



## PENGENALAN GERAKAN SIKAP DASAR PENCAK SILAT BAKTI NEGARA BERBASIS APLIKASI *MOBILE* MENGGUNAKAN *NEURAL NETWORK*

I P Julian Taruna S<sup>1</sup>, K Queena Fredlina<sup>2</sup>, I B Kresna Sudiatmika<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika, STMIK Primakara  
Denpasar, Bali, Indonesia 80226

juliantarunasuantara@gmail.com, naa.queena@gmail.com, kresna@primakara.ac.id

### Abstract

*Research on introducing basic stance or poses recognition specializing in sports is scarce. Researchers have not yet found journals or articles that discuss the introduction of the fundamental view of Pencak Silat as in this research. The research results can be used by athletes, teaching staff or organizations, and even the government who want to take advantage of the results of this research in the form of an application to introduce the fundamental stance of Pencak Silat. To realize this application, researchers use Neural Network or artificial intelligence where the model of recognition is trained on the Teachable Machine, which produces recognition accuracy above 90%, then integrated with mobile application programs based on Android that have been built using Flutter. After being integrated with the mobile application, This application recognition still produces accuracy above 90%, so it can be concluded this application can work very well.*

**Keywords:** Artificial Intelligence, Mobile Apps, Neural Network, Pencak Silat, Poses Recognition

### Abstrak

Penelitian pengenalan sikap dasar atau poses recognition yang mengkhusus ke bidang olahraga sangatlah jarang ada dan peneliti belum menemukan jurnal atau artikel yang membahas mengenai pengenalan sikap dasar dari Pencak Silat seperti pada penelitian ini. Hasil penelitian ini dapat dipergunakan oleh atlet, tenaga pengajar ataupun organisasi bahkan juga pemerintah yang ingin memanfaatkan hasil penelitian ini yaitu, berupa aplikasi pengenalan sikap dasar dari Pencak Silat. Untuk mewujudkan aplikasi ini, peneliti menggunakan *Neural Network* atau kecerdasan buatan yang mana model dari pengenalan di *train* pada *Teachable Machine* yang menghasilkan akurasi pengenalan di atas 90%, kemudian diintegrasikan dengan program aplikasi *mobile* berbasis *Android* yang dibangun menggunakan *Flutter*, serta setelah diintegrasikan dengan aplikasi *mobile*, aplikasi pengenalan ini masih menghasilkan akurasi di atas 90%, sehingga dapat disimpulkan aplikasi ini dapat bekerja dengan sangat baik.

**Kata kunci:** Aplikasi *Mobile*, Kecerdasan Buatan, *Neural Network*, Pengenalan Gerakan, Pencak Silat

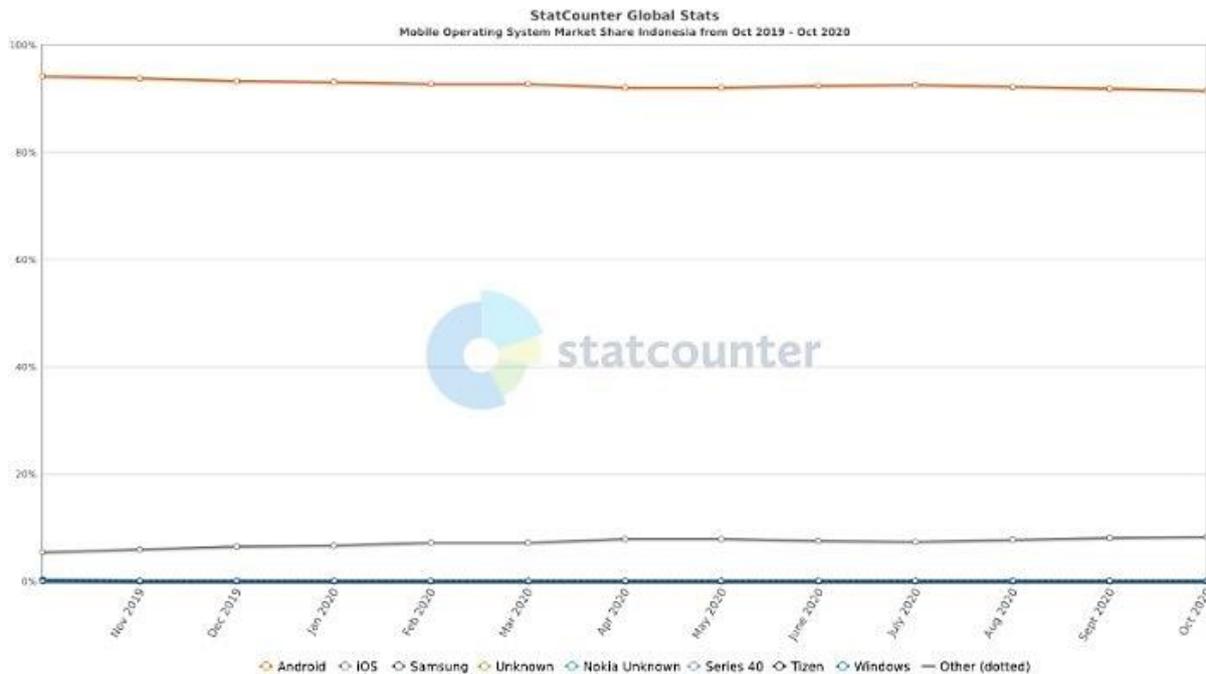
### 1. PENDAHULUAN

Berbagai macam teknologi dikembangkan mulai dari teknologi *robotic* sampai *Artificial Intelligence (AI)*. Inti di dalam proses kerja *Artificial Intelligence* yang akan dibahas dalam penelitian ini sendiri adalah *learning*, *reasoning* dan *self correction*. Pada penelitian ini memanfaatkan *AI* jenis *poses recognition*, yang dapat mengenali gerakan dari *object* yang bergerak secara *real-time*. *AI* jenis ini digunakan karena, studi kasus pada penelitian ini adalah bagaimana mengenali gerakan sikap dasar dari Pencak Silat Bakti Negara (PSBN) yang memiliki banyak sikap dan gerakan yang berbeda dengan pola gerakan masing-masing.

Penelitian ini diharapkan menghasilkan *output* aplikasi *mobile* yang dapat mengenali gerakan dari pencak silat.

Model pengenalan dari aplikasi ini menggunakan teknologi *Teachable Machine* yang dikembangkan oleh Google yang sudah menggunakan algoritma *RNN (Recurrent Neural Network)* didalamnya dan akan diintegrasikan dengan aplikasi *mobile* berbasis *Android* yang dibangun menggunakan *Flutter*.

Menurut Stat Counter Global Stats, dari Oktober 2019 – Oktober 2020 pengguna dari *Android* yang lebih dominan di Indonesia daripada *iOS* dan lainnya, dengan persentasi 90% ke atas pengguna *Android* meninggalkan pengguna *iOS* dan lainnya, pada *Mobile Operating System Market Share in Indonesia - November 2020* yang dapat dilihat pada data gambar statistik di bawah ini [1].



Gambar 1. Data Sistem Operasi – November 2020 [1]

Data pada Gambar 1 di atas menjadi acuan aplikasi ini dikembangkan ke arah aplikasi Android, melihat dari jumlah pengguna di Indonesia yang mendominasi ke arah Android *user*. Sehingga diharapkan nantinya, aplikasi ini akan memberikan dampak yang lebih masif.

Adapun masalah yang akan dikaji dalam penelitian ini berdasarkan latar belakang di atas yaitu, bagaimana cara mengintegrasikan model dari hasil pelatihan *Teachable Machine* terhadap pemrograman *Flutter*, sehingga menghasilkan aplikasi berbasis *mobile*.

Berdasarkan latar belakang dan perumusan masalah di atas, penelitian ini bertujuan untuk membantu pembelajaran dan atau proses latihan dari perguruan Pencak Silat Bakti Negara dengan memanfaatkan *AI (Artificial Intelligence)* berbasis Android.

Keterbatasan peneliti dalam penelitian serta untuk membuat pembahasan lebih terarah menyebabkan batasan masalah perlu penulis deskripsikan yaitu, penelitian ini menggunakan *Neural Network* dan hasil penelitian berupa aplikasi *mobile* bukan *website*, sikap yang bisa dikenali aplikasi hanya ada lima yang telah divalidasi serta studi kasus dilakukan di perguruan Pencak Silat Bakti Negara, versi Android yang dapat menggunakan aplikasi mulai dari versi Android 8.1 (Oreo) ke atas dan model pelatihan didapatkan dari hasil pelatihan di *Teachable Machine* dengan *datasets* sendiri serta *maintenance* dari aplikasi dilakukan setiap 6 bulan atau apabila diperlukan sewaktu-waktu.

Penelitian ini juga melihat dari penelitian-penelitian sebelumnya, agar dapat menentukan dan melihat perbedaan penelitian ini dengan penelitian terdahulu, seperti Syarifah Rosita Dewi yang melakukan penelitian dengan judul “*Deep Learning Object Detection* pada Video Menggunakan *Tensorflow* dan *Convolutional Neural Network*” menggunakan metode analisa data mendapatkan hasil bahwa klasifikasi meja dan kursi pada suatu citra digital menggunakan *convolutional neural network* dapat dinilai bekerja dengan baik [2].

Habib Astari Adi dan Ika Candradewi yang melakukan penelitian dengan judul “Sistem Pengenalan Isyarat Tangan untuk Mengendalikan Gerakan Robot Beroda Menggunakan *Convolutional Neural Network*” menggunakan metode *Contrast Stretching* mendapatkan hasil bahwa sistem pengenalan isyarat tangan menggunakan metode tersebut menghasilkan akurasi sebesar 97,5%, presisi 97,57%, sensitivitas 97,5%, spesivitas 99,77% dan f1 score 97,45% [3].

Mochamad Bagus Setiyo Bakti dan Yuliana Melita Pranoto yang melakukan penelitian dengan judul “Pengenalan Angka Sistem Syarat Bahasa Indonesia dengan Menggunakan Metode *Convolutional Neural Network*” menggunakan metode *Convolutional Neural Network* mendapatkan hasil bahwa dengan menggunakan arsitektur LeNet tingkat akurasi *training* dan *testing* mencapai 90% bahkan lebih [4].

Sunario Megawan dan Wulan Sri Lestari yang melakukan penelitian dengan judul “Deteksi *Spoofing* Wajah Menggunakan Faster R-CNN Dengan Arsitektur Resnet50

pada Video” menggunakan metode *Faster R-CNN* mendapatkan hasil bahwa rata-rata nilai akurasi yang dihasilkan pada tahap *training* adalah 97,07%, dengan jumlah *epoch* sebanyak 21, yang artinya pengujian menunjukkan bahwa model yang dihasilkan berhasil menentukan *bounding box* [5].

Hamdani Mubarak yang melakukan penelitian dengan judul “Identifikasi Ekspresi Wajah Berbasis Citra Menggunakan Algoritma *Covolution Neural Network (CNN)*” menggunakan metode *Backpropagation* mendapatkan hasil bahwa penelitian ini cukup baik dan didapat akurasi *training* sebesar 99,6% dan akurasi *testing* sebesar 88,89% yang dilakukan pada *epochs* 1000 [6].

Ada beberapa perbedaan dari penelitian ini dengan penelitian yang sebelumnya yang tercantum di dalam *state of the art* yaitu, ada pada algoritma yang dipakai pada penelitian sebelumnya paling banyak menggunakan *Convolution Neural Network* yang menghasilkan persentase akurasi 80%-90%. Sedangkan peneliti disini menggunakan algoritma *Recurrent Neural Network*, namun peneliti memanfaatkan *Teachable Machine* yang sudah menggunakan algoritma RNN didalamnya untuk membantu melakukan proses pelatihan dan perhitungan terhadap model dan menghasilkan persentase akurasi 90%-100%.

Penelitian ini tentunya berdasarkan pengetahuan yang telah ada seperti, sikap dasar pencak silat yang merupakan sikap-sikap statis yang dapat dilakukan untuk melatih kekuatan otot-otot, sebelum melakukan gerak dinamis sehingga pijakan kaki dapat menjadi kokoh. Pembentukan sikap merupakan dasar dari pembentukan gerak yang meliputi sikap jasmaniah dan sikap rohani [7]. Sikap dasar dari pencak silat itu semuanya sama, namun karena begitu banyaknya aliran pencak silat di Indonesia, sikap dasar dari setiap perguruan pun memiliki namanya masing-masing.

Pada perguruan Pencak Silat Bakti Negara sendiri ada beberapa sikap dasar seperti, sikap sempurna, jangkar kodok, jangkar kuda, jurus satu (ular menyebrang sungai) dan jurus dua (menghadang jalan).

Pada Penelitian ini menggunakan algoritma RNN (*Recurrent Neural Network*) untuk mengoptimalkan prediksi pengenalan terhadap obyek, hal ini dikarenakan konsep dasar dari RNN untuk melakukan prediksi terhadap data bukan hanya menggunakan data yang diberikan pada saat ini, melainkan juga mengambil data hasil dari prediksi atau *input* sebelumnya [8].

Namun untuk membantu perhitungan algoritma yang sangat kompleks peneliti menggunakan *Teachable Machine* yang merupakan alat *Graphic User Interface* berbasis *website* yang dikembangkan Google dan dapat digunakan membuat model klasifikasi yang sudah berisi algoritma RNN. Ada tiga langkah sederhana yang harus dilakukan untuk membuat model pelatihan yaitu, pertama mengumpulkan

dan mengelompokkan data ke dalam kelas atau kategori yang telah ditentukan, langkah kedua melakukan pelatihan model sesuai kebutuhan dan dapat langsung di lihat hasil dari pelatihan apakah dapat mengklasifikasikan dengan benar dan akurat, serta langkah terakhir adalah ekspor model yang telah dibuat ke proyek yang sedang dikerjakan baik berbasis pada *website* ataupun aplikasi [9].

Hasil dari model yang telah dilatih akan diimplementasikan ke dalam program aplikasi yang telah dibangun menggunakan *tools Flutter*, di sini peneliti memilih menggunakan *Flutter* karena *SDK (Software Development Kit)* yang dikembangkan oleh Google ini dapat mengembangkan aplikasi dengan cepat dan memiliki beberapa keunggulan seperti, *fast development, expressive and flexible, and native performance* [10].

## 2. METODE PENELITIAN

### 2.1 Metode pengumpulan data dan instrumen penelitian

#### 1. Metode Pengumpulan Data

Penelitian ini menggunakan metode pengumpulan data melalui wawancara dan observasi. Pada sesi wawancara dilakukan terhadap pelatih dari perguruan Pencak Silat Bakti Negara, dengan tujuan untuk mengetahui permasalahan dan mendapatkan data awal. Sedangkan sesi observasi pada penelitian ini dilakukan untuk mengetahui gerakan sikap dasar dari *Pencak Silat Bakti Negara* yang akan ada pada aplikasi. Berdasarkan hasil wawancara dan observasi didapatkan hasil sebagai berikut:

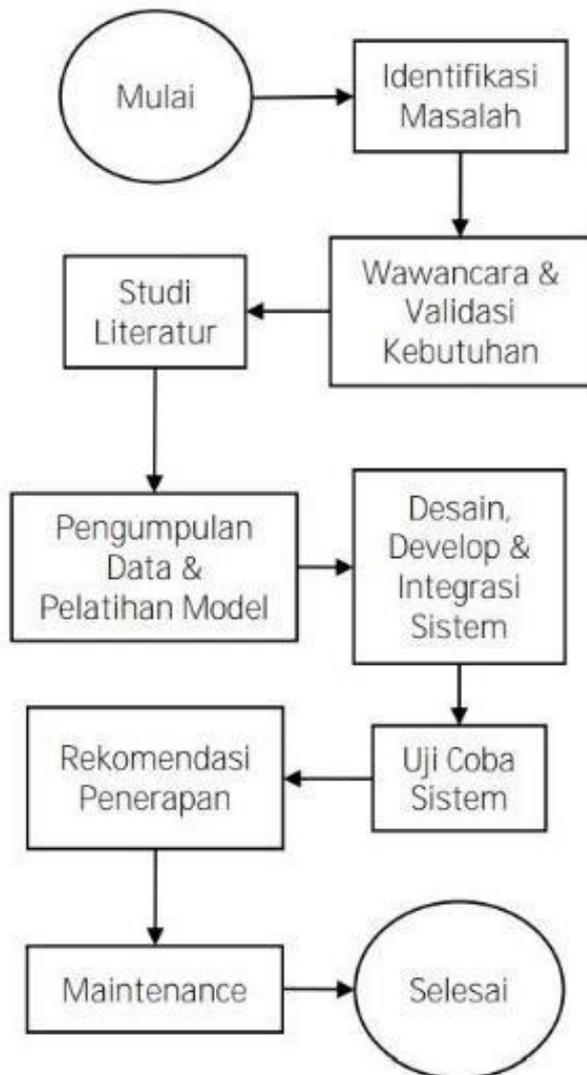
Terdapat lima sikap yang akan digunakan untuk pengenalan sikap dasar yang telah divalidasi sebelumnya melalui observasi terhadap 18 orang secara acak, mulai dari pelatih dan atlet di perguruan Pencak Silat Bakti Negara dengan persentase 77,8% mengatakan setuju atau cukup dengan lima sikap yang bisa dikenali di awal pengembangan aplikasi ini yaitu, sikap sempurna, jangkar kuda, jangkar kodok, jurus satu (ular menyebrang sungai), dan jurus dua (menghadang jalan), 16,6% mengatakan perlu tambahan sikap dari tendangan dasar dan 5,6% mengatakan perlu tambahan hormat perguruan di awal sebelum ke sikap dasar yang lainnya.

#### 2. Instrumen Penelitian

Instrument yang digunakan dalam penelitian ini berupa buku catatan atau *note* pada *smartphone* ataupun alat rekam yang digunakan untuk mendapatkan informasi saat melakukan wawancara dengan informan.

#### 2.2 Tahapan Penelitian

Jika digambarkan maka alur atau tahapan penelitian pada penelitian ini adalah seperti pada Gambar 2 berikut:



Gambar 2. Tahapan Penelitian

### 1. Identifikasi Masalah

Penelitian ini diawali dengan mengidentifikasi masalah yang ada pada tempat studi kasus dan didapatkan masalah yaitu keterbatasan untuk melakukan proses latihan akrobat pandemi saat ini.

### 2. Wawancara dan Validasi Kebutuhan

Pada sesi wawancara dilakukan terhadap pelatih untuk mendalami masalah yang ditemukan menghasilkan solusi yaitu, membangun aplikasi pengenalan sikap dasar perguruan Pencak Silat Bakti Negara yang dijadikan sebagai bahan pembelajaran atau pelatihan bagi atlet yang ingin melakukan latihan dari rumah masing-masing melalui *smartphone* mereka.

### 3. Studi Literatur

Tahapan ini perlu dilakukan untuk mengetahui teknologi atau algoritma apa yang akan digunakan dalam penelitian

ini agar aplikasi yang dibuat dapat melakukan pengenalan terhadap sikap dasar dengan akurat.

### 4. Pengumpulan Data dan Pelatihan Model

Proses pengumpulan data berupa gambar dengan *format .jpg* dilakukan secara langsung di tempat studi kasus. Data yang diambil merupakan data kualitatif karena bukan berupa angka tetapi berupa gambar sebanyak 200 *datasets* gambar dari setiap sikap dasar yang menggunakan 20 sampel atlet yang akan diambil gambarnya pada tiap sikap dasar, 1 orang atlet akan diambil gambarnya dengan 10 posisi yang berbeda sehingga dengan 20 orang atlet akan menghasilkan 200 data gambar pada setiap sikap dasar, total keseluruhan data dari lima sikap dasar tersebut adalah 1000 *datasets*.

Tahap selanjutnya setelah mengumpulkan data yaitu pelatihan model pada *Teachable Machine* untuk membuat model pengenalan *pose* atau gerakan sikap dasar pencak silat. Ada 3 langkah dalam pembuatan model ini, yaitu:

#### A. Input dan Klasifikasi Data

Pada tahap ini data yang telah terkumpul diklasifikasikan ke dalam lima kelas yang di dalam tiap kelas terdapat 200 *datasets yang dilatih*. Pada tahapan ini model pengenalan dijadikan satu yang dapat mengenali Sikap Sempurna, Jangkar Kuda dan Jangkar Kodok, Jurus 1 Ular Menyebang Sungai dan Jurus 2 Menghadang Jalan dengan total 1000 *datasets*.

#### B. Train Model

Pada tahap ini semua data yang telah di *input* akan di *training*, dari 1000 *datasets* yang ada akan dilatih sebanyak 30 *epochs*, yang dibatasi setiap interasi hanya 16 *batch size* dengan 0,0001 *learning rate*, dengan total 63 interasi untuk menyelesaikan pelatihan 1000 *datasets*. Hasil *accuracy* dari *epoch* akhir sebesar 0,99 dan hasil *loss* dari *epoch* akhir sebesar 0,00. Hasil *accuracy* dari setiap kelas dari 30 *epoch* itu yaitu, hasil *class* Sikap Sempurna sebesar 1,00, hasil Jangkar Kodok sebesar 1,00, hasil dari Jangkar Kuda sebesar 1,00, hasil dari Jurus 1 UMS sebesar 0,97 dan hasil dari Jurus 2 MJ sebesar 1,00.

#### C. Export Model

Hasil dari pembuatan model ini berupa *clustering* yang mana model ini melakukan pengenalan berdasarkan kemiripan *pose* atau gerakan dari data sikap yang telah di *training* sebelumnya dan rata-rata menghasilkan persentase akurasi di atas 90%. Model pelatihan yang sudah jadi kemudian di *upload* dan *link* dari model ini di *input* ke komputer untuk nanti di integrasikan dengan sistem yang dibuat dengan pemrograman *Flutter*.

### 5. Desain Pengembangan Sistem dan Integrasi Sistem

Sebelum masuk proses koding dari aplikasi, terlebih dahulu harus ada rancangan desain atau *user interface (UI)* dari

aplikasi yang akan dibuat, agar pada saat pembuatan aplikasi tinggal mengikuti desain yang telah dibuat sebelumnya. Setelah UI selesai maka *coding* dari program dengan menggunakan bahasa *Dart* pada *Flutter* bisa dilakukan dan dilakukan integrasi dari model pengenalan dengan sistem yang dibuat menggunakan *Flutter*.

## 6. Uji Coba Sistem

Setelah semua tahapan dilalui, kini memasuki tahap akhir yaitu uji coba dari sistem dengan pengujian *integration testing*. Pengujian ini dilakukan untuk menguji apakah model pengenalan yang telah diintegrasikan dengan sistem aplikasi Android yang telah dibangun dapat tetap mengenali dengan akurat atau tidak.

## 7. Rekomendasi Penerapan Sistem

Apabila semua tahapan telah dilalui, aplikasi juga telah lulus uji coba, maka aplikasi ini bisa direkomendasikan untuk diimplementasikan langsung pada tempat studi kasus.

## 8. Maintenance

Tahapan maintenance ini sudah di luar dari tahapan penelitian, tahap ini masuk kedalam pemeliharaan aplikasi yang dilakukan secara berkala setiap 6 bulan sekali atau sewaktu-waktu bilamana ada kebutuhan tambahan dari aplikasi pengenalan ini.

# 3. HASIL DAN PEMBAHASAN

## 3.1 Integrasi Model dengan Program

Pada penelitian ini, model pengenalan sikap dasar Pencak Silat yang telah di *train* pada *Teachable Machine* dapat diintegrasikan dengan program yang dibangun menggunakan *Flutter* dengan memanfaatkan *teachable package* versi terbaru, yaitu *teachable 0.0.2* yang dirilis pada tanggal 3 Juli 2021. *Teachable package* ini dipanggil menggunakan *syntac* atau sering disebut *widget* pada *Flutter* seperti pada *code program* di bawah ini:

```
child: Teachable(
  path: "pose/index.html",
  results: (res) {
    var resp = jsonDecode(res);
    // print("The values are ${}");
    setState(() {
      pose =
        (resp['Sikap Sempurna'] * 100.
0).toString();
    });
  },
),
```

Pada model pengenalan yang didapat dari *Teachable Machine* tersebut tidak di *download*, melainkan di *upload* dan *link* dari hasil *upload* model pengenalan tersebut

nantinya akan ditambahkan ke dalam *URL* di dalam logika dalam bahasa *Html* untuk *Teachable Machine*, seperti pada *code program* di bawah ini:

```
const URL =
  "https://teachablemachin
  e.withgoogle.com/models/tkmGLLNzJ/";
```

Selain itu diperlukan juga beberapa *permission* untuk menjalankan sistem pengenalan ini, seperti pada *code program* di bawah ini :

```
void main() async {
  WidgetsFlutterBinding.ensureInitia
  lized();

  await
  Permission.camera.request(); await
  Permission.microphone.reques
  t();
  runApp(Tengklung());
}
```

Pada aplikasi ini juga ditambahkan *handling error* untuk membatasi ketika akurasi semisal keluar 50% atau 80%, *alert* atau *pop up* apa yang akan muncul untuk memberikan *feedback* apakah dengan akurasi itu, gerakan yang dilakukan sudah benar atau salah, yang mana menggunakan *If else* untuk membuat logika *handling error* tersebut, seperti pada *code program* di bawah ini :

```
hasillatihan(String data) {
  temp = double.tryParse(data);

  if (temp! <= 10.0) {
    return 'MASIH SALAH';
  } else if (temp! >= 11.0 &&
  temp
  ! <= 70.0) {
    return 'HAMPIR BENAR';
  } else {
    return 'SUDAH BENAR';
  }
}
```

Pada logika *if else* ini juga perlu dibuat logika tambahan untuk men-*trigger alert* yang akan keluar saat logika *if else* membaca data akurasi yang keluar dari model, karena model yang ada melakukan prediksi secara terus menerus dan *realtime* yang membuat *alert* yang akan muncul juga harus *realtime* dapat me-*refresh* logika *if else* yang akan keluar, maka dari itu peneliti menambahkan *timer* yang mana akan di *refresh* setiap 1 detik untuk men-*trigger* logika *if else* untuk memberikan *alert* hasil akurasi apakah gerakan sudah benar atau salah, seperti pada *code program* di bawah ini:

```

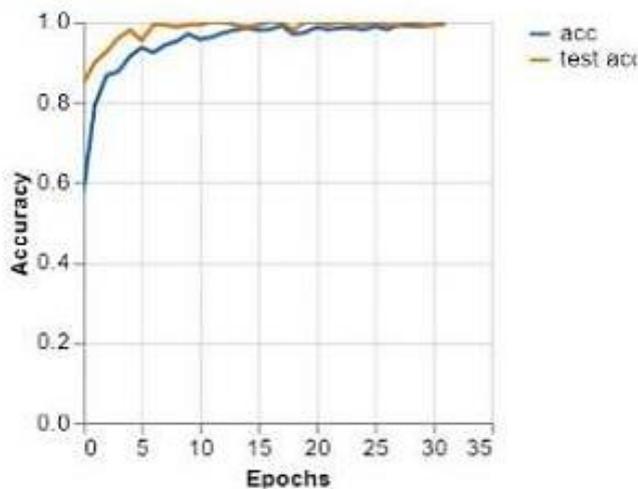
@Override
void initState() {
  timer =
  Timer.periodic(Duration(
  seconds: 1), (_) {
    setState(() {
      // textHasilLatihan();
      hasilLatihan(pose);
    });
  });
  super.initState();
}

@Override
void dispose()
// TODO: implement dispose
timer!.cancel();
super.dispose();
}
    
```

### 3.2 Perhitungan Algoritma RNN

Pada pembahasan ini tidak menampilkan proses perhitungan manual dari algoritma RNN, melihat dari sisi perhitungan yang sangat kompleks dan data matriks dari setiap data gambar yang terlalu besar, maka peneliti menggunakan *tools* dari *Teachable Machine* yang sudah memakai algoritma *Recurrent Neural Network* didalamnya dan menghasilkan *accuracy* dari *epoch* akhir sebesar 0,98 dan hasil *loss* dari *epoch* akhir sebesar 0,00 yang bisa dilihat pada Gambar 3 di bawah.

Accuracy per epoch



Sedangkan hasil perhitungan dari pelatihan, untuk *accuracy* dari setiap kelas yaitu, Sikap Sempurna sebesar 1.00, Jangkar Kodok sebesar 0.93, Jangkar Kuda sebesar 1.00, Jurus 1 UMS sebesar 0,97 dan Jurus 2 MJ sebesar 1.00 serta bisa dilihat pada Gambar 4 di bawah ini:

Accuracy per class

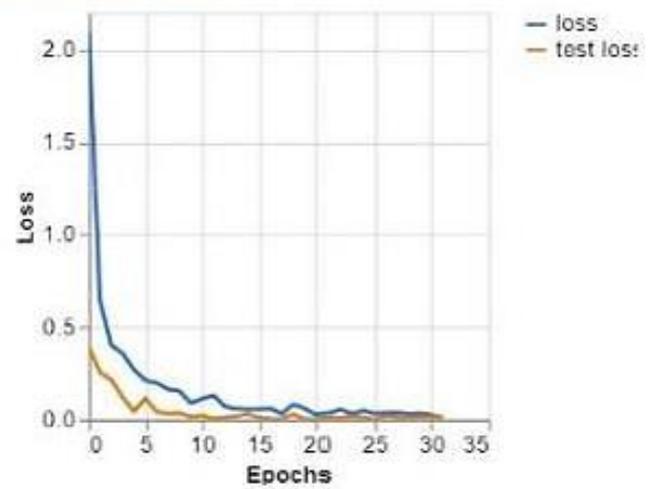
CLASS	ACCURACY	EPOCH
Sikap Sempurna	1.00	30
Jangkar Kodok	1.00	30
Jangkar Kuda	1.00	30
Jurus 1 UMS	0.97	30
Jurus 2 MJ	1.00	30

Gambar 4. Hasil Perhitungan Accuracy Class dari Pelatihan

### 3.3 Perhitungan Keseimbangan Klasifikasi

Model yang sudah dilatih untuk melakukan prediksi untuk menentukan apakah gerakan sikap dasar dari perguruan Pencak Silat Bakti Negara yang dilakukan sudah benar atau salah. Dengan *user* yang sudah mencoba prediksi dari model sebanyak 25 orang acak, baik atlet pencak silat maupun orang awam. Dari klasifikasi model memprediksi 16 orang bergerak benar dan 9 orang masih salah, namun kenyataannya 17 orang bergerak benar dan 8 orang masih salah.

Loss per epoch



Gambar 3. Hasil Perhitungan Accuracy dan Loss dari Pelatihan

Pada penelitian ini melihat performa dari prediksi akan menggunakan *Confusion Matrix* untuk memetakan hasil

dari prediksi model yang sudah dilatih ke dalam tabel dengan 4 kombinasi berbeda dari nilai prediksi dan aktual

yang dihasilkan [11].

**Tabel 1.** Tabel *Software* dan *Hardware*

<i>N</i> = 25	Aktual : Positif (1)	Aktual : Negatif (0)
Prediksi : Positif (1)	TP : 15	FP : 1
Prediksi : Positif (0)	FN : 1	TN : 8
	16	9

Keterangan pada tabel di atas:

1. True Positif (TP): model memprediksi orang bergerak benar dan memang orang itu bergerak dengan benar [11].
2. True Negatif (TN): model memprediksi orang masih salah dan memang orang itu masih bergerak dengan salah [11].
3. False Positif (FP): model memprediksi orang bergerak benar dan ternyata prediksi salah, ternyata orang masih bergerak salah [11].
4. False Negatif (FN): model memprediksi orang masih salah bergerak dan ternyata prediksi salah, ternyata orang sudah melakukan gerakan dengan benar [11].

Seperti pada keterangan di atas bahwa FP adalah kesalahan tipe 2 yang mana kesalahan ini sangat berbahaya, misalkan jika orang diprediksi telah bergerak dengan benar namun kenyataannya masih bergerak dengan salah, maka orang itu bisa saja akan mengalami cedera karena selalu bergerak dengan sikap atau teknik yang masih salah karena prediksi yang diterima mengatakan gerakan yang dilakukannya sudah benar.

Pada penelitian ini, hasil yang telah didapat dan dipetakan pada tabel *Confusion Matrix* di atas, maka dapat dihitung baik dari nilai *accuracy*, *precision*, *recall* dan F-1 Score-nya, sebagai berikut:

#### A. Accuracy (A)

Perhitungan ini dapat menggambarkan seberapa akurat model di dalam mengklasifikasikan dengan benar dan tepat, menggunakan rumus  $A = (TP+TN) / (TP+FP+FN+TN)$  (1) [11].

Seperti persamaan 1, maka perhitungan *Accuracy* sebagai berikut:

$$\begin{aligned} Accuracy &= (TP+TN) / (TP+FP+FN+TN) \\ &= (15+8) / (15+1+1+8) \\ &= 0.92 \\ &= 0.92 * 100\% \\ &= 92\% \end{aligned}$$

#### B. Precision (P)

Perhitungan ini dapat menggambarkan seberapa akurat antara data yang diminta dengan hasil prediksi dari model, dengan rumus  $P = (TP) / (TP+FP)$  (2) [11]. Seperti persamaan 2, maka perhitungan *Precision* sebagai berikut:

$$\begin{aligned} Precision &= (TP) / (TP+FP) \\ &= (15) / (15+1) \\ &= 0.9375 \\ &= 0.9375 * 100\% \\ &= 94\% \end{aligned}$$

#### C. Recall

Perhitungan ini dapat menggambarkan keberhasilan dari model yang dapat menemukan kembali sebuah informasi dengan rumus  $R = (TP) / (TP+FN)$  (3) [11].

Seperti persamaan 3, maka perhitungan *Recall*, sebagai berikut:

$$\begin{aligned} Recall &= (TP) / (TP+FN) \\ &= (15) / (15+1) \\ &= 0.9375 \\ &= 0.9375 * 100\% \\ &= 94\% \end{aligned}$$

#### D. F-1 Score (F-1 S)

Perhitungan ini akan menggambarkan perbandingan dari rata-rata antara *precision* dan *recall* yang dibobotkan dengan rumus  $F-1 S = (2 * R * P) / (R + P)$  (4) [20]. Seperti persamaan 4, maka perhitungan F-1 Score sebagai berikut:

$$\begin{aligned} F-1 S &= (2 * R * P) / (R + P) \\ &= (2 * 0.9375 * 0.9375) / (0.9375 + 0.9375) \\ &= 1.7578 / 1.875 \\ &= 0.94 * 100\% \\ &= 94\% \end{aligned}$$

Jika hasil perhitungan memiliki jumlah data *False Negatif* dan *False Positif* yang sangat mendekati atau simetris maka *Accuracy* dapat dijadikan sebagai acuan untuk melihat performa algoritma. Namun jika jumlahnya tidak mendekati, sebaiknya menggunakan F-1 Score sebagai acuannya [11].

#### 3.4 Akurasi Pengenalan

Pada penelitian ini, dari segi akurasi yang dihasilkan dari model pengenalan saat di *train* pada *Teachable Machine*

yaitu, Sikap Sempurna sebesar 100%, Jangkar Kodok sebesar 93%, Jangkar Kuda sebesar 100%, Jurus 1 UMS (Ular Menyebrang Sungai) sebesar 97% dan Jurus 2 MJ (Menghadang Jalan) sebesar 100%, hal ini membuat akurasi rata-rata semua *class* di atas 90% dari 30 *epoch*.



Gambar 5. Hasil Akurasi saat Bergerak dengan Benar

Saat model diintegrasikan dengan program, sistem pengenalan menghasilkan akurasi rata-rata di atas 70% saat pengguna sudah melakukan gerakan dengan benar seperti pada Gambar 5, sedangkan di sisi lain sistem memberikan akurasi rata-rata dibawah 10% saat pengguna masih bergerak salah seperti pada Gambar 6.

Pada aplikasi ini sudah berisi *feedback* berupa teks di bagian bawah akurasi saat gerakan sudah diprediksi benar atau salah. Selain *feedback* yang telah dibuat, pengguna juga akan diberikan pengetahuan melalui deskripsi berupa gambar dan teks mengenai bagaimana bentuk dari sikap yang ada dan dapat dipelajari pada aplikasi ini. Untuk tampilan aplikasi dapat dilihat pada Gambar 7.



Gambar 6. Hasil Akurasi saat Masih Bergerak dengan Salah



Gambar 7. Hasil Akurasi saat Masih Bergerak dengan Salah

#### 4. KESIMPULAN

Model pengenalan sikap dasar Pencak Silat yang di *train* pada *Teachable Machine* dapat diintegrasikan dengan aplikasi yang dibangun menggunakan *Flutter* dengan memanfaatkan *widget inAppWebView* dari *Flutter* dan memanfaatkan *teachable package* dari *Flutter* yang menghasilkan akurasi pengenalan diatas 70% untuk gerakan sikap dasar yang diprediksi benar dan akurasi dibawah 30% untuk gerakan yang diprediksi salah dari semua sikap dasar yang bisa dikenali model pengenalan, serta pada bagian *feedback* saat melakukan prediksi juga sudah *user friendly* yang dapat memudahkan pengguna, karena sudah ada *alert* berupa teks di bawah akurasi yang muncul ketika pengguna sudah bergerak dengan benar ataupun saat gerakan yang dilakukan masih salah.

Walaupun penelitian ini bisa dibilang telah sukses dapat membuat sistem atau model pengenalan sikap dasar dari pencak silat, namun berdasarkan data *end user* atau masukan-masukan yang telah peneliti terima dari pengguna yang akan menggunakan aplikasi ini nantinya, rata-rata mengatakan aplikasi ini mudah untuk dipahami dan digunakan, namun ada juga beberapa saran seperti perlunya ada keterangan jika gerakan pengguna masih salah, jadi diberikan *feedback* dimana letak kesalahannya, kemudian tata letak dari beberapa UI juga perlu dirapikan lagi dan perlunya *guiding* saat perngguna baru menggunakan aplikasi ini.

#### Ucapan Terima Kasih

Peneliti mengucapkan terima kasih kepada semua pihak yang terlibat didalam penelitian ini, baik dari dosen pembimbing, Kampus STMIK Primakara yang sudah menyediakan ruang untuk membuat penelitian ini juga PSPS Bakti Negara Ranting Bayad sebagai tempat studi kasus penelitian ini dilakukan.

**DAFTAR PUSTAKA**

- [1] Admin, "Mobile Operating System Market Share Indonesia Oct 2019 - Oct 2020," *Statcounter Global Stats*, November 2020, [Online]. Tersedia: <https://gs.statcounter.com/os-market-share/mobile/indonesia> [Diakses: 1 Desember 2020].
- [2] S. R. DEWI, "Deep Learning Object Detection Pada Video Menggunakan Tensorflow Dan Convolutional Neural Network," *Universitas Islam Indonesia*, Yogyakarta, 2018.
- [3] H. A. A., Candradewi, I, "Sistem Pengenal Isyarat Tangan Untuk Mengendalikan Gerakan Robot Beroda menggunakan Convolutional Neural Network," *Indones. J. Electron. Instrum. Syst.*, vol. 9, pp. 192–202, 2019.
- [4] Bakti, M. B. S., "Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network," *Sekolah Tinggi Teknik Surabaya*, Surabaya, p. 16, 2019.
- [5] W. S. Lestari dan S. Megawan, "Deteksi Spoofing Wajah Menggunakan FasterR CNN dengan Arsitektur Resnet50 pada Video," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 9, pp. 261–267, 2020, doi:10.22146/v9i3.231.
- [6] H. Mubarak, "Identifikasi Ekspresi Wajah Berbasis Citra Menggunakan Algoritma Covolution Neural Network (CNN)," *Universitas Islam Negeri Maulana Malik Ibrahim*, Malang, 2019.
- [7] A. M. Agung Nugroho, "Melatih Sikap Dan Gerak Dasar Pencak Silat Bagi Pesilat Pemula," *Jurnal Olahraga Prestasi*, vol. 1, p. 18, 2005, doi: 10.21831/jorpres.v1i2.6859.
- [8] Steven Sen dkk., "Komparasi Metode Multilayer Perceptron (MLP) dan Long Short Term Memory (LSTM) dalam Peramalan Harga Beras," *Ultimatic*, Vol. XII, No. 1, 2020. ISSN: 2085-4552.
- [9] M. Carney dkk., "Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification," *presented at the CHI '20: CHI Conference on Human Factors in Computing Systems*, April 2020, doi: 10.1145/3334480.3382839.
- [10] Flutter Intern, "Flutter" Flutter Developer, 2020, [Online]. Tersedia: <https://flutter.dev/> [Diakses: 17 Juli 2021].
- [11] Admin, "Confusion Matrix", *Binus University School of Computer Science*, November 2020, [Online]. Tersedia: <https://socs.binus.ac.id/2020/11/01/confusion-matrix/> [Diakses: 19 Agustus 2021].