

IMPLEMENTASI *MACHINE LEARNING* UNTUK IDENTIFIKASI ORANG BATUK/BERSIN

Sandy Bhawana Mulia¹, Nur Wisma Nugraha², Muhammad Hanif Robbani³
Teknik Otomasi Manufaktur & Mekatronika, Politeknik Manufaktur Bandung^{1,2,3}
email¹: sandy@ae.polman-bandung.ac.id

Abstract

The most important thing in machine learning is feature extraction from a raw data. In this journal will be explained regarding Implementation of Machine Learning for identification people coughing/sneezing. The main objective of this research is how to classify the sound of sneezing and coughing in the situation as difficult as the noise contained in the raw data. The development of artificial intelligence is quite rapid especially on sound classification using machine learning can help to distinguish coughs and sneezes based on voice. In this final project, the Long Short Term algorithm Memory (LSTM) is used because the algorithm is capable classify the sound pattern of a data. For produce a cough detection program the author makes 3 The main programs are Pre-Processing, Training, and Prediction. Through the method used, Machine Implementation Learning to identify people coughing/sneezing capable achieved an average value of 68.52% accuracy, 88.10% precision and recall 62.03%.

Keywords: Cough, LSTM, Machine Learning, Sneeze.

Abstrak

Hal terpenting pada *machine learning* adalah fitur ekstraksi dari sebuah data mentah. Pada jurnal ini akan dijelaskan mengenai implementasi *machine learning* untuk identifikasi orang batuk/bersin. Tujuan utama penelitian ini adalah bagaimana mengklasifikasi suara bersin dan batuk pada situasi yang sulit seperti noise yang terdapat pada data mentah. Perkembangan kecerdasan buatan yang cukup pesat terutama pada klasifikasi suara menggunakan machine learning dapat membantu untuk membedakan batuk dan bersin berdasarkan suara. Dalam tugas akhir ini, algoritma *Long Short Term Memory* (LSTM) digunakan karena algoritma ini mampu mengklasifikasikan pola suara dari suatu data. Untuk menghasilkan program pendeteksi batuk penulis membuat 3 program utama yaitu *Pre-Processing*, *Training*, dan *Prediction*. Melalui metode yang digunakan, implementasi *machine learning* untuk identifikasi orang batuk/bersin mampu mencapai nilai rata-rata akurasi 68,52%, presisi 88,10% dan recall 62,03%.

Kata Kunci: Batuk, Bersin, LSTM, *Machine Learning*.

I. PENDAHULUAN

Batuk merupakan mekanisme pertahanan tubuh di saluran pernapasan dan merupakan gejala suatu penyakit atau reaksi tubuh terhadap iritasi di tenggorokan karena adanya lendir, makanan, debu, asap dan sebagainya [1]. Pada seseorang yang mengalami batuk kronis, baik frekuensi dan intensitas batuk sangat penting untuk menentukan keparahan batuk yang dialami oleh orang tersebut secara umum [2]. Tingkat kekuatan batuk dapat diukur secara objektif berdasarkan aliran batuk yang keluar [3].

Bersin adalah keluarnya udara semi otonom yang terjadi dengan keras lewat hidung dan mulut. Udara ini dapat mencapai kecepatan 70 m/detik (250 km/jam). Bersin dapat menyebarkan penyakit lewat butir-butir air yang terinfeksi yang diameternya antara 0,5 hingga 5 μm . Sekitar 40.000 butir air seperti itu dapat dihasilkan dalam satu kali bersin [4].

Baik batuk maupun bersin ketika hal tersebut dialami oleh seseorang maka dua hal tersebut akan mengakibatkan keluarnya saliva dari dalam mulut, dimana di dalam saliva dapat mengandung berbagai macam jenis bakteri, virus, serta mikroorganisme lainnya yang dapat membahayakan tubuh. melalui pengukuran secara subjektif demikian muncul kebutuhan untuk menciptakan alat pendeteksi batuk untuk mengatasi penyebaran mikroorganisme yang berbahaya tersebut.

Dalam penelitian ini diharapkan penulis dapat mengembangkan sistem pendeteksi batuk dan bersin dengan menggunakan metode *machine learning* berbasis *python* untuk mengklasifikasikan jenis batuk dan bersin dengan menambahkan sistem penyemprotan berkala setiap 2 jam, serta penulis mencoba untuk menambahkan sistem IOT

sebagai sarana pengirim data dari komputer menuju aktuator yang akan digunakan.

II. KAJIAN PUSTAKA

A. *Machine Learning*

Machine Learning merupakan sebuah bentuk analisis dari suatu problem dengan menggunakan suatu model matematika. Model matematika digunakan untuk menangkap fitur yang menjadi kunci saat mesin sedang belajar, kemudian model matematika tersebut dapat menghasilkan sebuah solusi dari suatu permasalahan [5].

Sebagai contoh, untuk menentukan masalah dalam mendefinisikan sebuah proses kimia terhadap senyawa yang dihasilkan. Dengan menggunakan cara konvensional dimana hal tersebut membutuhkan seorang ahli kimia dengan mengaplikasikan pengetahuannya mengenai reaksi kimia untuk mendapatkan jawaban mengenai proses kimia terhadap senyawa yang dihasilkan.

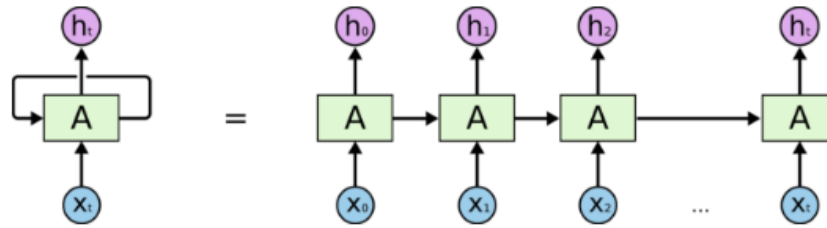
Contoh lainnya adalah bagaimana cara membuat desain dari *speech translation* atau kompresi algoritma gambar/video. Kedua tugas tersebut membutuhkan sebuah tim ahli seperti ahli bahasa, psikolog, dan teknisi dalam pemrosesan sinyal, dimana hal tersebut membutuhkan sebuah diskusi yang sangat panjang. Hal tersebut sangat berdampak pada penggunaan dana yang cukup besar serta tidak efisien [6].

B. *Long Short Term Memory Network (LSTM)*

Long short term memory network (LSTM) adalah salah satu modifikasi dari recurrent neural network atau RNN. Banyak modifikasi dari RNN, tetapi LSTM merupakan salah

satu yang populer di antaranya. LSTM hadir untuk melengkapi kekurangan RNN yang tidak dapat memprediksi kata berdasarkan informasi lampau yang disimpan dalam jangka waktu lama. Gbr 1 di bawah merupakan jaringan

LSTM (*Long Short Term Memory*) dimana di dalam sebuah jaringan LSTM terdiri dari berbagai macam sel yang di representasikan oleh huruf A dan X sebagai input serta h sebagai outputnya.



An unrolled recurrent neural network.

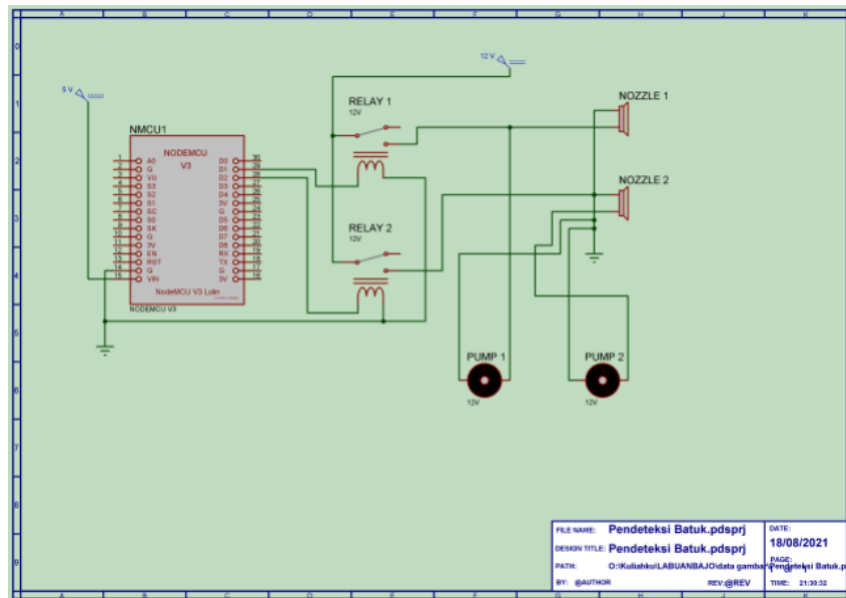
Gbr 1. Jaringan LSTM

Dengan demikian, LSTM mampu mengingat kumpulan informasi yang telah disimpan dalam jangka waktu panjang, sekaligus menghapus informasi yang tidak lagi relevan. LSTM lebih efisien dalam memproses, memprediksi, sekaligus mengklasifikasikan data berdasarkan urutan waktu tertentu.

III. METODE

A. Desain Elektrik.

Pembuatan model rangkaian elektrikal menggunakan software proteus. Dalam software proteus sudah tersedia berbagai macam library elektrik yang dapat digunakan diantaranya *library node MCU*, *relay*, motor dc dll.

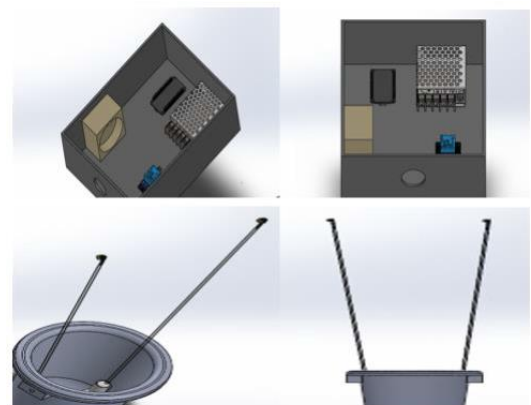


Gbr 2. Wiring Diagram

Di dalam perancangan elektrik ini terdapat Node MCU yang berfungsi sebagai penangkap data yang akan dikirimkan melalui broker MQTT, dengan menggunakan sumber 5VDC sebagai powernya. Kemudian digunakan 2 Output untuk mengirim 2 data *High* dan *Low* yang berbeda dimana pada D1 berfungsi untuk mengirim data dari wilayah 1 dan D2 berfungsi untuk mengirim data dari wilayah 2. Data *High* D1 dan D2 yang berupa tegangan 3 VDC selanjutnya diteruskan untuk mengaktifkan relay 12 VDC. Penggunaan relay 12 VDC berdasarkan kepada sumber tegangan yang dibutuhkan untuk mengaktifkan aktuator baik motor DC dan *nozzle*. Dalam perancangan *elektrikal nozzle* ini menggunakan *library* dari speaker karena *library* untuk *nozzle* sendiri tidak tersedia dalam aplikasi proteus yang digunakan.

B. Desain Mekanik

Desain perancangan mekanik menggunakan *software Solidwork*.



Gbr 3. Konstruksi Mekanik

Di dalam perancangan mekanik terdapat 1 buah panel box yang berfungsi sebagai pelindung dari komponen elektrik seperti: terminal listrik, *Power Supply*, *Node MCU*, dan *Relay 12VDC*. Kemudian terdapat ember sebagai wadah tempat menyimpan sanitizier yang akan di semprotkan. Terminal yang terdapat pada *panel box* berfungsi sebagai penghubung adaptor 5VDC, kemudian *power supply* sebagai sumber tegangan 12VDC. Sedangkan didalam ember yang berisi disinfektan terdapat 2 buah *fuel pump* yang berfungsi sebagai pompa yang akan mengalirkan disinfektan tersebut melalui *waterpass* menuju masing-masing *nozzle* berdasarkan wilayah.

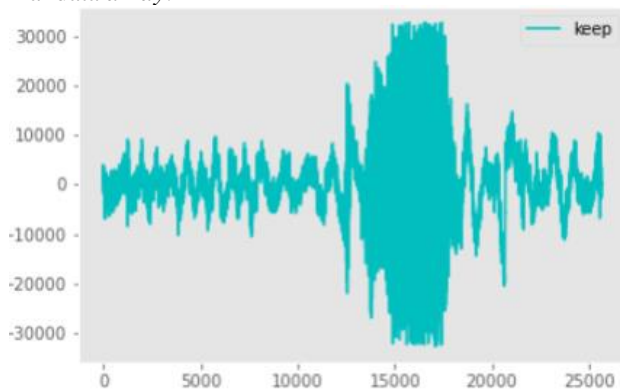
IV. HASIL DAN PEMBAHASAN

Dalam penelitian ini, program yang akan dibuat menggunakan python untuk menyelesaikan masalah dalam mengklasifikasi batuk atau bersin. Program yang dibuat akan diklasifikasikan kedalam 3 kategori yaitu: *Pre-processing*, *Training*, dan *Prediction*.

A. *Pre-processing*

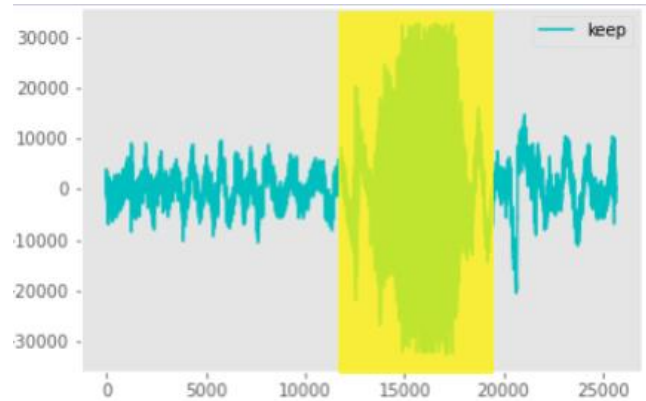
Pre-processing merupakan program awal yang dibuat untuk mengekstrak fitur-fitur yang terdapat pada dataset yang digunakan dimana hal ini bertujuan untuk memberikan label atau informasi ketika data set yang digunakan akan di *training*. Dalam fitur ekstraksi ini data set yang digunakan akan dibuang sebagian sinyalnya untuk menghilangkan noise.

Dalam menghilangkan *noise* yang dihasilkan maka perlu adanya *threshold* untuk mengatur seberapa besar sinyal yang akan dibuang, dimana dalam pengaturan nilai *threshold* ini dapat melihat dari bentuk sinyal yang dihasilkan karena sinyal yang akan diproses akan memiliki nilai *amplitude* yang berbeda dengan sinyal *noise* yang dihasilkan. Selanjutnya sinyal akan dipartisi menjadi potongan-potongan dengan rentan waktu 1 detik dan sinyal akan di *re-sample* ke dalam bentuk 16.000 *sample rate* yang berupa nilai data *array*.



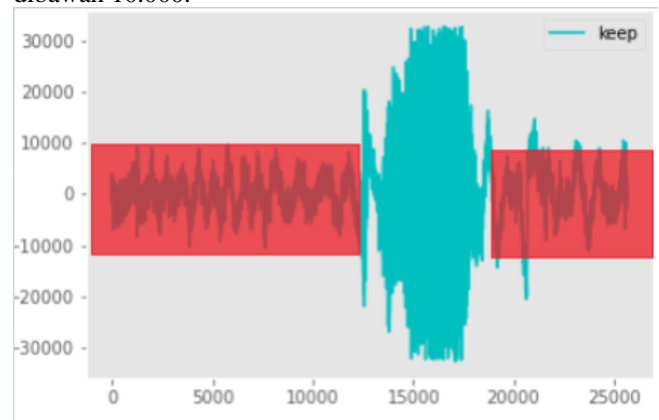
Gbr 4. Sinyal bersin

Pada gbr 4 di atas merupakan contoh sinyal bersin yang akan diproses. Dengan melihat sinyal tersebut kita dapat mengetahui karakteristik dari sinyal bersin yang memiliki nilai *amplitude* berbeda dengan *noise* yang dihasilkan.



Gbr 5. Asumsi Sinyal bersin

Dengan mengasumsikan area berwarna kuning merupakan sinyal bersin karena *amplitudanya* yang berbeda. Selanjutnya diatur nilai *threshold* sebesar 10.000. nilai *threshold* ini akan menghilangkan data yang memiliki nilai dibawah 10.000.



Gbr 6. Asumsi sinyal noise

Pada Gbr 6 area yang berwarna merah merupakan nilai *threshold* yang sudah diatur sebelumnya, seperti yang dapat dilihat area yang berwarna merah tersebut akan dihilangkan karena kita asumsikan sinyal tersebut sebagai *noise*.



Gbr 7. Sinyal bersin setelah dipre-processing

Gbr 7 di atas merupakan bentuk sinyal yang sudah di aplikasikan nilai *thresholdnya*. Data ini merupakan sinyal bersin yang akan diproses oleh mesin untuk di *training*. Selanjutnya data akan di partisi kedalam 1 detik, hal ini bertujuan untuk menyetarakan waktu yang digunakan. Setelah dipartisi maka data akan di *sampling* ulang kedalam 16.000 *sample rate* berdasarkan *library tensorflow* yang

digunakan. Terakhir data tersebut akan dikonversi ke dalam bentuk data *array*.

B. Training

Program *training* data berfungsi untuk melatih data set yang sudah siap untuk *di-training* melalui proses *Pre-*

Processing dengan tujuan komputer mampu melakukan komputasi dengan menggunakan data hasil training yang sudah disimpan.

	Standard	Standard	Standard	Standard	Standard
1	epoch	accuracy	loss	val_accuracy	val_loss
2	0	0.7653061151504517	0.5803020000457764	0.8999999761581421	0.32867664098739624
3	1	0.9285714030265808	0.29130321741104126	1.0	0.1564299762248993
4	2	0.9693877696990967	0.14992168545722961	1.0	0.14058755338191986
5	3	0.9846938848495483	0.11175244301557541	0.8999999761581421	0.3271884322166443
6	4	0.9897959232330322	0.09686050564050674	0.949999988079071	0.14583246409893036
7	5	1.0	0.05539326369762421	1.0	0.09130927175283432
8	6	1.0	0.045840948820114136	1.0	0.07414272427558899

Gbr 8. Hasil Training

Pada saat melakukan training sistem akan menampilkan 5 poin utama yaitu: epoch, accuracy, val_accuracy, loss, dan val_loss. Epoch merupakan jumlah iterasi yang dilakukan pada saat *training* data set dimana epoch 0 yang ada pada Gbr 8 berarti iterasi ke-1. Disini sistem akan melakukan *training* menggunakan 1 data set sebagai *sample* uji dan sisanya sebagai data *training*, untuk epoch 1 atau iterasi ke-2 sistem akan kembali mengambil 1 data set sebagai *sample* uji yang berbeda dari data sebelumnya dan data set lainnya akan digunakan sebagai data *training*. Epoch yang digunakan saat *training* berjumlah 100 namun hanya terjadi 8 epoch atau 8 kali.

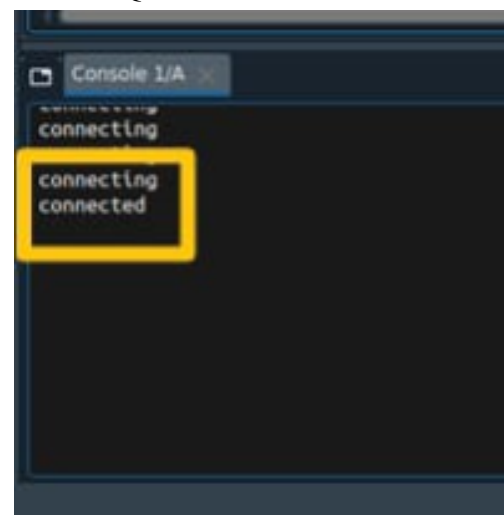
Accuracy adalah matrix yang digunakan dalam klasifikasi. Dalam *accuracy* akan ditampilkan persentase hubungan hasil *training* menurut sistem dengan skala 0 sampai 1, dimana 0 berarti 0% dan 1 berarti 100%. Sedangkan Val_accuracy adalah representasi dari performa model neural network terhadap validasi data, maka ketika nilai val_accuracy tidak meningkat atau berada dalam nilai 1 secara terus menerus maka sistem akan menganggap training sudah mencapai maksimal. Training untuk sistem ini menghasilkan nilai accuracy dan val_accuracy sebesar 1 atau 100%.

Loss adalah nilai training data yang dibuang ketika melakukan training data dikarenakan sistem tidak mengenali data tersebut dan menganggap data tersebut tidak penting. Sedangkan Val_loss merupakan data yang terbuang saat validasi data pada proses training. Sama seperti Loss, Val_loss membuang data yang dianggap tidak penting. Maka dari itu dapat dikatakan nilai loss dan Val_loss harus semakin kecil dari setiap iterasi, hal ini menandakan bahwa sistem bekerja dengan baik karena persentase data yang

dibuang semakin kecil karena sistem semakin mampu untuk mengenali data yang dibutuhkan. Nilai loss terakhir yang dihasilkan pada sistem ini sebesar 4%. Hal ini menandakan jumlah data yang dibuang, sedangkan nilai terakhir val_loss untuk sistem ini sebesar 7% untuk pembuangan data saat validasi.

C. Prediction

Program prediksi memiliki bertujuan untuk melakukan keputusan dalam menentukan hasil keluaran. Ketika program running program akan menampilkan user interface yang sudah dibuat untuk dioperasikan dan kernel/command box akan menampilkan proses koneksi antara komputer dengan broker MQTT.



Gbr 9. Kernel terkoneksi

Gbr 9 di atas menunjukkan bahwa komputer telah berhasil terkoneksi dengan broker MQTT, dan program telah siap dijalankan. Untuk menjalankan program dapat

dioperasikan menggunakan user interface, dimana kita dapat memilih metode manual atau otomatis.

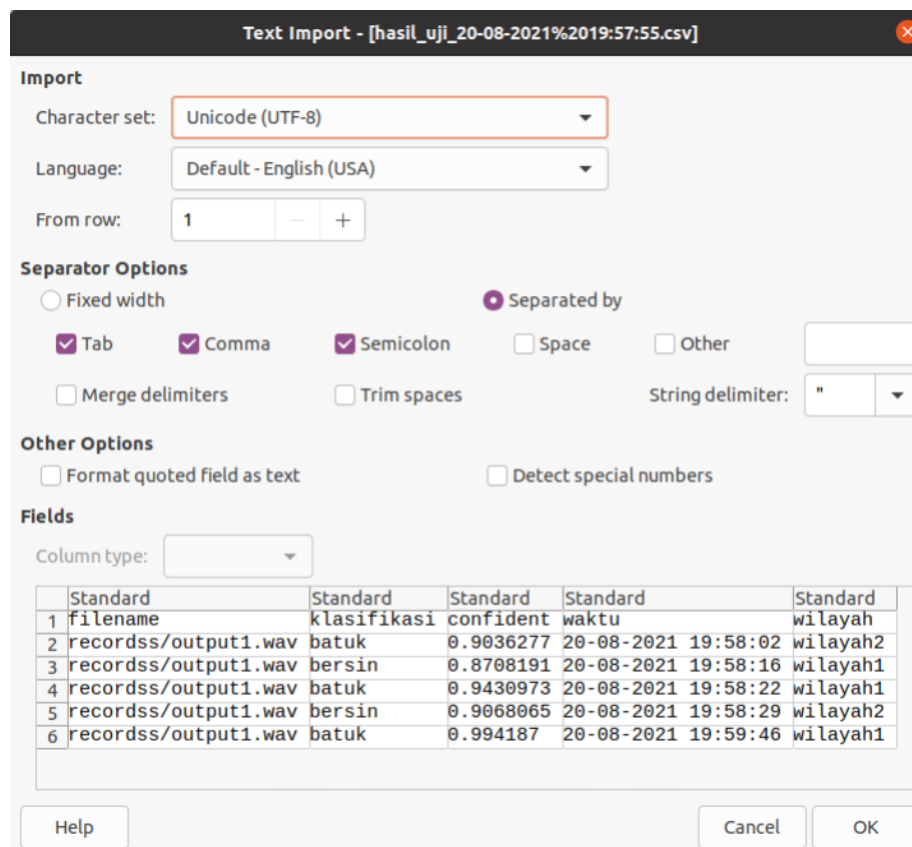
```

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-08-23 13:16:51.786642: I tensorflow/compiler/jit/xla_gpu_device.cc:99] Not creating XLA devices,
tf_xla_enable_xla_devices not set
2021-08-23 13:16:57.306447: I tensorflow/compiler/mlir/nlir/nlir_graph_optimization_pass.cc:116] None of the MLIR
optimization passes are enabled (registered 2)
2021-08-23 13:16:57.439865: I tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency: 2096155000 Hz
recordss/output1.wav, Hasilnya adalah: background confident : 0.8728536
recording...
recordss/output1.wav, Hasilnya adalah: background confident : 0.8695472
recording...
recordss/output1.wav, Hasilnya adalah: background confident : 0.95085454
recording...
recordss/output1.wav, Hasilnya adalah: background confident : 0.9533265
recording...
  
```

Gbr 10. Prediksi pada kernel

Selanjutnya mode otomatis yang dijalankan akan ditampilkan pada kernel untuk memberi informasi bahwa sistem melakukan perekaman suara seperti pada Gbr 10 di atas tampilan kernel/console dapat di gunakan untuk memastikan program telah berjalan. Pada gambar 10 juga menampilkan bahwa program telah berjalan dan akan

melakukan pendeteksian secara otomatis. Kemudian apabila sistem mendeteksi objek yang dianggap sebagai batuk atau bersin maka data tersebut akan dimasukkan kedalam sebuah file yang berbentuk '.csv' sebagai riwayat pendeteksian yang terjadi.



Gbr 11. Riwayat pendeteksian

Gbr 11 di atas merupakan tampilan riwayat pendeteksian dalam format csv, dimana informasi yang terdapat diantaranya filename recording atau nama data yang digunakan selama merakam suara yang kemudian akan di klasifikasikan. Kemudian ada klasifikasi terhadap suara yang dideteksi diantaranya batuk dan bersin. Selanjutnya ada confident merupakan persentase seberapa yakin sistem mengklasifikasikan data tersebut. Dan terakhir ada waktu

dan wilayah, untuk wilayah dibedakan menjadi 2 wilayah diantaranya wilayah 1 dan wilayah 2.

V. KESIMPULAN

Implementasi *machine learning* untuk identifikasi orang batuk/bersin menggunakan algoritma LSTM telah berhasil dilakukan. Melalui percobaan menggunakan dataset dan pengujian sendiri didapatkan nilai rata-rata akurasi

sebesar 75% (lebih rendah dibandingkan hasil penelitian Wu Dan & F. Weninger and B. Schuller) [7][8], presisi 69,36%, dan recall 95%. Untuk percobaan menggunakan dataset gabungan [9] dan percobaan terhadap orang yang berbeda melalui jarak maksimal 4 meter didapatkan nilai rata-rata akurasi sebesar 68,52%, presisi 88,10% dan recall 62,03%.

UCAPAN TERIMA KASIH

Kami ucapan terima kasih kepada tim redaksi Journal of Energy and Electrical Engineering yang telah bersedia menampung tulisan ini, serta Politeknik Manufaktur Bandung yang telah mendukung setiap penelitian yang dibuat.

REFERENSI

- [1] "Batuk." <https://id.wikipedia.org/wiki/Batuk>.
- [2] Kai K Lee, Sergio Matos, Katie Ward, Gerrard F Rafferty, John Moxham and S. S. B. David H Evans, "Sound: a non-invasive measure of cough intensity," 2017.
- [3] A. Jasmine and A. K. Jayanthi, "Sensor-based system for automatic cough detection and classification," Test Eng. Manag., vol. 83, no. October 2015, pp. 13826–13834, 2015.
- [4] "Bersin," [Online]. Available: <https://id.wikipedia.org/wiki/Bersin>.
- [5] Hasibuan, F. C., & Rahayu, A. U. (2022). Identifikasi Persediaan Makanan di dalam Lemari Pendingin Berbasis Raspberry Pi dan Deep Learning. Electrician: Jurnal Rekayasa dan Teknologi Elektro, 16(1), 94-101.
- [6] O. Simeone, "A brief introduction to machine learning for engineers," Found. Trends Signal Process., vol. 12, no. 3–4, pp. 200–431, 2018, doi:10.1561/2000000102.
- [7] D. Wu, "An Audio Classification Approach Based on Machine Learning," Proc. - 2019 Int. Conf. Intell. Transp. Big Data Smart City, ICITBS 2019, pp. 626–629, 2019, doi: 10.1109/ICITBS.2019.00156.
- [8] F. Weninger and B. Schuller, "AUDIO RECOGNITION IN THE WILD: STATIC AND DYNAMIC CLASSIFICATION ON A REAL-WORLD DATABASE OF ANIMAL VOCALIZATIONS Felix Weninger and Bj " Institute for Human-Machine Communication, Technische Universitat Munchen 80290 Munchen, Germany," Ger. Res., pp. 337–340, 2011.
- [9] E. C. Knight and E. M. Bayne, "Classification threshold and training data affect the quality and utility of focal species data processed with automated audio-recognition software," Bioacoustics, vol. 28, no. 6, pp. 539–554, 2019, doi: 10.1080/09524622.2018.1503971.

BIOGRAFI PENULIS



Sandy Bhawana Mulia, Lahir di Bandung, bekerja sebagai tenaga pengajar di Program Studi Teknologi Rekayasa Mekatronika Politeknik Manufaktur Bandung dengan bidang penelitian Kontrol, *Power System, IoT, Machine Learning*.



Nur Wisma Nugraha, Lahir di Magetan, bekerja sebagai tenaga pengajar di Program Studi Teknologi Rekayasa Mekatronika Politeknik Manufaktur Bandung dengan bidang penelitian Kontrol.



Muhammad Hanif Robbani, Lahir di Cimahi, lulusan tahun 2021 pada Program Diploma 4 Teknologi Rekayasa Otomasi Politeknik Manufaktur Bandung.