

KLASIFIKASI TEKS HADIS BUKHARI TERJEMAHAN INDONESIA MENGUNAKAN RECURRENT CONVOLUTIONAL NEURAL NETWORK (CRNN)

Muhammad Yuslan Abu Bakar^{*1}, Adiwijaya²

^{1,2} Universitas Telkom Bandung

¹ abubakar.yuslan@gmail.com, ² adiwijaya@telkomuniversity.ac.id

^{*}Penulis Korespondensi

(Naskah masuk: 15 Juli 2020, diterima untuk diterbitkan: 18 Oktober 2021)

Abstrak

Hadis merupakan sumber hukum dan pedoman kedua bagi umat Islam setelah Al-Qur'an dan banyak sekali hadis yang telah diriwayatkan oleh para ahli hadis selama ini. Penelitian ini membangun sebuah sistem yang dapat melakukan klasifikasi teks hadis Bukhari terjemahan berbahasa Indonesia. Topik ini diangkat untuk memenuhi kebutuhan umat Islam dalam mengetahui apa saja informasi mengenai anjuran dan larangan yang terdapat dalam suatu hadis. Klasifikasi teks memiliki tantangannya tersendiri terkait dengan jumlah fitur yang sangat banyak (dimensi sangat besar) sehingga waktu komputasi menjadi besar dan mengakibatkan sulitnya mendapatkan hasil yang optimal. Pada penelitian ini, digunakan salah satu metode hibrid dalam dunia *deep learning* dengan menggabungkan Convolutional Neural Network dan Recurrent Neural Network, yaitu Convolutional Recurrent Neural Network (CRNN). Convolutional Neural Network dipilih sebagai metode seleksi dan reduksi data dikarenakan dapat menangkap informasi spasial yang saling berhubungan dan berkorelasi. Sementara Recurrent Neural Network digunakan sebagai metode klasifikasi dengan mengungkap kemampuan utamanya yaitu dapat menangkap informasi kontekstual yang sangat panjang khususnya pada data sekuens seperti data teks dengan mengandalkan 'memori' yang dimilikinya. Hasil penelitian menyajikan beberapa hasil klasifikasi menggunakan *deep learning*, dimana hasil akurasi terbaik diberikan oleh Convolutional Recurrent Neural Network (CRNN), yakni sebesar 80.79%.

Kata kunci: *convolutional neural network, recurrent neural network, klasifikasi teks, reduksi dimensi*

TEXT CLASSIFICATION OF HADITH OF BUKHARI (INDONESIAN LANGUAGE TRANSLATION) USING RECURRENT CONVOLUTIONAL NEURAL NETWORK (CRNN)

Abstract

Hadith is a source of law and guidance for Muslims after the Qur'an and many hadith have been narrated by hadith experts so far. This research builds a system that can classify Bukhari hadith in Indonesian translations. This topic was raised to meet the needs of Muslims in knowing what information about the suggestions and prohibitions that exist in a hadith. Text classification has its own challenges related to several features whose dimensions are very large so that it increases computing time and causes difficulties in getting optimal results. This research uses a hybrid method in deep learning by combining a Convolutional Neural Network and a Recurrent Neural Network, namely Convolutional Recurrent Neural Network (CRNN). Convolutional Neural Network was chosen as a method of selecting and reducing data that can be determined as spatial information that is interrelated and correlated. While Recurrent Neural Networks are used as a classification method by carrying out capabilities that can be used as very long contextual information specifically on sequential data such as text data by relying on the 'memory' it has. This research presents several classification results using deep learning, where the best accuracy results are given by the Convolutional Recurrent Neural Network (CRNN), which is equal to 80.79%.

Keywords: *convolutional neural network, recurrent neural network, text classification, dimension reduction*

1. PENDAHULUAN

Islam merupakan salah satu agama terbesar di dunia, khususnya di Indonesia yang mencakup

sebanyak 12.7% dari total populasi penganut Agama Islam diseluruh dunia (*Muslim Population by Country 2020*, n.d.). Bagi penganut Agama Islam,

terdapat dua sumber utama dalam ajarannya yaitu Al-Qur'an dan Hadis. Hadis merupakan panduan bagi umat Muslim yang bersumber dari perbuatan dan perkataan Nabi Muhammad SAW. dan menjadi pedoman bagi umat Muslim untuk lebih memahami Islam agar dapat menjalankan kehidupannya di dunia.

Telah banyak hadis yang diriwayatkan oleh para ahli hadis ke dalam beberapa kitab. Salah satu ahli hadis yang telah meriwayatkan banyak hadis adalah Al-Bukhari. Saat ini, ada sekitar 7000 hadis sahih yang telah diriwayatkan oleh Bukhari dan telah diterjemahkan ke dalam Bahasa Indonesia. Jumlah hadis yang tergolong ke dalam teks yang cukup besar membuat tingkat kesulitan dalam proses klasifikasi menjadi tinggi. Hal ini dikarenakan terdapat banyak sekali fitur/kata yang diproses sehingga dimensi dari data menjadi sangat besar dan membuat proses klasifikasi menjadi lama dan menurunkan hasil klasifikasi. Metode *deep learning* digunakan dalam penelitian ini untuk mengatasi masalah yang telah dipaparkan sebelumnya.

1.1. Penelitian Terkait

Hadis terbagi ke dalam beberapa kategori diantaranya hadis berdasarkan tingkat keabsahannya seperti sahih, hasan, maudo', dan dhaif seperti yang telah dikerjakan oleh (Aldhlan et al., 2013). Dalam penelitiannya, mereka menyebutkan bahwa beberapa atribut pada data yang digunakan diindikasikan sebagai nilai yang hilang. Oleh sebab itu, mereka menggunakan suatu mekanisme yang disebut dengan Missing Data Detector (MDD) untuk mengatasi data yang hilang tersebut. MDD yang digabungkan dengan *classifier* Decision Tree dapat meningkatkan hasil akurasi dengan sangat baik dari 50.1502% menjadi 97.597% dan menyimpulkan bahwa Decision Tree merupakan pendekatan yang layak untuk klasifikasi hadis dilihat dari kemudahannya dalam aturan induksi.

Al-Kabi et al. mengklasifikasikan hadis ke dalam delapan kategori berdasarkan klasifikasi Al-Bukhari menggunakan metode TF-IDF (Al-Kabi et al., 2005). Metode yang digunakan akan menghitung bobot dari masing-masing istilah (*term*) yang mengacu pada setiap kelas. Bobot suatu istilah akan lebih tinggi apabila sering muncul pada suatu kelas dan jarang muncul dikelas yang lainnya. Sebanyak 15 data dari masing-masing kelas digunakan sebagai data latih dan 5 data dari masing-masing kelas digunakan sebagai data uji. Dengan menggunakan TF-IDF, penelitian ini berhasil mendapatkan akurasi rata-rata sekitar 83.2%.

Hal serupa juga telah dikerjakan oleh Harrag (Harrag & El-Qawasmah, 2009) dalam penelitiannya namun dengan jumlah kelas yang lebih banyak yakni 14 kategori yang semuanya tersebar ke dalam 453 dokumen berbahasa Arab. Mereka menggunakan Artificial Neural Network (ANN) sebagai *classifier*. Dikarenakan jumlah kata-kata unik yang terdapat

dalam dokumen bisa sangat banyak dan bervariasi, Singular Value Decomposition (SVD) digunakan dalam penelitian untuk mencari fitur yang relevan bagi proses klasifikasi. Penggunaan SVD sebagai seleksi fitur berhasil meningkatkan performansi dengan akurasi sebesar 88.33% dibandingkan tanpa menggunakan proses seleksi fitur.

Hadis juga dikategorikan ke dalam hadis anjuran, larangan, dan informasi seperti yang telah dikerjakan oleh (Abu Bakar et al., 2019; Al Faraby et al., 2018; Fauzan et al., 2018; Hidayati et al., 2020; Mediameer et al., 2019; Purbolaksiono et al., 2020). Pengkategorian hadis tersebut dilakukan terhadap hadis Bukhari terjemahan Indonesia. (Fauzan et al., 2018) mengelompokkan masing-masing hadis tersebut ke dalam salah satu dari tiga kelas yang ada atau yang biasa disebut dengan klasifikasi multi-class. Pada penelitiannya, mereka menggunakan metode Latent Semantic Indexing sebagai seleksi fitur dan Bootstrap Aggregating (Bagging) untuk meningkatkan hasil akurasi dengan metode klasifikasi Support Vector Machine (SVM). LSI merupakan salah satu metode representasi fitur yang paling bagus untuk semantic kata. LSI bertujuan untuk mencari hubungan antara suatu istilah (*term*) terhadap dokumen. LSI akan menghitung korelasi semantic antara *term* dan dokumen pada representasi teks tidak terstruktur (Al-Anzi & AbuZeina, 2017).

Pekerjaan yang serupa juga telah dilakukan oleh (Al Faraby et al., 2018) dalam penelitiannya mengenai klasifikasi teks hadis ke dalam kategorisasi multi-label. Mereka menggunakan dataset yang sama dengan penelitian (Fauzan et al., 2018). SVM dan Neural Network dipilih sebagai metode klasifikasi dengan bantuan TF-IDF sebagai metode ekstraksi fitur. Beberapa Teknik pencocokan string berbasis aturan sederhana dibangun sebagai pembandingan pada metode yang diusulkan. Hal ini dilakukan karena pada saat penelitian tersebut dikerjakan, mengklasifikasikan hadis ke dalam ketiga kelompok (anjuran, larangan, dan informasi) merupakan pekerjaan yang sangat baru. Penelitian tersebut mendapatkan hasil akurasi yang cukup baik yaitu sekitar 88% menggunakan metrik pengukuran F1-Score.

(Mediameer et al., 2019) menggunakan ekstraksi fitur berbasis aturan pada metode TF-IDF dan Word2Vec. Percobaan dilakukan pada 2000 data multi-label hadis Bukhari terjemahan Indonesia. Metode yang diusulkan bekerja efektif pada dataset yang digunakan dan mampu melampaui metode sebelumnya dengan nilai Hamming Loss sebesar 0.0623.

Adapun (Abu Bakar et al., 2019; Hidayati et al., 2020) mengkategorikan klasifikasi hadis tersebut ke dalam bentuk klasifikasi multi-label. Klasifikasi multi-label adalah mengelompokkan setiap data ke dalam lebih dari satu kelas (G. Chen et al., 2017; R. Li et al., 2017). Hal ini berlaku pada data hadis

dimana suatu hadis mungkin mengandung makna anjuran dan larangan sekaligus. Dikarenakan hal itu, penelitian (Abu Bakar *et al.*, 2019; Hidayati *et al.*, 2020) memfokuskan untuk mengerjakan klasifikasi hadis terjemahan Indonesia tersebut ke dalam bentuk klasifikasi multi-label. Keduanya menggunakan metode klasifikasi yang dibantu dengan suatu metode seleksi fitur untuk meningkatkan hasil akurasi. Backpropagation dengan bantuan Information Gain yang dikerjakan oleh (Abu Bakar *et al.*, 2019) mendapatkan hasil cukup baik dengan akurasi sebesar 88.42% sedangkan k-Nearest Neighbor dengan bantuan Latent Semantic Analysis pada penelitian (Hidayati *et al.*, 2020) berhasil mendapatkan nilai akurasi sebesar 90.28%.

Klasifikasi hadis termasuk ke dalam salah satu kelompok kategorisasi teks. Saat ini, sangat banyak penelitian yang membahas kategorisasi teks (Daeli & Adiwijaya, 2020; Herlambang & Wijoyo, 2019; Hmeidi *et al.*, 2015; Pinheiro *et al.*, 2015; Purnomoputra & Wisesty, 2019; Shimpikar & Govilkar, 2017). Penelitian mengenai kategorisasi teks telah jauh berkembang dengan adanya metode-metode canggih seperti *neural network* dan *deep learning* (Johnson & Zhang, 2015; X. Zhang *et al.*, 2015). *Deep learning* merupakan salah satu metode yang dapat secara otomatis mencari fitur dari sejumlah data dengan proses pembelajaran tanpa harus berdasarkan aturan-aturan yang dibuat oleh manusia, khususnya untuk memproses data yang sangat banyak apabila dilakukan oleh manusia (Liang *et al.*, 2017).

(Abdi *et al.*, 2019) dalam penelitiannya mengenai *deep learning* membahas klasifikasi analisis sentimen menggunakan metode Recurrent Neural Network (RNN) dengan bantuan Long Short-Term Memory (LSTM) untuk mendapatkan keuntungan dari proses sekuensial dan menghilangkan kelemahan pada metode tradisional lainnya yaitu kehilangan informasi pada proses pembelajaran. Penelitian ini mengelompokkan data ke dalam dua kelas yaitu kelas positif dan kelas negatif. Metode yang dibangun dapat meningkatkan nilai F1-Score hingga 15% dibandingkan dengan metode pembandingan. Sementara itu, Chen *et al.* (B. Chen *et al.*, 2019) menggunakan dua lapisan Convolutional Neural Network (CNN) untuk menangkap fitur-fitur pada teks sebelum dilakukan klasifikasi menggunakan LSTM. Metode yang diajukan berhasil mengungguli beberapa metode seperti SVM, CNN, dan LSTM.

RNN sebagai metode klasifikasi berbasis *deep learning* telah banyak digunakan karena kemampuannya dalam melakukan proses data sekuensial (G. Chen *et al.*, 2017; Lai *et al.*, 2015; S. Li *et al.*, 2019; Luo, 2017). Luo (Luo, 2017) menerapkan RNN khususnya model LSTM untuk melakukan klasifikasi hubungan pada catatan klinis menggunakan dataset i2b2/VA. Level segmen LSTM digunakan pada penelitian tersebut untuk

melakukan klasifikasi hubungan dengan cara memisahkan konsep dan konteks dari suatu teks ke dalam lima segmen. Metode yang diajukan pada penelitian tersebut dengan hanya menggunakan fitur *word embedding* berhasil mendapatkan nilai f-measure hingga 0.800.

RNN juga digunakan sebagai media untuk pelatihan terhadap lapisan *pooling* pada CNN agar *pooling* dapat lebih beradaptasi dengan data dan komponen lainnya pada jaringan sehingga meningkatkan akurasi (S. Li *et al.*, 2019). Setiap fitur pada wilayah *pooling* dipindai sebagai suatu urutan dan menjadi masukan bagi lapisan *recurrent*. Hasil keluaran terakhir dari lapisan *recurrent* merupakan fitur gabungan dari wilayah *pooling local* dan akan menjadi suatu fitur gabungan. Hasilnya menunjukkan bahwa metode yang dibangun dapat mengungguli performansi dari fungsi *pooling* lainnya khususnya pada dataset citra CIFAR-10 yang digunakan pada penelitian tersebut.

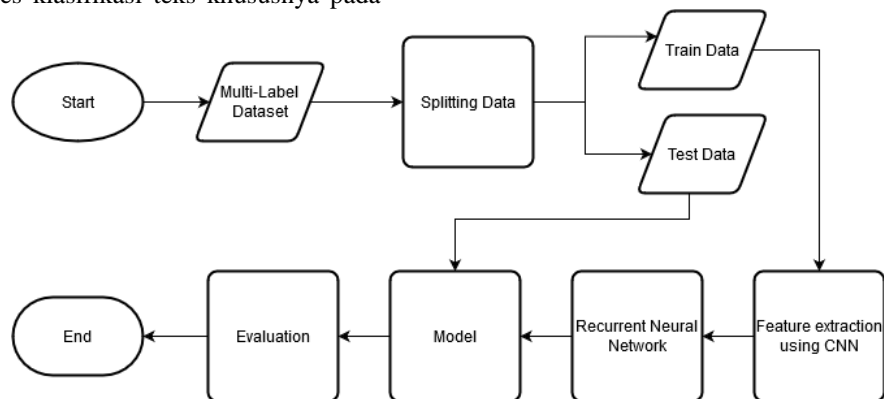
Sementara itu, CNN dengan kemampuannya dalam melakukan seleksi dan ekstraksi fitur dikombinasikan dengan RNN dalam melakukan klasifikasi berbasis urutan telah banyak dilakukan dan mendapatkan hasil performansi yang baik (G. Chen *et al.*, 2017; Choi *et al.*, 2017; Sivaram *et al.*, 2019). Penelitian (Choi *et al.*, 2017) memanfaatkan CNN sebagai metode untuk ekstraksi fitur dan RNN sebagai metode untuk meringkas fitur yang telah diekstrak. Mereka membandingkan metode yang diusulkan dengan beberapa jenis struktur pada CNN yang biasa digunakan pada penandaan musik. Metode CRNN yang dibangun dapat menghasilkan performansi yang baik daripada metode pembandingan dengan mengatur beberapa parameter yang tepat. Sementara itu, hal yang serupa juga telah dikerjakan dengan menggunakan data yang berbeda, yaitu data gambar untuk pendeteksian keakuratan wajah (Sivaram *et al.*, 2019). Penelitian tersebut memanfaatkan suatu lapisan yang ada pada CNN yaitu lapisan *pooling* untuk metode ekstraksi fitur sebelum akhirnya diteruskan menuju LSTM untuk proses deteksi wajah. Metode yang diusulkan dapat mengimbangi empat metode deteksi citra lainnya seperti CFAN (J. Zhang *et al.*, 2014), SDM (deng *et al.*, 2015), TCDN (Z. Zhang *et al.*, 2014), dan RCPR (Burgos-Artizzu *et al.*, 2013).

Penelitian ini mencoba untuk menggabungkan metode CNN sebagai ekstraksi fitur dengan beberapa struktur RNN sebagai metode klasifikasi (CRNN) untuk melakukan klasifikasi multi-label pada dataset hadis Bukhari terjemahan Indonesia sebanyak 7000 data teks. Gabungan kedua metode ini dikenal dengan nama Convolutional Recurrent Neural Network (CRNN). Penelitian ini membandingkan pengaruh penggunaan CNN sebagai metode ekstraksi fitur terhadap hasil klasifikasi dibandingkan jika tanpa menggunakan CNN.

2. METODE PENELITIAN

Penelitian ini menggabungkan metode CNN sebagai metode ekstraksi fitur dengan RNN sebagai metode klasifikasi untuk melihat seberapa besar pengaruh CNN dalam meningkatkan performansi sistem pada proses klasifikasi teks khususnya pada

dataset hadis Bukhari terjemahan Indonesia. Gambar 1 menunjukkan blok diagram dari metode yang diusulkan. Proses utama pada penelitian ini ada pada ekstraksi fitur menggunakan CNN.

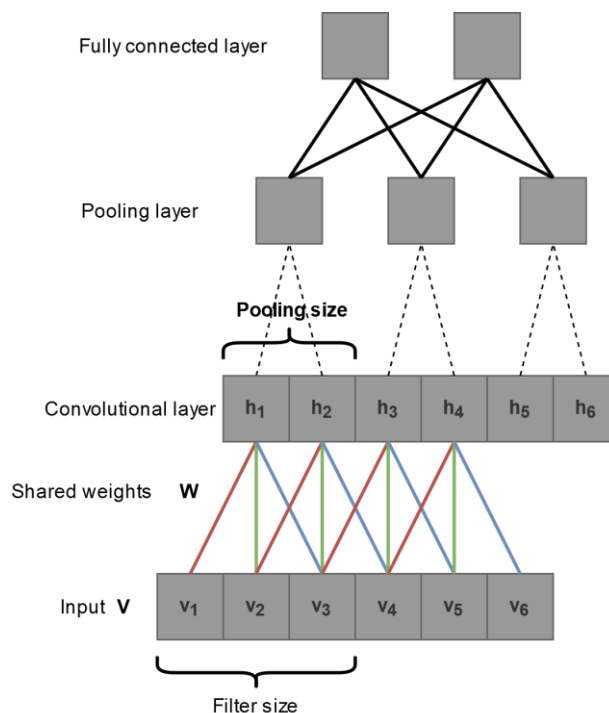


Gambar 1. Blok diagram dari metode yang diusulkan.

2.1. Convolutional Neural Network (CNN)

CNN merupakan suatu jaringan *feed-forward* yang terdiri dari berbagai kombinasi lapisan yaitu lapisan konvolusi yang bertindak sebagai penyaring fitur, lapisan *pooling* yang bertindak sebagai seleksi fitur, dan lapisan *fully connected (dense)* dimana lapisan konvolusi terhubung ke suatu bagian kecil pada lapisan sebelumnya yang disebut dengan filter (Hu et al., 2015; Prusa & Khoshgoftaar, 2017). Filter ini digunakan untuk menangkap informasi spasial

yang saling berhubungan dan berkorelasi. Biasanya ukuran yang digunakan pada filter ini relatif kecil (Prusa & Khoshgoftaar, 2017). Pada Gambar 2, terlihat bahwa terdapat tiga lapisan utama pada CNN. Dua lapisan pertama, yaitu lapisan konvolusi dan lapisan *pooling* merupakan lapisan yang berperan pada proses ekstraksi fitur (Yamashita et al., 2018).



Gambar 2. Arsitektur standar dari CNN yang terdiri dari lapisan konvolusi, lapisan *pooling*, dan lapisan *dense* (Hu et al., 2015).

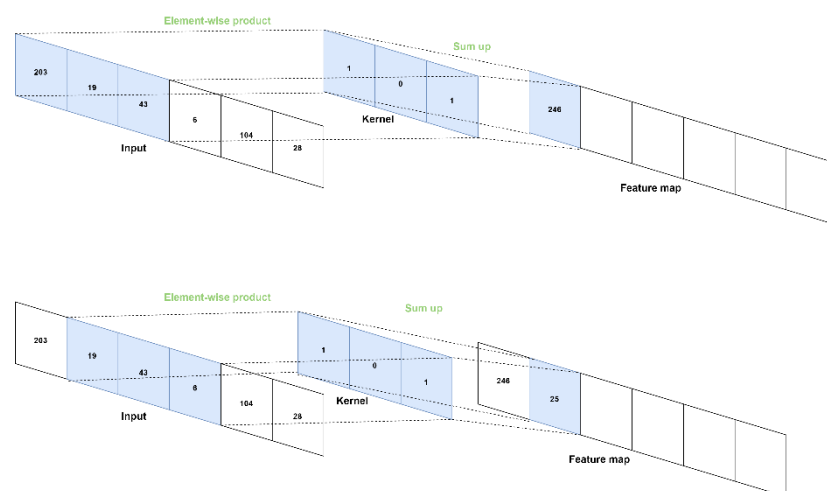
2.1.1. Lapisan konvolusi

Lapisan ini merupakan salah satu jenis lapisan dengan operasi linear yang digunakan untuk proses ekstraksi fitur dimana sejumlah kecil himpunan bilangan yang disebut dengan kernel diterapkan keseluruh masukan (himpunan bilangan). Suatu operasi yang disebut dengan *element-wise product* diterapkan antara masukan dengan kernel dan hasilnya dijumlahkan sehingga menghasilkan suatu himpunan bilangan baru yang disebut dengan *feature map* (lihat Gambar 3). Pada Gambar 2, kita melihat bahwa terdapat suatu parameter yang disebut dengan *shared weights* yang dikenal dengan nama kernel. Kernel ini menjadi kunci pada proses konvolusi dimana berperan dalam mengekstrak seluruh fitur dari masukan yang diterima. Kernel akan bergerak menelusuri semua masukan dengan nilai yang selalu sama (lihat Gambar 3) yang merupakan bobot bersama.

Beberapa *hyperparameter* yang digunakan pada operasi ini adalah ukuran dan jumlah kernel (Yamashita et al., 2018). Dikarenakan penelitian ini merupakan suatu klasifikasi teks, yangmana

masukan terdiri dari himpunan bilangan dalam bentuk matriks 1 (satu) dimensi, maka kernel yang digunakan pada lapisan konvolusi juga merupakan kernel dengan bentuk matriks 1 (satu) dimensi. Pada Gambar 3 terlihat bahwa ukuran kernel yang digunakan adalah matriks dengan dimensi 1 x 3.

Salah satu hal terpenting pada proses konvolusi adalah fungsi aktivasi. Fungsi aktivasi digunakan untuk merubah level aktivasi dari suatu unit (*neuron*) menjadi suatu sinyal keluaran. Fungsi aktivasi nonlinear yang saat ini paling banyak digunakan karena mudah diimplementasikan dan perhitungan yang sederhana adalah *rectified linear unit* (ReLU) dengan fungsi $f(x) = \max(0, x)$ (Krizhevsky et al., 2017; LeCun et al., 2015). ReLU dapat menangani masalah yang sering terjadi ketika proses *backpropagation* dikarenakan turunan fungsi ini akan menghasilkan nilai 0 jika $f(x) = 0$ dan bernilai 1 jika $f(x) = x$. Hal ini akan menyebabkan jaringan tidak akan mengalami masalah *vanishing gradient* maupun *exploding gradient* (Ide & Kurita, 2017; Talathi & Vartak, 2015).



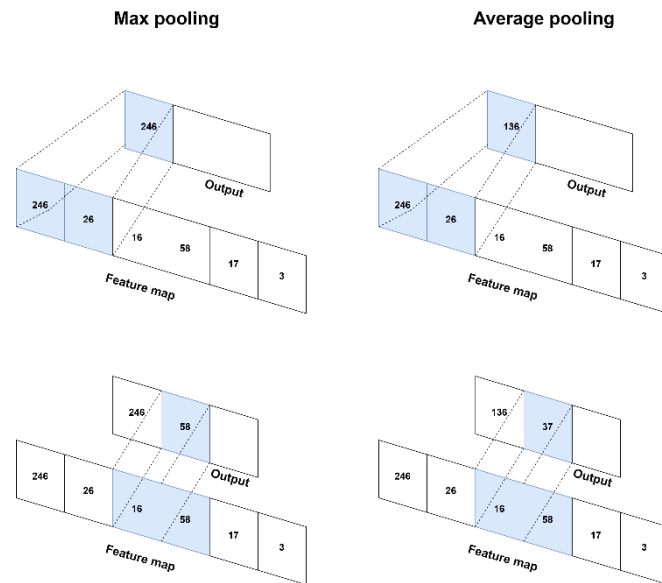
Gambar 3. Proses yang terjadi pada lapisan konvolusional.

2.1.2. Lapisan pooling

Lapisan *pooling* merupakan lapisan yang melakukan proses pengambilan sampel kecil dari *feature map* yang telah dihasilkan pada proses konvolusional. Pengambilan sampel ini dilakukan untuk mereduksi fitur dengan cara mengambil beberapa fitur penting dari *feature map*.

Beberapa teknik *pooling* yang sering digunakan pada CNN adalah *max pooling* dan *average pooling*. *Max pooling* akan mengambil nilai

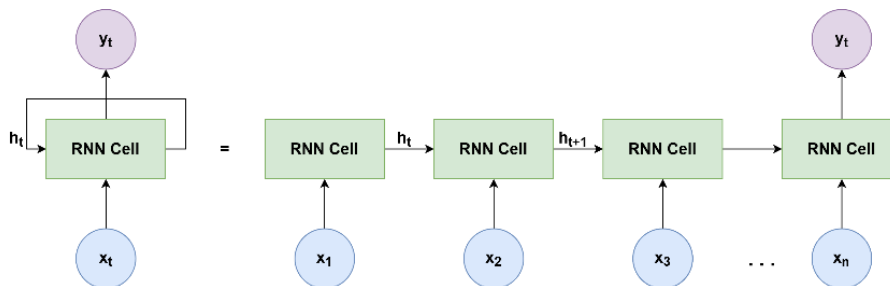
terbesar dari sejumlah kecil matriks sebagai fitur utama yang dipilih sedangkan *average pooling* akan mengambil rata-rata dari sejumlah kecil matriks sebagai fitur utama yang dipilih (lihat Gambar 4). Penelitian ini menggunakan *max pooling* sebagai metode ekstraksi fitur dikarenakan *max pooling* mengambil fitur terpenting dari data untuk selanjutnya menjadi fitur utama pada proses klasifikasi.

Gambar 4. Proses *pooling* pada CNN

2.2. Recurrent Neural Network (RNN)

Salah satu metode klasifikasi berbasis *deep learning* yang memiliki kemampuan untuk memodelkan data sekuens dan menangkap ketergantungan jangka panjang (Wang, 2018) khususnya pada data seperti teks dan suara. Tidak seperti *feedforward* pada *neural network* biasa, RNN dapat menggunakan status internal (memori) untuk memproses urutan masukan. Ini membuatnya dapat

digunakan untuk tugas-tugas yang tidak tersegmentasi seperti rekognisi tulisan tangan atau pengenalan ucapan yang saling terhubung. Di jaringan saraf lain, semua masukan independen satu sama lain. Namun dalam RNN, semua masukan terkait satu sama lain (lihat Gambar 5).



Gambar 5. Struktur jaringan saraf pada RNN.

Pertama, RNN mengambil x_1 dari urutan masukan dan kemudian outputnya h_1 yang bersamaan dengan x_2 adalah masukan untuk langkah selanjutnya. Jadi, h_1 dan x_2 adalah masukan untuk langkah selanjutnya. Demikian pula, h_2 dari berikutnya adalah masukan dengan x_3 untuk langkah selanjutnya dan seterusnya. Dengan cara ini, ia terus mengingat konteks saat berlatih.

Formula yang digunakan pada *current state* adalah seperti pada persamaan (1):

$$h_t = f(h_{t-1}, x_t) \quad (1)$$

dan aktifkan fungsi aktivasi menggunakan persamaan (2):

$$h_t = \text{ReLU}(W_{hh}h_{t-1} + W_{xh}x_t) \quad (2)$$

W adalah bobot, h adalah vektor dari lapisan tersembunyi, W_{hh} adalah bobot pada lapisan tersembunyi sebelumnya, W_{xh} adalah bobot pada keadaan masukan saat ini, **ReLU** adalah fungsi aktivasi, yang mengimplementasikan non-linearitas yang menekan aktivasi ke kisaran $[0, \infty]$. Fungsi aktivasi atau fungsi ambang atau fungsi transfer digunakan untuk mengubah tingkat aktivasi unit (neuron) menjadi sinyal keluaran (Karlik & Olgac, 2011). Fungsi ReLU dengan mudah bekerja dengan menetapkan nilai pada 0, yaitu $f(x) = \max(0, x)$. Sederhananya, ketika $x < 0$, maka akan menghasilkan 0 dan sebaliknya, akan menghasilkan

fungsi linear ketika $x \geq 0$ (Agarap, 2018) seperti persamaan (3) berikut:

$$\text{ReLU}(x) = f(x) = \max(0, x) \quad (3)$$

dan untuk lapisan keluaran adalah seperti persamaan (4):

$$y_t = \text{sigmoid}(W_{hy}h_t) \quad (4)$$

y_t adalah status keluaran. W_{hy} adalah bobot pada status keluaran dan sigmoid adalah fungsi aktivasi untuk lapisan keluaran. Sigmoid adalah fungsi aktivasi non-linear yang sebagian besar digunakan dalam jaringan saraf *feedforward*. Ini adalah fungsi nyata terdiferensiasi terbatas, didefinisikan untuk nilai masukan asli, dengan turunan positif dimana-mana dan beberapa tingkat kelancaran (Nwankpa et al., 2018). Fungsi Sigmoid dihitung menggunakan persamaan (5):

$$f(x) = \frac{1}{1 + \exp^{-x}} \quad (5)$$

Fungsi Sigmoid digunakan dalam model multi-label dimana ia mengembalikan probabilitas masing-masing kelas, nilai di atas 0,5 menjadi 1 dan lainnya menjadi 0, dimana 1 menunjukkan kelas target.

2.3. Convolutional Recurrent Neural Network (CRNN)

Convolutional Recurrent Neural Network (CRNN) memanfaatkan CNN sebagai metode ekstraksi fitur dan RNN sebagai metode untuk melakukan ringkasan dari fitur yang telah diekstrak (Choi et al., 2017). RNN berperan sebagai lapisan terakhir dari CNN dalam melakukan tahapan klasifikasi. Hal ini dilakukan karena kemampuan RNN dalam mendapatkan konteks makna dari kalimat yang terdiri dari susunan kata-kata yang saling terkait. RNN juga mampu menerima masukan data dengan dimensi beragam, tidak seperti CNN yang harus dilakukan metode *padding* terlebih dahulu untuk menyeragamkan panjang data pada keseluruhan dataset.

Selain digunakan pada teks, CRNN juga digunakan pada data gambar (Zuo et al., 2015) dan suara (Sigita et al., 2016). Metode yang digunakan pada penelitian (Zuo et al., 2015) mempelajari dependensi spasial antara daerah gambar untuk meningkatkan kekuatan diskriminatif representasi gambar. Lapisan CNN pertama kali diproses untuk menghasilkan fitur tingkat menengah. Lapisan RNN kemudian dipelajari untuk menyandikan dependensi spasial. CRNN dapat mempelajari representasi gambar yang lebih baik, terutama untuk gambar dengan ketergantungan kontekstual spasial yang jelas. Metode yang dibangun mencapai kinerja yang kompetitif untuk beberapa dataset seperti ILSVRC 2012, SUN 397, dan MIT indoor.

Pada penelitian yang kami lakukan, CNN dilakukan untuk mendapatkan fitur-fitur penting dari data teks hadis untuk meningkatkan performansi sistem. CNN juga digunakan untuk mengurangi waktu komputasi secara signifikan. RNN yang memiliki sistem memori sangat cocok digunakan

pada data teks saling berhubungan satu sama lain membentuk konteks makna kata. Memori yang digunakan sangat berguna untuk mengingat kata yang diproses diawal sebagai nilai masukan pada saat memproses kata saat ini.

Berikut adalah algoritma dari CRNN yang dibangun pada penelitian ini:

Langkah-1: representasi data untuk masukan CNN

Langkah-2: terapkan beberapa filter dan dikumpulkan untuk menangkap informasi kontekstual dan mengekstrak fitur-fitur penting dari setiap data

Langkah-3: x_t adalah masukan pada langkah waktu t . Misalnya, x_t bisa berupa vektor *one-hot* yang sesuai dengan kata kedua dari sebuah kalimat.

Langkah-4: h_t adalah status tersembunyi saat langkah t . Ini adalah "memori" jaringan. h_t dihitung berdasarkan keadaan tersembunyi sebelumnya dan masukan pada langkah saat ini (lihat persamaan 1). Fungsi f biasanya adalah nonlinier seperti tanh atau ReLU. Dalam persamaan 2, kita menggunakan ReLU sebagai fungsi f . $h_{(t-1)}$, diperlukan untuk menghitung status tersembunyi pertama serta menggunakan matriks identitas untuk inisialisasi bobot.

Langkah-5: y_t adalah output pada langkah t seperti yang bisa kita lihat dalam persamaan 4.

Langkah-6: Ulangi langkah 1-5 hingga mencapai kesalahan minimum.

2.4. Fungsi Optimasi

Fungsi optimasi digunakan pada saat melatih jaringan dengan mengubah nilai pada semua bobot menggunakan prinsip *gradient descent*. Sederhananya, masalah optimasi melibatkan sesuatu untuk menemukan parameter yang meminimalkan fungsi matematika. Fungsi yang ingin kita minimalkan saat membangun model *deep learning* yang rumit (melibatkan beberapa parameter seperti skalar, vektor, atau matriks), secara konseptual hanyalah menemukan parameter yang dapat meminimalkan fungsi tersebut (Ketkar, 2017).

Sebagai contoh pada *stochastic gradient descend* (SGD), kita asumsikan bahwa x adalah sebuah vektor. Kita ingin meminimalkan nilai $L(x)$, dimana kita ingin merubah nilai x sehingga $L(x)$ akan berkurang. Misalkan u merupakan sebuah unit vektor atau arah dari x yang seharusnya berubah dan α merupakan besarnya arah (skalar) dari perubahan ini. Semakin besar nilai α akan menyebabkan perubahan arah u yang besar, dimana hal ini tidak diinginkan.

Maka dari itu, kita ingin menemukan u sehingga pada persamaan (6):

$$\lim_{\alpha \rightarrow 0} L(x + \alpha u) \quad (6)$$

dapat diminimalkan. Karena itu (persamaan 7):

$$\lim_{\alpha \rightarrow 0} L(x + \alpha u) = u^T \nabla_x L(x) \quad (7)$$

Karenanya, kita sebenarnya ingin mencari u sehingga persamaan (8):

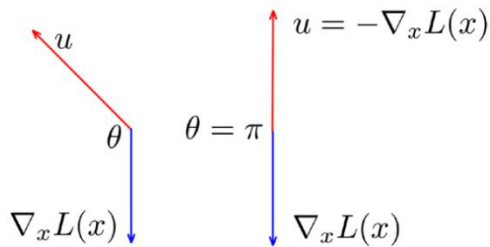
$$u^T \nabla_x L(x) \quad (8)$$

dapat diminimalkan. Perhatikan bahwa $\nabla_x L(x)$ merupakan turunan dari $L(x)$.

Diketahui bahwa u^T dan $\nabla_x L(x)$ adalah vektor, karena itu persamaan (9):

$$u^T \nabla_x L(x) = |u| \cdot |\nabla_x L(x)| \cos \theta \quad (9)$$

dimana θ adalah sudut diantara dua vektor (lihat Gambar 6).



Gambar 6. Menemukan arah yang diinginkan saat perubahan (Ketkar, 2017).

Nilai dari $\cos \theta$ akan minimal saat $\theta = \pi$, atau dengan kata lain vektor menunjuk kearah yang berlawanan. Karena itu, mengatur arah dari $u =$

Tabel 1. Representasi dataset.

Data hadis	Label (Kelas)		
	Anjuran	Larangan	Informasi
Aku datang untuk menjelaskan Lailatul Qodar kepada kalian, namun fulan dan fulan saling berdebat sehingga akhirnya diangkat (lailatul qodar), dan semoga menjadi lebih baik buat kalian, maka itu intailah (lailatul qodar) itu pada hari yang ketujuh, enam dan lima.	1	0	1
Tinggalkanlah dia, karena sesungguhnya malu adalah bagian dari iman. Islam dibangun diatas lima (landasan); persaksian tidak ada ilah selain Allah dan sesungguhnya Muhammad utusan Allah, mendirikan shalat, menunaikan zakat, haji dan puasa Ramadhan.	0	1	1
	0	0	1

Tabel 2. Informasi jumlah data per label.

Label (Kelas)	Jumlah data
Anjuran	1335
Larangan	844
Informasi	6634

3. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan terjemahan bahasa Indonesia dari hadis sahih Bukhari dengan 7000 data yang sebelumnya dilabeli dengan 3 (tiga) label yaitu 'anjuran', 'larangan', dan 'informasi' untuk data multi-label. Pada data multi-label, setiap data dapat terdiri dari salah satu kelas maupun lebih dari satu kelas (Abu Bakar et al., 2019). Saat ini, ada 7000 data yang telah diberi label dan divalidasi oleh Universitas Islam Negeri Bandung. Data ini menjadi masukan untuk sistem. Tabel 1 menunjukkan bentuk dataset yang digunakan pada penelitian ini.

$-\nabla_x L(x)$ akan mendapatkan tujuan yang diharapkan.

Salah satu metode optimisasi perubahan bobot jaringan yang telah dimodifikasi dari SGD adalah Adaptive Moment Estimation (Adam). Metode ini menghitung pembaruan dengan mempertahankan rata-rata tertimbang secara eksponensial dari g_i dan $(g_i)^2$ untuk setiap parameter (dilambangkan oleh subskrip i). Aturan pembaruan untuk Adam adalah seperti pada persamaan (10):

$$x_i = x_i - \frac{\alpha}{E[(g_i^{S-1})]^2} E[g_i^{S-1}] \quad (10)$$

Penting untuk diketahui bahwa $E[g_i^{S-1}]$ dan $E[(g_i^{S-1})]^2$ merupakan bias menuju nol pada langkah-langkah awal untuk tingkat peluruhan yang kecil (ada dua tingkat peluruhan disini – satu untuk $E[g_i^{S-1}]$ dan satu lagi untuk $E[(g_i^{S-1})]^2$ – yang dinotasikan oleh ρ_1 dan ρ_2). Nilai bias ini dapat diperbaiki dengan perhitungan pada persamaan (11) dan persamaan (12):

$$E[g_i^{S-1}] = \frac{E[g_i^{S-1}]}{1-\rho_1} \quad (11)$$

dan

$$E[(g_i^{S-1})]^2 = \frac{E[(g_i^{S-1})]^2}{1-\rho_2} \quad (12)$$

Penelitian ini fokus pada melihat seberapa banyak sistem yang dibangun dapat mengurangi waktu proses klasifikasi secara umum dengan tetap mempertahankan hasil akurasi yang baik. Untuk alasan ini, beberapa metode *deep learning* lain digunakan sebagai metode pembanding. Adapun metode yang digunakan adalah Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), dan Long Short-Term Memory (LSTM). Semua metode dijalankan dengan iterasi sebanyak 25 epochs.

Untuk menjalankan proses itu, diperlukan perangkat keras dengan spesifikasi yang tinggi.

Percobaan dilakukan dengan menggunakan Bahasa Pemrograman Python 3 di Google Collaboratory. Google Collaboratory menyediakan 12 GB RAM, Intel Xeon Processors 2.3 GHz, Tesla K80 GPU 12 GB, dan Disk 320 GB yang sangat cukup untuk menjalankan keseluruhan proses.

Seperti yang terlihat pada Gambar 2, ada beberapa lapisan di CNN. Lapisan pertama adalah lapisan masukan. Jumlah masukan untuk lapisan ini diperoleh dari panjang urutan maksimal data. Setelah lapisan masukan ini, ada lapisan konvolusi dan lapisan *pooling* yang digunakan untuk mengurutkan dan menilai fitur yang dianggap penting untuk proses klasifikasi. Fitur-fitur yang dipilih digunakan untuk mengklasifikasikan data ke dalam 3 (tiga) kelas.

Dari Gambar 7, kita dapat melihat bahwa lapisan masukan atau lapisan *embedding* memiliki 1647 masukan dari panjang urutan maksimal data.

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 1647, 32)	528512
conv1d (Conv1D)	(None, 1647, 32)	2080
max_pooling1d (MaxPooling1D)	(None, 823, 32)	0
conv1d_1 (Conv1D)	(None, 823, 32)	3104
max_pooling1d_1 (MaxPooling1D)	(None, 274, 32)	0
simple_rnn (SimpleRNN)	(None, 128)	20608
dense (Dense)	(None, 3)	387
Total params: 554,691		
Trainable params: 554,691		
Non-trainable params: 0		
None		

Gambar 7. Ringkasan model CRNN yang dibangun.

Adapun untuk hasil *precision* dari masing-masing kelas adalah sebesar 0.396 untuk kelas anjuran, 0.863 untuk kelas larangan, dan 1.0 untuk kelas informasi dengan total rata-rata *precision* untuk ketiganya adalah sebesar 0.753. Data *precision* yang sangat besar untuk kelas informasi didapatkan karena persebaran data yang tidak

Dengan beberapa lapisan penyatuan, akhirnya akan memiliki 274 fitur yang dipilih untuk diklasifikasikan ke dalam 3 (tiga) kelas. Model dilatih dalam 25 *epochs* menggunakan fungsi optimisasi perubahan bobot Adam dan mengkonsumsi waktu sekitar 5 detik untuk setiap *epochs* dengan akurasi sebesar 90.80% pada data latih (5600 sampel) dan 80.79% akurasi pada data uji (1400 sampel). Hasil akurasi yang didapatkan dari metode yang diusulkan (CRNN) melampaui hasil akurasi dari seluruh metode pembandingan. Adapun untuk waktu komputasi, metode yang diusulkan melampaui beberapa metode pembandingan meskipun lebih lambat dibandingkan dengan metode MLP dan CNN (masing-masing sekitar 1 detik untuk setiap *epochs*). Tabel 3 menampilkan statistik perbandingan dari metode yang digunakan.

merata. Hampir keseluruhan data termasuk ke dalam kelas informasi sehingga sistem yang dibangun bekerja cenderung *overfitting* terhadap kelas informasi. Hal ini terbukti dengan hasil sistem yang dibangun menunjukkan bahwa seluruh data yang diklasifikasi paling tidak termasuk ke dalam kelas informasi walaupun data tersebut sebenarnya bukan termasuk ke dalam kelas informasi.

Tabel 3. Hasil pengujian menggunakan beberapa metode pembandingan.

Metode	Waktu / epoch	Loss latih	Akurasi latih	Loss uji	Akurasi uji
Multi-Layer Perceptron (MLP)	< 1 detik	0.2162	73.23%	0.5209	72.50%
Convolutional Neural Network (CNN)	1 detik	0.0345	78.77%	0.3838	74.07%
Recurrent Neural Network (Simple)	27 detik	0.2499	71.52%	0.3580	72.50%
Recurrent Neural Network (LSTM)	52 detik	0.5352	71.48%	0.5381	72.50%
Convolutional Recurrent Neural Network (CRNN)	5 detik	0.0222	90.80%	0.2885	80.79%

Tabel 4. Perbandingan hasil klasifikasi benar dan salah pada data latih dan data uji.

Metode	Data latih		Data uji	
	Benar	Salah	Benar	Salah
Multi-Layer Perceptron (MLP)	4101	1499	1015	385
Convolutional Neural Network (CNN)	4411	1189	1037	363
Recurrent Neural Network (Simple)	4005	1595	1015	385
Recurrent Neural Network (LSTM)	4003	1597	1015	385
Convolutional Recurrent Neural Network (CRNN)	5085	515	1132	268

Secara keseluruhan, metode yang diusulkan yaitu CRNN dapat mengungguli metode *deep learning* lainnya dalam hal akurasi dan nilai *loss*. CNN berpengaruh sangat baik dalam mengurangi tingkat *loss*. Terlihat pada Tabel 3, metode yang menggunakan lapisan konvolusi dan *pooling* pada CNN sebagai ekstraksi fitur (CNN dan CRNN) bekerja sangat baik dalam mengurangi tingkat *loss* hingga mencapai angka 0.02 hanya menggunakan epochs sebesar 25. Hasil ini juga dibuktikan dengan perolehan data klasifikasi yang dianggap benar dan salah oleh setiap metode seperti yang ditunjukkan pada Tabel 4. Ini menandakan bahwa CNN dapat mengambil fitur yang sangat penting pada data untuk diteruskan menuju proses klasifikasi. CNN juga dapat menangani masalah yang umum terjadi pada RNN, yaitu *vanishing gradient* dan *exploding gradient* yang menyebabkan proses melatih jaringannya menjadi sangat lambat. Ini terbukti dari waktu eksekusi setiap *epoch* maksimal hanya 5 detik apabila menggunakan CNN sebagai metode reduksi dimensi.

4. KESIMPULAN

Convolutional Recurrent Neural Network (CRNN) yang berbasis pada *deep learning* dipilih sebagai metode seleksi dan reduksi dimensi dikarenakan metode ini melakukan segmentasi pada data, kemudian memilih fitur-fitur yang dianggap penting menggunakan suatu lapisan yang dinamakan lapisan *pooling* dan pada akhirnya melakukan klasifikasi data menggunakan memori dari konteks sebelumnya. CNN pada CRNN membantu melakukan seleksi dan reduksi fitur dengan sangat cepat sehingga akan mengurangi waktu komputasi dengan sangat signifikan.

Dari hasil pengujian, hampir seluruh data termasuk ke dalam salah satu kelas, yaitu kelas informasi sehingga menyebabkan metode yang dibangun menjadi *overfitting* terhadap kelas informasi. Hal ini disebut dengan data yang tidak seimbang (*imbalanced data*) seperti yang dapat dilihat pada Tabel 2. Kedepannya, penelitian ini akan dikembangkan dengan mencoba untuk menyamakan persebaran data pada tiap kelasnya. Beberapa metode inisialisasi bobot dan fungsi aktivasi pada *deep learning* juga akan diterapkan untuk mendapatkan hasil maksimal.

DAFTAR PUSTAKA

- ABDI, A., SHAMSUDDIN, S. M., HASAN, S., & PIRAN, J. 2019. Deep learning-based sentiment classification of evaluative text based on Multi-feature fusion. *Information Processing and Management*, 56(4), 1245–1259. <https://doi.org/10.1016/j.ipm.2019.02.018>
- ABU BAKAR, M. Y., ADIWIJAYA, & AL FARABY, S. 2019. Multi-Label Topic Classification of Hadith of Bukhari (Indonesian Language Translation) Using Information Gain and Backpropagation Neural Network. *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*, 344–350. <https://doi.org/10.1109/IALP.2018.8629263>
- AGARAP, A. F. 2018. *Deep Learning using Rectified Linear Units (ReLU)*. 1, 2–8. <http://arxiv.org/abs/1803.08375>
- AL-ANZI, F. S., & ABUZEINA, D. 2017. Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing. *Journal of King Saud University - Computer and Information Sciences*, 29(2), 189–195. <https://doi.org/10.1016/j.jksuci.2016.04.001>
- AL-KABI, M. N., GHASSAN, K., AL-SHALABI, R., AL-SINJILAWI, S. I., & AL-MUSTAFA, R. S. 2005. Al-Hadith Text Classifier. *Journal of Applied Sciences*, 5(3), 584–587. <https://doi.org/10.3923/jas.2005.584.587>
- AL FARABY, S., JASIN, E. R. R., KUSUMANINGRUM, A., & ADIWIJAYA. 2018. Classification of hadith into positive suggestion, negative suggestion, and information. *Journal of Physics: Conference Series*, 971(1). <https://doi.org/10.1088/1742-6596/971/1/012046>
- ALDHLAN, K. A., ZEKEI, A. M., ZEKEI, A. M., & ALRESHIDI, H. A. 2013. Novel mechanism to improve hadith classifier performance. *Proceedings - 2012 International Conference on Advanced Computer Science Applications and Technologies, ACSAT 2012*, 512–517. <https://doi.org/10.1109/ACSAT.2012.93>
- BURGOS-ARTIZZU, X. P., PERONA, P., & DOLLAR, P. 2013. Robust Face Landmark Estimation under Occlusion. *2013 IEEE International Conference on Computer Vision*,

- 1513–1520.
<https://doi.org/10.1109/ICCV.2013.191>
- CHEN, B., HUANG, Q., CHEN, Y., CHENG, L., & CHEN, R. 2019. Deep Neural Networks for Multi-class Sentiment Classification. *Proceedings - 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018*, 854–859.
<https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00142>
- CHEN, G., YE, D., XING, Z., CHEN, J., & CAMBRIA, E. 2017. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. *Proceedings of the International Joint Conference on Neural Networks, 2017-May*, 2377–2383.
<https://doi.org/10.1109/IJCNN.2017.7966144>
- CHOI, K., FAZEKAS, G., SANDLER, M., & CHO, K. 2017. Convolutional recurrent neural networks for music classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2392–2396.
<https://doi.org/10.1109/ICASSP.2017.7952585>
- DAELI, N. O. F., & ADIWIJAYA. 2020. Sentiment analysis on movie reviews using Information gain and K-nearest neighbor. *JOURNAL OF DATA SCIENCE AND ITS APPLICATIONS*, 3(1), 1–7.
<https://doi.org/10.34818/JDSA.2020.3.22>
- DENG, J., SUN, Y., LIU, Q., & LU, H. 2015. Low rank driven robust facial landmark regression. *Neurocomputing*, 151(P1), 196–206.
<https://doi.org/10.1016/j.neucom.2014.09.052>
- FAUZAN, H., ADIWIJAYA, A., & AL-FARABY, S. 2018. Pengklasifikasian Topik Hadits Terjemahan Bahasa Indonesia Menggunakan Latent Semantic Indexing dan Support Vector Machine. *Jurnal Media Informatika Budidarma*, 2(4), 131.
<https://doi.org/10.30865/mib.v2i4.948>
- HARRAG, F., & EL-QAWASMAH, E. 2009. Neural network for Arabic text classification. *2nd International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2009*, 778–783.
<https://doi.org/10.1109/ICADIWT.2009.5273841>
- HERLAMBAH, A. D., & WIJOYO, S. H. 2019. Algoritma Naive Bayes untuk Klasifikasi Sumber Belajar Berbasis Teks pada Mata Pelajaran Produktif di SMK Rumpun Teknologi Informasi dan Komunikasi. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 6(4), 430.
<https://doi.org/10.25126/jtiik.2019641323>
- HIDAYATI, D. C., AL FARABY, S., & ADIWIJAYA, A. 2020. Klasifikasi Topik Multi Label pada Hadis Shahih Bukhari Menggunakan K-Nearest Neighbor dan Latent Semantic Analysis. *JURIKOM (Jurnal Riset Komputer)*, 7(1), 140.
<https://doi.org/10.30865/jurikom.v7i1.2013>
- HMEIDI, I., AL-AYYOUB, M., ABDULLA, N. A., Almodawar, A. A., Abooraig, R., & Mahyoub, N. A. 2015. Automatic Arabic text categorization: A comprehensive comparative study. *Journal of Information Science*, 41(1), 114–124.
<https://doi.org/10.1177/0165551514558172>
- HU, W., HUANG, Y., WEI, L., ZHANG, F., & LI, H. 2015. Deep Convolutional Neural Networks for Hyperspectral Image Classification. *Journal of Sensors*, 2015, 1–12.
<https://doi.org/10.1155/2015/258619>
- IDE, H., & KURITA, T. 2017. Improvement of learning for CNN with ReLU activation by sparse regularization. *Proceedings of the International Joint Conference on Neural Networks, 2017-May*, 2684–2691.
<https://doi.org/10.1109/IJCNN.2017.7966185>
- JOHNSON, R., & ZHANG, T. 2015. Semi-supervised convolutional neural networks for text categorization via region embedding. *Advances in Neural Information Processing Systems, 2015-Janua*, 919–927.
- KARLIK, B., & OLGAC, V. A. 2011. Performance Analysis of Various Activation Functions in Artificial Neural Networks. *International Journal of Artificial Intelligence And Expert Systems (IJAE)*, 1(4), 111–122.
<https://doi.org/10.1088/1742-6596/1237/2/022030>
- KETKAR, N. 2017. Deep Learning with Python. In *Deep Learning with Python*. Apress.
<https://doi.org/10.1007/978-1-4842-2766-4>
- KRIZHEVSKY, A., SUTSKEVER, I., & HINTON, G. E. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90.
<https://doi.org/10.1145/3065386>
- LAI, S., XU, L., LIU, K., & ZHAO, J. 2015. Recurrent convolutional neural networks for text classification. *Proceedings of the National Conference on Artificial Intelligence*, 3, 2267–2273.
- LECUN, Y., BENGIO, Y., & HINTON, G. 2015. Deep learning. *Nature*, 521(7553), 436–444.
<https://doi.org/10.1038/nature14539>
- LI, R., LIU, W., LIN, Y., ZHAO, H., & ZHANG, C. 2017. An Ensemble Multilabel Classification for Disease Risk Prediction. *Journal of Healthcare Engineering*, 2017.
<https://doi.org/10.1155/2017/8051673>
- LI, S., LI, W., COOK, C., ZHU, C., & GAO, Y. 2019. A fully trainable network with RNN-

- based pooling. *Neurocomputing*, 338(Shuai Li), 72–82. <https://doi.org/10.1016/j.neucom.2019.02.004>
- LIANG, H., SUN, X., SUN, Y., & GAO, Y. 2017. Text feature extraction based on deep learning: a review. *Eurasip Journal on Wireless Communications and Networking*, 2017(1), 1–12. <https://doi.org/10.1186/s13638-017-0993-1>
- LUO, Y. 2017. Recurrent neural networks for classifying relations in clinical notes. *Journal of Biomedical Informatics*, 72, 85–95. <https://doi.org/10.1016/j.jbi.2017.07.006>
- MEDIAMER, G., ADIWIJAYA & FARABY, S. A. 2019. 1. Development of rule-based feature extraction in multi-label text classification. *International Journal on Advanced Science, Engineering and Information Technology*, 9(4), 1460–1465. <https://doi.org/10.18517/ijaseit.9.4.8894>
- MUSLIM POPULATION BY COUNTRY 2020. (n.d.). Retrieved October 18, 2020, from <https://worldpopulationreview.com/country-rankings/muslim-population-by-country>
- NWANKPA, C., IJOMAH, W., GACHAGAN, A., & MARSHALL, S. 2018. *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*. 1–20. <http://arxiv.org/abs/1811.03378>
- PINHEIRO, R. H. W., CAVALCANTI, G. D. C., & REN, T. I. 2015. Data-driven global-ranking local feature selection methods for text categorization. *Expert Systems with Applications*, 42(4), 1941–1949. <https://doi.org/10.1016/j.eswa.2014.10.011>
- PRUSA, J. D., & KHOSHGOFTAAR, T. M. 2017. Improving deep neural network design with new text data representations. *Journal of Big Data*, 4(1). <https://doi.org/10.1186/s40537-017-0065-8>
- PURBOLAKSONO, M. D., RESKYADITA, F. D., ADIWIJAYA, SURYANI, A. A., & HUDA, A. F. 2020. Indonesian text classification using back propagation and sastrawi stemming analysis with information gain for selection feature. *International Journal on Advanced Science, Engineering and Information Technology*, 10(1), 234–238. <https://doi.org/10.18517/ijaseit.10.1.8858>
- PURNOMOPUTRA, R. B., & WISESTY, U. N. 2019. Sentiment Analysis of Movie Reviews using Naïve Bayes Method with Gini Index Feature Selection. *Journal of Data Science and Its Applications*, 2(July), 85–94. <https://doi.org/10.34818/jdsa.2019.2.36>
- SHIMPIKAR, S., & GOVILKAR, S. 2017. A Survey of Text Summarization Techniques for Indian Regional Languages. *International Journal of Computer Applications*, 165(11), 29–33. <https://doi.org/10.5120/ijca2017914083>
- SIGTIA, S., BENETOS, E., & DIXON, S. 2016. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 24(5), 927–939. <https://doi.org/10.1109/TASLP.2016.2533858>
- SIVARAM, M., PORKODI, V., MOHAMMED, A. S., & MANIKANDAN, V. 2019. *Detection of accurate facial detection using hybrid deep convolutional recurrent neural network*. 1844–1850. <https://doi.org/10.21917/ijsc.2019.0256>
- TALATHI, S. S., & VARTAK, A. 2015. *Improving performance of recurrent neural network with relu nonlinearity*. <http://arxiv.org/abs/1511.03771>
- WANG, B. 2018. Disconnected recurrent neural networks for text categorization. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1, 2311–2320. <https://doi.org/10.18653/v1/p18-1215>
- YAMASHITA, R., NISHIO, M., DO, R. K. G., & TOGASHI, K. 2018. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4), 611–629. <https://doi.org/10.1007/s13244-018-0639-9>
- ZHANG, J., SHAN, S., KAN, M., & CHEN, X. 2014. *Coarse-to-Fine Auto-Encoder Networks (CFAN) for Real-Time Face Alignment* (pp. 1–16). https://doi.org/10.1007/978-3-319-10605-2_1
- ZHANG, X., ZHAO, J., & LECUN, Y. 2015. Character-level Convolutional Networks for Text Classification. *Advances in Neural Information Processing Systems*, 649–657. <http://arxiv.org/abs/1502.01710>
- ZHANG, Z., LUO, P., LOY, C. C., & TANG, X. 2014. Facial Landmark Detection by Deep Multi-task Learning. *Proceedings of International Conference on European Conference on Computer Vision*, 94–108. https://doi.org/10.1007/978-3-319-10599-4_7
- ZUO, Z., SHUAI, B., WANG, G., LIU, X., WANG, X., WANG, B., & CHEN, Y. 2015. Convolutional recurrent neural networks: Learning spatial dependencies for image representation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2015-Octob*, 18–26. <https://doi.org/10.1109/CVPRW.2015.7301268>