

BAB III

METODOLOGI PENELITIAN

3.1. Objek Penelitian

Objek yang diteliti pada penelitian ini adalah gambar *facial expression* yang menunjukkan 7 jenis emosi yaitu *Happy, Anger, Sadness, Fear, Surprise, Disgust dan Neutral* [25] seperti pada gambar 3.1. Pada umumnya, manusia menggunakan isyarat yang berbeda untuk menunjukkan emosi, seperti *facial expression, gerakan tangan, dan suara*. Namun, isyarat yang paling menunjukkan emosi adalah *facial expression, yang dimana facial expression* merepresentasikan 55% emosi manusia sedangkan isyarat lainnya hanya menunjukkan 7% dari emosi manusia [6]. Oleh karena itu, dapat dikatakan bahwa *facial expression* memiliki peran yang berpengaruh besar dalam menunjukkan ekspresi manusia yang beragam.



Gambar 3.1. Dataset RAF-DB

3.2. Data Collection Methods

3.2.1. Data Collections

Penelitian ini juga menggunakan *dataset Real-Word Affective Face Database* (RAF-DB) yang bersumber dari *website* <http://www.whdeng.cn/RAF/model1.html> dan dipublikasikan pada tahun 2017. RAF-DB juga merupakan *dataset* yang berjumlah 29.672 gambar *facial expression* yang sangat beragam. Tetapi *dataset* ini dibagi 2 menjadi *basic emotion* dan *compound emotion*. Untuk penelitian ini hanya akan menggunakan *basic emotion* yang berjumlah sebanyak 15.339 gambar dengan 12.271 *data training* dan 3.068 *data testing* dengan persentase pembagiannya untuk *data training* 80% dan *data testing* 20% menggunakan pembagian presentase data yang sama dengan peneliti yang sama-sama menggunakan *dataset* RAF-DB [27]. *Data training* digunakan untuk melakukan *training* pada model sedangkan *data testing* digunakan untuk memvalidasi hasil model yang didapat. Jenis emosi yang akan diklasifikasi adalah *Happy, Anger, Sadness, Fear, Surprise, Disgust* dan *Neutral* [25]. Berikut jumlah gambar untuk masing-masing ekspresi pada tabel 3.1 [25]:

Tabel 3.1. Jumlah Dataset RAF-DB

No.	Jenis Emosi	Training	Testing	Jumlah
1.	<i>Happiness</i>	4772	1185	5957
2.	<i>Anger</i>	705	162	867

3.	<i>Sadness</i>	1982	478	2460
4.	<i>Fear</i>	281	74	355
5.	<i>Surprise</i>	1290	329	1619
6.	<i>Disgust</i>	717	160	877
7.	<i>Neutral</i>	2524	680	3204

3.2.2. Variabel Penelitian

Variabel penelitian adalah karakter, atribut atau segala sesuatu yang terbentuk, atau yang menjadi perhatian dalam suatu penelitian sehingga mempunyai variasi antara satu objek yang satu dengan objek yang lain dalam satu kelompok tertentu kemudian ditarik kesimpulannya. Variabel penelitian dibagi menjadi 2, yaitu variabel independen dan variabel dependen.

3.2.2.1. Variabel Independen

Variabel independen adalah variabel yang mempengaruhi variabel dependen dan tidak bergantung pada variabel lainnya. Pada penelitian ini, variabel independennya adalah gambar *facial expression* pada *dataset* RAF-DB.

3.2.2.2. Variabel Dependen

Variabel dependen adalah variabel yang dipengaruhi oleh variabel *independent* dan hasil variabel tersebut bergantung kepada variabel lain. Pada penelitian ini, variabel dependennya adalah jenis

ekspresi yang dideteksi yaitu *Happy, Anger, Sadness, Fear, Surprise, Disgust* dan *Neutral*.

3.3. Metode Penelitian

CRISP-DM merupakan kerangka yang menerjemahkan masalah bisnis menjadi *task Data Mining* dan sering digunakan pada proyek *Data Mining*. Kerangka ini merupakan proses KDD (*Knowledge Discovery*) yang sering diadopsi oleh industri [21]. Selain itu juga, CRISP-DM merupakan metode yang paling favorit digunakan [28]. Lalu, ada metode *Data Mining* lainnya yang bernama SEMMA, yang digunakan di *SAS Enterprise Miner Tool* dan tahapnya yang terdiri dari *Sample, Explore, Modify, Model, Assesment*.

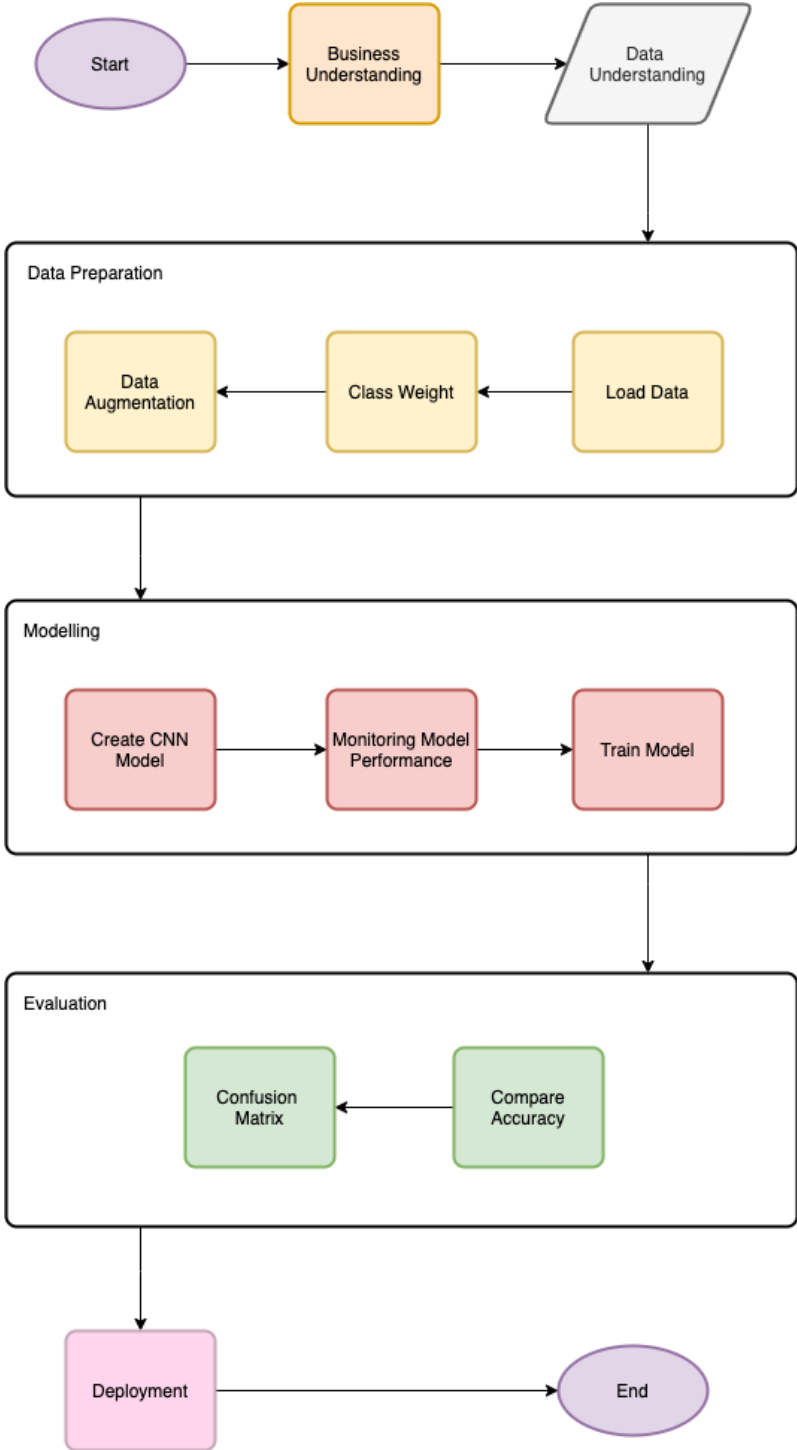
Tabel 3.2. Kelebihan dan Kekurangan CRISP-DM dan SEMMA

Metode	Kelebihan	Kekurangan
SEMMA	<ul style="list-style-type: none"> • <i>User Support</i> yang bagus • Mendukung berbagai teknik <i>Data Mining</i> • Proses yang berulang 	<ul style="list-style-type: none"> • Dirancang untuk bekerja dengan <i>SAS Enterprise Miner Tool</i> • Tidak memasukan tahapan <i>business understanding</i> pada prosesnya
CRISP-DM	<ul style="list-style-type: none"> • Prosesnya didefinisikan secara jelas • Keseluruhan tahapan digunakan untuk <i>transform</i> masalah menjadi bentuk <i>Data Mining</i>. • Proses yang berulang • Mempunyai dokumentasi untuk kasus yang tipikal untuk membantu proses 	<ul style="list-style-type: none"> • Proses yang panjang dan rumit • Butuh penyesuaian untuk tahap <i>data preparation</i> dan <i>modelling</i> tergantung dari pendekatan yang digunakan.

	<ul style="list-style-type: none">• Mendukung berbagai teknik <i>Data Mining</i>.	
--	---	--

Berdasarkan tabel 3.2., dapat disimpulkan CRISP-DM merupakan metode yang sesuai dengan kebutuhan penelitian dikarenakan SEMMA hanya berfokus pada *software SAS Enterprise Miner Tool* sedangkan pada penelitian ini akan menggunakan *software Jupyter Notebook*. Selain itu juga, tidak ada pembahasan *business understanding* atau tujuan dari penelitian pada tahap SEMMA.

3.4. Alur Penelitian



Gambar 3.2. Alur Penelitian

Penelitian ini akan membuat *model deep learning* dengan arsitektur CNN dan mencari perbandingan model dengan akurasi yang terbaik. Berikut tahap-tahap yang akan diimplementasikan seperti pada gambar 3.2 :

3.3.1. *Business Understanding*

Tahapan ini mendeskripsikan latar belakang dan tujuan dari kegiatan penelitian yaitu pengenalan emosi berdasarkan ekspresi wajah menggunakan *Convolutional Neural Network* dimana model yang dihasilkan sesuai dengan tujuan yang ada pada penelitian.

3.3.2. *Data Understanding*

Pada tahap ini, *dataset* yang digunakan akan diperjelas mengenai isi dan jenis gambar yang akan digunakan. Pada tahap ini, ada beberapa hal yang perlu dilakukan seperti mengumpulkan data awal, mendeskripsikan data, mengeksplorasi data, dan memverifikasi kualitas data [28].

3.3.3. *Data Preparation*

Tahap ini mempersiapkan data yang akan digunakan oleh model. Tahap ini juga perlu analisa yang jelas dan pengetahuan yang cukup banyak untuk menentukan preparasi yang diperlukan pada penelitian. Penelitian ini juga tidak melakukan *cleansing data* dikarenakan data didapat dari penelitian sebelumnya yang sudah disiapkan [25]. Berikut tahap-tahap preparasi yang akan dilakukan:

1. *Load Data*

Load Data merupakan tahapan yang dilakukan untuk membaca data dari format atau bentuk lain ke format yang dibutuhkan saat melakukan *training* model. *Dataset* yang digunakan mempunyai data berbentuk *file* teks seperti pada gambar 3.3., oleh karena itu data teks ini diubah menjadi bentuk *dataframe*.

```
train_00002.jpg 5
train_00003.jpg 4
train_00004.jpg 4
train_00005.jpg 5
train_00006.jpg 1
train_00007.jpg 5
train_00008.jpg 4
train_00009.jpg 4
train_00010.jpg 1
train_00011.jpg 4
train_00012.jpg 1
```

Gambar 3.3. Data Teks RAF-DB

2. *Class Weight*

Class Weight merupakan salah satu cara untuk menangani *class imbalance* yang terjadi pada *dataset*, dimana *class imbalance* merupakan masalah yang muncul jika jumlah masing-masing klasifikasi tidak seimbang. *Dataset* RAF-DB merupakan *dataset* yang mempunyai masalah *class imbalance* dikarenakan jumlah masing-masing jenis ekspresi berbeda jauh seperti pada tabel 3.1. Berikut rumus yang digunakan untuk menghitung *Class Weight* :

$$Class\ Weight = \frac{1}{sample} * \frac{total}{2}$$

Rumus 3. 1. Rumus Class Weight

Berdasarkan rumus 3.1., *sample* merupakan jumlah klasifikasi atau jenis yang akan dihitung bobotnya. Lalu total merupakan jumlah total semua data yang akan di *training*. Lalu total dibagi 2 untuk membantu menjaga *loss* dengan besaran yang sama.

3. Data Augmentation

Data Augmentation merupakan teknik untuk manipulasi sebuah data tanpa kehilangan inti atau arti dari data tersebut. Berikut teknik *data augmentation* yang digunakan pada penelitian ini di tabel 3.3 [23]:

Tabel 3.3. Teknik Data Augmentasi

No.	Teknik	Fungsi
1.	<i>Flip</i>	Gambar di putar balik secara horizontal atau vertikal.
2.	<i>Rotation</i>	Gambar diputar berdasarkan angka derajat yang ditentukan.
3.	<i>Zoom</i>	Gambar di perbesar atau diperkecil.
4.	<i>Shift</i>	Gambar bergeser berdasarkan axis secara horizontal atau vertikal
5.	<i>Shear</i>	Gambar dimiringkan dan diregangkan berdasarkan axis.

3.3.4. *Modelling*

Pada tahap ini, akan dibuatnya model untuk pengenalan ekspresi wajah menggunakan arsitektur CNN dengan 2 arsitektur yaitu DenseNet dan Xception. DenseNet merupakan arsitektur yang populer dikarenakan hasil akurasi mereka yang melebihi standar model lainnya pada proses pengenalan objek atau *image processing* [29]. Xception digunakan dikarenakan hasil akurasi terbaik yang didapat menggunakan imageNet *dataset* dibandingkan dengan *arsitektur* lain untuk deteksi gambar [10]. Berikut tahapan model yang akan dilakukan :

1. *Create CNN Model*

Penelitian ini menggunakan metode *Deep Learning* menggunakan arsitektur CNN. Alasan utama penggunaan CNN adalah fitur *weight sharing* dimana jumlah parameter yang di *train* mengurang dan dapat mengurangi *overfitting* pada model [23]. Selain itu juga, untuk implementasi model dalam skala besar CNN lebih mudah penggunaannya dibanding *neural network* lainnya. [23]. Berikut arsitektur yang akan digunakan pada penelitian ini adalah Xception dan DenseNet.

2. *Monitoring Model Performance*

Pada tahapan ini, dilakukannya konfigurasi *monitoring* terlebih dahulu sebelum *training model*. Konfigurasi ini dilakukan untuk

mencegah terjadinya kesalahan atau *overfitting* pada model. Berikut parameter yang digunakan untuk *monitoring* :

a. *Learning Rate*

Learning rate merupakan salah satu parameter yang digunakan untuk menghitung bobot nilai koreksi pada saat proses *training*. Nilai *learning rate* mempunyai cakupan dari 0 sampai dengan 1. Semakin besar nilai *learning rate* maka semakin berkurang ketelitian pada model dan juga sebaliknya jika nilai *learning rate* semakin kecil [30]. Oleh karena itu digunakan *learning rate scheduler*, untuk menyesuaikan nilai *learning rate* saat melakukan *training*. Untuk penelitian ini akan menggunakan rumus *step decay* sebagai *learning rate scheduler*nya seperti pada rumus 3.1. *Step Decay* ini menurunkan *learning rate*-nya dengan faktor setiap beberapa *epoch*. Pada rumus 3.2, lr_0 merupakan nilai *learning rate* awalnya, lalu *drop* itu merupakan seberapa banyak penurunan yang dilakukan dan terakhir adalah *epoch drop* dimana *learning rate* akan turun setiap banyak nilai *epoch drop* tersebut.

$$learning\ rate = lr_0 * drop^{\lfloor \frac{epoch}{epochs\ drop} \rfloor}$$

Rumus 3. 2. Rumus *Step Decay*

b. *Early Stopping*

Early Stopping merupakan teknik yang digunakan untuk menghentikan *training model* untuk mencegah *overfitting model* [30]. *Early Stopping* ini dipanggil dengan menggunakan fungsi *callbacks* pada *Library Keras*.

c. *Model Checkpoint*

Model Checkpoint merupakan strategi yang digunakan untuk menyimpan model berdasarkan parameter yang ditentukan. Pada penelitian ini, model *checkpoint* digunakan untuk menyimpan model dengan akurasi validasi terbaik, sehingga model yang disimpan ini bisa memuat bobot model dan dilanjutkan lagi dengan model yang sama untuk meningkatkan akurasi [30]. *Model Checkpoint* ini juga dipanggil dengan menggunakan fungsi *callbacks* pada *Library Keras*.

d. *Callbacks*

Callbacks merupakan parameter yang disediakan oleh *Library TensorFlow* dan *Keras* untuk memantau kinerja pada parameter-parameter tertentu dalam menjalankan *training* model. *Callbacks* yang dijalankan ini bergantung kepada parameter yang ditentukan untuk eksekusi [30].

3. *Train Model*

Pada tahapan ini, model yang sudah dibangun arsitekturnya bisa di *training* dengan *dataset* RAF-DB. Model di *training* sebanyak

dengan *epoch* yang ditentukan. Jika model berhenti di tengah proses *training*, model dapat dijalankan lagi dari awal dan menggunakan model *checkpoint* untuk memuat bobot model terakhir dengan akurasi yang paling tinggi.

3.3.5. *Evaluation*

Tahap ini melakukan interpretasi terhadap hasil *modelling* yang dilakukan. Evaluasi dilakukan pada model untuk menentukan apakah model ini sesuai dengan kebutuhan atau tidak. Evaluasi model yang dilakukan akan berdasarkan akurasi dan *confusion matrix*.

3.3.6. *Deployment*

Pada tahap *deployment*, model yang dibuat akan di *demo* dengan *Library OpenCV* dan mendeteksi wajah menggunakan kamera pada *laptop*. Ekspresi yang muncul pada wajah akan dideteksi dan dimunculkan juga hasil deteksi tersebut pada *video* secara *real time*.

3.5. *Tools yang digunakan*

Penelitian ini akan menggunakan pemrograman *Python* dikarenakan *Python* mempunyai *Library* yang mendukung dalam pembuatan model *deep learning*. Berikut *Library* yang akan digunakan :

Tabel 3.4. *Library Python* yang digunakan

<i>Library</i>	Fungsi
<i>Tensorflow</i>	Untuk pembuatan model <i>deep learning</i> dengan arsitektur DenseNet dan Xception
Keras	Untuk pembuatan model <i>deep learning</i> dengan arsitektur DenseNet dan Xception

OpenCV	<i>Pre-processing</i> data gambar dan deteksi <i>facial landmark</i> pada gambar.
Pandas	<i>Library</i> digunakan untuk membaca data yang digunakan menjadi bentuk <i>dataframe</i>
sklearn	Digunakan untuk pembuatan <i>confusion matrix</i> .