode utama. Seluruh proses penulisan, pengelolaan, dan *debugging* kode Python dilakukan di dalam VS Code. Fitur utamanya seperti terminal terintegrasi (untuk menjalankan perintah pip dan python) serta dukungan ekstensi untuk Python sangat mempermudah dan mempercepat alur kerja pengembangan sistem ini.

3. *Library dan Framework*

a. *Mediapipe*

Framework (kerangka kerja) dan *open-source* (sumber terbuka) yang dikembangkan oleh Google untuk membangun *pipeline* pemrosesan media yang kompleks dan multimodal. Secara sederhana, *MediaPipe* adalah sebuah "kotak peralatan" canggih yang memungkinkan pengembang untuk menganalisis video, audio, dan data sensor lainnya secara *real-time* dengan performa tinggi.

b. *Scikit-learn*

Library ini digunakan dalam beberapa bagian penting, seperti `train_test_split` untuk membagi data, `resample` untuk undersampling, `compute_class_weight` untuk menangani ketidakseimbangan kelas, dan `classification_report` serta `confusion_matrix` untuk mengevaluasi performa model.

c. *Numpy*

NumPy digunakan untuk operasi *array* numerik, termasuk normalisasi dan konversi label. Selain itu, *library* ini juga berperan dalam pengolahan label untuk menghitung *class weight* agar pelatihan model lebih adil terhadap kelas minoritas.

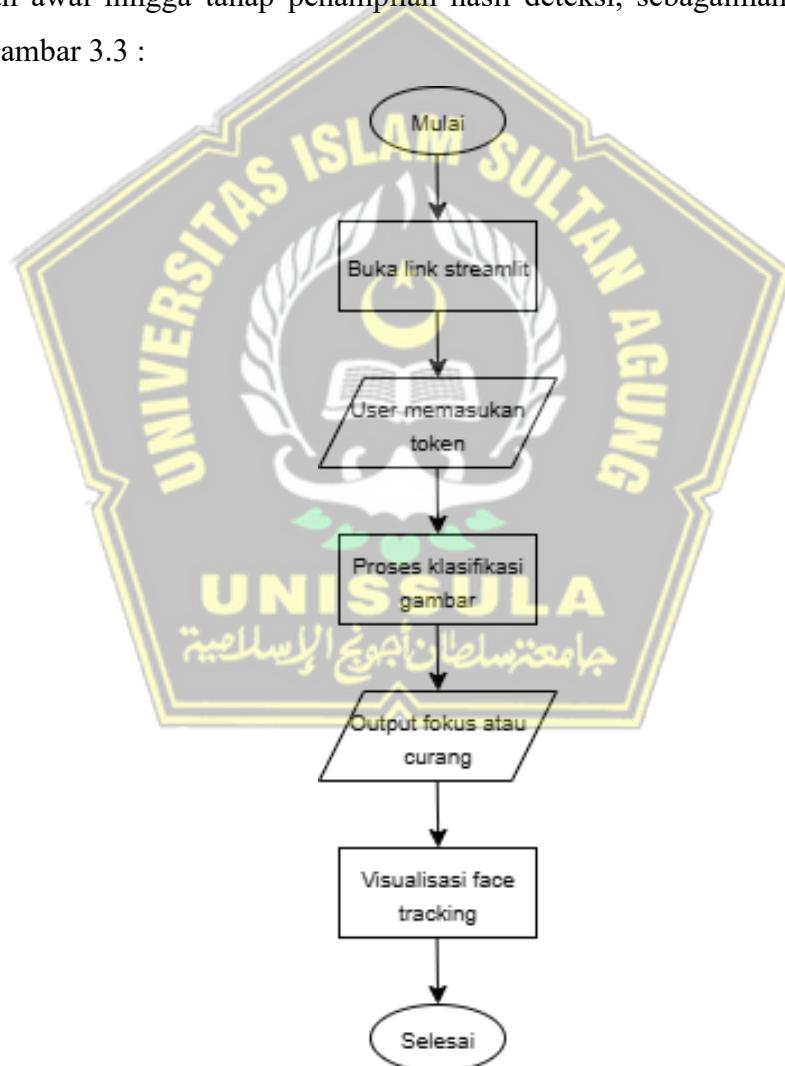
d. *Google Text-to-Speech (gTTS)*

gTTS digunakan untuk menghasilkan ucapan yang tidak hanya jelas secara fonetik, tetapi juga memiliki intonasi, ritme, dan nada yang natural, meniru cara manusia berbicara. Layanan ini dapat diakses oleh

para pengembang melalui berbagai cara, salah satunya adalah melalui pustaka Python gTTS, yang menyederhanakan proses interaksi dengan *Application Programming Interface (API)* Google.

3.3 Perancangan Sistem

Perancangan sistem aplikasi bertujuan untuk membangun antarmuka pengguna (UI) yang interaktif dan mudah diakses menggunakan *framework streamlit*. Alur kerja sistem disusun sedemikian rupa untuk memandu pengguna mulai dari halaman awal hingga tahap penampilan hasil deteksi, sebagaimana ditunjukkan pada Gambar 3.3 :



Gambar 3. 3 Flowcart alur sistem streamlit

Gambar 3.3 merupakan flowchart streamlit yang menjelaskan alur kerja aplikasi web dalam menjalankan sesi pengawasan ujian dari awal hingga akhir. Tahapan alur kerja streamlit tersebut dijelaskan sebagai berikut :

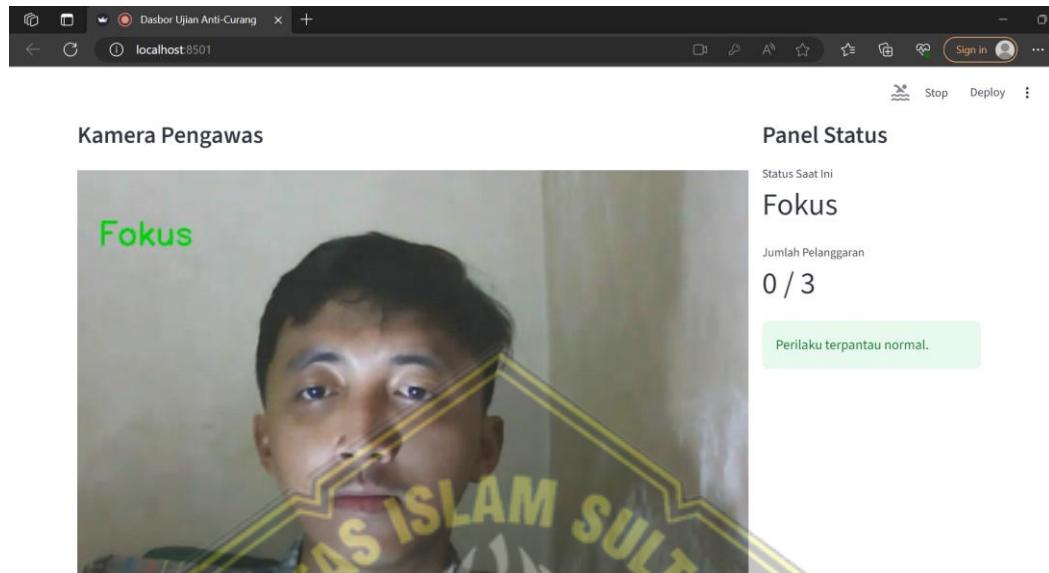
- a. Tahap pertama, pengguna mengakses tautan aplikasi web yang dikembangkan dengan *Streamlit*.
- b. Selanjutnya, pengguna memasukkan token ujian pada kolom yang telah disediakan, kemudian menekan tombol “Mulai Ujian” untuk menjalani proses validasi.
- c. Setelah token valid, sistem akan mengaktifkan kamera dan memulai proses inti. Untuk setiap *frame* video, sistem akan melakukan serangkaian proses internal:
 - Deteksi wajah dan landmarks menggunakan MediaPipe.
 - Ekstraksi fitur numerik (rasio posisi hidung, MAR, dll.).
 - Klasifikasi fitur menggunakan model SVM yang telah dilatih.
- d. Hasil dari proses klasifikasi akan menjadi output utama. Model svm akan menentukan apakah perilaku pengguna tergolong "Fokus" atau "Curang".
- e. *Output* dari klasifikasi akan divisualisasikan secara langsung. Ini mencakup:
 - Teks "Fokus" atau "Curang" yang digambar di atas video.
 - Panel status yang menampilkan jumlah pelanggaran secara *real-time*.
- f. Sesi akan berakhir jika pengguna menekan tombol "Akhiri Ujian" atau jika jumlah pelanggaran mencapai batas maksimal, yang akan mengembalikan pengguna ke halaman login.

3.4 Perancangan *User Interface*

3.4.1 Halaman Awal Sistem

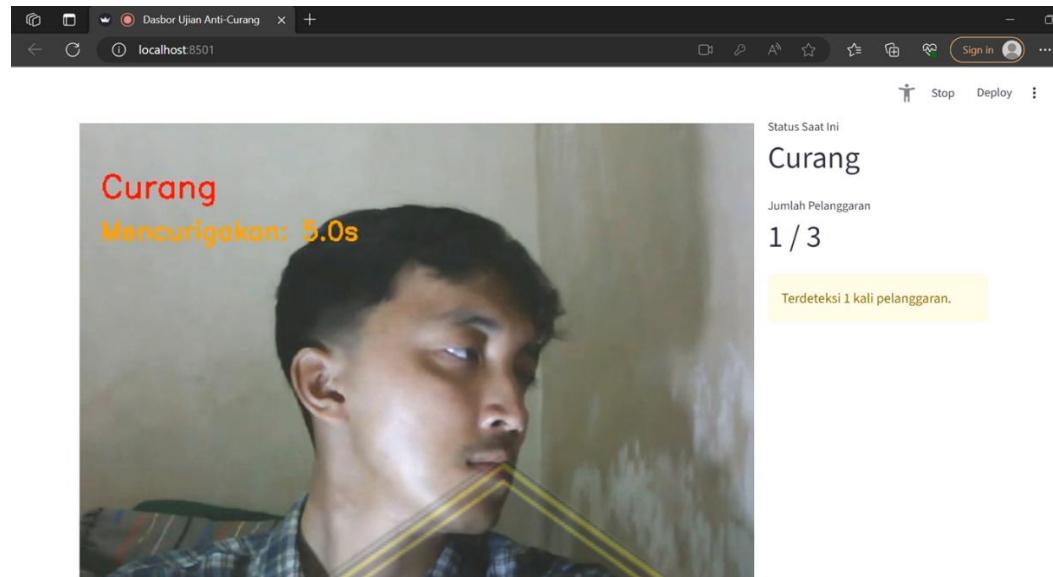


3.4.2 Tampilan Hasil Sistem



Gambar 3.5 Tampilan hasil sistem mendekripsi perilaku fokus

Gambar 3.5 menampilkan antarmuka pengguna (UI) dari Dasbor Pemantauan Ujian berbasis *Streamlit* yang sedang berjalan. Tampilan terbagi menjadi dua kolom utama yang menyajikan informasi secara *real-time*. Kolom Kiri: Kamera PengawasIni adalah area utama yang menampilkan aliran video langsung dari kamera web pengguna. Di dalam video, sistem secara aktif melakukan analisis dan memberikan umpan balik visual instan. Pada screenshot ini, terlihat teks "Fokus" berwarna hijau digambar di atas video, yang mengindikasikan bahwa model SVM telah mengklasifikasikan perilaku pengguna sebagai normal atau tidak curang. Kolom Kanan: Panel StatusPanel ini berfungsi sebagai pusat informasi dan rangkuman dari sesi pengawasan.



Gambar 3.6 Tampilan hasil sistem mendeteksi perilaku curang

Gambar 3.6 menampilkan respons sistem secara real-time ketika mendeteksi perilaku yang diklasifikasikan sebagai "Curang". Dalam skenario ini, subjek melakukan gerakan menoleh ke samping secara persisten. Setelah perilaku ini berlangsung selama 5 detik, seperti yang ditunjukkan oleh teks "Mencurigakan: 5.0s" pada video, sistem mengonfirmasi ini sebagai sebuah status pelanggaran. Hasilnya, label prediksi di atas video berubah menjadi "Curang" dengan warna merah. Panel Status di sebelah kanan langsung diperbarui untuk merefleksikan deteksi ini. "Status Saat Ini" berubah menjadi "Curang", dan "Jumlah Pelanggaran" bertambah dari 0 menjadi 1 / 3. Sebuah pesan peringatan berwarna kuning juga muncul untuk memberikan notifikasi yang jelas kepada pengawas. Perubahan visual ini juga diiringi dengan peringatan suara "Pelanggaran ke satu" yang diputar secara otomatis. Keseluruhan respons ini menunjukkan bahwa sistem tidak hanya berhasil mengklasifikasikan perilaku curang, tetapi juga mampu menerapkan logika timer yang telah ditentukan dan memberikan umpan balik multimodal (visual dan audio) yang terintegrasi dan informatif.

BABI IV

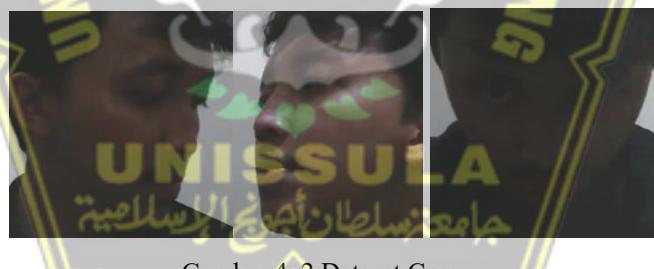
HASIL DAN ANALISIS PENELITIAN

4.1 Hasil Pengumpulan Dataset

Tahap pengumpulan dataset menjadi fondasi penting dalam penelitian ini, karena kualitas serta keragaman data yang diperoleh sangat berpengaruh terhadap kemampuan generalisasi maupun tingkat akurasi model *Support Vector Machine* (SVM) yang dilatih. Tahap ini bertujuan untuk membangun dataset citra wajah yang merepresentasikan dua kategori perilaku pada ujian online, yaitu “Fokus” (tidak curang) dan “Curang”. Dataset tersebut disusun secara khusus agar dapat digunakan sebagai dasar dalam proses klasifikasi.



Gambar 4. 1 Dataset Fokus (tidak curang)



Gambar 4. 2 Dataset Curang

Gambar 4.1 dan Gambar 4.2 memperlihatkan proses pengumpulan dataset yang dilakukan melalui aplikasi kustom. Aplikasi ini dikembangkan menggunakan bahasa pemrograman *Python* dengan memanfaatkan pustaka *OpenCV* untuk akses kamera serta *MediaPipe* sebagai alat deteksi wajah. Spesifikasi perangkat lunak yang digunakan adalah webcam eksternal dari laptop Lenovo IdeaPad dengan resolusi 720p. Proses pengambilan data dilakukan dalam kondisi pencahayaan ruangan yang konsisten untuk menjaga kualitas gambar. Setiap gambar yang disimpan adalah hasil *cropping* otomatis di sekitar area wajah yang terdeteksi untuk

memastikan data yang dilatih relevan dan bebas dari *noise* latar belakang yang tidak perlu.

Data dikumpulkan untuk dua kelas yang berbeda, yang masing-masing merepresentasikan skenario perilaku yang kontras. Proses ini dilakukan melalui antarmuka interaktif pada aplikasi yang memungkinkan peneliti memilih label data yang akan direkam.

1. Kelas 0 : Fokus (Tidak Curang)

Kelas ini berisi gambar-gambar yang mewakili kondisi peserta yang sedang berkonsentrasi pada ujiannya. Skenario yang direkam untuk kelas ini mencakup variasi postur alami seperti menatap lurus ke layar, membaca soal dengan sedikit gerakan mata, dan ekspresi berpikir.

2. Kelas 1 : Curang

Kelas ini berisi gambar-gambar yang mewakili berbagai macam gerakan yang diidentifikasi sebagai indikasi kecurangan. Untuk memastikan model dapat mengenali berbagai modus, skenario yang direkam sangat beragam, meliputi :

- Menoleh secara perlahan atau cepat ke arah kiri dan kanan.
- Menundukkan kepala untuk melihat ke bawah (simulasi melihat catatan atau ponsel).
- Gerakan bibir yang mengindikasikan berbicara atau berbisik.

Setelah melalui proses pengumpulan data, terkumpul sejumlah gambar yang telah terbagi ke dalam dua folder sesuai dengan labelnya.

Tabel 4. 1 Komposisi Dataset Final

No	Nama Folder (label)	Deskripsi Kelas	Jumlah Gambar
1	0_fokus	Peserta dalam kondisi fokus pada ujian.	100
2	1_curang	Peserta melakukan gerakan indikasi curang.	100

Berdasarkan Tabel 4.1, dataset yang berhasil dikumpulkan terdiri dari 100 gambar, dengan distribusi yang relatif seimbang antara kelas "Fokus" dan

"Curang". Keseimbangan jumlah sampel ini merupakan aspek penting untuk mencegah model menjadi bias terhadap salah satu kelas selama proses pelatihan.

Keberhasilan pengumpulan data dengan berbagai skenario, terutama pada kelas "Curang". Dengan merekam tidak hanya gerakan menoleh yang ekstrem tetapi juga lirikan cepat dan gerakan menunduk, dataset ini dirancang untuk melatih model yang lebih akurat dan mampu mengenali indikasi kecurangan yang lebih halus. Proses *cropping* wajah secara otomatis juga memastikan bahwa fitur yang akan diekstraksi pada tahap selanjutnya benar-benar berasal dari area wajah, sehingga meningkatkan relevansi data.

4.2 Hasil Ekstrasi Fitur Visual

Setelah dataset gambar berhasil dikumpulkan, tahap selanjutnya adalah Ekstraksi Fitur Visual. Tahap ini dilakukan dengan tujuan menyederhanakan data gambar mentah yang bersifat kompleks menjadi representasi numerik yang lebih terstruktur dan bermakna berupa vektor fitur. Representasi tersebut kemudian dimanfaatkan oleh model *Support Vector Machine* (SVM) dalam proses klasifikasi. Untuk mengekstrak fitur dari setiap gambar wajah di dalam dataset, penelitian ini memanfaatkan teknologi face mesh dari pustaka Google *MediaPipe*.

Berdasarkan analisis terhadap perilaku "Fokus" dan "Curang" dari dataset yang telah dikumpulkan, tiga fitur kuantitatif utama dirancang untuk menangkap esensi dari gerakan-gerakan tersebut. Penggunaan rasio sebagai dasar perhitungan memastikan bahwa fitur-fitur ini tidak terpengaruh oleh faktor-faktor seperti jarak subjek ke kamera atau ukuran wajah dalam *frame*.

- Fitur 1 : Rasio Posisi Hidung (Fitur Arah Horizontal)

Fitur ini dirancang secara spesifik untuk mengukur gerakan menoleh ke kiri atau ke kanan, yang merupakan indikator kecurangan paling dominan dalam dataset 1_curang. Fitur dihitung dengan mengambil posisi sumbu-x dari ujung hidung (*Landmark* 1) dan menormalisasikannya berdasarkan lebar wajah (jarak horizontal antara titik pipi terluar kiri, *Landmark* 234, dan kanan, *Landmark* 454). Pada sampel di folder 0_fokus, nilai rasio ini cenderung konsisten di sekitar 0.5 (menandakan wajah lurus). Sebaliknya, pada sebagian besar sampel di folder 1_curang, di mana

subjek menoleh, nilai rasio ini akan menyimpang secara signifikan (mendekati 0 atau 1), memberikan sinyal numerik yang jelas untuk dideteksi oleh model.

- Fitur 2 : *Mouth Aspect Ratio* (MAR) (Fitur Deteksi Berbicara)

Untuk mendeteksi aktivitas berbicara atau berbisik, yang merupakan skenario kecurangan lainnya. Dihitung dengan mengukur jarak vertikal absolut antara titik bibir atas (*Landmark* 13) dan bibir bawah (*Landmark* 14). Nilai ini kemudian dinormalisasi dengan lebar wajah untuk menjaga konsistensi. Mayoritas gambar pada dataset 0_fokus menunjukkan mulut dalam keadaan tertutup (nilai MAR rendah). Beberapa sampel pada 1_curang yang merekam skenario berbicara akan memiliki nilai MAR yang lebih tinggi, memberikan dimensi pembeda lainnya bagi model.

- Fitur 3: Rasio Posisi Vertikal (Fitur Arah Vertikal)

Fitur ini dirancang untuk mendeteksi gerakan kepala menunduk, yang seringkali berkorelasi dengan tindakan melihat catatan atau perangkat lain di bawah meja. Dihitung dengan mengukur posisi sumbu-y dari ujung hidung (*Landmark* 1) dan menormalisasikannya berdasarkan tinggi wajah (jarak vertikal antara titik dahi, *Landmark* 10, dan dagu, *Landmark* 152). Gambar pada kelas 1_curang, yang memperlihatkan subjek dalam posisi menunduk, cenderung menghasilkan nilai rasio vertikal yang lebih tinggi. Sebaliknya, gambar pada kelas 0_fokus, di mana posisi kepala peserta relatif tegak, menunjukkan nilai rasio vertikal yang lebih rendah .

Untuk setiap gambar dalam dataset, ketiga nilai fitur numerik yang telah dihitung tersebut digabungkan menjadi sebuah vektor fitur tunggal. Vektor ini adalah representasi matematis dari postur wajah pada gambar tersebut.

```
Vektor Fitur = [Rasio Posisi Hidung, MAR, Rasio Posisi Vertikal].
```

Misalnya, sebuah gambar dari folder 1_curang di mana subjek menoleh ke kiri sambil sedikit menunduk mungkin akan direpresentasikan oleh vektor seperti [0.21, 0.01, 0.65]. Sebaliknya, gambar dari 0_fokus mungkin memiliki vektor [0.49, 0.02, 0.55]. Proses ini diotomatisasi dan diterapkan pada seluruh 200

gambar dalam dataset, yang secara efektif mengubah koleksi gambar menjadi sebuah tabel data numerik yang siap untuk dianalisis.

4.3 Hasil *Preprocessing* Data

Setelah seluruh data gambar dari dataset diubah menjadi format vektor fitur, tahap selanjutnya adalah *Preprocessing* Data. Tahap ini berfokus pada persiapan akhir kumpulan data numerik agar siap digunakan untuk melatih dan mengevaluasi model machine learning. Tujuan utamanya adalah untuk memastikan data terstruktur dengan benar dan terbagi secara proporsional untuk menghasilkan evaluasi kinerja model yang valid dan tidak bias.

Langkah pertama dalam pemrosesan data adalah mengumpulkan seluruh vektor fitur yang telah diekstraksi menjadi satu kesatuan data. Dalam konteks ini, data disusun menjadi dua komponen utama sesuai standar machine learning :

- Matriks Fitur (X) merupakan sebuah matriks yang menyimpan representasi fitur dari setiap gambar, di mana masing-masing baris pada matriks tersebut menggambarkan vektor fitur dari satu citra..
- Vektor Target atau Label (y) adalah sebuah vektor yang memuat informasi kelas untuk setiap baris pada matriks fitur, dengan ketentuan nilai 0 mewakili kelas "Fokus" dan nilai 1 mewakili kelas "Curang".

Pada tahap ini, semua sampel data yang gagal diekstraksi fiturnya (misalnya karena wajah tidak terdeteksi) secara otomatis tidak disertakan, memastikan bahwa dataset yang akan diproses adalah data yang bersih dan lengkap. Berdasarkan informasi bahwa total sampel yang berhasil diekstraksi adalah 200 gambar, dengan komposisi seimbang (100 "Fokus" dan 100 "Curang"), maka hasil dari proses pembagian data 80:20.

Tabel 4. 2 Pembagian dataset

Jenis Data	Proporsi	Total Sampel	Jumlah Sampel Kelas 0 (Fokus)	Jumlah Sampel Kelas 1 (Curang)
Data Latih	80%	160	80	80
Data Uji	20%	40	20	20
Total	100%	200	100	100

Berdasarkan Tabel 4.2 diatas keseluruhan dataset yang digunakan dalam penelitian ini terdiri dari 200 citra wajah, yang telah dikategorikan secara seimbang ke dalam dua kelas: 100 sampel untuk kelas 'Fokus' (0) dan 100 sampel untuk kelas 'Curang' (1). Dalam tahap persiapan data, dataset tersebut dipartisi menjadi set data latih dan set data uji. Sebanyak 80% dari data, atau 160 sampel, ditetapkan sebagai data latih, sementara sisa 20% atau 40 sampel digunakan sebagai data uji. Dengan menerapkan metode pembagian terstratifikasi, proporsi seimbang 50:50 antar kelas secara konsisten dipertahankan pada kedua partisi tersebut. Hasilnya, data latih berisi 80 sampel 'Fokus' dan 80 sampel 'Curang', dan data uji berisi 20 sampel 'Fokus' dan 20 sampel 'Curang', yang mana ini merupakan fondasi ideal untuk melatih dan mengevaluasi kinerja model secara adil dan tidak bias.

4.4 Hasil Pelatihan Model SVM

Setelah melewati tahap pra-pemrosesan data, penelitian berlanjut pada proses pelatihan dan pengujian model. Algoritma *Support Vector Machine* (SVM) dilatih dengan menggunakan 159 sampel data latih. Pada tahap pelatihan ini, model berusaha mengenali pola dari vektor fitur yang tersedia untuk kemudian menentukan batas keputusan (*hyperplane*) yang paling optimal dalam memisahkan dua kelas, yaitu "Fokus" dan "Curang". Kinerja dari model yang telah dilatih kemudian dievaluasi secara objektif menggunakan 40 sampel data uji yang belum pernah dilihat sebelumnya, untuk mengukur kemampuan generalisasi model terhadap data baru.

Tabel 4. 3 Parameter Hyperplane dari Model SVM yang Dilatih

Parameter	Hasil (nilai)
Jumlah support vector (kelas fokus)	48
Jumlah support vector (kelas curang)	48
<i>Total Support Vectors</i>	96
<i>Nilai Intercept Bias</i>	3.66648823

Dari tabel 4.3 menyajikan parameter-parameter kunci yang secara matematis mendefinisikan *hyperplane* yang telah dipelajari oleh model SVM selama proses pelatihan. Parameter ini memberikan wawasan tentang kompleks batas keputusan yang dibuat oleh model.

Tabel tersebut menunjukkan bahwa model menggunakan total 96 sampel dari 157 data latih sebagai support vectors. Jumlah ini terbagi secara seimbang, dengan 48 sampel dari kelas 'Fokus' dan 48 sampel dari kelas 'Curang'. *Support vectors* adalah titik-titik data kritis yang posisinya paling dekat dengan batas keputusan dan secara langsung "menopang" atau menentukan posisi dari *hyperplane* dan margin. Jumlah *support vector* yang signifikan ini (lebih dari separuh data latih) mengindikasikan bahwa batas keputusan antara kedua kelas cukup kompleks, sehingga banyak sampel data yang diperlukan untuk secara presisi mendefinisikan "zona abu-abu" di antara perilaku fokus dan curang.

Parameter Nilai *Intercept* (Bias) sebesar 3.666 adalah konstanta matematis yang menentukan posisi final dari *hyperplane* di dalam ruang fitur. Nilai ini berfungsi untuk menggeser batas keputusan dari titik pusat agar dapat memposisikan diri secara optimal di antara kedua kelas data. Secara keseluruhan, kombinasi dari posisi ke-96 *support vectors* dan nilai *intercept* inilah yang membentuk fungsi keputusan yang disimpan dalam file *svm_model.pkl* dan digunakan untuk memprediksi data baru secara *real-time*.

Adanya peningkatan performa yang sangat signifikan dan menghasilkan model yang jauh lebih seimbang dan dapat diandalkan. Kekuatan utama dari model ini terletak pada presisinya yang sempurna (1.00) untuk kelas "Curang", yang berarti setiap kali sistem memberikan peringatan "Curang", prediksi tersebut 100% akurat. Ini adalah hasil yang sangat krusial dalam konteks aplikasi pengawasan, karena memastikan bahwa sistem tidak akan pernah salah menuduh mahasiswa yang berperilaku jujur. Di sisi lain, *recall* sebesar 60% untuk kelas "Curang" menunjukkan bahwa model berhasil mendeteksi mayoritas (12 dari 20) insiden kecurangan, meskipun masih ada ruang untuk peningkatan sensitivitas. Keseimbangan antara presisi yang sempurna dan *recall* yang baik ini merupakan

sebuah kompromi yang sangat dapat diterima untuk implementasi praktis. Secara keseluruhan, evaluasi ini memvalidasi bahwa model yang dihasilkan efektif dan cukup andal untuk diimplementasikan ke dalam sistem deteksi *real-time*.

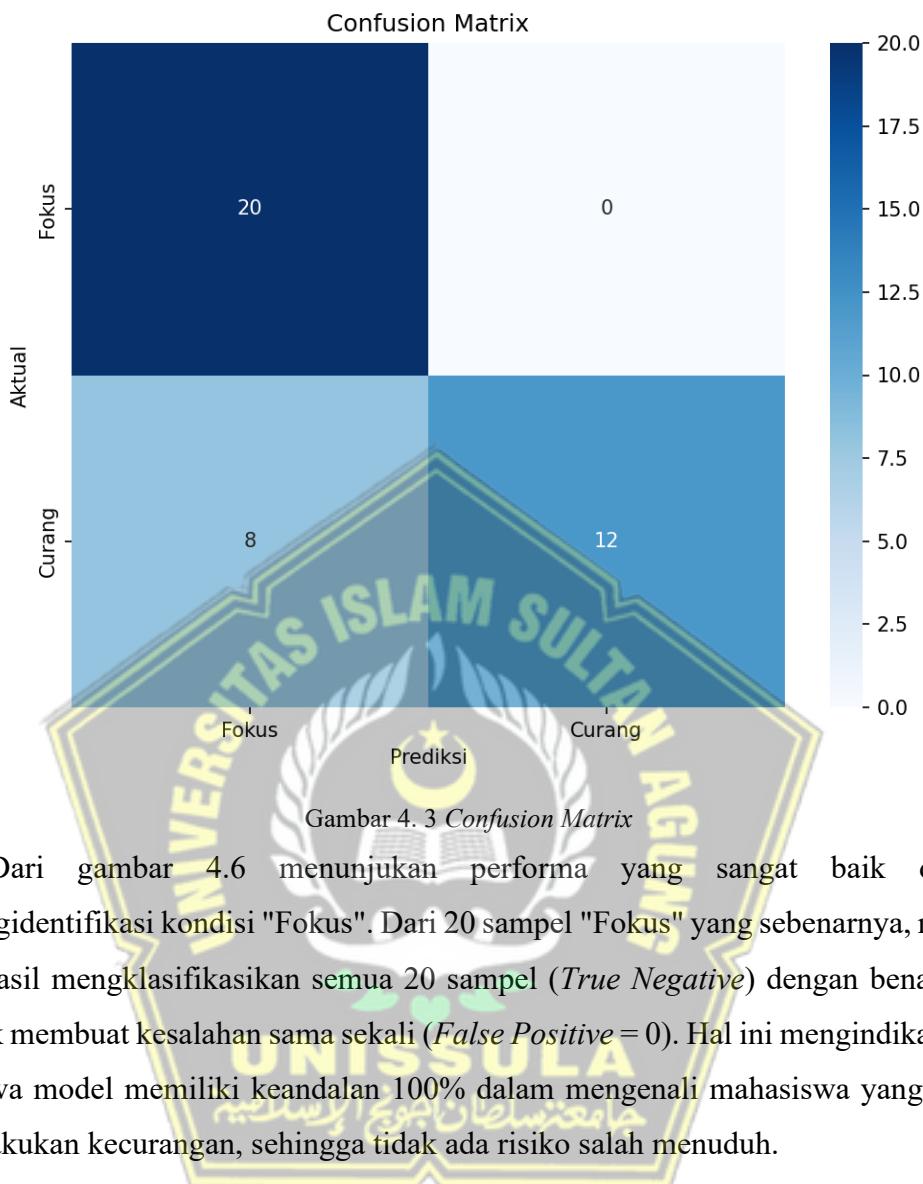
4.5 Evaluasi Sistem

Setelah model dilatih dan diimplementasikan ke dalam sebuah prototipe aplikasi real-time, tahap terakhir dari analisis adalah evaluasi sistem secara menyeluruh. Tahap ini bertujuan untuk menilai kinerja aplikasi secara fungsional dan kualitatif dalam sebuah skenario penggunaan yang disimulasikan. Evaluasi ini berbeda dari pengujian model sebelumnya, karena tidak hanya berfokus pada metrik numerik (akurasi), tetapi juga pada fungsional, performa, dan efektivitas antarmuka aplikasi secara keseluruhan.

4.5.1 Confusion Matrix

Confusion matrix divisualisasikan menggunakan *Confusion Matrix Display* dari *Scikit-learn*, sehingga lebih mudah dalam mengamati pola kesalahan model. Hal ini penting untuk melihat apakah model kesulitan dalam membedakan kelas-kelas yang berdekatan secara visual. Untuk analisis yang lebih mendalam, digunakan *confusion matrix* guna mengevaluasi jumlah prediksi yang benar dan salah pada masing-masing kelas. *Matrix* ini menyajikan informasi sebagai berikut :





Dari gambar 4.6 menunjukkan performa yang sangat baik dalam mengidentifikasi kondisi "Fokus". Dari 20 sampel "Fokus" yang sebenarnya, model berhasil mengklasifikasikan semua 20 sampel (*True Negative*) dengan benar dan tidak membuat kesalahan sama sekali (*False Positive* = 0). Hal ini mengindikasikan bahwa model memiliki keandalan 100% dalam mengenali mahasiswa yang tidak melakukan kecurangan, sehingga tidak ada risiko salah menuduh.

Pada sisi lain, untuk 20 sampel "Curang" yang sebenarnya, model berhasil mendeteksi 12 kasus dengan benar (*True Positive*), namun gagal mendeteksi 8 kasus lainnya (*False Negative*), yang salah diklasifikasikan sebagai "Fokus". Ini menunjukkan bahwa meskipun model mampu menangkap mayoritas (60%) dari perilaku curang, masih ada 40% kasus kecurangan yang terlewat.

4.5.2 Classification Report

Selain *confusion matrix*, ditampilkan juga *classification report* yang berisi metrik evaluasi penting lainnya :

- *Precision* : proporsi prediksi benar dibanding seluruh prediksi pada kelas tertentu.
- *Recall* : proporsi prediksi benar dibanding jumlah aktual pada kelas tersebut.
- *F1-Score* : harmonisasi antara *precision* dan *recall*.

Tabel 4. 4 Classification Report

Kelas	Deskripsi	Precision	Recall	F1-Score
0	Fokus	0.71	1.00	0.83
1	Curang	1.00	0.60	0.75
<i>Accuracy</i>				0.80
<i>Macro avg</i>		0.86	0.80	0.79
<i>Weighted avg</i>		0.86	0.80	0.79

Dari tabel 4.3 hasil pengujian model pada 40 sampel data uji, model klasifikasi *Support Vector Machine* (SVM) yang dikembangkan berhasil mencapai tingkat akurasi keseluruhan sebesar 80%. Laporan tersebut menunjukkan bahwa model memiliki kinerja yang berbeda untuk masing-masing kelas. Untuk kelas "Curang" (1), model mencapai nilai presisi yang sempurna (1.00). Ini mengindikasikan bahwa setiap kali model memprediksi seorang mahasiswa melakukan kecurangan, prediksi tersebut dijamin 100% benar, sehingga tidak ada kasus mahasiswa jujur yang salah dituduh. Namun, nilai recall untuk kelas ini adalah 0.60, yang berarti model berhasil mendeteksi 60% dari semua kasus kecurangan yang sebenarnya terjadi, sementara 40% sisanya tidak terdeteksi.

Sebaliknya, untuk kelas "Fokus" (0), model mencapai nilai *recall* yang sempurna (1.00), menandakan bahwa model mampu mengidentifikasi semua mahasiswa yang benar-benar fokus tanpa ada yang terlewat. Nilai presisi untuk kelas ini adalah 0.71, yang berarti ketika model memprediksi "Fokus", prediksi tersebut benar sebanyak 71% dari total kasus.

Secara keseluruhan, hasil evaluasi ini menunjukkan bahwa model yang dikembangkan memiliki performa yang baik dan dapat diandalkan, terutama karena presisinya yang sempurna dalam mendeteksi kecurangan. Meskipun masih ada ruang untuk meningkatkan sensitivitas (*recall*), keseimbangan yang tercapai antara

tidak adanya tuduhan palsu dan kemampuan mendeteksi mayoritas kasus kecurangan menjadikan model ini efektif untuk implementasi pada sistem pengawasan ujian *real-time*.

4.5.3 Evaluasi *Face Tracking*

Evaluasi terhadap kinerja *face tracking* dilakukan untuk memastikan bahwa input data untuk tahap ekstraksi fitur memiliki kualitas yang tinggi dan stabil. Pengujian ini berfokus pada kecepatan, akurasi, dan ketangguhan dari modul *MediaPipe Face Mesh* yang diimplementasikan. Berdasarkan pengujian secara *real-time*, modul *O* menunjukkan kinerja yang sangat efisien, mampu memproses aliran video tanpa menyebabkan jeda (lag) yang signifikan, yang krusial untuk responsivitas aplikasi.

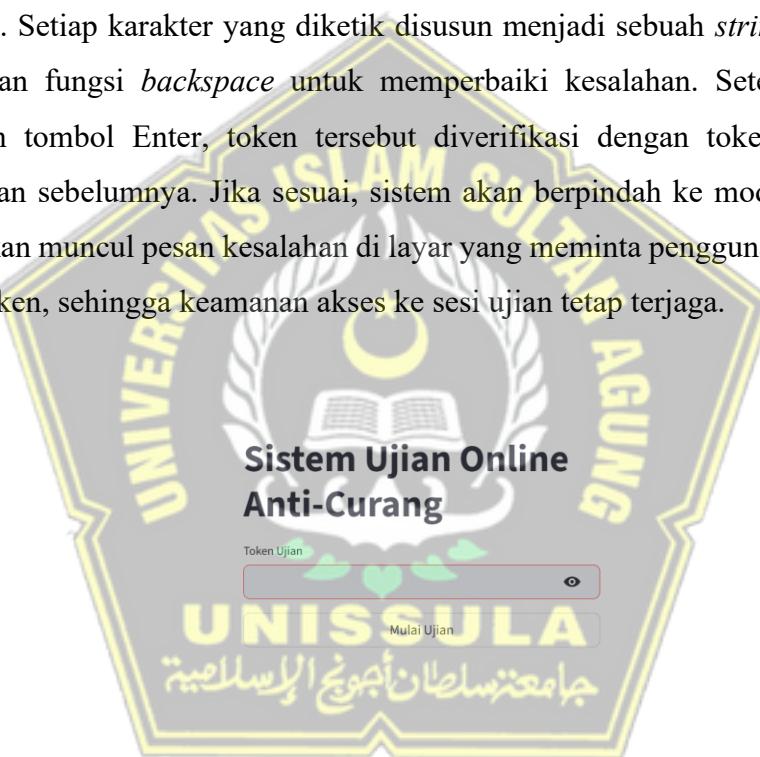
Model terbukti sangat akurat dalam mendeteksi dan mengunci posisi wajah subjek secara cepat sesaat setelah kamera diaktifkan. Pelacakan 468 *landmarks* wajah juga berjalan dengan sangat stabil; titik-titik tersebut terlihat melekat erat pada kontur wajah bahkan saat subjek melakukan gerakan wajar seperti menoleh, menunduk, atau berbicara. Selain itu, sistem pelacakan menunjukkan tingkat ketangguhan (*robustness*) yang baik terhadap perubahan pose kepala dan kondisi pencahayaan ruangan yang standar. Secara keseluruhan, kinerja modul *face tracking* dinilai sangat akurat, sehingga dapat menyediakan data koordinat *landmarks* yang konsisten dan akurat sebagai *input* bagi model SVM, yang menjadi fondasi penting bagi akurasi sistem deteksi secara keseluruhan.

4.6 Hasil Implementasi Sistem

Tahap implementasi sistem *real-time* (secara langsung) merupakan fase akhir dalam alur penelitian, di mana model SVM yang telah dilatih diintegrasikan ke dalam sebuah prototipe aplikasi yang dapat berfungsi secara langsung. Tujuan utama dari tahap ini adalah merancang sistem yang mampu menerima masukan berupa video dari kamera, memprosesnya secara berurutan pada setiap frame, serta menampilkan hasil klasifikasi dan peringatan secara langsung. Sistem ini dikembangkan dalam bentuk aplikasi tunggal (*main_app.py*) dengan pendekatan arsitektur berbasis keadaan (*state-based architecture*) untuk mengatur jalannya

interaksi dengan pengguna. Seluruh tampilan grafis (GUI) pada *prototipe* dibangun menggunakan pustaka *OpenCV*, yang digunakan untuk pembuatan jendela, penyajian citra, penyajian teks, serta pengolahan input dari perangkat pengguna.

Sistem ini dirancang dengan dua mode utama : *LOGIN* dan *EXAM*. Saat dijalankan, sistem pertama kali masuk ke mode *LOGIN*, di mana pengguna diminta melakukan proses autentifikasi. Pada tahap ini, layar akan menampilkan antarmuka untuk memasukkan token ujian. Mekanisme input token disimulasikan menyerupai kotak teks dengan cara membaca penekanan tombol keyboard secara terus-menerus. Setiap karakter yang diketik disusun menjadi sebuah *string* token, serta disediakan fungsi *backspace* untuk memperbaiki kesalahan. Setelah pengguna menekan tombol Enter, token tersebut diverifikasi dengan token yang sudah ditentukan sebelumnya. Jika sesuai, sistem akan berpindah ke mode *EXAM*; jika tidak, akan muncul pesan kesalahan di layar yang meminta pengguna memasukkan ulang token, sehingga keamanan akses ke sesi ujian tetap terjaga.



Gambar 4. 4 Tampilan antarmuka layar login sistem

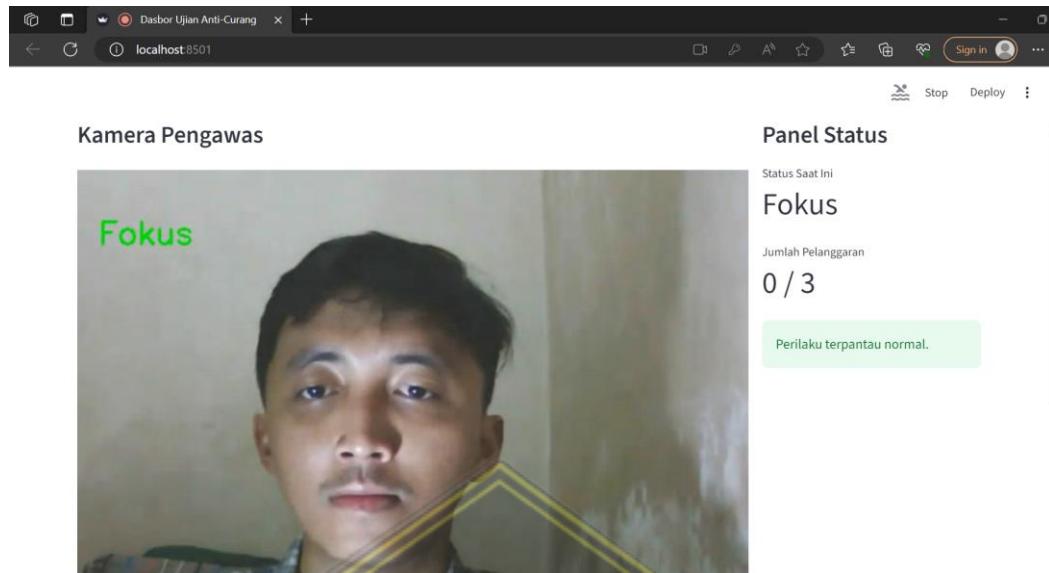
Pada gambar 4.3 menampilkan hasil pelaksanaan dari halaman masuk aplikasi yang berbasis *Streamlit*. Halaman ini berperan sebagai portal otentifikasi awal sebelum peserta dapat mengakses sesi ujian yang diawasi. Antarmuka dirancang dengan gaya minimalis dan profesional agar penggunanya dapat mengaksesnya dengan mudah. Di tengah layar, terdapat judul "Sistem Ujian Online Anti-Kecurangan". Langsung di bawahnya, terdapat suatu kolom input "Token Ujian"

yang mewajibkan peserta untuk mengetikkan kode unik yang sudah diberikan. Demi keamanan, masukan ini diatur sebagai kata sandi sehingga karakter yang dimasukkan akan disembunyikan. Pengguna selanjutnya menekan tombol "Mulai Ujian" untuk mengonfirmasi token. Apabila token valid, pengguna akan diarahkan ke halaman pemantauan utama, jika tidak, pesan kesalahan akan ditampilkan. Halaman login ini telah berhasil diterapkan sebagai langkah keamanan awal untuk memastikan hanya peserta yang terverifikasi yang dapat mengakses sistem ujian.

Setelah tahap autentikasi berhasil dilalui, sistem secara otomatis bertransisi ke keadaan *EXAM* dan langsung mengaktifkan perangkat kamera untuk menjalankan proses pemantauan ujian. Pada fase ini, aplikasi bekerja dalam sebuah *loop* pemrosesan berkelanjutan berbasis *real-time*. Setiap frame yang ditangkap dari aliran video diproses secara berurutan melalui beberapa tahap: deteksi wajah serta pemetaan 468 titik *landmarks* menggunakan pustaka *MediaPipe*, ekstraksi tiga parameter numerik dengan prosedur yang sama seperti pada proses pelatihan, kemudian klasifikasi perilaku menggunakan model SVM yang telah dimuat sebelumnya. Hasil prediksi yang diperoleh, berupa label klasifikasi (Fokus atau Curang) beserta nilai tingkat kepercayaan, divisualisasikan secara langsung dalam bentuk teks yang ditampilkan pada *frame* video yang sedang berjalan.

4.6.1 Implementasi Kelas 0 (Fokus)

Tampilan hasil implementasi Penerapan Metode *Support Vector Machine* (SVM) Berbasis *Face Tracking* untuk Deteksi Kecurangan pada Ujian Online Secara *Real-Time*. Implementasi untuk Kelas 0 (Fokus) merepresentasikan kondisi ideal di mana sistem berhasil mengidentifikasi bahwa peserta ujian sedang berkonsentrasi dan tidak melakukan pelanggaran.

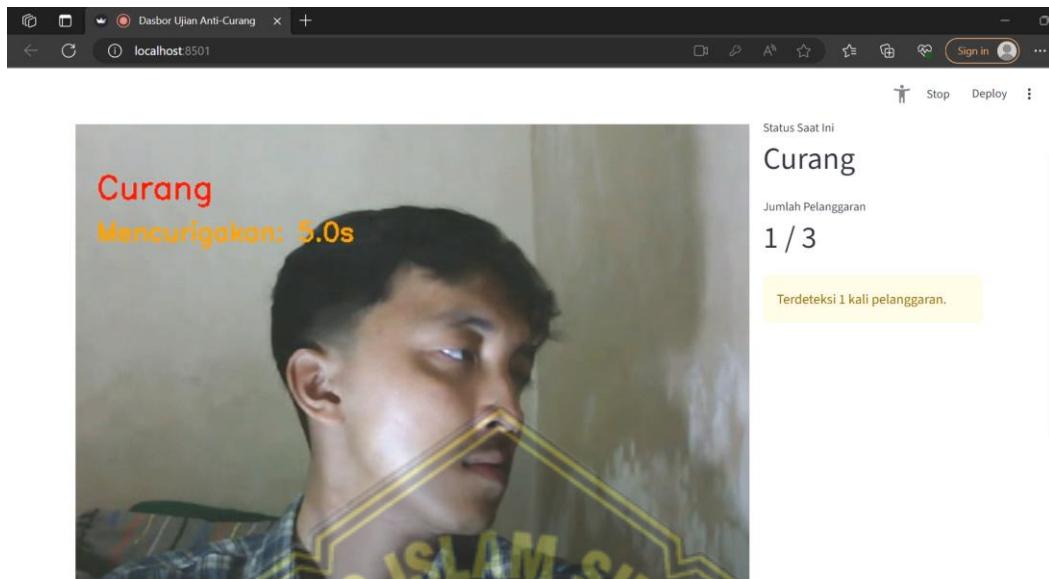


Gambar 4. 5 Implementasi hasil dari kelas 0 (fokus)

Pada gambar 4.5 menampilkan dasbor pemantauan ujian ketika sistem berfungsi dalam keadaan optimal. Antarmuka dibagi menjadi dua kolom utama:

- Kolom Kiri (Kamera Pengawas): Bagian ini menampilkan siaran video langsung dari kamera milik pengguna. Dalam bingkai video, sistem secara aktif mengevaluasi posisi wajah dan memberikan umpan balik langsung berupa teks "Fokus" berwarna hijau. Ini menunjukkan bahwa model SVM telah berhasil mengategorikan perilaku pengguna sebagai normal atau curang.
- Kolom Kanan (Panel Status): Panel ini berperan sebagai sumber informasi. Terdapat indikasi bahwa "Status Saat Ini" adalah "Fokus", yang sejalan dengan label pada video. "Jumlah Pelanggaran" menunjukkan "0 / 3", menegaskan bahwa tidak ada insiden kecurangan yang terdaftar. Pesan status di bawahnya, "Perilaku yang teramat normal", memberikan ringkasan kondisi yang mudah dipahami oleh pengawas.

4.6.2 Implementasi Kelas 1 (Curang)



Gambar 4. 6 Implementasi hasil dari kelas 1 (curang)

Pada gambar 4.6 menampilkan sistem label "Curang" berwarna merah. Di bawahnya, terdapat tulisan "Mencurigakan: 5.0s" yang menandakan bahwa logika timer telah berfungsi dan mencapai batas 5 detik, sehingga mengonfirmasi kejadian itu sebagai sebuah pelanggaran. Secara bersamaan, Panel Status di sebelah kanan mengalami pembaruan. "Status Saat Ini" berubah menjadi "Curang", "Jumlah Pelanggaran" meningkat menjadi 1 / 3, dan sebuah pesan peringatan muncul sebagai notifikasi yang jelas. Ini menunjukkan bahwa sistem tidak hanya berhasil mengidentifikasi perilaku curang dengan akurat, tetapi juga mampu menerapkan logika timer yang ditentukan dan memberikan umpan balik multimodal yang terintegrasi.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil implementasi dan evaluasi sistem yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut :

1. Penelitian ini telah berhasil merancang dan mengimplementasikan sebuah prototipe sistem deteksi kecurangan pada ujian online secara real-time. Sistem ini secara efektif mengintegrasikan teknologi deteksi wajah *MediaPipe* untuk pelacakan landmarks dan metode klasifikasi *Support Vector Machine* (SVM) untuk pengambilan keputusan. Arsitektur yang dibangun mampu memproses aliran video dari kamera, mengekstrak fitur-fitur perilaku kunci, dan mengklasifikasikannya sebagai "Fokus" atau "Curang" dengan umpan balik visual dan audio secara langsung.
2. Implementasi sistem ke dalam sebuah aplikasi *prototipe* yang fungsional telah berhasil dilakukan. Aplikasi ini tidak hanya mampu melakukan deteksi secara real-time, tetapi juga dilengkapi dengan fungsionalitas pendukung yang esensial seperti sistem otentikasi pengguna berbasis token, logika penghitungan insiden pelanggaran secara akumulatif, serta sistem peringatan langsung (visual dan audio). Hal ini menunjukkan bahwa model machine learning yang dikembangkan dapat diintegrasikan ke dalam sebuah alur kerja aplikasi yang lengkap dan logis.

5.2 Saran

Meskipun penelitian ini telah berhasil mencapai tujuannya, terdapat beberapa keterbatasan yang membuka peluang untuk pengembangan dan penelitian di masa mendatang. Berikut adalah beberapa saran yang dapat dipertimbangkan :

1. Pengembangan dataset yang lebih komprehensif

Kinerja model *machine learning* sangat bergantung pada kualitas dan kuantitas data latih. Disarankan untuk pengembangan selanjutnya agar dataset diperluas dengan melibatkan lebih banyak subjek dengan karakteristik demografis yang beragam. Selain itu, pengumpulan data sebaiknya dilakukan dalam berbagai kondisi lingkungan yang berbeda, seperti variasi pencahayaan, latar belakang, dan kualitas kamera, untuk meningkatkan kemampuan generalisasi dan ketangguhan (*robustness*) model.

2. Eksplorasi fitur yang lebih canggih

Penelitian ini menggunakan tiga fitur dasar yang efektif. Untuk meningkatkan akurasi, penelitian selanjutnya dapat mengeksplorasi fitur-fitur yang lebih canggih. Contohnya termasuk implementasi analisis arah pandangan mata (*gaze tracking*) untuk mendeteksi secara presisi ke mana pengguna melihat, estimasi pose kepala 3D untuk mendapatkan sudut rotasi kepala (*pitch, yaw, roll*) yang lebih akurat, dan deteksi objek untuk mengidentifikasi keberadaan benda-benda terlarang seperti ponsel atau buku di area yang terpantau.

3. Penggunaan model klasifikasi alternatif dan arsitektur web

Meskipun SVM memberikan hasil yang baik, disarankan untuk melakukan studi komparatif dengan algoritma klasifikasi lain seperti *Random Forest*, *Gradient Boosting*, atau Jaringan Saraf Tiruan (*Neural Networks*) yang mungkin dapat memberikan performa yang lebih tinggi. Selain itu, untuk skalabilitas, arsitektur sistem dapat dikembangkan menjadi aplikasi berbasis web penuh, di mana pemrosesan video dilakukan di sisi server, sehingga memungkinkan sistem diakses dari perangkat mana pun melalui browser tanpa memerlukan instalasi perangkat lunak dari sisi pengguna.

DAFTAR PUSTAKA

- Alamsyah, H., Cahyana, Y., & Pratama, A. R. (t.t.). *Deteksi Fake Review Menggunakan Metode Support Vector Machine dan Naïve Bayes Di Tokopedia.*
- Andi Saenong, & Rahmat, R. (2024). Implementasi Algoritma SVM Untuk Sistem Deteksi dan Pengawasan Keamanan Kendaraan di Area Parkir Menggunakan Kamera. *Simkom*, 9(2), 267–277. <https://doi.org/10.51717/simkom.v9i2.570>
- Arief, M. I., Sari, R., & Susilawati, F. E. (2025). Teknik Markerless-Based Face Tracking untuk Deteksi Dinamika Wajah dalam Pembuatan Filter Visual 2D dan 3D. *Jurnal Informasi dan Teknologi*, 2(1), 12–23.
- Bimantoro, F., Gede, I., Suta Wijaya, P., & Aohana, M. R. (t.t.). *Pendeteksian Kecurangan Ujian Melalui CCTV Menggunakan Algoritma YOLOv5* (Vol. 3).
- Frenica, A., & Soim, S. (t.t.). *Implementasi Algoritma Support Vector Machine (SVM) untuk Deteksi Banjir.* 8(2), 2023.
- Hendrastuty, N., Rahman Isnain, A., & Yanti Rahmadhani, A. (2021a). *Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine.* 6(3). <http://situs.com>
- Hendrastuty, N., Rahman Isnain, A., & Yanti Rahmadhani, A. (2021b). *Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine.* 6(3). <http://situs.com>
- Ilmiah, J., & Grafis, K. (2023). *Pemanfaatan Tools Live2D terhadap Animasi menggunakan Metode Face Tracking.* 16(1), 186–195. <https://doi.org/10.51903/pixel.v16i1.1242>
- Japit, S., Risyani, Y., Selamat, T., Bombongan, C., & Yuliana, Y. (2024). Deteksi Anomali Transaksi E-Commerce Menggunakan Support Vector Machine Berbasis Data Mining. *Jurnal Minfo Polgan*, 13(2), 1976–1980. <https://doi.org/10.33395/jmp.v13i2.14325>
- Jurnal, H., Justitia, A., Sakti, W., & Adi, D. (2021). *RANCANG BANGUN SISTEM UJIAN ONLINE BERBASIS WEB MOBILE* (Vol. 1, Nomor 3).

- Karinasaari, I., Badriyah, T., Elektronika, P., & Surabaya, N. (t.t.). DETEKSI DINI PENYAKIT IUGR (INTRA UTERINE GROWTH RETRICTION) DENGAN METODE SVM (SUPPORT VECTOR MACHINE). *Kumpulan jurnal Ilmu Komputer (KLIK)*, 07.
- Mushtofa, Z., Rusilowati, A., Sulhadi, S., Marwoto, P., & Mindiyarto, B. N. (2021). Analisis Perilaku Kecurangan Akademik Siswa dalam Pelaksanaan Ujian di Sekolah. *Jurnal Kependidikan: Jurnal Hasil Penelitian dan Kajian Kepustakaan di Bidang Pendidikan, Pengajaran dan Pembelajaran*, 7(2), 446. <https://doi.org/10.33394/jk.v7i2.3302>
- Niswa, K., & Widasari, E. R. (2017). *DETEKSI GESTUR TANGAN MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM) UNTUK LENGAN PROSTETIK BIONIK* (Vol. 1, Nomor 1). <http://j-ptiik.ub.ac.id>
- Pangestu, M. P., Wiyono, S., & Af'ida, D. I. (2024). Platform Ujian Online Berbasis Pendekripsi Gerakan Kecurangan Menggunakan Kamera. *Infomatek*, 26(1), 55–62. <https://doi.org/10.23969/infomatek.v26i1.11208>
- Pratama, A. J., Putra Kharisma, A., & Arwani, I. (2021). *Pengembangan Aplikasi Pendekripsi Kecurangan dalam Ujian Daring menggunakan Konsep Context Aware pada Platform Android* (Vol. 5, Nomor 5). <http://j-ptiik.ub.ac.id>
- Putri Iskandar, A., Muhammad Ikhsan Thohir, Ivana Lucia Kharisma, Kamdan, & Anggun Fergina. (2024a). Implementasi Deteksi Langsung Pada Sistem Ujian Online Menggunakan Algoritma Convolutional Neural Network. *Jurnal CoSciTech (Computer Science and Information Technology)*, 5(2), 483–492. <https://doi.org/10.37859/coscitech.v5i2.7270>
- Putri Iskandar, A., Muhammad Ikhsan Thohir, Ivana Lucia Kharisma, Kamdan, & Anggun Fergina. (2024b). Implementasi Deteksi Langsung Pada Sistem Ujian Online Menggunakan Algoritma Convolutional Neural Network. *Jurnal CoSciTech (Computer Science and Information Technology)*, 5(2), 483–492. <https://doi.org/10.37859/coscitech.v5i2.7270>
- Sari, K., & Arvita, Y. (2022). *Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM) Perancangan Sistem Absensi Facial Recognition Menggunakan*

- CNN dan Liveness Detector pada BPR Central Dana Mandiri* (Vol. 1, Nomor 2). JAKAKOM. <http://ejournal.unama.ac.id/index.php/jakakom>
- Septiawan, Y. (t.t.). Perbandingan Akurasi Metode Deteksi Ujaran Kebencian dalam Postingan Twitter Menggunakan Metode SVM dan Decision Trees yang Dioptimalkan dengan Adaboost. *IJCCS*, x, No.x, 1–5.
- Syarif, M., Rahman, A., 18, S. N., Id, M. M. A., & Korespondensi, P. (2023). *Jurnal Restikom : Riset Teknik Informatika dan Komputer*. 5(2), 137–147. <https://restikom.nusaputra.ac.id>
- Tantika, R. S., & Kudus, A. (2022). Penggunaan Metode Support Vector Machine Klasifikasi Multiclass pada Data Pasien Penyakit Tiroid. *Bandung Conference Series: Statistics*, 2(2), 159–166. <https://doi.org/10.29313/bcss.v2i2.3590>
- Vebriani, M., & Yustanti, W. (2024). Klasifikasi Deteksi Link Phising DANA Kaget Menggunakan Metode Support Vector Machine Berbasis Website. *Journal of Informatics and Computer Science*, 06. <https://danakagetvezridd>.
- View of Klasifikasi Tanaman Obat Herbal Menggunakan Metode Support Vector Machine*. (t.t.).
- Wicaksono, F. B., & Yamasari, Y. (2025). Pengembangan Model Pengawas Berbasis Kecerdasaan Buatan untuk Ujian Online. *Journal of Informatics and Computer Science*, 06.
- Wong, W. P., Pheh, K.-S., & Tan, C. S. (2024). Effects of a Character Strengths Intervention on Emerging Adults' Identity Formation, Self-Efficacy, and Self-Doubt. *Makara Human Behavior Studies in Asia*, 28(1). <https://doi.org/10.7454/hubs.asia.1140424>
- Yahya, R. D., Wibowo, S. A., & Vendyansyah, N. (2024). ANALISIS SENTIMEN UNTUK DETEKSI UJARAN KEBENCIAN PADA MEDIA SOSIAL TERKAIT PEMILU 2024 MENGGUNAKAN METODE SUPPORT VECTOR MACHINE. Dalam *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Nomor 3).