Studi Fitur Text to Speech pada Penerjemah Bahasa Isyarat BISINDO di Raspberry Pi 5

Jonathan Tanaka[1], Olfat Harits Alatas[2], Romy Budhi Widodo[3]
Human-Machine Interaction Research Center-Teknik Informatika, Universitas Ma Chung
e-mail: 312310023@student.machung.ac.id [1], 312210018@student.machung.ac.id [2], romy.budhi@machung.ac.id [3]

Kata Kunci:

BISINDO
Raspberry Pi
Pyttsx3
Text-to-Speech
Multithreading

ABSTRAK

Komunikasi antara masyarakat dengar dan penyandang tunarungu pengguna Bahasa Isyarat Indonesia masih terkendala karena sistem penerjemah BISINDO berbasis Raspberry Pi hasil penelitian sebelumnya hanya menghasilkan keluaran teks pada layar, sehingga kurang praktis digunakan untuk komunikasi dua arah secara langsung antara kedua belah pihak. Penelitian ini bertujuan menyempurnakan sistem tersebut dengan menambahkan fitur *Text-to-Speech* agar hasil klasifikasi gestur dapat langsung didengar dalam bentuk suara. Fitur *Text-to-Speech* dikembangkan menggunakan *library* pyttsx3 yang diintegrasikan dengan *engine* eSpeak dan *synthesizer* MBROLA, serta diterapkan teknik *multithreading* agar proses sintesis suara tidak menghambat tangkapan *frame* kamera dan klasifikasi gestur secara *real-time*. Pengujian dilakukan melalui pengujian mandiri terhadap 141 gestur BISINDO dan pengujian oleh responden yang memperagakan seluruh 141 gestur dengan rancangan *Latin Square*, dinilai dari keberhasilan pengenalan dan kejelasan suara. Hasil pengujian mencatat keberhasilan 90,07% pada pengujian mandiri dan 89,83% pada pengujian responden, dengan 96,58% keluaran suara dinilai jelas.

Keyword:

BISINDO
Raspberry Pi
Pyttsx3
Text-to-Speech
Multithreading

ABSTRACT

Communication between hearing people and deaf individuals who use Indonesian Sign Language remains constrained because the Raspberry Pi based BISINDO translator from previous research only produces text output on screen, making it less practical for direct two-way communication between both parties. This research aims to improve that system by adding a Text-to-Speech feature so gesture classification results can be heard directly as speech. The Text-to-Speech feature was developed using the pyttsx3 library integrated with the eSpeak engine and the MBROLA synthesizer, with multithreading applied so the speech synthesis process does not hinder camera frame capture and real-time gesture classification. Testing was carried out through independent testing on 141 BISINDO gestures and testing by respondents who performed all 141 gestures following a Latin Square design, assessed on recognition success and voice clarity. Results recorded a 90.07% success rate in independent testing and 89.83% in respondent testing, with 96.58% of the voice output rated clear.

Copyright © SENASIF 2026
ISSN: 2598-0076

I. PENDAHULUAN

Berkomunikasi salah satu hak dasar bagi setiap warga negara tanpa terkecuali penyandang tunarungu dan tunawicara. Penyandang tunarungu dan tunawicara di Indonesia dapat berkomunikasi melalui dua bahasa visual yaitu Sistem Isyarat Bahasa Indonesia atau SIBI yang dibentuk secara resmi oleh pemerintah Indonesia dan dipergunakan dalam lingkup pendidikan dan Bahasa Isyarat Indonesia atau BISINDO yang dibuat langsung oleh komunitas tunarungu melalui Gerakan untuk Kesejahteraan Tunarungu Indonesia (GERKATIN) [1]. Terdapat penelitian terhadap mahasiswa tunarungu menunjukkan Bahasa Isyarat Indonesia atau BISINDO lebih sering dipakai dan para mahasiswa merasa lebih efektif dibandingkan menggunakan bahasa isyarat SIBI dalam penggunaan kehidupan sehari-hari [2], karena BISINDO ini melekat pada identitas dan citra sosial penggunanya [3]. Oleh karena itu, hal tersebut menjadi alasan mengapa Bahasa Isyarat Indonesia atau BISINDO dibahas dalam pengembangan sistem penerjemah bahasa isyarat pada penelitian ini.

Pengembangan sistem penerjemah BISINDO di Program Studi Teknik Informatika Universitas Ma Chung dikembangkan secara bertahap. Awalnya diinisiasi dengan menggunakan sarung tangan dengan sensor Flexpoint USB Glove Kit untuk mengklasifikasikan bahasa isyarat angka [4], namun penelitian tersebut terhambat karena *software* bawaan sarung tangan tidak lagi mendapat pembaruan. Lalu dikembangkan lebih lanjut dengan beralih ke pendekatan berbasis kamera dengan model klasifikasi hybrid, yaitu Random Forest untuk gestur statis serta Transformer dan LSTM untuk gestur dinamis yang dijalankan pada Raspberry Pi 5 dengan Pi Camera v1 [5]. Penelitian tersebut sudah terbukti dapat mengenali gestur, tetapi *output*-nya masih berupa teks di layar sehingga lawan bicara akan sering mengalihkan fokusnya antara layar dan pengguna bahasa isyarat hal tersebut membuat komunikasi menjadi kurang praktis dan efisien.

Oleh karena itu, penulis meningkatkan sistem tersebut dengan menambahkan fitur *Text-to-Speech* (TTS) menggunakan *library* bawaan Python yaitu pyttsx3 dan juga memakai *engine* eSpeak dan diperlengkapi *synthesizer* MBROLA sebagai *output* suara dengan logat Bahasa Indonesia. Dan juga penulis menggunakan teknik *multithreading* agar proses TTS tidak mengganggu performa pengambilan *frame* kamera secara real-time. Tujuan dari menambahkan fitur *Text-to-Speech* yaitu membuat hasil klasifikasi gestur dapat langsung didengar oleh lawan bicara yang dimana dapat memberikan manfaat antara penyandang tunarungu dan lawan bicaranya untuk dapat berkomunikasi tanpa harus saling memandang layar.

II. METODE

Penelitian ini menggunakan kembali *hardware* dan *software* dari penelitian sebelumnya yaitu Raspberry Pi 5 dengan spesifikasi RAM 16 GB yang terhubung dengan Pi Camera v1, beserta *virtual environment* Python yang telah berisi model *Random Forest* dan model *TensorFlow Lite* untuk gestur dinamis [5]. Raspberry Pi 5 digunakan karena dapat menjalankan beban komputasi

Copyright © SENASIF 2026
ISSN: 2598-0076

menengah hingga berat untuk *machine learning* dan *computer vision* berkat prosesor quad-core ARM Cortex-A76 2,4 GHz [6], dan Raspberry Pi 5 ini sudah banyak dimanfaatkan dalam penelitian *embedded* lain [7]. Penggunaan TensorFlow Lite disistem ini juga dipertahankan karena dapat meringankan model *machine learning* melalui teknik *quantization* sehingga dapat dijalankan pada perangkat dengan sumber daya terbatas [8].



Gambar 1. Arsitektur Sistem Keseluruhan

Arsitektur sistem secara keseluruhan ditunjukkan pada Gambar 1. Dimulai dari Pi Camera v1 menangkap *frame* tangan pengguna secara real-time, kemudian dilanjutkan ke tahap ekstraksi *landmark* tangan menggunakan *library MediaPipe* yang menghasilkan 21 titik koordinat pada tangan. Titik koordinat tersebut menjadi *input* bagi tahap klasifikasi gestur, di mana model *Random Forest* menangani gestur statis dan model *Transformer* menangani gestur dinamis, sehingga menghasilkan teks sesuai gestur yang dikenali. Teks ini kemudian masuk ke fitur *Text-to-Speech* yang dibuat menggunakan *library* pyttsx3 dengan *engine* eSpeak dan *synthesizer* MBROLA, sebelum akhirnya keluar sebagai suara melalui *speaker* eksternal.

Sebelum fitur *Text-to-Speech* bekerja, sistem melewati tahap deteksi dan klasifikasi gestur yang telah dirancang dari penelitian sebelumnya [5]. Pi Camera v1 menangkap *frame* video secara berkelanjutan, lalu MediaPipe mengekstrak titik-titik *landmark* tangan dari setiap *frame* tersebut. Titik-titik *landmark* ini menjadi masukan bagi dua model klasifikasi yang bekerja sesuai jenis gesturnya yaitu *Random Forest* menangani gestur statis seperti abjad dan angka, sedangkan *Transformer* menangani gestur dinamis yang melibatkan gerakan berurutan. Begitu salah satu model mengklasifikasikan gestur menjadi teks, teks tersebut diteruskan ke fitur TTS yang dikembangkan pada penelitian ini untuk diucapkan. Alur ini berulang pada setiap *frame*, sehingga pengguna dapat membaca teks di layar sekaligus mendengar suara hasil terjemahan hampir bersamaan dengan gestur yang sedang diperagakan.

Dengan mengawali penelitian ini dapat melakukan persiapan dengan meng-*install engine* eSpeak pada *hardware* yang kita gunakan yaitu Raspberry Pi 5, melalui package manager *apt* yaitu dengan *command* "*sudo apt install espeak*" dan *library* pyttsx3 melalui pip dengan *command "pip*

Copyright © SENASIF 2026
ISSN: 2598-0076

install pyttsx3" pada *virtual environment* yang sama dengan sistem klasifikasi yang sudah dibuat oleh penelitian sebelumnya. Lalu pada *synthesizer* MBROLA tidak memiliki paket biner resmi untuk arsitektur ARM sehingga di-*install* dengan mengompilasi kode sumber dari repositori resminya, kemudian basis data suara Bahasa Indonesia id1 ditempatkan pada direktori */usr/share/mbrola/id1* agar dikenali *engine* eSpeak melalui penamaan suara berawalan "*mb-*" [9][10]. Dalam program yang sudah dibuat, mesin TTS dari *library* pyttsx3 diinisiasikan sekali saja di awal program supaya tidak membebani proses inisialisasi yang berulang. Lalu diatur juga kecepatan bicara dengan memperlambat dari nilai standar, karena pada saat dicoba dengan nilai standarnya suara masih terdengar cepat dan diatur juga basis data suara yaitu dengan menggunakan Bahasa Indonesia. Pengaturan ini sejalan dengan penerapan pyttsx3 pada sistem komunikasi berbasis suara nyata [11], termasuk pada komunikasi dua arah manusia robot yang berbahasa Indonesia menggunakan fitur TTS yang serupa [12]. Program juga dapat mengoreksi pengucapan untuk abjad Bahasa Indonesia, karena mesin TTS secara standar membunyikan huruf sebagai bunyi fonetik, bukan sebagai nama huruf sesuai ejaan Bahasa Indonesia misalnya, huruf 'B' akan otomatis diucapkan "Bé" bukan sekadar bunyi "b". Kemudian *Synthesizer* MBROLA dipakai pada penelitian ini sebagai alat bantu tambahan pada kualitas suara yang dikeluarkan karena kualitas suara *diphone* yang dihasilkan lebih natural dibandingkan *formant synthesis* bawaan eSpeak, hal ini juga terbukti pada penelitian dengan adaptasi MBROLA untuk Bahasa Arab [13].



Gambar 2. Alur Kerja Fitur Text-to-Speech

Alur kerja fitur *Text-to-Speech* secara lebih rinci ditunjukkan pada Gambar 2. Teks hasil klasifikasi gestur yang dikirim dari model klasifikasi pertama kali diterima oleh *library* pyttsx3, yang bertugas mengatur *rate* atau kecepatan bicara, volume suara, dan bahasa yang dipakai. Selanjutnya pyttsx3 meneruskan teks tersebut ke *engine* eSpeak untuk diubah menjadi fonem beserta informasi *pitch*. Fonem hasil pemrosesan eSpeak kemudian digabungkan oleh MBROLA melalui teknik penggabungan *diphone* menjadi satu kesatuan suara yang utuh. Suara hasil sintesis ini pada akhirnya diputar melalui *speaker* eksternal sehingga terjemahan gestur BISINDO dapat didengar langsung oleh lawan bicara.

Copyright © SENASIF 2026

ISSN: 2598-0076

Pada pemrosesan sintesis suara berjalan lambat dan bersifat *blocking* saat diminta mengucapkan sesuatu yang dimana program akan berhenti dan menunggu sampai ucapan selesai sebelum bisa lanjut ke *task* lainnya. Jika dibiarkan seperti itu maka kamera akan berhenti mengambil gambar setiap kali sistem mengeluarkan suara. Untuk menangani permasalahan yang ada, proses TTS dijalankan di jalur eksekusi terpisah dari alur program utama dan dilengkapi penanda status supaya tidak ada dua ucapan yang tumpang tindih dan berjalan bersamaan. Cara ini tetap efektif meski Python punya *Global Interpreter Lock*, karena GIL dinonaktifkan sementara saat program menunggu proses *input/output* selesai [14]. Jalur-jalur proses yang ada juga diatur agar otomatis berhenti saat program utama ditutup, sehingga tidak ada proses yang menggantung di latar belakang. Cara memisahkan proses suara dari alur utama semacam ini juga dipakai pada perangkat *embedded* lain, misalnya pada *portable virtual text reader* berbasis OCR untuk pengguna dengan gangguan penglihatan [15]. Meskipun demikian, Raspberry Pi tetap perlu diperhatikan potensi *overheating*-nya jika dipakai untuk komputasi dengan beban yang berat dalam waktu cukup lama [16].

Pengujian pada program yang sudah dirancang, dilakukan terhadap 141 kosakata gestur BISINDO dari penelitian sebelumnya yang terdiri atas 46 gestur statis dan 95 gestur dinamis. Tahap pertama adalah pengujian mandiri oleh penulis terhadap seluruh 141 gestur untuk menilai ketepatan sistem klasifikasi setelah penambahan fitur suara tersebut. Lalu tahap kedua melibatkan tiga responden mahasiswa Teknik Informatika Universitas Ma Chung angkatan 2023 yang menguji seluruh 141 gestur dalam tiga blok yaitu Blok A, B, C yang disusun dengan rancangan Latin Square 3×3, sehingga pengaruh kelelahan responden tersebar merata ke seluruh blok gestur. Kemudian tahap ketiga adalah pengujian *frame per second*/FPS pada kamera untuk membandingkan performa sistem saat fungsi TTS dijalankan pada *thread* atau jalur terpisah dan saat dijalankan langsung pada jalur utama program. Setiap percobaan dinilai dari keberhasilan sistem mengenali gestur dan kejelasan suara keluaran TTS, dengan tingkat keberhasilan dihitung sebagai persentase jumlah gestur yang berhasil dikenali terhadap jumlah gestur yang diuji.

III. HASIL DAN PEMBAHASAN

Hasil yang didapat dari pengujian mandiri oleh penulis mencatat 127 dari 141 gestur BISINDO yang berhasil dikenali dan diucapkan dengan tepat atau setara dengan tingkat keberhasilan 90,07%. Lalu pada gestur dinamis mencapai 91,58% sedikit lebih tinggi dibandingkan gestur statis yang mencapai 86,96%. Pengujian responden mencatat hasil yang konsisten yaitu 380 dari 423 percobaan yang berhasil dikenali atau 89,83% dengan selisih hanya 0,24% terhadap pengujian mandiri. Rincian kedua pengujian ditunjukkan pada Tabel 1.

Dari 14 gestur yang gagal pada pengujian mandiri oleh penulis, hanya huruf ‘E’ yang murni disebabkan oleh kegagalan teknis pada fitur TTS. Hal tersebut disebabkan oleh s*ynthesizer mb-id1* yang tidak dapat memproses huruf beraksen ‘É’ yang dipaksa untuk membunyikan vokal /e/ taling

Copyright © SENASIF 2026
ISSN: 2598-0076

tertutup. Tiga belas kegagalan lainnya dimulai dari penggabungan huruf ‘O’ dengan angka ‘0’ dan huruf ‘V’ dengan angka ‘2’ sampai kekeliruan pada gestur ‘MENDENGAR’, ‘TURUN’, ‘MINUM’, dan ‘MENANGIS’, sudah terjadi pada model klasifikasi *hybrid* Random Forest dan Transformer di penelitian sebelumnya [5]. Dan fitur TTS yang dikembangkan pada penelitian ini tidak mengubah lapisan klasifikasi itu.

Dari 380 gestur yang berhasil dikenali responden, 367 gestur di antaranya atau 96,58% dinilai jelas suaranya. Lima gestur statis, yaitu huruf ‘E’, ‘O’, ‘0’, ‘V’, dan ‘2’, bermasalah di kedua tahap pengujian yaitu mandiri maupun responden karena masalahnya ada di sistem, bukan pada cara responden memperagakan gestur. Sembilan gestur lain, seperti ‘KITA’, ‘BAWAH’, dan ‘SELATAN’, gagal dikenali oleh dua responden atau lebih padahal saat diuji mandiri oleh penulis gestur tersebut berhasil dikenali, kemungkinan besar pada perbedaan sudut peragaan antar orang.

Angka 96,58% kejelasan suara ini sejalan dengan penerapan pyttsx3 pada penelitian Dwijayanti et al. untuk komunikasi dua arah manusia-robot berbahasa Indonesia [12]. Penelitian tersebut menempatkan pyttsx3 sebagai komponen keluaran suara dan fokus pengujian pada penelitian tersebut adalah model *Speech-To-Text*, dengan *word error rate* 24,99% pada dalam dataset dan 25,31% pada di luar dataset. Perbandingan ini menunjukkan pyttsx3 layak dipakai untuk komunikasi Bahasa Indonesia secara *real-time*, meskipun penelitian tersebut tidak melaporkan angka kejelasan suara TTS secara langsung seperti pada penelitian ini.

Dan juga pada penelitian yang dilakukan oleh Hasan U. Zaman juga memakai TTS berbasis eSpeak pada Raspberry Pi untuk perangkat *portable virtual text reader* bagi pengguna dengan gangguan penglihatan [15]. Sistem tersebut mengonversi hasil OCR menjadi suara secara langsung, mirip dengan pendekatan penelitian ini yang mengonversi hasil klasifikasi gestur menjadi suara. Penelitian tersebut berhenti pada klaim kualitatif bahwa sistem akurat dan portabel, tanpa angka kejelasan suara yang terukur sehingga penelitian ini melengkapi dengan metrik kejelasan suara eksplisit dari sisi responden.

Dengan 367 dari 380 gestur yang berhasil dikenali juga dinilai jelas suaranya, sistem ini cukup andal dipakai sebagai alat bantu komunikasi harian, bukan sekadar purwarupa laboratorium. Kejelasan suara yang tinggi ini penting karena gestur yang berhasil dikenali tidak banyak berguna kalau lawan bicara tidak bisa menangkap kata yang diucapkan sistem.

Perhitungan Persentase Keberhasilan

Tingkat keberhasilan sistem pada setiap kategori pengujian dihitung dengan membandingkan jumlah gestur yang berhasil dikenali dan diucapkan dengan benar terhadap jumlah gestur yang diuji, sebagaimana dirumuskan pada Persamaan 1.

$$P = \frac{\Sigma Berhasil}{\Sigma Diuji} \times 100\% \qquad 1$$

Copyright © SENASIF 2026

ISSN: 2598-0076

Persamaan 1 menghasilkan seluruh angka persentase keberhasilan pada Tabel 1, baik untuk gestur statis dan dinamis di pengujian mandiri maupun untuk rekapitulasi pengujian responden. Kedua tahap pengujian sama-sama berada di atas 89%.

Rekapitulasi Hasil Pengujian

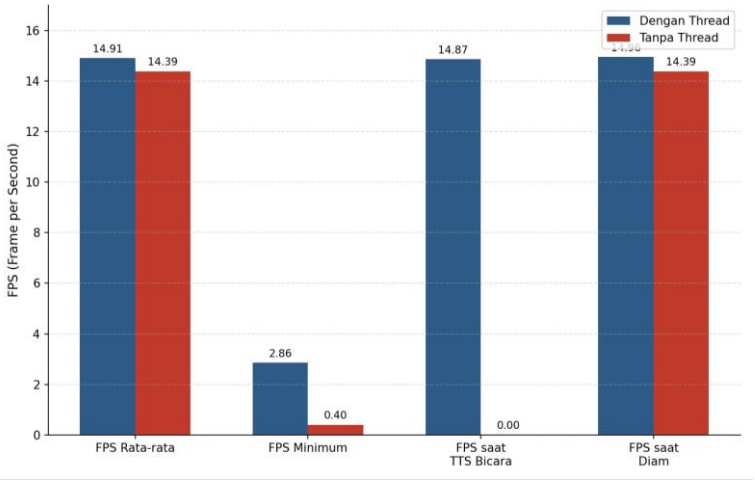
Kemudian pada Tabel 1 merangkum hasil pengujian mandiri berdasarkan kategori gestur serta hasil pengujian gabungan dari tiga responden. Kolom Jumlah Diuji menunjukkan total gestur atau percobaan yang diperagakan sedangkan kolom Berhasil dan Tidak Berhasil menunjukkan hasil klasifikasi beserta pengucapan TTS lalu pada kolom persentase dihitung memakai Persamaan 1.

Tabel 1 Rekapitulasi Hasil Pengujian Mandiri dan Responden

Kategori	Jumlah Diuji	Berhasil	Tidak Berhasil	Persentase Keberhasilan
Gestur Statis (Mandiri)	46	40	6	86,96%
Gestur Dinamis (Dinamis)	95	87	8	91,58%
Total Pengujian Mandiri	141	127	14	90,07%
Total Pengujian Responden	423	380	43	89,83%

Pengujian Performa Kamera dengan *Multithreading*

Lalu selanjutnya yaitu Pengujian FPS pada kamera yang membandingkan dua cara eksekusi fungsi TTS yaitu dengan dijalankan pada *thread* terpisah dan dijalankan langsung pada jalur utama program dan secara bersamaan juga engambil nilai FPS beserta status TTS pada setiap *frame* gambar yang berhasil ditangkap. Gambar 3 menunjukkan bahwa sistem dengan *thread* terpisah menjaga FPS pada kisaran 14,87–14,96 baik saat TTS berbicara maupun diam dengan selisih hanya 0,09 FPS. Sedangkan tanpa *thread* terpisah, FPS turun menjadi 0,00 saat TTS berbicara karena kamera berhenti total mengambil gambar sampai proses ucapan suara selesai, sehingga FPS minimum yang tercatat hanya 0,40 dan jauh di bawah 2,86 pada sistem dengan *thread* terpisah. Memisahkan TTS ke *thread* sendiri menjaga *loop* deteksi kamera tetap berjalan *real-time*.



Gambar 3. Perbandingan FPS Kamera Dengan *Multithreading* dan Tanpa *Multithreading*

Copyright © SENASIF 2026
ISSN: 2598-0076

IV. SIMPULAN

Penambahan fitur *Text-to-Speech* pada penerjemah BISINDO menggunakan Raspberry Pi 5 berhasil mengubah hasil klasifikasi gestur yang sebelumnya hanya tampil sebagai teks di layar menjadi keluaran suara. Lawan bicara tidak perlu lagi mengalihkan fokusnya dengan melihat layar dan pengguna bahasa isyarat, karena sistem langsung mengucapkan hasil terjemahan begitu sebuah gestur dikenali.

Kombinasi pyttsx3, eSpeak, dan MBROLA terbukti mampu mengubah teks hasil klasifikasi menjadi suara Bahasa Indonesia yang jelas didengar, baik pada pengujian mandiri oleh penulis maupun pengujian oleh responden. Sebagian besar gestur BISINDO mulai dari abjad Bahasa Indonesia, angka, hingga kosakata-kosakata dapat diucapkan oleh sistem dengan tepat. Kegagalan yang masih tercatat sebagian besar berasal dari kesalahan klasifikasi gestur pada model *Random Forest* dan *Transformer* yang sudah ada pada penelitian sebelumnya dan bukan kesalahan dari fitur TTS yang dikembangkan dan dirancang pada penelitian ini. Salah satu kendala teknis yang jelas dan murni kesalahan dari sistem berasal dari sisi *engine* TTS terjadi pada huruf berimbuhan aksen ‘E’, di mana *synthesizer mb-id1* belum bisa memproses bunyi vokal tersebut dengan benar.

Penerapan *multithreading* juga menjawab kekhawatiran bahwa proses pengucapan suara akan membebani kinerja kamera. Dengan memisahkan proses TTS ke *thread* tersendiri, kamera tetap bisa menangkap gestur secara *real-time* tanpa terganggu, baik saat sistem sedang mengucapkan hasil terjemahan maupun saat diam. Tanpa pemisahan ini, kamera berhenti total setiap kali suara sedang diproses, yang tentu saja mengganggu alur komunikasi dua arah yang menjadi tujuan utama penelitian. Secara keseluruhan, sistem penerjemah BISINDO dengan fitur TTS ini sudah bisa digunakan sebagai alat bantu komunikasi yang lebih praktis bagi penyandang tunarungu, tanpa mengorbankan performa real-time yang sudah dibangun pada penelitian sebelumnya.

DAFTAR RUJUKAN

[1] Nisria, Mustafa, and Hadis, “Implementasi Bisindo Dalam Berkomunikasi Pada Sesama Anak Tunarungu,” *PINISI JOURNAL OF EDUCATION*, Nov. 2022, Accessed: Jun. 24, 2026. [Online]. Available: https://eprints.unm.ac.id/id/eprint/25584

[2] A. S. Nugraheni, A. P. Husain, and H. Unayah, “Optimalisasi Penggunaan Bahasa Isyarat dengan SIBI dan BISINDO Pada Mahasiswa Difabel Tunarungu di Prodi PGMI UIN Sunan Kalijaga,” *Jurnal Holistika*, vol. 5, no. 1, p. 28, Feb. 2023, doi: 10.24853/holistika.5.1.28-33.

[3] B. Asriandhini and C. H. Rahmawati, “Bahasa Isyarat Indonesia Sebagai Konstruksi Identitas dan Citra Sosial Tuli Di Purwokerto,” *JRK (Jurnal Riset Komunikasi)*, vol. 12, no. 1, Jun. 2021, doi: 10.31506/jrk.v12i1.10059.

Copyright © SENASIF 2026

ISSN: 2598-0076

[4] A. F. Conelia, “Pengembangan Klasifikasi Angka Bahasa Isyarat dengan Menggunakan Metode K-Nearest Neighbor,” *Tugas Akhir/Skripsi thesis, Universitas Ma Chung.*, Sep. 2023, Accessed: Jun. 24, 2026. [Online]. Available: http://repository.machung.ac.id/id/eprint/1145

[5] O. H. Alatas, “Pengembangan Sistem Klasifikasi Bahasa Isyarat BISINDO Secara Real-Time Dengan Raspberry Pi,” *Tugas Akhir/Skripsi thesis, Universitas Ma Chung.*, Jan. 2026, Accessed: Jun. 24, 2026. [Online]. Available: http://repository.machung.ac.id/id/eprint/1092

[6] Raspberry Pi Ltd, “Raspberry Pi 5.” Accessed: Jul. 04, 2026. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-5/

[7] H. Ghael, L. D. Solanki, and G. Sahu, “A Review Paper on Raspberry Pi and its Applications,” *International Journal of Advances in Engineering and Management (IJAEM)*, Jan. 2020, Accessed: Jul. 04, 2026. [Online]. Available: https://www.researchgate.net/publication/348993428_A_Review_Paper_on_Raspberry_Pi_and_its_Applications

[8] R. David *et al.*, “Tensorflow Lite Micro: Embedded Machine Learning On Tinyml Systems,” in *Proceedings of Machine Learning and Systems*, Mar. 2021. Accessed: Jul. 04, 2026. [Online]. Available: https://doi.org/10.48550/arXiv.2010.08678

[9] J. Duddington, “eSpeak: Speech Synthesizer,” SourceForge. Accessed: Jul. 04, 2026. [Online]. Available: https://espeak.sourceforge.net/

[10] T. Dutoit, V. Pagel, N. Pierret, F. Bataille, and O. van der Vrecken, “The MBROLA project: towards a set of high quality speech synthesizers free of use for non commercial purposes,” in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP ’96*, IEEE, pp. 1393–1396. doi: 10.1109/ICSLP.1996.607874.

[11] N. M. Bhat, “pyttsx3 Documentation,” Jul. 14, 2021.

[12] S. Dwijayanti, A. R. Akbar, and B. Y. Suprapto, “Designing Human-Robot Communication in the Indonesian Language Using the Deep Bidirectional Long Short-Term Memory Algorithm,” *Jurnal Elektronika dan Telekomunikasi*, vol. 24, no. 1, pp. 1–11, Aug. 2024, doi: 10.55981/jet.595.

[13] A. Chabchoub and A. Cherif, “An Automatic MBROLA Tool For High Quality Arabic Speech Synthesis,” *Int. J. Comput. Appl.*, vol. 36, 2011, doi: 10.5120/4452-6228.

[14] G. Van Rossum and Jr. , F. L. Drake, *Python Reference Manual*. Centrum voor Wiskunde en Informatica (CWI), 1995.

[15] H. U. Zaman, S. Mahmood, S. Hossain, and I. I. Shovon, “Python Based Portable Virtual Text Reader,” in *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*, IEEE, Oct. 2018, pp. 1–6. doi: 10.1109/ICACCAF.2018.8776778.

[16] S. Monk, *Raspberry Pi Cookbook*, 4th ed. O’Reilly Media, 2023.

Copyright © SENASIF 2026
ISSN: 2598-0076