

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Teknologi *Artificial Intelligence* (AI) berkembang pesat khususnya pada *Machine Learning*. *Machine Learning* merubah cara pemrograman yang mana biasanya melakukan program dengan cara *Traditional Programming* dimana memasukkan *rules* dan data pada mesin dahulu kemudian baru bisa menghasilkan output atau *answer*. Dengan cara ini mengharuskan programmer membuat *rules* sendiri agar nantinya mesin dapat mengikuti *rules* tersebut. Sedangkan pada *Machine Learning* programmer hanya perlu memasukkan data sekaligus label atau *answer* pada mesin, kemudian mesin akan menganalisa dari data dan label tersebut sehingga menghasilkan sebuah pola yang nantinya ditampung pada model yang telah dilatih.

Dengan hal ini programmer tidak perlu memikirkan *rules* sendiri yang biasanya menguras pikiran dan *code* pun menjadi berpuluh-puluh baris hanya untuk klasifikasi saja. Dengan menggunakan *Machine Learning* permasalahan klasifikasi akan menjadi efisien, namun metode ini dapat menimbulkan masalah baru yaitu *overfitting*. Suatu model dapat dikatakan *overfitting* ketika mesin sangat baik dalam mengenali data yang ada pada direktori *training* namun kurang baik dalam mengenal data validasi atau data uji yang belum dikenalnya. Hal ini terjadi karena mesin sudah sangat terbiasa atau terpatok pada data *trainingnya*.

Dari permasalahan tersebut dapat dikatakan dataset dan model latih menjadi salah-satu komponen yang sangat penting dalam mendukung tercapainya ketepatan yang akurat (tidak *overfitting*), tanpa harus bergantung pada jumlah *epoch* yang sangat banyak pada saat proses *training* model. Berdasarkan pengujian [1] pada data *training* dan data validasi dapat dimanfaatkan untuk dikombinasikan pada saat proses training model, dengan hal ini nantinya penguji dapat memonitoring apakah terjadi *overfitting* pada model atau tidak. Kemudian pada pengujian [2] dan

[3] menggunakan metode arsitektur CNN dalam pengujiannya. Hasil yang didapat akurasi seiring *epoch* bertambah meningkat secara signifikan, namun pada hal ini ternyata *epoch* yang diberikan pada saat *training* sangat banyak yaitu 10.000 *epoch* dan 5000 *epoch*. Selain memakan waktu saat proses *training* hal ini pun sangat rentan untuk menyebabkan terjadinya *overfitting* dikarenakan model latih akan terbiasa memprediksi terlalu spesifik pada data training, akibatnya model akan sulit memprediksi data validasi yang lebih general.

Namun setelah mereview [4] dan [5] hasilnya memang terlihat baik yaitu 99,68% akurasi pada data *training* dan 83,67% pada data validasi. Walau terdapat sedikit *overfitting* namun dapat disimpulkan bahwa metode CNN berfungsi dengan sangat baik ketika digunakan untuk klasifikasi gambar atau citra. Pada [6] dibuktikan bahwa proses *Image Augmentation* dibutuhkan untuk memanipulasi data, dibanding melakukan *epoch* yang berlebihan. Penguji dapat menggantinya dengan memperbanyak variasi pada data *training* dengan memanipulasi data, agar jumlah *epoch* dapat diminimalisir tanpa mengurangi tingkat ketepatan yang signifikan. Dan menurut [7] *optimizer* yang paling cocok untuk kasus klasifikasi gambar menggunakan CNN yaitu *Adam Optimizer*. Dimana pada pengujiannya *Adam Optimizer* sedikit lebih baik dibandingkan dengan *stochastic optimization method* lainnya. Selain itu CNN memiliki *layer convolution* dan *maxpooling* yang sangat baik dalam pengoptimalan data *training* yang jumlahnya banyak, sehingga dapat dimungkinkan menghasilkan model latih yang tidak *overfitting* [8].

Pada penelitian ini data *training* akan dilakukan *pre-processing* terlebih dahulu agar dimensi dan pola pada data lebih serasi. Dikarenakan dataset yang digunakan berupa gambar maka tahap *pre-processing* dapat dilakukan dengan *Image Augmentation* menggunakan *ImageDataGenerator*. Kelebihan dari *ImageDataGenerator* yaitu dapat melakukan pelabelan data secara otomatis, manipulasi data seperti merubah rotasi, *mirroring*, *zooming*, mengompres dimensi agar pola pada data *training* lebih searasi, serta pembagian direktori antara data *training* dan data validasi. Pada model latihnya menggunakan arsitektur CNN dimana dengan ini mesin dapat mengenali gambar tidak hanya berdasarkan pola

pada keseluruhan piksel namun juga berdasarkan atribut yang ada pada gambar tersebut. Lalu model dapat di-*compile* dengan *Adam Optimizer* karena ia dapat mencapai hasil yang baik dengan cepat. Hasil empiris menunjukkan bahwa *Adam Optimizer* bekerja selangkah lebih baik dibandingkan dengan *stochastic optimization method* lainnya. Serta membatasi *epoch* yang dilakukan dengan melakukan monitoring pada tiap *epoch* menggunakan *EarlyStopping* callback pada validasi *loss*, agar dapat mengurangi resiko terjadinya *overfitting* pada model *Binary Classification* dengan 2 label yang akan diprediksi yaitu Kucing dan Anjing.

### 1.2 Perumusan Masalah

Dari latar belakang yang dipaparkan diatas. Dapat dirumuskan permasalahannya adalah apakah dengan menerapkan *Image Augmentation* menggunakan *ImageDataGenerator*, membangun arsitektur model dengan CNN, dan menggunakan *Adam Optimizer*, serta membatasi *epoch* menggunakan *EarlyStopping* callback, dapat mengurangi resiko terjadinya *overfitting* pada model *Binary Classification*. Dengan 2 label yang akan diprediksi yaitu Kucing dan Anjing.

### 1.3 Tujuan Penelitian

Adapun maksud dan tujuan dari penelitian ini adalah mengetahui apakah dengan menerapkan :

1. *Pre-processing* data dengan *Image Augmentation* menggunakan *ImageDataGenerator*.
2. Membangun arsitektur model dengan CNN.
3. Menggunakan *Adam Optimizer* untuk proses *compile* model.
4. Serta membatasi *epoch* yang dilakukan dengan melakukan monitoring pada tiap *epoch* menggunakan *EarlyStopping* callback pada validasi *loss*.

Dapat mengurangi resiko terjadinya *overfitting* pada model *Binary Classification*, dengan 2 label yang akan diprediksi yaitu Kucing dan Anjing.

#### 1.4 Batasan Masalah

Adapun batasan masalah dalam penelitian ini sebagai berikut :

1. Pendeteksian dilakukan pada gambar kucing dan anjing, dalam satu gambar tidak memiliki gambar kucing serta anjing sekaligus.
2. Jumlah dataset yang digunakan sebanyak 14.047 gambar dengan 2 kategori yaitu Kucing sebanyak 7.023 gambar dan Anjing sebanyak 7.024 gambar.
3. Pembagian dataset pada direktori *training* sebesar 80% dan validasi sebesar 20%.
4. Pengujian berfokus pada *Binary Classification* dimana hanya terdapat 2 *class* atau label yang dilatih dan diprediksi.
5. Optimizer yang digunakan yaitu *Adam Optimizer*.
6. Hasil dari penelitian ini berfokus untuk mendapatkan model yang tidak mengalami *overfitting*.

#### 1.5 Manfaat Penelitian

Dengan dilakukannya penelitian ini diharapkan mampu memberikan manfaat berupa diketahui apakah dengan menerapkan metode yang dipaparkan diatas dapat mengurangi resiko terjadinya *overfitting* pada model *Binary Classification* dengan 2 label yang akan diprediksi yaitu Kucing dan Anjing. Serta dapat dijadikan referensi untuk penelitian lain yang berkaitan.