

DAFTAR PUSTAKA

- [1] Yudha, P., 2020. Analisis Kelayakan Bisnis Usaha Air Minum Dalam Kemasan (Studi Kasus Pada Bum Desa Lancar Jaya Desa Lambur Kecamatan Mrebet Kabupaten Purbalingga) (Doctoral dissertation, IAIN Purwokerto).
- [2] Mambela, S., 2018. Tinjauan umum masalah psikologis dan masalah sosial individu penyandang tunanetra. Buana Pendidikan: Jurnal Fakultas Keguruan dan Ilmu Pendidikan Unipa Surabaya, 14(25), pp.65-73.
- [3] Siahaan, M., Jasa, C.H., Anderson, K., Valentino, M., Lim, S. and Yudianto, W., 2020. Penerapan Artificial Intelligence (AI) Terhadap Seorang Penyandang Disabilitas Tunanetra. Journal Of Information System And Technology, 1(2), pp.186-193.
- [4] Rizki, P., 2021. Rancang Bangun Sistem Alarm Pengisian Air Minum Untuk Penyandang Tunanetra Berbasis Arduino (Doctoral dissertation, Universitas Widya Dharma Klaten).
- [5] Singgeta, R.L. and Rumondor, R., 2018. Rancang bangun dispenser otomatis menggunakan sensor ultrasonik berbasis mikrokontroller atmega2560. Jurnal Ilmiah Realtech, 14(1), pp.31-36.
- [6] Febriandirza, A. and Sahuri, A.A.G.A., 2021. Rancang Bangun Dispenser Otomatis Untuk Tunanetra Berbasis Microcontroller. Pseudocode, 8(2), pp.143-152.
- [7] Furui S. 2001. Digital Speech Processing, Synthesis and Recognition. Marcel Dekker Inc. New York.
- [8] Rabiner L., Juang B.H. 1993, Fundamental of Speech Recognition. Prentice-Hall Inc. New Jersey.
- [9] Turang, D. A. O. (2015, December). Pengembangan Sistem Relay Pengendalian Dan Penghematan Pemakaian Lampu Berbasis Mobile. In Seminar Nasional Informatika (SEMNASIF). Vol. 1, No. 1.
- [10] Satya, T. P., Al Fauzan, M. R., & Admoko, E. M. D. (2019). Sensor ultrasonik HCSR04 berbasis arduino due untuk sistem monitoring ketinggian. JFA (Jurnal Fisika dan Aplikasinya), 15(2), 36-39.
- [11] Fauziah, F., Tritoasmoro, I.I. and Rizal, S., 2021. Sistem Keamanan Berbasis Pengenalan Suara Sebagai Pengakses Pintu Menggunakan Metode Mel Frequency Cepstral Coefficient (mfcc). eProceedings of Engineering, 8(6).
- [12] N. Dave, "Feature Extraction Methods LPC , PLP and MFCC In Speech Recognition," Int. J. Adv. Res. Eng. Technol., vol. 1, no. VI, pp. 1–5, 2013.
- [13] A. F. Permana, "Implementasi Hidden Markov Model Untuk Aplikasi Pengenalan Ucapan Sebagai Kendali Gerak Robot Mobil," Institut Pertanian Bogor, 2011.
- [14] S. C. Joshi and A. N. Cheeran, "MATLAB Based Feature Extraction Using Mel Frequency Cepstrum Coefficients for Automatic Speech Recognition," Int. J. Sci. Eng. Technol. Res., vol. 3, no. 6, pp. 1820–1823, 2014.

- [15] A. Mustofa, "Sistem Pengenalan Penutur dengan Metode Mel-Frequency Wrapping," *J. Tek. Elektro*, vol. 7, no. 2, pp. 88–96, 2007.
- [16] L. R. Mehta and A. S. Dabhade, "Comparison Study of MFCC and LPC for Marathi Isolated Word," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 2, no. 6, pp. 2133–2139, 2013.
- [17] V. Tiwari, "MFCC and Its Applications in Speaker Recognition," *Int. J. Emerg. Technol.*, vol. 1, no. 1, pp. 19–22, 2010.
- [18] Sanjaya, M. and Salleh, Z., 2014. Implementasi Pengenalan Pola Suara Menggunakan Mel-Frequency Cepstrum Coefficients (Mfcc) Dan Adaptive Neuro-Fuzzy Inferense System (Anfis) Sebagai Kontrol Lampu Otomatis. *ALHAZEN Journal of Physics*, 1(1), pp.43-54.
- [19] Data Sheet Arduino Nano 33 BLE Sense. <https://docs.arduino.cc/hardware/nano-33-ble-sense> (3 Februari 2023)
- [20] Perancangan Alat Ukur Kecepatan Kendaraan Menggunakan ATmega 16. Olivia . *E-Journal Teknik Elektro dan Komputer* (2015), ISSN : 2301-8402
- [21] Hymel, S., Banbury, C., Situnayake, D., Elium, A., Ward, C., Kelcey, M., ... & Reddi, V. J. (2022). Edge Impulse: An MLOps Platform for Tiny Machine Learning. arXiv preprint arXiv:2212.03332.
- [22] Suprianto, "Pengertian dan Prinsip Kerja *Solenoid valve*". (Semarang: Dukuh Kemplang, 2015). hal.1.
- [23] Romi Shaputra, dkk., "Kran Air Otomatis Pada Tempat Berwudhu Menggunakan Sensor Ultrasonik Berbasis Arduino Uno". *Jurnal Sigma Teknika*, Vol.2, No.2 : 192-201, 2021. hal.195.
- [24] Adipura, "Katup Otomatis *Solenoid valve*". (Kebumen: Adipura Publisher, 2014). hal.14. (1 Februari 2023)
- [25] <https://electronic-info.eu/2018081207/mp34dt05-a-mems-digital-microphone>
- [26] Tri, M., Jaya, S., Puspitaningrum, D., & Susilo, B. (2016). PENERAPAN SPEECH RECOGNITION PADA PERMAINAN TEKA-TEKI SILANG MENGGUNAKAN METODE HIDDEN MARKOV MODEL (HMM) BERBASIS DESKTOP. *Jurnal Rekursif*, 4(1).
- [27] Sutikno, S., Anam, K., & Shaleh, A. (2021). Sistem Navigasi Kursi Roda Elektrik untuk Pasien Penyandang Cacat Fisik Menggunakan Metode *Convolutional Neural Network*. *Jurnal Rekayasa Elektrika*, 17(1), 29–35. <https://doi.org/10.17529/jre.v17i1.16376>
- [28] Irfan Reza, M., Rizqika Akbar, S., & Fitriyah, H. (2018). *Kontrol Lampu Berbasis Voice Command Pada Raspberry PI* (Vol. 2, Issue 9). <http://j-ptiik.ub.ac.id>
- [29] Karyadi Kusuma, A., & Dewanta, F. (n.d.). *IMPLEMENTATION OF AUTOMATIC FILLING SMART DISPENSER USING DATABASE AND WEB SERVER WITH IOT BASED*.
- [30] Fitri, Y., Retnawaty, F., & Zecha, N. (2018). FREQUENCY CEPSTRUM COEFFICIENTS (MFCC) DAN JARINGAN SYARAF TIRUAN (JST) PROPAGASI BALIK. In *Jurnal Photon* (Vol. 8, Issue 2).

- [31] Santoso, S., Hartayu, R., Anam, C., & Dimas Abdul Aziz. (2022). Simulasi Simulasi Ekstraksi Fitur Suara menggunakan Mel-Frequency Cepstrum Coefficient. *Jurnal Sains Dan Informatika*, 8(1), 80–87. <https://doi.org/10.34128/jsi.v8i1.357>



LAMPIRAN

Lampiran 1. Data Hasil Pengujian Respon Waktu Sistem *Smart Dispenser* Tunanetra dari masing-masing perintah

PANAS		DINGIN		CAMPUR	
Pengujian Ke	Waktu Respon (s)	Pengujian Ke	Waktu Respon (s)	Pengujian Ke	Waktu Respon (s)
1	1.01	1	1.36	1	1.56
2	1.37	2	1.37	2	1.45
3	1.18	3	1.11	3	1.29
4	1.23	4	1.08	4	1.44
5	1.43	5	1.23	5	1.18
6	1.23	6	1.05	6	1.06
7	1.56	7	1.42	7	1.27
8	1.33	8	1.12	8	1.48
9	1.35	9	1.47	9	1.13
10	1.36	10	1.02	10	1.40
Rata-Rata	1.31	Rata-Rata	1.22	Rata-Rata	1.33

Lampiran 2. Dokumentasi Pengujian *Smart Dispenser* Tunanetra



Lampiran 3. Kode Program Python

```
1 import tensorflow as tf
2 from tensorflow.keras.models import Sequential
3 from tensorflow.keras.layers import Dense, InputLayer, Dropout, Conv1D, Conv2D, Flatten,
  Reshape, MaxPooling1D, MaxPooling2D, AveragePooling2D, BatchNormalization,
  TimeDistributed, Permute, ReLU, Softmax
4 from tensorflow.keras.optimizers import Adam
5 # Data augmentation for spectrograms, which can be configured in visual mode.
6 # To learn what these arguments mean, see the SpecAugment paper:
7 # https://arxiv.org/abs/1904.08779
8 sa = SpecAugment(spectrogram_shape=[int(input_length / 20), 20], mF_num_freq_masks=0,
  F_freq_mask_max_consecutive=0, mT_num_time_masks=1, T_time_mask_max_consecutive=1,
  enable_time_warp=False, W_time_warp_max_distance=6, mask_with_mean=False)
9 train_dataset = train_dataset.map(sa.mapper(), num_parallel_calls=tf.data.AUTOTUNE)
10
11 EPOCHS = args.epochs or 100
12 LEARNING_RATE = args.learning_rate or 0.005
13 # this controls the batch size, or you can manipulate the tf.data.Dataset objects yourself
14 BATCH_SIZE = 32
15 train_dataset = train_dataset.batch(BATCH_SIZE, drop_remainder=False)
16 validation_dataset = validation_dataset.batch(BATCH_SIZE, drop_remainder=False)
17
18 # model architecture
19 model = Sequential()
20 # Data augmentation, which can be configured in visual mode
21 model.add(tf.keras.layers.GaussianNoise(stddev=0.45))
22 model.add(Reshape((int(input_length / 20), 20), input_shape=(input_length, )))
23 model.add(Conv1D(8, kernel_size=3, padding='same', activation='relu'))
24 model.add(MaxPooling1D(pool_size=2, strides=2, padding='same'))
25 model.add(Dropout(0.25))
26 model.add(Conv1D(16, kernel_size=3, padding='same', activation='relu'))
27 model.add(MaxPooling1D(pool_size=2, strides=2, padding='same'))
28 model.add(Dropout(0.25))
29 model.add(Flatten())
30 model.add(Dense(classes, name='y_pred', activation='softmax'))
31
32 # this controls the learning rate
33 opt = Adam(learning_rate=LEARNING_RATE, beta_1=0.9, beta_2=0.999)
34 callbacks.append(BatchLoggerCallback(BATCH_SIZE, train_sample_count, epochs=EPOCHS))
35
36 # train the neural network
37 model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
38 model.fit(train_dataset, epochs=EPOCHS, validation_data=validation_dataset, verbose=2,
  callbacks=callbacks, class_weight=ei_tensorflow.training.get_class_weights(Y_train))
39
40 # Use this flag to disable per-channel quantization for a model.
41 # This can reduce RAM usage for convolutional models, but may have
42 # an impact on accuracy.
43 disable_per_channel_quantization = False
```

26%

SIMILARITY INDEX

25%

INTERNET SOURCES

7%

PUBLICATIONS

14%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Nasional Student Paper	3%
2	repository.ar-raniry.ac.id Internet Source	2%
3	docplayer.info Internet Source	2%
4	journal.wima.ac.id Internet Source	1%
5	journal.uinsgd.ac.id Internet Source	1%
6	repository.its.ac.id Internet Source	1%
7	Submitted to Sriwijaya University Student Paper	1%
8	robocraze.com Internet Source	1%
9	lib.ui.ac.id Internet Source	1%