

IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI KATA PADA CITRA TEKS

Donni S. Silalahi¹, Mayanda Mega Santoni^{2*}, Anita Muliawati³

¹²³Program Studi Informatika, Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia
Email: donnissilalahi@gmail.com¹, megasantoni@upnvj.ac.id²

Abstrak

Optical Character Recognition (OCR) adalah proses untuk mencari dan mengenali karakter dalam sebuah gambar dan kemudian mengekstrak karakter tersebut sehingga menjadi karakter digital (*editable character*). Dalam penelitian ini, akan merancang sebuah sistem untuk mengenali kata bahasa Indonesia dalam sebuah gambar / citra. Penelitian ini akan melalui beberapa tahapan, dimulai dari tahap praproses citra, kemudian tahap segmentasi karakter pada citra dengan menggunakan algoritma *Connected Component Analysis (CCA)* lalu dilanjutkan ke tahap mengklasifikasikan karakter tersebut dengan menggunakan metode *Convolutional Neural Network (CNN)*. Setelah itu, akan dilakukan penggabungan karakter menjadi sebuah kata. Penelitian ini menggunakan data alfabet yang digunakan untuk melatih model CNN, dan data citra teks bahasa Indonesia untuk diklasifikasi oleh model. Hasil yang didapat yaitu akurasi klasifikasi alfabet sebesar 97%, kemudian untuk klasifikasi kata didapat akurasi sebesar 34,54%. Akurasi pada klasifikasi kata ini dipengaruhi oleh praproses citra yang dilakukan sebelumnya, hasil dari praproses tersebut banyak citra yang mengalami kegagalan sehingga salah saat diklasifikasi. Kegagalan praproses disini yaitu seperti citra yang blur atau rusak sehingga salah diklasifikasi dan juga bisa salah disegmentasi, kemudian terdapat citra teks yang mengandung huruf “i” dan “j” yang dianggap sebagai 2 karakter berbeda.

Kata Kunci: *Optical Character Recognition, Connected Component Analysis, Convolutional Neural Network*

1 PENDAHULUAN

Dalam perkembangan ilmu pengetahuan saat ini, dapat membuat akses semua orang menjadi serba mudah. Salah satunya dalam masalah ini ilmu pengetahuan yang dapat dimanfaatkan adalah pengolahan citra digital. Salah satunya dikenal dengan metode OCR, dimana metode ini bertujuan untuk mengenali gambar berisi teks menjadi teks yang dapat dikenali oleh mesin (Shreya dkk., 2019). Pada penelitian ini, metode ini digunakan untuk mengenali dan mengekstrak karakter dari sebuah gambar / citra teks bahasa Indonesia.

Penelitian terkait metode untuk identifikasi citra teks atau karakter banyak dilakukan menggunakan metode *Neural Network*. *Neural Network* memiliki cara kerja yang sama seperti sel otak manusia ketika memproses informasi, dimana *Neural Network* menyimpan dan menggunakan pengetahuan yang didapat dari pengalaman sehingga sangat cocok digunakan dalam masalah prediksi ataupun masalah klasifikasi yang sedang dibahas dalam penelitian ini. *Neural Network* memiliki beberapa metode yang banyak digunakan untuk klasifikasi karakter, yaitu *Multilayer Perceptrons (MLP)* dan *Convolutional Neural Network (CNN)*. CNN merupakan bagian dari *Deep Learning* dan juga merupakan pengembangan dari MLP dimana CNN dapat memproses citra 2 dimensi sedangkan MLP hanya 1 dimensi.

Penelitian mengenai MLP pernah dilakukan untuk mengenali karakter plat kendaraan menggunakan metode *Canny* untuk deteksi tepian dan algoritma *Backpropagation* arsitektur MLP untuk mengenali karakter (Haryoko dan Pramono, 2016). Dari penelitian tersebut mampu mengenali karakter dengan tingkat keberhasilan 94,29%. Penelitian mengenai CNN pernah dilakukan untuk mengenali karakter pada plat kendaraan Indonesia (Notonogoro dkk., 2018) dan mencapai akurasi hingga 95,45%. Penelitian lain juga pernah dilakukan untuk pengenalan karakter pada tulisan tangan Korea (Kim dan Xie, 2014) dari 2 *database* berbeda dan mendapat nilai akurasi tertinggi yaitu 99,71%.

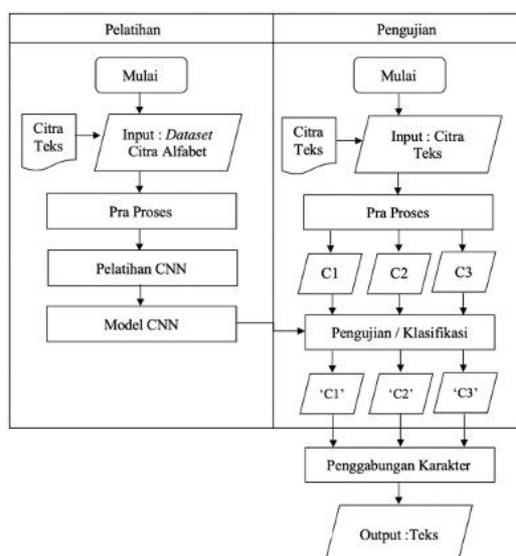
Dari kedua metode tersebut di atas, penulis menggunakan CNN untuk melakukan identifikasi atau klasifikasi kata pada citra teks. CNN memiliki tingkat akurasi yang lebih tinggi pada beberapa penelitian yang dilakukan untuk pengenalan karakter. CNN juga memiliki cara kerja yang hampir mirip dengan MLP dan CNN diyakini cenderung lebih efektif dan mudah dilatih, karena pada CNN dilakukan konvolusi pada input untuk mereduksi ukuran citra (matriks). Selain itu pada penelitian ini, penulis juga menggunakan metode *Connected Component Analysis* (CCA) untuk proses segmentasi teks menjadi karakter-karakter.

2 METODOLOGI PENELITIAN

2.1 Pengumpulan Data

Pada penelitian ini terdapat dua kelompok data yang berbeda. Kelompok pertama yaitu kelompok data citra yang terdiri dari 26 alfabet baik huruf kecil maupun kapital. Data ini digunakan untuk data latih dan data *test* pada proses pembuatan model CNN. Kelompok kedua yaitu citra dari 200 teks yang akan digunakan pada tahap pengujian. Kedua kelompok citra ini terdiri dari 5 jenis *font* yang berbeda yaitu Arial, Bodoni, Calibri, Helvetica dan Times New Roman. Kedua kelompok data ini nanti akan di cetak (*printed image*) terlebih dahulu lalu diambil citranya menggunakan beberapa *device*.

2.2 Tahapan Penelitian



Gambar 1. Perancangan Sistem

Seperti yang dapat dilihat pada **Gambar 1**, terdapat 2 tahapan yang akan dilakukan yaitu tahap pelatihan dan pengujian. Pada tahap pelatihan, proses ini menerima data *input* berasal dari *dataset* alfabet, data tersebut di praproses terlebih dahulu, kemudian masuk ke pelatihan CNN untuk melatih setiap karakter alfabet. Hasil akhir dari pelatihan karakter ini yaitu sebuah model yang digunakan

untuk mengenali karakter alfabet lain pada saat proses pengujian / klasifikasi. Pada proses pelatihan ini terdapat 3 proses yang dilakukan yaitu konvolusi dan *pooling* yang dikenal dengan ekstraksi fitur, dan proses *fully connected*.

Proses konvolusi melakukan ekstraksi fitur yang ada pada citra *input* dengan mengaplikasikan filter yang telah ditentukan pada citra *input*. Proses pengaplikasian terus bergeser sampai semua area pada citra *input* berhasil diaplikasikan dan setiap pengaplikasian tersebut dilakukan operasi perkalian matriks. Hasil akhir proses konvolusi ini berupa *output* yang disebut *feature map*, dan selanjutnya akan diterapkan fungsi aktivasi menggunakan ReLu (*Rectified Linear Unit*). *Pooling* akan mereduksi dimensi dari *feature map* dengan mengaplikasikan filter baru di semua area *feature map*. Proses ini bertujuan untuk mempercepat komputasi dan menghindari kemungkinan *overfitting*. Proses *Fully Connected Layer* merupakan proses klasifikasi, yang memperoleh *input* dari hasil proses ekstraksi fitur. *Fully Connected Layer* hanya memproses *input* satu dimensi, sehingga *feature map* dari ekstraksi fitur yang masih multidimensi akan dilakukan proses *flatten* menjadi satu dimensi. Proses ini biasa digunakan pada MLP karena *input* berupa satu dimensi dan akan diklasifikasikan secara linear. Hasil dari proses ini kemudian diberikan fungsi aktivasi *softmax* untuk mendapatkan hasil klasifikasi.

Pada tahap pengujian menerima *input* dari data citra teks kemudian dilakukan praproses pada citra tersebut sebelum dilakukan klasifikasi. Pertama, citra masukan dilakukan *preprocessing*, adapun proses yang dilakukan pada tahap ini adalah proses *thresholding*. *Thresholding* yaitu mengubah citra teks menjadi citra biner sehingga dapat terlihat jelas posisi dari objek dan *background*. Kedua, hasil citra biner dilakukan proses segmentasi karakter dengan menggunakan *Connected Component Analysis* untuk memberi label pada tiap objek pada citra. Kemudian dilakukan segmentasi objek sesuai label yang bertujuan untuk mendapatkan karakter-karakter dari citra teks (biner) yang kemudian masing-masing karakter tersebut digunakan untuk proses pengenalan karakter. Ketiga, setelah didapatkan karakter-karakter tersebut kemudian dijadikan *input* ke proses pengujian / klasifikasi. Proses yang dilakukan disini yaitu proses pengenalan karakter menggunakan model hasil dari proses pelatihan yang telah dilakukan CNN tadi. Pada proses pengujian tadi maka dihasilkan beberapa karakter hasil klasifikasi dan dilakukan penggabungan karakter-karakter sehingga menjadi sebuah kata.

2.3 Pengujian Sistem

Pada tahap ini akan dilakukan pengujian terhadap sistem setelah dimasukkan semua citra *input*. Pengujian dilakukan pada hasil klasifikasi kata pada citra teks yang dilakukan oleh CNN. Adapun formula yang digunakan untuk menghitung nilai akurasi pada CNN dapat dilihat pada persamaan berikut:

$$\text{Akurasi Klasifikasi} = \frac{\text{Jumlah Kata Terklasifikasi Benar}}{\text{Total Citra Teks}} \times 100\% \quad (1)$$

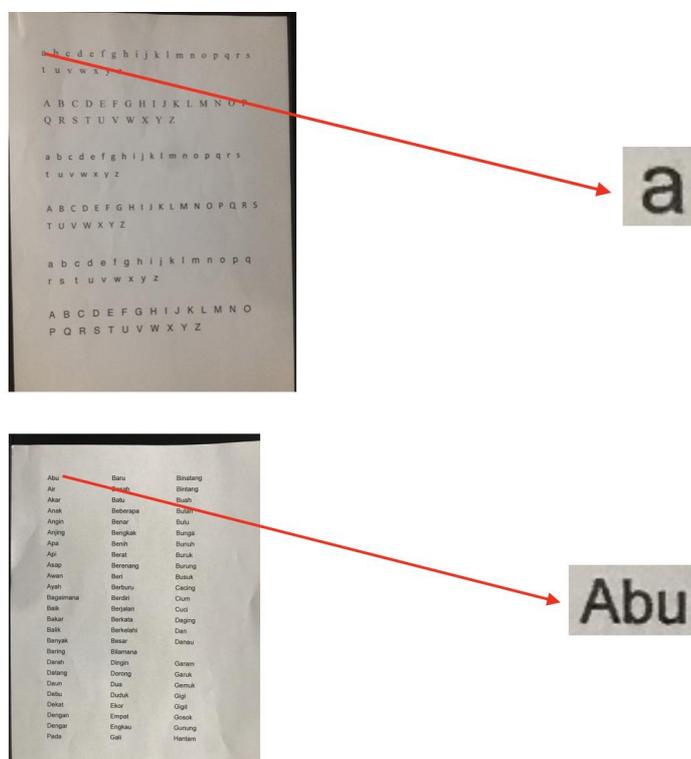
3 HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Terdapat dua kelompok data pada penelitian ini. Kelompok pertama yaitu citra yang terdiri dari 26 alfabet baik huruf kecil dan kapital. Kelompok kedua terdiri dari citra dari 200 kata yang akan digunakan pada tahap pengujian sistem. Kedua kelompok tersebut berasal dari 5 jenis. Namun, pada proses pengambilan citra peneliti menggunakan 5 *device* berbeda yang berguna untuk mengukur kemampuan sistem pada kondisi citra yang bervariasi. *Device* yang digunakan terdiri dari *device Scanner*, Samsung Galaxy J5, iPhone 4, iPhone 5s dan iPhone 7. Adapun spesifikasi kamera pada masing-masing *smartphone* tersebut yaitu, 13 MP f/1.9 di Samsung Galaxy J5, 5 MP f/2.8 di iPhone 4, 8 MP f/2.2 di iPhone 5s dan 12 MP f/1.8 di iPhone 7. Kelompok ketiga yaitu 200 daftar

kata swadesh yang berasal dari Laboratorium Kebinekaan Bahasa dan Sastra, Kemdikbud dan telah diurutkan berdasarkan abjad dari A ke Z.

Proses pengambilan citra dimulai dari mencetak (*print*) kelompok data ini dengan ukuran kertas A4. Data alfabet diketik dengan menggunakan ukuran *font* 20, sedangkan data citra teks diketik dengan ukuran *font* 16. Dalam satu kertas A4 dapat berisi data alfabet sebanyak 3 *font* dan dapat berisi kurang lebih 70 kata. Setelah dicetak, citra diambil dengan melakukan *scan* dengan alat *scanner* dan memfoto menggunakan 4 *smartphone* lainnya pada semua data. Aturan yang dilakukan pada saat pengambilan foto yaitu jarak dari *device smartphone* dengan data (kertas) 15 cm. Setelah dilakukan pengambilan citra oleh semua *device*, proses berlanjut ke tahap *cropping* menggunakan aplikasi Paint, awalnya dengan mengubah size (*resize*) citra hasil *device* terlebih dahulu ke ukuran 35 kemudian mulai meng-*crop* citra alfabet menjadi per huruf dan citra teks menjadi per kata. Aturan yang diterapkan pada proses *cropping* ini yaitu citra alfabet di-*crop* dengan ukuran 32x32 piksel per huruf. Proses *cropping* pada citra teks, akan di-*crop* dengan ukuran lebar 32 piksel dan panjang menyesuaikan dengan panjang dari kata dan masih memiliki beberapa ruang di sisi kanan dan kiri. Hasil *cropping* pada citra alfabet dan teks dapat dilihat pada **Gambar 2**.



Gambar 2. Proses Cropping Citra Alfabet dan Teks

Dalam penamaan citra hasil *cropping* juga dilakukan aturan penamaan yaitu untuk *font* Arial, Bodoni, Calibri, Helvetica dan Times New Roman masing-masing diberi kode F1, F2, F3, F4 dan F5 secara berurutan. Penamaan pada *device Scanner*, Samsung Galaxy J5, iPhone 4, iPhone 5s dan iPhone 7 masing-masing diberi kode D1, D2, D3, D4 dan D5 secara berurutan. Urutan penamaan masing-masing citra ini mengikuti format 'kode alfabet/kode kata_kode font_kode device'. Pada citra alfabet kecil dari a – z diberi kode A1 – A26. Penamaan untuk huruf kapital dari A – Z akan diberikan penambahan tanda kutip setelah angka untuk menjadi pembeda dari alfabet kecil. Maka kode yang akan diberikan yaitu A1' – A26' secara berurutan. Penamaan pada 200 citra teks diberi kode T1 – T200 secara berurutan sesuai urutan penamaan berdasarkan. Hasil akhir dari proses ini menghasilkan data citra alfabet sebanyak 1.300 citra dan data citra teks sebanyak 5.000 citra.

3.2 Praproses Data Citra

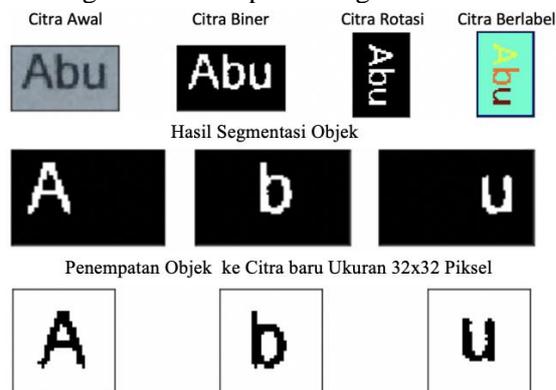
Pada data citra alfabet praproses yang dilakukan ialah *thresholding* semua citra alfabet kemudian diubah ke dalam format .csv. Awalnya setiap citra alfabet dilakukan *thresholding* kemudian objek pada citra diubah posisinya agar berada di tengah citra. Proses pengubahan ke dalam format .csv ialah mengubah citra menjadi array 1 dimensi atau dengan kata lain dari ukuran 32x32 piksel menjadi 1x1024 piksel. Data format .csv ini berisi 1.300 baris yang mana merupakan jumlah citra itu sendiri dan berisi 1.025 kolom, dimana 1.024 kolom berisi nilai piksel setiap citra sedangkan 1 kolom pertama yaitu kolom label yang berisi nilai kelas setiap citra. Aturan pemberian nilai kelas pada setiap citra alfabet yaitu citra dengan huruf a – z masing-masing diberi nilai kelas 0 – 25 secara berurutan, seperti dapat dilihat pada **Gambar 3**. Data citra alfabet dalam format .csv ini yang menjadi data pembelajaran pada tahap pelatihan.

label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	...	pixel1019	pixel1020	pixel1021	pixel1022	pixel1023	pixel1024
0	255	255	255	255	255	255	255	255	255	255	255	255
0	255	255	255	255	255	255	255	255	255	255	255	255
0	255	255	255	255	255	255	255	255	255	255	255	255
0	255	255	255	255	255	255	255	255	255	255	255	255

Gambar 3. Data Citra Alfabet Dalam Format .csv

Pada data citra teks dilakukan praproses sebelum citra masuk sebagai *input* ke model CNN untuk diklasifikasi. Pertama praproses yang dilakukan pada citra teks ialah *thresholding* citra, kemudian citra biner hasil *thresholding* ini dirotasi terlebih dahulu sebelum akhirnya diterapkan proses *Connected Component Analysis* (CCA) untuk memberi label setiap objek dalam citra. Adapun rotasi tadi dilakukan untuk mencegah kesalahan pengurutan label objek oleh CCA. Selanjutnya, akan dilakukan segmentasi setiap objek yaitu memisahkan objek satu per satu berdasarkan labelnya dan masing-masing objek akan ditempatkan pada suatu citra baru tersendiri.

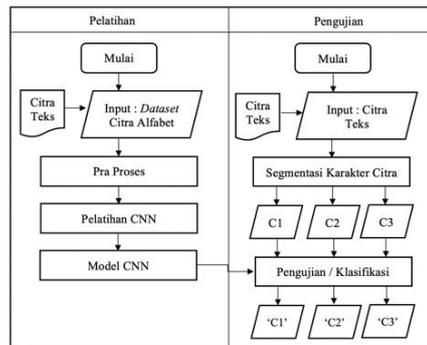
Akan tetapi, citra baru tersebut masih memiliki ukuran yang sama dengan citra awalnya, sedangkan untuk menjadi *input* ke dalam model CNN harus memiliki ukuran yang sama dengan yang dipelajari oleh model yaitu 32x32 piksel. Oleh karena itu, akan dilakukan praproses lagi dimana objek dalam citra akan ditempatkan dalam sebuah citra baru dengan ukuran 32x32 piksel dan posisinya berada ditengah citra baru tersebut. Hasil akhir yang didapatkan dari praproses citra teks ini adalah kumpulan citra biner hasil segmentasi yang berisi setiap objek dari citra awal dengan ukuran baru 32x32 piksel serta memiliki *background* warna putih dan objek warna hitam. Seluruh praproses yang dilakukan pada data citra teks tadi dapat dilihat pada **Gambar 4**. Pada praproses citra ini, dari 5.000 citra, didapatkan total 2.016 citra yang memenuhi kriteria keberhasilan atau disegmentasi bagus dan layak masuk ke dalam proses klasifikasi. Semua citra teks hasil praproses ini selanjutnya akan disebut sebagai citra hasil proses segmentasi.



Gambar 4. Praproses Citra Teks

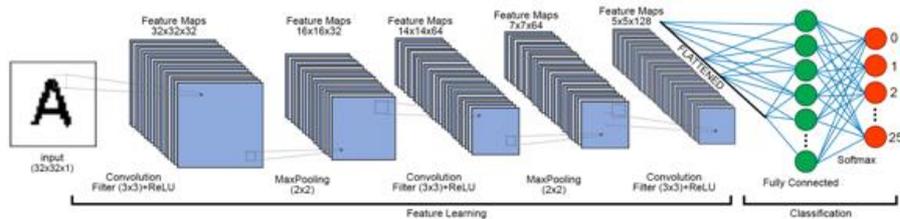
3.3 Perancangan CNN

Proses pelatihan dan proses pengujian dapat dilihat pada **Gambar 5**.



Gambar 5. Proses Pelatihan dan Pengujian pada CNN

Dalam proses pelatihan akan menghasilkan model, dimulai dengan menyusun arsitektur modelnya (CNN) terlebih dahulu. Seperti yang sudah dijelaskan sebelumnya, arsitektur model terdiri dari proses convolution untuk pembelajaran ciri dan proses fully connected untuk pengklasifikasian. Dalam model ini, peneliti membuat 3 proses convolution yang mana setiap prosesnya memiliki nilai parameter yang berbeda dan masing-masing diikuti dengan proses *pooling*. Setelah proses konvolusi lalu masuk ketahap klasifikasi dengan melakukan proses *flatten* dan *fully connected* Arsitektur CNN yang telah disusun ini dapat dilihat pada **Gambar 6**.



Gambar 6. Arsitektur CNN

Selanjutnya, arsitektur model yang telah dibuat akan di-*compile* dengan argumen yang berisi *loss function* menggunakan *categorical_crossentropy*, *optimizer* menggunakan adam dan *metrics* menggunakan *accuracy*. Setelah itu, didapatkan ringkasan model yang telah dibuat seperti pada **Gambar 7**, dengan jumlah parameter yang dilatih sebanyak 505.754.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 32, 32)	320
max_pooling2d_1 (MaxPooling2)	(None, 16, 16, 32)	0
dropout_1 (Dropout)	(None, 16, 16, 32)	0
conv2d_2 (Conv2D)	(None, 14, 14, 64)	18496
max_pooling2d_2 (MaxPooling2)	(None, 7, 7, 64)	0
dropout_2 (Dropout)	(None, 7, 7, 64)	0
conv2d_3 (Conv2D)	(None, 5, 5, 128)	73856
dropout_3 (Dropout)	(None, 5, 5, 128)	0
flatten_1 (Flatten)	(None, 3200)	0
dense_1 (Dense)	(None, 128)	409728
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 26)	3354
Total params: 505,754		
Trainable params: 505,754		
Non-trainable params: 0		

Gambar 7. Model Summary

3.4 Pembagian Data untuk Pelatihan Model CNN

Proses pelatihan model dimulai dengan menerima dan membaca data citra alfabet yang telah diubah ke dalam format .csv sebanyak 1.300 data atau *rows*. Data ini dibagi terlebih dahulu menjadi 2 yaitu data *train* dan data *test*. Data *train* akan digunakan untuk pelatihan atau pembelajaran oleh CNN saat pembentukan model. Data *test* digunakan sebagai bahan evaluasi terhadap performa model yang sudah terbentuk. Performa model model tersebut dapat digambarkan dalam bentuk *Confusion Matrix* dan *Classification Report*. Dalam penelitian ini, dari 1.300 data akan dibagi dengan perbandingan 80:20, dimana 80% menjadi data *train* dan 20% menjadi data *test*. Data test terdiri dari semua data citra alfabet yang diambil menggunakan *device 5* yang mana berjumlah 260 citra, sehingga dapat mencapai 20% untuk data *test*. Sisanya dari *device 1 – device 4* yaitu 1.040 citra atau 80% akan menjadi data *train*. Sebelum digunakan, kedua data ini akan dijadikan *array* dengan ukuran pikselnya yaitu 32x32 piksel dan dengan jumlah *channel 1*. Pada saat pelatihan, data *train* akan dibagi lagi menggunakan metode *K-Fold Cross Validation*.

3.5 Pemilihan Hyperparameters Model CNN

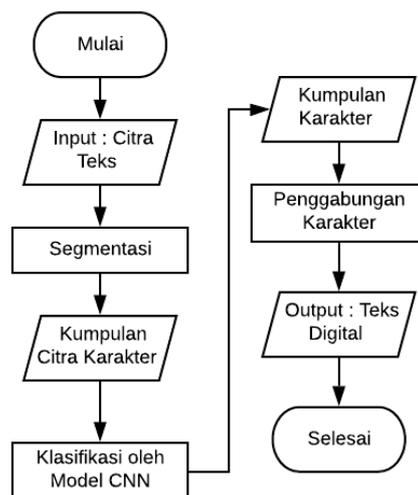
Pada saat pembentukan model, terdapat paling tidak 3 *hyperparameters* yang dapat diubah untuk menghasilkan model yang baik. Seperti yang telah disebutkan sebelumnya *hyperparameters* tersebut yaitu *batch size*, *epoch* dan *learning rate*, dan tidak ada nilai pasti setiap *hyperparameters* tersebut untuk menghasilkan sebuah model yang baik. Oleh karena itu, peneliti melakukan beberapa percobaan untuk mencari kombinasi nilai yang cocok demi menghasilkan model yang baik. Salah satu ukuran sebuah model yang baik yaitu memiliki nilai *validation loss* yang rendah dan memperhatikan *accuracy* dan *validation accuracy* yang tinggi dan tidak memiliki selisih terlalu jauh. Dari beberapa percobaan yang telah dilakukan, terpilih kombinasi nilai *hyperparameter* terbaik yaitu nilai *epoch* sebanyak 20, nilai *batch size* sebesar 50 dan nilai *learning rate* sebesar 0,001 dengan akurasi sebesar 0,9700 (97%) dan *validation loss* sebesar 0,0062. Kemudian setelah dilakukan juga evaluasi performa model yang telah dibentuk menggunakan percobaan terhadap data *test*. Hasil yang didapat yaitu *test loss* sebesar 0.0282 dan *test accuracy* sebesar 0.9884 (98%).

3.6 Implementasi Metode Penelitian

Pada tahap ini akan mengimplementasikan sistem berdasarkan perancangan yang telah dilakukan sebelumnya. Dalam pengimplementasiannya, penulis menggunakan bahasa pemrograman python dan menggunakan *software Jupyter Notebook* untuk menjalankan kode program. Dalam pembuatan model *Convolutional Neural Network*, penulis menggunakan *packages Keras* dan *Tensorflow* sebagai *back-end*-nya. Penulis juga menggunakan *library* pendukung seperti *numpy*, *opencv*, *PIL*, *pandas* dan beberapa *library* lainnya.

3.7 Pengujian Metode Penelitian

Pada tahap ini proses yang dilakukan ialah menguji model yang telah dibentuk dengan menggunakan data citra teks sebagai data ujinya. Proses pengujian dapat dilihat pada flowchart **Gambar 8**.



Gambar 8. Flowchart Pengujian Sistem

3.8 Akurasi

Akurasi model CNN yang didapat yaitu sebesar sebesar 97%, dan *test accuracy* sebesar 98%. *Classification report* dari model CNN ini dapat dilihat pada **Gambar 9**.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	10
2	1.00	1.00	1.00	10
3	1.00	1.00	1.00	10
4	1.00	1.00	1.00	10
5	1.00	1.00	1.00	10
6	1.00	1.00	1.00	10
7	1.00	1.00	1.00	10
8	1.00	0.70	0.82	10
9	1.00	1.00	1.00	10
10	1.00	1.00	1.00	10
11	0.77	1.00	0.87	10
12	1.00	1.00	1.00	10
13	1.00	1.00	1.00	10
14	1.00	1.00	1.00	10
15	1.00	1.00	1.00	10
16	1.00	1.00	1.00	10
17	1.00	1.00	1.00	10
18	1.00	1.00	1.00	10
19	1.00	1.00	1.00	10
20	1.00	1.00	1.00	10
21	1.00	1.00	1.00	10
22	1.00	1.00	1.00	10
23	1.00	1.00	1.00	10
24	1.00	1.00	1.00	10
25	1.00	1.00	1.00	10
accuracy			0.99	260
macro avg	0.99	0.99	0.99	260
weighted avg	0.99	0.99	0.99	260

Gambar 9. Classification Report Model CNN

Perhitungan akurasi keberhasilan pada proses klasifikasi kata dari masing-masing *device* dan *font* dapat dilihat pada **Tabel 1**. Dalam tabel akurasi klasifikasi ini, dapat dilihat bahwa akurasi tertinggi diperoleh dari data citra teks dengan jenis *font* 1 (Arial) dan *font* 4 (Helvetica) menggunakan *device* 5 (iPhone 7), yaitu sebesar 58,0%. Selain itu, akurasi terendah diperoleh dari data citra teks dengan jenis *font* 2 (Bodoni) dengan menggunakan *device* 3 (iPhone 4), yaitu sebesar 0,5%.

Tabel 1. Akurasi Keberhasilan Klasifikasi Kata Berdasarkan Device dan Font

	D1	D2	D3	D4	D5
F1	57,0%	54,5%	22,0%	56,0%	58,0%
F2	3,5%	1,0%	0,5%	1,5%	6,0%
F3	43,5%	55,5%	6,5%	42,5%	57,5%
F4	57,0%	57,5%	35,0%	57,5%	58,0%
F5	33,5%	39,5%	3,0%	26,5%	30,5%

Kesalahan klasifikasi kata dipengaruhi oleh beberapa kasus, seperti kegagalan dalam proses segmentasi dan juga model yang dalam beberapa kondisi tidak mampu untuk mengklasifikasi karakter dengan benar. Adapun kasus kesalahan yang terjadi dapat dilihat pada **Tabel 2**.

Tabel 2. Kesalahan Klasifikasi Kata Dalam Beberapa Kasus

Hasil Proses Segmentasi	Hasil Klasifikasi	Kasus
	aiir	Kata mengandung huruf "i"
	ahv	Citra teks rusak dan model tidak mampu mengklasifikasi dengan benar
	ijmia	Citra rusak dan segmentasi salah mengakibatkan klasifikasi salah

4 KESIMPULAN

Dari hasil penelitian dan pengimplementasian sistem yang dilakukan untuk mengklasifikasi kata pada citra teks menggunakan *Convolutional Neural Network* ini, dapat ditarik kesimpulan sebagai berikut :

1. Metode *Convolutional Neural Network* (CNN) dapat digunakan untuk mengenali teks pada suatu citra menggunakan model yang telah dibentuk.
2. Model CNN yang terbentuk memiliki *accuracy* sebesar 97%, dan *test accuracy* sebesar 98%.
3. Kombinasi nilai *hyperparameter* terbaik yang digunakan pada model yaitu *epoch* sebanyak 20, nilai *batch size* sebesar 50 dan nilai *learning rate* sebesar 0,001.
4. Model digunakan pada proses pengujian untuk melakukan klasifikasi dan mampu mengklasifikasi 1.727 citra dengan benar.

Referensi

- Deng, li & Abdel-Hamid, Ossama & Yu, Dong. (2013). "A Deep Convolutional Neural Network Using Heterogeneous Pooling For Trading Acoustic Invariance With Phonetic Confusion". *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88.*, 1988 International Conference on. 6669-6673. [10.1109/ICASSP.2013.6638952](https://doi.org/10.1109/ICASSP.2013.6638952).
- Haryoko, A., & Pramono, S. H. (2016). "Pengenalan Karakter Plat Kendaraan Bermotor Berbasis Citra Dengan Menggunakan Metode Canny Dan Algoritma Backpropagation". *Jurnal Ilmiah NERO*, 2(2), 123–130.
- Kim, I. J., & Xie, X. (2014). "Handwritten Hangul Recognition Using Deep Convolutional Neural Networks". *International Journal on Document Analysis and Recognition*, 18(1), 1–13. <https://doi.org/10.1007/s10032-014-0229-4>.
- Nasuha, A., Sardjono, T. A., & Purnomo, M. H. (2018). "Pengenalan Viseme Dinamis Bahasa Indonesia Menggunakan Convolutional Neural Network". *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 7(3), 258–265. <https://doi.org/10.22146/jnteti.v7i3.433>.
- Notonogoro, I. W., Jondri, & Arifianto, A. (2018). "Indonesian License Plate Recognition Using Convolutional Neural Network". *2018 6th International Conference on Information and Communication Technology, ICoICT 2018*, 0(c), 366–369. <https://doi.org/10.1109/ICoICT.2018.8528761>.
- Rizki, A., Nugroho, A. S., Jamal, A., Handoko, D., Gunawan, M., Witjaksono, A., & Yogantara, W. W. (2010). "Connected Component Analysis Sebagai Metode Pencarian Karakter Plat Dalam Sistem Pengenalan Plat Nomor Kendaraan". *Technology*, (Sitia).
- S. Shreya, Y. Upadhyay, M. Manchanda, R. Vohra, and G. D. Singh. (2019). "Optical Character Recognition Using Convolutional Neural Network". *Proc. 2019 6th Int. Conf. Comput. Sustain. Glob. Dev. INDIACom 2019*, pp. 55–59.
- Sehgal, I. (2019). "Connected Component Labeling for Binary Images". *International Journal of Advanced Research*, 7(8), 916–927. <https://doi.org/10.21474/ijar01/9580>.