Pengembangan Aplikasi Simulasi Sesi *Speaking* Tes *IELTS* Berbasis *Android*

1st Ayub Rosihan Ambarita
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ayubambarita@student.telkomuniversity.ac.id

2nd Casi Setianingsih
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
setiacasie@telkomuniversity.ac.id

3rd Astri Novianty
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
astrinov@telkomuniversity.ac.id

***Abstrak*—*International English Language Testing System* atau biasa dikenal dengan *IELTS* adalah sebuah tes kemampuan ystem Inggris yang diselenggarakan oleh Universitas *Cambridge*, *British Council*, dan IDP *Education Australia*. Dalam tes *IELTS* terdiri dari 4 sesi yaitu *Reading*, *Writing*, *Listening*, dan *Speaking*. Menurut ystem ic pada tahun 2022, sesi *Speaking* menempati posisi kedua dengan nilai terendah. Hal ini dikarenakan persiapan yang dilakukan untuk sesi *Speaking* ini cukup kompleks dan memakan waktu yang cukup lama. Saat ini terdapat simulasi yang diadakan oleh pihak resmi penyelenggara tes *IELTS*. Namum proses simulasi ini berbayar dan proses penilaian cukup lama. Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah mesin yang dapat mengevaluasi simulasi sesi *Speaking* tes *IELTS* secara langsung dan akurat. Untuk mengatasi permasalahan tersebut akan di kembangkan sebuah ystem *Cloud* yang menjalankan layanan API sebagai penghubung antara model *Machine Learning* dan *Android*. API akan menjalankan proses prediksi model Machine Learning terhadap *request* dari *Android* berupa penilaian terhadap matriks *Fluency*, *Lexical*, *Grammar* dan *Pronunciation*. Sehingga, hasil integrasi ystem keseluruhan ini akan menjadi sebuah aplikasi simulasi sesi *Speaking* tes *IELTS* berbasis *Android* yang dapat digunakan sebagai sarana pelatihan simulasi sesi *Speaking* tes *IELTS* yang dapat melakukan penilaian secara cepat dan murah.**

Kata kunci*— *IELTS*, *Speaking*, *Application Programming Interface, Cloud*, *Android

I. PENDAHULUAN

International English Language Testing System (IELTS) adalah sebuah tes kemampuan bahasa Inggris yang diselenggarakan oleh Universitas Cambridge, British Council, dan IDP Education Australia dan menjadi salah satu prasyarat wajib untuk seseorang yang ingin bekerja, belajar, atau bermigrasi ke negara yang menggunakan bahasa Inggris sebagai bahasa utama [1]. Dengan banyaknya manfaat yang diperoleh saat mendapatkan sertifikasi ini, maka banyak orang yang ingin mengikuti tes ini baik untuk kebutuhan akademik, umum, atau hanya sekedar ingin mendapatkan sertifikat ini. Tes ini terdiri dari 4 sesi yaitu Reading, Writing, Listening, dan Speaking [2]. Menurut statistik resmi yang dikeluarkan oleh pihak penyelenggara tes di tahun 2022, sesi Speaking menempati posisi kedua dengan nilai terendah [3]. Hal ini dikarenakan memerlukan proses pelatihan yang melelahkan dan kompleks karena perlunya untuk merekam percakapan diri sendiri ataupun mencari teman bicara untuk mendapatkan feedback dari pengucapan yang telah diucap. Terdapat beberapa matriks evaluasi pada sesi Speaking tes IELTS yaitu Fluency, Lexical, Grammar, dan Pronunciation dan di setiap matriks evaluasi memiliki kriteria masing masing di setiap band [4].

Untuk mempersiapkan sebelum mengikuti tes *IELTS*, terdapat simulasi atau latihan yang diadakan oleh pihak resmi penyelenggara tes *IELTS*. Simulasi ini mencakup format tes dengan berbagai contoh pertanyaan dan jawaban tes, untuk membantu peserta dalam mempersiapkan tes *IELTS* yang sebenarnya [5]. Biaya untuk melakukan simulasi di website resmi *IELTS* berkisar sekitar Rp700.000 [6] dan hasil penilaian berupa *feedback* report akan diberikan lima hari setelah pengambilan simulasi tes [7]. Dari kedua hal tersebut, banyak peserta yang mengurungkan niatnya untuk mengambil simulasi tes dikarenakan biaya yang cukup mahal dan proses menunggu hasil penilaian cukup lama.

Dari permasalahan yang didapatkan, maka diperlukan sebuah pengembangan sistem cloud untuk dapat menjalankan sebuah API yang menghubungkan antara model *Machine Learning* dan *Android*. Pengembangan sistem yang dilakukan adalah dengan membuat sebuah API yang berbasis pada *Framework Flask* berbasis bahasa pemrograman *Python*. Sistem API akan melakukan serving terhadap sistem prediksi oleh model *Machine Learning*, dan akan menyediakan sebuah *endpoint* yang akan diakses oleh *Android* untuk melakukan proses *request*. Sistem API ini akan di integrasikan dengan layanan *Cloud* yaitu *Google Cloud Platform (GCP)* sebagai media untuk melakukan *service* dalam melakukan proses prediksi dari model *Machine Learning* dan *Cloudinary* sebagai media penyimpanan data berupa audio dari setiap request yang diberikan oleh *Android*. Dengan sistem API yang terintegrasi, mampu mengoptimalkan penggunaan aplikasi yang dapat diakses secara online dan melakukan penilaian untuk mendapatkan hasil dari simulasi yang dilakukan pada aplikasi simulasi sesi *speaking* pada tes IELTS.

II. KAJIAN TEORI

A. IELTS

International English Language Testing System (IELTS) adalah sebuah tes kemampuan bahasa Inggris internasional terpopuler di dunia yang biasa digunakan untuk keperluan bekerja, belajar, atau bermigrasi ke negara yang

menggunakan bahasa Inggris sebagai bahasa utama [8]. Tes IELTS diselenggarakan dan dikembangkan oleh Universitas Cambridge, British Council, dan IDP Education Australia [1]. Tes IELTS dapat digunakan untuk 2 tujuan yaitu untuk kebutuhan akademik atau kebutuhan umum. Tes IELTS ini terdiri dari 4 sesi yaitu Reading, Writing, Listening, dan Speaking [2].

B. Application Programming Interface (API)

Application Programming Interface (API) adalah seperangkat aturan yang ditetapkan yang menjelaskan bagaimana komputer atau aplikasi berkomunikasi satu sama lain. API berada di antara aplikasi dan *server* website, bertindak sebagai lapisan perantara yang memproses transfer data antar sistem [9]. Terdapat banyak keuntungan menggunakan API seperti mempermudah kolaborasi, mudah dalam inovasi, dan menambah keamanan sistem. API memiliki berbagai jenis protokol yang dapat digunakan seperti *Simple Object Access Protocol* (SOAP), XML-RPC, JSON-RPC, dan *Representational State Transfer* (REST) yang saat ini banyak digunakan.

C. Flask

Flask adalah sebuah website *framework* yang ditulis menggunakan bahasa *Python*. *Framework* ini termasuk dalam jenis *microframework* karena tidak memerlukan suatu alat atau pustaka tertentu dalam penggunaannya. *Flask* dapat berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu website. Sehingga dengan *Flask* dapat membuat sebuah website yang terstruktur dan dapat mengatur perilaku suatu website dengan lebih mudah [10].

D. Docker

Docker merupakan suatu platform perangkat lunak yang memungkinkan pengguna untuk dengan cepat membuat, menguji, dan meluncurkan aplikasi. Dalam Docker, perangkat lunak dikemas dalam unit standar yang disebut sebagai "kontainer", yang mengandung semua komponen yang diperlukan untuk menjalankan perangkat lunak, termasuk pustaka, alat sistem, kode program, dan pengaturan waktu proses. Dengan memanfaatkan Docker, aplikasi dapat dengan cepat diimplementasikan dan ditingkatkan dalam berbagai lingkungan, dengan keyakinan bahwa kode akan beroperasi sesuai yang diharapkan [11].

E. Google Cloud Platform

Google Cloud Platform (GCP) merupakan sekumpulan layanan *cloud* publik yang disajikan secara langsung oleh pengembang di Google. GCP menyediakan berbagai layanan yang di-*hosting* untuk keperluan komputasi, penyimpanan, serta pengembangan aplikasi yang beroperasi pada infrastruktur *hardware Google* [12]. GCP menghadirkan berbagai layanan seperti Cloud Storage, Virtual Machine (VM) Instance, Container Registry, Cloud Run, dan lainnya.

F. Cloudinary

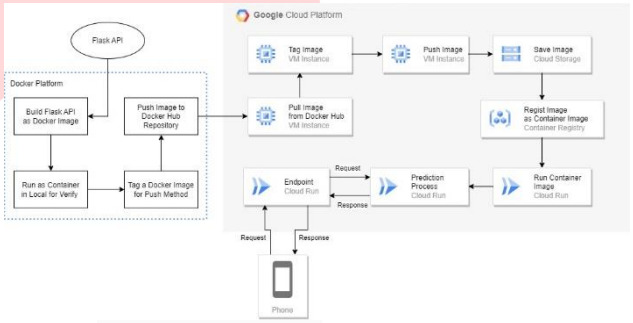
Cloudinary adalah solusi Software-as-a-Service (SaaS) untuk mengelola semua aset media aplikasi web atau seluler di dalam *cloud*. *Cloudinary* memberikan solusi lengkap untuk semua kebutuhan gambar dan video, termasuk unggah, penyimpanan, administrasi, transformasi, dan pengiriman yang dioptimalkan. Unggahan, pemrosesan, dan pengiriman media ditangani di *server Cloudinary*, yang secara otomatis dapat melibatkan beban tinggi dan lonjakan lalu lintas [13]

G. Python

Python adalah sebuah bahasa pemrograman dasar yang dirilis oleh Guido Van Rossum pada 1991. *Python* dapat digunakan untuk membuat aplikasi, perintah komputer, melakukan analisis data dan memungkinkan untuk mengimplementasikan pemrograman berbasis objek. Sebagai *general-purpose language*, *Python* bisa digunakan untuk membuat program apa saja dan menyelesaikan berbagai permasalahan. Selain itu, *Python* juga dinilai mudah untuk dipelajari dan termasuk bahasa pemrograman tingkat tinggi. Python banyak digunakan oleh banyak orang yang berprofesi *back-end* developer, *IT*, *ML* developer sampai *data scientist* [14].

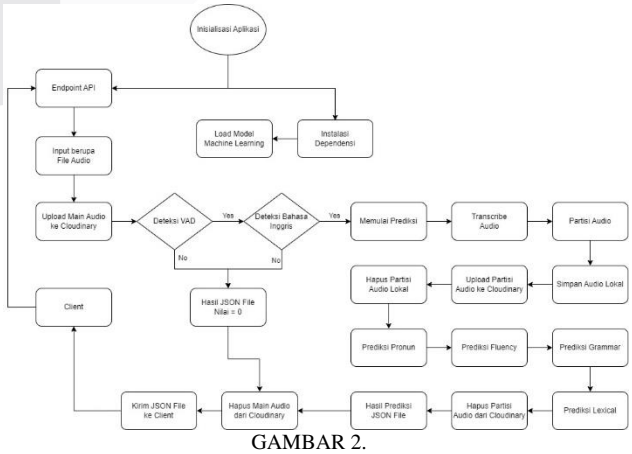
III. METODE

A. Desain Sistem



GAMBAR 1.
Desain Sistem Cloud

Gambar 1 merupakan desain sistem Cloud secara keseluruhan yang melakukan service terhadap sebagai proses utama dalam melakukan proses prediksi dengan model *Machine Learning*. *Google Cloud Platform dengan service Cloud Run* didalamnya, membuat API dapat diakses secara daring. API yang digunakan pada sistem adalah dengan menggunakan *framework Flask*. Sebelum di deploy ke dalam platform Google Cloud, bentuk API di build menjadi sebuah *Docker Image* terlebih dahulu sehingga yang berfungsi untuk menyesuaikan dan mengoptimalkan environment maupun dependensi yang digunakan dalam sistem API.



GAMBAR 2.
Sistem API

Gambar 2 merupakan sistem API yang berperan penting dalam menguhubungkan model *Machine Learning* (ML) dengan *Android*. Setiap tahap dari sistem API mulai dari inisialisasi aplikasi, menyediakan *endpoint* yang akan di gunakan oleh Android untuk melakukan request. Selanjutnya, dengan mempersiapkan penyimpanan cloud yang akan digunakan untuk menyimpan data hasil request dan penggunaan kembali data yang disimpan untuk dilanjutkan proses prediksi pada model *Machine Learning*. Sistem penyimpanan cloud yang digunakan adalah Cloudinary yang berfungsi sebagai media penyimpanan. Tahap selanjutnya dalam sistem API adalah proses prediksi data hasil *request* dari *Android* ke dalam model *Machine Learning*. Untuk dapat melakukan proses prediksi diperlukan model *Machine Learning* sebagai proses utama untuk mendapatkan hasil penilaian dari metriks penilaian IELTS. Hasil penilaian dari setiap proses prediksi dalam satu kali request akan dikembalikan ke *Android* dengan berupa file JSON. Perlu dicatat bahwa setiap audio yang disimpan setelah hasil prediksi didapatkan akan dihapus dari media penyimpanan Cloudinary. Sistem API ini dikembangkan untuk dapat diakses secara online dengan menggunakan Google Cloud Platform.

B. Analisis Kebutuhan Sistem

Berdasarkan desain sistem maka dibutuhkan beberapa aspek yang digunakan untuk selanjutnya diimplementasikan agar sistem API dapat berjalan. Berikut ini aspek yang dibutuhkan:

1. Analisis Perangkat yang Digunakan

Pada penelitian ini digunakan perangkat lunak untuk menunjang proses implemetasi sistem yang dibuat. Berikut ini spesifikasi perangkat lunak yang digunakan:

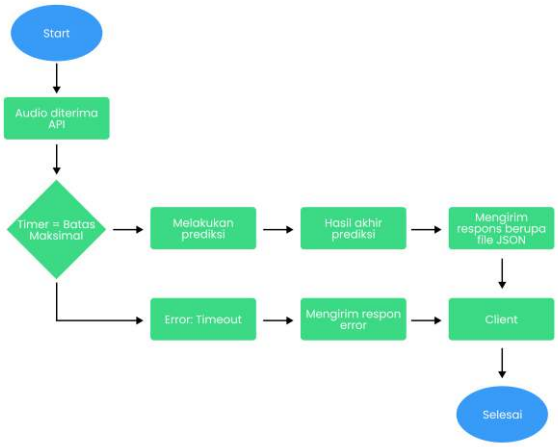
a. Bahasa pemrograman Python versi 3.8
b. Sistem Operasi Windows 10 Pro
c. Visual Studio Code untuk pembuatan kode program
d. Docker Desktop versi 4.21.1
e. Google Cloud Platform Console

C. Proses Pengembangan Sistem

Pengembangan yang dilakukan pada sistem ini dibagi menjadi:

1. Sistem Pengaturan Waktu *Timeout*

Pengaturan waktu respon API yang dibuat adalah pada saat *Android* mengirimkan masukan berupa sebuah data audio, dan API menerima masukan tersebut lalu diprediksi untuk mendapatkan hasil akhir, dan mengirimkan respon berupa file JSON. Pada sistem API, pengaturan waktu respon di buat dalam layanan *Google Cloud* pada fitur *Cloud Run*, yang memberikan layanan dan menjalankan Flask API yang telah di *deploy*.



GAMBAR 3.
Sistem Pengaturan Waktu *Timeout*

Pada saat API menerima masukan dari *Android*, API akan memberikan status pada log *Cloud Run* berupa proses jika audio sedang diproses untuk diprediksi, atau status berupa respon *timeout* ke *Android* jika audio yang dijalankan pada *Cloud Run* tidak selesai untuk memproses permintaan dalam waktu yang ditentukan.

Standar timeout yang dimiliki oleh layanan *Cloud Run* adalah selama 5 menit (300 detik), namun dalam sistem API ini *timeout* yang di atur adalah selama 450 detik. Lama timeout yang diatur pada *Cloud Run* dibuat berdasarkan lama waktu saat melakukan proses prediksi pada API secara lokal mulai dari *upload* audio hingga mendapatkan hasil akhir secara lokal.

2. Sistem API

Pengembangan sistem penghubung aplikasi atau *Application Programming Interface* (API) yang dilakukan adalah dengan mengimplementasikan struktur API dari *framework Flask*. Untuk dapat menggunakan *framework* ini memerlukan inisialisasi aplikasi dan membuat *endpoint* sehingga API dapat diakses oleh *Android* yang dapat dilihat pada gambar 2.

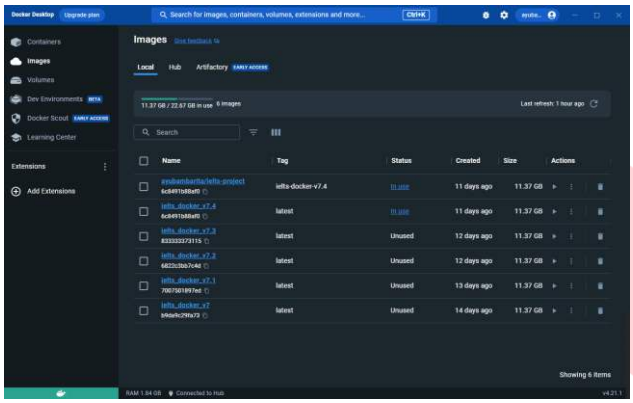
Dalam sistem ini untuk dapat mengakses API, *Android* dapat mengakses dua tipe *endpoint* dengan metode GET dan metode POST. *Endpoint* "/" dengan metode GET berfungsi untuk melakukan verifikasi apakah layanan API sudah berjalan, sedangkan untuk *endpoint* dengan metode POST untuk melakukan prediksi dengan model *Machine Learning*.

Pengaplikasi fungsi model *Machine Learning* di letakkan ke dalam dua parameter yang berbeda. Fungsi model *Machine Learning* perlu di definisikan terlebih dahulu pada parameter utama setelah aplikasi *Flask* di inisialisasikan, seperti fungsi untuk melakukan download model hingga load model *Machine Learning*. Sedangkan, fungsi untuk melakukan proses penilaian audio dengan model *Machine Learning* di aplikasikan pada fungsi metode POST dengan parameter "/*upload*", dimana pada akhir dari proses penilaan dari setiap *request* yang masuk, API akan mengembalikan sebuah respon berupa hasil dari proses penilaan file JSON.

Untuk memperlengkapi API dalam melakukan proses penilaan, diperlukan dependensi dari setiap fungsi baik itu dari segi model *Machine Learning*, aplikasi Flask API itu sendiri, hingga *cloud* penyimpanan data seperti *Cloudinary*.

Inisialisasi dependensi memiliki tujuan untuk mengaktifkan fungsi serta memudahkan dalam melakukan instalasi fungsi dan integrasi layanan dengan fitur yang di gunakan.

Dari gambar 1 dapat dilihat bahwa, Flask API di build ke dalam bentuk docker image terlebih dahulu dalam sebuah environment docker. Sebelum di deploy kedalam *Google Cloud*, *image* di jalankan sebagai *container* secara lokal untuk memastikan bahwa image berfungsi dengan baik dan kompatibel dengan *environment* lokal, sehingga membantu mengidentifikasi masalah potensial dan kesalahan.



GAMBAR 4.
API yang sudah dibentuk menjadi Docker Image

Setelah *container* dipastikan sudah berjalan dengan baik secara lokal, *image* yang akan di *deploy* perlu di simpan ke dalam sebuah repositori *docker hub* sebagai tempat penyimpanan versi yang berbeda dari setiap *image* sehingga memudahkan pengembang untuk mengelola *image* jika ada beberapa spesifikasi *image* tertentu yang ingin di gunakan kembali, dan memudahkan integrasi dengan *environment Cloud Run* yang digunakan pada sistem ini.

Proses *deployment* dalam *Google Cloud Platform* memiliki beberapa tahapan proses dengan menggunakan beberapa layanan diantaranya, *Cloud Run*, *Cloud Storage*, *VM Instance*, *Container Registry*. Layanan ini memiliki fungsi masing-masing yang berhubungan satu sama lain dalam melakukan proses *deploy docker image* yang telah dibuat sebelumnya. *VM Instance* digunakan sebagai media untuk melakukan proses *download docker image* secara spesifik dari repositori *Docker Hub*, selanjutnya *Cloud Storage* digunakan untuk menyimpan *docker image* dalam format file "*versi_image.tar*" dan menyimpan docker image ke dalam *Container Registry* untuk dapat di *deploy* ke dalam layanan *Cloud Run*.

Setelah *docker image* di simpan dalam *Container Registry*, *Cloud Run* akan melanjutkan proses terakhir dalam deployment dengan menjalankan image sebagai container yang akan dijadikan sebagai *service* API untuk dapat menghasilkan sebuah link yang dapat diakses oleh *Android* untuk melakukan verifikasi API dan melakukan proses prediksi model *Machine Learning*.

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Respon Timeout

Pengujian yang dilakukan untuk mengukur waktu respon *timeout* model *machine learning* membutuhkan data audio yang memiliki durasi berbeda-beda dan masing-masing durasi memiliki 3 jenis data yang berbeda pula

TABEL 1.
Data audio untuk pengujian respon timeout

No.	Durasi Audio	Ukuran Data ke-1	Ukuran Data ke-1	Ukuran Data ke-1
1.	10 s	264 KB	47 KB	47 KB
2.	20 s	313 KB	317 KB	501 KB
3.	30 s	714 KB	456 KB	703 KB
4.	60 s	954 KB	985 KB	1438 KB
5.	120 s	1748 KB	1808 KB	1788 KB
6.	135 s	1983 KB	1990 KB	2004 KB
7.	150 s	2398 KB	2199 KB	2391 KB
8.	165 s	2469 KB	2391 KB	2384 KB
9.	180 s	2810 KB	2786 KB	2810 KB
10.	240 s	3775 KB	3771 KB	3741 KB
11.	300 s	3902 KB	4030 KB	4047 KB

Hasil pengujian respon timeout dapat dilihat pada tabel berikut:

TABEL 2.
Hasil pengujian respon timeout

No	Durasi Audio	Total Pengujian setiap data ke-n		
		Pengujian Data ke-1	Pengujian Data ke-2	Pengujian Data ke-3
1.	Audio 10 s	16.468 s	23.838 s	20.048 s
2.	Audio 20 s	34.179 s	28.543 s	31.288 s
3.	Audio 30 s	38.291 s	36.36 s	46.768 s
4.	Audio 60 s	53.785 s	66.133 s	66.696 s
5.	Audio 120 s	142.375 s	173.094 s	152.493 s
6.	Audio 135 s	160.629 s	148.988 s	167.825 s
7.	Audio 150 s	199.598 s	201.494 s	172.356 s
8.	Audio 165 s	184.467 s	271.273 s	188.625 s
9.	Audio 180 s	199.067 s	237.127 s	213.944 s
10.	Audio 240 s	287.946 s	261.749 s	269.817
11.	Audio 300 s	*Timeout*	*Timeout*	*Timeout*

Dari pengujian yang dilakukan sebanyak 33 kali pengujian untuk mengukur waktu respon timeout, diperoleh perhitungan akurasi uji alfa sebagai berikut:

$$Uji\ Alfa = \frac{Total\ Berhasil}{33\ x\ 1\ Aksi}\ x\ 100\% = \frac{30}{33}\ x\ 100\% = 90\%$$

Dari pengujian yang dilakukan, didapatkan hasil nilai uji alfa terhadap sistem API untuk pengujian waktu respon timeout adalah 90%.

Hasil dari pengujian yang dilakukan didapatkan kesimpulan bahwa berdasarkan pengaturan waktu respon timeout pada Cloud Run dalam rentang waktu dari 0 sampai 450 detik, terdapat 30 data pengujian yang berhasil dicatat waktu responnya. Sedangkan, 3 data pengujian dengan durasi 300 detik menghasilkan output *timeout error* dikarenakan proses prediksi yang dilakukan sudah lebih dari rentang

waktu maksimal yaitu 450 detik. Maka dapat dikatakan sistem berjalan 90% sesuai dengan tujuan yang diinginkan

B. Hasil Pengujian Waktu Proses API

Pengujian kedua yang dilakukan terhadap API adalah proses pengujian untuk menghitung berapa lama waktu yang diperlukan dalam melakukan proses instalasi model *Machine Learning* dan lama waktu dalam melakukan proses prediksi terhadap request yang diberikan kepada API.

1. Waktu Proses Instalasi model *Machine Learning*

TABEL 3.
Hasil pengujian waktu proses instalasi model *Machine Learning*

No.	*Load Model Process Name*	*Duration* (s)
1.	*Whisper Language Detect Process*	8.6168 s
2.	*Whisper Audio Speech Recognition Process*	8.5994 s
3.	*Grammar Load Model Process*	22.9103 s
4.	*Lexical Load Model Process*	11.2347 s
5.	*Pronunciaton Load Model Process*	24.5863 s
6.	*Fluency Check Model Process*	0.0002 s
7.	*Fluency Load Model Process*	0.4909 s
8.	*Voice Activity Detector Process*	1.4624 s
9.	*Errant Load Model Process*	0.7894 s
10.	*Natural Language Processing Model*	0.5097 s
11.	*Total Load Model Process*	79.2008 s

Instalasi model Machine Learning dilakukan hanya sebanyak satu kali, ketika pertama kali API berjalan di dalam *service Cloud Run*. Waktu total yang diperlukan dalam melakukan instalasi dependensi model *Machine Learning* adalah 78.7319 s, setelah instalasi sistem API selesai dilakukan maka aplikasi sudah berada dalam status siap untuk menerima *request*.

2. Waktu Proses Prediksi model *Machine Learning*

Pengujian yang dilakukan untuk mengukur waktu proses prediksi model machine learning adalah dengan menggunakan data audio yang memiliki durasi berbeda-beda dan masing-masing durasi memiliki 3 jenis data yang berbeda pula.

TABEL 4.
Data audio yang diuji

No.	Durasi Audio	Ukuran Data ke-1	Ukuran Data ke-1	Ukuran Data ke-1
1.	10 s	264 KB	47 KB	47 KB
2.	20 s	313 KB	317 KB	501 KB
3.	30 s	714 KB	456 KB	703 KB
4.	60 s	954 KB	985 KB	1438 KB
5.	120 s	1748 KB	1808 KB	1788 KB
6.	135 s	1983 KB	1990 KB	2004 KB
7.	150 s	2398 KB	2199 KB	2391 KB
8.	165 s	2469 KB	2391 KB	2384 KB
9.	180 s	2810 KB	2786 KB	2810 KB
10.	240 s	3775 KB	3771 KB	3741 KB
11.	300 s	3902 KB	4030 KB	4047 KB

Parameter yang diukur dalam pengujian waktu proses prediksi ini adalah sebagai berikut:

TABEL 5.
Parameter pengujian waktu proses prediksi

No.	Parameter pengujian waktu proses prediksi
1.	Upload main audio ke *Cloudinary*
2.	Proses *Voice Activity Detector*
3.	Proses *Language Detect*
4.	Proses Dalam *Transcribe*
5.	Proses Prediksi *Pronun*
6.	Proses Prediksi *Fluency*
7.	Proses Prediksi *Grammar*
8.	Proses Prediksi *Lexical*
9.	Proses hapus main audio dari *Cloudinary*

Dari 9 parameter pengujian waktu proses prediksi pada tabel 5, akan diambil akumulasi seluruh parameter sesuai dengan audio yang diuji berdasarkan durasinya.

Dengan data audio yang diuji pada tabel 4, didapatkan masing-masing total pengujian dari seluruh parameter sebagai berikut

TABEL 6.
Hasil pengujian waktu proses prediksi model *Machine Learning*

No.	Durasi Audio	Total Pengujian setiap data ke-n		
		Pengujian Data ke-1	Pengujian Data ke-2	Pengujian Data ke-3
1.	Audio 10 s	16.468 s	23.838 s	20.048 s
2.	Audio 20 s	34.179 s	28.543 s	31.288 s
3.	Audio 30 s	38.291 s	36.36 s	46.768 s
4.	Audio 60 s	53.785 s	66.133 s	66.696 s
5.	Audio 120 s	142.375 s	173.094 s	152.493 s
6.	Audio 135 s	160.629 s	148.988 s	167.825 s
7.	Audio 150 s	199.598 s	201.494 s	172.356 s
8.	Audio 165 s	184.467 s	271.273 s	188.625 s
9.	Audio 180 s	199.067 s	237.127 s	213.944 s
10.	Audio 240 s	287.946 s	261.749 s	269.817
11.	Audio 300 s	*Timeout*	*Timeout*	*Timeout*

Pengujian untuk aksi nomor 1 dilakukan sebanyak 1 kali pengujian, dari pengujian tersebut diperoleh perhitungan akurasi uji alfa sebagai berikut:

$$Uji\ Alfa = \frac{Total\ Berhasil}{1\ x\ 1\ Aksi}\ x\ 100\% = \frac{1}{1}\ x\ 100\% = 100\%$$

Untuk pengujian selanjutnya, dilakukan sebanyak 33 kali pengujian untuk aksi nomor 2, dari pengujian tersebut diperoleh perhitungan akurasi uji alfa sebagai berikut:

$$Uji\ Alfa = \frac{Total\ Berhasil}{33\ x\ 1\ Aksi}\ x\ 100\% = \frac{30}{33}\ x\ 100\% = 90\%$$

Dari kedua pengujian yang dilakukan, didapatkan hasil nilai uji alfa terhadap setiap aksi yang diuji pada sistem API adalah 100% untuk pengujian waktu proses instalasi model Machine Learning dan 90% untuk pengujian waktu proses prediksi model *Machine Learning*.

V. KESIMPULAN

Hasil dari pengujian yang dilakukan, didapatkan bahwa sistem penghubung aplikasi (API) dapat beroperasi pada aplikasi dengan baik.

Dari pengujian sistem pengaturan waktu Respon API yang dilakukan dengan mengamati secara langsung menggunakan *Logs Explorer*, dapat disimpulkan bahwa semua *request* yang dikirim ke API akan berhasil (tanpa respon *request timeout error*) pada rentang durasi audio 0 sampai 240 detik, sehingga persentase yang didapatkan dari proses pengujian waktu respon *timeout* adalah sebesar 90%.

Dilihat dari pengujian pada sistem API di dapatkan bahwa seluruh aksi pada skenario pengujian alfa sistem API dikatakan berhasil sesuai dengan hasil yang di inginkan, dengan hasil akhir bahwa API berhasil melakukan *load* model *Machine Learning* ketika API pertama kali dimulai sebanyak 1 kali dengan persentase sebesar 100% dan API berhasil mencatat waktu proses prediksi yang ditampilkan pada *Log Explorer* untuk melakukan *monitoring* terhadap *service* yang dilakukan *Cloud Run* ketika melakukan prediksi dari setiap *request* sebanyak 33 kali dengan persentase sebesar 90%.

REFERENSI

[1] Universitas Cambridge; British Council; IDP Education Australia, “What is IELTS?,” [Online]. Available: https://www.ielts.org/about-ielts/what-is-ielts. [Diakses 17 Oktober 2022].

[2] A. Hashemi dan S. Daneshfar, “A Review of the IELTS Test: Focus on Validity, Reliability, and Washback,” *Indonesian Journal of English Language Teaching and Applied Linguistics,* vol. 3, no. 1, pp. 42-43, 2018.

[3] IELTS, “Test taker performance 2022,” [Online]. Available: https://www.ielts.org/for-researchers/test-statistics/test-taker-performance. [Diakses 1 Agustus 2023].

[4] IELTS, “IELTS scoring in detail,” [Online]. Available: https://www.ielts.org/for-organisations/ielts-scoring-in-detail. [Diakses 1 Agustus 2023].

[5] Universitas Cambridge; British Council; IDP Education Australia, “IELTS practice test,” [Online]. Available: https://www.ielts.org/usa/ielts-practice-test. [Diakses 12 November 2022].

[6] Universitas Cambridge; British Council; IDP Education Australia, “IELTS Academic – Practice Test 1 (Timed),” IELTS Progress Check, [Online]. Available: https://www.ieltsprogresscheck.com/product/ielts-academic-practice-test-1-timed/. [Diakses 12 November 2022].

[7] Universitas Cambridge; British Council; IDP Education Australia, “What is IELTS Progress Check?,” IELTS Progress Check, [Online]. Available: https://www.ieltsprogresscheck.com/#:~:text=What%20is%20IELTS%20Progress%20Check%3F. [Diakses 12 November 2022].

[8] IDP Education, “Apa itu IELTS?,” [Online]. Available: https://www.idp.com/indonesia/ielts/what-is-ielts/. [Diakses 1 Agustus 2023].

[9] IBM, “What is an API?,” [Online]. Available: https://www.ibm.com/topics/api. [Diakses 1 December 2022].

[10] R. Maulid, “Mengenal Flask, Library Machine Learning Python Idaman Developer,” Yayasan Multimedia Nusantara & Xeratic, 1 September 2021. [Online]. Available: https://dqlab.id/mengenal-flask-library-machine-learning-python-idaman-developer. [Diakses 30 July 2023].

[11] Amazon Web Services, “Apa itu Docker?,” Amazon Web Services, [Online]. Available: https://aws.amazon.com/id/docker/. [Diakses 30 July 2023].

[12] Ngoolie Media Tech, “Mengenal Google Cloud Computing: Apa itu, Produk, Foundation,” Ngoolie Media Tech, 10 Desember 2022. [Online]. Available: https://ngoolie.id/komputer/google-cloud-computing/. [Diakses 1 Desember 2022].

[13] Cloudinary, “Get Started With Cloudinary,” 17 Mei 2023. [Online]. Available: https://cloudinary.com/documentation/cloudinary_get_started. [Diakses 2 Agustus 2023].

[14] Algoritma Data Science Academy, “Mengenal Lebih Jauh Apa Itu Python Dan Kegunaannya,” Algoritma Data Science Academy, 22 April 2022. [Online]. Available: https://algorit.ma/blog/data-science/apa-itu-python-2022/. [Diakses 1 Desember 2022].