

SKRIPSI

**PENERAPAN KLASIFIKASI GERAKAN KESEHATAN YOGA
MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK
BERDASARKAN KOREKSI POSTUR MANUSIA**

DISUSUN OLEH

ELZA AHMAD RAIHAN

2009020091



UMSU

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN**

2024

LEMBAR PENGESAHAN

Judul Skripsi : PENERAPAN KLASIFIKASI GERAKAN KESEHATAN YOGA
MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK
BERDASARKAN KOREKSI POSTUR MANUSIA

Nama Mahasiswa : ELZA AHMAD RAIHAN

NPM : 2009020091

Program Studi : TEKNOLOGI INFORMASI

Menyetujui
Komisi Pembimbing



Dr. Al-Khowarizmi, S.Kom., M.Kom
NIDN. 0127099201

Ketua Program Studi



Fatma Sari Hutagalung, S.Kom., M.Kom
NIDN. 0117019301

Dekan



Dr. Al-Khowarizmi, S.Kom., M.Kom
NIDN. 0127099201

PERNYATAAN ORISINALITAS

**PENERAPAN KLASIFIKASI GERAKAN KESEHATAN YOGA
MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK
BERDASARKAN KOREKSI POSTUR MANUSIA**

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Juni 2024

Yang membuat pernyataan



Elza Ahmad Raihan

NPM. 2009020091

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Elza Ahmad Raihan
NPM : 2009020091
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**PENERAPAN KLASIFIKASI GERAKAN KESEHATAN YOGA
MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK
BERDASARKAN KOREKSI POSTUR MANUSIA**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Juni 2024

Yang membuat pernyataan



ELZA AHMAD RAIHAN

NPM. 2009020091

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Elza Ahmad Raihan
Tempat dan Tanggal Lahir : Medan, 01 Agustus 2002
Alamat Rumah : Jln. Prof HM Yamin SH No.134
Telepon/Faks/HP : 081362732432
E-mail : elzaraihan08@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD : SD SWASTA HUSNI THAMRIN MEDAN TAMAT: 2014
SMP : SMPN 29 MEDAN TAMAT: 2017
SMA : SMK NEGERI 1 PERCUT SEI TUAN TAMAT: 2020

KATA PENGANTAR

Alhamdulillah puji dan syukur peneliti panjatkan kehadirat Allah Subhanahu wa Ta'ala atas segala limpahan rahmat dan hidayah-Nya kepada peneliti sehingga dapat menyelesaikan perkuliahan dan penelitian skripsi ini , guna memenuhi salah satu syarat dalam dalam memperoleh gelar Sarjana Komputer pada Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara.

Pada kesempatan ini peneliti mengucapkan terima kasih kepada :

1. Bapak Prof. Dr. Agussani, MAP, selaku Rektor Universitas Muhammadiyah Sumatera Utara.
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom, selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi.
3. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom, dan Bapak Mhd. Basri, S.Si., M.Kom, selaku Ketua Program Studi Teknologi Informasi dan Sekretaris Program Studi Teknologi Informasi.
4. Bapak Dr. Al-Khowarizmi , selaku Dosen Pembimbing dan Mentor yang cukup baik dimana beliau yang telah banyak meluangkan waktunya untuk memberikan bimbingan, arahan dan saran dalam penelitian skripsi ini bahkan beberapa permasalahan kehidupan mahasiswa.
5. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom dan selaku Dosen Penguji yang telah membantu dan memberikan saran dalam pengerjaan skripsi ini dan juga membantu dalam memberikan arahan dukungan , semangat dan juga perhatian yang cukup baik bagi kebutuhan penulis selama awal perkuliahan
6. Ayah kandung penulis yaitu Zulkifli dan Juga Ibu Kandung saya yaitu Elita

Hanum Nasution , yang selalu menjadi Orang tua , partner hingga sahabat diskusi serta canda tawa yang cukup baik dalam kehidupan sehingga perkuliahan ini dapat berjalan dengan baik dimana niat baik dan perjuangan yang cukup panjang akan membuahkan hasil yang baik nantinya.

7. Adik Kandung Penulis yaitu Raisya Elza Nabilla yang dimana menjadi motivasi utama penulis hingga saat ini untuk bisa menjadi panutan yang baik bagi beliau dan adik kecil lainnya
8. Sahabat-sahabat kecil saya yaitu Fadhil Muhammad , Dhafaruddin , Difo Heriandi , Niko Nugraha , Khaira Ummah , Linda Indriyani dan Almarhum Khoirun Nisa yang selalu memotivasi saya untuk menyelesaikan pendidikan sebaik baik mungkin dan selalu mendukung apa yang telah saya jalani selama ini .
9. Sahabat Perjuangan saya yaitu Bima Rifandy Nasution , Hadi Ikhwan dan Rima Yulika yang selalu menjadi tempat terbaik untuk berdiskusi panjang terkait pikiran dan memberikan semangat secara mental dan perjuangan dalam kehidupan dan penyelesain skripsi ini
10. Sahabat KKN yang telah menjadi keluarga kedua saya yaitu ,M Fahri Fahroza , Januar Yuda , Sulis Hafiza , Rio Satya Yuda ,Farizi Erlangga , Perdinal Nasution , Ryan Pratama , Dimas Fadlurohman , Muhammmad Ilham Mufit , Abdillah Syahputra , Ahmad Iqbal Syahputra dan Siti Raisha Rushainy yang terimakasih telah menemani perjalanan perkuliahan menjadi cukup menyenangkan dan mendukung selalu apa yang penulis lakukan dalam kehidupan serta proses perjalanan akademis
11. Teman Perjuangan yang ikut dalam pembangunan Himpunan Mahasiswa

Teknologi Informasi dan juga teman teman satu angkatan Prodi Teknologi Informasi yang selalu memberikan keyakinan dan arahan terkait kebaikan bersama

12. Teman dan Keluarga Besar Core Team GDSC UMSU 2022 – 2023 , 2023 – 2024 yang cukup membantu penulis dalam pembangunan mentalitas penulis dan juga ikut membantu membangun dasar fondasi komunitas teknologi pada Universitas , terkhususnya Siti Raisha Rushainy sebagai Partner Community / MBKM dan kompetisi lainnya dan M Rizky Pratama Siregar yang paling dekat dan berdiskusi selama perjalanan perkuliahan , Terimakasih atas kekompakan dan kebahagiaan yang telah kita jalani selalu.
13. Semua pihak yang terlibat baik secara langsung dan tidak langsung baik dari perkuliahan ataupun kegiatan non kuliah yang telah membangun diri penulis yang dimana penulis tidak bisa menyebutkan satu persatu nama tersebut .
Terimakasih telah hadir

Semoga Allah Subhanahu wa Ta'Ala dapat memberikan balasan atas kebaikan-kebaikan yang telah diberikan kepada peneliti baik di dunia maupun di akhirat kelak. Saya menyadari bahwa skripsi ini masih jauh dari kesempurnaan, oleh karena itu saya mengharapkan saran dan kritikan yang membangun dari pembaca untuk perbaikan-perbaikan dimasa yang akan datang.

Medan, 27 Juli 2024

Peneliti



Elza Ahmad Raihan

ABSTRAK

PENERAPAN KLASIFIKASI GERAKAN KESEHATAN YOGA MENGUNAKAN CONVOLUTIONAL NEURAL NETWORK BERDASARKAN KOREKSI POSTUR MANUSIA

Pola kehidupan pasca pandemi telah mengalami perubahan yang cukup banyak dari penerapan transformasi digital salah satu yang berdampak yaitu pola meditasi seperti praktik yoga yang dapat dilakukan secara mandiri dirumah tanpa harus berinteraksi secara langsung dengan instruktur Penelitian ini juga bertujuan untuk mengembangkan sistem klasifikasi gerakan yoga menggunakan Convolutional Neural Network (CNN) berdasarkan koreksi postur manusia. Dengan memanfaatkan model Movenet , sistem ini dapat mengenali dan mengklasifikasikan berbagai pose yoga untuk memberikan umpan balik yang akurat terkait postur yang benar. Data pelatihan dikumpulkan dari foto yoga dan diproses menjadi gambar pose yang dianalisis menggunakan CNN. Hasil dari penelitian ini menunjukkan bahwa sistem yang dikembangkan mampu mencapai tingkat akurasi yang tinggi dalam mengidentifikasi pose yoga, yang berpotensi membantu pengguna dalam memperbaiki postur dan mengurangi risiko cedera. Sistem ini juga diterapkan dalam aplikasi mobile, sehingga memudahkan pengguna dalam mengakses koreksi postur secara real-time. Dengan demikian, penelitian ini memberikan kontribusi signifikan dalam bidang kesehatan dan teknologi dengan menawarkan solusi inovatif untuk praktik yoga yang lebih aman dan efektif.

Kata Kunci : *Convolutiona Neural Network , Yoga , Mobile App*

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PERNYATAAN PERSETUJUAN PUBLIKASI.....	iv
RIWAYAT HIDUP	v
KATA PENGANTAR.....	vi
ABSTRAK.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
BAB I PENDAHULUAN	1
13.1 Latar Belakang Masalah	1
13.2 Batasan Masalah.....	1
13.3 Rumusan Masalah.....	1
13.4 Tujuan Penelitian.....	1
13.5 Manfaat Penelitian	1
BAB II LANDASAN TEORI.....	7
2.1 Deep Learning	7
2.2 Convolutional Neural Network.....	8
2.2.1 Lapisan Konvolusi.....	8
2.2.2 Lapisan Pooling	9
2.2.1 Lapisan Fully Connected	10
2.3 Movenet.....	11
2.4 TensorFlow.....	12
2.5 Gerakan Kesehatan Yoga.....	13
BAB III METODOLOGI PENELITIAN.....	14
3.1 Metode Penelitian dan Pengembangan Produk	14
3.2 Alur Penelitian.....	16
3.2.1 Studi Literatur.....	16
3.2.2 Pengumpulan Dataset	17
3.2.3 Pelatihan Model Movenet.....	19
3.2.4 Testing dan Evaluasi Model.....	19
3.2.5 Implementasi Model dan Pengujian Performa Model pada Aplikasi.....	20
3.3 Kebutuhan Hardware dan Software.....	20
3.3.1 Kebutuhan Hardware.....	20

3.3.2	Kebutuhan Software	21
BAB IV_HASIL DAN PEMBAHASAN		22
4.1	Rancangan Desain Tampilan Android	22
4.2	Implementasi Kotlin Tampilan Android	25
4.3	Pembangunan Model dan Training Data.....	26
4.4	Implementasi Machine Learning dalam Android	35
BAB V KESIMPULAN DAN SARAN.....		38
5.2	Kesimpulan.....	38
5.2	Saran.....	39
DAFTAR PUSTAKA		40
DAFTAR LAMPIRAN.....		38

DAFTAR TABEL

HALAMAN

Tabel 3. 1 Kebutuhan Hardware

25

DAFTAR GAMBAR

Gambar 2. 1 Konsep Deep Learning	11
Gambar 2. 2 Proses Lapisan Konvolusi	13
Gambar 2. 3 Proses Lapisan Pooling	14
Gambar 2. 4 Proses Lapisan Fully Connected	14
Gambar 2. 5 Arsitektur MoveNet	15
Gambar 3. 1 Alur Penelitian dan Pengembangan	16
Gambar 3. 2 Pose Chair	18
Gambar 3. 3 Pose Cobra	18
Gambar 3. 4 Pose Downdog	18
Gambar 3. 5 Pose Tree	18
Gambar 3. 5 Pose Warrior	18
Gambar 4.1 Tampilan Awal Onboarding	22
Gambar 4.2 Tampilan Login dan Register	23
Gambar 4.3 Tampilan Halaman Pilihan Yoga dan Detail Panduan	24
Gambar 4.4 Implementasi Layout pada Kotlin	25
Gambar 4.5 Tampilan Layout Activity	25
Gambar 4.6 Google Drive Source Google Collab	26
Gambar 4.7 Perintah Instalasi Impor Awal	27
Gambar 4.8 Impor Model Movenet dalam Prediksi Pose	27
Gambar 4.9 Hasil Model Movenet dalam Keypoint Deteksi	28
Gambar 4.10 Pelatihan Data Gambar Label Yoga	29
Gambar 4.11 Testing Data terkait Pose	30
Gambar 4.12 Arsitektur Model yang dihasilkan	30
Gambar 4.13 Hasil Validasi Akurasi	32
Gambar 4.14 Akurasi Model	33
Gambar 4.15 Hasil Laporan Klasifikasi dengan Confusion Matrix	34
Gambar 4.16 Hasil Deteksi Box Pose Webcam	35
Gambar 4.17 Hasil Pose Tree	36
Gambar 4.18 Hasil Pose Warrior	36
Gambar 4.19 Hasil Pose Downdog	36
Gambar 4.20 Hasil Pose Cobra	36
Gambar 4.21 Hasil Pose Chair	36

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan Teknologi terkhususnya dibidang *Artificial Intelligence* atau kecerdasan buatan terkhususnya dibagian *Deep Learning* suatu kemajuan yang sangat dibutuhkan dalam memahami menganalisis dan mengimplementasi pekerjaan manusia dimana konsep pembelajaran mesin tersebut mengadopsi otak manusia Dimana adanya pelatihan data berupa gambar, video, maupun suara sebagai bahan untuk dilatih. Computer Vision sudah berkembang pesat sejak dahulu kala, banyak peneliti yang terus mengembangkannya agar menjadi suatu sistem sempurna. Salah satu metode yang diciptakan yaitu Convolutional Neural Network (CNN) dan akan digunakan dalam perancangan ini di bantu dengan MoveNet sebagai pengenalan struktur gerakan (Karnadi et al., n.d.)

Penyakit kardiovaskular menjadi penyebab kematian Indonesia dimana gaya hidup masyarakat indonesia yang cenderung lebih ke makanan cepat saji , dimana kita tau makanan cepat saji terdapat lemak , protein ,tinggi garam dan juga rendah serat yang menyebabkan banyak permasalahan pada kesehatan masyarakat(Ayuningtyas, 2023)

Kesehatan mental yoga dapat membantu dalam mengelola dan meningkatkan keseimbangan emosional , yang dapat membawa dampak positif terhadap kualitas hidup disamping itu juga Gerakan Kesehatan yoga dapat memberikan dampak positif terhadap Kesehatan Fisik bagi pengguna

yang sering melakukan aktivitas tersebut. Beberapa penelitian, termasuk "*Role of yoga in the prevention and management of various cardiovascular diseases and their risk factors: A comprehensive scientific evidencebased review*"(Mooventhan & Nivethitha, 2020), menemukan bahwa yoga efektif dalam mengurangi risiko faktor penyakit kardiovaskular seperti penyakit jantung, tekanan darah tinggi dan kadar kolesterol disamping itu yoga juga efektif dalam menangani penyakit kardiovaskular tanpa melalui tahapan operasi , disamping itu juga Penelitian seperti "*The effects of yoga on student mental health: a randomisedcontrolled trial*" (Elstad et al., 2020)memberikan hasil bahwa yoga mengurangi masalah emosional seperti masalah tidur di kalangan mahasiswa dengan efek setidaknya tiga bulan setelah periode mereka memulai kursus yoga aktif. Pengalaman yoga ini memiliki dampak positif pada keseluruhan penelitian tersebut

Dalam konteks klasifikasi yoga, banyak orang memilih untuk melakukan perawatan kesehatan mental dengan melakukan yoga tetapi banyak dari mereka berpikir bahwa yoga harus dilakukan secara berkelompok atau memiliki instruktur yang dapat memandu mereka yang dimana biaya memulai kelas yoga terkesan mahal disamping itu melakukan yoga secara mandiri di rumah tanpa pengawasan instruktur, sulit bagi pengguna untuk memastikan bahwa mereka menjalankan gerakan dan postur yoga dengan benar. Hal ini dapat mengakibatkan eksekusi gerakan yang tidak efektif atau bahkan berpotensi menyebabkan cedera(Sharma et al., 2022)

Saat ini banyak pemula yang memulai yoga dengan Handphone mereka untuk mempelajari cara melakukan pose yoga dan mulai melakukannya, namun saat

melakukan itu mereka bahkan tidak tahu apakah pose yoga yang mereka lakukan sudah benar atau tidak. Untuk mengatasi keterbatasan ini, banyak upaya telah dilakukan. Teknik *Computer Vision* dan *Machine Learning* telah digunakan untuk membangun perangkat lunak AI yang berfungsi sebagai instruktur. Perangkat lunak ini akan memberikan keakuratan kinerja pose yang sedang dijalankan. Dengan menggunakan software ini seseorang dapat melakukan yoga tanpa bimbingan seorang pelatih

Umpan balik instan tentang kebenaran atau kesalahan gerakan menjadi penting dalam pengembangan keterampilan yoga. Ketika seseorang berlatih sendiri, tanpa bantuan langsung, sulit bagi mereka untuk mendapatkan umpan balik yang cepat untuk perbaikan postur atau gerakan.

Penerapan *Convolutional Neural Network* (CNN) dalam klasifikasi pose yoga adalah suatu pendekatan yang menggunakan teknologi deep learning untuk mengenali dan mengklasifikasikan posisi tubuh dalam konteks praktik yoga. CNN adalah tipe arsitektur jaringan saraf tiruan yang khusus dirancang untuk mengekstrak fitur dari data spasial, seperti gambar atau, dalam hal ini, representasi visual dari gerakan tubuh.(Zuo et al., 2019)

Prosesnya dimulai dengan memberikan model CNN sejumlah besar data latihan yang berisi citra atau frame video dari berbagai pose yoga yang berbeda. Model CNN akan belajar untuk mengekstrak pola dan fitur yang representatif dari pose-pose ini selama fase pelatihan. Ini melibatkan serangkaian lapisan konvolusi dan lapisan pooling yang berfungsi untuk mengekstrak fitur-fitur penting dari gambar, secara hierarkis.(Yudistira, 2021)

Ketika model CNN telah dilatih dengan baik, dapat digunakan untuk mengklasifikasikan pose yoga dalam situasi waktu nyata. Misalnya, saat seseorang berlatih yoga dan dicapai citra dari gerakan mereka, model CNN dapat memberikan output yang mengidentifikasi jenis pose yang sedang dilakukan.

1.2 Rumusan Masalah

Berdasarkan permasalahan latar belakang diatas , maka Penelitian ini bertujuan untuk mengembangkan sebuah sistem yang mampu mengklasifikasi gerakan yoga dengan akurat menggunakan model CNN atau *Movenet* dimana teknologi digunakan untuk memberikan kontribusi dalam mempertimbangkan koreksi postur manusia untuk meningkatkan kualitas hasil klasifikasi untuk para pengguna pemula yang ingin memulai meditasi Yoga

1.3 Batasan Masalah

Adapun Batasan masalah dalam Penelitian ini adalah sebagai berikut :

1. Penelitian ini hanya menggunakan Model *Convolutional Neural Network* dan *Movenet* sebagai Arsitektur utama dalam melakukan klasifikasi gerakan yoga
2. Sampel Dataset Gerakan Yoga hanya terbatas untuk 5 gerakan yaitu Gerakan *Downdog* , *Warrior* , *Tree* , *Cobra* dan *Chair*
3. Dataset yang digunakan untuk melatih dan menguji model klasifikasi gerakan yoga akan terdiri dari gambar dan video gerakan yoga dengan resolusi yang telah ditentukan sebelumnya. Dataset ini akan mencakup berbagai jenis gerakan yoga yang umum dilakukan.

4. Implementasi Sistem terbatas hanya melalui Android untuk melakukan klasifikasi gerakan sehingga pengguna dapat menggunakan lebih mudah dan cepat
5. Umpan balik koreksi postur manusia akan disimulasikan melalui dataset latihan, dan tidak akan melibatkan pengguna yang memberikan umpan balik secara langsung ke dalam sistem

1.4 Tujuan Penelitian

Adapun tujuan penelitian ini adalah melakukan implementasi *Convolutional Neural Network* dalam Klasifikasi Gerakan Kesehatan Yoga disamping itu juga terdapat tujuan khusus penelitian ini yaitu :

1. Menggunakan hasil Implementasi untuk memberikan penyesuaian gerakan yoga yang baik serta menciptakan aplikasi yang gratis untuk memulai perawatan kesehatan mental dengan yoga
2. Sistem yang dikembangkan juga bertujuan untuk menjadi alat yang memfasilitasi belajar mandiri dalam praktik yoga. Pengguna akan dapat menggunakan sistem ini untuk memeriksa postur tubuh mereka sendiri saat melakukan gerakan yoga, sehingga dapat meningkatkan kesadaran mereka terhadap postur yang benar dan memperbaiki teknik latihan mereka secara mandiri.
3. Mengembangkan model klasifikasi gerakan yoga yang mampu mengidentifikasi gerakan dengan akurat. Model ini diharapkan dapat membedakan antara jenis gerakan yang dilakukan dengan postur yang benar

1.5 Manfaat Penelitian

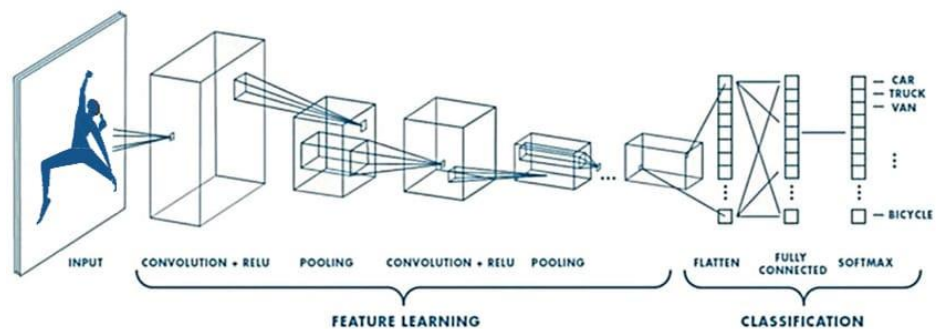
1. Dengan panduan melalui aplikasi dan penerapan machine learning yang lebih baik tentang deteksi pose yoga ini sangat berguna untuk menghindari potensi isiko cedera yang mungkin timbul akibat postur yang tidak tepat dan menciptakan panduan berdasarkan tingkat keterampilan dan kebutuhan individu
2. Sistem klasifikasi yang dikembangkan juga dapat digunakan sebagai alat untuk pembelajaran mandiri dalam praktik yoga. Individu dapat menggunakan sistem ini untuk memeriksa postur tubuh mereka sendiri dan memperbaiki teknik latihan mereka secara mandiri, meningkatkan kemampuan mereka untuk berlatih yoga dengan lebih efektif.
3. Penelitian ini juga dapat memberikan kontribusi pada pengembangan teknologi kesehatan dengan menggabungkan teknologi machine learning dan yoga. Dengan mengembangkan sistem yang mampu mengklasifikasi gerakan yoga dengan akurat, penelitian ini dapat membuka pintu untuk pengembangan aplikasi lain yang memanfaatkan teknologi untuk meningkatkan kesehatan dan kesejahteraan.

BAB II

LANDASAN TEORI

2.1 Deep Learning

Deep learning adalah sebuah ilmu cabang dari kecerdasan buatan atau *Machine Learning* yang menggunakan *Convolutional Neural Network* untuk mempelajari representasi data secara bertingkat. Ilmu ini terinspirasi bagaimana struktur dan fungsi otak manusia dapat bekerja secara umum, di mana neuron-neuron jaringan saraf otak manusia saling terhubung dalam jaringan yang kompleks dimana pola ini digunakan juga dalam *Convolutional Neural Network* (Santoso & Ariyanto, n.d.). Deep learning memungkinkan komputer untuk belajar dari data secara langsung tanpa diprogram secara eksplisit. Model deep learning dapat mengenali pola kompleks dalam gambar, teks, suara, dan data lain untuk menghasilkan wawasan dan prediksi yang cukup akurat (Hindarto et al., 2023)



Gambar 2. 1 Konsep Deep Learning
Sumber : (He & Dong, 2023)

2.2 Convolutional Neural Network

Convolutional Neural Network merupakan salah satu cabang dari Algoritma Deep Learning dimana ini hasil pengembangan dari metode MLP (Multi Layer Perceptron) dari pengembangan tersebut ditemukan bahwa CNN mengalami peningkatan signifikan dalam metode untuk pengenalan citra, dikarenakan CNN mengadopsi konsep system pengenalan citra pada *visual cortex* manusia, sehingga memiliki kemampuan yang cukup baik dalam pengelolaan informasi citra digital (Purna Irawan et al., 2022.)

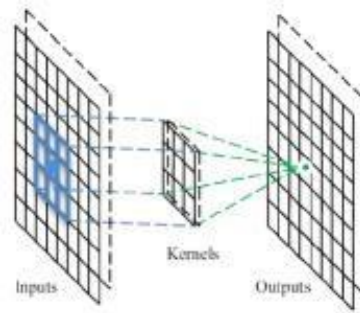
2.2.1 Lapisan Konvolusi

Lapisan konvolusi adalah elemen kunci dalam CNN yang memungkinkan "penglihatan" dan pemahaman terhadap data spasial. Ia berperan penting dalam ekstraksi fitur, transformasi data, dan lokalisasi fitur. Ia menjadi kunci utama dalam memahami pola-pola kompleks yang tersembunyi dalam data spasial, seperti gambar dan video (Putro Eko Cahyono & Rolly Maulana Awangga, 2020).

Pada lapisan ini dilakukan perkalian titik antara dua matriks, dimana satu matriks adalah himpunan parameter yang dapat dipelajari, yang dikenal dengan kernel, dan matriks lainnya adalah bagian terbatas dari bidang reseptif. Secara matematis suatu perkalian konvolusi untuk fungsi diskrit satu dimensi, dapat dinyatakan dalam persamaan berikut :

$$f(x) * g(x) = \sum f(\langle)g(x - \langle) \quad (1) \quad (2.2)$$

dimana $g(x)$ adalah filter atau kernel konvolusi yang dioperasikan dengan cara menggeser pada sinyal masukan $f(x)$.



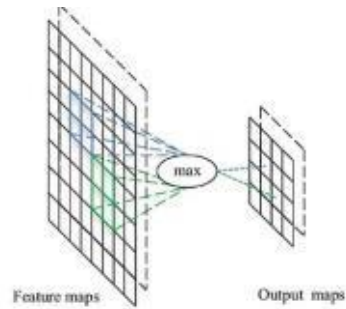
Gambar 2. 2 Proses Lapisan Konvolusi

Sumber : Moayyed et al., 2022

Dalam proses lapisan konvolusi terdapat 3 bagian utama dalam pemrosesannya seperti mekanisme berbagi bobot pada fitur yang sama , sehingga dapat mengurangi beberapa parameter yang tidak begitu penting , kedua adanya konektivitas antar lokal dimana ini mempelajari bagaimana hubungan antar piksel , serta ditutup dengan penglokasian objek tertentu(Rere et al., 2019)

2.2.2 Lapisan Pooling

Lapisan pooling, bagaikan jembatan yang menghubungkan kompleksitas data input dengan efisiensi komputasi dan robustness jaringan saraf tiruan convolutional neural network (CNN). Ia bukan sekadar penyederhanaan data, tetapi umumnya lapisan ini digunakan untuk mengurangi dimensi peta fitur dan parameter jaringan. Umumnya pada lapisan ini biasanya menggunakan strategi *max-pooling*



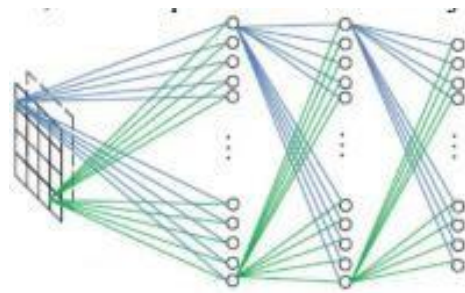
Gambar 2. 3 Proses Lapisan Pooling

Sumber : (Moayyed et al., 2022)

Lapisan pooling sama halnya dengan lapisan konvolusi juga translation invariant. Hal ini karena pada proses komputasinya mengambil piksel ketetanggaan dalam perhitungan, dengan mengurangi resolusi peta fitur (Rere et al., 2019)

2.2.3 Lapisan Fully Connected

Lapisan fully connected merupakan komponen yang cukup penting dalam *Convolutional Neural Network*, pada dasarnya lapisan ini berperan sebagai Mesin Voting yang mengintegrasikan informasi dari seluruh feature map dan menghasilkan klasifikasi akhir (Mehindra Prasmatio et al., 2020)



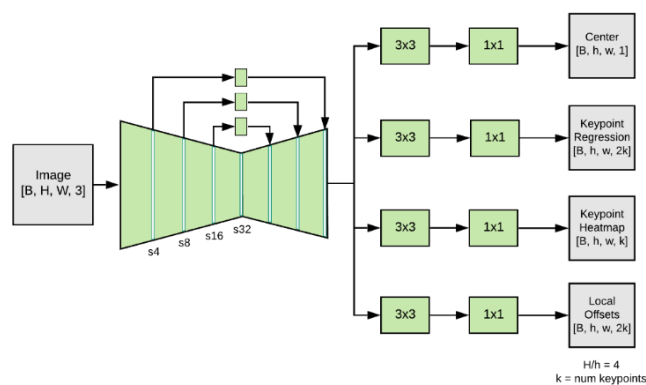
Gambar 2. 4 Proses Lapisan Fully Connected

Sumber : (Motamed, 2022)

Lapisan ini bekerja sebagai masukan untuk serangkaian lapisan terhubung sepenuhnya yang melaksanakan tugas klasifikasi. Seperti diperlihatkan pada Gambar 2.4

2.3 Movenet

MoveNet adalah model pengembangan berbasis Google dikembangkan oleh *Include Health*, sebuah perusahaan kesehatan digital, pada tahun 2021 perusahaan tersebut meminta bantuan dari Google untuk mendukung perawatan pasien jarak jauh. Model ini sebenarnya sangat serupa dengan model PoseNet dimana model PoseNet merupakan versi web dari model MoveNet yang menggunakan bahasa TensorFlow.js, dan untuk penggunaan versi mobile atau androidnya dapat menggunakan menggunakan TensorFlow Lite. Umumnya terdapat dua versi yang dikembangkan dalam model MoveNet yaitu Model Lightning yang berorientasi pada kinerja dan Model Thunder yang berorientasi pada akurasi model tersebut (Jo & Kim, 2022)



Gambar 2. 5 Arsitektur MoveNet

(Sumber : <https://blog.tensorflow.org/2021/05/next-generation-pose-detection-with-movenet-and-tensorflowjs.html>)

Seperti yang ditunjukkan pada Gambar 2.5 , MoveNet menghitung keempatnya proses secara bersamaan. Kemudian terdapat Heatmap keypoint yang merupakan peta probabilitas dimana ini menunjukkan lokasi keypoint tubuh manusia. Heatmap ini dihasilkan dari model regresi heatmap. Setelah itu model regresi keypoint memprediksi koordinat x dan y dari keypoint tubuh manusia. Selanjutnya setiap piksel dikalikan dengan bobotnya, yaitu berbanding terbalik dengan jarak dari titik kunci yang diregresi. Dengan cara ini, poin-poin penting dari orang-orang di latar belakang dikecualikan dari perhitungan. Pada akhirnya, itu kumpulan poin-poin penting diselesaikan secara maksimal nilai peta panas di setiap saluran titik kunci (Jo & Kim, 2022)

2.4 TensorFlow

TensorFlow adalah *Open Source Tools* yang digunakan untuk membuat, mengevaluasi, dan melatih model *machine learning*. Ini adalah kerangka kerja terkenal yang diadopsi dalam *machine learning* yang dapat berinteraksi melalui library Python. Dalam penelitian ini, akan menggunakan TensorFlow Lite, dimana ini merupakan alat paket yang diadopsi dari model TensorFlow untuk kebutuhan model perangkat seluler . TensorFlow versi ringan ini adalah alat canggih dan industrial yang melayani model pembelajaran mendalam untuk implementasi pada ponsel atau papan mikrokontroler (Abed et al., 2023)

Dalam bidang object detection terdapat framework tensorflow object detection API yang merupakan suatu alat yang dapat digunakan

untuk mempermudah proses constructing, training dan deployment pada suatu model object detection. Framework tensorflow object detection API menyediakan pretrained object detection model bagi user, namun memungkinkan jika user ingin menggunakan pretrained object detection model yang lain, seperti Faster R-CNN, SSD, Retinanet, Resnet50 dan masih banyak lagi.(Dufan J. P. Manajang et al., 2020)

2.5. Gerakan Kesehatan Yoga

Yoga adalah gerakan meditasi yang melatih disiplin terhadap fisik, mental, dan spiritual dimana meditasi ini berasal dari India. *United Nation* atau Persatuan Bangsa Bangsa menyadari bahwa yoga adalah pendekatan holistik yang diperlukan sehingga memerlukan penyebaran tentang manfaat gerakan ini yang lebih luas demi kesejahteraan masyarakat. Meditasi Yoga telah didokumentasikan sebagai salah satu dari beberapa sistem terapi tradisional di *World Health Organization*.(Singh PK, 2017)

Dalam beberapa artikel penelitian yoga memberikan dampak berupa kekuatan dan pengendalian diri untuk menghadapi tantangan dalam individu manusia. Gerakan meditasi ini telah dipraktikkan secara luas di seluruh dunia dan menjadi lebih menjadi meditasi yang cukup dikenal saat ini. Sebuah studi tentang evaluasi efek terapi yoga atau dari temuan artikel yang cukup terpilih menunjukkan bahwa yoga meningkatkan kualitas hidup secara mental dan kesejahteraan seseorang individu, dalam terapi yoga ditemukan bahwa meditasi ini memiliki hubungan dengan mekanisme anti stress tubuh (Gohel et al., 2021)

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian dan Pengembangan Produk

Metode penelitian adalah bentuk pembuktian ilmiah guna memperoleh informasi agar tercapainya tujuan dan manfaat yang diinginkan. Biasanya dalam penelitian diperlukan upaya yang harus didasari dengan bagian ciri dari keilmuan seperti, masuk akal, berdasarkan pengalaman atau ada bukti empiris, dan terstruktur. Penelitian yang dilakukan haruslah dapat diterima oleh akal sehat manusia atau rasional, jadi apa pun langkah-langkah yang diambil dalam penelitian tersebut dapat diterima oleh penalaran seseorang.

Pembuktian akan upaya penelitian ini dapat dilihat dan dipahami dengan penggunaan prosedur tertentu dimana penelitian harus terstruktur dan juga sistematis sehingga penelitian ini memiliki sifat logis atau masuk akal . secara umum biasanya penelitian memiliki beberapa metode yang sering digunakan sebagai pendekatan antara lain :

1. Penelitian berbasis Riset dan Pengembangan (*Research and Development*) Penelitian riset ini adalah suatu kajian tentang pencarian pola dan urutan pertumbuhan sebagai fungsi waktu. Objek penelitiannya adalah perubahan atau kemajuan yang dicapai oleh individu seperti penulis saat ini .Tujuan penelitian ini adalah untuk mengetahui perkembangan model *machine learning* dan aplikasi yang sedang dikembangkan

2. Penelitian Kualitatif (*Qualitative Research*)

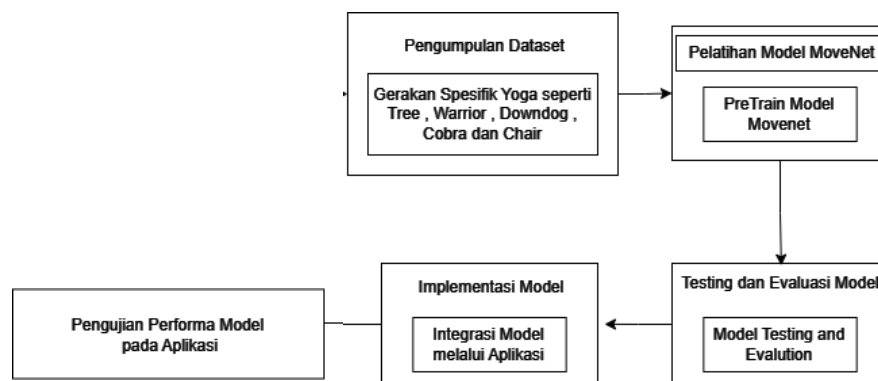
Penelitian kualitatif ini adalah bentuk penelitian yang digunakan menjawab permasalahan secara mendalam dalam konteks pengerjaan situasi yang berkembang, biasanya penelitian dilakukan secara wajar dan alami sesuai dengan kondisi objektif di lapangan dan pada tahapan ini sebagai penulis telah mewawancarai terhadap beberapa individu yang memerlukan aplikasi meditasi trainer yoga

Kemudian dalam pengembangan aplikasi atau evaluasi model machine learning dilakukan pengembangan yang cukup berbeda dengan penelitian dimana pada pengembangan tersebut kita akan menggunakan metode Agile , dimana peneliti memungkinkan merespons perubahan kebutuhan dengan cepat dan fleksibel , membangun aplikasi dan model secara iteratif untuk meningkatkan kualitasnya

Dari beberapa upaya pada jenis penelitian yang umum digunakan, untuk pendekatan penelitian ini maka dalam penelitian ini peneliti menggunakan metode pendekatan pada jenis penelitian R&D (*Research and Development*) atau sering disebut penelitian pengembangan. Metode penelitian yang digunakan ini untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut dan dalam pengembangan produk peneliti akan membangun dalam konsep Agile

3.2 Alur Penelitian

Dalam pelaksanaan penelitian ini, ada serangkaian langkah yang akan diambil untuk mencapai kesimpulan dari penelitian ini. Rinciannya mengenai langkah-langkah penelitian dapat ditemukan dalam diagram berikut:



Gambar 3. 1 Alur Penelitian dan Pengembangan

3.2.1 Studi Literatur

Pada tahapan awal alur penelitian sebagai peneliti, saya mengumpulkan, meninjau, dan menganalisis literatur yang relevan dan terkait dengan topik penelitian seperti pengumpulan data informasi, membaca dan mencatat serta mengolah informasi menjadi bahan penelitian. Disamping itu sebagai penulis, studi literatur digunakan untuk memahami pengetahuan yang sudah ada tentang topik yang diteliti, mengidentifikasi celah pengetahuan, dan membentuk dasar yang kuat untuk penelitian yang akan dilakukan.

Melalui studi literatur , penulis mempelajari banyak tentang teori yang berhubungan dengan yoga dan deep learning , dengan alat bantu pengujian menggunakan tensorflow sebagai platform pengujian model dan nantinya diharapkan dapat mengembangkan kedalam produk berupa mobile app kemudian untuk studi literatur berasal dari buku , jurnal serta studi yang terdahulu.

3.2.2 Pengumpulan Dataset

Tahap pemrolehan data ini merupakan bagian integral dari penelitian ini. Dataset yang digunakan untuk melatih dan mengevaluasi model terdiri dari beberapa gerakan yoga yang diperoleh dari sumber kaggle. Data yang digunakan dalam penelitian ini terklasifikasi ke dalam 5 kategori seperti seperti downdog , tree , warrior , chair dan cobra Penelitian ini bertujuan untuk mengetahui performa model Deep Learning MoveNet dalam deteksi dan klasifikasi jenis gerakan yoga maka dalam penerapannya, Movenet seperti halnya model Deep Learning lainnya memerlukan adanya dataset yang berupa citra citra beserta labelnya untuk digunakan dalam proses pelatihan/training atau pada kasus MoveNet merupakan proses transfer learning.

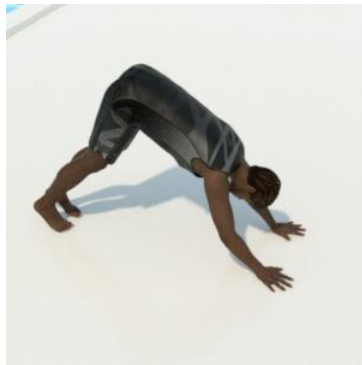
Berikut merupakan sampel gambar beberapa gerakan yoga yang nantinya akan digunakan untuk sampel dataset pengembangan klasifikasi yoga



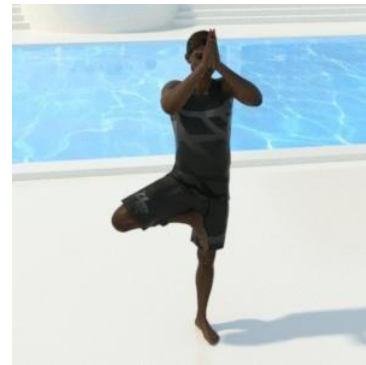
Gambar 3.2 Pose Chair



Gambar 3.3 Gambar Pose Cobra



Gambar 3.4 Pose Downdog



Gambar 3.5 Pose Tree



Gambar 3.6 Pose Warrior

Pada pengumpulan dataset ini pastikan , kita memiliki gambar yang cukup banyak untuk pose pose yang berbeda , pada penelitian ini , peneliti akan menggunakan data training sebanyak 942 data gambar dengan pose yang berbeda kemudian untuk data testing sebanyak 675 data gambar

3.2.3 Pelatihan Model MoveNet

Dikarenakan model MoveNet merupakan model pra terlatih / pretrained, proses pelatihan ini merupakan proses transfer learning yang hanya melatih lapisan akhir untuk menyesuaikan model untuk mendeteksi kelas kelas dalam dataset yang diberikan.

Sebelum proses pelatihan ini dilakukan, perlu dilakukan sedikit konfigurasi terhadap model MoveNet yang akan digunakan, seperti direktori dataset, ukuran citra input dan kedalaman citra lalu mengatur konfigurasi pelatihan seperti jumlah epoch, batch size, dan callback untuk angka akurasi target. Setelah konfigurasi selesai dilakukan dan sesuai dengan model, maka proses pelatihan dapat berlangsung. Model akan melakukan proses pelatihan terhadap dataset sebanyak jumlah epoch yang diberikan atau hingga angka akurasi yang diberikan pada callback. Setelah proses pelatihan selesai dilakukan, maka model secara otomatis di ekspor dalam bentuk tflite(tfl) dan dapat digunakan untuk prediksi gerakan.

3.2.4 Testing dan Evaluasi Model

Setelah model berhasil dilatih dan menghasilkan akurasi pelatihan yang memuaskan, maka selanjutnya dapat dilakukan evaluasi performa model terhadap gerakan pose yoga. Dalam proses evaluasi dilakukan prediksi terhadap kelompok gerakan yang memiliki label untuk mengetahui hasil aslinya dan kelompok gambar ini harus tidak pernah digunakan dalam proses pelatihan untuk memastikan integritas hasil evaluasi. Proses evaluasi dilakukan dengan melakukan prediksi pada direktori yang mengandung

kelompok dataset evaluasi, dan menghasilkan angka evaluasi yang akan dilakukan analisa pada proses selanjutnya.

3.2.5 Implementasi Model dan Pengujian Performa Model pada Aplikasi

Langkah selanjutnya adalah mengintegrasikan model ke dalam aplikasi yang akan digunakan untuk klasifikasi gerakan yoga secara praktis. Ini melibatkan pengembangan antarmuka pengguna yang ramah pengguna, yang memungkinkan pengguna untuk merekam atau memperoleh video gerakan yoga dan mendapatkan klasifikasi gerakan secara real-time dari model MoveNet dan tahapan terakhir adalah pengujian performa model yang diimplementasikan pada aplikasi yang dikembangkan. Ini melibatkan uji coba aplikasi dengan pengguna yang sesungguhnya atau dengan data yang relevan untuk mengevaluasi seberapa baik model MoveNet berkinerja dalam mengklasifikasikan gerakan yoga dalam situasi praktis

3.3 Kebutuhan Hardware & Software

3.3.1 Kebutuhan Hardware

Dalam pengimplementasian penelitian ini , diperlukan beberapa perangkat keras yang menggunakan arsitektur x64 dengan spesifikasi minimal tertentu. Spesifikasi ini menjadi kebutuhan kritis untuk menjalankan model dan memberikan landasan evaluasi kinerja kecepatan model dalam konteks penelitian ini.

Tabel 3. 1 Kebutuhan Hardware

No	Hardware	Spesifikasi
1	<i>Processor</i>	Intel Core i7-7700HQ
2	<i>RAM</i>	8 GB
3	<i>GPU</i>	Geforce GTX 1050 Ram 16Gb
4	Penyimpanan	1 <i>Terabyte</i>
5	Sistem Operasi	<i>Windows 10 / Windows 11</i>

Dalam konteks pelaksanaan penelitian, dapat dipilih untuk menggunakan sistem yang memiliki spesifikasi seperti yang telah dijelaskan sebelumnya atau memanfaatkan komputasi awan melalui platform seperti Google Collab atau layanan cloud lainnya untuk menjalankan model

3.3.2 Kebutuhan Software

Pada perancangan sistem ini dibutuhkan beberapa perangkat lunak (*Software*) sebagai pendukung untuk melakukan penelitian ini. Ada dua *software* yang akan digunakan diantaranya yaitu *Android Studio*, dan Google Colaboratory. Pada perancangan ini peneliti menggunakan *Android Studio* sebagai *software IDE* untuk merancang *Mobile App* yang akan dikembangkan, Dan peneliti menggunakan Google Colaboratory sebagai tempat untuk merancang logika dari *Machine Learning* yang akan digunakan peneliti untuk membuat Sistem Klasifikasi Gerakan Yoga

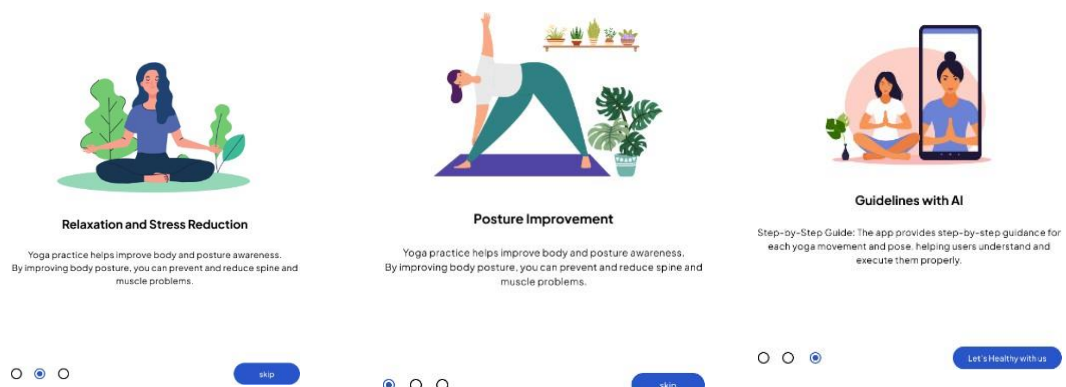
BAB IV

HASIL DAN PEMBAHASAN

4.1 Rancangan Desain Tampilan Android

Pada tahapan awal ini sebelum membangun dataset kita akan fokus dalam membangun tampilan awal aplikasi berbasis android dimana penggunaan figma digunakan untuk memberikan gambaran seperti apa tampilan aplikasi android dalam implementasi klasifikasi gerakan yoga dengan *convolutional neural network*

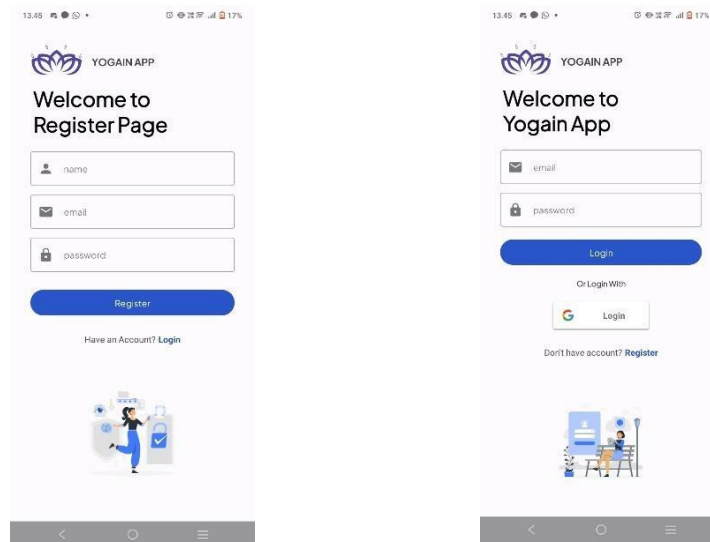
Nantinya setelah tampilan UX Design berhasil dibuat maka kita akan implementasi gambaran tersebut ke android menggunakan pemograman kotlin sehingga lebih memudahkan menentukan layout pada pemograman android berbasis Kotlin



Gambar 4.1 Tampilan Awal Onboarding

Diatas merupakan gambar onboarding dimana user akan diberitahu gambaran aplikasi yang mereka gunakan , dimana gambar (a) menunjukkan tentang penjelasan yoga bermanfaat dalam relaksasi dan pereda stress, kemudian gambar (b) menjelaskan tentang penggunaan aplikasi yang bisa mengimprovisasi gerakan secara mandiri dan gambar (c) menjelaskan

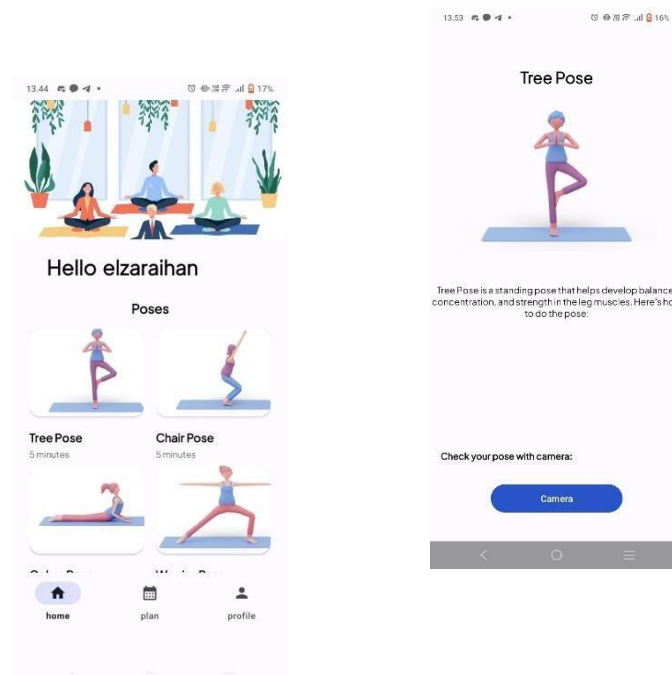
panduan menggunakan machine learning untuk memudahkan proses panduan meditasi kemudian setelah itu kita juga akan meracanakan tampilan Login dan Register Account untuk dibangun dimana user akan menggunakan akun untuk melakukan percobaan aplikasi dimana tampilannya akan sebagai berikut



Gambar 4.2 Tampilan Login dan Register

Pada tampilan register . pengguna dapat mendaftarkan akun baru dimana adanya pengisian nama , email dan juga password dimana akun diperlukan untuk akses penggunaan aplikasi tersebut , kemudian untuk gambar (b) merupakan tampilan login dimana diperlukan akun untuk membuka akses aplikasi tersebut berdasarkan data pendaftaran sebelumnya di menu register

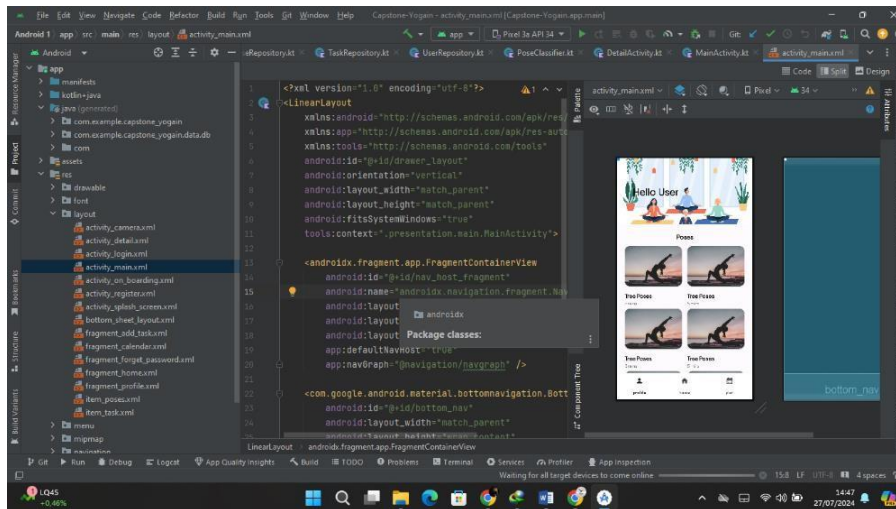
Kemudian pada tahap selanjutnya tidak lupa juga untuk membangun tampilan berupa halaman utama pilihan klasifikasi gerakan yoga apa yang akan kita gunakan dalam mendeteksi pilihan gerakan kita untuk mengkoreksi postur yang kita gunakan dan sebelum melakukan deteksi kamera secara real time maka ada tampilan yang memberikan gambar statis dan penjelasan terkait pose yang sedang kita pilih



Gambar 4.5 Tampilan Halaman Pilihan Yoga dan Detail Panduan

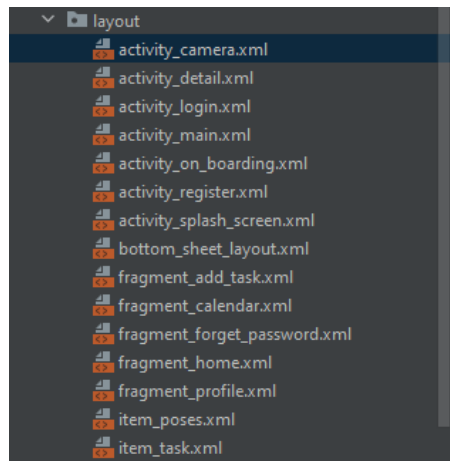
Pada gambar (a) pengguna dapat memilih pose yoga apa yang mereka lakukan pada hari itu dan dapat memilih pilihan pose yoga tepat dimana sebelum memulai pergerakan pose yoga adanya panduan gambar dan penjelasan tentang gerakan tersebut sebelum memulai pada button camera untuk aktivasi kamera dalam deteksi gerakan yoga

4.2 Implementasi Kotlin Tampilan Android



Gambar 4.4 Implementasi Layout pada Kotlin

Pada tahapan selanjutnya, adanya implementasi rancangan desain tampilan kedalam bentuk android nantinya dimana disini saya membangun dengan bahasa pemrograman kotlin dimana pembangunan layout dibangun dengan android studio dimana terdapat widget yang dapat digunakan dalam memilih komponen komponen UI yang sesuai kebutuhan kita dalam interaksi interface tampilan nantinya



Gambar 4.5 Tampilan Layout Activity

Pada pengembangan layout android, peneliti melakukan pembagian beberapa layout dimana layout berbeda memiliki fungsi yang berbeda dimana nantinya layout tersebut akan

digabung dan dilakukan penyatuan dalam bentuk APK nantinya dalam implementasi android

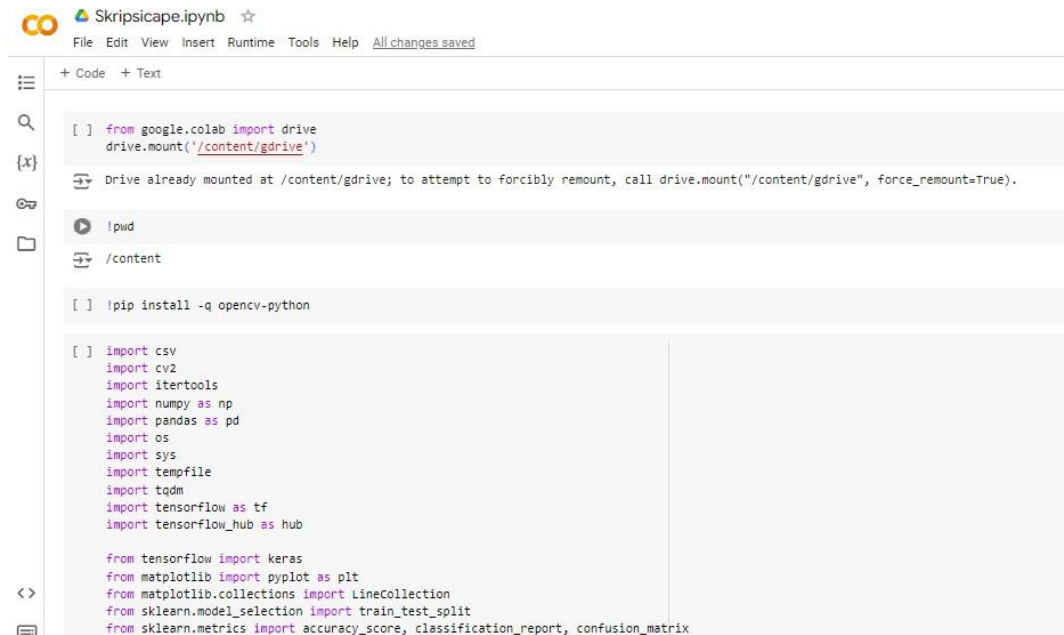
4.3 Pembangunan Model dan Training serta Testing Data

Pada tahapan ini hasil pengumpulan dataset dikumpulkan dalam folder yang berisikan label folder dengan isi gambar yang berbeda sesuai dengan label yoga yang dimiliki, kemudian pastikan semua dataset dan folder model dibangun didalam google drive agar nantinya dapat digunakan pada Google Colab nantinya dalam pelatihan dan pembangunan data



Gambar 4.6 Google Drive Source Google Collab

Pada drive ini telah dibagi menjadi 3 folder dimana folder app merupakan folder yang berisi aplikasi mobile app yang dibangun sebelumnya, kemudian folder datasets dimana terdapat 2 bagian yaitu testing dan training dimana data tersebut digunakan untuk melakukan pelatihan serta testing data nantinya dimana folder model merupakan output yang akan digunakan dari hasil pelatihan data yang telah dibangun pada google collaborator



```

from google.colab import drive
drive.mount('/content/gdrive')

Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).

!pwd

/content

[ ] !pip install -q opencv-python

[ ] import csv
import cv2
import itertools
import numpy as np
import pandas as pd
import os
import sys
import tempfile
import tqdm
import tensorflow as tf
import tensorflow_hub as hub

from tensorflow import keras
from matplotlib import pyplot as plt
from matplotlib.collections import LineCollection
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

```

Gambar 4.7 Perintah Instalasi Impor Awal

Pada tahap selanjutnya google collaborator akan dihubungkan google drive dengan google collab dengan perintah `from.google.colab import drive` dimana dengan secara otomatis akan mengimpor google drive sebagai alat pengumpulan data , Pada barisselanjutnya adanya penggunaan modul library untuk persiapan pemrosesan data dan persiapan untuk penggunaan opencv untuk melakukan testing secara langsung pada google collaborator nantinya



```

[ ] # Download Model Movenet
!wget -q -O movenet_thunder.tflite https://tfhub.dev/google/lite-model/movenet/singlepose/thunder/tflite/float16/4?lite-format=tflite
!git clone https://github.com/tensorflow/examples.git
pose_sample_rpi_path = os.path.join(os.getcwd(), 'examples/lite/examples/pose_estimation/raspberry_pi')
sys.path.append(pose_sample_rpi_path)

fatal: destination path 'examples' already exists and is not an empty directory.

[ ] import utils
from data import BodyPart
from ml import Movenet
movenet = Movenet('/content/gdrive/MyDrive/YOGA/model/movenet_thunder.tflite')

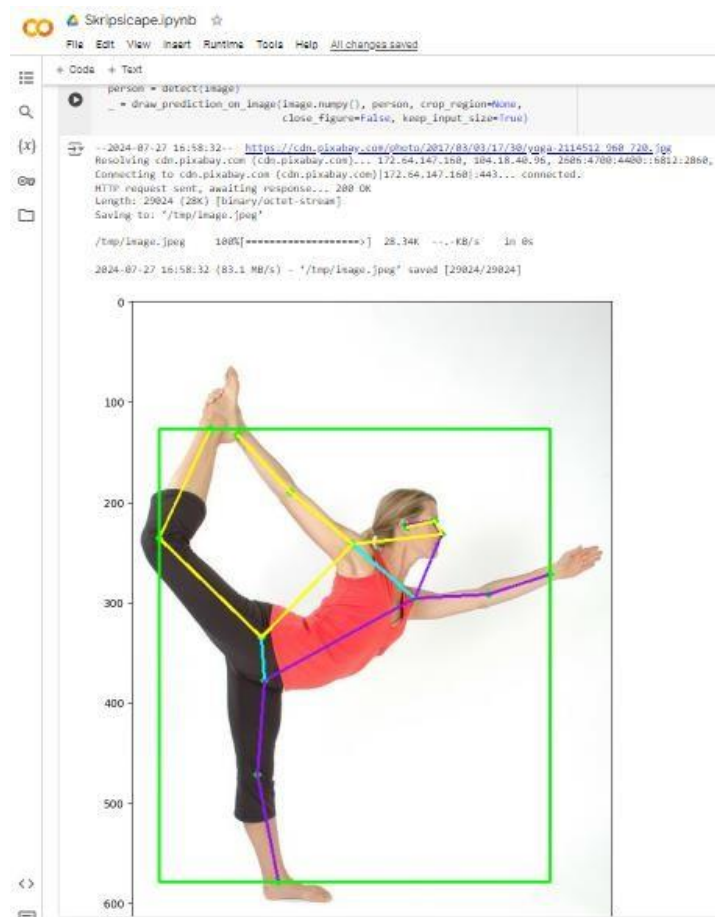
```

Gambar 4.8 Impor Model Movenet dalam Prediksi pose

Kemudian setelah melakukan koneksi dengan google drive dengan google collab , selanjutnya movenet akan digunakan sebagai model utama dalam pendeteksian pose pada tubuh manusia dalam bentuk gambar , dimana nantinya gambar manusia akan diberikan titik keypoint dalam menentukan pose tubuh manusia , disamping itu juga movenet akan

diimpor kedalam drive google pribadi dimana hasil mode merupakan movenet thunder tflite yang dapat digunakan nantinya

Kemudian setelah proses panjang dalam pelatihan model movenet sebagai penggunaan deteksi model pose ,dimana data dasar pelatihan model movenet digunakan berdasarkan gambar yang digunakan yang telah kita simpan pada google drive sebelumnya dimana gambar ini digunakan sebagai pelatihan model untuk mendeteksi gambar dalam menentukan pose berdasarkan label yang sudah kita buat sebelumnya , maka lakukan proses input gambar untuk melihat apakah model movenet telah berhasil mendeteksi sebuah pose pada gambar dengan menggunakan keypoint



Gambar 4.9 Hasil Model Movenet dalam Keypoint Deteksi

Pada hasil testing model movenet yang digunakan telah menunjukkan hasil terkait model movenet dapat digunakan dalam mendeteksi pose gerakan pada tubuh manusia , maka model keypoint tersebut dapat digunakan

Pada tahapan selanjutnya setelah model movenet telah dilatih sebagai dasar penggunaan model pendeteksian pose maka perlunya training data yang baru terkait gambar baru yang kita miliki pada google drive dimana data tersebut memiliki gambar yang berbeda sesuai label yang telah diberikan , dimana proses training diharapkan menghasilkan data pendeteksian gambar berdasarkan gerakan yang dimiliki oleh gambar

```

[ ] dataset_in = "/content/gdrive/MyDrive/YOGA/datasets/TRAIN"
dataset_out = "/content/gdrive/MyDrive/YOGA/dataset/"
# split_into_train_valid(dataset_in, dataset_out, valid_split=0.2)

[ ] images_in_train_folder = "/content/gdrive/MyDrive/YOGA/datasets/TRAIN/"
images_out_train_folder = "poses_images_out_train"
csvs_out_train_path = 'train_data.csv'

preprocessor = MoveNetPreprocessor(
    images_in_folder=images_in_train_folder,
    images_out_folder=images_out_train_folder,
    csvs_out_path=csvs_out_train_path,
)

preprocessor.process(per_pose_class_limit=None)

Preprocessing chair
100% ██████████ 85/85 [05:25<00:00, 3.82s/it]
Preprocessing cobra
100% ██████████ 232/232 [13:04<00:00, 3.38s/it]
Preprocessing downward
100% ██████████ 223/223 [11:59<00:00, 3.23s/it]
Preprocessing tree
22% ██████ | 36/168 [02:00<07:21, 3.56s/it] /usr/local/lib/python3.10/dist-packages/numpy/lib/function_base.py:520: RuntimeWarning: Mean of empty slice.
  avg = a.mean(axis, **keepdims_kw)
/usr/local/lib/python3.10/dist-packages/numpy/core/_methods.py:129: RuntimeWarning: invalid value encountered in scalar divide
  ret = ret.dtype.type(ret / rcount)
Preprocessing warrior2
100% ██████████ 160/160 [08:41<00:00, 3.26s/it]
Preprocessing warrior2
100% ██████████ 252/252 [14:00<00:00, 3.33s/it] Skipped /content/gdrive/MyDrive/YOGA/datasets/TRAIN/cobra/guy3_cobra036_flipped.jpg. No pose was confidentially detected.

```

Gambar 4.10 Pelatihan Data Gambar Label Yoga

Pada tahapan ini adanya preprocessing beberapa pose yang telah kita tentukan dimana , proses ini memakan waktu dan nantinya data training itu akan dapat digunakan proses testing model dalam memprediksi output tanpa penyesuaian bobot dan mengetahui hitungan akurasi dan kinerja model

```

images_in_test_folder = "/content/gdrive/MyDrive/YOGA/datasets/TEST/"
images_out_test_folder = "poses_images_out_test"
csvs_out_test_path = 'test_data.csv'

preprocessor = MoveNetPreprocessor(
    images_in_folder=images_in_test_folder,
    images_out_folder=images_out_test_folder,
    csvs_out_path=csvs_out_test_path,
)

preprocessor.process(per_pose_class_limit=None)

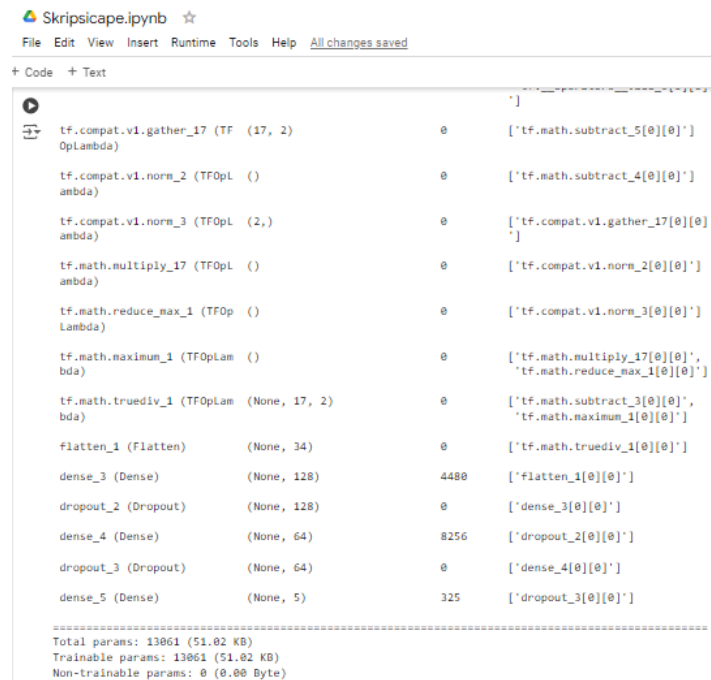
Preprocessing chair
100%|██████████| 168/168 [09:30<00:00, 3.40s/it]
Preprocessing cobra
100%|██████████| 232/232 [13:08<00:00, 3.40s/it]
Preprocessing dowdog
100%|██████████| 97/97 [05:49<00:00, 3.60s/it]
Preprocessing tree
86%|██████████| 59/69 [02:41<00:35, 3.56s/it]/usr/local/lib/python3.10/dist-packages/numpy/lib/function_base.py:520: RuntimeWarning: Mean of empty slice.
  avg = a.mean(axis, **keepdims_kw)
/usr/local/lib/python3.10/dist-packages/numpy/core/_methods.py:129: RuntimeWarning: invalid value encountered in scalar divide
  ret = ret.dtype.type(ret / rcount)
100%|██████████| 69/69 [04:15<00:00, 3.70s/it]
Preprocessing warrior2
100%|██████████| 109/109 [05:57<00:00, 3.28s/it]Skipped /content/gdrive/MyDrive/YOGA/datasets/TEST/cobra/guy3_cobra036_flipped.jpg. No pose was confidently detected.

```

Gambar 4.11 Testing Data terkait Pose

Proses ini mempersiapkan data gambar pengujian dengan mengonversinya ke format yang sesuai untuk model MoveNet. Tujuannya adalah untuk memastikan bahwa model dapat mengevaluasi gambar dengan benar dan mengidentifikasi pose dalam data pengujian. Peringatan dan kesalahan menunjukkan area yang mungkin memerlukan perhatian lebih lanjut untuk memastikan keakuratan.

Kemudian proses tersebut menghasilkan arsitektur model yang dapat digunakan nantinya dimana hasilnya sebagai berikut



Layer Name	Shape	Output	Connections
tf.compat.v1.gather_17 (TF OPLambda)	(17, 2)	0	['tf.math.subtract_5[0][0]']
tf.compat.v1.norm_2 (TFOPLambda)	(1,)	0	['tf.math.subtract_4[0][0]']
tf.compat.v1.norm_3 (TFOPLambda)	(2,)	0	['tf.compat.v1.gather_17[0][0]']
tf.math.multiply_17 (TFOPLambda)	(1,)	0	['tf.compat.v1.norm_2[0][0]']
tf.math.reduce_max_1 (TFOPLambda)	(1,)	0	['tf.compat.v1.norm_3[0][0]']
tf.math.maximum_1 (TFOPLambda)	(1,)	0	['tf.math.multiply_17[0][0]', 'tf.math.reduce_max_1[0][0]']
tf.math.truediv_1 (TFOPLambda)	(None, 17, 2)	0	['tf.math.subtract_3[0][0]', 'tf.math.maximum_1[0][0]']
flatten_1 (Flatten)	(None, 34)	0	['tf.math.truediv_1[0][0]']
dense_3 (Dense)	(None, 128)	4480	['flatten_1[0][0]']
dropout_2 (Dropout)	(None, 128)	0	['dense_3[0][0]']
dense_4 (Dense)	(None, 64)	8256	['dropout_2[0][0]']
dropout_3 (Dropout)	(None, 64)	0	['dense_4[0][0]']
dense_5 (Dense)	(None, 5)	325	['dropout_3[0][0]']

Total params: 13061 (51.02 KB)
 Trainable params: 13061 (51.02 KB)
 Non-trainable params: 0 (0.00 Byte)

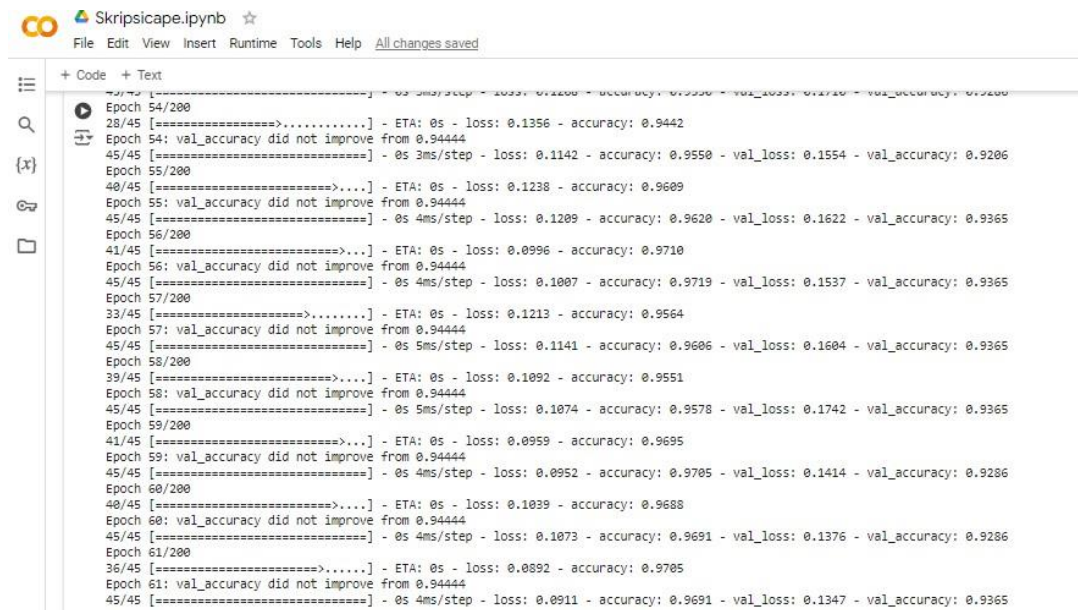
Gambar 4.12 Arsitektur Model yang dihasilkan

Dalam hasil model **yang didapatkan** menunjukkan bahwa

1. **Total Params:** Jumlah total parameter dalam model adalah 13,061. Ini termasuk semua bobot dan bias dalam jaringan.
2. **Trainable Params:** Semua parameter (13,061) dapat dilatih. Ini berarti setiap parameter dapat diperbarui selama proses pelatihan untuk meminimalkan loss dan meningkatkan akurasi model.
3. **Non-trainable Params:** Tidak ada parameter yang bersifat non-trainable (0). Ini menunjukkan bahwa semua parameter dalam model ini dioptimalkan selama pelatihan.

Keseluruhan ukuran parameter adalah 51.02 KB, menandakan bahwa model ini relatif ringan dan kompak untuk keperluan komputasi.

Kemudian proses selanjutnya adanya validasi akurasi yang telah dibangun dimana ada implementasi kode Dengan menggunakan teknik seperti checkpointing dan early stopping, model dioptimalkan untuk performa dan ketahanan terhadap overfitting. Penggunaan strategi ini penting dalam pengembangan model machine learning yang robust dimana hasilnya menunjukkan bahwa model memiliki kemampuan klasifikasi yang cukup baik, namun ada ruang untuk peningkatan, baik melalui peningkatan arsitektur model, penggunaan data yang lebih banyak, atau teknik regularisasi tambahan. Penggunaan strategi seperti early stopping dan checkpointing sangat penting dalam mengoptimalkan performa dan stabilitas model.



```

Skripsicape.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

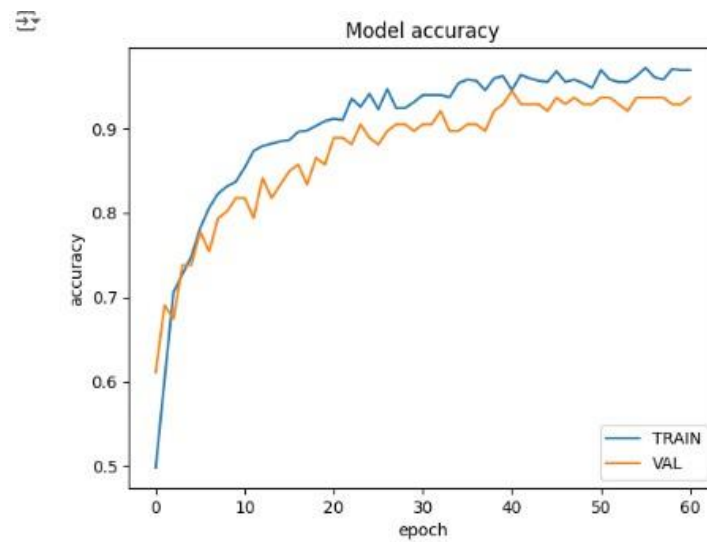
Epoch 54/200 [=====>.....] - ETA: 0s - loss: 0.1356 - accuracy: 0.9442
Epoch 54: val_accuracy did not improve from 0.94444
45/45 [=====] - 0s 3ms/step - loss: 0.1142 - accuracy: 0.9550 - val_loss: 0.1554 - val_accuracy: 0.9286
Epoch 55/200 [=====>.....] - ETA: 0s - loss: 0.1238 - accuracy: 0.9609
Epoch 55: val_accuracy did not improve from 0.94444
45/45 [=====] - 0s 4ms/step - loss: 0.1209 - accuracy: 0.9620 - val_loss: 0.1622 - val_accuracy: 0.9365
Epoch 56/200 [=====>.....] - ETA: 0s - loss: 0.0996 - accuracy: 0.9710
Epoch 56: val_accuracy did not improve from 0.94444
45/45 [=====] - 0s 4ms/step - loss: 0.1007 - accuracy: 0.9719 - val_loss: 0.1537 - val_accuracy: 0.9365
Epoch 57/200 [=====>.....] - ETA: 0s - loss: 0.1213 - accuracy: 0.9564
Epoch 57: val_accuracy did not improve from 0.94444
45/45 [=====] - 0s 5ms/step - loss: 0.1141 - accuracy: 0.9606 - val_loss: 0.1604 - val_accuracy: 0.9365
Epoch 58/200 [=====>.....] - ETA: 0s - loss: 0.1092 - accuracy: 0.9551
Epoch 58: val_accuracy did not improve from 0.94444
45/45 [=====] - 0s 5ms/step - loss: 0.1074 - accuracy: 0.9578 - val_loss: 0.1742 - val_accuracy: 0.9365
Epoch 59/200 [=====>.....] - ETA: 0s - loss: 0.0959 - accuracy: 0.9695
Epoch 59: val_accuracy did not improve from 0.94444
45/45 [=====] - 0s 4ms/step - loss: 0.0952 - accuracy: 0.9705 - val_loss: 0.1414 - val_accuracy: 0.9286
Epoch 60/200 [=====>.....] - ETA: 0s - loss: 0.1039 - accuracy: 0.9688
Epoch 60: val_accuracy did not improve from 0.94444
45/45 [=====] - 0s 4ms/step - loss: 0.1073 - accuracy: 0.9691 - val_loss: 0.1376 - val_accuracy: 0.9286
Epoch 61/200 [=====>.....] - ETA: 0s - loss: 0.0892 - accuracy: 0.9785
Epoch 61: val_accuracy did not improve from 0.94444
45/45 [=====] - 0s 4ms/step - loss: 0.0911 - accuracy: 0.9691 - val_loss: 0.1347 - val_accuracy: 0.9365

```

Gambar 4.13 Hasil Validasi Akurasi

Pada hasil yang dilakukan menunjukkan bahwa model mencapai akurasi pelatihan tinggi, sekitar 97% atau lebih, yang menunjukkan kemampuan model dalam mempelajari pola dari data pelatihan kemudian untuk Akurasi validasi cenderung lebih rendah dari akurasi pelatihan, yang umum terjadi dan menunjukkan tingkat generalisasi model. Dengan val_accuracy sekitar 93%-94%, model menunjukkan performa yang baik, meskipun ada potensi untuk perbaikan lebih lanjut. Meskipun ada sedikit perbedaan antara akurasi pelatihan dan validasi, mekanisme early stopping membantu mencegah overfitting, memastikan model tetap general.

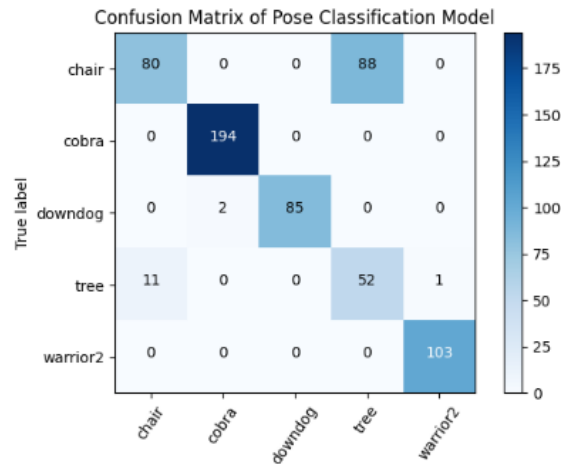
Kemudian untuk hasil model accuracy menunjukkan hasil seperti berikut :



Gambar 4.14 Akurasi Model

Grafik menunjukkan bahwa model dilatih dengan baik dan memiliki performa yang kuat dalam klasifikasi, namun ada ruang untuk perbaikan dalam hal generalisasi. Strategi seperti regularisasi, penambahan data, atau fine-tuning hyperparameters dapat dipertimbangkan untuk lebih meningkatkan akurasi validasi. Meskipun ada sedikit perbedaan antara akurasi pelatihan dan validasi, performa keseluruhan model masih berada pada tingkat yang tinggi, menjadikannya cukup andal untuk penggunaan praktis.

Classification Report:				
	precision	recall	f1-score	support
chair	0.88	0.48	0.62	168
cobra	0.99	1.00	0.99	194
downdog	1.00	0.98	0.99	87
tree	0.37	0.81	0.51	64
warrior2	0.99	1.00	1.00	103
accuracy			0.83	616
macro avg	0.85	0.85	0.82	616
weighted avg	0.90	0.83	0.84	616

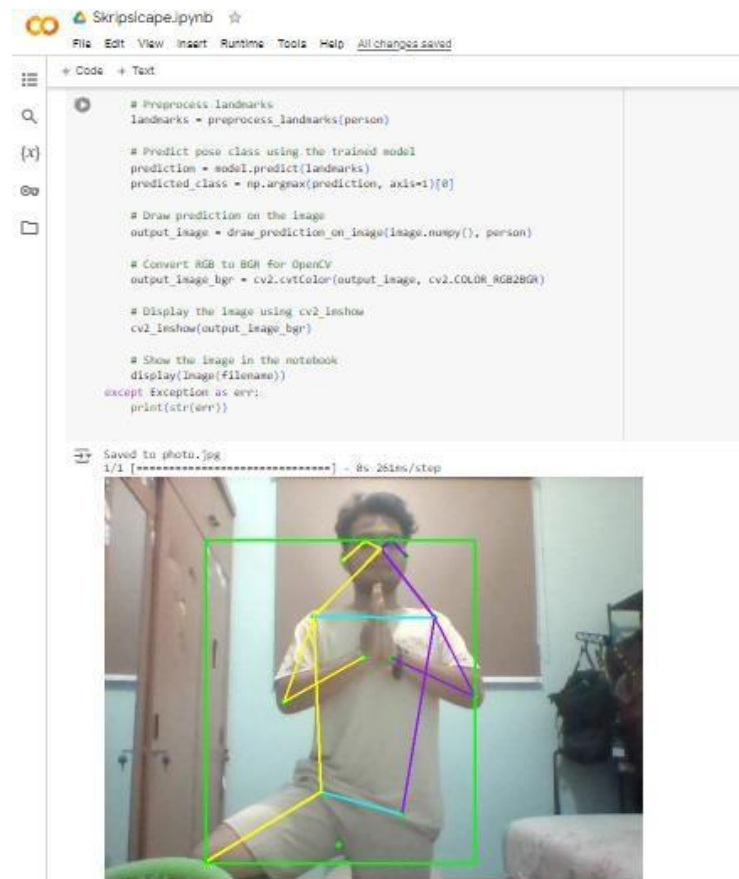


Gambar 4.15 Hasil Laporan Klasifikasi dengan Confusion Matrix

Laporan klasifikasi memberikan gambaran mengenai kinerja model dalam melakukan prediksi terhadap lima kelas pose yang berbeda. Metrik utama yang digunakan adalah precision, recall, dan F1-score, yang masing-masing memberikan informasi penting tentang kualitas prediksi model dan untuk keseluruhan model menunjukkan dengan akurasi 92%. Namun, terdapat kelemahan signifikan pada kelas Chair yang sering salah diklasifikasikan sebagai Tree. Precision dan recall untuk kelas ini lebih rendah dibandingkan kelas lain, yang menunjukkan kebutuhan untuk penanganan lebih lanjut, seperti menambah data pelatihan atau meningkatkan fitur pembeda untuk kelas tersebut.

Performa sangat baik dicapai pada kelas Cobra, Downdog, dan Warrior, dengan precision, recall, dan F1-score yang hampir sempurna. Hal ini menunjukkan model mampu membedakan kelas-kelas ini dengan baik. Secara keseluruhan, model

menunjukkan potensi yang kuat, tetapi ada ruang untuk peningkatan, khususnya dalam menangani ketidakseimbangan dan kesalahan klasifikasi pada kelas tertentu. Kemudian untuk hasil penggunaan secara langsung akan dicoba dengan menggunakan webcam dimana dilakukan pada google collaborator



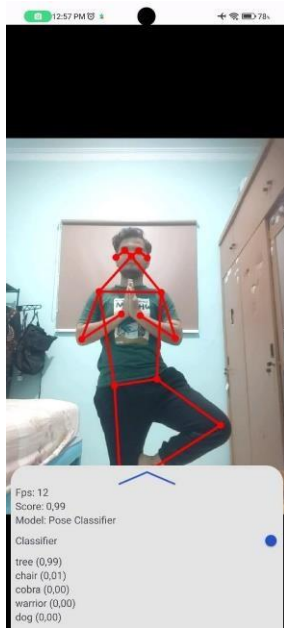
Gambar 4.16 Hasil Deteksi Box Pose Webcam

Pada webcam yang sudah dijalankan menunjukkan bahwa gambar tersebut dapat menggunakan model movenet untuk mendeteksi pose yang sedang dilakukan

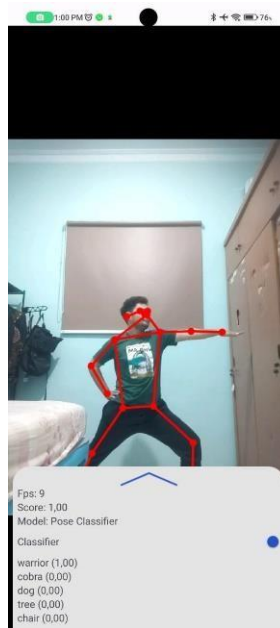
4.4 Implementasi Machine Learning dalam Android

Kemudian setelah proses validasi akurasi dan pelatihan dan testing data melalui google collaboratory , proses akhirnya dilakukan tflite dimana ini menjadi

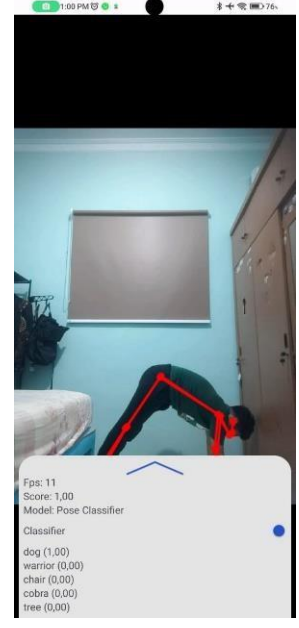
API dimana framework ini dihasil dari model yang sudah dibangun dan dapat digunakan untuk menjalankan model tersebut pada perangkat sumber daya terbatas yaitu mobile application yang telah kita bangun pada kotlin sebelumnya dan menunjukkan hasil yang akurat dimana terbukti pada gambar berikut



Gambar 4. 17 Hasil Tree



Gambar 4.18 Hasil Warrior



Gambar 4.19 Hasil Downdog



Gambar 4.20 Hasil Cobra



Gambar 4.21 Hasil Chair

Pada hasil gambar tersebut menunjukkan adanya akurasi yang tepat dalam gerakan tersebut , dimana deteksi terkait akurasi gerakan tersebut berhasil dimana untuk hasil pose yang berbeda memiliki nilai akurasi yang berbeda seperti Hasil Pose Tree menunjukkan angka akurasi sekitar 0,99 , kemudian untuk hasil pose Warrior Menunjukkan angka akurasi hingga 1.00 , lalu pose dowlndog menunjukkan hasil angka hingga 1.00 , untuk pose cobra juga menunjukkan angka 1.00 dan untuk pose chair sekitar 0.81 . Dimana hasil berikut sudah cukup dapat digunakan pada mobile app dimana model yang dibangun sebelumnya juga memiliki akurasi yang cukup baik dan memiliki kemampuan integrasi yang cukup baik pada penggunaannya di mobile app . sehingga implementasi machine learning pada mobile app berjalan cukup baik dimana model tersebut berjalan secara akurat dan juga ringan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan pada penelitian ini , maka dapat ditarik kesimpulan sebagai berikut :

1. Pada pelatihan dan pembangunan model menggunakan movenet dengan menggunakan google collaboratory. Model mencapai akurasi pelatihan tinggi, sekitar 97% atau lebih, yang menunjukkan kemampuan model dalam mempelajari pola dari data pelatihan. Kemudian untuk Akurasi validasi cenderung lebih rendah dari akurasi pelatihan, yang umum terjadi dan menunjukkan tingkat generalisasi model. Dengan val_accuracy sekitar 93%-94%, model menunjukkan performa yang baik, meskipun ada potensi untuk perbaikan lebih lanjut. Meskipun ada sedikit perbedaan antara akurasi pelatihan dan validasi, mekanisme early stopping membantu mencegah overfitting, memastikan model tetap general dimana ini tetap dapat digunakan ketika nantinya diconvet menggunakan tflite untuk dapat diimpor kedalam mobile application
2. Hasil model yang diimplementasikan kedalam mobile application menunjukkan bahwa gerakan yang kita lakukan secara real time pada handphone memberikan hasil feedback yang cukup akurat dalam mengklasifikasi gerakan kita dimana jika gerakan tidak sesuai maka klasifikasi tersebut tidak terbaca sehingga kita harus mengoreksi

gerakan secara benar dengan panduan gambar sebelumnya dan dibantu dengan bantuan titik keypoint dalam memberikan titik gerakan yang benar dalam melakukan gerakan yoga

5.2 Saran

1. Adanya penambahan fitur fitur yang baru pada mobile application dimana membantu para user / pengguna dalam rutintias yoga yang mereka lakukan seperti pemberian rekomendasi panduan kesehatan yang lain berdasarkan gerakan yoga yang sering dilakukan dan sering digunakan
2. Adanya penambahan pose yoga yang jauh lebih variatif dan memberikan penjelasan secara real time terkait manfaat dari yoga oleh expert dan mentor secara langsung
3. Pengembangan atau perbandingan dengan jenis klasifikasi yoga dengan arsitektur atau model lainnya yang berfokus pada pose estimation model atau human activity recognition sebagai pembanding

DAFTAR PUSTAKA

- Abed, A. A., Al-Ibadi, A., & Abed, I. A. (2023). Real-time multiple face mask and fever detection using YOLOv3 and TensorFlow lite platforms. *Bulletin of Electrical Engineering and Informatics*, 12(2), 922–929. <https://doi.org/10.11591/eei.v12i2.4227>
- Ayuningtyas, E. Y. (2023). Penerapan Senam Yoga Terhadap Tekanan Darah Pada Penderita Hipertensi Di Kelurahan Jebres Surakarta. 1(4), 131–145. <https://doi.org/10.59680/anestesi.v1i4.529>
- Dufan J. P. Manajang, Sherwin R.U.A. Sompie, & Agustinus Jacobus. (2020). Implementasi Framework Tensorflow Object Detection dalam mengklasifikasi jenis kendaraan bermotor. *Jurnal Teknik Informatika*, 15(Jurnal Teknik Informatika), 171–178.
- Elstad, T., Ulleberg, P., Klonteig, S., Hisdal, J., Dyrdal, G. M., & Bjorndal, A. (2020). The effects of yoga on student mental health: a randomised controlled trial. *Health Psychology and Behavioral Medicine*, 8(1), 573–585. <https://doi.org/10.1080/21642850.2020.1843466>
- Gohel, M., Phatak, A., Kharod, U., Pandya, B., Prajapati, B., & Shah, U. (2021). Effect of long-term regular Yoga on physical health of Yoga practitioners. *Indian Journal of Community Medicine*, 46(3), 508–510. https://doi.org/10.4103/ijcm.IJCM_554_20
- Hindarto, H., Sumarno, S., & Rosid, M. A. (2023). Buku Ajar Kecerdasan Buatan/Artificial Intelegent (AI). In *Buku Ajar Kecerdasan Buatan/Artificial Intelegent (AI)*. Umsida Press. <https://doi.org/10.21070/2022/978-623-464-034-2>
- Jo, B. J., & Kim, S. K. (2022). Comparative Analysis of OpenPose, PoseNet, and MoveNet Models for Pose Estimation in Mobile Devices. *Traitement Du Signal*, 39(1), 119–124. <https://doi.org/10.18280/ts.390111>
- Karnadi, B., Lubis, C., Agus,), & Dharmawan, B. (n.d.). *Jurnal Ilmu Komputer dan Sistem Informasi Integrasi Metode Convolutional Neural Networks dengan Arsitektur Model PoseNet untuk Pengembangan Sistem Klasifikasi Gerakan serta Monitoring Repetisi pada Olahraga Bulu Tangkis*.
- Mehindra Prasmatio, R., Rahmat, B., & Yuniar, I. (2020). ALGORITMA CONVOLUTIONAL NEURAL NETWORK. In *Jurnal Informatika dan Sistem Informasi (JIFoSI)* (Vol. 1, Issue 2).
- Mooventhan, A., & Nivethitha, L. (2020). Role of yoga in the prevention and management of various cardiovascular diseases and their risk factors: A comprehensive scientific evidence-based review. In *Explore* (Vol. 16, Issue 4, pp. 257–263). Elsevier Inc. <https://doi.org/10.1016/j.explore.2020.02.007>
- Purna Irawan, Y., Susilawati, I., & Kunci, K. (n.d.). *Klasifikasi Jenis Aglaonema Berdasarkan Citra Daun Menggunakan Convolutional Neural Network (CNN)*.
- Putro Eko Cahyono, & Rolly Maulana Awangga. (2020). *Tutorial Gender Classification Using The You Look Only Once (YOLO)* (Vol. 1). books.google.com.
- Rere, L. M. R., Usna, S., & Soegijanto, D. (2019). Studi Pengenalan Ekspresi Wajah Berbasis Convolutional Neural Network. *Seminar Nasional Teknologi Informasi Dan Komunikasi STI&K (SeNTIK)*, 3.
- Santoso, A., & Ariyanto, G. (n.d.). IMPLEMENTASI DEEP LEARNING BERBASIS KERAS UNTUK PENGENALAN WAJAH. *Jurnal Teknik Elektro*, 18(01). <https://www.mathworks.com/discovery/convol>
- Sharma, A., Shah, Y., Agrawal, Y., & Jain, P. (2022). *REAL-TIME RECOGNITION OF YOGA POSES USING COMPUTER VISION FOR SMART HEALTH CARE A PREPRINT*.
- Singh PK. (2017, July 1). *International Day of Yoga 2017-World Health Organization*. WHO.
- Yudistira, N. (2021). Peran Big Data dan Deep Learning untuk Menyelesaikan Permasalahan Secara Komprehensif. *EXPERT: Jurnal Manajemen Sistem Informasi Dan Teknologi*, 11(2), 78. <https://doi.org/10.36448/expert.v11i2.2063>

- Zuo, X., Yang, X., Dou, Z., & Wen, J. R. (2019). RUCIR at TREC 2019: Conversational Assistance Track. *28th Text REtrieval Conference, TREC 2019 - Proceedings*. <https://doi.org/10.1145/1122445.1122456>
- Motamed, S., & Askari, E. (2022). Recognition of Attention Deficit/Hyperactivity Disorder (ADHD) Based on Electroencephalographic Signals Using Convolutional Neural Networks (CNNs). *Journal of Information Systems and Telecommunication*, *10*(39), 222–228. <https://doi.org/10.52547/jist.16399.10.39.222>
- Moayyed, H., Mohammadpourfard, M., Konstantinou, C., Moradzadeh, A., Mohammadi-Ivatloo, B., & Aguiar, A. P. (2022). Image Processing Based Approach for False Data Injection Attacks Detection in Power Systems. *IEEE Access*, *10*, 12412–12420. <https://doi.org/10.1109/ACCESS.2021.3131506>
- He, X., & Dong, F. (2023). A deep learning-based mathematical modeling strategy for classifying musical genres in musical industry. *Nonlinear Engineering*, *12*(1). <https://doi.org/10.1515/nleng-2022-0302>

DAFTAR LAMPIRAN

Lampiran 1 : Surat Plagiasi

Skripsi Final Print.pdf

ORIGINALITY REPORT

20% SIMILARITY INDEX	19% INTERNET SOURCES	6% PUBLICATIONS	10% STUDENT PAPERS
--------------------------------	--------------------------------	---------------------------	------------------------------

PRIMARY SOURCES

1	repository.umsu.ac.id Internet Source	3%
2	Submitted to Universitas Muhammadiyah Sumatera Utara Student Paper	2%
3	core.ac.uk Internet Source	2%
4	repositori.usu.ac.id Internet Source	2%
5	Submitted to Sriwijaya University Student Paper	1%
6	repository.umi.ac.id Internet Source	1%
7	journal.untar.ac.id Internet Source	1%
8	repository.unej.ac.id Internet Source	1%
9	repository.its.ac.id Internet Source	1%

Lampiran 2 : Source Code

```
[ ] !pip install -q opencv-python

[ ] import csv
import cv2
import itertools
import numpy as np
import pandas as pd
import os
import sys
import tempfile
import tqdm
import tensorflow as tf
import tensorflow_hub as hub

from tensorflow import keras
from matplotlib import pyplot as plt
from matplotlib.collections import LineCollection
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

[ ] # Download Model Movenet
!wget -q -O movenet_thunder.tflite https://tfhub.dev/google/lite-model/movenet/singlepose/thunder/tflite/float16/4?lite-format=tflite
!git clone https://github.com/tensorflow/examples.git
pose_sample_rpi_path = os.path.join(os.getcwd(), 'examples/lite/examples/pose_estimation/raspberry_pi')
sys.path.append(pose_sample_rpi_path)

[ ] import utils
from data import BodyPart
from ml import Movenet
movenet = Movenet('/content/gdrive/MyDrive/VGA/model/movenet_thunder.tflite')

[ ] # Define function to run pose estimation using Movenet Thunder
def detect(input_tensor, inference_count=3):
    """Runs detection on an input image.

    Args:
        input_tensor: A [height, width, 3] tensor of type tf.float32.
            Note that height and width can be anything since the image
            will be immediately resized according to the needs of the
            model within this function.
        inference_count: Number of times the model should run repeatedly
            on the same input image to improve detection accuracy.

    Returns:
        A person entity detected by the Movenet.SinglePose
    """
    image_height, image_width, channel = input_tensor.shape

    # Detect pose using the full input image
    movenet.detect(input_tensor.numpy(), reset_crop_region=True)

    # Repeatedly using previous detection result to identify the
    # region of interest and only cropping that region to improve
    # detection accuracy
    for _ in range(inference_count - 1):
        person = movenet.detect(input_tensor.numpy())

def draw_prediction_on_image(
    image, person, crop_region=None, close_figure=True,
    keep_input_size=False):
    """Draws the keypoint predictions on image.

    Args:
        image: A numpy array with shape [height, width, channel] representing
            the pixel values of the input image.
        person: A person entity returned from the Movenet.SinglePose model.
        close_figure: Whether to close the plt figure after the function returns.
        keep_input_size: Whether to keep the size of the input image.

    Returns:
        A numpy array with shape [out_height, out_width, channel] representing
        the image overlaid with keypoint predictions.
    """
    # Draw the detection result on top of the image.
    image_np = utils.visualize(image, [person])

    # Plot the image with detection results.
    height, width, channel = image.shape
    aspect_ratio = float(width) / height
    fig, ax = plt.subplots(figsize=(12 * aspect_ratio, 12))
    im = ax.imshow(image_np)

    if close_figure:
        plt.close(fig)

    if not keep_input_size:
        image_np = utils.keep_aspect_ratio_resizer(image_np, (512, 512))
```

```

[ ] def evaluate_model(interpreter, x, y_true):
    """Evaluates the given TFLite model and return its accuracy."""
    input_index = interpreter.get_input_details()[0]["index"]
    output_index = interpreter.get_output_details()[0]["index"]

    # Run predictions on all given poses.
    y_pred = []
    for i in range(len(y_true)):
        # Pre-processing: add batch dimension and convert to float32 to match with
        # the model's input data format.
        test_image = x[i: i + 1].astype('float32')
        interpreter.set_tensor(input_index, test_image)

        # Run inference.
        interpreter.invoke()

        # Post-processing: remove batch dimension and find the class with highest
        # probability.
        output = interpreter.tensor(output_index)
        predicted_label = np.argmax(output()[0])
        y_pred.append(predicted_label)

    # Compare prediction results with ground truth labels to calculate accuracy.
    y_pred = keras.utils.to_categorical(y_pred)
    return accuracy_score(y_true, y_pred)

# Evaluate the accuracy of the converted TFLite model
classifier_interpreter = tf.lite.Interpreter(model_content=tflite_model)
classifier_interpreter.allocate_tensors()
print('Accuracy of TFLite model: %s' %
      evaluate_model(classifier_interpreter, x_test, y_test))

```