

BAB III

METODOLOGI PENELITIAN

3.1 SUBJEK DAN OBJEK PENELITIAN

Subjek pada penelitian ini yang akan dilakukan adalah rambu lalu lintas dengan menggunakan 43 kelas. Objek pada penelitian ini menjadi sasaran penelitian, objek tersebut menentukan klasifikasi citra rambu yang berupa data gambar.

3.2 ALAT DAN BAHAN

Pada penelitian ini terdapat beberapa alat dan bahan yang digunakan sebagai berikut:

3.2.1 Perangkat keras

Dibutuhkannya perangkat keras yang akan digunakan dalam penelitian adalah sebagai berikut:

- a. *Laptop Asus X40IU, series processor AMD-E2, Ram 2 GB, SSD 120 GB*
- b. Komputer Lab PSD
- c. *Handphone*

3.2.2 Perangkat Lunak

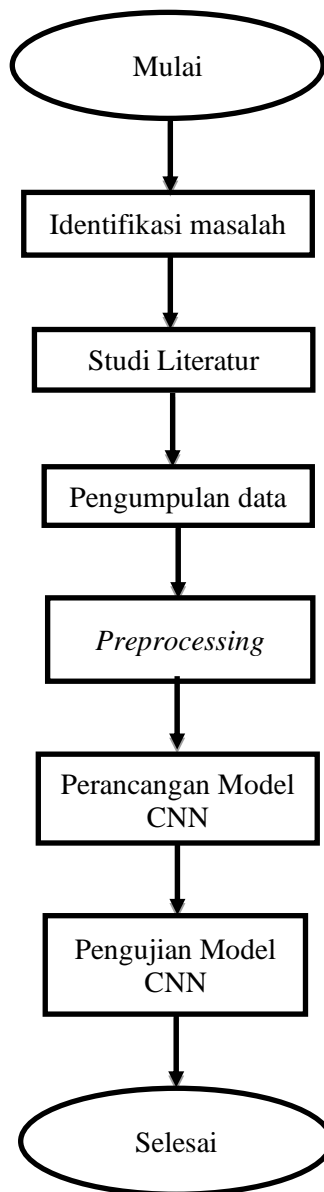
- a. Google colab
- b. Python 3.10
- c. Google Chrome
- d. Kaggle

3.2.3 Dataset

Dataset yang akan digunakan pada penelitian ini merupakan sekumpulan gambar yang diperoleh dari kaggle dengan link <https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>. Data tersebut merupakan dataset rambu dari Jerman yang bernama *German Traffic Sign Recognition Benchmark (GTSRB)*, terdapat 43 kelas citra rambu pada dataset tersebut[53]. Jumlah keseluruhan data untuk penelitian ini sebanyak 51.850 citra.

3.3 DIAGRAM ALUR PENELITIAN

Secara umum, pada proses klasifikasi gambar terutama pada rambu lalu lintas dengan menggunakan algoritma *Convolutional Neural Network* (CNN) diawali dengan mengidentifikasi masalah dari penelitian terdahulu terkait dengan rambu lalu lintas. Dapat dilihat proses pada *flowchart* pada gambar 3.1 dibawah ini.



Gambar 3.1 *Flow Chart* Alur Penelitian

3.3.1 Identifikasi Masalah

Setelah melihat alur penelitian diatas membutuhkan beberapa tahap yang diawali dengan identifikasi masalah, dimana pada tahap ini

mengidentifikasi masalah mengenai rambu lalu lintas, dimana agar dapat mengenali pada rambu lalu lintas, sehingga tidak terjadi hal-hal yang tidak diinginkan saat berkendara disaat bertambahnya pengguna jalan yang semakin meningkat.

3.3.2 Studi Literatur

Tahap selanjutnya melakukan studi literatur yang berkaitan dengan permasalahan dan metode yang akan digunakan, dengan melakukan pengumpulan data yang berhubungan dengan permasalahan yaitu melakukan perbandingan klasifikasi penanda rambu lalu lintas. Algoritma yang akan digunakan yaitu *Convolutional Neural Network* (CNN). Pada tahap ini dilakukan agar memperkuat permasalahan yang dibahas pada penelitian ini.

3.3.3 Pengumpulan Data

Pada tahap pengumpulan data ini, data yang akan digunakan merupakan dataset yang diperoleh dari suatu website yang bernama Kaggle. Dapat diunduh melalui link <https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>. Contoh gambar rambu *German Traffic Sign Recognition Benchmark* (GTSRB) yang dapat di lihat pada gambar 3.2. Dataset yang akan digunakan pada penelitian ini merupakan citra rambu lalu lintas dengan jumlah 12.631 untuk data *testing*, dan 39.209 untuk data *training*. Pada gambar 3.3 merupakan grafik dari jumlah dataset dari jumlah yang kecil hingga jumlah yang terbesar. Pada tabel 3.1 dibawah ini merupakan jumlah data dari setiap kelasnya yang akan digunakan sebelum dilakukannya proses *training* dan *validation*.

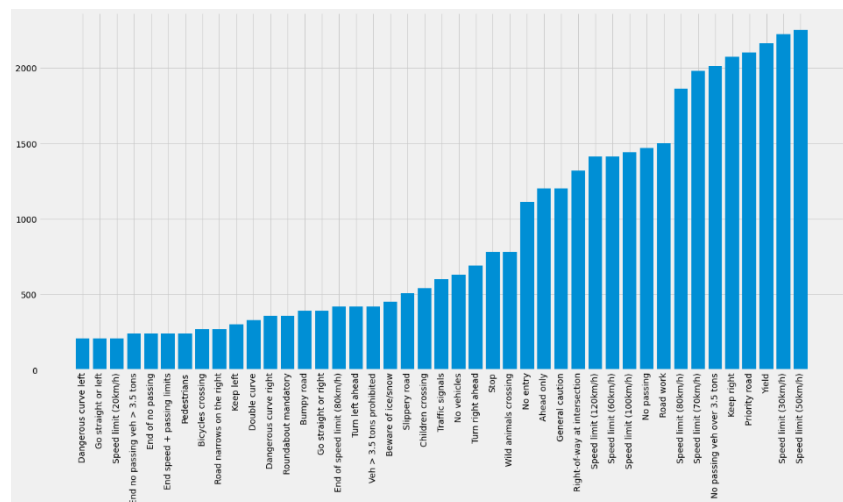
Tabel 3.1 Jumlah Dataset

No	Kelas rambu	Training
0	Batas kecepatan 20km/j	210
1	Batas kecepatan 30km/j	2,220
2	Batas kecepatan 50km/j	2,250
3	Batas kecepatan 60km/j	1,410
4	Batas kecepatan 70km/j	1,980
5	Batas kecepatan 80km/j	1,860

6	Akhir batas kecepatan	420
7	Batas kecepatan 100km/j	1,440
8	Batas kecepatan 120km/j	1,410
9	Tidak boleh lewat	1,470
10	Tidak boleh yang melewati lebih dari 3,5 ton	2,010
11	Jalan ke kanan di persimpangan berikutnya	1,320
12	Jalan utama	2,100
13	Yield	2,160
14	Stop	780
15	Tidak ada kendaraan	630
16	Tidak ada truk	420
17	No entry	1,110
18	Hati-hati	1,200
19	Kurva berbahaya kiri	210
20	Kurva berbahaya kanan	360
21	Jalan berliku	330
22	Jalan bergelombang	390
23	Jalan licin	510
24	Jalan menyempit di sebelah kanan	270
25	Pekerjaan jalan	1,500
26	Lampu lalu lintas	600
27	Jalan kaki	240
28	Penyeberangan anak-anak	540
29	Penyeberangan sepeda	270
30	Waspada terhadap salju	450
31	Pelintasan satwa liar	780
32	Akhir batas kecepatan	240
33	Belok kanan	689
34	Belok kiri	420
35	Hanya lurus	1,200
36	Hanya lurus atau kanan	390
37	Hanya lurus atau kiri	210
38	Tetap kanan	2,070
39	Tetap kiri	300
40	Bundaran wajib	360
41	Akhiri batas menyalip	360
42	Akhiri batas menyalip truk > 3.5 ton	240
Jumlah		39.209



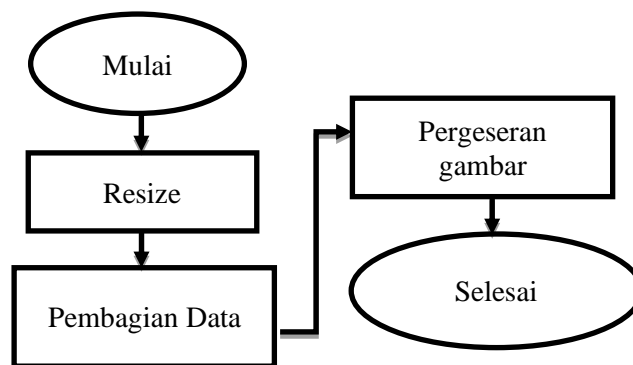
Gambar 3.2 rambu lalu lintas German *Traffic Sign Recognition Benchmark* (GTSRB) gambar per kelas dalam dataset



Gambar 3.3 jumlah dataset pada 43 kelas

3.3.4 *Preprocessing*

Tahap proses *Preprocessing* pada gambar 3.4 dibawah ini digunakan untuk menyiapkan gambar yang akan diproses pada langkah berikutnya.



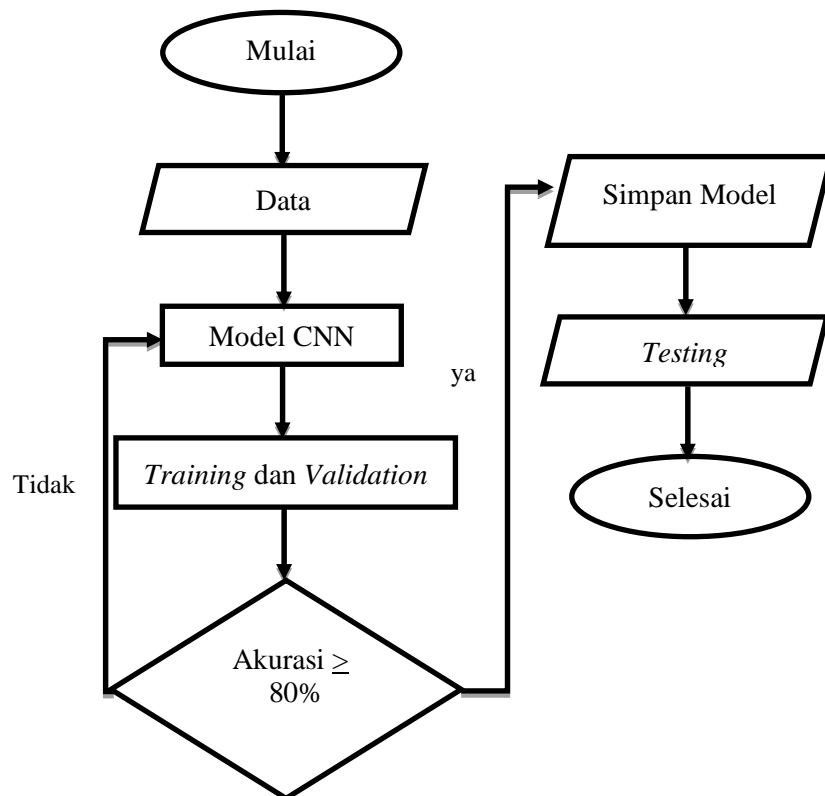
Gambar 3.4 *Preprocessing*

Dimana dalam proses *resize* berfungsi untuk mengubah ukuran pada

gambar agar semua gambar berukuran yang sama diatur dengan ukuran 30 x 30 pixel. Kemudian jumlah data *training* akan dibagi menjadi data *training* dan *validation* dengan pembagian 70% untuk data *training* sedangkan 30% untuk data validasi. Dilakukannya pergeseran gambar tinggi dan lebar gambar sebesar 0.1.

3.3.5 Perancangan model CNN

Berikut ini merupakan proses selanjutnya dari proses *preprocessing*, dapat dilihat pada gambar 3.5. Diawali dengan data yang dimasukkan yaitu berupa citra rambu, dimana perancangan model dengan menggunakan algoritma CNN sebagai pengklasifikasi citra rambu.



Gambar 3.5 *flowchart* perancangan model CNN

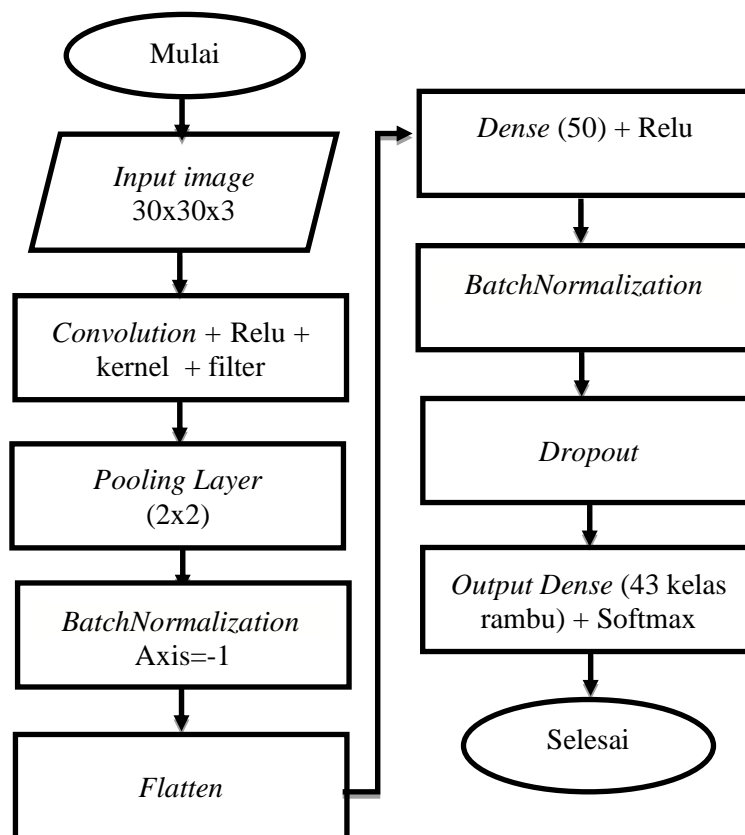
Data tersebut akan diproses *training* atau sering disebut proses *learning* (pembelajaran), guna untuk melatih model CNN yang telah dibuat, sehingga dapat membedakan maupun memahami jenis citra pada rambu lalu lintas sesuai dengan kelasnya masing-masing. Data akan di proses *training* (latih) dengan total keseluruhan sebanyak 39.209, data tersebut akan dibagi menjadi data *training* dan data *validation* (validasi). Untuk data *training* sebanyak 27.446 citra sedangkan data

validation sebanyak 11.763. Data validasi sendiri digunakan untuk memvalidasi model selama poses pelatihan untuk mengetahui layak tidaknya model yang akan digunakan sebelum dilakukanya proses *testing*. Tabel 3.2 berikut merupakan pembagian data *training* dan data *validation*.

Tabel 3.2 Pembagian dataset

Dataset	Jumlah data
Data Training	27.446
Data Validation	11.763
Total	39.209

Jumlah data *training* dan *validation* tersebut sebanyak 39.209. Proses *training* dilakukan mulai dari *epoch* 1 sampai *epoch* 30, sehingga menghasilkan nilai akurasi sesuai tujuan yaitu $\geq 80\%$. Selanjutnya model akan disimpan, kemudian digunakan untuk proses selanjutnya yaitu *testing*.



Gambar 3.6 *flowchart* model CNN

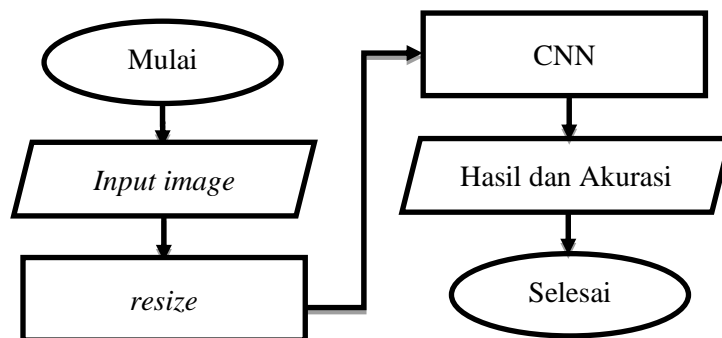
Pada umumnya rancangan arsitektur memiliki 2 tahap, yang pertama tahap *feature learning* adalah Teknik yang memungkinkan sistem dapat berjalan secara otomatis untuk merepresentasikan gambar menjadi *feature* yang berupa angka, langkah ini dimulai dari layer *convolution* dan *maxpooling*. *Input* citra model CNN menggunakan citra yang digunakan berukuran 30x30x3, dimana angka 30 merupakan ukuran *pixel* yang ditentukan sedangkan angka 3 merupakan semacam citra yang mempunyai 3 *channel* warna yaitu *Red*, *Green*, dan *Blue* (RGB). Citra tersebut akan diproses melalui tahap *feature learning*. Pada lapisan konvolusi memiliki beberapa percobaan dengan jumlah filter yaitu 5, 10, dan 15. Filter sendiri merupakan jumlah lapisan pada *layer* konvolusi dengan ukuran tertentu yang dapat ditentukan berbagai variasi. Pada penelitian ini menggunakan ukuran 5x5, 10x10, dan 15x15 ukuran tersebut disebut kernel. Tujuannya untuk membandingkan kinerja atau hasil akurasi dari variasi filter dan ukuran kernelnya. Pada lapisan *Pooling layer* menggunakan *maxpooling* dengan kernel (2x2). *BatchNormalization* biasanya bekerja paling baik setelah fungsi aktivasi. *BatchNormalization* digunakan agar distribusi *input* ke lapisan tertentu tidak mengalami perubahan karena pembaruan parameter dari setiap *batch*. Pada intinya lapisan ini mencoba untuk menyamakan nilai *input* yang selalu berubah pada lapisan sebelumnya selama proses *training*.

Tahap yang kedua *Classification* adalah tahap dimana hasil *feature learning* akan dilakukan proses klasifikasi sesuai dengan *subclass* yang telah ditentukan. Terdapat lapisan *flatten*, dimana pada proses ini mengubah *feature map* hasil dari layer sebelumnya yaitu *pooling layer* ke dalam bentuk vektor. Karena hasil *feature map* sebelumnya masih berbentuk multidimensional *array*, sehingga perlu melakukan “*flatten*” atau *reshape feature map* agar bisa digunakan sebagai *input* dari *fully-connected layer*. Pada *layer fully-connected layer*, terdapat *layer dense* yang berfungsi untuk

menambahkan *hidden layer*. Nilai 50 menandakan nilai neuron yang digunakan pada lapisan ini yang dapat diatur sesuai keinginan masing-masing, dengan aktivasi yang digunakan yaitu Relu. Terdapat *layer BatchNormalization* lagi, kemudian diteruskan ke *layer dropout*. Pada *output* menggunakan *layer dense* dengan aktivasi *softmax*, menggunakan aktivasi tersebut dikarenakan klasifikasi yang digunakan menggunakan lebih dari dua kelas.

3.3.6 Pengujian model CNN

Pada gambar 3.7 dijelaskan proses *testing*, guna untuk menguji ketepatan klasifikasi pada model CNN yang telah dibuat pada proses *training*.



Gambar 3.7 Alur *Testing*

Dimulai dari memasukkan *image* baru yaitu data *test* sebanyak 12.631 citra yang digunakan untuk proses pengujian, kemudian terdapat proses *resize* dengan menyesuaikan ukuran citra menjadi 30x30 *pixel*, citra tersebut akan diklasifikasikan oleh CNN.

Output dari proses klasifikasi CNN merupakan citra rambu dengan total keseluruhan sebanyak 12.631 dari 43 kelas, kemudian menghasilkan tingkat akurasi yang dihasilkan dari perhitungan *confusion matrix*. Akurasi itu sendiri merupakan kemampuan dalam pengukuran yang berupa variabel untuk menilai tolak ukur dari nilai sebenarnya dari model CNN pada klasifikasi rambu lalu lintas.

