

Deteksi Pulpitis Menggunakan MFCC Dan CNN1D Dalam Lingkup Penggunaan Flask Sebagai Backend

1st Nanda Putri Hermina

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

nandaputri@student.telkomuniversity.ac.id

2nd Sofia Saidah

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

sofiasaidahsfi@telkomuniversity.ac.id

3rd Bambang Hidayat

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

bhidayat@telkomuniversity.ac.id

Abstrak— Pulpitis adalah peradangan pada jaringan pulpa gigi yang dapat disebabkan oleh berbagai faktor seperti infeksi bakteri, trauma pada gigi, atau kerusakan gigi. Sakit gigi bisa sangat mengganggu aktivitas seseorang. Ketika seseorang mengalami sakit gigi mereka mungkin sulit untuk berkonsentrasi, berbicara atau bahkan makan dengan nyaman. Sebelum terjadi kerusakan gigi yang lebih parah maka kami membuat alat yang dapat mendeteksi pulpitis dengan biaya yang terjangkau dan *realtime* yaitu deteksi pulpitis menggunakan sinyal suara dengan algoritma *machine learning* dengan ekstraksi MFCC dan CNN 1D. Model ini dapat mendeteksi gigi yang sehat maupun gigi yang mengalami pulpitis dengan akurasi 92%.

Kata kunci—CNN, MFCC, backend, website, pulpitis, machine learning.

I. PENDAHULUAN

Kesehatan gigi dan mulut merupakan aspek penting dari tubuh manusia yang dapat meningkatkan kualitas hidup. Jika kesehatan gigi dan mulut tidak dijaga dengan baik, hal ini dapat menyebabkan rasa sakit dan ketidaknyamanan yang bisa mengganggu aktivitas sehari-hari, sehingga seseorang tidak dapat dianggap sepenuhnya sehat. [1].

Gigi adalah salah satu jaringan tubuh yang paling keras jika dibandingkan dengan jaringan lainnya. Strukturnya terdiri dari beberapa lapisan, mulai dari enamel yang kuat, dentin (tulang gigi) di bagian dalam, hingga pulpa yang mengandung pembuluh darah, saraf, serta bagian-bagian lain yang memperkuat gigi. Meskipun begitu, gigi mudah sekali mengalami kerusakan. Gigi berfungsi sebagai bagian dari alat pengunyahan dalam sistem pencernaan manusia. [2].

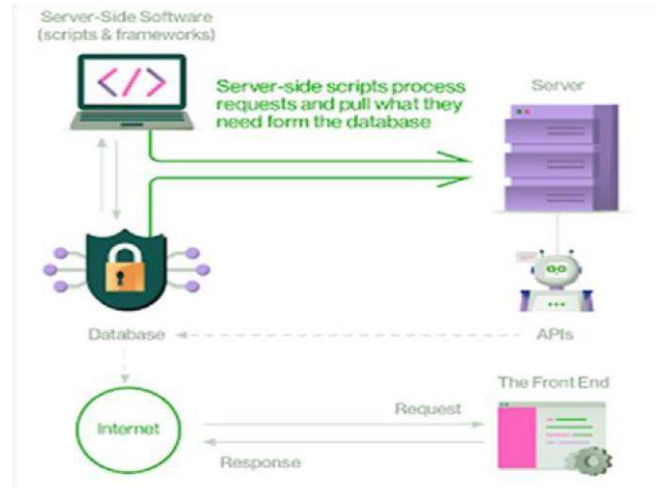
Pulpitis adalah proses peradangan pada pulpa saluran akar gigi. Mahkota gigi mengandung ruang pulpa yang memiliki rongga yang berisi pleksus pembuluh darah, jaringan ikat, dan ujung saraf. Semua jaringan ini bersama-sama membentuk pulpa, yang fungsi utamanya adalah untuk menyehatkan bagian koronal dan akar [3].

II. METODOLOGI

A. Backend Website

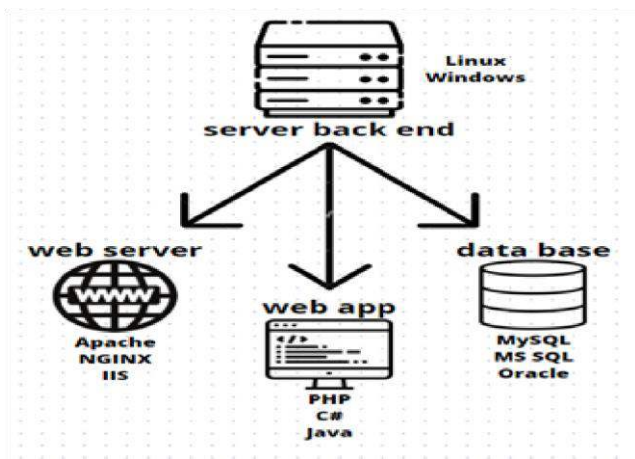
Backend Website adalah sumber untuk menjalankan website aplikasi. Bagian belakang situs web terdiri dari tiga yaitu *divisions-server*, *website application*, dan *database* [4]. Pengembangan *backend* dimulai dengan menulis kode yang

mendefinisikan fungsionalitas inti aplikasi. dari sana *backend development* akan menghost kode di *server*, mengelola operasi basis data, dan menangani API. Tiap *backend* aplikasi dapat terlihat sangat berbeda dari satu aplikasi ke aplikasi lainnya, baik itu menggunakan *server* berbasis *cloud* dan *database*.



GAMBAR 1.
Cara kerja website

Backend adalah proses di balik layar yang membuat situs web dan aplikasi bekerja. Selain itu *backend* juga menggerakkan basis data, *server* dan aplikasi pada dasarnya, semua yang terjadi di sisi *server*. *Backend developer* yang membangun basis data, mengintegrasikan API, dan menyiapkan teknologi sisi *server* yang menghidupkan situs web. Tanpa *backend development*, situs web tidak akan berfungsi [5].



Gambar 2.
Bagian dari backend

1. Server

Server backend terdiri dari perangkat keras dan sistem operasi di *backend* yang menghost semua aplikasi yang diperlukan untuk dijalankan pada aplikasi web. Ini adalah sistem yang sebenarnya menjalankan semua proses dan menangani semua tugas yang membentuk aplikasi web yang lengkap.

2. Website Application

Aplikasi web adalah perangkat lunak aplikasi yang tidak memerlukan instalasi dan dapat diakses dari *server* jarak jauh melalui *browser* web. Aplikasi web dibuat untuk interaksi, yang memungkinkan pengguna untuk mengirim dan menggunakan data antara *browser* dan *server* web. Interaksi ini bisa sesederhana masuk ke akun, atau serumit melakukan pembayaran dengan kartu kredit. Jika sebuah situs web dapat dikatakan ditentukan oleh kontennya, maka aplikasi web ditentukan oleh interaksinya dengan pengguna [6].

3. Database

Konsep *backend database* ditemukan oleh Microsoft pada tahun 1989. *Backend database* adalah *database* yang diakses oleh pengguna secara tidak langsung melalui aplikasi eksternal, bukan dengan pemrograman aplikasi yang disimpan di dalam *database* itu sendiri atau dengan manipulasi data tingkat rendah (misalnya melalui perintah SQL). *Backend database* menyimpan data tetapi tidak menyertakan elemen aplikasi pengguna akhir seperti kueri yang tersimpan, formulir, makro atau laporan.

B. Machine Learning

Machine Learning adalah pendekatan statistik untuk mempelajari dan membuat kesimpulan tentang data yang menggunakan berbagai algoritma yang cocok untuk menjawab berbagai jenis pertanyaan [7]. Selama beberapa dekade terakhir, kemajuan teknologi dalam penyimpanan dan kekuatan pemrosesan telah memungkinkan beberapa produk inovatif berdasarkan *machine learning*.

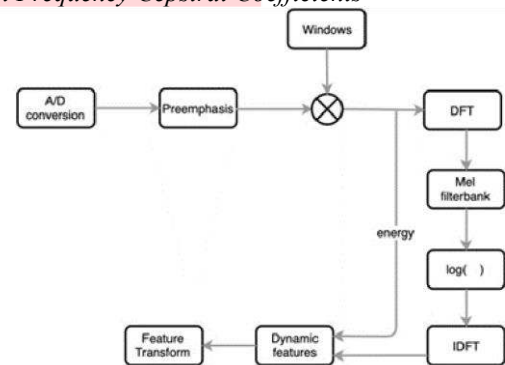
Machine learning berperan sebagai komponen penting dari bidang ilmu data yang terus berkembang. Melalui penggunaan metode statistik, algoritme dilatih untuk membuat klasifikasi atau prediksi, dan untuk mengungkap wawasan utama dalam proyek penggalian data. Wawasan ini kemudian mendorong pengambilan keputusan dalam aplikasi dan bisnis, yang idealnya berdampak pada metrik pertumbuhan utama. Algoritma pembelajaran mesin biasanya

dibuat menggunakan kerangka kerja yang mempercepat pengembangan solusi.

C. Flask

Flask web adalah kerangka kerja web mikro yang ditulis dalam bahasa python. Flask di klasifikasikan sebagai *microframework* karena tidak memerlukan alat atau *libraries* tertentu, tidak memiliki lapisan abstraksi *database*, *form validation*, atau komponen lain di mana pustaka pihak ketiga yang sudah ada sebelumnya menyediakan fungsi-fungsi umum. Namun, Flask mendukung ekstensi yang dapat menambahkan fitur aplikasi seolah-olah mereka diimplementasikan dalam Flask itu sendiri. Ekstensi tersedia untuk pemetaan objek relasional, *form validation*, penanganan unggahan, berbagai teknologi otentikasi, dan beberapa alat terkait *framework* [8].

D. Mel Frequency Cepstral Coefficients



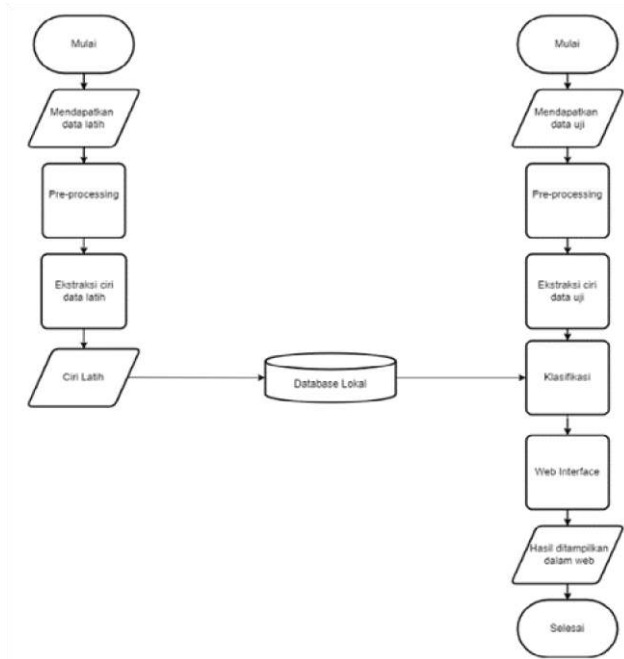
GAMBAR 3.
Proses MFCC

MFCC pertama kali diusulkan untuk pengenalan suara. MFCC adalah pendekatan berbasis *bank filter*, desain filter sedemikian rupa sehingga mirip dengan persepsi frekuensi pendengaran manusia. Para peneliti telah menunjukkan bahwa fitur *bank filter* yang dihitung secara langsung lebih kuat untuk mengenali ucapan dalam kondisi bising [9].

MFCC adalah sekumpulan koefisien yang menangkap bentuk spektrum daya sinyal suara. Koefisien-koefisien ini diperoleh dengan terlebih dahulu mengubah sinyal audio menjadi domain frekuensi menggunakan teknik seperti *Discrete Fourier Transform (DFT)*, dan kemudian menerapkan mel-scale untuk memperkirakan persepsi pendengaran manusia terhadap frekuensi suara. Terakhir, koefisien *cepstral* dihitung dari spektrum berskala mel.

III. DESAIN SISTEM

Flowchart ini menggambarkan secara visual proses dari awal hingga akhir, memastikan setiap tahap dilaksanakan secara sistematis dan efisien. Adapun langkah-langkah dalam *flowchart* tersebut meliputi pengumpulan data, *preprocessing*, ekstraksi ciri, hingga klasifikasi dan penampilan hasil pada antarmuka web. Setiap komponen dalam *flowchart* ini berperan penting dalam mencapai tujuan penelitian, yaitu menghasilkan hasil analisis yang akurat dan dapat diandalkan.



GAMBAR 4.
Diagram Alir Sistem

A. Mendapatkan Data Uji

Mendapatkan data uji adalah proses mengumpulkan atau menyiapkan kumpulan data yang akan digunakan untuk menguji dan mengevaluasi kinerja model atau sistem setelah model dilatih. Data uji ini harus berbeda dari data latih agar dapat memberikan penilaian yang objektif terhadap kemampuan model dalam mengeneralisasi pola dari data yang belum pernah dilihat sebelumnya.

B. Preprocessing

Preprocessing sinyal ucapan meliputi tahap-tahap berikut: proses input sinyal ucapan digital yang telah difilter, *slicing* sinyal ucapan dengan tumpang tindih *frame*, pemrosesan sinyal di jendela, konversi spektrum, dan normalisasi spektrum frekuensi [10].

C. Ekstraksi Ciri Data Uji

Ekstraksi ciri data uji adalah proses mengambil fitur atau atribut penting dari data uji yang akan digunakan oleh model untuk melakukan prediksi atau klasifikasi. Langkah ini bertujuan untuk menyederhanakan data dan mengisolasi informasi yang relevan sehingga model dapat bekerja secara lebih efektif dan efisien saat mengevaluasi data uji.

D. Klasifikasi

Klasifikasi adalah proses dalam *machine learning* di mana model memetakan data input ke dalam kategori atau kelas yang telah ditentukan sebelumnya. Berdasarkan ciri-ciri yang diekstraksi dari data, model menentukan label atau kategori yang paling sesuai untuk setiap data uji, memungkinkan pengambilan keputusan otomatis atau analisis berdasarkan kelompok yang diidentifikasi.

E. Web Interface

Web interface adalah antarmuka berbasis web yang memungkinkan pengguna berinteraksi dengan aplikasi atau sistem melalui *browser*. Ini menyediakan tampilan grafis dan kontrol yang memudahkan pengguna untuk mengakses, melihat, dan berinteraksi dengan data atau hasil yang dihasilkan oleh sistem secara intuitif dan mudah diakses dari berbagai perangkat.

F. Tampilan Hasil Deteksi

Tampilan hasil deteksi adalah presentasi visual atau laporan yang menunjukkan output dari proses deteksi apakah gigi tersebut pulpitis atau tidak.

IV. HASIL DAN PEMBAHASAN

A. Memuat Model Machine Learning

```

app = Flask(__name__)

model = load_model('voice_classification_models.h5')
UPLOAD_FOLDER = 'static/uploads'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

ALLOWED_EXTENSIONS = {'wav'}

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
  
```

GAMBAR 5.
Memuat model ML

Proses pertama pada subsistem *backend website* yaitu memuat model *machine learning* yang sudah dilatih sebelumnya. Proses ini juga menyiapkan folder sebagai tempat penyimpanan data yang akan dimuat ke dalam *website* dan proses ini ada tahap memeriksa file di tempat penyimpanan file yang boleh diproses yaitu dengan format wav.

B. Ekstraksi Fitur Pada Backend

```

def extract_features(signal, sr, max_len=400, apply_gaussian=True):
    mfccs = librosa.feature.mfcc(y=signal, sr=sr, n_mfcc=40)
    if apply_gaussian:
        mfccs = gaussian_filter1d(mfccs, sigma=1)

    if mfccs.shape[1] > max_len:
        mfccs = mfccs[:, :max_len]
    else:
        mfccs = np.pad(mfccs, ((0, 0), (0, max_len - mfccs.shape[1])), mode='constant')

    return mfccs
  
```

GAMBAR 6.
Ekstraksi fitur menggunakan MFCC

Proses kedua yaitu ekstraksi fitur untuk data yang akan masuk ke dalam *website* agar proses klasifikasi data seragam dengan saat proses latih data. Ekstraksi fitur pada tahap ini sama dengan ekstraksi fitur saat pelatihan data.

C. Konversi Audio Ke File Wav

```

def convert_to_wav(source_path, target_path):
    audio = AudioSegment.from_file(source_path)
    audio.export(target_path, format="wav")
  
```

GAMBAR 7.
Konversi audio ke file wav

Proses ketiga yaitu mengubah suara rekaman yang masuk menjadi wav dikarenakan model yang sudah dilatih menggunakan wav sehingga perlu ada keselarasan antara data latih dan data uji agar hasil deteksi bisa maksimal.

D. Memanggil Kembali Fungsi Yang Sudah Didefinisikan

```
@app.route('/upload_audio', methods=['POST'])
def upload_audio():
    if 'audio' not in request.files:
        print('No audio part found') # Debugging
        return jsonify(error='No audio part'), 400

    file = request.files['audio']
    if file.filename == '':
        print('No selected file') # Debugging
        return jsonify(error='No selected file'), 400

    if file and allowed_file(file.filename):
        filename = secure_filename(file.filename)
        filepath = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        file.save(filepath)

        print(f'File saved to {filepath}') # Debugging

        # Convert to wav format
        wav_path = os.path.join(app.config['UPLOAD_FOLDER'], 'converted_audio.wav')
        convert_to_wav(filepath, wav_path)

        signal, sr = librosa.load(wav_path, sr=None)
        features = extract_features(signal, sr)
        features = np.expand_dims(features, axis=0)
        features = np.expand_dims(features, axis=-1)

        prediction = model.predict(features)
        label = 'Sehat' if np.argmax(prediction) == 0 else 'Pulpitis'
        print(f'Prediction: {label}') # Debugging
        return jsonify(result=label)

    print('File type not allowed') # Debugging
    return jsonify(error='File type not allowed'), 400
```

GAMBAR 8.

Memanggil kembali fungsi yang sudah didefinisikan sebelumnya

Proses terakhir terdapat proses seperti mengirim pesan error ke javascript jika tidak ada audio yang diinginkan, dan juga tidak ada file yang di unggah sesuai. Tahap ini juga memanggil ekstraksi fitur dan juga memanggil model yang sudah di upload agar dapat melakukan prediksi lalu mengirim pesan bahwa hasil deteksi pulpitis atau sehat ke javascript untuk ditampilkan.

V. KESIMPULAN

Berdasarkan hasil dan pembahasan. Peran *backend website* untuk mendeteksi pulpitis menggunakan MFCC dengan penggunaan flask sebagai *backend*-nya sangat penting untuk menjaga fungsionalitas, keamanan dan kinerjanya. Penelitian ini telah berhasil mengembangkan sistem untuk mendeteksi pulpitis. Performa model yang baik dibuktikan dengan laporan klasifikasi yang menunjukkan bahwa model mencapai akurasi 92%. Penelitian ini hanya berfokus pada deteksi pulpitis, diharapkan penelitian selanjutnya dapat mendeteksi penyakit gigi selain pulpitis melalui sinyal audio

REFERENSI

- [1] W. W. Putri and N. Nina, "Hubungan Antara Frekuensi Menyikat Gigi, Cara Menyikat Gigi dan Kebiasaan Makan dengan Kejadian Karies," *Journal of Public Health Education*, vol. 1, no. 01, pp. 13–19, Aug. 2021, doi: 10.53801/jphe.v1i01.13.
- [2] I. G. S. Kencana, "Relationship Between Education Level And Knowledge Of Dental Caries And Tooth Brushing Skills In Pregnant Women In South Denpasar District 2021," *Jurnal Kesehatan Gigi (Dental Health Journal)*, vol. 8, no. 2, pp. 80–89, Aug. 2021, doi: 10.33992/jkg.v8i2.1502.
- [3] Z. Mohammadi, P. V. Abbott, S. Shalavi, and M. Yazdizadeh, "Postoperative Pain Following Treatment of Teeth with Irreversible Pulpitis: A Review.," *PubMed*, vol. 83, no. 1, pp. 44–53, Jan. 2017, [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/29919992>.
- [4] M. Kaluža, M. Kalanj, and B. Vukelić, "A comparison of back-end frameworks for web application development," *Zbornik Veleučilišta U Rijeci*, vol. 7, no. 1, pp. 317–332, Jan. 2019, doi: 10.31784/zvr.7.1.10.
- [5] P. P. Arhandi, "Pengembangan Sistem Informasi Perijinan Tenaga Kesehatan Dengan Menggunakan Metode Back End Dan Front End," *Jurnal Teknologi Informasi*, Mar. 2016, doi: 10.36382/jti-tki.v7i1.192.
- [6] G. Cherubin, J. Hayes, and M. Juarez, "Website fingerprinting defenses at the application layer," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 2, pp. 186–203, Apr. 2017, doi: 10.1515/popets-2017-0023.
- [7] K. Ahn, H. Rakha, and A. Trani, "Microframework for modeling of High-Emitting Vehicles," *Transportation Research Record Journal of the Transportation Research Board*, vol. 1880, no. 1, pp. 39–49, Jan. 2004, doi: 10.3141/1880-05.
- [8] M. Algren, W. Fisher, and A. E. Landis, "Machine learning in life cycle assessment," in *Elsevier eBooks*, 2021, pp. 167–190. doi: 10.1016/b978-0-12-817976-5.00009-7.
- [9] N. Singh, R. A. Khan, and R. Shree, "MFCC and Prosodic feature Extraction Techniques: A Comparative study," *International Journal of Computer Applications*, vol. 54, no. 1, pp. 9–13, Sep. 2012, doi: 10.5120/8529-2061.
- [10] A. Keerio, B. K. Mitra, P. Birch, R. Young, and C. Chatwin, "On preprocessing of speech signals," *Zenodo (CERN European Organization for Nuclear Research)*, Nov. 2008, doi: 10.5281/zenodo.1332328.