

Penggunaan Convolutional Recurrent Neural Network dan RLSA untuk Mengambil Data pada Akta Kelahiran

Stephanie Liem, Kartika Gunadi, Alvin Nathaniel Tjondrowiguno
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658

E-Mail: stephie_liem@yahoo.co.id, kgunadi@petra.ac.id, alvin.tjondrowiguno@gmail.com

ABSTRAK

Akta kelahiran merupakan salah satu dokumen yang wajib untuk dimiliki oleh seorang warga negara. Dokumen ini berisikan informasi seputar kelahiran seseorang dan merupakan pencatatan resmi bukti pengakuan negara akan keberadaan orang tersebut. Akta kelahiran banyak digunakan sebagai dasar penetapan identitas dalam dokumen lainnya dimana salah satunya adalah ijazah. Universitas Kristen Petra sebagai salah satu lembaga pendidikan di Indonesia membutuhkan data akta kelahiran mahasiswanya sebagai bukti nyata merupakan warga negara yang sah dan sebagai dasar dalam penerbitan ijazah. Selama ini pengambilan informasi dilakukan secara manual namun dengan berkembangnya teknologi, kini memungkinkan untuk mengambil informasi-informasi yang terdapat pada akta kelahiran secara otomatis. Penelitian akan pengambilan informasi terhadap akta kelahiran belum pernah dilakukan sebelumnya tetapi penelitian serupa telah dilakukan terhadap Kartu Tanda Penduduk menggunakan metode Template Matching dengan akurasi 17-39%, .

Penelitian ini menggunakan metode *Run Length Smoothing Algorithm* dan *Convolutional Recurrent Neural Network*. *Run Length Smoothing Algorithm* digunakan untuk mensegmentasi setiap kata yang ada pada akta kelahiran menjadi sebuah gambar tersendiri. Kata yang terdapat pada gambar akan kemudian diubah menjadi teks dalam bentuk *string* menggunakan *Convolutional Recurrent Neural Network*. Penentuan kata mana yang adalah informasi yang diinginkan menggunakan bantuan urutan kata dan kata kunci-kata kunci tertentu.

Hasil akhir yang akan ditampilkan adalah informasi berupa nama lengkap, tanggal lahir, tempat dan jenis kelamin dari pemilik akta dalam bentuk *string*. Berdasarkan pengujian yang dilakukan, didapatkan nilai akurasi sebesar 12.936% untuk pengambilan informasi dan 60.068% untuk prediksi kata oleh CRNN.

Kata Kunci: *Neural Network, Convolutional Recurrent Neural Network, Run Length Smoothing Algorithm, Akta Kelahiran, Image Segmentation*

ABSTRACT

Birth certificate is one of the documents that is mandatory for every citizen to have. This document records the information upon someone's birth and an official acknowledgement of a country on that person's existence. Birth certificate is a legal document and is an acceptable form of identification for other documents such as a diploma. As one of Indonesia's learning institute, Petra Christian University needs its students birth certificate as a solid proof upon their identification and as a base to publish a diploma. The extraction of information is being done manually but with the rapid development of technology, it is now possible to obtain the information within a birth certificate automatically. Research about information

extraction on Birth Certificate hasn't been done yet before, but similar research with the object of Identity Card has been done using Template Matching with the accuracy of 17-39%.

This research uses Run Length Smoothing Algorithm and Convolutional Recurrent Neural Network as its primary methods. Run Length Smoothing Algorithm is used to segment words in a birth certificate image. The word in an image will then be translated into a text in string form by Convolutional Recurrent Neural Network. To know which words that contain the wanted information, the sequence of the words and specific keywords are being used.

The result of this research will be information upon the full name, birth date, place of birth and the gender of the birth certificate holder. The result from tests that were done is an accuracy of 12.936% upon finding the wanted information and 60.068% for words translation from image to string by CRNN.

Keywords: *Neural Network, Convolutional Recurrent Neural Network, Run Length Smoothing Algorithm, Birth Certificate, Image Segmentation*

1. PENDAHULUAN

Akta Kelahiran merupakan salah satu dokumen yang harus dimiliki oleh semua warga Indonesia, dimana dokumen ini merupakan pencatatan resmi mengenai keberadaan seseorang dan hak kewarganegaraannya. Universitas Kristen Petra merupakan salah satu perguruan tinggi swasta nasional yang ada di Indonesia. Setiap mahasiswa yang ada harus dicatat data-data pribadinya sesuai dengan dokumen-dokumen resmi miliknya, dimana salah satu dokumen tersebut adalah Akta Kelahiran. Akta Kelahiran ini digunakan sebagai bukti nyata mahasiswa merupakan warga Indonesia yang diakui dan sebagai dasar salah satu syarat proses kelulusan yaitu penerbitan ijazah. Dibutuhkan sebuah sistem yang mampu secara otomatis mengambil data yang ada pada akta kelahiran tersebut dimana data yang diperlukan adalah nama lengkap, tanggal lahir, tempat lahir dan jenis kelamin dari pemilik akta kelahiran.

Proses otomatisasi pengambilan informasi pada Akta Kelahiran dapat dilakukan dengan pertama menentukan daerah mana pada Akta Kelahiran yang merupakan teks dan memisahkan per kata yang ada. Proses segmentasi per kata ini menggunakan metode *Run Length Smoothing Algorithm* [2]. Proses ini digunakan karena kemudahan dalam konfigurasi *threshold* dan kecepatannya yang cenderung lebih cepat [6].

Selanjutnya setelah berhasil dipisahkan kata-kata tersebut harus diproses agar dapat dikenali, dimana prosesnya disebut *Optical Character Recognition* (OCR). Salah satu cara untuk melakukan proses OCR adalah dengan memanfaatkan metode tradisional seperti *template matching* ataupun dengan *neural network* seperti *Convolutional Neural Network* atau *Recurrent Neural Network* untuk mengartikan kata pada citra menjadi bentuk teks.

Dalam penelitian [5] serupa mengenai pengambilan informasi dari sebuah dokumen resmi yaitu KTP, digunakan metode *template matching* untuk mengenali setiap karakter yang ada. Penelitian ini menunjukkan hasil yang kurang memuaskan karena dari total 320 karakter hanya dapat dikenali sekitar 17-39% saja. Metode *template matching* ini juga dinilai tidak cocok untuk digunakan pada dokumen Akta Kelahiran karena adanya perbedaan *font*, tidak seperti KTP yang memiliki satu *font* yang sama.

Penelitian ini menggunakan *Convolutional Recurrent Neural Network* [7] untuk mengenali teks dan diubah menjadi bentuk string.

2. DASAR TEORI

2.1 Akta Kelahiran

Akta kelahiran adalah salah satu dokumen yang wajib dimiliki oleh semua orang yang berperan sebagai bukti sah mengenai status dan peristiwa kelahirannya. Dokumen ini diterbitkan oleh Kantor Catatan Sipil (Dinas Kependudukan dan Pencatatan Sipil) masing-masing daerah [9]. Akta kelahiran akan sangat berpengaruh terhadap masa depan seseorang, tanpa akta kelahiran seseorang tidak dapat sekolah, menikah dan membuat berbagai dokumen kependudukan lainnya seperti paspor.

Informasi yang terdapat pada Akta Kelahiran dapat dibedakan menjadi 5 bagian yang meliputi:

1. Nomor Registrasi Akta Kelahiran
Nomor Registrasi Akta Kelahiran ini digunakan untuk membedakan satu Akta Kelahiran dengan Akta Kelahiran lainnya. Format pengisian nomor ini berbeda antara Akta Kelahiran model lama dan model baru.
2. Nomor Stbld
Stbld adalah kepanjangan dari Staatsblad dari bahasa Belanda yang berarti perundang-undangan yang dikeluarkan oleh Kolonial Belanda pada zamannya. Stbld ini masih digunakan hingga sekarang dalam penggolongan Akta Kelahiran masyarakat Indonesia. Terdapat beberapa Stbld yang sering dipakai yaitu Stbld 1917 No. 130 jo. 1919 No. 81 untuk WNA dan WNI keturunan Cina, Stbld 1920 No. 751 jo. 1927 No. 564 untuk WNI Pribumi non Nasrani, dan Stbld 1933 No. 75 jo. 1936 No. 607 untuk WNI Pribumi Nasrani.
3. Tempat Kelahiran
Pada bagian ini disebutkan tempat kota kelahiran pemilik akta kelahiran.
4. Tanggal Kelahiran
Bagian ini berisikan tanggal, bulan, tahun dan jam kelahiran pemilik akta kelahiran.
5. Informasi pemilik akta
Pada bagian ini dituliskan beberapa informasi mengenai pemilik akta kelahiran seperti jenis kelamin, anak keberapa, nama lengkap, nama ayah dan nama ibu.

2.2 Segmentasi

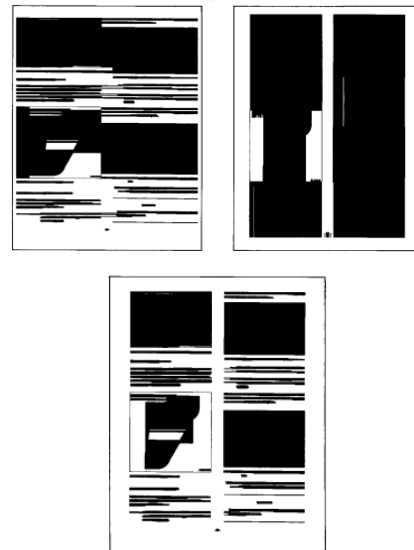
Segmentasi digunakan untuk membagi sesuatu ke dalam segmen-segmen atau bagian-bagian lain. Pada objek citra, segmentasi digunakan untuk membagi citra menjadi bagian-bagian tertentu. Tujuan dari dilakukan segmentasi adalah agar dapat menemukan daerah (*Region of Interest*) yang diinginkan pada citra, dimana daerah tersebut dapat memudahkan dan meningkatkan proses analisa data lebih lanjutnya. Pada umumnya proses segmentasi ini dilakukan pemberian label terhadap pixel-pixel yang ada pada citra agar pixel dengan label yang sama memiliki karakteristik tertentu yang sama pula.

2.2.1 Run Length Smoothing Algorithm

Run Length Smoothing Algorithm (RLSA) adalah sebuah metode yang digunakan untuk *block segmentation* dan *text discrimination*. RLSA akan digunakan untuk melakukan segmentasi kata-kata pada citra agar kemudian dapat dilanjutkan pada proses CRNN. RLSA melakukan perhitungan dengan merubah pixel hitam (0) dan putih (1) menurut dua aturan yaitu:

1. Semua pixel 1 pada gambar asli akan diganti dengan 0 jika pixel 1 terdekat lainnya kurang dari atau sama dengan limit C pada hasil
2. Semua pixel 0 pada gambar asli tetap akan menjadi pixel 0 pada hasil

Perhitungan algoritma ini dilakukan dua kali yaitu secara vertikal dan horizontal. Hasil dari keduanya akan diberlakukan operasi logika AND untuk mendapat hasil akhir [10]. Penentuan parameter *threshold* (C) akan sangat berperan penting dalam penentuan hasil segmentasi. Contoh hasil RLSA dapat dilihat pada Gambar 1.



Gambar 1. Hasil RLSA Horizontal, Vertikal dan Setelah Operasi AND [10]

2.3 Neural Network

Neural network adalah sistem yang memproses informasi yang memiliki karakteristik menyerupai jaringan saraf biologis [3]. Sistem ini akan belajar dengan sendirinya untuk menghasilkan *output* paling optimal dengan menggunakan jaringan-jaringan yang terhubung seperti yang ada pada jaringan saraf manusia.

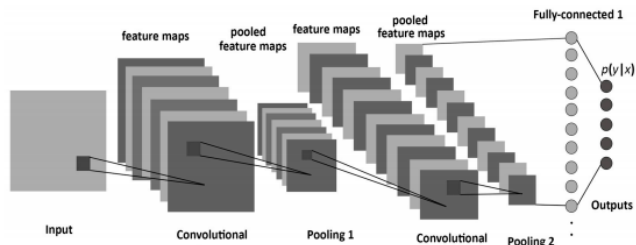
2.3.1 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah *neural network* yang dikhususkan untuk memproses data yang memiliki topologi menyerupai *grid* [4]. CNN pada dasarnya merupakan *neural network* yang menggunakan metode konvolusi untuk menghitung mutlipikasi matriks-matriks yang diolah dalam masing-masing *layer* yang ada pada arsitektur *neural network*.

CNN memiliki tiga *layer* utama yaitu *Convolutional Layer*, *Pooling Layer* dan *Fully-Connected Layer* [1]. Proses konvolusi menggunakan beberapa filter berupa matriks dalam ukuran tertentu ($n \times n$) yang digeserkan ke seluruh bagian *image* (input). Pada setiap bagian yang ada di matriks dilakukan operasi *dot product*, penambahan bias dan *hyperbolic tangent*. Hasil dari konvolusi ini kemudian menghasilkan sebuah *feature map*. *Feature map* ini akan

kemudian digunakan pada *layer* berikutnya. Kumpulan dari setiap *feature maps* ini akan menjadi hasil akhir dari *Convolutional Layer*.

Pada *Pooling Layer*, hasil *feature maps* dari *Convolutional Layer* akan kemudian dikecilkan dengan mencari nilai maksimumnya. Hal ini dilakukan untuk mengurangi jumlah parameter dan komputasi dari jaringan. Sedangkan pada *Fully-Connected Layer* setiap *layer* saling dihubungkan agar menjadi *multi-layer perceptron*. Arsitektur CNN dapat dilihat pada Gambar 2.

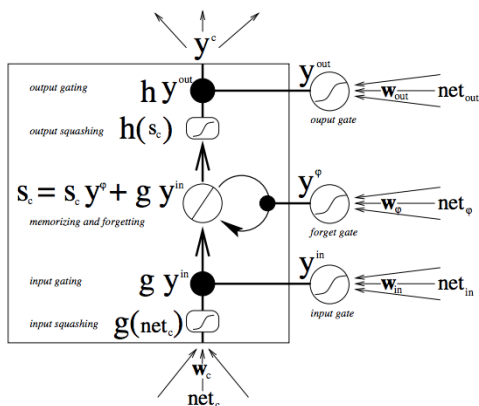


Gambar 2. Arsitektur CNN Error! Reference source not found.

2.3.2 Recurrent Neural Network

Recurrent Neural Network (RNN) sebuah *neural network* yang memiliki koneksi *feedback* (*closed loop*) pada *node*-nya [3]. Berbeda dengan *neural network* pada umumnya yang memperlakukan input dan outputnya sebagai sesuatu yang independen antara satu dengan yang lainnya, RNN memperhitungkan hubungan antar input dan outputnya.

Pada RNN *output* dari sebuah proses bergantung dari perhitungan proses sebelumnya. Hasil komputasi pada proses-proses sebelumnya akan dijadikan faktor dalam perhitungan komputasi proses saat ini. Agar dapat mengetahui hasil komputasi dan keputusan yang diambil pada *node*-node atau proses-proses sebelumnya dibutuhkan adanya *memory*. Salah satu tipe RNN yang mampu menggunakan kembali hasil komputasi proses-proses sebelumnya adalah *Long Short Term Memory* (LSTM) [8]. LSTM memiliki tiga *gate* yang dapat mengatasi permasalahan ini yaitu *input gate*, *output gate* dan *forget gate*. *Gate*-*gate* ini digunakan oleh LSTM untuk menentukan informasi mana saja yang ingin disimpan. Arsitektur LSTM dapat dilihat pada Gambar 3.



Gambar 3. Arsitektur LSTM [8]

2.3.3 Connectionist Temporal Classification

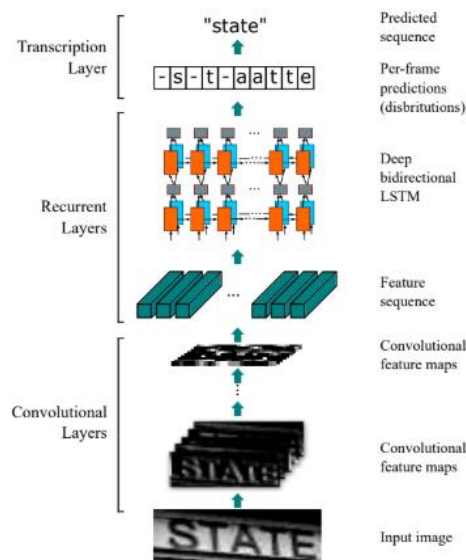
Connectionist Temporal Classification (CTC) adalah sebuah metode yang digunakan untuk mengklasifikasi hasil akhir agar berupa *string* dari sebuah matriks. CTC digunakan sebagai tahap terakhir dalam rangkaian *neural network* yang bertujuan untuk mengenali karakter. Hasil dari *neural network* sebelumnya (CNN/RNN) pada umumnya akan berbentuk matriks, matriks

inilah yang oleh CTC akan diolah agar didapatkan teks *string* sebagai hasil akhir dari rangkaian proses yang ada.

2.3.4 Convolutional Recurrent Neural Network

Convolutional Recurrent Neural Network (CRNN) adalah model *neural network* yang menggabungkan CNN, RNN dan CTC untuk memprediksi suatu kata yang terdapat pada citra yang diinputkan. CRNN menyatukan keunggulan CNN dan RNN sehingga proses klasifikasi tidak terbatas pada per huruf saja melainkan per kata. Dengan menggunakan kata sebagai inputan maka tidak dibutuhkan proses segmentasi lebih untuk memisahkan huruf-huruf pada citra input.

CRNN terdiri dari tiga komponen yaitu *Convolutional Layer* (ekstraksi *feature sequence*), *Recurrent Layer* (prediksi label *distribution*) dan *Transcription Layer* (translasi prediksi per-*frame* menjadi hasil akhir) [7]. Pada tahap ekstraksi *feature sequence*, digunakan *convolution* dan *pooling layer* dari CNN tradisional. Tahap ini menghasilkan *feature vector* yang saling berurutan (*feature sequence*) yang akan diteruskan kepada tahap label *distribution*. Setelah itu dilakukan prediksi untuk setiap *frame* yang ada pada *feature sequence* menggunakan *bidirectional LSTM*. Hasil prediksi ini akan diubah menjadi sebuah label *sequence* oleh CTC pada *transcription layer*. Label *sequence* inilah yang akan menjadi hasil akhir dari keseluruhan proses CRNN. Arsitektur CRNN dapat dilihat pada Gambar 4.



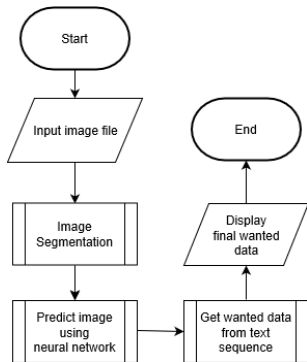
Gambar 4. Arsitektur CRNN [7]

3. DESAIN SISTEM

Program pertama-tama akan menerima *input* dari *user* berupa *image* Akta Kelahiran dalam format .JPG atau .JPEG. *File* gambar tersebut kemudian akan diolah hingga dapat terdeteksi dan tersegmentasi kata-kata yang ada di dalamnya. Segmentasi per kata ini dilakukan menggunakan metode *Run Length Smoothing Algorithm*.

Setelah didapatkan seluruh gambar yang merepresentasikan sebuah kata hasil segmentasi sebelumnya, gambar-gambar tersebut akan dimasukkan ke dalam proses selanjutnya yaitu prediksi kata yang ada pada gambar. Kata pada setiap gambar akan diolah dan diproses menggunakan CRNN (*Convolutional Recurrent Neural Network*) yang telah di-*train* sebelumnya agar dapat menghasilkan kata dalam bentuk tipe data *string*.

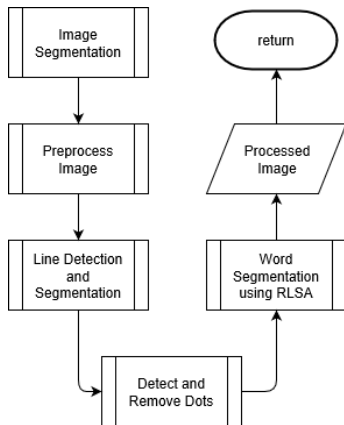
Kemudian, akan dicari informasi-informasi yang diinginkan berdasarkan urutan *string* hasil prediksi CRNN tersebut. Hal ini digunakan dengan cara memperhatikan kata kunci apa saja yang mengikuti setiap informasi yang diinginkan. *Flowchart* proses gambaran besar sistem dapat dilihat pada Gambar 5.



Gambar 5. Gambaran Besar Sistem

3.1 Image Segmentation

Sub-proses *Image Segmentation* bertujuan untuk mengolah citra Akta Kelahiran yang di-*input*-kan oleh *user* agar dapat menjadi potongan-potongan kata. Langkah pertama pada proses ini adalah *preprocessing*, dimana citra diolah terlebih dahulu agar didapatkan *Region of Interest* yaitu daerah tengah akta kelahiran yang mengandung informasi yang diinginkan. Setelah itu akan dilakukan proses *Line Detection and Segmentation* untuk mensegmentasi hasil dari *Region of Interest* menjadi per baris. Kemudian akan dilanjutkan dengan pencarian dan penghapusan titik-titik agar tidak mengganggu proses selanjutnya yaitu segmentasi per kata menggunakan metode RLSA (*Run Length Smoothing Algorithm*). *Flowchart* alur *Image Segmentation* dapat dilihat pada Gambar 6.

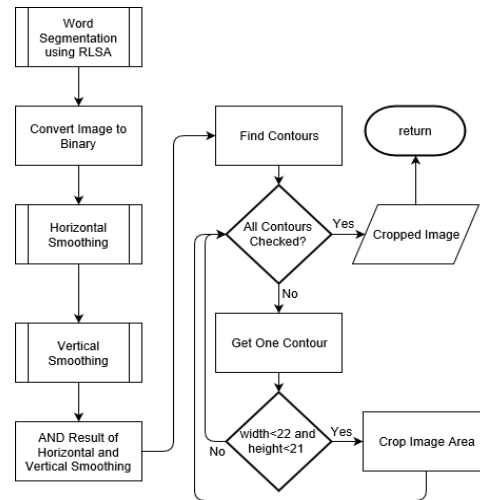


Gambar 6. Proses *Image Segmentation*

3.1.1 Run Length Smoothing Algorithm

Untuk dapat memecah setiap baris menjadi per kata digunakan metode RLSA. Pertama-tama, gambar diubah ke dalam bentuk *binary*, setelah itu dilakukan proses RLSA. Pada proses RLSA sendiri terbagi menjadi dua proses yaitu *smoothing* secara horizontal dan *smoothing* secara vertikal. Hasil dari kedua proses ini akan digabungkan dengan metode *AND*. Dari hasil penggabungan ini kemudian akan dicari bagian mana saja yang merupakan daerah tulisan (kontur) dan bukan *background*. Pada setiap kontur yang ada dilakukan pengecekan jika memiliki lebar lebih kecil dari 22 dan tinggi lebih kecil dari 21 pixel akan di-*crop* pada daerah tersebut. Hasil akhir berupa gambar-gambar hasil *crop*

yang masing-masing berisikan sebuah kata yang ada pada *input image*. *Flowchart* proses RLSA dapat dilihat pada Gambar 7.



Gambar 7. Proses *Run Length Smoothing Algorithm*

3.2 Neural Network

Pada penelitian ini digunakan penggabungan dari *neural network* CNN dan LSTM. Rincian layer-layer yang digunakan dapat dilihat pada Tabel 1. Digunakan *Convolutional Layer* dan *Pooling Layer* dari CNN, namun *Fully-connected Layer* yang pada umumnya ada pada CNN digantikan dengan LSTM. Hal ini dilakukan agar *neural network* dapat mengenali citra per kata dan tidak per huruf saja. Pada *layer-layer* milik CNN digunakan ReLU sebagai *activation function*.

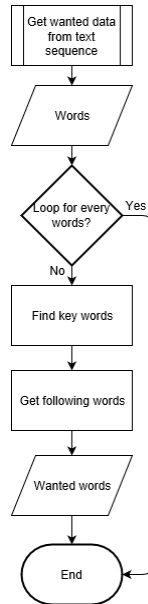
Setelah selesai melalui proses CNN dan LSTM, maka akan dilanjutkan dengan pengolahan menggunakan CTC. Hasil akhir dari proses CTC ini akan menghasilkan sebuah kata hasil prediksi gambar *input* dalam bentuk *string*. Untuk *Convolutional Layer*, *Pooling Layer*, LSTM dan CTC digunakan bantuan fungsi dari *library* Tensorflow.

Tabel 1. Rincian *Layer* pada CNN-LSTM

Layer	Operation	Kernel Size
1	<i>Convolutional</i>	3
2	<i>Convolutional</i>	3
	<i>Pooling</i>	2
3	<i>Convolutional</i>	3
4	<i>Convolutional</i>	3
	<i>Pooling</i>	2
5	<i>Convolutional</i>	3
6	<i>Convolutional</i>	3
	<i>Pooling</i>	2
7	<i>Convolutional</i>	3
8	<i>Convolutional</i>	3
	<i>Pooling</i>	3
9	LSTM	
10	LSTM	

3.3 Get Data from Text Sequence

Pada proses ini akan dicari data mana saja yang diinginkan dari sekumpulan string hasil prediksi *neural network*, dimana pada penelitian ini yang diinginkan adalah nama lengkap, tanggal lahir, tempat lahir dan gender dari pemilik akta kelahiran. Karena *input* yang dimasukkan ke dalam *neural network* berurutan dari kata pertama pada baris pertama dan seterusnya, maka dapat dikatakan bahwa kata yang dihasilkan oleh *neural network* memiliki urutan yang sama seperti pada citra akta kelahiran yang di-*input* oleh *user* pertama kali. Proses ini akan mencari kata-kata *keywords* yang biasanya muncul sebelum kata yang adalah informasi yang diinginkan. *Keyword* untuk informasi yang diinginkan berbeda-beda dan dapat memiliki jumlah yang tidak sama akibat banyaknya kemungkinan yang terjadi. *Flowchart* proses *Get Data from Text Sequence* dapat dilihat pada Gambar 8.



Gambar 8. Proses *Get Data from Text Sequence*

4. IMPLEMENTASI SISTEM

Implementasi sistem dilakukan pada computer dengan spesifikasi:

- RAM: 8GB, DDR4
- Memory: 120 GB SDD + 1T HDD
- CPU: Intel Core i7
- GPU: NVIDIA GeForce GTX 1060
- OS Windows 10 Pro

Implementasi pengkodean sistem, menggunakan Bahasa pemrograman Python dengan versi 3.6.6. Adapun beberapa *library* yang mendukung sistem ini adalah:

- PIL
- OpenCV
- Tensorflow
- Flask
- Scipy
- Numpy

Aplikasi ini menggunakan tampilan pada web ini dibuat dengan menggunakan arsitektur *Service Oriented*. Data-data yang di-*input* dan dihasilkan diakses melalui berbagai *HTTP Method*, antara lain:

- GET (Menerima data olahan dalam bentuk *array* ataupun *object*)

- POST (Memasukkan data dari web ke program untuk diolah)

5. ANALISA DAN PENGUJIAN

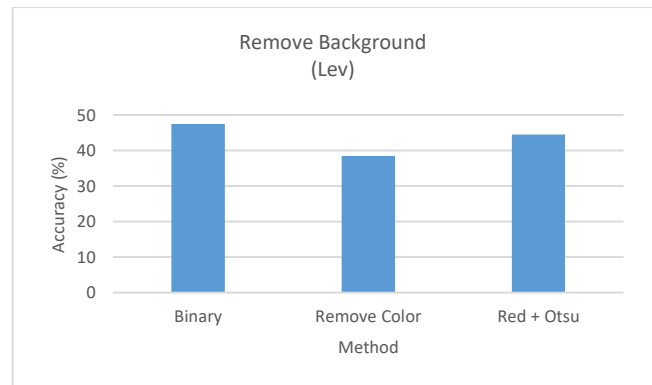
5.1 Image Segmentation

Pengujian image segmentation dikhususkan pada sub-proses *Preprocessing* dan *Word Segmentation* using RLSA.

5.1.1 Preprocessing

Subproses utama yang diuji adalah subproses *remove background*. Pengujian dilakukan terhadap tiga metode yaitu *binary threshold*, *remove color with HSV channel*, dan *otsu thresholding with red channel*.

Pada metode *binary threshold*, citra diubah menjadi *grayscale* dan jika nilai pada pixel lebih besar dari *threshold* maka akan berwarna hitam dan jika tidak akan berwarna putih. *Threshold* yang digunakan adalah 70 hingga 255 dan 127 hingga 255. Metode *remove color* dilakukan dengan mengubah citra menjadi *channel HSV* dan kemudian mengubah pixel yang memiliki nilai HSV diantara *range (0,0,153) hingga (100,255,255)* menjadi warna putih. Metode terakhir adalah metode *red channel + otsu*, dimana pada awalnya *channel blue* dan *green* pada citra diubah menjadi 0 sehingga hanya didapatkan *red channel*. Setelah itu *red channel* ini akan ubah menjadi *grayscale* dan diberlakukan *otsu binarization*.



Gambar 9. Grafik Akurasi Pengujian *Remove Background*

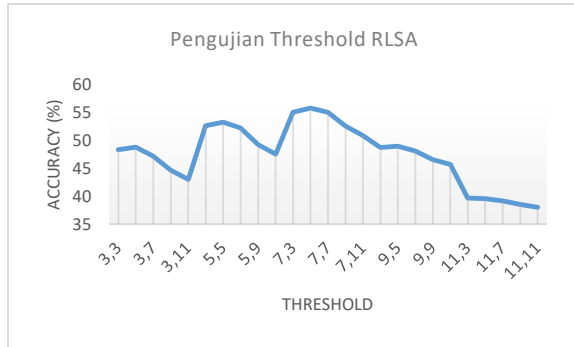
Pengujian dilakukan dengan menghitung akurasi kata yang dihasilkan oleh *neural network* berdasarkan inputan gambar hasil segmentasi ketika metode tersebut. Akurasi dihitung dengan menggunakan metode *Levenshtein Distance*. Berdasarkan pengamatan akan hasil akurasi dari pengujian tersebut didapati bahwa hasil akurasi terbaik didapatkan dari penggunaan metode *binary threshold* dengan tingkat akurasi 47.528%. Rincian hasil pengujian dapat dilihat pada Gambar 9.

5.1.2 Threshold Run Length Smoothing Algorithm

Percobaan dilakukan dengan mengganti vertikal dan horizontal *threshold* pada RLSA agar ditemukan hasil dengan akurasi yang paling optimal. *Threshold* akan berpengaruh pada jarak maksimum pixel hitam satu dengan yang lainnya agar pixel putih diantaranya dapat diubah menjadi hitam juga (digolongkan sebagai satu kesatuan). Hasil proses RLSA ini akan menghasilkan segmentasi-segmentasi kata. Hasil segmentasi ini akan kemudian dimasukkan pada *neural network*. Hasil prediksi kata akan kemudian dicocokkan dengan kata yang benar dan dihitung akurasinya dengan menggunakan metode *Levenshtein Distance*.

Percobaan dilakukan pada 85 akta kelahiran yang sudah tersegmentasi per baris dengan sempurna menggunakan metode

background removal binary threshold dengan pembersihan noise yang tersisa secara manual. Kombinasi nilai *threshold* yang digunakan adalah 3 hingga 11 dengan interval 2 pada masing-masing horizontal dan vertikal *threshold*.



Gambar 10. Chart Hasil Pengujian *Threshold RLSA*

Berdasarkan pengamatan terhadap hasil pengujian yang dapat dilihat pada Gambar 10, diperoleh bahwa nilai maksimum akurasi OCR adalah 55.817% yang dihasilkan dengan menggunakan konfigurasi kombinasi *threshold* 7 untuk horizontal *threshold* dan 5 untuk vertikal *threshold*.

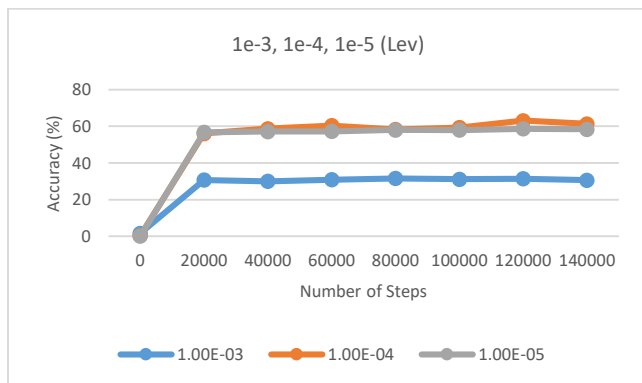
5.2 Pengujian *Neural Network*

Pengujian dilakukan pada *learning rate* serta *number of steps* (epoch) dari *neural network*.

5.2.1 *Learning Rate* dan *Number of Steps* (epoch)

Pengujian dilakukan terhadap pengaruh konfigurasi *learning rate* dan *number of steps* terhadap akurasi prediksi dari *neural network*. Pengujian dilakukan terhadap 85 akta kelahiran *training* dan 10 akta kelahiran *testing* yang sudah tersegmentasi kata per katanya dengan sempurna, dimana menggunakan metode *background removal binary threshold* dengan pembersihan *noise manual*. Nilai *learning rate* yang diuji meliputi 1e-3, 1e-4 dan 1e-5. Sedangkan nilai *number of steps* (epoch) yang diuji mulai 0 hingga 140000 dengan interval 20000.

Perhitungan akurasi dilakukan dengan dua metode yaitu *Levenshtein Distance* dan *Jaro-Winkler*. Setiap kata yang diprediksi oleh *neural network* akan dibandingkan dengan kata yang seharusnya, setiap nilai akurasi akan ditambah dan dibagi dengan jumlah total kata yang ada pada akta tersebut. Total akurasi pada seluruh akta yang ada akan kemudian ditambahkan dan dirata-rata untuk mendapat hasil akurasi akhir untuk setiap parameter percobaan.



Gambar 11. Grafik Perbandingan Hasil Akurasi *Learning Rate*

Berdasarkan pengujian akan tiga parameter *learning rate* dan delapan *number of steps* yang telah dilakukan, didapati bahwa kombinasi yang menghasilkan nilai akurasi terbaik adalah kombinasi *learning rate* 1e-4 dan *step* 120000 dengan nilai 63.077%. Perbandingan hasil akurasi dapat dilihat secara lengkap pada Gambar 11.

5.2.2 Dataset Tanpa Citra dengan Nama

Pengujian terhadap data *testing* dengan tidak melibatkan kata-kata yang adalah nama. Tujuan dari pengujian ini adalah untuk mencari tahu apakah dengan menghilangkan kata-kata yang berupa nama pada data *testing* yang diuji akan berdampak pada tingkat akurasi prediksi *neural network*. Rincian hasil pengujian dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian Data *Testing* Tanpa Nama

Number of Steps	With	Without	Diff
0	0.348	0.992	0.644
20000	54.030	61.387	7.357
40000	56.836	64.885	8.049
60000	58.312	66.208	7.896
80000	55.900	63.596	7.696
100000	57.035	64.088	7.053
120000	60.068	68.116	8.048
140000	57.180	65.146	7.966
		AVG	6.839

5.3 Proses Keseluruhan

Pengujian dilakukan dari proses awal hingga akhir dengan menggunakan konfigurasi terbaik pada masing-masing subprosesnya. Pengujian dilakukan terhadap 85 akta *training* dan 10 akta *testing*. Konfigurasi yang digunakan untuk metode *remove background* adalah *binary threshold*, *threshold RLSA* yang digunakan adalah 7 (*horizontal*) dan 5 (*vertikal*), serta model *neural network* yang digunakan adalah model hasil *learning rate* 1e-4 pada *step* 120000.

Hasil akhir dari pengujian ini adalah kata yang merupakan informasi yang diinginkan. Informasi yang dicari meliputi nama, tanggal lahir, tempat lahir dan jenis kelamin. Akurasi hasil yang berupa kata dihitung dengan menggunakan metode *Levenshtein Distance*.

Untuk pengujian yang dilakukan pada 10 akta baru didapati tingkat akurasi sebesar 12.936%. Pengujian selanjutnya yang dilakukan pada 85 akta yang sudah pernah menjadi dataset dalam proses *training neural network*, didapati akurasi berupa 29.854%.

5.4 Analisis

Berdasarkan hasil akurasi pengujian pengambilan informasi pada akta kelahiran yang cenderung rendah dilakukan analisa akan penyebab-penyebab rendahnya akurasi tersebut. Beberapa penyebab yang ada antara lain ada pada kondisi dataset, jarak kata dan *neural network*.

Salah satu penyebab dari rendahnya akurasi adalah kondisi dataset yang digunakan. Jenis-jenis kondisi dataset yang gagal dalam proses penelitian ini adalah citra dengan *brightness* tidak merata dan cenderung tinggi, citra dengan *brightness* rendah, citra tidak mengandung seluruh area akta kelahiran, citra dengan akta kelahiran yang lecek dan citra yang mengandung *noise* berupa bayangan dan *blur*.

Penyebab lainnya adalah jarak antar baris dan kata yang berbeda-beda pada akta kelahiran. Jarak yang berbeda-beda ini mengakibatkan tidak mungkin digunakan sebuah *threshold* dan nilai *variable* tertentu yang dapat sesuai untuk merepresentasikan jarak yang ada. Oleh karena itu, maka dengan *threshold* yang digunakan untuk metode *line segmentation* dan *word segmentation* (RLSA) tidak dapat menghasilkan segmentasi yang sempurna untuk setiap kondisi baris dan kata. Pada kasus tertentu didapati bahwa ada dua baris yang dianggap menjadi sebaris dan dua kata yang dianggap menjadi satu kata.

Recurrent Neural Network bekerja dengan cara mengingat urutan dari data yang diinputkan. Pada prosesnya, RNN akan memperhitungkan hasil komputasi sebelumnya untuk menentukan *output* saat ini. Pada sistem *neural network* yang digunakan pada penelitian ini, dimana input berupa sebuah kata yang terdiri dari beberapa huruf, RNN akan memperhitungkan prediksi huruf-huruf yang telah terprediksi sebelumnya untuk menentukan prediksi huruf terkini dalam sebuah inputan kata. Pada penelitian ini, objek yang digunakan adalah akta kelahiran dimana salah satu informasi yang berada di dalamnya adalah nama. Nama merupakan suatu identitas unik yang membedakan seorang dengan yang lainnya, hal ini berakibat pada banyaknya variasi nama yang ada. Oleh karena keterbatasan data yang dimiliki, maka tidak memungkinkan untuk mendapatkan seluruh variasi nama yang ada. Hal ini akan mempengaruhi kemampuan *neural network* untuk memprediksi kata yang adalah nama. Karena tidak memungkinkan untuk *neural network* mempelajari seluruh variasi nama yang ada, maka tentu akan berdampak pada akurasi prediksi *neural network*.

6. KESIMPULAN

Setelah dilakukan perancangan sistem, pengimplementasian, dan pengujian terhadap aplikasi yang telah dibuat, dapat ditarik kesimpulan sebagai berikut:

- Metode *Run Length Smoothing Algorithm* dapat digunakan untuk menentukan area kata-kata pada sebuah citra Akta Kelahiran dan metode *Convolutional Recurrent Neural Network* dapat digunakan mendapatkan informasi pada citra Akta Kelahiran dalam bentuk teks digital, dimana diperoleh hasil akurasi keseluruhan sebesar 12.936% (Levenshtein Distance)
- Berdasarkan pengujian pada bab 5, disimpulkan bahwa konfigurasi *threshold* dengan akurasi terbaik adalah 7 untuk horizontal *threshold* dan 5 untuk vertical *threshold* serta konfigurasi *neural network* dengan akurasi terbaik adalah dengan *learning rate* $1e-4$ dan *step* 120000
- Rendahnya akurasi disebabkan oleh kurangnya jumlah dan variasi pada *dataset training neural network*, adanya variasi pada jarak baris dan kata pada akta kelahiran serta kualitas *dataset* yang tidak optimal
- Pengujian terhadap RLSA dan *neural network* dimana citra yang diinputkan adalah baris yang tersegmentasi sempurna adalah 55.817% (Levenshtein Distance)
- Pengujian terhadap *neural network* murni, dimana citra yang diinputkan adalah kata-kata yang berhasil tersegmentasi secara sempurna menghasilkan akurasi sebesar 60.068% (Levenshtein Distance)

- Berdasarkan pengujian pada bab 5, ditemukan bahwa tingkat *brightness* yang menghasilkan tingkat akurasi tertinggi adalah ketika dikalikan dengan faktor 1
- Berdasarkan pengujian *preprocessing* untuk *remove background* pada subbab 5.2.1 disimpulkan bahwa metode terbaik adalah dengan menggunakan *binary threshold*

Saran untuk pengembangan kedepannya adalah:

- Menambahkan jumlah dan variasi dataset pada *neural network* agar prediksi *neural network* dapat menjadi lebih baik
- Penggunaan metode CNN tanpa RNN untuk memprediksi informasi-informasi yang diinginkan yaitu nama, tanggal lahir, tempat lahir dan jenis kelamin
- Pengembangan proses menghilangkan *background* untuk akta kelahiran agar dapat dihasilkan hasil yang lebih baik
- Pengembangan proses segmentasi baris dan kata yang lebih adaptif agar dapat mengikuti kondisi masing-masing akta

7. DAFTAR PUSTAKA

- [1] Albelwi, S., & Mahmood, A. 2017. A Framework for Designing the Architectures of Deep Convolutional Neural Network. *Entropy*, 5.
- [2] Bishop, C. 1998. *Neural Networks and Pattern Recognition*. London: Academic Press.
- [3] Fausett, L. 1994. *Fundamentals of Neural Networks*. New Jersey: Prentice-Hall, Inc.
- [4] Goodfellow, I., Bengio, Y., & Courville, A. 2016. *Deep Learning*. Cambridge: MIT Press.
- [5] Ryan, M., & Hanafiah, N. 2015. An Examination of Character Recognition on ID Card using Template Matching Approach. *Procedia Computer Science*, 520-529.
- [6] Shafait, F., Keysers, D., & Breuel, T. M. 2006. Performance Comparison of Six Algorithms for Page Segmentation. Dalam H. Banke, & A. L. Spitz, *Document Analysis Systems VII* (hal. 368-379). New Zealand: Springer International Publishing.
- [7] Shi, B., Bai, X., & Yao, C. 2016. 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*, 2298-2304.
- [8] SkyMind. 2018, Mei 31. *A Beginner's Guide to Recurrent Neural Network and LSTMs*. Retrieved from <https://deeplearning4j.org/lstm#recurrent>
- [9] Sukabumi, D. K. 2018, September 18. *Akta Kelahiran - Dukcapil Kabupaten Sukabumi*. Retrieved from <https://www.dukcapilkabsukabumi.org/pelayanan/akta-kelahiran/>
- [10] Wong, K. Y., Casey, R. G., & Wahl, F. M. 1982. Document Analysis System. *IBM Journal of Research and Development*, 26(6), 647-656.