 ISSN: 2684-8473

Implementasi Ekstraksi Fitur untuk Klasifikasi Suara Urban Menggunakan Deep Learning

Via Wahyuningtyas

Informatika, Universitas Ahmad Dahlan, Yogyakarta Indonesia
via1700018067@webmail.uad.ac.id

INFORMASI ARTIKEL

Histori Artikel
Diterima : 25 September 2020
Direvisi : 25 Februari 2021
Diterbitkan : 08 Maret 2021

Kata Kunci:
MFCC
Audio
CNN
classification
feature extraction

ABSTRAK

Pengolahan suara merupakan konsep yang sangat penting untuk semua jenis sistem yang membutuhkan interaksi manusia dalam kegiatan sehari-hari. Salah satu teknik yang digunakan dalam pengolahan suara adalah ekstraksi ciri suara dan klasifikasi yang memiliki pengaruh langsung dalam sistem pengenalan suara. Namun, teknik pengenalan audio yang telah dikembangkan sangat beragam yang bertujuan untuk memperbaiki dan meningkatkan efisiensi akurasi, pengenalan pola, pemrosesan sinyal, ekstraksi dan tingkat pengenalan untuk menghasilkan klasifikasi yang akurat. Dalam penelitian ini, untuk menganalisis dan membuktikan bahwa menggunakan metode *Melf-Frequency Cepstral Coefficients* (MFCC) untuk mengekstraksi data suara dari sampel yang berupa input suara dari lingkungan perkotaan dapat diimplementasikan dengan baik, kemudian melakukan klasifikasi dengan menggunakan Convolutional Neural Network (CNN) untuk menyempurnakan model dengan skor Akurasi Klasifikasi yang baik. Dari hasil penelitian, bahwa model berkinerja dengan sangat baik dan juga dapat di prediksi dengan baik saat diuji terhadap data audio baru.

I. Pendahuluan

Suara urban memiliki pengaruh besar terhadap cara kita memandang tempat. Namun, perencanaan kota terutama berkaitan dengan kebisingan, hanya karena suara-suara yang mengganggu[1]. dan energi yang terkandung dalam suara/bunyi dapat meningkat secara cepat dan dapat menempuh jarakyang sangat jauh[2]. Pengolahan suara merupakan konsep yang sangat penting untuk semua jenis sistem yang membutuhkan interaksi manusia dalam kegiatan sehari-hari. Salah satu teknik yang digunakan dalam pengolahan suara adalah ekstraksi ciri suara dan klasifikasi yang memiliki pengaruh langsung dalam sistem pengenalan suara[3]. Suara juga selalu ada di sekitar baik secara langsung atau tidak langsung, selalu berhubungan dengan audio data. Di sekitar lingkungan bisa terdengar suara seperti kegiatan sehari-hari yaitu saat sedang bercakap, mendengarkan musik, derai hujan, mobil bejalan, atau lainnya yang dapat didengar setiap hari. Otak manusia terus memproses dan memahami audio ini baik secara sadar atau tidak sadar untuk memberikan informasi tentang lingungan sekitar.

Permasalahan utama yang terjadi apabila hendak mengenali suatu pola tertentu adalah bagaimana proses akuisisi data dilakukan sehingga menghasilkan data yang representatif dan konsisten terhadap sampel yang diberikan. Terdapat banyak penelitian mengenai pengenalan pola suara, namun, Teknik pengenalan audio yang telah dikembangkan sangat beragam yang bertujuan untuk memperbaiki dan meningkatkan efisiensi akurasi, pengenalan pola, pemrosesan sinyal, ekstraksi dan tingkat pengenalan untuk menghasilkan klasifikasi yang akurat. Klasifikasi adalah menggolongkan obyek pada kelasnya masing-Masing[4]. Berbagai model telah diterapkan pada berbagai penelitian, namun dari semua itu memiliki masalah yang sama dengan akurasi maksimum yang diperoleh[3]. Oleh karena itu penelitian ini diharapkan dapat mengetahui ciri bentuk pola suara melalui sampel audio dalam format yang dapat dibaca komputer (seperti file .wav) dari beberapa durasi detik. Ekstraksi ciri bertujuan untuk menajamkan perbedaan pola sehingga akan memudahkan dalam memisahkan kategori-kategori klas untuk proses klasifikasi.

Tujuan utama dari penulisan ini adalah untuk menganalisis dan membuktikan bahwa menggunakan metode *Melf-Frequency Cepstral Coefficients* (MFCC) untuk mengekstraksi data suara dari sampel yang berupa input suara dari lingkungan perkotaan dapat diimplementasikan dengan baik, kemudian melakukan klasifikasi

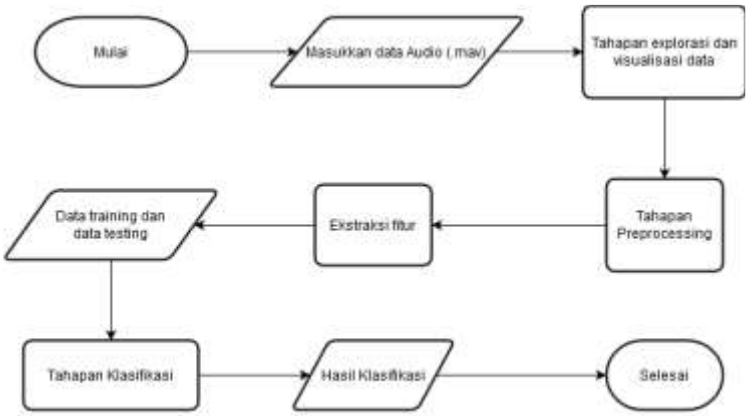
dengan menggunakan *Convolutional Neural Network* (CNN) untuk menyempurnakan model agar meningkatkan skor Akurasi Klasifikasi dengan tujuan sehingga komputer dapat mengidentifikasi suara secara baik dan konsisten.

II. Material dan Metode

A. Desain Sistem

Sistem ini terdapat beberapa tahap, yaitu tahap pra proses, tahap ekstraksi fitur, dan tahap uji coba klasifikasi. Pada tahap pra proses dilakukan menggunakan librosa untuk preprocessing, proses pengambilan data sampel dari berkas audio format (.wav) yang akan diolah, kemudian dilanjutkan dengan pembagian data sampel kedalam bagian-bagian yang lebih kecil dengan ukuran sama yang biasa disebut dengan frame. Panjang frame ditentukan oleh nilai bit-depth, sample rate dan Audio channels yang didapat pada proses pembacaan berkas audio. Tahap ekstraksi fitur, Tahap ekstraksi fitur, merupakan tahapan dilakukannya pengambilan nilai 6 fitur pada tiap data frame yang terbentuk kemudian dilakukan pengambilan nilai mean, standard deviasi, kurtosis, dan skewness sehingga pada satu berkas mp3 yang diekstraksi, didapatkan 24 atribut[5]. Namun, dalam penelitian ini menggunakan berkas file .wav. Proses yang dilakukan adalah dengan melakukan ekstraksi fitur pada suara seperti Pitch, MFCC, Wavelet, ZCR dan energi[6]. Oleh karena itu, di penelitian ini menggunakan *Melf-Frequency Cepstral Coefficients* (MFCC).

Tahap klasifikasi, merupakan tahapan yang dilakukan untuk menguji hasil ekstraksi fitur dalam merepresentasikan suara lingkungan perkotaan yang dimiliki. Data nilai atribut yang dihasilkan pada tahapan ekstraksi fitur akan digunakan sebagai data training dan data testing untuk dilakukan klasifikasi pada tahapan ini dengan menggunakan metode *Convolutional Neural Network* (CNN) agar skor klasifikasi akurat guna mengetahui kelayakan fitur yang dihasilkan untuk mengkelompokkan klas suara dilingkungan perkotaan. Gambaran diagram alir ini ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alir Sistem Utama

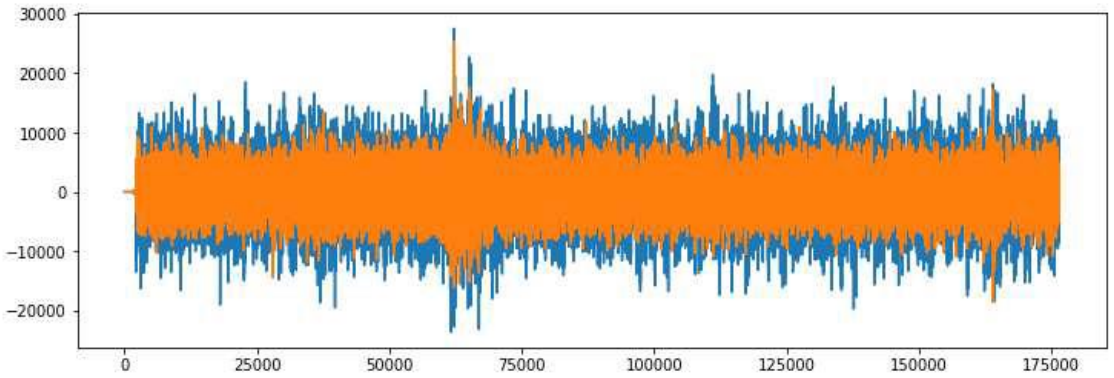
B. Method

1. Tahap Pra-Pemrosesan

   a. Sampling rate
      Fungsi muatan Librosa mengubah laju pengambilan sampel menjadi 22,05 KHz yang dapat kita gunakan sebagai tingkat perbandingan kami.
      Original sample rate: 44100
      Librosa sample rate: 22050
   b. Bit-depth
      Fungsi beban Librosa juga akan menormalkan data sehingga nilainya berkisar antara -1 dan 1. Ini menghilangkan kerumitan dataset yang memiliki kisaran kedalaman bit yang luas.
      Original audio file min~max range: -23628 to 27507
      Librosa audio file min~max range: -0.50266445 to 0.74983937
   c. Audio Channels
      Librosa juga akan mengonversi sinyal menjadi mono, artinya jumlah saluran akan selalu 1.

```
#Audio Channels
import matplotlib.pyplot as plt

# Original audio with 2 channels
plt.figure(figsize=(12, 4))
plt.plot(scipy_audio)
```

```
[<matplotlib.lines.Line2D at 0x1c178aa2b0>,
 <matplotlib.lines.Line2D at 0x1c178aa400>]
```



Gambar 2. Tahap Pra Pemrosesan

2. Ekstraksi Fitur

Ekstraksi fitur dengan menggunakan metode *Melf-Frequency Cepstral Coefficients* (MFCC) merupakan ekstraksi fitur yang paling sering digunakan dalam pemrosesan suara, karena dapat mempresentasikan sinyal dengan baik. Langkah-langkah proses MFCC berdasarkan pada perbedaan dari frekuensi yang terdengar oleh pendengaran manusia melalui panca indranya, sehingga dapat seperti layaknya manusia merepresentasikan sinyal suara. Dibawah Ini menunjukkan librosa menghitung serangkaian 40 MFCC lebih dari 173 frame.

```
In [7]: mfccs = librosa.feature.mfcc(y=librosa_audio, sr=librosa_sample_rate, n_mfcc=40)
        print(mfccs.shape)
```

```
(40, 173)
```

Gambar 3. Tahap Ekstraksi Fitur

3. Metode *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah pengembangan dari *Multilayer Perceptron (MLP)* yang didesain untuk mengolah data dua dimensi. *CNN* termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada penelitian ini, Metode *CNN* untuk melakukan klasifikasi yang memiliki nilai akurasi yang akurat.

III. Hasil dan Pembahasan

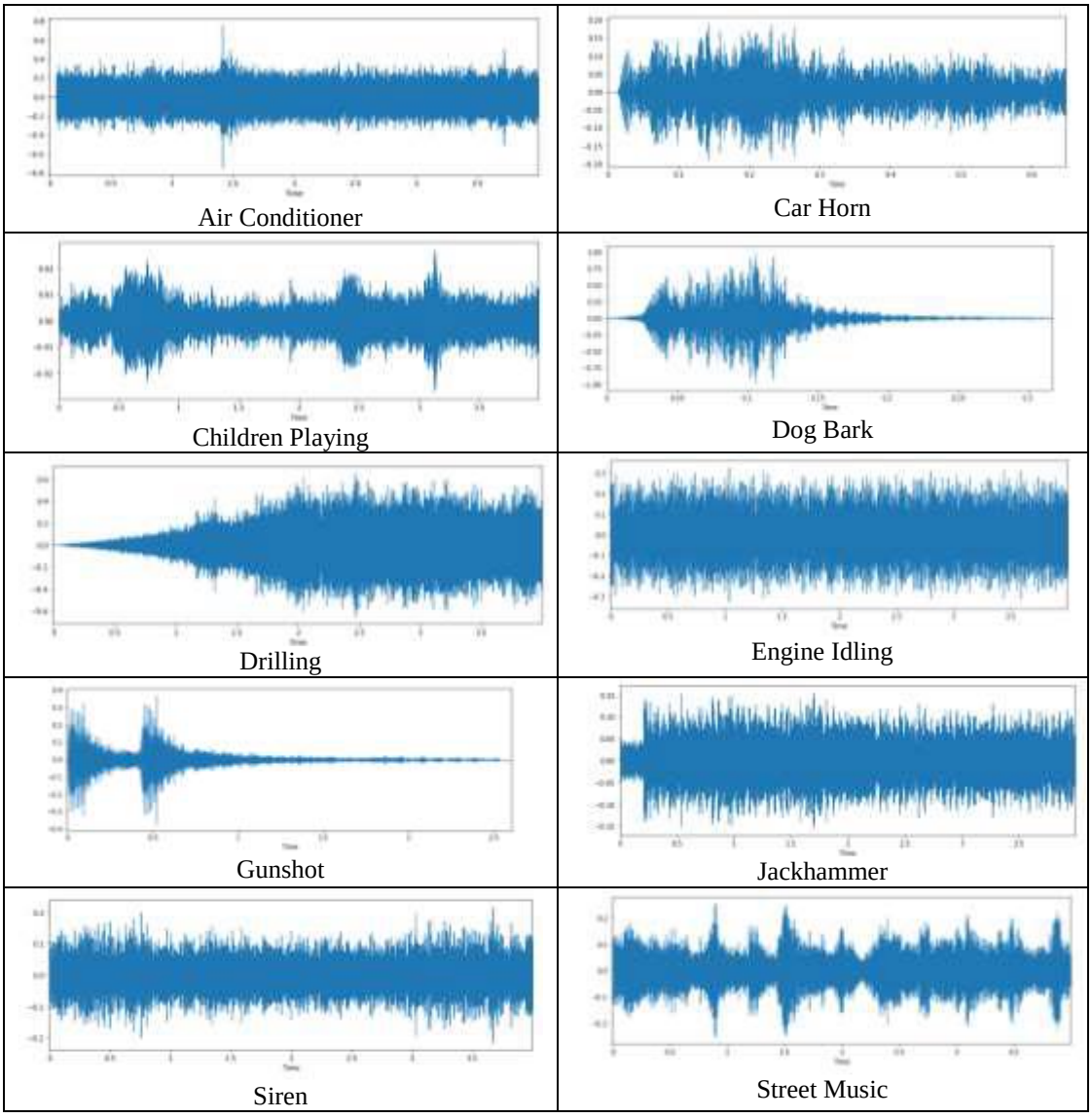
Lingkungan pengujian sistem ini dilakukan dengan menggunakan sebuah laptop dengan spesifikasi prosessor Intel® Celeron® CPU @ 1.10GHz dengan RAM sebesar 4.0 GB. Data yang digunakan untuk pengujian merupakan berkas MP3 yang diunduh dari website https://urbansounddataset.weebly.com/urbansound8k.html. Data pengujian dibagi menjadi dua yaitu data untuk *training*, dan data untuk *testing*. Dataset berisi 8732 kutipan suara (<= 4s) dari suara lingkungan perkotaan dari 10 kelas, yaitu *Air Conditioner, Car Horn, Children Playing, Dog bark, Drilling, Engine Idling, Gun Shot, Jackhammer, Siren*, dan *Street Music.*

A. Tahapan Eksplorasi dan Visualisasi Data

Data yang akan kami analisis untuk setiap kutipan suara pada dasarnya adalah array satu dimensi atau vektor dari nilai amplitudo. Untuk analisis audio, akan menggunakan perpustakaan berikut:

1. IPython.display.Audio = Ini memungkinkan kami memutar audio langsung di Jupyter Notebook.
2. Librosa = librosa adalah paket Python untuk pemrosesan musik dan audio oleh Brian McFee dan akan memungkinkan kita untuk memuat audio di notebook kita sebagai array numpy untuk analisis dan manipulasi.

Melakukan pemeriksaan data untuk setiap pola secara visual dengan menggunakan librosa untuk mengimplementasikan file audio (.mav) dan matplotlib untuk menampilkan bentuk gelombang.



Gambar 4. Tahap Visualisasi Data

Setelah melakukan beberapa visual dari audio tersebut, maka dapat dilihat perbedaannya dari bentuk gelombang. Tetapi masih ada beberapa kemiripan bentuk gelombang suara antara :

1. *Air Conditioner*, *Drilling*, *Idling Machine*, dan *jackhammer* (mesin penghancur beton) yang memiliki bentuk serupa.
2. *Dog Bark*, *Gunshot* dan *Car Horn* yang memiliki bentuk serupa.
3. *Children Playing* dan *Street Music* yang memiliki bentuk serupa.

B. Arsitektur model Convolutional Neural Network (CNN)

Menggunakan model *Convolution Neural Network (CNN)* dengan menggunakan Keras dan backend Tensorflow. Dengan menggunakan model sekuensial, dimulai dengan arsitektur model sederhana, yang terdiri dari empat lapisan konvolusi Conv2D, dengan lapisan hasil akhir kami menjadi lapisan padat. Lapisan konvolusi dirancang untuk deteksi fitur. Ia bekerja dengan menggeser jendela filter ke input dan melakukan perkalian matriks dan menyimpan hasilnya dalam peta fitur. Operasi ini dikenal sebagai konvolusi.

```
num_rows = 40
num_columns = 174
num_channels = 1

x_train = x_train.reshape(x_train.shape[0], num_rows, num_columns, num_channels)
x_test = x_test.reshape(x_test.shape[0], num_rows, num_columns, num_channels)

num_labels = yy.shape[1]
filter_size = 2

# Construct model
model = Sequential()
model.add(Conv2D(filters=16, kernel_size=2, input_shape=(num_rows, num_columns, num_channels), activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Dropout(0.2))

model.add(Conv2D(filters=32, kernel_size=2, activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Dropout(0.2))

model.add(Conv2D(filters=64, kernel_size=2, activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Dropout(0.2))

model.add(Conv2D(filters=128, kernel_size=2, activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Dropout(0.2))
model.add(GlobalAveragePooling2D())

model.add(Dense(num_labels, activation='softmax'))
```

Gambar 5. Algoritma *Convolution Neural Network (CNN)*

C. Training dan Predicting

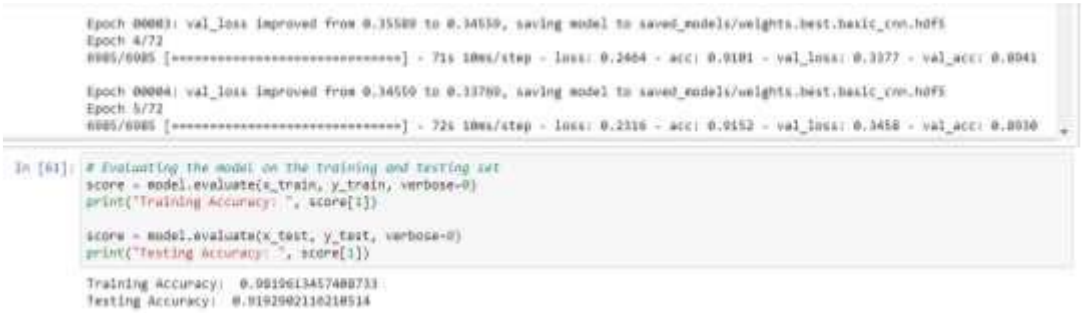
Tahapan pelatihan model dilakukan *CNN* dapat memakan waktu yang cukup lama, kami akan mulai dengan jumlah zaman yang rendah dan ukuran kumpulan yang rendah. Jika kita dapat melihat dari output bahwa model sedang konvergen, kita akan menambah kedua bilangan.

1. Training

Tahapan training akan meninjau keakuratan model pada pelatihan dan set data uji. Hasil skor akurasi Pelatihan dan Pengujian keduanya tinggi dan sangat baik.

Training Accuracy : 0.919613457408733
Testing Accuracy : 0.9192902116210514

```
In [60]: from keras.callbacks import ModelCheckpoint
         from datetime import datetime

         #num_epochs = 12
         #num_batch_size = 128

         num_epochs = 72
         num_batch_size = 256

         checkpointer = ModelCheckpoint(filepath='saved_models/weights.best.basic_cnn.hdf5',
                                        verbose=1, save_best_only=True)
         start = datetime.now()

         model.fit(x_train, y_train, batch_size=num_batch_size, epochs=num_epochs, validation_data=(x_test, y_test), callbacks=[checkpoint

         duration = datetime.now() - start
         print("Training completed in time: ", duration)

Train on 6985 samples, validate on 1747 samples
Epoch 1/72
6985/6985 [==============================] - 76s 11ms/step - loss: 0.2628 - acc: 0.9085 - val_loss: 0.3790 - val_acc: 0.8775

Epoch 00001: val_loss improved from inf to 0.37902, saving model to saved_models/weights.best.basic_cnn.hdf5
Epoch 2/72
6985/6985 [==============================] - 73s 10ms/step - loss: 0.2668 - acc: 0.9059 - val_loss: 0.3559 - val_acc: 0.8878

Epoch 00002: val_loss improved from 0.37902 to 0.35580, saving model to saved_models/weights.best.basic_cnn.hdf5
Epoch 3/72
6985/6985 [==============================] - 72s 10ms/step - loss: 0.2410 - acc: 0.9184 - val_loss: 0.3456 - val_acc: 0.8930
```



Gambar 6. Proses Training

2. Predicting

Memodifikasi metode sebelumnya untuk menguji prediksi model pada file .wav audio yang ditentukan.

```
In [50]: def print_prediction(file_name):
             prediction_feature = extract_features(file_name)
             prediction_feature = prediction_feature.reshape(1, num_rows, num_columns, num_channels)

             predicted_vector = model.predict_classes(prediction_feature)
             predicted_class = le.inverse_transform(predicted_vector)
             print("The predicted class is:", predicted_class[0], '\n')

             predicted_proba_vector = model.predict_proba(prediction_feature)
             predicted_proba = predicted_proba_vector[0]
             for i in range(len(predicted_proba)):
                 category = le.inverse_transform(np.array([i]))
                 print(category[0], "\t\t : ", format(predicted_proba[i], '.32f') )
```

Gambar 7. Proses Prediksi model

3. Analisis

Melakukan prediksi menggunakan subbagian file audio sampel yang sudah ada. Hasil dapat dilihat bahwa model berkinerja dengan sangat baik dan juga dapat di prediksi dengan baik saat diuji terhadap data audio baru. Skor akurasi tertinggi di class :

a. *Audio Air Conditioner*, di class air_conditioner : 0.90663295842681884765625000000000
b. *Audio Drilling*, di class drilling : 0.99598699808120727539062500000000
c. *Audio Street music*, di class street_music : 0.96923023462295532226562500000000
d. *Audio Dog Bark*, di class dog_bark : 0.84292083978652954101562500000000

```
In [51]: # Class: Air Conditioner

filename = '../UrbanSound Dataset sample/audio/100852-0-0-0.wav'
print_prediction(filename)
```

```
The predicted class is: air_conditioner

air_conditioner                 :  0.90663295984268188476562500000000
car_horn                :  0.00000379312382392527069896459579
children_playing                :  0.00372877437621355056762695312500
dog_bark                :  0.00003181818829034455120563507080
drilling                :  0.00387497572228312492370605468750
engine_idling           :  0.00299200275912880897521972656250
gun_shot                :  0.00765613839030265808105468750000
jackhammer              :  0.07329261302947998046875000000000
siren           :  0.00018024632299784570932388305664
street_music            :  0.00160682143177837133407592773438
```

Gambar 8. Percobaan pada *audio air conditioner*

```
In [52]: # Class: Drilling

filename = '../UrbanSound Dataset sample/audio/103199-4-0-0.wav'
print_prediction(filename)
```

```
The predicted class is: drilling

air_conditioner                 :  0.000709912739694118499975585937500
car_horn                :  0.000000017771748517247942800167 95
children_playing                :  0.000014050696336196859814643859863
dog_bark                :  0.000000471112429067943594418466 09
drilling                :  0.995986998081207275390625000000000
engine_idling           :  0.000003546584139257902279496192 93
gun_shot                :  0.000000032232076563332157093100 25
jackhammer              :  0.000529039069078862667083740234 38
siren           :  0.000000983402628662588540464639 66
street_music            :  0.002754879882559180259704589843 75
```

Gambar 9. Percobaan Audio Drilling

```
In [53]: # Class: Street music

filename = '../UrbanSound Dataset sample/audio/101848-9-0-0.wav'
print_prediction(filename)
```

```
The predicted class is: street_music

air_conditioner                 :  0.000114960152131970971822738647 46
car_horn                :  0.000792882812675088644027709960 94
children_playing                :  0.017915384843945503234863281250 00
dog_bark                :  0.002579237101599574089050292968 75
drilling                :  0.000079045399616006761789321899 41
engine_idling           :  0.000060611935623455795266548156 738
gun_shot                :  0.000000000074822682771813475710 59
jackhammer              :  0.000004578259904519654810428619 38
siren           :  0.009223079308886745452880859375000
street_music            :  0.969230234622955322265625000000 00
```

Gambar 10. Percobaan Audio Street Music

```
In [55]: filename = '../Evaluation audio/dog_bark_1.wav'
print_prediction(filename)
```

```
The predicted class is: dog_bark

air_conditioner                 :  0.000530691642779856920242309570 31
car_horn                :  0.018079748377203941345214843750 00
children_playing                :  0.009588890708982944488525390625 00
dog_bark                :  0.842920839786529541015625000000 00
drilling                :  0.022515686228871345520019531250 00
engine_idling           :  0.002860577078536152839660644531 25
gun_shot                :  0.092330768704414367675781250000 00
jackhammer              :  0.001473494106903672218322753906 25
siren           :  0.007028585299849510192871093750 00
street_music            :  0.002670840360224246978759765625 00
```

Gambar 11. Percobaan pada Dog Bark

IV. Kesimpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa pengklasifikasian suara lingkungan perkotaan dapat dilakukan dengan suatu metode ekstraksi ciri *Melf-Frequency Cepstral Coefficients (MFCC)* kemudian, Menggunakan *Convolutional Neural Network (CNN)* untuk memperoleh hasil akurasi yang sangat baik untuk pengklasifikasian suara lingkungan perkotaan.

Daftar Pustaka

[1] L. M. Aiello, R. Schifanella, D. Quercia, and F. Aletta, "Chatty maps: Constructing sound maps of urban areas from social media data," *R. Soc. Open Sci.*, vol. 3, no. 3, 2016.

[2] M. D. Egan, J. D. Quirt, M. Z. Rousseau, R. T. Beyer, and S. Walker, "Architectural Acoustics View online : https://doi.org/10.1121/1.398174 View Table of Contents : https://asa.scitation.org/toc/jas/86/2 Published by the Acoustical Society of America Architectural Acoustics : Principles and Practice The Journal of the Acoustical Society of America 104 , 3151 (1998); https://doi.org/10.1121/1.423953 REVIEWS Fourier Analysis," vol. 4, no. 1989, 2010.

[3] Yousra F. Al-Irhaim dan Enaam Ghanem Saeed," Arabic Word Recognition Using Wavelet Neural Network *" J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019.

[4] Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 905–910, 2007.

[5] B. K. Baniya, D. Ghimire, and J. Lee, "Automatic music genre classification using timbral texture and rhythmic content features," *Int. Conf. Adv. Commun. Technol. ICACT*, vol. 2015-August, no. 3, pp. 434–443, 2015.

[6] S. Zahid, F. Hussain, M. Rashid, M. H. Yousaf, and H. A. Habib, "Optimized Audio Classification and Segmentation Algorithm by Using Ensemble Methods," *Math. Probl. Eng.*, vol. 2015, 2015.