**Implementasi Metode *CNN* Untuk Klasifikasi Wajah Suku Moi Berbasis Web**

**SKRIPSI**

**Diajukan Sebagai Salah Satu Syarat**

**untuk Memperoleh Gelar Sarjana Komputer**

**Pada Fakultas Teknik Universitas Muhammadiyah Sorong**



**Disusun Oleh :**

**MUHAMMAD THAUFIK**

**NIM : (201955202091)**

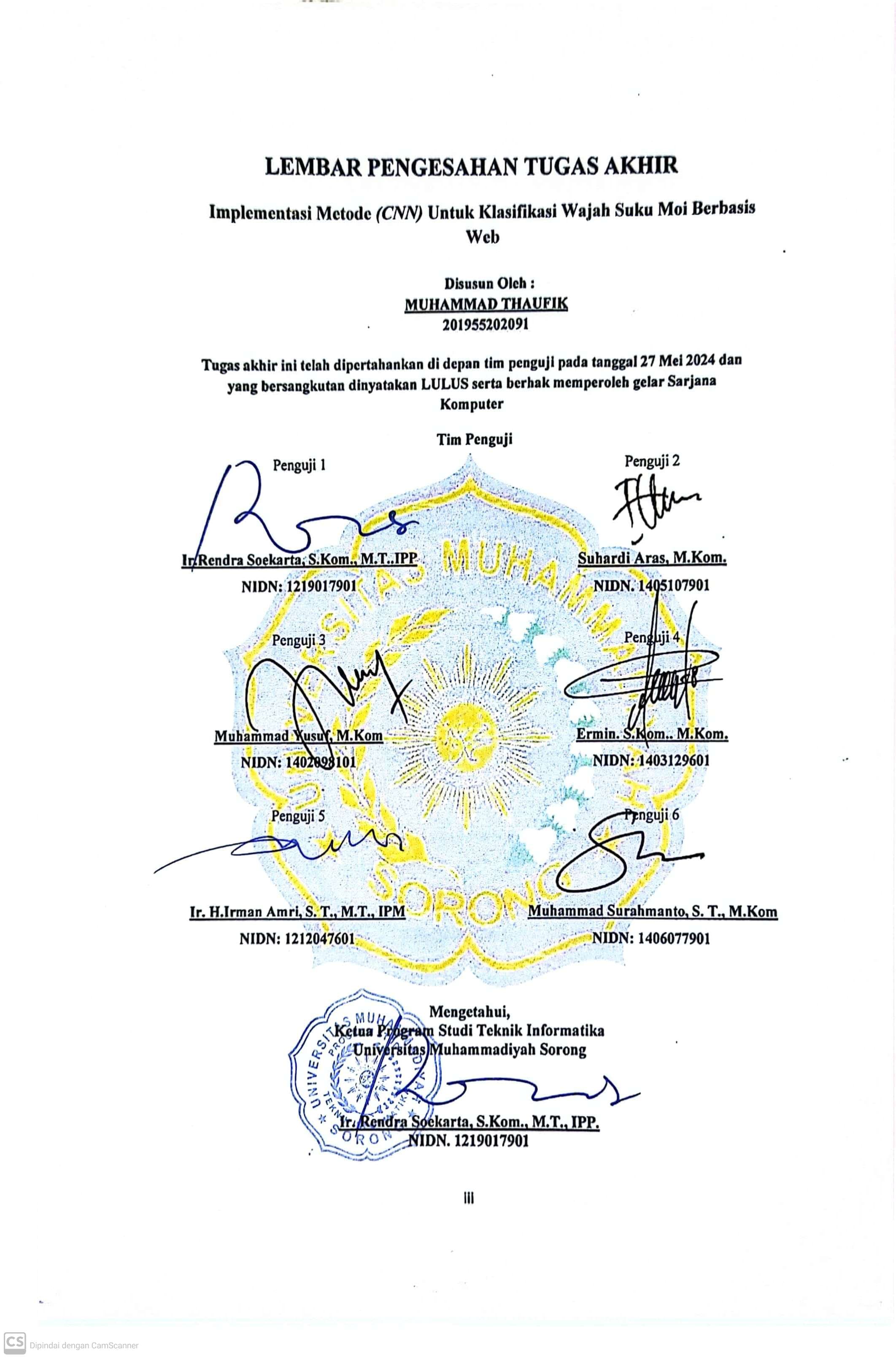
**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MUHAMMADIYAH SORONG**

**2024**

****

****

**ABSTRAK**

**Muhammad Thaufik 201955202091**, **Implementasi Metode *(CNN)* Untuk Klasifikasi Wajah Suku Moi Berbasis Web,**

Pengenalan wajah suku moi adalah sistem yang digunakan untuk mengetahui seberapa besar tingkat akurasi dari implementasi deep learning dalam mengenali wajah ras suku moi. Sistem dirancang khusus yang bertujuan untuk mengetahui bahwa sistem tersebut bisa bekerja dengan baik untuk mengidentifikasi individu orang moi berdasarkan wajah.

Penelitian ini bertujuan untuk mendorong penyusunan dataset pada wajah suku moi yang terletak di Papua Barat Daya penyusunan dataset tersebut dapat digunakan dalam imlpementasi aplikasi-aplikasi pengenalan wajah penelitian ini perlu dilakukan dikarenakan belum adanya dataset tentang pengenalan wajah suku moi di Papua Barat Daya. Metode yang digunakan adalah *Convolution Neural Network (CNN)* yang telah terbukti efektif dalam menghasilkan fitur gambar tingkat tinggi dan dapat digunakan untuk pengenalan wajah, *CNN* adalah metode pembelajaran mesin yang berspesialisasi dalam gambar dan secara otomatis menghasilkan fitur utama dari ilustrasi tanpa memerlukan hasil dari fitur manual. Selain itu *CNN* dikenal lebih efisien dibandingkan metode neural network lainnya, terutama dalam hal penggunaan memori dan pemrosesan citra digital yang kompleks.

Peneltian ini menggunakan dataset berupa gambar berformat jpg yang berjumlah 2 kelas yaitu kelas moi dan kelas non moi dataset yang digunakan berjumlah 627 dataset, 124 data test ( 20%) dan 501 data train (80%) sehingga mendapatkan tingkat akurasi menggunakan *confusion matrix* sebesar 94%.

***Kata kunci\_\_\_****Pengenalan wajah, Klasifikasi etnis, CNN,Deep learning,Djanggo*

## **KATA PENGANTAR**

**Bismillahirrahmanirrahim**

*Alhamdulillahirabbil’aalamiin*, segala puji bagi *Allah Subhanahu Wa Ta’ala* senantiasa tercurahkan karena atas ridha dan rahmat-Nya penulis dapat menyelesaikan laporan hasil penelitian ini dengan judul **“Implementasi Metode *(CNN)* Untuk Klasifikasi Wajah Suku Moi Berbasis Web”** sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) pada Program Studi Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Sorong.

Adapun selama pelaksanaan penulisan laporan ini, penulis mendapatkan banyak bimbingan, dukungan dan doa sehingga dapat menyelesaikannya. Untuk itu, penulis ingin mengucapkan terimakasih kepada :

1. Bapak Dr. H. Muhammad Ali, M.M.., M.H sebagai Rektor Universitas Muhammadiyah Sorong yang telah memberikan kesempatan kepada penulis untuk melanjutkan pendidikan di Universitas Muhammadiyah Sorong.
2. Bapak Ir. H. Hendrik Pristianto, S.T., M.T., IPM sebagai dekan Fakultas Teknik Universitas Muhammadiyah Sorong.
3. Bapak Ir. Rendra Soekarta, S.Kom., M.T., IPP. selaku Ketua Program Studi Teknik Informatika, Universitas Muhammadiyah Sorong.
4. Bapak Ir. Rendra Soekarta, S.Kom., M.T., IPP selaku Dosen Pembimbing I dan BapakSuhardi Aras,M.Kom. selaku Dosen Pembimbing II yang telah memberikan bimbingan, pengetahuan, dan waktu sehingga penulis dapat menyelesikan laporan ini.
5. Seluruh Bapak dan Ibu Dosen di Lingkungan Fakultas Teknik khususnya Program Studi Teknik Informatika.
6. Ibu Miryam Diana Kalagison.S.AN.,M.Si yang telah membantu dalam pengumpulan dataset selama penelitian.
7. Yang teristimewa Ayah dan Ibunda Ucapan Terimakasih yang sebesarbesarnya karena telah mendoakan penulis tiada henti-hentinya, selalu ada dalam berbagai kondisi serta dukungan untuk menyelesaikan studi ini.
8. Untuk Teman Dekat, Teman Seperjuangan, dan Teman-Teman Kelas Teknik Informatika Angkatan 2019, Camping Selebew dan Bacoters yang telah menjadi rumah kedua, menjadi sandaran berbagai kondisi dan partner dalam segala hal.

Semoga Allah Subhanahu Wa Ta’ala memberikan rahmat dan pahala kepada seluruh pihak yang telah membantu penulis dalam menyelesaikan laporan proposal penelitian ini. Akhir kata, penulis ucapkan terimakasih dan semoga laporan proposal penelitian ini dapat bermanfaat bagi penulis dan pembaca.

|  |
| --- |
| Sorong, 9 Februari 2024 |
| Penyusun |

# **DAFTAR ISI**

[ABSTRAK ii](#_Toc163027104)

[KATA PENGANTAR v](#_Toc163027105)

[DAFTAR ISI vii](#_Toc163027106)

[DAFTAR GAMBAR x](#_Toc163027107)

[DAFTAR TABEL xii](#_Toc163027108)

[BAB I 1](#_Toc163027109)

[PENDAHULUAN 1](#_Toc163027110)

[1.1 Latar Belakang 1](#_Toc163027111)

[1.2. Rumusan Masalah 3](#_Toc163027112)

[1.3 Tujuan Penelitian 4](#_Toc163027113)

[1.4 Batasan Masalah 4](#_Toc163027114)

[BAB II 5](#_Toc163027115)

[LANDASAN TEORI 5](#_Toc163027116)

[2.1 *State Of The Art* 5](#_Toc163027117)

[2.2 Literatur Terkait 12](#_Toc163027118)

[2.2.1 Suku Moi 12](#_Toc163027119)

[*2.2.2* *Deep Learning* 13](#_Toc163027120)

[*2.2.3* *Convulutional Neural Network (CNN)* 13](#_Toc163027121)

[*2.2.4* *Efficientnet* 15](#_Toc163027122)

[*2.2.5* *Google Colaboratory* 16](#_Toc163027123)

[*2.2.6* *Website* 17](#_Toc163027124)

[*2.2.7* *Djanggo* 18](#_Toc163027125)

[*2.2.8* *Python* 18](#_Toc163027126)

[*2.2.9* *Tensorflow* 19](#_Toc163027127)

[*2.2.10* *Visual Studio Code* 19](#_Toc163027128)

[*2.2.11* *Draw.io* 20](#_Toc163027129)

[*2.2.12* *Figma* 21](#_Toc163027130)

[*2.2.13* *Flowchart* 21](#_Toc163027131)

[*2.2.14* *UML* 23](#_Toc163027132)

[*2.2.15* *Extreme Programming* 27](#_Toc163027133)

[*2.2.16 Blackbox Testing* 28](#_Toc163027134)

[*2.2.17* *Usability Testing* 28](#_Toc163027135)

[*2.2.18* *Confusion Matrix* 29](#_Toc163027136)

[BAB III 31](#_Toc163027137)

[METODOLOGI PENELITIAN 31](#_Toc163027138)

[3.1 Lokasi dan Waktu Penelitian 31](#_Toc163027139)

[3.1.1 Lokasi Penelitian 31](#_Toc163027140)

[3.2 Alur Penelitian 34](#_Toc163027141)

[3.3 Tahapan Penelitian 35](#_Toc163027142)

[3.3.1 Identifikasi Masalah 35](#_Toc163027143)

[3.3.2 Metode Pengumpulan Data 35](#_Toc163027144)

[3.4 Pengembangan Sistem 38](#_Toc163027145)

[*3.4.1* *Plaining* 38](#_Toc163027146)

[*3.4.2* *Design* 38](#_Toc163027147)

[*3.4.3* *Coding* 44](#_Toc163027148)

[*3.4.4* *Testing* 44](#_Toc163027149)

[*3.4.5* *Release* 45](#_Toc163027150)

[BAB IV 46](#_Toc163027151)

[HASIL DAN PEMBAHASAN 46](#_Toc163027152)

[4.1. Hasil Tahapan Penelitian 46](#_Toc163027153)

[4.1.1 Identifkasi Masalah 46](#_Toc163027154)

[4.2. Hasil pengujian I 48](#_Toc163027155)

[4.3. Hasil Pengujian II 50](#_Toc163027156)

[4.4. Hasil pengujian III 52](#_Toc163027157)

[4.5. Implementasi Model CNN 53](#_Toc163027158)

[4.4.1 Persiapan Dataset 53](#_Toc163027159)

[4.4.2 Preprocessing dataset 54](#_Toc163027160)

[4.4.3 Pembuatan Model 57](#_Toc163027161)

[4.4.4 Training Model CNN 61](#_Toc163027162)

[4.4.5 Pengujian model 63](#_Toc163027163)

[4.4.6 *Saving Model* 65](#_Toc163027164)

[*4.6.* Implementasi *Website* 66](#_Toc163027165)

[4.5.1. Tampilan Halaman Utama 66](#_Toc163027166)

[4.5.2. Tampilan Halaman Klasifikasi 67](#_Toc163027167)

[*4.5.3.* Tampilan Halaman *Probability* 67](#_Toc163027168)

[4.7. Pengujian Sistem 68](#_Toc163027169)

[4.8. *Usability Testing* 69](#_Toc163027170)

[BAB V 82](#_Toc163027171)

[PENUTUP 82](#_Toc163027172)

[5.1 Kesimpulan 82](#_Toc163027173)

[5.2 Saran 82](#_Toc163027174)

[DAFTAR PUSTAKA 84](#_Toc163027175)

[LAMPIRAN 87](#_Toc163027176)

# **DAFTAR GAMBAR**

[Gambar 2. 1 *State Of The Art* 5](#_Toc147599557)

[Gambar 2. 2 Metode CNN 14](#_Toc147599558)

[Gambar 2. 3 Metode *Extreme Programming* 28](#_Toc147599559)

[Gambar 3. 1 Lokasi Penelitian 31](#_Toc157382738)

[Gambar 3. 2 Alur Penelitian 34](#_Toc157382739)

[Gambar 3. 3 Model Pengembangan CNN 36](#_Toc157382740)

[Gambar 3. 4 Flowchart Sistem 39](#_Toc157382741)

[Gambar 3. 5 Use case Diagram Sistem 40](#_Toc157382742)

[Gambar 3. 6 Activity Diagram Deteksi 41](#_Toc157382743)

[Gambar 3. 7 Tampilan Halaman Menu Utama 42](#_Toc157382744)

[Gambar 3. 8 Tampilan Halaman Menu Deteksi 43](#_Toc157382745)

[Gambar 3. 9 Tampilan Halaman Menu Hasil 44](#_Toc157382746)

[Gambar 4. 1 Dataset Pengujian ke-I 49](#_Toc157382856)

[Gambar 4. 2 Dataset Pengujian ke-II 51](#_Toc157382857)

[Gambar 4. 3 Dataset Pengujian ke- III 54](#_Toc157382858)

[Gambar 4. 4 Preprocessing Dataset 55](#_Toc157382859)

[Gambar 4. 5 Pembuatan Model EfficientNet 58](#_Toc157382860)

[Gambar 4. 6 Hasil Training Model 61](#_Toc157382861)

[Gambar 4. 7 Training Akurasi 62](#_Toc157382862)

[Gambar 4. 8 Akurasi Model *Confusion Matrix* 64](#_Toc157382863)

[Gambar 4. 9 Akurasi model *F1-Score* 65](#_Toc157382864)

[Gambar 4. 10 Saving Model 65](#_Toc157382865)

[Gambar 4. 11 Halaman Utama 66](#_Toc157382866)

[Gambar 4. 12 Activity Halaman Klasifikasi 67](#_Toc157382867)

[Gambar 4. 13 Activity Halaman *Probability* 67](#_Toc157382868)

[Gambar 4. 14 Kuisioner Pertanyaan Pertama 69](#_Toc157382869)

[Gambar 4. 15 Kuisioner Pertanyaan Kedua 70](#_Toc157382870)

[Gambar 4. 16 Kuisioner Pertanyaan Ketiga 71](#_Toc157382871)

[Gambar 4. 17 Kuisioner Pernyataan Kesatu 72](#_Toc157382872)

[Gambar 4. 18 Kuisioner Pernyataan Kedua 73](#_Toc157382873)

[Gambar 4. 19 Kuisioner Pernyataan Ketiga 74](#_Toc157382874)

[Gambar 4. 20 Kuisioner Pernyataan Keempat 75](#_Toc157382875)

[Gambar 4. 21 Kuisioner Pernyataan Kelima 76](#_Toc157382876)

# **DAFTAR TABEL**

[Tabel 2. 1 Tabel Perbandingan Penelitian 11](#_Toc155746826)

[Tabel 2. 2 Flowchart Sistem 22](#_Toc155746827)

[*Tabel 2. 3 Simbol Activity Diagram 25*](#_Toc155746828)

[Tabel 2. 4 Simbol *Use Case* *Diagram* 26](#_Toc155746829)

[Tabel 3. 1 Aktivitas Penelitian 33](#_Toc155746830)

[Tabel 4. 1 Jumlah Dataset 48](#_Toc158405788)

[Tabel 4. 2 Hasil Training Dataset 49](#_Toc158405789)

[Tabel 4. 3 Jumlah Dataset 50](#_Toc158405790)

[Tabel 4. 4 Hasil Training Dataset 51](#_Toc158405791)

[Tabel 4. 5 Jumlah Dataset 52](#_Toc158405792)

[Tabel 4. 6 Hasil Training Dtaset 52](#_Toc158405793)

[Tabel 4. 7 Tabel Pengujian Sistem 68](#_Toc158405794)

[Tabel 4. 8 Skor kuisioner 76](#_Toc158405795)

[*Tabel 4. 9 Hasil Kuisioner ke- 1* 78](#_Toc158405796)

[Tabel 4. 10 Hasil Kuisioner ke- 2 78](#_Toc158405797)

[Tabel 4. 11 Hasil Kuisioner ke- 3 78](#_Toc158405798)

[Tabel 4. 12 Hasil Kuisioner ke- 4 79](#_Toc158405799)

[Tabel 4. 13 Hasil Kuisioner ke- 5 80](#_Toc158405800)

[Tabel 4. 14 Hasil Kuisioner ke- 6 80](#_Toc158405801)

[Tabel 4. 15 Hasil Kuesioner ke- 7 81](#_Toc158405802)

[Tabel 4. 16 Hasil Kuesioner ke- 8 81](#_Toc158405803)

# **BAB I**

# **PENDAHULUAN**

## **1.1 Latar Belakang**

Wajah manusia dapat meninggalkan berbagai informasi seperti jenis kelamin, umur, ekspresi, suku bangsa dan sebagainya. Penelitian tentang klasifikasi ras, negara dan suku berdasarkan citra wajah telah membuat topik populer diantara komunitas peneliti *face recognition* (Hartono, Perwitasari, dan Sujaini 2020)*.* Menurut hasil kerja sama antara Badan Pusat Statistik(BPS)dan *Institut of South Asian Studies* (ISEAS) telah dianalisis bahwa terdapat sekitar 633 suku yang diterima dari pengelompokan suku dan sub suku yang ada di indonesia (Putri et al. 2021).

Penelitian yang dilakukan oleh oleh Munarto and Darma (2021) yang berfokus pada penggunaan *CNN* untuk menganalisis citra wajah manusia. Tujuan dari ini untuk mengembangkan model klasifikasi yang dapat secara akurat mengidentifikasi jenis kelamin dan usia dari data citra wajah yang diberikan. Pada penelitian tersebut *CNN* digunakan sebagai metode untuk menunjukkan potensi besar dalam meningkatkan tingkat akurasi dalam mengklasifikasikan jenis kelamin dan usia dari citra wajah hasil yang diperoleh juga memberkan landasan yang kuat untuk penelitian lanjutan dalam bidang pengolahan citra dan kecerdasan buatan. Selain itu pada penelitian tersebut dapat digunakan sebagai dasar untuk mengembangkan aplikasi yang lebih canggih dalam identifikasi individu berdasarkan ciri-ciri wajah mereka. Secara keseluruhan penelitian ini meneruskan pemahaman yang lebih mendalam tentang potensi dan tantangan dalam penggunaan *CNN* dalam pengenalan wajah, sertamemberikan sumbangan berharga bagi perkembangan dalam bidang tersebut.

Penelitian yang dilakukan oleh Armandhani, Wihandika, dan Rahman (2019) yang melakukan penelitian dengan judul klasifikasi gender berbasis wajah menggunakan metode Local Binary Pattern dan Random KNN dengan tujuan mengembangkan sistem klasifikasi gender otomatis berdasarkan gambar wajah yang menggunakan teknologi penglihatan komputer. Metode yang digunakan adalah Local Binary Pattern dan Random KNN yang berfungsi untuk mengekstraksi fitur untuk tujuan identifikasi gender. Penelitian ini memberikan partisipasi penting dalam perluasan pengembangan sistem pengenalan gender yang dapat digunakan dalam berbagai konteks yang dimulai dari keamanan hingga pengalaman pengguna pada peneliti tersebut penelitian ini akan lebih lanjut dalam pengembangan teknologi pengenalan wajah yang lebih canggiih dan akurat.

Penelitian yang dilakukan oleh Kimmo Karkkainen (2019) yang mengelompokkan dataset wajah dari berbagai ras jenis kulit putih, kulit hitam, india, asia timur, asia tenggara, timur tengah, dan latin (Kimmo karkkainen 2019). Sehingga hasil penelitian tersebut mendorong penyusunan dataset wajah suku – suku di indonesia yang nantinya dapat digunakan dalam implmentasi aplikasi – aplikasi pengenalan wajah, salah satunya adalah dataset wajah suku moi yang tinggal di wilayah kota dan kabupaten Sorong.

Penelitian tersebut perlu dilakukan dikarenakan belum adanya dataset tentang pengenalan suku moi di Papua Barat Daya, dan akan berguna pada pengembangan aplikasi identifikasi berbasis pengenalan wajah. Maka peneliti melakukan upaya khusus untuk mengumpulkan data dari etnis masyarakat moi papua dengan memanfaatkan teknologi *Deep Learning* yang merupakan pengembangan dari *Machine Learning*.

*Deep learning* memungkinkan model secara otomatis mempelajari komputasi data yang semakin kompleks (Peryanto, Yudhana, dan Umar 2020) teknologi kecerdasan buatan tersebut telah membuktikan keefektifan model *Deep learning* seperti *Convolution Neural Network (CNN)* yangtelah terbukti berhasil dalammenghasilkan fitur gambar tingkat tinggi dan digunakan untuk pemeriksaan wajah, *CNN* adalah metode pembelajaran mesin yang berspesialisasi dalam gambar dan secara otomatis menghasilkan fitur utama dari ilustrasi tanpa memerlukan hasil dari fitur manual. Selain itu *CNN* dikenal lebih efektif dibandingkan metode neural network lainnya, dan terpenting dalam hal penggunaan memori dan pemrosesan citra digital yang kompleks.

Berdasarkan permasalahan dan pembahasan diatas, penulis mengambil judul “**Implementasi (CNN) Untuk Klasifikasi Wajah Suku Moi Berbasis Web**“ yang diharapkan dapat mengatasi permasalahan diatas.

## **1.2. Rumusan Masalah**

Dari latar belakang permasalahan,adapun rumusan masalah yang dapat diambil dalam penelitian ini yaitu :

1. Bagaimana mengimplementasikan deep learning dengan metode *Convolutuional Neural Network* dalam mengenali wajah ras suku moi berbasis web?
2. Berapa besar tingkat akurasi dari implementasi deep learning dalam mengenali wajah ras suku moi?

## **Tujuan Penelitian**

Adapun tujuan dalam pembuatan proposal tugas akhir ini yaitu :

1. Merancang dan membuat sistem yang dapat mengenali wajah ras suku moi berbasis web.
2. Mengetahui tingkat akurasi dari implementasi *deep learning* untuk mengenali wajah ras suku moi.

## **Batasan Masalah**

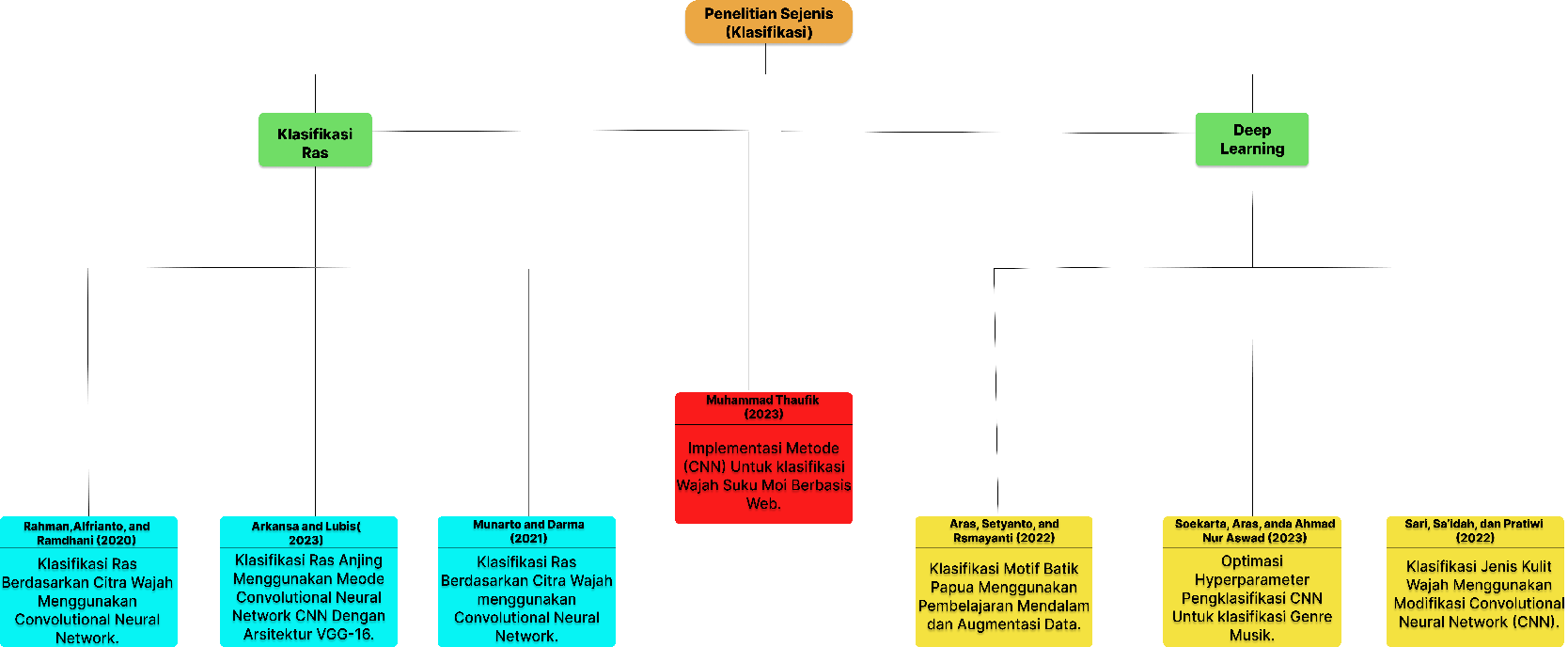
Dari masalah yang telah dirumuskan, batasan masalah dalam penelitian yang akan dilakukan yaitu:

1. Jenis wajah yang diklasifikasikan yaitu suku moi dataset suku moi terdiri dari 2 class suku moi dan non suku moi.
2. Penelitian ini dilakukan di Sorong.
3. Dataset dapat dari hasil foto sendiri.
4. Sistem menggunakan metode *Convolution Neural Network.*
5. Sistem mendeteksi secara single target.
6. Sistem diimplementasikan pada *website.*
7. Training dilakukan pada *google colab* dengan menggunakan bahasa *python.*
8. Sistemmenggunakan forum input gambar.
9. Sistem tidak mengenali batas usia serta jenis kelamin.

# **BAB II**

# **LANDASAN TEORI**

## **2.1 *State Of The Art***

*State of The Art* merupakan kumpulan jurnal yang dijadikan referensi untuk penelitian ini. *State of The Art* juga memberikan penjelasan tentang perbedaan antara penelitian sebelumnya dan saat ini gambar yang dipakai pada perbandingan penelitian bisa dilihat pada gambar 2.1 :

*Gambar 2. 1 State Of The Art*

Penelitian ini bertujuan untuk memperkenalkan klasifikasi dari berbagai Ras atau Suku menggunakan Deplearning, terdapat penelitian-penelitian terdahulu yang melakukan implementasi Deplearning untuk mengklasifikasi Ras, Rahman, Alfianto, and Ramdhani (2020) mengembangkan sistem perangkat lunak otomatis yang dapat mengidentifikasi ras seseorang seperti ras Caucasoid, Negroid, dan Mongoloid. Klasifikasi Ras Anjing dengan memanfaatkan metode *CNN* dengan arsitektur *VGG-16* oleh Arkansa and Lubis. Munarto and Darma (2021) klasifikasi Gender dan Usia Berdasarkan CitraWajah Manusia Menggunakan *CNN*. Namun belum ada penelitian yang meneiti *deeplearning* tentang klasifikasi. Akan tetapi ada beberapa penelitian ynag menggunakan *deeplearning* untuk mengklasifikasi terdapat beberapa penelitian yang mengambil tentang *deeplearning* contoh nya pada penelitian Aras, Setyanto, and Rismayani (2022) klasifikasi Motif Batik Papua Menggunakan Pembelajaran Mendalam dan *Augmentasi* Data. *Optimasi Hyperparameter* Pengklasifikasi *CNN* untuk klasifikasi Genre Musik oleh Soekarta, Aras, and Ahmad Nur Aswad (2023) dan Sari, Sa’idah, and Pratiwi (2022) klasifikasi Jenis Kulit Wajah Menggunakan Modifikasi *CNN.*

1. Penelitian dengan judul Klasifikasi Motif Batik Papua Menggunakan Pembelajaran Mendalam dan *Augmentasi* Data oleh Aras, Setyanto, and Rismayani (2022). Penelitian ini bertujuan untuk meningkatkan akurasi klasifikasi motif batik dengan membandingkan arsitektur pembelajaran mendalam yang berbeda dan mengeksplorasi dampak teknik augmentasi data (Aras, Setyanto, dan Rismayani 2022).Metode yang dimanfaatkan dalam penelitian tersebut adalah *CNN MobilNetV2* dengan *pre-trained transfer learning*. Hasil pada penelitian ini mendapatkan akurasi sebesar 78,79%. Perbedaan dengan penelitian peneliti yang dilakukan, dalam penelitian peneliti akan mengklasifikasikan wajah suku moi dan non moi sedangkan penelitian tersebut mengklasifikasi motif batik papua. **J43**
2. Penelitian dengan judul Optimasi Hyperparameter Pengklasifikasi *CNN* untuk klasifikasi Genre Musik oleh Soekarta, Aras, and Ahmad Nur Aswad (2023) penelitian ini bertujuan untuk meningkatkan akurasi model klasifikasi dengan menemukan kombinasi terbaik dari *hyper* parameter seperti ukuran batch, zaman, dan pemisahan dataset(Rendra Soekarta, Suhardi Aras 2023). Metode yang digunakan dalam penelitian tersebut adalah *CNN* sebagai model klasifikasi untuk klasifikasi genre musik. Hasil dari penelitian ini mendapatkan akurasi sebesar 72% dengan pemisahan data 85:15, ukuran batch 3, dan 500 zaman Perbedaan dengan penelitian peneliti yang dilakukan, dalam penelitian peneliti akan mengklasifikasikan wajah suku moi dan non moi sedangkan penelitian tersebut mengklasifikasi *genre* musik. **J44**
3. Penelitian dengan judul Klasifikasi Ras Berdasarkan Citra Wajah menggunakan *Convolutional Neural Network* oleh Rahman, Arifianto, and Ramdhani (2020) penelitian ini bertujuan untuk mengembangkan sistem perangkat lunak otomatis yang dapat mengidentifikasi ras seseorang seperti ras Caucasoid, Negroid, dan Mongoloid berdasarkan gambar wajah mereka (Rizza Aulia Rahman, Anditya Arifianto 2023). Metode yang digunakan adalah *CNN* untuk klasifikasi ras berdasarkan gambar wajah hasil akurasi training sebanyak 91.63% sesudah dilatih dengan sekitar 5400 citra wajah dan akurasi testing sebanyak 89.739% untuk 600 citra wajah. Pada hasil penelitian diatas mendapatkan hasil cukup tinggi dikarenakan ras Caucasoid, Negroid, dan Mongoloid cukup mencolok untuk membedakan nya sehingga sedikit mudah dalam mengklasifikasikan perbedaan pada wajah dan warna kulit tersebut, sedangkan penelitian yang dilakukan peneliti adalah mengklasifikasi ras suku moi dan non moi melalui wajah cukup sulit untuk membedakan nya dikarenakan tingkat kesamaan pada wajah tersebut hampir mirip. **J17**
4. Penelitian dengan judul Klasifiasi Emosi Berdasarkan Ciri Wajah Menggunakan *Convolutional Neural Network (CNN)* oleh Arkansa and Lubis. Penelitain ini bertujuan untuk dapat mengetahui perilaku konsumen serta pengujian konsumen dikarenakan emosi mempengaruhi perilaku konsumen (Arkansa dan Lubis 2023). Metode yang digunakan *CNN* dalam mengklasifikasikan emosi berdasarkan ciri wajah sehingga pada pengujian mengklasifikasikan emosi pada wajah mendapatkan akurasi sebanyak 86,42% dengan rata-rata akurasi 80,75%. **J46**
5. Penelitian dengan judul Implementasi *Convolutional Neural Networks* *CNN* untuk klasifikasi ekspresi citrah wajah pada *FER2013* dataset oleh Alamsyah and Pratama (2020). Penelitian ini bertujuan akan mengevaluasi kinerja *CNN* dalam menandai ekspresi wajah kebahagiaan dan kesedihan dalam kondisi data unideal menggunakan dataset *FER2013* (Alamsyah dan Pratama 2020). Metode yang dimanfaatkan dalam penelitian ini adalah *CNN*. Tingkat akurasi yang dihasilkan dalam mengenali ekspresi wajah menggunakan *CNN* pada dataset *FER2013* sebesar 66%. Perbedaan dengan penelitian peneliti yang akan dilakukan adalah dalam penelitian peneliti akan menklasifikasi suku moi melalui wajah sedangkan penelitian tersebut mengklasifikasi ekspresi wajah. **J40** **J 40**
6. Penelitian dengan judul judul Eksperimen Layer Pooling menggunakan Standar Deviasi untuk Klasifikasi Dataset Citra Wajah dengan Metode *CNN* oleh Pratama et al. (2023) Penelitian tersebut bertujuan untuk bereksperimen dengan memodifikasi Pooling Layer *CNN* dengan menggunakan perhitungan standar deviasi alih-alih metode pengumpulan maks dan pengumpulan rata-rata yang umum digunakan (Pratama et al. 2023) hasil akurasi tertinggi mencapai 87,853% hasil penelitian menunjukkan bahwa max pooling masih meninggalkan akurasi yang lebih baik dibandingkan melalui perhitungan standar deviasi. Perbedaan dengan penelitian peneliti yang dilakukan dalam penelitian peneliti hanya berfokus pada mengklasifikasikan wajah suku moi dan non moi tanpa ada nya eksperimen dan memodifikasi pada pooling layer sedangkan penelitian tersebut berfokus pada memodifikasi Pooling Layer *CNN* dengan menggunakan perhitungan standar deviasi. **J20**
7. Penelitian dengan judul Klasifikasi Ras Anjing Menggunakan Metode *Convolutional Neural Networks* *CNN* Dengan Arsitektur *VGG-16* oleh Arkansa and Lubis (2023). Penelitian tersebut bertujuan untuk membangun sistem yang dapat mendeteksi anjing(Arkansa dan Lubis 2023). Metode yang dimanfaatkan dalam penelitian ini adalah *Mask R-CNN* merupakan kerangka kerja yang secara konseptual sederhana, fleksibel, dan umum. Hasil pelatihan pada penelitian tersebut mendapatkan akurasi sebesar 88,5% dan  *loss* sebesar 0.49. Perbedaan dengan penelitian peneliti yang dilakukan, dalam penelitian peneliti yaitu mengklasifikasikan wajah suku moi dan non moi sedangkan penelitian tersebut mengklasifikasikan ras anjing. **J45**
8. Penelitian dengan judul Klasifikasi Usia Berdasarkan Citra Wajah Menggunakan Algoritma *Artificial Neural Network* dan *Gabor filter* oleh Melangi (2020). Penelitian ini bertujuan untuk mengklasifikasikan kelompok usia berdesarkan gambar wajah menggunakan Filter Gabor dan algoritma Jaringan syaraf Tiruan (Melangi 2020). Metode yang digunakan pada penelitian tersebut adalah Filter Gabor dan jaringan Saraf Tiruan mencapai akurasi 83% dalam pengenalan usia berdasarkan gambar wajah. Perbedaan dengan penelitian peneliti yang dilakukan dalam penelitian peneliti mengklasifikasikan wajah suku moi dan non moi menggunakan metode *CNN* sedangkan penelitian tersebut mengklasifikasikan kelompok usia berdesarkan gambar wajah dengan menggunakan metode *Artificial Neural Network* dan *Gabor filter.***J5(J.5)**
9. Penelitian dengan judul Klasifikasi Gender dan Usia Berdasarkan CitraWajah Manusia Menggunakan *Convolution Neural Network* oleh Munarto and Darma (2021) Penelitian ini bertujuan untuk membangun model klasifikasi yang dapat secara akurat mengklasifikasikan jenis kelamin dan usia dari data citra wajah yang ada(Munarto dan Darma 2021)*.* Metode yang dipakai dalam penelitian tersebut adalah *CNN* penelitian tersebut dapat mengklasifikasikan gender dengan kinerja akurasi sebesar 74,14%, presisi 78,07% dan sensitivitas 70,65%. Perbedaaan dengan penelitian peneliti adalah dalam penelitian peneliti yang dilakukan hanya mengklasifikasikan wajah suku moi dan non moi menggunakan metode *CNN* sedangkan penelitian tersebut mengklasifikasikan gender dan usia melalui citra wajah manusia dengan memanfaatkan *CNN.***J7**
10. Penelitian dengan judul Klasifikasi Gender berbasis Wajah menggunakan Metode *Local Binary Pattern* dan *Random KNN* oleh Armandhani, Wihandika, and Rahman (2019) Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi gender otomatis berdasarkan gambar wajah menggunakan teknik penglihatan komputer (Armandhani, Wihandika, dan Rahman 2019). Metode yang dimanfaatkan dalam penelitian tersebut adalah metode *Pola Biner Lokal (LBP)* yang digunakan untuk ektraksi fitur tekstur dari gambar yang telah di proses sebelumnya sehingga akurasi yang di peroleh untuk klasifikasi gender menggunakan metode *LBP* mencapai 72,5%. **J9**

Tabel yang digunakan pada perbandingan penelitian dapat dilihat pada tabel 2.1 :

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NO | *Fitur* | Penelitian | | | | | | | | | | |
| PP | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 |
| 1 | Menggunakan  Algoritma CNN |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Implementasi pada Web |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Form input gambar |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Hasil klasifikasi |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Hasil  Akurasi |  |  |  |  |  |  |  |  |  |  |  |

Tabel 2. 1 Tabel Perbandingan Penelitian

Berdasarkan informasi yang tertertera di atas. PP dapat diartikan dengan penelitian peneliti sementara P1-P10 dapat diartikan pada penelitian jurnal terkait. Pada kolom yang terdapat checklist, menunjukkan bahwa fitur-fitur tersebut ada dalam antara kedua jenis penelitian, baik itu dalam penelitian oleh para peneliti maupun dalam jurnal-jurnal tersebut. Pada poin awal, yang melibatkan pemanfaatan algoritma CNN, terdapat kesesuaian antara peneliti yang sedang dilaksanakan dan penelitian pada jurnal-jurnal terkait P1 hingga P10. Meskipun demikian, perlu dicatatat bahwa ada tiga jurnal penelitian tersebut tidak memanfaatkan algoritma CNN diantara nya P8, dan P10. Namun tetap memiliki persamaan dalam aspek klasifikasi wajah berdasarkan kelompok etnis atau ras. Pada poin kedua yaitu implementasi pada web antara lain pada P1, P2, dan P9 yang menggunakan implementasi pada web disisi lain pada P3, P5, P6,P7, P8, dan P10 lainnya, tidak ada indikasi yang menjelaskan platform pelaksanaan yang digunakan, baik itu berupa Android atau platform Web. Pada poin ketiga yaitu forum input gambar terdapat kesesuaian antara peneliti yang sedang dilakukan dan penelitian pada jurnal-jurnal terkait anatara P1 hingga P10 disisi lain hanya ada satu penelitian yang menggunakan forum input gambar di antaranya pada P8. Pada poin keempat yang melibatkan hasil klasifikasi, terdapat kesesuaian antara peneliti yang sedang dilakukan dan penelitian pada jurnal-jurnal terkait dari P1 hingga P10. Pada poin kelima terdapat kesamaan antara peneliti yang dilakukan dan penelitian pada jurnal-jurnal terkait P1 hingga P10.

## **2.2 Literatur Terkait**

## **Suku Moi**

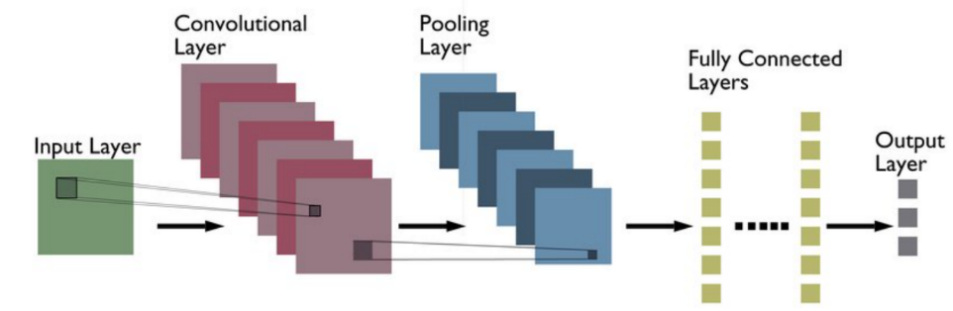
Suku moi merupakan salah satu suku bangsa yang ditempati di daerah terpencil di ujung timur indonesia, tepatnya di wilayah Kepala Burung Papua Barat. Suku moi dikenal sebagai kelompok kedudukan yang berpengaruh dalam wilayah tersebut, dianggap sebagai pemilik yang mempunyai peran penting dalam aspek sosial, politik dan budaya (Makawewe, Syafriny, dan Lakat 2022).

## ***Deep Learning***

*Deep Learning*  dapat mempelajari metode komputasinya sendiri dengan memanfaatkan otaknya sendiri. Teknologi *deep learning* ini merupakan salahsatu teknologi yang amat populer dalam menandai suatu aktivitas atau objek, dengan tingkat akurasi yang lebih tinggi dibandingkan dengan menggunakan metode sebelumnya. Proses *mechine learning* tersebut dikerjakan pada komputer yang mempunyai fungsi mengklasifikasikan data gambar membentuk hasil klasifikasi berupa prediksi. Teknologi *Deep Learning* mengidentifikasikan konsep yang kompleks sebagai rangkaian konsep paling sederhana. *Deep Learning*  adalah algoritma jaringan saraf tiruan yang mengambil data sebagai masukan dan memprosesnya dengan memanfaatkan beberapa lapisan tersembunyi, kemudian melakukan transformasi nonlinier terhadap data yang di masukkan untuk menghitung niali keluaran (Muhammad Haris Diponegoro, Sri Suning Kusumawardani, dan Indriana Hidayah 2021).

## ***Convulutional Neural Network (CNN)***

*Convulutional Neural Network (CNN)* merupakan salah satu algoritma *DeepLearning* yang dapat dimanfaatkan untuk melakukan pengambilan gambar menetapkan weight dan bias pada berbagai aspek dan objek gambar shingga dapat digunakan untuk membedakan gambar satu dengan gambar lainnya *Convulutional Neural Network (CNN)* terdiri dari *input* layer,*output* layer dan *hidden* layer (Haris Alfikri et al. 2022) gambar metode *CNN* yang digunakan dapat dilihat pada gambar 2.2 :



Gambar 2. 2 Metode CNN

*CNN* memiliki kategori komponen utama pada layer atau lapisannya yaitu sebagai berikut :

1. *Input Layer*

*Input layer* adalah lapisan yang bertujuan untuk mengimput gambar yang akan dilatih

1. *Convolutinoal Layer*

*Convolutional Layer* merupakan konvolusi dari output pada layer sebelumnya, Layer ini merupakan proses utama di balik *CNN* dan merupakan lapisan utama terpenting dalam sisitem yang digunakan pada *CNN.* Dalam beberapa proses, *Convolutional Layer* menggunakanfillter di setiap citra yang menjadi *output*.

1. *Pooling Layer*

*Pooling layer* adalah lapisan untuk mengurangi nilai piksel pada gambar dan dapat mengurangi parameter.

1. *Fully Conected Layer*

*Fully Conected Layer* adalah lapisan untuk mengubah gambar menjadi bentuk *vector* untuk menemukan objek sebenarnya.

1. *Output layer*

*Output layer* adalah lapisan yang menampilkan hasil klasifikasi.

## ***Efficientnet***

*EfficientNet* merupakan salah satu arsitektur *Convolutional Neural Network (CNN*) yang dirancang oleh tim *Google Brain,* di pimpin olehMingxing Tan dan Quoc V. Le. Yang bertujuan untuk melakukan pendekatan model *scalling* dimana mereka melakukan peningkatan pada kedalaman model, lebar model, dan resolusi gambar secara bersamaan untuk memperbaiki performa model (Azahro Choirunisa, Karlita, dan Asmara 2021). Ada beberapa lapisan model yang dipakai dalam arsitektur *EfficientNet* dalam model yang dipakai yaitu :

1. Base model yang merupakan bagian dari *EfficientNet* yang berfungsi untuk mengekstraksi fitur-fitur penting dari gambar yang di masukan lalu,
2. *Conv2D Layer* merupakan lapisan dengan 640 filter dan ukuran kernel 1x1 yang bertujuan untuk melakukan transformasi linear pada fitur-fitur yang diekstraksi oleh *base model,*
3. Setelah itu *flatten Layer* digunakan untuk meratakan output dari lapisan sebelumnya menjadi satu vektor dimensi.
4. *Dense Layer* merupakan lapisan *Dense* dengan 320 unit neuron serta menggunakan insialisasi bobot *“he\_uniform”* yang berfungsi untuk memproses fitur-fitur yang telah diratakan sebelumnya secara linear.
5. *BatchNormalization* *Layer* digunakanuntuk menormalkan output dari lapisan sebelumnya kemudian.
6. *ReLU Layer* merupakan fungsi aktivasi yang umum digunakan dalam jaringan saraf tiruan yang bersifat sederhana dan efektif.
7. *Dropout Layer* yang berfungsi untuk mencegah overfitting dengan secara acak.
8. Lalu yang terakhir *Aktivasi softmax* berfungsi menghasilkan distribusi probabilitas dari setiap kelas.

## ***Google Colaboratory***

*Google Colab* adalah IDE untuk pemrograman *Python* yang dimana pemrosesannya akan dilakukan oleh server *Google* dengan komponen perangkat keras berkinerja tinggi, di sisi perangkat lunak, *Google Colab* menyediakan sebagian besar pusataka yang diperlukan. Faktanya *Google Colab* menyediakan semua versi yang dibutuhkan misalnya *TensorFlow* versi 1.x dan 2.x yang dapat disediakan serta versi *Python* dari versi 2.x hingga 3.x disisi *hardware*, *Google Collab* memberikan layanan berupa media penyimpanan bawaan seperti *Google Drive,* prosesor berupa *CPU,GPU,TPU,* dan *RAM.* Dengan jaminan *server* yang normal hampir semua pemrosesan tidak menjadi masalah dalam menjalankan *Google Colab* selama koneksi ke jaringan internet lancar (Gelar Guntara 2023).

## ***Website***

*Website* merupakan suatu tempat diinternet yang berisi berbagai informasi terkait dengan profil pemilik situs. *Website* juga berupa sebuah halaman yang berisi kumpulan halaman situs *web* yang terdapat di internet, dan berperan sebagai sarana untuk menyampaikan informasi, komunikasi, atau melakukan transaksi (Figo, Setiawan, dan Widyassari 2023).

Secara umum *website* mempunyai fungsi sebagai berikut :

1. Fungsi komunikasi, beberapa fasilitas menyediakan fungsi komunikasi seperti obrolan *web, email* dan lainnya.
2. Fungsi informasi, fitur informasi situs *web* seperti berita, profil, *library,* referensi, dan lain-lain.
3. Fungsi intertainment, *website* mempunyai fungsi hiburan, misalnya *web-web* yang menyediakan game *on-line*, *music* *online* dan lain-lain.
4. Fungsi transaksi, *web* dapat digunakan sebagai media transaksi.

## ***Djanggo***

*Djanggo* adalah *framework* full-stack yang akan membantu membangun aplikasi *web* dengan bahasa pemrograman *phyton framework* serta membantu membangun *web* lebih cepat daripada mencatat kode dari nol. Full-stack merupakan *djanggo* yang memiliki kedua sisi yang disebut dengan *front-end* dan *back-end,* yang merupakan sisi depan yang akan dipandang oleh pengguna, dan sisi belakang yang berhubungan dengan *database* dan logika bisnins. *Djanggo* merupakan *framework web* tingkat tinggi pyhton yang memungkinkan pembangunan atau pengembangan dengan cepat dan memiliki desain pragmatis yang bersih (Dirgantara, Suryadarma, dan Mandiri 2019).

## ***Python***

*Python* ialah sebuah bahasa pemrograman tingkat tinggi yang umumnya dimanfaatkan oleh para pengembang dalam proses pembuatan maupun pengembangan aplikasi pada masa kini, dengan dukungan penuh terhadap konsep pemrograman berorientasi objek, *Python* telah dirancang sedemikian rupa agar memudahkan proses pembelajaran dan penggunaan. Tak hanya itu, *Python* dikenal akan fleksibilitasnya yang memungkinkan pengembang untuk mengekspresikan ide-ide kompleks dalam kode yang lebih sederhana (Graciela Fausten Novindri dan Ocsa Nugraha Saian 2022).

## ***Tensorflow***

*Tensorflow* adalah sebuah *framework* *deep learning* yang menjadi pilihan utama dalam komunitas pilihan dan industri. Dikembangkan oleh tim *Google Tensorflow* tidak hanya gratis dan *open source,* tetapi juga berperan penting dalam berbagai aplikasi di bidang data *sciense.* Salah satu aspek penting dari *Tensorflow* adalah *Tensorflow Object Detection API,* suatu alat yang sangat berguna selama fase pembuatan, pelatihan, dan penerapan model deteksi objek. *Tensorflow* merupakan alat yang sangat bermanfaat selama berbagai tahap dalam pengembangan model deteksi objrk.tahapan tersebut melipu tifase perancangan,pelatihan dan implementasi model. Dengan menggunakan *API* ini pengguna dapat dengan mudah membangun , melatih, dan menerapkan model deteksi objek yang akurat dan efisien(Maryati 2021).

## ***Visual Studio Code***

*Visual Studio Code* adalah sebuah perangkat lunak pengedit kode sumber bersifat open source dan dikembangkan oleh *microsoft.* Aplikasi ini kompatibel dengan sistem operasi *Windows, Linux,* dan *MacOS.* Tujuan utama dari *Visual Studio Code* adalah demi membantu menyederhanakan proses *coding* dalam berbagai bahasa pemrograman termasuk C++, C#, *Java, Pyhton, PHP* dan *GO.* Keunggulan utamanya adalah kemampuan untuk mengenali jeniis bahasa pemrograman yang digunakan dan memberikan kemampuan penyorotan sintaksis warna untuk membedakan. Selain itu *Visual Studio Code* juga menonjol dalam kemampuanya untuk mengenali jenis bahasa pemrograman yang dipakai dan memberikan penyorotan sintaksis warna yang konsisten, fitur ini membantu pengguna untuk dengan cepat membedakan elemen-elemen kode dan meningkatkan keterbacaan kode sumber. *Visual Studio Code* juga dilengkapi dengan berbagai fitur lain yang meningkatkan pengalaman pengkodean seperti penyelesaian otomatis, navigasi cepat, debugging terintegrasi dan integrasi dengan sistem kontrol versi seperti Git (Makawewe et al. 2022).

## ***Draw.io***

*Draw.io* adalah platform desain khusus yang memungkinkan pengguna menggabar diagram secara *online.* Aplikasi tersebut dapat diakses melalui *browser* yang membantu *HTML5* dan memerlukan koneksi internet. Kelebihan dari *Draw.io* adalah integrasi dengan *Google Drive* untuk penyimpanan file, dan kemampuan mengekspor diagram berisi format *JPG, PNG, SVG,* dan *XML* hal ini memungkinkan pengguna untuk berbagai diagram dengan format yang sesuai dengan kebutuhan mereka baik untuk keperluan presentasi, dokumentasi, atau kolaborasi. Dengan kombiansi kemudahan penggunaan, integrasi dengan *Google Drive,* dan flektabilitas dalam mengekspor file, *Draw io* menjadi pilihan yang populer bagi individu maupun timyang membutuhkan alat desain diagram yang dapat diakses secara online danmudah digunakan. Hal ini memfasilitasi proses komunikasi visual, kolaborasi dan dokumentasi proyek dengan lebih efisien dan efektif (Eko 2022)*.*

## ***Figma***

*Figma* merupakan aplikasi berbasis web yang digunakan untuk mngembangkan antarmuka pengguna dan pengalaman pengguna *(UI dan UX).* Media ini berguna untuk merancang antarmuka pengguna pada berbagai platform, seperti situs web, aplikasi seluler, dan proyek-proyek lainnya. *Figma* juga memungkinkan *desainer* untuk bekerja sama secara kolaboratif dan membuat desain dengan mudah platform ini memiliki fungsi penting dalam membangun antarmuka pengguna untuk berbagai platform, termasuk situs web, aplikasi seluler, dan proyek-proyek lainnya. Selain itu *figma* juga memberikan kemampuan bagi para desainer untuk bekerja secara kolaboratif memungkinkan pengguna untuk berbagiproyek dan membuat perubahan secara bersama-sama dalam waktu nyata. *Figma* juga menawarkan berbagai fitur yang memudahkan desainer dalam membuat desain dengan mudah pengguna dapat dengan cepat membuat prototipe interaktif, membuat komponen yang dapat digunakan kembali dan mengatur alur kerja pengguna dengan efisien (Rancangan Ui et al. 2023).

## ***Flowchart***

*Flowchart* adalah sebuah alat atau media yang menguraikan langkah yang harus diambil untuk menyelesaikan suatu masalah komputasi. Dengan menggunakan simbol-simbol grafus khusus, *flowchart* memvisualisasikan urutan langkah-langkah secara sistematis dan jelas *flowchart* berfungsi sebagai alat yang sangat berguna dalam berbagai macam konteks mulai dari pengembangan perangkat lunak, sampai proses bisnins dalam perkembangan perangkat lunak *flowchart* dapat digunakan untuk merancang alur logika dalam sebuah program, menjelaskan algoritma, atau merancang antar muka pengguna hal ini dilakukan dengan menjelaskan langkah-langkah menggunakan simbol grafis khusus (Ratumurun dan Joseph 2023).

Keuntungan menggunakan flowchart dalam pemecahan masalah komputasi adalah sebagai berikut :

1. Membiasakan cara berpikir secara terstruktur dan sistematis.
2. Mempermudah pengecekan serta identifikasi bagian-bagian prosedur yang tidak valid atau berlebihan.
3. Memudahkan pengembangan prosedur secara keseluruhan.

Symbol yang digunakan pada *flowchart* sistem dapat dilihat pada tabel 2.2 :

Tabel 2. 2 Flowchart Sistem

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Fungsi** |
| 1. |  | Terminator | Simbol awal dan akhir suatu kegiatan |
| **No** | **Simbol** | **Nama** | **Fungsi** |
| 2. |  | Conector | Penyambungan proses pada halaman yang sama |
| 3. |  | Process | Proses operasional komputer |
| 4. |  | *Decision* | Untuk menunjukkan bahwa suatu kondisi tertentu mengarah pada dua kemungkinan ya/tidak |
| 5. |  | *Flow Direction* | Penghubung antara simbol satu dengan simbol lainnya |
| 6. |  | *Manual*  *Operation* | Menunjukkan pengolahan yang tidak dilakukan komputer |
| 7. |  | *Predefined*  *Process* | Melakukan atau melaksanakan suatu bagian prosedur yang disebut dengan sub bagian |

## ***UML***

Dalam perkembangan teknik pemrograman berorientasi objek, muncul langkah penting dalam upaya standarisasi dalam teknik pengembangan komponen perangkat lunak dengan menggunakan pendekatan tersebut yakni dengan pengenalan *Unified Modeling Language (UML)*. *UML* adalah sebuah bahasa pemodelan visual yang dimanfaatkan untuk merancang, menggambarkan, dan memahami struktur serta interaksi berbagai komponen dalam perangkat lunak berbasis objek (Sunarto Usna , Sudjiran 2023).

1. Activity diagram

*Activiy diagram* menunjukkan aliran dari serangkaian aktivitas. Biasanya digunakan untuk untuk menunjukkan perilaku suatu operasi, aliran *event use case.* *Activity diagram* merupakan salah satu bentuk pemodelan yang dipakai untuk merepresentasikan aktivitas dalam suatu sistem. Diagram ini berguna untuk menggambarkan bagaimana aktivitas-aktivitas dalam sistem berjalan tanpa perlu memperhatikan kode pemrograman atau antarmuka pengguna. Dalam *activity diagram,* alur kerja atau urutan aktivitas dari sistem, proses bisnis, atau bahkan menu dalam perangkat lunak dapat dijelaskan dengan jelas. Diagram ini membantu pemahaman tentang bagaimana suatu sistem atau proses berinteraksi dan memberikan pandangan visual yang mudah dipahami bagi mereka yang terlibat dalam pengembangan atau analisis sistem (Based et al. 2023).

Symbol yang digunakan pada *flowchart* dapat dilihat pada tabel 2.3 :

*Tabel 2. 3 Simbol Activity Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Fungsi** |
| 1. |  | Status awal | Bagaimana objek dibentuk atau dimulai. |
| 2. |  | Activity | Menunjukan kelas yang saling berinteraksi dengan satu sama lain |
| 3. |  | Percabangan | Untuk menunjukkan bahwa suatu kondisi tertentu mengarah pada dua kemungkinan ya/tidak |
| 4. |  | Penggabungan | Penggabungan lebih aktivitas menjadi satu. |
| 5. |  | Status akhir | Penghubung antara simbol satu dengan simbol lainnya |
| 6. |  | Swimlane | Menunjukkan pengolahan yang tidak dilakukan komputer |

1. Use case diagram

*Use case* adalah sbuah gambaran bagaimana sistem berjalan dari sudut pandang pengguna. Ini sebuah uraian tugas atau aktivitas yang harus dikerjakan oleh sistem serta komponen yang terlibat. *Use case* menggunakan scenario untuk menggambarkan urutan langkah-langkah yang menjelaskan interaksi antara pengguna dan sisitem. Dengan demikian, *use case* mengidentifikasi fitur-fitur utama dari sistem yang dimana pengguna bisa berinteraksi dengan sistem, dan bagaimana koneksi antara pengguna dan fungsionalitas sistem terjalin (Setiyani 2021).

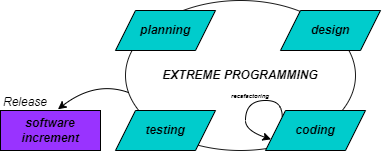
Symbol yang digunakan pada *Use Case Diagram* dapat dilihat pada tabel 2.4 :

Tabel 2. 4 Simbol Use Case Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Fungsi** |
| 1. |  | Status awal | Bagaimana objek dibentuk atau dimulai. |
| 2. |  | Activity | Menunjukan kelas yang saling berinteraksi dengan satu sama lain |
| 3. |  | Percabangan | Untuk menunjukkan bahwa suatu kondisi tertentu mengarah pada dua kemungkinan ya/tidak |
| **No** | **Simbol** | **Nama** | **Fungsi** |
| 4. |  | Penggabungan | Penggabungan lebih aktivitas |
| 5. | <<includee>> | Status akhir | Penghubung antara simbol satu dengan simbol lainnya |
| 6. | <<extend>> | Swimlane | Menunjukkan pengolahan yang tidak dilakukan komputer |

## ***Extreme Programming***

*Extreme Programming* merupakan sebuah pendekatan peningkatan perangkat lunak yang menekankan prinsip sedehana, komunikasi, efektif dan siklus umpan balik yang kuat. Metodologi *XP* dirancang khusus untuk diterapkan oleh tim kecil yang mmerlukan pengembangan perangkat lunak secara cepat, terutama dalam konteks lingkungan yang berubah dengan pesat, terutama dalam konteks lingkungan yang senantiasa berubah dengan cepat. Dengan kesederhanaan pada intinnya, *XP* mendorong tim untuk terus berkomunikasi secara aktif dan memanfaatkan umpan balik untuk menciptakan perangkat lunak berkualitas tinggi. Pendekatan *XP* sendiri memenuhi tantangan pengembangan perangkat lunak skala kecil namun tetap responsif terhadap perubahan yang terjadi pada sekitar proyek (Septiani dan Habibie 2022).



Gambar 2. 3 Metode Extreme Programming

# ***Blackbox Testing***

Metode pengujian *Black Box* atau dikenal juga dengan *Black Box Testing,* adalah metode yang sangat mudah dimanfaatkan dikarenakan hanya memerlukan penentuan batas bawah dan atas dari data yang diharapkan. Pendekatan ini sangat efektif untuk menguji fungsionalitas perangkat lunak tanpa harus memiliki pemahaman mendalam tentang cara kerja kode internal. Dalam metode *Black Box Testing,* perkiraan jumlah data pengujian dapat dihitung berdasarkan jumlah f*ield* data yang akan diuji, serta aturan dan kasus-kasus batas atas dan bawah yang harus diuji (Febriyanti, Sudana, dan Piarsa 2021).

## ***Usability Testing***

*Usability Testing* merupakan salah satu jenis pendekatan dalam evaluasi kegunaan yang digunakan untuk menguji produk secara langsung dengan pengguna. Tujuannya adalah untuk mengidantifikasi permasalahan yang muncul selama penggunaan, mengumpulkan data baik kualitatif maupun kuantitatif, menngukur kemudahan penggunaan, terhadap produk yang dievaluasi (Utami 2020).

Berikut rumus *usability testing* :

Keterangan

Y = Nilai presentase yang dicari (..%)

X= Jumlah nilai kategori jawaban dikalikan dengan frekuensi ( ∑ = N x R)

## ***Confusion Matrix***

*Confusion Matrix* merupakan metode yang dipakai dalam teori data mining untuk menghitung keakuratan suatu model atau algoritma dengan menggunakan *Confusion Matrix,* kita dapat mengukur performa model dalam melakukan klasifikasi data. Dengan digunakan nya matrix ini, kita dapat menghitung berbagai ukuran evaluasi seperti akurasi, presisi, *recall* dan *skor F1* (Normawati dan Prayogi 2021)*.*

Rumus confusion matrix untuk menghitung *accuracy*, *precision,* dan *recall* seperti berikut.

Keterangan :

1. *TP (True Positive)* = Banyak dokumen pada kelas 1 yang bener diklasifikasikan pada kelas 1.
2. *TN (True Negative)* = Banyak dokumen pada kelas 0 yang bener diklasifikasikan pada kelas 0.
3. *FP (False Positive)* = Banyak dokumen pada kelas 0 yang salah diklasifikasikan pada kelas 1.
4. *FN (False Negative)* = Banyak dokumen pada kelas 1 yang salah diklasifikasikan pada kelas 0.

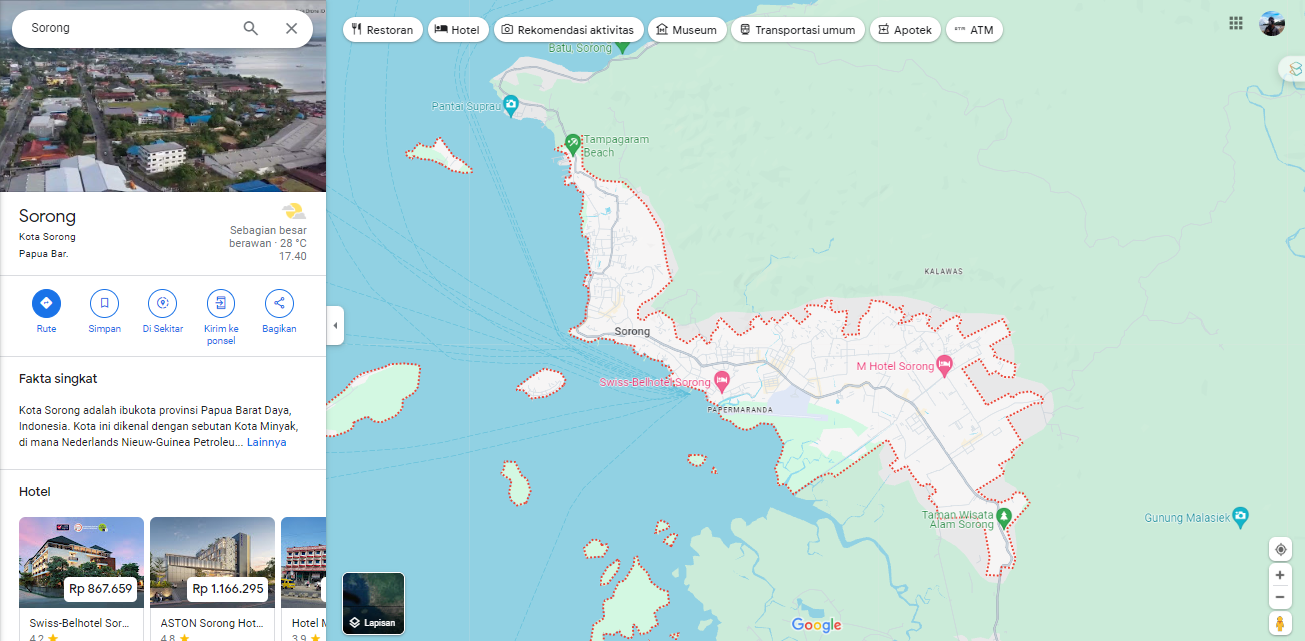
# **BAB III**

## **METODOLOGI PENELITIAN**

## **Lokasi dan Waktu Penelitian**

## **Lokasi Penelitian**

Lokasi yang dijadikan tempat penelitian yaitu di Kota Sorong Tempat penelitian dapat dilihat pada gambar 3.1:



Gambar 3. 1 Lokasi Penelitian

Kota Sorong merupakan kota terbesar di wilayah kepala burung Papua Barat Daya keberadaan Suku Moi turut memberikan warna dan keberagaman budaya, Kota Sorong lebih dikenal sebagai pusat bisnis dan perdagangan, kehadiran Suku Moi dan budaya mereka memberikan kekayaan kultural yang menarik. Suku Moi di Kota Sorong masih mempertahankan beberapa aspek kehidupan tradisional mereka, meskipun telah terjadi berbagai proses akulturasi dengan budaya modern. Selain itu keberadaan Suku Moi di Kota Sorong juga memainkan peran penting dalam melestarikan dan memperkenalkan warisan budaya mereka kepada penduduk setempat dan pengunjung dari luar daerah. Hal ini membantu memperkenalkan pemahaman dan toleransi antarbudaya di Kota Sorong serta memberikan apresiasi terhadap keragaman budaya yang kaya di wilayah Papua Barat Daya.**3.1.2 Waktu Penelitian**

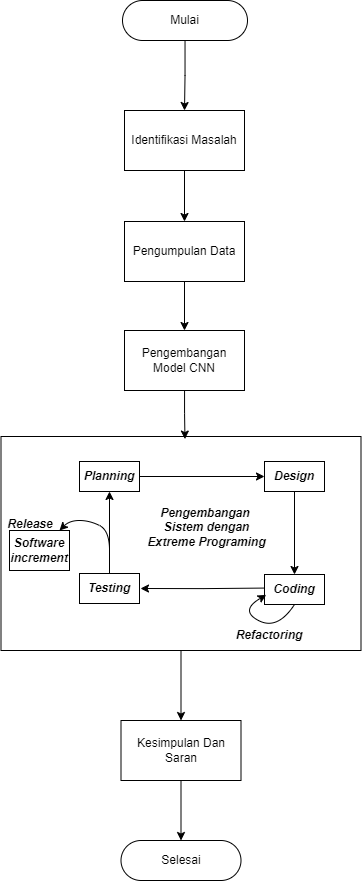
Waktu penelitian yang akan dilakukan yaitu pada bulan September 2023 sampai bulan Maret 2024 *waktu penelitian dapat dilihat pada gambar 3.1:*

Tabel 3. 1 Aktivitas Penelitian

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Aktivitas | Bulan (2023/2024) | | | | | | | | |
| 8 | 9 | 10 | 11 | 12 | 1 | 2 | 3 | 4 |
| 1 | Pengajuan judul |  |  |  |  |  |  |  |  |  |
| 2 | Penyusunan proposal |  |  |  |  |  |  |  |  |  |
| 3 | Pengajuan proposal |  |  |  |  |  |  |  |  |  |
| 4 | Pengumpulan data |  |  |  |  |  |  |  |  |  |
| 5 | Pembuatan sistem |  |  |  |  |  |  |  |  |  |
| 6 | Analisis dan Pembahasan |  |  |  |  |  |  |  |  |  |
| 7 | Implementasi dan pengujian |  |  |  |  |  |  |  |  |  |
| 8 | Seminar  hasil |  |  |  |  |  |  |  |  |  |
| 9 | Seminar tutup |  |  |  |  |  |  |  |  |  |

## **Alur Penelitian**

Langkah-Langkah yang akan dilakukan pada penelitian ini. Untuk alur penelitian nya dapat dilihat dari gambar 3.2 :



Gambar 3. 2 Alur Penelitian

## **Tahapan Penelitian**

## **Identifikasi Masalah**

Tahapan sangat penting dalam penelitian adalah melakukan observasi pada lokasi tersebut untuk mengidentifikasi kelemahan yang mungkin ada dan mengetahui akar permasalahan dari setiap permasalahan yang dihadapi. Agar brtujuan untuk memahami atau mendalami permasalahan yang dihadapi pada lokasi tersebut. Dari permasalahan yang penulis dapatkan yaitu model yang dibuat untuk mengimplementasi klasifikasi wajah suku moi belum ada oleh karena itu penulis melakukan upaya khusus untuk membuat model tersebut agar dapat terimplementasi pada pengenalan wajah suku moi.

## **Metode Pengumpulan Data**

Berikut ini adalah metode pengumpulan data yang digunakan dalam penelitian ini :

1. Studi Literatur

Studi Literatur atau studi kepustakaan dilakukan dengan cara mengumpulkan data melalui teks tertulis atau publikasi elektronik, seperti jurnal, e-book, dan seterusnya yang dapat diperoleh dari internet.

1. Wawancara

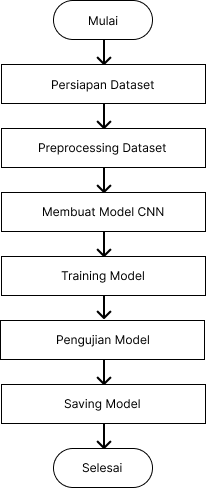
Wawancara adalah suatu metode pengumpulan data yang melibatkan pertemuan langsung antara peneliti dan narasumber di mana informasi diperoleh melalui interaksi tatap muka dan dialog tanya jawab.

1. Observasi

Observasi adalah suatu metode pengumpulan data yang meliputi pengamatan secara langsung ke tempat penelitian untuk mengumpulkan informasi.

1. **Pengembangan Model *CNN***

Pada peneltian ini digunakan model dari arsitektur yang akan diuji untuk dapat mengklasifikasi pengenalan wajah suku moi. Berikut adalah langkah-langkah dalam pengembangan model *CNN* sebagai arsitektur adalah sebagai berikut :



Gambar 3. 3 Model Pengembangan CNN

1. *Dataset*

Dataset yang digunakan terdapat 2 kelas wajah suku moi dan non suku moi.

1. *Preprocessing Data*

Dalam tahap ini dilakukan penyesuaian pada batch size dan pembagian data yang akan digunakan dalam folder validasi, selain itu juga terdapat proses data augmentasi yang melibatkan pengaturan ukuran gambar dan rotasi gambar.

1. Membuat Model *CNN*

Pada tahap ini melibatkan pembuatan sistem yang menggunakan algoritma *CNN* dengan arsitektur untuk mengklasifikasikan wajah suku moi. Sistem ini dirancang untuk mengklaasifikasikan jenis wajah suku moi dan non suku moi.

1. *Training Model*

Pada tahap ini melibatkan penentuan jumlah *Epoch* dan menguji model arsitektur untuk mengevaluasi tingkat akurasi dalam proses klasifikasi gambar.

1. Pengujian Model

Pada tahap ini akan melakukan evaluasi keakuratan klasifikasi gambar, dengan evaluasi ini menggunakan *confusion matrix* sebagai alat untuk mengukur sejauh mana tingkat akurasi model tersebut.

1. *Saving Model*

Pada tahap ini dilakukan untuk minyimpan model atau arsitektur yang akan dipakai.

## **Pengembangan Sistem**

*Metode Extreme Programming* digunakan dalam pengembangan perangkat lunak karena menekankan pada pengkodean dan fleksibilitas ada beberapa fase saat mengembangkan perangkat lunak antara lain:

## ***Plaining***

Analisis kebutuhan adalah tahapan penting dalam pengembangan sistem. Kebutuhan sistem dapat dikelompokkan menjadi dua yaitu Kebutuhan hardware dan kebutuhan software :

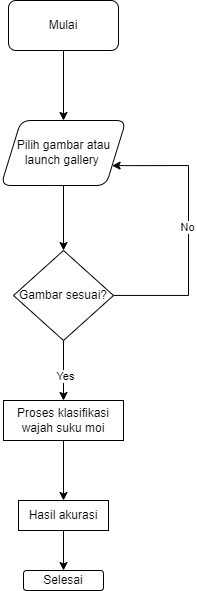
1. Kebutuhan *hardware*
2. Internet/*Access* Point
3. *SSD 240 GB*
4. Laptop *Asus P039K4A* Ram 4GB
5. Kebutuhan *software*
6. Sistem Operasi *Windows 10 Home Single Language*
7. Web *Browser Chrome*
8. *Google Colaboratory*
9. *Visual Studio*

## ***Design***

Dalam proses pemodelan sistem, diselesaikan berdarasarkan hasil analisis kebutuhan seperti *design interface*, *activity diagram, usecase,* dan *flowchart.* Hal ini dilakukan berdasarkan hasil analisis kebutuhan untuk menghubungkan interaksi antara pengguna dan sistem.

1. Flowchart Sistem

*Flowchart Sistem* digunakan untuk menggambarkan alur dari sistem. Berikut gambar *Flowchart sistem* yang telah dibuat yang akan digunakan pada *sistem* dapat dilihat pada gambar 3.4 :

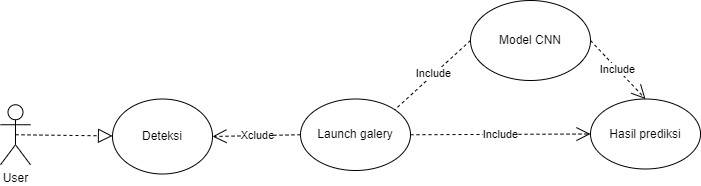


Gambar 3. 4 Flowchart Sistem

Berdasarkan gambar diatas, Flowchart sistem dimulai dengan pengguna melakukan input gambar dari galeri pada aplikasi. Setelah pengguna meng input gambar mengunggah gambar, sistem akan menjalankan proses klasifikasi pada gambar yang telah diinputkan. Proses klasifikasi ini akan melibatkan algoritma atau model machine learning yang akan menganalisis gambar tersebut untuk mengidentifikasi objek, karakteristik, atau informasi tertentu dalam gambar tersebut.

1. *Use case* Diagram Sistem

*Use case* digunakan untuk menggambarkan interaksi antara user dan sistem. Berikut gambar yang merupakan *usecase diagram sistem* dapat dilihat pada gambar3.5 :



Gambar 3. 5 Use case Diagram Sistem

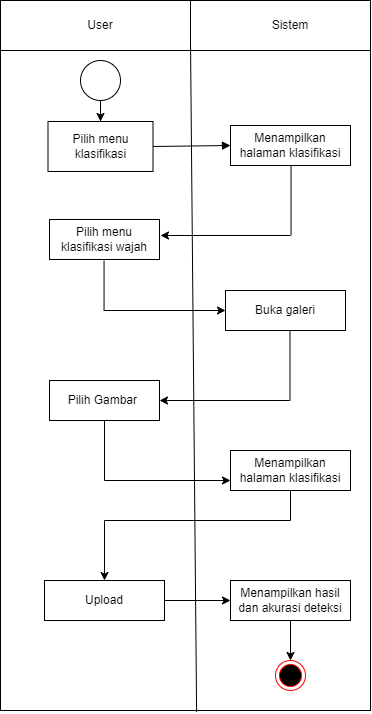
Berdasarkan gambar diatas, user merupakan pengguna yang menjalankan sistem. Pada saat sistem dijalankan user dapat memilih menu deteksi lalu user akan mangambil gambar melalui gallery dari situ sistem akan menganalisis gambar tersebut lewat model CNN setelah itu sistem akan menmprediksi kan hasil dari gambar tersebut.

1. *Activity Diagram*

*Activity diagram* digunakan untuk membantu memahami proses secara keseluruhan *activity diagram* dalam analisis perancangan sistem.

1. *Activity Diagram Deteksi*

Berikut gambar 3.6 *Activity diagram Deteksi* :



Gambar 3. 6 Activity Diagram Deteksi

Berdasarkan gambar di atas, pada saat user menjalankan aplikasi dan memilih menu klasifikasi, sistem akan menampilkan halaman klasifikasi. Kemudian terdapat menu pilih menu klasifikasi wajah. Ketika user menekan klasifikasi wajah tersebut maka sistem akan membuka gallery dan user akan memilih gambar wajah yang akan di deteksi setelah itu sistem akan kembali ke halaman klasifikasi maka akan muncul *button* upload ketika user tekan *button* tersebut sistem akan menampilkan hasil deteksi.

1. *Design Interface*
2. Tampilan Halaman Menu Utama



Gambar 3. 7 Tampilan Halaman Menu Utama

Pada gambar diatas merupakan tampilan halaman menu utama dari aplikasi yang akan dibuat menu tersebut terdiri dari 2 fitur yaitu fitur Home dan klasifikasi, fiur home menampilkan halaman utama sedangkan fitur klasifikasi untuk menampilkan halaman dimana user akan menginput gambar untuk memulai mengklasifikasi, aset yang dipake diambil dari *google* yaitu: *https://images.app.goo.gl/kSnvyaCePumPy7wo6*.

1. Tampilan halaman menu deteksi



Gambar 3. 8 Tampilan Halaman Menu Deteksi

Pada gambar diatas merupakan menu deteksi, menu ini terdiri dari satu button yaitu pilih gallery dimana ketika user menekan button tersebut maka sistem akan menampilkan gambar dari galery untuk dipilih.

1. Tampilan halaman menu hasil



Gambar 3. 9 Tampilan Halaman Menu Hasil

Pada gambar diatas merupakan menu tampilan hasil. Menu tersebut akan menjelaskan kepada user seberapa besar tingkat akurasi yang didapatkan dalam gambar tersebut.

## ***Coding***

Pada tahapan ini developer sepenuhnya fokus pada tahap *coding* (pengkodean) dengan menggunakan bahasa pemrograman *Phyton.* Pembuatan *codingan* menggunakan *Google Colab* untuk menguji dan melatih model tujuan utama dalam melatih model untuk mencapai hasil pelatihan yang optimal. Adapun yang perlu diperhatikan dalam pengembangan meningkatkan kualitas program yaitu *Refactoring* tujuannya agar menjadikan (*XP)* lebih efektif dalam mencapai hasil pelatihan yang optimal.

## ***Testing***

Pada tahap ini, setiap fungsi dan fitur dalam dalam perangkat lunak penguji akan memeriksa secara saksama, jika ditemukan ada nya bug atau cacat bila perlu penguji akan mencatatnya dengan teliti agar membantu pengembang dalam memperbaiki masalah secara langsung pada lokasi di mana kesalahan terjadi. Pengujian awal aplikasi dilakukan menggunakan metode *Blackbox Testing*. Dalam pengujian tersebut akan terfokuskan pada pemeriksaan apakah fitur-fitur yang telah dibuat dalam aplikasi sesuai dengan perpindahan *activity*. Pada tahap kedua, pengujian dilakukan melalui *usability* testing dimana pengguna aplikasi langsung menguji aplikasi tersebut. Pengujian ini bertujuan untuk menerima masukan positif dan saran untuk memperbaiki terkait desain dan fungsi aplikasi yang telah dibuat.

* + 1. ***Release***

Pada tahap ini *release* sebagai perangkat lunak bertahap atau *Incremental* *Release* dalam artian pengembangan perangkat lunak yang dimana sistem tersebut melibatkan pengenalan fitur baru ke dalam project secara berulang dan terus menerus bukan hanya sekali pada akhir Project dalam kerangka *XP* setiap inkrementasi dalam proses ini tidak hanya sekadar menambahkan fitur baru, tetapi juga berfokus pada memberikan nilai tambah kepada pengguna serta meningkatkan kualitas keseluruhan sistem. Dengan demikian pendekatan ini memungkinkan sistem untuk terus berkembang dan beradaptasi sesuai dengan kebutuhan yang berkembang dan umpan balik yang diterima selama prosespengembangan.

# **BAB IV**

# **HASIL DAN PEMBAHASAN**

## **Hasil Tahapan Penelitian**

## Identifkasi Masalah

Berdasarkan hasil identifikasi masalah yang didapatkan penulis bahwa, model klasifikasi wajah suku moi sebelumnya belum mencapai tingkat akurasi yang diharapkan, setelah melakukan proses pengembangan secara teliti, model tersebut berhasil diimplementasikan dan diuji secara menyeluruh sehingga hasilnya menunjukkan bahwa model *efficienet,* memberikan performa yang signiifikan lebih baik dalam mengenali dan mengklasifikasi fitur wajah suku moi sehingga langkah ini tidak hanya meningkatkan akurasi secara keseluruhan, tetapi juga memberikan dasar yang lebih kukuh untuk penelitian lebih lanjut.

1. Metode pengumpulan data
2. Studi Literatur

Beradasarkan hasil dari studi literatur yang penulis gunakan yaitu berupa jurnal yang memiliki keterkaitan anatara penelitian penulis dan penelitian terdahulu, penulis menggunakan enam jurnal terkait yang dimana dibagi menjadi dua pembahasan antara lain, jurnal yang membahas klasifikasi ras serta jurnal yang membahas metode *deeplearnig*.

1. Wawancara

Berdasarkan hasil dari wawancara tersebut penulis melakukan pengujian terhadap orang-orang yang berada di dalam kampus maupun luar kampus untuk mengetahui sistem yang telah dibuat, serta meminta tanggapan secara langsung mengenai sistem yang sudah diuji terkait *UI/UX* dan fitur yang dipakai dalam sistem tersebut.

1. Observasi

Berdasarkan hasil dari observasi yang penulis lakukan adalah penulis melakukan pengumpulan dataset yang meliputi pengambilan gambar secara langsung, penulis mengumpulkan gambar dataset wajah orang moi dan non moi dari beberapa tempat yang ada di Kota Sorong, kemudian setelah dataset tersebut terkumpul data akan ditraining menggunakan *google colab* tujuan nya agar bisa mengetahui berapa persen gambar dataset yang didapatkan dari wajah-wajah trsebut.

## **Hasil pengujian I**

Dalam penelitian ini penulis menggunakan dataset klasifikasi wajah yang terdiri dari 2 kelas yaitu moi yang mengacu pada beberapa jenis wajah orang moi dan non moi merujuk pada beberapa jenis wajah lainnya. Dataset yang digunakan oleh penulis dapat dilihat pada tabel 4.1 :

Tabel 4. 1 Jumlah Dataset

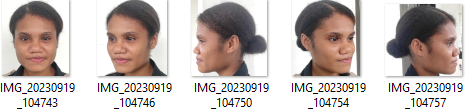
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Kelas** | **Jumlah** | | |
| Train | Test | Validation |
| **1.** | **Moi** | 271 Gambar | 90 Gambar | 90 Gambar |
| **2.** | **Non Moi** | 211 Gambar | 70 Gambar | 70 Gambar |
| **Total** | | 482 Gambar | 160 Gambar | 160 Gambar |

Berdasarkan tabel di atas, dataset terdiri dari tiga kelas kemudaian dibagi menjadi tiga folder yaitu *Train,**Test,* dan *Validation*. Pada folder *train* mengandung dua kelas Moi dan Non moi berjumlah 482 gambar. Folder *test* mengandung dua kelas dengan total 160 gambar. Dan folder mengandung dua kelas yang berjumlah 160 gambar.

Tabel 4. 2 Hasil Training Dataset

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Epoach*** | ***Train akurasi*** | ***Val akurasi*** | ***Test akurasi*** | ***Precission*** | ***F1 score*** | ***Recall*** |  |
| 10 | 0.8379 | 0.7812 | 0,86 | 0.91/0.81 | 0.87/0.85 | 0.83/0.90 |
| 20 | 0.8853 | 0.9062 | 0,77 | 0.78/0.75 | 0.80/0.73 | 0.81/0.71 |
| 30 | 0.9052 | 0.9062 | 0,75 | 0.90/0.65 | 0.74/0.76 | 0.62/0.91 |
| 50 | 0.9327 | 0.9062 | 0,82 | 0.84/0.79 | 0.84/0.79 | 0.83/0.80 |

**Berdasarkan tabel diatas penulis melakukan training dataset melalui Google Colab. Training dilakukan menggunakan 3 folder antara lain tes, train, dan validation sehingga total dataset yang digunakan berjumlah 802 *Val\_akurasi* yang didapatkan, sehingga pada epoach 10 penulis mendapatkan akurasi terbaik sebesar 0,86. Pada foto tersebut penulis melakukan crop gambar yang dimana masih belum berfokus pada wjah yang dideteksi dikarenakan latar belakang pada gambar tersebut masih terlihat. Berikut gambar dataset yang dipakai dimana gambar tersebut masih belum berfokus pada wajah bisa dilihat pada gambar 4.1 :



*Gambar 4. 1 Dataset Pengujian ke-I*

## **Hasil Pengujian II**

Dalam penelitian ini penulis menggunakan dua kelas folder train dan test pada folder train penulis menggunakan 98 gambar yang di pakai dalam kelas moi maupun nonmpoi sedangkan dalam folder test penulis menggunakan 28 gambar yang di pakai dalam kelas moi maupun nonmoi pembagian dataset tersebut dapat dilihat pada tabel 4.3 :

Tabel 4. 3 Jumlah Dataset

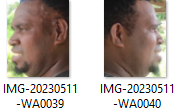
|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kelas** | **Jumlah** | |
| Train | Test |
| **1.** | **Moi** | 98 Gambar | 25 Gambar |
| **2.** | **Non Moi** | 98 Gambar | 25 Gambar |
| **Total** | | 196 Gambar | 50 Gambar |

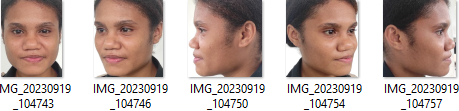
Dalam penelitian ini penulis menggunakan Google Colab untuk men training dataset tersebut sehingga menghasilkan akurasi yang berbeda-beda hasil training dataset dapat dilihat pada tabel 4.4 :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Epoach*** | | ***Train akurasi*** | ***Val akurasi*** | ***Test akurasi*** | | ***Precission*** | ***F1 score*** | ***Recall*** |  |
|  |  |  |  |  |  | |  |  |
| 10 | | 10 | 0,95000 | 0,78 | | 0,69/1,00 | 0,82/0,72 | 1,00/0,56 |
| 20 | | 0,9688 | 0,95000 | 0.80 | | 0,71/1,00 | 0,83/0,75 | 1.00/0.60 |
| 30 | | 0,9886 | 0,95000 | 0.86 | | 0,78/1,00 | 0,88/0,84 | 1.00/0.72 |
| 50 | | 0,9943 | 1.00000 | 0.88 | | 0.81/1.00 | 0.89/0.86 | 1.00/0.76 |

Tabel 4. 4 Hasil Training Dataset

Pada hasil training dataset pengujian ke dua berdasarkan tabel diatas, training dilakukan hanya menggunakan 2 folder antara lain tes dan train sehingga total dataset yang digunakan berjumlah 246 dimana pada gambar tersebut penulis melakukan crop yang sudah berfokus pada wajah yang dideteksi dan menghilangkan sedikit latar yang terlihat sehingga pada epoach 50 penulis mendapatkan akurasi terbaik sebesar 0,88.. Berikut gambar dataset yang dipakai dimana gambar tersebut sudah berfokus pada wajah bisa dilihat pada gambar 4.2 :





Gambar 4. 2 Dataset Pengujian ke-II

## **Hasil pengujian III**

Dalam penelitian ini penulis menggunakan Google Colab untuk mentraining dataset tersebut sehingga menghasilkan akurasi yang berbeda-beda hasil training dataset dapat dilihat pada tabel 4.5 :

Tabel 4. 5 Jumlah Dataset

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kelas** | **Jumlah** | |
| Train | Test |
| **1.** | **Moi** | 359 Gambar | 90 Gambar |
| **2.** | **Non Moi** | 142 Gambar | 36 Gambar |
| **Total** | | 501 Gambar | 126 Gambar |

Dalam penelitian ini penulis menggunakan Google Colab untuk mentraining dataset tersebut sehingga menghasilkan akurasi yang berbeda-beda hasil training dataset dapat dilihat pada tabel 4.6 :

Tabel 4. 6 Hasil Training Dtaset

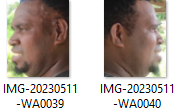
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Epoach*** | ***Train akurasi*** | ***Val akurasi*** | ***Test akurasi*** | ***Precission*** | ***F1 score*** | ***Recall*** |
| 10 | 0.9378 | 0.9608 | 0.91 | 0.92/0.90 | 0.94/0.84 | 0.97/0.78 |
| 20 | 0,9444 | 0.9412 | 0.93 | 0.93/0.94 | 0.95/0.87 | 0.98/0.81 |
| ***Epoach*** | ***Train akurasi*** | ***Val akurasi*** | ***Test akurasi*** | ***Precission*** | ***F1 score*** | ***Recall*** |
| 30 | 0,9667 | 0.9608 | 0,87 | 0.85/1.00 | 0.92/0.71 | 0.85/1.00 |
| 50 | 0,9622 | 0.9608 | 0.94 | 0.94/0.94 | 0.96/0.88 | 0.98/0.83 |

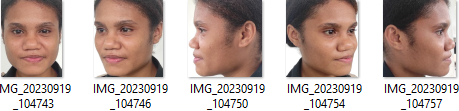
Pada hasil training dataset pengujian ke- tiga berdasarkan tabel diatas, training dilakukan menggunakan 2 folder antara lain folder tes dan folder train pada tahap ini penulis menggunakan dataset dengan total 627 gambar sehingga *Val\_akurasi* yang didapatkan, sehingga pada epoach 50 penulis mendapatkan akurasi terbaik sebesar 0,94.

## **Implementasi Model CNN**

## Persiapan Dataset

Pada tahap ini penulis menggunakan dataset pada pengujian ketiga yang terdiri dari 627 dataset lalu akan dibagi menjadi dua kelas masing-masing yang terdiri dari dua simbol jenis wajah suku moi dan non suku moi. Selanjutnya dataset di upload ke *Google Drive* yang memungkinkan *Google Colaboratory* untuk menyimpan data*.*Berikut ini adalah gambar beberapa contoh dataset yang akan digunakan dalam penelitian ini dapat dilihat pada gambar 4.3:

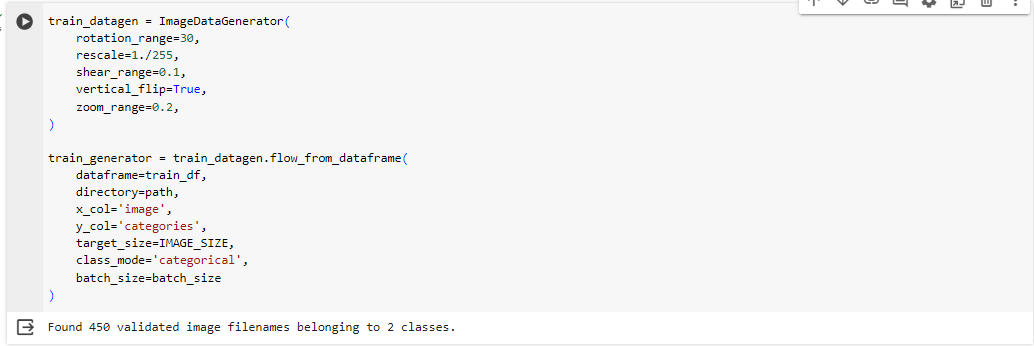


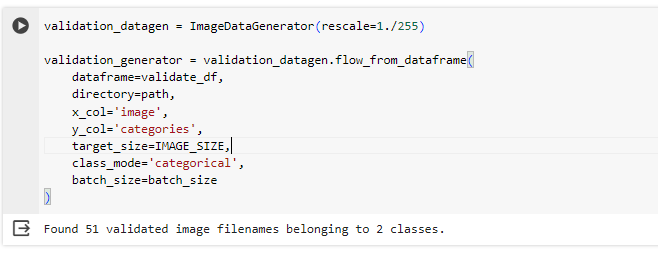


Gambar 4. 3 Dataset Pengujian ke- III

## Preprocessing dataset

Pada tahap ini, jumlah data yang akan disebar ke *neural network* diatur dan pembagian data pada *validation* untuk klasifikasi dilakukan. Berikut *source code Google colab* yang digunakan dapat dilihat pada gambar 4.4:





Gambar 4. 4 Preprocessing Dataset

Pada gambar diatas sistem akan mengatur jumlah dataset pada folder test yang berjumlah 126 gambar dataset. Dan pada folder train yang terdiri dari 501 dataset, terbagi menjadi 450 dataset dikarenakan 51 dataset ditransfer ke folder validation. Selanjutnya pada gambar diatas penulis menggunakan beberapa augmentasi yang bertujuan untuk memungkinkan penulis dalam meningkatkan akurasi data latih ada beberapa parameter lapisan yang dipakai dalam meningkatkan akurasi :

1. *Rotation\_range=30:*

Parameter tersebut memiliki fungsi yang bertujuan untuk mengacu pada augmentasi data dengan rotasi yang dimana gambar-gambar yang digunakan akan diputar secara acak dalam rentang sudut 0 hingga 30 derajat hal ini membantu penulis dalam meningkatkan invariansi rotasi model terhadap variasi sudut pandang objek.

1. *Rescale=1./255:*

Parameter tersebut memiliki fungsi untuk menormalisasikan gambar yang secara umum digunakan dalam pra-pemrosesan gambar yang bertujuan untuk membagi setiap nilai piksel dalam gambar dengan angka 255 serta mengubah rentang intensitas piksel menjadi interval 0,1 dengan menggunakan parameter ini akan membantu penulis dalam menghindari masalah numerik dan mempermudah menyatukan gambar selama proses pelatihan model.

1. *Shear\_range=0.1:*

Parameter tersebut memiliki fungsi untuk menciptakan efek distorsi yang berguna untuk meningkatkan kemampuan model dalam mengenali pola objek dalam berbagai orientasi.

1. *Vertical\_flip=True:*

Parameter ini berfungsi untuk menentukan serta memperkenalkan variasi dengan membalikkan gambar secara vertikal dengan menggunakan parameter tersebut bisa membantu model untuk lebih baik dalam mengidentifikasi objek yang mungkin muncul dalam orientasi yang berbeda.

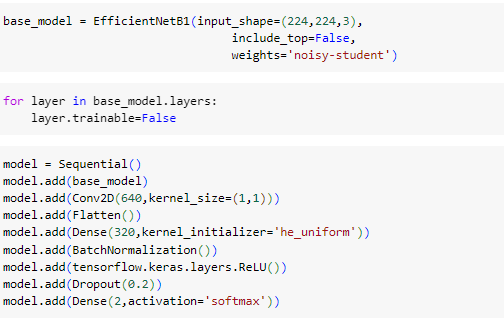
1. *Zoom\_range=0.2:*

Parameter ini mengontrol perbesaran acak yang diterapkan pada gambar dengan memperbesar atau memperkecil gambar secara acak maka kita mperkenalkan variasi dalam skala objek, yang membantu model untuk mempelajari fitur-fitur relevan dalam berbagai ukuran objek.

## Pembuatan Model

Pada tahap ini penulis menggunakan arsitektur *EficientNet* dikarenakan merupakan arsitektur terbaru yang memiliki kemampuan untuk mengklasifikasikan gambar dengan tingkat akurasi yang tinggi dibandingkan dengan arsitektur yang lain *EficientNet* jugadilkembangkan dengan pendekatan yang inovatif dalam merancang skala arsitektur yang memperhitungkan keseimbangan antara kedalaman gambar, lebar gambar dan resolusi gambar yang diinput.

Berikut *source code Google colab* yang digunakan dapat dilihat pada gambar 4.5:

**

Gambar 4. 5 Pembuatan Model EfficientNet

Pada gambar di atas penulis mengggunakan arsitektur *EfficientNetB1* sebagai arsitektur dasar. Keunggulan pada model ini dalam mengatasi proses klasifikasi gambar model tersebut mempunyai fungsi dalam tingkat akurasi yang tinggi dan jumlah parameter yang rendah model *EfficientNetB1* memberikan solusi yang efisien dalam penggunaan *Deep Learning* dalam model yang dibuat terdapat *Noisy student* yang berfungsi untuk meningkatkan kinerja model dengan memanfaatkan data tambahan secara efektif serta menghasilkan representasi fitur yang lebih dalam dan generalisasi yang lebih baik. Selain menggunakan *EfficientNetB1* sebagai dasar model, terdapat lapisan-lapisan *fully conected* yang digunakan dalam arsitektur *EfficientNetB1* sebagai berikut :

1. *Conv2D (640, kernel\_size=(1,1)*:

Lapisan tersebut berfungsi untuk mengubah dimensi representasi fitur yang dihasilkan oleh *EfficientNetB1* dengan menggunakan kernel berukuran (1,1) lapisan ini dapat menghasilkan representasi fitur dengan kedalaman 640.

1. *Flatten( )* :

Lapisan tersebut berfungsi untuk meratakan representasi fitur menjadi vektor satu dimensi sehingga dapat disambungkan dengan lapisan-lapisan *fully conected* berikutnya.

1. *Dense (320, kernel\_intilizer=’he\_uniform’)* :

Lapisan tersebut adalah lapisan *fully conected* dengan 320 unit neuron yang bertujuan untuk mempercepat konvergensi pelatihan model dengan mengatur distribusi awal bobot secara cerdas.

1. *BatchNormalization ( )* :

Lapisan ini berfungsi untuk mempercepat pelatihan dan meningkatkan stabilitas model dengan normalisasi setiap *batch* data akan dimasukkan selama proses pelatihan

1. *ReLU( )* :

Lapisan ini berfungsi untuk memfungsikan aktivasi *Relu (Rectified Liniear Unit )* digunakan untuk memperkenalkan non-lineritas ke dalam model, serta meningkatkan kemampuan model untuk mempelajari representasi fitur yang lebih kompleks.

1. *Dropout(0.2)* :

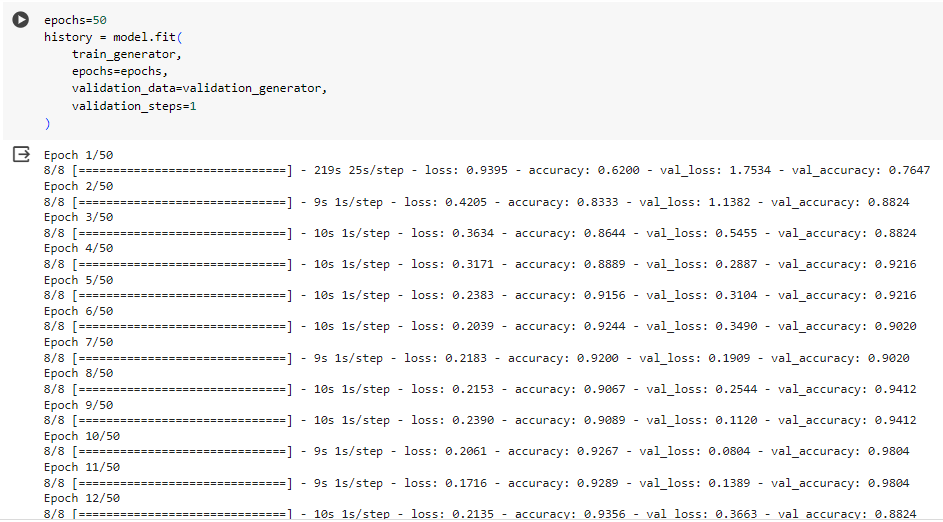
Lapisan ini berfungsi untuk mencegah *overvitting* dengan secara acak mengabaikan sebagaian unit neuron selama pelatihan dengan tingkat dropout sebesar 0.2 yang berarti 20% dari unit neuron akan diabaikan secara acak pada setiap literasi pelatihan.

1. *Dense (2, activation=’softmax’)*

Lapisan ini adalah output yang terdiri dari 2 unit neuron sesuai dengan jumlah kelas yang akan diprediksi lapisan tersebut juga berfungsi untuk mengaktivasi softmax yang digunakan sehingga menghasilkan probabilitas pada kelas-kelas yang berbeda serta memungkinkan model untuk menghasilkan prediksi kelas yang valid.

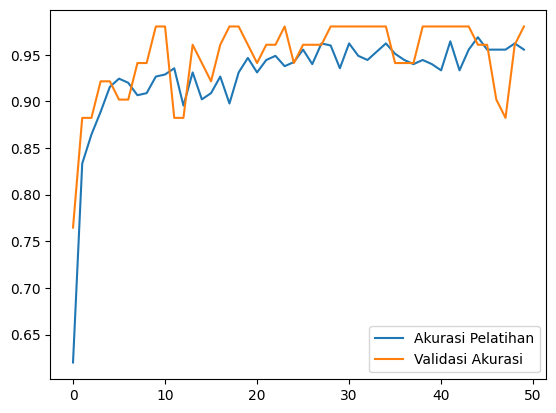
Dengan menambahkan lapisan-lapisan diatas maka akan memberikan model menjadi lebih adaptif terhadap data latih yang telah mengalami augmentasi. Lapisan-lapisan tersebut akan mengarahkan pada pencapaian representasi fitur yang lebih baik serta meningkatkan kemampuan model untuk mengklasifikasikan suatu gambar dengan akurasi yang tinggi.

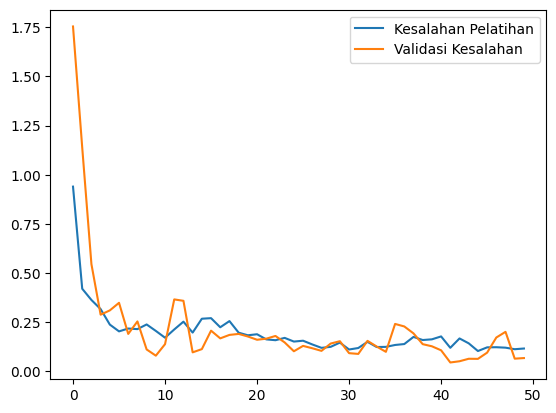
## Training Model CNN

Setelah pembuatan model selesai penulis mulai melakukan training dataset. Berikut *Source Code Google Colab* untuk melakukan training dataset dapat dilihat pada gambar 4.6 :

Gambar 4. 6 Hasil Training Model

Pada gambar diatas, jumlah epoach yang digunakan adalah 50 epoach yang dimana sudah di latih selama beberapa kali sehingga mendapatkan hasil nilai tertinggi sebesar 94% dan mendapatkan nilai val\_accuracy tertinggi sebesar 98%.

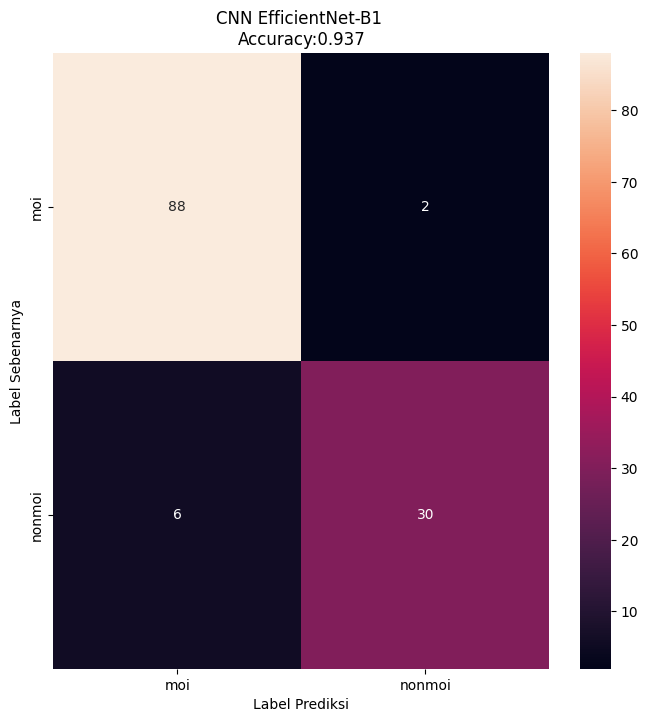




Gambar 4. 7 Training Akurasi

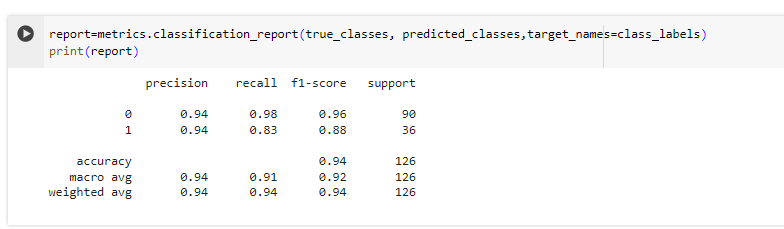
Pada epoach pertama loss: 0.9395 – accuracy : 0.6200 – Val\_loss: 1.7534 – Val\_acucuracy: 0.7647. Sehingga pada epoach 50 loss: 0.1167 – accuracy : 0.9556 – Val\_loss: 0.0682 – Val\_acucuracy : 0.9804. Hal ini dapat dikatakan cukup baik karena statistik pada akurasi masih sedikit stabil.

## Pengujian model ke- III

Pada tahap ini, evaluasi tingkat akurasi arsitektur EfficientNet serta penilaian label prediksi pada sistem yang telah dibuat dilakukan dengan menggunakan *Confusion Matrix* dan *F1- Score.* Model *Confusion Matrix*  yang dipakai mengunakan hasil pengujian model ke-III.

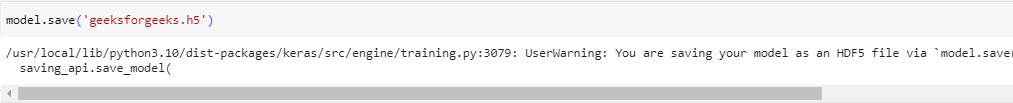
*Gambar 4. 8 Akurasi Model Confusion Matrix*

Berdasarkan gambar di atas prediksi pada arsitektur *EfficientNet* menggunakan *Confusion Matrix.* Pada hasil pengujian ke- III kelas moi terdapat 88 gambar yang diprediksi berhasil dan 2 gambar di prediksi gagal. Pada pengujian kelas nonmoi terdapat 30 gambar yang diprediksi berhasil dan 6 gambar yang diprediksi gagal. Perhitungan *Confusion Matrix* secara manual :

Gambar 4. 9 Akurasi model F1-Score

Prediksi pada arsitektur *EfficientNet* menggunakan *F1-Score* pada pengujian kelas seperti yang ditunjukkan pada gambar diatas. Pada kelas angka (0) yang dimana disebut sebagai kelas moi dan pada kelas angka (1) yang dimana disebut sebagai kelas moi. *Precision* 94%,*recall* 98% dan *F1-Score* 96% dapat dicapai pada pengujian kelas moi sedangkan pada kelas nonmoi yang didapatkan *Precision* 94%,*recall* 83% dan *F1-Score* 88%. Akurasi sebesar 94% telah dicapai dengan menggunakan model *EfficientNet* yang menunjukkan kemampuan luar biasa model dalam mengenali dan mengklasifikasikan data.

## Saving Model

Pada tahap ini adalah langkah yang dilakukan untuk menyimpan model yang telah dibuat. Berikut *Source Code* yang digunakan untuk menyimpan model *Google Colab* bisa dilihat pada gambar 4.10 :

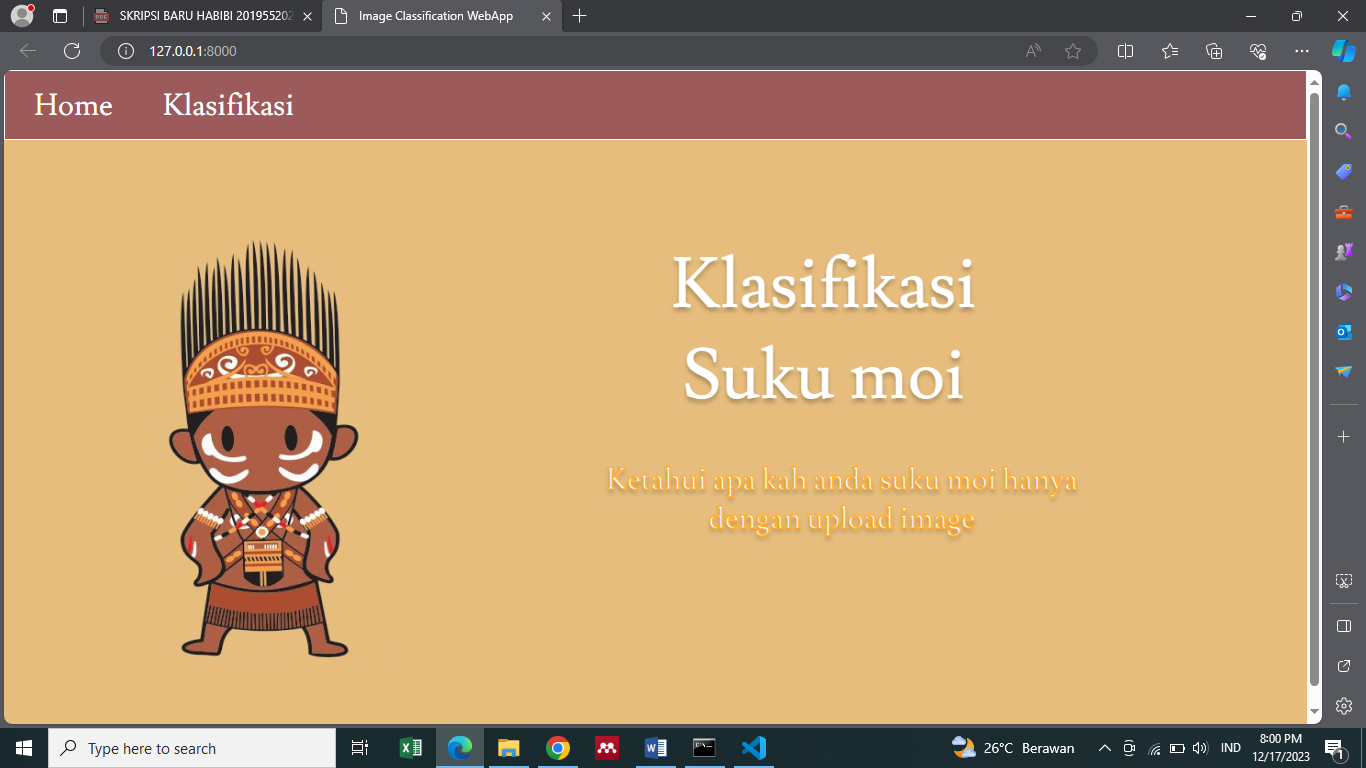
Gambar 4. 10 Saving Model

Berdasarkan gambar di atas pada tahap ini dilakukan untuk mengubah model *Google Colab* menjadi h5 seperti yang ditunjukan pada gambar di atas yang dimana memungkinkan sistem tersebut untuk berjalan pada aplikasi *web*.

## **Implementasi *Website***

Pada aplikasi ini dibuat menggunakan bahasa *pyhton* yang akan diimplementasikan pada website dan kemudian akan dilakukan training pada *google colab* menggunakan bahasa *pyhton.*

## Tampilan Halaman Utama



Gambar 4. 11 Halaman Utama

Pada gambar di atas merupakan menu utama dari 3 fitur anatara lain *Home page*, Klasifikasi dan *Probability*. Pada gambar diatas merupakan tampilan halaman utama dimana ketika *user* memasuki halaman utama.

## Tampilan Halaman Klasifikasi

Gambar 4. 12 Activity Halaman Klasifikasi

Pada gambar diatas merupakan tampilan menu ke- 2 yang dimana ketika *user* menekan klasifikasi *user* akan di bawah ke halaman galeri untuk memilih gambar yang akan *user* pilih untuk di klasifikasikan setelah itu *user* akan menekan *upload* dan akan meembawa *user* ke halaman *probability*.

## Tampilan Halaman *Probability*



Gambar 4. 13 Activity Halaman Probability

Pada gambar diatas merupakan tampilan menu ke- 3 yang dimana sistem akan menampilkan halaman *probability*, pada halaman tersebut menampilkan presentase yang memiliki dua class antara lain moi dan nonmoi serta angka probability yang ditampilkan.

## **Pengujian Sistem**

Pada tahap ini, pengujian dilakukan untuk menguji semua fitur aplikasi yang telah dibuat untuk memastikan bahwa semua fitur berfungsi dengan baik. Tabel berikut menunjukkan hasil pengujian yang dilakukan bisa dilihat pada tabel 4.7 :

Tabel 4. 7 Tabel Pengujian Sistem

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | | Aktivitas  Pengujian | | Skenario  Pengujian | Hasil yang diharapkah | | Kesimpulan | |
| 1. | | Menekan menu klasifikasi | | User akan menekan menu klasifikasi yang ada pada menu utama | User akan menuju ke halaman klasifikas | | Sesuai | |
| 2. | | Menekan *button* klasifikasi | | User akan menekan *button* dengan keterangan klasifikasi | User akan menuju ke halaman galeri | | Sesuai | |
| 3. | Menekan *button upload* | | User akan menekan *button upload* | | | User akan menuju ke halaman *probability* | Sesuai |

Berdasarkan tabel di atas dapat disimpulkan bahwa aplikasi dapat berjalan dengan baik dan memenuhi harapan pengembang dari hasil analisis yang dapat disimpulkan yaitu bahwa aplikasi tersebut dapat berjalan secara optimal dan memenuhi semua kemampuan yang telah dilakukan oleh pengembang. Performa sistem selama melakukan pengujian memperlihatkan konsistensi yang memuaskan dalammenjalankan fungsi-fungsi utama dan fitur-fitur yang disediakan.

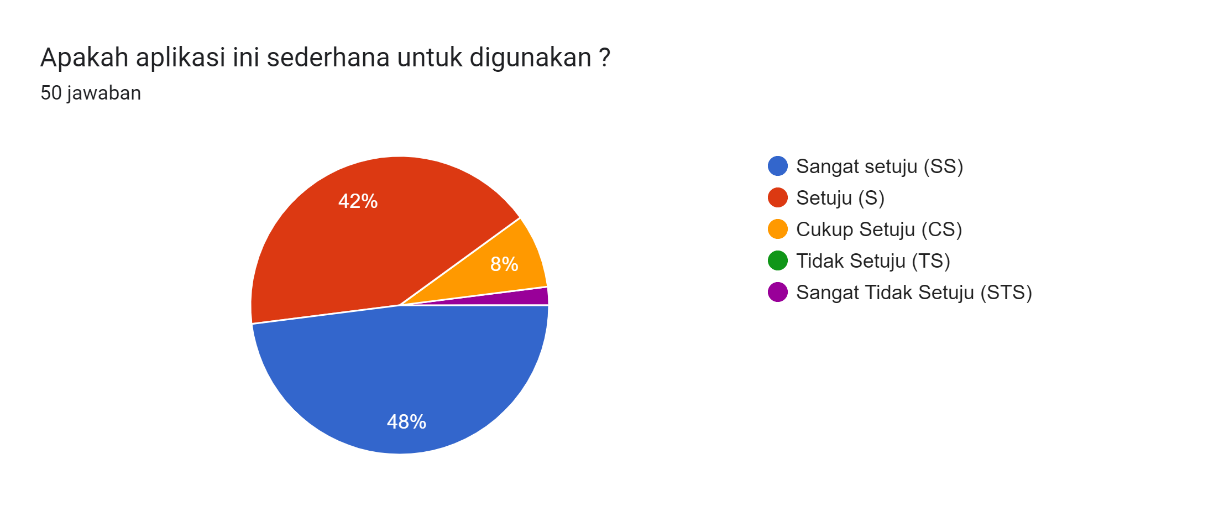
## ***Usability Testing***

Berdasarkan hasil *usabilitiy testing* yang di dapatkan penulis melakukan pengujian sistem serta melakukan tesing secara langsung dengan total wajah berjumlah 32 orang yang disertai oleh 16 wajah orang moi dan 16 wajah orang non moi serta mendapatkan 50 responden yang telah penulis dapatkan dalam *google form* berikut hasil kuisioner yang dikumpulkan dari 50 responden melalui *google form* serta data presentase dari kuisioner yang telah didapatkan dapat dilihat pada gambar 4.14 :



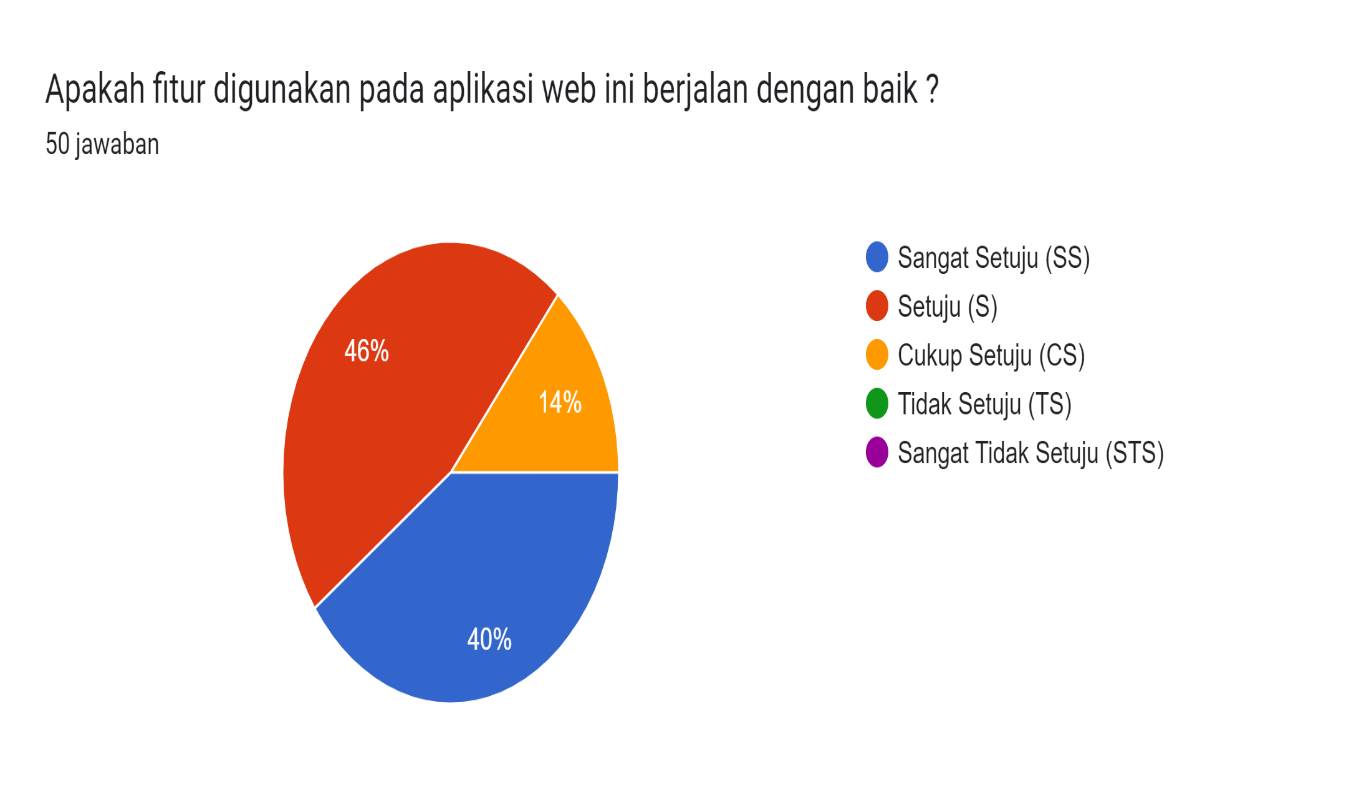
Gambar 4. 14 Kuisioner Pertanyaan Pertama

Berdasarkan tanggapan dari 50 responden terhadap pertanyaan yang diberikan, pertanyaan awalnya adalah apakah aplikasi ini berfungsi dengan baik ? Ditemukan bahwa 40% dari responden sangat setuju, yang diwakili oleh 20 jawaban. Dan sebanyak 48% responden yang mengisi jawaban setuju dan diwakili oleh 24 jawaban. Disusul dengan 12% dari reponden yang mengisi jawaban cukup setuju dan diwakli oleh 6 jawaban.



Gambar 4. 15 Kuisioner Pertanyaan Kedua

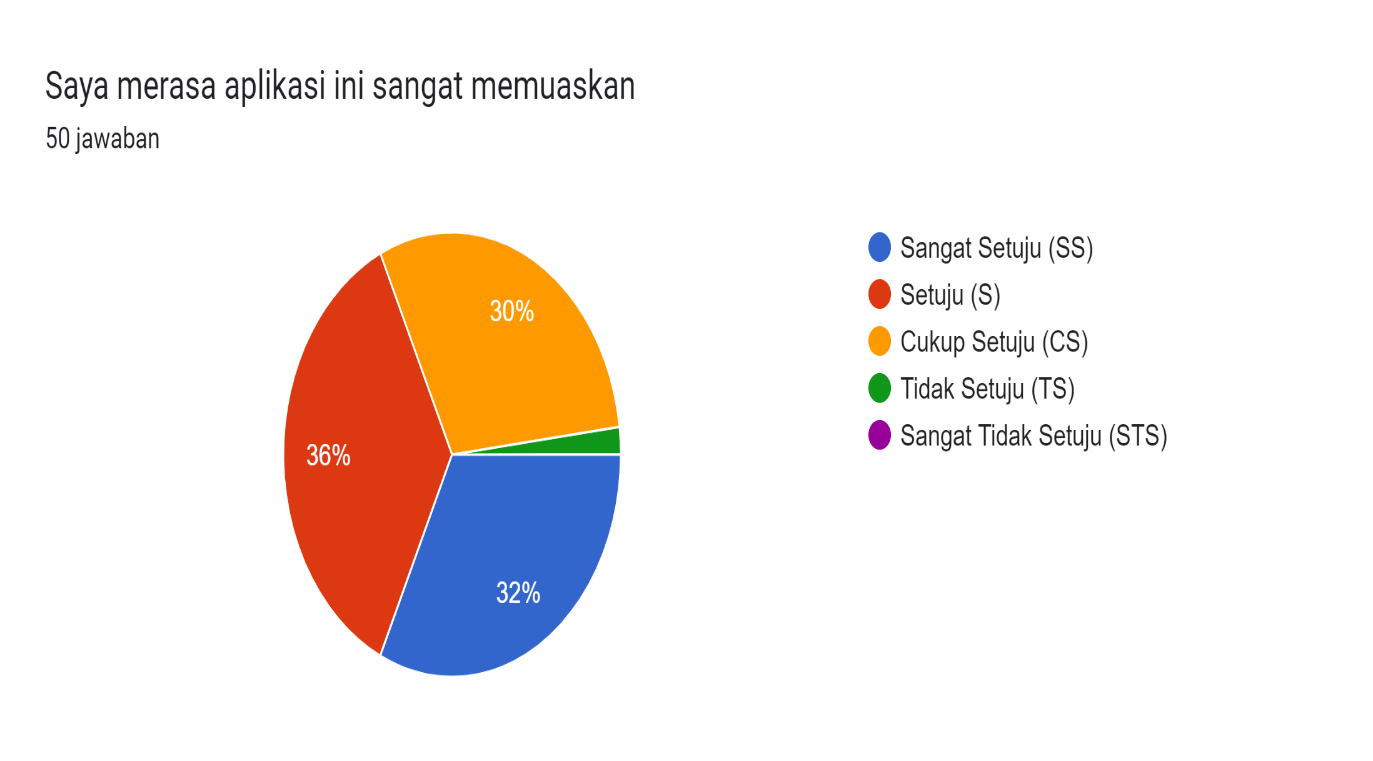
Berdasarkan tanggapan dari 50 responden terhadap pertanyaan yang diberikan, pertanyaan pertanyaan kedua adalah apakah aplikasi ini sederhana untuk digunakan ? Ditemukan bahwa 48% dari responden sangat setuju, yang diwakili oleh 24 jawaban. Dan sebanyak 42% responden yang mengisi jawaban setuju dan diwakili oleh 21 jawaban. Disusul dengan 8% dari reponden yang mengisi jawaban cukup setuju dan diwakli oleh 4 jawaban. Kemudian disusul lagi dengan 2% dari reponden yang mengisi jawaban sangat tidak setuju yang diwakili oleh 1 jawaban.

Gambar 4. 16 Kuisioner Pertanyaan Ketiga

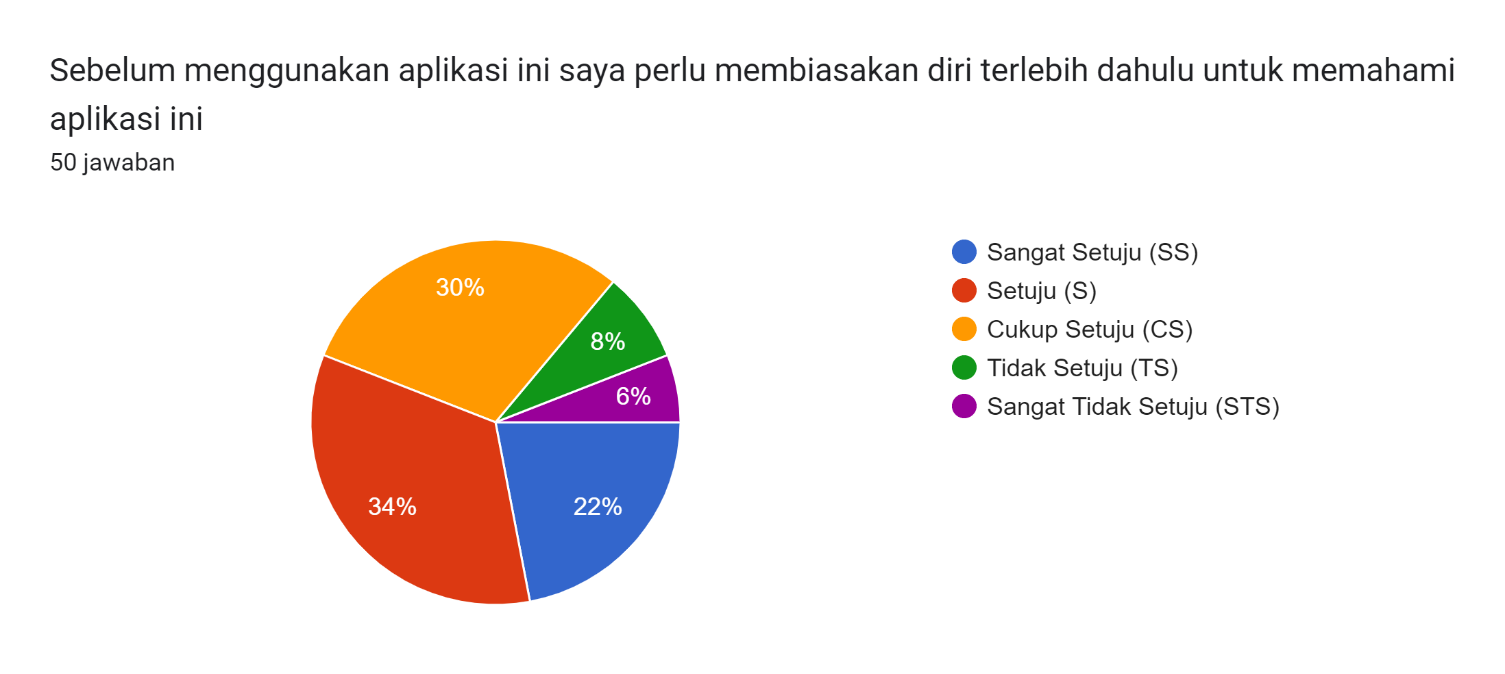
Berdasarkan tanggapan dari 50 responden terhadap pertanyaan yang diberikan, pertanyaan ketiga adalah apakah apakah fitur yang digunakan pada aplikasi web ini berjalan dengan baik ? Ditemukan bahwa 40% dari responden sangat setuju, yang diwakili oleh 20 jawaban. Dan sebanyak 46% responden yang mengisi jawaban setuju dan diwakili oleh 23 jawaban. Disusul dengan 14% dari reponden yang mengisi jawaban cukup setuju dan diwakli oleh 7 jawaban.

Gambar 4. 17 Kuisioner Pernyataan Kesatu

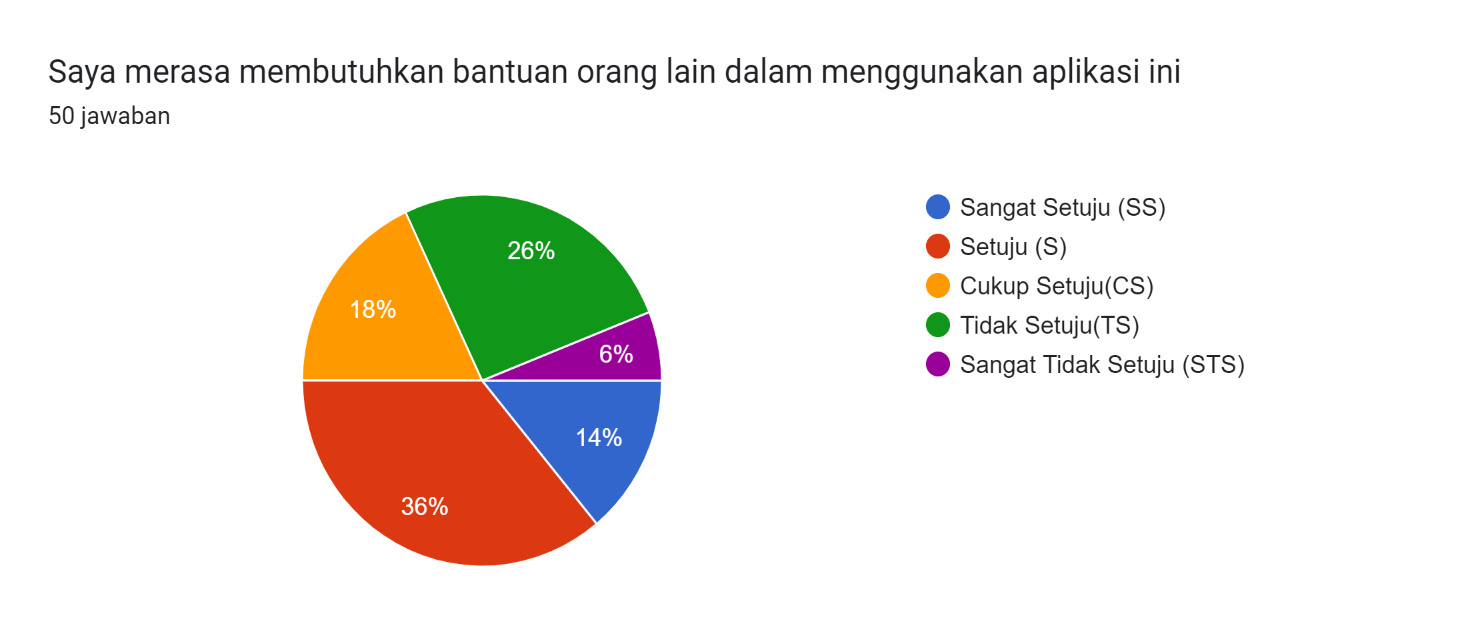
Berdasarkan tanggapan dari 50 responden terhadap pernyataan yang diberikan, pernyataan keempat adalah saya merasa desain antar muka aplikasi web ini mudah dipahami. Ditemukan bahwa 26% dari responden sangat setuju, yang diwakili oleh 13 jawaban. Dan sebanyak 64% responden yang mengisi jawaban setuju dan diwakili oleh 32 jawaban. Disusul dengan 8% dari reponden yang mengisi jawaban cukup setuju dan diwakli oleh 4 jawaban. Kemudian disusul lagi dengan 2% dari reponden yang mengisi jawaban tidak setuju yang diwakili oleh 1 jawaban.

Gambar 4. 18 Kuisioner Pernyataan Kedua

Berdasarkan tanggapan dari 50 responden terhadap pernyataan kedua yang diberikan, pada pernyataan keempat adalah saya merasa merasa aplikasi ini aplikasi web ini sangat memuaskan. Ditemukan bahwa 32% dari responden yang menjawab sangat setuju , yang diwakili oleh 16 jawaban. Dan sebanyak 36% responden yang mengisi jawaban setuju dan diwakili oleh 18 jawaban. Disusul dengan 30% dari reponden yang mengisi jawaban cukup setuju dan diwakli oleh 15 jawaban. Kemudian disusul lagi dengan 2% dari reponden yang mengisi jawaban tidak setuju yang diwakili oleh 1 jawaban.

 Gambar 4. 19 Kuisioner Pernyataan Ketiga

Berdasarkan tanggapan dari 50 responden terhadap pernyataan ketiga yang diberikan, pada pernyataan ketiga adalah sebelum menggunakan aplikasi ini saya perlu membiasakan diri terlebih dahulu untuk memahami aplikasi ini. Ditemukan bahwa 22% dari responden yang menjawab sangat setuju , yang diwakili oleh 11 jawaban. Dan sebanyak 34% responden yang mengisi jawaban setuju yang diwakili oleh 17 jawaban. Disusul dengan 30% dari responden yang mengisi jawaban cukup setuju dan diwakli oleh 15 jawaban. Kemudian disusul dengan 8% dari responden yang mengisi jawaban tidak setuju yang diwakili oleh 4 jawaban. Setelah itu disusul lagi dengan 6% dari responden yang mengisi jawaban sangat tidak setuju yang diwakili oleh 3 jawaban.

Gambar 4. 20 Kuisioner Pernyataan Keempat

Berdasarkan tanggapan dari 50 responden terhadap pernyataan keempat yang diberikan, pada pernyataan keempat adalah saya membutuhkan bantuan orang lain dalam menggunakan aplikasi ini. Ditemukan bahwa 14% dari responden yang menjawab sangat setuju , yang diwakili oleh 7 jawaban. Dan sebanyak 36% responden yang mengisi jawaban setuju yang diwakili oleh 18 jawaban. Disusul dengan 18% dari responden yang mengisi jawaban cukup setuju dan diwakli oleh 9 jawaban. Kemudian disusul dengan 26% dari responden yang mengisi jawaban tidak setuju yang diwakili oleh 13 jawaban. Setelah itu disusul lagi dengan 6% dari responden yang mengisi jawaban sangat tidak setuju yang diwakili oleh 3 jawaban.

Gambar 4. 21 Kuisioner Pernyataan Kelima

Berdasarkan tanggapan dari 50 responden terhadap pernyataan yang diberikan, pada pernyataan kelima adalah saya merasa secara keseluruhan penggunaan aplikasi ini memberikan hasil yang memuaskan. Ditemukan bahwa 40% dari responden sangat setuju, yang diwakili oleh 20 jawaban. Dan sebanyak 44% responden yang mengisi jawaban setuju dan diwakili oleh 22 jawaban. Disusul dengan 16% dari reponden yang mengisi jawaban cukup setuju dan diwakli oleh 8 jawaban.

Berdasarkan hasil kuisioner yang di dapatkan pada gambar diatas maka akan dihitung dengan skala likert dari 1 hingga 8

Tabel 4. 8 Skor kuisioner

|  |  |  |
| --- | --- | --- |
| Skala Jawaban | Keterangan | Skor |
| SS | Sangat Setuju | 5 |
| S | Setuju | 4 |
| CS | Cukup Setuju | 3 |
| TS | Tidak Setuju | 2 |
| Skala Jawaban | Keterangan | Skor |
| STS | Sangat Tidak Setuju | 1 |

Rumus berikut digunakan untuk menghitung nilai persentase uji usability testing :

Keterangan :

Y = Nilai presentase yang dicari (..%)

X = Jumlah nilai kategori jawaban dikalikan dengan frekuensi ( ∑ = N x R)

N = Nilai dari setiap jawaban.

R = Frekuensi

Skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel (5 x 50 = 250)

Setelah mendapatkan nilai Y, maka selanjutnya mencari interval (rentang jarak dan interpretasi persen menggunakan rumus berikut :

Rumus interval

I = 100/Jumlah Skor (Likert)

Maka I = 100/5 = 20

Hasil (I) = 20

Berikut kriteria interpretasi skor berdasarkan interval yaitu:

Angka 0% - 19,99% = Sangat Tidak Setuju

Angka 20% - 39,99% = Tidak Setuju

Angka 40% - 59,99% = Cukup Setuju

Angka 60% - 79,99% = Setuju

Angka 80% - 100% = Sangat Setuju

*Tabel 4. 9 Hasil Kuisioner ke- 1*



Tabel 4. 10 Hasil Kuisioner ke- 2



*Tabel 4. 11 Hasil Kuisioner ke- 3*



Tabel 4. 12 Hasil Kuisioner ke- 4



Tabel 4. 13 Hasil Kuisioner ke- 5



Tabel 4. 14 Hasil Kuisioner ke- 6



Tabel 4. 15 Hasil Kuesioner ke- 7



Tabel 4. 16 Hasil Kuesioner ke- 8



Berdasarkan hasil perhitungan usability testing terhadap 8 kuisioner di atas, dapat disimpulkan bahwa pembuatan aplikasi web pada pengenalan wajah suku moi sudah tepat karena telah mencapai lebih dari 80% menyatakan sangat setuju.

# 

# **BAB V**

# **PENUTUP**

# **Kesimpulan**

Hasil dari implementasi dari implementasi dan pengujian model EfiicientNet Convolutional Neural Network pada jenis wajah suku moi diperoleh kesimpulan sebagai berikut :

1. Sistem tersebut berhasil diimplementasikan pada klasifikasi wajah suku moi menggunakan metode *Convolutional Neural Network* (CNN) berbasis *web* serta menggunakan bahasa *python* sehingga mendapatkan skor *usability testing* lebih dari 80%, yang dimana menunjukkan bahwa penelitian tersebut telah dilakukan sesuai dengan perancangan sistem.
2. Pada proses pelatihan klasifikasi wajah suku moi dengan dua kelas, moi dan non moi menggunakan model *EfficientNet* dapat mencapai tingkat akurasi tertinggi sebesar 94%, *Val\_Accuracy 96%,* dan angka *F1 score* dapat mencapai 96%

# **Saran**

Adapun saran dari penulis dalam pengembangan sistem klasifikasi wajah suku moi mencakup ide-ide sebagai berikut :

1. Dapat meningkatkan tingkat akurasi lebih dari hasil yang didapatkan pada penelitian tersebut.
2. Wajah yang digunakan pada perbedaan warna kulit harus terlihat lebih mencolok agar tingkat membandingkan dalam suatu objek lebih mudah.
3. Pada penelitian selanjutnya tambahkan beberapa fitur agar bisa melakukan klasifikasi secara multitarget atau real detection.
4. Pada penelitian selanjutnya agar menambahkan dataset yang lebih banyak.
5. Pada penelitian selanjutnya agar membuat tampilan web agar lebih menarik.
6. Untuk kedepan nya agar sistem tersebut bisa di kembangkan di *Android.*
7. Pada penelitian selanjutnya agar kedepan nya sistem tersebut bisa mengatasi pengenalan Cahaya.
8. Pada penelitian selanjutnya agar peembagian data seimbang diperlukan sehingga memperoleh hasil yang lebih baik.

## **DAFTAR PUSTAKA**

Alamsyah, Derry, dan Dicky Pratama. 2020. “Implementasi Convolutional Neural Networks (CNN) untuk Klasifikasi Ekspresi Citra Wajah pada FER-2013 Dataset.” *Jurnal Teknologi Informasi* 4(2):350–55. doi: 10.36294/jurti.v4i2.1714.

Aras, Suhardi, Arief Setyanto, dan Rismayani. 2022. “Classification of Papuan Batik Motifs Using Deep Learning and Data Augmentation.” *2022 4th International Conference on Cybernetics and Intelligent System, ICORIS 2022* (Diusulkan untuk mengklasifikasikan motif-motif tersebut menggunakan deep learning menggunakan arsitektur Vgg16 dan Resnet50 dengan fine tuning, untuk menambahkan data dengan berbagai kombinasi guna memperoleh performa yang lebih baik.):1–5. doi: 10.1109/ICORIS56080.2022.10031320.

Arkansa, Sandy Danish, dan Chairisni Lubis. 2023. “KLASIFIKASI RAS ANJING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DENGAN ARSITEKTUR VGG-16.” *Jurnal Ilmu Komputer dan Sistem Informasi* 11(1):1–7. doi: 10.24912/jiksi.v11i1.24078.

Armandhani, Ruri, Randy Cahya Wihandika, dan Muh. Arif Rahman. 2019. “Klasifikasi Gender berbasis Wajah menggunakan Metode Local Binary Pattern dan Random KNN.” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* 3(8):1–8.

Azahro Choirunisa, Nadia, Tita Karlita, dan Rengga Asmara. 2021. “Deteksi Ras Kucing Menggunakan Compound Model Scaling Convolutional Neural Network.” *Technomedia Journal* 6(2):1–16. doi: 10.33050/tmj.v6i2.1704.

Based, Desktop, Bolt Stock, System At, P. T. Ikoma, dan Echo Robotech. 2023. “Jurnal Teknologi Pelita Bangsa.” 14(2):71–76.

Dirgantara, Universitas, Marsekal Suryadarma, dan Universitas Nusa Mandiri. 2019. “Perancangan Sistem Informasi Manajemen Invoice Menggunakan Generator Freamwork Django-Python Berbasis Website Pada Pt. Lampuind Tekno Elektrik.” *Jurnal Sistem Informasi Universitas Suryadarma* 10(1):1–14. doi: 10.35968/jsi.v10i1.999.

Eko, Suharyanto. 2022. “Perancangan Aplikasi Pengenalan Budaya Nusantara Berbasis Android Dengan Metode Rad.” *Jurnal Ilmu Komputer JIK* 5(01):30–39.

Febriyanti, Ni Made Dwi, A. A. Kompiang Oka Sudana, dan I. Nyoman Piarsa. 2021. “Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen.” *Jurnal Ilmiah Teknologi dan Komputer* 2(3):1–10.

Figo, Denni, Cevin Eris Setiawan, dan Adhika Widyassari. 2023. “Aplikasi Pembelajaran Berbasis WEB.” *JIIFKOM (Jurnal Ilmiah Informatika & Komputer) STTR Cepu* 2(1):40–46.

Gelar Guntara, Rangga. 2023. “Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7.” *Jurnal Teknologi Dan Sistem Informasi Bisnis* 5(1):55–60. doi: 10.47233/jteksis.v5i1.750.

Graciela Fausten Novindri, dan Pratyaksa Ocsa Nugraha Saian. 2022. “Implementasi Flask Pada Sistem Penentuan Minimal Order Untuk Tiap Item Barang Di Distribution Center Pada Pt Xyz Berbasis Website.” *Jurnal Mnemonic* 5(2):81–85. doi: 10.36040/mnemonic.v5i2.4670.

Haris Alfikri, Reza, Mardi Siswo Utomo, Herny Februariyanti, Eko Nurwahyudi, Program Studi Sistem Informasi, Universitas Stikubank Semarang, Jalan Tri Lomba Juang, Kota Semarang, dan Jawa Tengah. 2022. “Pembangunan Aplikasi Penerjemah Bahasa Isyarat Menggunakan.” 16(2):183–97.

Hartono, Seno, Anggi Perwitasari, dan Herry Sujaini. 2020. “Komparasi Algoritma Nonparametrik untuk Klasifikasi Citra Wajah Berdasarkan Suku di Indonesia.” *Jurnal Edukasi dan Penelitian Informatika (JEPIN)* 6(3):1–7. doi: 10.26418/jp.v6i3.43268.

Kimmo karkkainen, Jungseok joo. 2019. “untuk Pengukuran dan Mitigasi Bias.” *FairFace: Kumpulan Data Atribut Wajah untuk Ras, Gender, dan Usia yang Seimbang untuk Pengukuran dan Mitigasi Bias* 1548–58.

Makawewe, Navalin R., Reny Syafriny, dan Ricky M. S. Lakat. 2022. “PUSAT BUDAYA SUKU MOI DI SORONG.” 11(2):42–49.

Maryati, Indra. 2021. “Implementasi Framework Tensorflow Object Detection Dalam Mengklasifikasi Jenis Kendaraan Bermotor.” *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)* 8(4):1–8. doi: 10.35957/jatisi.v8i4.1269.

Melangi, Sudirman. 2020. “Klasifikasi Usia Berdasarkan Citra Wajah Menggunakan Algoritma Artificial Neural Network dan Gabor Filter.” *Jambura Journal of Electrical and Electronics Engineering* 2(2):60–67. doi: 10.37905/jjeee.v2i2.6956.

Muhammad Haris Diponegoro, Sri Suning Kusumawardani, dan Indriana Hidayah. 2021. “Tinjauan Pustaka Sistematis: Implementasi Metode Deep Learning pada Prediksi Kinerja Murid.” *Jurnal Nasional Teknik Elektro dan Teknologi Informasi* 10(2):131–38. doi: 10.22146/jnteti.v10i2.1417.

Munarto, Ri, dan Ardian Darma. 2021. “Klasifikasi Gender dan Usia Berdasarkan Citra Wajah Manusia Menggunakan Convolutional Neural Network.” *Setrum : Sistem Kendali-Tenaga-elektronika-telekomunikasi-komputer* 10(2):30–43. doi: 10.36055/setrum.v10i2.12991.

Normawati, Dwi, dan Surya Allit Prayogi. 2021. “Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter.” *Jurnal Sains Komputer & Informatika (J-SAKTI* 5(2):1–15.

Peryanto, Ari, Anton Yudhana, dan Rusydi Umar. 2020. “Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network.” *Format : Jurnal Ilmiah Teknik Informatika* 8(2):1–10. doi: 10.22441/format.2019.v8.i2.007.

Pratama, Yovi, Errissya Rasywir, Fachruddin Fachruddin, Desi Kisbianty, dan Beni Irawan. 2023. “Eksperimen Layer Pooling menggunakan Standar Deviasi untuk Klasifikasi Dataset Citra Wajah dengan Metode CNN.” *Building of Informatics, Technology and Science (BITS)* 5(1):200–210. doi: 10.47065/bits.v5i1.3604.

Putri, Tiara Khumaira, Mahda Laina Arnumukti, Khusnul Khatimah, Egidya Zalsabila, dan Sudianto Sudianto. 2021. “Journal of dinda.” 1(2):82–87.

Rancangan Ui, Manajemen, Ux Sistem Digital Pengajuan Peminjaman Permodalan Biaya Pertanian Menggunakan Perangkat Lunak Figma Sudjiran, Mohamad Saefudin, Safrido Ahmad Perdana, Program Studi Manajemen Informatika, Program Studi Sistem Informasi, dan Stmik Jakarta STI. 2023. “Digital System Ui/Ux Design Management Submission of Agricultural Cost Loans Using Figma Software.” *Issue Period* 7(1):74–85. doi: 10.52362/jisicom.v7i1.1090.

Ratumurun, Samuel, dan Chricela Natalia Joseph. 2023. “Implementasi Model Flowchart Perancangan Sistem Informasi Akuntansi untuk Permintaan Dana/Advance.” *PUBLIC POLICY (Jurnal Aplikasi Kebijakan Publik & Bisnis)* 4(1):1–11.

Rendra Soekarta, Suhardi Aras, Ahmad Nur Aswad. 2023. “JURNAL RESTI untuk Klasifikasi Genre Musik.” (158):1–6.

Rizza Aulia Rahman, Anditya Arifianto, Kurniawan Nur Ramdhani. 2023. “Klasifikasi Ras Anjing Berdasarkan Citra Menggunakan Convolutional Neural Network.” *Jurnal Algoritme* 3(2):1–11. doi: 10.35957/algoritme.v3i2.3389.

Septiani, Noer Azni, dan Faauzan Yusuf Habibie. 2022. “Penggunaan Metode Extreme Programming Pada Perancangan Sistem Informasi Pelayanan Publik.” *Jurnal Sistem Komputer dan Informatika (JSON)* 3(3):1–9. doi: 10.30865/json.v3i3.3931.

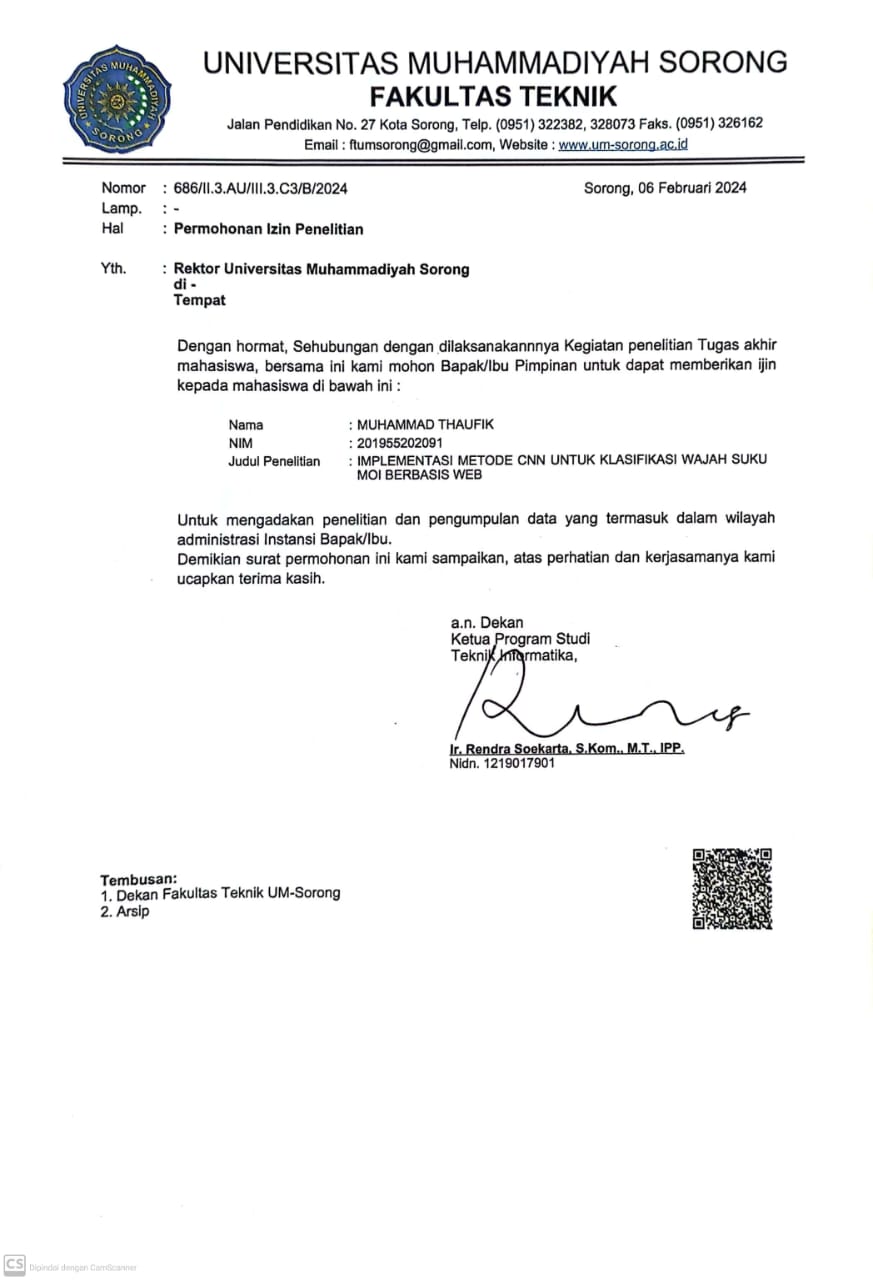
Setiyani, Lila. 2021. “Desain Sistem : Use Case Diagram Pendahuluan.” *Prosiding Seminar Nasional : Inovasi & Adopsi Teknologi 2021* (September):1–15.

Sunarto Usna , Sudjiran, M. Hidayatullah. 2023. “Aplikasi Penjualan Pada Bengkel Bintoro Motor Service Dan Sparepart Berbasis Web.” *Jurnal Ilmiah SIKOMTEK* 13(1):24–29.

Utami, Nengah Widya. 2020. “Pendidikan Ganesha Dengan Metode Usability Testing.” *Sistem Informasi Akuntansi* 9(1):(halaman 112).

## **LAMPIRAN**

1. **Surat Penelitian**

****

1. **Dokumentasi pengambilan dataset**

****

****

****

****

****

1. **Dokumentasi wawancara**

****

1. **Sourcode Aplikasi**

**(Manage)**

#!/usr/bin/env python

"""Django's command-line utility for administrative tasks."""

import os

import sys

def main():

"""Run administrative tasks."""

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'cifar.settings')

try:

from django.core.management import execute\_from\_command\_line

except ImportError as exc:

raise ImportError(

"Couldn't import Django. Are you sure it's installed and "

"available on your PYTHONPATH environment variable? Did you "

"forget to activate a virtual environment?"

) from exc

execute\_from\_command\_line(sys.argv)

if \_\_name\_\_ == '\_\_main\_\_':

main()

**(Views)**

import os

import uuid

import urllib.request

from django.http import HttpResponse

from django.shortcuts import render

from django.core.files.storage import FileSystemStorage

from PIL import Image

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing.image import load\_img, img\_to\_array

import json

import keras

from tensorflow import keras

import numpy as np

import tensorflow as tf

import collections

import matplotlib.pyplot as plt

from efficientnet.tfkeras import EfficientNetB1

from django.templatetags.static import static

from django.core.files.temp import NamedTemporaryFile

from django.core.files import File

BASE\_DIR = os.path.dirname(os.path.abspath(\_\_file\_\_))

def reverse\_dictionary(dictionary):

"""

Mengembalikan dictionary yang dibalik.

Args:

dictionary: Dictionary yang akan dibalik.

Returns:

Dictionary yang dibalik.

"""

# Membuat dictionary baru

reversed\_dictionary = {}

# Menambahkan key dan value ke dictionary baru

for key, value in dictionary.items():

reversed\_dictionary[value] = key

# Mengembalikan dictionary baru

return reversed\_dictionary

# Load the model

model = keras.models.load\_model(os.path.join(BASE\_DIR , 'model\_n1.h5'))

# Load the label mapping

with open(os.path.join(BASE\_DIR , 'label\_mapping.json'), 'r') as fp:

label\_mapping = json.load(fp)

# label\_mapping = reverse\_dictionary(label\_mapping)

print(label\_mapping)

# Menambahkan defaultdict ke label\_mapping

# label\_mapping = collections.defaultdict(lambda: 'Unknown', label\_mapping)

def get\_class\_from\_probability(probabilities):

"""

Menentukan kelas prediksi berdasarkan probabilitas.

Args:

probabilities: List berisi probabilitas untuk setiap kelas.

Returns:

Kelas prediksi.

"""

# Mengambil indeks kelas dengan probabilitas tertinggi.

class\_index = np.argmax(probabilities)

# Mengembalikan kelas prediksi.

return class\_index

def get\_exclude(val):

list\_input = list(label\_mapping.values())

# Nilai yang akan dieksklusi

nilai\_eksklusi = val

# Seleksi list dengan mengecualikan nilai eksklusi

list\_hasil = list\_input.copy()

list\_hasil.remove(nilai\_eksklusi)

return list\_hasil[0]

def load\_image(img\_path, show=False):

img = tf.keras.preprocessing.image.load\_img(img\_path, target\_size=(224, 224))

img\_tensor = tf.keras.preprocessing.image.img\_to\_array(img) # (height, width, channels)

img\_tensor = np.expand\_dims(img\_tensor, axis=0) # (1, height, width, channels), add a dimension because the model expects this shape: (batch\_size, height, width, channels)

img\_tensor /= 255. # imshow expects values in the range [0, 1]

if show:

plt.imshow(img\_tensor[0])

plt.axis('off')

plt.show()

return img\_tensor

# Membuat fungsi prediksi

def predict(image\_path):

# Load the image

image = load\_image(image\_path)

prediction = model.predict(image)

# Get the predicted class and probability

class\_predicted = tf.argmax(prediction, axis=1)

probability\_predicted = round(float(tf.math.reduce\_max(prediction, axis=1)), 2)

# Map the predicted class to the label

class\_predicted = label\_mapping[str(class\_predicted.numpy()[0])]

other\_class = get\_exclude(class\_predicted)

other\_probability = abs(90 - probability\_predicted)

class\_predicted = [class\_predicted, other\_class]

probability\_predicted = [probability\_predicted, other\_probability]

return class\_predicted, probability\_predicted

ALLOWED\_EXT = {'jpg', 'jpeg', 'png', 'jfif'}

# classes = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck']

# model = load\_model(os.path.join(BASE\_DIR , 'model.hdf5'))

def allowed\_file(filename):

return '.' in filename and filename.rsplit('.', 1)[1] in ALLOWED\_EXT

from django.conf import settings

from django.core.files.storage import FileSystemStorage

fs = FileSystemStorage(location=os.path.join(settings.STATICFILES\_DIRS[0], 'images'))

from django.conf import settings

def success(request):

error = ''

if request.method == 'POST':

if 'link' in request.POST:

link = request.POST.get('link')

try:

resource = urllib.request.urlopen(link)

unique\_filename = str(uuid.uuid4())

filename = unique\_filename + ".jpg"

img\_path = os.path.join(settings.STATICFILES\_DIRS[0], 'images', filename)

with NamedTemporaryFile(delete=True) as temp\_file:

temp\_file.write(resource.read())

temp\_file.flush()

fs.save(img\_path, File(temp\_file))

img = filename

print(f"Filename : {img}")

print(f"Img\_path : {img\_path}")

class\_result, prob\_result = predict(img\_path, model)

predictions = {

"class1": class\_result[0],

"class2": class\_result[1],

"prob1": prob\_result[0],

"prob2": prob\_result[1],

}

except Exception as e:

print(str(e))

error = 'This image from this site is not accessible or inappropriate input'

if len(error) == 0:

return render(request, 'success.html', {'img': fs.url(img\_path), 'predictions': predictions})

else:

return render(request, 'index.html', {'error': error})

elif 'file' in request.FILES:

uploaded\_file = request.FILES['file']

if uploaded\_file and allowed\_file(uploaded\_file.name):

filename = fs.save(uploaded\_file.name, uploaded\_file)

img\_path = os.path.join(settings.MEDIA\_ROOT,f'{filename}')

img\_path2 = os.path.join(settings.MEDIA\_URL,f'{filename}')

img = filename

print(f"uploaded\_file : {uploaded\_file}")

print(f"Filename : {img}")

print(f"ImgPath : {img\_path}")

print(f"Img\_path : {img\_path2}")

class\_result, prob\_result = predict(img\_path)

print(f"class\_result : {class\_result} \n prob\_result : {prob\_result}")

try:

predictions = {

"class1": class\_result[0],

"class2": class\_result[1],

"prob1": prob\_result[0],

"prob2": prob\_result[1],

}

except:

predictions = {

"class1": class\_result[0],

"prob1": prob\_result[0],

}

else:

error = "Please upload images of jpg, jpeg and png extension only"

if len(error) == 0:

return render(request, 'success.html', {'img': img\_path2, 'predictions': predictions})

else:

return render(request, 'index.html', {'error': error})

else:

return render(request, 'index.html')

def pred1(request):

return render(request, "index.html", {'mode': 0})

def pred2(request):

return render(request, "index.html", {'mode': 1})

def tentang(request):

return render(request, "tentang.html")

def home(request):

# css\_url = static('css/style.css')

# print(f"css\_url {css\_url}")

return render(request, "begin.html")