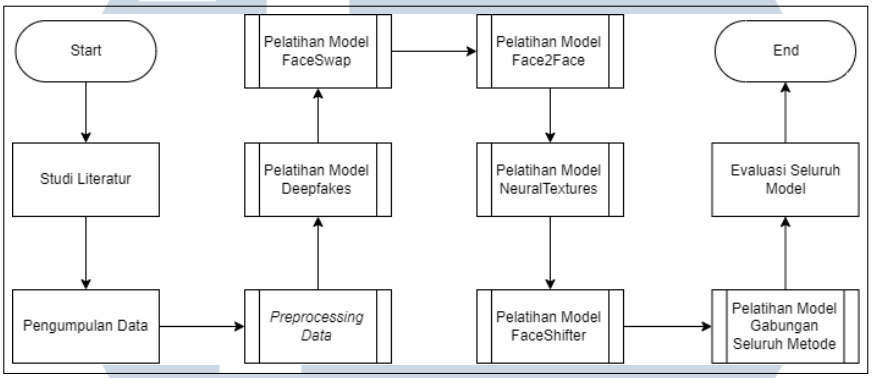
BAB 3
METODOLOGI PENELITIAN

3.1 Gambaran Umum Penelitian



Gambar 3.1. Alur Penelitian

Metodologi yang akan digunakan pada penelitian ini akan meliputi beberapa tahapan seperti pada gambar 3.1.

3.2 Studi Literatur

Studi literatur akan dilakukan sebagai langkah awal untuk memahami teori-teori dasar dan pendekatan yang telah digunakan pada penelitian-penelitian sebelumnya. Studi ini mencakup penelusuran terhadap berbagai penelitian terdahulu yang relevan, khususnya dalam bidang deteksi video *deepfake*. Fokus utama studi ini adalah untuk mengidentifikasi pendekatan, algoritma, serta *dataset* yang telah digunakan dalam penelitian sebelumnya, serta mengevaluasi kelebihan, kekurangan, dan keterbatasan penelitian-penelitian tersebut.

Literatur yang dikaji mencakup penelitian yang menggunakan arsitektur CNN, terutama *pre-trained models*, dan RNN, terutama LSTM serta penggabungan metode-metode tersebut untuk meningkatkan akurasi deteksi dan kemampuan generalisasi *model*. Selain itu, kajian juga mencakup pemanfaatan teknik augmentasi data, penggunaan fitur spasial dan temporal, serta strategi evaluasi model seperti *cross-dataset testing* yang penting dalam menilai generalisasi model deteksi video *deepfake*. Sumber literatur yang digunakan berasal dari berbagai sumber berupa jurnal, artikel, hasil prosiding, dan *website*.

3.3 Pengumpulan Data

3.3.1 Sumber Dataset

Penelitian ini menggunakan dataset publik *FaceForensics++*. Dataset ini dipilih karena secara luas digunakan dalam berbagai penelitian sebelumnya dan mencerminkan variasi teknik manipulasi wajah dalam pembuatan video *deepfake*.

FaceForensics++ terdiri dari 1000 video asli yang diambil dari YouTube. Dari 1000 video asli tersebut, dibuat 5000 video palsu dengan menerapkan lima metode manipulasi wajah yang berbeda, di mana setiap metode menghasilkan 1000 video manipulasi. Metode-metode tersebut adalah *DeepFakes*, *Face2Face*, *FaceSwap*, *FaceShifter*, dan *NeuralTextures*.

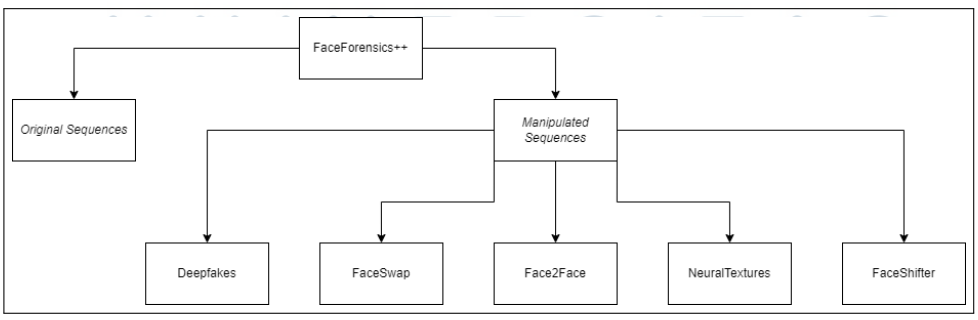
Dataset ini tersedia dalam tiga tingkat kompresi video RAW (tanpa kompresi), c23 (kompresi sedang), dan c40 (kompresi tinggi). Dalam penelitian ini, digunakan video dengan kompresi c23 dan seluruh metode manipulasi wajah.

3.3.2 Karakteristik Dataset

Dataset *FaceForensics++* memiliki karakteristik sebagai berikut:

- Terdiri dari 1.000 video asli dan 5.000 video palsu yang merupakan hasil manipulasi dari lima metode *deepfake* yang berbeda.
- Rata-rata video berdurasi sekitar 15 detik dengan *frame rate* sebesar 25 FPS.
- Resolusi video bersifat variatif dari 480p, 720p, dan 1080p.

3.3.3 Struktur *Dataset* FaceForensics++



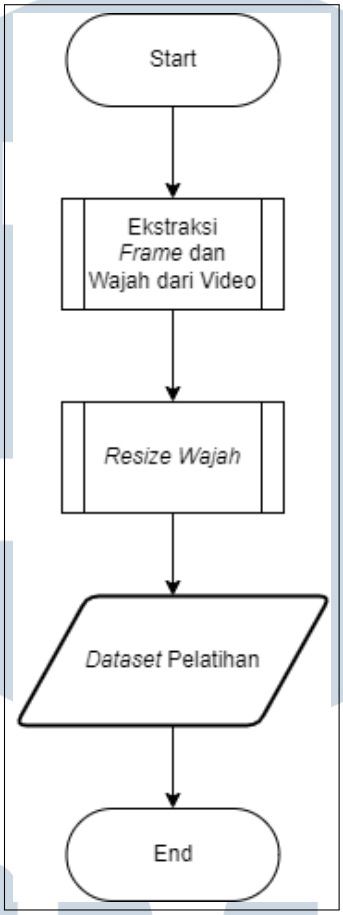
Gambar 3.2. *Subset Dataset* FaceForensics++

Dataset FaceForensics++ dikategorikan berdasarkan jenis video, yaitu video asli dan video yang telah dimanipulasi secara digital atau deepfake. Setiap kategori mencakup beberapa subset yang berasal dari metode pembuatan video yang berbeda, sebagaimana ditampilkan pada gambar 3.2.

1. **Dataset Video Asli:**
    - *Original Sequences*
2. **Dataset Video Deepfake:**
    - *Deepfakes*
    - *Face2Face*
    - *FaceSwap*
    - *NeuralTextures*
    - *FaceShifter*

Penggunaan berbagai jenis metode manipulasi dalam *dataset* ini bertujuan untuk merepresentasikan keragaman teknik yang digunakan dalam pembuatan video *deepfake*. Setiap metode memiliki karakteristik manipulasi yang berbeda, baik dari sisi teknik pembuatan, kualitas visual, maupun jenis artefak yang dihasilkan.

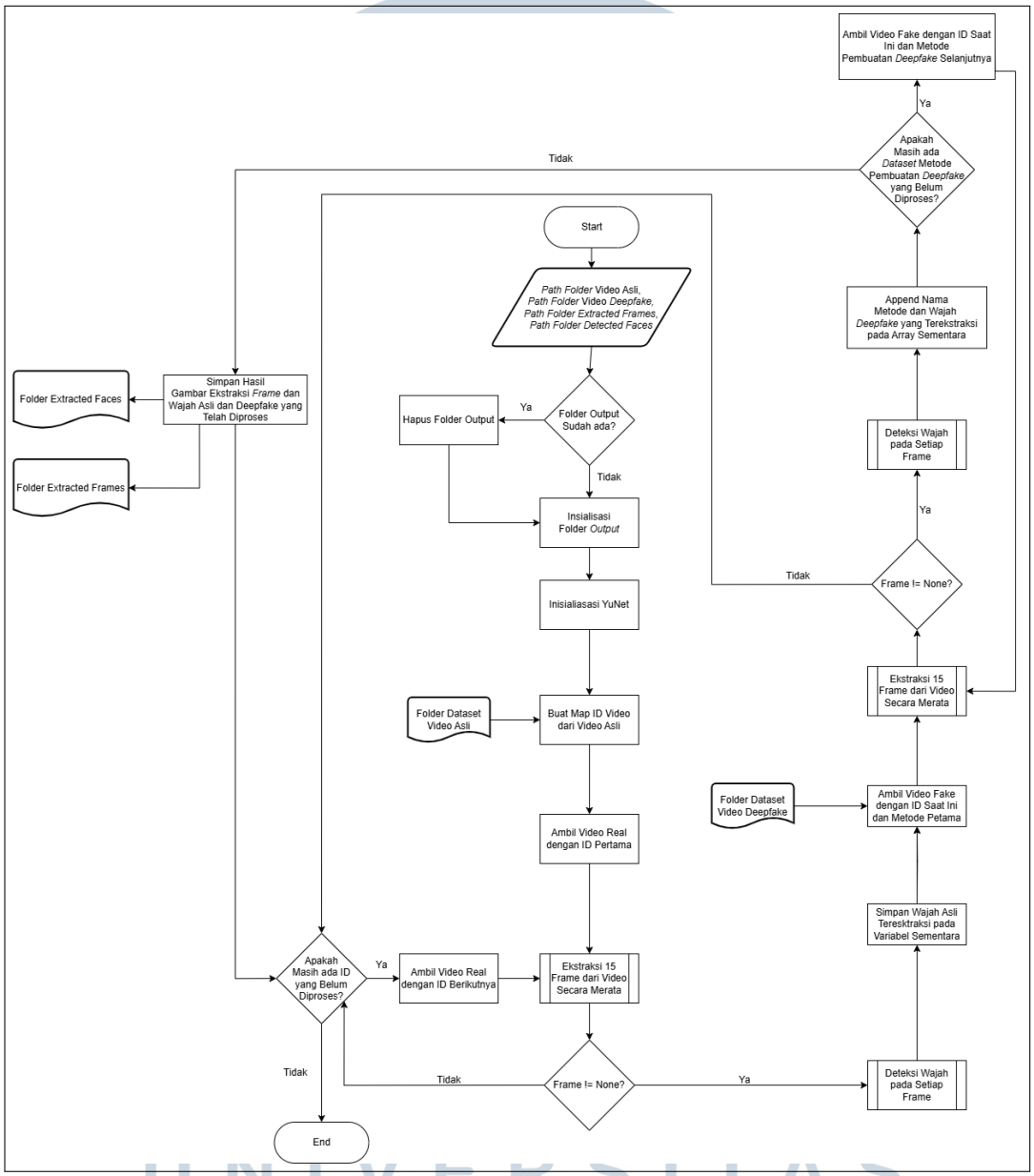
3.4 *Pre-processing Data*



Gambar 3.3. Tahapan *Preprocessing*

Gambar 3.3 menunjukkan tahapan-tahapan yang akan dilakukan pada tahap *preprocessing*.

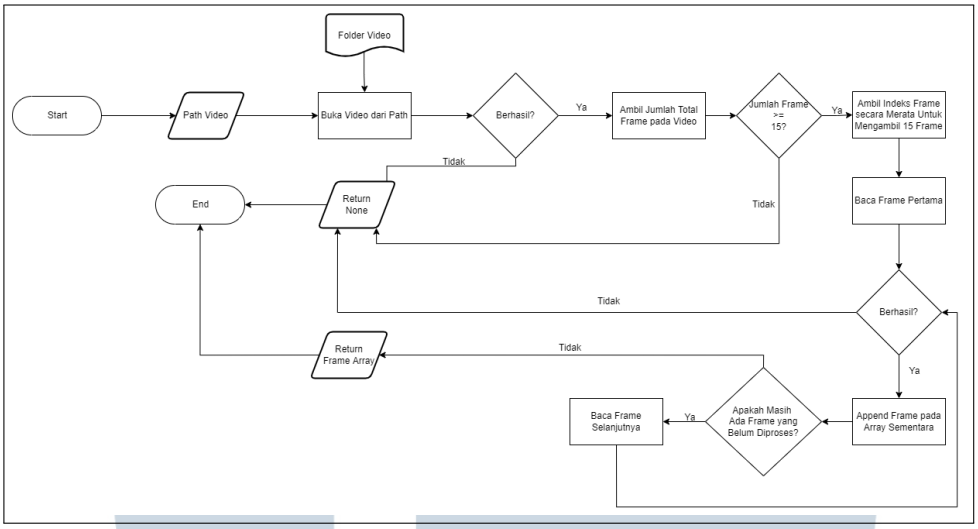
3.4.1 Ekstraksi *Frame* dan Wajah dari Video



Gambar 3.4. Ekstraksi *Frame* dan Wajah dari Video

Dikarenakan besarnya jumlah dan ukuran data video, tidak seluruh *frame* dari tiap video digunakan dalam proses pelatihan. Oleh karena itu, dilakukan serangkaian proses seleksi *frame* dan deteksi wajah seperti yang ditunjukkan pada Gambar 3.4, dengan langkah-langkah sebagai berikut.

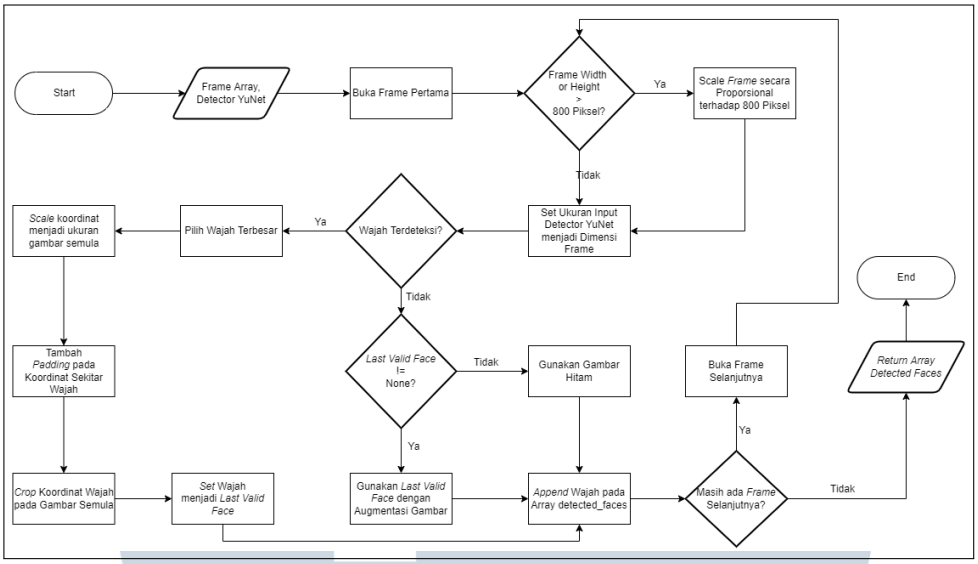
1. Ekstraksi *Frame* Secara Merata (lihat Gambar 3.5)



Gambar 3.5. *Flow* Ekstraksi 15 *Frame* dari Video Secara Merata

- Semua video digunakan tanpa memotong durasi video.
- Dari setiap video, 15 *frames* diekstraksi secara merata berdasarkan durasi video. 15 *frames* dipilih dengan pertimbangan dari segi efisiensi komputasi serta merujuk pada penelitian oleh Abidin et al. yang menunjukkan bahwa peningkatan jumlah frame dari 20 hingga 60 tidak memberikan peningkatan akurasi yang signifikan [13]. Dengan demikian, 15 *frames* dianggap cukup representatif untuk menangkap informasi temporal tanpa membebani proses pelatihan secara berlebihan.
- Apabila *frame* tidak dapat digunakan, maka seluruh video untuk semua *subset* akan diabaikan.
- Ekstraksi dilakukan tanpa mengubah *frame rate* asli video.

2. Deteksi Wajah dengan YuNet (lihat Gambar 3.6)



Gambar 3.6. *Flow* Deteksi Wajah pada Setiap *Frame*

- Setiap *frame* diproses menggunakan model deteksi wajah YuNet.
- Jika *frame* memiliki *width* atau *height* lebih dari 800 piksel, maka akan dilakukan *scaling* secara proporsional terhadap 800 piksel.
- Hasil deteksi wajah yang disimpan akan digunakan sebagai data pelatihan model.
- Akan ditambahkan *padding* di sekitar wajah untuk menangkap lebih banyak konteks visual.
- Jika deteksi wajah gagal, *fallback* yang akan digunakan adalah augmentasi dari wajah sebelumnya jika ada atau *placeholder* hitam jika tidak ada wajah dari *frame* sebelumnya.

3. Penyamaan Identitas Video antara Real dan Fake

- Identitas video ditentukan dari nama file real (misalnya ‘004.mp4‘) dan dicocokkan dengan file fake yang memiliki awalan nama sama (misalnya ‘004_*.mp4‘).
- Hanya identitas yang memiliki pasangan lengkap dari seluruh metode pembuatan video *deepfake* yang diproses.
- Baik video real maupun semua pasangan video *deepfake* akan melalui proses ekstraksi dan deteksi wajah.

4. Seleksi Identitas Video Valid

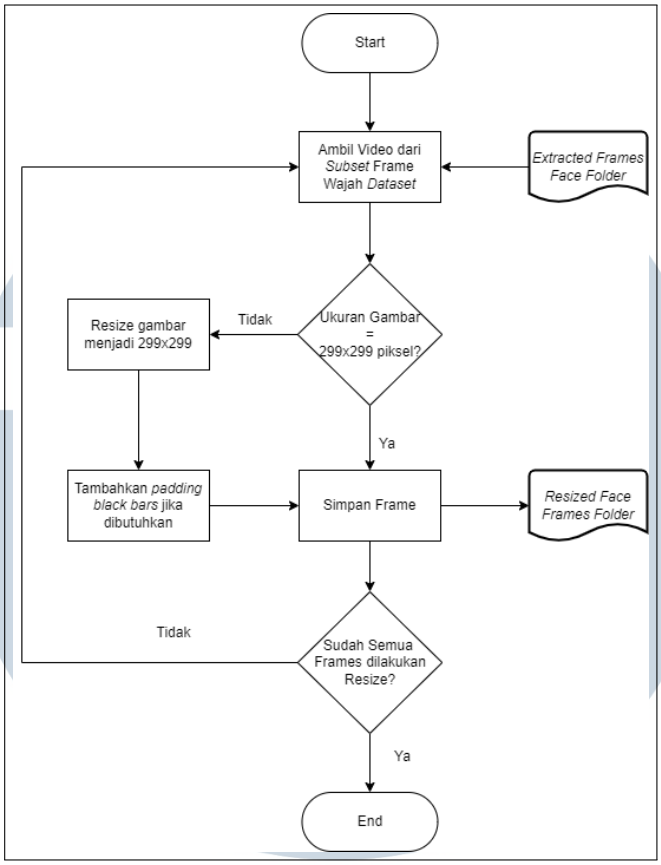
- Proses dilanjutkan hingga berhasil memproses seluruh 1.000 identitas video yang valid.

5. Struktur Dataset Akhir

   - Dataset terdiri dari 1.000 video real dan 5.000 video fake yang terbagi merata ke dalam lima metode *deepfake* (Deepfakes, Face2Face, FaceSwap, NeuralTextures, dan FaceShifter).
   - Setiap video menghasilkan 15 *frame* wajah, sehingga total terdapat:
     - 15.000 *frame* wajah real,
     - 75.000 *frame* wajah fake (15.000 untuk setiap metode).
   - Dataset ini digunakan untuk proses pelatihan, validasi, dan pengujian model klasifikasi.

3.4.2 *Resize Frame* Wajah

Setelah seluruh proses seleksi video, ekstraksi frame, dan deteksi wajah telah selesai dilakukan, maka tahap selanjutnya adalah penyesuaian ukuran gambar menjadi 299x299 piksel RGB wajah seperti pada gambar 3.7 agar sesuai dengan kebutuhan InceptionV3 dengan tahapan sebagai berikut.



Gambar 3.7. *Flow Resize Wajah*

1. Setiap gambar wajah yang telah terdeteksi dan diekstrak akan dibuah ukurannya menjadi 299x299 piksel dengan 3 saluran warna (RGB), sesuai dengan *expected input shape* dari arsitektur InceptionV3.
2. Untuk menjaga proporsi asli gambar wajah sebelumnya agar tidak terdistorsi, *aspect ratio* gambar akan dipertahankan dengan menambahkan *padding* berupa latar belakang hitam (*black bars*), sehingga hasil akhir akan memiliki ukuran 299x299 piksel tanpa merubah bentuk wajah apabila ukuran gambar sebelumnya belum 299x299 piksel.
3. Gambar *frame* hasil *resizing* inilah yang akan digunakan sebagai *input* untuk proses pelatihan model.

3.4.3 Pembagian Subset Dataset untuk Pelatihan dan Evaluasi

Setelah seluruh gambar wajah berhasil diproses melalui tahap *resizing*, dataset yang terbentuk kemudian dibagi menjadi enam *subset* berdasarkan metode

pembuatan video *deepfake*. Pembagian ini bertujuan agar setiap model dapat dilatih dan dievaluasi secara spesifik terhadap satu jenis teknik manipulasi wajah, sehingga performa model terhadap masing-masing jenis *deepfake* dapat dianalisis secara mendalam.

1. Terdapat lima *subset*, masing-masing mewakili satu metode pembuatan *deepfake* sebagai berikut.
    - DeepFakes
    - Face2Face
    - FaceSwap
    - NeuralTextures
    - FaceShifter
2. Terdapat satu *subset* yang mewakili gabungan dari seluruh metode pembuatan video *deepfake*.
3. Setiap *subset* terdiri dari:
    - 1.000 video asli yang sama di seluruh subset, yaitu 15.000 gambar wajah hasil ekstraksi.
    - 1.000 video *deepfake* yang dibuat dari metode terkait, yaitu 15.000 gambar wajah hasil manipulasi.
4. Pada subset gabungan seluruh metode, *dataset* tetap memiliki jumlah video asli dan *deepfake* yang sama. Hal ini dilakukan dengan membagi secara rata dari lima metode pembuatan *deepfake* (masing-masing 160 video untuk *training*, 25 untuk *validation*, 25 untuk *testing*). Hal ini dilakukan untuk memastikan bahwa total jumlah data dan distribusi label pada subset gabungan tetap setara dengan subset lain.
5. Dengan demikian, tiap *subset* berisi total 30.000 gambar wajah (15.000 gambar asli dan 15.000 gambar *deepfake*) yang digunakan untuk pelatihan, validasi, dan pengujian model.

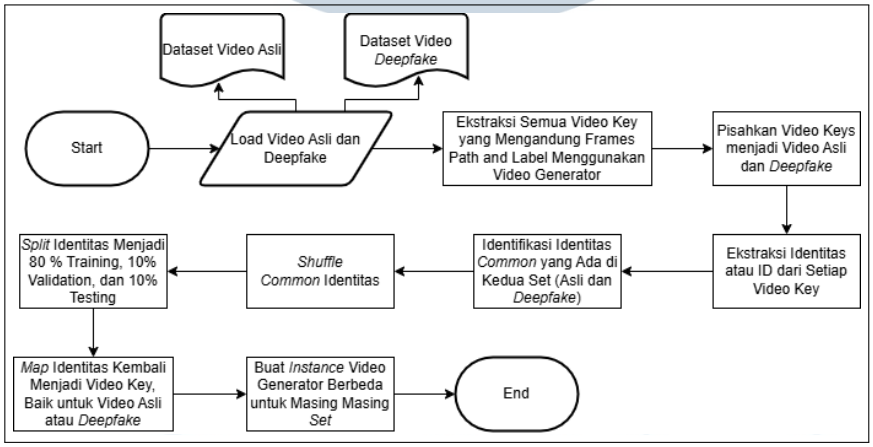
3.5 Pelatihan Model

Setelah proses *pre-processing* data selesai, tahap berikutnya adalah pelatihan model menggunakan data yang telah disiapkan sebelumnya. Pelatihan ini mencakup *data splitting*, perancangan arsitektur model, strategi *loading* input data, penerapan *fine-tuning*, dan proses pelatihan aktual.

3.5.1 *Data Splitting*

Setiap subset yang telah terbentuk pada tahap sebelumnya kemudian dibagi lagi menjadi tiga bagian untuk keperluan pelatihan dan evaluasi model dengan perbandingan sebagai berikut.

- 80% data digunakan untuk pelatihan.
- 10% data digunakan untuk validasi.
- 10% data digunakan untuk pengujian.



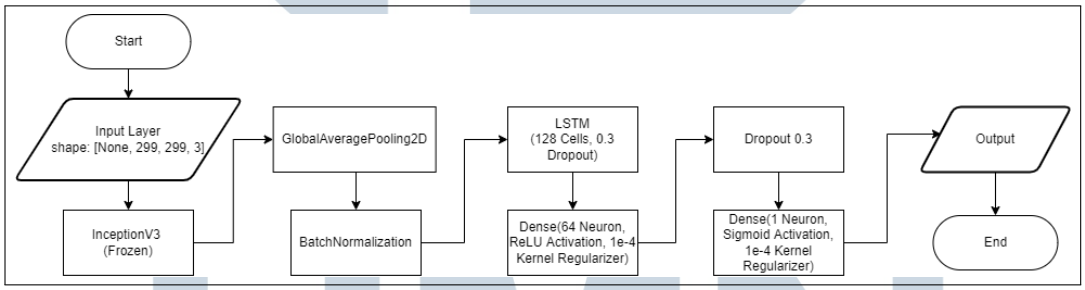
Gambar 3.8. *Flow Data Splitting*

Data splitting akan dilakukan dengan tahapan seperti pada gambar 3.8. Video key yang sama akan digunakan untuk seluruh pelatihan *subset* model.

1. *Load frames* yang telah di praproses sebelumnya, baik untuk video asli ataupun *deepfake* yang ingin digunakan.
2. Ekstraksi seluruh video key yang mengandung *path* dan *label* dari *frame* yang dikelompokkan berdasarkan video.

3. Pisahkan video keys menjadi video asli dan *deepfake*.
4. Ekstraksi identitas dari setiap video key.
5. Identifikasi identitas yang terdapat baik pada video asli ataupun *deepfake*.
6. *Shuffle* identitas tersebut.
7. *Split* identitas berdasarkan *splitting data* yang telah ditentukan sebelumnya.
8. *Map* identitas tersebut kembali menjadi video key, baik untuk video asli ataupun *deepfake* yang ingin digunakan.
9. Menggunakan video key yang telah terbentuk, bentuklah video generator untuk *training*, *validation*, dan *testing*.

3.5.2 Perancangan Arsitektur Model



Gambar 3.9. Arsitektur Model

Model yang digunakan merupakan kombinasi dari InceptionV3 dan *Long Short-Term Memory* (LSTM) untuk mengolah data video secara spasial dan temporal. Berikut adalah arsitektur model yang akan digunakan. Gambar 3.9 menunjukkan arsitektur model yang akan digunakan dengan penjelasan sebagai berikut.

- **Input**: Model menerima input berupa urutan frame video dengan dimensi `(None, 299, 299, 3)`, di mana setiap frame berukuran 299x299 piksel dengan 3 saluran warna (RGB).
- **Ekstraksi fitur spasial**: Setiap frame diproses secara individual menggunakan *base model* CNN yaitu *InceptionV3* yang telah dilatih sebelumnya pada dataset ImageNet, tanpa lapisan klasifikasi terakhir yang

dilatih untuk dataset ImageNet. Ekstraksi fitur dilakukan menggunakan lapisan `TimeDistributed` agar CNN dapat diterapkan pada setiap frame secara paralel. Seluruh lapisan InceptionV3 akan di *freeze* untuk memanfaatkan *transfer learning*.

- **Pooling spasial**: Fitur hasil ekstraksi dari setiap frame kemudian diringkas menggunakan *Global Average Pooling 2D* secara `TimeDistributed`, yang mereduksi dimensi spasial menjadi vektor fitur berdimensi tetap per frame. Hal ini berfungsi untuk mengurangi kompleksitas dan menjaga informasi global pada setiap frame.
- **Pemodelan temporal**: Vektor fitur sekuensial tersebut selanjutnya diproses oleh satu lapisan `LSTM` dengan 128 unit neuron dan *dropout* sebesar 0.3 untuk mencegah *overfitting*. Lapisan ini bertugas menangkap pola temporal dalam pergerakan wajah antar frame.
- **Klasifikasi**: Hasil dari LSTM diteruskan ke lapisan `Dense` dengan 64 neuron dan fungsi aktivasi ReLU, dengan *L2 regularization* 1×10^{-4} untuk mengurangi risiko *overfitting*. Setelah itu, dilakukan `Dropout` sebesar 0.3. Output akhir dihasilkan melalui lapisan `Dense` dengan 1 neuron dan fungsi aktivasi sigmoid untuk klasifikasi biner (real atau fake), juga dengan kernel regularizer 1×10^{-4}.
- **Kompilasi model**: Model dikompilasi menggunakan optimizer Adam dengan learning rate sebesar 1e-3, fungsi loss `binary crossentropy`, serta metrik akurasi untuk evaluasi performa.

3.5.3 Konfigurasi *Hyperparameter*

Untuk memperoleh performa yang optimal, penelitian ini menggunakan konfigurasi hyperparameter sebagai berikut:

- ***Learning Rate***: 3×10^{-4}
- ***Batch Size***: 32
- ***Weight Decay L2 Regularization***: 1×10^{-4}
- **Jumlah *Epoch***: 32

- ***Early Stopping***: 3 (Model akan berhenti berlatih jika tidak ada peningkatan pada metrik *validation loss* selama 3 epoch berturut-turut, mencegah *overfitting*).

3.5.4 Strategi *Loading* Input Data

Strategi *loading* data dalam penelitian ini menggunakan pendekatan generator khusus yang dirancang untuk memproses data video dalam bentuk urutan *frame*. Karena data video memiliki ukuran besar dan terdiri dari banyak frame per video, maka diperlukan metode pemuatan data secara dinamis agar efisien dan tidak membebani RAM. Oleh karena itu, diimplementasikan sebuah generator yang memanfaatkan `Sequence` dari Keras. Generator ini bertanggung jawab untuk hal-hal berikut.

- **Pengelompokan Frame Berdasarkan Video:** Setiap folder berisi frame-frame dari beberapa video. Berdasarkan pola nama *file*, *frame* yang berasal dari video yang sama dikelompokkan menjadi satu *entry*. *Frame-frame* tersebut diurutkan secara numerik untuk mempertahankan urutan temporal dari video aslinya. Setiap entri disimpan dalam struktur data yang menyimpan daftar path frame serta label kelasnya.

- **Augmentasi Data:** Khusus untuk data *training*, setiap *frame* akan memiliki kemungkinan untuk mengalami proses augmentasi seperti *horizontal flip*, penyesuaian *brightness/contrast*, modifikasi *hue/saturation/value*, dan efek *blur*. Hal ini meningkatkan variabilitas data dan membantu model mencapai generalisasi yang lebih baik, sementara augmentasi gambar tidak akan dilakukan pada data validasi dan testing.

- **Penyusunan dan *Loading* Batch:** Untuk setiap batch pelatihan, generator memilih sejumlah video. Jumlah frame dari tiap video disesuaikan dengan batas maksimum frame yang telah ditentukan. Jika jumlah frame kurang dari batas tersebut, maka akan ditambahkan *frame* kosong sampai jumlah *frame* yang ditentukan.

- ***Preprocessing* Frame:** Setiap *frame* yang dimuat melewati serangkaian langkah *preprocessing* sebelum dimasukkan sebagai *input* ke dalam model. Urutan langkahnya adalah sebagai berikut.

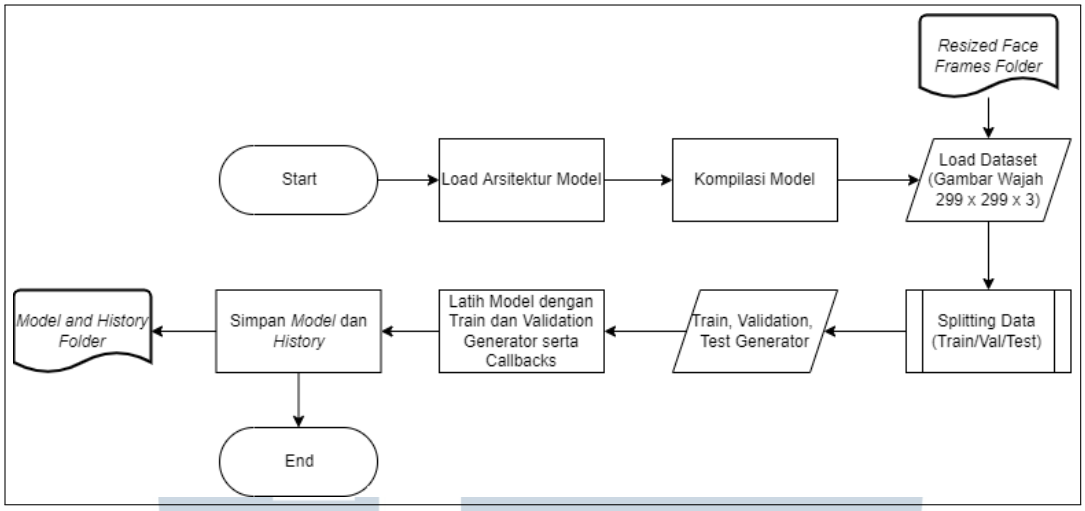
    1. *Frame* dibaca dari *file* menggunakan OpenCV.

2. Gambar dikonversi dari BGR ke RGB, sesuai dengan format yang diharapkan oleh model Keras.
3. Gambar diubah ukurannya menjadi dimensi input 299x299 piksel yang dibutuhkan oleh arsitektur InceptionV3.
4. Nilai piksel dinormalisasi sesuai standar pra-pemrosesan InceptionV3 (mengubah *range* piksel [0, 255] menjadi [-1, 1]) melalui fungsi `preprocess_input` bawaan Keras.

- **Output Generator:** Generator mengembalikan dua buah tensor untuk setiap iterasi: satu tensor berisi urutan frame untuk tiap video dalam satu batch, dan satu tensor lagi berisi label untuk masing-masing video.
- **Pembagian Data:** Generator dapat melakukan pemisahan data yang konsisten untuk keperluan *training*, *validation*, dan *testing* melalui pemilihan subset video.
- **Pengacakan pada Akhir Tiap Epoch:** Untuk mencegah model menghafal urutan data, generator akan mengacak urutan video setiap kali satu epoch pelatihan selesai pada fase *training*.

3.5.5 Proses Pelatihan Model

Proses pelatihan model dilakukan dengan menggabungkan arsitektur yang telah dirancang, konfigurasi *hyperparameter*, dan strategi pemuatan data menggunakan generator. Gambar 3.10 menunjukkan tahapan-tahapan yang akan dilakukan pada setiap model yang akan dilatih.



Gambar 3.10. *Flow* Proses Pelatihan Model

1. **Load Arsitektur Model**: Model InceptionV3-LSTM dimuat atau didefinisikan sesuai dengan kebutuhan *input*.

2. **Kompilasi Model**: Model dikompilasi menggunakan *optimizer* Adam, fungsi *loss* `binary_crossentropy`, dan metrik akurasi.

3. **Load Dataset (Gambar Wajah 299×299×3)**: *Dataset* berupa gambar wajah hasil *preprocessing* dimuat dari *folder* yang telah berisi *frame* yang telah diubah ukurannya menjadi 299×299 piksel.

4. **Splitting Data (Train/Val/Test)**: Dataset dibagi berdasarkan identitas menjadi tiga bagian: training (80%), validation (10%), dan testing (10%).

5. ***Train*, *Validation*, *Test* Generator**: Berdasarkan hasil pembagian data, tiga generator dibentuk untuk *loading batch* data secara dinamis ke dalam memori.

6. **Latih Model dengan *Train* dan *Validation* Generator serta *Callbacks***: Model dilatih menggunakan generator *training* dan divalidasi menggunakan generator *validation*. *Callbacks* seperti `EarlyStopping` digunakan untuk mengatur proses pelatihan secara efisien.

7. **Simpan Model dan *History***: Model dengan *weights* terbaik serta *history* pelatihan yang diperoleh disimpan ke direktori tertentu untuk evaluasi lebih lanjut.

3.6 Evaluasi Model

Setelah proses pelatihan selesai, model akan dievaluasi menggunakan metrik-metrik berikut. Evaluasi akan dilakukan terhadap setiap model yang dibuat, dan hasilnya akan dibandingkan untuk menentukan model mana yang memiliki hasil terbaik. Evaluasi dilakukan dalam dua skenario, yaitu *intra-subset evaluation* dan *cross-subset evaluation*.

1. *Accuracy*: Mengukur proporsi atau persentase prediksi yang benar, baik positif maupun negatif, dari total jumlah prediksi. Metrik ini memberikan gambaran umum tentang performa model dalam mengklasifikasikan video *deepfake* dan video *asli*.

2. *Precision*: Mengukur proporsi prediksi positif yang benar dari semua prediksi positif. Metrik ini menunjukkan kemampuan model untuk menghindari false positive (mengidentifikasi video asli sebagai video *deepfake*).

3. *Recall*: Mengukur proporsi video *deepfake* yang berhasil diidentifikasi dari total video *deepfake* yang ada. Metrik ini menunjukkan kemampuan model dalam mendeteksi semua video *deepfake*.

4. *F1-score*: Merupakan rata-rata harmonik dari *precision* dan *recall*. Menyeimbangkan kedua metrik tersebut dan memberikan ukuran tunggal untuk mengevaluasi performa model.

5. *AUC-ROC*: Mengukur kemampuan model untuk membedakan antara dua kelas. Nilai yang lebih tinggi menunjukkan model yang lebih baik dalam membedakan antara video asli dan video *deepfake* pada berbagai *threshold*.

3.6.1 Evaluasi *Intra-Subset*

Evaluasi pertama dilakukan dengan menguji setiap model pada *subset* data yang sama dengan data pelatihannya (misalnya, model yang dilatih pada subset Face2Face diuji pada data Face2Face). Hasil dari evaluasi ini akan menunjukkan kemampuan model dalam mengenali pola *deepfake* spesifik dari satu metode manipulasi wajah.

3.6.2 Evaluasi *Cross-Subset*

Untuk mengukur kemampuan generalisasi model, setiap model juga diuji pada *subset* lain yang berbeda dari data latihnya. Sebagai contoh, model yang dilatih pada data Face2Face akan diuji pada subset DeepFakes, FaceSwap, FaceShifter, dan NeuralTextures.

Evaluasi ini dilakukan dengan menggunakan metrik **AUC-ROC** karena fokus utamanya adalah pada kemampuan model dalam membedakan antara video asli dan *deepfake* tanpa tergantung pada ambang klasifikasi tertentu. Nilai AUC-ROC yang tinggi menunjukkan model yang lebih general dan tidak hanya belajar spesifik terhadap satu metode manipulasi.