JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

PEMBANGUNAN APLIKASI PENERJEMAH BAHASA ISYARAT DENGAN METODE CNN BERBASIS ANDROID

Reza Haris Alfikri[1)], Mardi Siswo Utomo[2)], Herny Februariyanti[3)], Eko Nurwahyudi[4)]

[1,2]Program Studi Teknik Informatika, Universitas Stikubank Semarang
[3,4]Program Studi Sistem Informasi, Universitas Stikubank Semarang
[1,2,3,4]Jalan Tri Lomba Juang, Mugassari, Semarang, Kota Semarang, Jawa Tengah

Email: [1]rezaharis32@gmail.com, [2]mardi@edu.unisbank.ac.id, [3]hernyfeb@edu.unisbank.ac.id, [4]eko@edu.unisbank.ac.id

Abstract

Speech impairment is a condition where a person does not have the ability to hear or speak. Because of these problems, individuals with speech impairments use sign language as the language used to communicate with other individuals. Sign language is a language that uses lips, body, and hands to express meaning in communication. Most people in Indonesia do not understand and are reluctant to learn about the use of sign language, so this results in limitations in communicating when meeting with individuals who are speech impaired. This study has the aim that the general public can communicate with individuals with speech impairments by carrying out the sign language translation process and learning about the use of sign language with an Android-based sign language translator application developed by the researcher. The sign language translator application was developed on the Android platform, with the aim that it can be used by the wider community. The researcher developed the Tensorflow Lite model as a sign language translator model, by implementing the Convolutional Neural Network (CNN) method and the total number of sign language datasets was 1820 data. The type of sign language used by researchers in developing the model is American Sign Language (ASL). Researchers used 3 total epochs when conducting the model training process. Among them are 100 Epoch with 95% accuracy result, 150 Epoch with 93% accuracy result, and 200 Epoch with 97% accuracy result. The training model with the highest accuracy results is deployed to the Android application. However, after the sign language translator model was deployed on an Android-based application, the accuracy dropped to 73% with the problem of several sign languages experiencing incorrect predictions. Sign language translation can still be done, although there are some sign languages that have mispredicted. The sign language translator application has been developed by researchers with various supporting features that can be used by users. Some of these features include login, register, forgot password, home, translation, dictionary, profile, edit profile, about me, and the last one is bookmarks. Users can use the translation feature to carry out the sign language translation process, and the dictionary feature to learn to use sign language.

Keywords: *Speech Impaired, Sign Language, Communication, Convolutional Neural Network (CNN), Android.*

Abstrak

Tunawicara merupakan keadaan dimana seseorang tidak memiliki kemampuan untuk mendengar ataupun berbicara. Karena adanya permasalahan tersebut, individu dengan penyandang tunawicara menggunakan bahasa isyarat sebagai bahasa yang digunakan untuk melakukan komunikasi dengan individu lainnya. Bahasa isyarat adalah bahasa yang menggunakan gerak bibir, tubuh, dan juga tangan untuk mengekspresikan maksud dalam komunikasi. Sebagian besar masyarakat di Indonesia tidak memahami dan enggan belajar mengenai penggunaan bahasa isyarat, sehingga hal tersebut mengakibatkan adanya batasan dalam melakukan komunikasi jika bertemu dengan individu penyandang tunawicara. Penelitian ini memiliki tujuan agar masyarakat umum dapat melakukan komunikasi dengan individu penyandang tunawicara dengan melakukan proses penerjemahan bahasa isyarat dan belajar mengenai penggunaan bahasa isyarat dengan aplikasi penerjemah bahasa isyarat berbasis Android yang dikembangkan oleh peneliti. Aplikasi penerjemah bahasa isyarat dikembangkan pada *platform* Android, dengan tujuan agar dapat digunakan oleh masyarakat secara luas. Peneliti mengembangkan model *Tensorflow Lite* sebagai model penerjemah bahasa isyarat, dengan mengimplementasikan metode *Convolutional Neural Network* (CNN) dan jumlah total *datasets* bahasa isyarat sebanyak 1820 data. Jenis bahasa isyarat yang digunakan oleh peneliti pada pengembangan model tersebut adalah berjenis *American Sign Language* (ASL). Peneliti menggunakan 3 total *Epoch* ketika melakukan proses *training* model. Diantaranya adalah 100 Epoch dengan hasil akurasi 95%, 150 Epoch dengan hasil akurasi 93%, dan 200 Epoch dengan hasil akurasi 97%. *Training* model dengan hasil akurasi tertinggi dilakukan *deploy* ke aplikasi Android. Namun, setelah model penerjemah bahasa isyarat dilakukan *deployment* pada aplikasi berbasis Android, akurasi turun hingga 73% dengan permasalahan beberapa beberapa bahasa

JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

isyarat mengalami salah prediksi. Penerjemahan bahasa isyarat masih dapat dilakukan, walaupun terdapat beberapa bahasa isyarat yang mengalami salah prediksi. Aplikasi penerjemah bahasa isyarat telah dikembangkan oleh peneliti dengan berbagai fitur pendukung yang dapat digunakan oleh *users*. Beberapa fitur tersebut diantaranya adalah *login*, *register*, *forgot password*, *home*, *translation*, *dictionary*, *profile*, *edit profile*, *about me*, dan yang terakhir adalah *bookmarks*. *Users* dapat menggunakan fitur *translation* untuk melakukan proses penerjemahan bahasa isyarat, dan fitur *dictionary* untuk belajar menggunakan bahasa isyarat.

Kata Kunci: Tunawicara, Bahasa Isyarat, Komunikasi, *Convolutional Neural Network* (CNN), Android.

1. PENDAHULUAN

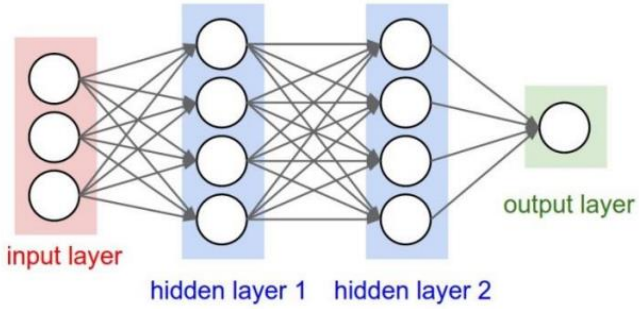
Tunawicara merupakan keadaan individu yang mengalami kesulitan dalam berkomunikasi. Hal ini dapat disebabkan oleh tidak berfungsinya organ-organ bicara seperti rongga mulut, lidah, langit-langit dan pita suara. Selain itu, kurang atau tidak berfungsinya organ pendengaran, keterlambatan dari perkembangan bahasa, kerusakan otot pada sistem saraf dan struktur otot. Karena hal tersebut, penyandang tunawicara menggunakan bahasa isyarat untuk berkomunikasi dengan individu lainnya [1]. Cukup banyak orang yang tidak faham mengenai penggunaan bahasa isyarat, sehingga ketika bertemu dengan orang yang menyandang tunawicara, akan timbul kesulitan dalam melakukan komunikasi.

Berdasarkan tren penggunaan sistem operasi dari tahun 2012 sampai dengan 2021, sistem operasi Android menduduki peringkat teratas untuk versi sistem operasi *mobile*, dilanjutkan dengan IOS dan seterusnya [2]. Ponsel Android sendiri sudah eksis di kalangan masyarakat sejak tahun 2008, hingga sampai saat ini versi sistem operasinya telah berkembang sampai dengan versi ke 12 [3]. Pengembangan aplikasi Android juga lebih fleksibel, jika dibandingkan dengan aplikasi berbasis IOS yang mengharuskan kita untuk menggunakan sistem operasi MAC dalam pengembangannya.

Dengan perkembangan teknologi yang sangat pesat ini, terutama pada bidang *Machine Learning*. Terdapat berbagai *neural networks* yang sering digunakan oleh *developers* untuk mengembangkan aplikasi mereka. Diantaranya adalah *Recurrent Neural Network* (RNN), *Generative Adversial Networks* (GAN), dan *Convolutional Neural Network* (CNN). Setiap *neural networks* tersebut memiliki keunikannya tersendiri. Namun, yang membedakan CNN dengan yang lain adalah untuk CNN pengolahan datasets dilakukan dengan melakukan preprocessing terlebih dahulu. Hasil dari gambar yang diolah dengan melakukan preprocessing, akan memiliki akurasi yang lebih tinggi dibandingkan tanpa melakukan preprocessing.

CNN cukup membantu *developers*, dengan *preprocessing* yang meningkatkan tingkat akurasi. Begitu juga Android yang ramah akan masyarakat, dengan harga terjangkau dan jumlah *users* yang cukup banyak di dunia. Oleh sebab itu, aplikasi penerjemah bahasa isyarat yang dikembangkan diharapkan dapat membantu masyarakat, khususnya yang tidak paham mengenai penggunaan bahasa isyarat agar dapat berkomunikasi dengan individu yang menyandang tunawicara.

Convolutional Neural Network (CNN) merupakan salah satu algoritma *Deep Learning* yang dapat digunakan untuk melakukan *input image*, menetapkan *weight* dan *bias* ke dalam berbagai aspek dan objek di dalam gambar, sehingga dapat digunakan untuk membedakan satu gambar dengan gambar yang lainnya. CNN merupakan variasi dari *Multi Layer Perception* (MLP) yang terinspirasi dari *neural networks* pada manusia [4].



Gambar 1. Arsitektur *Multi Layer Perception* (MLP) [4]

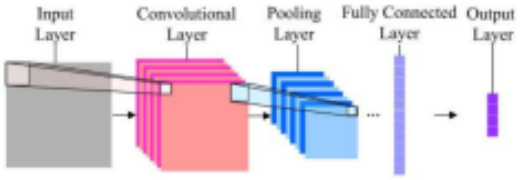
Sebuah arsitektur dari MLP pada Gambar 1 memiliki *input layer*, *hidden layer*, dan juga *output layer* dengan masing-masing *layer* tersebut berisi *ji neuron* (lingkaran putih). MLP menerima input data satu dimensi dan mempropagasikan data tersebut pada jaringan hingga menghasilkan output. Setiap hubungan antar parameter bobot satu dimensi yang menentukan kualitas mode. Di setiap data input pada layer, dilakukan operasi linear dengan nilai weight yang sudah ada,

JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

kemudian hasil komputasi akan ditransformasi menggunakan operasi non linear yang disebut sebagai fungsi aktivasi [5].

Metode pada CNN mempunyai dua tahap, yang pertama adalah klasifikasi gambar yang menggunakan *feedforward* dan tahap yang kedua adalah *learning stage* yang menggunakan *backpropagation method*. *Feedforward process* adalah tahap pertama yang mengenalkan beberapa *layers* untuk melakukan klasifikasi gambar yang menggunakan *updated weights* dan *biases* dari *backpropagation process*. Tahap ini juga akan digunakan kembali selama proses *testing* dilakukan. *Backpropagation process* adalah tahap kedua, yang terdapat hasil dari *feedforward process* dan dapat terlacak dari *output layer* sampai *first layer*. Indikasi dari data yang telah terlacak adalah dengan didapatkannya *weights* dan *bias* dengan nilai yang baru. Sebelum proses klasifikasi, akan dilakukan *preprocessing* terlebih dahulu dengan cara melakukan *wrapping* dan cropping pada datasets, dengan tujuan agar fokus gambar yang ada pada *datasets*, fokus pada objek yang akan dilakukan proses klasifikasi [4]. Berdasarkan pada penelitian "Sistem Pemberian Saran Resep Kuliner Indonesia Menggunakan Metoda *Case Based Reasoning* Dengan Algoritma Similaritas *Czekanowski* Berbobot", nilai bobot (*weights*) suatu bahan masakan akan semakin meningkat lebih tinggi bila bahan masakan tersebut ada dalam pembentuk sebagian besar resep masakan, namun tidak semua resep masakan menggunakannya. Bahan masakan disini dapat dikatakan sebagai *value*, sedangkan untuk pembentuk resep masakan dapat dikatakan sebagai pembentuk dari *value* tersebut [6].

Layers yang ada pada CNN, menggunakan *filter* untuk mengekstraksi objek dari citra input. *Filter* berisi *weight* yang digunakan untuk mendeteksi garis tepi, kurva, dan warna dari objek. Konvolusi akan menghasilkan transformasi linear dari citra input yang sesuai dengan informasi *spasial* pada data citra. *Filter* diaplikasikan secara berulang sehingga menghasilkan serangkaian bidang *receptive* [7]. Berikut adalah gambaran dari beberapa layers yang ada pada CNN.



Gambar 2. *Convolutional Neural Network* (CNN) Layers [7]

Convolutional Layer merupakan lapisan yang digunakan untuk melakukan operasi konvolusi pada *output layer*. *Layer* ini termasuk blok utama pada CNN yang di dalamnya terdiri dari kumpulan *filter* yang dipelajari secara acak untuk melakukan operasi konvolusi, dengan tujuan sebagai ekstraksi fitur untuk mempelajari representasi fitur dari *input layer*. Tujuan dilakukannya operasi konvolusi pada data citra yaitu untuk mengekstraksi fitur dari input citra, Konvolusi tersebut akan menghasilkan transformasi *linear* dari data yang di *input* sesuai informasi *spasial* yang tersedia pada data. *Weight* pada layer akan melakukan pembuatan spesifikasi untuk kernel konvolusi, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN [7].

Pooling Layer adalah lapisan yang berfungsi untuk mengurangi ukuran *spasial* dari fitur konvolusi, dengan tujuan untuk menimalisir sumber daya komputasi yang dibutuhkan untuk memproses data melalui pengurangan dimensi dari *feature map*, sehingga mempercepat proses komputasi karena parameter yang digunakan semakin sedikit [7]. Terdapat dua macam *pooling* yang sering digunakan, diantaranya adalah *average pooling* dan *max pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max pooling* adalah nilai maksimal [8].

Fully Connected Layer merupakan lapisan yang digunakan untuk melakukan transformasi pada dimensi data, agar data dapat diklasifikasikan secara *linear* [7]. *Layer* ini mendapatkan *input* dari proses sebelumnya, untuk menentukan fitur mana yang paling berkolerasi dengan kelas tertentu. Fungsi dari *layer* ini adalah untuk menyatukan semua *node* menjadi satu dimensi [8].

Activation Function atau sering juga disebut dengan *neuron* merupakan fungsi *non-linear* yang memungkinkan sebuah jaringan syaraf tiruan untuk menyelesaikan permasalahan *non-trivial*. *Activation function* pada arsitektur CNN terletak pada perhitungan akhir dari *output feature map* atau sesudah proses perhitungan konvolusi [7].

Tensorflow Lite merupakan evolusi dari *Tensorflow Mobile* yang sudah mendukung *deployment* untuk perangkat *mobile*. Selama adanya tren pengembangan *Machine Learning* pada perangkat *mobile* dan juga ekspektasi *users* yang tinggi, *Tensorflow Lite* telah dikembangkan dengan maksimal, sehingga dapat digunakan dengan ringan pada perangkat *mobile*. Beberapa dari optimasi yang telah dilakukan pada *Tensorflow Lite* diantaranya adalah *hardware acceleration* melalui *silicon layer*, optimasi pada *frameworks* seperti *Android Neural Network API* (ANN API) dan *Artificial Neural Network* (ANN) [9].

Keras merupakan perangkat lunak jaringan yang berbasis *open source*, dan ditulis dengan menggunakan bahasa pemrograman *Python*. *Keras* juga dapat dijalankan menggunakan *MXNet*, *Tensorflow*, *Deeplearning4j*, dan *Theano* yang

JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

khusus dirancang guna mempercepat eksperimen yang berhubungan dengan *Deep Learning* [7]. Fitur yang menonjol dari *Keras* adalah *Tensorflow* dan *Theano* digunakan sebagai *backend*. *Keras* juga dapat berjalan lancar di kedua CPU dan GPU. Keras mendukung hampir semua model jaringan saraf mulai dari *convolutional*, *pooling*, *recurrent*, *embedding*, dan lainnya. Selanjutnya, model ini dapat dikombinasikan untuk membangun model yang lebih kompleks [10].

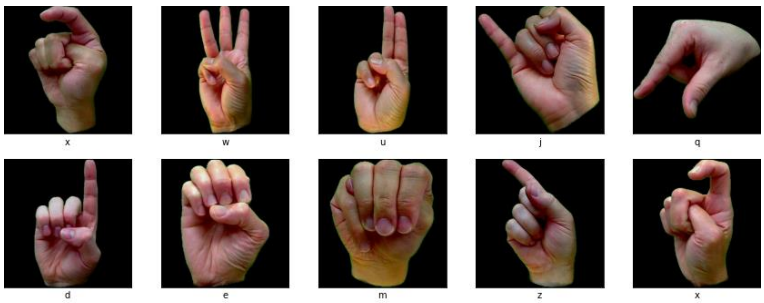
Android adalah sistem operasi yang menyediakan *platform* terbuka untuk membuat aplikasi *smartphone* sesuai dengan keinginan para *developers* dimana sistem operasi ini berbasis Linux [11]. Awalnya, Android dikembangkan oleh perusahaan kecil di Silicon Valley yang bernama Android Inc. Kemudian, Google mengambil alih sistem operasi tersebut pada tahun 2005 dan mencanangkannya sebagai sistem operasi yang bersifat *open source*, sehingga siapapun dapat menggunakannya secara gratis Sistem operasi Android terus berkembang dari versi pertamanya yang merupakan versi 1.0, hingga sekarang sudah berkembang sampai dengan versi ke 12 [12].

Python adalah bahasa pemrograman *interpretative* multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Multi paradigm programming* juga didukung oleh *Python*. *Python* menyediakan berbagai macam fitur, salah satunya adalah bahasa pemrograman yang dinamis dan dilengkapi dengan manajemen memori secara otomatis. *Python* didistribusikan dengan beberapa lisensi, namun lisensi dari *Python* tidak bertentangan baik menurut definisi *open source* maupun *General Public License* (GPL) [10]. *Kotlin* adalah bahasa pemrograman berbasis *Java Virtual Machine* (JVM) yang dikembangkan oleh JetBrains. *Kotlin* merupakan bahasa pemrograman yang pragmatis untuk Android yang mengkombinasikan *Object Oriented* (OO) dan pemrograman fungsional. *Kotlin* juga merupakan bahasa pemrograman yang interopabilitas yang membuat bahasa ini dapat digabungkan dalam satu *project* dengan bahasa pemrograman Java. Bahasa pemrograman ini bahkan dapat digunakan untuk pengembangan aplikasi berbasis *desktop*, *web*, dan bahkan *backend* [13].

2. METODE PENELITIAN

2.1 Pengumpulan Datasets

Dalam melaksanakan pengembangan sistem aplikasi penerjemah bahasa isyarat, langkah pertama yang dilakukan adalah pengumpulan *datasets* yang berupa kumpulan gambar tangan yang membentuk sebuah simbol bahasa isyarat. *Datasets* tersebut nantinya akan digunakan sebagai *input* untuk model *Tensorflow Lite* yang akan diproses oleh sistem. Pada penelitian ini akan menggunakan *datasets* yang bersumber dari *website* bernama Kaggle pada url berikut https://www.kaggle.com/ayuraj/asl-dataset. Kaggle merupakan *website* yang berisi tentang *Machine Learning* dan *Data Science*, pada *website* tersebut terdapat berbagai *open datasets* yang dapat digunakan dengan bebas oleh *developers*.



Gambar 3. Sampel Datasets *American Sign Language* (ASL)

Datasets bahasa isyarat yang digunakan adalah berjenis *American Sign Language* (ASL) yang telah diunggah pada tahun 2019 oleh Ayush Takur, Machine Learning Engineer yang berasal dari India. Kumpulan gambar pada *datasets* yang digunakan memiliki ukuran dimensi sebesar 400 x 400 pixel, dengan jumlah gambar sejumlah 1820 buah. Gambar 3 merupakan beberapa sampel gambar dari *datasets* yang digunakan.

2.2 Preprocessing Datasets

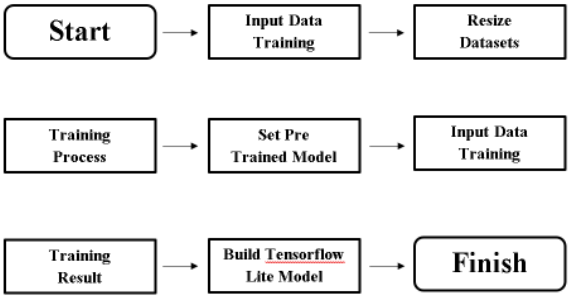
Data *preprocessing* merupakan tahapan dimana data akan dilakukan pengisian data yang kosong, menghilangkan duplikasi data, memeriksa inkonsistensi data, pembersihan data serta memperbaiki kesalahan pada data [14]. *Datasets* dari *American Sign Language* (ASL) memiliki ukuran sebesar 400 x 400 pixel dan akan menjadi ukuran yang sangat besar untuk melakukan proses *training* data. Maka dari itu, pada *preprocessing* data ini, akan dilakukan *resize* dimensi ukuran gambar yang sebelumnya berukuran 400 x 400 pixel, menjadi 20 x 20 *pixel.*

JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

2.3 Training Datasets

Proses *training datasets* merupakan tahapan dalam proses pengembangan model, dimana pada proses ini *datasets* dilatih agar dapat memperoleh akurasi yang tinggi dari klasifikasi yang dilakukan. Pada tahap ini *datasets* akan dilakukan proses *training* dengan menggunakan metode CNN. Pengujian training *datasets* dilakukan dengan menggunakan 100 *Epoch*, 120 *Epoch*, dan 200 *Epoch*. *Pre-trained Model* juga akan digunakan pada proses training, dengan tujuan didapatkannya akurasi tinggi ketika proses training selesai.

Pre-trained Model yang digunakan pada proses *training* ini adalah berjenis *EfficientNet Lite4* yang merupakan turunan dari *EfficientNet Lite*. Desain dari *EfficientNet Lite* memang sudah dirancang agar mendapatkan performa bagus ketika dijalankan pada *MobileCPU*, GPU (*Graphics Processing Unit*), dan *EdgeTPU*. *EfficientNet Lite* memiliki kemampuan agar dapat dijalankan pada perangkat *mobile* dan memiliki lima jenis varian, dimulai dari *EfficientNet Lite0* yang memiliki latensi terendah, hingga *EfficientNet Lite4* yang memiliki akurasi paling tinggi [15]. Dibawah ini adalah alur *training datasets* hingga ke pengembangan model *Tensorflow Lite* yang nantinya akan dilakukan *deployment* ke perangkat Android.



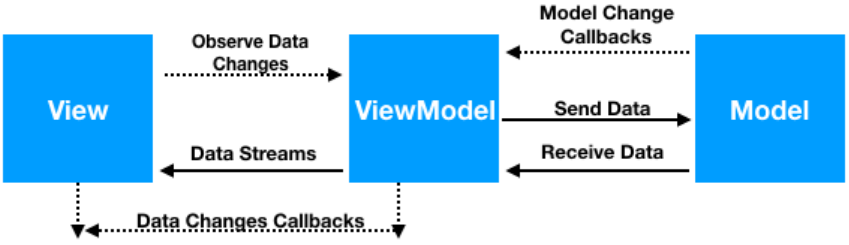
Gambar 4. Alur *Training Datasets*

2.4 Pengembangan Aplikasi Android

Setelah model *Tensorflow Lite* selesai dibuat, akan dilakukan pengembangan aplikasi berbasis Android. Karena nantinya *deployment* model tersebut akan dilakukan pada aplikasi Android. Aplikasi ini sendiri akan dikembangkan dengan menggunakan Android Studio dan mempunyai beberapa fitur. Diantaranya adalah fitur *login*, *register*, *forgot password*, *profile*, *about me*, *dictionary* (kamus yang memuat daftar bahasa isyarat), *edit profile*, *bookmarks*, dan yang terakhir adalah fitur deteksi yang akan menerjemahkan bahasa isyarat menjadi sebuah tulisan *alphabet*.

Pada fitur *login*, *register*, dan juga pada *profile*, akan dikembangkan dengan menggunakan *service* dari Firebase. Untuk pola arsitektur yang digunakan pada pengembangan aplikasi ini adalah dengan menggunakan Arsitektur MVVM (*Model-View-ViewModel*). Pola arsitektur ini digunakan dengan tujuan agar dapat memisahkan penggunaan data dan juga *view* agar tidak dalam satu *activity*. Dengan dipisahkannya antara penggunaan data dan juga *view*, akan dapat memudahkan kita untuk melakukan *maintain project*, jika sewaktu-waktu terjadi masalah. Dibawah ini adalah bentuk pola arsitektur MVVM yang ada pada Android.

Kemudian dalam pengembangan halaman *Dictionary*, disini peneliti menggunakan *Room Database* sebagai *database* lokal. Penggunaan *database* lokal ini berguna agar pada halaman *Dictionary* tersebut dapat diakses secara *offline* dan dapat digunakan untuk pengembangan halaman *Bookmarks*.



Gambar 5. Pola Arsitektur MVVM

Dalam pengembangan fitur dictionary dan juga bookmarks, untuk memunculkan sebuah list data, dilakukan terlebih dahulu pembuatan adapter. Adapter tersebut berguna untuk melakukan looping item yang terdapat pada data dictionary,

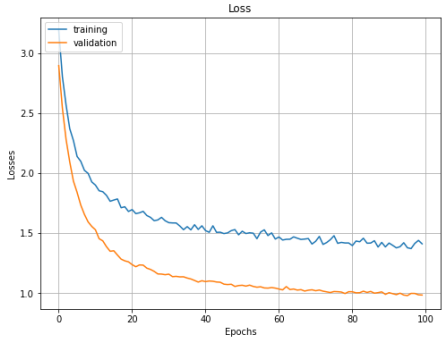
JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

dan akan memunculkan item tersebut sesuai dengan jumlah item yang ada.

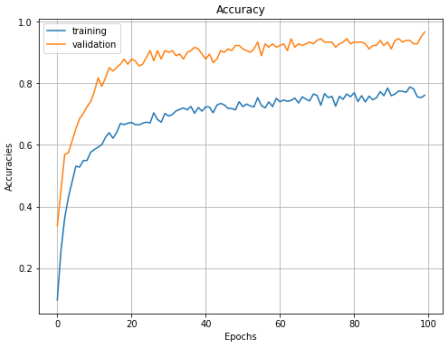
3. HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian Epoch

Pengujian dilakukan dengan tujuan untuk melihat perbedaan pengaruh dari hasil pelatihan datasets dengan jumlah langkah pelatihan (*Epoch*) yang berbeda. Sebagai perbandingannya, pengujian dilakukan dengan menggunakan 100 *Epoch*, 150 *Epoch*, dan 200 *Epoch*. Berikut adalah hasil dari beberapa pelatihan yang telah dilakukan.

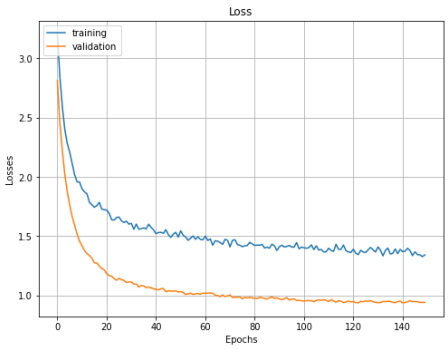


Gambar 6. *Training Loss* Sistem *Epoch* 100



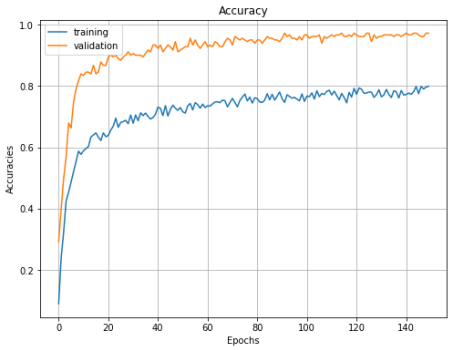
Gambar 7. *Training Accuracy* Sistem *Epoch* 100

Berdasarkan Gambar 6 dan Gambar 7 dapat diketahui bahwa dengan melakukan pelatihan dengan jumlah Epoch 100, dapat mencapai akurasi hingga 95% dan mendapatkan *Training Loss* sebesar 0,95. *Training Loss* tersebut masih tergolong cukup besar, dan perlu dilakukan *training* kembali dengan *Epoch* yang lebih besar.



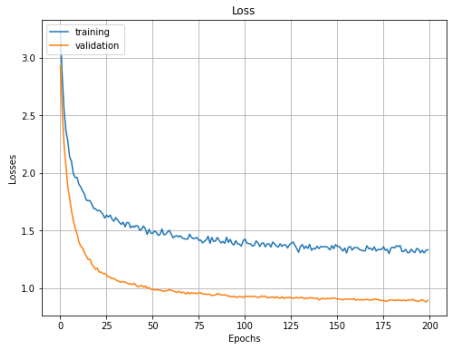
Gambar 8. *Training Loss* Sistem *Epoch* 150

JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

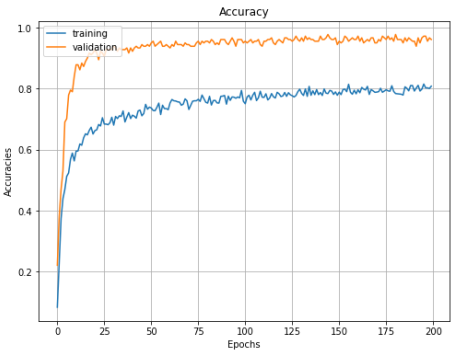


Gambar 9. *Training Accuracy* Sistem *Epoch* 150

Berdasarkan Gambar 8 dan Gambar 9 dapat diketahui bahwa dengan melakukan pelatihan dengan jumlah *Epoch* 150, tingkat akurasi menjadi turun hingga 93% dan mendapatkan *Training Loss* sebesar 0,93. Dengan bertambahnya *Epoch*, seharusnya tingkat akurasi semakin bertambah. Tetapi untuk pelatihan yang telah dilakukan ini, dengan 150 *Epoch* tingkat akurasi menjadi berkurang sebanyak 2%.



Gambar 10. *Training Loss* Sistem *Epoch* 200



Gambar 11. *Training Accuracy* Sistem *Epoch* 200

Berdasarkan Gambar 10 dan Gambar 11, dapat diketahui bahwa *training* dengan 200 *Epoch* dapat menambah akurasi sampai dengan 97% dan mendapatkan *Training Loss* sebesar 0,89. Jumlah langkah pelatihan (*Epoch*) yang dilakukan secara berulang-ulang sampai dengan 200 kali, membuat model menjadi semakin akurat dan mengurangi tingkat *Training Loss*. Model dengan hasil akurasi tertinggi akan dilakukan *deployment* pada fitur penerjemahan bahasa isyarat.

4.2 Hasil Pengujian dan Analisis

Pada bagian ini, akan dijelaskan beberapa hal mengenai hasil pengujian dari pelatihan menggunakan *Epoch* berjumlah 100, 150, dan 200.

JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

Tabel 1. Hasil Pengujian Sistem

Jumlah Datasets	Image Size	Epoch	Accuracy (%)	Training Loss
1820	20 x 20 px	100	95%	0,95
		150	93%	0,93
		200	97%	0,89

Berdasarkan tabel hasil pengujian diatas, dapat disimpulkan bahwa dengan *datasets* sejumlah 1820 gambar, ukuran *image* 20 x 20 pixel dan *Epoch* berjumlah 100, dapat menghasilkan akurasi sampai dengan 95% dengan *Training Loss* sebanyak 0,95. Kemudian dengan *Epoch* berjumlah 150, dapat menghasilkan akurasi yang lebih sedikit kecil dari sebelumnya, yaitu 93%, dengan *Training Loss* sebanyak 0,93. Pengujian yang terakhir yaitu dengan menggunakan *Epoch* sejumlah 200, dapat menghasilkan akurasi sebesar 97% dengan *Training Loss* sebesar 0,89.

Dari hasil pengujian tersebut dapat diketahui bahwa perbedaan jumlah *Epoch* dapat berpengaruh dalam tingkat akurasi yang dihasilkan. Semakin besar jumlah *Epoch* maka semakin besar pula tingkat akurasi yang dihasilkan [16]. Selain akurasi, ada juga *Training Loss* yang mengindikasikan seberapa baik model cocok dengan data pelatihan. Semakin besar *Training Loss*, dapat diartikan model tidak begitu bagus untuk digunakan. Namun, seiring banyaknya pelatihan yang dilakukan, nilai dari *Training Loss* akan menjadi berkurang [17].

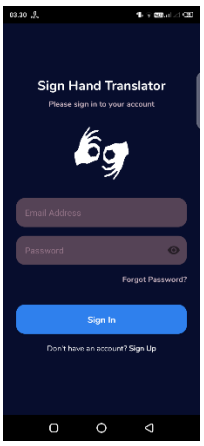
4.3 Hasil Pengembangan Aplikasi Android

Telah dilakukan pengembangan aplikasi berbasis Android dengan menggunakan Android Studio untuk aplikasi penerjemah bahasa isyarat. Berikut merupakan beberapa tampilan UI dari aplikasi yang telah dibuat dan dikembangkan oleh peneliti.

Gambar 12. *Splash Screen*

Pada Gambar 12 menunjukkan tampilan halaman *Splash Screen* yang akan muncul ketika aplikasi pertama kali dibuka. Halaman ini dikembangkan dengan menggunakan *Handler* yang berguna untuk memunculkan halaman ini secara sesaat. Tampilan dari halaman ini akan muncul selama 1,5 detik, dan setelah itu halaman akan melakukan navigasi otomatis menuju halaman *Sign In*.

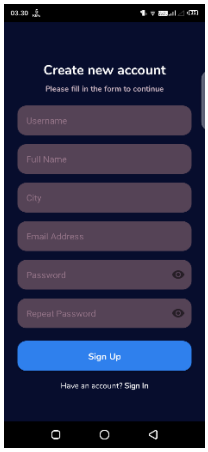
JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index



Gambar 13. Tampilan Halaman *Sign In*

Pada Gambar 13 menunjukkan halaman *Sign In* yang dapat digunakan untuk melakukan aktivitas *login* pada aplikasi penerjemah bahasa isyarat. Komponen penting yang diperlukan untuk melakukan aktivitas *login* pada halaman ini adalah *email* dan *password* yang sudah terdaftar melalui halaman *Sign Up*. Setelah mengisikan *email* dan *password* pada form yang tersedia, kita dapat melakukan klik pada *button Sign Up* untuk melakukan *login*. Kemudian aplikasi akan melakukan verifikasi data pada *service* Firebase. Kita juga dapat melakukan *reset password* melalui *text* "*Forgot Password"* yang terdapat diatas *button Sign In*, dan setelah melakukan klik pada *text* tersebut, halaman *Sign In* akan melakukan navigasi menuju halaman *Forgot Password.*

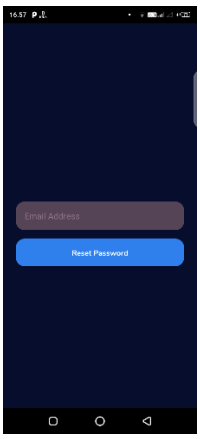
Pada halaman ini juga terdapat fitur penting untuk pengamanan *password* ketika melakukan aktivitas *login*, yaitu adalah fitur *hide password* yang dapat diaktifkan melalui *icon* berbentuk mata pada bagian *form password*. Kemudian, bagian bawah *button Sign In* terdapat *text "Don't have an account?* ***Sign Up****"*. Pada text tersebut dapat dimanfaatkan untuk melakukan pembuatan akun, yang nantinya setelah melakukan klik pada *text* tersebut halaman *Sign In* akan melakukan navigasi menuju halaman *Sign Up* untuk pembuatan akun aplikasi penerjemah bahasa isyarat. Halaman ini juga sudah dilakukan pengembangan, agar ketika jika kita sebelumnya sudah pernah melakukan *login*, maka halaman ini akan langsung menavigasikan kita menuju halaman *Home*.



Gambar 14. Tampilan Halaman *Sign Up*

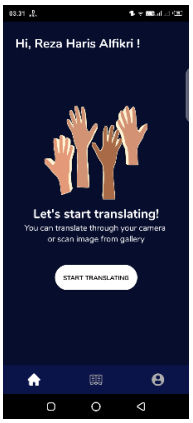
Pada Gambar 14 menunjukkan halaman *Sign Up* yang dapat digunakan untuk melakukan pendaftaran akun pada aplikasi penerjemah bahasa isyarat. Halaman ini akan tampil setelah melakukan klik pada *text "Don't have an account?* ***Sign Up****"* di halaman *Sign Up*. Kita dapat melakukan pendaftaran akun untuk aplikasi penerjemah bahasa isyarat dengan melakukan pengisian data pada *form* yang tersedia, dan jika sebelumnya kita sudah memiliki akun, kita dapat kembali ke halaman *Sign In* dengan menekan *button back* atau memanfaatkan *text "Have an account?* ***Sign In****"* untuk kembali ke halaman *Sign In*. Pada saat pengisian form pendaftaran akun, kita juga dapat menyembunyikan *password* kita dengan cara melakukan klik pada *icon* berbentuk mata seperti pada halaman *Sign Up.* Aplikasi akan melakukan *request* pendaftaran akun ke *service* Firebase, setelah kita melakukan pengisian data dan melakukan klik pada *button Sign In*.

JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index



Gambar 15. Tampilan Halaman *Forgot Password*

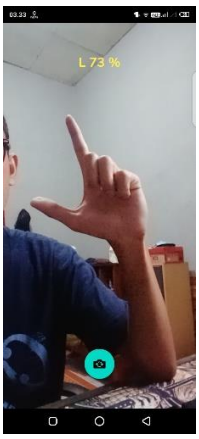
Pada Gambar 15 menunjukkan halaman *Forgot Password* yang dapat digunakan untuk melakukan *reset password*. Kita dapat mengakses halaman ini dengan melakukan klik pada *text "Forgot Password"* yang ada pada halaman *Sign In*. Untuk melakukan *reset password*, dapat dilakukan dengan mengisi *form email* yang tersedia dan melakukan klik pada *button Reset Password*. Setelah itu aplikasi akan melakukan *request reset password* ke *service* Firebase, dan jika *request reset password* berhasil dilakukan, kita akan mendapatkan *email* konfirmasi untuk melakukan *reset password*.



Gambar 16. Tampilan Halaman *Home*

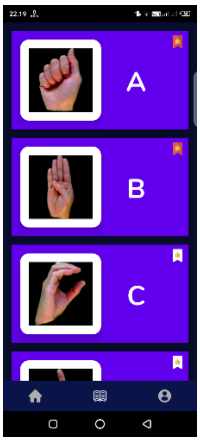
Pada Gambar 16 menunjukkan halaman utama dari aplikasi penerjemah bahasa isyarat. Kita dapat melakukan penerjemahan dengan melakukan navigasi ke halaman *Translation*, dengan melakukan klik pada *button Start Translating*. Pada halaman ini juga terdapat *bottom navigation* yang dapat digunakan untuk melakukan navigasi ke halaman *Dictionary* dan *Profile*. Halaman ini juga menggunakan *service* dari Firebase yang berguna untuk memanggil data *name* yang sebelumnya telah dibuat.

JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index



Gambar 17. Tampilan Halaman *Translation*

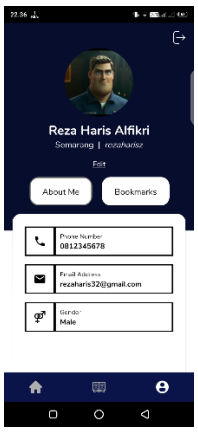
Pada Gambar 17 menunjukkan halaman *Translation*, yang merupakan fitur utama dari aplikasi penerjemah bahasa isyarat ini. Proses penerjemahan dilakukan dengan cara menunjukkan tangan kita yang membentuk simbol bahasa isyarat, setelah itu hasil terjemahan akan muncul dalam bentuk *text* beserta dengan akurasi yang didapatkan. Model *Tensorflow Lite* yang sebelumnya telah dikembangkan, berhasil dilakukan *deployment* pada halaman ini, dengan hasil akurasi sebesar 73%, untuk *alphabet* L seperti yang tertera pada Gambar 17. Tingkat akurasi tersebut tergolong berkurang, dari yang sebelumnya memiliki tingkat akurasi 97%, ketika dilakukan *testing* pada saat pengembangan model. Ketika proses prediksi berlangsung pada halaman ini, sebagian *label* mengalami salah prediksi dan beberapa *label* lainnya sudah dapat diprediksi dengan benar. Kita juga dapat mengubah arah kamera yang secara *default* menggunakan kamera depan ke kamera belakang, dengan cara melakukan klik pada *icon* kamera dan seketika kamera akan berubah menggunakan kamera belakang.



Gambar 18. Tampilan Halaman *Dictionary*

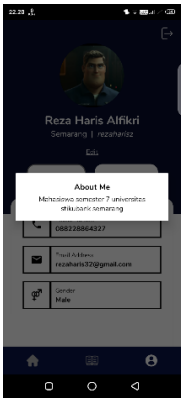
Pada Gambar 18 menunjukkan halaman *Dictionary* yang berisi *list* mengenai bahasa isyarat yang sudah ada. Bahasa isyarat yang tampil pada halaman ini adalah menggunakan data dari *database* lokal yang melakukan *import* data dari file *json* saat halaman ini pertama kali dibuka. Sedangkan untuk data dari gambar, masih menggunakan url, sehingga kita harus dalam keadaan *online* untuk mengakses gambar tersebut. Data dari gambar tersebut berasal dari Github *repository* milik peneliti, yang di dalamnya terdapat list dari gambar bahasa isyarat dengan tipe *American Sign Language* (ASL). Pada halaman ini, kita juga dapat menambahkan bahasa isyarat *favorite* pilihan kita dengan melakukan klik pada *icon bookmark*. Jika *icon* tersebut berubah warna dari putih menjadi *orange*, maka bahasa isyarat berhasil ditambahkan ke *database* lokal dan bahasa isyarat yang berhasil ditambahkan tersebut dapat diakses melalui halaman *Bookmarks*. Halaman ini dapat diakses melalui *bottom navigation* dengan cara melakukan klik pada *icon dictionary* yang ada pada *bottom navigation*.

JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index



Gambar 19. Tampilan Halaman *Profile*

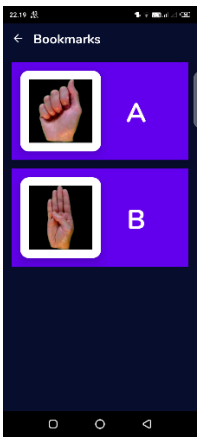
Pada Gambar 19 menunjukkan halaman *Profile* yang menunjukkan identitas dari akun kita yang sebelumnya telah dibuat pada halaman *Sign Up*. Halaman ini dapat diakses melalui *bottom navigation* dengan cara melakukan klik pada *icon profile* yang ada pada *bottom navigation*. Pada halaman ini terdapat beberapa fitur, diantaranya adalah fitur untuk melakukan aktivitas *logout* akun dengan cara melakukan klik pada *icon logout*. Kemudian ada *text "Edit"* yang berguna untuk menavigasikan kita menuju halaman *Edit Profile*, dibawahnya terdapat *button About Me* yang berguna untuk membuka *modal About Me*, dan *button Bookmarks* yang berguna untuk melakukan navigasi ke halaman *Bookmarks*. Selain itu, pada halaman ini juga menampilkan beberapa informasi pada akun. Seperti foto *profile*, nama lengkap, kota asal, negara, nomor telepon, *email*, dan yang terakhir adalah jenis kelamin.



Gambar 20. *Pop Up Modal About Me*

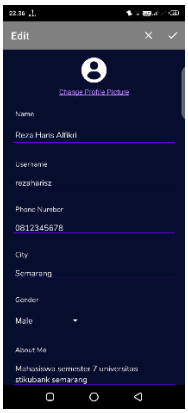
Pada Gambar 20 menunjukkan tampilan deskripsi dari *About Me* yang berbentuk *Pop Up Modal*. Deskripsi dari *About Me* ini dapat diubah melalui halaman *Edit Profile*. Halaman ini mengakses data dari *service* Firebase dari akun yang telah terdaftar sebelumnya.

JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index



Gambar 21. Tampilan Halaman *Bookmarks*

Pada Gambar 21 menunjukkan tampilan dari halaman *Bookmarks* yang berisi mengenai *Bookmarks* dari bahasa isyarat yang ditambahkan melalui halaman *Dictionary*. *List* bahasa isyarat yang tertampil pada halaman *Bookmarks* ini mengakses data dari *Room Database* yang merupakan *database* lokal. Untuk melakukan navigasi ke halaman sebelumnya, terdapat dua opsi yang dapat dilakukan. Pertama yaitu melalui *button back* yang telah dikembangkan yang letaknya ada pada bagian kiri atas halaman, dan yang kedua adalah melalui *button back* dari bawaan *smartphone* Android.



Gambar 22. Tampilan Halaman *Edit Profile*

Pada Gambar 22 menunjukkan halaman *Edit Profile*, yang dapat digunakan untuk merubah informasi *profile* dari akun yang telah digunakan untuk login pada halaman *Sign In*. Kita dapat melakukan perubahan untuk informasi pada akun kita, mulai dari foto *profile* yang dapat diubah dengan melakukan klik pada *text "Change Profile Picture"*. Kemudian kita juga dapat melakukan perubahan pada *name*, *username*, *phone number*, *city*, *gender*, dan yang terakhir adalah *about me*. Untuk bagian *gender* (jenis kelamin), telah dikembangkan dengan menggunakan *spinner*. Sehingga kita dapat memilih *gender* kita hanya dengan melakukan klik, dan tidak perlu mengetik *gender* kita secara manual.

Data yang dirubah pada halaman ini, akan dikirim ke *service* Firebase untuk dilakukan perubahan pada akun kita. Untuk menyimpan perubahan tersebut, kita harus melakukan klik pada *icon* centang yang ada pada bagian kanan atas layar, dan jika ingin membatalkan perubahan, kita perlu melakukan klik pada *icon* silang yang letaknya di sebelah *icon* centang atau kita juga dapat melakukan klik pada tombol *back* dari bawaan *smartphone* Android.

4. KESIMPULAN

Penelitian ini berhasil mengembangkan model guna untuk melakukan prediksi *images* dari bahasa isyarat dengan menggunakan metode *Convolutional Neural Network* (CNN). Pengembangan dilakukan dengan menggunakan *Pre-Trained Model* dengan tujuan untuk mendapatkan hasil akurasi yang tinggi. *Pre-Trained Model* yang digunakan adalah berjenis *Efficient-Net Lite1* dengan hasil akurasi yang didapatkan mencapai 97% dengan 200 *Epoch*. Pengembangan diuji dengan menggunakan 3 jumlah *Epoch*, yang mana adalah 100 *Epcoh*, 150 *Epoch*, dan 200 *Epoch*. Hasil dari pengujian

JURNAL TEKNOINFO
Volume 16, Nomor 2, Juli 2022, Page 183-197
ISSN: 1693-0010(Print), ISSN: 2615-224X(Online)
Available online at https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

Epoch tersebut diantaranya adalah *Epoch* 100 dengan hasil akurasi sampai dengan 95% dan *Training Loss* sebesar 0,95, kemudian *Epoch 150* mendapatkan akurasi sampai dengan 93% dan *Training Loss* sebesar 0,93, dan yang terakhir adalah *Epoch* 200 dengan hasil akurasi mencapai 97% dengan *Training Loss* sebesar 0,89. Akurasi tertinggi didapatkan oleh *Epoch* 200, dan model dengan akurasi tertinggi dikembangkan menjadi model dengan jenis *Tensorflow Lite* agar dapat dilakukan *deployment* pada aplikasi berbasis Android.

Setelah model tersebut dilakukan *deployment*, akurasi turun hingga 73% untuk prediksi pada bahasa isyarat *alphabet* L. Untuk sebagian label yang lain, ada yang mengalami salah prediksi dan mengakibatkan bahasa isyarat tidak dapat terprediksi dengan akurat. Namun, aplikasi masih dapat digunakan untuk melakukan proses penerjemahan, walaupun tidak semua bahasa isyarat dapat diprediksi dengan akurat.

Aplikasi penerjemah bahasa isyarat berhasil dikembangkan dengan arsitektur MVVM (*Model-View-ViewModel*), dengan total 10 halaman dan 1 *Pop Up* Modal. Beberapa halaman dan *Pop Up Modal* tersebut diantaranya adalah halaman *Splash Screen, Sign In*, *Sign Up*, *Forgot Password*, *Home*, *Translation*, *Dictionary*, *Profile*, *Edit Profile*, *Bookmarks*, dan yang terakhir adalah *Pop Up Modal* untuk *About Me*. Semua halaman, dan *Pop Up Modal* tersebut berfungsi dengan baik, dan untuk *database* yang digunakan yaitu menggunakan *database* lokal dan *online*. Untuk penggunaan *database* lokal digunakan pada halaman *Dictionary* dan *Bookmarks,* dan untuk penggunaan *database online* digunakan pada halaman *Sign In*, *Sign Up*, *Forgot Password*, *Profile*, *Edit Profile*, dan *Pop Up Modal* pada *About Me*. Database yang digunakan adalah *Room Database* sebagai *database* lokal dan Firebase sebagai *database online*.

Walaupun masih belum sempurna dan terdapat beberapa kekurangan seperti salah prediksi pada beberapa label dan kurang akuratnya prediksi setelah dilakukan *deployment*. Diharapkan aplikasi ini dapat membantu masyarakat agar dapat melakukan komunikasi dengan penyandang tunawicara, dan belajar mengenai bahasa isyarat dengan kamus yang telah disediakan pada halaman *Dictionary*.

DAFTAR PUSTAKA

[1] I. W. P. Suyadnya, I. P. W. A. Candra, N. A. N. Ginarsa, and I. M. Suartika, "Alat Bantu Komunikasi Terintegrasi bagi Penyandang Tuna Wicara Berbasis Sensor Gerak dan *OpenWrt*," *E-Journal SPEKTRUM*, vol. 5, no. 2, pp. 176-177, 2018, doi: 10.24843/SPEKTRUM.2018.v05.i02.p22.

[2] S. Liu, "Global number of internet users 2012-2021 by operating system," https://www.statista.com/statistics/543185/worldwide-internet-connected-operating-system-population/, 2021, diakses tanggal 14 Desember 2021.

[3] J. R. Raphael, "Android versions: A living history from 1.0 to 12," https://www.computerworld.com/article/3235946/android-versions-a-living-history-from-1-0-to-today.html, 2021 diakses tanggal 17 Desember 2021.

[4] N. F. Nissa, A. Janiati, N. Cahya, Anton, and P. Astuti, "Application of Deep Learning Using *Convolutional Neural Network* (CNN) Method for Women's Skin Classification," *Scientific Journal of Informatics*, vol. 8, no. 1, pp. 145-146, 2021, doi: 10.15294/sji.v8i1.26888.

[5] I. W. Suartika E. P, A. Y. Wijaya, and R. Soelaiman, "Klasifikasi Citra Menggunakan *Convolutional Neural Network* (CNN) pada Caltech 101," *JURNAL TEKNIK ITS*, vol. 5, no. 1, pp. 1-2, 2016, doi: 10.12962/j23373539.v5i1.15696.

[6] M. S. Utomo, S. Wibisono, W. Hadikurniawati, and H. Februariyanti, "SISTEM PEMBERIAN SARAN RESEP KULINER INDONESIA MENGGUNAKAN METODA CASE BASED REASONING DENGAN ALGORITMA SIMILARITAS CZEKANOWSKI BERBOBOT," *Prosiding SENDI_U*, pp. 257, 2019, ISBN: 978-979-3649-99-3.

[7] M. R. Alwanda, R. P. K. Ramadhan, and D. Alamsyah, "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," *Jurnal Algoritme*, vol. 1, no. 1, pp. 47-49, 2020, doi: 10.35957/algoritme.v1i1.434.

[8] E. N. Arrofiqoh, and Harintaka, "Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi," *GEOMATIKA,* pp. 3, doi: 10.24895/JIG.2018.24-2.810.

[9] O. Asling, "Mobile Object Detection using TensorFlow Lite and Transfer Learning," *Thesis*, Computer Science and Technology, KTH Royal Institute of Technology, Stockholm, 2018.

[10] C. N. Ihsan, "Klasifikasi Data Radar Menggunakan Algoritma *Convolutional Neural Network* (CNN)," *Journal of Computer and Information Technology*, vol. 4, no. 2, pp. 116-117, 2021, doi: 10.25273/doubleclick.v4i2.8188.

[11] A. S. Wijaya, and J. F. Andry, "PERANCANGAN APLIKASI *E-COMMERCE* BERBASIS ANDROID PADA UD HOKY CELLULER SHOP", *Jurnal TEKNOINFO*, vol. 5, no. 2, pp. 98, 2021, doi: 10.33365/jti.v15i2.1065.

[12] A. Kadir, "Pemrograman Aplikasi Android," *Penerbit Andi*, *Yogyakarta*, pp. 2-3, 2013. Available: https://andipublisher.com/produk-from-zero-to-a-pro-pemrograman-aplikasi-androidcd.

[13] N. S. Sibarani, G. Munawar, and B. Wisnuadhi, "Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin," *Industrial Research Workshop and National Seminar*, vol. 9, pp. 320, 2018, doi: 10.35313/irwns.v9i0.

[14] Y. Pristyanto, "PENERAPAN METODE *ENSEMBLE* UNTUK MENINGKATKAN KINERJA ALGORITME KLASIFIKASI PADA IMBALANCED DATASET," *Jurnal TEKNOINFO*, vol. 13, no. 1, pp. 13, 2019, doi: 10.33365/jti.v13i1.184.

[15] R. Liu, "Higher accuracy on vision models with EfficientNet-Lite," https://blog.tensorflow.org/2020/03/higher-accuracy-on-vision-models-with-efficientnet-lite.html, 2020, diakses tanggal 07 Desember 2021.
[16] R. N. Nurfita, and G. Ariyanto, "Implementasi Deep Learning berbasis Tensorflow untuk Pengenalan Sidik Jari," *jurnal Teknik elektro*, vol. 18, no. 1, pp. 25, 2018, doi: 10.23917/emitor.v18i01.6236
[17] Baeldung, "What is a Learning Curve in Machine Learning?," https://www.baeldung.com/cs/learning-curve-ml, 2020, diakses tanggal 25 Desember 2021.