

### BAB 3

## METODOLOGI PENELITIAN

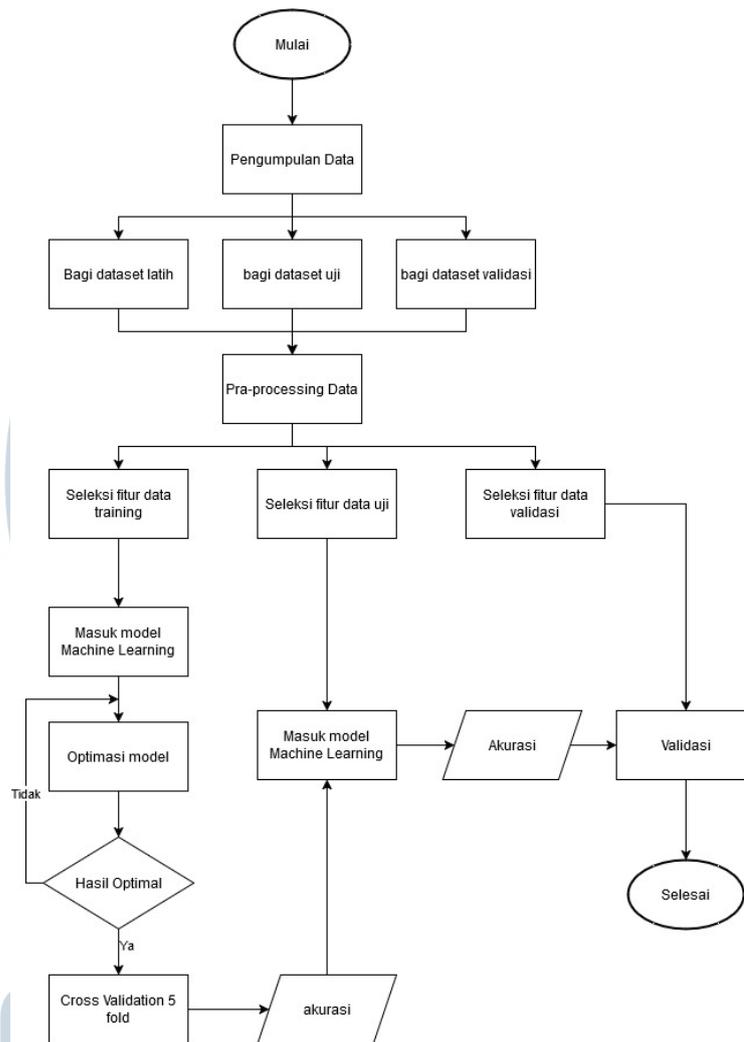
Flowchart pada Gambar 3.1 menunjukkan tahapan penelitian ini. Penelitian dimulai dengan tahap pengumpulan data, di mana data yang relevan dikumpulkan untuk mendukung penelitian. Setelah data terkumpul, data dibagi menjadi tiga bagian: *dataset latih*, *dataset uji*, dan *dataset validasi*, untuk memastikan evaluasi model yang menyeluruh.

Langkah berikutnya adalah *pra-pemrosesan data* yang melibatkan pembersihan dan persiapan data agar dapat digunakan untuk pelatihan model. Setelah itu, dilakukan *seleksi fitur* untuk setiap subset data (latih, uji, dan validasi). Seleksi fitur ini bertujuan memilih fitur-fitur yang paling relevan menggunakan metode seperti *Lasso*, *SelectKBest*, atau *PCA* untuk meningkatkan efisiensi model.

*Dataset latih* yang sudah diproses kemudian dimasukkan ke dalam model *machine learning* untuk proses pelatihan. Setelah proses pelatihan selesai, model akan menghasilkan nilai akurasi awal, dan model yang telah dilatih akan disimpan (*save model*). Model yang telah disimpan ini kemudian digunakan untuk mengevaluasi *dataset uji* guna mendapatkan akurasi pada data uji. Akurasi dari data uji kemudian disimpan kembali untuk digunakan pada tahap berikutnya.

Model yang telah disimpan dengan *dataset uji* kemudian digunakan untuk memvalidasi performa model dengan *dataset validasi*. Jika hasil validasi menunjukkan akurasi yang optimal, penelitian dinyatakan selesai. Jika tidak, siklus optimasi dan evaluasi ulang dilakukan hingga hasil yang diinginkan tercapai.

UIN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.1. Flowchart penelitian

### 3.1 Studi Literatur

Tahap studi literatur bertujuan untuk memperdalam pemahaman tentang topik penelitian ini dengan mempelajari berbagai sumber yang relevan. Ini termasuk jurnal, artikel ilmiah, dan publikasi lainnya yang akan membantu membangun dasar yang kuat bagi penelitian. Fokus utama dari studi ini adalah mempelajari berbagai metode seleksi fitur, seperti LASSO (*Least Absolute Shrinkage and Selection Operator*), *SelectKBest*, dan PCA (*Principal Component Analysis*), yang sudah terbukti efektif dalam mendeteksi *deepfake*. Dengan memahami metode-metode ini secara mendalam, peneliti bisa menentukan cara terbaik untuk mendapatkan hasil yang optimal.

Selain itu, literatur yang dipilih juga akan mencakup penelitian terkait

dengan teknologi *deepfake* dan algoritma pendeteksiannya. Literatur ini juga akan membantu peneliti memahami karakteristik data yang digunakan, teknik *machine learning* yang relevan, serta masalah yang mungkin muncul selama proses penelitian. Dengan begitu, studi literatur ini akan memberikan dasar yang kuat untuk menentukan metode deteksi dan seleksi fitur yang tepat dalam penelitian ini.

### 3.2 Akuisisi Data

Akuisisi data merupakan langkah penting dalam penelitian ini, di mana data dikumpulkan dari sumber terpercaya, yaitu *dataset Deepfake and Real Images* yang diunduh dari platform *Kaggle* pada tanggal 10 November 2024. *Dataset* ini berisi total 190.000 gambar dengan resolusi tetap sebesar 255x255 piksel dan tiga saluran warna (RGB), serta format *.jpg*. *Dataset* ini mencakup dua kategori utama: "real" dan "fake." Pelabelan dilakukan secara manual, dengan memberikan label numerik untuk setiap kategori, yaitu "0" untuk gambar asli (real) dan "1" untuk gambar deepfake (fake).

*Dataset* ini telah dibagi ke dalam tiga *subset* utama, yaitu *train*, *test*, dan *validation*, yang disediakan langsung dalam *dataset*. *Subset* ini mempermudah proses pelatihan, pengujian, dan validasi model, sehingga dapat menghemat waktu dalam tahap pengelolaan data. *Subset* pelatihan digunakan untuk melatih model, *subset* pengujian digunakan untuk mengevaluasi kinerja model, sementara *subset* validasi memastikan model mampu bekerja dengan baik pada data baru dan tidak mengalami *overfitting*.

### 3.3 Pra-Pemrosesan Data

Pra-pemrosesan data merupakan tahap penting untuk memastikan kualitas data sebelum digunakan dalam proses pelatihan model. Proses ini meliputi beberapa langkah utama seperti penghapusan duplikat, penanganan data hilang, normalisasi, serta penyesuaian ukuran gambar agar seragam. Gambar dari dataset dimuat dan diubah menjadi array numerik menggunakan pustaka *Pillow* dan *NumPy*. Setiap gambar diubah ukurannya menjadi 128x128 piksel dan diubah menjadi array 1D untuk memudahkan pemrosesan lebih lanjut. Gambar yang gagal diproses akan dicatat agar dapat diidentifikasi dan diperbaiki di kemudian hari.

Data yang telah dimuat dikonversi menjadi *DataFrame* untuk memanfaatkan fungsi bawaan dari pustaka *pandas*. Duplikat dihapus menggunakan fungsi

drop\_duplicates. Jika ditemukan data yang hilang, dilakukan imputasi menggunakan nilai median dari masing-masing fitur dengan bantuan *SimpleImputer* dari pustaka *scikit-learn*. Setelah data bersih, dilakukan normalisasi menggunakan metode *Min-Max Scaling* agar setiap fitur memiliki rentang nilai antara 0 dan 1. Proses ini penting untuk memastikan model dapat belajar dengan lebih efektif dan stabil.

Data yang telah diproses kemudian disimpan dalam format *.pkl* secara terpisah untuk masing-masing kelas data. Gambar asli (real) dan gambar palsu (fake) disimpan di file yang berbeda untuk memudahkan pengelolaan dan pengolahan data selanjutnya. Setiap file *.pkl* berisi data gambar yang telah dinormalisasi dan label yang sesuai, di mana label 0 digunakan untuk gambar asli dan label 1 untuk gambar palsu.

Sebagai contoh, gambar dari folder *Validation\_Real* diproses dan disimpan ke file *Validation\_Real.pkl*, sementara gambar dari folder *Validation\_Fake* disimpan ke file *Validation\_Fake.pkl*. Proses ini diterapkan pula pada subset data *training* dan *testing*. Setiap file *.pkl* akan diverifikasi untuk memastikan data telah disimpan dengan benar.

Dengan langkah-langkah ini, data yang digunakan untuk pelatihan, validasi, dan pengujian telah dipastikan bersih, terstruktur, dan siap mendukung proses pengembangan model yang lebih akurat.

### 3.4 Seleksi Fitur

Pada tahap seleksi fitur, beberapa metode digunakan untuk memilih fitur-fitur yang paling relevan dalam mendeteksi konten *deepfake*. *SelectKBest* dipilih untuk menyeleksi fitur terbaik berdasarkan skor statistik tertentu, sehingga hanya fitur yang paling penting yang akan dimasukkan ke dalam model.

Selain itu, Lasso (*Least Absolute Shrinkage and Selection Operator*) digunakan dalam seleksi fitur untuk memilih fitur-fitur yang paling relevan dalam model prediktif, sekaligus menghilangkan fitur-fitur yang tidak penting atau redundan. PCA (*Principal Component Analysis*) fitur yang memiliki variansi tertinggi, sehingga mengurangi kompleksitas data tanpa mengorbankan informasi penting.

Parameter yang digunakan dalam metode seleksi fitur adalah sebagai berikut:

1. *SelectKBest*:  $k=1000$ ,  $\text{scoring} = \text{chi2}$

2. PCA:  $n\_component = 0.95$

3. LASSO:  $alpha = 0.1$

Penggunaan parameter pada metode seleksi fitur dalam penelitian ini dilakukan secara independen, tanpa mengombinasikan metode satu dengan yang lain, untuk mengevaluasi efektivitas masing-masing teknik dalam mendeteksi konten *deepfake*. Pada metode *SelectKBest*, nilai  $k = 1000$  dipilih untuk menjaga keseimbangan antara mempertahankan informasi yang relevan dan mengurangi kompleksitas model, sementara skor *chi-square* ( $chi2$ ) digunakan karena efektif dalam mengevaluasi hubungan antara fitur dan label pada data kategorikal seperti "real" dan "fake." Untuk PCA (*Principal Component Analysis*), parameter  $n\_component=0.95$  memastikan bahwa 95% dari total variansi data terjaga, memungkinkan pengurangan dimensi tanpa kehilangan informasi penting, sehingga model dapat dilatih lebih cepat dan lebih efisien. Sedangkan pada LASSO (*Least Absolute Shrinkage and Selection Operator*), nilai  $alpha=0.1$  dipilih untuk memberikan tingkat regularisasi yang moderat, yang secara efektif menghilangkan fitur-fitur yang tidak signifikan sambil mencegah *overfitting* pada dataset berdimensi tinggi. Dengan pendekatan ini, performa setiap metode seleksi fitur dapat diukur secara individual, memberikan wawasan lebih dalam tentang keunggulan dan kelemahan masing-masing teknik dalam konteks deteksi *deepfake*.

### 3.5 Pelatihan Model dengan SVM

Setelah proses seleksi fitur, model dilatih menggunakan algoritma *Support Vector Machine* (SVM). Algoritma SVM digunakan untuk memisahkan gambar *deepfake* dan gambar asli (*real*) dengan mencari batas optimal (*hyperplane*) yang memisahkan kedua kelas secara maksimal. Algoritma ini dipilih karena kemampuannya menangani data berdimensi tinggi dan distribusi fitur yang kompleks, yang menjadi karakteristik utama dalam mendeteksi *deepfake*.

Pelatihan model dilakukan dengan menggunakan pustaka *scikit-learn*, dengan konfigurasi parameter sebagai berikut:

- $C = 1.0$ : Parameter ini mengontrol keseimbangan antara margin pemisahan yang lebar dan kesalahan klasifikasi pada data pelatihan. Nilai  $C = 1.0$  memberikan toleransi moderat untuk kesalahan klasifikasi guna menghindari *overfitting*.

- **Kernel = 'RBF'** (*Radial Basis Function*): *Kernel* ini dipilih karena kemampuannya untuk memetakan data yang tidak terpisahkan secara linear ke ruang dimensi yang lebih tinggi, sehingga pola-pola kompleks dalam data dapat diidentifikasi dengan baik.
- **Gamma = 'scale'**: Parameter ini secara otomatis menyesuaikan nilai *gamma* berdasarkan jumlah fitur dalam dataset, untuk memastikan stabilitas dan konsistensi dalam klasifikasi.

Proses pembentukan *hyperplane* pada SVM dimulai dengan menentukan bobot  $w$  dan bias  $b$  yang optimal. SVM berusaha memaksimalkan margin antara kelas dengan meminimalkan fungsi objektif dengan constraint untuk data yang tidak dapat dipisahkan secara linear, SVM menggunakan *kernel trick* dengan memetakan data ke ruang dimensi yang lebih tinggi. Dalam kasus ini, *Radial Basis Function* (RBF) digunakan untuk menangkap pola kompleks dalam data.

Pelatihan model dilakukan menggunakan metode *5-fold cross-validation*. Dalam metode ini, data pelatihan dibagi menjadi lima subset. Pada setiap iterasi, empat subset digunakan untuk melatih model, sementara satu subset digunakan untuk validasi. Proses ini diulangi sebanyak lima kali, sehingga semua subset menjadi bagian dari data validasi. Hasilnya kemudian dirata-rata untuk mengukur performa keseluruhan model. Metode ini membantu mengurangi risiko *overfitting* dan memastikan model dapat bekerja secara konsisten pada data baru.

Selain itu, pelatihan dilakukan secara bertahap:

1. Model pertama dilatih menggunakan *dataset latih* yang telah melalui proses seleksi fitur.
2. Setelah pelatihan selesai, model disimpan (*save model*) untuk mencegah pengulangan pelatihan yang memakan waktu.
3. Model yang telah dilatih kemudian dievaluasi menggunakan *dataset uji* untuk mendapatkan performa awal.
4. Jika performa sudah sesuai, model digunakan pada *dataset validasi* untuk mengukur kemampuan generalisasi.

Kombinasi parameter  $C = 1.0$ , *kernel* RBF, dan *gamma* = '*scale*' dipilih berdasarkan pertimbangan teoretis dan hasil uji coba awal. Konfigurasi ini memungkinkan model untuk mengidentifikasi pola-pola kompleks yang menjadi

ciri khas *deepfake*, seperti perbedaan tekstur kulit, artefak yang dihasilkan oleh algoritma generatif, atau inkonsistensi pencahayaan. Dengan pendekatan ini, model diharapkan mampu mendeteksi *deepfake* secara andal dengan tingkat akurasi yang tinggi.

### 3.6 Evaluasi Model

Tahap akhir dalam penelitian ini adalah evaluasi model, yang dilakukan menggunakan *dataset* berbeda dari data yang digunakan dalam pelatihan. Tujuan dari evaluasi ini adalah untuk mengukur performa model dalam mendeteksi *deepfake* secara objektif, serta untuk membandingkan hasil model sebelum dan sesudah dilakukan seleksi fitur. Proses ini menggunakan beberapa metrik evaluasi utama, yaitu *precision*, *recall*, dan *F1-score*.

Evaluasi dilakukan dengan memanfaatkan *confusion matrix*, yang memberikan gambaran menyeluruh mengenai distribusi hasil prediksi model. *Confusion matrix* terdiri dari empat elemen utama:

- **True Positive (TP):** Jumlah prediksi yang benar, di mana gambar *deepfake* diklasifikasikan sebagai *deepfake*.
- **True Negative (TN):** Jumlah prediksi yang benar, di mana gambar asli (*real*) diklasifikasikan sebagai *real*.
- **False Positive (FP):** Jumlah prediksi salah, di mana gambar asli (*real*) diklasifikasikan sebagai *deepfake*.
- **False Negative (FN):** Jumlah prediksi salah, di mana gambar *deepfake* diklasifikasikan sebagai gambar asli (*real*).

Berdasarkan *confusion matrix*, beberapa metrik evaluasi digunakan untuk menilai performa model:

#### 1. Precision

Precision mengukur proporsi prediksi positif (*deepfake*) yang benar terhadap total prediksi positif. Nilai precision yang tinggi menunjukkan bahwa model memiliki tingkat kesalahan yang rendah dalam memberikan prediksi positif yang salah.

#### 2. Recall

Recall mengukur kemampuan model untuk mendeteksi semua sampel yang

benar-benar positif (*deepfake*). Nilai recall yang tinggi menunjukkan bahwa model dapat mendeteksi mayoritas sampel positif dengan baik.

### 3. F1-Score

F1-Score adalah rata-rata harmonis antara precision dan recall, yang memberikan keseimbangan antara keduanya. Nilai F1-Score yang tinggi menunjukkan bahwa model memiliki keseimbangan yang baik antara precision dan recall, terutama dalam situasi dengan data yang tidak seimbang.

Evaluasi model dilakukan menggunakan pustaka *scikit-learn* dengan fungsi `confusion_matrix` dan `classification_report`. Fungsi-fungsi ini secara otomatis menghitung nilai *precision*, *recall*, dan *F1-score* berdasarkan hasil prediksi model.

Diharapkan, hasil evaluasi ini dapat memberikan pemahaman yang lebih jelas mengenai efektivitas metode seleksi fitur yang digunakan, serta menunjukkan kontribusi signifikan terhadap efisiensi waktu pelatihan dan akurasi deteksi *deepfake*. Dengan evaluasi yang baik, model dapat diandalkan untuk diaplikasikan pada berbagai skenario nyata dalam mendeteksi konten *deepfake*.

