

APLIKASI WEBSITE PENERJEMAH BAHASA ISYARAT SEBAGAI MEDIA PEMBELAJARAN SLB NEGERI CICENDO KOTA BANDUNG

Ario Syawal Muhammad
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

Marisa W. Paryasto
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

ariosyawal@student.telkomuniversity.ac.id

setiacasie@telkomuniversity.ac.id

marisaparyasto@telkomuniversity.ac.id

Abstrak — Bahasa isyarat membantu penyandang tunarungu berkomunikasi, tetapi masyarakat umum masih sangat terbatas dalam memahami dan menggunakan bahasa ini. Hal ini menimbulkan hambatan bagi penyandang tunarungu dan orang yang tidak mengerti bahasa isyarat untuk berkomunikasi. Perkembangan teknologi informasi di era modern memberikan peluang besar untuk membuat solusi untuk mengatasi tantangan komunikasi. Proyek ini mengembangkan sebuah aplikasi berbasis *website* bernama U-SIBI sebagai wadah untuk sistem penerjemah bahasa isyarat, *website* ini juga bertujuan untuk menjadi media pembelajaran yang efektif untuk anak-anak serta guru pada SLB Negeri Cicendo Bandung. Pengembangan *website* dilakukan dengan menggunakan *framework* Laravel 8, serta melakukan prototyping UI/UX dengan menggunakan Figma. *Website* U-SIBI juga mengaplikasikan *flask* sebagai *back-end* untuk menghubungkan algoritma *machine learning* yang telah dikembangkan. Semua komponen dan fitur yang telah dikembangkan berjalan sesuai dengan spesifikasi yang diinginkan, ini terbukti dari hasil pengujian alfa yang berhasil mencapai 100%. Dengan demikian, hasil akhir dari *website* U-SIBI dapat diterima dengan baik oleh masyarakat SLB Negeri Cicendo Bandung sebagai media pembelajaran yang efektif.

Kata kunci— laravel 8, figma, flask, penerjemah bahasa isyarat.

I. PENDAHULUAN

Masalah utama yang dihadapi penyandang disabilitas khususnya tunarungu adalah kurangnya aksesibilitas terhadap informasi dan komunikasi. Orang yang menggunakan bahasa isyarat sebagai cara utama untuk berkomunikasi dapat lebih mudah diakses jika mereka menerjemahkan bahasa isyarat ke dalam teks atau ucapan. Akibatnya, masalah ini menunjukkan betapa pentingnya mencari cara untuk memberi komunitas ini akses yang lebih baik ke informasi. Bahasa isyarat adalah bahasa yang berkomunikasi dengan menggunakan bahasa tubuh, gerak bibir, dan bentuk tangan, serta gerak tangan, dan wajah untuk mengungkapkan pikirannya [1].

Kurangnya kebutuhan khusus penyandang tunarungu sering mengakibatkan minimnya perhatian terhadap penyediaan fasilitas komunikasi yang sesuai. Akibatnya,

penyandang tunarungu pada SLB Negeri Cicendo Kota Bandung harus mempelajari materi bahasa isyarat diluar dari jam pelajaran. Terlebih lagi media belajar mereka berpusat pada sebuah buku atau kamus, Hanya saja metode pembelajaran ini kurang interaktif dan masih tradisional, serta masih sedikit tersedia media digital yang digunakan untuk pembelajaran bahasa isyarat.

Membaca buku adalah hal yang paling dasar dilakukan untuk mempelajari suatu ilmu, begitu juga dengan penyandang tunarungu menggunakan buku untuk mempelajari bahasa isyarat agar mereka berinteraksi satu sama lain. Salah satu buku yang membahas tentang pembelajaran bahasa isyarat adalah “Kamus Sistem Bahasa Isyarat Indonesia” [2]. Tujuan penelitian ini adalah untuk mengembangkan dan menerapkan sistem penerjemah bahasa isyarat yang efisien berbasis website untuk SLB Negeri Cicendo Bandung.

II. KAJIAN TEORI

Membaca buku adalah hal yang paling dasar dilakukan untuk mempelajari suatu ilmu, begitu juga dengan teman tunarungu menggunakan buku untuk mempelajari bahasa isyarat agar mereka berinteraksi satu sama lain. Salah satu buku yang membahas tentang pembelajaran bahasa isyarat adalah “KAMUS SISTEM BAHASA ISYARAT INDONESIA”. Kamus ini merupakan hasil kerjasama oleh Kementerian Pendidikan dan Budaya dan Kementerian Lingkungan Kehidupan [2].



Gambar 1. Kamus SIBI

Kelebihan dari kamus ini ialah memberikan rincian gerakan dan penjelasan untuk bahasa isyarat, serta dapat memperagakan gerakan bahasa isyarat secara langsung. Sedangkan kekurangannya membaca kamus saja dianggap kurang efektif dan interaktif dikarenakan tidak adanya komunikasi dengan orang lain dan tidak ada yang dapat memberi tahu kan apakah gerakan tangan yang dilakukan sudah benar atau belum [3].



Gambar 2. website kamus sibi

Terdapat beberapa website yang menyediakan pembelajaran bahasa isyarat untuk penyandang tunarungu, dan masyarakat umum. Website ini dapat diakses dengan mengunjungi laman website resmi Kamus SIBI yang dibuat oleh Kementerian Pendidikan dan Budaya Indonesia (KEMDIKBUD) [2]. Kelebihan dari menggunakan website ini ialah lebih mudah digunakan dan dapat diakses setiap saat sesuai dengan kebutuhan, website ini juga sudah memiliki penjelasan secara rinci untuk gerakan bahasa isyarat yang memperagakan abjad, imbuhan, dan angka sehingga lebih memungkinkan pembelajaran yang interaktif. Sedangkan kelemahan dari penggunaan website ini adalah sangat tergantung dengan jaringan internet dan memerlukan perangkat elektronik seperti smartphone atau laptop untuk dapat mengakses laman website tersebut.

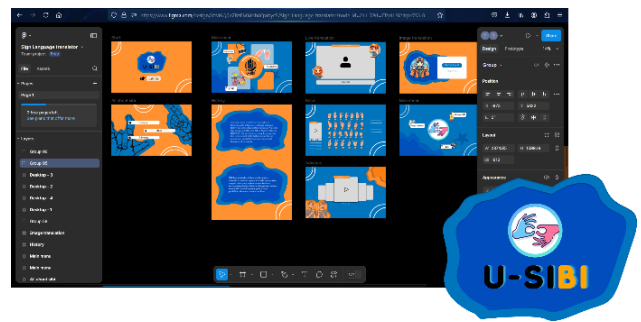
III. METODE

1. Pembuatan Antarmuka Situs Web U-SIBI.

Langkah pertama dalam mengembangkan aplikasi berbasis web "U-SIBI" adalah menentukan framework yang akan digunakan. Framework dapat menghemat banyak waktu dan tenaga dalam mengembangkan aplikasi web, framework beroperasi dengan menyediakan berbagai struktur dan kerangka kerja yang telah dirancang sebelumnya [4].

Laravel adalah *framework* PHP yang menekankan kesederhanaan dan fleksibilitas desain. Laravel dirilis di bawah lisensi MIT dan kode sumber tersedia di *Github*. Seperti *framework* PHP lainnya, Laravel dibangun di atas MVC (*Model-View-Controller*). Laravel hadir dengan alat baris perintah yang disebut "*Artisan*" yang dapat digunakan untuk mengemas dan menginstal bundel. Menurut survei yang dilakukan oleh *Sitepoint.com* pada bulan Desember 2013 mengenai popularitas framework PHP, Laravel berada di posisi teratas. Hal ini menjadikan Laravel sebagai *framework* PHP terbaik tahun 2014 [5].

Versi *Laravel* yang akan digunakan pada projek ini adalah *Laravel* 8, alih-alih menggunakan versi *Laravel* terbaru seperti 10 dan seterusnya *Laravel* 8 lebih mudah untuk digunakan dikarenakan komunitas pengguna yang lebih aktif serta dokumentasi yang lebih lengkap.



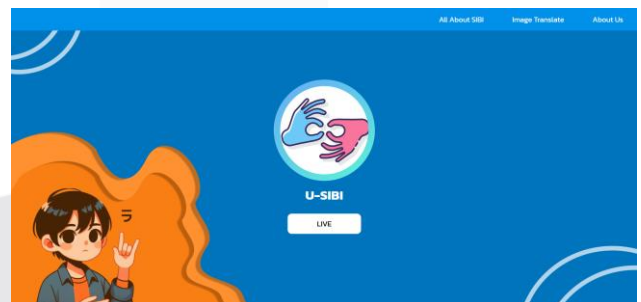
Gambar 3. Pembuatan Prototipe dan Desain Awal Figma.

Desain awal dan pembuatan prototipe UI/UX dilakukan dengan menggunakan aplikasi bernama Figma. Situs *web* ini memiliki total 9 halaman, termasuk halaman *loading*.



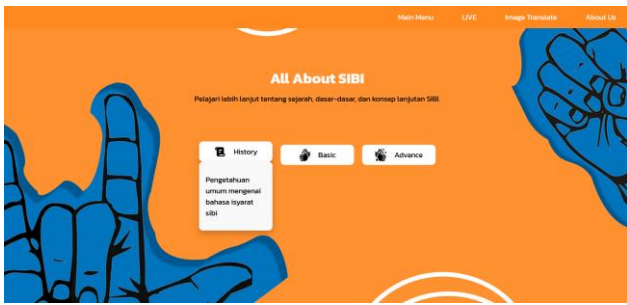
Gambar 4. halaman home website U-SIBI.

Halaman *Home* memiliki desain yang cukup sederhana, dengan menggunakan palet warna antara biru, oranye, dan juga putih ini menyesuaikan dengan warna dari logo SLB Negeri Cicendo Bandung sebagai *client* utama dari projek ini. Halaman ini menampilkan logo dari website U-SIBI yang ditempatkan tepat di tengah halaman serta *loading bar* sebelum berpindah ke halaman berikutnya.



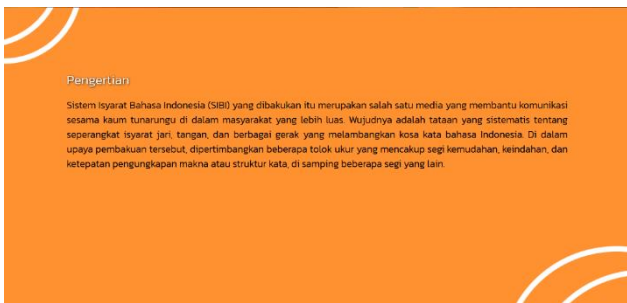
Gambar 5. halaman menu website U-SIBI.

Pada halaman *menu* terdapat gambar logo U-SIBI ditengah halaman serta tombol "*LIVE*" untuk fitur penerjemahan secara langsung tepat dibawah tulisan U-SIBI, menandakan bahwa fitur tersebut adalah inti dari *website* ini. Pada bagian atas halaman terdapat *navigation bar* yang berisi dua menu lain dari *website* ini yaitu "*All About SIBI*", "*Image Translate*", dan "*About Us*" yang berisikan informasi mengenai tim pengembang.



Gambar 6. halaman All About SIBI.

Halaman web U-SIBI ini memiliki antarmuka elemen desain melingkar putih dengan latar belakang oranye terang. Opsi navigasi disebut "History", "Basic", dan "Advance", dengan tiga tombol putih di tengah halaman yang sesuai dengan setiap opsi dengan ikon. Sisi kiri dan kanan halaman terdapat dua gambar tangan berwarna biru yang menunjukkan Bahasa isyarat, memberikan visual tambahan untuk memperkuat tema bahasa isyarat. Menu navigasi dengan link ke "Main Menu", "LIVE", "Image Translate", dan "About Us" terletak di bagian atas halaman. Halaman ini mudah digunakan dan mudah dipahami pengguna karena desain yang menarik.



Gambar 7. halaman History.

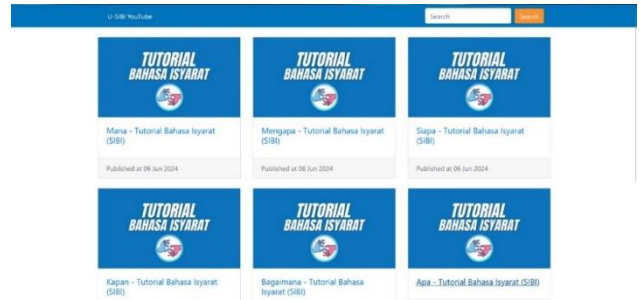
Setiap topik yang dibahas pada halaman History memiliki seksi halaman masing-masing sebesar 100vh (Viewport Height) saat pengguna menggulir halaman ke bawah, dan memberikan warna background yang berbeda untuk setiap seksi topik. Paragraf yang ada pada setiap topik ditulis langsung di dalam program HTML tanpa menggunakan database atau back-end.



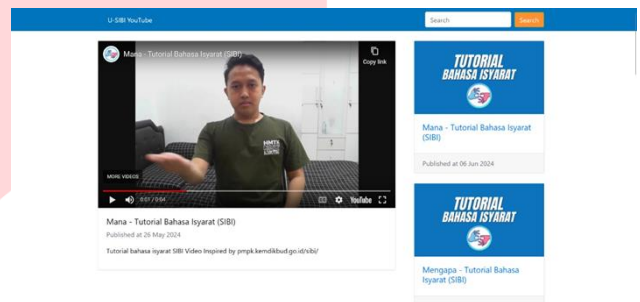
Gambar 8. halaman Basic.

Alfabet dan penomoran Sistem Bahasa Isyarat Indonesia (SIBI) divisualisasikan pada halaman web U-SIBI ini. Bentuk gambar tangan yang melambangkan huruf A sampai Z disusun dalam dua baris dengan latar belakang biru, sehingga membentuk "SIBI ALPHABET" di bagian atas halaman. Setiap gambar memiliki label oranye di bagian bawah. "NOMOR SIBI" ditampilkan di bagian bawah

halaman. Angka 1 sampai 10 digambar tangan dan disusun dalam dua baris dengan latar belakang biru dan label oranye di bawah setiap gambar. Pengguna akan mudah mempelajari dan mengidentifikasi simbol bahasa isyarat karena desain halamannya yang lugas namun bermanfaat.

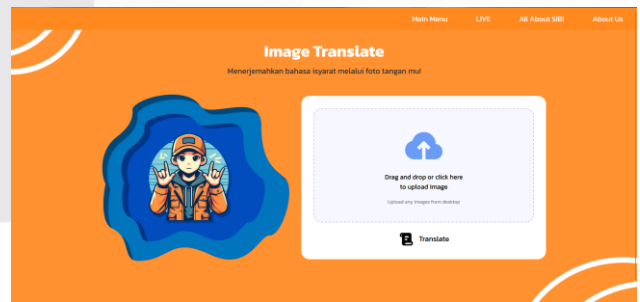


Gambar 9. halaman Advance.



Gambar 10. halaman Advance (2).

Tutorial yang disediakan akan berbentuk video atau MP4, dikarenakan bahasa isyarat yang ditampilkan memiliki pergerakan tangan dan perubahan raut wajah yang sesuai dengan kata yang disampaikan. Video tutorial yang akan ditampilkan pada halaman ini akan diakses dari playlist YouTube pada channel U-SIBI sendiri dan diperagakan oleh tim U-SIBI, pengambilan data akan dilakukan menggunakan YouTube Data API sehingga video tidak akan memakan memori, hanya memerlukan internet untuk mengakses video tutorial bahasa isyarat SIBI. Berikut adalah function dari halaman dengan tag: 'watch' untuk mengambil data dari YouTube.

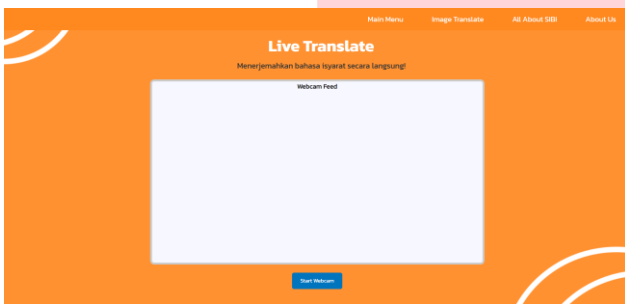


Gambar 11. halaman Image Translate.

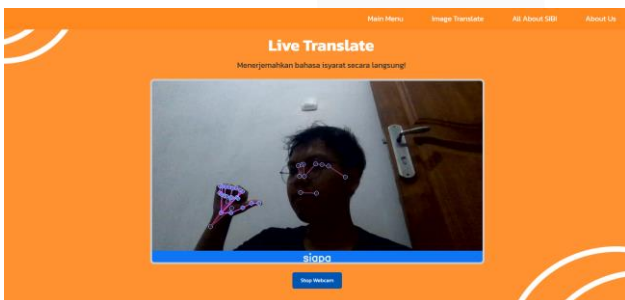


Gambar 12. halaman Image Translate (2).

Halaman web ini merupakan fitur "*Image Translate*" yang berfungsi menerjemahkan bahasa isyarat melalui foto tangan. Sisi luar halaman yang dibersihkan tampak garis besar karakter yang mengenakan topi dan mantel oranye, dikelilingi oleh alas bedak berwarna biru. Sisi kanan halaman berisi wilayah transfer gambar dengan konten "*drag and drop or click here*", yang memungkinkan klien mentransfer foto tangan mereka untuk penerjemahan.

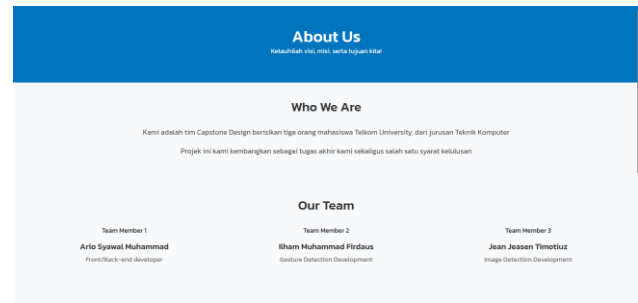


Gambar 13. halaman Live Translate.



Gambar 14. halaman Live Translate (2).

Halaman *web* U-SIBI ini menampilkan fitur "*Live Translate*" yang memungkinkan terjemahan langsung bahasa isyarat melalui *webcam*. Halaman ini memiliki latar belakang oranye dengan elemen desain lingkaran putih di sudut kiri atas dan kanan bawah. Di tengah halaman terdapat area kosong besar berlabel "*Webcam Feed*" tempat pengguna dapat melihat tampilan kameranya. Di bawah bagian ini terdapat tombol biru berlabel "*Start Webcam*" yang memungkinkan pengguna memulai sesi terjemahan langsung.



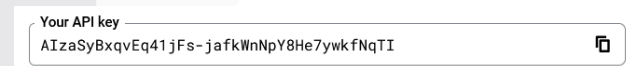
Gambar 15. halaman About Us.

Halaman ini akan berisikan nama anggota tim serta peran-perannya dalam pengembangan *website*. Visi, tujuan, dan ambisi tim pengembangan dijelaskan di bagian "*About Us*" di situs *web* U-SIBI ini.

2. Integrasi YouTube API.

API atau *Application Programming Interface*, adalah antarmuka yang memungkinkan perangkat lunak terhubung dan berinteraksi dengan sistem lain melalui teknologi web, seperti dalam bentuk layanan web. API sangat penting dalam pengembangan sistem perangkat lunak terdistribusi yang komponennya saling terhubung secara longgar. Aplikasi seperti aplikasi seluler, aplikasi berbasis cloud, aplikasi web, dan perangkat pintar sering menggunakan API untuk mengakses sumber daya data. Keunggulan API terletak pada kesederhanaannya yang membuatnya mudah untuk diintegrasikan dan digunakan oleh aplikasi lain, meskipun tidak memiliki antarmuka pengguna yang terlihat dan biasanya tidak berinteraksi langsung dengan pengguna akhir. API terutama digunakan untuk komunikasi antar mesin dan integrasi antar sistem perangkat lunak [6].

Sebuah *API-key* dibutuhkan untuk mengakses data dari *YouTube*. *API-key* merupakan suatu kode yang digunakan untuk mengidentifikasi dan mengautentikasi aplikasi atau pengguna. Kode API ini diperoleh melalui platform pasar seperti *google cloud console*. Selain itu, *API-key* juga berfungsi sebagai *identifier* dan menyediakan token yang dapat digunakan untuk proses autentikasi [7]. Token *API-key* dari *YouTube* dapat diperoleh secara gratis melalui *Marketplace Google Cloud Console* menggunakan produk "*YouTube Data API v3*". Berikut adalah token *API-key* yang telah diperoleh dari *YouTube Data API v3*.



Gambar 16. API Key YouTube Data

Terdapat dua metode inti yang harus dibuat untuk mengambil data dari *YouTube*, yang pertama adalah "*_videoLists*" untuk mendapatkan daftar video berdasarkan kata kunci dari pencarian yang akan dilakukan dan "*_singleVideo*" untuk mengambil detail setiap video yang dipilih seperti deskripsi, tanggal, dan lainnya. Ada enam parameter konfigurasi yang harus diambil melalui kedua metode tersebut.

Tabel 1. Parameter *YouTube API*

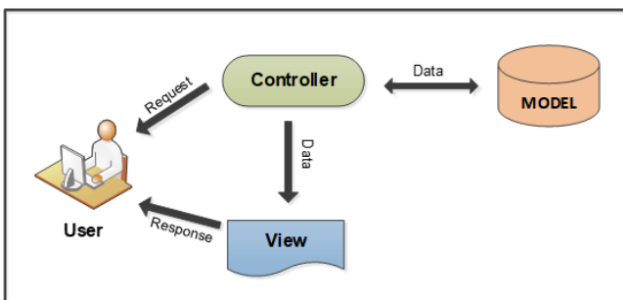
Parameter	Penjelasan
'part'	bagian data yang akan diambil dari API, yaitu 'snippet' yang mengandung informasi dasar video.
'country'	Mengambil kode negara dari sebuah video
'apiKey'	API Key yang diambil dari konfigurasi Laravel
'maxResults'	Jumlah maksimum hasil pencarian yang ingin diambil
'youTubeEndPoint'	Endpoint pencarian YouTube
'type'	jenis data yang akan diambil

3. Mengaplikasikan Flask.

Flask adalah kerangka web ringan yang ditulis dengan Python. Saat ini merupakan salah satu kerangka web Python yang paling banyak digunakan di kalangan startup dan secara luas dianggap sebagai alat sempurna untuk solusi cepat dan mudah bagi sebagian besar bisnis. Pada intinya, ini menyediakan serangkaian perpustakaan yang kuat untuk menangani tugas-tugas pengembangan web yang paling umum [8].

Algoritma yang telah dirancang dan dikembangkan untuk penerjemahan bahasa isyarat entah melalui gesture atau gambar dibangun menggunakan bahasa pemrograman *Python*. Untuk mengaplikasikan kedua algoritma ke dalam *environment* aplikasi web *Laravel* dibutuhkan semacam *environment* lain yang mendukung penggunaan bahasa pemrograman *Python*.

Laravel memerlukan controller untuk menghubungkan sebuah API atau model dari luar, dikarenakan *Laravel* adalah salah satu framework aplikasi web yang mengikuti arsitektur MVC atau *Model-View-Controller*. MVC adalah arsitektur aplikasi yang memfasilitasi pengembangan atau pemeliharaan aplikasi yang lebih efektif dan efisien dengan memisahkan antarmuka pengguna, data, dan antarmuka proses. Hal ini dimungkinkan karena pemrogram dan pengembang dapat lebih fokus pada proses bisnis dan membatasi pengembangan dan pemeliharaan pada komponen aplikasi yang terhubung langsung dengan aktivitas bisnis yang bersangkutan [9].



Gambar 17. Alur Kerja MVC.

3.1 Mengubah algoritma menjadi bentuk Flask.

Agar penggunaan flask lebih optimal, maka kedua algoritma LSTM dan YOLOv8 akan digabungkan menjadi satu program flask. Alhasil hanya perlu menjalankan satu server *flask* untuk menggunakan dua algoritma sekaligus.

Hal yang penting dalam pembuatan sebuah aplikasi *flask* adalah pembuatan rute aplikasi flask tersebut, pada umumnya menggunakan fungsi '@*app.route*'. Dalam aplikasi Flask, rute berfungsi untuk menghubungkan URL tertentu dengan metode Python yang mengelola permintaan. Rute berfungsi sebagai saluran untuk logika aplikasi yang diluncurkan saat mengakses titik akhir URL tertentu.

Tabel 2. App Routes Flask

Rute	Penjelasan
@ <i>app.route</i> ('/')	berfungsi untuk memberikan tanda atau verifikasi bahwa server <i>flask</i> sudah berjalan. Ini juga dapat berfungsi sebagai <i>endpoint</i> dasar.
@ <i>app.route</i> ('/video_feed')	menampilkan <i>video</i> secara <i>real-time</i> dari kamera pengguna, dengan deteksi pose dan tangan serta prediksi <i>gesture</i> yang ditampilkan pada <i>video</i> , rute ini memberikan umpan <i>video</i> langsung dengan hasil deteksi dan prediksi.
@ <i>app.route</i> ('/probabilities')	memberi tahu klien tentang tindakan yang telah diidentifikasi oleh model LSTM dan tingkat kepercayaan terhadap tindakan tersebut.
@ <i>app.route</i> ('/predict', methods=['POST'])	memungkinkan pengguna mengirimkan gambar untuk klasifikasi menggunakan model YOLOv8. Label kelas yang diantisipasi dan tingkat kepercayaan terkait dikembalikan oleh rute ini, yang menawarkan layanan prediksi gambar

Sebelum menjalankan aplikasi *flask* yang telah dibuat, harus melakukan *install* untuk setiap *requirements* atau *library* yang dibutuhkan oleh aplikasi *flask* tersebut. Untuk mempersingkat waktu *install*, semua *library* yang dibutuhkan beserta dengan versinya akan dimasukkan ke dalam satu *file.txt*.

```

File Edit View
Flask==2.1.1
opencv-python==4.5.5.64
numpy==1.22.4
mediapipe==0.10.11
Pillow==9.1.1
ultralytics==8.0.3
tensorflow==2.10.0
protobuf==3.20.*
  
```

Gambar 18. File requirements untuk aplikasi flask.

Aplikasi tersebut akan diisolasi menggunakan *Virtual Environment* atau 'venv'. Dengan mengaplikasikan 'venv' dapat mengelola dan menginstal paket *Python* secara lokal di lingkungan virtual tanpa memengaruhi proyek lain atau sistem utama *Python*. Hal ini memfasilitasi kontrol versi dan pengelolaan berbagai dependensi yang diperlukan oleh aplikasi Flask.

IV. HASIL DAN PEMBAHASAN

1. Stress Test Website.

Stress testing pada sebuah website dilakukan untuk melakukan evaluasi terhadap kemampuan suatu sistem atau aplikasi. Pada sebuah website pengujian ini biasanya digunakan untuk melihat kapasitas maksimum dalam menerima request dari banyak pengguna sekaligus, pengujian ini juga berguna untuk memonitor metrik performa seperti throughput, response time, dan penggunaan CPU dan memori. Stress tool yang akan digunakan untuk menguji website U-SIBI ada JMeter.

Pengujian awal dilakukan dengan menentukan tiga konfigurasi yaitu *Number of Threads* untuk mensimulasikan beberapa banyak pengguna sekaligus, *Ramp-Up Period* untuk waktu yang dibutuhkan untuk menambahkan pengguna, dan *Loop Count* untuk menentukan jumlah pengulangan simulasi yang terjadi.

Berdasarkan hasil pengujian awal pada halaman *menu*, *endpoint /menu* memiliki kinerja yang sangat responsif dengan waktu respons yang konsisten (waktu pengambilan sampel) kurang dari 200 ms. Semua permintaan berhasil diproses tanpa kesalahan apa pun. Hal ini menunjukkan bahwa *endpoint* ini stabil dan dapat diandalkan. Pengukuran data respons yang konsisten menunjukkan bahwa server menyajikan data terstruktur dan terkadang statis, yang menunjukkan efisiensi dalam pengelolaan konten. Selain itu, waktu koneksi yang sangat singkat menunjukkan bahwa server berada dalam kondisi optimal dan jaringan antara JMeter dan server sangat cepat, memungkinkan permintaan dan tanggapan hampir seketika. Berikut adalah tabel keseluruhan dari hasil *stress test* yang telah dilakukan.

Tabel 3. hasil Stress Test website U-SIBI.

No	Sample Time (ms)	Status (%)	Bytes (KB)	Sent Bytes (bytes)	Latency (ms)	Connect Time (ms)
1	112 – 153	100%	4,989	125	111 – 150	0 – 2
2	112 – 434	100%	4,989	125	117 - 177	0 – 1
3	0 – 9863	70.20% - 82.60%	11,66 – 179,48	0 – 128	0 – 9862	0 – 7
4	18018 – 44548	100%	2.428,79 – 8.937,66	127	1500 – 2559	1 – 2

- *Stress Testing /menu* dengan konfigurasi lebih berat.

Pada pengujian dengan konfigurasi yang tinggi, meskipun ada sedikit peningkatan dalam waktu respons dan latensi maksimum, *endpoint /menu* masih sangat responsif dan

menunjukkan kinerja yang konsisten. *Endpoint* ini ideal untuk digunakan dalam lingkungan produksi karena koneksinya tetap andal dan efisien. Pengujian menunjukkan bahwa server dapat menangani permintaan dengan baik dan menyediakan data terstruktur dan konsisten.

- *Stress Testing* semua halaman dengan konfigurasi lebih berat.

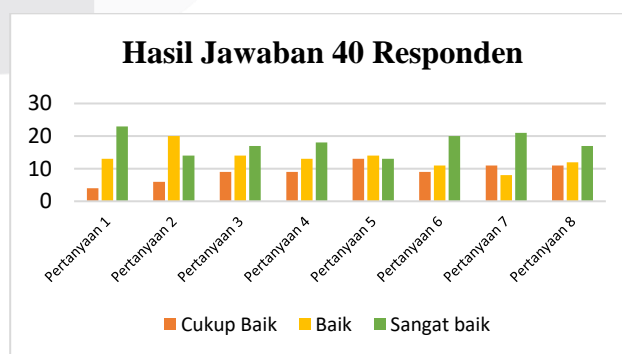
Waktu respons rata-rata untuk semua label sangat tinggi, dengan setiap label memiliki waktu respons lebih dari 2,8 detik, yang menunjukkan penurunan kinerja yang dapat berdampak signifikan pada pengalaman pengguna. Tingkat kesalahannya juga sangat tinggi, terutama 100% untuk halaman Watch. Hal ini menunjukkan bahwa hampir semua permintaan ke beberapa endpoint gagal. Throughput berada di antara 11,0 dan 15,0 permintaan per detik, yang cukup rendah mengingat waktu respons dan tingkat kesalahan yang tinggi, menunjukkan bahwa server kesulitan menangani beban.

- *Stress Testing* aplikasi Flask.

Hasil pengujian ini tidak memuaskan. Waktu respons yang sangat lama, rata-rata lebih dari 18 detik, mengindikasikan bahwa server membutuhkan waktu lama untuk memproses permintaan, yang dapat memengaruhi pengalaman pengguna dan mungkin memerlukan pengoptimalan server atau mengindikasikan bahwa mungkin ada kemacetan dalam kode aplikasi. Meskipun waktu respons yang lama, semua permintaan berhasil diproses tanpa kesalahan. Hal ini menunjukkan bahwa server stabil dan dapat diandalkan meskipun tidak dioptimalkan dalam hal kecepatan.

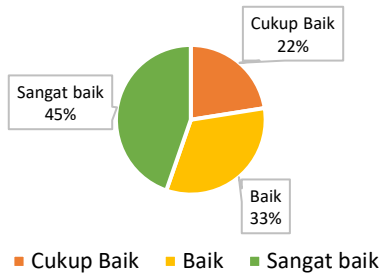
2. Pengujian Beta.

Mayoritas pengujian Beta akan dilakukan secara *online* di SLB Negeri Cicendo, namun cakupan data pengujian beta tidak harus berasal dari SLB Negeri Cicendo, siapa saja yang berminat untuk menjadi *tester* dapat melakukannya. *Tester* dalam hal ini adalah siswa dan guru di SLB Negeri Cicendo, para *tester* akan diberikan tugas untuk mencoba *website* U-SIBI hingga para *tester* berhasil melihat setiap fitur yang disediakan oleh *website*, terutama untuk melihat apakah informasi mengenai bahasa isyarat SIBI yang diberikan sudah benar.



Gambar 19. Visualisasi hasil jawaban 40 responden.

Total Keseluruhan Jawaban 8 Pertanyaan



Gambar 20. Visualisasi total keseluruhan jawaban.

Berdasarkan data *beta testing*, sebagian besar responden memberikan umpan balik yang sangat positif, dengan 45% dari jawaban dinilai "Sangat Baik" dan 33% "Baik," menunjukkan tingkat kepuasan yang tinggi. Meskipun demikian, terdapat 22% jawaban "Cukup Baik," yang mengindikasikan adanya area yang memerlukan perbaikan untuk meningkatkan kepuasan pengguna. Pertanyaan 1 dan 5 menerima jumlah tertinggi dari penilaian "Sangat Baik," menandakan aspek yang sangat diapresiasi oleh responden, sementara Pertanyaan 3 dan 6 memiliki jumlah "Cukup Baik" yang relatif tinggi, menunjukkan potensi perbaikan.

3. Pengujian Usability.

Hasil perhitungan usability testing berdasarkan jawaban dari responden pada beta testing sebagai berikut:

Tabel 4. Tabel nilai indeks usability test

Pertanyaan	Nilai Index	Keterangan
1	89.5%	Sangat Baik
2	84%	Sangat Baik
3	84%	Sangat baik
4	84.5%	Sangat Baik
5	80%	Baik
6	85.5%	Sangat Baik
7	85%	Sangat Baik
8	83%	Sangat Baik

Hasil *usability testing* pada tabel diatas didasarkan hasil kuesioner *beta testing*. Hasil perhitungan nilai indeks didapatkan dengan menggunakan rumus berikut:

$$\text{Nilai Indeks} = \frac{\sum \text{Nilai Skala} \times \text{Jumlah responden}}{\text{Nilai maksimum skala}}$$

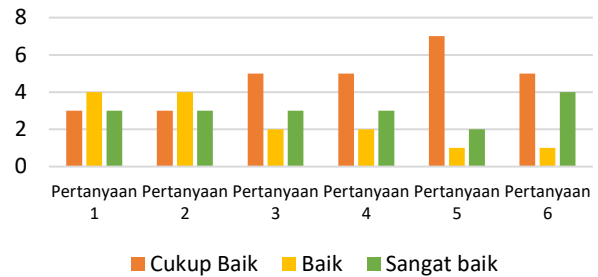
Berdasarkan hasil tabel diatas dapat disimpulkan bahwa para responden memiliki opini **Sangat Baik** untuk semua hal atau fitur yang bersangkutan dengan pertanyaan yang telah diberikan.

4. User Acceptance Test.

Tujuan UAT adalah untuk memastikan bahwa sistem produk mempunyai kualitas yang cukup untuk disetujui oleh

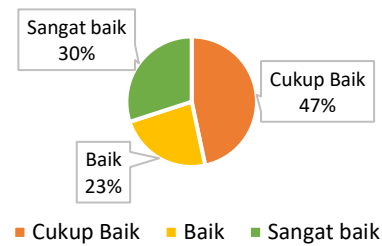
pengguna dan, pada akhirnya, sponsor atau klien. Penting untuk memikirkan kualitas sebagai sejauh mana hasil kerja memenuhi persyaratan yang ditentukan dan disimpulkan untuk menghasilkan sistem yang dapat digunakan. Definisi ini menyoroti dua poin penting: pertama, UAT ini tidak melibatkan pencarian setiap kelemahan kode, dan kedua, kualitas tidak mengharuskan setiap persyaratan dicatat untuk mendukung perbaikan masalah selama UAT [10].

Hasil Jawaban 10 Responden



Gambar 21. Visualisasi hasil jawaban 10 responden.

Total Keseluruhan Jawaban 6 Pertanyaan



Gambar 22. Visualisasi total keseluruhan jawaban.

Tabel 5. Nilai indeks UAT

Pertanyaan	Nilai Index	Keterangan
1	80%	Baik
2	80%	Baik
3	76%	Baik
4	76%	Baik
5	68%	Baik
6	78%	Baik

Hasil *User Acceptance Test* pada tabel diatas didasarkan hasil kuesioner UAT. Hasil perhitungan nilai indeks didapatkan dengan menggunakan rumus berikut:

$$\text{Nilai Indeks} = \frac{\sum \text{Nilai Skala} \times \text{Jumlah responden}}{\text{Nilai maksimum skala}}$$

Berdasarkan hasil tabel diatas dapat disimpulkan bahwa para responden memiliki opini **Baik** untuk semua hal atau

fitur yang bersangkutan dengan pertanyaan yang telah diberikan.

V. KESIMPULAN

Dari hasil pengujian usability yang didasarkan kuesioner beta testing. Usability testing memiliki rata-rata nilai indeks 84.4375% yang masuk ke dalam kategori **Sangat Baik**, berdasarkan skala likert yang telah dibuat. Ini dapat diartikan bahwa 40 orang peserta yang telah mengisi kuesioner beta testing memiliki pendapat yang tinggi mengenai fitur dan kegunaan dari website U-SIBI. 10 orang peserta dari SLB Negeri Cicendo Bandung memiliki pendapat yang cukup positif bahwa *website* penerjemah Bahasa isyarat U-SIBI dapat menjadi media pembelajaran yang cukup efektif dalam mempelajari bahasa isyarat SIBI.

REFERENSI

- [1] J. Khikam Hikmalansya, D. Cahyono, dan S. Surabaya, "Aplikasi Pembelajaran Bahasa Isyarat Berbasis Android," 2016.
- [2] kemdikbud, "Kamus SIBI Kerjasama antara Kementerian Pendidikan dan Kebudayaan dengan Lembaga Penelitian dan Pengembangan Sistem Isyarat Bahasa Indonesia," kemdikbud.go.id. Diakses: 19 Oktober 2023. [Daring]. Tersedia pada: <https://pmpk.kemdikbud.go.id/sibi/>
- [3] Chris Burke, "Advantages & Disadvantages of Sign Language," theclassroom.com. Diakses: 19 Oktober 2023. [Daring]. Tersedia pada: <https://www.theclassroom.com/advantages-disadvantages-of-sign-language-12084066.html>
- [4] admin_dti, "Framework : Pengertian, Fungsi, Cara Kerja, Tipe-tipe, Jenis-jenis," jakarta.telkomuniversity.ac.id. Diakses: 30 April 2024. [Daring]. Tersedia pada: <https://jakarta.telkomuniversity.ac.id/framework-pengertian-fungsi-cara-kerja-tipe-dan-jenis/>
- [5] R. Abdul, "Mengenal Framework 'Laravel,'" 2014.
- [6] Mathias Biehl, *API ARCHITECTURE* . 2015.
- [7] Fortinet, "What Is an API Key?," Fortinet.com. Diakses: 13 Mei 2024. [Daring]. Tersedia pada: <https://www.fortinet.com/uk/resources/cyberglossary/api-key>
- [8] Matt Copperwaite dan Charles Leifer, *Learning Flask Framework*. Birmingham B3 2PB, UK.: Packt Publishing Ltd., 2015.
- [9] O. Widodo Purbo, "Enrichment: Journal of Management is Licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0) Enrichment: Journal of Management A Systematic Analysis: Website Development using Codeigniter and Laravel Framework," 2021.
- [10] Rob Cimperman, *UAT Defined: A Guide to Practical User Acceptance Testing (Digital Short Cut)*. 75 Arlington St., Suite 300 Boston, MA 02116: Pearson Education, 2006.