

**IMPLEMENTASI MACHINE LEARNING PADA APLIKASI PENERJEMAH BAHASA ISYARAT INDONESIA MENGGUNAKAN TENSORFLOW**

**SKRIPSI**

**MUHAMMAD ASRUL AJI PANGESTU**

**NPM 17670067**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK DAN INFORMATIKA**

**UNIVERSITAS PGRI SEMARANG**

**2021**



**IMPLEMENTASI MACHINE LEARNING PADA APLIKASI PENERJEMAH BAHASA ISYARAT INDONESIA MENGGUNAKAN TENSORFLOW**

**SKRIPSI**

**MUHAMMAD ASRUL AJI PANGESTU**

**NPM 17670067**

**Diajukan kepada Fakultas Teknik dan Informatika**

**Universitas PGRI Semarang untuk Memenuhi Salah Satu Syarat**

**Memperoleh Gelar Sarjana**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK DAN INFORMATIKA**

**UNIVERSITAS PGRI SEMARANG**

**2020**

**SKRIPSI**

**IMPLEMENTASI MACHINE LEARNING PADA APLIKASI PENERJEMAH BAHASA ISYARAT INDONESIA MENGGUNAKAN TENSORFLOW**

**Disusun dan diajukan oleh**

**MUHAMMAD ASRUL AJI PANGESTU**

**17670067**

**telah disetujui pembimbing untuk dilanjutkan untuk**

**Disusun menjadi skripsi**

**Pada tahun 2020**

|  |  |
| --- | --- |
| **Pembimbing Utama,**  **Aris Trijaka Harjanta, S.Kom, M.Kom**  **NIDN .0619048202** | **Pembimbing Pendamping,**  **Noora Qotrun Nada, ST, M. Eng**  **NIDN .0626028201** |

ABSTRAK

Komunikasi merupakan proses penyampaian informasi yang dilakukan oleh satu pihak untuk terhubung dengan lingkungan atau pihak lain. Pada umumnya terdapat dua jenis komunikasi yaitu komunikasi verbal dan non-verbal yang biasa digunakan oleh Tuli. Dalam kehidupan sosial Tuli di Indonesia memiliki dua bahasa isyarat yang sering digunakan untuk berkomunikasi yaitu Bisindo dan SIBI. Tetapi cara berkomunikasi dengan bahasa isyarat dapat membatasi ketika berkomunikasi dengan orang lain yang tidak tuli. Masyarakat umum atau lawan komunikasinya akan sulit memahami karena perbedaan cara berkomunikasi tersebut. Secara teoritis jika bahasa isyarat mengadopsi teknologi digital seperti *machine learning*, maka kita dapat dengan mudah untuk mempelajari bahasa isyarat dan mempermudah komunikasi. Oleh sebab itu, penulis akan mengimplementasikan *machine learning* menggunakan *library* Tensorflow pada aplikasi berbasis android dengan tujuan sebagai pembelajaran dan media untuk mempermudah komunikasi antara masyarakat dengan penyandang tunarungu.

Kata kunci: komunikasi, Tuli, machine learning*,* Tensorflow, Android, YOLOv4

prakata

Dengan memanjatkan puji syukur kehadirat Allah SWT yang telah memberikan Rahmat dan Karunia-Nya sehingga penyusunan Skripsi yang berjudul ”Implementasi *Machine Learning* Pada Aplikasi Penerjemah Bahasa Isyarat Indonesia Menggunakan Tensorflow**”** Dapat selesai tepat pada waktunya.Penyusunan Skripsi ini merupakan salah satu syarat untuk menyelesaikan Program Studi Informatika S-1 pada Fakultas Teknik dan Informatika Universitas PGRI Semarang. Oleh karena itu, dalam kesempatan ini dengan tulus hati penulis menyampaikan terima kasih kepada:

1. Allah SWT Yang Maha Pemurah Lagi Maha Penyayang, yang telah memberikan kesempatan dan kemudahan dalam menyelesaikan Skripsi dengan lancar.
2. Bapak dan Ibu tercinta yang telah banyak memberikan dorongan baik moral, material serta doa yang selalu dihadiahkan kepada penulis. Semoga ini menjadi awal langkah sukses untuk penulis.
3. Dr. Muhdi, S.H., M.Hum selaku Rektor Universitas PGRI Semarang yang telah memberi kesempatan kepada penulis untuk menimba ilmu di Universitas PGRI Semarang.
4. Drs. Slamet Supriyadi, M.Env.,St. selaku Dekat Fakultas Teknik dan Informatika Universitas PGRI Semarang
5. Bambang Agus Herlambang, S.KOM., M.KOM. selaku Ketua Program Studi Informatika yang selalu memberikan motivasi kepada seluruh mahasiswa Informatika.
6. Aris Tri Jaka Harjanta, S.KOM., M.KOM. Selaku Pembimbing Petama yang yang telah membimbing penulis dengan penuh dedikasi yang tinggi dan dorongan serta motivasi untuk terus memperbaiki proposal skripsi ini.
7. Noora Qotrun Nada, ST, M. Eng Selaku Pembimbing Kedua yang telah membimbing penulis dengan penuh dedikasi dan kesabaran.
8. Seluruh Dosen Pengajar, Staff dan Karyawan Fakultas Teknik dan Informatika Universitas PGRI Semarang
9. Teman-teman Informatika Universitas PGRI Semarang angkatan 2017 terkhusus Informatika Kelas B yang telah bersama-sama dalam keadaan suka maupun duka selama kurang lebih tiga setengah tahun ini.
10. Teruntuk sahabat-sahabat penulis yang tidak bisa penulis sebutkan satu per satu, yang telah membantu penulis dan bertukar pikiran dalam penulisan Skripsi ini.

Akhirnya dengan segala kerendahan hati penulis mengharapkan skripsi ini dapat bermanfaat bagi pembaca dan kita semua yang membutuhkan

Semarang,

Muhammad Asrul Aji Pangestu

daftar isi

[ABSTRAK iv](#_Toc79057461)

[PRAKATA V](#_Toc79057462)

[DAFTAR ISI VII](#_Toc79057463)

[DAFTAR TABEL IX](#_Toc79057464)

[DAFTAR GAMBAR X](#_Toc79057465)

[DAFTAR LAMPIRAN xi](#_Toc79057466)

[BAB I PENDAHULUAN 1](#_Toc79057467)

[A. Latar Belakang 1](#_Toc79057468)

[B. Identifikasi Masalah 2](#_Toc79057469)

[C. Rumusan Masalah 2](#_Toc79057470)

[D. Batasan Masalah 2](#_Toc79057471)

[E. Tujuan Penelitian 3](#_Toc79057472)

[F. Manfaat Penelitian 3](#_Toc79057473)

[G. Penegasan Istilah 4](#_Toc79057474)

[BAB II KAJIAN PUSTAKA 5](#_Toc79057475)

[A. Tinjauan Pustaka 5](#_Toc79057476)

[B. Landasan Teori 9](#_Toc79057477)

[C. Kerangka Berpikir 20](#_Toc79057478)

[BAB III METODE PENELITIAN 22](#_Toc79057479)

[A. Pendekatan Penelitian 22](#_Toc79057480)

[B. Populasi dan Sampel 22](#_Toc79057481)

[C. Variabel Penelitian 22](#_Toc79057482)

[D. Teknik Pengumpulan Data 24](#_Toc79057483)

[E. Pengujian Aplikasi 24](#_Toc79057484)

[BAB IV HASIL PENELITIAN DAN PEMBAHASAN 26](#_Toc79057485)

[A. Hasil Tahapan Permulaan 26](#_Toc79057486)

[B. Hasil Sistem Machine Learning 2](#_Toc79057487)

[C. Hasil Sistem Aplikasi Android 2](#_Toc79057488)

[Daftar pustaka x](#_Toc79057489)

daftar tabel

[Tabel 1 Tabel Perbandingnan Penelitian Sebelumnya 7](#_Toc61232372)

[Tabel 2 Simbol-simbol pada Activity Diagram 16](#_Toc61232373)

[Tabel 3 Simbol-simbol pada Class Diagram 17](#_Toc61232374)

[Tabel 4 Relationship Multiplicity 18](#_Toc61232375)

[Tabel 5 Variabel Huruf Bisindo 21](#_Toc61232376)

[Tabel 6 Jadwal Kegiatan 23](#_Toc61232377)

daftar gambar

[Gambar 2. 1 Huruf Bisindo 11](#_Toc61232439)

[Gambar 2. 2 Actor Use Case Diagram 15](file:///D:\000SKRIPSI\Asrul%20Proposal.docx#_Toc61232440)

[Gambar 2. 3 Use Case 15](file:///D:\000SKRIPSI\Asrul%20Proposal.docx#_Toc61232441)

[Gambar 2. 4 Kerangka Berpikir 19](#_Toc61232442)

[Gambar 3. 1 Langkah-langkah metode R&D 20](#_Toc61232452)

DAFTAR LAMPIRAN

BAB I

pendahuluan

1. Latar Belakang

Komunikasi merupakan proses penyampaian informasi yang dilakukan oleh satu pihak untuk terhubung dengan lingkungan atau pihak lain. Pada umumnya terdapat dua jenis komuniasi yaitu komunikasi verbal dan non-verbal yang biasa digunakan oleh Tuli. Dalam kehidupan sosial Tuli di indonesia memiliki dua bahasa isyarat yang sering digunakan untuk berkomunikasi yaitu Bisindo (Bahasa Isyarat Indonesia) dan SIBI (Sistem Isyarat Bahasa Indonesia).

SIBI merupakan bahasa isyarat yang diciptakan oleh Alm. Anton Widyatmoko mantan kepala sekolah SLB/B Widya Bakti Semarang bekerjasama dengan mantan kepala sekolah SLB/B di Jakarta dan Surabaya. Kamus SIBI sudah diterbitkan oleh pemerintah dan disebarluaskan melalui sekolah-sekolah, khususnya ke sekolah untuk penyandang tunarungu di Indonesia (SLB/B) sejak tahun 2001[1]. Bahasa Isyarat Indonesia (Bisindo) merupakan adaptasi dari *American Sign Language* (ASL) yang disesuaikan dengan budaya asli Indonesia.

Tetapi cara berkomunikasi dengan bahasa isyarat dapat membatasi ketika berkomunikasi dengan orang lain yang tidak tuli. Masyarakat umum atau lawan komunikasinya akan sulit memahami karena perbedaan cara berkomunikasi tersebut. Secara teoritis jika bahasa isyarat mengadopsi teknologi digital seperti *machine learning*, maka kita dapat dengan mudah untuk mempelajari bahasa isyarat dan mempermudah komunikasi.

Pada umumnya *machine learning* merupakan proses dimana komputer mempelajari suatu hal dengan menggunakan banyak data yang telah diproses. Namun untuk melatih komputer dari awal akan menghabiskan banyak waktu dan memerlukan kumpulan data yang cukup banyak. Oleh karena itu aplikasi akan menggunakan *library* Tensorflow dan *pre-trained model* denganalgiritma *Convolutional Neural Network* untuk mempercepat proses pelatihan model yang digunakan. Diharapkan dengan mengimplementasikan *machine learning* pada aplikasi berbasis android dapat mempermudah dalam mempelajari huruf isyarat sehingga tidak ada lagi batas dalam berkomunikasi antara masyarakat dengan penderita tunarungu.

Berdasarkan latar belakang yang telah diuraikan, penulis akan membangun sebuah aplikasi berbasis android untuk membantu menerjemahkan bahasa melalui *object detection* dan mengambil judul “Implementasi *Machine Learning* Pada Aplikasi Penerjemah Bahasa Isyarat Indonesia Menggunakan Tensorflow”.

1. Identifikasi Masalah

Permasalahan penelitian dapat diidentifikasi sebagai berikut:

1. Sulitnya masyarakat untuk berkomunikasi dengan penderita tunarungu.
2. Kurangnya pengetahuan tentang bahasa Isyarat bagi masyarakat umum.
3. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, yang menjadi rumusan masalah dalam penelitian ini adalah sebagain berikut:

1. Bagaimana merancang dan membuat aplikasi penerjemah huruf Bisindo?
2. Bagaimana model hasil pelatihan pada aplikasi penerjemah huruf Bisindo?
3. Bagaimana aplikasi dapat mendeteksi setiap huruf Bisindo?
4. Bagaimana tingkat akurasi pendeteksian huruf Bisindo?
5. Batasan Masalah

Agar permasalahan tidak berkembang terlalu jauh dan fokus pada pemecahan masalah, maka penulis melakukan pembatasan masalah sebagai berikut:

1. Objek yang akan diolah dan diproses adalah huruf isyarat Bisindo, tidak termasuk angka dalam Bisindo.
2. Algoritma yang digunakan adalah *Convolutional Neural Network*
3. Aplikasi yang akan dibangun berbasis Android.
4. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dijabarkan sebelumnya, maka tujuan dari penelitian ini yaitu:

* + 1. Merancang dan membuat aplikasi penerjemah huruf Bisindo berbasis android.
    2. Mendapatkan model hasil pelatihan untuk deteksi huruf Bisindo.
    3. Mengintegrasikan *machine learning* untuk mendeteksi setiap huruf Bisindo ke dalam aplikasi android.
    4. Mengetahui nilai akurasi dalam pendeteksian huruf Bisindo

1. Manfaat Penelitian

Manfaat dari penyusunan penelitian adalah:

* + - 1. Bagi Mahasiswa

Untuk mempraktekkan ilmu yang diperoleh di bangku kuliah dan menerapkannya dalam lingkungan dan kehidupan yang membutuhkan.

* + 1. Bagi Akademik

Dapat memberikan alternative pembelajaran kepada siswa sehingga siswa dapat mengakses materi pembelajaran kapanpun dan dimanapun.

* + 1. Bagi Pembaca

Sebagai tambahan ilmu pengetahuan, rujukan, dan bahan acuan apabila melakukan penelitian lebih lanjut serta tambahan pengetahuan dalam mempelajari masalah-masalah yang ada.

1. Penegasan Istilah
2. *Machine Learning* adalah proses dimana komputer mempelajari suatu hal dengan menggunakan banyak data.
3. *Object Detection* adalah teknik komputer dalam menemukan suatu objek pada gambar atau video.
4. *Model* adalah berkas hasil akhir yang siap didistribusikan untuk keperluan deteksi hufuf Bisindo.
5. *Transfer Learning* adalah metode memanfaatkan *model* yang sudah ada dan memodifikasi menjadi model baru.
6. Android adalah suatu sistem operasi yang berbasis Linux untuk perangkat mobile.

BAB II

kajian pustaka

1. Tinjauan Pustaka

Referensi dari penelitian terdahulu menjadi hal yang sangat penting bagi penulis sebagai bahan kajian untuk mengatahui hubungan antara penelitian sebelumnya dengan penelitian yang dilakukan saat ini. Penelitian terdahulu juga bertujuan untuk memberi kontribusi penelitian bagi penulis agar penelitian dengan tema ini dapat terus berkembang. Berikut beberapa ulasan tentang penelitihan yang dilakukan terdahulu yang berhubungan dengan *machine learning* dan metode yang digunakan.

Penelitian mengenai “Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode *Convolutional Neural Network”* yang dilakukan oleh Mochamad Bagus Setiyo Bakti (2019). Dalam penelitian ini menggunakan metode *Convolutional Neural Network* dalam mengenali angka Sistem Isyarat Bahasa Indonesia. Pada penelitian ini proses training menghasilkan akurasi berbeda dari setiap tahapnya yaitu dengan 24 *epoch* sampai 100 *epoch* dapat mencapai tingkat akurasi 96.44% dan nilai0.13%. Dari percobaan yang dilakukan, proses prediksi angka dua selalu mengalami kesalahan[2].

Penelitian mengenai “Perancangan Sistem Pengenalan Jenis Tanaman Obat Dengan Kamera Berbasis Android” yang dilakukan oleh Widhar Dwiatmoko (2020). Pada penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) untuk pengenalan tanaman obat pada aplikasi berbasis android. Hasil dari penelitian ini menghasilkan tingkat akurasi 100% dengan total *epoch* 240. Hasil 100% didapat karena menghilangkan halangan pada dataset berupa penghapusan warna pada *background,* foto dalam keadaan penuh tanpa terpotong dan posisi gambar yang di samaratakan. Namun terjadi perubahan tingkat akurasi menjadi 43% ketika data yang digunakan adalah data *augmentation*[3]*.*

Penelitian mengenai “Implementasi Algoritma YOLO pada Aplikasi Pendeteksi Senjata Tajam di Android” yang dilakukan oleh Christopher Nathanael Liunanda (2020). Penelitian ini menggunakan model YOLOv3 dan YOLOv3-tiny untuk mendeteksi senjata tajam. Dengan menambahkan dataset 900 gambar negatif (gambar yang tidak berisi objek senjata tajam) YOLOv3 mendapatkan tingkat mAP 59% dan YOLOv3-tiny sebesar 56.64%. Tingkat akurasi setelah konversi langsung dari model Darkflow menjadi model Tensorflow Lite yaitu 72.7% pada model YOLOv3 sedangkan pada YOLOv3-tiny 63.6%. Pada penelitian ini model masih rentang melakukan deteksi salah ke objek mirip senjata tajam, rentan gagal deteksi pada objek *blur,* dan refleksi cahaya yang tinggi pada senjata[4].

Penelitian mengenai “Deep Learning untuk Deteksi Tanda nomor Kendaraan Bermotor Menggunakan Algoritma Convolutional Neural Network Dengan Python dan Tensorflow” oleh Imam Taufiq (2018). Penelitian ini menggunakan *Convolutional Neural Network* CNN dalam mendeteksi tanda kendaraan bermotor. Akurasi yang dihasilkan dari penelitian ini adalah sekitar 99% dengan 502 dataset, 80% untuk training dan 20% untuk testing. Proses training membutuhkan lebih dari 25.000 step dengan jumlah *batch* 8 untuk mendapatkan tingkat akurasi sebesar 99% pada tanda nomor kendaraan[5].

Penelitian mengenai “Deep Learning Object Detection Pada Video Menggunakan Tensorflow dan *Convolutional Neural Network* (Studi Kasus: Klasifikasi Gambar Meja dan Kursi Motif Ukiran Jepara)” oleh Syarifah Rosita Dewi (2018). Pada penelitian ini menggunakan Algoritma *Convolutional Neural Network* (CNN) dan Tensorflow dalam mendeteksi motif-motif pada gambar meja dan kursi ukuran Jepara. Model hasil *training* dengan jumlah 250000 *steps* dan 2 *batch size* menghasilkan tinggkat akurasi berkisar antara 70% hingga 99%[6].

Penelitian mengenai “*Human Related-Health Actions Detection using Android Camera based on TensorFlow Object Detection API*” yang dilakukan oleh Fadwa Al-Azzo (2018). Penelitian ini menggunakan Tensorflow dan *pre-trained model* Faster-R-CNN-Resnet dalam mendeteksi gerakan-gerakan yang berkaitan dengan kesehatan seperti bersin, sakit perut, mual dan lain-lain. Dataset yang digunakan terdiri dari 56000 dan memiliki 4000000 *frames.* Tingkat akurasi yang didapat mencapai 93,8% untuk Faster R-CNN-Resnet dan 95,8% untuk SSD-Mobilenet[7].

Tabel 1 Tabel Perbandingnan Penelitian Sebelumnya

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Penulis | Judul Penelitian | Metode | Hasil |
| 1 | Mochamad Bagus Setiyo Bakti (2019) | Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network | Convolutional Neural Netwok (CNN) | Penelitian ini menggunakan metode CNN untuk mendeteksi angka Sistem Isyarat Bahasa Indonesia (SIBI). Penelitian ini menghasilkan akurasi mencapai 98,89%. Dan mengalami kegagalan pada proses deteksi angka 2 |
| 2 | Widhar Dwiatmoko (2020) | Perancangan Sistem Pengenalan Jenis Tanaman Obat Dengan Kamera Berbasis Android | Convolutional Neural Netwok (CNN) | Penelitian ini menghasilkan akurasi 100% dengan *epoch* 250 pada objek tanaman obat tanpa halangan dan gambar utuh tanpa terpotong, sedangkan pada gambar normal atau asli akurasi mencapai 43%. |
| 3 | Christopher Nathanael Liunanda, dkk. (2020) | Implementasi Algoritma YOLO pada Aplikasi Pendeteksi  Senjata Tajam di Android | Algoritma You Only Look Once (YOLO) | Penelitian yang dilakukan yaitu deteksi objek pada senjata tajam dengan object detection network YOLO. Tingkat akurasi YOLOv3 dalam mendeteksi objek senjata tajam mencapai 72.7% setelah dikonversi ke Tensorflow Lite. Sedangkan waktu interferensi rata-rata yaitu 956ms atau 0.9 detik. |
| 4 | Imam Taufiq (2018) | Deep Learning untuk Deteksi Tanda nomor Kendaraan Bermotor Menggunakan Algoritma Convolutional Neural Network Dengan Python dan Tensorflow | Algoritma CNN dan Tensorflow | Penelitian ini menggunakan CNN dalam mendeteksi tanda kendaraan bermotor. Akurasi yang dihasilkan dari penelitian ini adalah sekitar 99% dengan502 dataset, 80% untuk training dan 20% untuk testing. |
|  |  |  |  |  |
| 5 | Syarifah Rosita Dewi (2018) | Deep Learning Object Detection Pada Video Menggunakan Tensorflow dan Convolutional Neural Network | Algoritma CNN dan Tensorflow | Tingkat akurasi pada pengujian ini berkisar antara 70% hingga 99% dengan objek berupa meja dan kursi motif ukiran Jepara. |
| 6 | Fadwa Al-Azzo, dkk (2018) | *Human Related-Health Actions Detection using*  *Android Camera based on TensorFlow Object*  *Detection API* | Faster R-CNN-Resnet dan SSD-Mobilenet | Pengujian yang dilakukan berupa deteksi gerakan yang berkaitan dengan kesehatan seperti bersin, sakit perut, mual dan lain-lain. Tingkat akurasi mencapai 93,8% untuk Faster R-CNN-Resnet dan 95,8% untuk SSD-Mobilenet |

1. Landasan Teori

Landasan teori merupakan penjelasan dari konsep dasar dan teori-teori yang tersusun secara sistematis dan berkaitan dengan penelitian. Untuk mendukung penyusunan penelitian, dibutuhkan landasan-landasan teori yang mendukung penyusunan searah dengan landasan teori tersebut.

1. Bahasa Isyarat

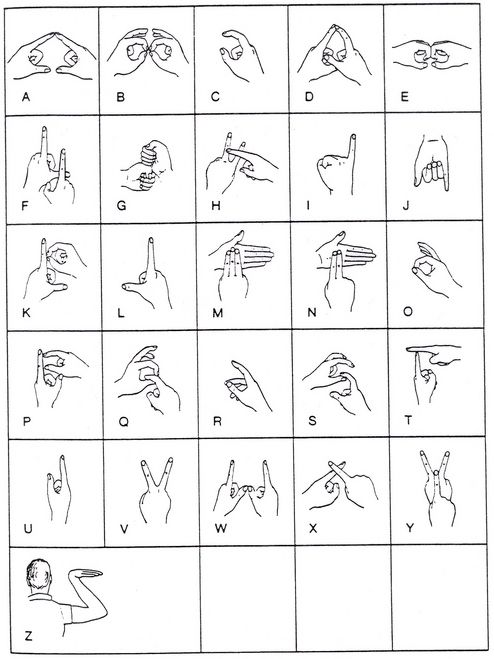
Bahasa isyarat merupakan bahasa yang mengutamakan komunikasi manual, yaitu menggunakan bahasa tubuh, tangan dan gerak bibir, bukan suara lisan. Kaum tunarungu adalah kelompok utama yang menggunakan bahasa ini, biasanya dengan mengkombinasikan bentuk tangan, orientasi dan gerak tangan, lengan, dan tubuh, serta ekspresi wajah untuk mengungkapkan pikiran mereka.

Bahasa isyarat unik dalam jenisnya di setiap negara. Bahasa isyarat bisa saja berbeda di negara-negara yang berbahasa sama. Contohnya, Amerika Serikat dan Inggris meskipun memiliki bahasa tertulis yang sama, memiliki bahasa isyarat yang sama sekali berbeda (American Sign Language dan British Sign Language).

Untuk di negara Indonesia sendiri terdapat dua jenis bahasa isyarat yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (Bisindo) namun sistem yang sekarang umum digunakan adalah Sistem Isyarat Bahasa Indonesia (SIBI) dimana sistem ini sama dengan bahasa isyarat yang diterapkan di Amerika (ASL - American Sign Language)[8].

1. Bisindo (Bahasa Isyarat Indonesia)

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].



Gambar 2. 1 Huruf Bisindo

1. *Machine Learning*

*Machine learning* atau bisa disebut juga dengan pembelajaran mesin merupakan bagian dari AI atau yang dikenal dengan (*Artificial Intelligence*). Dimana *machine learning* sendiri digunakan untuk menggantikan atau menirukan perilaku dari manusia untuk menyelesaikan sebuah permasalahan. Ciri tertentu dari mesin learning adalah dimana adanya proses pelatihan, pembelajaran, atau training. ML (*Machine Learning*) memiliki metode yaitu klasifikasi yang digunakan untuk memilah atau mengklasifikasi objek berdasarkan ciri tertentu sama seperti halnya manusia mencoba untuk membedakan benda satu dengan yang lainnya[9].

1. *Deep Learning*

*Deep learning* adalah salah satu bidang *machine learning* yang memanfaatkan banyak layer pengolahan informasi nonlinier untuk melakukan ekstraksi fitur, pengenalan pola, dan klasifikasi.

1. *Convolutional Neural Network*

*Convolutional network* atau yang dikenal dengan *Convolutional Neural Network* (CNN) adalah tipe khusus dari *neural network* untuk memproses data yang mempunyai topologi jala atau *grid-like topology*. Pemberian nama *Convolutional Neural Network* mengindikasikan bahwa jaringan tersebut menggunakan operasi matematika yang disebut konvolusi. Konvolusi sendiri adalah sebuah operasi linear. Jadi *convolutional network* adalah *neural network* yang menggunakan konvolusi minimal pada salah satu lapisannya[10].

1. *Transfer Learning*

*Transfer Learning* adalah suatu teknik atau metode yang memanfaatkan model yang sudah dilatih terhadap suatu dataset untuk menyesuaikan permasalahan lain yang serupa dengan cara menggunakannya sebagai *starting point*, memodifikasi dan mengupdate parameternya sehingga sesuai dengan dataset yang baru[11].

Melatih CNN dari awal sangat jarang dilakukan, karena memerlukan data dalam jumlah besar dan dapat memakan waktu yang cukup lama. Untuk itu, terdapat teknik yang disebut dengan *Transfer Learning* yang menggunakan jaringan yang sudah ada sebelumnya. Jaringan dasar dengan dataset terlatih seperti kumpulan dataset dari ImageNet yang mempunyai lebih dari 1,2 juta gambar dan 1000 kategori. Jaringan dari ImageNet ini berfungsi sebagai jaringan dasar yang digunakan pada *Transfer Learning*. Ketika jaringan terlatih digunakan sebagai *feature extractor*, maka layer yang terhubung dengan CNN dasar akan dihapus dan dua lapisan baru akan ditambahkan ke layer[12].

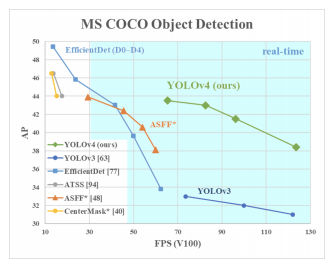
1. *Object Detection*

*Object Detection* adalah teknik visi komputer untuk menemukan contoh objek dalam gambar atau bideo. Algoritma deteksi objek biasanya memandaatkan pembelajaran mesin atau pembelajaran mendalam untuk menghasilkan hasil yang bermakna. Ketka manusia melihat gambar atau video, manusia dapat mengenali dan menemukan objek dalam beberapa saat berbeda dengan komputer yang memerlukan komputasi yang kompleks. Tujuan deteksi objek adalah untuk mereplikasi kecerdasan yang dimiliki manusia dalam melihat benda menggunakan komputer. Cara kerja deteksi objek deteksi objek menempatkan keberadaan objek dalam gambar dan menggambar kotak pembatas di sekitar objek itu. ini biasanya melibatkan dua proses, yaitu mengklasifikasikan jenis objek, dan kemudian menggambar kotak di sekitar objek itu[13].

1. YOLO (*You Only Look Once*)

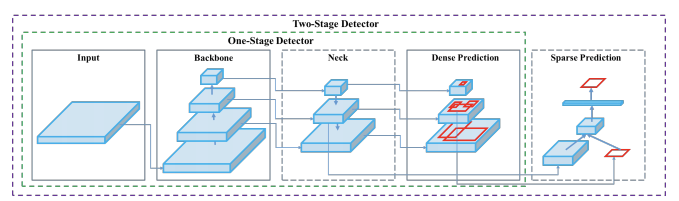
YOLO (*You Only Look Once*) pertama ditulis oleh Joseph Redom pada *custom framework* yang dinamakan Darknet. Cara kerja YOLO cukup sederhana, hanya sebuah Convolutional Neural Network yang akan memprediksi beberapa *bounding boxes* dan probabilitas *class labels* tiap *box* pada suatu *end-to-end differentiable network*. Darknet adalah *research framework* fleksibel yang ditulis dengan bahasa pemrograman tingkat rendah dan telah menghasilkan serangkaian deteksi objek *real-time* terbaik dalam *computer vision*: YOLO, YOLOv2, YOLOv3, dan YOLOv4[14].

YOLOv4 adalah model deteksi objek *real-time* yang dipublikasikan pada April 2020. YOLOv4 bekerja dengan memecah tugas deteksi objek menjadi dua bagian, regresi untuk mengidentifikasi posisi melalui *bounding boxes* dan klasifikasi untuk menentukan *object class*. YOLOv4 merupakan improvisai dari YOLOv3 yang dapat memperbaiki mAP (*mean Average Precision*) 10% dan meningkatkan angka FPS (*Frame per Second*) sebesar 12%.



Gambar 2. 2 Perbandingan YOLOv4 dengan YOLOv3 dan deteksi objek lain (sumber: YOLOv4: Optimal Speed and Accuracy of Object Detection)

YOLOv4 mengimprovisasi akurasi mAP dan kecepatan FPS dengan mengimplementasikan arsitektur baru yang terdiri dari CSPDarknet53 pada *Backbone* dan pada bagian Neck terdiri dari SPP dan PAN, sedangkan pada bagian Head YOLOv4 masih menggunakan YOLO head yang sama dengan YOLOv3.



Gambar 2. 3 Arsitektur deteksi objek

1. Tensorflow

Tensorflow adalah *library* perangkat lunak, yang dikembangkan oleh Tim Google Brain dalam organisasi penelitian Mesin Cerdas Google, untuk tujuan melakukan pembelajaran mesin dan penelitian jaringan syaraf dalam. Tensorflow kemudian menggabungkan aljabar komputasi teknik pengoptimalan kompilasi, mempermudah penghitungan banyak ekspresi matematis dimana masalahnya adalah waktu yang dibutuhkan untuk melakukan perhitungan[5].

1. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware, dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi sendiri.

Awalnya, Google Inc membeli Android Inc yang merupakan pendatang baru yang membuat software untuk ponsel. Kemudian untuk mengembangkan Android dibentuklah Open Handset Alliance, konsorsium dari tiga puluh empat perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Google mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Susunan lapisan tersebut merupakan arsitektur pembangun dari Aplikasi berbasis Android. Berikut gambar arsitektur sistem Android[15].

1. UML (*Unfied Modeling Language*)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML, perancang atau pemodel dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. UML lebih cocok diterapakan pada piranti berorientasi objek seperti C++, Java, C#, dan sebagainya. Tetapi UML juga tetap dapat digunakan untuk modeling aplikasi prosedural semisal VB atau C.

UML versi 2.0 mencakup 13 macam diagram dan perangkat yang berfungsi untuk menggambarkan sistem informasi berorientasi objek dengan sangat lengkap dan rinci. Meski demikan, tidak selalu ke-13 diagram dan perangkat tersebut digunakan saat para pengembang berupaya mengemabangkan perangkat lunak berorientasi objek. Beberapa diantara digaram UML yang digunakan adalah Use Case Diagram, Activity Diagram, Sequence Diagram, Class Diagram, dan Collaboration Diagram[16].

1. *Use Case Diagram*

*Use Case Diagram*, yaitu diagram yang digunakan untuk menggambarkan hubungan antara sistem dengan aktor. Diagram ini hanya menggambarkan secara global. Karena use case diagram hanya menggambarkan sistem secara global, maka elemen-elemen yang digunakan pun sangat sedikit[17].

*Use Case Diagram* memiliki beberapa komponen, yaitu:

* + - 1. *Actor*

Menggambarkan seseorang atau perangkat yang berinteraksi dengan sistem. *Actor* dapat memberi dan menerima informasi dari sistem, namun tidak memegang kendali pada *use case*.



Gambar 2. 4 Actor Use Case Diagram

* + - 1. *Use case*

Merupakan gambaran fungsionalitas suatu sistem yang berhubungan dengan *actor*, sehingga sistem yang akan dibagun lebih mudah dimengerti dan dipahami.

Gambar 2. 5 Use Case

* + - 1. *Relation*

*Relation* adalah komponen yang menghubungkan antara *actor* dengan *use case* atau sebalikya. Pada *Use Case Diagram* terdapat beberapa relasi yang digunakan, yaitu:

1. Association, interakasi antara komponen actor dengan use case yang digambarkan dengan garis antar komponen tersebut
2. Generalization, elemen yang menjadi spesialisasi dari elemen lain
3. Dependency, elemen yang bergantung pada elemen lain
   1. *Activity Diagram*

*Activity Diagram*, yaitu diagram yang digunakan untuk menggambarkan alur kerja (aktivitas) pada *use case* (proses), logika, proses bisnis dan hubungan antara aktor dengan alur-alur kerja *use-case.*[17]

Tabel 2 Simbol-simbol pada Activity Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| No | Gambar | Nama | Keterangan |
| 1 |  | *Activity* | Merupakan aktivitas atau pekerjaan yang dilakukan sistem pada alur kerja. |
| 2 |  | *Initial state* | Adalah simbol untuk mengawali suatu alur kerja pada *Activity Diagram.* |
| 3 |  | *Final State* | Adalah simbol untuk akhir suatu alur kerja pada *Activity Diagram.* |
| 4 |  | Decision | Berfungsi untuk menggambarkan pilihan suatu kondisi dan kemungkinan yang terjadi pada suatu alur kerja. |
| 5 |  | *Association* | Digunakan untuk menghubungkan suatu simbol atau komponen dengan komponen lainnya. |

* 1. *Class Diagram*

*Class Diagram* merupakan kumpulan dari beberapa *class* dan relasinya. *Class* identik dengan *entity* yang dipresentasikan dalam bentuk persegi dimana pada bagian atas dirulis nama *class*, kemudian kebawah ditulis *attributr* yang terdapat pada *class*, kemudian kebawah lagi ditulis *method-method* yang ada pada *class.*[17]

Tabel 3 Simbol-simbol pada Class Diagram

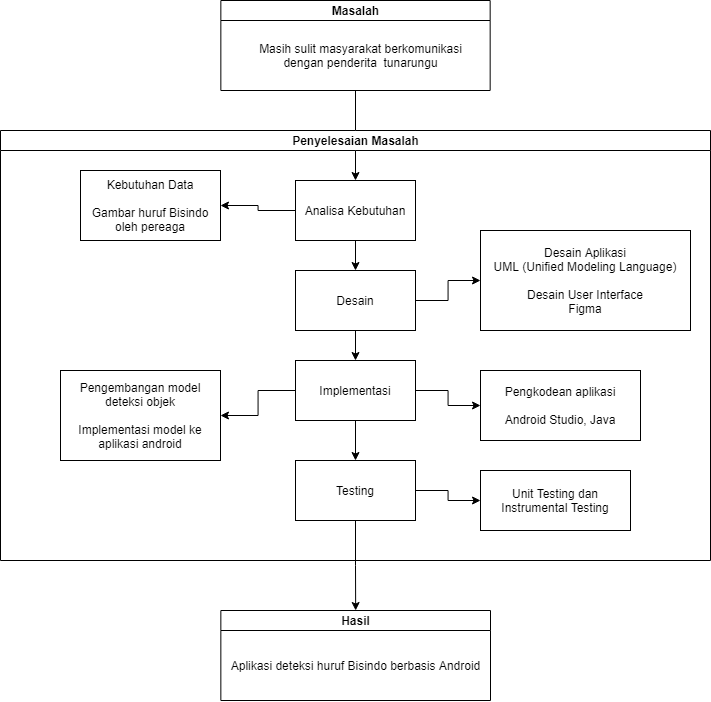
|  |  |  |  |
| --- | --- | --- | --- |
| No | Gambar | Nama | Keterangan |
| 1 |  | *Class* | Menjelaskan nama *class* dan *attributes* serta *operations.* |
| 2 |  | *Association* | Merupakan simbol relasi antar *class* yangbersifat umum. |
| 3 |  | *Dependency* | Merupakan simbol relasi antar *class* dengan makna kebergantungan antar *class.* |
| 4 |  | Aggregation | Merupakan simbol relasi antar *class* dengan makna semua bagian (*whole-part*)*.* |

Tabel 4 Relationship Multiplicity

|  |  |  |
| --- | --- | --- |
| No. | Nilai | Arti |
| 1 | 0..1 | Nol atau satu |
| 2 | 1 | Hanya Satu |
| 3 | 0..\* | Nola tau lebih |
| 4 | 1..\* | Satu atau lebih |
| 5 | n | Hanya n (dengan n > 1) |
| 6 | 0..n | Nol sampai n (dengan n > 1) |
| 7 | 1..n | Satu samapi n (dengan n > 1) |

1. Kerangka Berpikir

Kerangka berpikir merupakan model konseptual tentang bagaimana teori berhubungan dengan berbagai faktor yang telah diidentifikasi sebagai masalah yang penting. Berikut adalah kerangka berpikir penelitian ini disajikan dalam gambar:



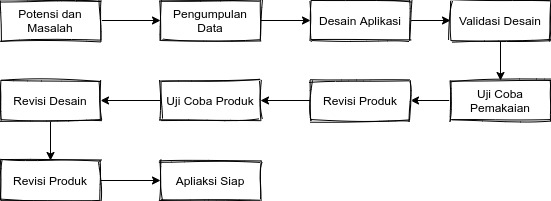
Gambar 2. 6 Kerangka Berpikir

BAB III

metode penelitian

1. Pendekatan Penelitian

Pada penelitian ini, metode yang digunakan adalah metode R&D (*Research and Development*). Metode *Research and Development* atau Penelitian dan Pengembangan ini adalah metode yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut. Untuk dapat menghasilkan produk tertentu digunakan penelitian yang bersifat analisis kebutuhan dan untuk menguji keefektifan produk tersebut supaya dapat berfungsi di masyarakat luas, maka diperlukan penelitian untuk menguji keefektifan produk tersebut (digunakan metode eksperimen)[18].



Gambar 3. 1 Langkah-langkah metode R&D

1. Populasi dan Sampel

Populasi dari penelitian ini adalah gambar huruf Bisindo yang ada. Sedangkan sampel pada penelitian ini adalah gambar huruf Bisindo sebanyak 1016 gambar.

1. Variabel Penelitian

Variabel yang digunakan dalam penelitian ini ditampilkan pada Tabel 5 berikut:

Tabel 5 Variabel Huruf Bisindo

|  |  |  |  |
| --- | --- | --- | --- |
| Isyarat | Huruf | Isyarat | Huruf |
| A | A | O | O |
| B | B | P | P |
| C | C | Q | Q |
| D | D | R | R |
| E | E | S | S |
| F | F | T | T |
| G | G | U | U |
| H | H | V | V |
| I | I | W | W |
| K | K | X | X |
| L | L | Z | Z |
| M | M |  |  |
| N | N |  |  |

1. Teknik Pengumpulan Data

Pada tahap ini peneliti melakukan pengumpulan data yang berkaitan dan akan digunakan dalam penelitian ini. Data yang dikumpulkan berupa foto peragaan huruf alphabet Bisindo yang diperagakan oleh tiga model. Dari foto tersebut kemudian diberi anotasi pada masing-masing gambar sesuai *class* hutuf Bisindo dengan menggunakan LabelImg sehingga data dapat digunakan pada *framework* Darknet.

1. Pengujian Aplikasi

Pengujian aplkasi dilakukan untuk mengetakui apakah aplikasi berjalan sesuai dengan apa yang diharapkan. Pada penelitian ini dilakukan melalui dua tahap pengujian.

1. Pengujian Akurasi Model

Pengujian model dilakukan untuk mengetahui tingkat akurasi mAP (*mean Average Precision*) dan kecepatan FPS (*Frame per Second*) pada model deteksi objek yang telah dibuat.

1. *Unit Testing*

Pengujian ini dilakukan untuk menguji validasi unit kode secara individual sehingga aplikasi android berjalan sesuai harapan.

1. *Instrumental Tesing*

Pengujian ini digunakan untuk mengetahui bagaimana *user interface* atau tampilan antar muka pada aplikasi android apakah sudah berjalan dengan baik atau belum.

BAB IV

**HASIH PENELITIAN DAN PEMBAHASAN**

1. Hasil Tahapan Permulaan

Tahap ini menjelaskan lebih detail mengenai sistem yang akan dikembangkan. Pengembangan aplikasi deteksi huruf Bisindo ini menggunakan motode Waterfall yang terdiri dari beberapa tahapan yaitu, analisis kebutuhan sistem dan kebutuhan data, desain atau perancangan aplikasi menggunakan UML (Unified Modeling Language) serta desain *user interface* menggunakan *software* Figma, kemudian tahap implementasi desain ke penulisan kode menggunakan bahasa pemrograman Java dan implementasi model deteksi huruf Bisindoke aplikasi android, dan tahap terakhir yaitu melakukan testing pada aplikasi android menggunakan Unit Testing dan Instrumental testing.

1. **Analisis Sistem**
2. Kebutuhan Sistem

Kebutuhan sistem meliputi *tools* atau *software* yang digunakan dalam mengembangkan aplikasi deteksi Bisindo. Berikut merupakan *tools* yang dibutuhkan:

1. LabelImg

LabelImg digunakan untuk memberikan label pada setiap dataset yang akan digunakan.

1. Google Colaboratory

Google Colaboraty digunakan untuk mengeksekusi Python pada Google cloud server yang berjalan pada Google hardware untuk memproses *dataset*, *training dataset*, konversi model ke Tensorflow *lite*.

1. Draw.io

Draw.io digunakan untuk merancang UML seperti *use case diagram*, *activity diagram* dan *class diagram*.

1. Figma

Figma digunakan untuk mendesain setiap halaman antarmuka aplikasi android.

1. Android Studio

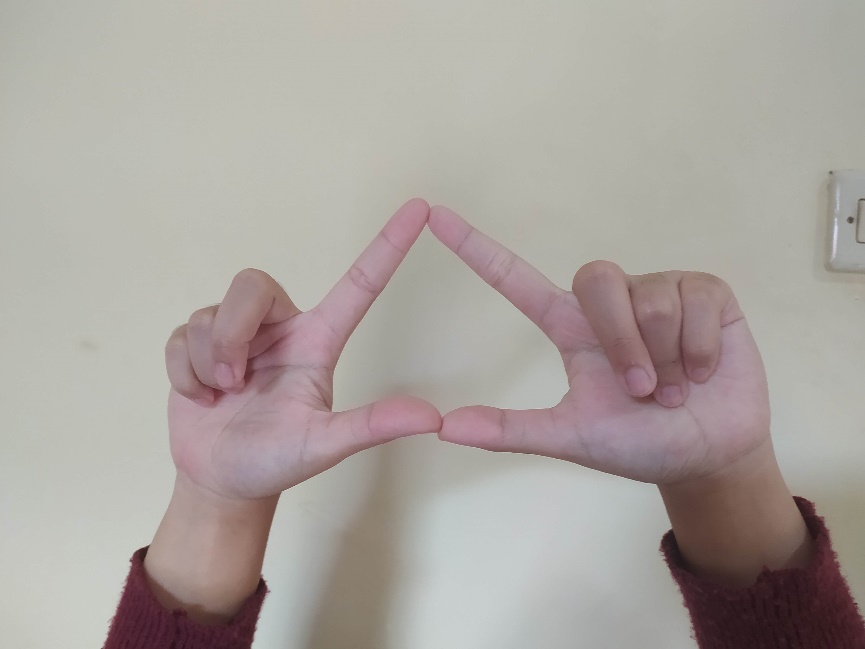
Android Studio digunakan dalam proses implementasi dari desain menjadi aplikasi android yang fungsional dengan menggunakan bahasa pemrograman Java maupun Kotlin.

1. Kebutuhan Data

Kebutuhan data yang diperlukan dalam pengembangan aplikasi deteksi huruf Bisindo adalah data gambar huruf Bisindo yang akan diolah terlebih dahulu melalui beberapa tahapan yaitu pengumpulan data, pelabelan data dan *training* data sehingga data tersebut dapat digunakan dalam proses deteksi huruf Bisindo.

1. Pengumpulan Data

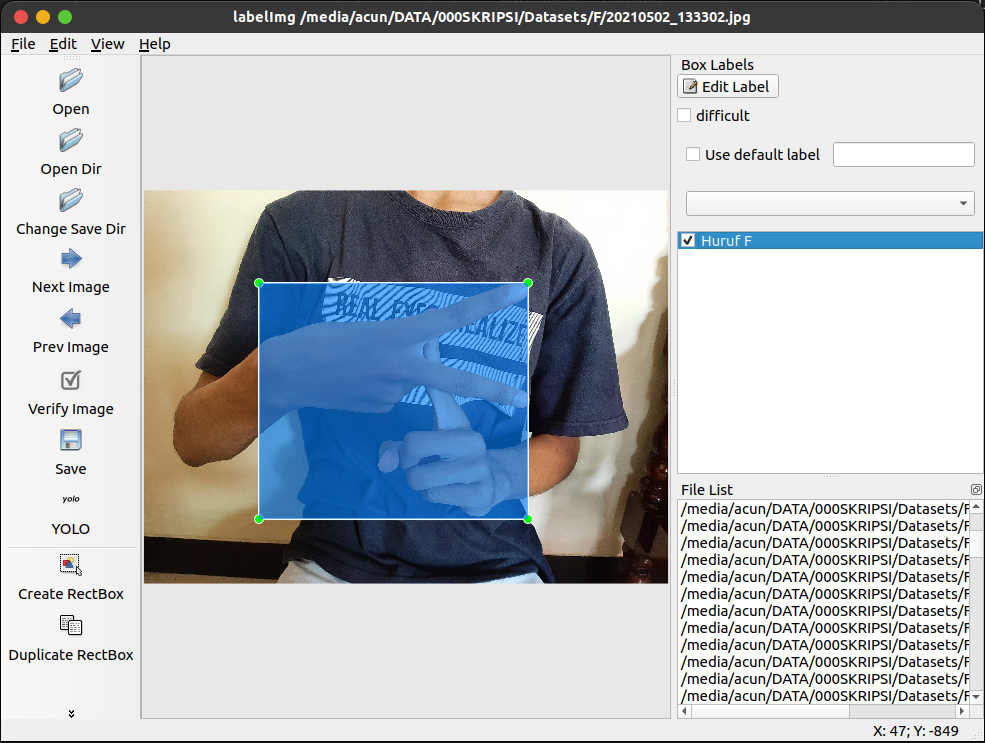
Data yang digunakan dalam pembuatan model menggunakan data gambar yang diambil dari foto peragaan huruf Bisindo. Peragaan huruf bisindo dilakukan oleh tiga peraga yang masing-masing memperagakan setiap huruf bisindo dari huruf A-Z terkecuali huruf J, Y dan R, karena dalam peragaan huruf tersebut terdapat gerakan yang akan sulit dideteksi dengan model deteksi objek. Dari tiga peraga tersebut didapatkan data gambar huruf bisindo sebanyak 1016 gambar.



Gambar 4. 1 Contoh data gambar peragaan isyarat

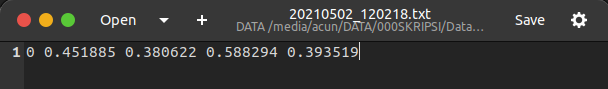
1. Pelabelan Data

Pelabelan adalah tahap dimana *dataset* diberikan label atau anotasi sebagai pengenal objek pada setiap gambar sesuai dengan *class* yang sudahdidefinisikan sebelumnnya yaitu huruf Bisindo. Proses pelabelan dilakukan secara manual pada 1016 dataset huruf bisindo dengan menggunakan LabelImg.



Gambar 4. 2 Contoh proses pelabelan menggunakan labelImg

Anotasi yang dihasilkan kemudian disimpan dalam berkas .txt sesuai dengan format yang dibutukan untuk training pada YOLOv4. Pada format pelabelan YOLO, sebuah file .txt dengan nama yang sama dengan gambar akan dibuat pada folder yang sama. Setiap file .txt menyimpan informasi gambar terkait, yaitu *object* *class, object coordinates, height* dan *weight.*

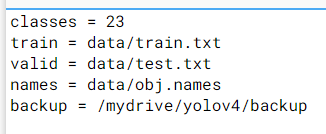


Gambar 4. 3 Contoh file hasil pelabelan gambar

1. Training Data

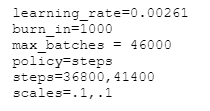
Proses *training* dilakukan untuk menghasilkan model yang akan diimplementasikan pada sistem untuk mendeteksi objek. Proses *training* memakan waktu yang cenderung cukup lama tergantung dari besarnya *dataset* yang digunakan dan spesifikasi *hardware*. Oleh karena itu, pada proses ini akan menggunakan teknik *transfer learning* untukmempercepat proses *training*. Proses *training* menggunakan *pre-weight* dari repository yolov4-tiny.weights.

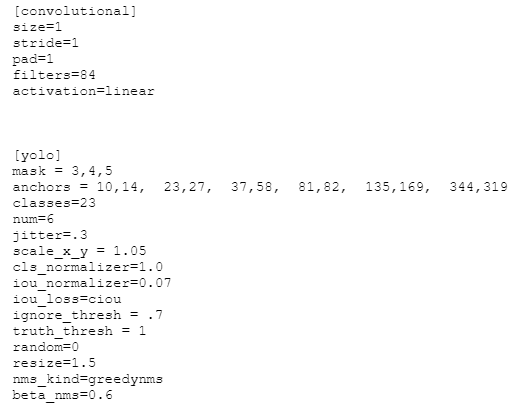
Sebelum melakukan proses *training* menggunakan Darknet, perlu dilakukan konfigurasi pada beberapa file yaitu yolov4-tiny-obj.cfg, obj.data dan obj.name sehingga sesuai dengan kebutuhan untuk menghasilkan model baru. File tersebut berisi konfigurasi *neural network*, dan informasi lokasi dari data gambar yang digunakan.



Gambar 4. 4 Konfigurasi file obj.data

Pada gambar 4.4 merupakan file obj.data yang berisi informasi jumlah *class* lokasi fileyang dibutuhkan dan lokasi untuk menyimpan backup file .*weight* yang dihasilkan selama proses *training.*





Gambar 4. 5 Konfigurasi convolutional dan yolo

1. **Desain Sistem**

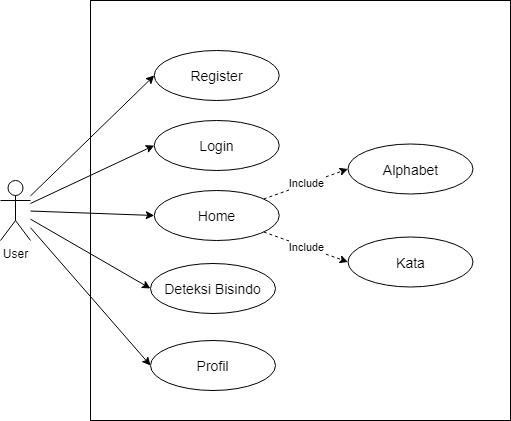
Tahap desain sistem merupakan perancangan suatu kesatuan yang terdiri dari komponen untuk mempermudah aliran informasi dalam mencapai suatu tujuan yang berkaitan dengan aplikasi. Tahap desain sistem dilakukan melalui dua tahap yaitu desain UML dan desain antarmuka atau *user interface.*

1. UML (*Unfied Modeling Language*)

Perancangan UML (*Unfied Modeling Language*) merupakan penyampaian model sistem yang mendukung paradigma *object oriented*. UML juga dapat digunakan sebagai *blue print* suatu sitem. Mempertimbangkan banyaknya diagram pada UML, maka pada tahap ini hanya menggunakan beberapa jenis diagram UML sederhana yaitu *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*.

1. *Use Case Diagram*

Use Case Diagram mempresentasikan initeraksi antar sistem serta kebutuhan fungsional dari sistem sendiri. Kebutuhan dalam sistem terdiri dari satu *user* dan beberapa fungsi atau akses yang bisa dilakukan *user*.

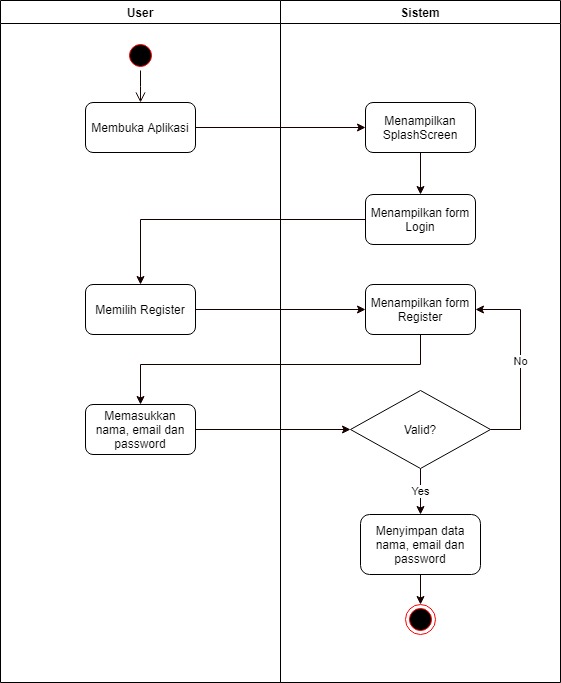


Gambar 4. 6 Use Case Diagram Aplikasi Belajar Bisindo

1. *Activity Diagram*

*Activity Diagram* pada dasarnya adalah diagram alir yang menggambarkan alur dari masing-masing *activity* ke *activity* lainnya. *Activity Diagram* dibuat sesuai dengan *Use Case* yang telah dibuat yaitu *Register*, *Login*, Deteksi Bisindo, dan Profil.

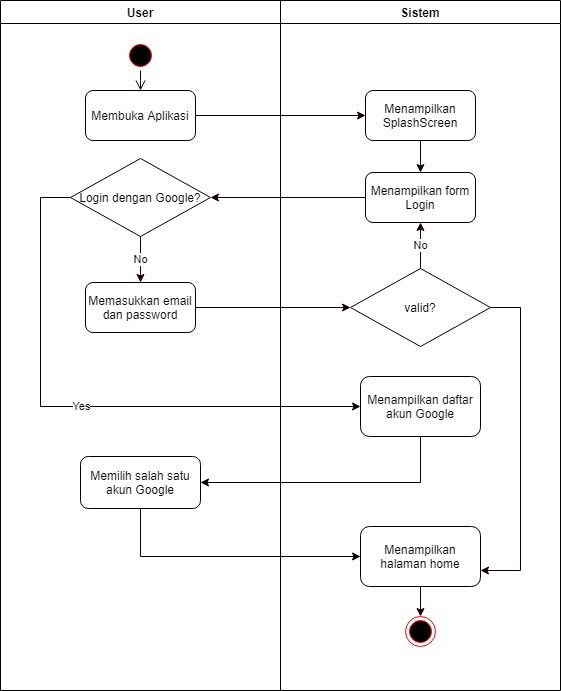
1. *Activity Diagram Register*



Gambar 4. 7 Diagram Activity Register

Gambar 4.2 menunjukkan alur bagaimana user melakukan registrasi akun baru. Dalam melakukan registrasi, user membuka aplikasi akan menampilkan *splashscreen.* Setelah *splashscreen* tampil akan beralih ke halaman *login* kemudian user memilih *register.* Pada halaman *register* ini *user* dapat menginputkan nama, email dan *password*.

1. *Activity Diagram Login*

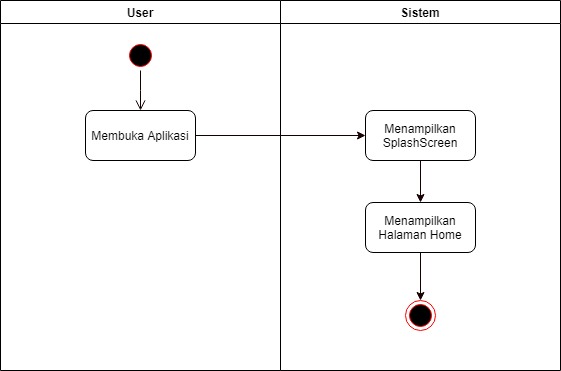


Gambar 4. 8 Activity Diagram Login

Gambar 4.3 merupakan gambaran diagram aktivitas login pada aplikasi android. Saat user membuka aplikasi sistem akan menampilkan *splashscreen* kemudian menampilkan halaman *login*. Di halaman *login,* user dapat memilih untuk melakukan *login* menggunakan Google atau menggunakan email dan *password*. Jika *user* memilik menggunakan Google maka user memilih tombol *login* *with* Google dan akan muncul daftar akun Google yang dapat dipilih. Namun, jika *user* memilih untuk melakukan *login* dengan email, *user* mengisi email dan *password* pada *form* yang ada di halaman *login*. Sistem akan memverifikasi kemudian mengalihkan ke halaman utama atau *home*.

1. *Activity Diagram Home*

Pada halaman Home terdapat menu alphabet dan kata yaitu menu untuk melihat alphabet atau kata dalam bahasa isyarat.

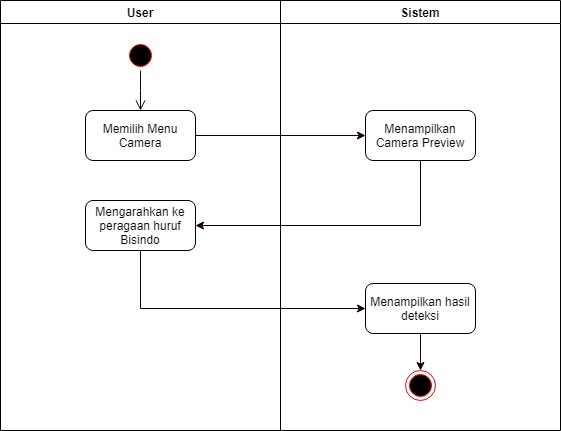


Gambar 4. 9 Activity Diagram Home

Gambar 4.9 merupakan alur bagaimana user membuka aplikasi yang kemudian aplikasi akan menampilkan splashscreen beberapa saat lalu barulah halaman Home akan tampil.

1. *Activity Diagram* Deteksi Bisindo

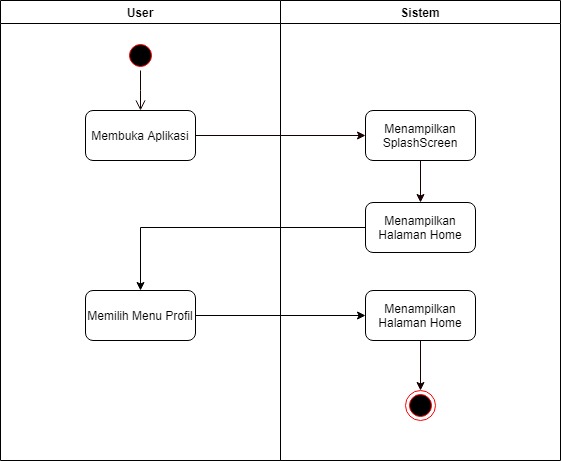
Halaman deteksi Bisindo menampilkan *camera preview* dan beberapa informasi pada bagian bawah atau sering disebut *bottom sheet.* Pada halaman inilah aplikasi dapat mendeteksi huruf apa yang diperagakan beserta presentase keakuratan hasil deteksi tersebut.



Gambar 4. 10 Activity Diagram Deteksi Bisindo

Gambar 4.10 menunjukkan *user* memilih menu camera, kemudian aplikasi menampilkan halaman deteksi yang terdiri dari *camera preview* yang dapat diarahkan ke huruf isyarat atau peragaan huruf bisindo. Setelah kamera diarahkan ke huruf isyarat, aplikasi akan mendeteksi huruf apa yang diperagakan dan berapa akurasi deteksi objek terebut.

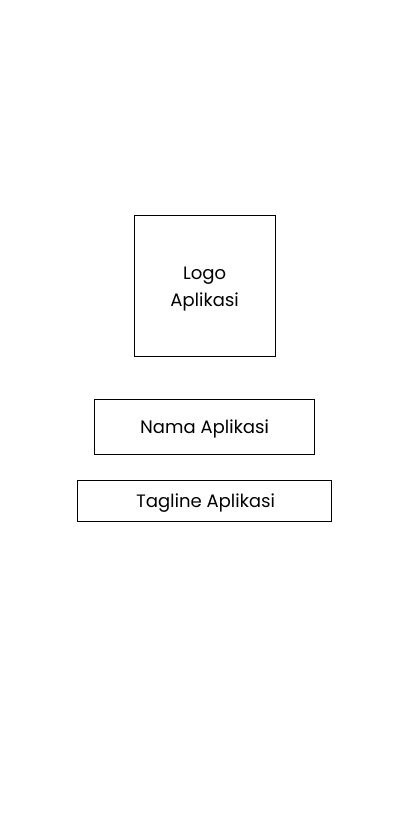
1. *Activity* *Diagram* Profil



Gambar 4. 11 Activity Diagram Profile

Pada gambar 4.11 menggambarkan user membuka aplikasi, kemudian sistem menampilkan *splashscreen* selama beberapa detik dan langsung *redirect* ke halaman home. Pada halaman awal user memilih menu profil di bagian *bottom navigation*, lalu sistem akan menampilkan halaman profil

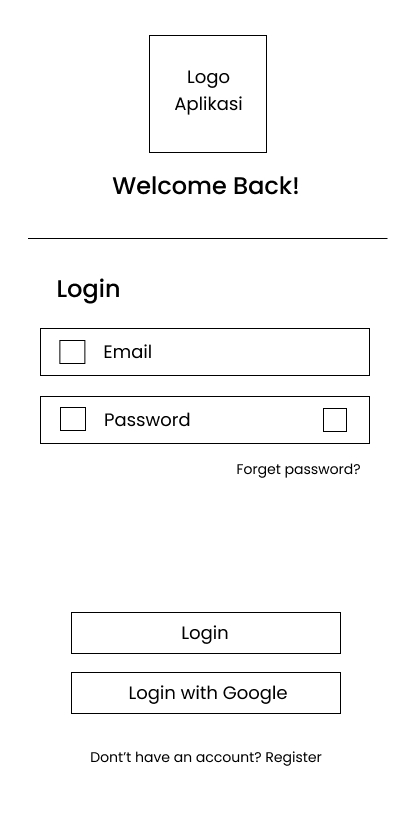
1. *Class Diagram*
2. Desain Antarmuka
3. Desain Tampilan *SplashScreen*



Gambar 4. 12 Desain Tampilan SplashScreen

Gambar 4.6 menunjukkan tampilan *splashscreen* dan halaman utama aplikasi. Tampilan *splashscreen* muncul 2 detik di awal aplikasi dibuka.

1. DesainTampilan *Login*

**

Gambar 4. 13 Desain Tampilan Login

Gambar 4.7 menunjukan tampilan *login* dan *register*. Jika belum melakukan *login*, halaman *login* akan muncul setelah tampilan splashscreen. Pada halaman *login* terdapat *form* untuk mengisi data *email* serta *password* dan terdapat tombol untuk melakukan aksi *login*. Jika pengguna belum memiliki akun, pada halaman terdapat tombol untuk menuju halaman *register*. Untuk mempermudah user, terdapat pilihan untuk *login* denganmenggunakan akun Google. Halaman *register* menampilan *form* yang digunakan untuk mengisi nama, email dan *password.*

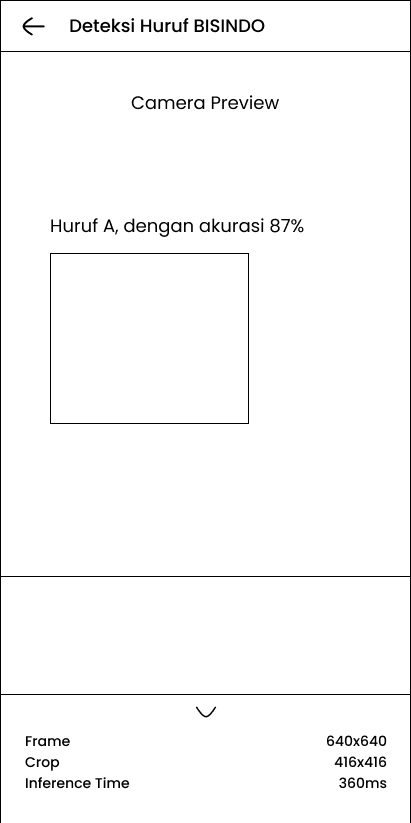
1. Desain Tampilan *Home*

**

Gambar 4. 14 Desain Tampilan Home

Jika telah melakukan login, tampilan menu utama seperti pada gambar 4.8 akan tampil setelah tampilan splashscreen. Pada halaman menu utama terdapat daftar artikel berita dan *bottom navigation menu* yang terdiri dari *home, camera* dan *profile.*

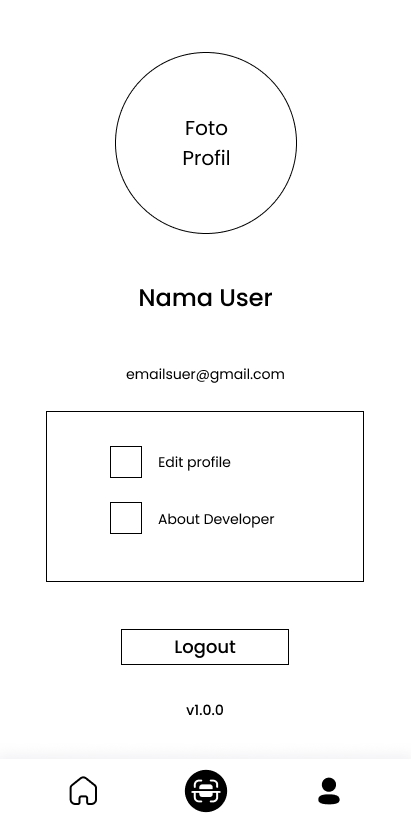
1. Desain Tampilan Deteksi



Gambar 4. 15 Desain Tampilan Deteksi

Pada gambar 4.9 menunjukan halaman untuk detaksi huruf isyarat. Di halaman ini terdapat *camera preview* untukmendeteksi huruf isyarat secara *real time.* Pada *camera preview*, jika berhasil mendeteksi huruf maka terdapat persegi berwarna biru, merah ataupun hijau beserta informasi huruf apa yang telah terdeteksi dan persentase akurasi. Di bagian bawah terdapat informasi *frame, crop* dan *inference time.*

1. Desain Tampilan Profil



Gambar 4. 16 Desain Tampilan Profil

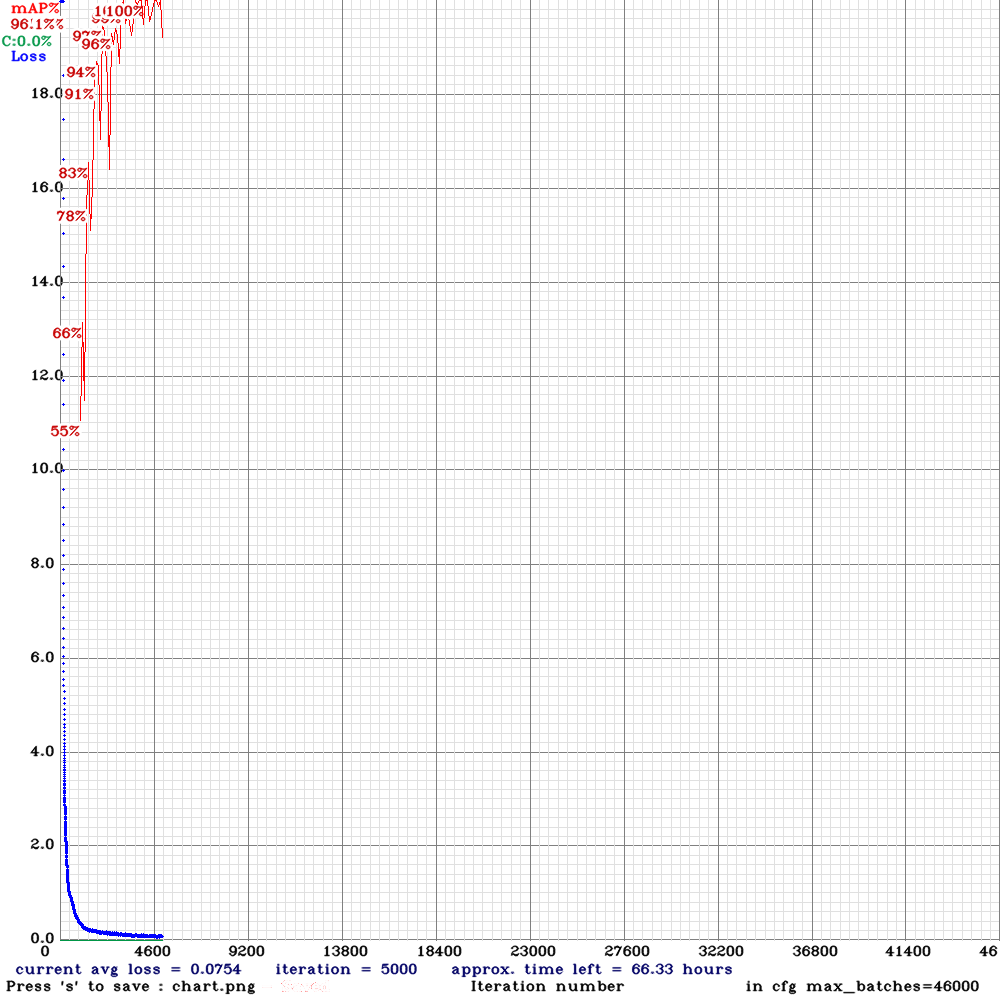
Gambar 4.10 adalah tampilan profil di mana terdapat informasi mengenai pengguna aplikasi yang telah *login*, informasi aplikasi, *developer* dan terdapat tombol untuk *logout*.

1. Hasil Sistem Machine Learning

Hasil sistem machine learning adalah hasil dari proses pelatihan dataset.

1. ***Loss curve***

*Loss curve* adalah grafik yang memvisualisasikan *loss* dan fungsi dari *training iterations*. *Loss* merupakan ukuran seberapa jauh prediksi model dari labelnya atau bisa disebut sebagai ukuran seberapa buruk model. Sedangkan *iteration* adalah *update* dari setiap *model’s weight* selama proses *training.*

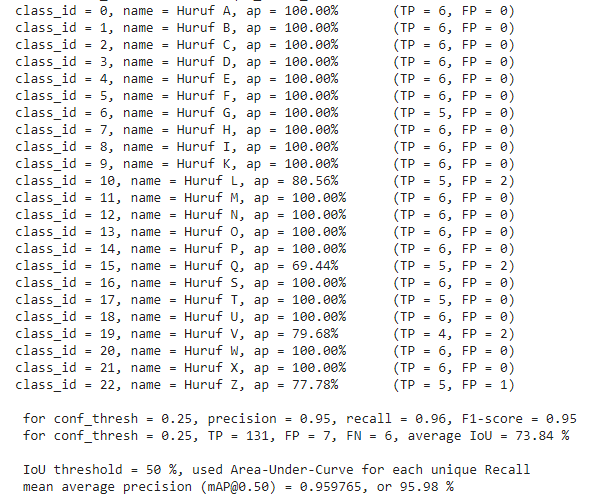


Gambar 4. 17 *Loss curve* hasil *training* model

Berdasarkan grafik pada Gambar 4.17, dapat diketahui bahwa hasil pelatihan awal menunjukkan tingkat akurasi yang rendah diketahui dari angka *average* *loss* yang tinggi. Kemudian setelah *iteration number* menunjukkan angka kisaran 4000 dan seterusnya, angka *average loss* menunjukkan angka yang rendah dan grafik mulai konstan.

1. ***Mean Average Precision* (mAP)**

*Mean Average Precision* adalah sebuah matrik untuk meringkas performa dari hasil *training*. *Average Precision* didapatkan dari perhitungan rata-rata dari setiap nilai *precision* masing-masing hasil yang relevan.



Gambar 4. 18 *Mean Average Precision* (mAP)

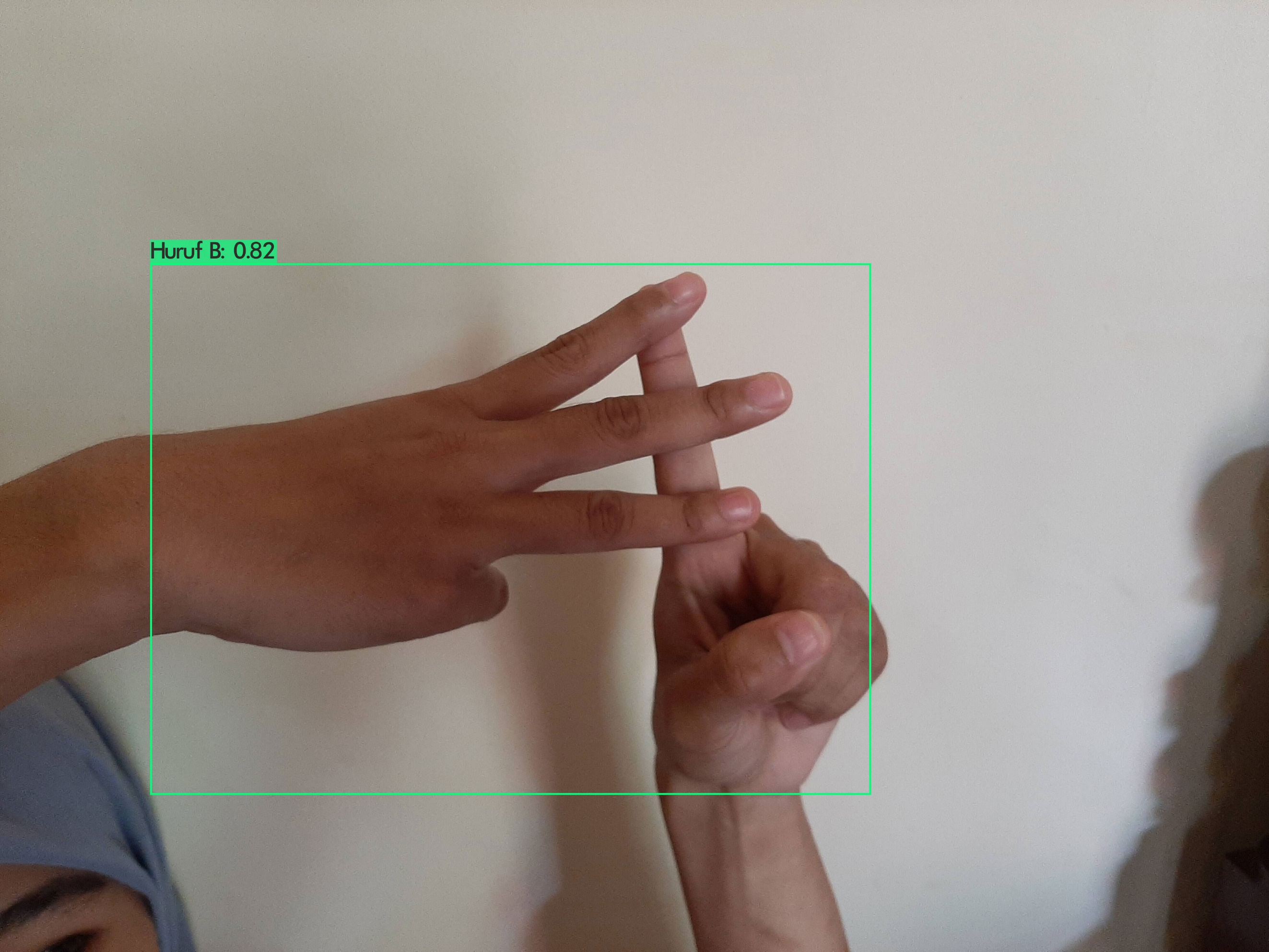
Pada Gambar 4. 18 menampilkan *Average Precision* dari masing-masing *class* yang diperoleh dari rata-rata seluruh dataset yang relevan beserta informasi tambahan yang berada di sebelah kanan berupa TP atau *True Positive* dan FP atau *False Positive*.

TP yaitudimana modelmemprediksi kelas positif dengan benar. Misal, model mendeteksi huruf A dan gambar tersebut memang benar-benar huruf A. Sedangkan FP yaitu dimana model secara keliru memprediksi kelas positif. Contoh, model menyimpulkan gambar yang terdeteksi adalah kelas A tetapi sebenarnya gambar tersebut bukan A.

Dapat dilihat pada Gambar 4. 18 diperoleh *mean average precision* hasil perhitungan *mean* dari AP tiap-tiap kelas dengan hasil 0.959765 atau 95.98 persen.

1. **Hasil Deteksi**

Hasil uji coba dari sebuah model menggunakan salah satu gambar yang menampilkan hasil deteksi berupa gambar yang ditandai kotak berwarna hijau dengan informasi nama *class* pada pojok kiri atas berikut dengan angka akurasi. Pada gambar 4. 19 berikut ini adalah hasil dari uji coba deteksi pada file 20210502\_121351.jpg



Gambar 4. 19 Hasil deteksi pada file 20210502\_121351.jpg



Gambar 4. 20Informasi hasil deteksi file20210502\_121351.jpg

Pada gambar 4. 20 dapat dilihat informasi mengenai hasil uji deteksi pada file 20210502\_121351.jpg di mana model berhasil mendeteksi huruf B dengan akurasi 82% dan dengan kecepatan deteksi 5.086000 *milli-seconds.*

1. Hasil Sistem Aplikasi Android

Daftar pustaka

[1] G. Gumelar, H. Hafiar, and P. Subekti, “Bahasa Isyarat Indonesia Sebagai Budaya Tuli Melalui,” *Inf. Kaji. Ilmu Komun.*, 2018.

[2] M. Bagus, S. Bakti, and Y. M. Pranoto, “Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network,” *Semin. Nas. Inov. Teknol.*, 2019.

[3] W. Dwiatmoko, P. S. Informatika, F. Komunikasi, D. A. N. Informatika, and U. M. Surakarta, “Perancangan sistem pengenalan jenis tanaman obat dengan kamera berbasis android,” 2020.

[4] C. N. Liunanda, S. Rostianingsih, and A. N. Purbowo, “Implementasi Algoritma YOLO pada Aplikasi Pendeteksi Senjata Tajam di Android,” pp. 1–7.

[5] I.-135610103 Taufiq, “DEEP LEARNING UNTUK DETEKSI TANDA NOMOR KENDARAAN BERMOTOR MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK DENGAN PYTHON DAN TENSORFLOW,” Feb. 2018.

[6] S. R. Dewi, “Deep learning object detection pada video menggunakan Tensorflow dan Convolutional Neural Network,” *Deep Learn. Object Detect. Pada Video Menggunakan Tensorflow Dan Convolutional Neural Netw.*, 2018.

[7] F. Al-Azzo, A. M. Taqi, and M. Milanova, “Human related-health actions detection using Android Camera based on TensorFlow Object Detection API,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 10, pp. 9–23, 2018, doi: 10.14569/IJACSA.2018.091002.

[8] A. Mardiyani, M. Heri P., and I. K. Edy P., “Pengenalan Bahasa Isyarat Menggunakan Metode PCA dan Haar Like Features,” *Fti-Its*, 2011.

[9] A. Ahmad, “Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning,” 2017, Accessed: Jan. 08, 2021. [Online]. Available: www.teknoindonesia.com.

[10] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*. 2015, doi: 10.1038/nature14539.

[11] “Pengenalan Deep Learning Part 8 : Gender Classification using Pre-Trained Network (Transfer Learning) | by Samuel Sena | Medium.” https://medium.com/@samuelsena/pengenalan-deep-learning-part-8-gender-classification-using-pre-trained-network-transfer-37ac910500d1 (accessed Jan. 11, 2021).

[12] O. Alsing, “Mobile Object Detection using TensorFlow Lite and Transfer Learning,” 2018.

[13] A. Ouaknine, “Review of Deep Learning Algorithms for Object Detection | by Arthur Ouaknine | Zyl Story | Medium.” https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852 (accessed Jan. 08, 2021).

[14] J. Solawetz, “Breaking Down YOLOv4.” https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/ (accessed Jun. 24, 2021).

[15] Nazruddin Safaat H, *Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android (Edisi Revisi)*. 2012.

[16] S. Dharwiyanti and R. S. Wahono, “Pengantar Unified Modeling LAnguage (UML),” *IlmuKomputer.com*, 2003.

[17] C. Prof. Dr. Sri Mulyani, Ak., “Analisis dan Perancangan Sistem Informasi Manajemen Keuangan Daerah: Notasi Permodelan Unifed Modeling Language,” *Metode Analisis Dan Perancangan Sistem*. 2016.

[18] S. Haryati, “Research and Development (R&D) Sebagai Salah Satu Model Penelitian dalam Bidang Pendidikan,” *Res. Dev. Sebagai Salah Satu Model Penelit. Dalam Bid. Pendidik.*, 2012.