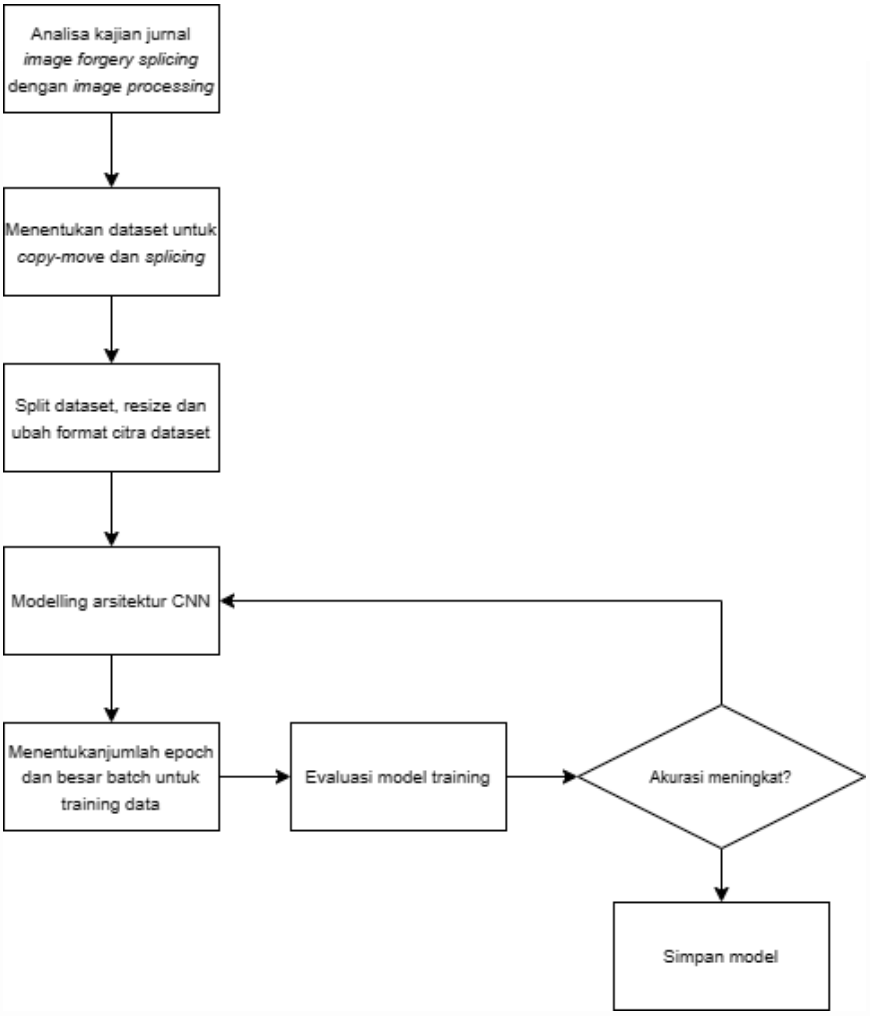
BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Pada penelitian untuk membuat model pendeteksi citra palsu dengan pendekatan *image processing* menggunakan metode *knowledge discovery in database* (KDD). Tahapan yang ada pada metode ini terdapat *pre-KDD, selection, pre-processing, transformation, data mining, evaluation,* dan *post-KDD.* Semua tahapan tersebut diterapkan pada proses pembuatan model deteksi citra palsu dengan pendekatan *image processing*. Kerangka kerja penelitian ini dapat dilihat pada gambar 3.1.



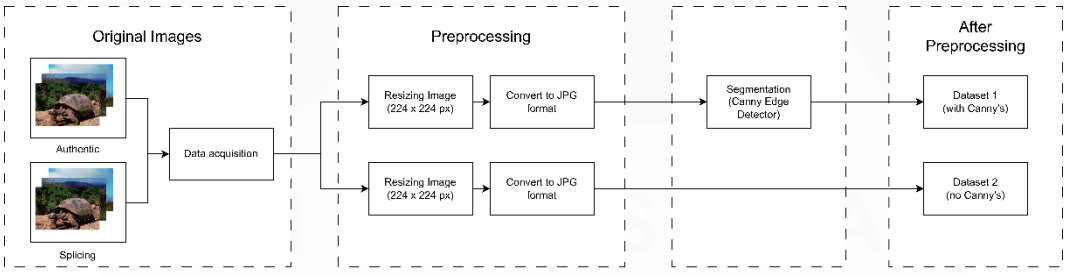
Gambar 3. 1 Kerangka kerja penelitian KDD

3.2 Tahapan Penelitian

Berdasarkan kerangka kerja pada gambar 3.1, berikut penjelasandari setiap tahapan kerangka kerja dengan menggunakan metode KDD:

1. *Pre-KDD*: tahapan pertama yang dilakukan adalah melakukan kajian pada jurnal terdahulu mengenai pendeteksi pemalsuan citra *splicing* menggunakna pendekatan *image processing*. Hasilnya ditemukan tidak sedikit yang menggunakan *machine learning* daripada *deep learning* untuk melakukan deteksi pemalsuan citra untuk *splicing* dengan pentekatan *image processing*. Selain itu terdapat beberapa publikasi yang sudah menggunakan *deep learning* tetapi untuk akurasi model pendeteksi masih di angka 70% untuk pendeteksi citra palsu untuk *splicing*.

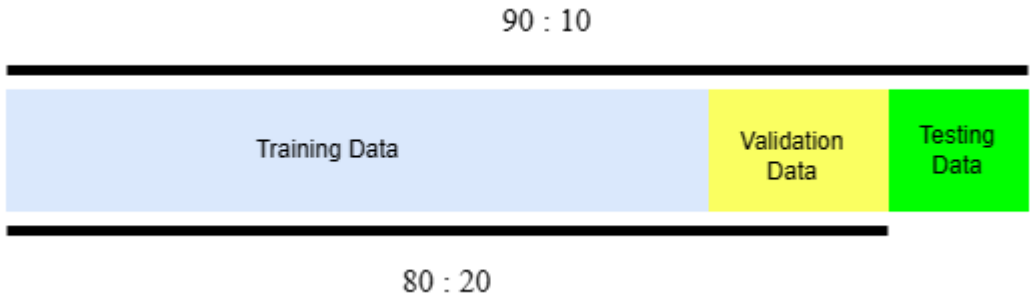
2. *Selection*: dataset untuk splicing menggunakan CASIA-v1 dan CASIA-v2, untuk dataset *splicing* berdasarkan pembagian topik dan dataset setiap anggota penelitian. Kedua dataset tersebut digabung menjadi satu dataset menjadi CASIAMIX. Untuk semua dataset yang digunakan tersedia di kanal internet seperti situs resmi dan github sehingga pengumpulan dataset dapat dilakukan dengan mengunduh dataset tersebut.
3. *Pre-processing*: dataset yang digunakan yaitu CASIAMIX, *pre-processing* yang dilakukan adalah mengubah semua format citra pada dataset CASIAMIX menjadi .jpg dan melakukan *resize* terhadap citra pada dataset menjadi ukutan 224 x 224 piksel. Kemudian tahapan *pre-processing* selanjutnya dengan menggandakan dataset menjadi dua dataset. Untuk dataset yang pertama pengaplikasian proses *image processing* mengunakan *Canny Edge Detector* untuk setiap citra yang ada dalam dataset dan kemudian dataset tersebut ditandai dengan nama dataset 1 *(with Canny Edge Detection*). Sedangkan untuk dataset yang kedua tidak ada melakukan proses *image processing*, bisa dikatakan masih dasar awal (*Base*) dan dataset kedua ditandai dengan nama dataset 2 (*no Canny Edge Detection/Base*). Untuk visualisasi mengenai proses *pre-processing* bisa dilihat pada gambar 3.2.



Gambar 3. 2 *Flow Diagram* pada tahap *pre-processing*

Selanjutnya tahapan yang dilakukan adalah melakukan pembagian dataset menjadi training data, *validation data*, dan *testing data*.

Untuk pembagian dataset CASIAMIX dibagi menjadi 90:10, 90% untuk train master dan 10% untuk *testing data*. Kemudian dari 90% yang berasal dari train master dilakukan pembagian data menjadi 80:20, yang dimana 80% untuk *training data* dan 20% untuk *validation data*. Untuk mengetahui gambaran pembagian data dapat dilihat pada gambar 3.3 dan hasil dari pre-processing untuk kedua dataset CASIAMIX dapat dilihat pada tabel 3.1 dan tabel 3.2.



Gambar 3. 3 Visualisasi skema pembagian data

Tabel 3. 1 Hasil *pre-processing* dataset CASIAMIX *Base*

	Training	***Validation***	***Testing***
Authentic (Au)	5370	1343	746
Splicing (Sp)	1122	281	156

Tabel 3. 2 Hasil *pre-processing* dataset CASIAMIX *Edge Detection*

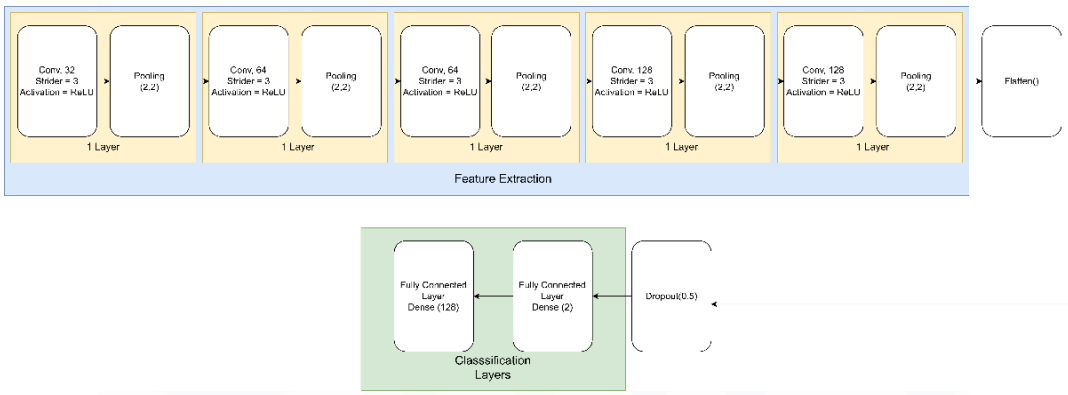
	Training	***Validation***	***Testing***
Authentic (Au)	5370	1343	746
Splicing (Sp)	1122	281	156

4. Transformation: kedua dataset yang telah melakukan proses pre-processing kemudian dibentuk model nya menggunakan arsitektur CNN dan arsitektur ResNet50 sebagai model pembanding. Penggunaan CNN untuk membangun model dikarenakan CNN memiliki fleksibilitas untuk menentukan jumlah *layer convolutional*, *layer max-pooling*, dan *layer fully connected*. Pembetukan model CNN menggunakan pythin sebagai

bahasa pemograman dengan menggunakan library Tensorflow dan Keras. Arsitektur CNN pada beberapa dataset CASIAMIX bisa dilihat pada tabel 3.3.

Tabel 3. 3 Struktur arsitektur model CNN untuk CASIAMIX *base* dan CASIAMIX *Edge Detection*

Layer	***Type***	***Activation Function***	***Output Shapes***	***Kernel Size***	***Total Filters***
0	*Input*				
1	*2D Convolution*	ReLU	222	3	32
2	*2D Max Poling*	ReLU	111	3	32
3	*2D Convolution*	ReLU	109	3	64
4	*2D Max Poling*	ReLU	54	3	64
5	*2D Convolution*	ReLU	52	3	64
6	*2D Max Poling*	ReLU	26	3	64
7	*2D Convolution*	ReLU	24	3	128
8	*2D Max Poling*	ReLU	12	3	128
9	*2D Max Poling*	ReLU	10	3	128
10	*2D Max Poling*	ReLU	5	3	128
	Flatten	-	-		
	Dropout		-		
	Dense	-	128		



Gambar 3. 4 *Flow Diagram* dari arsitektur model CNN

Pada model diatas memiliki 5 *convolutional layer* dan *max-pooling*. *Layer* pertama memiliki 32 filter, untuk *layer* kedua dan *layer* ketiga memiliki 64 filter, *layer* keempat dan *layer* kelima memiliki 128 filter. Kemudian kelima layer tersebut menggukan *activation function* ReLU. Kernel untuk setiap *convolutional layer* berukuran 3x3.Untuk ukuran pada max-pooling

layer adalah 2x2 dan diletakkan sesudah setiap *convolutional layer*. Optimzer yang digunakan pada model tersebut menggunakan Adam *optimizer*. Untuk kalkulasi *loss* pada model tersebut menggunakan metode *binary crossentropy* berdasarkan jumlah kelas yang ada pada dataset CASIAMIX yaitu dua.

5. *Data mining*: Model yang sudah dibuat menggunakan arsitektur CNN kemudian akan dilakukan *running* pada *training data* yang sudah disiapkan pada tahapan pre-processing. Sebelum melakukan *running* pada *training data*, jumlah pada epoch dan batch size ditentukan berdasarkan jumlah data. Setelah melakukan pelatihan pada *training data*, hasil akan disimpan menggunakan format file .h5 Untuk penjelasan bisa dilihat pada tabel 3.4.

Tabel 3. 4 Ringkasan model *training* setiap dataset

Model	***Optimizer***	***Batch Size***	**Jumlah *Epoch***
Deteksi *splicing* CASIAMIX *Base*	Adam	10	84
Deteksi *splicing* CASIAMIX *Edge Detection*	Adam	10	92

Perbedaan jumlah epoch yang dijalankan pada kedua model dengan dataset berbeda dikarenakan terdapat perbedaan citra yang ada pada masing-masing dataset. Penggunakan *image processing* yaitu *Edge Detection* pada dataset CASIAMIX membuat jumlah epoch yang dilakukan selisih 8 epoch dengan dataset CASIAMIX *base.*

6. *Interpretation/Evaluation:* evaluasi model dapat diperform dengan memeriksa akurasi dalam tahap pelatihan, validasi, dan pengujian. Untuk mengimplementasikan ini dalam bahasa Python, dapat digunakan metode model.evaluate(). Selain itu, penilaian model dapat dilakukan terhadap data validasi dan pengujian dengan memanfaatkan matriks kebingungan (confusion matrix) dan kurva ROC (Receiver Operating Characteristic). Langkah ini juga akan menentukan apakah diperlukan pelatihan ulang

model dengan melakukan beberapa penyesuaian guna meningkatkan akurasi. Hasil akurasi pengujian akan mencerminkan seberapa baik kinerja model sebenarnya. Selain akurasi, penggunaan matriks kebingungan dan kurva ROC juga akan menampilkan nilai-nilai precision, recall, F1-score, dan support.

7. *Post*-KDD: Hasil penelitian ini akan menggunakan model terakhir yang mencapai akurasi tertinggi dalam mendeteksi splicing pada dataset CASIAMIX base dan CASIAMIX Edge Detection.

3.3 Teknik Pengumpulan data

Teknik pengumpulan data yang digunakan pada penelitian ini adalah data sekunder. Data sekunder merupakan data yang diperoleh secara tidak langsung dari objeknya, tetapi melalui sumber lain, dalam hal ini data yang diperoleh berasal dari situs website resmi suatu institusi ataupun forum. Data yang diperoleh untuk digunakna pada penelitian ini adalah CASIA-v1 dan CASIA-v2 yang dapat diperoleh pada situs Kaggle dan Github. Terdapat banyak versi pada dataset tersebut, tetapi dataset yang dipakai pada penelitian ini bersumber dari Github dikarenakan pada situs resmi yaitu situs *Center for Biometrics and Security Research* (CBSR) dan Kaggle terdapat beberapa data yang hilang maupun *corrupt* dan untuk sumber Github, data sudah dilakukan revisi dan terdapat groundtruth pada folder dataset.

Tabel 3. 5 *Link* sumber Dataset

Dataset	**Kelas**	**Jumlah**	***Link***
CASIA-v1	*Authentic* (Au)	800	https://github.com/namtpham/casia1groundtruth
	Splicing (Sp)	462	
CASIA-v2	*Authentic* (Au)	7491	https://github.com/sunnyhaze/casia2.0-corrected-groundtruth
	Splicing (Sp)	1828	

\