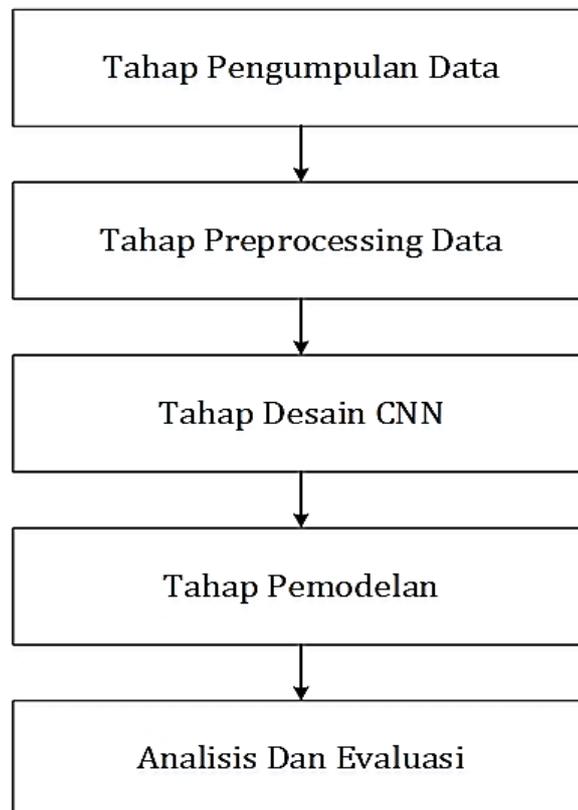


BAB III METODOLOGI PENELITIAN

3.1 Alur Penelitian

Tanda nomor atau plat pada kendaraan merupakan bagian terpenting yang dapat digunakan dalam mengenali kendaraan dengan pemiliknya. Hal ini juga dilakukan pada lokasi parkir di Institut Teknologi Sumatera dimana saat mahasiswa keluar dari lokasi parkir dengan membawa kendaraan harus melakukan pengecekan nomor polisi dengan STNK. Namun seiring dengan bertambahnya mahasiswa di ITERA tentunya untuk pada lokasi parkir sendiri mengalami permasalahan terlebih lagi jika satpam yang melakukan pengecekan STNK dan nomor plat hanya terbatas pada setiap gedung. Selain itu pada proses pengecekan nomor polisi pada lokasi parkir ITERA diharuskan mengeluarkan STNK terlebih dahulu dan itu memakan waktu yang cukup lama jika terjadi antrian kendaraan oleh mahasiswa, oleh karena itu diperlukan sebuah solusi untuk dapat mengatasi permasalahan tersebut.



Gambar 3.1. Alir penelitian.

Adapun untuk solusi yang ditawarkan pada penelitian ini yaitu dengan pengaplikasian *convolutional neural networks* untuk melakukan identifikasi karakter pada plat kendaraan. Untuk itu terdapat tahapan-tahapan yang dilakukan seperti yang terdapat pada Gambar 3.1. Alur penelitian merupakan tahapan-tahapan yang dilakukan pada penelitian ini. Alur penelitian ini juga sebuah prosedur serta teknik dalam perencanaan penelitian yang berguna sebagai panduan untuk membangun strategi yang menghasilkan *model* atau *blue print* penelitian. Gambar 3.1 merupakan tahapan-tahapan alur dari penelitian ini yang mana terdiri dari 5 tahap diantaranya yaitu survei, pengumpulan data, *preprocessing*, tahap implementasi, tahap analisis dan evaluasi.

3.1.1 Tahap Pengumpulan Data

Tahap pengumpulan data merupakan tahap yang dilakukan untuk mendapatkan data citra dari tanda nomor kendaraan guna membantu proses pelatihan dan pengujian pada penelitian ini. Pada tahap ini pengumpulan data dilakukan dengan cara mengambil gambar dengan kamera digital atau kamera smartphone pada tanda nomor kendaraan yang terdapat di Provinsi Lampung. Untuk jumlah data yang digunakan pada penelitian ini adalah 600 gambar, dimana 500 gambar digunakan untuk pelatihan dan 100 gambar untuk pengujian.

Adapun beberapa kriteria yang dilakukan pada tahap pengumpulan data sebagai berikut :

1. Data citra yang diambil dilakukan dari 2 sudut yang sama, artinya data citra yang diambil pada kendaraan merupakan bagian belakang atau bagian depan dari kendaraan.
2. Untuk semua data citra yang diambil memiliki plat nomor yang berbeda-beda.
3. Data citra yang diambil adalah tanda nomor kendaraan berwarna merah, hitam dan kuning.
4. Jarak pada saat pengambilan data tidak lebih dari 5 m.
5. Intensitas cahaya mempengaruhi dalam pengambilan data, artinya pengambilan data saat terdapat cahaya yang cukup.

6. Pada saat pengambilan data dilakukan dengan posisi kamera yang bergerak untuk memotret objek plat kendaraan yang diam.

3.1.2 Tahap *Preprocessing* Data

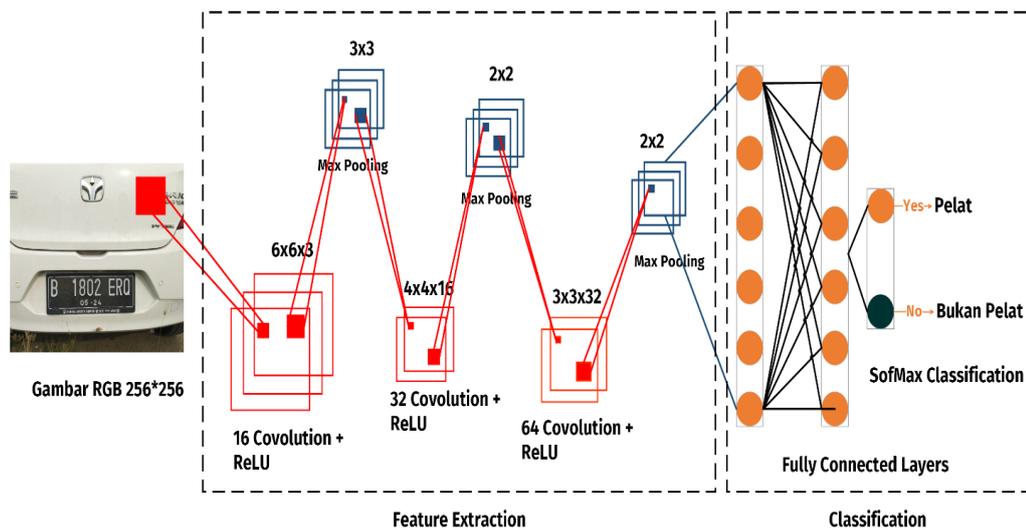
Tahap *preprocessing* merupakan tahap yang dilakukan untuk pemrosesan data citra asli sebelum data tersebut diolah dengan *convolutional neural networks*. Pada tahap ini data citra yang diperoleh dari akuisisi dilakukan proses pemilihan atau penyortiran. Hal ini bertujuan untuk mengelompokkan data citra yang memiliki kualitas baik dapat digunakan untuk proses pelatihan. Pemilihan data ini dilakukan sampai mendapatkan 600 data citra yang berguna untuk tahap pelatihan dan pengujian.

3.1.3 Tahap Desain CNN

Pada penelitian sebelumnya terkait dengan penyelesaian permasalahan identifikasi nomor polisi terdapat beberapa topologi CNN yang digunakan. Adapun topologi CNN yang digunakan oleh Novyantika (2018) yaitu menggunakan CNN dengan 2 *layer* konvolusi, dengan masing-masing ukuran konvolusi $6*6*3$, $4*4*16$ dan dengan ukuran data input $300*300*3$, dengan topologi CNN ini berhasil digunakan untuk mendeteksi objek plat. Lalu topologi CNN yang digunakan oleh Arsenovic dkk (2017) untuk mendeteksi plat yaitu menggunakan YOLO V3 yang merupakan pengembangan dari CNN.

Berdasarkan penelitian terkait, pada penelitian ini di desain CNN dengan menggunakan topologi yang hampir sama dengan topologi CNN oleh Novyantika (2018), hanya saja pada penelitian ini ditambahkan 1 *layer* konvolusi yang mana diharapkan itu dapat berguna untuk mendeteksi plat yang memiliki warna berbeda. Adapun desain dari CNN dapat dilihat pada Gambar 3.2. Arsitektur CNN pada Gambar 3.3 memiliki beberapa bagian yaitu data *input*, 3 *convolution layers + activation* (ReLU), *max pooling*, *fully connected layer*, *classification*. Data input merupakan gambar yang telah dilakukan *preprocessing* dan digunakan untuk pelatihan CNN tersebut. Pada Gambar 3.2 diberikan data citra dengan ukuran $256*256*3$ yang artinya data tersebut memiliki ukuran piksel $256*256$ dengan dengan ruang warna RGB sejumlah 3.

Selanjutnya setelah gambar di input dilakukan proses konvolusi layers dengan ukuran 6*6. Selanjutnya dilakukan *max pooling* 3*3 sehingga menghasilkan layer dengan ukuran 4*4. Kemudian dilakukan proses *max pooling* dengan ukuran 2*2 dan dihasilkan layers 3*3. Setelah itu dilakukan *max pooling* sekali lagi dan dilakukan proses *fully connected layers*. Pada *fully connected layers* ini melakukan proses untuk menyatukan semua layers yang telah diprediksi sebagai plat untuk kemudian dilakukan proses *softmax classification* untuk menentukan apakah itu benar plat atau bukan.



Gambar 3.2. Arsitektur CNN.

Adapun untuk penjelasan pada Gambar 3.2 sebagai berikut :

1. Convolution Layers

Convolution layers pada CNN merupakan bagian yang utama pada algoritma ini. Konvolusi adalah operasi matematika untuk menggabungkan dua set informasi. Proses konvolusi memanfaatkan apa yang disebut sebagai *filter*. Seperti layaknya gambar, filter memiliki ukuran tinggi, lebar, dan tebal tertentu. *Filter* ini diinisialisasi dengan nilai tertentu dan nilai dari filter inilah yang menjadi parameter yang terus di-*update* dalam proses *learning*.

Untuk memahami cara kerja dari proses konvolusi, digunakan sampel matiks yang merupakan data input dengan ukuran 800x600 berdasarkan Gambar 3.3. untuk proses konvolusi digunakan ukuran 6x6 dan menggunakan kernel atau filter untuk operasi vertical edge detection dengan ukuran 3x3. Pada proses konvolusi ini dilakukan perkalian *dot product* antara matriks ukuran 6x6 dengan kernel 3x3, sehingga diperoleh matriks 4x4.

1	3	0	7	8	5
6	0	0	4	1	1
7	2	1	1	1	3
1	2	4	6	7	9
1	8	4	7	0	4
1	0	2	1	2	3

*

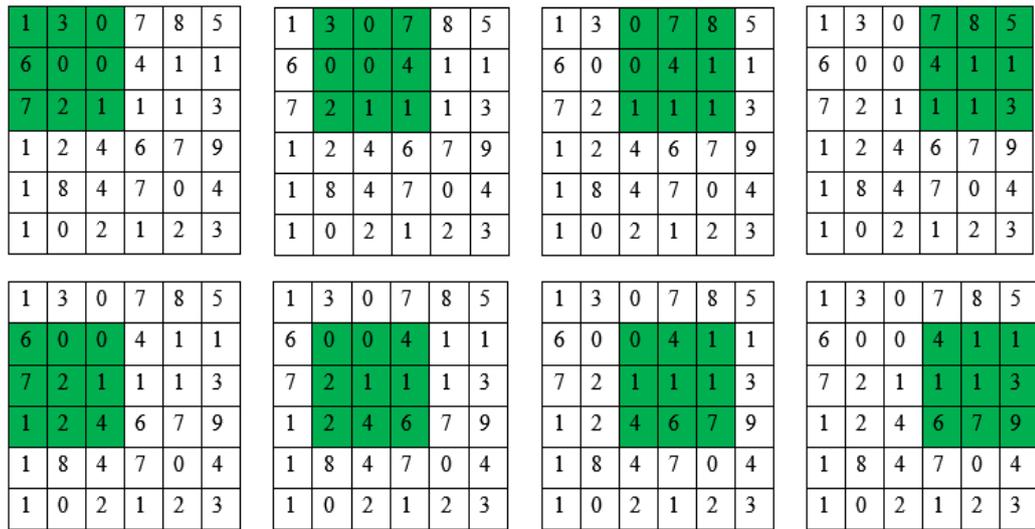
1	0	-1
1	0	-1
1	0	-1

=

13	-8	-9	3
9	-7	-4	-2
0	-2	1	-2
-7	-4	1	-2

Gambar 3.3. Operasi konvolusi.

Dengan menggunakan filter atau kernel 3*3 dan *stride* 1 maka proses perhitungan dapat divisualisasikan pada Gambar 3.4. Proses perhitungan konvolusi dengan ukuran *filter* 3x3 ini dimulai dari sudut kiri atas kemudian dilakukan *sliding window* atau perpindahan dengan *stride* sejumlah 1 sampai pojok kiri bawah. *Filter* yang digunakan pada gambar 3.3 tidak lebih dari satu *set* bobot, yaitu 3x3x3 = 9 + 1 sehingga bias = 10 bobot. Pada masing-masing posisi, jumlah piksel yang dihitung tersebut menggunakan persamaan 2.4 dan kemudian nilai baru diperoleh. Dengan *filter* 3*3 diperoleh matriks baru dengan ukuran 4*4 atau disebut juga sebagai *features map*. Ukuran gambar yang dihasilkan dari proses konvolusi semakin berkurang terus menerus, hal ini tidak begitu baik karena ukurannya dapat menjadi kecil. Selain itu bisa membatasi penggunaan *filter* dengan ukuran besar karena menghasilkan pengurangan ukuran lebih cepat. Untuk mengatasi ini maka digunakan *stride* sebesar 1.



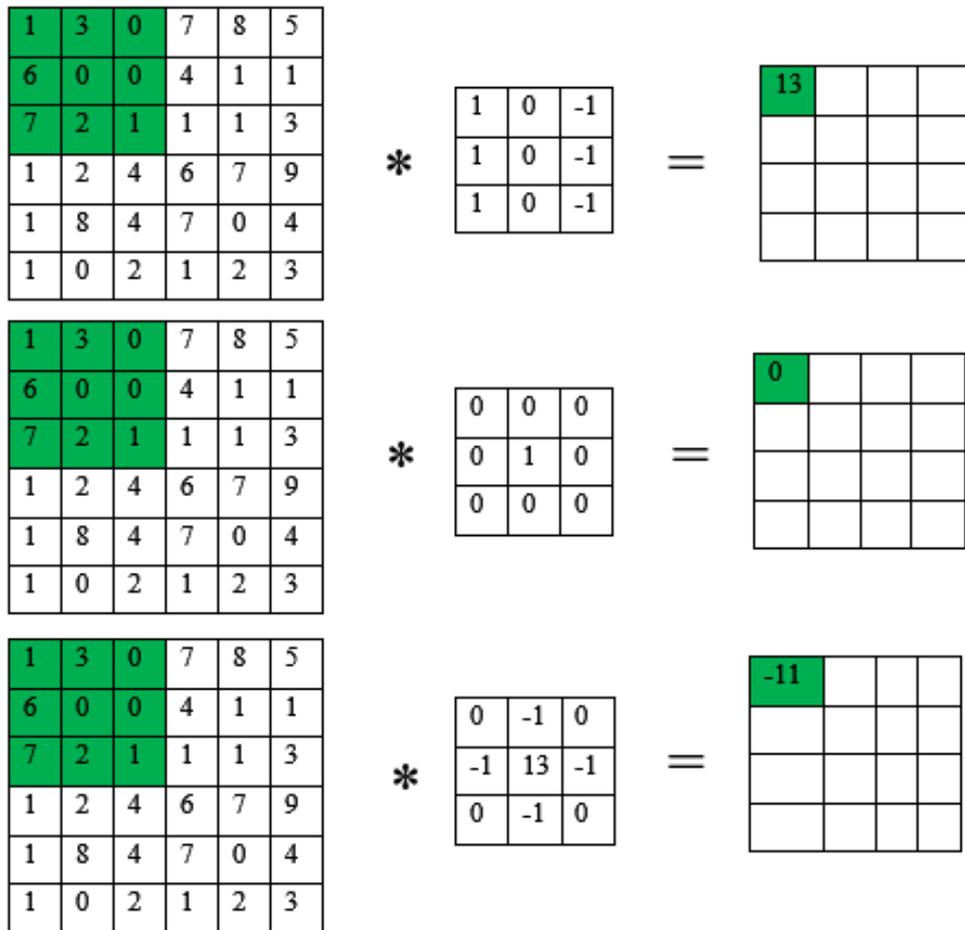
Gambar 3.4. Proses konvolusi.

2. Fungsi Aktivasi ReLU

Fungsi aktivasi ReLU digunakan untuk mencari nilai non-linear yang dihasilkan dari proses konvolusi. Rumus yang digunakan pada fungsi aktivasi ReLU seperti pada persamaan 3.1.

$$f(x) = \max(x, 0) \quad (3.1)$$

Dengan x merupakan input neuron atau *node*. Lalu angka 0 pada rumus ReLU merupakan unit linier yang dikoreksi jika input kurang dari 0. Artinya, jika input lebih besar dari 0, outputnya sama dengan input. Untuk lebih jelasnya lihat pada Gambar 3.5. Pada Gambar 3.5 kernel pertama menghasilkan 13, kernel kedua 0, dan kernel ketiga -11. Karena dari ketiga kernel tersebut diperoleh nilai paling besar 13 maka yang diambil adalah nilai tertinggi dan apabila pada hasil konvolusi memperoleh nilai negatif maka pada fungsi aktivasi ReLU diganti dengan nilai 0.



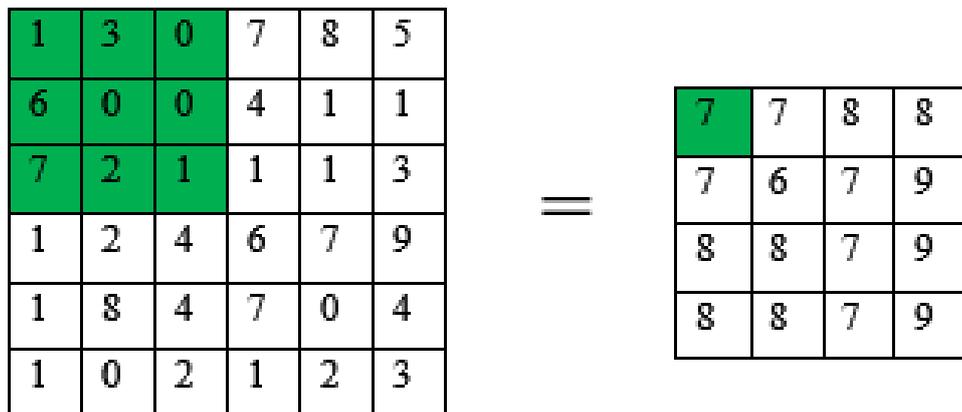
Gambar 3.5. Fungsi aktivasi ReLU.

3. Max Pooling

Pooling layer adalah lapisan untuk mengurangi dimensi dari *feature map* atau lebih dikenal untuk *downsampling*, sehingga mempercepat komputasi karena parameter yang harus di-*update* semakin sedikit dan mengatasi *overfitting*. Pada penelitian ini *pooling layer* yang digunakan adalah *max pooling*.

Max pooling sendiri lebih umum digunakan pada sebuah CNN, selain itu kelebihan dari *max pooling* dibandingkan dengan *average pooling* adalah jika pada *max pooling* nilai yang diambil adalah nilai tertinggi, sehingga dapat mengidentifikasi fitur yang ekstrim atau tajam. Selain itu untuk *max pooling* lebih baik dalam melakukan proses deteksi objek jika dibandingkan dengan *average pooling* yang tidak dapat mengestraksi fitur dengan baik karena memperhitungkan semuanya dan menghasilkan nilai rata-rata yang tidak penting untuk deteksi objek.

Max pooling pada Gambar 3.8 merupakan proses *pooling layer* yang berguna untuk mengurangi ukuran gambar. Pada Gambar 3.6 terdapat lapisan dengan ukuran 6×6, lalu kemudian dilakukan *filter* dengan menggunakan ukuran 3×3 dengan stride 1 maka diperoleh hasil *max pooling* dengan ukuran 4×4.



Gambar 3.6. Sampel *max pooling*.

4. *Fully Connected Layers*

Fully connected layers merupakan lapisan yang terhubung sepenuhnya di mana setiap piksel dianggap sebagai neuron atau *node* yang terpisah seperti jaringan saraf tiruan biasa. Lapisan terakhir yang terhubung sepenuhnya mengandung banyak neuron sebagai jumlah kelas yang harus diprediksi. Proses *fully connected layers* ini merupakan penyatuan setiap piksel yang dianggap sebagai plat yang kemudian menjadi sebuah output yang terdiri dari satu label kelas yaitu tanda nomor kendaraan sehingga lapisan yang terhubung terakhir hanya memiliki 1 neuron.

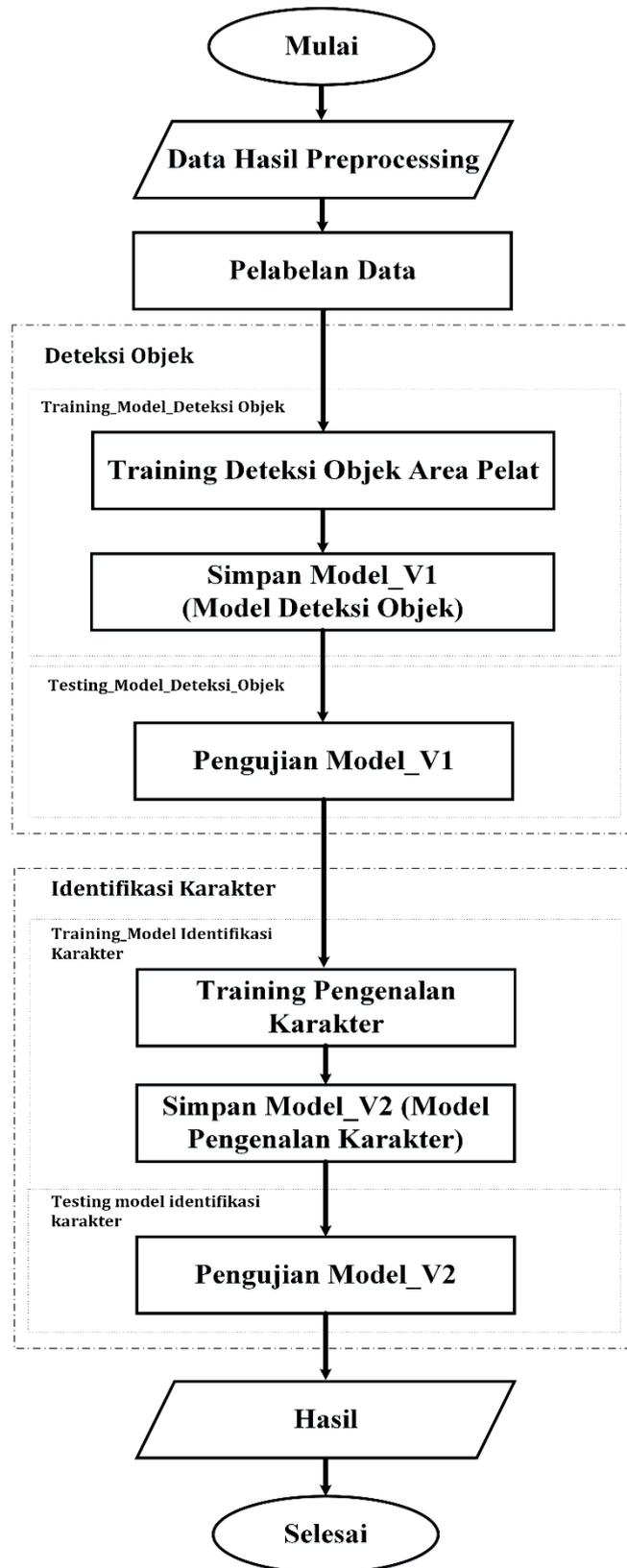
5. *Sofmax Classification*

Sofmax classification merupakan tahapan klasifikasi pada bagian mana saja pada setiap piksel yang memiliki pola sebagai plat yang diprediksi. Dalam penelitian ini setelah dilakukan proses CNN sebelumnya maka gambar pada CNN tersebut menghasilkan klasifikasi untuk menentukan apakah gambar tersebut merupakan plat atau bukan. Selanjutnya setelah dihasilkan plat maka dilakukan *bounding box* dengan *rectangle* di sekitar plat berwarna hijau.

Setelah proses deteksi plat pada CNN tersebut, maka untuk proses pada identifikasi karakter mengadopsi prinsip-prinsip yang terdapat pada arsitektur CNN Gambar 3.6. Untuk data input pada proses identifikasi atau pengenalan karakter tersebut digunakan ukuran 80×60 piksel. Lalu pada fitur ekstraksi dilakukan konvolusi 4 lapisan dengan ukuran filter 16, 32, 64, 128 dan diantaranya yang terdiri dari ukuran $6 \times 6 \times 3$, $4 \times 4 \times 16$, $3 \times 3 \times 32$, $3 \times 3 \times 64$, $3 \times 3 \times 128$. Penggunaan 4 lapisan konvolusi tersebut dikarenakan pada identifikasi karakter biasanya memiliki frekuensi yang lebih tinggi dan lebih luas. Kemudian pada prosesnya dilakukan inputan data dengan ukuran $80 \times 60 \times 3$. Lalu dilakukan konvolusi dengan 6×6 . Selanjutnya dilakukan proses *max pooling* 3×3 dan diperoleh matriks baru dengan ukuran 4×4 , proses ini dilakukan sampai pada lapisan konvolusi terakhir. Setelah proses pada ekstraksi fitur selesai maka dilakukan proses *fully connected layers*. Pada proses ini dilakukan penyatuan terhadap semua karakter yang diidentifikasi dan terakhir dilakukan *softmax classification* untuk mengklasifikasikan bahwa karakter yang diidentifikasi itu benar atau tidak.

3.1.4 Pemodelan

Tahap implementasi merupakan tahapan yang dilakukan untuk pengolahan data citra yang sebelumnya telah dilakukan pemrosesan data pada *preprocessing*. Adapun untuk tahap ini terdiri dari beberapa proses yaitu dimulai dari penggunaan data citra hasil preprocessing dan dilanjutkan dengan penerapan *convolutional neural networks* untuk pelatihan deteksi objek area plat dan deteksi karakter pada plat. Setelah proses tersebut selesai maka diperoleh model yang digunakan untuk pengujian dan pada tahap pengujian merupakan tahap yang dilakukan untuk menguji keakuratan model yang diperoleh. Untuk *flowchart* pengolahan citra yang digunakan pada penelitian ini terdapat pada Gambar 3.7.



Gambar 3.7. *Flowchart* penelitian.

Adapun langkah-langkah yang terdapat pada *flowchart* penelitian ini diantaranya sebagai berikut :

1. Data hasil *preprocessing*

Pada bagian ini merupakan bagian awal dari proses yang terjadi pada rancangan penelitian ini. Data citra merupakan tahapan awal yang digunakan sebagai data masukan pada program yang dibuat. Data citra yang digunakan ini merupakan data citra yang terdiri dari tiga jenis tanda nomor kendaraan yaitu data citra plat berwarna putih dengan warna latar hitam, data citra plat berwarna putih dengan warna latar merah dan data citra hitam dengan warna latar kuning. Untuk data citra yang digunakan sebagai masukan berjumlah 600 gambar.

2. Pelabelan data

Pada tahap ini dilakukan proses pelabelan gambar untuk pembuatan dataset berformat XML yang di konversi ke *TFRecord* agar dapat diproses oleh *Tensorflow* pada tahap implementasi. Sedangkan proses pelabelan gambar menghasilkan *file XML* yang berisi informasi mengenai data citra yang telah dilakukan *rectangle* pada proses pelabelan. Untuk struktur *file* untuk proses pelabelan yaitu data citra sejumlah 500 disimpan di *folder train* pada *folder Image* dan 100 data citra disimpan di *folder val* pada *folder Image*. Selanjutnya setelah program pelabelan dijalankan maka setiap data citra yang sudah dilabeli disimpan pada *file train* sejumlah 500 data dengan format XML didalam *folder annotations*, begitupun untuk 100 *data validation* setelah dilabel disimpan pada *folder val* dengan format XML pada *folder annotations*. Sehingga dihasilkan dataset dari data plat. Adapun untuk contoh proses pelabelan seperti terdapat pada Gambar 3.8 dan Gambar 3.9.



Gambar 3.8. Pelabelan plat hitam.



Gambar 3.9. Pelabelan plat merah.

3. *Object Detection*

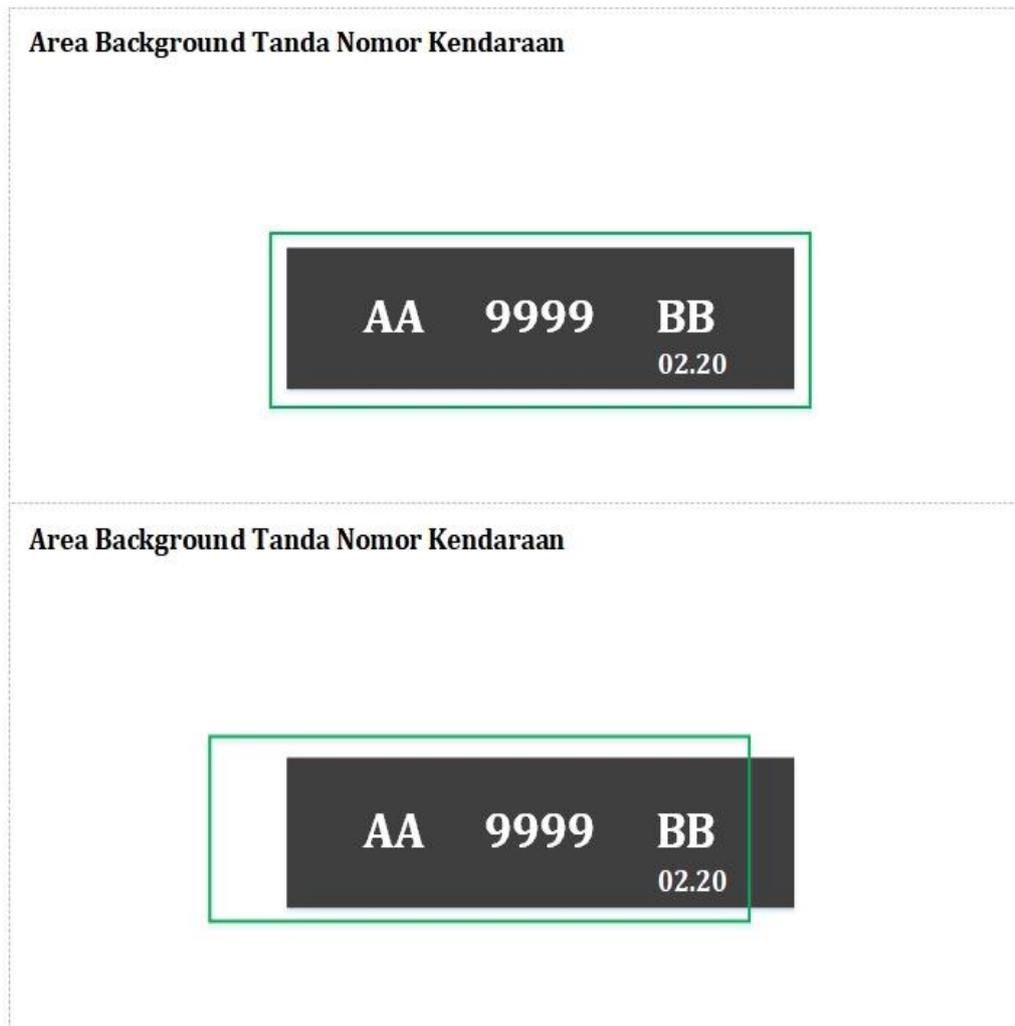
A. *Pelatihan Object Detection*

Proses yang terjadi pada bagian ini adalah data hasil *preprocessing* yang telah dimasukkan pada program langsung dilakukan pelatihan untuk mendeteksi

bagian plat dengan menggunakan model CNN jenis *Single Slot Detection* (SSD) *ResNet-50 Box Predictor*.

Pada tahap ini jumlah data yang dilatih yaitu sejumlah 500 gambar. Adapun proses yang terjadi yaitu melakukan memasukkan data hasil *preprocessing* yang telah dilakukan pelabelan pada *tensorflow*. Lalu langkah selanjutnya adalah dilakukan proses pelatihan *object detection* dengan menggunakan *convolutional neural networks*, setelah proses konvolusi selesai maka dilakukan *pooling layer* yang bertujuan untuk mengurangi *downsampling* dan mengurangi *overwriting* sehingga dapat mempercepat proses komputasi. Selanjutnya diaktifkan fungsi aktivasi ReLU (*Rectified Linear Unit*) dan menghasilkan prediksi kelas berupa gambar baru dengan label disekitar area plat seperti Gambar 3.10. dan juga diperoleh model dari proses pelatihan ini. Setelah model diperoleh maka model tersebut disimpan dalam format *file* tertentu yang dapat digunakan untuk tahap pengujian.

Pada tahap *object detection* ini juga akan digunakan YOLO V3 sebagai pembanding. YOLO merupakan sebuah pengembangan dari CNNs yang memang didesain untuk menyelesaikan permasalahan 2D dan 3D. YOLO sendiri dikenali mampu mendeteksi sebuah objek yang memiliki ruang warna RGB. YOLO V3 adalah peningkatan dari dua versi YOLO sebelumnya yang lebih baik tetapi sedikit lebih lambat dari versi sebelumnya. Model YOLO V3 memiliki fitur *multi-scale detection*, *feature extraction network* yang lebih baik, dan beberapa perubahan pada *loss function* [37]. Oleh karena itu dengan menerapkan YOLO V3 ini sebagai pembanding pada *object detection* dapat mendeteksi juga plat yang memiliki ruang warna RGB.



Gambar 3.10. Hasil prediksi deteksi plat.

Setelah dilakukan pelatihan deteksi objek terdapat bagian tertentu yang dapat diambil dari gambar tersebut atau disebut sebagai *region of interest* (ROI). Adapun bagian tertentu dari ROI tersebut memiliki kriteria diantaranya adalah :

1. Karakter pada area yang dideteksi objek platnya masih sepenuhnya dapat dikenali dan tidak hilang.
2. Gambar hasil dari deteksi platnya dapat menghasilkan gambar dengan area plat yang lebih besar atau tepat.

Dari hasil pelatihan deteksi akan diperoleh sebuah model berisi informasi dari area plat dari data latih, model tersebut berguna untuk melakukan deteksi objek plat pada proses pengujian. Performa model tersebut dapat diukur melalui nilai akurasi dari *Mean Average Precision* (mAP) dan *Average Recall* (AR). Adapun secara matematis untuk mAP dan AR dapat dirumuskan pada persamaan 3.2, pada persamaan 3.2 N menyatakan banyaknya kelas yang digunakan sedangkan untuk AP menyatakan *Average Precision* yang juga dirumuskan pada persamaan 3.3.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3.2)$$

Pada AP variabel r merupakan banyaknya nilai dari *recall* sehingga r_{i+1} dapat dihitung dari $r_1, r_2, r_3, \dots, r_n$. Lalu variabel *Pinterp* merupakan nilai presisi tertinggi pada nilai recall apapun, oleh karena itu pada persamaan 3.4 *Pinterp* digunakan konstanta *max* yang menyatakan nilai recall tertinggi.

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) P_{interp}(r_{i+1}) \quad (3.3)$$

$$P_{interp}(r) = \max_{r^l \geq r} p(r^l) \quad (3.4)$$

Sedangkan untuk AR dapat dirumuskan pada persamaan 3.5, variabel *recall(o)* adalah nilai dari recall itu sendiri dan o merupakan nilai dari IoU.

$$AR = 2 \int_{0.5}^1 recall(o) do \quad (3.5)$$

B. Tahap Pengujian Model Deteksi Objek Area Plat

Proses pengujian ini bertujuan untuk menguji model yang dihasilkan pada proses pelatihan deteksi area plat apakah model yang dihasilkan memperoleh *score* yang tinggi atau tidak. Data uji yang digunakan disini merupakan data original.

4. Identifikasi Karakter

A. Tahap Pelatihan Pengenalan Karakter

Tahap pelatihan pengenalan karakter merupakan tahapan untuk melatih algoritma dalam mengidentifikasi karakter yang terdapat pada plat. Pada tahap ini digunakan data dari hasil deteksi objek berupa deteksi area plat yang telah dilakukan segmentasi perkarakturnya. Tahap pengenalan karakter ini dilatih dengan menggunakan model CNN MobileNet V2 dan *weights* dari *imagenet* dan gambar di *resize* menjadi $80 * 80$ yang bertujuan untuk mempercepat proses komputasi dengan 3 *channels* RGB.

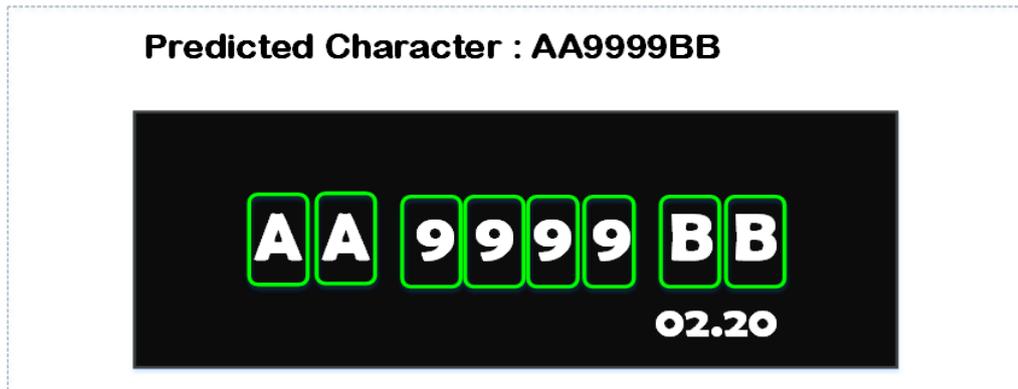
Proses pelatihan identifikasi karakter pertama kali adalah dilakukan *feeding* data dari dataset untuk dikonsumsi pada *tensorflow*. Lalu dilakukan *pooling layer* untuk mengurangi *downsampling* agar dapat mempercepat proses komputasi. Setelah itu maka aktifkan fungsi aktivasi ReLU, lalu kemudian setelah melewati *convolutinal layer*, *stride*, *padding* maka akan disatukan dengan *full connected layer* dan *softmax classification* untuk mengklasifikasikan hasil pelatihan dan disimpan pada *weights* dengan format model HDF5, JSON dan Numpy *Array* agar bisa digunakan pada tahap prediksi atau pengujian.

B. Tahap Pengujian Model pengenalan Karakter

Proses pengujian ini bertujuan untuk menguji model yang dihasilkan pada proses pelatihan identifikasi karakter apakah model yang dihasilkan mampu mendeteksi karakter dengan tepat. Tahap ini merupakan tahap untuk menguji data dari hasil pengujian menggunakan model deteksi area plat sejumlah 100 gambar.

5. Hasil Identifikasi Karakter

Hasil akhir dari proses identifikasi karakter adalah berupa gambar dengan mengekstraksi karakter yang terdapat pada plat tersebut. Untuk rancangan hasil dari identifikasi karakter tersebut dapat dilihat pada Gambar 3.11. Pada rancangan hasil tersebut hasil dari prediksi identifikasi karakter dengan memberikan *bounding box* berwarna hijau untuk setiap karakter yang berhasil diidentifikasi.



Gambar 3.11. Hasil identifikasi karakter.

3.1.5 Analisis Dan Evaluasi

Tahap analisis dan evaluasi merupakan tahap untuk melakukan evaluasi terhadap akurasi model yang digunakan pada proses pengujian menggunakan data uji. Adapun untuk menghitung akurasi yang berhasil diidentifikasi dapat digunakan persamaan 3.6. dengan ketentuan menghitung setiap karakter yang berhasil diidentifikasi dibagi dengan total karakter benar dan dikalikan dengan 100%. Untuk persamaan 3.6 ini berlaku pada setiap data citra yang dilakukan pengujian.

$$Accuracy = \frac{\text{Jumlah prediksi benar}}{\text{Total data}} \times 100\% \quad (3.6)$$

Adapun keterangan dari persamaan 3.6 adalah sebagai berikut :

1. Jumlah prediksi benar : merupakan total dari karakter yang benar atau sesuai fakta setelah dilakukan proses pengujian pada setiap data citra.

-
2. Total seluruh karakter : merupakan total keseluruhan data yang digunakan untuk pengujian data citra.