

Implementasi *Convolutional Recurrent Neural Network* untuk Identifikasi Plat Nomor Mobil pada Sistem Parkir Otomatis

WINARNO SUGENG, THETA DINNARWATY PUTRI, FAKHRUDIN RIZKY HUSAINI

Program Studi Informatika, Institut Teknologi Nasional, Bandung
Email: winarno@itenas.ac.id

Received 26 Juni 2023 | *Revised* 23 Juli 2023 | *Accepted* 5 Agustus 2023

ABSTRAK

Beberapa tempat di kota Bandung telah menggunakan sistem otomatis dalam proses pelayanan sistem parkir mobil, namun sistem parkir tersebut masih menggunakan tiket parkir yang digunakan sebagai tanda kendaraan memasuki area parkir. Banyak orang yang mengalami kehilangan tiket parkir, sehingga perlu ada penanganan terhadap masalah tersebut. Dengan adanya masalah tersebut, teknologi Optical Character Recognition (OCR) dapat digunakan sebagai pembaharuan untuk mengenali karakter pada plat nomor mobil sehingga tiket parkir tidak diperlukan lagi. Metode yang digunakan sebagai deteksi teks yaitu Character Region Awareness for Text detection (CRAFT) dan metode untuk pengenalan teks yaitu Convolutional Recurrent Neural Network (CRNN). Hasil training model dengan dataset synthetic yang dibuat dari python library telah menghasilkan model terbaik yang memiliki akurasi sebesar 85,73% terhadap 1500 citra uji synthetic dan akurasi sebesar 77,5 % terhadap 40 citra plat nomor asli yang didapat dari google image dan Kaggle.com.

Kata kunci: CRAFT, CRNN, sistem parkir otomatis, real-time

ABSTRACT

Several places in the city of Bandung have used an automated system in the process of servicing the car parking system, but the parking system still uses a parking ticket which is used as a sign for vehicles entering the parking area. Many people experience lost parking tickets, so there needs to be a solution to this problem. Given this problem, Optical Character Recognition (OCR) technology can be used as an update to recognize characters on car license plates so that parking tickets are no longer needed. The method used for text detection is Character Region Awareness for Text detection (CRAFT) and the method for text recognition is Convolutional Recurrent Neural Network (CRNN). The results of model training with a synthetic dataset made from the python library have produced the best model which has an accuracy of 85.73% for 1500 synthetic test images and an accuracy of 77.5% for 40 original license plate images obtained from Google Image and Kaggle.com.

Keywords: CRAFT, CRNN, automatic parking system, real-time

1. PENDAHULUAN

Dengan seiring berkembangnya teknologi, sarana sistem parkir mobil telah dapat bekerja secara otomatis. Kemudahan dari teknologi tersebut telah diimplementasikan pada beberapa tempat yang memiliki sistem pelayanan parkir mobil. Saat ini di beberapa tempat besar seperti rumah sakit dan mall sudah menggunakan kamera dan komputer, namun tetap saja pencatatan dan pengecekan plat nomor kendaraan masih dilakukan secara manual oleh petugas parkir dengan melakukan pencatatan pada pintu masuk dan keluar (**Munawar, dkk, 2021**). Permasalahan yang terjadi adalah beberapa orang yang mengalami kehilangan tiket parkir, sehingga menjadi masalah mobil tidak bisa keluar dan harus berurusan dengan petugas parkir, dalam hal ini perlu ada penanganan terhadap masalah tersebut.

Optical Character Recognition (OCR) adalah algoritma yang berfungsi untuk memindai citra dan dijadikan teks (**Ismail, dkk, 2020**). Saat ini teknik OCR telah diintegrasikan dengan *deep learning* untuk meningkatkan akurasi dalam mengenali teks. Deep learning memiliki kemampuan yang sangat baik dalam visi komputer dengan kapabilitasnya yang signifikan dalam memodelkan berbagai data kompleks seperti data gambar (**Harani, dkk, 2019**). Salah satu metode dengan pendekatan *deep learning* yaitu *Character-Region Awareness for Text detection* (CRAFT) dan *Convolutional-Recurrent Neural Network* (CRNN).

Pada penelitian tahun 2017, telah melakukan eksperimen dengan mengusulkan arsitektur jaringan saraf baru, yang disebut *Convolutional Recurrent Neural Network* (CRNN). Arsitektur tersebut mengintegrasikan keunggulan kedua jaringan saraf *Convolutional Neural Network* (CNN) dan *Recurrent Neural Network* (RNN). CRNN mampu mengambil citra masukan dan menghasilkan prediksi yang panjang, seperti kata-kata (**Shi, dkk, 2017**). Dan pada tahun 2019 telah dilakukan penelitian yang mengusulkan detektor teks baru yang disebut *Character Region Awareness for Text Detection* (CRAFT). Hasil dari penelitian tersebut dapat mendeteksi area teks secara efektif dengan mengelompokkan setiap karakter menjadi satu. Dengan begitu, berbagai bentuk teks tetap dapat dideteksi oleh CRAFT (**Baek, dkk, 2019**). Dengan menggabungkan kedua penelitian tersebut dapat menjadi suatu sistem yang dapat mengenali sebuah teks pada plat nomor mobil, dalam hal ini plat nomor mobil akan berfungsi sebagai pengenalan mobil saat parkir, ini merupakan solusi untuk menggantikan fungsi karcis parkir sebagai tanda pengenalan kendaraan.

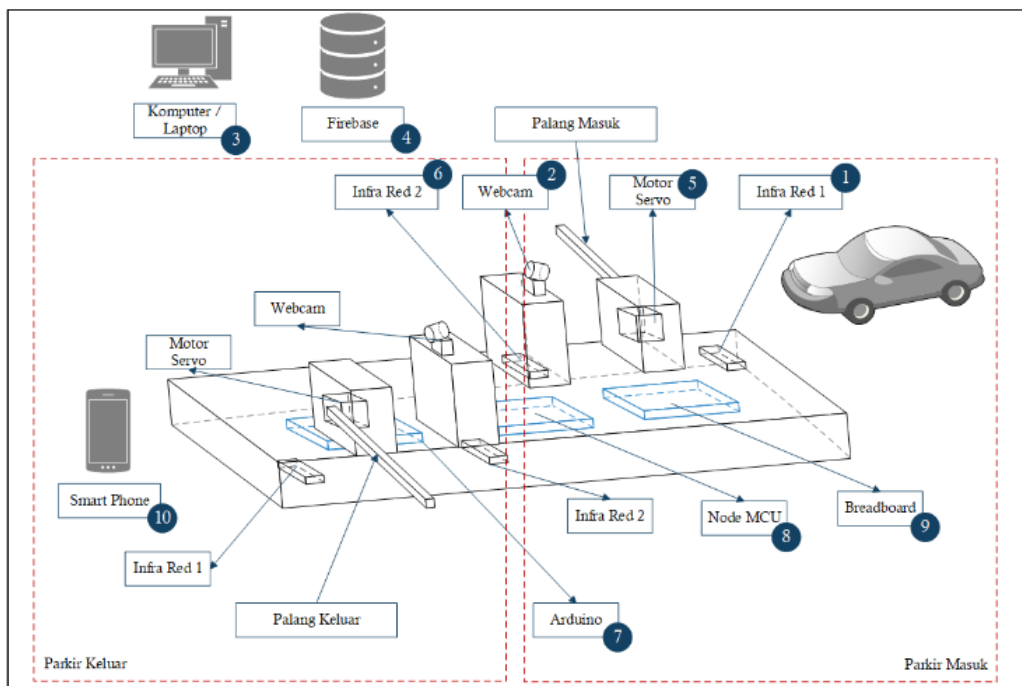
Plat nomor kendaraan merupakan sebuah identitas dari sebuah kendaraan, pada plat nomor kendaraan terdapat gabungan huruf dan angka (**Ramadhani, 2021**). Plat nomor kendaraan tersebut dikeluarkan oleh instansi yang berwenang sebagai tanda izin untuk kendaraan tersebut agar dapat digunakan di jalan umum. Setiap plat nomor kendaraan memiliki nomor seri yang berbeda dan dikelompokkan berdasarkan provinsi yang terdaftar di wilayah Indonesia. Kode huruf sebelum nomor kendaraan menunjukkan wilayah provinsi pendaftaran kendaraan, dan kode huruf setelah nomor kendaraan menunjukkan subwilayah provinsi tempat kendaraan tersebut didaftarkan (**Aprilino & Al Amin, 2022**). Setiap kendaraan memiliki plat nomor, sehingga dapat digunakan sebagai tanda masuk tempat parkir sebagai pengganti tiket parkir.

Berdasarkan uraian sebelumnya, dalam penelitian yang dilakukan pembaharuan sistem parkir dengan mengganti tiket parkir menjadi pengenalan karakter pada plat nomor kendaraan sebagai tanda masuk tempat parkir, kondisi ini perbaikan dari penelitian sebelumnya yang dilakukan oleh Munawar dkk yang telah disampaikan pada paragraf awal. Metode pengenalan karakter plat nomor yang digunakan pada sistem ini menggunakan metode *Character-Region Awareness for Text detection* (CRAFT) sebagai pendeteksi setiap bagian kode plat

(Youngmin, dkk, 2019) (Fcakyon, 2022) dan metode *Convolutional-Recurrent Neural Network* (CRNN) sebagai pengenalan karakter untuk setiap bagian kode plat (Bgshih, 2018). Adapun mikrokontroler yang digunakan sebagai penggerak palang pintu parkir adalah *Arduino Uno* dan *NodeMCU* sebagai modul wifi yang dapat menerima data dari *Firebase* melalui jaringan. Sebagai pemahaman dasar Mikrokontroler merupakan komputer kecil berbentuk chip IC (*Integrated Circuit*) untuk melakukan operasi tertentu. Mikrokontroler sistem mikroprosesor lengkap yang terkandung di dalam sebuah chip yang dapat diprogram (Al Khairi, 2021). *Arduino Uno* adalah papan sirkuit berbasis mikrokontroler, memiliki input/output digital, analog input, resonator kristal keramik 16 MHz, Koneksi USB, soket adaptor, pin header ICSP, dan tombol reset (Ihsan, 2023). *NodeMCU ESP32* merupakan mikrokontroler dengan ketersediaan modul WiFi dan Bluetooth dalam chip sehingga mendukung sistem *aplikasi Internet of Things* (Iqbal, 2022). Kontribusi dari hasil penelitian adalah sistem dapat mengurangi kemacetan antrian parkir dengan mengganti pencacatan plat nomor kendaraan secara manual menjadi secara otomatis yang berakibat kemacetan jalan raya dapat dihidari.

2. METODOLOGI PENELITIAN

Pada bagian ini dibahas mengenai rancangan sistem yang dibangun. Dalam melakukan pembaharuan sistem parkir, terdapat beberapa komponen yang dibutuhkan agar sistem dapat berfungsi dengan baik. Gambaran sistem beserta komponen-komponen yang dibutuhkan dapat dilihat pada Gambar1.



Gambar 1. Gambaran Sistem

Pada Gambar 1 terdapat 2 bagian yaitu bagian parkir masuk pada sebelah kanan dan bagian parkir keluar pada sebelah kiri. Berikut penjelasan dari beberapa komponen utama yang diperlukan dalam membangun sistem.

1. Komponen pertama yaitu *infrared* 1, komponen ini digunakan sebagai pendeteksi jika terdapat mobil,
2. Webcam, komponen ini digunakan sebagai pendeteksi plat nomor mobil,
3. Komputer, komponen ini digunakan sebagai penerima data citra plat nomor dari webcam dan data tersebut diproses untuk dilakukan pengenalan karakter pada plat nomor,

4. *Firebase*, *firebase* merupakan layanan *realtime database* yang disediakan oleh *Google* (Lyalin, 2023). Pada sistem ini *firebase* digunakan sebagai penyimpanan data (Wicaksono, 2020), seperti data parkir masuk dan keluar, status deteksi plat, status portal membuka atau tidak, dan data akun untuk setiap kartu,
5. Motor servo, komponen ini digunakan sebagai penggerak portal masuk maupun keluar,
6. *Infrared 2*, komponen ini digunakan sebagai pendeteksi mobil ketika mobil sudah melewati portal,
7. *Arduino Uno*, komponen ini merupakan mikrokontroler yang berfungsi sebagai pengendali motor servo dan *infrared*,
8. *NodeMCU*, komponen ini berfungsi sebagai pengirim koneksi antara komputer dan *Arduino* agar saling terhubung,
9. *Breadboard*, komponen ini merupakan sebuah *board* yang berfungsi untuk merancang sebuah rangkaian elektronik agar dapat saling terhubung menggunakan kabel *jumper*,
10. *Smartphone*, *smartphone* yang digunakan harus memiliki *NFC Reader* agar dapat membaca kartu *NFC*. Kartu *NFC* digunakan sebagai simulasi pembayaran seperti *E-money*.

Komponen-komponen yang telah dipaparkan sebelumnya merupakan komponen utama yang digunakan pada pembaharuan sistem parkir ini. Sistem parkir masuk dimulai dari komponen *infrared 1* yang digunakan untuk memberitahu jika mobil mendekati palang parkir. Setelah *infrared 1* mendeteksi adanya mobil, *Arduino Uno* mengirim status deteksi plat sama dengan 1, selanjutnya *NodeMCU* menerima status tersebut dan mengirimkan status ke *Firebase*. Kemudian sistem mengambil status deteksi plat untuk memerintahkan webcam agar memulai mengambil *video* secara *realtime*. Jika status deteksi sama dengan 1 maka sistem mulai memproses pengenalan karakter plat nomor, dan jika 0, sistem tidak akan melakukan pemrosesan. Setelah karakter pada plat nomor dikenali, sistem akan mengirim karakter yang dikenali tersebut ke *firebase* beserta waktu mobil memasuki parkir, mengubah status deteksi menjadi 0, dan mengubah status palang menjadi 1. *NodeMCU* kemudian mengambil status palang dan mengirimkan status tersebut ke *Arduino Uno* untuk membuka palang parkir. Jika mobil telah melewati palang dan *infrared2* mendeteksi adanya pergerakan mobil, *Arduino Uno* menutup palang dengan mengubah status palang sama dengan 0 dan mengirimkan ke *NodeMCU*. Status yang diterima *NodeMCU* kemudian dikirim ke *firebase*.

Cara kerja sistem yang telah dipaparkan sebelumnya merupakan cara kerja sistem parkir masuk. Untuk cara kerja sistem parkir keluar terdapat perbedaan yaitu ketika sistem telah mengenali karakter pada plat nomor, pengendara perlu menempelkan kartu terlebih dahulu pada *NFC Reader*. Jika kartu pengendara tidak terdaftar atau nomor plat pada kartu tidak sesuai dengan plat nomor yang dikenali, maka palang tidak akan terbuka. Oleh sebab itu, sebelum menggunakan layanan sistem parkir, pengendara harus mendaftarkan kartu terlebih dahulu dengan memasukkan nomor plat kendaraan yang digunakan.

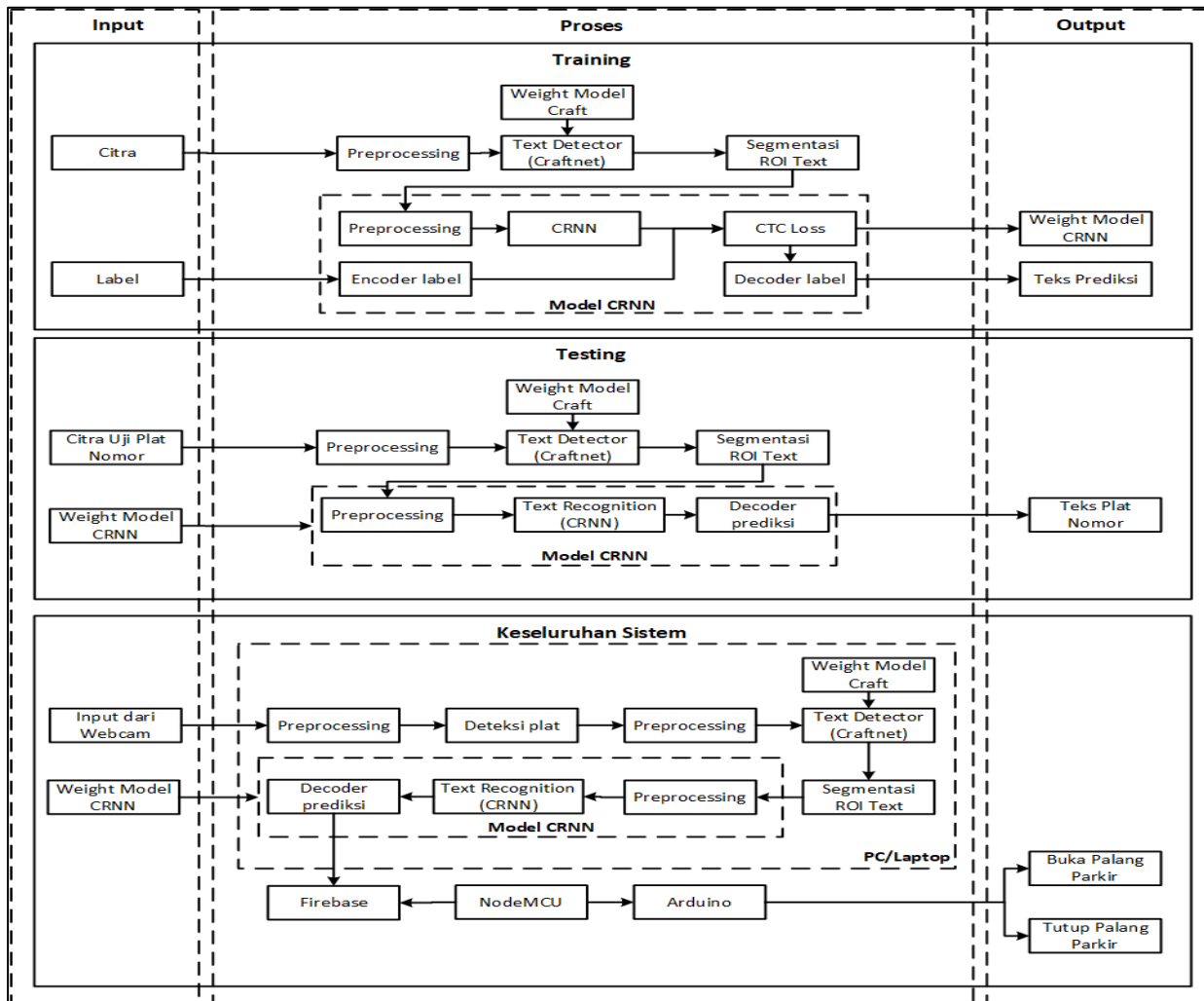
2.1. Desain Sistem

Pada desain sistem terdapat blok diagram sistem dan arsitektur model yang digunakan dalam pengenalan karakter pada plat nomor, dan *dataset* yang digunakan untuk melatih model.

Pada Gambar 2 telah dipaparkan blok diagram sistem mengenai proses *training* model, *testing* model, dan keseluruhan sistem. Blok diagram sistem pada gambar 2 merupakan tahapan pembuatan model CRNN untuk melakukan pengenalan karakter pada plat nomor mobil dan pembuatan model CRNN.

Berikut penjelasan dari masing-masing tahapan pada blok diagram sistem yang sudah dipaparkan. Pada tahapan *training* terdapat *input*, *proses* dan *output* dengan penjelasan berikut.

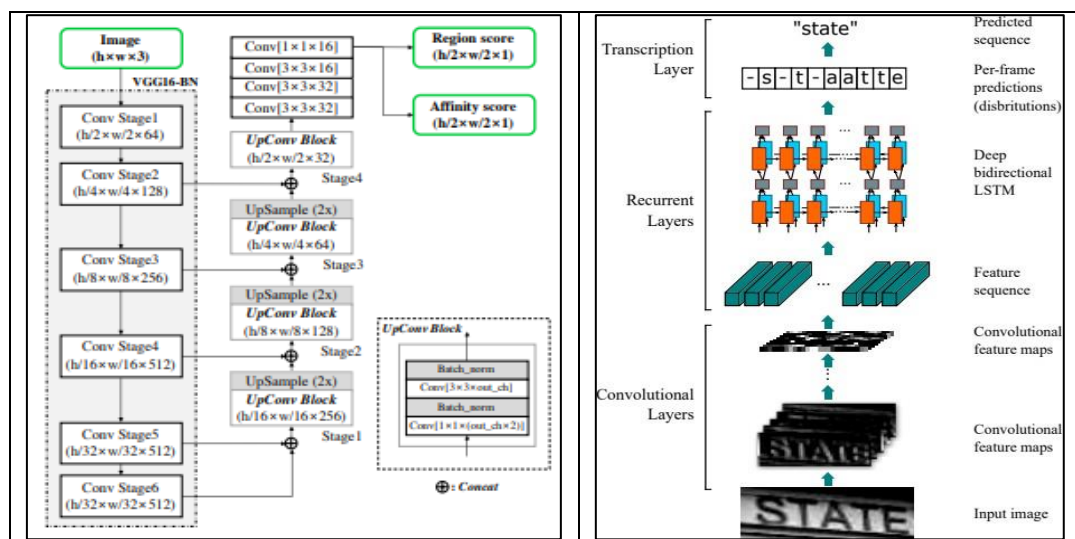
1. *Input*, pada tahap pertama, *dataset* dipersiapkan terlebih dahulu sebagai data latih model, *dataset* ini terdiri dari beberapa citra angka atau huruf beserta label yang berkaitan dengan masing-masing citra,



Gambar 2. Blok Diagram Sistem

2. Proses, pada tahap proses terdapat dua proses *preprocessing* yaitu tahap *preprocessing* untuk CRAFT model dan untuk CRNN model mengikuti arsitektur pada Gambar 3. Pada tahap proses *preprocessing* citra yang pertama dilakukan perubahan ukuran lebar dan tinggi citra menjadi kelipatan 32 dari ukuran asli citra, dan menormalisasi citra. Hasil citra yang sudah di *preprocessing* selanjutnya dimasukkan ke dalam CRAFT untuk mengetahui tata letak karakter pada citra dan *weight* CRAFT ini menggunakan *weight pretrained*. Tata letak karakter pada citra tersebut selanjutnya di lakukan segmentasi ROI menjadi 3 bagian yaitu kode plat, nomor polisi, dan kode wilayah. Masing-masing citra tersebut dilakukan *preprocessing* kembali untuk diproses oleh CRNN. Pada tahap proses *preprocessing* ini dilakukan perubahan citra menjadi *grayscale*, perubahan ukuran citra menjadi 100px x 50px, dan menormalisasi citra sehingga citra memiliki nilai negatif dan positif. Masing-masing citra yang sudah di *preprocessing* selanjutnya diproses oleh CRNN untuk mengetahui prediksi dari citra *dataset*. Sebelum proses *encoder* label, dilakukan terlebih

dahulu pembuatan kamus yang didalamnya terdapat huruf dan angka A sampai Z dan 0 sampai 9 beserta index dari masing-masing karakter. Selanjutnya pada proses *encoder* label, label dataset tersebut dipecah menjadi beberapa karakter dan melaraskan karakter tersebut dengan kamus yang sudah dibuat sebelumnya untuk mengetahui index dari tiap karakter pada label dan jumlah karakter dari label. Hasil dari prediksi CRNN, ukuran prediksi, hasil dari *encoder label* yang berisi index dari tiap karakter pada label dan jumlah karakter dari label di proses oleh *CTC loss* dengan melaraskan hasil dan ukuran prediksi dengan hasil. Kemudian hasil prediksi yang sudah ditingkatkan akurasi oleh *CTC loss* tersebut akan dilakukan proses *decoder* label dengan melaraskan index dari tiap karakter pada hasil prediksi dengan index dari tiap karakter pada kamus untuk mengetahui karakter aslinya.



Gambar 3. Arsitektur CRAFT (Baek, Lee, Han, Yun, & Lee, 2019) dan CRNN (Shi, Bai, & Yao, 2017)

3. *Output*, pada tahap ini terdapat 2 keluaran yaitu *weight* dari model yang sudah di latih dan hasil prediksi yang sudah di *decoder*.

Weight dari model yang sudah di latih tersebut selanjutnya, dilakukan testing untuk memastikan *weight* model tersebut sudah dapat digunakan atau belum. Berikut penjelasan tahapan testing dari model yang sudah dilatih:

1. *Input*, pada tahap pertama, citra uji plat nomor dipersiapkan terlebih dahulu sebagai citra yang akan diuji dan *weight* terbaik dari model yang sudah dilatih, citra uji ini merupakan citra yang berbeda dari citra *dataset* latih,
2. Proses, pada tahap ini sama seperti tahap proses *training* namun terdapat perbedaan yaitu hasil prediksi dari CRNN langsung dilakukan proses *decoder label* untuk mengetahui karakter aslinya.
3. *Output*, pada tahap ini terdapat keluaran yaitu hasil prediksi yang sudah di *decoder* berupa nomor plat kendaraan.

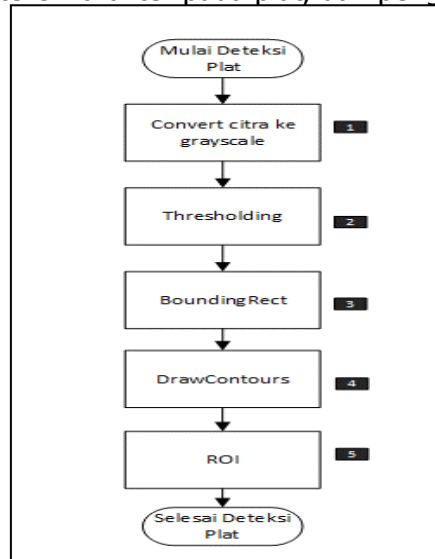
Pada blok diagram keseluruhan sistem dipaparkan tahapan keseluruhan sistem dengan *weight* dari model CRNN yang sudah layak digunakan. Berikut penjelasan dari blok diagram tahapan keseluruhan sistem.

1. *Input*, pada tahap pertama terdapat masukan yaitu *weight* CRNN model yang sudah di latih dan citra yang didapat dari tiap frame pada *webcam*,

2. Proses, pada tahap proses dilakukan proses *preprocessing* citra terlebih dahulu dengan dilakukan perubahan citra menjadi *grayscale* dan menormalisasi citra untuk dilakukan proses deteksi plat. Selanjutnya dilakukan pencarian kontur yang memiliki 4 sudut untuk mengambil plat nomor pada citra. Hasil pencarian kontur tersebut kemudian dilakukan *cropping* dan kemudian dilakukan proses prediksi karakter pada plat nomor, proses tersebut sama seperti proses *testing*. Hasil prediksi tersebut kemudian dikirimkan ke *firebase* beserta tanggal dan jam saat proses berlangsung serta mengubah nilai status portal menjadi 1, nilai 1 merupakan status portal terbuka dan nilai 0 status portal tertutup. Selanjutnya pada NodeMCU dilakukan proses menghubungkan koneksi wifi dan *firebase* untuk mengambil nilai status portal. Nilai status portal tersebut dikirimkan ke *Arduino Uno* untuk dilakukan pengkondisian jika nilai status portal 0 atau 1. Jika nilai status portal sama dengan 1 maka motor servo akan berputar 90 derajat untuk membuka portal dan jika nilai 0 maka motor servo berputar kembali seperti semula. Pada saat kondisi portal terbuka atau nilai status portal sama dengan 1 terdapat kondisi untuk menutup portal jika terdapat halangan pada *sensor infrared* dan proses ini di proses oleh Arduino.
3. *Output*, pada tahap ini terdapat 2 keluaran yaitu pergerakan pada aktuator *motor servo* untuk membuka dan menutup portal.

Metode yang digunakan untuk pengenalan teks yaitu metode *Convolutional Recurrent Neural Network*. Agar sistem dapat berjalan dengan sesuai, diperlukan tahapan *training* dan *testing* seperti yang sudah dijelaskan sebelumnya. Untuk melakukan pelatihan *model*, perlu ditentukan arsitektur yang akan digunakan. Arsitektur tersebut telah ditetapkan oleh Baoguang shi, Xiang Bai dan Cong Yao pada tahun 2017.

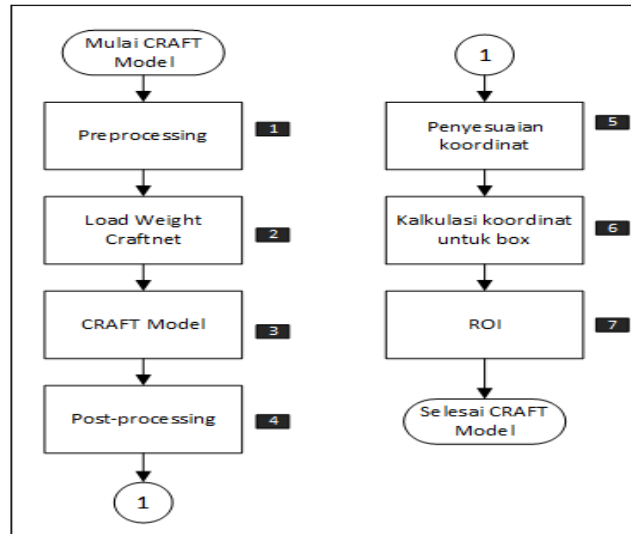
Pada sistem pengenalan plat nomor seperti terlihat pada Gambar 4 terdapat beberapa *proses* yaitu, proses deteksi plat, deteksi karakter pada plat, dan pengenalan karakter pada plat.



Gambar 4. Proses Deteksi Plat

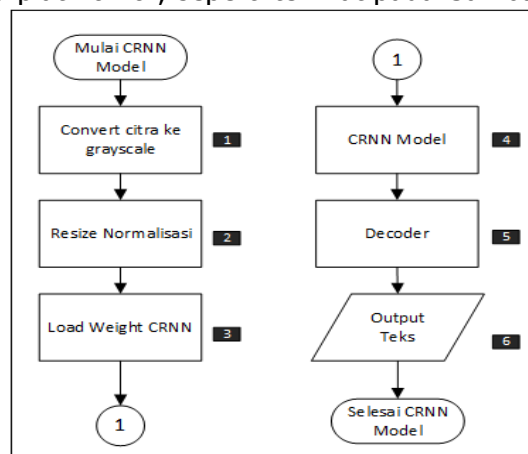
Deteksi karakter menggunakan CRAFT model, seperti terlihat pada Gambar 5.

Implementasi *Convolutional Recurrent Neural Network* untuk Identifikasi Plat Nomor Mobil pada Sistem Parkir Otomatis



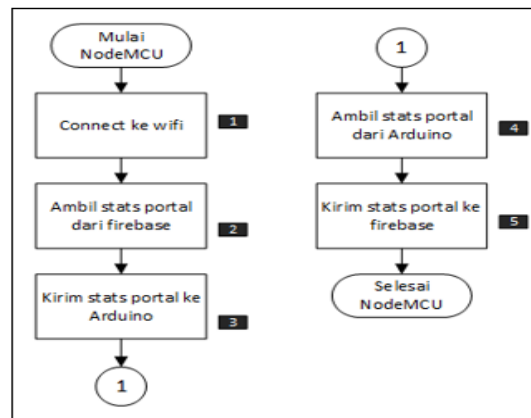
Gambar 5. Flowchart Deteksi Karakter pada Plat Nomor

Proses pengenalan karakter pada plat nomor menggunakan CRNN model. Berikut proses pengenalan karakter pada plat nomor, seperti terlihat pada Gambar 6.



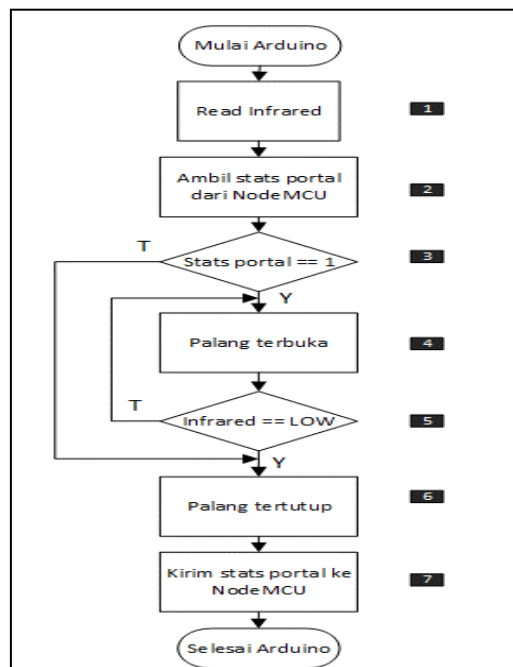
Gambar 6. Flowchart Pengenalan Karakter

Proses NodeMCU esp32 modul wifi yang digunakan untuk menghubungkan Arduino dengan firebase, seperti terlihat pada Gambar 7.



Gambar 7. Flowchart NodeMCU

Pada Gambar 8 merupakan pengendalian sensor dan aktuator dalam hal ini buka tutup palang portal.



Gambar 8. Pengendalian Sensor dan Aktuator

3. HASIL DAN PEMBAHASAN

Dalam pembuatan pembaharuan sistem parkir ini digunakan bahasa C untuk memprogram *Arduino Uno*, *python* untuk memprogram deteksi dan pengenalan plat nomor mobil, *html* digunakan sebagai tampilan untuk deteksi dan pengenalan plat nomor, dan *java* digunakan untuk memprogram registrasi kartu dan *tap* kartu nfc yang diimplimentasikan pada *mobile smartphone*.

Software yang digunakan untuk membangun sistem yaitu *Android Studio* dengan bahasa pemrograman *java*, *Pycharm* dengan bahasa *python*, dan *Arduino IDE* dengan bahasa C. Adapun *hardware* yang digunakan yaitu *laptop*, *Arduino*, *nodeMCU*, *breadboard*, dua buah *webcam*, dua buah *motor servo*, 4 buah *infrared*, dan akrilik sebagai kerangka sistem parkir. Pada pembaharuan sistem ini juga diperlukan sebuah *database realtime* yaitu *Firebase* yang disediakan oleh *Google*.

3.1. Pengujian Alfa

Proses pengujian dibagi menjadi dua bagian, pengujian fungsionalitas tiap sub komponen yang digunakan dan pengujian keseluruhan sistem. Pengujian fungsionalitas terdiri dari : registrasi Kartu berbasis NFC, deteksi plat nomor polisi, pengujian deteksi karakter plat nomor polisi.

Pengujian registrasi kartu berbasis NFC, dimulai dari melakukan tap kartu NFC ke sistem kemudian membaca tag ID dari kartu tersebut, dilanjutkan memberikan identitas nama pemilik kendaraan dan nomor plat kendaraan , kemudian dilakukan *upload* data ke *firebase*

Pengujian sistem pengenalan plat nomor terdapat beberapa *proses* sesuai dengan proses pada Gambar 3 yaitu, proses deteksi plat, deteksi karakter pada plat, dan pengenalan karakter pada plat.

1. Pada tahap awal terdapat proses konversi citra asli menjadi *grayscale* yang dapat dilihat pada Gambar 9 merupakan citra asli dan citra yang sudah dikonversi menjadi *grayscale*.

Implementasi *Convolutional Recurrent Neural Network* untuk Identifikasi Plat Nomor Mobil pada Sistem Parkir Otomatis



Gambar 9. Citra Asli dan Citra *Grayscale*

2. Tahap selanjutnya terdapat proses transformasi citra yaitu proses *thresholding*, dengan mengubah bentuk citra *grayscale* menjadi citra *biner*. Hasil proses transformasi citra dapat dilihat pada Gambar 10 yang menjadi citra hitam putih.



Gambar 10. Citra Hitam Putih

3. Tahap selanjutnya dilakukan pencarian kontur yang memiliki 4 sudut pada setiap area citra hasil proses *preprocessing* dengan mencari koordinat (x, y) dengan proses *BoundingRect*.
4. Koordinat yang telah didapat sebelumnya digunakan sebagai titik yang akan dilakukan proses pembuatan garis berwarna hijau dalam bentuk persegi. Gambar 11 merupakan hasil citra asli yang telah dilakukan proses *drawcontours*.



Gambar 11. Hasil Pembuatan Garis

5. Titik koordinat sebelumnya juga digunakan untuk proses segmentasi citra atau *Region of Interest* (ROI). Gambar 12 merupakan hasil dari proses ROI.



Gambar 12. Hasil *Region of Interest*

Selanjutnya terdapat pengujian deteksi karakter menggunakan CRAFT model sesuai dengan proses Gambar 4

1. Tahap awal terdapat proses *preprocessing*, pada proses ini dilakukan perubahan ukuran citra dan menormalisasikan citra dalam bentuk tensor. Dapat dilihat pada Gambar 13 citra yang sudah dinormalisasi dalam bentuk tensor.

```
tensor([[[[ 1.2043,  1.2043,  1.2043, ...,  1.5810,  1.5810,  1.5810],
 [ 1.2043,  1.2043,  1.2043, ...,  1.5810,  1.5810,  1.5810],
 [ 1.2043,  1.2043,  1.2043, ...,  1.5810,  1.5810,  1.5810],
 ...,
 [-2.1179, -2.1179, -2.1179, ..., -2.1179, -2.1179, -2.1179],
 [-2.1179, -2.1179, -2.1179, ..., -2.1179, -2.1179, -2.1179],
 [-2.1179, -2.1179, -2.1179, ..., -2.1179, -2.1179, -2.1179]]],
 [[ [ 1.2381,  1.2381,  1.2381, ...,  1.5882,  1.5882,  1.5882],
 [ 1.2381,  1.2381,  1.2381, ...,  1.5882,  1.5882,  1.5882],
 [ 1.2381,  1.2381,  1.2381, ...,  1.5882,  1.5882,  1.5882],
 ...,
 [-2.0357, -2.0357, -2.0357, ..., -2.0357, -2.0357, -2.0357],
 [-2.0357, -2.0357, -2.0357, ..., -2.0357, -2.0357, -2.0357],
 [-2.0357, -2.0357, -2.0357, ..., -2.0357, -2.0357, -2.0357]]],
 [[ [ 1.1759,  1.1759,  1.1759, ...,  1.5768,  1.5768,  1.5768],
 [ 1.1759,  1.1759,  1.1759, ...,  1.5768,  1.5768,  1.5768],
 [ 1.1759,  1.1759,  1.1759, ...,  1.5768,  1.5768,  1.5768],
 ...,
 [-1.8044, -1.8044, -1.8044, ..., -1.8044, -1.8044, -1.8044],
 [-1.8044, -1.8044, -1.8044, ..., -1.8044, -1.8044, -1.8044],
 [-1.8044, -1.8044, -1.8044, ..., -1.8044, -1.8044, -1.8044]]]])
```

Gambar 13. Hasil Proses *Preprocessing* dalam Bentuk Tensor

2. Tahap selanjutnya terdapat proses memuat bobot dari CRAFT model yang telah di training. Tahap ini dilakukan di balik layar.
3. Tahap CRAFT model juga sama seperti tahap sebelumnya yang dilakukan di balik layar. Tahap ini merupakan tahap dimana model memprediksi tata letak karakter.
4. Pada tahap *post-processing* ini dilakukan proses *thresholding* untuk memfilter prediksi dari hasil CRAFT model agar lebih akurat. Berikut hasil dari proses *post-processing* pada Gambar 14.

```
[array([[420.    ,  10.    ],
 [629.    ,  10.    ],
 [629.    , 143.99997],
 [420.    , 143.99997]], dtype=float32), array([[116.82887,  12.610857],
 [367.21268,  24.53392],
 [361.26697, 149.39366 ],
 [110.88236, 137.4706  ]], dtype=float32), array([[ 8.,  26.],
 [ 81.,  26.],
 [ 81., 144.],
 [ 8., 144.]], dtype=float32)]
```

Gambar 14. Hasil Proses *Post-Processing*

5. Tahap selanjutnya Gambar 15 dilakukan penyesuaian koordinat hasil *post-processing* dengan rasio tinggi dan lebar citra. Masing-masing nilai tersebut dikalikan. Hasil dari tahap ini adalah sebagai berikut.

```
[[ [120.09375  2.859375 ]
 [179.85469  2.859375 ]
 [179.85469  41.17499 ]
 [120.09375  41.17499 ]

 [ 33.405525  3.605917 ]
 [104.99988  7.0151677]
 [103.299774  42.71725 ]
 [ 31.705425  39.308   ]

 [ 2.2875  7.434375 ]
 [ 23.160938  7.434375 ]
 [ 23.160938  41.175   ]
 [ 2.2875  41.175   ]]]
```

Gambar 15. Hasil Penyesuaian Koordinat

6. Tahap selanjutnya Gambar 16 dilakukan proses kalkulasi koordinat untuk pembuatan box. Tahap ini dilakukan untuk mencari rasio dari hasil penyesuaian koordinat sebelumnya dibagi dengan lebar dan tinggi citra. Hasil tahap ini sebagai berikut.

Implementasi *Convolutional Recurrent Neural Network* untuk Identifikasi Plat Nomor Mobil pada Sistem Parkir Otomatis

```
[[[0.65625 0.04846398]
 [0.98281252 0.04846398]
 [0.98281252 0.69788121]
 [0.65625 0.69788121]]

 [[0.18254385 0.06111724]
 [0.57376982 0.11890115]
 [0.56447964 0.7240212 ]
 [0.17325369 0.66623727]]

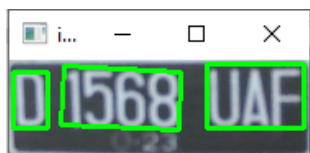
 [[0.0125 0.12600635]
 [0.1265625 0.12600635]
 [0.1265625 0.69788134]
 [0.0125 0.69788134]]]
```

Gambar 16. Hasil Kalkulasi Koordinat

7. Tahap terakhir Gambar 17 yaitu *Region of Interest* (ROI) atau segmentasi citra sesuai koordinat yang telah didapatkan sebelumnya. Pada Gambar 18 tahap dilakukan penggambaran garis (*draw contours*) dan segmentasi citra. Hasil dari masing-masing tersebut sebagai berikut.



Gambar 17. Hasil *Region of Interest*



Gambar 18. Hasil Pembuatan Garis

Pengujian selanjutnya yaitu pengujian pengenalan karakter pada plat nomor menggunakan CRNN model seperti proses pada Gambar 5. Berikut proses pengenalan karakter pada plat nomor.

1. Pada Gambar 19 tahap awal terdapat proses *preprocessing* yang dilakukan pada tahap 1 dan 2. Pada tahap yang pertama terdapat proses mengubah masing-masing citra yang telah disegmentasi menjadi citra *grayscale*. Berikut hasil citra yang telah dikonversi.

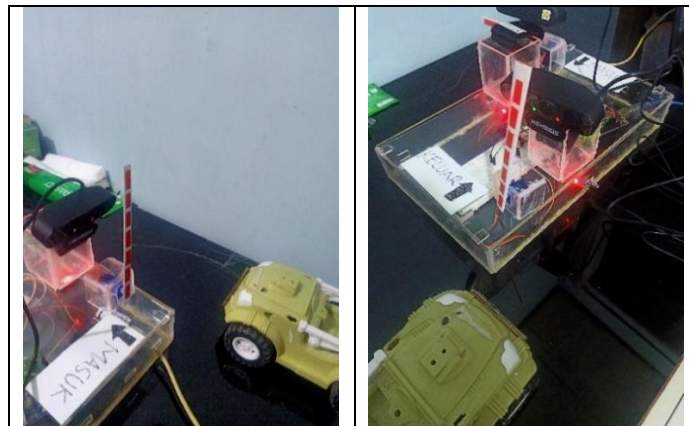


Gambar 19. Hasil Masing-Masing Citra *Grayscale*

2. Proses Gambar 20 *preprocessing* selanjutnya yaitu proses *resize* normalisasi. Pada tahap ini dilakukan proses perubahan ukuran dan menormalisasikan citra dalam bentuk tensor. Berikut hasil masing-masing citra yang telah dilakukan proses *resize* normalisasi dalam bentuk tensor.

```
tensor([[[[-0.1765, -0.1765, -0.1765, ..., -0.4118, -0.4118, -0.4118],
 [-0.1137, -0.1137, -0.1137, ..., -0.4039, -0.4039, -0.4039],
 [-0.1137, -0.1137, -0.1137, ..., -0.4039, -0.4039, -0.4039],
 ...,
 [-0.1373, -0.1373, -0.1373, ..., -0.4510, -0.4510, -0.4510],
 [-0.1294, -0.1294, -0.1294, ..., -0.4667, -0.4667, -0.4667],
 [-0.0902, -0.0902, -0.0902, ..., -0.4667, -0.4667, -0.4667]]]])
tensor([[[[-0.3255, -0.3333, -0.3333, ..., -0.4196, -0.4275, -0.4196],
 [-0.3725, -0.3725, -0.3647, ..., -0.4196, -0.4196, -0.4118],
 [-0.3804, -0.3569, -0.3333, ..., -0.3882, -0.4039, -0.4196],
 ...,
 [-0.4631, -0.4353, -0.4353, ..., -0.4510, -0.4745, -0.4745],
 [-0.4510, -0.4510, -0.4510, ..., -0.4745, -0.4745, -0.4824],
 [-0.4588, -0.4588, -0.4588, ..., -0.4667, -0.4745, -0.4745]]]])
tensor([[[[-0.3490, -0.3490, -0.3490, ..., -0.3176, -0.1688, -0.0588],
 [-0.3804, -0.3647, -0.3490, ..., -0.3961, -0.2941, -0.2235],
 [-0.3490, -0.3176, -0.2627, ..., -0.3961, -0.3020, -0.2471],
 ...,
 [-0.5059, -0.4980, -0.4824, ..., -0.4745, -0.4275, -0.3961],
 [-0.5059, -0.5059, -0.4902, ..., -0.4824, -0.4631, -0.4275],
 [-0.5137, -0.5137, -0.5137, ..., -0.4980, -0.4510, -0.4275]]]])]
```

Gambar 20. Hasil Masing-Masing Tensor *Resize* Normalisasi



Gambar 23. kondisi Palang Masuk dan Keluar Terbuka

5. Selanjutnya terdapat kondisi jika sensor infrared terdapat halangan atau tidak. Jika nilai sensor *LOW* maka lanjut ke tahap nomor 6, dan jika *HIGH* maka kembali ke tahap 4.
6. Pada Gambar 24 proses palang tertutup, pada proses ini arduino menggerakkan motor servo kembali seperti semula.



Gambar 24. Palang Masuk Tertutup dan Palang Keluar Tertutup jika Mobil Menghalangi Sensor Infrared

7. Selanjutnya proses kirim stats portal ke NodeMCU untuk meng-*update* nilai pada *firebase*.

Pengujian terakhir yaitu pengujian *taping* kartu *NFC* pada parkir keluar. Proses ini dilakukan untuk pencocokan nomor plat kendaraan yang dikenali sistem dengan nomor plat yang didaftarkan pada proses pendaftaran kartu *NFC* diawal.

3.2. Pengujian *Beta*

Pengujian model terhadap citra *synthetic* yang dibuat dari *python library* memiliki akurasi 85,73% dengan jumlah 1500 citra *synthetic*. Sedangkan untuk pengujian model terhadap citra plat nomor asli yang didapat dari *google image* dan *Kaggle.com* memiliki akurasi 77,5% dengan jumlah 40 citra plat nomor asli. Pengujian sistem secara *realtime* dengan 3 plat nomor uji beserta kartu *NFC* yang didaftarkan. Plat nomor D 1198 VUA dengan kartu *NFC* ktp, Z 5278 DAE dengan *SIM*, dan D 1568 UAF tidak memiliki kartu *NFC*. Berikut Gambar 25 plat nomor beserta kartu *NFC* yang digunakan dan hasil pengujian palang masuk dan palang keluar dalam bentuk Tabel 1 dan Tabel 2.



Gambar 25. Plat Nomor Uji Beserta Kartu NFC

Tabel 1. Pengujian Beta Palang Masuk

No Plat	Jml Percobaan	Hasil Prediksi	Status
D 1568 UAF	Percobaan ke - 1	U 1568 UAF	Salah
	Percobaan ke - 2	D 1568 UAF	Benar
Z 5278 DAE	Percobaan ke - 1	Z 5278 DAE	Benar
D 1198 VUA	Percobaan ke - 1	D 1198 VUA	Benar

Tabel 2. Pengujian Beta Palang Keluar

No Plat	Jml Percobaan	Hasil Prediksi	Status
D 1568 UAF	Percobaan ke - 1	O 1568 UAF	Salah
	Percobaan ke - 2	O 1568 VAF	Salah
	Percobaan ke - 3	D 1568 UAF	Benar
Z 5278 DAE	Percobaan ke - 1	L 5278 DAE	Salah
	Percobaan ke - 2	Z 5278 DAE	Benar
D 1198 VUA	Percobaan ke - 1	O 1198 VUA	Salah
	Percobaan ke - 2	U 1198 UUA	Salah
	Percobaan ke - 3	D 1198 VUA	Benar

Dari hasil pengujian sistem yang diusulkan dapat menjadi alternatif pembaharuan sistem parkir dalam hal pencatatan plat nomor kendaraan secara otomatis dalam sistem perpustakaan yang selama ini masih banyak mengandalkan pencatatan manual oleh tenaga parkir di pintu masuk dan keluar. Kendala sistem saat melakukan pengujian secara *real time* terdapat masalah pada proses deteksi plat, karena proses deteksi plat pada penelitian ini menggunakan proses pencarian kontur yang memiliki 4 sudut, bagian mobil terdapat kontur 4 sudut selain dari plat nomor, jika hal ini ditemukan deteksi plat nomor tidak berhasil dilakukan.

4. KESIMPULAN

Pada penelitian ini telah diimplementasikan metode CRAFT sebagai deteksi teks dan CRNN sebagai pengenalan teks pada plat nomor. Penggabungan kedua metode tersebut dilakukan untuk pembaharuan pada sistem parkir otomatis dan mampu melakukan pengenalan karakter pada plat nomor kendaraan. Proses *training model* dengan *dataset synthetic* telah menghasilkan model terbaik yang memiliki akurasi sebesar 85,73% terhadap 1500 citra uji *synthetic* dan akurasi sebesar 77,5 % terhadap 40 citra plat nomor asli yang didapat dari *google image* dan Kaggle.com.

DAFTAR RUJUKAN

- Al Khairi, M. H. (2021). *Mikrokontroler: Pengertian, Struktur dan Jenis-Jenisnya*. Diambil kembali dari Mahir Elektro: <https://www.mahirelektro.com/2020/10/pengertian-mikrokontroler-struktur-dan-jenisnya.html>
- Aprilino, A., & Al Amin, I. H. (2022). Implementasi Algoritma Yolo Dan Tesseract Ocr Pada Sistem Deteksi Plat Nomor Otomatis. *Jurnal Teknoinfo*, 16(1), 54-59.

- Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. (2019). Character Region Awareness for Text Detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Bgshih. (2018). *Convolutional Recurrent Neural Network*. Diambil kembali dari Github: <https://github.com/bgshih/crnn>
- Fcakyon. (2022). *CRAFT: Character-Region Awareness For Text detection*. Diambil kembali dari Github: <https://github.com/fcakyon/craft-text-detector>
- Harani, N. H., Prianto, C., & Hasanah, M. (2019). Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python. *Jurnal Teknik Informatika*, 47-53.
- Ihsan. (2023). *Pengertian Arduino UNO Mikrokontroler ATmega328*. Diambil kembali dari CaraTekno: <https://www.caratekno.com/pengertian-arduino-uno-mikrokontroler/>
- Iqbal, M. (2022). *Mikrokontroler ESP32*. Diambil kembali dari Telkom University: <https://miqbal.staff.telkomuniversity.ac.id/mikrokontroler-esp32/>
- Ismail, W. S., Purnawan, P. W., Riyanto, I., & Nazori. (2020). Sistem Perekaman Pelat Nomor Mobil pada Palang Pintu Parkir Menggunakan Web Kamera dan Mikrokontroler. *JURNAL MATRIX*, 10(3), 103-112.
- Lyalin, D. (2023). *Developer documentation for Firebase*. Diambil kembali dari Firebase: <https://firebase.google.com/>
- Munawar, I., Rusman, P., & Rizky, A. (2021). Automatic VLP'S Recognition For Smart Parking System. *Doctoral dissertation, Nusa Putra University*.
- Ramadhani, N. (2021). *Mengenal Lebih Jauh Sejarah Plat Nomor Kendaraan serta Daftar Lengkapnya*. Diambil kembali dari Akseleran: <https://www.akseleran.co.id/blog/plat-nomor/>
- Shi, B., Bai, X., & Yao, C. (2017). An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 2298-2304.
- Wicaksono, N. (2020). *Apa itu Firebase? Pengertian, Jenis-Jenis, dan Fungsi Kegunaannya*. Diambil kembali dari Dicoding Indonesia: <https://www.dicoding.com/blog/apa-itu-firebase-pengertian-jenis-jenis-dan-fungsi-kegunaannya/>
- Youngmin, Bado, Dongyoon, Sangdoon, & Hwalsuk. (2019). Character Region Awareness for Text Detection. *Clova AI Research NAVER Corp*, 1-12.