BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Analisis Deepfake

2.1.1.1 Pentingnya Analisis Deepfake di Platform Digital

Teknologi pengenalan wajah telah menjadi salah satu pilar penting dalam kemajuan sistem cerdasberbasis visual. Dalam beberapa tahun terakhir, implementasi sistem deteksi wajah mengalami perkembangan pesat, khususnya dengan munculnya pendekatan berbasis deep learning yang mampu meningkatkan akurasi dan kecepatan deteksi secara signifikan (Gulo et al., 2025). Deteksi wajah tidak hanya berfungsi sebagai fitur keamanan, tetapi juga telah digunakan dalam berbagai bidang seperti sistem absensi otomatis, kontrol akses, pengawasan, hingga hiburan digital. Analisis deepfake sangat penting untuk dilakukan, khususnya di lingkungan akademik dan institusi pendidikan seperti Universitas Darma Persada, karena penyebaran video manipulatif dapat menyebabkan kesalahpahaman publik, pencemaran nama baik, serta menurunkan kredibilitas institusi. Dengan adanya sistem deteksi otomatis berbasis Convolutional Neural Network (CNN), pihak kampus dapat memverifikasi keaslian video dengan cepat dan akurat.

2.1.1.2 Teknik dalam Analisis Deepfake

Analisis deepfake dilakukan dengan beberapa pendekatan:

1 Analisis visual, yaitu memeriksa ketidaksesuaian pada wajah, pencahayaan, dan gerakan bibir.

2 Analisis temporal, yang memeriksa perubahan antar-frame video.

3 Pendekatan berbasis CNN, yang menggunakan jaringan saraf tiruan untuk mengenali pola mikro pada citra wajah yang tidak alami.

2.1.1.3 Machine Learning dalam Analisis Deepfake

Pada tahun 1957 sampai 1974, popularitas dari AI mulai meningkat. Kemudahan akses pada komputer membuat manusia semakin cepat beradaptasi dengan machine learning yang dibuktikan dengan algoritma machine learning yang semakin kompleks dan pengaplikasian algoritma machine learning pada masalah-masalah tertentu. Namun yang menjadi hambatan pada masa-masa ini adalah kurangnya sumber daya komputasional untuk menyimpan maupun memproses artificial intelligence (SKRIPSI Pemodelan Convolutional Vision Transformer Pada Pemrosesan Gambar Spektogram Untuk Deteksi North Atlantic Right Whales Up-Call, n.d.).

Pada 1980an, AI kembali populer karena hadirnya "Algorithmic Toolkit" dan pendanaan yang dilakukan investor. Pada masa ini Deep learning diperkenalkan oleh John Hopfield dan David Rumelhart dimana deep learning merupakan teknik agar mesin bisa belajar dari pengalaman. Expert system juga mulai masuk ke ranah AI dibawa oleh Edward Feigenbaum. Expert system meniru bagaimana proses pengambilan keputusan dari seorang pakar. Program akan mendapatkan informasi tentang bagaimana cara pakar menangani atau menjawab masalah-masalah tertentu dan program tersebut dapat menggunakan informasi tersebut untuk menjawab pertanyaan-pertanyaan yang diajukan oleh non-pakar ke

program(SKRIPSI Pemodelan Convolutional Vision Transformer Pada Pemrosesan Gambar Spektogram Untuk Deteksi North Atlantic Right Whales Up-Call, n.d.).

AI mulai redup pada tahun 1990 sampai 2000an karena berkurangnya investasi dari pemerintah pada AI, serta turunnya minat publik. Milestones dari perkembangan AI yang mulai tercapai juga menjadi salah satu faktor yang

me

\mpengaruhi redupnya AI, seperti Gary Kasparov yang dikalahkan dalam permainan catur oleh program IBM bernama Deep Blue (SKRIPSI Pemodelan Convolutional Vision Transformer Pada Pemrosesan Gambar Spektogram Untuk Deteksi North Atlantic Right Whales Up-Call, n.d.)

2.1.2 Data Mining dalam Analisis Deepfake

Data Mining digunakan untuk menemukan pola tersembunyi dalam data besar. Dalam konteks deteksi deepfake, proses data mining melibatkan:

1 Ekstraksi data video menjadi frame (gambar per detik).

2 Pembersihan dan seleksi data agar hanya wajah yang relevan yang digunakan.

3 Transformasi data menjadi format numerik yang bisa diproses oleh CNN.

Hasil dari data mining ini menjadi *input dataset* untuk melatih model CNN agar mampu mengidentifikasi perbedaan halus antara video asli dan hasil manipulasi.

2.1.3 Prediksi dalam Analisis Deepfake

Prediksi dalam konteks ini adalah hasil klasifikasi sistem terhadap suatu video: apakah termasuk kategori real (asli) atau fake (manipulatif).

Model CNN melakukan proses prediksi berdasarkan fitur spasial dan temporal yang telah dipelajari selama pelatihan. Semakin banyak data yang digunakan, semakin tinggi kemampuan model dalam membuat prediksi akurat.

2.1.4 Metode CNN dalam Analisis Deepfake

Convolutional Neural Network (CNN) merupakan algoritma Deep Learning yang paling banyak digunakan untuk analisis citra dan video. CNN terdiri dari beberapa lapisan:

1 Convolution Layer – mengekstraksi fitur dari gambar menggunakan filter.

2 Pooling Layer – mengurangi dimensi data agar efisien.

3 Fully Connected Layer – melakukan klasifikasi berdasarkan fitur yang diperoleh.

Dalam deteksi deepfake, CNN mampu mendeteksi ketidaksesuaian pola wajah, bayangan, atau tekstur kulit yang tidak alami akibat manipulasi.

2.1.5 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu metode Deep Learning yang banyak digunakan dalam pengolahan citra digital karena kemampuannya dalam mengekstraksi fitur visual secara otomatis. CNN bekerja dengan memanfaatkan operasi konvolusi untuk mengenali pola penting pada citra,

seperti tepi, tekstur, dan bentuk objek.

Arsitektur CNN terdiri dari beberapa lapisan utama, yaitu *convolution layer*, *pooling layer*, dan *fully connected layer*. Convolution layer berfungsi untuk mengekstraksi fitur visual menggunakan filter tertentu, sedangkan pooling layer digunakan untuk mengurangi dimensi data agar proses komputasi menjadi lebih efisien. Fully connected layer kemudian digunakan untuk melakukan proses klasifikasi berdasarkan fitur yang telah dipelajari.

Banyak studi dalam literatur menyebutkan bahwa sistem deteksi wajah berbasis CNN memiliki ketahanan tinggi terhadap gangguan visual seperti latar belakang yang kompleks dan perubahan pencahayaan ekstrem (Anwar, 2025). Ini membedakan CNN dari pendekatan klasik yang sering gagal ketika menghadapi citra dengan kualitas buruk atau wajah yang tidak menghadap langsung ke kamera.

Dalam penelitian ini, CNN digunakan untuk melakukan klasifikasi citra wajah dengan memanfaatkan dataset sebanyak **2.408 gambar**, yang terdiri dari **1.204 gambar asli (real)** dan **1.204 gambar manipulatif (fake)**. Dataset ini digunakan untuk melatih model CNN agar mampu membedakan karakteristik visual antara citra asli dan citra hasil manipulasi. Pendekatan berbasis citra ini bertujuan untuk mengidentifikasi artefak visual yang umumnya muncul pada hasil deepfake.

2.1.6 Transformer dalam Analisis Video Deepfake

Transformer merupakan arsitektur Deep Learning yang awalnya dikembangkan untuk pemrosesan data sekuens, dan kemudian diadaptasi untuk

analisis citra dan video. Berbeda dengan CNN yang berfokus pada fitur spasial, Transformer menggunakan mekanisme *self-attention* untuk memahami hubungan antar elemen dalam suatu urutan data secara global.

Dalam tugas pengenalan ekspresi wajah, CNN memiliki keunggulan dalam mengekstraksi fitur penting dari gambar secara otomatis, membuatnya sangat efektif dalam klasifikasi visual. Meskipun demikian, CNN memiliki keterbatasan, seperti sensitivitas terhadap pencahayaan, sudut pengambilan gambar, serta kesulitan dalam mendeteksi ekspresi mikro atau micro-expressions. Oleh karena itu, penelitian beralih ke pendekatan yang lebih canggih, salah satunya adalah Vision Transformer (ViT). Berbeda dengan CNN, ViT memanfaatkan attention mechanism yang memungkinkan model untuk menangkap hubungan spasial antar piksel secara lebih kontekstual melalui pemrosesan gambar dalam bentuk patch. Pendekatan ini telah terbukti mampu mengenali pola visual dengan tingkat akurasi yang tinggi (Muhammad Fakhri Fadhlurrahman et al., 2025)

Dalam analisis video deepfake, Transformer memiliki keunggulan dalam menangkap dependensi temporal antar frame video(Shafa Meira Wahyono, 2025). Kemampuan ini memungkinkan model untuk mendeteksi ketidakkonsistenan gerakan wajah, sinkronisasi ekspresi, serta perubahan visual antar frame yang sering menjadi indikator manipulasi video.

Pada penelitian ini, Transformer diterapkan pada dataset video yang terdiri dari **70 video**, yaitu **35 video asli (real)** dan **35 video deepfake (fake)**. Dataset video ini digunakan untuk melatih model Transformer agar mampu mempelajari

pola temporal dan spasial secara bersamaan, sehingga sistem dapat membedakan video asli dan video hasil manipulasi secara lebih akurat.

2.1.7 CRISP-DM

Model CRISP-DM (Cross Industry Standard Process for Data Mining) digunakan sebagai kerangka penelitian dalam pengolahan data deepfake. Tahapannya meliputi:

1 Business Understanding – memahami tujuan penelitian, yaitu mendeteksi video deepfake di lingkungan kampus.

2 Data Understanding – mengumpulkan dan menganalisis dataset video.

3 Data Preparation – melakukan preprocessing seperti cropping wajah, normalisasi, dan augmentasi.

4 Modeling – menerapkan CNN untuk mendeteksi video deepfake.

5 Evaluation – mengukur performa model.

6 Deployment – mengimplementasikan sistem dalam bentuk website.

2.1.1 Pemodelan UML

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan untuk membantu proses pengembangan perangkat lunak, terutama dalam tahap analisis dan desain sistem. UML memungkinkan pengembang untuk merepresentasikan sistem secara visual melalui berbagai diagram yang menjelaskan struktur, hubungan, dan perilaku komponen sistem. Dengan UML, komunikasi antara pengembang, perancang, dan pemangku kepentingan menjadi lebih efektif karena setiap elemen sistem divisualisasikan dengan jelas.

Menurut (Zulianda Syahputra, 2015) UML singkatan dan Unified Modeling Language yang berarti bahasa pemodelan standar. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dan sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan terhadap sistem yang kita buat? Dan sebagainya dapat dijawab dengan UML.

2.1.7.1 Use Case Diagram

Use Case Diagram merupakan salah satu jenis diagram dalam Unified Modeling Language (UML) yang berfungsi untuk menggambarkan interaksi antara aktor (pengguna) dengan sistem yang dikembangkan. Diagram ini menjelaskan fungsi-fungsi utama yang dapat dilakukan oleh pengguna terhadap sistem, serta batasan tanggung jawab antara pengguna dan sistem.

Menurut Booch et al. (2023), use case diagram adalah alat bantu utama dalam tahap analisis kebutuhan karena memvisualisasikan bagaimana sistem digunakan oleh berbagai jenis aktor untuk mencapai tujuan tertentu. Diagram ini membantu pengembang dalam mengidentifikasi fitur fungsional, hubungan antaraktor, dan prioritas proses yang akan diimplementasikan.

Dalam penelitian ini, use case diagram digunakan untuk memodelkan sistem deteksi video deepfake berbasis web yang menerapkan algoritma Convolutional

Neural Network (CNN). Sistem ini dirancang agar dapat digunakan oleh beberapa pihak di lingkungan Universitas Darma Persada, antara lain admin, mahasiswa, dan dosen.

Berikut adalah komonen yang terdapat pada Use Case Diagram :

a. Actor

1 Mewakili setiap pengguna atau sistem eksternal yang berinteraksi langsung dengan sistem utama.

2 Berfungsi sebagai titik awal interaksi, yaitu siapa yang menggunakan sistem dan kapan fungsi dijalankan.

3 Contoh: User, Admin, Petugas, Sistem Pembayaran, API eksternal.

b. Use Case

1 Menunjukkan fungsi atau layanan yang disediakan oleh sistem.

2 Menggambarkan bagaimana sistem merespons permintaan aktor dan memenuhi kebutuhan pengguna.

3 Contoh: Login, Daftar Akun, Pinjam Buku, Lihat Data, Cetak Laporan.

c. Association

1. Menggambarkan hubungan langsung antara actor dan use case.
2. Menunjukkan interaksi atau komunikasi yang terjadi antara pengguna dan sistem.

3. Biasanya digambarkan dengan garis penghubung dari actor ke use case.

d. Generalization / Inheritance (Pewarisan)

1 Menunjukkan hubungan hierarki antara elemen yang lebih umum dan yang lebih spesifik.

2 Elemen turunan mewarisi sifat atau perilaku dari elemen induknya.

3 Contoh: Actor User bisa diturunkan menjadi Admin dan Member.

e. Dependency

1 Menggambarkan ketergantungan antar elemen dalam sistem.

2 Artinya, satu elemen bergantung pada elemen lain untuk bisa berjalan dengan baik.

3 Biasanya digunakan untuk menunjukkan urutan logis antar proses.

f. Include

1 Menunjukkan bahwa satu use case selalu menyertakan use case lain sebagai bagian dari prosesnya.

2 Biasanya digunakan untuk menghindari pengulangan langkah yang sama di beberapa use case.

3 Notasi: <<include>>

4 Contoh: “Proses Checkout” selalu menyertakan “Validasi Pembayaran”.

g. Extend

1 Menunjukkan bahwa sebuah use case dapat diperluas dengan fungsionalitas tambahan dalam kondisi tertentu.

2 Bersifat opsional, hanya dijalankan jika kondisi khusus terpenuhi.

3 Notasi: <<extend>>

4 Contoh: “Login” bisa diperluas dengan “Tampilkan Peringatan Gagal Login” jika password salah.

h. Communicates

1 Menunjukkan pertukaran informasi atau interaksi antara dua elemen, seperti antara actor dengan sistem atau antar actor.

2 Biasanya digunakan untuk menjelaskan alur komunikasi dalam proses tertentu.

3 Notasi: <<communicates>>

2.1.7.2 Activity Diagram

Menggambarkan alur aktivitas mulai dari unggah video, proses deteksi oleh model CNN, hingga hasil klasifikasi.

Berikut adalah komponen yang ada pada Activity Diagram :

a. Start State

1 Menunjukkan keadaan awal dalam diagram aktivitas.

2 Menjadi titik awal proses atau alur kerja dalam sistem dimulai.

3 Biasanya digambarkan sebagai lingkaran hitam kecil di awal diagram.

b. State

1 Menggambarkan kondisi atau situasi tertentu yang dialami oleh suatu objek dalam sistem.

2 State bisa berubah karena adanya aksi atau peristiwa tertentu.

3 Contoh: “Menunggu Persetujuan”, “Sedang Diproses”, “Selesai”.

c. Activity

1 Menunjukkan aktivitas atau tindakan yang dilakukan oleh sistem atau pengguna.

2 Biasanya merupakan langkah-langkah dalam suatu proses bisnis.

3 Digambarkan dengan persegi panjang sudut tumpul.

4 Contoh: “Memasukkan Data”, “Memvalidasi Input”, “Menyimpan ke Database”.

d. State Transition

1 Menunjukkan perpindahan dari satu state ke state lainnya.

2 Terjadi karena adanya kondisi, aksi, atau kejadian tertentu.

3 Digambarkan dengan panah dari satu state ke state lain.

4 Contoh: Dari state “Menunggu” → ke “Diproses” setelah tombol “Setujui” ditekan.

e. Transition to Self

1 Menggambarkan perubahan atau transisi dari satu state ke state yang sama.

2 Artinya sistem tetap dalam kondisi yang sama, tetapi terjadi aksi

tertentu di dalamnya.

3 Digambarkan dengan panah melingkar kembali ke state asalnya.

4 Contoh: State “Menunggu Konfirmasi” terus memperbarui status hingga disetujui.

f. Horizontal Synchronization

1 Menunjukkan proses atau aktivitas yang berjalan bersamaan secara sejajar (horizontal).

2 Dipakai untuk mengkoordinasikan beberapa alur aktivitas yang berjalan paralel di arah horizontal.

3 Biasanya digambarkan dengan garis tebal horizontal.

4 Contoh: Dua aktivitas dijalankan bersamaan di jalur berbeda tetapi waktunya sejajar.

g. Vertical Synchronization

1 Menggambarkan sinkronisasi aktivitas yang berjalan paralel secara vertikal.

2 Digunakan untuk memastikan bahwa semua aktivitas paralel selesai terlebih dahulu sebelum melanjutkan ke langkah berikutnya.

3 Biasanya digambarkan dengan garis tebal vertikal.

4 Contoh: Setelah beberapa proses selesai secara paralel, sistem melanjutkan ke satu langkah akhir bersama.

h. Decision

1 Menunjukkan titik percabangan alur aktivitas berdasarkan kondisi tertentu.

2 Menggambarkan pilihan “ya/tidak” atau “benar/salah” yang menentukan jalur berikutnya.

3 Biasanya digambarkan dengan belah ketupat (diamond).

4 Contoh: Jika login berhasil → lanjut ke halaman utama, jika gagal → tampilkan pesan error.

i. End State

1 Menunjukkan titik akhir dari alur aktivitas atau proses dalam sistem.

2 Digambarkan dengan lingkaran hitam di dalam lingkaran putih.

3 Artinya proses telah selesai atau berhenti di titik tersebut.

2.1.7.3 Sequence Diagram

Sequence Diagram merupakan salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan alur interaksi dan urutan pesan (message) yang terjadi antara objek atau komponen di dalam sistem dalam suatu skenario tertentu. Diagram ini menjelaskan bagaimana suatu proses atau fungsi dijalankan secara berurutan dari awal hingga menghasilkan keluaran sesuai dengan kebutuhan pengguna.

Menurut Fowler (2023), sequence diagram berperan penting dalam mendeskripsikan dinamika sistem dan menjadi panduan utama bagi pengembang perangkat lunak dalam memahami urutan komunikasi antarobjek secara kronologis. Diagram ini menunjukkan bagaimana objek berinteraksi melalui pesan (messages) yang dikirimkan, waktu eksekusi setiap langkah, serta dependensi antarproses dalam sistem.

Sedangkan menurut Deshmukh & Kumar (2022), sequence diagram memberikan manfaat besar dalam memastikan bahwa proses komunikasi antar komponen sistem sesuai dengan logika bisnis yang diinginkan. Diagram ini membantu pengembang mendeteksi potensi kesalahan desain sejak tahap awal karena mampu menunjukkan alur input–proses–output dengan jelas.

Dalam konteks penelitian ini, Sequence Diagram digunakan untuk menggambarkan interaksi antara pengguna (admin, dosen, mahasiswa) dengan sistem pendeteksi video deepfake berbasis CNN. Diagram ini memperlihatkan tahapan-tahapan utama saat sistem menerima video, memproses data menggunakan model CNN, dan menampilkan hasil klasifikasi kepada pengguna.

2.1.8 Database dan DBMS

Database Management System (DBMS) adalah perangkat lunak yang digunakan untuk membuat, mengelola, memelihara, dan memanipulasi database. DBMS bertugas sebagai penghubung antara pengguna dan basis data agar proses pengolahan data dapat dilakukan secara efisien, aman, dan terkontrol.

Menurut Silberschatz, Korth, & Sudarshan (2020), DBMS adalah sistem perangkat lunak yang menyediakan cara yang sistematis dan terorganisir untuk menyimpan serta mengambil informasi dari basis data. DBMS juga memastikan konsistensi, integritas, keamanan, dan ketersediaan data dalam sistem.\

Beberapa fungsi utama DBMS antara lain:

1 Data Definition – mendefinisikan struktur data dan relasi antar tabel.

2 Data Manipulation – melakukan operasi insert, update, delete, dan select terhadap data.

3 Data Security – mengatur hak akses pengguna agar tidak terjadi penyalahgunaan data.

4 Data Integrity – menjaga konsistensi antar data agar tidak terjadi duplikasi atau konflik.

5 Data Recovery – memulihkan data jika terjadi kegagalan sistem atau kehilangan data.

2.1.9 Software dan Pemrograman Terkait

Perangkat lunak dan bahasa pemrograman merupakan komponen penting dalam proses pengembangan sistem berbasis kecerdasan buatan. Dalam penelitian ini, perangkat lunak yang digunakan harus mendukung proses pengolahan video, pelatihan model CNN, serta integrasi dengan antarmuka web dan database.

Menurut Pressman (2020), pemilihan perangkat lunak dan bahasa pemrograman yang tepat sangat menentukan keberhasilan suatu proyek perangkat lunak karena akan memengaruhi kinerja sistem, efisiensi pemrosesan, dan

kemudahan pengembangan. Dalam konteks deteksi video deepfake, software yang dipilih harus mampu menangani data multimedia dalam jumlah besar serta mendukung framework machine learning modern.

Penelitian ini memanfaatkan kombinasi bahasa pemrograman Python, framework deep learning TensorFlow/Keras, library OpenCV, serta MySQL untuk manajemen basis data. Kombinasi ini memungkinkan sistem bekerja secara efisien dari tahap input video hingga menghasilkan hasil deteksi yang dapat diakses melalui web.

2.1.9.1 Bahasa Pemrograman Python dan Library Terkait

Python merupakan bahasa pemrograman tingkat tinggi yang populer di bidang kecerdasan buatan, data science, dan pengembangan web. Python memiliki sintaks yang sederhana dan mudah dibaca, serta didukung oleh berbagai pustaka (library) yang sangat kuat untuk pengolahan data, citra, dan video.

Menurut Lutz (2021), Python menjadi bahasa utama dalam pengembangan sistem berbasis AI karena memiliki komunitas besar, pustaka komprehensif, serta integrasi lintas platform yang baik. Python mendukung berbagai framework deep learning seperti TensorFlow, Keras, dan PyTorch yang sangat efisien untuk pelatihan model CNN.

Beberapa pustaka Python yang digunakan dalam penelitian ini antara lain:

1 NumPy – digunakan untuk operasi numerik dan manipulasi array multidimensi yang dibutuhkan dalam proses pelatihan model CNN.

2 Pandas – digunakan untuk mengelola dan memproses data hasil deteksi

atau metadata video.

3 Matplotlib & Seaborn – digunakan untuk visualisasi hasil pelatihan model seperti grafik akurasi dan loss.

4 Flask/Django – framework web Python untuk membangun antarmuka pengguna yang terhubung dengan model CNN dan database MySQL.

Dengan kombinasi pustaka tersebut, sistem pendeteksi deepfake dapat mengimplementasikan model CNN secara efisien, melakukan proses training dan testing, serta menampilkan hasil klasifikasi melalui web interface.

2.1.10 Kesimpulan

Dari penjelasan di atas dapat disimpulkan bahwa pemilihan perangkat lunak dan bahasa pemrograman memiliki peran vital dalam pengembangan sistem pendeteksi video deepfake berbasis CNN.

Penggunaan Python dengan dukungan pustaka TensorFlow, Keras, dan OpenCV memungkinkan penelitian ini untuk menghasilkan sistem yang efisien, fleksibel, dan akurat. Selain itu, integrasi dengan MySQL memastikan hasil deteksi tersimpan dengan baik dan dapat diakses dengan mudah oleh pengguna di lingkungan Universitas Darma Persada.