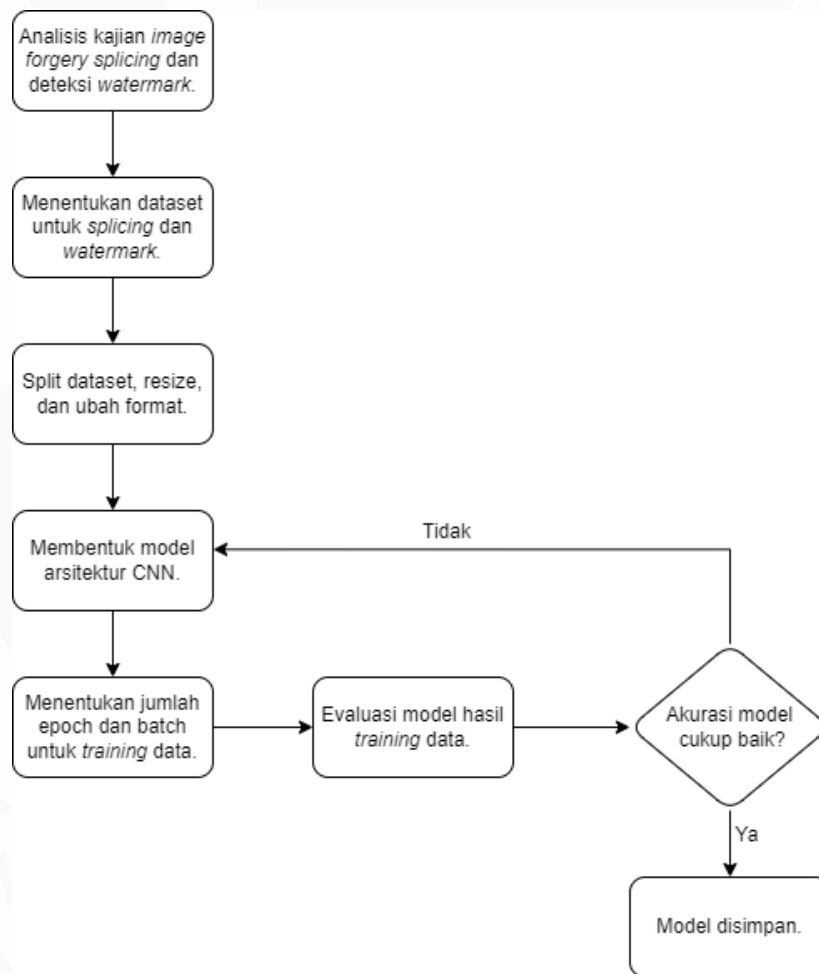


BAB III

METODE PENELITIAN

3.1. Metode Penelitian

Penelitian yang dilakukan bertujuan untuk membuat model deteksi *digital watermark* dan *splicing* menggunakan metode *knowledge discovery in database* (KDD). Tahap-tahap KDD terdiri atas *pre-KDD*, *selection*, *pre-processing*, *transformation*, *data mining*, *evaluation*, dan *post-KDD*. Seluruh tahapan tersebut akan diterapkan pada proses pembuatan model deteksi *splicing dataset* DVMM dan deteksi *watermark* pada *dataset Watermarked/not watermarked images*. *Framework* penelitian menggunakan KDD tertera pada Gambar 3.1.



Gambar 3.1 *Framework* penelitian

3.2 Tahapan Penelitian

Melalui *framework* penelitian, tahapan-tahapan KDD diterapkan pada penelitian ini. Penjelasan mengenai tahap penelitian menggunakan KDD adalah sebagai berikut:

1. *Pre-KDD*: melakukan kajian mengenai permasalahan *image forgery*, terutama kajian mengenai deteksi *image forgery* teknik *splicing* dan deteksi *watermark*. Hasilnya ditemukan bahwa mayoritas publikasi sebelumnya menggunakan *machine learning*, bukan *deep learning* untuk mendeteksi *splicing* pada gambar. Selain itu, masih sedikit penelitian yang mendeteksi *watermark* untuk mendeteksi secara aktif aktivitas *image forgery*. Mayoritas publikasi sebelumnya menggunakan algoritma *deep learning* untuk mendeteksi jenis penyerangan terhadap *watermark* gambar digital, bukan *watermark*-nya.
2. *Selection*: untuk *splicing*, *dataset* yang digunakan adalah DVMM berdasarkan hasil pembagian *dataset* untuk setiap anggota penelitian. Pertimbangan lainnya adalah klasifikasi data DVMM hanya data *authentic* dan *splicing* yang cocok dengan tujuan penelitian. Untuk deteksi *watermark* menggunakan *dataset* “*Watermarked/not watermarked images*” yang diunggah oleh Felice Polano di website Kaggle. Kedua *dataset* tersebut tersedia di internet sehingga pengumpulan *dataset* dilakukan dengan mengunduh keduanya.
3. *Pre-processing*: untuk *dataset* DVMM, *pre-processing* yang dilakukan hanya membagi kedua kelas, yaitu *authentic* (Au) dan *splicing* (Sp) menjadi data *training*, data *testing*, dan data validasi. Dengan menggunakan python, *dataset* DVMM dibagi menjadi 90% untuk *train master* dan 10% untuk data *testing*. Dari 90% data DVMM yang menjadi *train master*, dibagi lagi menjadi 70% data *training* dan 30% data validasi. Proses yang sama juga berlaku untuk data *Watermarked/not watermarked images*, hanya saja sebelum dibagi, *dataset watermark* akan diubah formatnya menjadi .jpg dan diubah ukurannya menjadi 224x224 px menggunakan python. Jumlah *dataset* awal DVMM adalah 933 untuk Au dan 912 untuk Sp sedangkan *Watermark/Not Watermarked Images* memiliki jumlah awal 15.099 untuk *no-watermarked* dan 15.128 untuk

watermarkd. Hasil *pre-processing* kedua *dataset* menjadi data *training*, data validasi, dan data *testing* tertera pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.1 Hasil *pre-processing* dataset DVMM

	<i>Training</i>	<i>Validation</i>	<i>Testing</i>
<i>Authentic</i> (Au)	587	252	94
<i>Splicing</i> (Sp)	574	246	92

Tabel 3.2 Hasil *pre-processing* dataset Watermarked/not watermarked images.

	<i>Training</i>	<i>Validation</i>	<i>Testing</i>
No-Watermarked	9.512	4.077	1.510
Watermarked	9.530	4.085	1.513

4. *Transformation*: *dataset* yang sudah melalui *pre-processing* akan dibentuk arsitektur CNN-nya. CNN memiliki fleksibilitas untuk menentukan jumlah *layer convolutional*, *layer max-pooling*, dan *layer fully connected*. Tujuannya supaya bisa menyesuaikan dengan kebutuhan penelitian. Selain itu, pada tahap ini juga harus ditentukan jenis aktivasi, penggunaan *drop out*, *loss function*, *optimizer*, dan *input*. CNN melakukan *feature extraction* secara otomatis sehingga tidak perlu dibuat secara manual. Tahap ini berakhir pada proses *compile* model. Pembentukan arsitektur CNN menggunakan bahasa pemrograman python yang memanggil library Keras dan Tensorflow. Struktur arsitektur CNN pada *dataset* DVMM untuk mendeteksi *splicing* tertera pada Tabel 3.3.

Tabel 3.3 Struktur arsitektur model CNN untuk *dataset* DVMM

<i>Layer</i>	<i>Type</i>	<i>Activation Function</i>	<i>Output Shapes</i>	<i>Kernel Size</i>	<i>Total Filters</i>
0	<i>Input</i>				
1	<i>2D Convolution</i>	ReLU	128	3	32
2	<i>2D Max Poling</i>	ReLU	126	3	32
3	<i>2D Convolution</i>	ReLU	63	3	64
4	<i>2D Max Poling</i>	ReLU	61	3	64
5	<i>2D Convolution</i>	ReLU	30	3	64
6	<i>2D Max Poling</i>	ReLU	28	3	64
7	<i>2D Convolution</i>	ReLU	14	3	64
8	<i>2D Max Poling</i>	ReLU	12	3	64
	<i>Flatten</i>	-	6		
	<i>Dense</i>	-	2304		

Struktur model CNN yang dibuat untuk *dataset* DVMM memiliki 4 lapisan *convolutional* dan *max-pooling*. Keempat lapisan tersebut menggunakan fungsi aktivasi ReLU. *Layer* pertama memiliki 32 filter, *layer* kedua sampai *layer* keempat memiliki 64 filter. Seluruh *layer convolutional* memiliki ukuran kernel 3x3. Ukuran *layer max-pooling* adalah 2x2 dan diletakkan sesudah setiap lapisan *convolutional*. Model di-*compile* dengan *optimizer* Adam dan *loss* dikalkulasikan menggunakan metode *binary cross entropy* untuk mencatat setiap *error* yang muncul saat proses *training* data.

Tabel 3.4 Struktur arsitektur model CNN untuk dataset Watermarked/not watermarked images

<i>Layer</i>	<i>Type</i>	<i>Activation Function</i>	<i>Output Shapes</i>	<i>Kernel Size</i>	<i>Total Filters</i>
0	<i>Input</i>				
1	<i>2D Convolution</i>	ReLU	224	3	32
2	<i>2D Max Poling</i>	ReLU	222	3	32
3	<i>2D Convolution</i>	ReLU	111	3	64
4	<i>2D Max Poling</i>	ReLU	109	3	64
5	<i>2D Convolution</i>	ReLU	54	3	64
6	<i>2D Max Poling</i>	ReLU	52	3	64
7	<i>2D Convolution</i>	ReLU	26	3	128
8	<i>2D Max Poling</i>	ReLU	24	3	128
9	<i>2D Convolution</i>	ReLU	12	3	128
10	<i>2D Max Poling</i>	ReLU	10	3	128
	<i>Flatten</i>	-	5		
	<i>Dense</i>	-	3200		

Untuk model arsitektur CNN deteksi *watermark* seluruh jenis aktivasi, ukuran kernel, dan metode kalkulasi *loss* sama seperti model sebelumnya. Namun terdapat perbedaan pada jumlah *layer convolutional* dan *max-pooling*. Pada model ini, jumlah *layer* tersebut masing-masing adalah 5 dengan *layer* pertama memiliki 32 filter, *layer* kedua dan ketiga memiliki 64 filter, *layer convolutional* keempat dan kelima memiliki 128 filter. Selain itu, jenis *optimizer* yang digunakan juga berbeda, yaitu RMSprop.

5. *Data mining*: arsitektur CNN yang sudah membentuk model akan dilatih menggunakan data *training* yang sudah disiapkan pada tahap *pre-processing*. Jumlah epoch dan batch size ditentukan berdasarkan jumlah data dan kemampuan komputasi perangkat. Pada umumnya, semakin besar jumlah data maka semakin besar juga ukuran *batch*-nya. Model yang sudah dilatih bisa disimpan ke perangkat dalam format .h5. Rangkuman model *training* bisa dilihat pada Tabel 3.5.

Tabel 3.5 Rangkuman Model Training Kedua Dataset

Model	<i>Optimizer</i>	<i>Batch Size</i>	<i>Jumlah Epoch</i>
Deteksi <i>splicing</i> DVMM	Adam	10	61
Deteksi <i>watermark</i>	RMSprop	16	76

Batch size untuk model deteksi *watermark* lebih besar karena jumlah *dataset watermark* lebih besar daripada DVMM. Hasil epoch yang dihasilkan dari proses *training* data juga berbeda karena menggunakan metode *early stopping*. Proses *training dataset* DVMM berhenti pada epoch ke-61 sedangkan *dataset watermark* berhenti pada epoch ke-76.

6. *Interpretation/Evaluation*: evaluasi model bisa dilakukan dengan mengecek akurasi *training*, *validation*, dan *testing*. Syntax python yang bisa digunakan adalah `model.evaluate()`. Selain itu, evaluasi model bisa dilakukan pada data validasi dan data *testing* menggunakan bantuan *confusion matrix* dan *ROC curve*. Tahap ini juga akan menentukan apakah model perlu dilatih ulang dengan beberapa penyesuaian untuk meningkatkan akurasi. Hasil akurasi *testing* mencerminkan akurasi model yang sebenarnya. Selain akurasi, dalam penggunaan *confusion matrix* dan *ROC curve*, akan ditampilkan juga nilai *precision*, *recall*, *F1-score*, dan *support*.
7. *Post-KDD*: model akhir yang memiliki akurasi terbaik untuk mendeteksi *splicing* pada *dataset* DVMM dan model untuk mendeteksi *watermark* akan digunakan sebagai hasil penelitian ini.

3.3 Teknik Pengumpulan Data

Pengumpulan data adalah tahap penting dalam setiap penelitian karena berpengaruh terhadap keberhasilan dan validasi hasil penelitian. Dengan menggunakan teknik pengumpulan data yang tepat, peneliti dapat memperoleh informasi yang akurat, relevan, dan bisa digunakan. Pada penelitian ini, data yang digunakan adalah data sekunder. Data adalah data yang dibuat oleh orang lain dengan tujuan yang berbeda. Penelitian ini menggunakan data DVMM yang tersedia di internet dan data *Watermarked/not watermarked images* yang diunggah oleh Felice Polano pada website Kaggle. Keduanya tersedia secara *open source* sehingga bisa diunduh oleh peneliti untuk kebutuhan penelitian deteksi pemalsuan gambar digital ini. Sumber dataset yang digunakan dapat dilihat pada tabel 3.6.

Tabel 3.6 Tabel Sumber Dataset

<i>Dataset</i>	<i>Kelas</i>	<i>Jumlah</i>	<i>Link</i>
DVMM	<i>Authentic (Au)</i>	933	https://www.ee.columbia.edu/in/dvmm/downloads/AuthSplicedDataset/AuthSplicedDataset.htm
	<i>Splicing (Sp)</i>	912	
<i>Watermarked/ not watermarked images</i>	<i>No-Watermarked</i>	15.328	https://www.kaggle.com/datasets/felicepollano/watermarked-not-watermarked-images
	<i>Watermarked</i>	15.359	

