

BAB 3

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi Penelitian yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Memilih Masalah

Pada proses ini, dilakukan pemilihan masalah. Masalah yang dipilih adalah seberapa akurat algoritma CNN dalam mendeteksi ekspresi wajah dan bagaimana cara implementasinya.

2. Merumuskan Masalah

Berdasarkan topik permasalahan yang dipilih, dilakukan analisis untuk membuat sistem yang sesuai. Diputuskanlah untuk membuat website untuk mendeteksi ekspresi wajah.

3. Analisis Kebutuhan

Dilakukan analisis kebutuhan yang disesuaikan dalam masalah yang ada. Kebutuhan tersebut adalah *dataset* dan *library* yang digunakan untuk menyelesaikan masalah.

4. Implementasi Algoritma

Algoritma akan diimplementasikan pada tahap ini. Perancangan akan meliputi pembuatan *flowchart*, dan *training dataset*. Setelah selesai dirancang, proses selanjutnya adalah pembangunan hasil implementasi sistem ke aplikasi web. Aplikasi ini akan berisi kamera yang akan mendeteksi ekspresi wajah pengguna.

5. Ujicoba Aplikasi

Setelah aplikasi selesai, tahap yang akan dilakukan selanjutnya adalah ujicoba hasil dari implementasi. Hal yang akan diuji ialah akurasi dari sistem dalam mendeteksi ekspresi wajah.

6. Penulisan Laporan

Tahap akhir dari rancang bangun ini adalah penulisan laporan berupa skripsi yang akan berisi hasil dari keseluruhan penelitian. Penulisan akan dilakukan dengan terstruktur dan menggunakan bahasa Indonesia yang baik dan benar dan sesuai dengan kaidah penulisan skripsi.

3.2 Perancangan Sistem

Sistem dirancang dengan *flowchart*, dan rancangan proses *loading* model untuk menampilkannya dalam bentuk kamera.

3.2.1 Flowchart dan Dataset

A. Dataset

Dataset ekspresi wajah yang digunakan adalah FER 2013. *Dataset* tersebut berisikan informasi berupa gambar wajah *grayscale* dengan skala 48x48 yang memiliki 7 kategori ekspresi yaitu marah (*angry*), jijik (*disgust*), takut (*fear*), senang (*happy*), sedih (*sad*), kaget (*surprised*), dan netral (*neutral*). *Dataset* tersebut disajikan didalam sebuah *file* dengan format *.csv* yang berisikan tabel dengan kolom nomor, kode *pixel* yang berbentuk *string* (masing-masing kode *pixel* berjumlah 48x48 dan dipisahkan oleh spasi), kode ekspresi, dan kode penggunaan (Carrier and Courville, 2013). Gambar 3.1 adalah visualisasi dari data *.csv* tersebut.

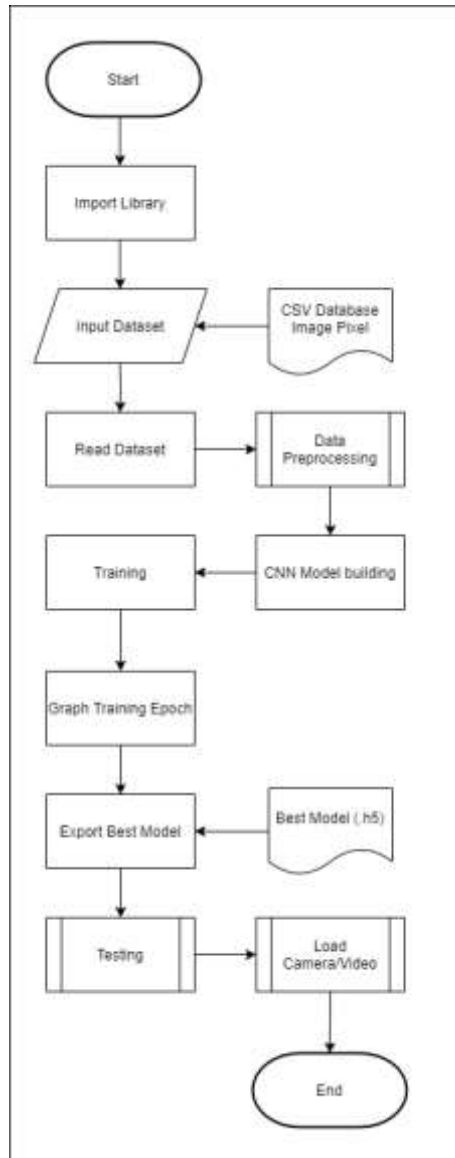
	emotion	pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training
...
35882	6	50 36 17 22 23 29 33 39 34 37 37 37 39 43 48 5...	PrivateTest
35883	3	178 174 172 173 181 188 191 194 196 199 200 20...	PrivateTest
35884	0	17 17 16 23 28 22 19 17 25 26 20 24 31 19 27 9...	PrivateTest
35885	3	30 28 28 29 31 30 42 68 79 81 77 67 67 71 63 6...	PrivateTest
35886	2	19 13 14 12 13 16 21 33 50 57 71 84 97 108 122...	PrivateTest

[35887 rows x 3 columns]

Gambar 3.1 Bentuk dari *dataset FER2013*

B. Flowchart Utama

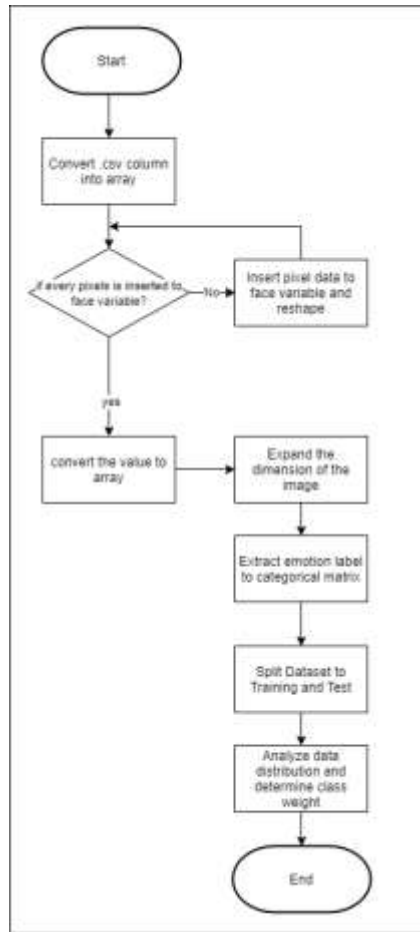
Pada Gambar 3.2 terdapat alur proses CNN pada penelitian ini. Tahap pertama adalah *import library* dan *dataset* yang dibutuhkan. Setelah itu, *dataset* masuk dan di proses. *Dataset* yang digunakan bersumber dari kompetisi *Kaggle* dengan judul “*Challenge in Representation Learning: Facial Expression Recognition Challenge*” yang diadakan pada tahun 2013. *Dataset* yang digunakan bernama “*FER2013*” berupa file dengan format *.csv* yang berisikan *string* dari *pixel* gambar (Carrier and Courville, 2013). *Output* dari proses *training* ini akan berupa data *model* yang paling ideal.



Gambar 3.2 *Flowchart* Utama

C. Flowchart Preprocessing

Gambar 3.3 merupakan *flowchart preprocessing* yang melalui beberapa proses. *Preprocessing* ini bertujuan untuk memindahkan data yang ada pada *dataset* kedalam variabel *array*. Proses ini penting dilakukan karena data yang ada di dalam *dataset* berbentuk berupa *string* berisi kode *pixel* yang dipisahkan oleh spasi.

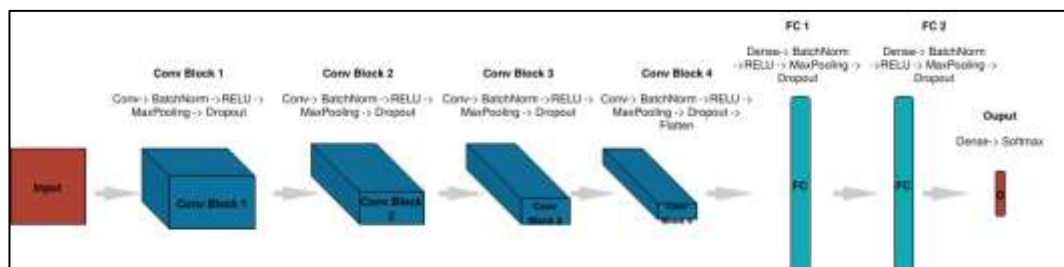


Gambar 3.3 *Flowchart Preprocessing*

Preprocessing dimulai dengan konversi kolom pada *dataset .csv pixels* dan memindahkannya ke variabel *array*, setelah itu dilakukan proses *reshape*. Setelah semua kolom dimasukan ke variabel *array*, akan dilakukan ekspansi dimensi dari gambar tersebut untuk menyelipkan label ekspresi pada tiap tiap *array image*. Pada tahap akhir *preprocessing*, dilakukan ekstraksi label emotion yang dimasukan ke *categorical matrix*. Sebelum data digunakan untuk membangun model, lakukan pengecekan distribusi data untuk menghindari *overfitting* dan *class bias* di mana akan ditentukan sebuah nilai *class weight* sebagai pertimbangan dalam proses *training*.

D. Flowchart Model

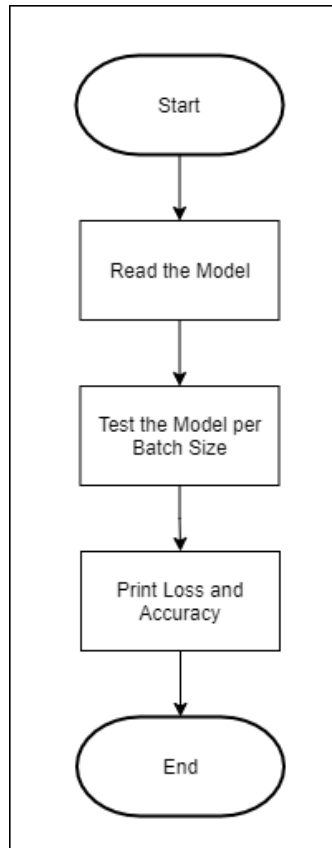
Pada Gambar 3.4 terdapat proses model CNN yang arsitekturnya terinspirasi dari jurnal *Challenge in Representation Learning* (Goodfellow et al., 2015). *Dataset* akan diproses kedalam *model* dengan ukuran (48,48,1) melalui 4 *convolutional layer* dan 2 *fully connected layer*. Tiap *convolutional layer* akan melalui berbagai proses seperti *batch normalization*, *relu activation*, *max pooling*, dan *dropout*. Lalu setelah diproses oleh *convolutional layer*, *dataset* akan melalui *fully connected layer* yang akan mengolah data sehingga bisa diklasifikasikan.



Gambar 3.4 Alur Proses Model CNN
(Goodfellow et al., 2015)

E. Flowchart Testing

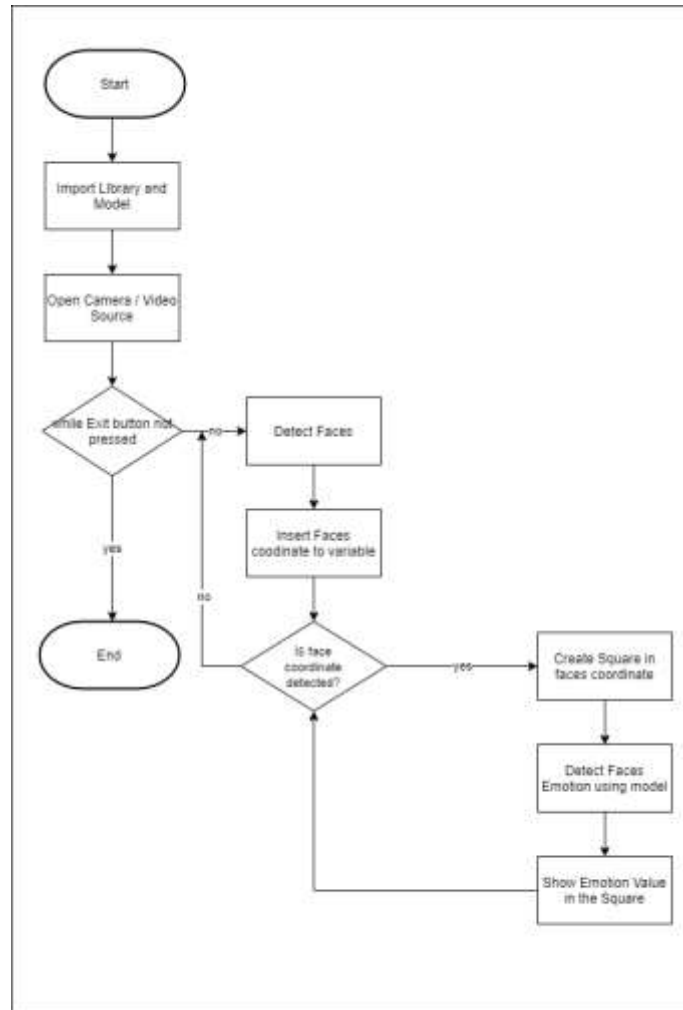
Setelah *model* dihasilkan melalui proses *training*, Gambar 3.5 menunjukkan proses *testing* dari *model* tersebut. Proses *testing* ini sendiri akan menggunakan data validasi untuk menguji *model* yang telah dilatih.



Gambar 3.5 *Flowchart Testing*

F. Flowchart Camera/Video Source

Gambar 3.6 menunjukkan proses implementasi *model* secara *real-time*. Proses ini sendiri akan langsung menampilkan kamera atau video yang bisa mendeteksi wajah dan mengklasifikasikan ekspresi wajah. Untuk klasifikasi wajah dari kamera atau video digunakan *haar cascades classifier* yang akan selalu aktif mencari wajah yang ditampilkan melalui kamera. Ketika wajah terdeteksi oleh kamera, akan dibuat sebuah persegi dan teks untuk menentukan ekspresi wajah yang terdeteksi.



Gambar 3.6 *Flowchart* Kamera atau *Video Source*