

Sign Language Translator Using Deep Learning

1st Hanifah Marta Ardilah

Falkutas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

hanifahmrtardilah@student.telkomuniversity.ac.id

2nd Casi Setianingsih

Falkutas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

3rd Randy Erfa Saputra

Falkutas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

resaputra@telkomuniversity.ac.id

Abstrak— Penelitian ini menjelaskan tentang salah satu permasalahan komunikasi yang terjadi di kehidupan masyarakat. Permasalahan komunikasi tersebut terjadi antara teman tuli dan teman dengar. Teman dengar berkomunikasi secara verbal, sedangkan teman tuli berkomunikasi dengan non-verbal. Untuk teman tuli berkomunikasi dengan menggunakan bahasa isyarat sebagai medianya. Solusi paling umum untuk saat ini adalah dengan menggunakan orang lain sebagai penerjemah untuk berkomunikasi antara teman tuli dan teman dengar. Tetapi, solusi tersebut tidak menjawab solusi yang efektif dikarenakan seorang penerjemah tidak setiap waktu akan tersedia, berbeda dengan program komputer.

Solusi yang diusulkan menerapkan teknik deep learning LSTM dan YOLOv5, sistem akan mengartikan bahasa isyarat SIBI. Selain itu, melalui pengembangan aplikasi Android dan pengembangan Backend memastikan operasi aplikasi yang andal dan efisien.

Produk berupa aplikasi dengan fitur deep learning LSTM untuk fitur deteksi motion dengan akurasi 95,56%, YOLOv5 untuk deteksi realtime memiliki mAP@0.5IoU 99,5%, dan LSTM untuk Text-Correction 92,5%. Kemudian animasi gerakan SIBI untuk fitur pembelajaran, dan implementasi ke Android melalui backend. Pada tes reliabilitas aplikasi ditemukan bahwa nilai r_{11} adalah 0,914428422. Oleh karena itu, dapat disimpulkan bahwa pengujian ini memiliki tingkat reliabilitas yang sangat tinggi. Respon waktu dalam hasil tes server juga menunjukkan dikisaran 1-5 detik dan memory usage rata-rata 130mb.

Kata kunci— SIBI, Machine Learning, Android, Backend, Animasi

I. PENDAHULUAN

Menurut laporan BPS, pada tahun 2011 yang telah diperbaharui pada Juni 2021, jumlah teman tuli di Bandung sebanyak 219 [1]. Bahasa yang digunakan antara teman tuli dan teman dengar untuk melakukan komunikasi mempunyai perbedaan. Komunikasi pada dasarnya merupakan kebutuhan setiap orang, yang dikatakan sesuatu yang esensial untuk kehidupan manusia. Komunikasi dibagi menjadi 2 macam yaitu komunikasi verbal dan komunikasi non-verbal. Pada dasarnya komunikasi yang biasa digunakan pada kehidupan sehari-hari adalah komunikasi verbal. Tetapi, teman tuli sulit untuk menggunakan komunikasi yang dilakukan secara verbal. Komunikasi yang teman tuli lakukan adalah menggunakan bahasa isyarat. Bahasa isyarat merupakan bahasa yang tidak memanfaatkan suara dalam berkomunikasi, melainkan memanfaatkan komunikasi manual, bahasa tubuh dan gerak bibir. Dalam berkomunikasi

tersebut akan mengkombinasikan bahasa tersebut dengan bentuk tangan, orientasi dan gerak tangan, lengan dan tubuh, serta ekspresi wajah untuk mengutarakan pikirannya [2]. Sehingga, terdapat keterbatasan komunikasi antara teman tuli dan teman dengar. Di Indonesia, ada 2 standar bahasa isyarat yaitu BISINDO dan SIBI. Yang akan digunakan pada penelitian ini SIBI. SIBI adalah bahasa isyarat yang sudah diresmikan oleh pemerintah yang digunakan dalam pembelajaran di SLB (Sekolah Luar Biasa) yang diambil dari referensi ASL (diambil dari wawancara Ibu Rini selaku guru di SLB Negeri Cicendo). SIBI telah diatur dalam Undang-Undang Nomor 2 Tahun 1989 dan telah dibakukan oleh Menteri Pendidikan dan Kebudayaan pada 30 Juni 1994 bahwa SIBI adalah bahasa isyarat yang telah diresmikan oleh pemerintah yang digunakan sebagai standar di SLB [3].

Pada zaman modern ini hampir setiap penduduk mampu mengakses teknologi, seperti contoh penggunaan *smartphone* berbasis Android. Sistem operasi yang paling banyak digunakan di Indonesia adalah Android, terbukti dari penggunaannya yang mencapai 89.77% per Oktober 2022 [4]. Sehingga, salah satunya dapat menggunakan pengembangan suatu aplikasi sebagai media pembelajaran berbasis Android. Dengan ini diharapkan dapat membantu pembelajaran bahasa isyarat secara mandiri. Oleh karena itu, tujuan dikembangkannya aplikasi media untuk pembelajaran bahasa isyarat lebih praktis dan sebagai alat pendamping dalam proses berkomunikasi satu sama lain. Dalam pengembangan *developer* Android juga bersifat *open-source* yang berarti dapat dimodifikasi oleh *developer*. Oleh karena itu, dengan adanya permasalahan tersebut peneliti tertarik melakukan penelitian yang berjudul, “Sign Language Translator Using Deep Learning”.

II. KAJIAN TEORI

A. Backend API aplikasi

pertama kita akan membuat code API menggunakan bahasa pemrograman JavaScript dengan *framework* NodeJS dengan menggunakan database server PostgreSQL. Dimana ada table yaitu *usercontributor*, *article*. Lalu untuk fitur aplikasi lainnya yang telah dibuat oleh anggota yang bertanggung jawab salah satunya *text to animation* sebelum diintegrasikan ke lingkungan server dilakukan pengecekan kode dan memastikan tidak ada kode yang error sehingga mempermudah dalam mendeploy dan implementasinya.

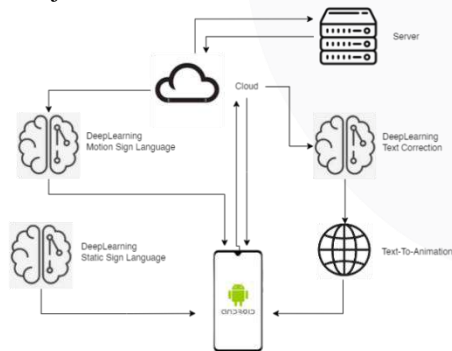
Berikut kode untuk backend API user autentifikasi menggunakan kode otp.

B. Server On Premises

Server sebagai *resource* dalam pengembangan dan menjalankan aplikasi *backend* serta fitur lainnya yang akan di produksi. Dengan instalasi sistem operasi yang digunakan yaitu Ubuntu yang di install pada server. Sistem operasi tersebut akan menjadi dasar untuk mengelolah aplikasi backend dengan menginstall database server yaitu PostgreSQL dan untuk mengelolah *backend* agar menjadi endpoint *public* ada beberapa *tools* yang diinstall pada vm instance ubuntu seperti docker, kuberspray, ansible, metallb dan server mengubah endpoint API dari alamat ip local menjadi endpoint menggunakan domain URL lalu memonitoring resource server dan aplikasi dari endpoint domain tersebut.

III. METODE

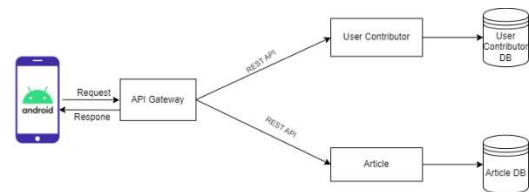
Pada penelitian ini Sistem terbagi menjadi beberapa sub sistem seperti pada Gambar 3.1, setiap sub sistem akan terintegrasi dan akan mempengaruhi kerja subsistem lainnya yang akan membentuk satu sistem keseluruhan. Arsitektur tersebut merupakan arsitektur utama dari sistem yang akan dirancang. Aplikasi yang dirancang berbasis Android menggunakan bahasa pemrograman Kotlin yang dikerjakan oleh rekan team saya Luthfi Yafi Alfiansyah. Teknologi yang digunakan pada sistem adalah *Artificial Intelligence* dan *Machine Learning* Serta Server. *Artificial Intelligence* digunakan sebagai penerjemah dari gerakan bahasa isyarat ke dalam teks bahasa Indonesia, sedangkan *Machine Learning* digunakan untuk melakukan pembuatan model pada sistem yang dikerjakan oleh rekan team saya Nurul amelia dan fikri putra hidayat. Selain itu juga terdapat fitur text-to-animation yang dikerjakan oleh rekan team saya rizqi alpriansyah. User bisa menginputkan kata atau kalimat dan nantinya akan dirubah menjadi sebuah video animasi.



GAMBAR 3.1
Arsitektur utama sistem

Adapun pada pembuatan pada aplikasi ini backend API yang akan diimplementasikan ke server on premises dan cloud VPS.

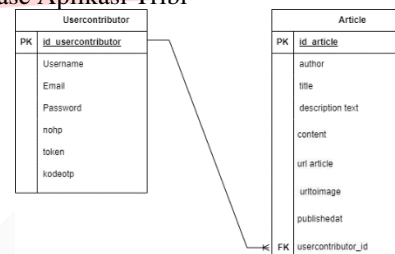
A. Arsitektur REST API



GAMBAR 3. 2
Arsitektur REST API

Aplikasi *back end* menggunakan REST API sebagai intraksi antara client dan data dengan melalui metode HTTP *request* yaitu GET, POST, PUT, DELETE. Lalu untuk Android mengakses backend API melalui API Gateway sebagai pintu gerbang yang mengelolah permintaan android melakukan request HTTP ke backend. API gateway akan meneruskannya ke REST API sesuai dengan request pada client atau Android lalu API gateway menerima respon dari REST API dan mengirim respon tersebut ke Android sesuai dengan request dan Android akan memproses data sesuai dengan kebutuhan.

B. Database Aplikasi Tribi

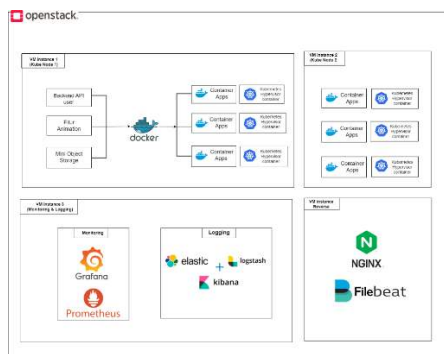


GAMBAR 3. 3
Entity Relationship Diagram Tribi

Untuk table *usercontributor* memiliki beberapa kolom sebagai identitas dari *user* dimana *user* memasukan email, *password*, *photo profile* dan *nohp* setelah memasukan email untuk memverifikasi bahwa email dari *user* yang telah inputan akan dikirimkan kode otp ke email user yang dimana kode otp telah digenerate berupa angka dan huruf acak terdiri dari 4 digit karakter tujuannya untuk memverifikasikan bahwa email yang dimasukkan benar-benar pemilik sah dari akun tersebut. Pada table artikel berisi mengenai artikel yang berkaitan SIBI, disabilitas serta pembelajaran Bahasa isyarat artikel diambil dari website yang telah dipublish.

C. Server On Premises

Pada server on premises dengan menggunakan opensource yaitu Openstack untuk bisa install beberapa VM instance agar dapat mempermudah dalam mengelolah database dan *storage*. Dengan membuat VM sistem operasi ubuntu didalamnya dapat mengatur dan mengkonfigurasi lingkungan pengembangan yang sesuai, seperti menjalankan web server, database server, dan bahasa pemrograman yang dibutuhkan untuk aplikasi.



GAMBAR 3.4
Desain Server VM

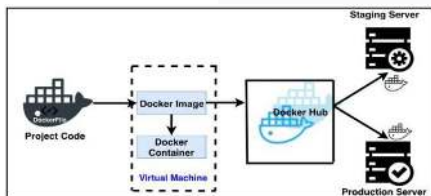
1. Virtual Machine

Didalam vm ini terdapat ada beberapa instalasi seperti docker, *tools provisioning* dari kubernetes yaitu kubespray, ansible, metallb, database server dan *storage*. Dimulai dengan menginstall docker terlebih dahulu dimana docker semua depensi yang digunakan dalam pembuatan backend akan dimasukkan ke docker image untuk instruksi dalam pembuatan docker image yaitu Dockerfile. Berikut instalasi docker dan dockerfile pada vm ubuntu.

```
ubuntu@ubuntu-dev-1:~$ sudo apt install docker.io -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (20.10.21-0ubuntu1~20.04.1).
0 upgraded, 0 newly installed, 0 to remove and 39 not upgraded.
```

GAMBAR 3.5
Instalasi Docker

Setelah docker berhasil terinstall maka selanjutnya akan membuat dockerfile berisi seperti *frame work* pendukung yang digunakan pada aplikasi yang akan dideploy ke docker tersebut.



GAMBAR 3.6
Arsitektur Docker [5]

Pada aplikasi dimana hasil kodingan *backend* yang berisi depensi, *library*, *codingan* dan *environment* lainnya akan dideploy kedalam cluster kubernetes akan dimasukkan kedalam dockerfile terlebih dahulu lalu akan di build ke docker images didalam vm instance ubuntu tetapi sebelum membuild docker images semua kode *backend* di push kedalam github beserta file-file pendukung lainnya sehingga untuk memindahkan file kode kedalam vm instance hanya perlu menclone github berisi code tersebut tanpa perlu membuat ulang pada vm ubuntu dan dengan github kita dapat melihat setiap ada perubahan *version* pada kode.

```
GNU nano 4.8 Docker#1
# Build Docker Image
FROM node:18.12
# setting working directory
WORKDIR /node/app/
# copy package.json and package-lock.json to the container
COPY package*.json /node/app/
# Installing the dependencies into the container
RUN npm install
# copy code to application
COPY . .
# container exposed network port number
EXPOSE 3000
# command to run within the container
CMD ["npm", "start"]
```

GAMBAR 3.7
Dockerfile

Pada gambar 3.7 merupakan dockerfile untuk backend aplikasi dan flash python yang telah dibuat rekan team saya alpiansyah. Dockerfile dan code backend API telah berada di vm selanjutnya akan membuat docker image tersebut agar mempermudah dalam mengolahnya pada cluster pada kubernetes. Dimana docker image ini berjalan pada *container* berisi depensi dari dockerfile tersebut dan untuk orkestrasi menggunakan kubernetes.

```
ubuntu@ubuntu-dev-1:~$ docker build -t hanifahart/app-nodejs:14.15.8
Sending build context to Docker daemon 227.848
Step 1/7 : FROM node:14.15
---> 000000000000
Step 2/7 : WORKDIR /app
---> 000000000000
Step 3/7 : COPY package*.json /app/
---> 000000000000
Step 4/7 : RUN npm install
---> 000000000000
Step 5/7 : CMD ["npm", "start"]
---> 000000000000
```

GAMBAR 3.8

Proses pembuatan docker image

Pada gambar 3.8 membuat docker image dengan Langkah-langka sesuai dari isi dari konfigurasi dockefilanya. Setelah di *build* imagenya push docker image ke repository dockerhub sesuai dengan docker image yang telah dibuat sebelumnya.

```
ubuntu@ubuntu-dev-1:~$ sudo docker push hanifahart/app-nodejs:14.15.8
The push refers to repository [docker.io/hanifahart/app-nodejs]
Pushed
14.15.8: Pushed
hanifahart/app-nodejs:14.15.8: Pushed
```

GAMBAR 3.9

Push docker image ke Dockerhub

Aplikasi yang berada di docker container akan diimplementasikan ke Kubernetes dimana Kubernetes membantu dalam mengelola dan mengorkestrasi banyak container dalam skala besar.

Untuk *tools* pendukung dalam pengimplementasian ini dengan menginstall Kubespray agar dapat mengotomatiskan proses instalasi Kubernetes secara penuh. Ini mempermudah pengelolaan infrastruktur Kubernetes, termasuk instalasi dan konfigurasi cluster, sehingga meminimalkan upaya dan waktu yang diperlukan dalam pengelolaan *backend*.

```
ubuntu@ubuntu-dev-1:~$ wget https://github.com/kubernetes-sigs/kubespray/archive/refs/tags/v2.11.0.tar.gz
2023-03-23 14:03:04 - https://github.com/kubernetes-sigs/kubespray/archive/refs/tags/v2.11.0.tar.gz
Resolving github.com (github.com)... 20.285.243.165
Connecting to github.com (github.com):20.285.243.165... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/kubernetes-sigs/kubespray/tar.gz/refs/tags/v2.11.0 [following]
2023-03-23 14:03:04 - https://codeload.github.com/kubernetes-sigs/kubespray/tar.gz/refs/tags/v2.11.0
Resolving codeload.github.com (codeload.github.com)... 20.285.243.165
Connecting to codeload.github.com (codeload.github.com):20.285.243.165... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: 'v2.11.0.tar.gz'
v2.11.0.tar.gz [
  <-->
] 2.45M 2.15MB/s in 1.1s
2023-03-23 14:03:04 (2.15 MB/s) - 'v2.11.0.tar.gz' saved [2566996]
```

GAMBAR 3.10

Install Kubespray

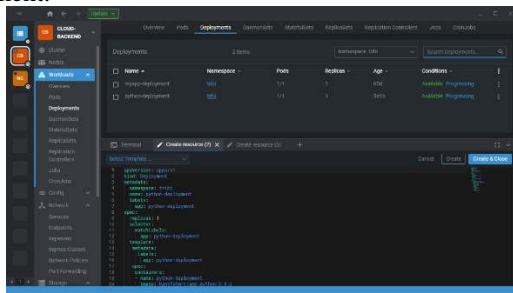
Setelah diinstall dan file tersebut di *extract* atau *unzip* dan dapat dilihat *tools provisioning*. Lalu kode konfigurasi cluster dicopy ke file configfix.yaml tersebut ke openlens agar mempermudah *monitoring*, *visualisasi* pada saat deployment ke cluster Kubernetes.

```
configfix.yaml
# Kubernetes Cluster Configuration
# This file is used to configure the Kubernetes cluster
# It contains the configuration for the Kubernetes cluster
# The configuration is used to create the Kubernetes cluster
# The configuration is used to create the Kubernetes cluster
```

GAMBAR 3.11

configfix.yaml

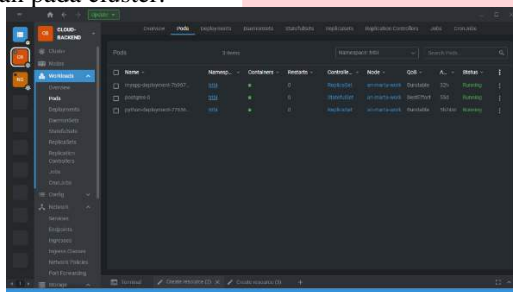
Selanjutnya dengan memasukkan code .yaml ke Create resource setelah itu create maka file tersebut akan ke deploy dapat dilihat ada 2 file yaitu myapp-deployment dan python-deployment.



GAMBAR 3. 12

Mendeploy file myapp-deployment.yaml ke cluster

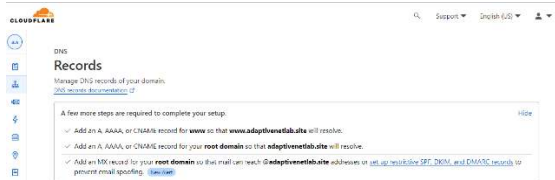
Dan *deployment* telah berhasil tidak terdapat error jika sudah *available* dan *progressing* dan file *deployment* ini akan tersimpan pada cluster.



GAMBAR 3. 13

pod cluster kubernetes

Pods ini secara otomatis terbuat saat *deployment* digunakan untuk melihat apakah file yang *deployment* tersebut terdapat ada error atau tidak. Lalu diimplementasikan ke loadbalancer agar mendapatkan *external ip* sehingga backend bisa dapat diakses tidak hanya didalam cluster kubernetes tetapi bisa diakses diluar cluster kubernetes tetapi masih dalam lingkup server on premises belum bisa diakses secara public melalui internet tetapi *external ip* ini yang akan dihubungkan dengan *ip public*. Maka dibutuhkan reverse proxy bertujuan tidak hanya membuat *external ip* dari server on premises ke *public* agar android dapat mengakses backend aplikasi secara public dengan konfigurasi nginx pada server on premises ke cloud vps agar bisa saling terhubung dimana dari *ip local* menjadi *ip public*. Cloudflare digunakan sebagai DNS management dimana dari *external ip* yang didapat dari loadbalancer di server on premises di hubungkan ke *ip public* agar bisa di akses melalui internet backend tersebut lalu di *toward* ke domain URL.



GAMBAR 3. 14

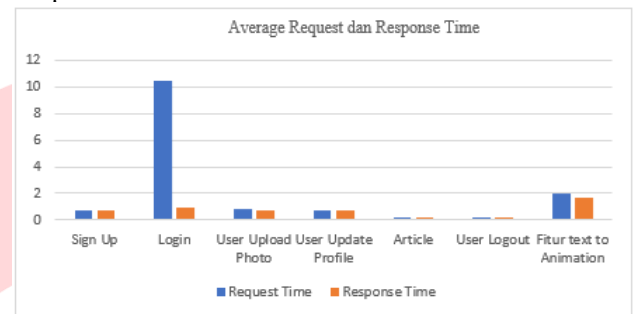
CloudFlare DNS Record

sub domain yang digunakan untuk Endpoint API atau backend dan minio sehingga android bisa akses API dan *storage* yang telah dibuat.

IV. HASIL DAN PEMBAHASAN

Hasil pengujian sistem yang digunakan pada server untuk backend aplikasi dengan mencoba mengakses endpoint API yang diproduksi oleh android yang telah di pointing ke domain url seperti *sign up*, *login*, *user upload photo*, *user update profile*, *article*, *user logout*, serta fitur text to animation. Percobaan akan diulang sebanyak 10 kali dengan sample data yang berbeda setiap kali endpoint API. Untuk mengetahui waktu lama *request* dan *response* time serta CPU dan memori *usage*.

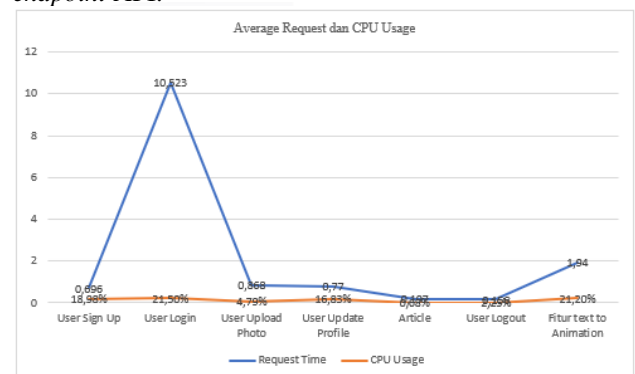
A. Hasil nilai Rata-rata lama waktu *request* dan *response* endpoint API.



GAMBAR 4. 1

Grafik Average Request dan Response Time

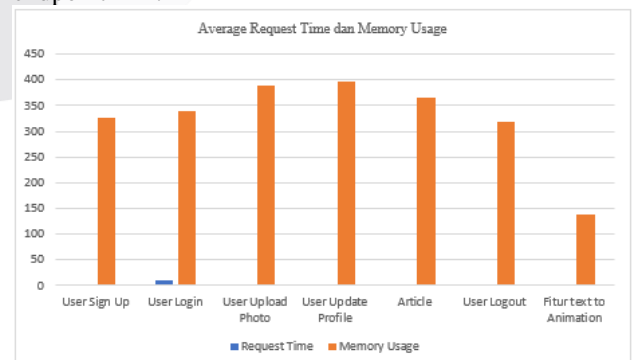
B. Hasil nilai rata-rata saat request time maka CPU *usage* endpoint API.



GAMBAR 4. 2

Average Request dan CPU Usage

C. Hasil nilai rata-rata saat request time maka memori *usage* endpoint API.



GAMBAR 4. 3

Average Request dan Memory Usage

V. KESIMPULAN

Dapat disimpulkan jika penggunaan cpu dan memori pada server akan mengalami peningkatan saat ada *user* mengakses aplikasi atau server menerima *request* dari *user* sesuai dengan kondisi yang diperintahkan oleh *user*. Jika kondisi *request* mengirim data ke server maka penggunaan cpu lebih besar dari pada memori sehingga waktu *request* nya lebih lama. Hal ini menandakan bahwa server memerlukan lebih banyak waktu untuk mengeksekusi kode dan logika yang terkait dengan *request* tersebut, dan beban kerja yang tinggi pada CPU menyebabkan peningkatan waktu proses.

REFERENSI

- [1] B. Kabupaten, "Banyaknya Desa Menurut Keberadaan Penyandang Cacat di Kabupaten/Kota Provinsi Jawa Barat (Bukan Tepi Laut)," Badan Pusat Statistik Kabupaten Bekasi, 2021. [Online]. Tersedia: <https://bekasikab.bps.go.id/statictable/2021/06/18/1717/banyaknya-desa-menurut-keberadaan-penyandang-cacat-di-kabupaten-kota-provinsi-jawa-barat-bukan-tepi-laut-2011.html>. [Diakses: 07-Nov-2022].
- [2] J. K. Hikmalansya, "Aplikasi Pembelajaran Bahasa Isyarat berbasis android," Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi, vol. 1, no. 2, 2016.
- [3] K. Perempuan, "Siaran pers," Komnas Perempuan Komisi Nasional Anti Kekerasan Terhadap Perempuan, 23-Sep-2021. [Online]. Tersedia: [https://komnasperempuan.go.id/siaran-pers-detail/siaran-pers-komnas-perempuan-tentang-peringatan-hari-bahasa-isyarat-internasional-jakarta-23-september-2021#:~:text=Penerapan%20SIBI%20ini%20diresmikan%20dalam,Sekolah%20Luar%20Biasa%20\(SLB\)](https://komnasperempuan.go.id/siaran-pers-detail/siaran-pers-komnas-perempuan-tentang-peringatan-hari-bahasa-isyarat-internasional-jakarta-23-september-2021#:~:text=Penerapan%20SIBI%20ini%20diresmikan%20dalam,Sekolah%20Luar%20Biasa%20(SLB).). [Diakses: 08-Nov-2022].
- [4] S. GlobalStats, "Mobile Operating System Market Share Indonesia," StatCounter Global Stats, 2022. [Online]. Tersedia: <https://gs.statcounter.com/os-market-share/mobile/indonesia>. [Diakses: 08-Nov-2022].
- [5] Shah, J. and Dubaria, D. (2019) 'Building modern clouds: Using Docker, Kubernetes & Google Cloud Platform', 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)[Preprint].doi:10.1109/ccwc.2019.8666479.