

Pengenalan Bunyi Nada Untuk Mendeteksi Keretakan Badan Gitar Klasik Menggunakan Algoritma Fast Fourier Transform Dan Support Vector Mechine

LISA KRISTIANA ¹, FREDERIKUS L KLAU²

^{1, 2}Program Studi Informatika Institut Teknologi Nasional Bandung, Indonesia

Email : lisa@itenas.ac.id ², fredyklau29@gmail.com

Received DD MM YYYY | Revised DD MM YYYY | Accepted DD MM YYYY

ABSTRAK

Gitar merupakan salah satu alat musik berdawai yang dapat dimainkan dengan cara memetik menggunakan jari. Alat musik gitar digolongkan lagi ke dalam beberapa kategori, diantaranya gitar akustik, gitar flamenco dan gitar klasik. Gitar klasik merupakan salah satu jenis gitar yang senarnya berbahan nilon dan sering digunakan untuk permainan karya tunggal. Permainan gitar klasik dapat menghasilkan keluaran berupa bunyi yang beraturan atau yang seringkali dikenal dengan sebutan nada. Dalam penggunaannya, gitar klasik ini sering mengalami keretakan terutama di bagian badan gitar. Hal ini dikarenakan bahannya yang sangat tipis dan mudah retak atau lubang jika terkena benturan. Dari masalah sederhana ini, akan dibuat suatu sistem komputasi untuk membantu mendeteksi keretakan gitar tersebut. Pada Penelitian ini, digunakan metode Fast Fourier Transform untuk merubah nilai sinyal audio gitar dari digital ke analog, mengurangi noise dan menambah kualitas audio, serta metode Support Vector Mechine untuk menemukan hyperplane terbaik yang memisahkan dua buah class pada input space. Sehingga setelah dilakukan pengujian, penggunaan kernel RBF dari SVM dengan $\gamma=1$ dan $\text{cost}=1000$ menghasilkan accuracy sebesar 85%, precision 94%, recall 82% dan f1-Score 56%. Dari hasil ini, perhitungan FFT dan SVM dalam hal ini penggunaan kernel RBF mencapai hasil yang baik.

Kata kunci: Fats Fourier Transform, Support Vector Mechine, Deteksi Keretakan Gitar

ABSTRACT

Guitar is one of the stringed musical instruments that can be played by strumming using fingers. Guitar instruments are further classified into several categories, including acoustic guitar, flamenco guitar and classical guitar. Classical guitar is one type of guitar whose strings are made of nylon and often used for single work games. Classical guitar playing can produce an output in the form of irregular sounds or often known as tones. In its use, this classical guitar often experiences a rift especially in the body part of the guitar. This is because the material is very thin and easy to crack or hole if hit. From this simple problem, a computing system will be created to help detect the guitar cracks. In this study, the Fast Fourier Transform method was used to change the value of guitar audio signals from digital to analog, reduce noise and improve audio quality, as well as the Mechine Support Vector method to find the best hyperplane that separates two classes in input space. So after testing, the use of

RBF kernel from SVM with $\gamma = 1$ and cost = 1000 resulted in accuracy of 85%, precision 94%, recall 82% and f1-Score 56%. From these results, fast fourier transform and Support Vector Machine calculations in this case the use of the Radial Basis Function kernel achieved good results

Keywords: *Fats Fourier Transform, Support Vector Mechine, Deteksi Keretakan Gitar.*

1. PENDAHULUAN

1.1. Latar Belakang

Keretakan gitar klasik sering terjadi dikarenakan adanya benturan saat gitar tersebut digunakan. Hal ini didukung juga oleh pendapat yang mengatakan bahwa bahan dari gitar klasik tersebut sangatlah tipis dan mudah retak saat terjadi benturan. Keretakan ini biasa diatasi dengan cara reparasi oleh tukang reparasi ataupun membeli baru. Akibat dari keretakan pada badan gitar ini adalah suara yang kemudian dihasilkan akan terdengar berubah.

Begitu banyak solusi yang ditawarkan dalam menyelesaikan masalah dalam mengetahui keretakan gitar tersebut. Salah satu caranya adalah dengan melibatkan sistem komputasi. Yang menjadi parameter penelitian di sini adalah bunyi nada yang di dalamnya ada nilai gelombang frekuensi yang dapat diolah secara digital.[2]

[12] Oleh karena itu, dengan adanya sistem komputasi ini, dapat membantu dalam menentukan kelas keretakan pada gitar klasik. Pada peneltian ini, implementasinya ada pada dua metode yang berbeda. Di mana metode pertama adalah metode Fast Fourier Transform yang penyelesaian masalah dilakukan dengan cara menerjemhkan audio sinyal analog menjadi sinyal digital untuk menemukan kualitas audio yang lebih maksimal. Kemudian ada metode Support Vector Machine, di mana metode ini akan mengklasifikasi sebuah nilai dengan prinsip Structural Risk Minimization (SRM) yang bertujuan untuk menemukan hyperplane terbaik yang memisahkan dua buah class pada input space. Sehingga untuk mengetahui jenis keretakan gitar klask, dibuatlah sebuah sistem yang di dalamnya dilibatkan kedua metode tersebut.

1.2. Rumusan Masalah

Dari konsep yang ada pada latar belakang di atas, masalah yang muncul adalah bagaimana mengolah bunyi nada yang dihasilkan untuk mencari nilai ciri menggunakan algoritma fast fourier transform dan bagaimana mengklasifikasikan nilai ciri tersebut menggunakan metode Support Vector Mechine guna menentukan hasil retak atau tidak retaknya sebuah gitar.

1.3. Tujuan

Tujuan dari penelitian ini adalah untuk menerapkan metode fast fourier transform dan support vector mechine untuk mengetahui keretakan pada badan gitar klasik melalui bunyi nada.

1.4. Ruang Lingkup

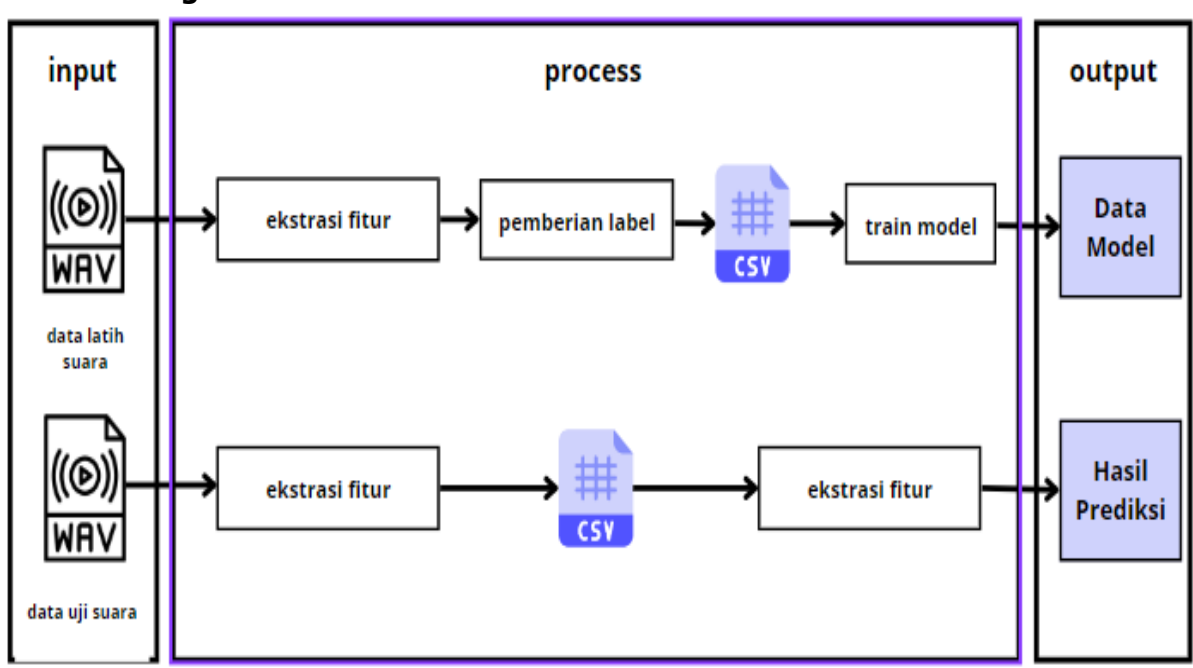
Ruang lingkup dari sistem yang dibangun ini berupa :

1. Pengolahan datanya hanya sebatas pada satu jenis alat musik saja yakni gitar klasik.
2. Inputan berupa suara dan format suara yang dikenal adalah wav.

3. Frekuensi sampling standar dengan durasi petikan senar selama 44100 Hz
4. Petikan yang dikenali berupa petikan senar 1 (Nada E), senar 2 (Nada B), senar 3 (Nada G), senar 4 (Nada D), senar 5 (Nada A), senar 6 (Nada E).
5. Aplikasi ini dibangun dengan dataset private yang jumlahnya masih sangat sedikit.

2. METODOLOGI PENELITIAN

2.1. Blok Diagram



Gambar 1. Blok Diagram

Gambar diatas merupakan rangkaian secara umum dari diagram blok sistem. Adapun rangkaian-rangkaianannya sebagai berikut :

Data Latih

- Data latih nada gitar adalah masukan nada berupa file .wav. Data tersebut direkam menggunakan software cool edit pro 2 pada ruang kedap suara untuk menghindari noise yang tidak dibutuhkan.
- Ekstraksi fitur merupakan proses pengenalan nada gitar yang masuk berupa audio, kemudian akan dikenali nilai cirinya agar nantinya dapat digunakan untuk proses klasifikasi.
- Pemberian Label adalah proses memberi label pada nilai ciri yang telah dikenali saat ekstraksi fitur.
- Simpan nilai ke dalam database merupakan hasil nilai dari perhitungan masing-masing ciri dari audio dan sudah diberi label pada ciri-ciri tersebut.

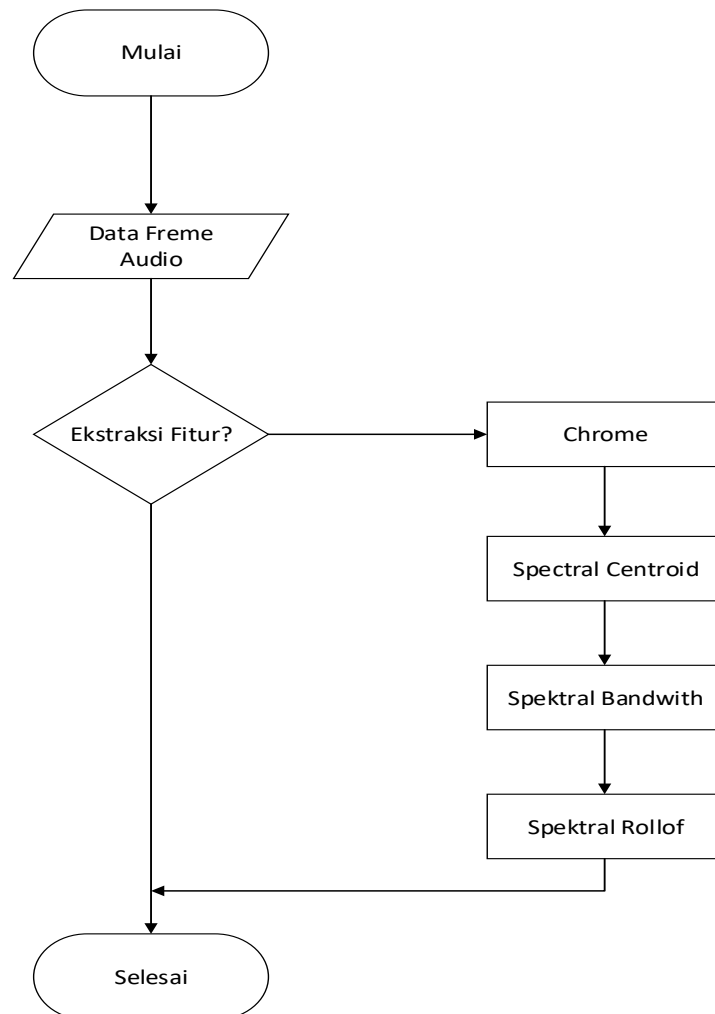
Data Uji

- Nada inputan data uji adalah masukan nada berupa file .wav. Data tersebut direkam menggunakan software cool edit pro 2 pada ruang kedap suara untuk menghindari noise yang tidak dibutuhkan.
- Ekstraksi Fitur merupakan proses nada gitar yang masuk berupa audio, kemudian akan dikenal cirinya agar dapat digunakan untuk proses identifikasi.

- Klasifikasi merupakan proses mengidentifikasi suatu inputan untuk menentukan inputan tersebut berada pada suatu kelas.
- Hasil prediksi merupakan hasil identifikasi kelas dari data uji yang diinputkan.

2.2. Flowchart Ektrasi Fitur

Flowchart Dempster-Shafer digunakan pada tahapan *Retrieve Case-based Reasoning* untuk menghitung bobot dari setiap gejala yang telah dipilih oleh pengguna sehingga bisa mengetahui nilai probabilitasnyanya.

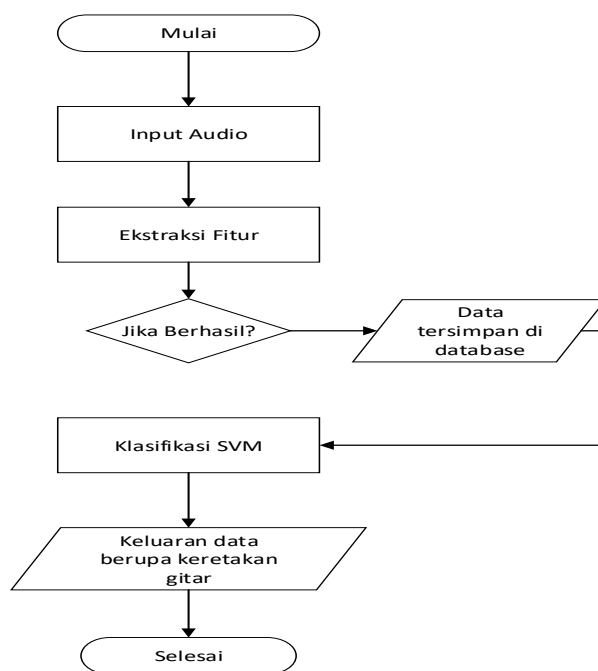


Gambar 2. Ekstrasi Fitur

2.4 Flowchart Sistem

Input Audio berupa rekaman suara gitar yang berektensi WAV* dan dilakukan secara banyak kali sesuai dengan kebutuhan. Convert audio dengan menggunakan librosa di mana audio tersebut akan diinisialisasi Hasil dari inisialisasi data tersebut akan disimpan sebagai dataset. Klasifikasi Dengan SVM di mana jarak data akan dihitung, pengambilan K-labeled dan pengklasifikasian.

Pengenalan Bunyi Nada Untuk Mendeteksi Keretakan Badan Gitar Klasik Menggunakan Algoritma Fast Fourier Transform Dan Support Vector Mechine



Gambar 3. Flowchart Sistem

3. HASIL DAN PEMBAHASAN

Data latih pada penelitian tersebut adalah data latih yang berasal dari rekaman audio gitar klasik dengan jumlah data adalah sebagai berikut:

Tabel 3 Data Latih

Senar Gitar	Jenis Gitar	Data Latih
Senar 1	Gitar Retak	11
Senar 1	Gitar Tidak retak	11
Senar 2	Gitar Retak	11
Senar 2	Gitar Tidak Retak	11
Senar 3	Gitar Retak	11
Senar 3	Gitar Tidak Retak	11
Senar 4	Gitar Retak	11
Senar 4	Gitar Tidak Retak	11
Senar 5	Gitar Rusak	11
Senar 5	Gitar Tidak Rusak	11
Senar 6	Gitar Rusak	11
Senar 6	Gitar Tidak Rusak	11

Jumlah keseluruhan data adalah 132 data. Data tersebut diambil menggunakan rekaman dari aplikasi audobe audition dengan frekuensi sampling 44100 Hz, 16 bits, channel mono, dan petikan senar sebanyak 4 kali dengan durasi waktu untuk setiap petikan adalah 4 detik dan disimpan dengan format wav. Dari data ini dilakukan proses ekstraksi fitur untuk mencari nilai dari fitur-fitur yang akan digunakan dalam penelitian ini.

3.1 DATA UJI

Data Uji didapatkan dari proses rekaman juga yang dilakukan dengan metode audobe audition dengan frekuensi sampling 44100 Hz, 16 bits, channel mono, dan petikan senar sebanyak 4 kali dengan durasi waktu untuk setiap petikan adalah 4 detik dan disimpan dengan format wav. Jumlah data uji pada sistem ini dapat dilihat pada gambar berikut:

Tabel 4 Data Uji

Senar Gitar	Jenis Gitar	Data Uji
Senar 1	Gitar Retak	5
Senar 1	Gitar Tidak retak	5
Senar 2	Gitar Retak	5
Senar 2	Gitar Tidak Retak	5
Senar 3	Gitar Retak	5
Senar 3	Gitar Tidak Retak	5
Senar 4	Gitar Retak	5
Senar 4	Gitar Tidak Retak	5
Senar 5	Gitar Rusak	5
Senar 5	Gitar Tidak Rusak	5
Senar 6	Gitar Rusak	5
Senar 6	Gitar Tidak Rusak	5

3.2 PENGUJIAN SISTEM

Pada tahap ini dilakukan pengujian terhadap sistem yang sudah dibangun, pengujian yang digunakan oleh pengembang atau disebut pengujian *alpha* yang bertujuan untuk mengetahui kinerja sistem identifikasi komentar negatif berbahasa Indonesia di sosial media dengan mendapatkan nilai *accuracy*, *precision*, *recall*, *f1-score*. Pada Tabel 5 disajikan daftar pengujian yang dilakukan.

Tabel 5 Tabel Daftar Pengujian

No Uji	Skenario Pengujian	Hasil yang Diharapkan
Uji 1	Melakukan proses ekstraksi fitur guna mendapatkan dataset	Menampilkan dataset pada h
Uji 2	Melakukan proses <i>traing data</i>	Menampilkan hasil <i>traing</i> berupa metode SVM dengan kernel linear, kernel rbf, matriks confusion, dan grafik perbandingan kernel
Uji 3	Melakukan proses prediksi data	Menampilkan hasil prediksi data

3.2.1 PENGUJIAN EKSTRAKSI FITUR

Proses pengujian untuk fungsionalitas Ekstraksi Fitur pada data dapat dijelaskan pada Tabel 6.

Tabel 6 Pengujian Ekstraksi Fitur

Identifikasi			
No uji	Uji-01		
Nama Butir Uji	Proses Ekstraksi Fitur		
Tujuan	Menampilkan dataset berupa data dengan format CSV.		
Deskripsi	Sistem akan melakukan perhitungan terhadap data latih.		
Kondisi Awal	Audio dipanggil secara otomatis oleh sistem.		
Skenario Uji			
Sistem mengambil fitur tertentu dari karakter yang diamati.			
Sistem mengenali karakter yang diamati dan menampilkan nilai.			
Kriteria Evaluasi Uji			
Sistem menampilkan hasil ekstraksi fitur berupa dataset.			
Kasus dan Hasil Uji			
Input	Harapan	Pengamatan	Kesimpulan
File audio berformat WAV.	Sistem menampilkan hasil ekstraksi berupa nilai dataset	Sistem menampilkan hasil nilai dataset	Hasil sesuai harapan
Source Code			
<pre><h3>Klik Extraction untuk membuat dataset </h3> <form action = "/"feature" method = "POST"> @app.route('/feature', methods = ['GET', 'POST']) def feature(): if request.method == 'GET': return render_template('feature.html') if request.method == 'POST': df_tidakrusak = pd.read_csv('tidakrusak.csv') df_tidakrusak['label'] = 1 df_rusak = pd.read_csv('rusak.csv') df_rusak['label'] = 0 df = pd.concat([df_tidakrusak, df_rusak]).reset_index(drop=True) return render_template('feature.html', df=df.to_html(), titles=[''])</pre>			
Output			

	chroma_stft	spectral_centroids	spectral_bandwidth	spectral_rolloff	zero_crossing_rate	label
0	0.246400	1808.749303	1768.196743	3798.810513	0.044992	1
1	0.284875	1569.788739	1510.209272	3053.631562	0.050250	1
2	0.290948	1824.880293	1757.185517	3817.498688	0.043099	1
3	0.246645	1217.295412	1376.081418	2510.158682	0.028000	1
4	0.361266	1004.611188	1316.892712	2167.998251	0.020350	1
5	0.209732	1173.013406	1384.925387	2517.023877	0.023718	1
6	0.246400	1808.749303	1768.196743	3798.810513	0.044992	1
7	0.290948	1824.880293	1757.185517	3817.498688	0.043099	1
8	0.200140	800.860036	1031.298435	1235.741149	0.021346	1
9	0.344147	1408.519909	1627.228891	3063.085478	0.035124	1
10	0.427510	1459.040125	1632.407636	3004.706032	0.037480	1
11	0.443356	2281.626125	1961.378930	4595.487286	0.068107	1
12	0.283842	2056.187907	1877.920845	4280.296280	0.057227	1
13	0.290948	1824.880293	1757.185517	3817.498688	0.043099	1
14	0.200140	800.860036	1031.298435	1235.741149	0.021346	1
15	0.361266	1004.611188	1316.892712	2167.998251	0.020350	1
16	0.209732	1173.013406	1384.925387	2517.023877	0.023718	1
17	0.246400	1808.749303	1768.196743	3798.810513	0.044992	1
18	0.283842	2056.187907	1877.920845	4280.296280	0.057227	1
19	0.246400	1808.749303	1768.196743	3798.810513	0.044992	1

3.2.2 PENGUJIAN TRAIN DATA

Proses pengujian untuk fungsionalitas *text preprocessing stemming* dijelaskan pada Tabel 7.

Tabel 7 Pengujian Train Data

Identifikasi			
No uji	Uji-02		
Nama Butir Uji	Pengujian <i>Training</i> Data		
Tujuan	Menampilkan hasil <i>traing</i> berupa menemukan hasil kerja dari metode svm yang menguji data menggunakan kernel linear dan RBF.		
Deskripsi	Sistem akan melakukan proses <i>traing</i> pada dataset.		
Kondisi Awal	Hasil training data berupa nilai precision,recall,f1-score, support dan accuracy.		
Skenario Uji			
Sistem melanjutkan proses dari hasil ekstraksi fitur. Sistem menemukan hasil <i>train</i> data.			
Kriteria Evaluasi Uji			
Sistem menampilkan hasil <i>training</i> berupa menemukan nilai dari kernel linear dan RBF.			
Kasus dan Hasil Uji			
Input	Harapan	Pengamatan	Kesimpulan
dataset format <i>csv</i>	Sistem menampilkan hasil <i>training</i>	Sistem menampilkan hasil <i>training</i>	Hasil sesuai harapan
Source Code			
Klik Train untuk melakukan Training Data dari data Feature Extraction sebelumnya			


```
<form action = "/train" method = "POST">

@app.route('/train', methods = ['GET', 'POST'])
def train():
    if request.method == 'GET':
        return render_template('train.html')

    if request.method == 'POST':
        df_tidakrusak = pd.read_csv('tidakrusak.csv')
        df_tidakrusak['label'] = 1

        df_rusak = pd.read_csv('rusak.csv')
        df_rusak['label'] = 0

        df = pd.concat([df_tidakrusak, df_rusak]).reset_index(drop=True)

        scaler = StandardScaler()

        X = df.drop(columns=['label'])
        X_std = scaler.fit_transform(X)
        y = df['label']

        X_train, X_test, y_train, y_test = train_test_split(X_std, y.values, test_size=0.30, random_state=42)

        # model = GridSearchCV(SVC(), param_grid, refit = True, verbose = 3)
        modell = SVC(C=1000, gamma=1, kernel='rbf')
        modell.fit(X_train, y_train)
        print('akurasi model adalah: ')
        print(model.score(X_test, y_test))
        rbf = modell.score(X_test, y_test)
        predict1 = modell.predict(X_test)
        cl1 = classification_report(y_test, predict1, output_dict=True)
        cm1 = plot_confusion_matrix(modell, X_test, y_test)
        plt.savefig('static/img/cm1.png')
        df1 = pd.DataFrame(cl1).transpose()

        model2 = SVC(C=1000, gamma=1, kernel='linear')
        model2.fit(X_train, y_train)
        print('akurasi model adalah: ')
        print(model2.score(X_test, y_test))
        linear = model2.score(X_test, y_test)
        predict2 = model2.predict(X_test)
        cl2 = classification_report(y_test, predict2, output_dict=True)
        cm2 = plot_confusion_matrix(model2, X_test, y_test)
        plt.savefig('static/img/cm2.png')
        df2 = pd.DataFrame(cl2).transpose()
```

Output																																																													
Kernel Linear	Kernel RBF																																																												
Kernel Linear Akurasi untuk kernel linear adalah 0.55 Classification Report <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.480000</td><td>0.705882</td><td>0.571429</td><td>17.00</td></tr><tr><td>1</td><td>0.666667</td><td>0.434783</td><td>0.526316</td><td>23.00</td></tr><tr><td>accuracy</td><td>0.550000</td><td>0.550000</td><td>0.550000</td><td>0.55</td></tr><tr><td>macro avg</td><td>0.573333</td><td>0.570333</td><td>0.548872</td><td>40.00</td></tr><tr><td>weighted avg</td><td>0.587333</td><td>0.550000</td><td>0.545489</td><td>40.00</td></tr></table>		precision	recall	f1-score	support	0	0.480000	0.705882	0.571429	17.00	1	0.666667	0.434783	0.526316	23.00	accuracy	0.550000	0.550000	0.550000	0.55	macro avg	0.573333	0.570333	0.548872	40.00	weighted avg	0.587333	0.550000	0.545489	40.00	Kernel RBF Akurasi untuk kernel rbf adalah 0.85 Classification Report <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.941176</td><td>0.842105</td><td>0.891056</td><td>17.00</td></tr><tr><td>1</td><td>0.782609</td><td>0.857143</td><td>0.819150</td><td>23.00</td></tr><tr><td>accuracy</td><td>0.850000</td><td>0.850000</td><td>0.850000</td><td>0.85</td></tr><tr><td>macro avg</td><td>0.861893</td><td>0.849624</td><td>0.855103</td><td>40.00</td></tr><tr><td>weighted avg</td><td>0.850000</td><td>0.850752</td><td>0.850375</td><td>40.00</td></tr></table>		precision	recall	f1-score	support	0	0.941176	0.842105	0.891056	17.00	1	0.782609	0.857143	0.819150	23.00	accuracy	0.850000	0.850000	0.850000	0.85	macro avg	0.861893	0.849624	0.855103	40.00	weighted avg	0.850000	0.850752	0.850375	40.00
	precision	recall	f1-score	support																																																									
0	0.480000	0.705882	0.571429	17.00																																																									
1	0.666667	0.434783	0.526316	23.00																																																									
accuracy	0.550000	0.550000	0.550000	0.55																																																									
macro avg	0.573333	0.570333	0.548872	40.00																																																									
weighted avg	0.587333	0.550000	0.545489	40.00																																																									
	precision	recall	f1-score	support																																																									
0	0.941176	0.842105	0.891056	17.00																																																									
1	0.782609	0.857143	0.819150	23.00																																																									
accuracy	0.850000	0.850000	0.850000	0.85																																																									
macro avg	0.861893	0.849624	0.855103	40.00																																																									
weighted avg	0.850000	0.850752	0.850375	40.00																																																									

3.2.3 PENGUJIAN PREDIKSI

Proses pengujian untuk fungsionalitas *text preprocessing case folding* dijelaskan pada Tabel 8.

Tabel 8 Pengujian Prediksi

Identifikasi			
No uji	Uji-03		
Nama Butir Uji	Pengujian Prediksi data		
Tujuan	Menampilkan hasil prediksi FFT, SVM dan berupa informasi gitar retak atau tidak retak		
Deskripsi	Sistem akan melakukan proses prediksi dengan menghitung nilai data inputan dengan model yang telah dibuat.		
Kondisi Awal	Audio inputan dengan format WAV		
Skenario Uji			
Sistem melakukan proses input audio uji.			
Sistem mengekstraksi audio dan klasifikasi audio sebagai proses dari prediksi			
Kriteria Evaluasi Uji			
Sistem menampilkan hasil <i>FFT</i> dan serta hasil prediksi			
Kasus dan Hasil Uji			
Input	Harapan	Pengamatan	Kesimpulan
Audio Inputan dengan format wav	Sistem menampilkan hasil <i>FFT</i> dan SVM	Sistem menampilkan hasil <i>FFT</i> dan SVM	Hasil sesuai harapan
Source Code			
<pre><form action = "/predict" method = "POST" enctype = "multipart/form-data"> <input type="file" name="file" id="upload"/> <audio id="audio" controls> <source src="" id="src" /> </audio> @app.route('/predict', methods = ['GET', 'POST']) def predict(): if request.method == 'GET':</pre>			

```
        return render_template('predict.html')

    if request.method == 'POST':

        global namafile

        f = request.files['file']

        f.save(os.path.join(uploads_dir,
secure_filename(f.filename)))

        # return "<h2>Successfully uploaded</h2>"

        namafile = 'static/'+f.filename
        print(namafile)

        audio_test = glob(namafile)
        print(audio_test)
def get_feature_vector(y,sr):
    feat_vect_i = [ np.mean(func(y,sr)) for func
in fn_list_i]
    feat_vect_ii = [ np.mean(func(y)) for func in
fn_list_ii]

    feature_vector = feat_vect_i + feat_vect_ii
    return feature_vector

#build the matrix with normal audios featurized
audios = []
for file in audio_test:

    y,sr = librosa.load(file,sr=None)
    feature_vector = get_feature_vector(y, sr)
    audios.append(feature_vector)
    print('.', end= " ")

print(audios)

data = pd.DataFrame(audios,
columns=['chroma_stft',
'spectral_centroid',
'spectral_bandwidth',
'zero_crossing_rate'])

hasil_prediksi = model.predict(audios)

if hasil_prediksi == 0:
    print('Prediksi Retak')
    hasil = 'Prediksi Retak'
else:
    print('Prediksi Tidak Retak')
    hasil = 'Prediksi Tidak Retak'
plt.figure(figsize=(14, 5))
path = namafile
out,samples = librosa.load(path)
```

```

print(out.shape, samples)
librosa.display.waveplot(out, sr=samples)
# plt.show()
plt.savefig('static/img/plot1.png')

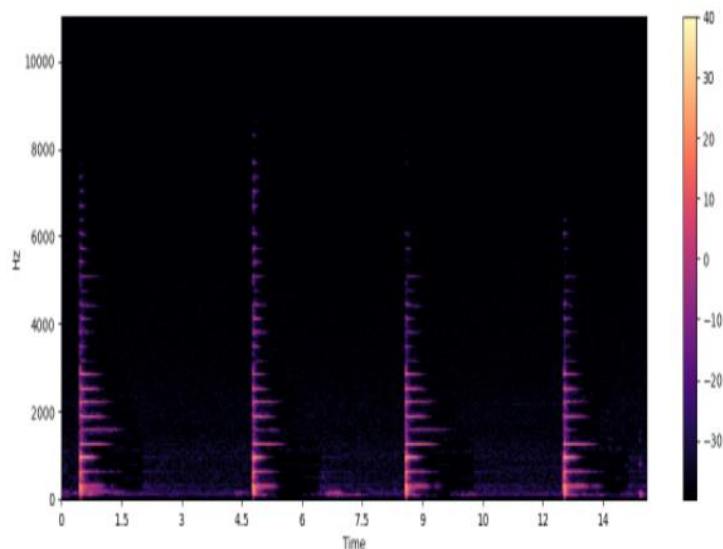
stft_array = librosa.stft(out)
stft_array_db = librosa.amplitude_to_db(abs(stft_array))

librosa.display.specshow(stft_array_db, sr=samples, x_axis='
time', y_axis='hz')
plt.colorbar()
# plt.show()
plt.savefig('static/img/plot2.png')

```

Output

FFT



SVM

chroma_stft	spectral_centroids	spectral_bandwidth	spectral_rolloff	zero_crossing_rate
00.443356	2281.626125	1961.37893	4595.487286	0.068107

Prediksi Tidak Retak

Dari Pengujian data uji sebanyak 5 untuk gitar rusak dan 5 untuk gitar tidak rusak, dapat dilihat pada tabel berikut untuk hasil pengujiannya

Tabel 9 Pengujian Akutasi

Senar Gitar	Jenis Gitar	Jumlah pengujian	Pengujian benar	Presentase
Senar 1	Retak	10	8	80 %
Senar 1	Tidak Retak	10	7	70 %

Senar 2	Retak	10	7	70 %
Senar 2	Tidak Retak	10	6	60 %
Senar 3	Retak	10	8	80 %
Senar 3	Tidak Retak	10	6	60 %
Senar 4	Retak	10	8	80 %
Senar 4	Tidak Retak	10	7	70 %
Senar 5	Retak	10	7	70%
Senar 5	Tidak Retak	10	8	80 %
Senar 6	Retak	10	6	60%
Senar 6	Tidak Retak	10	8	80%

4. KESIMPULAN

Pada penelitian ini telah diimplementasikan metode *Fast fourier transform* (FFT) dan SVM dalam hal ini kernel RBF untuk mendeteksi keretakan gitar berdasarkan bunyi nada gitar. Penggunaan FFT dapat membantu dalam proses pengenalan fitur dan menghasilkan dataset dari sebuah audio sehingga dapat digunakan proses identifikasi selanjutnya. Sedangkan penggunaan SVM ada pada proses klasifikasi data setelah diproses oleh metode FFT. Setelah dilakukan pengujian, penggunaan kernel RBF dari SVM dengan $\gamma=1$ dan $\text{cost}=1000$ menghasilkan *accuracy* sebesar 85%, *precision* 94%, *recall* 82% dan *f1-Score* 56%. Adapun kernel tersebut dibandingkan dengan perhitungan tanpa kernel(linear) dan menghasilkan *accuracy* sebesar 58 %, *precision* 82%, *recall* 59% dan *f1-Score* 35%. Dari hasil ini, perhitungan FFT dan SVM dalam hal ini penggunaan kernel RBF mencapai hasil yang baik.

5. SARAN

Dalam pengembangan sistem ke depannya untuk meningkatkan kinerja dan akurasi dari sistem identifikasi komentar negatif berbahasa Indonesia di sosial media terdapat beberapa saran sebagai berikut:

- a. Dapat mencoba metode klasifikasi lain untuk membandingkan hasil kinerja mana yang lebih baik.
- b. Data diklasifikasikan menjadi tiga kelas, yaitu positif, negatif, dan netral. Hasil dari klasifikasi tersebut dapat diketahui polaritas dari setiap kelasnya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Prodi Informatika Institut Teknologi Nasional Bandung. Penulis juga berterima kasih kepada semua pihak yang telah mendukung sehingga penelitian ini terlaksana dengan baik.

DAFTAR RUJUKAN

- [1]Prasetio, R., Hidayatno, A., & Santoso, I. (2014). Perancangan Sistem Perbaikan Nada Suara Manusia Dengan Menggunakan Metode Phase Vocoder Terhadap Nada Referensi

Musik. *Transmisi*, 16(4), 160-166–166. <https://doi.org/10.12777/transmisi.16.4.160-166>

- [2] Abdillah, F. N. (2017). IMPLEMENTASI ALGORITMA FAST FOURIER TRANSFORM (FFT) DAN ALGORITMA HARMONIC PRODUCT SPECTRUM (HPS) Pada TUNER GITAR BERBASIS ANDROID Firdaus Noor Abdillah Fakultas Ilmu Komputer Universitas Kuningan Jalan Tjut Nyak Dhien Cijoho Kuningan Telepon (0232. *Jurnal Nuansa Informatika*, 11(2), 18–25. <https://www.journal.uniku.ac.id/index.php/ilkom/article/viewFile/1326/982>
- [3] Abdillah, S. (2011). Penerapan Algoritma Decision Tree C4.5 Untuk Diagnosa Penyakit Stroke Dengan Klasifikasi Data Mining Pada Rumah Sakit Santra Maria Pemalang. *Jurnal Teknik Informatika*, 1–12.
- [4] Fetra, N., & Irsyad, M. (2015). Aplikasi Pencarian Chord dalam Membantu Penciptaan Lagu Menggunakan Algoritma Fast Fourier Transform (FFT) dan Metode Klasifikasi K-Nearest Neighbor (KNN). *Jurnal CoreIT*, 1(2 ISSN: 2460-738X), 30–36.
- [5] Gitar, A. (2015). *PEMBELAJARAN GITAR KLASIK DENGAN METODE ' PARTISIPATIVE LEARNING ' DI MULTIPLE INTELEGENCE SCHOOL MUSIC TALENTS ACADEMY SURABAYA Ian Taufan NIM: 072134219 Mahasiswa Sendratasik FBS Universitas Negeri Surabaya Agus Suwahyono S . Sn . M . Pd. 2*, 59–75.
- [6] Hadyansyah, I., Budiman, G., & Sunarya, U. (2013). *Applikasi identifikasi nada suara manusia menggunakan metode fft berbasis android*. 0–6.
- [7] Hanggarsari, P. N., Fitriawan, H., & Yuniati, Y. (2012). Simulasi Sistem Pengacakan Sinyal Suara Secara Realtime Berbasis Fast Fourier Transform (FFT). *ELECTRICAL Jurnal Rekayasa Dan Teknologi Elektro*, 3, 192–198.
- [8] Indrawan, A. (2019). *Mengenal Dunia Gitar Klasik*. 1–10.
- [9] Irtawaty, A. S. (2019). Implementasi Metode Fast Fourier Transform (Fft) Dalam Mengklasifikasikan Suara Pria Dan Wanita Di Laboratorium Jurusan Teknik Elektro Politeknik Negeri Balikpapan. *JTT (Jurnal Teknologi Terpadu)*, 7(2), 70–75. <https://doi.org/10.32487/jtt.v7i2.661>
- [10] Lagu, D. A. N., Pada, F., Bunyi, M., Waluyo, K. A., Noviandini, D., Sudjito, D. N., Fisika, J. P., Sains, F., Kristen, U., & Wacana, S. (2016). "Konser Fisika": Pembelajaran Fisika Dengan Mengintegrasikan Seni Musik Menggunakan Gitar Akustik, Zelscope, Dan Lagu Fisika Pada Materi Bunyi. *UPEJ Unnes Physics Education Journal*, 5(1), 1–11. <https://doi.org/10.15294/upej.v5i1.12694>
- [11] Muludi, K., Frank, A., & Loupatty, S. F. B. (2014). Chord Identification Using Pitch Class Profile Method with Fast Fourier Transform Feature Extraction. *IJCSI International Journal of Computer Science Issues*, 11(3), 139–144.
- [12] Parapat, I. M., Furqon, M. T., & Sutrisno. (2018). Penerapan Metode Support Vector Machine (SVM) Pada Klasifikasi Penyimpangan Tumbuh Kembang Anak. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(10), 3163–3169. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/2577>
- [13] Perdana, R. R., Soelaiman, R., & Fatichah, C. (2017). Implementasi Ekstraksi Fitur untuk Pengelompokan Berkas Musik Berdasarkan Kemiripan Karakteristik Suara. *Jurnal Teknik ITS*, 6(1). <https://doi.org/10.12962/j23373539.v6i1.22138>
- [14] Ritonga, A. S., & Purwaningsih, E. S. (2018). Penerapan Metode Support Vector Machine (SVM) Dalam Klasifikasi Kualitas Pengelasan Smaw (Shield Metal Arc Welding). *Ilmiah Edutic*, 5(1), 17–25.
- [15] Riyanto, S., Purwanto, A., & Supardi. (2009). Algoritma Fast Fourier Transform (FFT) Decimation In Time (DIT) dengan Resolusi 1/10 Hertz. *Seminar Nasional Penelitian, Pendidikan, Dan Penerapan MIPA*, 223–231.
- [16] Sasmito, G. W. (2017). Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*,

2(1), 6–12.

- [17] Sipasulta, R. Y., St, A. S. M. L., & Sompie, S. R. U. A. (2014). Simulasi Sistem Pengacak Sinyal Dengan Metode Fft (Fast Fourier Transform). *E-Journal Teknik Elektro Dan Komputer*, 3(2), 1–9.
- [18] Sujadi, H., Sopiandi, I., & Mutaqin, A. (2017). Sistem Pengolahan Suara Menggunakan Algoritma FFT (Fast Fourier Transform). *Sintak*, 101–107.
- [19] Syaifuddin, A. (2014). *FAST FOURIER TRANSFORM (FFT) UNTUK ANALISIS SINYAL SUARA DOPPLER ULTRASONIK*. 3(3), 181–188.
- [20] Tobing, F., & Agung, H. (2020). *PENERAPAN ALGORITMA FAST FOURIER TRANSFORM (FFT) PADA PROTOTYPE DOOR LOCK SYSTEM BERBASIS VOICE RECOGNITION The Application of the Fast Fourier Transform (FFT) Algorithm to the Voice Recognition Based Door Lock Prototype System*. 1, 248–254.