

Penerjemah Huruf Vokal Pada Bahasa Isyarat Indonesia (Bisindo) Menjadi Audio Berbasis *Image Processing*

Translator Of Vowels At Bahasa Isyarat Indonesia (Bisindo) Into Image Processing-Based Audio

1st Mohammad Zhillan Al Rashif
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
zhillanra@student.telkomunivers
ity.ac.id

2nd Suci Aulia
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
suciaulia@telkomuniversity.ac.id

3rd Sugondo Hadiyoso
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia
sugondo@telkomuniversity.ac.id

Abstrak—Di Indonesia sendiri terdapat bahasa isyarat yang sering digunakan dan sudah diakui serta disepakati agar dapat digunakan untuk saling bertukar informasi, yaitu Bahasa Isyarat Indonesia (BISINDO). Bahasa Isyarat Indonesia (BISINDO) merupakan bahasa isyarat yang menjadi bahasa sehari-hari bagi sebagian orang penyandang disabilitas khususnya seorang tunarungu. Proyek akhir ini dibuat, dengan tujuan agar lebih banyak orang yang mengetahui huruf vokal BISINDO. Pada Proyek Akhir ini, telah dirancang sebuah sistem penerjemah huruf vokal BISINDO per karakter menggunakan metode *Convolutional Neural Network* dengan arsitektur VGG-19 berbasis *image processing* dengan menggunakan bahasa pemrograman python. Data yang diambil merupakan gambar gerakan tangan huruf vokal dalam bahasa isyarat yang dibagi menjadi tiga yaitu 100 citra BISINDO, 150 citra BISINDO new, dan 250 citra BISINDO mix yang digunakan sebagai pengenalan untuk pendeteksian sistem penerjemah ini, yang selanjutnya akan diproses untuk menjalankan program penerjemah bahasa isyarat per karakter ini menjadi sebuah audio. Berdasarkan hasil pengujian dari 2 skenario, yaitu 100 citra untuk pengujian dan pelatihan, dan 150 citra untuk validasi data, dari hasil pengujian dan pelatihan diperoleh tingkat akurasi tertinggi sebesar 100% dengan menggunakan nilai *epoch* sebesar 14 dan *batch size* sebesar 4.

Kata kunci : *Convolutional Neural Network*, BISINDO, VGG-19.

Abstract—In Indonesia itself there is a sign language that is often used and has been recognized and agreed upon to be exchange information, namely Indonesian Sign Language (BISINDO). Indonesian (BISINDO) is the language used for everyday language for some

people with disabilities, especially a deaf person. This final project is made, with the aim that more people know BISINDO vowels. In this final project, a system for translating BISINDO vowels per character has been designed using the *Convolutional Neural Network* method with the VGG-19 architecture based on image processing using the python programming language. The data taken are pictures of hand gestures of vowels in sign language which are divided into three, namely 100 BISINDO images, 150 new BISINDO images, and 250 BISINDO mix images which are used as an introduction to detect this translator system, which will then be processed to run the language translator program. this per-character cue becomes an audio. Based on the test results from 2 scenarios, namely 100 images for testing and training, and 150 images for data validation from the results of testing and training, the highest accuracy level is 100% using an epoch value of 14 and a batch size of 4.

Keyword : *Convolutional Neural Network*, BISINDO, VGG-19.

I. PENDAHULUAN

Di Negara Kesatuan Republik Indonesia (NKRI) banyak sekali istilah-istilah yang digunakan untuk merujuk atau menyebut orang penyandang disabilitas, salah satunya yaitu penggunaan diksi “Tuna” yang berasal dari bahasa Jawa kuno yang berarti rusak atau rugi, dan pernah dipakai dalam dokumen resmi Negara yaitu Undang-undang no. 6 tahun 1974. Sebagai contoh penggunaan istilah “Tuna” terkait dengan orang penyandang disabilitas adalah : Tunarungu (Tidak dapat mendengar), Tunawicara (Tidak dapat berbicara), dan Tunanetra (Tidak dapat melihat). Keterbatasan yang paling menonjol yang dialami

oleh seorang tunarungu, tunanetra dan juga tunawicara ialah dalam hal berkomunikasi.

Sampai saat ini, cara yang paling sering digunakan untuk berkomunikasi khususnya bagi seorang tunawicara ialah dengan bahasa isyarat, di Indonesia sendiri terdapat 2 jenis bahasa isyarat. Pertama, Sistem Bahasa Isyarat Indonesia atau SIBI. Kedua, Bahasa Isyarat Indonesia atau BISINDO [1]. Kedua bahasa isyarat tersebut sudah lumrah digunakan, dengan metode yang disebut *hand on hand* atau gestur tangan ke tangan untuk saling bertukar informasi ketika berkomunikasi.

Dengan adanya suatu permasalahan tersebut, untuk meminimalisir kesalahpahaman mengartikan suatu kata atau kalimat, telah dirancang sistem penerjemah huruf vokal bahasa isyarat per karakter berbasis *Image Processing* dengan menggunakan bahasa pemrograman python dan juga menggunakan metoda CNN berarsitektur VGG-19. Dataset yang digunakan berupa gambar atau citra gerakan tangan yang digunakan sebagai acuan objek pengenalan untuk pendeteksian sistem penerjemah ini, yang selanjutnya akan diproses untuk menjalankan program penerjemah bahasa isyarat per karakter ke audio.

II. Dasar Teori

A. Bahasa Isyarat Indonesia (BISINDO)

BISINDO merupakan bahasa isyarat alami budaya asli Indonesia yang dengan mudah dapat digunakan dalam pergaulan isyarat Tuli sehari-hari. BISINDO merupakan bahasa isyarat yang dipelajari secara alami oleh Tuli sehingga BISINDO seperti halnya bahasa daerah dan memiliki keunikan di tiap daerah. Kecepatan dan kepraktisannya membuat Tuli lebih mudah memahami meski tidak mengikuti aturan bahasa Indonesia sebagaimana yang digunakan SIBI [1]. Bentuk gestur tangan pada BISINDO dapat dilihat pada gambar 1.

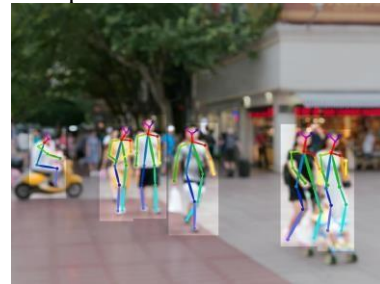


Gamabar 2.1 Gestur Tangan Bahasa Isyarat Indonesia (BISINDO)

B. Pengolahan Citra (*Image Processing*)

Citra yang baik adalah citra yang memiliki kualitas tinggi dan sesuai dengan gambar aslinya serta memiliki informasi yang lengkap dan jelas sesuai dengan apa yang kita inginkan. Namun seringkali citra mengalami penurunan kualitas citra misalnya, terdapat cacat pada citra (derau), terlalu kontras, kurang tajam warnanya, terlalu lembut dan

lain sebagainya. Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan computer, menjadi citra yang kualitasnya lebih baik [2]. Salah satu penggunaan *Image Processing* dapat dilihat pada Gambar 2. 2 berikut.



Gambar 2.2 Salah Satu Penggunaan *Image Processing*

C. Google Colab

Colaboratory, atau “Colab” merupakan produk dari *Google Research*. Colab memungkinkan siapa saja menulis dan mengeksekusi kode python arbitrer melalui browser, dan sangat cocok untuk *machine learning*, analisis data, serta Pendidikan. Secara lebih teknis, Colab merupakan layanan *notebook Jupyter* yang dihosting dan dapat digunakan tanpa penyiapan, serta menyediakan akses gratis ke *resource* komputasi termasuk GPU. *Resource* Colab tidak dijamin dan sidatnya terbatas, serta batas penggunaannya terkadang berfluktuasi. Hal ini diperlukan agar Colab dapat menyediakan *resource* secara gratis. Pengguna yang ingin memiliki akses lebih andal ke *resource* yang lebih baik dapat menggunakan Colab Pro. Memperkenalkan Colab Pro merupakan langkah pertama yang Google ambil untuk melayani pengguna yang ingin melakukan lebih banyak hal di Colab. Tujuan jangka Panjang pihak Google adalah untuk terus menyediakan versi gratis Colab, dan di saat yang bersamaan berkembang secara berkelanjutan untuk memenuhi kebutuhan pengguna Google [3]. Logo Google Colab dapat dilihat pada Gambar 2. 3 berikut.



Gambar 2.3 Logo Google Colab

D. *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah pengembangan dari Multilayer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis Deep Neural Network karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra [4] atau bisa dikatakan juga *Convolutional Neural Network* (CNN) merupakan salah satu jenis algoritma Deep Learning yang dapat menerima *input* berupa

gambar, menentukan aspek atau obyek apa saja dalam sebuah gambar yang bisa digunakan mesin untuk “belajar” mengenali gambar, dan membedakan antara satu gambar dengan yang lainnya. Pada bidang *image processing* terdapat beberapa algoritma yang dapat digunakan, metode *Convolutional Neural Network* (CNN) memiliki hasil yang paling signifikan dalam pengenalan citra digital [unesa] dan juga merupakan metode yang dapat mempelajari sendiri fitur pada citra yang complex [5].

E. Bahasa Pemrograman Python

Bahasa pemrograman Python merupakan bahasa pemrograman populer yang memiliki keunggulan sebagai berikut :

1. Mudah untuk digunakan dalam mengembangkan sebuah produk perangkat lunak, perangkat keras, *Internet of Things*, aplikasi web, maupun video game.
2. Selain memiliki keterbacaan kode yang tinggi, sehingga kode mudah dipahami, bahasa pemrograman ini memiliki *library* yang sangat banyak dan luas.
3. Merupakan bahasa yang mendukung ekosistem *Internet of Things* dengan sangat baik.

Internet of Things merupakan sebuah teknologi yang menghubungkan benda-benda di sekitar kita ke dalam sebuah jaringan internet yang menghubungkan satu sama lain. Teknologi yang mengusung semua benda dapat terhubung dalam satu jaringan internet ini tidak terlepas dari kebutuhan akan bahasa pemrograman dalam mengembangkan sistemnya dan bahasa pemrograman Python menawarkan dukungan yang sangat baik terhadap teknologi ini. Bahasa ini menjadi sangat populer, karena banyak sistem berbasis *Internet of Things* menggunakan bahasa ini. Terdapat berbagai macam *board* yang menggunakan bahasa pemrograman ini sebagai basisnya untuk menjalankan sistem *Internet of Things*, termasuk di dalamnya adalah Raspberry Pi [6]. Logo bahasa pemrograman python dapat dilihat pada Gambar 2. 4 berikut.



Gambar 2.4 Logo Bahasa Pemrograman Python

F. TensorFlow

TensorFlow merupakan *interface* untuk mengeksekusi perintah dengan menggunakan informasi yang dimiliki mengenai objek yang dikenali serta dapat membedakan antara objek

satu dengan objek lainnya [7]. TensorFlow menyediakan banyak model yang dapat digunakan untuk klasifikasi citra diantaranya adalah *Inception-v3* dan *MobileNets*, serta mendukung pemodelan seperti CNN, RNN, RBM, dan DBN [8]. Logo TensorFlow dapat dilihat pada Gambar 2. 5 berikut.



Gambar 2.6 Logo TensorFlow

G. Keras

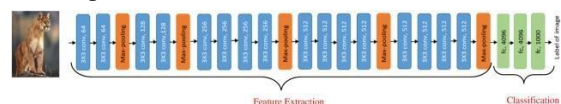
Keras merupakan *framework* yang dibuat untuk mempermudah pembelajaran terhadap komputer. Keras juga merupakan *library* CNN yang ditulis dengan Bahasa python dan mampu berjalan di atas TensorFlow, CNTK, atau Theano. *Library* ini menyediakan fitur yang digunakan dengan fokus mempermudah pengembangan lebih dalam tentang *Deep Learning* [9]. Logo dari *library* keras ini dapat dilihat pada Gambar 2. 6.



Gambar 2.7 Logo Keras

H. Arsitektur VGG-19

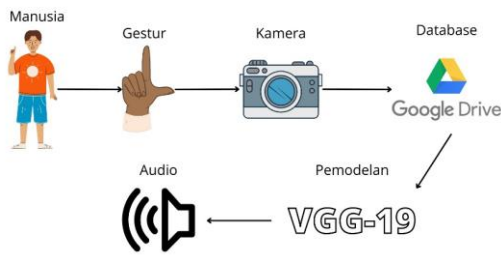
VGG (*Visual Geometry Group*) merupakan salah satu model arsitektur yang digunakan dalam metode pengolahan citra secara *Convolutional Neural Network* atau CNN. Arsitektur VGG19 merupakan arsitektur yang menggunakan konsep semakin banyak layer akurasi semakin baik [10]. Arsitektur VGG ini akan melakukan pemodelan dengan cara memberikan penambahan lapisan pada citra untuk dilakukan pengetesan dan pengujian dan mengurangi tingkat kesalahan dalam pengolahan citra pada CNN. VGG-19 memiliki 19 lapisan atau *layer* dengan pembagiannya adalah 16 *convolution layers* dan 3 *fully connected layers* yang akan mengklasifikasi sebuah citra menjadi 1000 kategori objek. VGG-19 dilatih dengan menggunakan *database* dari ImageNet yang memiliki 1000 kategori objek tersebut [11]. Ukuran citra standar atau *default* dari VGG-19 ini adalah 224x224 [12]. Arsitektur dari VGG-19 ini dapat dilihat pada Gambar 2. 7 berikut.



Gambar 2.8 Arsitektur VGG-19 [12]

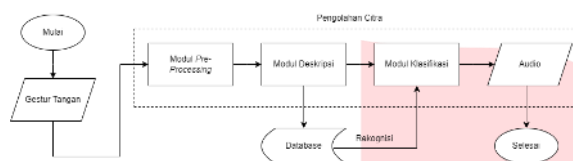
III. PEMBAHASAN

A. Blok Diagram Sistem Penerjemah



Gambar 3.1 Blok Diagram Sistem Penerjemah BISINDO menjadi audio

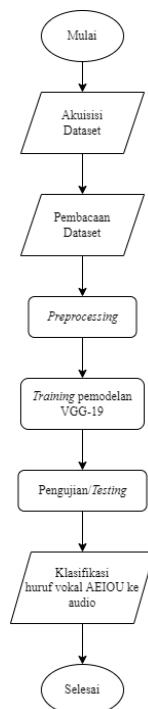
B. Diagram Alir Sistem Penerjemah



Gambar 3.2 Diagram Alir Sistem Penerjemah BISINDO menjadi audio

C. Diagram Alir Pendeteksian Citra

Pada proses penerjemahan dari bahasa isyarat ini akan melalui tahap pendeteksian dari citra gestur yang diambil menggunakan tangan dari penulis beserta partisipan lainnya. Sehingga diperlukan sistem dari pendeteksian ini. Diagram Alir dari pendeteksian citra ini ditunjukkan pada gambar 3.3 .



Gambar 3.3 Diagram Alir Sistem Pendeteksian Citra sampai Penerjemahan

Proses dari pendeteksian ini adalah sebagai berikut.

1. Akuisisi Dataset

Dataset baru yang berisi kumpulan citra gestur tangan huruf vokal dalam BISINDO ini langsung diambil oleh penulis tentunya dibantu oleh partisipan dengan melakukan pengambilan citra gerakan tangan huruf vokal BISINDO menggunakan kamera DSLR Canon EOS1300D 55mm dengan citra yang berformat *Joint Photographic Experts Group* (JPEG atau JPG) beresolusi 3.456 x 2.304. Dataset yang diambil berjumlah 250 citra yang terdiri dari 100 citra dataset awal, dan 150 citra dataset baru untuk validasi. Keseluruhan dataset ini terdiri dari huruf vokal AEIOU dengan dataset pertama masing-masing huruf berjumlah 20 citra dengan 4 kondisi dan pada dataset yang kedua masing-masing huruf berjumlah 30 citra dengan 4 kondisi sama dengan dataset pertama. Citra yang telah diambil untuk dijadikan dataset ini akan diupload ke Google Drive sesuai dengan kelas yang digunakan.

2. Pembacaan Dataset

Pada tahap ini akan dilakukan pembacaan dataset berupa gestur tangan huruf vokal BISINDO yang akan diterjemahkan ke audio.

3. Preprocessing

Pada tahap ini dilakukan pelabelan untuk huruf vokal BISINDO pada platform Google Colab, yang dijadikan sebagai acuan pembacaan dataset. Kemudian, pada dataset tersebut dilakukan pembagian untuk *training* dan *testing*. Pembagian ini ditetapkan jumlah citra untuk *training* sebesar 80% dari keseluruhan dataset, sedangkan untuk *testing* sebesar 20% dari keseluruhan dataset tersebut. Setelah dilakukan pembagian tersebut, maka dilakukanlah perubahan ukuran atau *resize* citra menjadi 64 x 64, dikarenakan resolusi tersebut yang terbaik dan dapat dilakukan pemodelan arsitektur VGG-19 untuk dilakukan pelatihan dan pengujian.

4. Pelatihan Pemodelan Arsitektur VGG-19

Setelah dilakukan tahap *resize* dan penentuan jumlah citra untuk *training* dan *testing* pada tahap *preprocessing*, selanjutnya dilakukan pemodelan arsitektur VGG-19 pada citra. Pemodelan ini dilakukan dengan cara penentuan model VGG pada program, penentuan jumlah kelas yang akan diberikan pemodelan sesuai dengan jumlah yang ada pada database yakni huruf vokal yaitu 5 kelas, lalu melakukan percobaan *epochs* dan *batch* untuk mendapatkan akurasi tertinggi atau maksimal yang selanjutnya akan dilakukan pemodelan dan disimpan dengan file .h5 di Google Drive. Nantinya berkas pelatihan pemodelan arsitektur VGG-19 yang telah disimpan di Google Drive berkecenderungan .h5 ini akan digunakan sebagai

database atau pusat klasifikasi pada sistem penerjemah ini, karena tanpa adanya berkas penyimpanan berekstensi .h5 ini sistem tidak dapat mengetahui dan tidak dapat mengenal bahasa isyarat yang dimasukkan pada tahap pengujian. Setelah disimpan di Google Drive, maka berkas penyimpanan ini akan dipanggil atau dideklarasikan pada tahap pengujian dan pengklasifikasian dengan program yang diberikan.

5. Pengujian

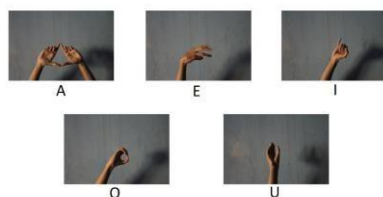
Pada tahap ini, dilakukan pengujian atau *testing* pada dataset berjumlah 20% dari keseluruhan dataset. Pengujian ini dilakukan dengan menggunakan file penyimpanan VGG-19 yang diberikan pada 20% dataset untuk diuji apakah penerjemahan berjalan baik atau tidak, dan seluruh kelas huruf vokal terbaca baik oleh sistem. Apabila terbaca semua, dan tidak ada *error* atau kesalahan, itu berarti hasil pemodelan arsitektur VGG-19 yang sudah disimpan tadi dapat dikatakan baik atau berhasil, sehingga dapat berjalan ke proses selanjutnya.

6. Klasifikasi Teks & A-Z

Setelah dilakukan proses *training* dan *testing* maka selanjutnya masuk ke tahap akhir yaitu pengklasifikasian huruf vokal AEIOU beserta audio dari huruf yang diklasifikasikan.

D. Dataset

Pada proyek akhir ini, dataset diambil dengan melibatkan 3 partisipan, yaitu penulis, dan juga 2 orang lain nya. Dataset pertama diambil dengan menggunakan gestur tangan dari orang A dengan mengambil gestur huruf vokal saja, kemudian dataset ini yang dijadikan data training dan testing untuk dataset awal. Lalu, dataset yang baru akan diambil dengan menggunakan gestur tangan dari orang lainnya (B), yang mana dataset ini diambil secara acak dan/atau *random* dengan mengambil gestur huruf vokal saja, yang kemudian dataset ini akan di training dan testing untuk dijadikan validasi apakah sistem penerjemah ini dapat berjalan atau tidak. Jika nilai dataset dapat diterjemahkan setelah melalui proses pelatihan dan pengujian, maka berkas dari data uji ini akan disimpan dan digunakan dalam proses penerjemahan huruf vokal bahasa isyarat per karakter ini. Dataset yang digunakan pada proyek akhir ini dapat dilihat di gambar 3.4.



Gambar 3.4 Citra BISINDO

IV. HASIL DAN ANALISA

Analisa dari hasil yang telah dilakukan dibutuhkan sebagai penilaian atas program yang telah dirancang dan dibuat ini. Simulasi yang dilakukan diantaranya adalah dengan melakukan pengtesan besaran *epoch* dan *batch size*.

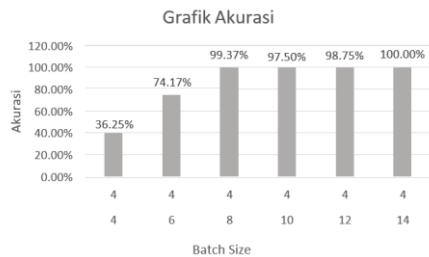
A. Simulasi Epoch pada Pemodelan Arsitektur VGG-19

Pada proses pengujian proyek akhir ini telah disimulasikan pengaruh besaran nilai *epochs* dan *batch size* pada arsitektur VGG-19 terhadap tingkat akurasi pada dataset yang dilakukan pelatihan. Pada proses pelatihan ini, dilakukan proses percobaan sebanyak 6 kali dengan menggunakan jumlah *epochs* yang berbeda disetiap percobaannya, sampai menemukan nilai akurasi yang paling tinggi atau maksimal. Setelah menemukan nilai akurasi yang maksimal, maka file penyimpanan dari pengujian *epochs* dan *batch size* ini akan digunakan untuk proses selanjutnya yaitu proses validasi pada citra AEIOU ke audio. Hasil dari pelatihan ini dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Hasil Percobaan Epoch dan Batch Size

Data Akurasi			
No.	Epochs	Batch Size	Akurasi
1	4	4	36,25%
2	6	4	74,17%
3	8	4	99,37%
4	10	4	97,50%
5	12	4	98,75%
6	14	4	100,00%

Berdasarkan hasil pelatihan epoch dan batch size pada **Tabel 4. 1**, maka dapat dihipunkan ringkasan pelatihan menjadi sebuah grafik yang dapat dilihat pada Gambar 4. 2.



Gambar 4.2 Grafik Percobaan *Epoch* dan *Batch Size*

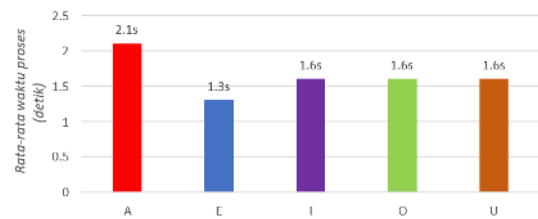
B. Rata-rata Waktu Proses

Pada tahap ini, akan ada tahap *training* atau pelatihan dan tahap *testing* atau pengujian. Setelah dilakukan pelatihan, maka tahap selanjutnya adalah pengujian validasi hasil penyimpanan pemodelan arsitektur VGG-19 pada data citra apakah citra tersebut dapat berjalan dengan baik atau tidak. Pada proses pengujian ini, akan dilakukan penghitungan rata-rata waktu proses yang akan dijadikan sebagai acuan berapa lama sistem ini dapat bekerja dengan menggunakan perangkat yang sudah ditentukan. Tabel rata-rata ini dapat dilihat pada tabel 4.2 di bawah ini.

Tabel 4. 2 Rata-rata Waktu Proses

No.	Kelas	Penguji Ke-...										Waktu Proses Pengujian (detik)
		1	2	3	4	5	6	7	8	9	10	
1.	A	1	2	2	1	2	1	2	1	2	1	2,1
2.	E	1	1	1	1	1	2	1	2	2	1	1,3
3.	I	2	2	2	2	1	2	2	1	1	1	1,6
4.	O	1	2	1	1	2	2	2	1	2	1	1,6
5.	U	2	1	2	1	2	1	2	2	2	1	1,6

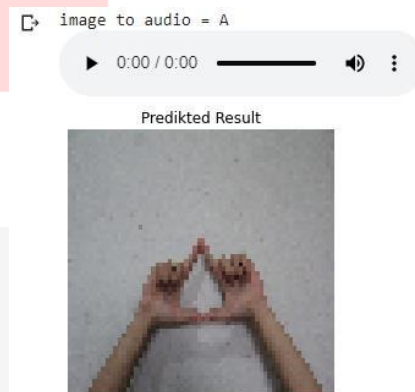
Berdasarkan tabel rata-rata waktu proses ini, dibuat grafik yang menunjukkan rata-rata dari waktu proses ini secara lengkap yang dapat dilihat pada gambar 4.3.



Gambar 4.3 Grafik Rata-rata Waktu Proses

C. Hasil Pengujian

Pada bagian ini, akan diperlihatkan bagaimana hasil dari proses pengujian dari pemodelan arsitektur VGG-19 ini pada kelas yang digunakan, yaitu data citra berupa 5 huruf vokal BISINDO (AEIOU). Berikutnya, akan diperlihatkan hasil dari pengujian yang telah dilakukan, salah satunya pada huruf A yang akan ditampilkan kelas "A" dan juga audionya yang dapat dilihat pada gambar 4.4, beserta hasil pengujian per folder huruf "U" yang dapat dilihat pada gambar 4.5.



Gambar 4.4 Hasil Pengujian pada huruf A

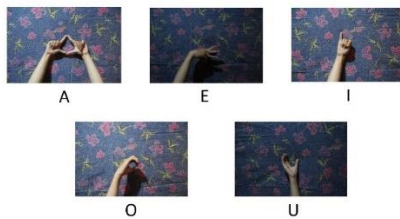


Gambar 4.5 Hasil Pengujian pada kata ADIK

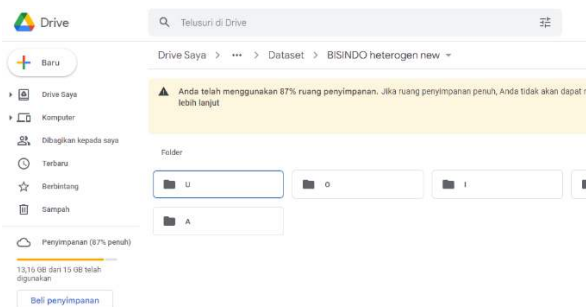
D. Pengujian Menggunakan Dataset Tambahan dengan *Background* Heterogen

Pada bagian ini, dilakukan pengujian dengan menggunakan dataset tambahan, yaitu gestur tangan yang menunjukkan bahasa isyarat Indonesia (BISINDO) dengan *background* bercorak (heterogen) dan juga ketentuan pengambilan citra

dari segi jarak serta tinggi dan juga ketentuannya sama dengan dataset sebelumnya. Berikut Gambar Citra pada dataset tambahan dapat dilihat pada gambar 4.6, serta penyimpanan pada Google Drive yang dapat dilihat pada gambar 4.7.

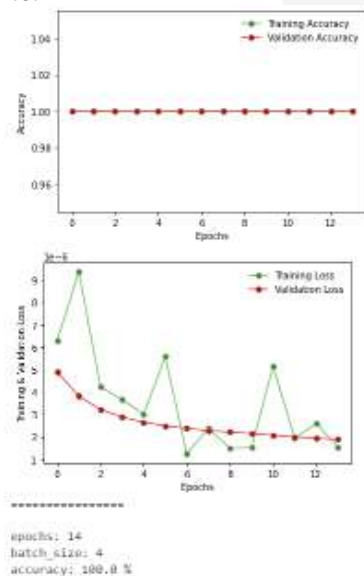


Gambar 4.6 Citra Dataset BISINDO heterogeny new



Gambar 4.7 Penyimpanan Citra Dataset BISINDO heterogeny new di Google Drive

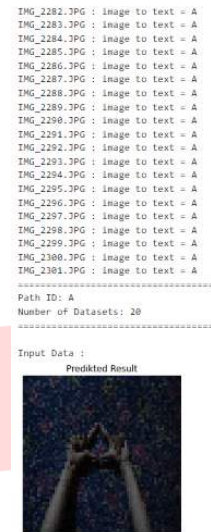
Sama seperti dataset sebelumnya, data yang di uji diambil sebesar 20% dan 80% untuk pelatihan citra, begitu pula dengan nilai *epoch* dan *batch size* yang dipakai yaitu 14 untuk *epoch* dan 4 untuk *batch size* dengan akurasi yang didapatkan yaitu 100%. Dari pemberian nilai *epoch* dan *batch size* maka didapatkanlah grafik yang dapat dilihat pada gambar 4.8.



Gambar 4.8 Grafik *Epoch* dan *Batch Size* dengan akurasi 100%

Setelah itu dilakukanlah pengujian kembali per

folder dan juga per citra yang akan diterjemahkan ke audio menggunakan file berkeestensi .h5 yang telah dilakukan pemodelan dan disimpan di Google Drive dengan nama “VGG19zhillanheterogennew.h5”, dan didapatkanlah hasilnya yaitu 97%. Hasil dari pengujian dataset ini dapat dilihat pada Gambar 4.9 dan 4.10.



Gambar 4.9 Hasil Pengujian Folder Pada Huruf A



Gambar 4.10 Hasil Pengujian Huruf U ke Audio

E. Perbandingan Penggunaan Ukuran Citra 64x64 dan 224x224 VGG-19

```
epoch 1/14 [=====] - 289.75/step - loss: 1.5272e-05 - accuracy: 1.0000 - val_loss: 1.1934e-05 - val_accuracy: 1.0000
epoch 2/14 [=====] - 385.65/step - loss: 9.6903e-06 - accuracy: 1.0000 - val_loss: 1.1765e-05 - val_accuracy: 1.0000
epoch 3/14 [=====] - 176.55/step - loss: 1.9199e-05 - accuracy: 1.0000 - val_loss: 1.0944e-05 - val_accuracy: 1.0000
epoch 4/14 [=====] - 176.55/step - loss: 9.8323e-06 - accuracy: 1.0000 - val_loss: 9.8323e-06 - val_accuracy: 1.0000
epoch 5/14 [=====] - 176.55/step - loss: 4.0863e-06 - accuracy: 1.0000 - val_loss: 9.9521e-06 - val_accuracy: 1.0000
epoch 6/14 [=====] - 176.55/step - loss: 6.1555e-06 - accuracy: 1.0000 - val_loss: 9.2130e-06 - val_accuracy: 1.0000
epoch 7/14 [=====] - 176.55/step - loss: 5.0572e-06 - accuracy: 1.0000 - val_loss: 7.6489e-06 - val_accuracy: 1.0000
epoch 8/14 [=====] - 176.55/step - loss: 1.5872e-05 - accuracy: 1.0000 - val_loss: 7.1164e-06 - val_accuracy: 1.0000
epoch 9/14 [=====] - 176.55/step - loss: 2.4163e-06 - accuracy: 1.0000 - val_loss: 6.7544e-06 - val_accuracy: 1.0000
epoch 10/14 [=====] - 176.55/step - loss: 4.7968e-06 - accuracy: 1.0000 - val_loss: 6.2284e-06 - val_accuracy: 1.0000
epoch 11/14 [=====] - 176.55/step - loss: 5.9300e-06 - accuracy: 1.0000 - val_loss: 5.8233e-06 - val_accuracy: 1.0000
epoch 12/14 [=====] - 176.55/step - loss: 4.1252e-06 - accuracy: 1.0000 - val_loss: 5.4055e-06 - val_accuracy: 1.0000
epoch 13/14 [=====] - 176.55/step - loss: 3.2082e-06 - accuracy: 1.0000 - val_loss: 5.1377e-06 - val_accuracy: 1.0000
epoch 14/14 [=====] - 176.55/step - loss: 7.2835e-06 - accuracy: 1.0000 - val_loss: 4.7444e-06 - val_accuracy: 1.0000
```

4.11 Proses Pemberian Nilai Epoch dan Batch Size Beserta Waktu Proses Pada Ukuran Citra 64x64

```

Epoch 1/14
3/3 [=====] - 2015 62s/step - loss: 9.9202e-06 - accuracy: 1.0000 - val_loss: 4.1689e-06 - val_accuracy: 1.0000
Epoch 2/14
3/3 [=====] - 2015 63s/step - loss: 1.1342e-05 - accuracy: 1.0000 - val_loss: 3.9279e-06 - val_accuracy: 1.0000
Epoch 3/14
3/3 [=====] - 2015 63s/step - loss: 5.2451e-06 - accuracy: 1.0000 - val_loss: 3.7134e-06 - val_accuracy: 1.0000
Epoch 4/14
3/3 [=====] - 2015 63s/step - loss: 1.9712e-05 - accuracy: 1.0000 - val_loss: 3.5107e-06 - val_accuracy: 1.0000
Epoch 5/14
3/3 [=====] - 2015 63s/step - loss: 8.2854e-06 - accuracy: 1.0000 - val_loss: 3.3378e-06 - val_accuracy: 1.0000
Epoch 6/14
3/3 [=====] - 1915 62s/step - loss: 1.4585e-05 - accuracy: 1.0000 - val_loss: 3.1531e-06 - val_accuracy: 1.0000
Epoch 7/14
3/3 [=====] - 1915 62s/step - loss: 1.8952e-05 - accuracy: 1.0000 - val_loss: 3.0180e-06 - val_accuracy: 1.0000
Epoch 8/14
3/3 [=====] - 1915 62s/step - loss: 2.5212e-05 - accuracy: 1.0000 - val_loss: 2.8191e-06 - val_accuracy: 1.0000
Epoch 9/14
3/3 [=====] - 1915 62s/step - loss: 3.6349e-06 - accuracy: 1.0000 - val_loss: 2.6464e-06 - val_accuracy: 1.0000
Epoch 10/14
3/3 [=====] - 1915 62s/step - loss: 6.4712e-06 - accuracy: 1.0000 - val_loss: 2.5272e-06 - val_accuracy: 1.0000
Epoch 11/14
3/3 [=====] - 1915 62s/step - loss: 7.3355e-06 - accuracy: 1.0000 - val_loss: 2.4080e-06 - val_accuracy: 1.0000
Epoch 12/14
3/3 [=====] - 1915 62s/step - loss: 2.6568e-06 - accuracy: 1.0000 - val_loss: 2.3188e-06 - val_accuracy: 1.0000
Epoch 13/14
3/3 [=====] - 1915 62s/step - loss: 9.5802e-06 - accuracy: 1.0000 - val_loss: 2.2352e-06 - val_accuracy: 1.0000
Epoch 14/14
3/3 [=====] - 1975 68s/step - loss: 5.9439e-06 - accuracy: 1.0000 - val_loss: 2.1577e-06 - val_accuracy: 1.0000

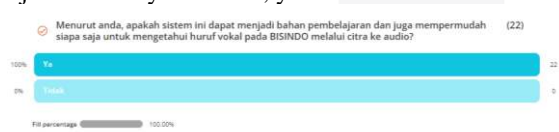
```

4.12 Proses Pemberian Nilai Epoch dan Batch Size Beserta Waktu Proses Pada Ukuran Citra 224x224

Berdasarkan gambar diatas, dijelaskan bahwa semakin tinggi ukuran citra yang diberikan, maka semakin lama waktu prosesnya.

F. Survey Kegunaan Perancangan Sistem Penerjemah Huruf Vokal BISINDO Menjadi Audio

Salah satu metode yang dilakukan pada perancangan ini adalah melakukan survey kepada pengguna yaitu masyarakat umum non-penyandang tunawicara dan tunarungu, sehingga melalui survey ini dapat diketahui penilaian masyarakat terhadap sistem ini, apakah dapat digunakan atau tidak. Jumlah responden yang mengikuti survei ini adalah sebanyak 22 responden, dengan pertanyaan yang diajukan sebanyak 2 butir, yaitu.



4.13 Hasil Survey Pertanyaan 1



4.14 Hasil Survey Pertanyaan 2

V. KESIMPULAN

Pada proyek akhir ini telah dirancang sistem untuk menerjemahkan huruf vokal pada BISINDO per karakter menggunakan metode *Convolutional Neural Network* dengan arsitektur VGG-19. Penerjemahan BISINDO per karakter ini menerjemahkan huruf vokal saja pada BISINDO yaitu AEIOU ke audio.

- Pada proyek akhir perancangan penerjemah BISINDO ini menggunakan total 250 dataset yang terdiri atas 100 data citra untuk huruf vokal AEIOU, dan 150 data citra baru huruf vokal AEIOU yang digunakan untuk validasi.
- Pengambilan citra ini ditentukan berbagai kondisi, yaitu kondisi terang dan gelap, serta background polos dan heterogen. Tinggi dan jarak kamera pun ditentukan,

yaitu 97 cm dan tinggi 110 cm.

- Kamera yang digunakan untuk pengambilan citra ini adalah Canon EOS1300D yang menghasilkan citra beresolusi 3.456 x 2.304 dengan format JPEG atau JPG.
- Data citra yang telah diambil ini akan disimpan di Google Drive sebagai pusat data dari proyek akhir ini yang kemudian dihubungkan dengan Google Colab.
- Pada arsitektur VGG-19, nilai ukuran citra standar adalah sebesar 224x224 tetapi pada proyek akhir ini dirubah atau dilakukan resize menjadi 64x64 agar mempersingkat waktu proses.
- Pada tahap pengujian dari proyek akhir ini, dilakukan 2 skenario dengan pembedanya adalah keperluan dari data citra yang diambil pada scenario pertama dan skenario kedua. Dari kedua skenario ini, dilakukan pembagian jumlah data latih dan data uji sebesar 80% untuk data latih, dan 20% untuk data uji. Pengujian pun dilakukan dengan melakukan pengujian epochs dan batch size pada data citra yang kemudian disimpan sebagai data uji yang akan digunakan pada tahap validasi dari proses pengujian pada data citra di proses penerjemahan.
- Pemberian epochs dan batch size dilakukan pada saat melakukan pemodelan arsitektur VGG-19 dan dilakukan beberapa kali dengan besaran yang berbeda sampai menemukan tingkat akurasi yang tertinggi, yaitu pada percobaan ke-6 epoch sebesar 14 dan batch size sebesar 4 dengan tingkat akurasi sebesar 100%. Hasil dari pengujian ini akan digunakan pada tahap validasi di proses penerjemahan.
- Proses validasi ini menghasilkan bahwa proses pengujian data citra berhasil dengan dapat menerjemahkan data citra menjadi huruf vokal AEIOU dan audio sesuai dengan data citra yang dimasukkan.

REFERENSI

- G. Gumelar, H. Hafiar, and P. Subekti, "KONSTRUKSI MAKNA BISINDO SEBAGAI BUDAYA TULI BAGI ANGGOTA GERKATIN," *INFORMASI*, vol. 48, no. 1, Jul. 2018, doi: 10.21831/informasi.v48i1.17727.
- K. Karnadi, "Pengembangan Aplikasi Digital Image Processing Dengan Microsoft Visual Basic," *J. Digit. Teknol. Inf.*, vol. 1, no. 1, p. 15, 2018, doi: 10.32502/digital.v1i1.933.
- G. Irfon and E. Soen, "Implementasi Cloud

- Computing dengan Google Colaboratory Pada Aplikasi Pengolah Data Zoom Participants,” vol. 6, no. 1, pp. 24–30, 2022.
- [4] S. R. Suartika E. P, I Wayan, Wijaya Arya Yudhi, “Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101,” *J. Tek. ITS*, vol. 5, no. 1, p. 76, 2016, [Online]. Available: <http://repository.its.ac.id/48842/>.
- [5] A. Rohim, Y. A. Sari, and Tibyani, “Convolution neural network (cnn) untuk pengklasifikasian citra makanan tradisional,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 7, pp. 7038–7042, 2019, [Online]. Available: <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5851/2789>.
- [6] T. M. Kadarina and M. H. Ibnu Fajar, “Pengenalan BAHASA PEMROGRAMAN PYTHON MENGGUNAKAN APLIKASI GAMES UNTUK SISWA/I DI WILAYAH KEMBANGAN UTARA,” *J. Abdi Masy.*, vol. 5, no. 1, Jul. 2019, doi: 10.22441/jam.2019.v5.i1.003.
- [7] H. G. GHIFARI, D. DARLIS, and A. HARTAMAN, “Pendeteksi Golongan Darah Manusia Berbasis Tensorflow menggunakan ESP32-CAM,” *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 9, no. 2, p. 359, 2021, doi: 10.26760/elkomika.v9i2.359.
- [8] N. Wiranda, H. S. Purba, and R. A. Sukmawati, “Survei Penggunaan Tensorflow pada Machine Learning untuk Identifikasi Ikan Kawasan Lahan Basah,” *IJEIS (Indonesian J. Electron. Instrum. Syst.)*, vol. 10, no. 2, p. 179, 2020, doi: 10.22146/ijeis.58315.
- [9] A. Santoso and G. Ariyanto, “Implementasi Deep Learning berbasis Keras untuk Pengenalan Wajah,” *Emit. J. Tek. Elektro*, vol. 18, no. 1, pp. 15–21, 2018, doi: 10.23917/emitor.v18i01.6235.
- [10] W. Setiawan, “Perbandingan Arsitektur Convolutional Neural Network Untuk Klasifikasi Fundus,” *J. Simantec*, vol. 7, no. 2, pp. 48–53, 2020, doi: 10.21107/simantec.v7i2.6551.
- [11] D. Hindarto and H. Santoso, “Plat Nomor Kendaraan Dengan Metode Convolutional Neural Network,” *J. Inov. Inform. Univ. Pradita*, no. September 2021, pp. 1–12, 2019.
- [12] M. Bansal, M. Kumar, M. Sachdeva, and A. Mittal, “Transfer learning for image classification using VGG19: Caltech-101 image data set,” *J. Ambient Intell. Humaniz. Comput.*, no. 0123456789, 2021, doi: 10.1007/s12652-021-03488-z.