

Implementasi API Pada Aplikasi Belajar Berbicara Bahasa Indonesia Untuk Anak *Down Syndrome* Berbasis *Mobile*

Rifky Nudhar Nurislami
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
rifkynudham@student.telkomuniversity.ac.id

Favian Dewanta
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
favian@telkomuniversity.ac.id

Retno Hendryanti
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
rehendry@telkomuniversity.ac.id

Abstrak — Anak-anak dengan *down syndrome* seringkali menghadapi tantangan dalam berkomunikasi akibat kesulitan dalam artikulasi, pemahaman kosakata, dan tata bahasa, yang dapat menghambat interaksi sosial dan perkembangan akademik mereka. Terapi wicara, penggunaan teknologi asistif, serta metode pembelajaran berbasis visual dan interaktif menjadi solusi untuk meningkatkan keterampilan komunikasi anak *down syndrome*. Penelitian ini bertujuan untuk mengembangkan aplikasi *mobile* berbasis teknologi pengenalan suara dan *machine learning* sebagai upaya meningkatkan kemampuan berbicara anak-anak dengan *down syndrome*. Aplikasi ini dirancang untuk memberikan pengalaman pembelajaran interaktif dengan kombinasi elemen audio dan visual yang disesuaikan dengan kebutuhan individu pengguna, serta memberikan umpan balik *real-time* untuk membantu mereka belajar secara mandiri. Hasil pengujian menunjukkan bahwa aplikasi ini mampu mendukung proses latihan berbicara anak dengan performa API yang optimal pada kondisi server aktif, ditunjukkan oleh waktu respons singkat, *throughput* yang stabil, serta latensi yang rendah. Pengembangan aplikasi ini diharapkan dapat memberikan kontribusi positif dalam meningkatkan kemampuan komunikasi anak-anak dengan *down syndrome*, sehingga mereka dapat lebih mandiri dan percaya diri dalam berinteraksi dengan lingkungan sekitar.

Kata kunci — Aplikasi berbasis *mobile*, *Down syndrome*, Belajar berbicara, Pengenalan kata dan frasa dasar, Latihan pelafalan.

I. PENDAHULUAN

Kebutuhan akan alat bantu berbicara bagi anak-anak dengan *down syndrome* muncul dari tantangan komunikasi yang mereka hadapi sehari-hari. Kesulitan dalam artikulasi, pemahaman kosakata, serta tata bahasa menyebabkan hambatan dalam komunikasi efektif, yang berdampak pada interaksi sosial dan perkembangan akademik mereka. Anak-anak dengan *down syndrome* sering mengalami kesulitan dalam menyampaikan pikiran dan kebutuhan mereka, yang dapat menyebabkan frustrasi baik bagi mereka maupun lingkungan sekitar. Meskipun mereka biasanya lebih baik dalam memahami bahasa yang didengar, mereka sering kesulitan dalam menyampaikan pikiran dan kebutuhan mereka, yang dapat menimbulkan frustrasi dan masalah perilaku [1]. Tantangan ini dapat semakin kompleks dengan adanya impulsivitas dan keterbatasan dalam memori kerja, yang turut mempengaruhi kemampuan mereka dalam belajar bahasa secara konvensional.

Dalam konteks ini, terapi wicara, penggunaan teknologi asistif, serta metode pembelajaran berbasis visual dan

interaktif menjadi solusi untuk meningkatkan keterampilan komunikasi anak *down syndrome*. Pengembangan aplikasi berbicara berbasis *mobile* merupakan inovasi untuk mendukung anak-anak dengan *down syndrome* dalam meningkatkan keterampilan komunikasi mereka. Aplikasi ini bertujuan untuk membantu anak-anak berlatih berbicara secara interaktif melalui pendekatan berbasis teknologi yang lebih fleksibel dan personal. Dengan adanya teknologi pengenalan suara dan *machine learning*, aplikasi ini dapat memberikan umpan balik secara *real-time*, yang memungkinkan pengguna untuk memperbaiki pengucapan mereka secara berulang tanpa tekanan dari lingkungan sosial.

Aplikasi "Berbicara Bahasa Indonesia untuk Anak dengan *Down Syndrome*" dikembangkan untuk mengatasi tantangan tersebut dengan mengintegrasikan teknologi pengenalan suara dan *machine learning*. Karena anak-anak ini membutuhkan akses yang lebih baik ke aplikasi dan perangkat lunak yang dirancang khusus untuk membantu mereka mengatasi tantangan komunikasi dan belajar [2]. Aplikasi ini dirancang untuk memberikan pengalaman pembelajaran interaktif dengan kombinasi elemen audio dan visual yang dapat disesuaikan dengan kebutuhan individu pengguna. Selain itu, aplikasi ini menawarkan fleksibilitas dalam penggunaannya, baik oleh anak-anak, orang tua, maupun pendidik, guna menciptakan lingkungan belajar yang inklusif dan mendukung perkembangan komunikasi anak-anak dengan *down syndrome*.

Selain manfaat bagi anak-anak, aplikasi ini juga memberikan keuntungan bagi orang tua dan pendidik dalam membantu proses pembelajaran. Dengan adanya fitur yang dapat merekam dan menganalisis pengucapan anak, orang tua dan pendidik dapat lebih mudah dalam memantau perkembangan anak secara objektif. Aplikasi ini juga dapat digunakan sebagai alat bantu dalam terapi wicara yang biasanya dilakukan secara langsung dengan tenaga profesional. Dengan kombinasi antara pembelajaran mandiri dan pendampingan dari orang tua serta guru, anak-anak dengan *down syndrome* dapat memperoleh pengalaman belajar yang lebih komprehensif.

Seiring berkembangnya teknologi, penerapan *machine learning* dalam dunia pendidikan semakin meluas. Penggunaan algoritma yang dapat menyesuaikan diri dengan kemampuan pengguna menjadikan proses belajar lebih efektif dan menyenangkan. Dalam konteks anak-anak dengan *down syndrome*, pendekatan berbasis teknologi ini sangat membantu dalam mengatasi keterbatasan metode

konvensional. Dengan desain antarmuka yang menarik dan fitur yang adaptif, aplikasi ini diharapkan dapat menjadi solusi berkelanjutan dalam meningkatkan kemampuan komunikasi anak-anak dengan *down syndrome*.

Dengan adanya penelitian dan pengembangan aplikasi ini, diharapkan bahwa anak-anak dengan *down syndrome* dapat lebih percaya diri dalam berkomunikasi, lebih mudah dalam memahami bahasa, lebih mudah untuk berkomunikasi, dan lebih siap dalam menghadapi interaksi sosial. Hal ini akan memberikan dampak positif dalam kehidupan mereka sehari-hari, baik dalam lingkungan keluarga, sekolah, maupun masyarakat luas.

II. KAJIAN TEORI

Dalam bidang pendidikan khusus, berbagai teori telah dikembangkan untuk memahami bagaimana anak-anak dengan kebutuhan khusus, termasuk *down syndrome*, dapat belajar secara optimal. Salah satu pendekatan yang relevan adalah teori pembelajaran multimodal, yang menekankan pentingnya kombinasi berbagai elemen seperti visual, audio, dan interaksi dalam meningkatkan pemahaman serta keterampilan anak-anak.

A. *Down Syndrome* dan Tantangan Komunikasi

Down syndrome merupakan kelainan genetik yang menyebabkan keterlambatan perkembangan, termasuk dalam aspek komunikasi [3]. Anak-anak dengan *down syndrome* umumnya memiliki kesulitan dalam artikulasi, tata bahasa, serta kosakata yang terbatas. Mereka lebih cenderung memahami bahasa secara visual dibandingkan auditori, sehingga metode pembelajaran yang mengombinasikan elemen visual dan auditori lebih efektif dalam meningkatkan kemampuan bahasa mereka [4].

B. Teknologi Pengenalan Suara dan *Machine Learning*

Penggunaan teknologi pengenalan suara memungkinkan sistem untuk menganalisis dan mengevaluasi ketepatan pengucapan pengguna. Dalam aplikasi ini, metode *Mel Frequency Cepstral Coefficient* (MFCC) digunakan untuk mengekstraksi fitur suara dan mengubah menjadi beberapa parameter [5]. MFCC mereplikasi pendengaran manusia dalam melakukan persepsi terhadap sinyal suara [6]. Sementara algoritma *random forest* berperan dalam mengklasifikasikan hasil pengucapan [7]. Kombinasi teknologi ini memberikan umpan balik berbasis teks secara *real-time* untuk membantu pengguna meningkatkan keterampilan berbicara mereka.

C. Pembelajaran Berbasis *Mobile* untuk Anak Berkebutuhan Khusus

Aplikasi *mobile* memberikan fleksibilitas dalam pembelajaran, memungkinkan anak-anak berlatih kapan saja dan di mana saja. Dengan pendekatan berbasis audio-visual, aplikasi ini dapat memberikan pengalaman belajar yang lebih interaktif dan menarik dibandingkan metode tradisional. Selain itu, elemen gamifikasi dalam aplikasi dapat meningkatkan motivasi pengguna dalam berlatih berbicara [8].

III. METODE

Penelitian ini menggunakan metode pengembangan aplikasi berbasis mobile dengan pendekatan yang berpusat pada pengguna (*user-centered design*).

A. Desain dan Pengembangan Aplikasi

Aplikasi dikembangkan menggunakan *flutter* sebagai *framework* utama, dengan *firebase* sebagai *database* penyimpanan data pengguna dan hasil latihan. Pengenalan suara diproses melalui API berbasis *cloud*, yang memungkinkan evaluasi dan analisis suara secara *real-time*.

B. Segmentasi dan Pemrosesan Data oleh API



GAMBAR 1 Alur Segmentasi

API tidak hanya berfungsi sebagai konektor antar sistem tetapi juga melakukan segmentasi dan analisis suara sebelum dikirim ke *machine learning*. Alur segmentasi yang terjadi pada API ditunjukkan oleh Gambar 1. Segmentasi ini dilakukan setiap 0,5 detik untuk memastikan pemrosesan suara lebih akurat. Jika energi dari segmen audio kurang dari 0,005, maka API akan mengklasifikasikan data sebagai tidak valid untuk mencegah *error* dalam evaluasi suara.

Segmentasi ini dilakukan dengan membaca file audio menggunakan *librosa*, menambahkan padding jika diperlukan, dan menghitung energi tiap segmen untuk

memastikan validitasnya. Segmentasi dilakukan setiap 0,5 detik. Artinya data yang diproses akan terbagi menjadi beberapa bagian setiap 0,5 detik tergantung pada panjang data. Data yang telah tersegmentasi dilakukan perhitungan energi. Jika setiap segmen memiliki energi di bawah 0,005, API akan menghasilkan penilaian yang tidak tervalidasi. Karena data yang memiliki energi rendah tidak dapat diproses. Saat energi yang dimiliki setiap segmen cukup, proses ekstraksi fitur bisa dilakukan.

Setelah segmentasi, data suara diekstraksi menggunakan metode *Mel Frequency Cepstral Coefficient* (MFCC) sebelum dikirim ke model *machine learning* berbasis *random forest* untuk dilakukan klasifikasi. Hasil analisis akan memberikan skor probabilitas yang menunjukkan tingkat kemiripan suara user dengan pola suara yang benar. Hasil dari MFCC diteruskan ke *machine learning* untuk diprediksi. Hasil dari proses yang dilakukan oleh *machine learning* berupa penilaian dan prediksi dari setiap segmen. Oleh karena itu API dibuat dengan merata-ratakan hasil dari setiap segmen tersebut. Nilai rata-rata tersebut menjadi nilai akhir dari pengujian yang dilakukan. Nilai akhir tersebut dikirimkan kembali ke aplikasi *mobile* dan juga data disimpan di *database* sebagai riwayat pengujian secara *real-time*.

C. Proses Pengujian

Pengujian aplikasi dilakukan di lingkungan pendidikan inklusif, seperti Sekolah Luar Biasa (SLB) dan komunitas pendukung *down syndrome*. Pengujian melibatkan anak-anak dengan *down syndrome* yang berpartisipasi dalam latihan berbicara menggunakan aplikasi ini. Parameter yang diukur meliputi efektivitas aplikasi dalam meningkatkan keterampilan berbicara, respons pengguna terhadap fitur interaktif, serta kecepatan dan akurasi pemrosesan suara.

D. Analisis Data dan Evaluasi

Data dikumpulkan dari hasil pengujian suara yang direkam, tingkat keberhasilan dalam pengucapan kata, serta umpan balik dari orang tua dan pendidik. Evaluasi dilakukan dengan membandingkan tingkat perkembangan komunikasi anak sebelum dan sesudah menggunakan aplikasi. Selain itu, aspek *usability* dan *user experience* juga dianalisis untuk menyempurnakan fitur aplikasi guna memastikan kemudahan dan kenyamanan penggunaannya.

IV. HASIL DAN PEMBAHASAN

A. Hasil Implementasi API

API yang diimplementasikan dalam aplikasi ini memiliki peran penting dalam menghubungkan aplikasi *mobile* dengan *backend* yang melibatkan *database* dan *machine learning*. API bertanggung jawab untuk mengatur aliran data, termasuk perekaman suara, pengiriman file audio, pemrosesan suara oleh sistem pengenalan suara, serta penyimpanan hasil evaluasi ke dalam *database* [9].

Dalam proses implementasi, API mampu melakukan konektivitas yang stabil antara aplikasi *mobile* dan *server cloud*, memungkinkan data yang dikirimkan

dari aplikasi dapat diproses dengan benar oleh sistem *backend*. API dirancang agar mampu menangani berbagai jenis *request* secara efisien dengan memastikan keamanan, latensi rendah, serta kompatibilitas terhadap berbagai format data.

Pada tahap pengujian, API berhasil menghubungkan aplikasi *mobile* dengan *database firebase* untuk mengambil dan menyimpan informasi *user*, termasuk data riwayat pengujian. API juga menangani proses pengiriman file audio untuk dievaluasi oleh model *machine learning*. Jika format file audio yang dikirim bukan ".wav", API akan secara otomatis melakukan konversi agar dapat diproses dengan baik oleh sistem pengenalan suara.

TABEL 1
Spesifikasi API

| Spesifikasi | Digunakan untuk konektivitas antar komponen |
|----------------------|--|
| Metode pengujian | Pemantauan waktu dan ukuran data saat proses pengiriman data hingga menerima hasil berlangsung |
| Mekanisme pengukuran | Pengukuran waktu dan ukuran data melalui terminal <i>flutter</i> dan <i>postman</i> |
| Prosedur Pengukuran | <ul style="list-style-type: none"> • <i>Running</i> kode <i>flutter</i> dan memeriksa <i>output</i> pada terminal • Akses url dengan <i>endpoint</i> yang sesuai |

Tabel 1 menunjukkan spesifikasi API, di mana metode pengujian dilakukan dengan pemantauan waktu pemrosesan dan ukuran data yang dikirim saat proses pengiriman berlangsung hingga hasil diterima. Pengujian dilakukan melalui terminal *flutter* dan *postman* untuk memastikan API dapat menangani beban kerja dalam berbagai skenario.

B. Pengujian API dan Performa

Pengujian API dilakukan dalam beberapa skenario untuk mengevaluasi efisiensi dan kecepatan responnya. Beberapa parameter yang diuji meliputi kecepatan transmisi data, ukuran paket data, serta kondisi *server* saat menangani permintaan.

```
I/flutter (31170): Response received at: 2025-01-20 18:30:28.402145
I/flutter (31170): Time taken from audio sent to result received: 35 712ms
I/flutter (31170): Request status: 200
I/flutter (31170): Response reason: OK
I/flutter (31170): Raw server response body: {"average_confidence":
63.25000000000001, "question": "Itu
apa", "result": "Benar", "tanggal": "2025-01-20", "waktu": "18:30:24"}
```

GAMBAR 2

Hasil Proses Dari *Flutter* Kondisi *Server* Tidak *Sleep*

Gambar 2 menunjukkan waktu yang dibutuhkan untuk mengirim file audio dan menerima hasil pengujian dalam kondisi *server* aktif: adalah sekitar 3,712 detik.

```

Body Cookies Headers (9) Test Results 200 OK 2.83 s 778 B
{} JSON Preview Visualize
1 {
2   "average_confidence": 76.125,
3   "question": "Itu Apa",
4   "result": "Benar",
5   "tanggal": "2025-01-16",
6   "waktu": "12:00:00"
7 }
    
```

GAMBAR 3 Hasil Proses Dari Postman Kondisi Server Tidak Sleep

Gambar 3 menunjukkan data pengujian menggunakan *postman*. Waktu pemrosesan lebih cepat, yaitu 2,83 detik dengan ukuran data yang ditransfer sebesar 778 Byte. Hasil yang ada pada API diantaranya adalah *average confidence*, pertanyaan yang diujikan, hasil, tanggal, dan waktu pengujian dilakukan oleh *user*. *Average confidence* merupakan nilai akhir yang didapat dari *machine learning*. Nilai dari setiap segmen dirata-ratakan untuk dijadikan nilai akhir dari pengujian. Dan *result* menunjukkan validasi ketepatan suara yang diujikan. Tanggal dan waktu menunjukkan waktu pengujian dilakukan.

```

I/flutter (31170): Response received at: 2025-01-20 18:28:54.127492
I/flutter (31170): Time taken from audio sent to result received: 33s 385ms
I/flutter (31170): Request status: 200
I/flutter (31170): Response reason: OK
I/flutter (31170): Raw server response body: {"average_confidence": 58.9, "question": "Itu apa", "result": "Salah", "tanggal": "2025-01-20", "waktu": "18:28:20"}
    
```

GAMBAR 4 Hasil Proses Dari Flutter Kondisi Server Sleep

Gambar 4 menunjukkan data pengujian dalam kondisi server *sleep*: Waktu pemrosesan meningkat drastis hingga 33,835 detik. Waktu tersebut lebih lama dibandingkan ketika server dalam keadaan aktif.

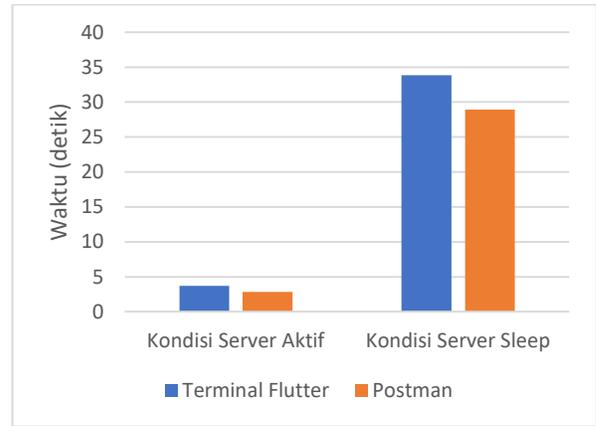
```

Body Cookies Headers (9) Test Results 200 OK 28.91 s 786 B
{} JSON Preview Visualize
1 {
2   "average_confidence": 76.125,
3   "question": "Itu Apa",
4   "result": "Benar",
5   "tanggal": "2025-01-16",
6   "waktu": "12:00:00"
7 }
    
```

GAMBAR 5 Hasil Proses dari Postman Kondisi Server Sleep

Gambar 5, pengujian dengan *postman* dalam kondisi server *sleep*, menunjukkan hasil serupa dengan pengujian dari terminal *flutter*. Waktu pemrosesan 4,925 detik lebih cepat, sekitar 28,91 detik dan ukuran data yang dikirimkan tidak jauh berbeda sebesar 786 Byte.

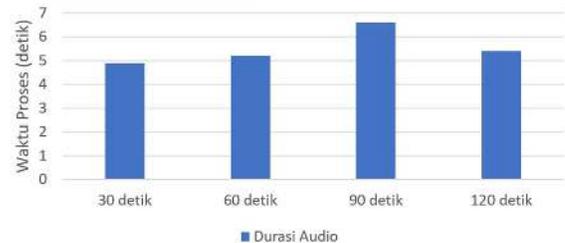
Pengujian tersebut baik kondisi server *sleep* atau aktif, respon dari server menghasilkan kode 200 OK. Hal tersebut merupakan respon yang menandakan bahwa konektivitas dan proses dapat berjalan dengan baik [10].



GAMBAR 6 Perbandingan Latensi Terhadap Kondisi Server

Gambar 6 menunjukkan bahwa terdapat perbedaan yang signifikan antara kondisi server aktif dan *sleep*. Perbedaan tersebut dikarenakan server perlu inialisasi ulang model *machine learning* yang diakses dari GitHub. Proses tersebut membutuhkan waktu lebih lama dari kondisi server aktif. Hal tersebut menyebabkan meningkatnya delay pada pemrosesan data.

Selain kondisi server, waktu yang dibutuhkan untuk yang diproses. Salah satu faktor penunjang jaringan adalah *throughput*. Semakin tinggi nilai *throughput* dalam suatu jaringan, semakin cepat juga data yang ditransmisikan [11]. Pada jaringan yang stabil, proses data dari pengiriman hingga menerima hasil akan terjadi dalam waktu yang singkat yaitu sekitar 2 hingga 5 detik dalam kondisi server aktif. Berbeda dengan jaringan yang tidak stabil. Proses pengiriman hingga menerima hasil akan membutuhkan waktu yang lebih lama. Hal tersebut dikarenakan meningkatnya *buffering* selama proses berjalan.



GAMBAR 7 Perbandingan Waktu Pemrosesan Terhadap Durasi Audio

Faktor yang mempengaruhi waktu pemrosesan juga dipengaruhi oleh ukuran data yang diproses [12]. Ukuran data yang diproses berbanding lurus dengan waktu yang dibutuhkan. Semakin besar ukuran data yang diproses, maka akan semakin banyak waktu yang dibutuhkan untuk memproses data. Berlaku untuk sebaliknya, yaitu semakin kecil ukuran data yang diproses, maka akan semakin sedikit waktu yang dibutuhkan. Waktu yang dibutuhkan untuk memproses mempengaruhi pengalaman *user* dalam pemakaian aplikasi dalam sehari-hari. Pemrosesan data yang memiliki waktu singkat akan memberikan pengalaman penggunaan aplikasi yang lebih baik. Sebagaimana

ditunjukkan pada Gambar 7, data yang diproses tidak terjadi perubahan ukuran data yang signifikan. Hal tersebut tidak akan menyebabkan terjadinya perubahan performa dari API secara signifikan juga.

C. Optimasi dan Tantangan API

Meskipun API telah berfungsi dengan baik, terdapat beberapa tantangan dalam implementasinya, termasuk latensi tinggi saat *server* dalam kondisi *sleep* dan kebutuhan untuk meningkatkan akurasi dalam analisis suara. Beberapa solusi yang dapat diterapkan untuk mengoptimalkan API meliputi:

- Menggunakan layanan *cloud* dengan latensi rendah, seperti *server* yang selalu aktif untuk mengurangi waktu respon pada kondisi *sleep*.
- Mengoptimalkan teknik *caching* agar data yang sering digunakan dapat diakses lebih cepat tanpa harus memproses ulang seluruh *pipeline* data.
- Meningkatkan efisiensi algoritma segmentasi dan ekstraksi fitur agar lebih cepat dalam mengolah suara sebelum dikirim ke *machine learning*.
- Menggunakan metode *load balancing* untuk memastikan *server* dapat menangani permintaan dalam jumlah besar tanpa penurunan performa.

V. KESIMPULAN

Berdasarkan hasil pengujian, API yang dikembangkan memiliki performa yang baik dalam menangani permintaan dari aplikasi mobile, dengan waktu respon tercepat 2,83 detik dan ukuran data yang dikirimkan sekitar 778 Byte dalam kondisi *server* aktif. Namun, dalam kondisi *server sleep*, waktu pemrosesan meningkat drastis hingga 33,835 detik, menunjukkan bahwa optimasi pada sisi *server* sangat diperlukan. Segmentasi audio yang dilakukan setiap 0,5 detik memastikan bahwa data yang diproses memiliki kualitas yang cukup baik untuk dianalisis oleh model *machine learning*. Dengan akurasi pemrosesan yang optimal dan waktu respons yang lebih cepat dalam kondisi *server* aktif, API ini memberikan pengalaman yang lebih baik bagi pengguna. Dengan menerapkan teknik optimasi seperti *load balancing* dan *server always-on*, API dapat ditingkatkan agar waktu respons tetap rendah bahkan dalam kondisi *server* tidak aktif. Penerapan strategi ini diharapkan mampu meningkatkan kecepatan pemrosesan hingga 5 kali lebih cepat dibanding kondisi tanpa optimasi. Selain itu ukuran data yang diproses oleh API menjadi faktor yang mempengaruhi pemrosesan. Semakin besar data yang diproses maka semakin banyak waktu yang dibutuhkan. Secara keseluruhan, API ini sudah memenuhi kebutuhan aplikasi dalam mendukung pembelajaran anak *down syndrome*, namun masih ada ruang untuk peningkatan, terutama dalam mengatasi latensi tinggi akibat inisialisasi *server*.

REFERENSI

- [1] A. N. Chamidah, "Intervensi Dini Gangguan Perkembangan Komunikasi Pada Anak Down Syndrome," *Dinamika Pendidikan*, vol. 22, no. 1, hlm. 28–37, 2017.
- [2] Y. Amaliyah, D. J. Amelia, N. Yuliantini, dan P. Setiono, "Desain Teknologi Pembelajaran Untuk Meningkatkan Kemampuan Membaca Pada Anak Down Syndrome," *Jurnal Kajian, Penelitian dan Pengembangan Kependidikan*, vol. 15, no. 4, hlm. 307–315, Okt 2024, doi: 10.31764.
- [3] N. Ashari dkk., "Analisis Perkembangan Bahasa Anak Sindrom Down," *JIVA: Journal of Behaviour and Mental Health*, vol. 5, no. 1, hlm. 35–48, Jun 2024, [Daring]. Tersedia pada: <https://journal.iainmanado.ac.id/index.php/JIVA/index>
- [4] R. N. L. Pasaribu, "Kajian Pola Penerimaan Bahasa Pada Anak Penderita Down Syndrome," *Prosiding Seminar Nasional Linguistik dan Sastra (SEMANTIKS)*, Bandung, hlm. 674–681, 2021. [Daring]. Tersedia pada: <https://jurnal.uns.ac.id/prosidingsemantiks>
- [5] T. Nasution, "Metoda Mel Frequency Cepstrum Coefficients (MFCC) untuk Mengenal Ucapan pada Bahasa Indonesia," *Jurnal Sains dan Teknologi Informasi*, vol. 1, no. 1, hlm. 22–31, 2012.
- [6] A. Wicaksono, N. Sukmawati, S. Adhy, dan Sutikno, "Aplikasi Speech Recognition Bahasa Indonesia Dengan Metode Mel-Frequency Cepstral Coefficient Dan Linear Vector Quantization Untuk Pengendalian Gerak Robot," *Prosiding Seminar Nasional Ilmu Komputer Undip*, hlm. 61–66, 2014.
- [7] I. Suyudi, S. Sudadio, dan S. Suherman, "Pengenalan Bahasa Isyarat Indonesia menggunakan Mediapipe dengan Model Random Forest dan Multinomial Logistic Regression," *Jurnal Ilmu Siber dan Teknologi Digital*, vol. 1, no. 1, hlm. 65–80, Feb 2023, doi: 10.35912/jisted.v1i1.1899.
- [8] Y. P. Pratama, V. Y. E. Erviana, I. Suwartini, dan M. N. A. Antono, "Gamifikasi Dalam Kelas Bahasa Indonesia Bagi Penutur Asing Anak (BIPA)," *Scholaria: Jurnal Pendidikan dan Kebudayaan*, vol. 15, no. 1, hlm. 60–67, 2025.
- [9] H. Y. Hermansyah dan Maryam, "Implementasi Teknologi Application Programming Interface Pada Perancangan Aplikasi Absensi Pegawai," *JIPi (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 3, hlm. 744–754, Sep 2023, doi: 10.29100/jipi.v8i3.3890.
- [10] Kurniawati L, M. Ed. Alimin Z, dan M. Pd. Dra. Asri P, "Program Intervensi Pengembangan Kecakapan Berbicara Anak Down Syndrome PEDAGOGIA : Jurnal Ilmu Pendidikan Program Intervensi Pengembangan Kecakapan Berbicara Anak Down Syndrome," hlm. 195–204
- [11] M. R. Wardana dan D. B. Santoso, "Analisis Throughput Distribusi Jaringan Nirkabel Pada Politeknik Bumi Akpelni," *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, vol. 8, no. 2, hlm. 558–567, Agu 2023, [Daring]. Tersedia

pada:
<https://tunasbangsa.ac.id/ejurnal/index.php/jurasik>
[12] M. B. Hartanto, T. Muhammad Fawa, dan D. P. Eko
Hendro, “Analisa Kinerja Database dan
Implementasi Cache Redis Pada Web Service

Lumen,” *Jurnal Alih Teknologi Komputer (ALTEK)*,
vol. 4, no. 2, Okt 2023, [Daring]. Tersedia pada:
<https://lumen.laravel.com/>,[https://github.com/laravel](https://github.com/laravel/lumen)
[/lumen,https://laravel.com/docs](https://laravel.com/docs)

