

SKRIPSI

**KLASIFIKASI GAMBAR ALAT MUSIK TRADISIONAL JAWA BARAT
BERBASIS ANDROID MENGGUNAKAN LIBRARY TENSORFLOW LITE**

Oleh:
IKHSAN FADILA HIDAYAT
065115161



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN
BOGOR**

2022

HALAMAN PENGESAHAN

Judul : KLASIFIKASI GAMBAR ALAT MUSIK TRADISIONAL JAWA BARAT
BERBASIS *ANDROID* MENGGUNAKAN *LIBRARY TENSORFLOW LITE*

Nama : Ikhsan Fadila Hidayat

NPM : 065115161

Mengesahkan,

Pembimbing Pendamping
Program Studi Ilmu Komputer
FMIPA - UNPAK

Pembimbing Utama
Program Studi Ilmu Komputer
FMIPA - UNPAK

Irma Anggraeni, S.T., M.Kom.

Dr Prihastuti Harsani, M.Si.

Mengetahui,

Ketua Program Studi Ilmu Komputer
FMIPA – UNPAK

Dekan
FMIPA - UNPAK

Arie Qur'ania, M.Kom.

Asep Denih, S.Kom., M.Sc., Ph.D.

PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI

Dengan ini saya yang bertanda tangan dibawah ini :

Nama : Ikhsan Fadila Hidayat
NPM : 065115161
Program Studi : Ilmu Komputer
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Universitas Pakuan Bogor

Menyatakan bahwa sejauh yang saya ketahui, karya tulis ini bukan merupakan karya tulis yang pernah dipublikasikan atau sudah pernah dipakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian-bagian di mana sumber informasinya dicantumkan dengan cara referensi yang semestinya.

Demikian pernyataan ini saya buat dengan sebenar-benarnya. Apabila kelak dikemudian hari terdapat gugatan, penulis bersedia dikenakan sanksi sesuai dengan peraturan yang berlaku.

Bogor, Desember 2022



Ikhsan Fadila Hidayat
Ikhsan Fadila Hidayat

PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Dengan ini saya yang bertanda tangan dibawah ini :

Nama : Ikhsan Fadila Hidayat
NPM : 065115161
Program Studi : Ilmu Komputer
Judul Skripsi : Klasifikasi Gambar Alat Musik Tradisional Jawa Barat
Berbasis *Android* Menggunakan *Library Tensorflow Lite*

Dengan ini saya menyatakan bahwa paten dan hak cipta dari produk skripsi dan tugas akhir di atas adalah benar karya saya dengan arahan dari pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun.

Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam daftar Pustaka dibagian akhir Skripsi ini.

Dengan ini saya melimpahkan Paten, hak cipta dan karya tulis saya kepada Universitas Pakuan.

Bogor, Desember 2022



Ikhsan Fadila Hidayat

HALAMAN PERSEMBAHAN

Sujud syukurku kusembahkan kepadaMu ya Allah, Tuhan Yang Maha Agung dan Maha Tinggi. Atas takdirmu saya bisa menjadi pribadi yang berpikir, berilmu, beriman dan bersabar. Semoga keberhasilan ini menjadi satu langkah awal untuk masa depanku, dalam meraih cita-cita.

Dengan ini saya persembahkan karya ini untuk, Alm Bapak Uyan Hidayat dan Alm Euis Sadiyah Terima kasih atas kasih sayang yang berlimpah dari mulai saya lahir, hingga saya sudah sebesar ini. Lalu teruntuk Bunda Leni Marlina, terima kasih juga atas limpahan doa yang tak berkesudahan. Serta segala hal yang telah Bunda lakukan, semua yang terbaik.

Terima kasih selanjutnya untuk Istri tercinta Qara Apriliana dan adik Zhafari Irsyad dan Adzkiya Dhiana Syifa, yang luar biasa dalam memberi dukungan dan doa yang tanpa henti.

Terima kasih juga yang tak terhingga untuk dosen pembimbing utama Dr Prihastuti Harsani, M.Si. dan pembimbing pendamping Irma Anggraeni, S.T., M.Kom. atas kesabaran kalian membimbingku, memberikanku berbagai macam pengetahuan, memberiku saran dan solusi sehingga aku dapat menyelesaikan tugas akhir ini.

Ucapan terima kasih ini saya persembahkan juga untuk seluruh teman-teman saya FMIPA ILKOM EF 2015. Terima kasih untuk memori yang kita rajut setiap harinya, atas tawa yang setiap hari kita miliki, dan atas solidaritas yang luar biasa. Sehingga masa kuliah selama ini menjadi lebih berarti. Semoga saat-saat indah itu akan selalu menjadi kenangan yang paling indah.

Untuk semua pihak yang saya sebutkan, terima kasih atas semuanya. Semoga Tuhan senantiasa membalas setiap kebaikan kalian. Serta kehidupan kalian semua juga dimudahkan dan diberkahi selalu oleh Allah SWT.

Saya menyadari bahwa hasil karya skripsi ini masih jauh dari kata sempurna, tetapi saya harap isinya tetap memberi manfaat sebagai ilmu dan pengetahuan bagi para pembacanya.

RIWAYAT HIDUP



Penulis dilahirkan di Bekasi pada tanggal 21 Februari 1997 dari pasangan Bapak Uyan Hidayat dan Ibu Euis Sadiyah sebagai anak pertama dari tiga bersaudara. Penulis memulai pendidikan Sekolah Dasar pada tahun 2004 dan menyelesaikan pendidikan Sekolah Dasar pada tahun 2009 di SDN Ciawi 01 IPK, Kabupaten Bogor, Jawa Barat. Kemudian penulis melanjutkan pendidikan Sekolah Menengah Pertama pada tahun yang sama dan menyelesaikan pada tahun 2012 di SMPN 1 Ciawi, Kabupaten Bogor, Jawa Barat. Penulis melanjutkan Sekolah Menengah Kejuruan pada tahun 2012 di SMK Wikrama, Kota Bogor, Jawa Barat dan meraih kelulusan di tahun 2015. Setelah lulus SMK penulis melanjutkan pendidikan perguruan tinggi ke Universitas Pakuan Bogor, Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi S1 Ilmu Komputer pada tahun 2015.

Penulis pernah mengikuti UKM (Unit Kegiatan Mahasiswa). Pada bulan Mei – Agustus 2022 penulis menyelesaikan tugas akhir yang berjudul “**Klasifikasi Gambar Alat Musik Tradisional Jawa Barat Berbasis *Android* Menggunakan *Library Tensorflow Lite***” dan dinyatakan lulus sebagai Sarjana Komputer setelah melalui sidang komprehensif pada tanggal 31 Agustus 2022 dengan predikat **Sangat Memuaskan**.

RINGKASAN

Ikhsan Fadila Hidayat, 2022. Klasifikasi Gambar Alat Musik Tradisional Jawa Barat Berbasis *Android* Menggunakan *Library Tensorflow Lite*.
Dibawah bimbingan **Dr Prihastuti Harsani, M.Si.** dan **Irma Anggraeni, S.T., M.Kom.**

Salah satu hasil kesenian budaya masyarakat Indonesia adalah alat musik tradisional. Sebagai alat yang digunakan untuk mengiringi musik tradisional, musik tradisional dapat diartikan sebagai musik yang memiliki adat istiadat atau nilai budaya yang diwariskan secara turun-temurun. Di zaman dengan kemajuan teknologi yang semakin canggih, alat musik tradisional kini semakin tertinggal dengan adanya alat musik modern. Sehingga masyarakat lebih mengenali alat musik modern dibandingkan dengan alat musik tradisional, khususnya dikalangan generasi muda. Pengetahuan akan jenis alat musik tradisional sudah semakin berkurang, hal ini di karenakan kurangnya ketertarikan masyarakat dengan budaya di nusantara. Maka dari itu Klasifikasi Gambar adalah solusi yang tepat untuk mengatasi masalah tersebut.

Klasifikasi gambar atau citra adalah sebuah pekerjaan untuk memasukan citra dan menempatkan ke dalam suatu kategori. Ini merupakan salah satu dari permasalahan yang ada pada *computer vision* yang dapat disederhakan dan memiliki berbagai macam aplikasinya. Salah satu aplikasi dalam klasifikasi citra adalah pengklasifikasian nama tempat pada suatu citra. Klasifikasi ini menerapkan Algoritma CNN (*Convolutional Neural Network*) yang merupakan salah satu metode *Deep Learning* dengan keunggulan, akurasi sangat tinggi dalam masalah pengenalan gambar, secara otomatis mendeteksi fitur penting tanpa pengawasan manusia, dan pembagian bobot.

Aplikasi klasifikasi gambar ini dapat digunakan oleh siswa/siswi, mahasiswa/mahasiswi, generasi milenial dan masyarakat umum. Dengan memanfaatkan fitur upload gambar pada perangkat *smartphone* untuk mendeteksi gambar tersebut.

KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT, karena izin,rahmat dan hidayah-Nya saya selaku penulis dapat menyelesaikan laporan hasil ini yang berjudul “Klasifikasi Gambar Alat Musik Tradisional Jawa Barat Berbasis *Android* Menggunakan *Library Tensorflow Lite*”. Penulisan laporan ini merupakan syarat menuju sidang skripsi di program Studi Ilmu Komputer FMIPA Universitas Pakuan Bogor.

Dalam laporan praktek lapangan ini saya selaku penulis laporan praktek lapang ingin mengucapkan terimakasih sebesar-besarnya kepada :

1. Ibu Dr Prihastuti Harsani, M.Si. Selaku Pembimbing Utama yang senantiasa memberikan bimbingan selama pembuatan laporan skripsi ini berlangsung.
2. Ibu Irma Anggraeni, S.T., M.Kom. Selaku Pembimbing Pendamping yang senantiasa memberikan pengarahan selama bimbingan laporan skripsi ini berlangsung.
3. Ibu Arie Qur'ania, M.Kom., Selaku Ketua Program Studi Ilmu Komputer.
4. Orang tua saya, keluarga yang selalu memberikan semangat, dukungan dan do'a kepada saya.
5. Seluruh teman – teman kelas E dan F Program Studi Ilmu Komputer Angkatan 2015, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Pakuan yang menyemangati dan memberikan saran kepada saya.

Penulis menyadari masih banyak kekurangan didalam penulisan usulan penelitian ini, baik dari tata tulis dan tata bahasa, serta keterbatasan materi yang dimuat di dalamnya. Oleh karena itu penulis berharap adanya masukan dan saran yang sifatnya membangun dalam penulisan usulan penelitian ini. Akhir kata semoga Penelitian ini dapat bermanfaat sesuai dengan maksud dan tujuan khususnya bagi penulis dan bagi para pembaca umumnya.

Bogor, Desember 2022

Ikhsan Fadila Hidayat
065115161

DAFTAR ISI

HALAMAN PENGESAHAN	I
PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI.....	II
PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA	III
HALAMAN PERSEMBAHAN	IV
RIWAYAT HIDUP	V
RINGKASAN	VI
KATA PENGANTAR.....	VII
DAFTAR GAMBAR.....	V
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan Penelitian	2
1.3 Ruang Lingkup	2
1.4 Manfaat	2
BAB II TINJAUAN PUSTAKA	3
2.1 Klasifikasi Citra/Gambar	3
2.2 Alat Musik Tradisional	3
2.3 Android.....	3
2.4 Android Studio	4
2.5 Tensorflow dan Tensorflow Lite	4
2.6 Deep Learning	5
2.7 Arsitektur Neural Network	5
2.8 Convolutional Neural Network	5
2.9 Convolutional Layer	6
2.10 Rectified Linear Unir Layer (ReLu).....	6
2.11 Max Pooling Layer	7
2.12 Fully Connected Layer	7
2.13 Softmax	8
2.14 Teachable Machine	8
2.15 Penelitian Terdahulu.....	8
BAB III METODE PENELITIAN	10

3.1 Tahapan Penelitian.....	10
3.2 Perencanaan.....	10
3.3 Penelitian.....	11
3.4 Perancangan.....	11
3.5 Pembuatan Aplikasi.....	11
3.6 Pengujian.....	11
3.7 Implementasi Aplikasi.....	11
BAB IV PERANCANGAN DAN IMPLEMENTASI.....	12
4.1 Perancangan.....	12
4.1.1 Rancangan <i>Flowchart</i> Sistem.....	12
4.1.2 Rancangan Desain <i>Interface</i>	13
4.1.3 Analisis Kebutuhan Data.....	14
4.2 Pembuatan Aplikasi.....	15
BAB V HASIL DAN PEMBAHASAN.....	17
5.1 Hasil.....	17
5.2 Pembahasan.....	18
5.3 Penentuan Parameter Model.....	18
5.3.1 Pengaruh Jumlah <i>Epoch</i>	18
5.3.2 Pengaruh Jumlah Data <i>Train</i>	19
5.3.3 Pengaruh Nilai <i>Learning Rate</i>	19
A. Tabel Hasil Penentuan Parameter Model Epoch 20.....	19
B. Tabel Hasil Penentuan Parameter Model Epoch 50.....	21
C. Tabel Hasil Penentuan Parameter Model Epoch 100.....	22
5.3.4 Model Hasil <i>Training</i>	24
5.4 Uji coba.....	26
BAB VI KESIMPULAN DAN SARAN.....	35
6.1 Kesimpulan.....	35
6.2 Saran.....	35
DAFTAR PUSTAKA.....	36
LAMPIRAN.....	39

DAFTAR GAMBAR

Gambar 1 . Alat musik tradisional Jawa Barat (angklung, kecapi, kendang)	3
Gambar 2 . Arsitektur CNN	5
Gambar 3 . Proses Ekstrak Feature Map Dari Convolutional Network	6
Gambar 4 . Grafik Fungsi Aktivasi ReLu	7
Gambar 5 . Proses Max Pooling Pada Feature Map	7
Gambar 6 . Tahapan Penelitian	10
Gambar 7 . Flowchart Sistem	12
Gambar 8 . Dataset Kaggle.com dan Google Image	13
Gambar 9 . Desain Interface	14
Gambar 10 . Pilihan Pembuatan Model Klasifikasi	16
Gambar 11 . Pemberian Label Kelas	16
Gambar 12 . File Model Klasifikasi	16
Gambar 13 . Hasil Akurasi	24
Gambar 14 . Confusion Matrix	24
Gambar 15 . Grafik Accuracy	26
Gambar 16 . Grafik Loss	26

DAFTAR TABEL

Tabel 1 . Tabel Penelitian Terdahulu	9
Tabel 2 . Tabel Data Training	14
Tabel 3 . Tabel Data Testing	15
Tabel 4 . Tabel Hasil User Interface	17
Tabel 5 . Tabel Hasil Train Dan Validation	25
Tabel 6 . Tabel Uji Coba Struktural	26
Tabel 7 . Tabel Uji Coba Fungsional	27
Tabel 8 . Tabel Uji Coba Validasi Keakuratan Kecapi	28
Tabel 9 . Tabel Uji Coba Validasi Keakuratan Kendang	29
Tabel 10 . Tabel Uji Coba Validasi Keakuratan Angklung	29
Tabel 11 . Tabel Confusion Matrix	30
Tabel 12 . Tabel Uji Kemiripan	31

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indonesia merupakan bangsa yang terdiri dari beraneka ragam budaya. Dengan beraneka ragam budaya tersebut Indonesia memiliki ratusan kelompok etnik atau suku bangsa. Menurut sensus BPS (Badan Pusat Statistik) di tahun 2010 Indonesia memiliki lebih dari 300 kelompok etnik atau suku bangsa, lebih tepatnya terdapat 1.340 suku bangsa di Tanah Air. Suku Jawa adalah kelompok terbesar di Indonesia dengan jumlah yang mencapai 41% dari total populasi. Salah satu kebudayaan yang menarik dimata masyarakat adalah kesenian yang beragam di setiap daerah.

Salah satu hasil kesenian budaya masyarakat Indonesia adalah alat musik tradisional. Sebagai alat yang digunakan untuk mengiringi musik tradisional, musik tradisional dapat diartikan sebagai musik yang memiliki adat istiadat atau nilai budaya yang diwariskan secara turun-temurun. Alat musik ini juga sangat enak didengarkan walaupun dipadukan dengan alat musik modern saat ini, sehingga alat musik tradisional ini akan selalu terjaga meskipun muncul berbagai alat musik modern lainnya.

Di zaman dengan kemajuan teknologi yang semakin canggih, alat musik tradisional kini semakin tertinggal dengan adanya alat musik modern. Sehingga masyarakat lebih mengenali alat musik modern dibandingkan dengan alat musik tradisional, khususnya dikalangan generasi muda. Pengetahuan akan jenis alat musik tradisional sudah semakin berkurang, hal ini di karenakan kurangnya ketertarikan masyarakat dengan budaya di nusantara.

Seiring dengan majunya teknologi, klasifikasi citra digital sangat dibutuhkan diberbagai bidang, seperti : bisnis, pertanian, informatika, kedokteran. Beberapa penelitian yang telah dilakukan misalnya Pengenalan metode CNN mengenai rambu-rambu lalu lintas (Visalini, 2017) dan Klasifikasi Citra Wayang Golek (Triano Nurhikmat, 2018).

Berdasarkan latar belakang di atas, penelitian ini menerapkan Algoritma CNN (*Convolutional Neural Network*) yang merupakan salah satu metode *Deep Learning* dengan keunggulan, akurasi sangat tinggi dalam masalah pengenalan gambar, secara otomatis mendeteksi fitur penting tanpa pengawasan manusia, dan pembagian bobot. Untuk melatih model objek menggunakan *teachable machine* dari *Google*, penelitian ini bertujuan untuk membantu masyarakat mengenali nama dari alat musik tradisional. Penelitian ini berfokus terhadap bagaimana mengklasifikasikan citra alat musik tradisional kedalam nama alat musik tradisional.

1.2 Tujuan Penelitian

Tujuan penelitian ini adalah membuat model Klasifikasi Gambar Alat Musik Tradisional Jawa Barat Berbasis *Android* Menggunakan *Library Tensorflow Lite*.

1.3 Ruang Lingkup

Pembuatan aplikasi ini dibatasi pada :

1. Aplikasi ini digunakan untuk pengklasifikasian gambar angklung, kendang, kecap.
2. Aplikasi ini dibuat dengan *Android Studio* dan *Tensorflow Lite*.
3. Aplikasi ini melatih model objek menggunakan *teachable machine* dari *Google*.

1.4 Manfaat

1. Sistem dapat mengenali serta memberikan akurasi pada gambar yang telah dideteksi.
2. Dapat dikembangkan sebagai aplikasi yang dapat mengetahui nama alat musik lain.
3. Memudahkan pengguna untuk mengetahui nama alat musik tradisional Jawa Barat.
4. Mengetahui tingkat akurasi dari hasil klasifikasi menggunakan CNN.

BAB II

TINJAUAN PUSTAKA

2.1 Klasifikasi Citra/Gambar

Klasifikasi gambar atau citra adalah sebuah pekerjaan untuk memasukan citra dan menempatkan ke dalam suatu kategori. Ini merupakan salah satu dari permasalahan yang ada pada *computer vision* yang dapat disederhakan dan memiliki berbagai macam aplikasinya. Salah satu aplikasi dalam klasifikasi citra adalah pengklasifikasian nama tempat pada suatu citra.

Setiap citra yang di input pada *training set data* diberikan label atau penamaan. Saat klasifikasi, label atau penamaan tersebut akan menjadi perbandingan dengan hadil hipotesis yang diberikan oleh model pembelajaran dan akan menghasilkan nilai *error*. Klasifikasi yang terawasi ini bisa sangat efektif dan akurat dalam mengklasifikasi citra tempat maupun objek lainnya. Banyak metode dan algoritma yang dapat mendukung proses klasifikasi yang terawasi terutama dengan teknik *Deep Learning* (Putri,2018).

2.2 Alat Musik Tradisional

Alat musik Tradisional merupakan alat musik yang lahir dan berkembang secara turun temurun (diwariskan dari nenek moyang) dan menjadi identitas atau ciri khas bagi suku atau daerah tertentu.

Berikut beberapa pengertian musik tradisional menurut para ahli :

1. Musik tradisional merupakan musik yang dipakai sebagai perwujudan dan nilai budaya yang sesuai dengan tradisi (Sedyawati, 1992).
2. Musik tradisional adalah suatu seni budaya yang sejak lama turun-temurun telah hidup dan berkembang pada daerah tertentu (Tumbijo, 1997).



Gambar 1. Alat musik tradisional Jawa Barat (angklung, kecapi, kendang)

2.3 Android

Android merupakan sebuah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Android adalah sistem operasi yang bersifat *open source* (sumber terbuka) disebut *open source* karena *source code* (kode sumber) dari sistem operasi android dapat dilihat, diunduh, dan dimodifikasi secara bebas. Dengan adanya aplikasi android nantinya akan membantu pengguna untuk memonitoring melalui smartphome. (Bagaslewi, 2017).

2.4 Android Studio

Android Studio adalah sebuah environment yang dibuat untuk membantu developer dalam mengembangkan aplikasi yang berbasis android. Android Studio menawarkan banyak fitur untuk meningkatkan produktivitas pengguna saat membuat aplikasi Android, seperti (Android, 2019) :

- Sistem versi berbasis *Gradle* yang fleksibel
- Emulator yang cepat dan kaya fitur
- Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
- *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
- Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
- Alat pengujian dan kerangka kerja yang ekstensif
- Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
- Dukungan C++ dan NDK
- Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*

2.5 Tensorflow dan Tensorflow Lite

Tensorflow adalah *library* perangkat lunak yang dikembangkan oleh Tim *Google Brain* Mesin Cerdas *Google* Asosiasi, yang bertujuan untuk melakukan pembelajaran mesin dan jaringan syaraf dalam penelitiannya. *Tensorflow* menggabungkan aljabar komputasi dengan teknik optimasi kompilasi, yang memfasilitasi perhitungan banyak ekspresi matematika. Fitur utama yang terdapat dalam *tensorflow* adalah:

1. Mendefinisikan, mengoptimalkan, dan menghitung secara matematis ekspresi wajah yang melibatkan *array multidimension (tensors)*.
2. Pemrograman pendukung jaringan syaraf dalam dan teknik *machine learning*.
3. Pemakaian *GPU (Graphics Processing Unit)* yang efisien, mengotomasi manajemen dan optimalisasi memori yang sama terhadap data yang digunakan. *Tensorflow* mampu menulis kode yang sama dan menjalankannya di *CPU* atau *GPU*. Lebih khususnya lagi *tensorflow* dapat mengetahui bagian mana yang harus dipindahkan ke *GPU*.
4. Skalabilitas komputasi yang tinggi pada keseluruhan mesin terhadap kumpulan data yang besar.

2.6 Deep Learning

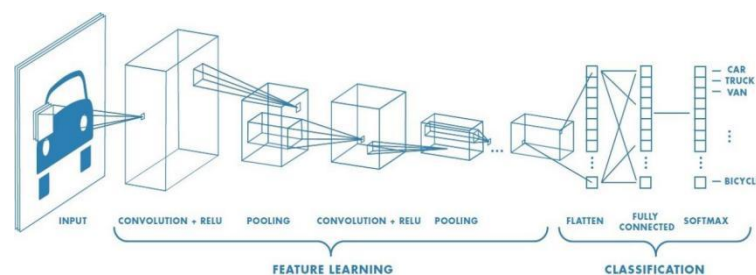
Deep Learning merupakan salah satu bidang dari *Machine Learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik *Deep Learning* memberikan arsitektur yang sangat kuat untuk *Supervised Learning*. Dengan menambahkan lebih banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik. Pada *Machine Learning* terdapat teknik untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi.

2.7 Arsitektur Neural Network

Arsitektur *Neural Network* menggunakan metode *Convolutional Neural Network* sebagai jalur *Artificial Neural Network* (ANN). *Artificial Neural Network* merupakan inti dari proses *Deep Learning* yang 3 sangat powerful untuk melakukan pengklasifikasian citra dengan kompleksitas yang tinggi (Boehmke and Greenwell, 2019). Pada *Deep Learning* dilakukan proses pelatihan dengan dataset citra dan mengubah nilai *pixel* suatu citra dimana proses klasifikasi dapat mendeteksi *patern* dari *input* citra (Chauhan and Ram, 2018).

2.8 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan pengembangan paling populer digunakan dari *Artificial Neural Network* (ANN), CNN umumnya digunakan untuk mengembangkan *neural network* dari data yang bersifat citra/gambar. Arsitektur CNN terdiri dari kumpulan *layer-layer* yang akan dilewati input citra dalam proses *training*. Dua layer pertama (*Convolutional Layer*, *Pooling Layer*) ialah *Convolution Layers* dan diikuti oleh *MaxPooling Layer*, Layer ketiga dan keempat ialah *Fully Connected Layer*, dan Layer terakhir ialah *output layer* (Lin and Shen, 2018). Gambar 2 menampilkan Arsitektur CNN secara umum:

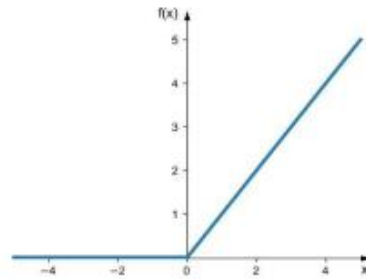


Gambar 2. Arsitektur CNN

Keunggulan menggunakan metode CNN:

1. Akurasi sangat tinggi dalam masalah pengenalan gambar.
2. Secara otomatis mendeteksi fitur penting tanpa pengawasan manusia.
3. Pembagian bobot.

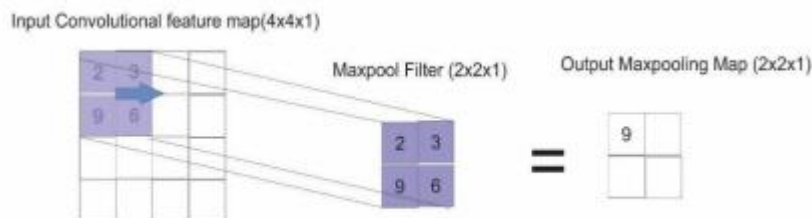
Dimana fungsi akan bernilai x jika nilai x lebih dari atau sama dengan 0, dan fungsi akan bernilai 0 jika nilai x bernilai kurang dari 0. Gambar 4 merupakan tampilan grafik fungsi aktivasi *ReLU*:



Gambar 4. Grafik Fungsi Aktivasi *ReLU*

2.11 Max Pooling Layer

Max Pooling Layer adalah sebuah *layer* lanjutan yang menerima input matriks dari *Convolutional Layer* yang pada *layer* ini dilakukan proses *subsampling* atau *pooling operation*. *Pooling operation* diterapkan pada tiap *feature map* hasil *Convolutional Layer* yang telah di aktivasi dengan fungsi aktivasi *ReLU*. Cara *Max Pooling Layer* memiliki kesamaan terhadap Kernel Filter pada *Convolutional Layer*, *Max Pooling Layer* akan menerapkan Kernel matriks baru pada *feature map* kemudian akan memilih nilai tertinggi dari setiap Kernel matriks melewati *feature map (Shift)*. Sehingga diakhir proses *Max Pooling* akan menghasilkan dimensi *feature map* yang lebih kecil dari sebelumnya. Gambar 5 menggambarkan proses *Max pooling* pada hasil *feature map* dari *convolutional layer*.



Gambar 5. Proses Max Pooling Pada Feature Map

2.12 Fully Connected Layer

Fully connected layer menerima hasil dari *Max Pooling Layer* yang sebelumnya telah dilakukan proses *flatten*. Proses *Flatten* ialah proses meratakan tiap *feature map* yang berbentuk *multidimensional array* menjadi *vector* n -dimensi, n disini berarti total kelas output yang harus dipilih program. Sebuah *fully connected layer* terdiri dari banyak *layer*, dan tiap *layer* memiliki banyak neuron yang melakukan proses perkalian antara input matriks dan bobot masing-masing input. Proses perkalian diformulasikan seperti formula (3):

$$u_i = f\left(\sum_{j=1}^N w_{ij}x_j + b_i\right) \quad (3)$$

Dimana hasil fungsi aktivasi dari nilai perkalian input matrix dan bobot. Dan hasilnya di jumlahkan dengan bias yang ada.

2.13 Softmax

Softmax adalah langkah akhir dari rangkaian Arsitektur CNN, Softmax berfungsi sebagai mengembalikan nilai probabilitas terbesar ke dalam kelas-kelas yang tepat.

Softmax function adalah perhitungan kemungkinan dari masing-masing kelas target atau semua kelas target yang memungkinkan dan membantu untuk menentukan kelas target untuk input yang diberikan. Softmax mempunyai keuntungan, seperti nilai rentang probabilitas yang dihasilkan dari 0 hingga 1, dan jumlah semua kemungkinan sama dengan satu. Ketika softmax digunakan untuk model klasifikasi multi, maka akan mengembalikan peluang dari masing-masing kelas dan kelas target akan memiliki probabilitas lebih tinggi dari kelas yang lain. Perhitungan softmax menggunakan eksponensial dari nilai masukan yang diberikan dan sum dari nilai eksponensial dari semua nilai dalam masukan (Sofia, 2018).

2.14 Teachable Machine

Teachable Machine adalah aplikasi berbasis web yang mampu membantu membuat *machine learning* model secara cepat, mudah dan dapat diakses oleh semua orang tanpa perlu ahli di bidang *machine learning* / kecerdasan buatan.

Kecerdasan buatan yang disediakan oleh *google* ini menggunakan sistem *learning* untuk menganalisa data tanpa terprogram secara eksplisit. Tujuan dibuatnya adalah untuk memudahkan semua kalangan untuk membuat model *machine learning* nya sendiri.

2.15 Penelitian Terdahulu

Penelitian terdahulu mengenai klasifikasi gambar berbasis android sudah banyak dipublikasikan diantaranya sebagai berikut:

- Nama : Royani Darma Nurfiti, 2018.
- Judul : Implementasi *Deep Learning* Berbasis *Tenserflow* untuk Pengenalan Sidik Jari.
- Deskripsi : Penelitian mengenai “Implementasi *Deep Learning* Berbasis *Tenserflow* untuk Pengenalan Sidik Jari” yang dilakukan oleh Royani Darma Nurfiti. Dataset yang digunakan adalah gambar sidik jari dengan menggunakan metode CNN untuk klasifikasi dan mendeteksi. Dalam penelitian ini ukuran datasetnya 24x24 pixel dengan jumlah gambar sidik jari sebanyak 80 gambar. Tingkat akurasi *training* yang dihasilkan dari penelitian ini yaitu sebesar 100%, yang mana penelitian ini telah berhasil melakukan

peningkatan keakurasian pelatihan yang sangat baik.

Nama : Muhammad Zufar dan Budi Setiyono, 2016.

Judul : “*Convolutional Neural Network (CNN) untuk Pengenalan Wajah secara Real-Time.*”

Deskripsi : Penelitian mengenai “*Convolutional Neural Network (CNN) untuk Pengenalan Wajah secara Real-Time*” yang dilakukan oleh Muhammad Zufar dan Budi Setiyono. Penelitian ini menggunakan *framework Open CV* dengan metode CNN. *Dataset* yang digunakan adalah data gambar yang wajah yang dilakukan secara *real-time*. Penelitian ini menghasilkan sebuah model CNN dengan kedalam 7 layer. *Input layer, convolutional layer C1, pooling layer P2, convolutional layer C3, pooling layer P4, hidden layer H dan output layer F* adalah layer pembangun jaringan pada model konvolusi. ketujuh layer yang dihasilkan telah berhasil mengklasifikasikan *dataset* gambar wajah dengan tingkat akurasi lebih dari 87%.

Nama : Triano Nurhikmat, 2018.

Judul : Implementasi *Deep Learning Untuk Image Classification* Menggunakan Algoritma *Convolutional Neural Network (Cnn)* Pada Citra Wayang Golek

Deskripsi : Penelitian menggunakan metode CNN dengan model Citra Wayang Golek pernah dilakukan oleh (Triano Nurhikmat, 2018). *Dataset* yang digunakan pada penelitian ini adalah data gambar wayang golek yang diambil dari situs pencarian *google*. Sedangkan sampel yang digunakan adalah wayang Cepot, Gatotkaca, dan Semar. Hasil akurasi dari penelitian ini didapatkan nilai akurasi sebesar 93%.

Tabel 1. Tabel Penelitian Terdahulu

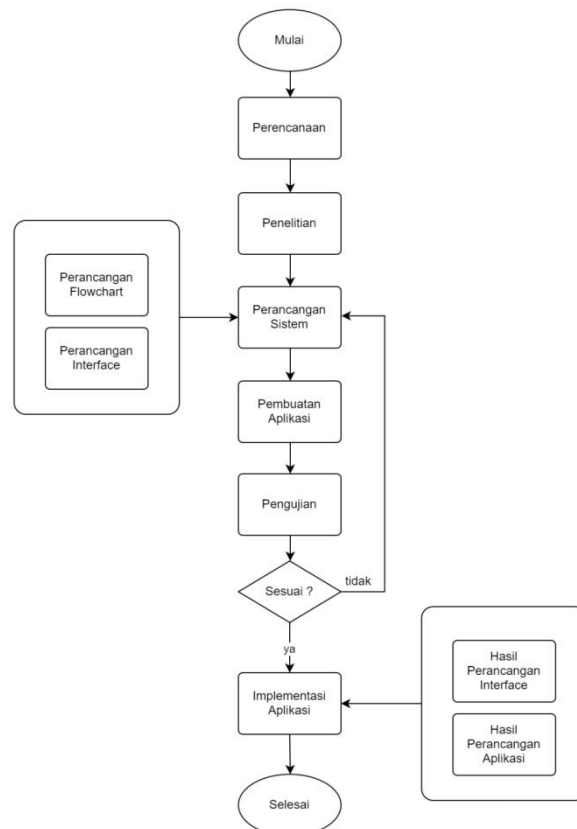
NO	Nama Peneliti	Objek	Metode	Akurasi
			CNN	
1	Royani Darma Nurfita, 2018	Pengenalan Sidik Jari	✓	100%
2	Triano Nurhikmat, 2018	Citra Wayang Golek	✓	93%
3	Muhammad Zufar dan Budi Setiyono, 2016.	Pengenalan Wajah	✓	87%
4	Ikhsan Fadila Hidayat, 2022	Citra Alat Musik Tradisional Jabar	✓	93%

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Tahap penelitian yang dilakukan dari awal hingga akhir digambarkan dalam bentuk alur sebagai berikut:



Gambar 6. Tahapan Penelitian

3.2 Perencanaan

Tahap perencanaan penelitian adalah penentuan sejumlah kegiatan yang akan dilaksanakan pada suatu periode tertentu dalam rangka mencapai tujuan yang ditetapkan. Dalam tahap ini dicari pokok permasalahan dan kebutuhan guna mengidentifikasi data yang dibutuhkan serta menganalisis untuk menentukan batasan masalah terhadap objek penelitian. Adapun kegiatan yang akan dilaksanakan dalam tahap ini yaitu:

- Membuat kerangka awal penelitian
- Estimasi kebutuhan awal penelitian
- Studi kepustakaan
- Penerapan dari aplikasi yang akan dirancang

3.3 Penelitian

Setelah melakukan tahap perencanaan, maka dilanjutkan dengan penelitian awal yang meliputi:

- Literatur yang terkait penelitian
- Mempelajari metode yang digunakan

3.4 Perancangan

Tahap ini berfokus pada penataan system yang akan dibuat berdasarkan prosedur. Ada dua tahap yang harus dilakukan selama proses perancangan system ini, yaitu perancangan struktur program atau membuat *flowchart* dan membuat perancangan desain *interface* pada aplikasi.

Adapun kebutuhan data yang diperlukan untuk keperluan penelitian. Berdasarkan kebutuhan penelitian dibutuhkan beberapa data berupa citra berformat jpg. Data didapatkan dari situs dataset terkemuka *kaggle.com* serta *google image*.

3.5 Pembuatan Aplikasi

Tahap ini merupakan pembuatan aplikasi terhadap rancangan yang telah dibuat. Pembuatan aplikasi dengan bahasa pemrograman *Android*, menggunakan *Library TensorflowLite*, serta menggunakan *Teachable Machine* untuk pembuatan model klasifikasi.

3.6 Pengujian

Pada tahap ini dilakukan pengujian perangkat lunak yang telah dirancang meliputi:

1. Uji Struktural, dilakukan dengan cara membandingkan kesesuaian hasil implementasi dengan rancangan.
2. Uji Fungsional, dilakukan dengan cara menguji setiap button, fungsi atau form yang dibuat, apakah sudah berfungsi dengan baik atau tidak.
3. Uji Validasi, dilakukan dengan cara menguji semua proses data yang ada pada sistem apakah bila dibanding dengan proses manual atau proses lainnya mampu memberikan data yang valid atau tidak.
4. Uji Coba menggunakan model klasifikasi *confusion matrix*.
5. Uji Validasi Kemiripan, dilakukan dengan cara memproses gambar alat musik yang memiliki kemiripan rupa atau bentuk yang berbeda dengan angklung, kecap, dan kendang.

3.7 Implementasi Aplikasi

Tahap ini merupakan tahap akhir dari perancangan sistem, dengan tujuan untuk mengimplementasikan hasil perancangan sistem menjadi sebuah aplikasi *mobile* agar mudah diakses.

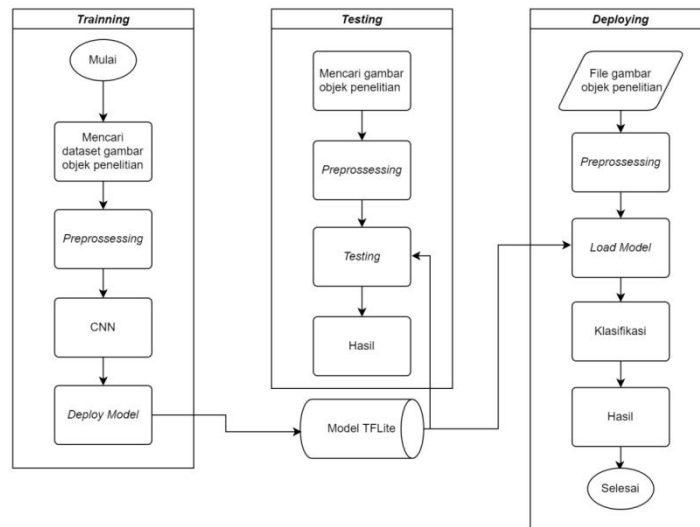
BAB IV

PERANCANGAN DAN IMPLEMENTASI

4.1 Perancangan

Pada tahap kali ini akan dijelaskan alur perancangan *flowchart* sistem yang akan dibuat dan perancangan desain *interface* aplikasi serta analisa kebutuhan data yang diperlukan.

4.1.1 Rancangan *Flowchart* Sistem



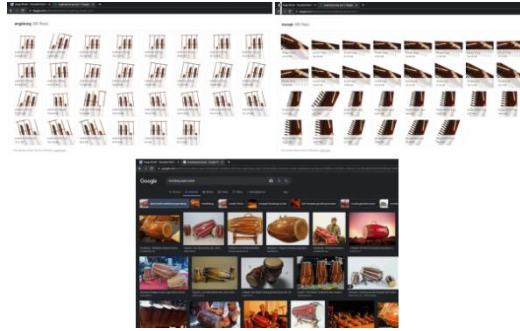
Gambar 7. Flowchart Sistem

Proses *Training* dan *Testing* serta Pembuatan model klasifikasi *TFLite* pada Gambar 7 menggunakan *Teachable Machine* dari *google* dan Proses *Deploying* menggunakan aplikasi *Android Studio*.

Bisa di lihat pada gambar 7 telah di perlihatkan bagaimana *flowchart* sistem tersebut. Berikut ini adalah penjelasan dari *flowchart* tersebut:

1. Proses *Training* (Pelatihan)

- a) Pengumpulan data citra didapatkan dari situs dataset *kaggle.com* serta *google image* untuk dijadikan dataset *training*. Data tersebut berjumlah 300 citra yang dibagi menjadi 3 bagian yaitu alat musik tradisional 100 citra angklung, 100 citra kecapi, 100 citra kendang.



Gambar 8. Dataset *Kaggle.com* dan *Google Image*

- b) Dataset yang telah dikumpulkan dirubah ukuran *pixel* menjadi 224x224px (ukuran *pixel default* pada *TeachableMachine*) lalu disimpan untuk dijadikan data *training*.
- c) Data *training* dimasukkan kedalam arsitektur *neural network* dan dilakukan proses *training*.
- d) Setelah proses *training* selesai maka model yang terbentuk akan di *convert* ke dalam *file tensorflow lite*.
- e) Data *tensorflow lite* yang terbentuk akan diunduh dan disimpan.

2. Proses *Testing* (Pengujian)

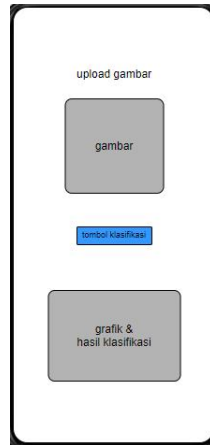
- a) Dataset yang akan digunakan pada proses pengujian berjumlah 100 tiap kelas dan diambil 15 gambar secara acak untuk dilakukan *testing* secara *default* oleh *TeachableMachine* dari *google*.
- b) Model yang telah dilatih pada proses pelatihan akan di *load* sebagai *classifier* yang digunakan untuk mengklasifikasikan objek alat musik tradisional.
- c) Hasil prediksi menunjukkan apakah model bekerja dengan baik atau tidak.

3. Proses *Deploying* (Penyematanan)

- a) Model dari proses pelatihan akan di sematkan pada aplikasi berbasis android.
- b) Model yang telah disimpan akan di *load* sebagai *classifier* yang digunakan untuk mengklasifikasikan objek alat musik tradisional.
- c) Hasil klasifikasi akan menunjukkan nama dari objek yang diuji. Contohnya ketika hasil menunjukkan kecapi maka sistem akan menunjukkan nama & hasil jumlah persen dari kecapi.

4.1.2 Rancangan Desain *Interface*

Tahap ini adalah media perantara antara sistem dengan pengguna. Berikut adalah tampilan rancangan desain *interface*:



Gambar 9. Desain *Interface*

Gambar 9 merupakan rancangan halaman antar muka aplikasi. Pada halaman ini penulis menggunakan fitur deteksi objek secara statis (*upload* foto), tombol yang berfungsi sebagai pemicu untuk proses klasifikasi berlangsung, serta grafik yang akan memunculkan nama objek dan persentase hasil klasifikasi gambar yang di proses.

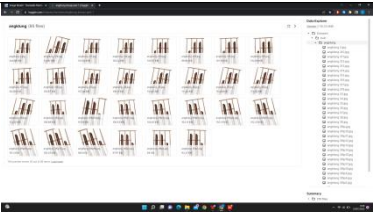
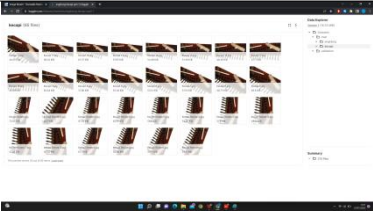
4.1.3 Analisis Kebutuhan Data

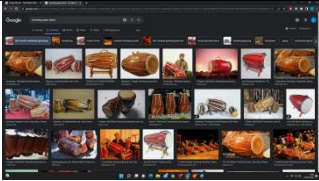
Analisis kebutuhan data adalah tahap yang diperlukan untuk keperluan penelitian. Berdasarkan penelitian ini maka dibutuhkan beberapa data berupa citra atau gambar yang dapat berformat jpg, jpeg, png. Data di dapatkan dari situs dataset terkemuka *kaggle.com* serta *google image*.

Data tersebut dibagi menjadi 3 bagian yaitu angklung, kecapi, kendang. Berikut ini:

1. Data *Training*, Berfungsi untuk membuat prediksi. Berikut tabel dari data *training* yang dikumpulkan.


Tabel 2. Tabel Data *Training*

Jenis Gambar	Jumlah Data	Gambar
Angklung	100	
Kecapi	100	

Kendang	100	
---------	-----	------------------------------------------------------------------------------------

2. *Data Testing*, Berfungsi untuk melihat keakuratan data. *Data testing* ini diambil 15 data secara acak dari 100 data *training* oleh sistem di *Teachable Machine* ketika proses pembuatan model klasifikasi berlangsung. Berikut tabel dari data *testing* yang dicoba.

Tabel 3. Tabel Data *Testing*

Jenis Gambar	Jumlah Data	Gambar
Angklung	15	
Kecapi	15	
Kendang	15	

4.2 Pembuatan Aplikasi

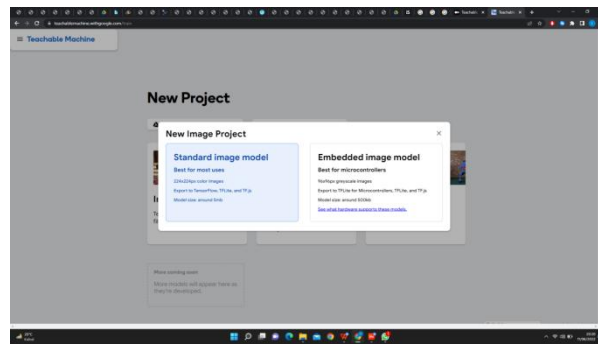
Pada pembahasan ini merupakan gambaran proses-proses pembuatan mulai dari pembuatan model klasifikasi serta pembuatan aplikasi.

4.2.1 Pembuatan Model Klasifikasi

Tahap ini merupakan tahap pembuatan model untuk mengklasifikasi alat musik dengan data yang telah dikumpulkan. Pembuatan model ini menggunakan laptop untuk pembuatannya, setelah jadi model lalu di sematkan kedalam aplikasi android. Berikut ini prosesnya:

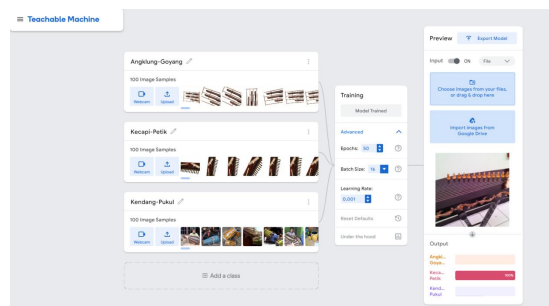
1. Pembuatan model klasifikasi melalui situs teachablemachine.withgoogle.com/train

2. Lalu pilih *image project*, lalu pilih *standard image model*



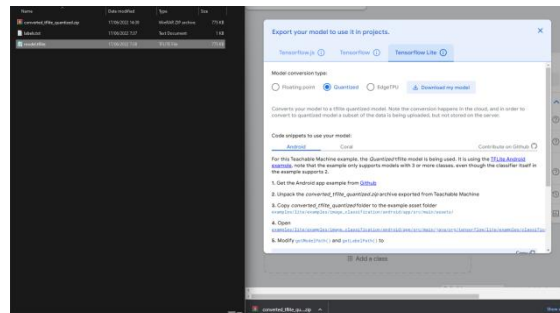
Gambar 10. Pilihan Pembuatan Model Klasifikasi

3. Lalu buat label kelas serta unggah file citra yang telah diperoleh



Gambar 11. Pemberian Label Kelas

4. Tahap selanjutnya adalah mengunduh file model yang sudah dibuat untuk di sematkan kedalam aplikasi mobile.



Gambar 12. File Model Klasifikasi

4.2.2 Pembuatan Aplikasi Mobile

Tahap ini merupakan tahap pembuatan aplikasi dimana semua objek atau bahan aplikasi disatukan. Disini penulis meniru aplikasi yang akan dibuat yang bersumber dari *Channel Youtube Golap Barman* dan *Blog Golap Barman* di situs <https://www.gbandroidblogs.com/2021/07/a-custom-image-classification-android-app-using-tensorflow-lite.html> (di akses pada bulan April 2022).

BAB V

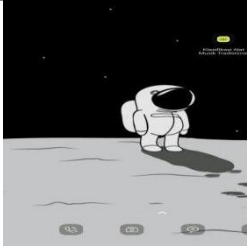
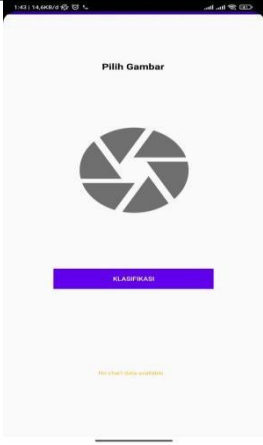

HASIL DAN PEMBAHASAN

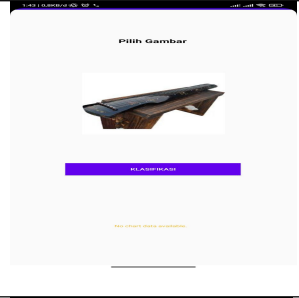
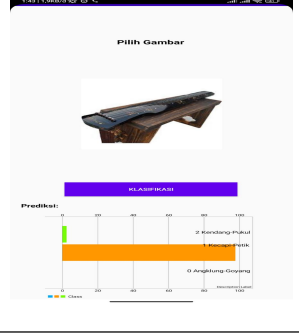
5.1 Hasil

Pada tahap perancangan dan implementasi yang sebelumnya telah dijelaskan mengenai rancangan Klasifikasi Gambar Alat Musik Tradisional Jawa Barat Berbasis *Android* Menggunakan *Library Tensorflow Lite*. Berikut ini adalah hasil beserta pembahasan yang akan digambarkan.

5.1.1 Hasil dari *User Interface*

Tabel 4. Tabel Hasil *User Interface*

Halaman	Tampilan	Keterangan
Ikon Aplikasi		Halaman ikon aplikasi adalah tampilan aplikasi yang telah diinstal diperangkat android.
Tampilan Awal Aplikasi		Pada halaman awal aplikasi terdapat label dan ikon Pilih Gambar untuk memilih gambar yang ingin di klasifikasi, selanjutnya terdapat tombol Klasifikasi untuk memproses klasifikasi gambar yang ingin di klasifikasi, selanjutnya terdapat Grafik untuk mengetahui presentase hasil proses klasifikasi gambar tersebut.
Tampilan Pilih Gambar		Pada halaman ini berisi file gambar yang ingin di pilih untuk proses klasifikasi.

Tampilan Setelah Pilih Gambar		Pada halaman ini tampilan setelah memilih gambar yang ingin di klasifikasi.
Tampilan Hasil Klasifikasi		Pada halaman ini hasil akhir dari proses klasifikasi gambar.

5.2 Pembahasan

Aplikasi klasifikasi gambar alat musik tradisional jawa barat berbasis android menggunakan *library tensorflow lite* merupakan aplikasi yang dapat memindai objek yang datanya telah dilatih dan dibuat sebagai model. Aplikasi ini menggunakan *tensorflow lite* sebagai *library* atau *framework* dalam pembuatannya serta menggunakan *teachable machine* dari *google* untuk proses pembuatan modelnya. Aplikasi ini juga menggunakan algoritma CNN yang sangat tepat digunakan untuk pengklasifikasian gambar dan dapat digunakan juga untuk deteksi gambar.

Perancangan Aplikasi klasifikasi gambar alat musik tradisional jawa barat berbasis android menggunakan *library tensorflow lite* ini dibuat untuk perangkat *mobile* khususnya untuk yang bersistem operasi android, dan dapat dijalankan pada sistem operasi android minimal *nougat 7.0* sampai *red velvet cake 11*.

5.3 Penentuan Parameter Model

Penentuan model terbaik, harus dicari nilai terbaik parameter-parameter dalam model CNN. Parameter yang dimaksud adalah pengaruh jumlah *epoch*, pengaruh data *train*, dan *learning rate*. Tujuan dari penentuan parameter model ini adalah untuk membandingkan model mana yang paling terbaik dengan memperhatikan nilai parameternya.

5.3.1 Pengaruh Jumlah *Epoch*

Epoch adalah ketika seluruh dataset sudah melalui proses *training* pada pembuatan model sampai dikembalikan ke awal dalam satu putaran. Dalam Neural Network satu epoch itu terlalu besar dalam proses pelatihan karena seluruh data diikuti kedalam proses training sehingga akan membutuhkan waktu. Penulis menetapkan jumlah *epoch* yang di uji yaitu 20, 50, 100 *epoch* dengan masing-masing dataset 100 dan 70 gambar untuk 3 alat musik.

5.3.2 Pengaruh Jumlah Data *Train*

Penelitian ini mencoba menggunakan jumlah data *train* yang berbeda. Data *train* yang digunakan dibagi menjadi tiga bagian, yaitu 70, dan 100. Dari masing-masing data *train* sebanyak 11 data gambar untuk 70 dataset, 15 data gambar untuk 100 dataset secara acak yang digunakan untuk proses validasi oleh sistem.

5.3.3 Pengaruh Nilai *Learning Rate*

Penelitian ini juga melakukan uji coba dengan menggunakan nilai *learning rate* sebesar 0.1, 0.001, dan 0.0001. Penentuan nilai dari *learning rate* sangat berpengaruh pada performa akurasi.

A. Tabel Hasil Penentuan Parameter Model Epoch 20

1. *Epoch* 20, Dataset 100 untuk setiap objek, *Learning Rate* 0,0001

Akurasi	Confusion Matrix	Keterangan												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>0.93</td> <td>15</td> </tr> <tr> <td>kecapi-petik</td> <td>0.93</td> <td>15</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>15</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	0.93	15	kecapi-petik	0.93	15	kendang-pukul	1.00	15		<p>Dari 15 sample citra angklung terdeteksi 1 citra kendang.</p> <p>Dari 15 sample citra kecapi terdeteksi 1 citra kendang.</p> <p>Dari 15 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	0.93	15												
kecapi-petik	0.93	15												
kendang-pukul	1.00	15												

2. *Epoch* 20, Dataset 100 untuk setiap objek, *Learning Rate* 0,001

Akurasi	Confusion Matrix	Keterangan												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>1.00</td> <td>15</td> </tr> <tr> <td>kecapi-petik</td> <td>0.87</td> <td>15</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>15</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	1.00	15	kecapi-petik	0.87	15	kendang-pukul	1.00	15		<ul style="list-style-type: none"> ◆ Dari 15 <i>sample</i> citra angklung terdeteksi dengan benar. ◆ Dari 15 <i>sample</i> citra kecapi terdeteksi 2 citra kendang. ◆ Dari 15 <i>sample</i> citra kendang terdeteksi dengan benar.
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	1.00	15												
kecapi-petik	0.87	15												
kendang-pukul	1.00	15												

3. *Epoch 20*, Dataset 100 untuk setiap objek, *Learning Rate 0,01*

Akurasi	Confusion Matrix	Keterangan												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>1.00</td> <td>15</td> </tr> <tr> <td>kecapi-petik</td> <td>1.00</td> <td>15</td> </tr> <tr> <td>kendang-pukul</td> <td>0.93</td> <td>15</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	1.00	15	kecapi-petik	1.00	15	kendang-pukul	0.93	15	<p>Confusion Matrix</p>	<p>Dari 15 sample citra angklung terdeteksi dengan benar.</p> <p>Dari 15 sample citra kecapi terdeteksi dengan benar.</p> <p>Dari 15 sample citra kendang terdeteksi 1 citra kecapi.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	1.00	15												
kecapi-petik	1.00	15												
kendang-pukul	0.93	15												

4. *Epoch 20*, Dataset 70 untuk setiap objek, *Learning Rate 0,0001*

Akurasi	Confusion Matrix	Keterangan												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>0.91</td> <td>11</td> </tr> <tr> <td>kecapi-petik</td> <td>0.91</td> <td>11</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>11</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	0.91	11	kecapi-petik	0.91	11	kendang-pukul	1.00	11	<p>Confusion Matrix</p>	<p>Dari 11 sample citra angklung terdeteksi 1 citra kendang.</p> <p>Dari 11 sample citra kecapi terdeteksi 1 citra kendang.</p> <p>Dari 11 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	0.91	11												
kecapi-petik	0.91	11												
kendang-pukul	1.00	11												

5. *Epoch 20*, Dataset 70 untuk setiap objek, *Learning Rate 0,001*

Akurasi	Confusion Matrix	Keterangan												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>1.00</td> <td>11</td> </tr> <tr> <td>kecapi-petik</td> <td>1.00</td> <td>11</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>11</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	1.00	11	kecapi-petik	1.00	11	kendang-pukul	1.00	11	<p>Confusion Matrix</p>	<p>Dari 11 sample citra angklung terdeteksi dengan benar.</p> <p>Dari 11 sample citra kecapi terdeteksi dengan benar.</p> <p>Dari 11 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	1.00	11												
kecapi-petik	1.00	11												
kendang-pukul	1.00	11												

6. *Epoch 20*, Dataset 70 untuk setiap objek, *Learning Rate 0,01*

Akurasi	Confusion Matrix	Keterangan												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>0.91</td> <td>11</td> </tr> <tr> <td>kecapi-petik</td> <td>0.82</td> <td>11</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>11</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	0.91	11	kecapi-petik	0.82	11	kendang-pukul	1.00	11	<p>Confusion Matrix</p>	<p>Dari 11 sample citra angklung terdeteksi 1 citra kendang.</p> <p>Dari 11 sample citra kecapi terdeteksi 2 citra kendang.</p> <p>Dari 11 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	0.91	11												
kecapi-petik	0.82	11												
kendang-pukul	1.00	11												

B. Tabel Hasil Penentuan Parameter Model Epoch 50

1. Epoch 50, Dataset 100 untuk setiap objek, Learning Rate 0,0001

Akurasi	Confusion Matrix	Keterangan												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>0.80</td> <td>15</td> </tr> <tr> <td>kecapi-petik</td> <td>1.00</td> <td>15</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>15</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	0.80	15	kecapi-petik	1.00	15	kendang-pukul	1.00	15		<p>Dari 15 sample citra angklung terdeteksi 3 citra kendang.</p> <p>Dari 15 sample citra kecapi terdeteksi dengan benar.</p> <p>Dari 15 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	0.80	15												
kecapi-petik	1.00	15												
kendang-pukul	1.00	15												

2. Epoch 50, Dataset 100 untuk setiap objek, Learning Rate 0,001

Akurasi	Confusion Matrix	Keterangan												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>1.00</td> <td>15</td> </tr> <tr> <td>kecapi-petik</td> <td>1.00</td> <td>15</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>15</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	1.00	15	kecapi-petik	1.00	15	kendang-pukul	1.00	15		<p>Dari 15 sample citra angklung terdeteksi dengan benar.</p> <p>Dari 15 sample citra kecapi terdeteksi dengan benar.</p> <p>Dari 15 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	1.00	15												
kecapi-petik	1.00	15												
kendang-pukul	1.00	15												

3. Epoch 50, Dataset 100 untuk setiap objek, Learning Rate 0,01

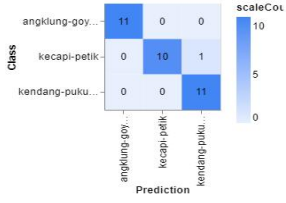
Akurasi	Confusion Matrix	Keterangan												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>1.00</td> <td>15</td> </tr> <tr> <td>kecapi-petik</td> <td>0.93</td> <td>15</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>15</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	1.00	15	kecapi-petik	0.93	15	kendang-pukul	1.00	15		<p>Dari 15 sample citra angklung terdeteksi dengan benar.</p> <p>Dari 15 sample citra kecapi terdeteksi 1 citra kendang.</p> <p>Dari 15 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	1.00	15												
kecapi-petik	0.93	15												
kendang-pukul	1.00	15												

4. Epoch 50, Dataset 70 untuk setiap objek, Learning Rate 0,0001

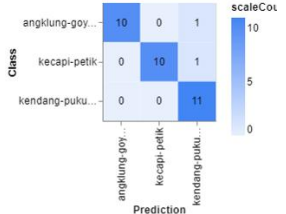
Akurasi	Confusion Matrix	Keterangan												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>0.91</td> <td>11</td> </tr> <tr> <td>kecapi-petik</td> <td>0.91</td> <td>11</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>11</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	0.91	11	kecapi-petik	0.91	11	kendang-pukul	1.00	11		<p>Dari 11 sample citra angklung terdeteksi 1 citra kecapi.</p> <p>Dari 11 sample citra kecapi terdeteksi 1 citra kendang.</p> <p>Dari 11 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	0.91	11												
kecapi-petik	0.91	11												
kendang-pukul	1.00	11												

5. Epoch 50, Dataset 70 untuk setiap objek, Learning Rate 0,001

Akurasi	Confusion Matrix	Keterangan

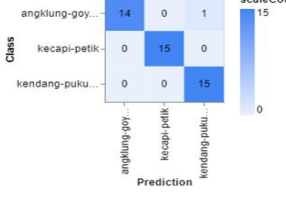
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>1.00</td> <td>11</td> </tr> <tr> <td>kecapi-petik</td> <td>0.91</td> <td>11</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>11</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	1.00	11	kecapi-petik	0.91	11	kendang-pukul	1.00	11	<p>Confusion Matrix</p> 	<p>Dari 11 sample citra angklung terdeteksi dengan benar.</p> <p>Dari 11 sample citra kecapi terdeteksi 1 citra kendang.</p> <p>Dari 11 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	1.00	11												
kecapi-petik	0.91	11												
kendang-pukul	1.00	11												

6. Epoch 50, Dataset 70 untuk setiap objek, Learning Rate 0,01

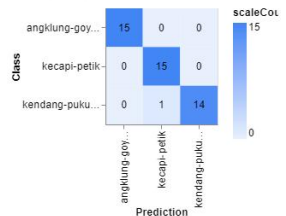
<p>Akurasi</p>	<p>Confusion Matrix</p>	<p>Keterangan</p>												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>0.91</td> <td>11</td> </tr> <tr> <td>kecapi-petik</td> <td>0.91</td> <td>11</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>11</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	0.91	11	kecapi-petik	0.91	11	kendang-pukul	1.00	11	<p>Confusion Matrix</p> 	<p>Dari 11 sample citra angklung terdeteksi 1 citra kendang.</p> <p>Dari 11 sample citra kecapi terdeteksi 1 citra kendang.</p> <p>Dari 11 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	0.91	11												
kecapi-petik	0.91	11												
kendang-pukul	1.00	11												

C. Tabel Hasil Penentuan Parameter Model Epoch 100

1. Epoch 100, Dataset 100 untuk setiap objek, Learning Rate 0,0001

<p>Akurasi</p>	<p>Confusion Matrix</p>	<p>Keterangan</p>												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>0.93</td> <td>15</td> </tr> <tr> <td>kecapi-petik</td> <td>1.00</td> <td>15</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>15</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	0.93	15	kecapi-petik	1.00	15	kendang-pukul	1.00	15	<p>Confusion Matrix</p> 	<p>Dari 15 sample citra angklung terdeteksi 1 citra kendang.</p> <p>Dari 15 sample citra kecapi terdeteksi dengan benar.</p> <p>Dari 15 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	0.93	15												
kecapi-petik	1.00	15												
kendang-pukul	1.00	15												

2. Epoch 100, Dataset 100 untuk setiap objek, Learning Rate 0,001

<p>Akurasi</p>	<p>Confusion Matrix</p>	<p>Keterangan</p>												
<p>Accuracy per class</p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>1.00</td> <td>15</td> </tr> <tr> <td>kecapi-petik</td> <td>1.00</td> <td>15</td> </tr> <tr> <td>kendang-pukul</td> <td>0.93</td> <td>15</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	1.00	15	kecapi-petik	1.00	15	kendang-pukul	0.93	15	<p>Confusion Matrix</p> 	<p>Dari 15 sample citra angklung terdeteksi dengan benar.</p> <p>Dari 15 sample citra kecapi terdeteksi dengan benar.</p> <p>Dari 15 sample citra kendang terdeteksi 1 citra kecapi.</p>
CLASS	ACCURACY	# SAMPLES												
angklung-goyang	1.00	15												
kecapi-petik	1.00	15												
kendang-pukul	0.93	15												

3. *Epoch* 100, Dataset 100 untuk setiap objek, *Learning Rate* 0,01

Akurasi	Confusion Matrix	Keterangan																												
<p><i>Accuracy per class</i></p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>1.00</td> <td>15</td> </tr> <tr> <td>kecapi-petik</td> <td>0.93</td> <td>15</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>15</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	1.00	15	kecapi-petik	0.93	15	kendang-pukul	1.00	15	<p><i>Confusion Matrix</i></p> <table border="1"> <thead> <tr> <th>Class</th> <th>angklung-goy...</th> <th>kecapi-petik</th> <th>kendang-puku...</th> </tr> </thead> <tbody> <tr> <th>angklung-goy...</th> <td>15</td> <td>0</td> <td>0</td> </tr> <tr> <th>kecapi-petik</th> <td>0</td> <td>14</td> <td>1</td> </tr> <tr> <th>kendang-puku...</th> <td>0</td> <td>0</td> <td>15</td> </tr> </tbody> </table>	Class	angklung-goy...	kecapi-petik	kendang-puku...	angklung-goy...	15	0	0	kecapi-petik	0	14	1	kendang-puku...	0	0	15	<p>Dari 15 sample citra angklung terdeteksi dengan benar.</p> <p>Dari 15 sample citra kecapi terdeteksi 1 citra kendang.</p> <p>Dari 15 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES																												
angklung-goyang	1.00	15																												
kecapi-petik	0.93	15																												
kendang-pukul	1.00	15																												
Class	angklung-goy...	kecapi-petik	kendang-puku...																											
angklung-goy...	15	0	0																											
kecapi-petik	0	14	1																											
kendang-puku...	0	0	15																											

4. *Epoch* 100, Dataset 70 untuk setiap objek, *Learning Rate* 0,0001

Akurasi	Confusion Matrix	Keterangan																												
<p><i>Accuracy per class</i></p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>0.82</td> <td>11</td> </tr> <tr> <td>kecapi-petik</td> <td>1.00</td> <td>11</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>11</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	0.82	11	kecapi-petik	1.00	11	kendang-pukul	1.00	11	<p><i>Confusion Matrix</i></p> <table border="1"> <thead> <tr> <th>Class</th> <th>angklung-goy...</th> <th>kecapi-petik</th> <th>kendang-puku...</th> </tr> </thead> <tbody> <tr> <th>angklung-goy...</th> <td>9</td> <td>0</td> <td>2</td> </tr> <tr> <th>kecapi-petik</th> <td>0</td> <td>11</td> <td>0</td> </tr> <tr> <th>kendang-puku...</th> <td>0</td> <td>0</td> <td>11</td> </tr> </tbody> </table>	Class	angklung-goy...	kecapi-petik	kendang-puku...	angklung-goy...	9	0	2	kecapi-petik	0	11	0	kendang-puku...	0	0	11	<p>Dari 11 sample citra angklung terdeteksi 2 citra kendang.</p> <p>Dari 11 sample citra kecapi terdeteksi dengan benar.</p> <p>Dari 11 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES																												
angklung-goyang	0.82	11																												
kecapi-petik	1.00	11																												
kendang-pukul	1.00	11																												
Class	angklung-goy...	kecapi-petik	kendang-puku...																											
angklung-goy...	9	0	2																											
kecapi-petik	0	11	0																											
kendang-puku...	0	0	11																											

5. *Epoch* 100, Dataset 70 untuk setiap objek, *Learning Rate* 0,001

Akurasi	Confusion Matrix	Keterangan																												
<p><i>Accuracy per class</i></p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>1.00</td> <td>11</td> </tr> <tr> <td>kecapi-petik</td> <td>1.00</td> <td>11</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>11</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	1.00	11	kecapi-petik	1.00	11	kendang-pukul	1.00	11	<p><i>Confusion Matrix</i></p> <table border="1"> <thead> <tr> <th>Class</th> <th>angklung-goy...</th> <th>kecapi-petik</th> <th>kendang-puku...</th> </tr> </thead> <tbody> <tr> <th>angklung-goy...</th> <td>11</td> <td>0</td> <td>0</td> </tr> <tr> <th>kecapi-petik</th> <td>0</td> <td>11</td> <td>0</td> </tr> <tr> <th>kendang-puku...</th> <td>0</td> <td>0</td> <td>11</td> </tr> </tbody> </table>	Class	angklung-goy...	kecapi-petik	kendang-puku...	angklung-goy...	11	0	0	kecapi-petik	0	11	0	kendang-puku...	0	0	11	<p>Dari 11 sample citra angklung terdeteksi dengan benar.</p> <p>Dari 11 sample citra kecapi terdeteksi dengan benar.</p> <p>Dari 11 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES																												
angklung-goyang	1.00	11																												
kecapi-petik	1.00	11																												
kendang-pukul	1.00	11																												
Class	angklung-goy...	kecapi-petik	kendang-puku...																											
angklung-goy...	11	0	0																											
kecapi-petik	0	11	0																											
kendang-puku...	0	0	11																											

6. *Epoch* 100, Dataset 70 untuk setiap objek, *Learning Rate* 0,01

Akurasi	Confusion Matrix	Keterangan																												
<p><i>Accuracy per class</i></p> <table border="1"> <thead> <tr> <th>CLASS</th> <th>ACCURACY</th> <th># SAMPLES</th> </tr> </thead> <tbody> <tr> <td>angklung-goyang</td> <td>0.91</td> <td>11</td> </tr> <tr> <td>kecapi-petik</td> <td>1.00</td> <td>11</td> </tr> <tr> <td>kendang-pukul</td> <td>1.00</td> <td>11</td> </tr> </tbody> </table>	CLASS	ACCURACY	# SAMPLES	angklung-goyang	0.91	11	kecapi-petik	1.00	11	kendang-pukul	1.00	11	<p><i>Confusion Matrix</i></p> <table border="1"> <thead> <tr> <th>Class</th> <th>angklung-goy...</th> <th>kecapi-petik</th> <th>kendang-puku...</th> </tr> </thead> <tbody> <tr> <th>angklung-goy...</th> <td>10</td> <td>0</td> <td>1</td> </tr> <tr> <th>kecapi-petik</th> <td>0</td> <td>11</td> <td>0</td> </tr> <tr> <th>kendang-puku...</th> <td>0</td> <td>0</td> <td>11</td> </tr> </tbody> </table>	Class	angklung-goy...	kecapi-petik	kendang-puku...	angklung-goy...	10	0	1	kecapi-petik	0	11	0	kendang-puku...	0	0	11	<p>Dari 11 sample citra angklung terdeteksi 1 citra kendang.</p> <p>Dari 11 sample citra kecapi terdeteksi dengan benar.</p> <p>Dari 11 sample citra kendang terdeteksi dengan benar.</p>
CLASS	ACCURACY	# SAMPLES																												
angklung-goyang	0.91	11																												
kecapi-petik	1.00	11																												
kendang-pukul	1.00	11																												
Class	angklung-goy...	kecapi-petik	kendang-puku...																											
angklung-goy...	10	0	1																											
kecapi-petik	0	11	0																											
kendang-puku...	0	0	11																											

5.3.4 Model Hasil *Training*

Setelah melalui beberapa proses pencarian model penulis menetapkan model yang terbaik dengan jumlah *epoch* 50, data *train* 100, dan *learning rate* 0.001. Berikut hasil pembuatan model yang digunakan:

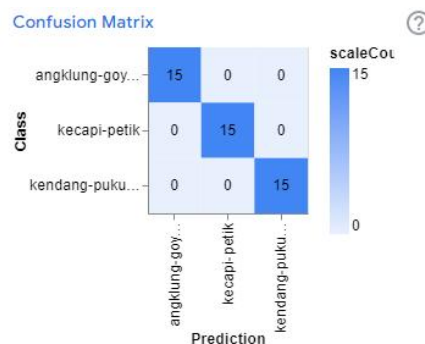
Hasil Akurasi *epoch* 50, data *train* 100, serta *learning rate* 0.001

Accuracy per class ?

CLASS	ACCURACY	# SAMPLES
angklung-goyang	1.00	15
kecapi-petik	1.00	15
kendang-pukul	1.00	15

Gambar 13. Hasil Akurasi

Dapat dilihat dari gambar13 hasil dari pembuatan model diambil *sample* sebanyak 15 secara acak oleh sistem dan mendapatkan hasil seperti di gambar13.



Gambar 14. *Confusion Matrix*

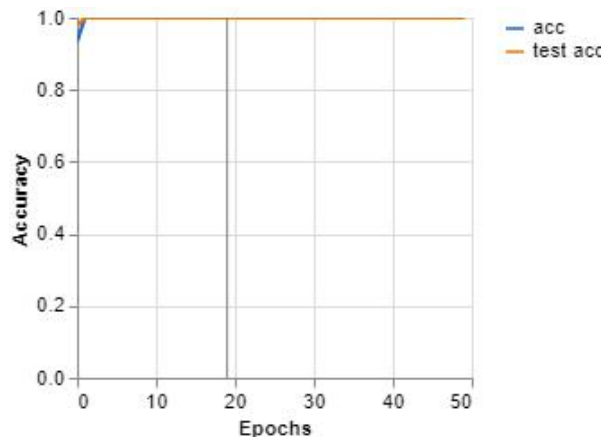
Hasil *confusion matrix* dalam gambar12 mengatakan:

1. Dari 15 *sample* citra angklung terdeteksi dengan benar.
2. Dari 15 *sample* citra kecapi terdeteksi dengan benar.
3. Dari 15 *sample* citra kendang terdeteksi dengan benar.

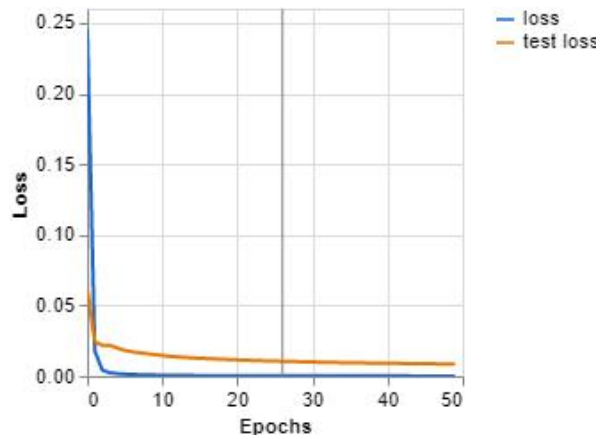
Setelah selesai pembuatan model di dapatkan hasil tingkat *accuracy* dan *loss* model dari data *train* dan *validation*. Berikut tabel hasil dari pelatihan dan *testing* beserta grafiknya :

Tabel 5. Tabel Hasil *Train* Dan *Validation*

Epoch	Data Train		Data Test		Epoch	Data Train		Data Test	
	Acc	Loss	Acc	Loss		Acc	Loss	Acc	Loss
1	0.9215	0.2387	0.9333	0.1795	26	0.9999	0.0002	0.9555	0.0981
2	0.9843	0.0402	0.9333	0.1105	27	0.9999	0.0002	0.9555	0.0981
3	0.9999	0.0051	0.9333	0.1433	28	0.9999	0.0002	0.9555	0.0980
4	0.9999	0.0022	0.9555	0.0917	29	0.9999	0.0002	0.9555	0.0979
5	0.9999	0.0014	0.9555	0.0896	30	0.9999	0.0002	0.9555	0.0979
6	0.9999	0.0010	0.9555	0.0955	31	0.9999	0.0001	0.9555	0.0978
7	0.9999	0.0008	0.9555	0.0989	32	0.9999	0.0001	0.9555	0.0978
8	0.9999	0.0007	0.9555	0.0992	33	0.9999	0.0001	0.9555	0.0977
9	0.9999	0.0006	0.9555	0.0987	34	0.9999	0.0001	0.9555	0.0976
10	0.9999	0.0006	0.9555	0.0986	35	0.9999	0.0001	0.9555	0.0976
11	0.9999	0.0005	0.9555	0.0986	36	0.9999	0.0001	0.9555	0.0975
12	0.9999	0.0005	0.9555	0.0986	37	0.9999	0.0001	0.9555	0.0975
13	0.9999	0.0004	0.9555	0.0985	38	0.9999	0.0001	0.9555	0.0975
14	0.9999	0.0004	0.9555	0.0985	39	0.9999	0.0001	0.9555	0.0975
15	0.9999	0.0003	0.9555	0.0984	40	0.9999	0.0001	0.9555	0.0975
16	0.9999	0.0003	0.9555	0.0983	41	0.9999	0.0001	0.9555	0.0974
17	0.9999	0.0003	0.9555	0.0983	42	0.9999	0.0001	0.9555	0.0974
18	0.9999	0.0003	0.9555	0.0983	43	0.9999	0.0001	0.9555	0.0974
19	0.9999	0.0003	0.9555	0.0983	44	0.9999	0.0001	0.9555	0.0973
20	0.9999	0.0002	0.9555	0.0983	45	0.9999	0.0001	0.9555	0.0973
21	0.9999	0.0002	0.9555	0.0983	46	0.9999	0.0001	0.9555	0.0973
22	0.9999	0.0002	0.9555	0.0983	47	0.9999	0.0001	0.9555	0.0972
23	0.9999	0.0002	0.9555	0.0982	48	0.9999	0.0001	0.9555	0.0972
24	0.9999	0.0002	0.9555	0.0982	49	0.9999	0.0001	0.9555	0.0971
25	0.9999	0.0002	0.9555	0.0981	50	0.9999	0.0001	0.9555	0.0971



Gambar 15. Grafik *Accuracy*



Gambar 16. Grafik *Loss*

5.4 Uji coba

Tahap berikutnya adalah uji coba aplikasi dengan cara dijalankan pada *Smartphone Android*. Pengujian dilakukan untuk mengetahui kelemahan dari aplikasi, seperti menu tidak berjalan, tombol tidak berfungsi, dan sebagainya. Dalam uji coba ini terdapat beberapa tahap yang akan dilakukan yaitu:

- a. Uji Coba Struktural
- b. Uji Coba Fungsional
- c. Uji Coba Validasi
- d. Uji Coba Validasi Kemiripan

a. Uji Coba Struktural

Uji coba struktural pada Tabel 6 merupakan tahap untuk pengecekan aplikasi yang dibuat apakah sudah sesuai dengan rancangan yang sebelumnya sudah dibuat.



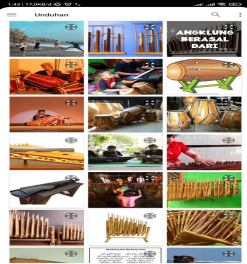
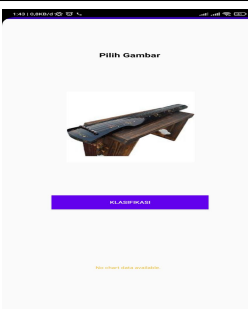
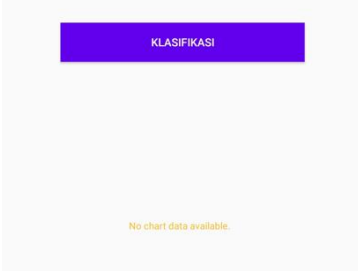

Tabel 6. Tabel Uji Coba Struktural

No	Tampilan Halaman	Hasil
1.	Halaman Awal Aplikasi	Sesuai
2.	Halaman Pilih Gambar	Sesuai
3.	Tampilan Setelah Memilih Gambar	Sesuai
4.	Tampilan Hasil Klasifikasi	Sesuai

b. Uji Coba Fungsional

Uji coba fungsional pada Tabel 7 dilakukan untuk mengetahui apakah sistem yang dibuat sudah berfungsi dengan baik. Pada tahap ini dilakukan percobaan untuk mengetahui apakah fungsi dari setiap tombol atau menu pada halaman dapat berfungsi dengan baik. Berdasarkan uji coba yang telah dilakukan maka secara keseluruhan tombol atau menu pada setiap halaman dapat berfungsi dan berjalan sesuai dengan keinginan. Hasil ini dapat dilihat pada Tabel 7 :

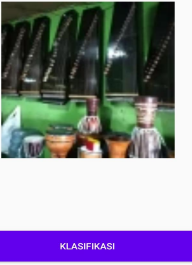
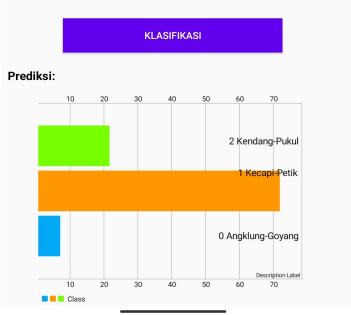
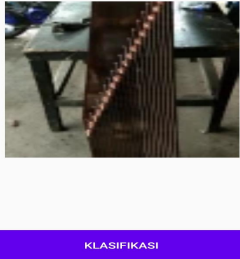

Tabel 7. Tabel Uji Coba Fungsional

No.	Input	Output	Hasil
1.	 <p>Pilih Gambar</p> <p>Tombol Pilih Gambar</p>		Berfungsi
2.	 <p>Pilih Gambar</p>		Berfungsi
3.	 <p>Tombol dan Grafik Hasil Klasifikasi</p>		Berfungsi

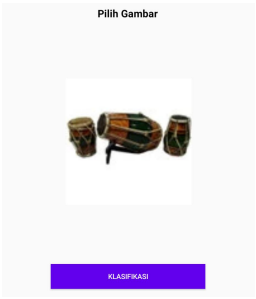
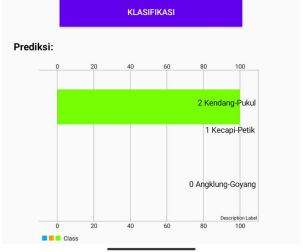
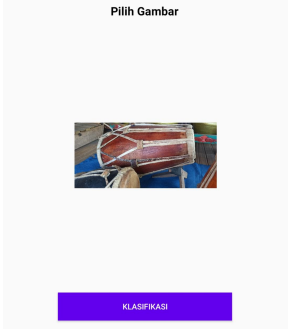
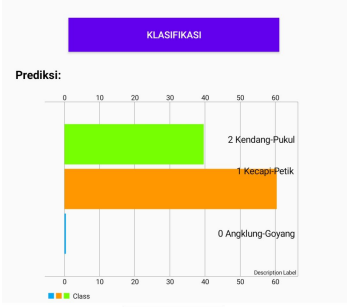
c. Uji Coba Validasi

Uji Validasi adalah tahap dimana merupakan pengecekan data dari suatu objek apakah sudah sesuai dengan data yang sudah di kumpulkan sebelumnya, dengan kata lain, validasi disini untuk memastikan apakah aplikasi yang dibuat sudah sesuai dengan harapan pengguna atau belum. Penulis menggunakan masing-masing lima gambar untuk setiap alat musik yang di uji, jadi total gambar yang di uji ada lima belas. Data yang di gunakan bersumber dari pencarian gambar *google* dan beberapa dari hasil foto penulis sendiri.

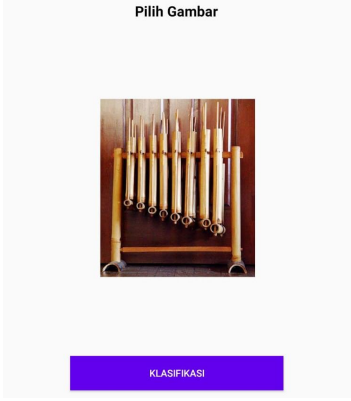
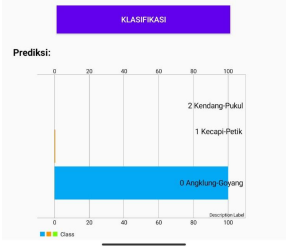
Tabel 8. Tabel Uji Coba Validasi Keakuratan Kecapi


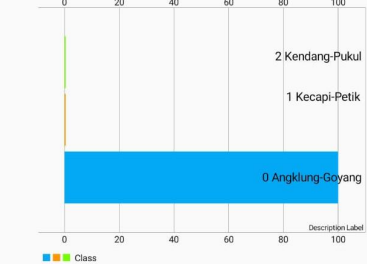
No	Citra	Label	Hasil
1.	<p>Pilih Gambar</p> 	kecapi	 <p>70-75% (Berhasil)</p>
2.	<p>Pilih Gambar</p> 	kecapi	 <p>90-100% (Berhasil)</p>

Tabel 9. Tabel Uji Coba Validasi Keakuratan Kendang

No	Citra	Label	Hasil
1.		kendang	 <p>100% (Berhasil)</p>
2.		kendang	 <p>40% (Gagal) Terklasifikasi Sebagai Kecapi 60%</p>

Tabel 10. Tabel Uji Coba Validasi Keakuratan Angklung

No	Citra	Label	Hasil
1.		angklung	 <p>100% (Berhasil)</p>

2.	<p>Pilih Gambar</p>  <p>KLASIFIKASI</p>	angklung	<p>KLASIFIKASI</p> <p>Prediksi:</p>  <p>100% (Berhasil)</p>
----	--------------------------------------------------------------------------------------------------------------------------	----------	-----------------------------------------------------------------------------------------------------------------------------------------------

Proses Uji Validasi menggunakan data uji sebanyak 6, untuk setiap kelas jenis alat musik tradisional sebanyak 2 gambar.

d. Metode *Confusion Matrix*

Hasil *confusion matrix* adalah sebagai berikut:

Tabel 11. Tabel *Confusion Matrix*

Matriks		Predict Class		
		Angklung	Kecapi	Kendang
Actual Class	Angklung	2	0	0
	Kecapi	0	2	0
	Kendang	0	1	1

Berdasarkan Tabel 11 diatas hasil prediksi dari model terhadap data uji validasi menunjukkan hasil yang baik. Prediksi terhadap alat musik tradisional Angklung di klasifikasikan kedalam Angklung, ini artinya klasifikasi terhadap gambar tersebut benar. Prediksi pada alat musik tradisional Kecapi di klasifikasikan kedalam Kecapi, ini artinya klasifikasi terhadap gambar tersebut benar. Kemudian yang terakhir adalah prediksi pada alat musik tradisional Kendang, terdapat satu gambar Kendang di klasifikasikan sebagai Kecapi. Perhitungan akurasi dari keseluruhan matriks diatas sebagai berikut:

$$\text{Overall Accuracy} = \frac{TTPall}{\text{Total Number of Testing Entires}}$$


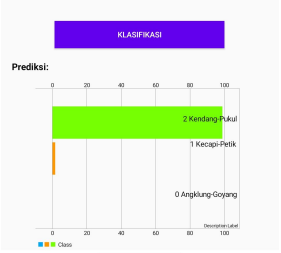

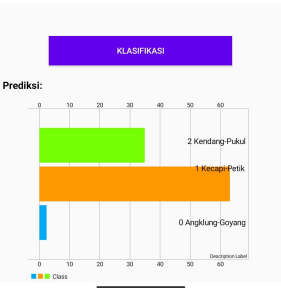
$$\text{Overall Accuracy} = \frac{5}{6} = 83\%$$


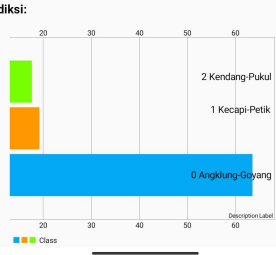

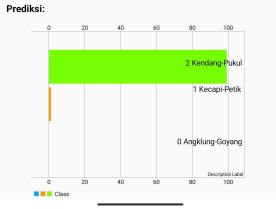

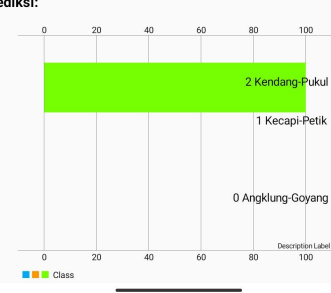
Jadi akurasi yang dihasilkan oleh model dengan input gambar dengan jumlah data uji 15 data didapatkan nilai 83%.




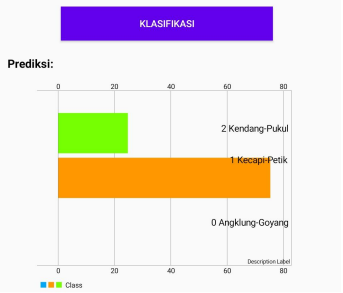

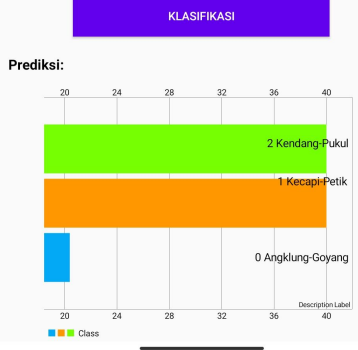
e. Uji Coba Validasi Kemiripan


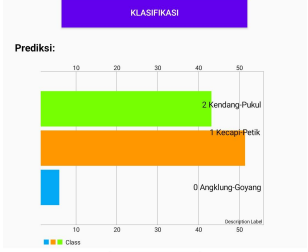
Uji Validasi Kemiripan adalah uji coba menggunakan gambar yang mirip dengan objek yang di gunakan dalam penelitian ini. Objek yang digunakan adalah alat musik calung, celempung yang ditujukan kemiripannya dengan angklung. Harpa, sasando, yang ditujukan kemiripannya dengan kecapi. Rebana, gong, jimbe yang ditujukan kemiripannya dengan kendang. Serta alat musik Piano dan Saxophone yang tidak memiliki kemiripan dengan angklung, kecapi, kendang. Uji kemiripan ini di uji coba tanpa dibuatkan model untuk gambar yang mempunyai kemiripan. Dan sumber gambar yang di uji bersumber dari pencarian *google image*.

Tabel 12. Tabel Uji Kemiripan

No	Citra	Label	Hasil
1.	<p>Pilih Gambar</p> 	<p>Gong (Kemiripan ditujukan dengan kendang)</p>	 <p>90-100%</p> <p>Terklasifikasi Sebagai Kendang</p>
2.	<p>Pilih Gambar</p> 	<p>Celempung (Kemiripan ditujukan dengan angklung)</p>	 <p>60-70%</p> <p>Terklasifikasi Sebagai Kecapi</p>

<p>3.</p>	<p>Pilih Gambar</p> 	<p>Calung (Kemiripan ditujukan dengan angklung)</p>	<p>KLASIFIKASI</p> <p>Prediksi:</p>  <p>60-70%</p> <p>Terklasifikasi Sebagai Angklung</p>
<p>4.</p>	<p>Pilih Gambar</p> 	<p>Rebana (Kemiripan ditujukan dengan kendang)</p>	<p>KLASIFIKASI</p> <p>Prediksi:</p>  <p>90-100%</p> <p>Terklasifikasi Sebagai Kendang</p>
<p>5.</p>	<p>Pilih Gambar</p> 	<p>Jimbe (Kemiripan ditujukan dengan kendang)</p>	<p>KLASIFIKASI</p> <p>Prediksi:</p>  <p>100%</p> <p>Terklasifikasi Sebagai Kendang</p>

<p>6.</p>	<p>Pilih Gambar</p> 	<p>Harpa (Kemiripan ditujukan dengan kecapi)</p>	 <p>90-100%</p> <p>Terklasifikasi Sebagai Kecapi</p>
<p>7.</p>	<p>Pilih Gambar</p> 	<p>Sasando (Kemiripan ditujukan dengan kecapi)</p>	 <p>70-80%</p> <p>Terklasifikasi Sebagai Kecapi</p>
<p>8.</p>	<p>Pilih Gambar</p> 	<p>Piano</p>	 <p>40%</p> <p>Terklasifikasi Sebagai Kendang dan Kecapi</p>

9.	<p>Pilih Gambar</p> 	Saxophone	 <p>40-45% Terklasifikasi Sebagai Kendang 50-55% Terklasifikasi Sebagai Kecapi</p>
----	-------------------------------------------------------------------------------------------------------	-----------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Berdasarkan Tabel 12 diatas hasil uji kemiripan gambar alat musik diatas dinyatakan sebagai berikut:

1. Calung yang ditujukan kemiripannya dengan Angklung terklasifikasi sebagai Angklung dengan tingkat kemiripan 60-70%.
2. Celempung yang ditujukan kemiripannya dengan Angklung terklasifikasi sebagai Kecapi dengan tingkat kemiripan 60-70%, dan hanya 0-10% terklasifikasi sebagai Angklung.
3. Harpa yang ditujukan kemiripannya dengan Kecapi terklasifikasi sebagai Kecapi dengan tingkat kemiripan 90-100%.
4. Sasando yang ditujukan kemiripannya dengan Kecapi terklasifikasi sebagai Kecapi dengan tingkat kemiripan 70-80%.
5. Rebana yang ditujukan kemiripannya dengan Gong terklasifikasi sebagai Gong dengan tingkat kemiripan 90-100%.
6. Gong yang ditujukan kemiripannya dengan Gong terklasifikasi sebagai Gong dengan tingkat kemiripan 90-100%.
7. Jimbe yang ditujukan kemiripannya dengan Gong terklasifikasi sebagai Gong dengan tingkat kemiripan 100%.
8. Piano yang tidak ditujukan kemiripannya dengan apapun terklasifikasi sebagai Kendang dan Kecapi dengan tingkat kemiripan 40%.
9. Saxophone yang tidak ditujukan kemiripannya dengan apapun terklasifikasi sebagai Kendang dengan tingkat kemiripan 40-45% dan juga terklasifikasi sebagai Kecapi dengan tingkat kemiripan 50-55%.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Kesimpulan yang dapat diambil dari Klasifikasi Gambar Alat Musik Tradisional Jawa Barat Berbasis *Android* Menggunakan *Library Tensorflow Lite* yaitu aplikasi ini pembuatannya menggunakan software Android Studio, dataset yang bersumber dari situs Kaggle.com serta google image, menggunakan library tensorflow lite, membuat model dengan *teachablemachine* dari google, serta menggunakan algoritma CNN untuk mendeteksi dan mengklasifikasi objek yang diteliti. Manfaat dari penelitian ini mempermudah dalam mendeteksi dan mengklasifikasikan objek yang ingin di implementasikan dalam dunia kecerdasan buatan bagi orang awam, sehingga siapapun dapat membuat aplikasi kecerdasan buaatannya sendiri.

Aplikasi ini telah melalui tahapan proses uji coba diantaranya uji coba struktural pada setiap halamannya, uji coba fungsional yang dilakukan pada setiap tombol yang ada dalam aplikasi, dan uji coba validasi yang dilakukan pada aplikasi ketika aplikasi berjalan, serta uji kemiripan dengan objek yang memiliki kemiripan dengan angkung, kecapi, kendang, serta ditambah dengan objek yang tidak memiliki kemiripan sama sekali dengan angklung, kecapi, kendang.

Hasil uji coba ini juga berdampak pada pengguna sehingga pengguna dapat dengan mudah menggunakan atau mengakses aplikasi hanya dengan memanfaatkan *smartphone android*. Berdasarkan hasil uji coba validasi diketahui dari 15 data yang di uji terdapat satu data yang gagal terklasifikasi, setelah dilakukan perhitungan metode *confolusion matrix* didapatkan ke akuratan aplikasi mendapatkan nilai 93%.

6.2 Saran

Pembuatan aplikasi pemanfaatan teknologi kecerdasan buatan pada sistem ini diharapkan dapat digunakan dan dimanfaatkan dengan baik. Untuk pengembangan berikutnya dapat ditambahkan fitur *real-time* menggunakan kamera serta dapat mendeteksi dan mengklasifikasi objek lebih dari satu secara bersamaan dengan keakuratan yang tinggi untuk kepentingan dunia kecerdasan buatan.

DAFTAR PUSTAKA

- Deng, L., & Yu, D. (2013).** Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4), 197–387.
<https://doi.org/10.1561/20000000039>
- Dewa, C. K., Fadhilah, A. L., & Afiahayati, A. (2018).** Convolutional Neural Networks for Handwritten Javanese Character Recognition. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 12(1), 83.
<https://doi.org/10.22146/ijccs.31144>
- Dobiki, J. (2018).** Analisis Ketersediaan Prasarana Persampahan di pulau Kumo dan pulau Kakara di Kabupaten Halmahera Utara. *Jurnal Spasial*, 5, 220-228.
- Fitriati, D. (2016).** Perbandingan Kinerja CNN LeNet 5 Dan Extreme Learning Machine Pada Pengenalan Citra Tulisan Tangan Angka. *Jurnal Teknologi Terpadu*, 2(1), 10–16.
- Gonzalez, R. C., Woods, R. E., & Masters, B. R. (2009).** Digital Image Processing, Third Edition. *Journal of Biomedical Optics*, 14(2), 029901.
<https://doi.org/10.1117/1.3115362>
- Goodfellow, I. (2014).** Front Matter. *Linear Algebra*, i–ii.
<https://doi.org/10.1016/b978-0-12-391420-0.09987-x>
- Hendriyana, & Maulana, Y. H. (2020).** Identifikasi Jenis Kayu menggunakan Convolutional Neural Network. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(1), 70–76.
- Hosseini, L., & Ramin Shaghghi Kandovan, 2017.** Hyperspectral Image Classification Based on Hierarchical SVM Algorithm for Improving Overall Accuracy. *Scientific Research Publishing*, 66-75.
- Kristanto, A. (2004).** *Kecerdasan Buatan*. Yogyakarta: Graha Ilmu.
- Kusum, & Kusumadewi, S. (2003).** *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Lecun, Y., Bengio, Y., & Hinton, G. (2015).** Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>

- LINA, Q. (2019, Januari 2).** *Apa itu Convolutional Neural Network?* Retrieved from Medium.com: <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>
- Priowidodo, S. (2019).** *Klasifikasi Gambar Dataset Fashion-Mnist.* 7(1), 34–38.
- Rich, E., & Knight, K. (1991).** *Artificial Intelligence.* New York: McGraw-Hill Inc
- Santoso, B., Azis, A. I., & Zohrahayaty. (2020).** *Machine Learning & Reasoning Fuzzy Logic.* Yogyakarta, DIY, Indonesia: Deepublish Publisher.
- Sena, S. 2018, Mei 27.** Pengenalan Deep Learning Part 1 : Neural Network. Diambil kembali dari Medium: <https://medium.com/@samuelsena/pengenalan-deep-learning8fbb7d8028ac>
- Sena, S. 2018, Mei 27.** Pengenalan Deep Learning Part 7 : Convolutional Neural Network (CNN). Diambil kembali dari Medium: <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>
- Siregar, A. H., Ginting, S., Seminar, P., & Penelitian, N. (n.d.).** *PUPUK ORGANIK MENGGUNAKAN KOMPOSTER.* 1–3.
- Sofia, N. (2018, Jun 9).** *CONVOLUTIONAL NEURAL NETWORK.* Retrieved Jul 2020, from Medium.com: <https://medium.com/@nadhifasofia/1-convolutional-neural-network-convolutional-neural-network-merupakan-salah-satu-metode-machine-28189e17335b#:~:text=Convolutional%20Neural%20Network%20adalah%20salah,banyak%20diimplementasikan%20dalam%20data%20citra.>
- Sri Kusumadewi. (2003).** *Rtificial Ntelligence. Artificial Intelligence (Teknik Dan Aplikasinya.*
- Stephen, Raymond, H. S. (2019).** Aplikasi Convolution Neural Network Untuk Mendeteksi Jenis-Jenis Sampah. *Explore - Jurnal Sistem Informasi Dan Telematika (Telekomunikasi, Multimedia, Dan Informasi),* 10(2), 122–132.
- Sulistya, R., & Murdaningsih, D. (2019, Juli 17).** *Sampah Indonesia Terus Meningkat Tiga Juta Ton Tiap Tahun.* Retrieved from Republika: <https://republika.co.id/berita/pus5ex368/sampah-indonesia-terus-meningkat-tiga-juta-ton-tiap-tahun>

Suyanto, S. M. (2011). *Artificial Intelligence (Edisi Revisi)*. Bandung: Informatika Bandung.

Triano Nurhikmat, 2018. Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (Cnn) Pada Citra Wayang Golek. *Tugas Akhir*

Visalini, S, 2017. Traffic Sign Recognition Using Convolutional Neural Network. *International Jurnal of Innovative Research in Computer and Communication Engineering*, Vol.5.

Wicaksono, A. F. (n.d.). *Tutorial Dasar Tensorflow*™. Retrieved from ui.ac.id:

https://ir.cs.ui.ac.id/alfan/tutorial/tf_intro.html

Yanuar, A. (2018, Juni 10). *Pengenalan Deep Learning*. Retrieved Juli 2020, from

UNIVERSITAS GADJAH MADA MENARA ILMU MACHINE LEARNING: <https://machinelearning.mipa.ugm.ac.id/2018/06/10/pengenalan-deep-learning/>

LAMPIRAN

```
MainActivity.java x
1 package com.example.cartoonclassification;
2
3 import androidx.annotation.Nullable;
4 import androidx.appcompat.app.AppCompatActivity;
5
6 import android.app.Activity;
7 import android.content.Intent;
8 import android.content.res.AssetFileDescriptor;
9 import android.graphics.Bitmap;
10 import android.graphics.Color;
11 import android.net.Uri;
12 import android.os.Bundle;
13 import android.provider.MediaStore;
14 import android.view.View;
15 import android.widget.Button;
16 import android.widget.ImageView;
17 import android.widget.SeekBar;
18 import android.widget.TextView;
19
20 import com.example.cartoonclassification.R;
21 import org.tensorflow.lite.support.tensorbuffer.TensorBuffer;
22 import org.tensorflow.lite.DataType;
23 import org.tensorflow.lite.Interpreter;
24 import org.tensorflow.lite.support.common.FileUtil;
25 import org.tensorflow.lite.support.common.TensorOperator;
26 import org.tensorflow.lite.support.common.TensorProcessor;
27 import org.tensorflow.lite.support.common.ops.NormalizeOp;
28 import org.tensorflow.lite.support.image.ImageProcessor;
29 import org.tensorflow.lite.support.image.TensorImage;
30 import org.tensorflow.lite.support.image.ops.ResizeOp;
31 import org.tensorflow.lite.support.image.ops.ResizeWithCropOrPadOp;
32 import org.tensorflow.lite.support.label.TensorLabel;
33
34
35 import java.io.FileInputStream;
36 import java.io.IOException;
37 import java.nio.MappedByteBuffer;
38 import java.nio.channels.FileChannel;
39 import java.util.ArrayList;
40 import java.util.Collections;
41 import java.util.List;
42 import java.util.Map;
43
44
45 import com.github.mikephil.charting.charts.BarChart;
```

```
MainActivity.java x
46 import com.github.mikephil.charting.charts.HorizontalBarChart;
47 import com.github.mikephil.charting.components.Legend;
48 import com.github.mikephil.charting.components.XAxis;
49 import com.github.mikephil.charting.data.BarData;
50 import com.github.mikephil.charting.data.BarDataSet;
51 import com.github.mikephil.charting.data.BarEntry;
52 import com.github.mikephil.charting.formatter.IndexAxisValueFormatter;
53 import com.github.mikephil.charting.utils.ColorTemplate;
54
55
56
57
58 public class MainActivity extends AppCompatActivity {
59
60     protected Interpreter tflite;
61     private MappedByteBuffer tfliteModel;
62     private TensorImage inputImageBuffer;
63     private int imageSizeX;
64     private int imageSizeY;
65     private TensorBuffer outputProbabilityBuffer;
66     private TensorProcessor probabilityProcessor;
67     private static final float IMAGE_MEAN = 0.0f;
68     private static final float IMAGE_STD = 1.0f;
69     private static final float PROBABILITY_MEAN = 0.0f;
70     private static final float PROBABILITY_STD = 255.0f;
71     private Bitmap bitmap;
72     private List<String> labels;
73     private HorizontalBarChart mBarChart;
74     ImageView imageView;
75     Uri imageuri;
76     Button buclassify;
77
78     TextView prediction;
79
80
81     @Override
82     protected void onCreate(Bundle savedInstanceState) {
83         super.onCreate(savedInstanceState);
84         setContentView(R.layout.activity_main);
85         imageView=(ImageView)findViewById(R.id.image);
86         buclassify=(Button)findViewById(R.id.classify);
87         prediction=(TextView)findViewById(R.id.predictions);
88
89
90
```

```
MainActivity.java x
91
92
93
94 imageView.setOnClickListener((v) -> {
97     Intent intent=new Intent();
98     intent.setType("image/*");
99     intent.setAction(Intent.ACTION_GET_CONTENT);
100     startActivityForResult(Intent.createChooser(intent, title: "Pilih Gambar"), requestCode: 12);
101 });
102
103
104 try{
105     tfLite=new Interpreter(loadModelFile( activity: MainActivity.this));
106 }catch (Exception e) {
107     e.printStackTrace();
108 }
109
110 buclassify.setOnClickListener((v) -> {
113
114     int imageTensorIndex = 0;
115     int[] imageShape = tfLite.getInputTensor(imageTensorIndex).shape(); // {1, height, width, 3}
116     imageSizeY = imageShape[1];
117     imageSizeX = imageShape[2];
118     DataType imageDataType = tfLite.getInputTensor(imageTensorIndex).dataType();
119
120     int probabilityTensorIndex = 0;
121     int[] probabilityShape =
122         tfLite.getOutputTensor(probabilityTensorIndex).shape(); // {1, NUM_CLASSES}
123     DataType probabilityDataType = tfLite.getOutputTensor(probabilityTensorIndex).dataType();
124
125     inputImageBuffer = new TensorImage(imageDataType);
126     outputProbabilityBuffer = TensorBuffer.createFixedSize(probabilityShape, probabilityDataType);
127     probabilityProcessor = new TensorProcessor.Builder().add(getPostprocessNormalizeOp()).build();
128
129     inputImageBuffer = loadImage(bitmap);
130
131     tfLite.run(inputImageBuffer.getBuffer(),outputProbabilityBuffer.getBuffer().rewind());
132     showresult();
133 });
134
135
136
137
138 }
139
140
141
```

```

142
143
144
145
146
147 private TensorImage loadImage(final Bitmap bitmap) {
148     // Loads bitmap into a TensorImage.
149     inputImageBuffer.load(bitmap);
150
151     // Creates processor for the TensorImage.
152     int cropSize = Math.min(bitmap.getWidth(), bitmap.getHeight());
153     // TODO(b/143564309): Fuse ops inside ImageProcessor.
154     ImageProcessor imageProcessor =
155         new ImageProcessor.Builder()
156             .add(new ResizeWithCropOrPadOp(cropSize, cropSize))
157             .add(new ResizeOp(imageSizeX, imageSizeY, ResizeOp.ResizeMethod.NEAREST_NEIGHBOR))
158             .add(getPreprocessNormalizeOp())
159             .build();
160     return imageProcessor.process(inputImageBuffer);
161 }
162
163 @ private MappedByteBuffer loadmodelfile(Activity activity) throws IOException {
164     AssetFileDescriptor fileDescriptor=activity.getAssets().openFd( fileName: "ta_model.tflite");
165     FileInputStream inputStream=new FileInputStream(fileDescriptor.getFileDescriptor());
166     FileChannel fileChannel=inputStream.getChannel();
167     long startoffset = fileDescriptor.getStartOffset();
168     long declaredLength=fileDescriptor.getDeclaredLength();
169     return fileChannel.map(FileChannel.MapMode.READ_ONLY,startoffset,declaredLength);
170 }
171
172 @ private TensorOperator getPreprocessNormalizeOp() {
173     return new NormalizeOp(IMAGE_MEAN, IMAGE_STD);
174 }
175 @ private TensorOperator getPostprocessNormalizeOp(){
176     return new NormalizeOp(PROBABILITY_MEAN, PROBABILITY_STD);
177 }
178
179 @ public static void barchart(BarChart barChart, ArrayList<BarEntry> arrayList, final ArrayList<String> xAxisValues) {
180     barChart.setDrawBarShadow(false);
181     barChart.setFitBars(true);
182     barChart.setDrawValueAboveBar(true);
183     barChart.setMaxVisibleValueCount(25);
184     barChart.setPinchZoom(true);
185
186     barChart.setDrawGridBackground(true);

```

```

MainActivity.java x
187 BarDataSet barDataSet = new BarDataSet(arrayList, label: "Class");
188 barDataSet.setColors(new int[]{Color.parseColor( colorString: "#03A9F4"), Color.parseColor( colorString: "#FF9800"),
189 Color.parseColor( colorString: "#76FF03"), Color.parseColor( colorString: "#E91E63"), Color.parseColor( colorString: "#2962FF")});
190 //barDataSet.setColors(ColorTemplate.COLORFUL_COLORS);
191 BarData barData = new BarData(barDataSet);
192 barData.setBarWidth(0.9f);
193 barData.setValueTextSize(0f);
194
195 barChart.setBackgroundColor(Color.TRANSPARENT); //set whatever color you prefer
196 barChart.setDrawGridBackground(false);
197 barChart.animateY( durationMillis: 2000);
198
199 //Legend l = barChart.getLegend(); // Customize the Ledgens
200 //l.setTextSize(10f);
201 //l.setFormSize(10f);
202 //To set components of x axis
203 XAxis xAxis = barChart.getXAxis();
204 xAxis.setTextSize(13f);
205 xAxis.setPosition(XAxis.XAxisPosition.TOP_INSIDE);
206 xAxis.setValueFormatter(new IndexAxisValueFormatter(xAxisValues));
207 xAxis.setDrawGridLines(false);
208
209 barChart.setData(barData);
210
211 }
212
213
214 private void showresult(){
215
216     try{
217         labels = FileUtil.LoadLabels( context: MainActivity.this, filePath: "ta_labels.txt");
218     }catch (Exception e){
219         e.printStackTrace();
220     }
221     Map<String, Float> labeledProbability =
222         new TensorLabel(labels, probabilityProcessor.process(outputProbabilityBuffer))
223             .getMapWithFloatValue();
224     float maxValueInMap =(Collections.max(labeledProbability.values()));
225
226     for (Map.Entry<String, Float> entry : labeledProbability.entrySet()) {
227         //if (entry.getValue()==maxValueInMap) {
228             String[] label = labeledProbability.keySet().toArray(new String[0]);
229             Float[] label_probability = labeledProbability.values().toArray(new Float[0]);
230
231             mBarChart = findViewById(R.id.chart);

```

```
MainActivity.java x
232     mBarChart.getXAxis().setDrawGridLines(false);
233     mBarChart.getAxisLeft().setDrawGridLines(false);
234     // PREPARING THE ARRAY LIST OF BAR ENTRIES
235     ArrayList<BarEntry> barEntries = new ArrayList<>();
236     for(int i=0; i<label_probability.length; i++)
237     {
238         barEntries.add(new BarEntry(i, y: label_probability[i]*100));
239     }
240
241     // TO ADD THE VALUES IN X-AXIS
242     ArrayList<String> xAxisName = new ArrayList<>();
243     for(int i=0; i<label.length; i++)
244     {
245         xAxisName.add(label[i]);
246     }
247     barChart(mBarChart, barEntries, xAxisName);
248     prediction.setText("Prediksi:");
249
250
251     //     }
252 }
253
254
255 @Override
256 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
257     super.onActivityResult(requestCode, resultCode, data);
258
259     if(requestCode==12 && resultCode==RESULT_OK && data!=null) {
260         imageuri = data.getData();
261         try {
262             bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), imageuri);
263             imageView.setImageBitmap(bitmap);
264         } catch (IOException e) {
265             e.printStackTrace();
266         }
267     }
268 }
269 }
270
271
```




```
activity_main.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context="com.example.cartoonclassification.MainActivity">
8
9
10    <TextView
11        android:id="@+id/classifytext"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:layout_above="@+id/image"
15        android:text="Pilih Gambar"
16        android:textColor="#000000"
17        android:textSize="20dp"
18        android:textStyle="bold"
19        app:layout_constraintBottom_toTopOf="@+id/image"
20        app:layout_constraintEnd_toEndOf="parent"
21        app:layout_constraintHorizontal_bias="0.496"
22        app:layout_constraintStart_toStartOf="parent"
23        app:layout_constraintTop_toTopOf="parent"
24        app:layout_constraintVertical_bias="0.207">
25
26    </TextView>
27
28    <ImageView
29        android:id="@+id/image"
30        android:layout_width="245dp"
31        android:layout_height="250dp"
32        android:layout_centerInParent="true"
33        android:src="@drawable/ic_baseline_camera_24"
34        app:layout_constraintBottom_toTopOf="@+id/classify"
35        app:layout_constraintEnd_toEndOf="parent"
36        app:layout_constraintStart_toStartOf="parent"
37        app:layout_constraintTop_toBottomOf="@id/classifytext"
38        app:layout_constraintVertical_bias="0.5" />
39
40    <Button
41        android:id="@+id/classify"
42        android:layout_width="300dp"
```

```
activity_main.xml x
43     android:layout_height="wrap_content"
44     android:layout_below="@+id/image"
45     android:layout_marginBottom="356dp"
46     android:background="@color/colorPrimary"
47     android:text="Klasifikasi"
48     android:textColor="#ffffff"
49     android:textSize="17sp"
50     app:layout_constraintBottom_toBottomOf="parent"
51     app:layout_constraintEnd_toEndOf="parent"
52     app:layout_constraintHorizontal_bias="0.495"
53     app:layout_constraintStart_toStartOf="parent" />
54
55     <com.github.mikephil.charting.charts.HorizontalBarChart
56     android:id="@+id/chart"
57     android:layout_width="390dp"
58     android:layout_height="300dp"
59     android:layout_marginTop="56dp"
60     app:layout_constraintEnd_toEndOf="parent"
61     app:layout_constraintHorizontal_bias="0.454"
62     app:layout_constraintStart_toStartOf="parent"
63     app:layout_constraintTop_toBottomOf="@+id/classify" />
64
65     <TextView
66     android:id="@+id/predictions"
67     android:layout_width="wrap_content"
68     android:layout_height="wrap_content"
69     android:text=""
70     android:textColor="#000000"
71     android:textSize="20sp"
72     android:textStyle="bold"
73     app:layout_constraintBottom_toTopOf="@+id/chart"
74     app:layout_constraintEnd_toEndOf="parent"
75     app:layout_constraintHorizontal_bias="0.045"
76     app:layout_constraintStart_toStartOf="parent"
77     app:layout_constraintTop_toBottomOf="@+id/classify"
78     app:layout_constraintVertical_bias="0.621" />
79
80 </androidx.constraintlayout.widget.ConstraintLayout>
```

Palette | Pixel | 31 | AppTheme | Default (en-us) | Attributes | Layout Validation

0dp | | | | |


Pilih Gambar



KLASIFIKASI

No internet available

Pilih Gambar



ImageView

KLASIFIKASI

ImageView

Emulator | Device File Explorer

Component Tree | Event Log | Layout Inspector

1:1 LF UTF-8 4 spaces

<< skripsi > gue > aplikasi > ImageClassification > app > src > main > assets

Name	Date modified	Type	Size
ta_labels.txt	17/06/2022 7:37	Text Document	1 KB
ta_model.tflite	17/06/2022 7:38	TFLITE File	775 KB