

Sistem Identifikasi Manusia Bergerak Jatuh berdasarkan Ekstraksi Suara dan Citra

Susetyo Bagas Bhaskoro¹, Afaf Fadhil Rifa'i², Eugenia Angela Salsabillah³

^{1,2,3} Jurusan Teknik Otomasi Manufaktur dan Mekatronika, Politeknik Manufaktur Bandung

Email: bagas@ae.polman-bandung.ac.id

Informasi Artikel:

Received:
22 Oktober 2021

Accepted:
01 Oktober 2022

Available:
15 Desember 2022

ABSTRAK

Jatuh merupakan masalah kesehatan utama di seluruh dunia, terutama dalam dunia kesehatan karena pasien jatuh merupakan masalah terparah yang terus terjadi. Kebanyakan pasien yang jatuh dari tempat tidur tidak disaksikan. Hal ini diperparah dengan berbagai masalah kesehatan yang bisa diakibatkan oleh jatuh. Tetap di lantai setelah jatuh dapat menyebabkan trauma, cedera serius, dan bahkan kematian. Oleh karena itu, diperlukan sistem pendeteksi jatuh agar orang yang jatuh dapat segera diberikan pertolongan sebelum menimbulkan masalah kesehatan yang serius. Maka pada penelitian ini akan dibuat sistem identifikasi jatuh berdasarkan suara dan citra menggunakan metode MFCC (Mel-Frequency Cepstrum Coefficients) untuk ekstraksi suara dan LVQ (Learning Vector Quantization) untuk klasifikasi. Pengolahan citra menggunakan metode CNN (Convolutional Neural Network). Pada sistem ini, keduanya tidak bekerja sama, tetapi pengolahan citra bekerja setelah pengolahan suara. Sistem ini mampu mendeteksi jatuh dengan akurasi keseluruhan 93,3% untuk 15 kali pengujian pengolahan suara dan citra.

Kata Kunci:

MFCC
LVQ
Deep Learning
CNN
TensorFlow

ABSTRACT

Falls are a major health problem around the world, especially in the world of healthcare because patient falls are the top worst problem that continues to occur. Most patients who fall out of bed are not witnessed. This is exacerbated by the various health problems that can result from falling. Remaining on the floor after a fall can cause trauma, serious injury and even death. Therefore, a fall detection system is needed so that people who fall can be given immediate help before they cause serious health problems. So in this study, we will create a fall identification system based on sound and image using the MFCC (Mel-Frequency Cepstrum Coefficients) method for sound extraction and LVQ (Learning Vector Quantization) for classification. Image processing using CNN (Convolutional Neural Network) method. In this system, both do not work together, but image processing works after sound processing. The system is able to detect falls with an overall accuracy of 93.3% for 15 times of sound and image processing tests.

1 PENDAHULUAN

Jatuh menjadi masalah kesehatan utama di seluruh dunia. Dalam dunia kesehatan, pasien jatuh merupakan masalah terburuk teratas yang terus terjadi [1]. Cedera karena jatuh dapat mengakibatkan masalah yang serius dengan tingkat morbiditas dan mortalitas yang sangat signifikan yang juga berdampak pada biaya. Sekitar 3 hingga 20 persen pasien rawat inap jatuh setidaknya sekali selama mereka dirawat di rumah sakit. Prevalensi cedera berkisar antara 30 hingga 51 persen. [2]. Diperkirakan bahwa 80% hingga 90% jatuh yang terjadi di rumah sakit tidak disaksikan, dengan 50% hingga 70% terjadi di sekitar tempat tidur [3].

Terbaring terlalu lama setelah jatuh dapat menyebabkan berbagai macam komplikasi medis terjadi, seperti penurunan aktivitas hidup sehari-hari, gangguan kognitif, cedera serius, bahkan dapat menyebabkan kematian. Tidak hanya itu, jatuh juga dapat menyebabkan trauma pada pasien [4].

Penelitian mengenai deteksi jatuh telah dilakukan sebelumnya, baik yang menggunakan *accelerometer* dan *gyroscope* yang dipasang di tubuh manusia, yang menggunakan suara, maupun yang menggunakan pendeteksian Gambar. Terdapat kelemahan pada pendeteksian orang jatuh menggunakan *accelerometer* dan *gyroscope*, yaitu orang tersebut harus selalu memakai alat tersebut di tubuh mereka dan sistem pengendalian kamera yang dibuat belum bisa mendeteksi apakah Gambar yang ditangkap melalui kamera merupakan Gambar jatuh yang sebenarnya, karena sistem yang dibuat hanya menggerakkan kamera berdasarkan sudut yang diterima dari sensor [5]. Sedangkan Penelitian deteksi orang jatuh menggunakan suara ada yang dilakukan dengan menggunakan manekin jatuh [6] dan ada juga yang menggunakan orang jatuh yang sebenarnya. Namun, penelitian menggunakan ekstraksi suara masih terdapat kesalahan pendeteksian yang disebabkan karena suara lingkungan sekitar yang ikut terekam dan suara benda uji yang mirip dengan suara buku jatuh [7].

Penelitian menggunakan suara dilakukan dengan proses ekstraksi suara, kemudian dilanjutkan dengan proses klasifikasi. Dari sebuah penelitian mengenai ekstraksi suara manusia disebutkan bahwa tingkat pengenalan suara tertinggi dan waktu ekstraksi tercepat adalah MFCC jika dibandingkan dengan metode ZCPA dan LPC [8]. Metode MFCC sering dianggap terbaik untuk pengenalan suara karena sesuai dengan karakteristik pendengaran manusia dan memiliki kemampuan anti-*noise* yang baik. Hal ini didukung dengan penelitian deteksi jatuh menggunakan metode MFCC didapat akurasi yang lebih tinggi jika dibandingkan dengan metode LPC, yaitu 82.82% untuk MFCC dan metode LPC 78.72% untuk metode klasifikasi yang sama [9].

Terdapat banyak metode untuk klasifikasi suara, namun berdasarkan penelitian yang telah dilakukan, metode LVQ lebih baik jika dibandingkan dengan metode *Backpropagation*. Metode *backpropagation* memperoleh hasil akurasi sebesar 69,44% sedangkan metode LVQ memperoleh hasil akurasi sebesar 80,56% [10]. Berbeda dengan VQ konvensional, LVQ mencoba merancang model referensi yang dapat membedakan kelas, alih-alih meminimalkan kesalahan rata-rata secara keseluruhan. LVQ adalah pendekatan untuk pendekatan distribusi kelas berdasarkan sampel yang tersedia [11].

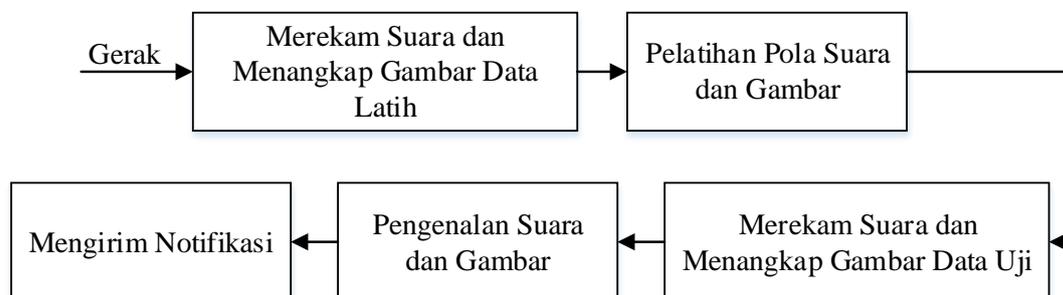
Terdapat banyak metode yang dapat digunakan untuk pengolahan citra, namun ada dua yang sering digunakan karena dinilai paling baik diantara metode lainnya. Diantaranya adalah *haar cascade classifier* dan CNN. Metode CNN lebih baik jika dibandingkan dengan metode *haar cascade classifier*, akurasi yang didapat yaitu 96,4% untuk metode CNN sedangkan dengan metode *haar cascade classifier* akurasi yang didapat 90% [12]. Metode

CNN (*Convolutional Neural Network*) dianggap mempunyai akurasi yang jauh lebih tinggi daripada metode pembelajaran mesin tradisional. CNN juga dinilai mempunyai konsistensi pendeteksian objek, baik untuk objek miring. CNN model dan *frameworks* dapat di-*training* dengan *custom dataset* untuk berbagai kondisi [13].

Oleh karena itu, pada penelitian ini ingin meningkatkan pendeteksian orang jatuh dengan menggunakan metode ekstraksi suara MFCC dan LVQ sebagai pengklasifikasian dan pengolahan citra dengan metode CNN sehingga jatuh dapat dideteksi sejak dini agar mendapat penanganan yang cepat sebelum terjadi komplikasi medis.

2 PERANCANGAN SISTEM

Perancangan sistem identifikasi manusia bergerak jatuh berdasarkan ekstraksi suara dan citra seperti pada Gambar 1 berikut.



Gambar 1. Blok Diagram Sistem

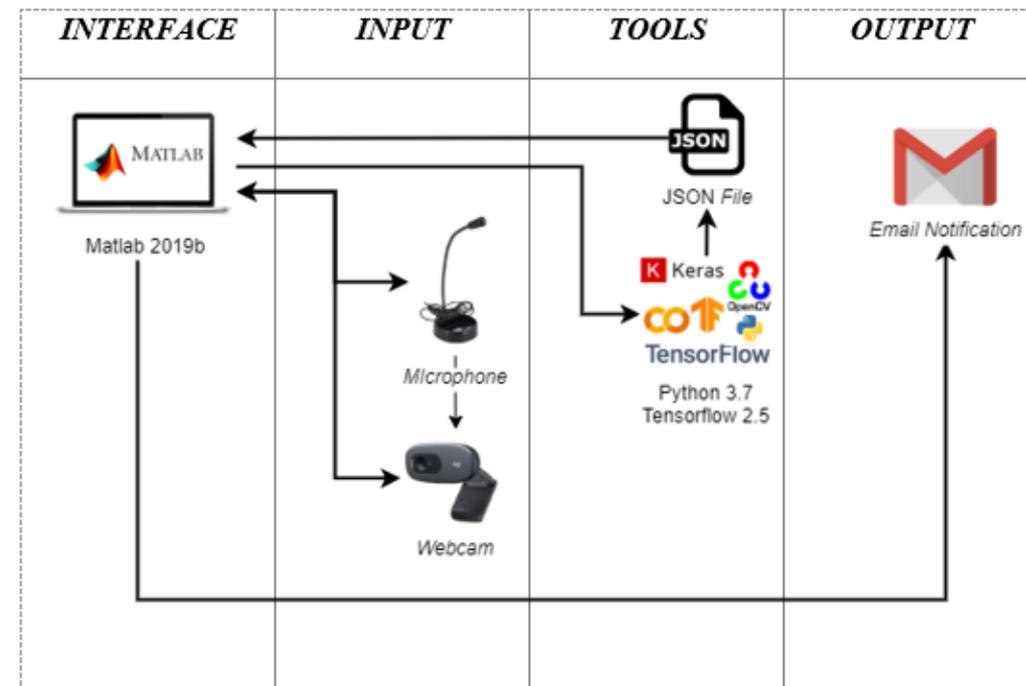
Data latih untuk proses ekstraksi suara diambil dengan merekam suara dan file suara yang telah direkam kemudian disimpan dalam format .wav selama 15 detik. Sedangkan untuk pengolahan citra data latih didapat dengan menangkap Gambar kemudian disimpan dalam format .png.

Suara yang telah direkam kemudian diekstraksi dengan menggunakan metode MFCC. Satu atau lebih pola uji yang berhubungan dengan bunyi suara dari kelas yang sama, digunakan untuk membuat pola representative dari ciri-ciri kelas tersebut. Hasilnya yang biasa disebut dengan pola referensi, dapat menjadi sebuah model yang mempunyai karakteristik bentuk statik dari ciri-ciri pola referensi [14].

Pada proses pengenalan Gambar menggunakan metode CNN. Pertama, data Gambar yang ada melalui proses *preprocessing* akan di-*resize*, *rescale*, dan sebagainya untuk menyamakan ukuran Gambar. Dan selanjutnya *dataset* yang ada di-*training* untuk mendapatkan model yang selanjutnya digunakan untuk proses prediksi.

Dari data uji, baik untuk suara maupun Gambar selanjutnya diklasifikasikan berdasarkan kelas yang telah dibuat untuk selanjutnya diprediksi data uji yang ada merupakan kelas yang mana. Jika hasil prediksi sistem menunjukkan jatuh, maka sistem akan mengirim notifikasi ke pengguna.

Pada sistem yang dibuat ini pengolahan citra bekerja setelah proses pengolahan suara menampilkan hasil prediksi pada antarmuka Matlab.



Gambar 2. Gambaran Umum Sistem

Gambaran umum sistem deteksi manusia jatuh yang dibuat dapat dilihat pada Gambar 3. Pertama sistem yang dibuat akan mendeteksi suara yang terjadi di sekitar tempat tidur. Suara akan direkam, diekstraksi, kemudian diklasifikasikan berdasarkan model yang ada oleh Matlab. Setelah pemrosesan suara selesai dan didapat hasil pengolahan suara, selanjutnya *webcam* akan menangkap Gambar untuk melakukan pengolahan citra pendeteksian jatuh. Gambar yang ditangkap akan diproses di Matlab dengan memanggil program prediksi oleh python menggunakan *tools* TensorFlow dan Keras. Hasil prediksi yang didapat berupa file json. File json hasil prediksi kemudian dipanggil oleh Matlab untuk ditampilkan pada antarmuka Matlab. Jika hasil prediksi yang didapat adalah manusia jatuh, maka sistem akan mengirimkan notifikasi orang jatuh melalui surel pengguna.

2.1 Perangkat yang Digunakan

2.1.1 Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini dapat dilihat pada Tabel berikut

Tabel 1. Perangkat Keras yang Digunakan

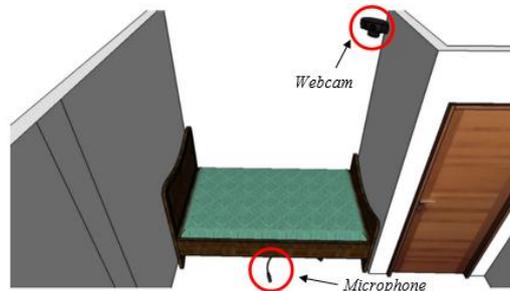
| Perangkat Keras | Tipe | Spesifikasi |
|-------------------|--------------------------|--------------------------------|
| Laptop | ACER AMD Ryzen 3 – 2200U | 8 GB RAM (Windows 10 – 64 Bit) |
| <i>Microphone</i> | Nubwo M-180 | 50 – 16000 Hz, -55dB ± 2d |
| Kamera | Logitech C270 | 720p / 30fps |

2.1.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini yaitu Matlab R2019b karena mendukung python 3.7 untuk dapat menjalankan proses prediksi TensorFlow dan Keras.

2.1.3 Skenario Area Penelitian

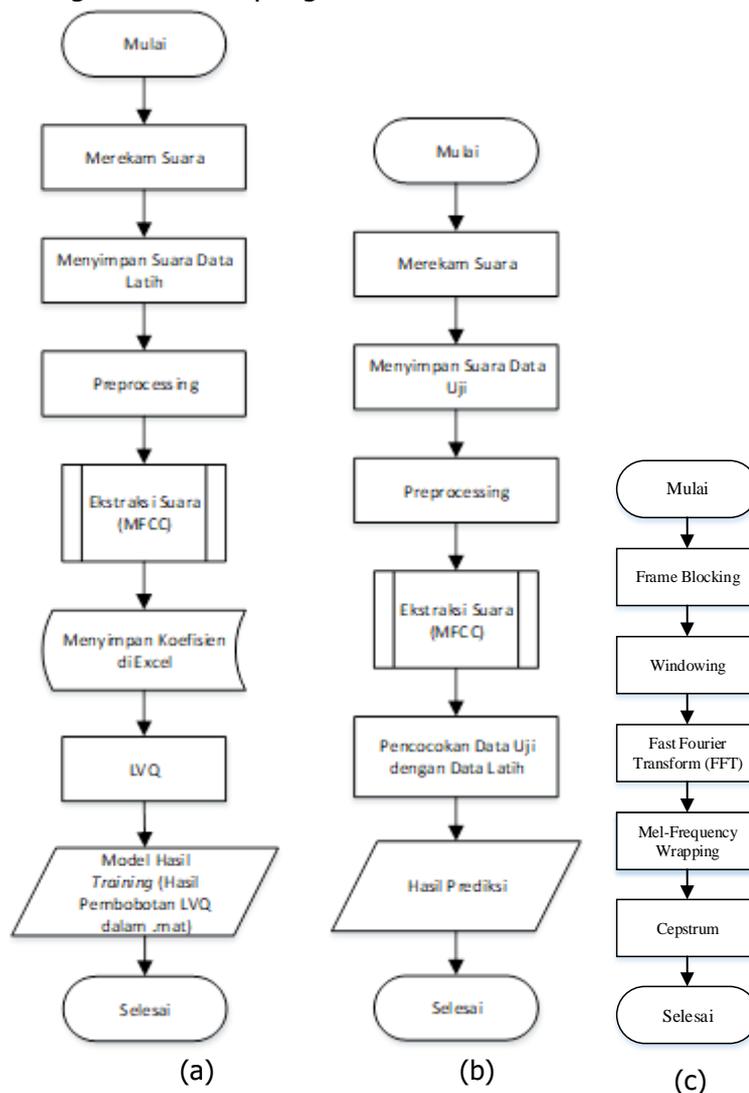
Dari Gambar berikut dapat dilihat bahwa kamera berada di atas tempat tidur dengan ketinggian dua meter dari tempat tidur, sedangkan mikrofon berada di bawah tempat tidur dengan jarak 20-30 cm dari subjek yang jatuh, dan ketinggian tempat tidur dari lantai adalah 60 cm.



Gambar 3. Area Penelitian

2.2 Pengolahan Suara

Berikut merupakan diagram alir dari pengolahan suara.



Gambar 4. (a) Diagram Alir Proses *Training*, (b) Diagram Alir Pengujian, (c) Metode MFCC

Berdasarkan diagram alir di atas, proses pengolahan suara diawali dengan perekaman suara untuk *training* (data latih). Suara yang telah direkam akan melalui proses *preprocessing* sinyal suara untuk selanjutnya dilakukan ekstraksi oleh MFCC. Hasil koefisien ekstraksi suara kemudian akan disimpan dalam excel. Dan selanjutnya proses *training* oleh LVQ yang hasil *training* modelnya akan disimpan dalam format .mat. Setelah itu dilakukan proses pengujian data suara dengan pencocokan koefisien hasil pembobotan LVQ yang telah dibuat. Data suara pengujian melalui proses *preprocessing*, kemudian ekstraksi oleh MFCC dan diprediksi untuk menentukan suara tersebut termasuk kelas yang mana.

2.2.1 Preprocessing Pengolahan Suara

1. Perekaman Suara dan Sampling

Untuk perekaman suara dilakukan selama 15 detik dengan f_s (*sampling frequency*) sebesar 8000Hz, yang berarti terdapat 8000 data tiap detik dalam satu kali perekaman. Sehingga dalam satu data suara selama 15 detik terdapat 120000 data.

2. DC Removal

Proses ini digunakan untuk menghilangkan komponen DC sehingga diperoleh normalisasi d dari data *input* suara, membuat rata-rata nilai sampel positif dan negatif sama [15].

$$D[i] = S[i] - \frac{\sum_{i=1}^n S[i]}{n} \quad (1)$$

3. Normalizing

Normalisasi yang dilakukan yaitu normalisasi amplitudo. Normalisasi amplitudo dilakukan untuk mengatasi tingkat energi yang tidak konsisten antara tiap isyarat. Sehingga untuk semua sinyal masukan memiliki nilai yang sama, yaitu 1 sebagai nilai maksimumnya [16].

$$N[i] = \frac{S[i]}{\max S[i]} \quad (2)$$

4. Preemphasis

Preemphasis merupakan filter yang bertujuan untuk untuk mempertahankan frekuensi – frekuensi tinggi pada spektrum sehingga dihasilkan data input suara dengan noise yang rendah [15].

$$y(n) = x(n) - \alpha x(n - 1) \quad (3)$$

Dimana α (α) adalah sebuah ketetapan filter *preemphasis*, dengan nilai $0.9 < \alpha < 1.0$ [17].

2.2.2 Mel-Frequency Cepstral Coefficients

Berikut merupakan diagram alir dari metode MFCC yang digunakan.

1. Frame Blocking

Frame Blocking (framing) adalah proses pembagian suara menjadi beberapa *frame* yang nantinya dapat memudahkan perhitungan dan analisa suara karena komputasi tidak bisa langsung menggunakan 120000 data.

$$Total\ frame = \left(\frac{(I - N)}{M} + 1 \right) \quad (4)$$

2. Windowing

Windowing digunakan untuk meminimalisir diskontinuitas sinyal pada permulaan dan akhir setiap *frame*. Fungsi *windowing* yang digunakan dalam penelitian ini adalah

window hamming karena fungsi *hamming* dapat membuat data pada awal *frame* dan akhir *frame* mendekati 0 dengan baik [16].

$$y(n) = x(n)w(n), 0 \leq n \leq N - 1 \quad (5)$$

3. *Fast Fourier Transform*

Fast Fourier Transform (FFT) mengkonversi setiap *frame* N sampel dalam domain waktu ke domain frekuensi. FFT merupakan sebuah algoritma cepat untuk implementasi *Discrete Fourier Transform* (DFT) yang dioperasikan pada sebuah sinyal waktu-diskret yang terdiri dari N sampel [14].

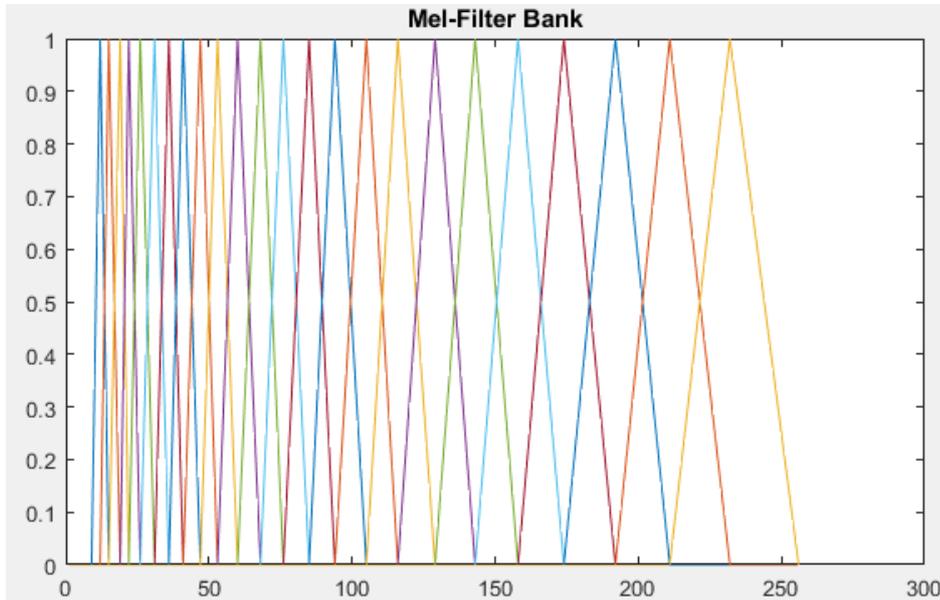
$$f(n) = \sum_{k=0}^{N-1} y_k e^{-2\pi fkn/N}, n = 0, 1, 2, \dots, N - 1 \quad (6)$$

4. *Mel-Frequency Wrapping*

Tahapan ini akan memfilter sinyal suara yang sudah diproses pada tahapan sebelumnya dan menciptakan sebuah pola yang dinamakan *mel-spectrum*. Sebelum difilter, harus ditentukan terlebih dahulu nilai pada *filterbank*. Jika nilai sudah ditentukan maka proses filtering dapat diaplikasikan ke sinyal suara yang sudah diproses pada tahapan sebelumnya dan akan menghasilkan *mel-spectrum*. Langkah pertama yaitu frekuensi linear dikonversi menjadi *mel-frequency*. Terlebih dahulu frekuensi linear dibagi menjadi frekuensi terendah dengan frekuensi tertinggi. Kemudian kedua frekuensi ini dikonversi dengan menggunakan rumus yang sudah dijelaskan pada Persamaan (7). Tiap filter pada *mel-filterbank* akan menghasilkan *mel-spectrum* yang besaran energinya bergantung dari sinyal suara yang masuk.

$$mel(f) = 1127 \times \log \left(1 + \frac{f}{700} \right) \quad (7)$$

Berikut merupakan *triangular filters* yang digunakan berjumlah 24 filter.



Gambar 5. *Mel-Filterbank*

5. *Cepstrum*

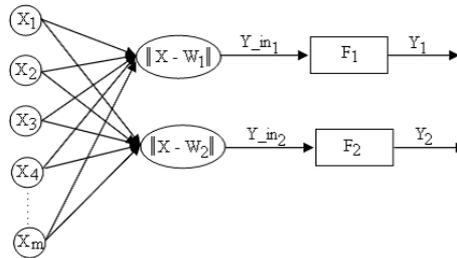
Tahapan ini merupakan tahapan akhir dari algoritma MFCC yang akan menghasilkan koefisien-koefisien yang digunakan untuk pengenalan suatu sinyal suara. Koefisien didapatkan dengan mengubah *mel-spectrum* menjadi domain waktu. Sinyal suara yang sudah diproses pada tahapan sebelumnya akan menghasilkan *mel-spectrum* yang memiliki besaran energi. Besaran energi ini akan diubah menjadi koefisien-koefisien.

Nilai yang terkandung di dalam koefisien-koefisien tersebut disebut dengan vektor akustik yang menjadi ciri dari suatu sinyal suara.

$$C_n = \sum_{k=1}^K (\log S_k) \cos \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right], n = 1, 2, \dots, K \quad (8)$$

2.2.3 Learning Vector Quantization

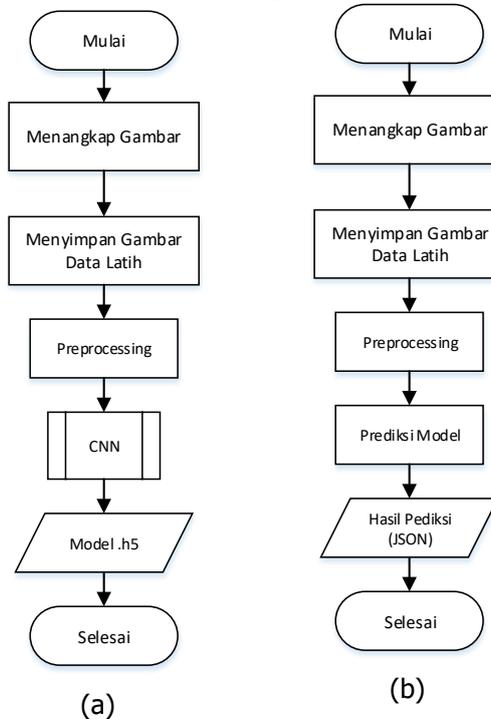
Learning Vector Quantization (LVQ) digunakan untuk proses klasifikasi obyek terhadap data fitur yang sudah tersimpan. LVQ adalah suatu metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi (*supervised*) [18]. LVQ bekerja dengan setiap unit output mempresentasikan sebuah kelas. Dengan kata lain LVQ adalah metode pengelompokan dimana target/kelasnya setiap kelompok/jumlah kelompok sudah ditentukan.



Gambar 6. Arsitektur Jaringan LVQ

2.3 Pengolahan Citra

Berikut merupakan diagram alir dari proses pengolahan citra.



Gambar 7. (a) Diagram Alir Proses *Training*, (b) Diagram Alir Pengujian

Berdasarkan diagram alir di atas, proses pengolahan citra diawali dengan menangkap Gambar *dataset* untuk *training* (data latih) yang terdiri dari *train set* dan *validation set*.

Dataset training disimpan di Google Drive karena proses *training* pengolahan citra dilakukan di Colab. Kemudian data *train* dan *validation* dimuat dari Google Drive untuk proses *training*. Selanjutnya *dataset* melalui proses *preprocessing* oleh Keras. Setelah *preprocessing*, Gambar diolah menggunakan metode CNN untuk mendapatkan model dengan format .h5. Kemudian Gambar yang akan diuji melalui *preprocessing* untuk memprediksi Gambar dari model yang ada. Hasil prediksi selanjutnya disimpan dalam format .json agar dapat dipanggil oleh Matlab.

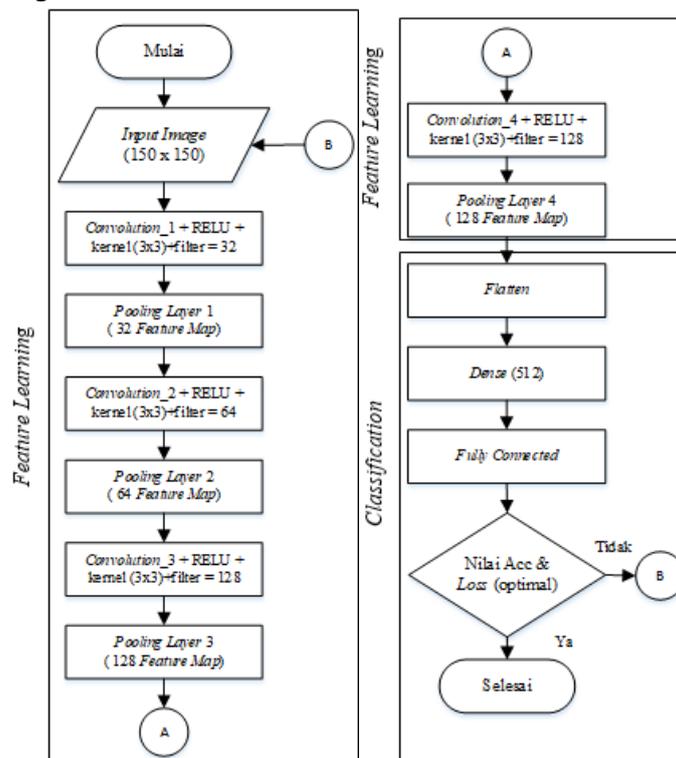
2.3.1 Preprocessing Pengolahan Citra

Pada penelitian ini data diambil secara manual yang terdiri dari empat kelas, yaitu jatuh, duduk, berdiri, dan tidak ada aktivitas apa pun. Data latih disimpan dalam format .png dan semua *dataset* selanjutnya dijadikan dalam satu format file .zip dan disimpan dalam Google Drive. Perbandingan antara data *train* dan *validation* yang digunakan yaitu 60 : 40.

Preprocessing pengolahan citra menggunakan *library* Keras melalui proses augmentasi oleh ImageDataGenerator. Proses augmentasi citra menggunakan *library* Keras dengan memberi parameter pada *Resize*, *Rescale*, *Rotation*, dan Pembalikan.

2.3.2 Convolutional Neural Network

Berikut merupakan diagram alir metode CNN.



Gambar 8. Diagram Alir Metode CNN

Diagram alir di atas menunjukkan tahapan dalam CNN. Umumnya dalam CNN memiliki 2 tahapan, yaitu tahap *feature learning* dan *classification*. Input Gambar pada model CNN menggunakan citra yang berukuran 150x150x3. Angka tiga yang dimaksud adalah sebuah citra yang memiliki 3 channel yaitu *Red*, *Green*, dan *Blue* (RGB). Citra masukan kemudian akan diproses terlebih dahulu melalui proses konvolusi dan proses *pooling* pada tahapan *feature learning*. *Feature learning* adalah teknik yang memungkinkan sebuah system

berjalan secara otomatis untuk menentukan representasi dari sebuah Gambar menjadi *features* yang berupa angka-angka yang merepresentasikan Gambar tersebut. Tahap *Classification* adalah sebuah tahap dimana hasil dari *feature learning* akan digunakan untuk proses klasifikasi berdasarkan subclass yang sudah ditentukan.

3 HASIL

Pada penelitian ini sistem identifikasi manusia jatuh menggunakan ekstraksi suara dan citra. Keduanya bekerja bergantian, pertama suara akan selalu aktif untuk memproses suara tiap 15 detik. Kemudian setelah proses suara selesai dan menghasilkan prediksi, maka selanjutnya *webcam* akan aktif untuk menangkap Gambar dan kemudian melakukan prediksi dengan memanggil fail prediksi dari Matlab.



Gambar 9. Waktu Proses Prediksi

Sehingga waktu yang dibutuhkan untuk menguji sistem keseluruhan adalah 70 detik, karena menggunakan perangkat yang tidak mendukung GPU. Laptop dengan GPU hanya memerlukan 37 detik untuk satu kali proses prediksi hingga mengirim notifikasi.

3.1 Pengujian Sistem Mendeteksi Manusia Jatuh

Pengujian ini dilakukan untuk mengetahui apakah sistem dapat mendeteksi suara manusia jatuh dengan baik. Data latih yang digunakan terdiri dari 20 suara manusia jatuh dari kasur atau tempat tidur, 15 suara naik ke tempat tidur, dan 15 suara turun dari tempat tidur.

Tabel 2. Pengujian Sistem Mendeteksi Manusia Jatuh

| Pengujian Ke- | Keadaan | Hasil Prediksi | | Hasil Sebenarnya | Hasil Prediksi Keseluruhan |
|---------------|------------------|------------------|-----------|------------------|----------------------------|
| | | Suara | Citra | | |
| 1 | Jatuh dari Kasur | Jatuh | Jatuh | Jatuh | Jatuh |
| 2 | Jatuh dari Kasur | Turun dari Kasur | Jatuh | | Jatuh |
| 3 | Jatuh dari Kasur | Turun dari Kasur | Jatuh | | Jatuh |
| 4 | Jatuh dari Kasur | Jatuh | Jatuh | | Jatuh |
| 5 | Jatuh dari Kasur | Jatuh | Jatuh | | Jatuh |
| 1 | Turun dari Kasur | Turun dari Kasur | Berdiri | Tidak Jatuh | Tidak Jatuh |
| 2 | Turun dari Kasur | Turun dari Kasur | Berdiri | | Tidak Jatuh |
| 3 | Turun dari Kasur | Naik ke Kasur | Berdiri | | Tidak Jatuh |
| 4 | Turun dari Kasur | Naik ke Kasur | Berdiri | | Tidak Jatuh |
| 5 | Turun dari Kasur | Turun dari Kasur | Berdiri | | Tidak Jatuh |
| 1 | Naik ke Kasur | Turun dari Kasur | Jatuh | Tidak Jatuh | Jatuh |
| 2 | Naik ke Kasur | Naik ke Kasur | Tidak Ada | | Tidak Jatuh |

| | | | | |
|---|---------------|---------------|-----------|-------------|
| 3 | Naik ke Kasur | Naik ke Kasur | Tidak Ada | Tidak Jatuh |
| 4 | Naik ke Kasur | Naik ke Kasur | Tidak Ada | Tidak Jatuh |
| 5 | Naik ke Kasur | Naik ke Kasur | Tidak Ada | Tidak Jatuh |

Sistem identifikasi jatuh yang andal harus membuat keputusan apakah jatuh telah terjadi berdasarkan fitur yang berbeda dari sinyal suara dan Gambar. Deteksi jatuh baik positif jika detektor mengenali jatuh dengan benar, atau negatif jika tidak. Karena keluarannya adalah biner, kualitas detektor tidak dapat dievaluasi hanya dari satu pengujian, melainkan perlu melakukan analisis statistik pada serangkaian pengujian. Ada 4 kemungkinan kasus:

- *True Positive* (TP): terjadi jatuh, sistem mendeteksinya.
- *False Positive* (FP): sistem menunjukkan jatuh, tetapi tidak terjadi.
- *True Negative* (TN): gerakan tidak jatuh dilakukan, sistem tidak menyatakan jatuh.
- *False Negative* (FN): terjadi jatuh tetapi sistem tidak mendeteksinya.

Untuk mengevaluasi respons terhadap 4 situasi ini, diusulkan 3 kriteria:

1. *Sensitivity*: kapasitas untuk mendeteksi jatuh,

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (9)$$

Dengan TP = 5 dan FN = 0, sehingga dalam penelitian ini didapat nilai $Sensitivity = \frac{5}{5+0} \times 100\% = 100\%$.

2. *Specificity*: kapasitas untuk mendeteksi hanya jatuh,

$$Specificity = \frac{TN}{TN + FP} \times 100\% \quad (10)$$

Dengan TN = 9 dan FP = 1, sehingga dalam penelitian ini didapat nilai $Specificity = \frac{9}{9+1} \times 100\% = 90\%$.

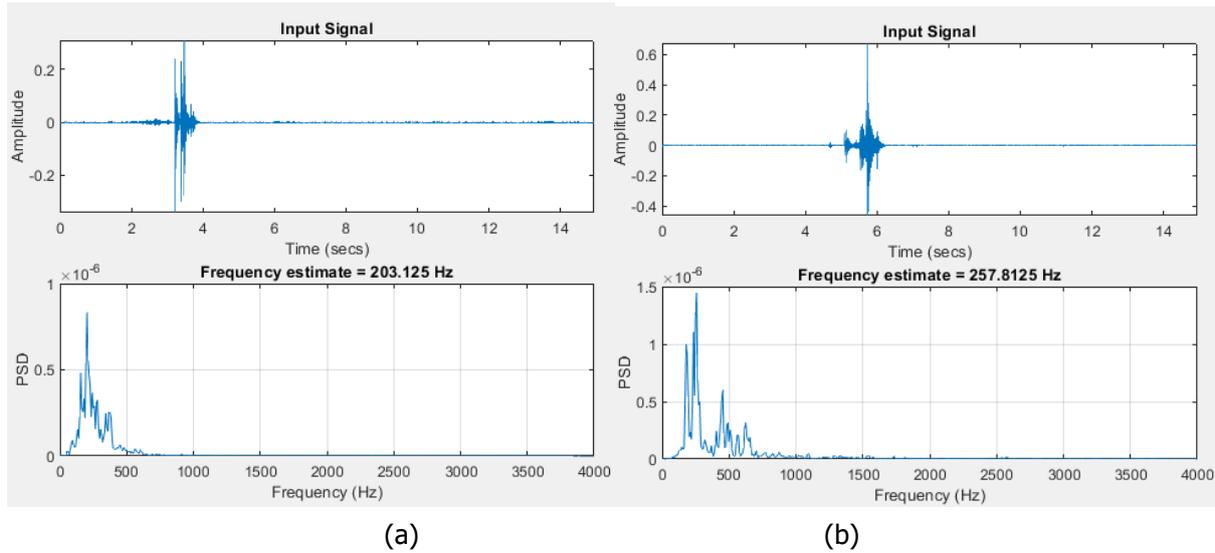
3. *Accuracy*: kemampuan sistem untuk membedakan antara jatuh dan tidak jatuh,

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \times 100\% \quad (11)$$

Dengan TP = 5, TN = 9, FN = 0, dan FP = 1, sehingga dalam penelitian ini didapat nilai $Accuracy = \frac{5+9}{5+0+9+1} \times 100\% = 93,3\%$.

3.2 Pengujian Berat Manusia Jatuh Berbeda

Pada pengujian ini, berat manusia jatuh yang diuji yaitu 54 kg dan 70 kg. Diambil frekuensi tertinggi untuk manusia dengan berat 54 kg, yaitu 203,125 Hz. Dan untuk berat 70 kg, yaitu 257,8125 Hz.



Gambar 10. Sinyal Frekuensi Manusia Jatuh (a) 54 kg, (b) 70 kg

Pada Tabel berikut merupakan perbandingan frekuensi suara yang dihasilkan berdasarkan berat badan manusia jatuh yang berbeda dengan 10 kali pengujian.

Tabel 3. Tabel Hasil Pengujian Berat Manusia Berbeda

| Pengujian Ke- | Frekuensi (Hz) | |
|---------------|----------------|-----------------------|
| | Berat 54 kg | Kolom 3 Berat 70kg |
| 1 | 203,125 | 210,9375 |
| 2 | 203,125 | 187,5 |
| 3 | 156,25 | 187,5 |
| 4 | 132,8125 | 257,8125 |
| 5 | 156,25 | 179,6875 |
| 6 | 156,25 | 210,9375 |
| 7 | 156,25 | 234,375 |
| 8 | 203,125 | 265,625 |
| 9 | 156,25 | 179,6875 |
| 10 | 203,125 | 218,75 |
| Rata-rata | 172,65625 | 213,28125 |

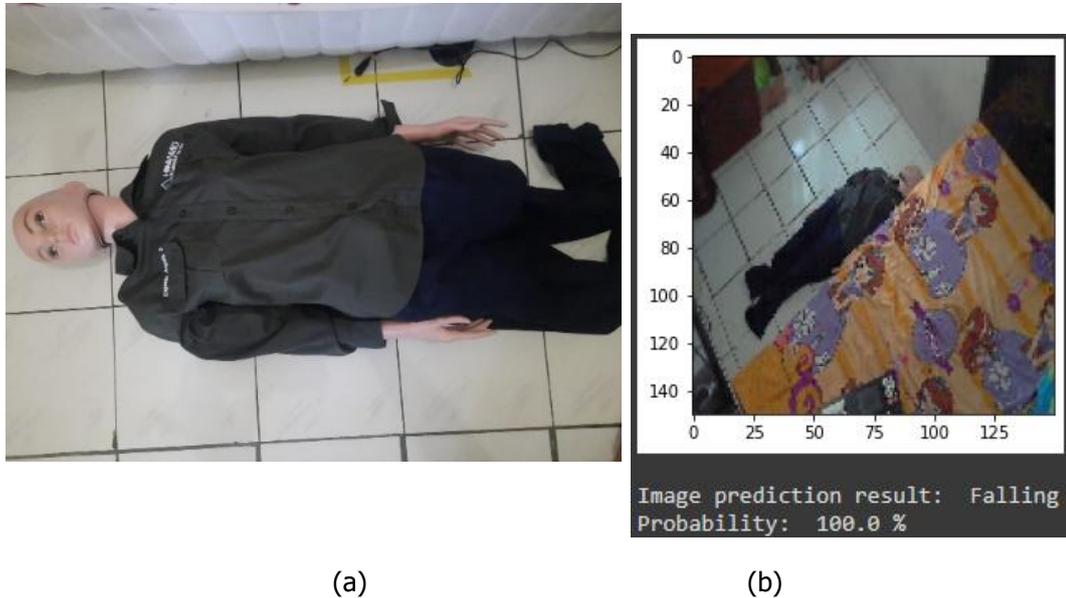
Dari Tabel di atas dapat dilihat bahwa frekuensi rata-rata manusia jatuh dengan berat 54 kg yaitu 172,65265 Hz, sedangkan untuk berat 70 kg yaitu 213,28125. Oleh karena itu, berat manusia mempengaruhi nilai frekuensi suara jatuhnya. Sehingga pada sistem yang dibuat ini, sistem akan mendeteksi jatuh jika frekuensi suara yang masuk berada diantara 156,25 hingga 257,8125 Hz akan diidentifikasi sebagai jatuh.

3.3 Pengujian Manusia Jatuh dengan Benda Lain

Berikut merupakan hasil pengujian manusia jatuh dengan benda lain yang mirip dengan manusia jatuh.

1. Pengujian Manusia Jatuh dengan Benda yang Bentuknya Mirip dengan Manusia

Pengujian dilakukan dengan menjatuhkan manekin. Pengujian dilakukan untuk mengetahui bagaimana sistem jika mendeteksi benda yang mirip dengan manusia. Hasil prediksi pengolahan citra untuk manekin jatuh adalah sebagai berikut.

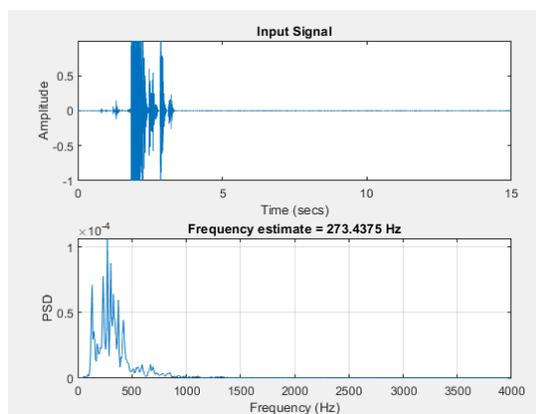


Gambar 11. (a) Manekin yang Diuji, (b) Hasil Prediksi Citra Manekin Jatuh

Dapat dilihat dari Gambar di atas bahwa manekin yang jatuh di lantai diprediksi sebagai falling, karena bentuk manekin jatuh mirip dengan posisi manusia jatuh. Selain itu, hal ini dikarenakan sistem yang dibuat saat ini hanya memiliki 4 kelas pengolahan citra yang hanya memprediksi bentuk tubuh manusia.

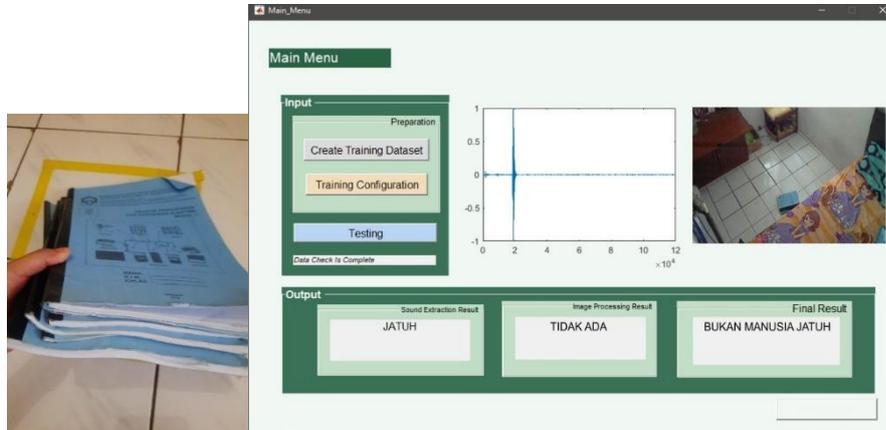
Namun, secara keseluruhan sistem tidak akan mendeteksi adanya jatuh, karena frekuensi suara manekin yang berbeda dengan suara manusia jatuh. Berikut merupakan frekuensi dari manekin jatuh.

Frekuensi manekin jatuh yaitu 273,4375 Hz, yaitu di atas frekuensi suara manusia jatuh. Frekuensi di bawah atau di atas frekuensi suara manusia jatuh akan diabaikan, sehingga sistem tidak akan mendeteksi adanya jatuh dari manekin yang terjatuh.



Gambar 12. Sinyal Frekuensi Suara Manekin Jatuh

2. Pengujian Manusia Jatuh dengan Buku yang Suaranya Mirip dengan Suara Manusia Jatuh
Buku yang diuji merupakan modul 5 modul praktikum yang suaranya mirip dengan suara manusia jatuh.

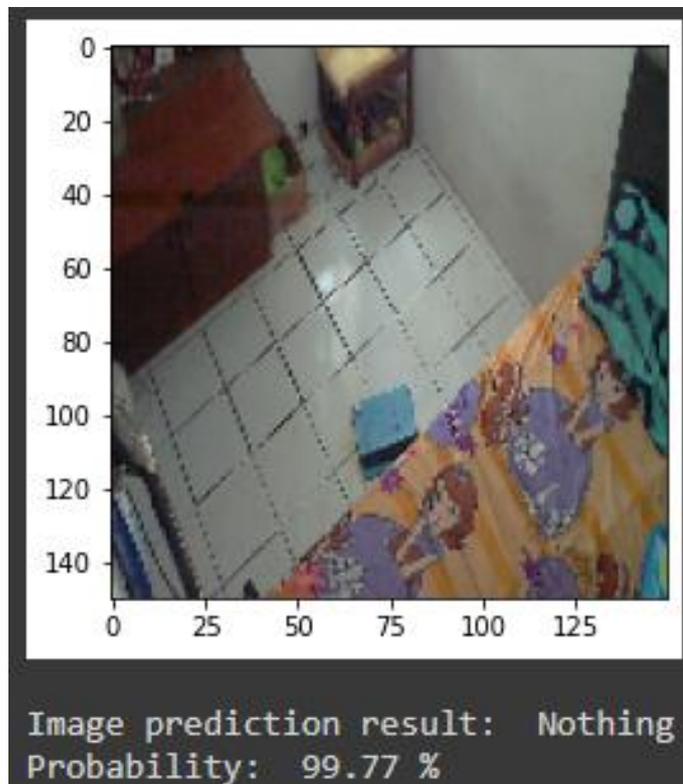


(a)

(b)

Gambar 13. (a) Buku Jatuh yang Diuji, (b) Hasil Prediksi

Walaupun hasil prediksi pengolahan suara menunjukkan jatuh karena frekuensi suara buku jatuh yaitu 150 – 200 Hz, namun pengolahan citra tidak mendeteksi adanya jatuh, karena untuk buku jatuh termasuk ke dalam kelas tidak ada dalam pengolahan citra sehingga hasil prediksi keseluruhan tidak menunjukkan jatuh.



Gambar 14. Hasil Prediksi Pengolahan Citra Buku Jatuh

Pada penelitian sebelumnya akurasi dari sistem deteksi jatuh yang dibuat yaitu 86,67% dan sistem yang dibuat sebelumnya belum mampu membedakan antara suara manusia jatuh dengan suara benda jatuh yang mirip, yaitu suara buku jatuh [7]. Sehingga sistem yang dibuat ini lebih baik jika dibandingkan dengan penelitian sebelumnya, karena akurasi yang didapat lebih besar dari penelitian sebelumnya, yaitu 93,3% dan sistem mampu membedakan suara manusia jatuh dengan benda jatuh lainnya yang frekuensinya sama.

4 KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan pada sistem identifikasi manusia bergerak jatuh yang telah dibangun, terdapat beberapa kesimpulan yang dapat ditarik diantaranya adalah sebagai berikut.

1. Berdasarkan dari hasil pengujian nilai akurasi dari sistem yang telah dibuat adalah sebesar 93.3%.
2. Sistem mampu membedakan manusia jatuh dengan benda jatuh lain yang frekuensinya sama.
3. Spesifikasi perangkat yang digunakan untuk menjalankan sistem mempengaruhi lamanya waktu prediksi. Laptop yang mendukung GPU lebih baik digunakan pada sistem berbasis *machine learning*.

5 REFERENSI

- [1] Boushon B, "How-to Guide: Reducing Patient Injuries from Falls," *Inst. Healthc. Improv.*, no. December, p. 60, 2012, [Online]. Available: <http://www.ih.org/resources/Pages/Tools/TCABHowToGuideReducingPatientInjuriesfromFalls.aspx>.
- [2] P. Potter *et al.*, "Evaluation of Sensor Technology to Detect Fall Risk and Prevent Falls in Acute Care," *Jt. Comm. J. Qual. Patient Saf.*, vol. 43, no. 8, pp. 414–421, 2017, doi: 10.1016/j.jcjq.2017.05.003.
- [3] P. Potter *et al.*, "Anatomy of inpatient falls: Examining fall events captured by depth-sensor technology," *Jt. Comm. J. Qual. Patient Saf.*, vol. 42, no. 5, pp. 225–231, 2016, doi: 10.1016/s1553-7250(16)42029-5.
- [4] E. J. Bisson, E. W. Peterson, and M. Finlayson, "Delayed Initial Recovery and Long Lie after a Fall among Middle-Aged and Older People with Multiple Sclerosis," *Arch. Phys. Med. Rehabil.*, vol. 96, no. 8, pp. 1499–1505, 2015, doi: 10.1016/j.apmr.2015.04.012.
- [5] R. A. MELITA, S. B. BHASKORO, and R. SUBEKTI, "Pengendalian Kamera berdasarkan Deteksi Posisi Manusia Bergerak Jatuh berbasis Multi Sensor Accelerometer dan Gyroscope," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 6, no. 2, p. 259, 2018, doi: 10.26760/elkomika.v6i2.259.
- [6] D. Litvak, Y. Zigel, and I. Gannot, "Fall detection of elderly through floor vibrations and sound," *Proc. 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS'08 - "Personalized Healthc. through Technol.*, pp. 4632–4635, 2008, doi: 10.1109/iembs.2008.4650245.
- [7] F. Nisrina Tsamara Hastuti, "SISTEM MONITORING ORANG JATUH BERBASIS EKSTRASI SUARA MENGGUNAKAN METODE MEL-FREQUENCY CEPSTRUM COEFFICIENT DAN LEARNING VECTOR QUANTIZATION PROGRAM STUDI EKSTRASI SUARA MENGGUNAKAN METODE MEL-FREQUENCY CEPSTRUM COEFFICIENTS," 2020.
- [8] E. Riyanto and Sutejo, "Perbandingan Metode Ekstraksi Ciri Suara MFFCC, ZCPA, dan LPC," vol. 10, pp. 1–8, 2014, [Online]. Available: <http://www.ejournal.himsya.ac.id/index.php/HIMSYATECH/article/view/61>.

- [9] M. Cheffena, "Fall Detection Using Smartphone Audio Features," *IEEE J. Biomed. Heal. Informatics*, vol. 20, no. 4, pp. 1073–1080, 2016, doi: 10.1109/JBHI.2015.2425932.
- [10] S. Tomia, Z. A. Leleury, and S. N. Aulele, "PERBANDINGAN METODE JARINGAN SARAF TIRUAN BACKPROPAGATION DAN LEARNING VECTOR QUANTIZATION DALAM DETEKSI HAMA PENGEREK BATANG (Studi Kasus: Kabupaten Seram Bagian Barat Provinsi Maluku)," vol. 11, pp. 13–26, 2017.
- [11] H. Song and S. Lee, "CONTRIBUTED ARTICLE LVQ Combined with Simulated Annealing for Optimal Design of Large-set Reference Models," vol. 9, no. 2, pp. 329–336, 1996.
- [12] P. C. Useche Murillo, R. J. Moreno, and J. O. Pinzon Arenas, "Comparison between CNN and Haar classifiers for surgical instrumentation classification," *Contemp. Eng. Sci.*, vol. 10, no. 28, pp. 1351–1363, 2017, doi: 10.12988/ces.2017.711157.
- [13] N. O'Mahony *et al.*, "Deep Learning vs. Traditional Computer Vision," *Adv. Intell. Syst. Comput.*, vol. 943, no. Cv, pp. 128–144, 2020, doi: 10.1007/978-3-030-17795-9_10.
- [14] A. Setiawan, A. Hidayatno, and R. R. Isnanto, "Aplikasi Pengenalan Ucapan dengan Ekstraksi Mel-Frequency Cepstrum Coefficients (MFCC) Melalui Jaringan Syaraf Tiruan (JST) Learning Vector Quantization (LVQ) untuk Mengoperasikan Cursor Komputer," *Transmisi*, vol. 13, no. 3, pp. 82-86–86, 2011, doi: 10.12777/transmisi.13.3.82-86.
- [15] N. Nurhamidah, E. C. Djamal, and R. Ilyas, "Perintah Menggunakan Sinyal Suara dengan Mel- Frequency Cepstrum Coefficients dan Learning Vector Quantization," *Semin. Nas. Apl. Teknol. Inf. 2017*, pp. 11–16, 2017.
- [16] S. B. Bhaskoro, *Komputasi Sinyal Suara Teori dan Contoh Aplikasinya*. Bandung: Politeknik Manufaktur Negeri Bandung, 2020.
- [17] R. Umar, I. Riadi, and A. Hanif, "Analisis Bentuk Pola Suara Menggunakan Ekstraksi Ciri Mel-Frequency Cepstral Coefficients (MFCC)," *CogITO Smart J.*, vol. 4, no. 2, p. 294, 2019, doi: 10.31154/cogito.v4i2.130.294-304.
- [18] S.B. Bhaskoro, "Aplikasi pengenalan gender menggunakan suara," *Semin. Nas. Apl. Teknol. Inf.*, vol. 2012, no. Snati, pp. 15–16, 2012.