

## Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python

Nisa Hanum Harani, Cahyo Prianto, Miftahul Hasanah  
Program Studi D4 Teknik Informatika Politeknik Pos Indonesia  
Bandung

nisahanum@poltekpos.ac.id, cahyoprianto@poltekpos.ac.id, mifta896@gmail.com

### ABSTRAK

Saat ini pengenalan karakter plat nomor kendaraan di Indonesia pada umumnya digunakan pada sistem parkir yang masih dilakukan manual, yaitu dengan masih mencatat karakter pada plat nomor kendaraan oleh petugas. Pengenalan karakter pada plat nomor kendaraan bukan hanya digunakan oleh sistem parkir saja, tetapi bisa juga digunakan pada lalu lintas. Pengenalan karakter plat nomor kendaraan Indonesia merupakan salah satu jenis deep learning karena dapat mengenali berbagai karakter angka dan huruf. Pembuatan sistem pengenalan karakter pada plat nomor kendaraan Indonesia memberikan alternatif pengenalan karakter plat kendaraan menggunakan metode Convolutional Neural Network (CNN) untuk mengenali objek pada gambar selayaknya manusia dengan pembelajaran pada sebuah komputer dengan menggunakan jaringan saraf tiruan. Hasil dari penelitian ini adalah sistem yang sudah ada pada komputer dapat mengenali karakter pada plat nomor kendaraan Indonesia yang sudah kita masukkan. Keseluruhan jumlah class yang kita gunakan pada penelitian tentang pengenalan karakter plat nomor kendaraan Indonesia adalah 37 class yaitu terdiri dari karakter huruf dan angka. Cara kerja dari sistem ini adalah kita memasukkan gambar plat nomor kendaraan Indonesia dengan format datanya JPG pada sistem, kemudian sistem akan mengolah gambar tersebut dan hasilnya akan langsung ditampilkan pada sistem. Sehingga pada saat kita masuk ke tempat parkir, petugas tidak lagi mencatat karakter plat nomor kendaraan secara manual, karena sudah dilakukan oleh sistem. Nantinya sistem ini akan dikembangkan menjadi smart parking yang mana kita tidak perlu lagi untuk men scan barcode karena data kita sudah terdaftar secara otomatis pada sistem. Kata kunci— Plat nomor kendaraan, deep learning, metode convolutional neural network.

### I. PENDAHULUAN

Jumlah kendaraan di Indonesia setiap tahunnya terus mengalami peningkatan. Berdasarkan data milik Korps Lalu Lintas Kepolisian Negara Indonesia yang dikutip oleh situs surat kabar Kompas, pada tahun 2013 jumlah kendaraan di Indonesia mencapai 104.211.000 unit, atau meningkat sebesar 11% dibandingkan dengan tahun sebelumnya.[1] Setiap kendaraan baik motor maupun mobil memiliki plat nomor. Plat nomor adalah salah satu bentuk identitas dari setiap kendaraan baik motor maupun mobil. Plat nomor kendaraan terdiri dari kombinasi huruf dan angka, yang mana setiap huruf dan angka tersebut mengandung informasi tentang kode propinsi dan kode daerah dimana kendaraan tersebut terdaftar.[2] Plat nomor dari tiap negara memiliki model yang berbeda, baik dari warna maupun bentuk susunan karakternya, sehingga pengenalan plat nomor yang dilakukan di negara lain belum tentu bisa mengenali plat nomor yang ada di Indonesia.

Meningkatnya jumlah kendaraan bermotor dan semakin padatnya arus lalu lintas di suatu daerah membuat sistem pengenalan citra *digital* plat nomor kendaraan oleh komputer menjadi penting. Sistem pengenalan citra digital plat nomor kendaraan diperlukan untuk mendukung ketertiban lalu lintas, mengetahui dengan cepat pemilik kendaraan, dan mempermudah pengaturan area parkir. Sistem pengenalan citra digital plat nomor kendaraan dapat dilakukan menggunakan teknik pengenalan pola. [2] Penerapan yang menggunakan pengenalan plat nomor salah satunya sebagai input parkir, dimana awalnya

pencatatan nomor plat kendaraan menggunakan cara manual sekarang bisa menggunakan gambar sebagai alat input. Teknologi ini secara tidak langsung mengurangi peran manusia dalam sistem parkir. [3] Penelitian mengenai pengenalan karakter pada plat nomor kendaraan terus mengalami perkembangan. Dimulai dengan penelitian untuk deteksi lokasi plat nomor sampai dengan pengenalan karakter pada plat nomor kendaraan.

Sekarang ini perkembangan teknologi bukan hanya pada bidang teknologi saja tetapi juga pada bidang komputerisasi contohnya adalah *computer vision* yang mengalami perkembangan yang sangat pesat. Dalam *computer vision* terdapat beberapa permasalahan dan salah satunya adalah pengenalan karakter plat nomor kendaraan / *Recognition Plate*. [4] *Recognition Plate* dengan *deep learning* atau jaringan saraf tiruan ini masih berkembang sebagai teknologi untuk menduplikasi kemampuan manusia dalam memahami informasi dari sebuah gambar agar komputer dapat mengenali objek pada gambar selayaknya manusia. [5]

*Deep learning* memiliki kemampuan yang sangat baik dalam visi *computer* dengan kapabilitasnya yang signifikan dalam memodelkan berbagai data kompleks seperti data gambar. Salah satu metode *Deep learning* pada saat ini memiliki hasil paling bagus dalam pengenalan citra adalah *Convolutional Neural Network* (CNN). Hal itu disebabkan karena metode CNN berusaha meniru sistem pengenalan citra pada *visual cortex* manusia sehingga memiliki kemampuan mengolah informasi citra. Namun CNN, seperti metode *Deep*

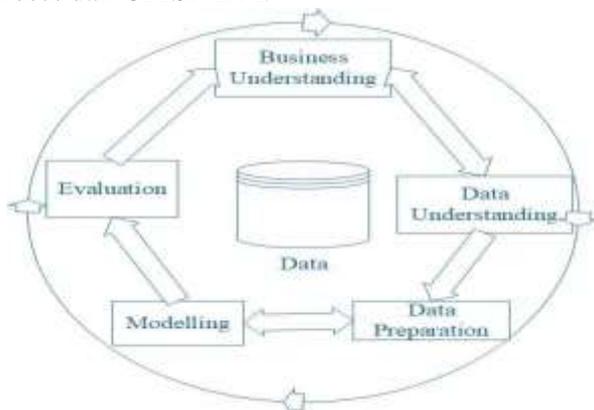
Jurnal Teknik Informatika, Vol. 11, No. 3, Agustus 2019  
*learning* lainnya, memiliki kelemahan yaitu proses pelatihan model yang lama.[6] CNN dirancang khusus untuk *deep learning* yaitu pada bidang *image processing*. [7] CNN digunakan karena dapat menangani masalah yang kompleks tersebut dan memiliki kinerja yang baik dari hasil penelitian terkait.[8] CNN merupakan variasi dari jaringan saraf tiruan yang memiliki bobot dan beberapa lapisan tersembunyi yang disusun menjadi arsitektur. [9]

CNN memiliki kemampuan klasifikasi yang diperuntukkan untuk pengolahan data gambar.[10] *Convolutional Neural Network* dapat kita gunakan untuk proses klasifikasi data yang sudah ter label dengan menggunakan metode *supervised learning*, dimana cara kerja dari *supervised learning* adalah sudah terdapat data yang akan dilatih dan terdapat variabel yang sudah kita targetkan sehingga tujuan dari metode ini adalah mengelompokkan suatu data ke data yang sudah ada. CNN sering digunakan untuk mengenali benda atau pemandangan, dan melakukan deteksi dan segmentasi objek.[11]

Berdasarkan uraian diatas, maka penulis ingin membuat sistem untuk mendeteksi pengenalan karakter pada plat nomor kendaraan Indonesia dengan menggunakan *python*. Adapun metode yang digunakan pada sistem tahap pengenalan karakter plat nomor kendaraan ini adalah metode *Convolutional Neural Network* (CNN).

## II. METODE

Pada penelitian ini peneliti menggunakan metode *Convolutional Neural Network*(CNN). Sedangkan metodologi yang digunakan adalah metodologi CRISP-DM (*Cross Standard Industries for Data Mining*) merupakan standar *data mining* yang disusun oleh tiga tambang penggagas data pasar, yaitu Daimler Chrysler (Daimler-Benz), SPSS (ISL), NCR. Kemudian berkembang dalam berbagai lokakarya (antara 1997-1999). Lebih dari 300 organisasi memberikan kontribusi untuk proses pemodelan ini dan akhirnya CRISP-DM 1.0 diterbitkan pada tahun 1999.[12] Gambar 1. Merupakan proses dari CRISP-DM.



Gambar 1. Proses CRISP-DM

## 2.1 Tahapan-Tahapan Diagram Alur Penelitian

Terdapat beberapa tahapan dalam CRISP-DM (*Cross Industries Process for Data Mining*), namun pada penelitian ini tahapan yang dilakukan hanya sampai pada tahapan *evaluation* dikarenakan tujuan dari penelitian ini adalah hanya untuk mendapatkan model hingga proses *testing*.

### 2.1.1 Business Understanding

Pada fase ini peneliti memahami permasalahan yang ada, menetapkan tujuan data *mining* dan mengembangkan rencana penelitian. Permasalahan yang ada sekarang adalah ditempat parkir pada saat kendaraan akan parkir itu masih dicatat secara manual oleh petugas. Kemudian sistem ini akan dikembangkan menjadi *smart parking*, yaitu pencatatan plat nomor kendaraan sudah dilakukan secara otomatis pada saat masuk ke tempat parkir, program ini diselesaikan dengan menggunakan metode *Convolutional Neural Network*.

### 2.1.2 Data Understanding

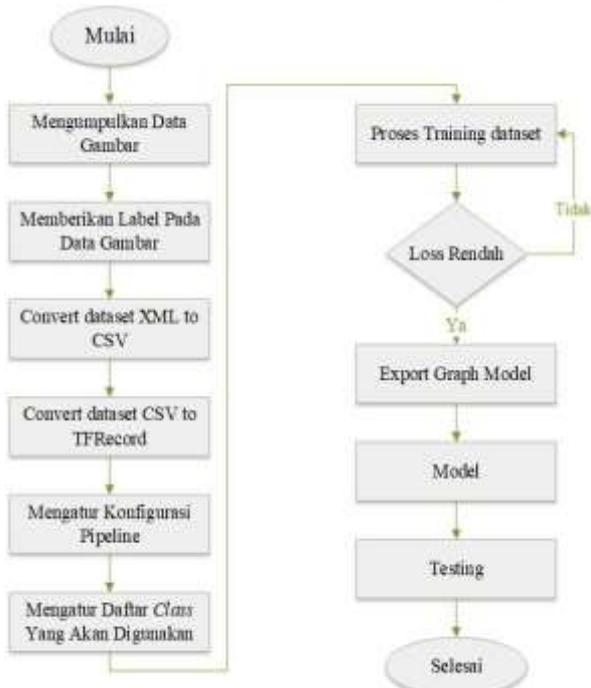
Pada tahap peneliti melakukan pengumpulan data. Data yang digunakan dalam penelitian berjumlah 1615 gambar data plat nomor kendaraan Indonesia, data tersebut sudah dibagi kedalam data *training* dan data *testing*. Data gambar plat nomor kendaraan Indonesia yang diambil langsung dari *kaggle* dan data yang diambil langsung dari tempat parkir dan jalanan oleh peneliti dengan menggunakan *smartphone*. Data tersebut diambil dari periode bulan September sampai bulan November, jika data yang diperoleh semakin banyak maka hasil dari akurasi datanya juga akan maksimal. Tabel 1. Dibawah ini merupakan tabel data awal pada pengenalan karakter pada plat nomor kendaraan Indonesia.

Tabel 1. Data Awal Plat Nomor Kendaraan

No.	Sumber Data	Jenis Data	Lokasi Data	Jumlah Data	Format Data
1.	<i>Kaggle</i> [13]	Data <i>Trainin</i>	<i>Kaggle.co</i> <i>m</i>	470	JPG
2.	Data dari <i>Internet</i> [14]	Data <i>Trainin</i> <i>g</i>	Google	100	JPG
2.	Data diambil dengan <i>Smartnh</i>	Gambar <i>Trainin</i> <i>g</i>	Tempat parkir dan jalan	939	JPG
3.	Data diambil dengan <i>Smartpho</i>	Data <i>Test</i>	Tempat parkir dan jalan	106	JPG

### 2.1.3 Data Preparation

Meliputi semua aktivitas untuk membuat *dataset* final. Pada penelitian mengenai pengenalan karakter pada plat nomor kendaraan Indonesia dengan menggunakan metode *convolutional neural network*. Bahasa pemrograman yang digunakan adalah *python*.



Gambar 2. Diagram Alur Penelitian

Pada gambar 2. Merupakan diagram alur penelitian pada pengenalan karakter plat nomor kendaraan, dijelaskan tahapan peneliti dari mendapatkan data awal hingga terbentuknya model sehingga dapat dilakukan proses *testing* data plat nomor kendaraan Indonesia.

### 2.1.3.1 Proses Pemberian Label Pada Gambar / Labelling

Pada proses ini, setelah gambar nya dikumpulkan pada 1 *Folder*, kemudian peneliti melakukan proses *labelling* gambar dengan menggunakan *tools* *Labellmg*. Data yang dihasilkan oleh *labellmg* ini adalah titik koordinat gambar, anotasi gambar plat nomor kendaraan dimana format yang dihasilkan dari proses memberikan label pada data gambar adalah berupa *.xml*. *Labellmg* menyimpan *file* *.xml* yang berisi data label untuk setiap gambar. *File* *.xml* ini akan digunakan untuk menghasilkan *TFRecords*. Gambar 3. Dibawah ini merupakan proses pemberian label image pada *dataset* plat nomor kendaraan Indonesia.



Gambar 3. Prose Labellmg

### 2.1.3.2 Mengubah File XML menjadi File CSV

Selanjutnya setelah kita melakukan prose labelling pada gambar langkah selanjutnya yaitu mengubah *file* *XML* menjadi *file* *CSV* agar bisa melanjutkan ke tahap berikutnya. Gambar 4. Dibawah ini merupakan proses *convert* data *.xml* to *.csv* beserta hasilnya.

```
(tensorflow) E:\tensorflow3\models\research\object_detection>python xml_to_csv.py
Successfully converted xml to csv.
```

```
(tensorflow) E:\tensorflow3\models\research\object_detection>python xml_to_csv.py
Successfully converted xml to csv.
```

Gambar 4. Proses convert data *.xml* to *.csv* beserta hasilnya

Pada gambar 4. Merupakan *source code* untuk mengubah *file xml* menjadi *csv* pada data *train* dan data *test*, dimana *source code* tersebut kita jalankan menggunakan *python*. Berikut gambar 5. Merupakan hasil dari proses *convert xml to csv*.

image_3344.jpg	447	193	E	20	22	59	97
image_3344.jpg	447	193	4	117	19	161	97
image_3344.jpg	447	193	1	168	17	196	97
image_3344.jpg	447	193	1	207	14	233	101
image_3344.jpg	447	193	2	241	19	282	99
image_3344.jpg	447	193	R	355	13	391	97
image_3344.jpg	447	193	J	391	20	427	94

Gambar 5. Hasil Proses Convert XML to CSV

### 2.1.3.3 Proses Membuat TFRecord

Selanjutnya kita akan menghasilkan *file TFRecord*, yang nantinya akan menghasilkan *file train.record* dan *file test.record*. Jika sudah mengubah data *train* dan data *test* dalam bentuk *CSV* maka selanjutnya adalah mengubah *file csv* menjadi *file* yang dapat dibaca oleh *tensorflow* yaitu *TFRecord*. Ini akan digunakan untuk melatih *classifier* deteksi pengenalan karakter pada plat nomor kendaraan Indonesia. Gambar 6. dan gambar 7. Dibawah ini merupakan proses *TFRecord* beserta hasilnya pada data *train* dan data *testing*.

```
(tensorflow) E:\tensorflow3\models\research\object_detection>python generate_tfrecord.py
--csv_input=data/train_labels.csv --type=train --output_path=train.record
C:\Users\Mifta\Anaconda3\envs\tensorflow\lib\site-packages\h5py\_init_.py:72: UserWarning:
h5py is running against HDF5 1.10.2 when it was built against 1.10.3, this may cause
problems
'{0}'.format(*version.hdf5_built_version_tuple)
Successfully created the TFRecords: E:\tensorflow3\models\research\object_detection\train
.record
```

Gambar 6. proses TFRecord beserta hasil data *train.record*

```
(tensorflow) E:\tensorflow3\models\research\object_detection>python generate_tfrecord.py
--csv_input=data/test_labels.csv --type=test --output_path=test.record
C:\Users\Mifta\Anaconda3\envs\tensorflow\lib\site-packages\h5py\_init_.py:72: UserWarning:
h5py is running against HDF5 1.10.2 when it was built against 1.10.3, this may cause
problems
'{0}'.format(*version.hdf5_built_version_tuple)
Successfully created the TFRecords: E:\tensorflow3\models\research\object_detection\test
.record
```

Gambar 7. proses TFRecord beserta hasil data *test.record*

### 2.1.3.4 Konfigurasi Pipeline

Selanjutnya setelah *file TFRecord* nya berhasil maka selanjutnya adalah membuat (mengatur)

Jurnal Teknik Informatika, Vol. 11, No. 3, Agustus 2019 konfigurasi *pipeline*, dimana konfigurasi ini akan digunakan untuk melakukan konfigurasi dari model *training*. Konfigurasi ini menggunakan model *faster\_rcnn\_inception\_v2\_coco\_2018\_01\_28* dimana ini sudah disediakan oleh *tensorflow* sendiri. Pada tahapan konfigurasi ini kita akan mengatur berapa banyak *class* yang kita gunakan. Pada penelitian ini peneliti menggunakan sebanyak 37 *class* yaitu huruf A hingga huruf Z, serta angka 0 hingga angka 9, dan objek *plate*. Selain mengatur berapa *class* yang digunakan kita juga mengatur berapa banyak step yang akan digunakan pada saat proses *training dataset*.

### 2.1.3.5 Mengatur Daftar Class / label Map Yang Akan Digunakan

Pada tahap ini kita akan mengatur terlebih dahulu label yang kita gunakan, pada gambar di atas terdapat 37 *class*. Label map memberi tahu pelatih masing-masing objek dengan mendefinisikan pemetaan nama kelas ke nomor ID kelas. Pada tabel 2. Dibawah ini merupakan sebagian dari *class* yang kita gunakan pada penelitian tentang pengenalan karakter pada plat nomor kendaraan Indonesia.

Tabel 2. Proses Pembuatan label Map

Tabel <i>classes.pbtxt</i>
<i>item</i> { <i>id</i> : 1 <i>name</i> : 'plate' }
<i>item</i> { <i>id</i> : 2 <i>nam</i> <i>e</i> : '1' }
<i>item</i> { <i>id</i> : 3 <i>name</i> : '2' }

### 2.1.3.6 Mengatur Tata Letak File dan Folder

+Data  
| - *faster\_rcnn\_inception\_v2\_coco\_2018\_01\_28*  
| - *classes.pbtxt*  
| - *train.TFRecord*  
-| - *test.record*  
+Training

Mengatur Tata Letak File dan folder ditujukan agar sesuai dengan syarat untuk dilakukannya proses *training dataset*. Jika syarat *file object detection* sudah terpenuhi, selanjutnya kita akan melakukan proses *training data*.

### 2.1.3.7 Proses Training Dataset

Selanjutnya setelah semua syarat untuk proses *training* sudah terpenuhi, kita akan melakukan proses *training dataset*.



Gambar 8. Proses Training Dataset

Pada gambar 8. Merupakan proses *training dataset*, peneliti akan melatih *dataset* dengan menggunakan *faster\_rcnn\_inception\_v2\_coco\_2018\_01\_28*. *Faster\_rcnn\_inception\_v2\_coco\_2018\_01\_28* adalah proses dari CNN yang sangat efisien. Convolutional hanya berarti bahwa perhitungan yang sama dilakukan di setiap lokasi dalam gambar. Untuk memulai proses training nya bisa menggunakan perintah pada gambar 9 berikut.

```
(tensorflow) E:\tensorflow3\models\research\object_detection\python
model_main.py --logtostderr --train_dir=training_fasterRCNN/ --pip
eline_config_path=training_fasterRCNN/plate_number_v2.config
```

Gambar 9. Script Training Dataset

Dapat dilihat dari Code pada gambar 9 diatas, bahwa output dari proses ini adalah menghasilkan file dengan nama *frozen\_inference\_graph.pb* yang merupakan hasil data training yang dilakukan oleh *faster\_rcnn\_inception\_v2\_coco\_2018\_01\_28*. Proses training data dapat dilihat seperti pada gambar 10 di bawah ini:

```
INFO:tensorflow:global step 99980: loss = 0.2522 (0.203 sec/step)
INFO:tensorflow:global step 99981: loss = 0.5516 (0.172 sec/step)
INFO:tensorflow:global step 99982: loss = 0.8992 (0.187 sec/step)
INFO:tensorflow:global step 99983: loss = 0.3643 (0.187 sec/step)
INFO:tensorflow:global step 99984: loss = 0.1343 (0.172 sec/step)
INFO:tensorflow:global step 99985: loss = 0.2001 (0.187 sec/step)
INFO:tensorflow:global step 99986: loss = 0.3319 (0.172 sec/step)
INFO:tensorflow:global step 99987: loss = 0.1693 (0.172 sec/step)
INFO:tensorflow:global step 99988: loss = 0.7458 (0.172 sec/step)
INFO:tensorflow:global step 99989: loss = 0.2191 (0.187 sec/step)
INFO:tensorflow:global step 99990: loss = 0.1267 (0.156 sec/step)
INFO:tensorflow:global step 99991: loss = 0.1343 (0.156 sec/step)
INFO:tensorflow:global step 99992: loss = 0.2042 (0.172 sec/step)
INFO:tensorflow:global step 99993: loss = 0.5663 (0.187 sec/step)
INFO:tensorflow:global step 99994: loss = 0.1867 (0.156 sec/step)
INFO:tensorflow:global step 99995: loss = 0.6181 (0.187 sec/step)
INFO:tensorflow:global step 99996: loss = 1.1754 (0.203 sec/step)
INFO:tensorflow:global step 99997: loss = 0.1723 (0.156 sec/step)
INFO:tensorflow:global step 99998: loss = 0.2224 (0.203 sec/step)
INFO:tensorflow:global step 99999: loss = 0.1549 (0.156 sec/step)
INFO:tensorflow:global step 100000: loss = 0.4603 (0.172 sec/step)
INFO:tensorflow:Stopping Training.
INFO:tensorflow:Finished training! Saving model to disk.
```

Gambar 10. Proses Training Data

Pada gambar 10 di atas, dapat dilihat bahwa *Tensorflow* akan menjalankan *training data* dengan parameter *loss*, dan waktu proses *training data* dengan

Jurnal Teknik Informatika, Vol. 11, No. 3, Agustus 2019  
100.000 *steps*. Pada saat melakukan proses *training* data kita memerlukan waktu sekitar 6 jam untuk melakukan *training* sebanyak 100.000 *steps* tersebut.

### III. HASIL/ MODELLING

Pada fase ini dilakukan dengan menggunakan aplikasi Anaconda Prompt, dan dimasukkan juga metode Convolutional Neural Network. Dari data yang telah dipilih pada fase data preparation yaitu data set training dan data set testing.

#### 3.1.1 Extract Model Dari Hasil Training

Setelah selesai proses training data, selanjutnya adalah kita meng-extract model menjadi model frozen inference graph sehingga nantinya dapat digunakan untuk pengenalan karakter pada plat nomor kendaraan. Untuk mengekstrak model yang sudah kita dapatkan bisa menggunakan file `export_inference_graph.py` kemudian menggunakan perintah pada gambar 11 berikut ini:

```
(tensorflow) E:\tensorflow\models\research\object_detection\python export_inference_graph.py --input_type image_tensor --pipeline_config_path training_fasterRCNN/plate_number_v2.config --trained_checkpoint_prefix training_fasterRCNN/model.cpkt-100000 --output_directory inference_graph
```

Gambar 11. Extract Model Dari Hasil Training

#### 3.1.2 Hasil Extract Model Training

Setelah kita berhasil melakukan proses extract model setelah melakukan proses training menjadi `frozen_inference_graph.pb` sehingga nantinya dapat digunakan untuk pengenalan karakter pada plat nomor kendaraan. Setelah kita mendapatkan model, kita dapat gunakan model tersebut untuk melakukan testing pada pengenalan karakter pada plat nomor kendaraan Indonesia. Jika kualitas dataset yang kita gunakan bagus, maka hasilnya juga akan maksimal. Gambar 12 dibawah ini merupakan perintah untuk melakukan proses testing data pada pengenalan karakter plat nomor kendaraan Indonesia.



Gambar 13. Tampilan pada proses testing

```
[INFO] Loading image "images\test\image_3302.jpg"  
Found: D1254RI
```

Gambar 14. Tampilan pada console anaconda

Pada gambar 13 dan gambar 14 kita melakukan proses testing

pada pengenalan karakter pada plat nomor kendaraan Indonesia, dimana file yang kita testing terdapat pada Folder `image/test/image_3302.jpg`. Kita dapat mengetahui

hasil testing dari plat nomor kendaraannya terdapat huruf dan angka, yaitu dengan plat D1254RI.

#### 4. Pengkajian

Pada tahap ini, penulis akan mengkaji terkait dengan penelitian

yang dilakukan. Tujuan dilakukannya penelitian adalah membuat sebuah program smart parking karena pada saat ini untuk masuk ke tempat parkir, petugas masih mencatat secara manual plat nomor kendaraan, dimana dalam penelitian ini menggunakan metode Convolutional neural Network(CNN) untuk mengolah datanya. Proses penelitian serta pengolahan data dilakukan dengan mengikuti tahapan dari metodologi penelitian CRISP-DM.

Data plat nomor kendaraan yang telah dikumpulkan kemudian

dimodelkan menggunakan bahasa pemrograman python pada aplikasi Anaconda. Dalam proses pemodelan tersebut dataset plat nomor kendaraan dibagi menjadi data training dan data testing untuk dapat mengenai karakter pada plat nomor kendaraan. Selanjutnya data gambar plat nomor kendaraan yang telah kita kumpulkan sebelumnya dibagi menjadi data train dan data test.

Data set tersebut terlebih dahulu kita beri label dengan menggunakan tools `labelImg`. Format hasil dari `labelImg` gambar ini adalah berupa file XML. Setelah kita melakukan proses label dataset gambar selanjutnya kita akan mengubah format dari file XML

menjadi format file CSV. Setelah berhasil langkah selanjutnya yaitu membuat file `TFRecord` pada data train dan data test, file `TFRecord` merupakan format yang dapat dibaca oleh `Tensorflow`.

Setelah berhasil membuat file `data.train.record` dan `data.test.record`, selanjutnya adalah kita akan melakukan proses training data, sebelum melakukan proses training data kita harus sudah mempunyai topologi objek deteksi. Pada penelitian ini penulis menggunakan topologi objek deteksi yaitu `faster_rcnn_inception_v2_coco_2018_01_28`. Selain topologi objek deteksi kita juga sudah harus menyiapkan file `classes.pbtxt` karena pada file `classes.pbtxt` ini kita dapat mengetahui masing-masing objek dengan mendefinisikan pemetaan nama kelas ke nomor ID kelas. Di sini kita menggunakan sebanyak 37 class, yaitu huruf A sampai huruf Z, angka 0 hingga angka 9 dan ditambah dengan kelas plate itu sendiri.

Selanjutnya kita akan melakukan proses training data. Tujuan dari kita melakukan training data adalah agar kita mendapatkan model yang akan digunakan untuk melakukan proses testing. Sebelumnya setelah kita melakukan proses training kita akan mengekstrak terlebih dahulu hasil dari proses training, setelah itu kita mendapatkan model yang kita akan gunakan.

Pada tahap evaluasi ini kita akan membahas tentang evaluasi model dalam memprediksi data baru (cross validation) dan untuk mengukur kinerja suatu metode klasifikasi (confusion matrix)

### 5.1 Cross Validation

Pada tahap cross validation ini, peneliti menggunakan data training sebanyak 1615 data gambar, dimana data itu dibagi menjadi 5 bagian untuk dilakukan 2 kali proses training. Data tersebut terdiri dari data training sebanyak 1294 data gambar plat nomor kendaraan dan data testing sebanyak 321 data gambar plat nomor kendaraan. Gambar 14 merupakan Source Code yang digunakan untuk menjalankan Cross Validation

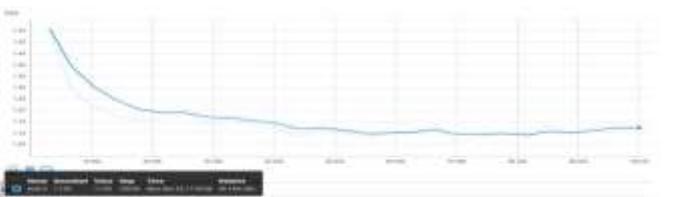
```
(Smartfarm) F:\DDS_group\tensorflow-framework-for-object-detection\models\research\object_detection>python model_main.py --model_dir=training_23_des_plat_FasterRCNN/ --pipeline_config_path=training_23_des_plat_FasterRCNN/plate_number_v2.config --num_train_steps=100000 --sample_1_of_n_eval_examples=1 --alsologtostderr
```

Gambar 14. Source Code yang digunakan untuk menjalankan Cross Validation



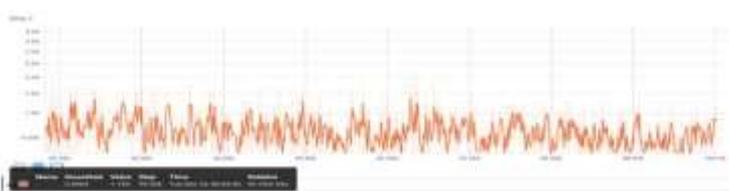
Gambar 15. Tensor board loss pada training dataset

Gambar 15 diatas merupakan tensor board loss training dataset pada pelatihan pertama. Dimana pada 100.000 step yang telah dilakukan loss nya adalah sebanyak 0.5278 sedangkan waktu yang diperlukan sekitar 4 ½ jam 10 detik.



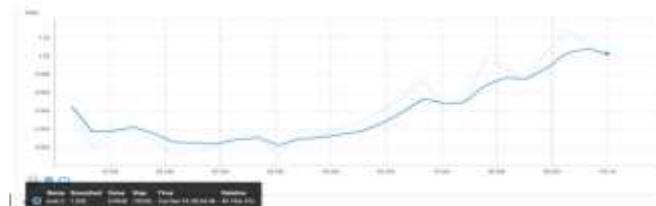
Gambar 16. Tensor board loss pada validasi dataset

Pada gambar 16 diatas merupakan Tensor board loss pada validasi dataset pada pelatihan pertama. Hasil loos dari proses evaluasi dataset pada gambar 6.3 adalah 1.125 dan waktu yang digunakan untuk validasi 100.000 step adalah 4 jam 19 menit 28 detik.



Gambar 17. Tensor board loss pada training dataset

Pada gambar 17 diatas merupakan Tensor board loss pada training dataset hasil training kedua. Dimana pada 100.000 step yang telah dilakukan loss nya adalah sebanyak 1.183 sedangkan waktu yang diperlukan sekitar 4 ½ jam 34 detik.



Gambar 18. Tensor board loss pada validasi dataset

Pada gambar 18 merupakan Tensor board loss pada validasi dataset pada pelatihan kedua. Hasil loos dari proses evaluasi dataset pada gambar 6.6 adalah 0.9948 dan waktu yang digunakan untuk validasi 100.000 step adalah 4 jam 19 menit 57 detik.

### 5.2 Confusion Matrix

Evaluasi performa dari metode klasifikasi dapat menggunakan parameter yang dikenal dengan performance metric yang terdiri dari TP, FP, FN, dan TN. Ukuran metode klasifikasi:

1. Accuracy atau tingkat pengenalan menyatakan persentase dari jumlah tuple dalam data uji yang diklasifikasikan dengan benar oleh model klasifikasi. Persamaan 1 merupakan rumus perhitungan nilai accuracy.

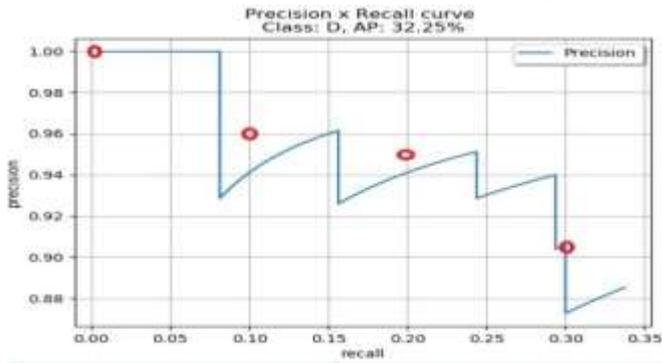
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

2. Pada tahap ini dapat dijelaskan bahwa recall dapat didefinisikan sebagai rasio dari jumlah total contoh positif yang di klasifikasikan dengan benar dan nilai recall yang baik menunjukkan klasifikasi dikenali dengan benar. Persamaan 2 merupakan rumus yang digunakan untuk menghitung nilai recall.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$

3. Pada tahap ini dapat dijelaskan bahwa nilai presisi didapatkan dari pembagian jumlah total contoh positif yang, diklasifikasikan dengan benar dengan jumlah total contoh Positif yang diprediksi. Persamaan 3 merupakan rumus yang digunakan untuk melakukan perhitungan pada nilai precision.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$



Gambar 22. Hasil Confusion Matrix pada Class D Cara menghitung AP dari kelas D adalah sebagai berikut:

$$AP = \frac{1}{11} (1 + 0,96 + 0,95 + 0,901)$$

$$AP = \frac{1}{11} (3,811)$$

$$AP = 34 \%$$

Gambar 22 merupakan hasil confusion matrix dari Class D, dimana gambar diatas terdapat precision dan recall. Jadi average precision (AP) dari class D adalah sebanyak 34%.

#### IV. KESIMPULAN

Berdasarkan penelitian yang dilakukan tentang pengenalan plat nomor kendaraan Indonesia dapat diambil kesimpulan yang diharapkan dapat memberikan jawaban terhadap tujuan dilakukannya penelitian merupakan sistem aplikasi pengenalan pola plat nomor kendaraan Indonesia yang di bangun secara umum dapat mengenali karakter plat nomor kendaraan Indonesia. Metode Convolutional Neural Network adalah metode yang tepat dalam persoalan mengidentifikasi huruf dan angka. Mendapatkan posisi mobil dan klasifikasi tiap karakter plat nomor pada gambar dapat dilakukan menggunakan topologi objek deteksi Faster R- CNN.

#### DAFTAR PUSTAKA

1. D. Avianto, "Pengenalan Pola Karakter Plat Nomor Kendaraan Menggunakan Algoritma Momentum Backpropagation Neural Network," *J. Inform.*, vol. 10, no. 1, pp. 1199–1209, Jan. 2016, doi: 10.26555/jifo.v10i1.a3352.
2. T. F. Abidin, A. A. AzZuhri, and F. Arnia, "Pengenalan Karakter Plat Nomor Kendaraan Bermotor Menggunakan Zoning dan Fitur Freeman Chain Code," *J. Rekayasa Elektr.*, vol. 14, no. 1, pp. 19–25, Apr. 2018, doi: 10.17529/jre.v14i1.8932.
3. David, "Pengenalan Pola Plat Nomor Kendaraan Menggunakan Jaringan Syaraf Tiruan Backpropagation," *J. Ilm. SISFOTENIKA*, vol. Vol. 3, no. 1, pp. 71–80, Jan. 2013.
4. A. Antonius, D. Triyanto, and I. Ruslianto, "Penerapan Pengolahan Citra Dengan Metode Adaptive Motion Detection Algorithm Pada Sistem Kamera Keamanan

- Dengan Push Notification Ke Smarthphone Android," *J. Coding Sist. Komput. Untan*, vol. 03, no. 2, pp. 54–65, 2015.
5. K. Kusumawati and D. W. Cahyadi, "Penerapan Teknologi Optical Character Recognition Untuk Mendeteksi Plat Nomor Kendaraan," *Pros. Semin. Nas. Inov. Teknol. – SNITek 2017*, pp. 12–20, Aug. 2017.
6. W. S. Eka Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, pp. A65–A69, Mar. 2016, doi: 10.12962/j23373539.v5i1.15696.
7. Giuseppe Ciaburro and Balaji Venkateswaran, *Neural Network With R*. Birmingham: Packt Publishing Ltd, 2017.
8. S. E. Limantoro, Y. Kristian, and D. D. Purwanto, "Pemanfaatan Deep Learning pada Video Dash Cam Untuk Deteksi Pengendara Sepeda Motor," *J. Nas. Tek. Elektro dan Teknol. Inf. JNTETI*, vol. 7, no. 2, pp. 167–173, Jun. 2018, doi: 10.22146/jnteti.v7i2.419.
9. David Molin, "Pedestrian detection using convolutional neural networks," *Linköping*, 2015, pp. 1–69.
10. Muhammad Zufar and Budi Setiyono, "Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real- Time," *SAINS DAN SENI ITS*, vol. 5, no. 2, p. A-72-A-77, 2016.
11. Tutut Furi Kusumaningrum, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Jamur Konsumsi di Indonesia Menggunakan Keras," *Yogyakarta*, 2018, p. 81.
12. C. Prianto and N. S. Harani, "The data mining analysis to determine the priorities of families who receiving assistance," *J. Phys. Conf. Ser.*, vol. 1280, p. 8, Nov. 2019, doi: 10.1088/1742-6596/1280/2/022027.
13. Imam Digmi, "Indonesian Plate Number Plate number of Indramayu / Cirebon Area (E)," *Indonesian Plate Number of Indramayu / Cirebon Area (E)*, 25-Jul-2018. [Online]. Available: <https://www.kaggle.com/imamdigmi/indonesian-plate-number>.
14. Tilik.ID, "Warna Dasar Plat Nomor Kendaraan di Indonesia Akan Berubah," *Warna Dasar Plat Nomor Kendaraan di Indonesia Akan Berubah*. [Online]. Available: <http://www.tilik.id/read/2018/07/26/5613/warna-dasar-plat-nomor-kendaraan-di-indonesia-akan-berubah>. [Accessed: 09-Jan-2020]