**KLASIFIKASI JENIS KELAMIN PADA CITRA WAJAH MENGGUNAKAN METODE *NAIVE BAYES***

**SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

**Oleh:**

**BELLA SITA ANDJANI NIM. 1341180055**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**AGUSTUS 2017**

# **HALAMAN PENGESAHAN**

**KLASIFIKASI JENIS KELAMIN PADA CITRA WAJAH MENGGUNAKAN METODE *NAIVE BAYES***

Disusun oleh:

BELLA SITA ANDJANI NIM. 1341180055

Skripsi ini telah diuji pada tanggal 10 Agustus 2017

Disetujui oleh:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | Penguji I | : | Dr.Eng. Cahya Rahmad, ST., M.Kom  NIP. 19720202 200501 1 002 | ........................... |
|  |  |  |  |  |
| 2. | Penguji II | : | Imam Fahrur Rozi,ST.,MT  NIP. 19840610 200812 1 004 | ........................... |
|  |  |  |  |  |
| 3. | Pembimbing I | : | Dr.Eng.Rosa.Andrie Asmara,S.T.,M T.  NIP. 1980101 0200501 1 001 | ........................... |
|  |  |  |  |  |
| 4. | Pembimbing II | : | Ulla Delfana Rosiani,S.T.,M T.  NIP. 19780327 200312 2 002 | ........................... |

Mengetahui,

|  |  |
| --- | --- |
| Ketua Jurusan  Teknologi Informasi | Ketua Program Studi  Teknik Informatika |
| Rudy Ariyanto, ST., MCs. | Ir. Deddy Kusbianto P., M.MKom. |
| NIP. 19711110 199903 1 002 | |  | | --- | | NIP. 19621128 198811 1 001 | |

# **PERNYATAAN**

Dengan ini saya menyatakan bahwa Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Malang, 10 Agustus 2017

Bella Sita Andjani

# **ABSTRAK**

**Andjani, Bella Sita.** “Klasifikasi Jenis Kelamin Pada Citra Wajah Menggunakan Metode Naive Bayes”. **Pembimbing: (1) Dr. Eng. Rosa Andrie Asmara, ST, M.Kom (2) Ulla Delfana Rosiani, S.T., M.T**

**Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2017.**

Saat ini perkembangan teknologi yang berkaitan dengan pengenalan wajah banyak dimanfaatkan pada aplikasi pengenalan data biologis (*biometrics*) seperti pengenalan jenis kelamin. Penerapan aplikasi yang memerlukan pengenalan jenis kelamin adalah proses segmentasi pasar untuk mengetahui trend demografis dari produk yang dipasarkan berdasarkan jenis kelamin. Selain itu, aplikasi ini juga dapat digunakan untuk pembatasan akses suatu ruangan.

Klasifikasi jenis kelamin pada citra wajah menggunakan metode *Naive Bayes* dapat digunakan untuk membedakan wajah wanita dan wajah pria berdasarkan fitur. Pembuatan data training berupa citra wajah dengan total 61 data dengan rincian 25 perempuan dan 36 laki-laki. Penggunaan fitur yang diperoleh dari deteksi mata, hidung dan mulut diekstraksi dengan metode *Principal Component Analysis* yang selanjutnya akan dilakukan proses klasifikasi jenis kelamin dengan metode *Naive Bayes* yang menghasilkan akurasi kecocokan sebesar 80 %.

**Kata Kunci :** Jenis Kelamin, Deteksi Wajah, *Haar Cascade Classifier, Principal Component Analysis, Naive Bayes*

# **ABSTRACT**

**Andjani, Bella Sita.** "Classification of Gender In Face Image Using Naive Bayes Method". **Advisor: (1) Dr.** **Eng.** **Rosa Andrie Asmara, ST, M.Kom (2) Ulla Delfana Rosiani, ST, MT**

**Thesis, Informatics Engineering Study Program, Department of Information Technology, State Polytechnic of Malang, 2017.**

Nowadays, the development of technology related to face recognition is widely used in introduction of biological data applications (biometrics) such as the gender’s introduction. The application’s which requires the introduction of gender is the process of market segmentation to determine the demographic trend of product marketed by gender. In addition, it also can be used to limit access to a room.

Classification of the sex in the face image using *Naive Bayes* method can be used to distinguish the face of the woman and the man's face based on features. Preparation of training data in the form of facial image with a total of 61 data with details 25 women and 36 men. The use of features derived from the detection of the eyes, nose and mouth in the extraction method of *Principal Component Analysis* which in turn will do the classification process sex with *Naive Bayes method* which yields 80% match accuracy.

**Keywords :** Gender, Face Detection, *Haar Cascade Classifier, Principal Component Analysis, Naive Bayes*

# **KATA PEGANTAR**

Puji Syukur kehadirat Allah AWT atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “KLASIFIKASI JENIS KELAMIN PADA CITRA WAJAH MENGGUNAKAN METODE NAIVE BAYES”. Skripsi ini disusun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Penulis mengucapkan terimakasih kepada Allah SWT karena atas limpahan rahmat-Nya, penulis dapat menyelesaikan skripsi tepat pada waktunya. Untuk itu, penulis menyampaikan rasa terima kasih kepada:

1. Tuhan Yang Maha Esa yang telah memberikan petunjuk dan hidayah-NYA dalam penyusunan skripsi ini sehingga dapat berjalan dengan baik.
2. Kedua Orang Tua dan Keluarga tercinta penulis yang telah banyak memberikan dukungan baik secara moral maupun material sehingga dalam penyusunan skripsi ini dapat berjalan dengan lancar dan sesuai dengan harapan penulis.
3. Bapak Rudy Ariyanto, ST., M.Cs., selaku ketua jurusan Teknologi Informasi
4. Bapak Dr. Eng. Rosa Andrie Asmara, ST., MT. Dan Ibu Ulla Delfana Rosiani,S.T.,M.T selaku dosen pembimbing skripsi yang telah memberikan waktu,kesempatan,ilmu dan bimbingan. Serta dosen-dosen pengajar Program Studi Teknik Informatika,Jurusan Teknologi Informasi, Politeknik Negeri Malang yang telah memberikan bimbingan dan ilmunya.
5. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Skripsi dari awal hingga akhir yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan skripsi dan laporan ini masih jauh dari sempurna.Karenanya,segala kritik dan saran yang membangun sangat penulis harapkan. Terima kasih.

Malang, 10 Agustus 2017

Penulis

# **DAFTAR ISI**

Halaman

[HALAMAN PENGESAHAN ii](#_Toc490549326)

[PERNYATAAN iii](#_Toc490549327)

[ABSTRAK iv](#_Toc490549328)

[ABSTRACT v](#_Toc490549329)

[KATA PEGANTAR vi](#_Toc490549330)

[DAFTAR ISI vii](#_Toc490549331)

[DAFTAR GAMBAR ix](#_Toc490549332)

[DAFTAR TABEL xi](#_Toc490549333)

[DAFTAR LAMPIRAN xii](#_Toc490549334)

[BAB I. PENDAHULUAN 1](#_Toc490549335)

[1.1 Latar Belakang 1](#_Toc490549336)

[1.2 Rumusan Masalah 2](#_Toc490549337)

[1.3 Tujuan 2](#_Toc490549338)

[1.4 Batasan Masalah 3](#_Toc490549339)

[1.5 Sistematika Penulisan 3](#_Toc490549340)

[BAB II. LANDASAN TEORI 5](#_Toc490549341)

[2.1 Biometrik Wajah 5](#_Toc490549342)

[2.2 Pengenalan Wajah 5](#_Toc490549343)

[2.3 Pengolahan Citra 8](#_Toc490549344)

[2.4 Deteksi Wajah (Face Detection) 8](#_Toc490549345)

[2.5 Deteksi Fitur (Feature Selection) 11](#_Toc490549346)

[2.6 Ekstraksi Fitur (Feature Extraction) 11](#_Toc490549347)

[2.7 Klasifikasi Jenis Kelamin 15](#_Toc490549348)

[2.8 EmguCV 20](#_Toc490549349)

[BAB III. METODOLOGI PENELITIAN 21](#_Toc490549350)

[3.1 Tahapan Penelitian 21](#_Toc490549351)

[3.2 Data 22](#_Toc490549352)

[3.3 Metode Pengolahan Data 22](#_Toc490549353)

[BAB IV. ANALISIS DAN PERANCANGAN 25](#_Toc490549354)

[4.1 Analisis Kebutuhan 25](#_Toc490549355)

[4.2 Perancangan Sistem 26](#_Toc490549356)

[4.3 Desain Interface 33](#_Toc490549357)

[4.4 Desain Database 40](#_Toc490549358)

[BAB V. IMPLEMENTASI 41](#_Toc490549359)

[5.1 Implementasi Proses 41](#_Toc490549360)

[5.2 Implementasi Metode 61](#_Toc490549361)

[BAB VI. PENGUJIAN DAN PEMBAHASAN 75](#_Toc490549362)

[6.1 Pengujian Proses 75](#_Toc490549363)

[6.2 Analisa Hasil Penelitian 85](#_Toc490549364)

[BAB VII. KESIMPULAN 86](#_Toc490549365)

[7.1 Kesimpulan 86](#_Toc490549366)

[7.2 Saran 86](#_Toc490549367)

[DAFTAR PUSTAKA 87](#_Toc490549368)

[LAMPIRAN-LAMPIRAN 88](#_Toc490549369)

# **DAFTAR GAMBAR**

Halaman

[Gambar 2. 1 Alur algoritma Haar Cascade Classifier 11](#_Toc490310500)

[Gambar 2. 2 Blok diagram proses ekstraksi fitur 13](#_Toc490310501)

[Gambar 2. 3 Alur proses klasifikasi Naive Bayes 17](#_Toc490310502)

[Gambar 2. 4 Alur metode Naive Bayes 19](#_Toc490310503)

[Gambar 3. 1 Metode Pengembangan Sistem 21](#_Toc490310504)

[Gambar 3. 2 Metode pengolahan data 22](#_Toc490310505)

[Gambar 4. 1 Blok Diagram Sistem 26](#_Toc490310506)

[Gambar 4. 2 Flowchart sistem 27](#_Toc490310507)

[Gambar 4. 3 Flowchart deteksi dengan Haar Cascade Classifier 29](#_Toc490310508)

[Gambar 4. 4 Alur sebuah cascade pendeteksian 30](#_Toc490310509)

[Gambar 4. 5 Alur ekstraksi fitur PCA 31](#_Toc490310510)

[Gambar 4. 6 Alur klasifikasi jenis kelamin 33](#_Toc490310511)

[Gambar 4. 7 Halaman utama sistem 34](#_Toc490310512)

[Gambar 4. 8 Halaman Training ( input citra) 35](#_Toc490310513)

[Gambar 4. 9 Halaman training (pre-processing) 36](#_Toc490310514)

[Gambar 4. 10 Halaman training (cropping fitur) 37](#_Toc490310515)

[Gambar 4. 11 Halaman training (proses PCA) 38](#_Toc490310516)

[Gambar 4. 12 Halaman Testing 39](#_Toc490310517)

[Gambar 5. 1 Tampilan Form Utama 41](#_Toc490310518)

[Gambar 5. 2 Tampilan Form Training 42](#_Toc490310519)

[Gambar 5. 3 Tampilan Form pre-processing 43](#_Toc490310520)

[Gambar 5. 4 Tampilan Form deteksi wajah dan fitur 44](#_Toc490310521)

[Gambar 5. 5 Tampilan Form Cropping fitur 45](#_Toc490310522)

[Gambar 5. 6 Tampilan Form Ekstraksi fitur dengan PCA 46](#_Toc490310523)

[Gambar 5. 7 Proses penyimpanan data ke database 52](#_Toc490310524)

[Gambar 5. 8 Data tesimpan 56](#_Toc490310525)

[Gambar 5. 9 Tampilan data training 57](#_Toc490310526)

[Gambar 5. 10 Tampilan Form Testing 58](#_Toc490310527)

[Gambar 5. 11 Objek wajah terdeteksi 59](#_Toc490310528)

[Gambar 5. 12 Hasil klasifikasi jenis kelamin dengan Naive Bayes 59](#_Toc490310529)

[Gambar 5. 13 Citra uji 61](#_Toc490310530)

[Gambar 5. 14 Citra deteksi fitur 62](#_Toc490310531)

[Gambar 5. 15 Pengambilan fitur 62](#_Toc490310532)

[Gambar 5. 16 Nilai fitur Mata kiri 63](#_Toc490310533)

[Gambar 5. 17 Nilai fitur mata kanan 63](#_Toc490310534)

[Gambar 5. 18 Nilai fitur hidung 63](#_Toc490310535)

[Gambar 5. 19 Nilai fitur mulut 63](#_Toc490310536)

[Gambar 5. 20 Nilai rata-rata fitur wajah 64](#_Toc490310537)

[Gambar 5. 21 Inputan Image 64](#_Toc490310538)

[Gambar 5. 22 Citra Uji 1 70](#_Toc490310539)

[Gambar 5. 23 Hasil klasifikasi citra uji 1 70](#_Toc490310540)

[Gambar 5. 24 Citra uji 2 71](#_Toc490310541)

[Gambar 5. 25 Hasil klasifikasi citra uji 2 71](#_Toc490310542)

[Gambar 5. 26 Perhitungan excel training perempuan 72](#_Toc490310543)

[Gambar 5. 27 Perhitungan excel training laki-laki 73](#_Toc490310544)

[Gambar 5. 28 Normalisasi nilai 73](#_Toc490310545)

[Gambar 5. 29 Perhitungan nilai keempat fitur 74](#_Toc490310546)

[Gambar 5. 30 Nilai probabilitas dan hasil klasifikasi 74](#_Toc490310547)

[Gambar 6. 1 Button pengambilan gambar 75](#_Toc490310689)

[Gambar 6. 2 Pesan error inputan citra 76](#_Toc490310690)

[Gambar 6. 3 Pesan fitur tidak terdeteksi 76](#_Toc490310691)

# **DAFTAR TABEL**

Halaman

[Tabel 4. 1 Tabel Detail data training 40](#_Toc490310696)

[Tabel 5. 1 Nilai Grayscale diubah menjadi matriks T 65](#_Toc490310702)

[Tabel 5. 2 Penggabungan data training 65](#_Toc490310703)

[Tabel 5. 3 Rata-rata 66](#_Toc490310704)

[Tabel 5. 4 Normalisasi Matriks 66](#_Toc490310705)

[Tabel 5. 5 Nilai kovarian 67](#_Toc490310706)

[Tabel 5. 6 Nilai EigenValue 67](#_Toc490310707)

[Tabel 5. 7 Nilai EigenVector 67](#_Toc490310708)

[Tabel 5. 8 Nilai EigenFeature 68](#_Toc490310709)

[Tabel 5. 9 Hasil PCA training 68](#_Toc490310710)

[Tabel 5. 10 Tabel rata-rata fitur training 69](#_Toc490310711)

[Tabel 6. 1 Pengujian deteksi wajah 79](#_Toc490310712)

[Tabel 6. 2 Pengujian tingkat akurasi klasifikasi 82](#_Toc490310713)

[Tabel 6. 3 Pengujian Blackbox 83](#_Toc490310714)

# **DAFTAR LAMPIRAN**

Lampiran 1. Pengujian deteksi wajah berdasarkan data training menggunakan algoritma Haar Cascade Classifier

Lampiran 2. Pengujian klasifikasi berdasarkan data training menggunakan metode Naive Bayes

Lampiran 3. Pengujian deteksi wajah berdasarkan data training menggunakan kamera handphone

Lampiran 4. Pengujian deteksi wajah berdasarkan data training dengan berbagai gaya ekspresi

Lampiran 5. Lembar bimbingan skripsi ( Pembimbing 1 )

Lampiran 6. Lembar bimbingan skripsi ( Pembimbing 2 )

Lampiran 7. Lembar revisi skripsi ( Penguji 1 )

Lampiran 8. Lembar revisi skripsi ( Penguji 2 )

Lampiran 9. Lembar verifikasi skripsi

Lampiran 10. Profil Penulis

# **BAB I. PENDAHULUAN**

## **Latar Belakang**

Saat ini perkembangan teknologi yang berkaitan dengan pengenalan wajah banyak dimanfaatkan pada aplikasi pengenalan data biologis (*biometrics*), pencarian dan pengindeksan database citra dan *video* digital, keamanan ruangan dan lain-lain. Pemanfaatan data biologis dapat memberikan informasi tentang identifikasi dari setiap individu yang menggunakan ciri-ciri fisik yang dapat membedakan dengan orang lain seperti, suara, wajah, sidik jari ataupun mata. Salah satu pembeda identitas individu yang satu dengan yang lainnya adalah wajah. Contoh informasi yang didapatkan dari wajah adalah umur, karakter, ekspresi dan juga jenis kelamin. Diantara beberapa *task* kategorisasi wajah, klasifikasi jenis kelamin merupakan informasi yang paling penting secara biologis [1] .

Klasifikasi merupakan sebuah proses pengelompokam fitur citra pelatihan yang telah disimpan dengan fitur citra uji. Salah satu bagian dari *image recognition* yang telah dikembangkan saat ini adalah pengenalan jenis kelamin (*gender recognition*). Kemiripan antara *gender recognition* dengan *face recognition* terletak pada proses ekstraksi fiturnya. Namun, sedikit berbeda pada proses pengklasifikasiannya. Kesuliltan dalam proses *gender recognition* terutama karena kekompleksan dari kondisi wajah, seperti posisi gambar, pencahayaan dan ekspresi gambar yang berbeda-beda yang memiliki dimensi serta reduksi tinggi sehingga harus melalui proses kompresi atau ekstraksi terlebih dahulu sebelum diolah datanya dengan metode klasifikasi.

Penelitian sebelumnya yang terkait dengan penelitian ini adalah penelitian yang dilakukan oleh Priska Choirina. Pada penelitian tersebut deteksi jenis kelamin berdasarkan citra wajah jarak jauh dengan metode *Haar Cascade Classifier* dapat digunakan untuk membedakan wajah wanita dan wajah pria dengan jarak kurang dari 200 cm. Data yang dilakukan untuk proses training adalah 150 wajah laki-laki dan 150 wajah perempuan, dari jumlah wajah tersebut diambil 100 citra wajah untuk setiap jarak berukuran 100, 150, dan 200 cm. Penggunaan fitur-fitur geometris yang diperoleh dari deteksi mata, hidung dan mulut diproses dengan pengukuran jarak antar fitur-fitur dari wajah yang akan dilakukan proses klasifikasi jenis kelamin. Untuk klasifikasi jenis kelamin dilakukan perbandingan 2 metode klasifikasi yaitu *Euclidean Distance* dan C4.5 namun pada penelitian tersebut terdapat beberapa masalah diantaranya pada tahap klasifikasi jarak antara tiap fitur tidak presisi serta *data set* yang kurang bagus sehingga hasil yang diperoleh kurang optimal dikarenakan kurangnya tingkat akurasi pada tahap klasifikasi [2] .

Algoritma *Naive Bayes* merupakan sebuah algoritma yang sederhana dengan mengandalkan probabilitas atau peluang, *Naive Bayes* mudah diimpelemntasikan sebab *classifie*r ini memiliki keuntungan yaitu hanya membutuhkan sejumlah kecil data pelatihan untuk hasil uji. *Naive Bayes* berpotensi cukup bagus dalam mengolah tingkat akurasi sebagai model klasifikasi jenis kelamin . Berdasarkan uraian di atas, maka perlu dibangun aplikasi “Klasifikasi Jenis Kelamin pada citra wajah menggunakan metode *Naive Bayes*”.

## **Rumusan Masalah**

Dengan memperhatikan uraian latar belakang diatas, maka dapat dirumuskan masalah yang terkait ,yaitu sebagai berikut :

1. Bagaimana mendeteksi wajah dari citra *input* menggunakan metode *Haar Cascade Classifier* ?
2. Bagaimana cara ekstraksi fitur citra menggunakan metode *Principal Component Analysis* ?
3. Bagaimana mengklasifikasikan jenis kelamin menggunakan metode *Naive Bayes* berdasarkan nilai PCA?

## **Tujuan**

Dari rumusan masalah diatas didapatkan beberapa tujua yaitu :

1. Untuk mengetahui hasil *output* dari *Haar Cascade Classifier* dalam mendeteksi wajah .
2. Untuk mengetahui hasil ekstraksi fitur menggunakan metode *Principal Component Analysis.*
3. Untuk menganalisis hasil tingkat akurasi jenis kelamin dengan metode *Naive Bayes.*

## **Batasan Masalah**

Dari permasalahan yang telah disebutkan diatas, perlu suatu batasan masalah agar perluasan masalah dapat dihindari yaitu sebagai berikut :

1. Input data berupa citra wajah yang berupa foto menggunakan kamera
2. Metode yang digunakan untuk untuk klasifikasi jenis kelamin mengguunakan metode *Naive Bayes*
3. Wajah yang digunakan Orang Indonesia
4. Pembuatan data training berupa citra wajah dengan total 61 data dengan rincian 25 Perempuan dan 36 Laki-laki.
5. Pembuatan data training dilakukan dalam kondisi pencahayaan yang cukup.
6. Citra input harus mempunyai 4 fitur yaitu mata kiri, mata kanan, hidung, dan mulut.

## **Sistematika Penulisan**

**BAB I PENDAHULUAN**

Bab ini menerangkan latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, metode penelitian dan, sistematika penulisan.

**BAB II LANDASAN TEORI**

Bab ini menerangkan tentang studi pustaka apa saja yang dibutuhkan dalam penulisan penelitian ini.

**BAB III METODOLOGI PENELITIAN**

Bab ini menerangkan tentang langkah-langkah dari metode, teknik, prosedur dan *tools* apa yang akan digunakan sehingga setiap tahap penelitian

**BAB IV ANALISA DAN PERANCANGAN**

Bab ini menerangkan analisa kebutuhan, dan perancangan sistem.

**BAB V IMPLEMENTASI**

Bab ini menjelaskan tentang implementasi sistem yang akan dipaparkan secara detail sesuai rancangan dan komponen (*tools*) bahasa pemrograman yang dipakai.

**BAB VI PENGUJIAN DAN PEMBAHASAN**

Bab ini menerangkan mengenai pengujian sistem serta analisa hasil penelitian.

**BAB VII KESIMPULAN**

Bab ini berisi kesimpulan dan saran hasil dari penelitian.

# **BAB II. LANDASAN TEORI**

## **Biometrik Wajah**

Dalam dunia teknologi informasi, biometric relevan dengan teknologi yang digunakan untuk menganalisa fisik dan kelakuan manusia dalam autentifikasi. Biometrik secara teoritis dapat lebih efektif untuk mengidentifikasikan pribadi seseorang karena biometric mengukur karakteristik masing-masing pribadi untuk membedakan setiap orang. Ketika digunakan untuk indentifikasi pribadi,teknologi biometrics mengukur dan menganalisa karakteristik tingkah laku dan fisiologis manusia. Mengindentifikasi karakteristik fisiologis seseorang yang didasarkan pada pengukuran langsung bagian *dari body-fingertips, hand geometry, facial geometry* dan *eye retinas* serta *irises* [1] .

## **Pengenalan Wajah**

Wajah merupakan bagian dari tubuh yang berperan penting dalam proses penyampaian ciri, identitas dan emosi seseorang. Kemampuan manusia dalam mengenali wajah sering terjadi secara tidak sadar, manusia mampu mengenali ribuan wajah sepanjang hidupnya dan mengindentifikasi wajah yang sekilas dikenalnya sampai beberapa tahun kemudian. Proses ini berlangsung begitu cepat dan dapat tersimpan cukup lama dalam memori manusia walaupun wajah yang dikenalnya memiliki banyak perubahan visual seperti adanya perubahan kondisi,ekspresi,sudut pandang,penuaan dan penambahan aksesoris seperti kacamata,topi sampai adanya perubahan gaya rambut.Oleh karena itu wajah digunakan sebagai acuan indikasi untuk mengenali seseorang.Pengenalan wajah merupakan salah satu bagian *system biometric* yang banyak diaplikasikan khusunya dalam pengklasifikasian jenis kelamin. Identifikasi jenis kelamin dengan citra wajah adalah beberapa aplikasi dari pengenalan wajah. Efisiensi dan akurasi menjadi faktor utama mengapa pengenalan wajah banyak diaplikasikan khusunya dalam system klasifikasi jenis kelamin.

Penelitian terhadap pengenalan wajah manusia sudah banyak dilakukan dengan kelebihan dan kekurangan tertentu, hal ini disebabkan karena wajah manusia merepresentasikan sesuatu yang kompleks dan mengembangkan model komputasi untuk pengenalan wajah manusia adalah hal yang sulit. Pengenalan wajah dibagi menjadi dua bagian yaitu dikenali dan tidak dikenali. Kesulitan muncul ketika wajah dipresentasikan dalam suatu pola yang berisi informasi unik yang membedakan dengan wajah lain [1] .

Metode pengenalan wajah memiliki dua prosedur,yaitu :

1. *Featured-based* yaitu pengenalan kontur wajah dengan mengenali bentuk mata,hidung, dan mulut. Karakteristik organ tersebut kemudian dinyatakan dalam bentuk vector dan vector tersebut terdiri dari bilangan-bilangan yang mewakili baris dan kolom.
2. *Image-based* yaitu analisis komponen wajah secara keseluruhan. Berdasarkan dari konsep ini, perhitungan model terbaik yang menjelaskan bentuk wajah dengan mengutip informasi paling relevan yang terkandung didalam wajah tersebut

Secara garis besar tahapan pengenalan wajah terdiri dari beberapa garis besar,yaitu:

1. Pengambilan Citra

Pengambilan citra ini dilakukan oleh user dan dapat dilakukan dengan beberapa pilihan media seperti : kamera digital, web camera, ponsel dan lain-lain. Setelah itu menyimpan data tersebut kedalam computer sehingga dapat digunakan untuk proses selanjutnya.

1. Image Processing

Tahap ini dilakukan oleh system,dalam pengenalan wajah biasa disebut pre-processing yang merupakan tahap dimana citra tersebut disiapkan agar dapat di proses lebih lanjut. Proses ini dilakukan untuk memudahkan nilai ekstrasi ciri. Pre-processing yang dilakukan adalah Resize, Grayscale, dan Threshold. Rezise dilakukan untuk mendapatkan inputan yang sama dari semua citra inputan,karena resize merupakan proses dimana ukuran citra inputan diperkecil dari ukuran aslinya. Sedangkan Grayscale merupakan teknik yang digunakan untuk mengubah citra berwarna (RGB) menjadi bentuk *grayscale* / tingkat keabuan (dari hitam ke putih). Dengan pengubahan ini,*matriks* penyusun citra yang sebelumnya 3 matriks akan berubah menjadi 1 *matriks*. Proses pembineran dilakukan dengan membulatkan keatas atau kebawah untuk setiap nilai keabuan dari *piksel* yang berada diatas atau bawah yang disebut dengan *threshold* ,dengan tujuan untuk mengatur jumlah derajat keabuan yang ada pada citra.

1. *Feature Extraction*

Tahapan ini dilakukan oleh system menggunakan metode tertentu,dimana akan dicari informasi yang merupakan ciri dari citra wajah. Ciri tersebut pada umumnya akan selalu tetap untuk setiap manusia dewasa karena sudah tidak mengalami masa pertumbuhan dan selalu berbeda-beda pada setiap orang, sehingga dapat dijadikan acuan untuk mendapatkan identitas atau klasifikasi gender dari suatu wajah. Dibalik kemudahan pengenalan wajah, terdapat beberapa masalah yang mungkin timbul ketika proses pengenalan wajah,yaitu :

1. Perubahan Skala

Citra seseorang dapat direpresentasikan berbeda yang diakibatkan perbedaan jarak saat pengambilan citra dengan media tertentu, semakin dekat jarak maka citra wajah akan semakin jelas dan besar.

1. Perubahan Posisi

Citra seseorang dapat direpresentasikan berbeda yang diakibatkan perubahan posisi seseorang ataupun perubahan sudut pengambilan.

1. Perubahan Cahaya

Citra seseorang dapat direpresentasikan berbeda yang diakibatkan perubahan kondisi yang terjadi pada saat pengambilan citra, seperti: intensitas cahaya pada ruangan, arah sumber cahaya dan karakteristik lensa dari media yang digunakan

1. Perubahan Detail dan Ekspresi

Citra seseorang dapat direpresentasikan berbeda yang diakibatkan perubahan detail secara fisik seperti tumbuhnya janggut,kumis,pemakaian aksesoris,perubahan gaya rambut, dan perubahan ekspresi yang dapat mengakibatkan perubahan visual citra .

## **Pengolahan Citra**

Pengolahan citra merupakan suatu proses yang dilakukan dengan *Input* berupa citra dan *Output* nya pun berupa citra. Proses ini dilakukan pada citra data *training* sebelum *feature extraction* (pengambilan ciri). Terdapat dua hal mendasar dalam memahami proses pembentukan citra,yaitu :

1. Geometri formasi citra yang menentukan lokasi suatu titik dalam pemandangan yang diproyeksikan pada bidang citra.
2. Fisik cahaya yang menentukan *brightness* suatu piksel citra sebagai fungsi pencahayaan.

Oleh karena itu, diperlukan penghubung antara notasi matematika untuk mengembangkan algoritma pengolahan citra dan notasi algoritma yang digunakan untuk pembuatan sebuah aplikasi yang disimpan kedalam sistem penyimpanan.

Tujuan utama dari pengolahan citra ini adalah untuk perbaikan data citra dengan menekan noise yang tidak diinginkan pada citra sehingga dapat diproses dan dianalisa lebih lanjut. Pada proses ini citra dinormalisasikan agar lebih siap untuk diolah pada proses selanjutnya dan kualitas yang dihasilkan tergantung pada proses pengolahan citra. Proses ini memiliki tahapan-tahapan tertentu dengan berbagai metode dan perhitungan sesuai dengan kebutuhan [1] .

## **Deteksi Wajah (*Face Detection*)**

Pendeteksian wajah pada citra merupakan suatu langkah penting menuju interaksi manusia dan computer berbasis visi yang cerdas. Pendeteksian wajah ini merupakan langkah awal dalam berbagai penelitian dibidang pengolahan wajah,termasuk pengenalan wajah,perkiraan pose, dan pengenalan ekspresi. Berbagai teknik telah disulkan untuk mendeteksi wajah didalah sebuah citra mauun video.[1]

Deteksi wajah adalah langkah awal untuk melakukan identifikasi wajah atau face recognition. Sebuah pendeteksi wajah mampu mengidentifikasi dan menemukan lokasi atau luas semua wajah yang ada didalam sebuah gambar tanpa memperhatikan pose,skala,orientasi,umur, dan ekspresi . Pendeteksian wajah dapat dilakukan dengan memanfaatkan library yang bersifat open source yaitu *emguCV*. *EmguCV* adalah bagian dari library *openCV* yang menggunakan bahasa pemrograman C#. Proses deteksi adanya citra wajah dalam sebuah gambar pad *OpenCV*, menggunakan sebuah metode yang dipublikasikan oleh Paul Viola Michael Jones tahun 2001. Umumnya disebut metode *Haar Cascade Classifier*.

Metode pendekatan ini adalah metode berbasis fitur (*feature-based methods*), *template matching*, metode berbasis tampilan (*appearance-based methods*). *Feature-based methods* bertujuan untuk mendeteksi keberadaan dan lokasi fitur seperti mata, hidung,dan mulut. Metode *template matching* adalah sebuah metode dengan menggunakan template yang disimpan untuk mendeskripsikan sebuah wajah secara utuh atau fitur wajah secara terpisah. Dalam metode template matching ini, keberadaan sebuah wajah dapat diketahui melalui perhitungan nilai korelasi antara citra masukan dan template yang disimpan. Metode *appearance-based* adalah metode yang menggunakan “*template*” yang dihasilkan dari *training* pada dataset citra dan digunakan untuk mendapatkan variabel yang mewakili keberadaan dari wajah itu sendiri. Dibandingan *feature-based* dan *template matching*, metode *appearance-based* memberikan hasil yang lebih baik ketika diterapkan pada sistem dengan sejumlah besar sampel [2] .

* + 1. **Metode *Haar Cascade Classifier***

Proses deteksi adanya citra wajah dalam sebuah gambar pada menggunakan sebuah metode yang dipublikasikan oleh Paul Viola dan Michael Jones tahun 2001. Umumnya disebut metode *Haar Classifier*. Metode ini merupakan metode yang menggunakan *statistical model* (*classifier*). Algoritma *Haar Cascade Classifier* digunakan dengan *libriray EmguCV* pada deteksi wajah dan deteksi fitur.

Pendekatan untuk mendeteksi objek dalam gambar menggabungkan empat konsep utama :

1. *Training* data pada *Haar*

Metode ini memerlukan 2 tipe gambar objek dalam proses *training* yang dilakukan, yaitu :

* *Positive Samples*
* *Negative Samples*

1. Sistem kerja Algoritma *Haar Cascade Classifier*

Algoritma *Haar* menggunakan metode statistikal dalam melakukan pendeteksian wajah. Metode ini menggunakan simple *haar-like features* dan juga *a cascade of boosted tree classifier*. *Classifier* ini menggunakan gambar berukuran tetap. Haar juga memiliki kemampuan untuk melakukan *scalling* sehingga dapat mendeteksi adanya wajah yang berukuran lebih besar ataupun lebih kecil dari gambar pada *classifier.*

1. *Haar Feature*

*Haar Feature* adalah fitur yang digunakan oleh Viola dan Jones didasarkan pada *Wavalet Haar.* *Wavalet Haar* adalah gelombang tunggal bujur sangkar ( satu interval tinggi dan satu interval rendah ). Untuk dua dimensi, satu terang dan satu gelap. Nilai dari *Haar-like feature* adalah perbedaan antara jumlah nilai-nilai piksel gray level. Kotak *Haar-like feature* dapat dihitung secara cepat menggunakan “*integral image*” .

1. *Integral Image*

*Integral Image* digunakan untuk menentukan ada atau tidaknya dari ratusan fitur *Haar* pada sebuah gambar dan pada skala yang berbeda secara efisien. Pada umumnya, pengintegrasian tersebut berarti menambahkan unit-unit kecil secara bersamaan. Dalam hal ini unit-unit kecil tersebut adalah nilai-nilai piksel. Nilai *integral* untuk masing-masing piksel adalah jumlah dari semua piksel-piksel dari atas sampai bawah. Dimulai dari kiri atas sampai kanan bawah, keseluruhan gambar itu dapat dijumlahkan dengan beberapa operasi bilangan bulat per piksel.

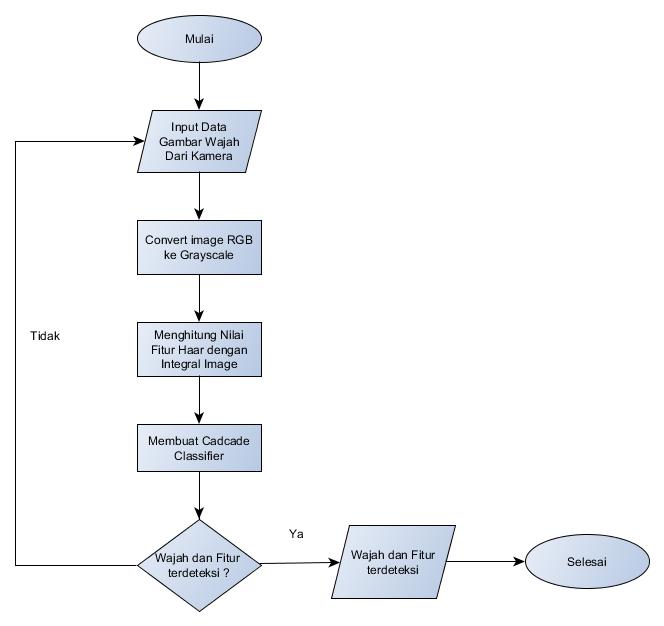
1. *Cascade Classifier*

*Cascade Classifier* adalah sebuah rantai *stage classifier*, dimana setiap *stage classifier* digunakan untuk mendeteksi apakah didalam *image sub window* terdapat objek yang diinginkan (*objek of interest*).

*Stageclassifier* dibangun menggunakan algoritma *adaptive-boost* (*AdaBoost*). Algoritma tersebut mengkombinasi *performance* banyak *weak classifier* untuk menghasilkan *strong classifier*. *Weak classifier* dalam hal ini adalah nilai dari *haar-like feature*. Jenis *AdaBoost* yang digunakan adalah *Gentle AdaBoost* [2] .

## **Deteksi Fitur (*Feature Selection*)**

Deteksi Fitur disini merupakan sebuah proses untuk mendeteksi beberapa object yang ada pada citra wajah. Pada proses ini dilakukan pemilihan fitur wajah, proses yang dilakukan sama seperti pendeteksian wajah sebelumnya (5.4) dengan menggunakan algoritma *Haar Cascade Classifier* yang terdapat pada *library* EmguCV. Fitur-fitur wajah yang dibutuhkan yaitu mata, hidung, dan mulut [1] . Berikut adalah gambaran alur algoritma Haar Cascade Classifier untuk deteksi wajah dan fitur :



Gambar 2. 1 Alur algoritma Haar Cascade Classifier

## **Ekstraksi Fitur (*Feature Extraction*)**

**Ekstraksi fitur citra** merupakan tahapan mengekstrak ciri / informasi dari objek di dalam citra yang ingin dikenali/dibedakan dengan objek lainnya. Fitur yang telah diekstrak yaitu mata,hidung,dan mulut yang kemudian digunakan sebagai parameter / nilai masukan untuk membedakan antara objek satu dengan lainnya pada tahapan identifikasi atau klasifikasi.

Eigenface merupakan salah satu metode pengenalan wajah berdasarkan *Principal Component Analysis* (PCA) yang mudah diimplementasikan. Eigenface dimulai dengan pemrosesan awal untuk mendapatkan hasil citra yang lebih baik. Setelah itu menghitung *eigenvector* dan *eigenvalue* dari citra wajah untuk dilakukan proses training image. Proses training wajah yaitu mencari *eigenvector, eigenvalue* dan *average image* yang diproyeksikan ke dalam subruang PCA [2] .

* + 1. **Algoritma PCA (*Principal Component Analysis*)**

Di tahun 1933, Hotelling mengajukan sebuah teknik untuk mengurangi dimensi sebuah ruang yang direpresentasikan oleh variabel statistik x1, x2, . . . ., xn, dimana variabel tersebut biasanya saling berkorelasi satu dengan yang lain. Selanjutnya Hotelling menyebut metode tersebut sebagai Principal Component Analysis (PCA) [2] .

Transformasi Karhunen-Loeve banyak digunakan untuk memproyeksikan atau mengubah suatu kumpulan data berukuran besar menjadi bentuk representasi data lain dengan ukuran yang lebih kecil. Transformasi Karhunen-Loeve terhadap sebuah ruang data yang besar akan menghasilkan sejumlah vektor basis ortonormal ke dalam bentuk kumpulan *vektor eigen* dari suatu matriks kovarian tertentu, yang dapat secara optimal merepresentasikan distribusi data.

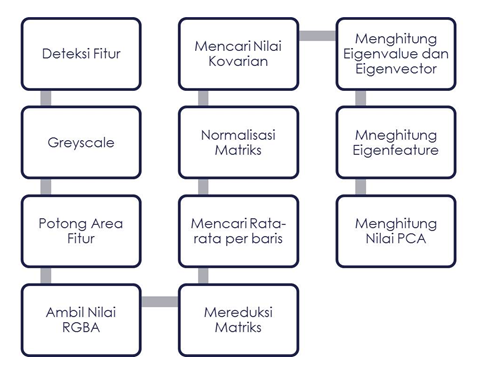
Pengenalan wajah dengan pendekatan Eigenface dapat digolongkan dalam suatu metoda appearance-based, ini dikarenakan pengenalan wajah eigenface menggunakan informasi mentah dari pixel citra yang kemudian direpresentasikan dalam metode tertentu yang selanjutnya digunakan untuk pelatihan dan klasifikasi indentitas citra. Ide dari metode ini adalah memproyeksikan sebuah wajah yang merupakan sebuah citra dapat dilihat sebagai sebuah vektor.

Metode PCA yang juga dikenal dengan nama Karhunen-Loeve transformation (KLT), sudah dikenal sejak 30 tahun lalu dalam dunia pengenalan pola. PCA memberikan transformasi ortogonal yang disebut dengan ‘eigenimage‘ yang mana sebuah image direpresentasikan kedalam bentuk proyeksi linier searah dengan eigenimage yang bersesuaian dengan nilai eigen terbesar dari matrix covariance. Secara praktis matrix covariance ini dibangun dari sekumpulan image training yang diambil dari berbagai obyek/kelas [2] .

Pengenalan citra wajah dengan metode Eigenface dilakukan berdasarkan pada pengurangan dimensi ruang wajah dengan menggunakan Principal Component Analysis (PCA) untuk memperoleh ciri wajah. Tujuan utama penggunaan metode PCA pada pengenalan wajah dengan menggunakan eigenfaces adalah membentuk (ruang wajah) dengan cara mencari vektor eigen yang berkoresponden dengan nilai eigen terbesar dari citra wajah. Vektor eigen ini menyatakan posisi dari Principal Component dalam suatu ruang dimensi n. Sebelum mengaplikasikan metode PCA dalam hal ini mereduksi dimensi dari vektor fitur yang ada maka nilai eigen dari citra wajah yang ada perlu diurutkan terlebih dahulu sehingga nantinya didapat nilai eigen yang tereduksi yang diinginkan.

*Principal Component Analysis* (PCA) merupakan salah satu hasil berharga dari aljabar linier terapan. Prosedur PCA pada dasarnya adalah bertujuan untuk menyederhanakan variable yang diamati dengan cara menyusutkan atau mereduksi dimensi dari fitur wajah yang sudah didapatkan dari deteksi fitur. Kemampuan ekstrasi fitur ini dapat digunakan untuk klasifikasi jenis kelamin. Ekstraksi fitur disini dibuat dengan bahasa C# untukditerapkan padaMetode *Principal Component Analysis* (PCA) [7] .

Berikut adalah alur blok diagram proses ekstraksi fitur dengan PCA :



Gambar 2. 2 Blok diagram proses ekstraksi fitur

Berikut langkah-langkah untuk mendapatkan nilai *Principal Component Analysis* untuk data latih:

1. Mereduksi Dimensi

Citra direduksi dari 2 dimensi menjadi 1 dimensi kemudian digabungkan dengan matriks reduksi citra yang lain (Matriks T) dengan tujuan untuk memudahkan dan mempercepat dalam proses perhitungan nilai rata-rata, jadi jumlah kolom merupakan jumlah citra yang akan diproses ,dengan cara melakukan partisi dimensi himpunan *x* menjadi *l* himpunan bagian,sehingga masing-masing *xi* menjadi berukuran *d* dimensi dengan persamaan berikut :

***d* =** .....................................................................................(2.1)

1. Menghitung Nilai Mean

Dari Matriks T dihitung nilai rata-rata per baris dengan persamaan :

***m = (jumlah tiap baris) / banyak data latih*** ..........................(2.2)

1. Normalisasi Matriks

Nilai rata-rata akan digunakan untuk menormalkan matriks T dengan cara mengurangkan setiap piksel pada matriks T dengan nilai rata-rata menggunakan persamaan berikut :

***A = T – m*** ..............................................................................(2.3)

1. Hasil selisih tiap piksel dengan nilai rata-rata digunakan untuk mencari nilai matriks kovarian (L).

***L = A’ x A*** .............................................................................(2.4)

Keterangan : A’ merupakan matriks *transpose* dari matriks A

1. Menghitung *Eigenvalue* dan *Eigenvector*.

*Eigen Value* adalah nilai karakteristik dari suatu matriks bujungsangkar. *Eigenvector* diambil berdasarkan *Eigen Value* yang lebih besar dari 0 (Eigen Value > 0). *Eigen Value* merupakan nilai ciri yang terdapat pada citra data latih. Jika pada suatu kolom terdapat nilai yang sama maka selanjutnya mengeleminasi *eigen vector* yang kolom *eigen value*-nya memiliki nilai dibawah ambang batas. Hal ini dimaksudkan untuk mengolah ciri yang perlu untuk di proses.

1. Menghitung Nilai *Eigenfeature*

Nilai *Eigenfeature* merupakan nilai ciri citra data latih.

***E = A x V*** ............................................................................(2.5)

1. Menghitung Nilai PCA

Setelah *eigenvector* didapatkan, selanjutnya menghitung nilai PCA menggunakan persamaan berikut :

***PCA\_train = E’ x A*** ..............................................................(2.6)

Jika nilai PCA data latih telah diperoleh maka selanjutnya menghitung nilai PCA data tes :

1. Mengambil nilai RGBA Citra Tes pada mode keabuan dan mereduksinya menjadi matriks satu kolon (Ts).
2. Menghitung nilai *Difference* dengan persamaan :

***Difference= Ts – m*** ...............................................................(2.7)

1. Menghitung nilai PCA data tes :

***PCA\_test = E’ x Difference*** ...................................................(2.8)

*Principal Component Analysis* (PCA) digunakan untuk mereduksi dimensi sekumpulan atau ruang gambar sehingga basis atau sistem koordinat yang baru dapat menggambarkan model yang khas dari kumpulan tersebut dengan lebih baik. Tujuan dari ekstraksi ciri PCA adalah memproyeksikan data latih dari suatu ruang dimensi tertentu ke ruang dimensi yang lebih rendah tanpa kehilangan banyak informasi yang terkandung. Metode PCA akan menghasilkan keluaran utama yaitu nilai reduksi PCA atau matriks penciri dari data latih .Inputan PCA berupa potongan fitur wajah seperti mata,hidung,mulut yang didalamnya berupa matrik berisi nilai piksel lalu diekstraksi sehingga menghasilkan keluaran nilai PCA berupa bilangan real [2] .

## **Klasifikasi Jenis Kelamin**

Klasifikasi merupakan salah satu task analisis data dimana dibentuk sebuah model (classifier) untuk menyatakan suatu objek ke salah satu kategori atau kelas yang sudah didefinisikan sebelumnya. Klasifikasi data merupakan proses yang terdiri atas dua langkah-langkah pertama adalah tahap pelatihan ( learning), dimana sebuah algoritma klasifikasi akan membentuk classifier dengan menganalisis atau belajar dari sekumpulan data latih. Pada langkah kedua, model yang terbentuk digunakan untuk mengklasifikasikan data uji. Akurasi dari sebuah classifier untuk sekumpulan data uji yang diberikan merupakan persentase dari data-data uji yang diklasifikasikan dengan benar oleh classifier.

Dari hasil ekstrasi fitur yang nilainya disimpan ke dalam database maka dapat dilakukan klasifikasi berdasarkan nilai-nilai PCA tersebut yang nantinya akan digunakan sebagai acuan untuk data *testing.* Maka disini dipilih metode Naive Bayes sebagai acuan untuk klasifikasi jenis kelamin.

Bayes merupakan teknik prediksi berbasis probabilitik sederhana yang mendasar pada penerapan teorema Bayes dengan asumsi independensi yang kuat. Dengan kata lain dalam Naïve Bayes, model yang digunakan adalah “model fitur independen”.[Naive Bayes *Classifier*](https://en.wikipedia.org/wiki/Naive_Bayes_classifier) merupakan salah satu metode *machine learning* yang memanfaatkan perhitungan probabilitas dan statistik. Metode ini dikemukakan oleh ilmuwan Inggris yaitu [Thomas Bayes](https://en.wikipedia.org/wiki/Thomas_Bayes) untuk memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya [3] .

Algoritma *Naive Bayes Classifier* merupakan algoritma yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang paling tepat. Dalam penelitian ini yang menjadi data uji adalah data baru hasil ekstrasi fitur. Ada dua tahap pada klasifikasi jenis kelamin. Tahap pertama adalah pelatihan terhadap fitur yang diketahui kategorinya. Sedangkan tahap kedua adalah proses klasifikasi jenis kelamin yang belum diketahui kategorinya [6] .

Dalam algoritma *Naive Bayes Classifier* setiap data direpresentasikan dengan pasangan atribut ”x1, x2, x3, ...xn” dimana x1 adalah data pertama, x2 adalah data kedua dan seterusnya. Sedangkan V adalah himpunan kategori. Pada saat klasifikasi algoritma akan mencari probabilitas tertinggi dari semua kategori fitur yang diujikan (Vmap), dimana persamaanya adalah sebagai berikut:

|  |  |
| --- | --- |
| ...........................(2.9) |  |

Untuk P(x1,x2,x3,....xn) nilainya konstan untuk semuakategori (Vj) sehingga persamaan dapat ditulis sebagai persamaan dapat ditulus sebagai berikut:

....................(2.10)

Persamaan diatas dapat disederhanakan menjadi sebagai berikut:

|  |  |
| --- | --- |
| .........................(2.11) |  |

Keterangan:  
Vj = Kategori fitur

P(xi|Vj) = Probabilitas xi pada kategori Vj

P(Vj) = Probabilitas dari Vj

Cara kerja Naive Bayes Classifier melalui dua tahapan, yaitu :

* Learning  (Pembelajaran)

Naive Bayes adalah suatu metode yang termasuk ke dalam *supervised learning*, maka akan dibutuhkan pengetahuan awal untuk dapat mengambil keputusan. Langkah-langkah :

Langkah 1 : Hitung setiap data training

Langkah 2 : Hitung probabilitas pada setiap kategori P(vj).

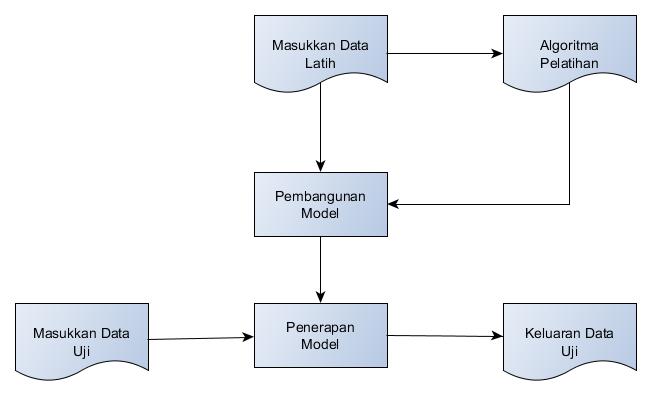
Langkah 3 : Tentukan frekuensi setiap kata wk pada setiap kategori P (wk|vj)

* Classify (Pengklasifikasian). Langkah-langkahnya adalah :

Langkah 1 : Hitung P(vj) Π P (wk | vj) untuk setiap kategori.

Langkah 2 :Tentukan kategori dengan nilai P(vj) Π P(wk | vj) maksimal.

Berikut adalah Alur proses klasifikasi Naive Bayes :



Gambar 2. 3 Alur proses klasifikasi Naive Bayes

* + 1. **Algoritma Naive Bayes**

Algoritma Naive Bayes merupakan salah satu algoritma yang terdapat pada teknik klasifikasi. Naive Bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman dimasa sebelumnya sehingga dikenal sebagai Teorema Bayes. Teorema tersebut dikombinasikan dengan Naïve dimana diasumsikan kondisi antar atribut saling bebas. Klasifikasi Naive Bayes diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya [6] .

Persamaan dari teorema Bayes adalah :

......................................................................(2.12)

Keterangan :

*X* : Data dengan class yang belum diketahui

*H* : Hipotesis data *X* merupakan suatu class spesifik / kategori

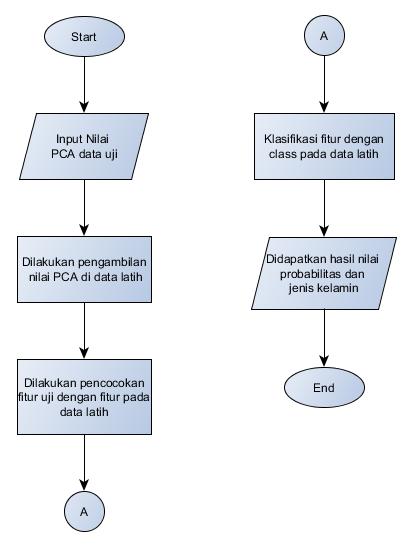
*P(H|X)* : Probabilitas hipotesis *H* berdasarkan kondisi *X*

*P(H)* : Probabilitas hipotesis *H*

*P(X|H)* : Probabilitas *X*  berdasarkan kondisi pada hipotesis *H*

*P(X)* : Probabilitas *X*

Alur dari Alur metode *Naive Bayes* ialah sebagai berikut :



Gambar 2. 4 Alur metode Naive Bayes

Adapun alur dari metode Naive Bayes adalah sebagai berikut :

1. Input nilai ekstrasi PCA pada data uji
2. Dilakukan pengambilan nilai PCA yaitu dengan matriks 10 x 10 yang ada pada data latih
3. Dilakukan pencocokan fitur uji dengan fitur pada data latih
4. Proses klasifikasi fitur dengan class yaitu perempuan atau laki-laki pada data latih
5. Maka didapatkan sebuah hasil yaitu berupa nilai probabilitas dengan jenis kelamin

Klasifikasi dengan *Naive Bayes* bekerja berdasarkan teori probabilitas yang memandang semua fitur dari data sebagai bukti dalam probabilitas. Hal ini memberikan karakteristik *Naive Bayes* sebagai berikut:

1. Mampu menangani secara kuantitatif dan data diskrit
2. Kokoh untuk titik noise yang diisolasi
3. Hanya memerlukan sejumlah kecil data pelatihan untuk mengestimasi parameter (rata-rata dan variasi dari variable) yang dibutuhkan untuk klasifikasi.
4. Mampu mengangani nilai yang hilang dengan mengabaikan instansi selama perhitungan estimasi peluang.
5. Cepat dan efiasiensi ruang.
6. Kokoh terhadap atribut yang tidak relevan.

## **EmguCV**

EmguCV adalah *library* *OpenCV* yang ditunjukkan untuk bahasa pemrograman C# yang digunakan pada deteksi wajah dan deteksi fitur. *OpenCV* dirilis dibawah lisensi BSD *(Berkeley Software Distribution)* sehingga bersifat gratis baik untuk penggunaan akademis maupun 19 komersial. *OpenCV* disusun dengan bahasa C++,C,python,Java dan dapat berjalan di Windows,Linux,Mac OS,iOS dan android. *OpenCV* dirancang untuk efisiensi komputasi dandengan focus yang kuat pada apliakasi. *OpenCV* digunakan hinggan saat ini dan memiliki lebih dari 47 ribu orang dari komunitas pengguna dan diperkirakan jumlah *download* melebihi 9 juta. *OpenCV* telah lama dipergunakan para peneliti dan pengembangan dalam berbagai bidang *computer vision*, seperti inspeksi produk, *medical imaging, camera calibration, security, user interface, streo vision* dan robotika [4] .

# **BAB III. METODOLOGI PENELITIAN**

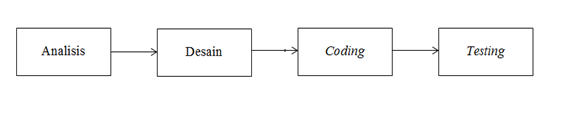
## **Tahapan Penelitian**

Metodologi yang digunakan dalam membangun sebuah aplikasi klasifikasi jenis kelamin pada citra wajah menggunakan metode *Naive Bayes*,

Berikut langkah – langkah yang diperlukan antara lain :

1. Metode Pengembangan Sistem

Dalam tahap pengembangan sistem menggunakan metode pendekatan terstruktur dengan menggunakan model sekuensial linier. Dimana tahapan-tahapan model ini adalah analisis, desain, *coding,* dan *testing.*



Gambar 3. 1 Metode Pengembangan Sistem

1. Analisis

Tahap analisis adalah tahap pengumpulan informasi yang berkaitan dengan pembangunan sistem pengenalan wajah, baik kebutuhan data,perangkat keras, maupun perangkat lunak. Pada tahap analisis dilakukan pula cara-cara perhitungan metode *Eigenface* untuk pengenalan wajah.

1. Desain

Setelah tahap analisis selesai dilakukan, selanjutnya adalah tahap desain. Tahap ini merancang model dan alur pembangunan sistem penegnalan wajah sesuai dengan analisis yang telah dilakukan. Proses perancangan desain sangat mempengaruhi proses pembuatan system.

1. *Coding*

Setelah desain dan alur dirancang, selanjutnya adalah proses implementasi *coding*. Tahap ini merupakan tahap utama karena pada tahap inilah proses pembangunan system yang dapat menyelesaikan masalah dan mengolah data-data yang terkumpul. Pada tahap ini pula proses hasil desain yang telah dibuat diimplementasikan dalam bentuk kode yang dapat menghasilkan sebuah aplikasi.

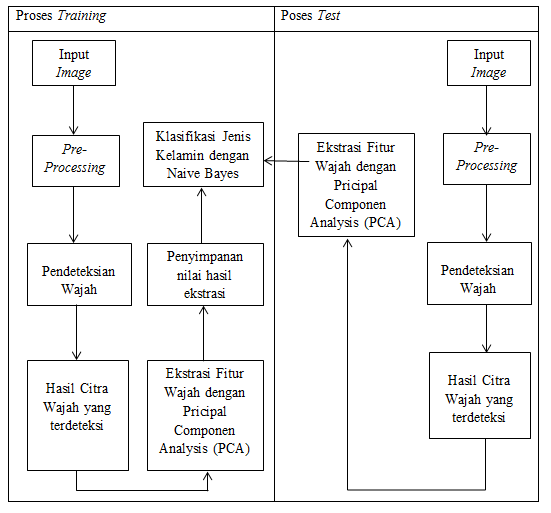
1. *Testing*

Setelah perangkat lunak berhasil dibuat selanjutnya adalah pengujian terhadap system pengenalan wajah menggunakan metode *Eigenface* . Pengujian dilakukan dengan menguji tingkat akurasi pengenalan wajah pada system yang dibangun.

## **Data**

Komponen yang diperlukan untuk melakukan penelitian ini salah satunya adalah data. Data yang diperlukan sebagai objek pengujian aplikasi ini yaitu data citra wajah laki-laki dan citra wajah perempuan untuk dilakukan *training* data dan memiliki format JPG. Data yang digunakan pada proses training dan testing diperoleh dari penelitian sebelumnya [1].

## **Metode Pengolahan Data**



Gambar 3. 2 Metode pengolahan data

Pada Gambar 3.2 merupakan proses pada aplikasi klasifikasi jenis kelamin pada citra wajah berdasarkan ekstraksi ciri menggunakan *Naive Bayes* yang terbagi menjadi dua bagian utama yaitu proses *training* dan proses *test*.

Pada proses *training* terdapat beberapa tahap yang dilakukan yaitu Pre-Processing, pendeteksian wajah menggunakan *Haar Cascade Classifier* yang ada pada *library EmguCv*, ekstraksi fitur wajah dengan *Principal Component Analysis* (PCA), penyimpanan hasil dari ekstrasi fitur ke dalam penyimpanan dan selanjutnya klasifikasi hasil jenis kelamin dengan Metode *Naive Bayes*. Proses *test* sama halnya dengan proses data *training* perbedaannya adalah nilai yang digunakan untuk pengklasifikasian jenis kelamin berdasarkan nilai yang sudah didapatkan pada data *training*.

Proses deteksi wajah adalah poses utama dimana diperlukan baik dalam proses pendeteksian fitur untuk melakukan ekstraksi nilai-nilai dari fitur wajah. Berdasarkan hasil pengujian pada lampiran 1 didapatkan hasil akurasi 100 % pada proses *face detection* menggunakan algoritma *Haar Cascade Classifier*.

Pada *Pre-processing* diatas dilakukan untuk memudahkan nilai ekstrasi ciri. *Pre-processing* yang dilakukan adalah *Resize* dan *Grayscale*. Pada tahap *resize* citra terjadi proses normalisasi dimensi citra wajah,yaitu proses pembesaran atau pengecilan dimensi citra wajah menjadi dimensi yang telah ditentukan yakni 480×720,tujuannya yaitu untuk menyamakan dimensi *image* dari tiap citra yang di masukkan. Sehingga pada proses ekstraksi fitur nanti tidak ada perbedaan dimensi dari matriks data citra wajah. Sedangkan *Grayscale* citra merupakan tahapan pertama dari proses penyelarasan, pada tahap ini terjadi pengkonversian citra warna RGB menjadi citra berwarna keabuan. Citra warna RGB terdiri dari 3 parameter warna yaitu merah (*red*), hijau (*green*), biru (*blue*) yang kemungkin akan sulit untuk dilakukan,oleh karena itu diperlukan penyamaan parameter yaitu dengan melakukan tahap *grayscalling* . Perhitungan digunakan untuk mengubah citra berwarna yang mempunyai nilai matriks masing-masing RGB menjadi citra *grayscale* dengan cara mengambil rata-rata dari nilai RGB.

*Principal Component Analysis* relatif mudah menangani sejumlah data yang cukup besar serta kemampuannya menangani data-data untuk dimensi komplek seperti fitur pada citra wajah yang diekstrak berupa dimensi *eigenvektor* dan *eigenvalues* sehingga diperoleh suatu hasil yaitu ruang wajah *(eigenspace)* atau *eigenfaces.*

Klasifikasi jenis kelamin berdasarkan citra wajah dilakukan untuk mengetahui apakah citra inputan wajah berjenis kelamin laki-laki atau perempuan. Dalam proses ini akan diimplementasikan menggunakan metode *Naive Bayes.* Maka dari hasil tabel pada lampiran 2 dapat dilihat bahwa tingkat akurasi serta nilai probabolitas *Naive Bayes* cukup baik dalam mengklasifikasikan jenis kelamin berdasarkan ciri. Hal ini dikarenakan faktor dari proses *resizing* dan ketepatan deteksi fitur wajah yang mempengaruhi jarak-jarak antara fitur.

# **BAB IV. ANALISIS DAN PERANCANGAN**

## **Analisis Kebutuhan**

Analisis kebutuhan sistem digunakan untuk mengetahui kebutuhan-kebutuhan apa saja yang diperlukan untuk merealisasikan sistem yang akan dibuat. Dibagian ini akan dibagi menjadi dua bagian yaitu analisis kebutuhan fungsional dan analisis kebutuhan non-fungsional.

* + 1. **Kebutuhan fungsional**

Kebutuhan fungsional yang dibutuhkan pada aplikasi klasifikasi jenis kelamin pada citra wajah menggunakan metode *Naive Bayes* adalah sebagai berikut:

* Sistem dapat menampilkan inputan *file image*.
* Sistem dapat melakukan pendeteksian wajah serta fitur-fitur wajah ,seperti mata,hidung,mulut dengan metode *Haar Cascade Classifier*.
* Sistem dapat melakukan ekstraksi fitur dengan metode PCA (*Principal Component Analysis*).
* Sistem dapat menampilkan hasil setiap perhitungan dari ekstraksi fitur.
* Sistem dapat menyimpan hasil ekstraksi fitur.
* Sistem dapat melakukan klasifikasi jenis kelamin dengan metode *Naive Bayes*.
* Sistem dapat menampilkan hasil akhir yaitu klasifikasi jenis kelamin (Laki-laki dan Perempuan).
  + 1. **Kebutuhan non fungsional**

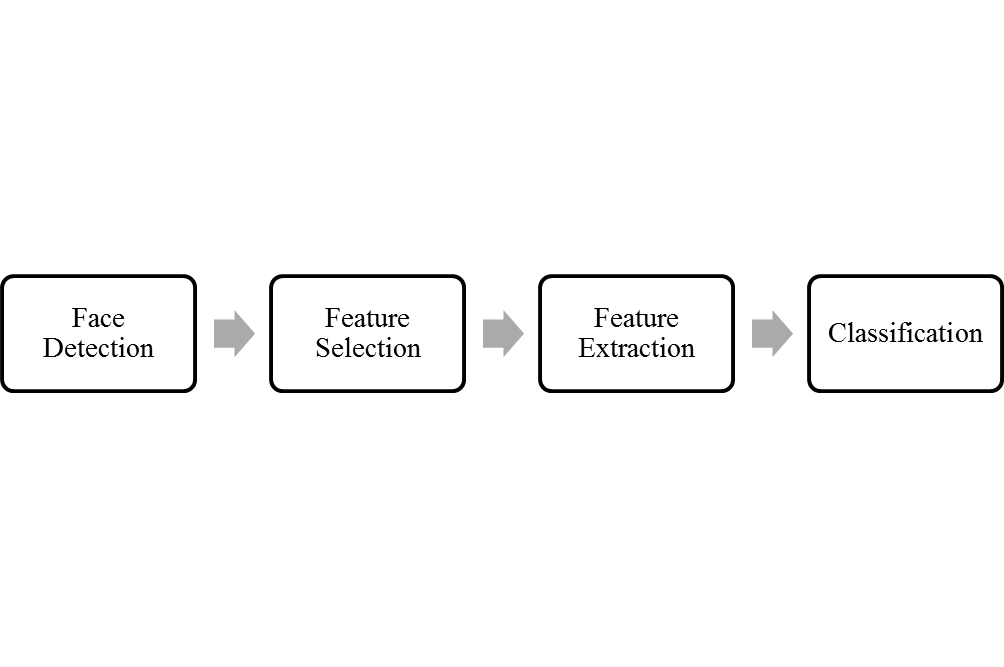
Untuk kebutuhan non fungsional sistem ini dibutuhkan perangkat lunak (*software*) sebagai berikut :

* Microsoft Wndows 10
* Microsoft Visual C# .NET
* Microsoft .NET Framework 4.5
* SQL Server 2014
* EmguCV 2.2.1
* Microsoft Visio 2013
* Microsoft Excel 2016

## **Perancangan Sistem**

* + 1. **Blok Diagram Sistem**

Secara umum, Blok Diagram dari aplikasi yang akan dibangun ditampilkan pada gambar berikut ini :



Gambar 4. 1 Blok Diagram Sistem

* *Face Detection*

Deteksi wajah berdasarkan jenis kelamin ini menggunakan library *EmguCV* dengan menggunakan algoritma *Haar Cascade Classifier*. Didalam metode ini sistem pendeteksian wajah khusus untuk deteksi wajah dengan posisi menghadap kamera (*frontal face*) sehingga tidak dapat mendeteksi wajah yang bukan dalam posisi tersebut.

* *Feature Selection*

Pada proses ini dilakukan pemilihan fitur wajah, proses yang dilakukan sama seperti pendeteksian wajah sebelumnya dengan menggunakan *library* EmguCV. Fitur-fitur wajah yang dibutuhkan yaitu mata, hidung, dan mulut.

* *Feature Extraction*

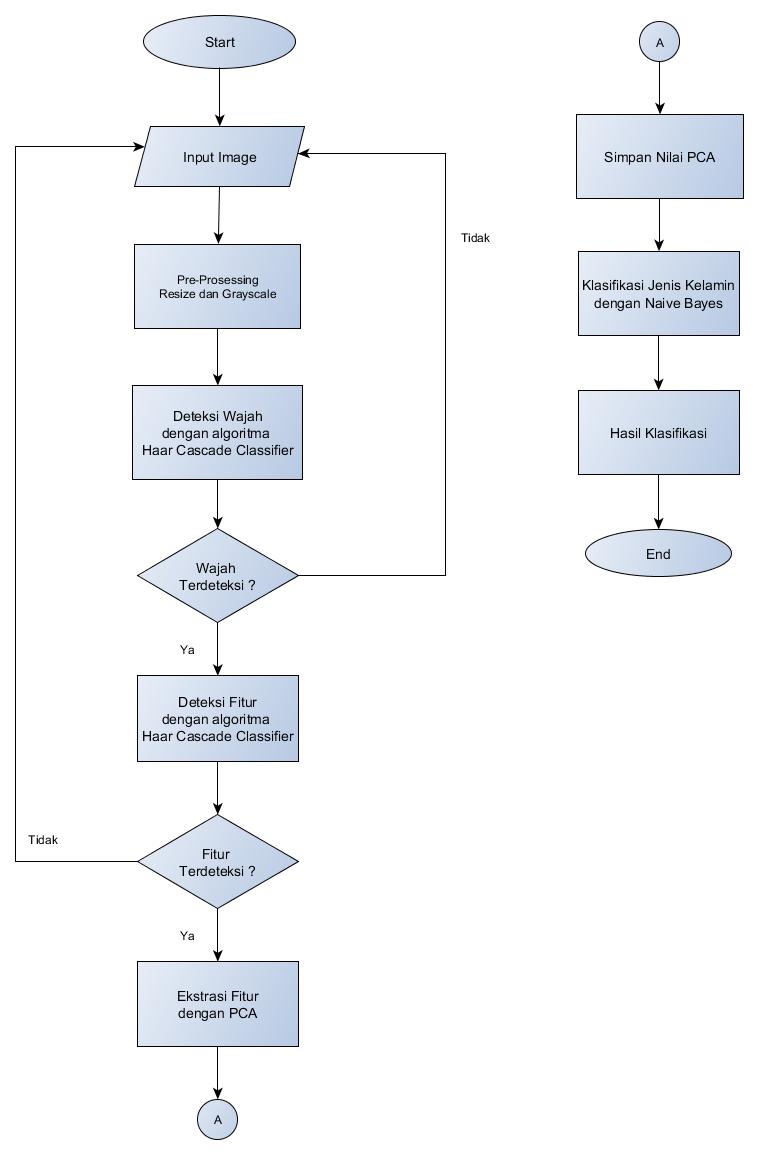
Ekstrasi Fitur merupakan suatu pengambilan ciri atau fitur dari *image* yang nantinya akan menghasilkan sebuah nilai berupa matriks kemudian disimpan di database lalu di analisis untuk proses selanjutnya yaitu klasifikasi *gender*. Ekstraksi fitur disini menggunakan metode *Principal Component Analysis* (PCA) dimana bahasa pemrograman yang digunakan adalah C#.

Metode tersebut bekerja dengan cara mengurangi jumlah dimensi dari citra masukan dan menangkap variasi total dari citra masukan tersebut yang kemudian direduksi menjadi nilai yang lebih kecil (bilangan real) dibandingkan nilai masukan.

* *Classification*

Klasifikasi merupakan suatu pengelompokan sebuah objek yang diuji kedalam kelas yang ditentukan. Pada proses klasifikasi disini menggunakan metode Naive Bayes dimana hasil klasifikasi adalah jenis kelamin yang terdiri dari laki-laki dan perempuan dimana penerapannya pada pengujian aplikasi menggunakan bahasa pemrograman C#.

* + 1. **Analisis Pembuatan Data Training**



Gambar 4. 2 Flowchart sistem

Keterangan :

1. Input image

Pada proses pertama untuk membuat suatu dataset untuk data *training* adalah dengan mengambil foto yang ada di penyimpanan.

1. Deteksi wajah dengan algoritma *Haar Cascade Classifier*

Pengambilan image yang nanti didapatkan objek wajah manusia yang akan dilakukan pendeteksian wajah dengan metode *Haar Cascade Classifier* dengan *library* EmguCV.

1. Deteksi Fitur Wajah (Mata, hidung dan mulut)

Setelah wajah didapatkan maka selanjutnya dilakukan pendeteksian fitur wajah seperti maa,hidung,mulut dengan algoritma *Haar Cascade Classifier.*

1. Ekstraksi fitur dengan *Principal Component Analysis*

Dari hasil citra wajah yang sudah didapatkan, langkah selanjutnya adalah melakukan proses ekstraksi fitur untuk mereduksi serta mengambil nilai-nilai dari setiap fitur wajah.

1. Simpan Nilai ekstraksi fitur PCA

Setelah nilai-nilai dari fitur wajah didapatkan, maka langkah selanjutnya adalah melakukan penyimpanan nilai-nilai tersebut kedalam database.

1. Klasifikasi jenis kelamin dengan metode *Naive Bayes*

Dari hasil ekstraksi fitur yang nilainya disimpan ke dalam database maka dapat dilakukan klasifikasi berdasarkan nilai-nilai tersebut yang nantinya akan digunakan sebagai acuan untuk data *testing*.

* + 1. **Analisis Pendeteksian Wajah**

Aplikasi pendeteksian wajah berdasarkan jenis kelamin ini menggunakan library EmguCV dengan menggunakan algoritma *Haar Cascade Classifier.* Didalam metode ini sistem pendeteksian wajah khusus untuk deteksi wajah dengan posisi menghadap pada kamera (*frontal face*) sehingga tidak dapat mendeteksi wajah yang bukan dalam posisi tersebut. Berikut ini merupakan flowchart proses pendeteksian wajah dengan *Haar Cascade Classifier* :



Gambar 4. 3 Flowchart deteksi dengan Haar Cascade Classifier

1. *Convert image* RGB ke *grayscale*

Nilai citra RGB diubah menjadi gambar yang terdiri dari warna putih dan gradasi warna hitam yaitu *grayscale*.

1. Menentukan Fitur *Haar* (*Haar Feature*)

Setelah image diubah menjadi *grayscale* kemudian dilakukan pencarian posisi wajah yaitu dengan cara mencari fitur-fitur yang memiliki tingkat pembeda yang tinggi untuk mengevaluasi setiap fitur terhadap data latih dengan menggunakan nilai fitur tersebut. Dalam proses pencarian wajah fitur yang memiliki batas terbesar antara wajah dan bukan wajah dianggap sebagai fitur yang terbaik.

*Haar feature* memproses gambar dalam kotak-kotak, dimana dalam satu kotak terdapat beberapa piksel. Kemudian diproses dan didapatkan perbedaan nilai (threshold) yang menandakan daerah gelap dan terang.

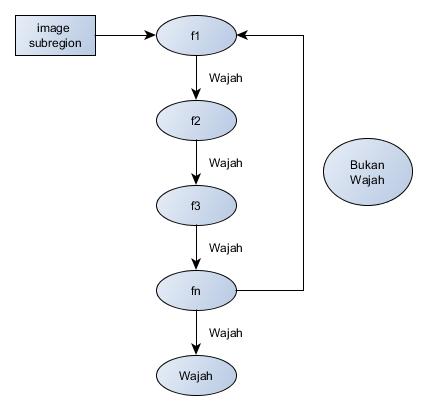
1. Menentukan dan menghitung nilai *Haar Feature* dengan *Integral Image*

Menghitung nilai fitur pada *haar feature* adalah melakukan pengurangan nilai piksel pada area hitam dengan piksel pada area putih jika nilai perbedaannya diatas nilai ambang (*threshold*), maka fitur tersebut ada.

Proses pencarian nilai fitur ini dilakukan secara iterative mulai dari ujung kiri atas sampai ujung kanan bawah dengan pergeseran sebesar piksel pada citra.

1. Pembuatan *Cascade Classifier*

Pada proses ini menggunakan klasifikasi bertingkat. Filter pada masing-masing level mengklasifikasikan gambar yang sebelumnya telah difilter. Jika satu dari filter-filter tersebut gagal mka daerah pada gambar diklasifikasikan sebagai “Bukan Wajah”. Saat filter berhasil melewatkan *image region, image region* kemudian masuk pada filter yang selanjutnya. *Image region* yang telah melalui semua filter akan dianggap sebagai “wajah”. Berikut adalah alur *filter cascade classifier*.



Gambar 4. 4 Alur sebuah cascade pendeteksian

* + 1. **Analisis Ekstraksi Fitur Wajah**

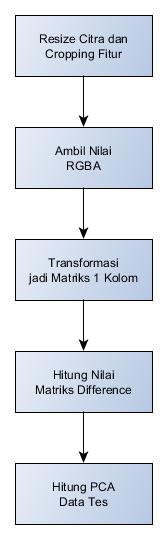
Berikut ini merupakan alur sistem dari proses ekstraksi fitur wajah adalah sebagai berikut:

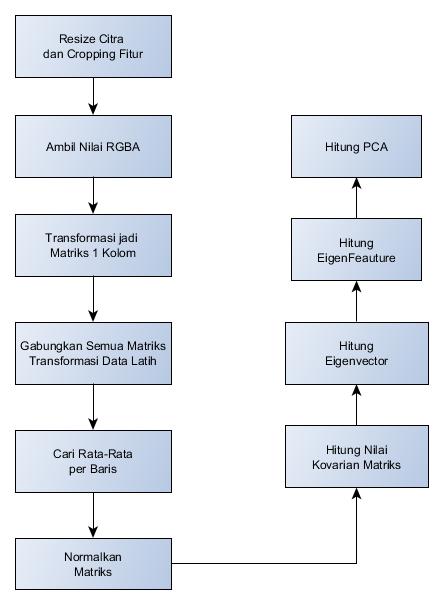
**Proses Ekstraksi**

**Data Tes**

**Proses Ekstraksi**

**Data Latih**





Gambar 4. 5 Alur ekstraksi fitur PCA

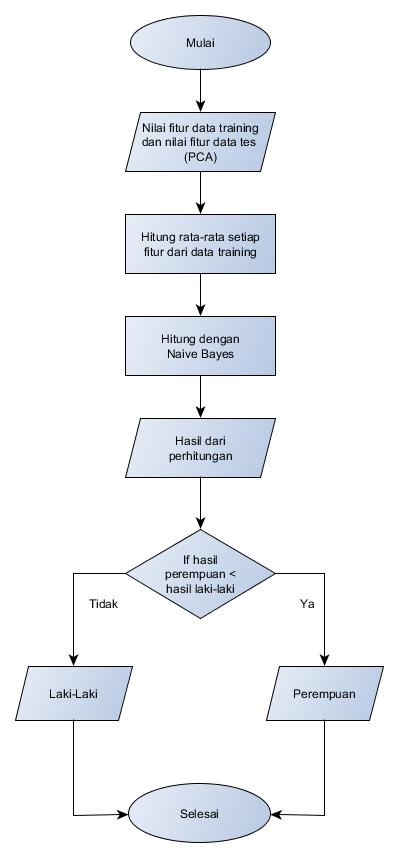
Proses PCA dibagi menjadi dua yaitu PCA Data Latih dan PCA Data Tes. Untuk PCA data latih diambil dari transformasi matriks tiap-tiap citra yang digabungkan menjadi satu matriks. Kemudian dicari rata-rata per baris. Setelah itu, matriks dinormalkan dengan cara mengurangkan matriks gabungan dengan nilai rata-rata per baris. Lalu menghitung nilai kovarian matriks dengan cara mentransposkan matriks yang telah dinormalisasi kemudian mengalikannya dengan matriks normal. Dari matriks kovarian akan didapatkan *Eigenvector* yang nantinya digunakan untuk menghitung *eigenfeature*. Setelah *eigenfeature* didapatkan maka dihitung nilai PCA data latih.

Untuk mendapatkan nilai PCA data tes dilakukan dengan cara mentransformasi matriks nilai RGBA kemudian menghitung nilai *Difference*. Lalu nilai PCA data tes akan dihasilkan dengan mengalikan matriks *EigenFeature* dan *Difference*.

Jika kedua macam nilai PCA tersebut telah didapatkan maka tingkat kemiripan akan dihitung dengan metode Naive bayes.

* + 1. **Analisis Klasifikasi Jenis Kelamin**

Dalam tahap ini, setelah dilakukan ekstraksi fitur maka didapatkan nilai dari proses tersebut. Selanjutnya, dari data *training* yang sudah dilakukan ekstraksi fitur maka nilai-nilai tersebut dikumpulkan untuk digunakan pada tahap klasifikasi. Pada tahap ini digunakan metode Naive Bayes sebagai klasifikasi jenis kelamin. Berikut adalah alur klasifikasi jenis kelamin pada citra wajah :



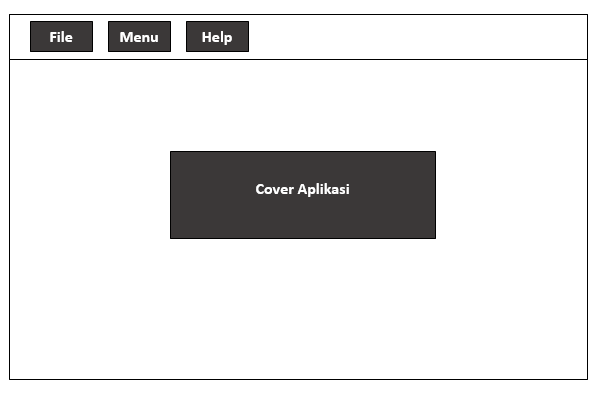
Gambar 4. 6 Alur klasifikasi jenis kelamin

## **Desain Interface**

Pada aplikasi Klasifikasi Jenis Kelamin Pada Citra Wajah ini terdapat 3 form diantaranya adalah :

* + 1. **Form Menu Utama**

Form Menu merupakan form tampilan awal yang berisikan menu pada aplikasi ini. Di dalamnya terdapat 3 *strip menu*



Gambar 4. 7 Halaman utama sistem

Keterangan :

1. *File* : Berisikan menu exit yang fungsinya untuk keluar dari aplikasi.
2. *Menu* : Berisikan 2 menu yaitu Pelatihan Dan Pengujian
3. *Help* : Berisikan menu about yang didalamnya terdapat sebuah jurnal aplikasi.
   * 1. **Form Pelatihan ( *Training* )**

Form Training merupakan form untuk mengambil data latih atau *training* untuk aplikasi ini. Di dalamnya terdapat beberapa sub menu diantaranya sebagai berikut :

* *Input* Citra

*Input* Citra merupakan sebuah langkah awal untuk melatih sebuah data. Di dalam sub menu ini terdapat 6 *button*, 2 *picture box*, 1 label, 2 *text box* dan 3 *group box*.



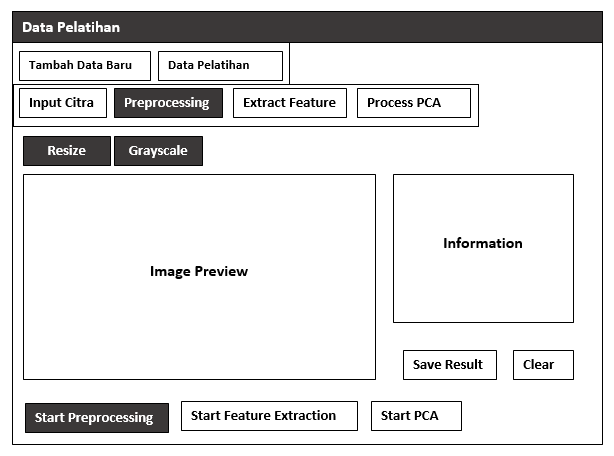
Gambar 4. 8 Halaman Training ( input citra)

Keterangan :

1. *Sub menu* Tambah Data Baru : digunakan untuk membuat data *training* baru.
2. *Sub menu* Data Pelatihan : digunakan untuk tempat penyimpanan data *training.*
3. *Button Open File :* berfungsi untuk mengambil data image pada *drive*.
4. *File :* berisikan letak *image* yang akan *diinputkan*
5. *Picture Box Image Preview :* berfungsi untuk menampilkan *image* data *training*.
6. *Picture Box Information :* berfungsi untuk menampilkan informasi *image.*
7. *Button Start Preprocessing :* berfungsi untuk memulai *preprocessing image.*
8. *Button Feature Extraction :* berfungsi untuk memulai *ekstraksi fitur.*
9. *Button Start PCA :* berfungsi untuk memulai proses *PCA.*
10. *Button Save Result :* berfungsi untuk menyimpan hasil *training* data.
11. *Button Clear :* digunakan untuk menghapus data.

* Preprocessing

*Preprocessing* merupakan sebuah *sub menu* untuk memproses image yang ada. Didalam sub menu tersebut ada 2 proses yaitu *resize* dan grayscale serta terdapat 2 *picture box* dan 5 *button.*



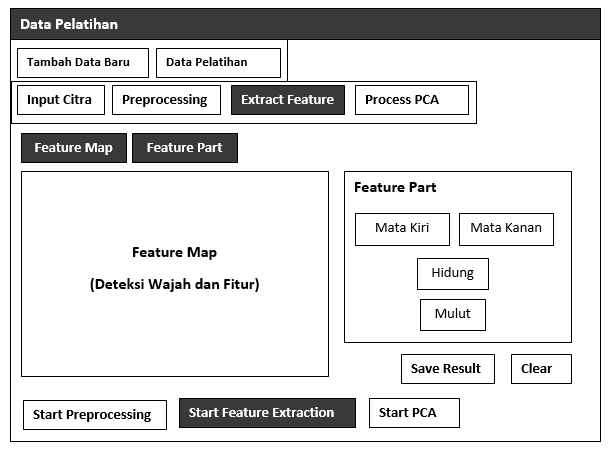
Gambar 4. 9 Halaman training (pre-processing)

Keterangan :

1. *Picture Box Image Preview :* berfungsi untuk menampilkan *image* yang telahdiubah *yaitu* di *resize* sertadi *grayscale* .
2. *Picture Box Information :* berfungsi untuk menampilkan informasi *image.*
3. *Button Start Preprocessing :* berfungsi untuk memulai *preprocessing image.*
4. *Button Feature Extraction :* berfungsi untuk memulai *ekstraksi fitur.*
5. *Button Start PCA :* berfungsi untuk memulai proses *PCA.*
6. *Button Save Result :* berfungsi untuk menyimpan hasil *training* data.
7. *Button Clear :* digunakan untuk menghapus data.

* Ekstraksi Fitur

Ekstraksi Fitur merupakan sebuah sub menu untuk medeteksi wajah serta fitur yang dibutuhkan. Didalam menu tersebut terdapat 2 bagian yaitu *feature map* dan *feature part* serta 2 *picture box* dan 5 *button.*



Gambar 4. 10 Halaman training (cropping fitur)

Keterangan :

1. *Picture Box Feature Map :* berfungsi untuk menampilkan *image* yang telah berhasil di deteksi
2. *Picture Box Feature part :* berfungsi untuk menampilkan fitur wajah yang dibutuhkan yaitu mata hidung dan mulut.
3. *Button Start Preprocessing :* berfungsi untuk memulai *preprocessing image.*
4. *Button Feature Extraction :* berfungsi untuk memulai *ekstraksi fitur.*
5. *Button Start PCA :* berfungsi untuk memulai proses *PCA.*
6. *Button Save Result :* berfungsi untuk menyimpan hasil *training* data.
7. *Button Clear :* digunakan untuk menghapus data.

* Proses PCA

Proses PCA merupakan sebuah menu untuk memproses fitur menggunakan metode *Principal Component Analysis .* Didalam menu tersebut terdapat 2 *radio button* , 1 tabel *gridview* , dan 5 *button.*

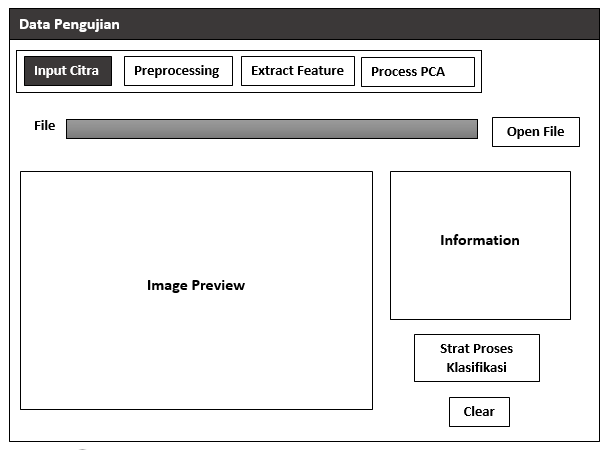


Gambar 4. 11 Halaman training (proses PCA)

Keterangan :

1. *Radio button Male dan Female :* berfungsi untuk memilih sebuah *gender* yang akan diproses pada proses selanjutnya yaitu klasifikasi.
2. *Picture box tampilan fitur :* digunakan untuk menampilkan gambar masing-masing fitur.
3. *Tabel data gridview :* digunakan untuk menampilkan hasil nilai *PCA* pada masing-masing fitur
4. *Button Start Preprocessing :* berfungsi untuk memulai *preprocessing image.*
5. *Button Feature Extraction :* berfungsi untuk memulai *ekstraksi fitur.*
6. *Button Start PCA :* berfungsi untuk memulai proses *PCA.*
7. *Button Save Result :* berfungsi untuk menyimpan hasil *training* data.
8. *Button Clear :* digunakan untuk menghapus data.
   * 1. **Form Pengujian ( *Testing* )**

Form Pengujian merupakan form yang digunakan untuk klasifikasi dengan metode *Naive Bayes*. Didalmnya terdapat 3 *button* dan2 *picture box.*



Gambar 4. 12 Halaman Testing

Keterangan :

1. *Button Open File :* berfungsi untuk mengambil data image pada *drive*.
2. *Button Start* proses klasifikasi : berfungsi untuk memulai proses klasifikasi jenis kelamin.
3. *Picture Box Image Preview :* berfungsi untuk menampilkan *image* data *training*.
4. *Picture Box Information :* berfungsi untuk menampilkan informasi *image.*
5. *Button Clear :* digunakan untuk menghapus data.

## **Desain Database**

Berikut ini merupakan desain *table* (Nama *database* :tbPelatihan) pada *training* untuk penyimpanan nilai-nilai dari data *training* :

Tabel 4. 1 Tabel Detail data training

|  |  |
| --- | --- |
| Nama Atribut | Tipe Data |
| IdPelatihan | Int |
| NamaFile | Nvarchar |
| Gender | Nvarchar |
| Hidung | Ntext |
| Mulut | Ntext |
| MataKanan | Ntext |
| MataKiri | Ntext |

# **BAB V. IMPLEMENTASI**

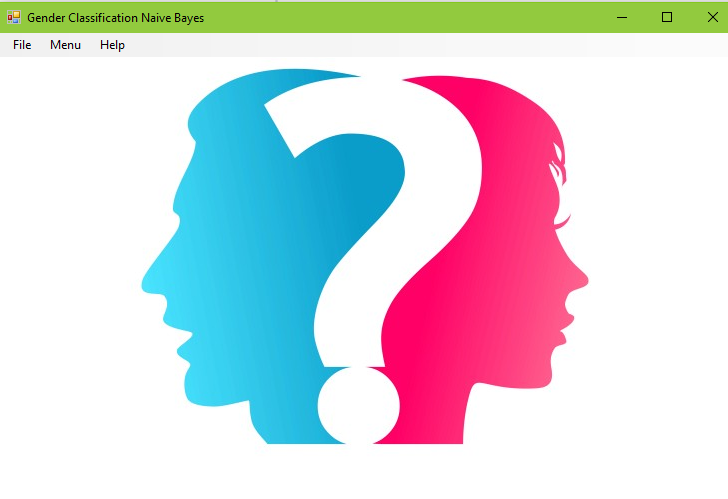
## **Implementasi Proses**

Pada tahap ini akan dilakukan implementasi aplikasi klasifikasijenis kelamin pada citra wajah menggunakan metode *Naive Bayes* dengan menggunakan bahasa pemrograman Visual C#. Dibawah ini merupakan langkah-langkah penggunaan aplikasi tersebut .

* + 1. **Proses pembuatan data training**

Untuk melakukan proses penambahan Data *Training,* ada beberapa proses yang harus dilakukan yaitu mengambil data objek wajah, dalam aplikasi ini fitur yang digunakan ialah mata,hidung dan mulut. Berikut ini adalah proses dalam pembuatan data *Training* :

1. Form Utama

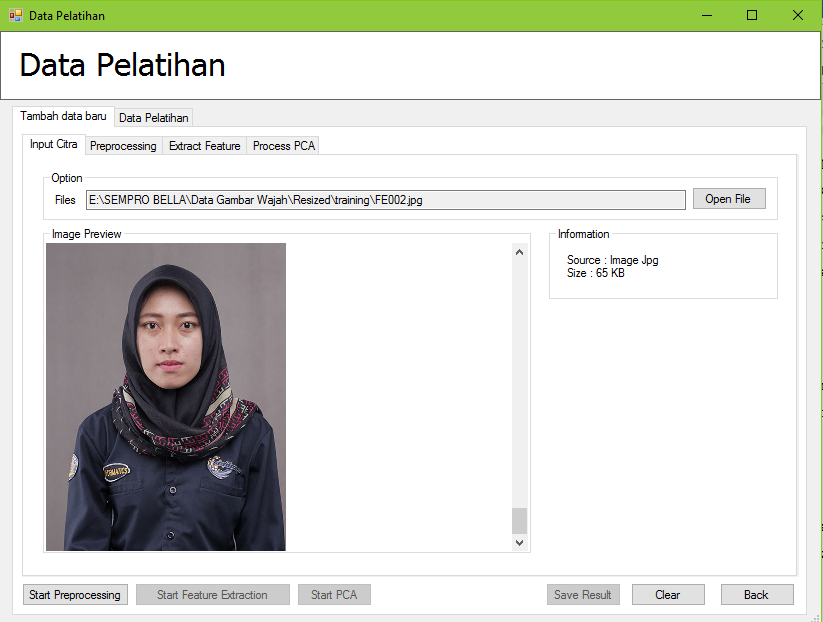


Gambar 5. 1 Tampilan Form Utama

Berikut adalah penjelasan dari menu utama :

* *File* 🡪 Pada menu *file* terdapat *sub menu* *exit* yang berfungsi untuk keluar dari aplikasi
* Menu 🡪 Pada Menu terdapat 2 *sub menu* yaitu Pelatihan dan Pengujian. Pada menu pelatihan berfungsi untuk melatih sebuah data sedangkan pada pengujian berfungsi untuk menguji hasil data latih yang sudah ada lalu di klasifikasikan.
* *Help* 🡪 Pada Menu *Help* terdapat *sub menu* *about* yang didalamnya berisi sebuah jurnal gambaran dari aplikasi.

1. Form Training



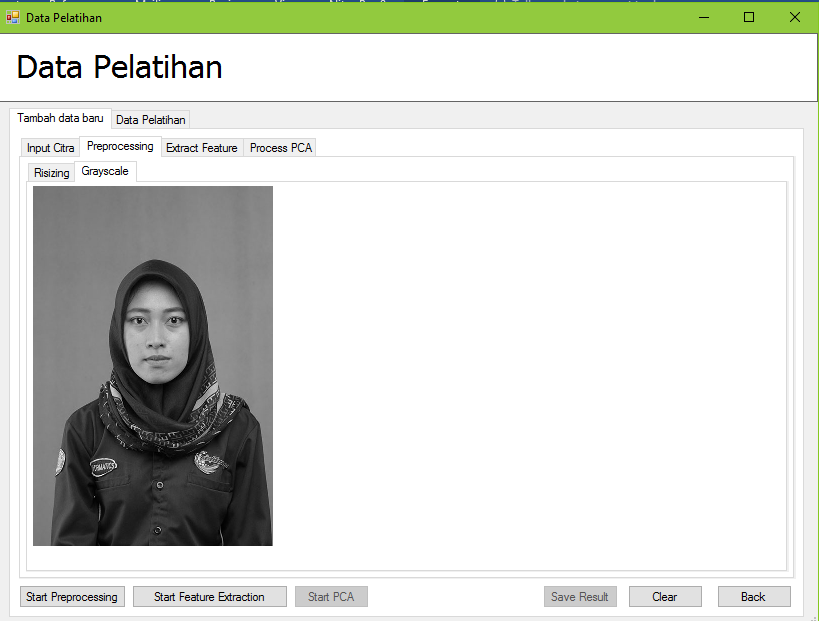
Gambar 5. 2 Tampilan Form Training

Langkah-langkah untuk melakukan pelatihan data yaitu:

1. Pada menu *input* citra klik *button* *open file* lalu pilih *image* yang diinginkan.

Berikut adalah *source code* untuk proses *input image* :

|  |
| --- |
| private void LoadFromFile(string nama)  {  this.\_frame = new Image<Bgr, Byte>(nama);  } |



Gambar 5. 3 Tampilan Form *pre-processing*

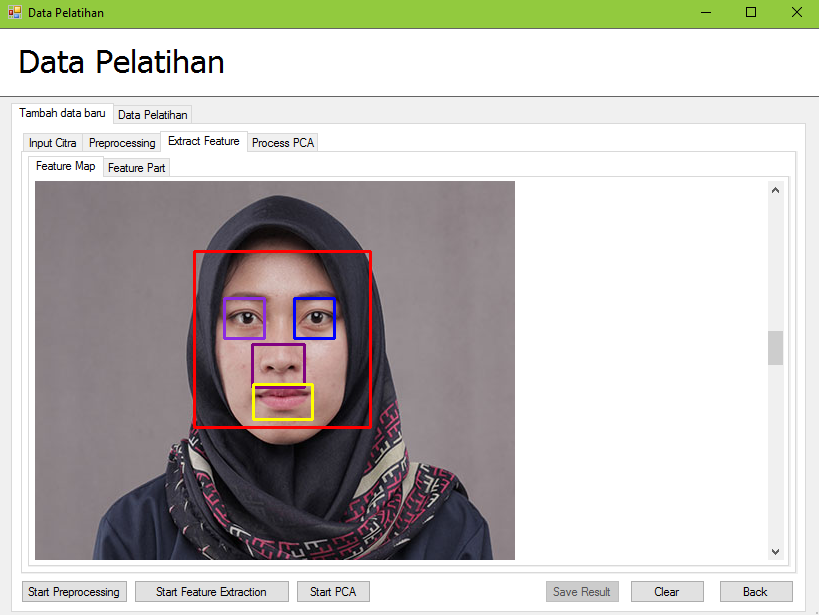
1. Selanjutnya klik *start preprocessing* untuk melakukan proses sebuah *image*. Proses tersebut yaitu *resize* *image* dan *grayscale.*

Berikut adalah *source code* untuk proses *resize* :

|  |
| --- |
| private void Resizes(string nama)  {  this.\_frame = new Image<Bgr, byte>(nama);  int width = 480;  int height = 720;    \_frame.Resize(width, height, Emgu.CV.CvEnum.INTER.CV\_INTER\_LINEAR, true);  } |

Berikut adalah *source code* untuk proses *grayscale* :

|  |
| --- |
| private void toGray()  {    this.\_grayFrame = this.\_frame.Convert<Gray, Byte>() ;  } |

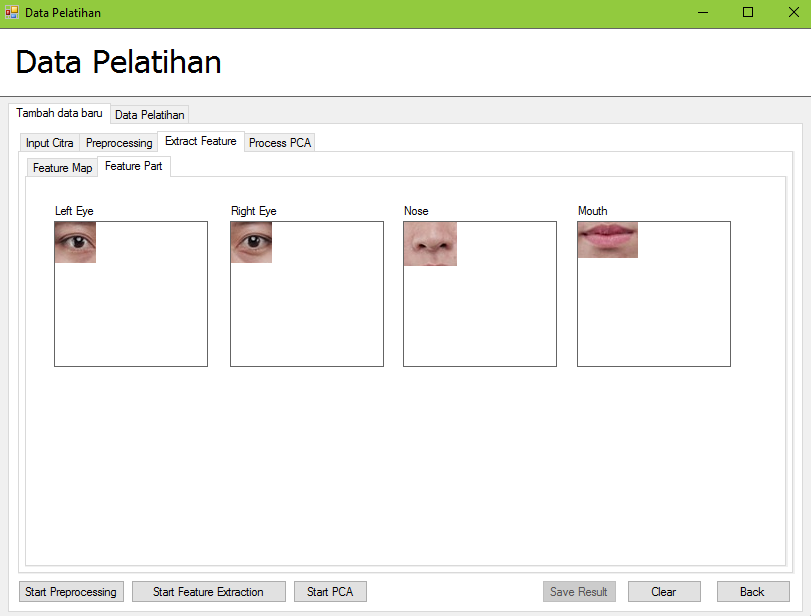


Gambar 5. 4 Tampilan Form deteksi wajah dan fitur

1. Setelah proses *grayscale* berhasil maka klik button *start feature extraction* untuk melakukan deteksi wajah serta fitur yang dibutuhkan.

Berikut adalah *source code* untuk proses deteksi wajah dan fitur mata, hidung, dan mulut :

|  |
| --- |
| foreach (Rectangle facerect in facesDetected)  {  FaceFeatures face = new FaceFeatures(facerect, grayImage.Cols, grayImage.Rows);  Console.WriteLine("Found Face");  grayImage.ROI = face.ProbableNoseLocation;  Rectangle[] noseDetected = noseClassifier.DetectMultiScale(  grayImage,  1.13,  3,  new Size(10,30),  Size.Empty);  grayImage.ROI = face.ProbableMouthLocation;  Rectangle[] mouthDetected = mouthClassifier.DetectMultiScale(  grayImage,  1.13,  3,  new Size(10,20),  Size.Empty);  grayImage.ROI = face.ProbableEyeLocation;  Rectangle[] doubelEyesDetected = eyeClassifier.DetectMultiScale(  grayImage,  1.13,  3,  new Size(10, 10),  Size.Empty);    grayImage.ROI = Rectangle.Empty;  face.AddNose(noseDetected);  face.AddMouth(mouthDetected);  face.AddEyes(doubelEyesDetected);    detectedFaces.Add(face);  } |



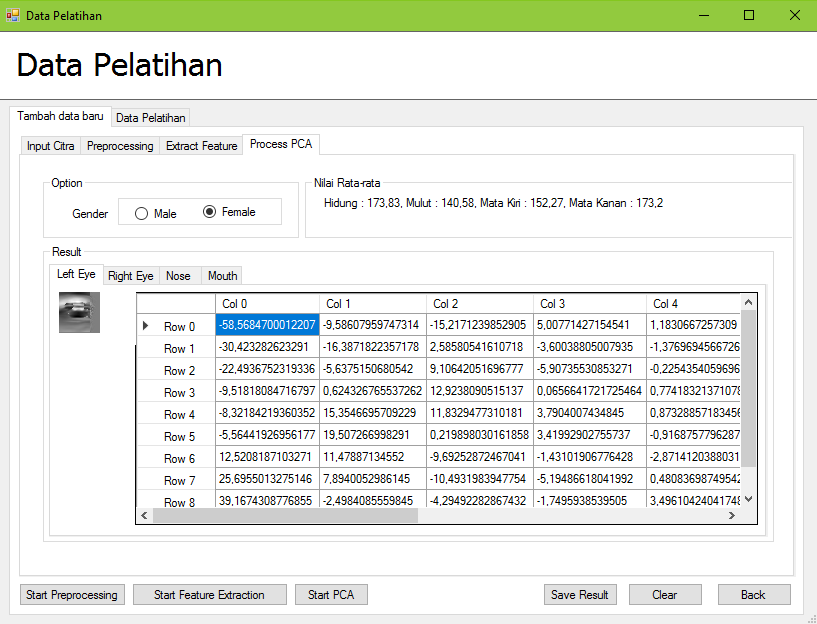
Gambar 5. 5 Tampilan Form Cropping fitur

1. Setelah wajah berhasil diambil maka saatnya untuk melakukan *cropping* fitur diantaranya yaitu mata kiri , mata kanan , hidung , dan mulut.

Berikut adalah *source code* untuk proses *cropping* fitur :

|  |
| --- |
| private Image<Bgr, byte> cropImage(Image<Bgr, byte> input, Rectangle rect)  {  var croppedImage = input.Clone();  croppedImage.ROI = rect;  return croppedImage;  } |

1. Form Ekstraksi Fitur



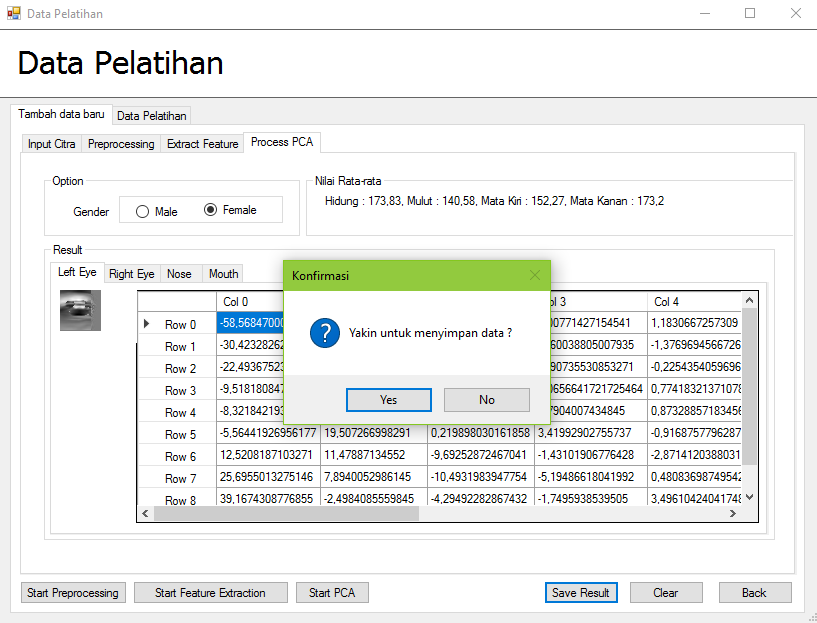
Gambar 5. 6 Tampilan Form Ekstraksi fitur dengan PCA

Langkah-langkah untuk melakukan proses ekstraksi fitur wajah adalah sebagai berikut :

1. Untuk menjalankan ekstraksi fitur dengan metode *Principal Component Analysis* , langkah pertama yaitu klik *button Start* *PCA*
2. Setelah itu muncul 4 fitur yang telah terdeteksi sebelumnya seperti mata kiri, mata kanan, hidung, dan mulut dengan nilai PCA pada masing-masing fitur wajah.
3. Klik *button female* atau *male* untuk mengisi jenis kelamin sesungguhnya
4. Selanjutnya mengambil nilai rata-rata atau *mean* pada masing-masing fitur sebagai inputan untuk proses selanjutnya yaitu klasifikasi

Berikut adalah *source code* untuk proses ekstrksifitur *PCA* :

|  |
| --- |
| StringBuilder sbNose, sbMouth, sbLeft, sbRight;  private Matrix<float> CalculatePCAofImage(Image<Gray, byte> FeatureImages, int WindowSize, MatrixBoxes MBox, PictureBox PBox)  {  int sX = FeatureImages.Width;  int sY = FeatureImages.Height;  int i, j;  FeatureImages.\_EqualizeHist();  Image<Gray, float> ImageFlo = FeatureImages.Convert<Gray, float>();  Matrix<float> Window = new Matrix<float>(WindowSize, WindowSize);  Matrix<float> Avg = new Matrix<float>(1, WindowSize);  Matrix<float> EigVals = new Matrix<float>(1, WindowSize);  Matrix<float> EigVects = new Matrix<float>(WindowSize, WindowSize);  Matrix<float> PCAFeatures = new Matrix<float>(WindowSize, WindowSize);  Image<Gray, float> TempIm = ImageFlo.CopyBlank();  for (i = 0; i < (sX - WindowSize); i++)  {  for (j = 0; j < (sY - WindowSize); j++)  {  CvInvoke.cvSetImageROI(ImageFlo, new Rectangle(new Point(i, j), new Size(WindowSize, WindowSize)));  CvInvoke.cvSetImageROI(TempIm, new Rectangle(new Point(i, j), new Size(WindowSize, WindowSize)));  CvInvoke.cvConvert(ImageFlo, Window);  CvInvoke.cvCalcPCA(Window, Avg, EigVals, EigVects, Emgu.CV.CvEnum.PCA\_TYPE.CV\_PCA\_DATA\_AS\_ROW);  try  {  CvInvoke.cvProjectPCA(Window, Avg, EigVects, PCAFeatures);  }  catch (Exception e)  {  throw (e);  }  CvInvoke.cvConvert(PCAFeatures, TempIm);  CvInvoke.cvResetImageROI(ImageFlo);  CvInvoke.cvResetImageROI(TempIm);  }  }  MBox.Matrix = Avg;  PBox.Image = TempIm.Bitmap;  return (Avg);  }  private void btnPca\_Click(object sender, EventArgs e)  {  Resizes(NamaFile);  resizeImageBox.Image = \_frame;  btnCapture.Enabled = false;  btnOpen.Enabled = false;  toGray();  grayscaleImageBox.Image = this.\_grayFrame;  TabControl1.SelectedIndex = 1;    File.Delete(Application.StartupPath + "\\left-eye.jpg");  File.Delete(Application.StartupPath + "\\right-eye.jpg");  File.Delete(Application.StartupPath + "\\nose.jpg");  File.Delete(Application.StartupPath + "\\mouth.jpg");  CascadeClassifier faceClassifier;  CascadeClassifier eyeClassifier;  CascadeClassifier mouthClassifier;  CascadeClassifier noseClassifier;  List<FaceFeatures> lastDetectedFaces = new List<FaceFeatures>();  faceClassifier = new CascadeClassifier(@"Resources\haarcascades\frontalface\_alt.xml");  eyeClassifier = new CascadeClassifier(@"Resources\haarcascades\eye.xml");  mouthClassifier = new CascadeClassifier(@"Resources\haarcascades\mouth.xml");  noseClassifier = new CascadeClassifier(@"Resources\haarcascades\nose.xml");  Image<Bgr, Byte> image = \_frame.Clone();  List<FaceFeatures> detectedFaces = new List<FaceFeatures>();  int jmlFeature = 0;  using (Image<Gray, Byte> grayImage = image.Convert<Gray, Byte>())  {  grayImage.\_EqualizeHist();  Rectangle[] facesDetected = faceClassifier.DetectMultiScale(  grayImage,  1.1,  3,  new Size(40, 40),  Size.Empty);  foreach (Rectangle facerect in facesDetected)  {  FaceFeatures face = new FaceFeatures(facerect, grayImage.Cols, grayImage.Rows);  Console.WriteLine("Found Face");  grayImage.ROI = face.ProbableNoseLocation;  Rectangle[] noseDetected = noseClassifier.DetectMultiScale(  grayImage,  1.13,  3,  new Size(10, 30),  Size.Empty);  grayImage.ROI = face.ProbableMouthLocation;  Rectangle[] mouthDetected = mouthClassifier.DetectMultiScale(  grayImage,  1.13,  3,  new Size(10, 20),  Size.Empty);  grayImage.ROI = face.ProbableEyeLocation;  Rectangle[] doubelEyesDetected = eyeClassifier.DetectMultiScale(  grayImage,  1.13,  3,  new Size(10, 10),  Size.Empty);  grayImage.ROI = Rectangle.Empty;  face.AddNose(noseDetected);  face.AddMouth(mouthDetected);  face.AddEyes(doubelEyesDetected);  detectedFaces.Add(face);  }  Image<Bgr, Byte> feaImg = \_frame.Clone();  foreach (FaceFeatures f in detectedFaces)  {  f.FeatureToImage(\_frame);  }  if (File.Exists(Application.StartupPath + "\\left-eye.jpg"))  {  leftIbox.Image = new Image<Bgr, Byte>(Application.StartupPath + "\\left-eye.jpg").ToBitmap();  jmlFeature++;  }  if (File.Exists(Application.StartupPath + "\\right-eye.jpg"))  {  rightIbox.Image = new Image<Bgr, Byte>(Application.StartupPath + "\\right-eye.jpg").ToBitmap();  jmlFeature++;  }  if (File.Exists(Application.StartupPath + "\\nose.jpg"))  {  nosIbox.Image = new Image<Bgr, Byte>(Application.StartupPath + "\\nose.jpg").ToBitmap();  jmlFeature++;  }  if (File.Exists(Application.StartupPath + "\\mouth.jpg"))  {  moutIbox.Image = new Image<Bgr, Byte>(Application.StartupPath + "\\mouth.jpg").ToBitmap();  jmlFeature++;  }  foreach (FaceFeatures f in detectedFaces)  {  f.DrawToImage(image, false);  }  feaMapImageBox.Image = image;//.ToBitmap();  }  TabControl1.SelectedIndex = 2;  if (jmlFeature >= 4)  {  doClassify();  TabControl1.SelectedIndex = 3;  }  else  {  MessageBox.Show("Proses Gagal, Feature wajah tidak lengkap..");  }  }  void Copy(string src, string dst)  {  if (File.Exists(dst))  {  File.Delete(dst);  }  File.Copy(src, dst);  }    private void button1\_Click(object sender, EventArgs e)  {  Close();  } |

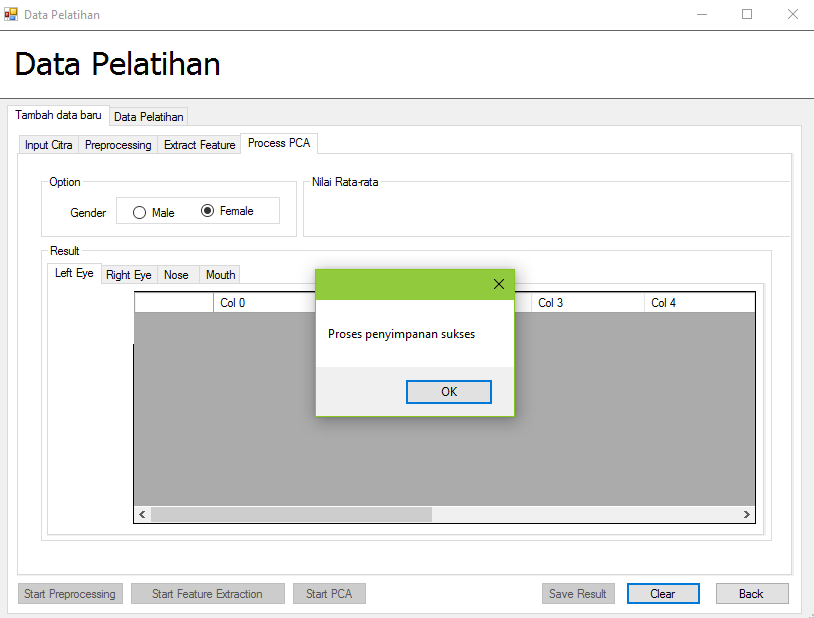


Gambar 5. 7 Proses penyimpanan data ke database

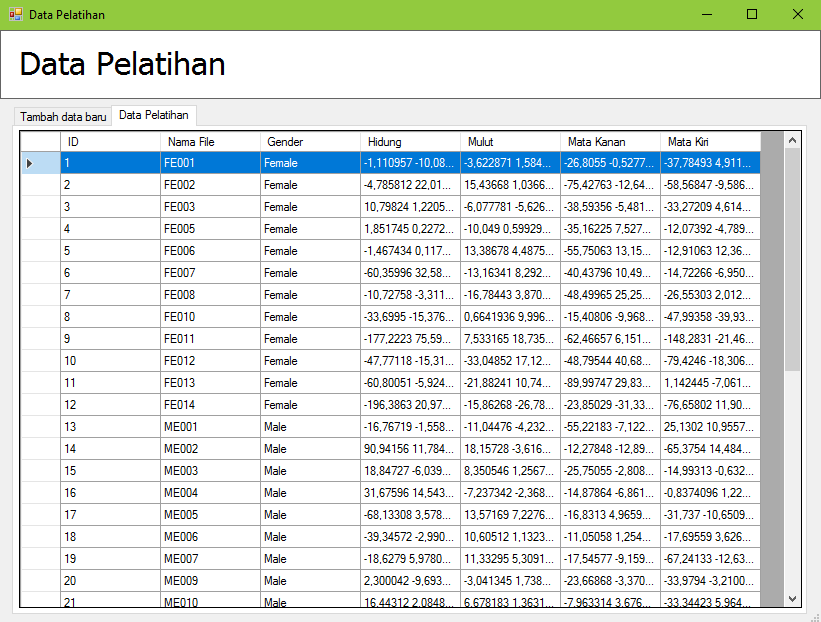
1. Setelah nilai rata-rata dan jenis kelamin terisi benar maka klik *button save result* untuk melakukan proses penyimpanan data pada *database* yang telah dibuat.

Berikut adalah *source code* untuk proses penyimpanannilai PCA training :

|  |
| --- |
| private static FormPelatihan \_DefaultInstance;  public static FormPelatihan DefaultInstance  {  get  {  if (\_DefaultInstance == null)  \_DefaultInstance = new FormPelatihan();  return \_DefaultInstance;  }  }  private Matrix<float> CalculatePCAofImage(Image<Gray, byte> FeatureImages, int WindowSize, MatrixBoxes MBox, PictureBox PBox)  {  int sX = FeatureImages.Width;  int sY = FeatureImages.Height;  int i, j;  Image<Gray, float> ImageFlo = FeatureImages.Convert<Gray, float>();  Matrix<float> Window = new Matrix<float>(WindowSize, WindowSize);  Matrix<float> Avg = new Matrix<float>(1, WindowSize);  Matrix<float> EigVals = new Matrix<float>(1, WindowSize);  Matrix<float> EigVects = new Matrix<float>(WindowSize, WindowSize);  Matrix<float> PCAFeatures = new Matrix<float>(WindowSize, WindowSize);  Image<Gray, float> TempIm = ImageFlo.CopyBlank();  for (i = 0; i < (sX - WindowSize); i++)  {  for (j = 0; j < (sY - WindowSize); j++)  {  CvInvoke.cvSetImageROI(ImageFlo, new Rectangle(new Point(i, j), new Size(WindowSize, WindowSize)));  CvInvoke.cvSetImageROI(TempIm, new Rectangle(new Point(i, j), new Size(WindowSize, WindowSize)));  CvInvoke.cvConvert(ImageFlo, Window);  CvInvoke.cvCalcPCA(Window, Avg, EigVals, EigVects, Emgu.CV.CvEnum.PCA\_TYPE.CV\_PCA\_DATA\_AS\_ROW);  try  {  CvInvoke.cvProjectPCA(Window, Avg, EigVects, PCAFeatures);  }  catch (Exception e)  {  throw (e);  }  CvInvoke.cvConvert(PCAFeatures, TempIm);  CvInvoke.cvResetImageROI(ImageFlo);  CvInvoke.cvResetImageROI(TempIm);  }  }  MBox.Matrix = Avg;  PBox.Image = TempIm.Bitmap;  return (PCAFeatures);  }  void testRecog()  {  String gender = RadioButton1.Checked ? "Male" : "Female";  Image<Gray, Byte>[] trainingImages = new Image<Gray, Byte>[1];  trainingImages[0] = new Image<Gray, byte>(Application.StartupPath + "\\nose.jpg");  String[] labels = new String[] { gender };  MCvTermCriteria termCrit = new MCvTermCriteria(16, 0.001);  PCA pcas = new PCA(  trainingImages,  labels,  5000,  ref termCrit);  Image<Gray, Byte> testImage = new Image<Gray, Byte>(Application.StartupPath + "\\nose.jpg");  String helo = pcas.Recognize(trainingImages[0]);  Console.WriteLine("Hasil " + helo);  mBoxNose.Matrix = pcas.EigenValues[0];  }  StringBuilder sbNose,sbMouth,sbLeft,sbRight;  private void btnPca\_Click(object sender, EventArgs e)  {  int WINDOW\_SIZE = 10;  String gender = RadioButton1.Checked ? "Male" : "Female";  StringBuilder sb = new StringBuilder();  sbNose = new StringBuilder();  sbMouth = new StringBuilder();  sbLeft = new StringBuilder();  sbRight = new StringBuilder();    Matrix<float> My\_Matrix\_Image = CalculatePCAofImage(new Image<Gray, Byte>(Application.StartupPath + "\\nose.jpg"), WINDOW\_SIZE, mBoxNose, pbNose);    float rataVals = 0;  for (int i = 0; i < My\_Matrix\_Image.Height; i++)  {  for (int j = 0; j < My\_Matrix\_Image.Width; j++)  {  rataVals += My\_Matrix\_Image.Data[i, j] ;  sbNose.Append(My\_Matrix\_Image.Data[i, j].ToString());  sbNose.Append(" ");  }  sbNose.Append(":");  }  rataVals /= (My\_Matrix\_Image.Height \* My\_Matrix\_Image.Width);  sb.Append("Hidung : " + rataVals.ToString());    My\_Matrix\_Image = CalculatePCAofImage(new Image<Gray, Byte>(Application.StartupPath + "\\mouth.jpg"), WINDOW\_SIZE, mBoxMouth, pbMouth);  rataVals = 0;  for (int i = 0; i < My\_Matrix\_Image.Height; i++)  {  for (int j = 0; j < My\_Matrix\_Image.Width; j++)  {  rataVals += My\_Matrix\_Image.Data[i, j];  sbMouth.Append(My\_Matrix\_Image.Data[i, j].ToString());  sbMouth.Append(" ");  }  sbMouth.Append(":");  }  rataVals /= (My\_Matrix\_Image.Height \* My\_Matrix\_Image.Width);  sb.Append(", Mulut : " + rataVals.ToString());  My\_Matrix\_Image = CalculatePCAofImage(new Image<Gray, Byte>(Application.StartupPath + "\\left-eye.jpg"), WINDOW\_SIZE, mBoxLeft, pbLeft);  rataVals = 0;  for (int i = 0; i < My\_Matrix\_Image.Height; i++)  {  for (int j = 0; j < My\_Matrix\_Image.Width; j++)  {  rataVals += My\_Matrix\_Image.Data[i, j];  sbLeft.Append(My\_Matrix\_Image.Data[i, j].ToString());  sbLeft.Append(" ");  }  sbLeft.Append(":");  }  rataVals /= (My\_Matrix\_Image.Height \* My\_Matrix\_Image.Width);  sb.Append(", Mata Kiri : " + rataVals.ToString());  My\_Matrix\_Image = CalculatePCAofImage(new Image<Gray, Byte>(Application.StartupPath + "\\right-eye.jpg"), WINDOW\_SIZE, mBoxRight, pbRight);  rataVals = 0;  for (int i = 0; i < My\_Matrix\_Image.Height; i++)  {  for (int j = 0; j < My\_Matrix\_Image.Width; j++)  {  rataVals += My\_Matrix\_Image.Data[i, j];  sbRight.Append(My\_Matrix\_Image.Data[i, j].ToString());  sbRight.Append(" ");  }  sbRight.Append(":");  }  rataVals /= (My\_Matrix\_Image.Height \* My\_Matrix\_Image.Width);  sb.Append(", Mata Kanan : " + rataVals.ToString());  TabControl1.SelectedIndex = 3;  lblInfoPCA.Text = sb.ToString();  btnSave.Enabled = true;  } |



Gambar 5. 8 Data tesimpan



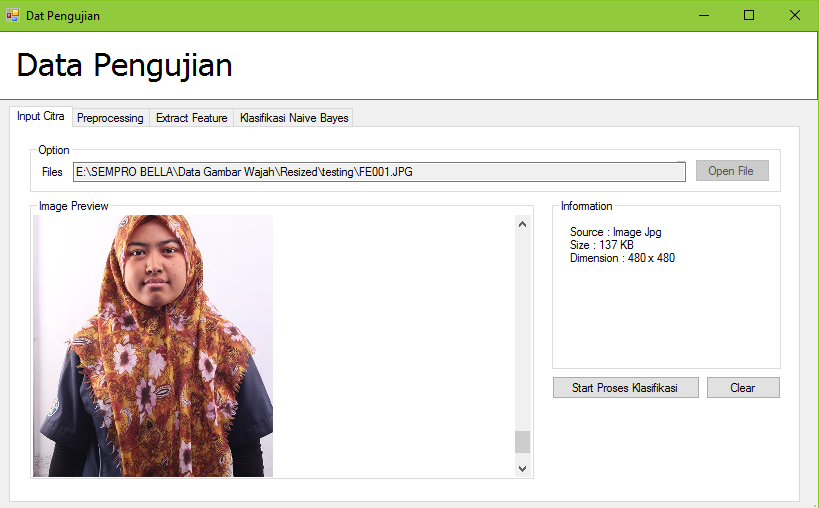
Gambar 5. 9 Tampilan data training

1. Berikut adalah tampilan pada data pelatihan yang sudah berhasil tersimpan.

Berikut adalah *source code* untuk proses penyimpananPCA :

|  |
| --- |
| private void btnSave\_Click(object sender, EventArgs e)  {  if (MessageBox.Show(this, "Yakin untuk menyimpan data ?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)  {    Copy(Application.StartupPath + "\\left-eye.jpg", Application.StartupPath + "\\Resources\\"+filesNames+"\_LEFT.jpg" );  Copy(Application.StartupPath + "\\right-eye.jpg", Application.StartupPath + "\\Resources\\" + filesNames + "\_RIGHT.jpg");  Copy(Application.StartupPath + "\\nose.jpg", Application.StartupPath + "\\Resources\\" + filesNames + "\_NOSE.jpg");  Copy(Application.StartupPath + "\\mouth.jpg", Application.StartupPath + "\\Resources\\" + filesNames + "\_MOUTH.jpg");  String gender = RadioButton1.Checked ? "Male" : "Female";  tbPelatihanTableAdapter.Insert(filesNames,gender,sbNose.ToString(),sbMouth.ToString(),sbRight.ToString(),sbLeft.ToString());    btnClear\_Click(sender, e);  MessageBox.Show("Proses penyimpanan sukses");  }  } |

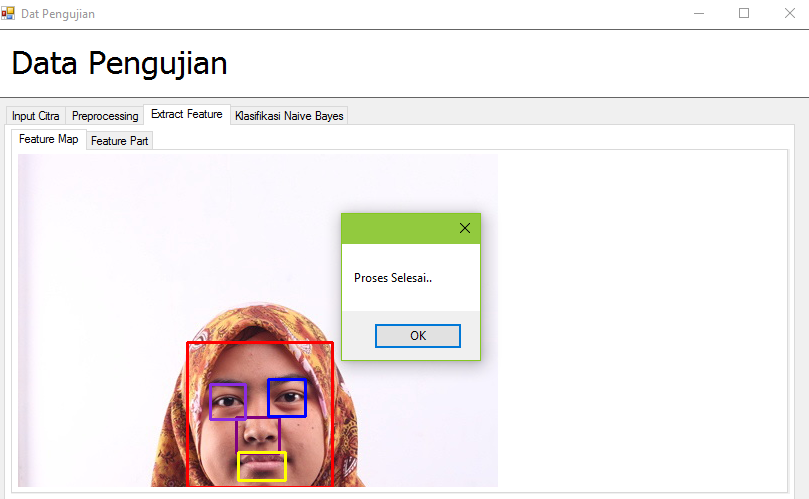
1. Form Testing



Gambar 5. 10 Tampilan Form Testing

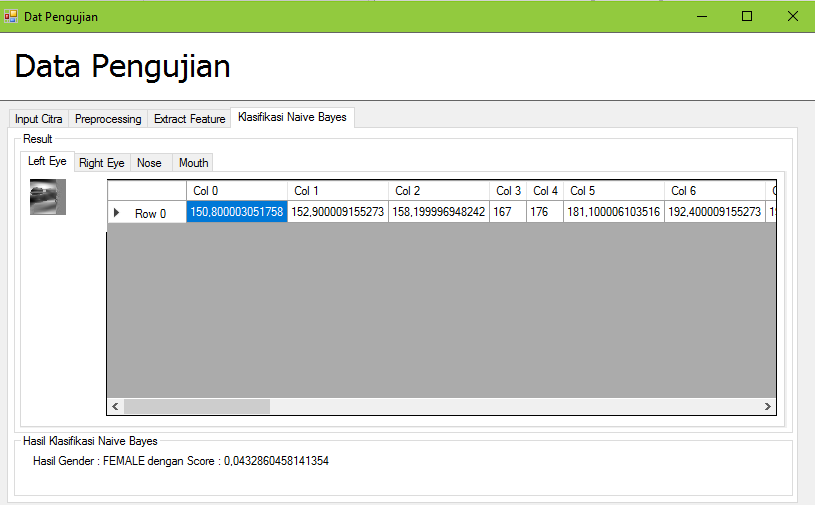
Langkah-langkah untuk melakukan pengujian, sebagai berikut :

1. Klik *button open file* untuk mengambil *image* yang ingin
2. Setelah *image* dipilih maka keluar pada *image box* yang sudah ada serta informasi dari *image* tersebut
3. Untuk melakukan klasifikasi maka klik *button start* proses klasifikasi



Gambar 5. 11 Objek wajah terdeteksi

1. Proses klasifikasi selesai , maka klik OK. Pada proses ini sudah otomatis mengeluarkan hasil *resize* , *grayscale* *image* serta deteksi wajah dan fitur.



Gambar 5. 12 Hasil klasifikasi jenis kelamin dengan Naive Bayes

1. Berikut adalah hasil klasifikasi jenis kelamin yang telah berhasil diuji dengan menghasilkan nilai *PCA* *10 x 1* serta nilai probabilitas menggunakan metode *Naive Bayes.*

Berikut adalah *source code* untuk *testing PCA* di tahap klasifikasi :

|  |
| --- |
| private void doClassify()  {  this.tbPelatihanTableAdapter.Fill(this.myDatabaseDataSet1.tbPelatihan);  string[,] LogArray = new string[dataGridView1.Rows.Count, dataGridView1.Columns.Count];  string[] Kelas = new string[dataGridView1.Rows.Count];  string[] Files = new string[dataGridView1.Rows.Count];  int i = 0, j = 0, k = 0;  int x = 0;  foreach (DataGridViewRow row in dataGridView1.Rows)  {  while (x < dataGridView1.Columns.Count)  {  LogArray[i, x] = row.Cells[x].Value != null ? row.Cells[x].Value.ToString() : string.Empty;  x++;  }  x = 0;  i++;  }  int jmlBaris = 10, jmlKolom = 10;  for (i = 0; i < dataGridView1.Rows.Count - 1; i++)  {  Files[i] = LogArray[i, 1];  Kelas[i] = LogArray[i, 2].ToUpper();  }  int WINDOW\_SIZE = 10;  PCA pcas = new PCA();  Matrix<float>[] ujiData = new Matrix<float>[4];  ujiData[0] = new Matrix<float>(1, WINDOW\_SIZE);  ujiData[1] = new Matrix<float>(1, WINDOW\_SIZE);  ujiData[2] = new Matrix<float>(1, WINDOW\_SIZE);  ujiData[3] = new Matrix<float>(1, WINDOW\_SIZE);  Matrix<float> My\_Matrix\_Mata\_Kiri\_Uji = CalculatePCAofImage(new Image<Gray, Byte>(Application.StartupPath + "\\left-eye.jpg"), WINDOW\_SIZE,mBoxLeft, pbLeft);  Matrix<float> My\_Matrix\_Mata\_Kanan\_Uji = CalculatePCAofImage(new Image<Gray, Byte>(Application.StartupPath + "\\right-eye.jpg"), WINDOW\_SIZE,mBoxRight,pbRight);  Matrix<float> My\_Matrix\_Hidung\_Uji = CalculatePCAofImage(new Image<Gray, Byte>(Application.StartupPath + "\\nose.jpg"), WINDOW\_SIZE,mBoxNose,pbNose);  Matrix<float> My\_Matrix\_Mulut\_Uji = CalculatePCAofImage(new Image<Gray, Byte>(Application.StartupPath + "\\mouth.jpg"), WINDOW\_SIZE,mBoxMouth,pbMouth);  for (j = 0; j < WINDOW\_SIZE; j++) |

## **Implementasi Metode**

Pada aplikasi ini menggunakan metode pengekstraksian fitur untuk mencari nilai-nilai dari setiap fitur yang nilainya akan diambil untuk pengklasifikasian jenis kelamin yaitu dengan *naive bayes*. Berikut ini merupakan perhitungan dari ekstraksi fitur sampai klasifikasi jenis kelamin.

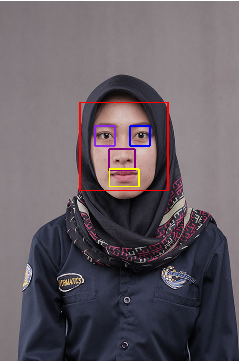
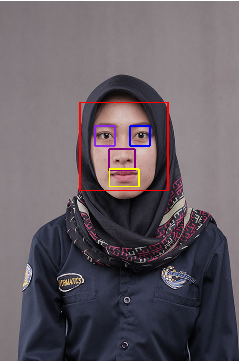
**5.2.1 Ekstraksi Fitur**

1. Citra input

Citra input yang digunakan adalah citra wajah yang tegak lurus menghadap kedepan. Pada proses ekstraksi wajah akan dilakukan konversi dari RGB ke *grayscale.*

Gambar 5. 13 Citra uji

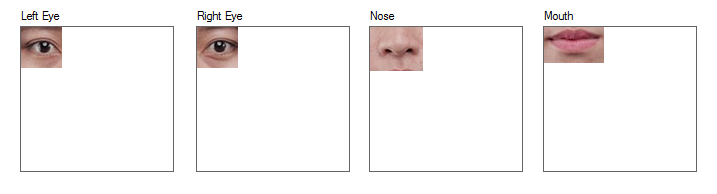
1. Pendeteksian wajah

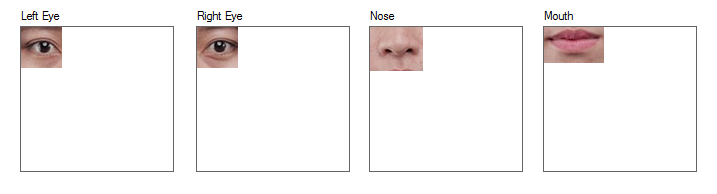
Pada proses ini dilakukan pendeteksian wajah dengan fitur yang dibutuhkan

Gambar 5. 14 Citra deteksi fitur

1. Cropping fitur

Paa proses ini dilakukan pengambilan fitur yang sudah terdeteksi diantaranya adalah mata , hidung, dan mulut yang nantinya akan diambil nilai ciri dari masing-masing *image.*



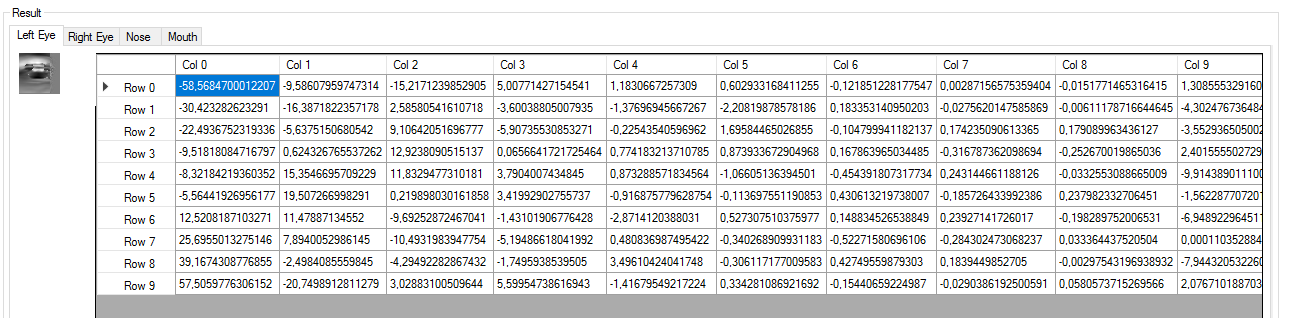


Gambar 5. 15 Pengambilan fitur

1. Proses Penerapan Metode *Principal Component Analysis (PCA)*

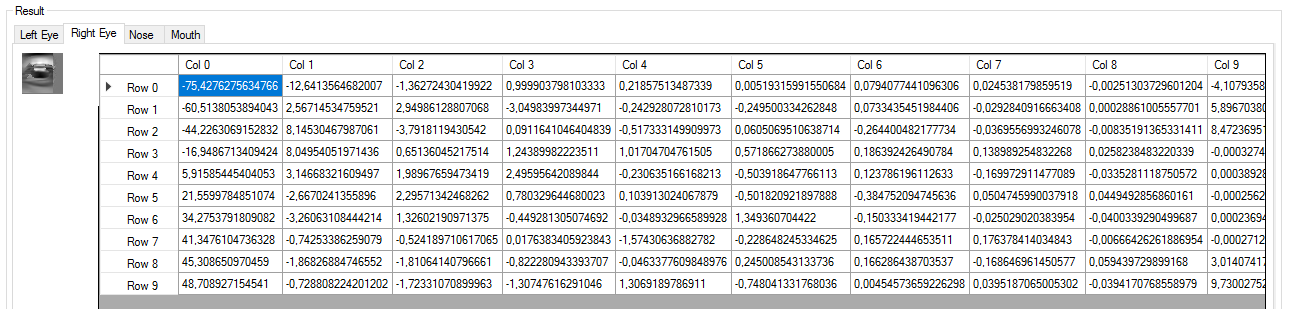
Pada Proses ini terdapat *matriks* fitur PCA dengan ukuran10 x 10 kemudian dirata-rata dengan cara di proses *resize* menjadi *matriks* nilai rata-rata dengan ukuran 10 x 1, dimana *matriks* tersebut akan digunakan untuk klasifikasi *naive bayes.* Berikut adalah hasil nilai *PCA* pada masing-masing fitur wajah :

Nilai ciri PCA dari mata kiri



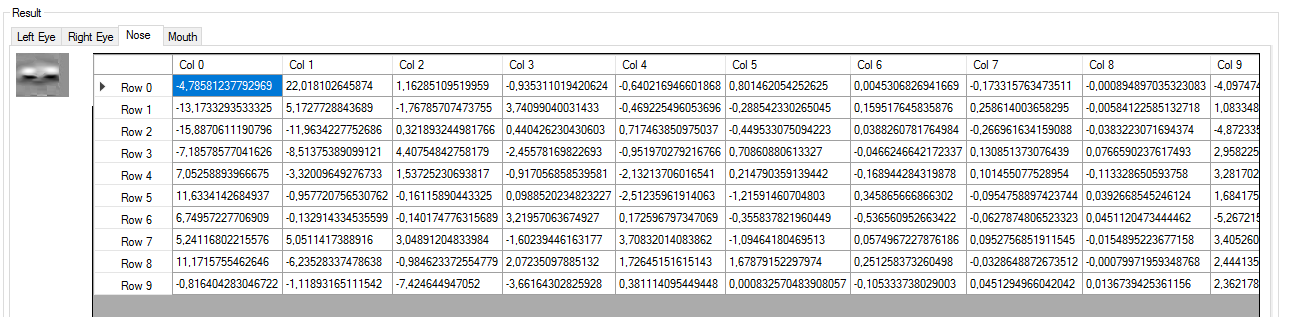
Gambar 5. 16 Nilai fitur Mata kiri

Nilai ciri PCA dari mata kanan



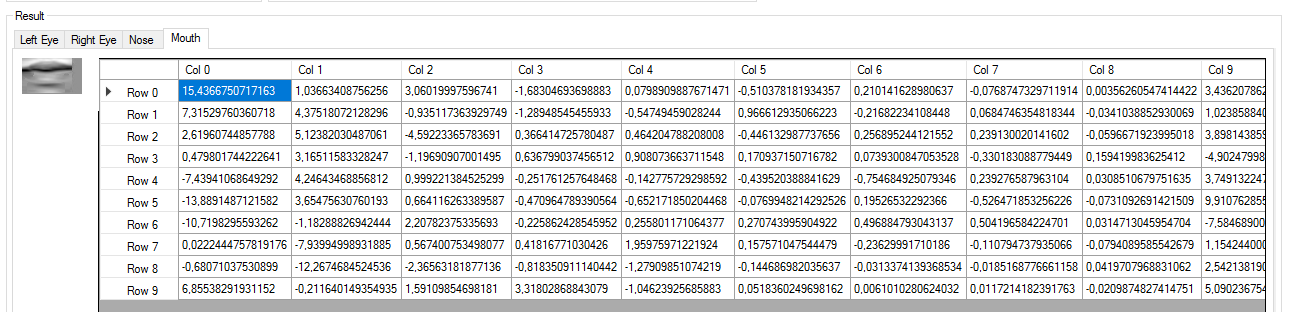
Gambar 5. 17 Nilai fitur mata kanan

Nilai ciri PCA dari hidung



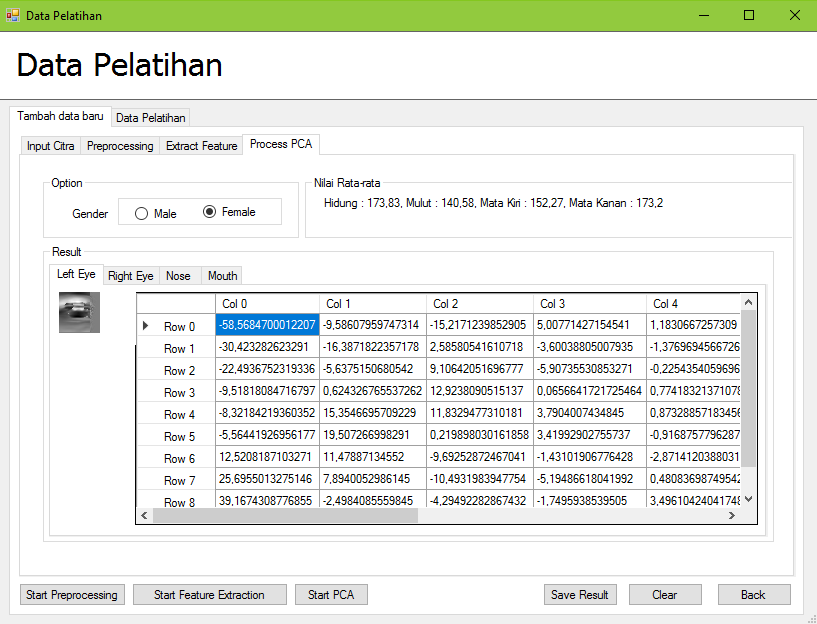
Gambar 5. 18 Nilai fitur hidung

Nilai ciri PCA dari mulut



Gambar 5. 19 Nilai fitur mulut

Berikut adalah nilai rata-rata untuk hasil PCA pada masing-masing fitur wajah :

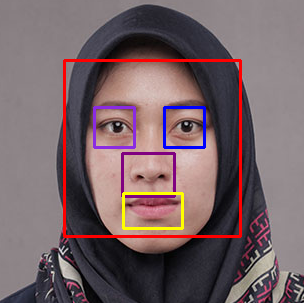


Gambar 5. 20 Nilai rata-rata fitur wajah

Proses dalam perhitungan metode ini memiliki beberapa tahap yaitu :

1. Mengambil Nilai *Grayscale*

Setelah fitur telah dideteksi maka diubah menjadi *Grayscale* kemudian area fitur yang didapat akan dipotong menjadi diambil nilai *Grayscale* dari potongan area fitur tersebut. Berikut adalah contoh potongan mata sebagai contoh perhitungan *ekstraksi* fitur menggunakan *PCA* dengan *library EmguCV :*



Gambar 5. 21 Inputan Image

1. Diubah transformasi menjadi 1 vektor

|  |
| --- |
| MATRIK T |
| 6 |
| 50 |
| 1 |
| 88 |
| 6 |
| 70 |
| 6 |
| 70 |
| 88 |
| 8 |
| 63 |
| 100 |

Matriks *Grayscale* ditransformasi menjadi matriks 1x1 kolam (Matriks T). kemudian matriks T tersebut digabungkan dengan kolom dari data *training.*

Tabel 5. 1 Nilai Grayscale diubah menjadi matriks T

|  |  |  |
| --- | --- | --- |
| Nilai Grayscale | | |
| 6 | 50 | 1 |
| 88 | 6 | 70 |
| 6 | 70 | 88 |
| 8 | 63 | 100 |

1. Digabungkan semua data latih yang ada

**M = jumlah tiap baris/banyaknya datatraining**

Tabel 5. 2 Penggabungan data training

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MATRIK T DENGAN TRAINING LAINNYA | | | | | TOTAL |
| 225 | 192 | 6 | 84 | 228 | 735 |
| 212 | 222 | 50 | 73 | 208 | 765 |
| 223 | 225 | 1 | 58 | 221 | 728 |
| 225 | 221 | 88 | 74 | 226 | 834 |
| 133 | 178 | 6 | 72 | 139 | 528 |
| 171 | 220 | 70 | 64 | 174 | 699 |
| 217 | 217 | 6 | 71 | 220 | 731 |
| 125 | 130 | 70 | 74 | 136 | 535 |
| 185 | 191 | 88 | 71 | 179 | 714 |
| 115 | 140 | 8 | 85 | 136 | 484 |
| 225 | 192 | 63 | 84 | 228 | 792 |
| 212 | 222 | 100 | 73 | 208 | 815 |

1. Mencari rata-rata

Tabel 5. 3 Rata-rata

|  |  |
| --- | --- |
| BARIS 1 | 147 |
| BARIS 2 | 153 |
| BARIS 3 | 145.6 |
| BARIS 4 | 166.8 |
| BARIS 5 | 105.6 |
| BARIS 6 | 139.8 |
| BARIS 7 | 146.2 |
| BARIS 8 | 107 |
| BARIS 9 | 142.8 |
| BARIS 10 | 96.8 |
| BARIS 11 | 77.4 |
| BARIS 12 | 105 |

1. Normalisasi matriks

**A = T – M**

Keterangan : A = Matriks Normalisasi

T = Matriks Tranformasi

M = Rata-rata

Tabel 5. 4 Normalisasi Matriks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MATRIK A | | | | |
| 78 | 45 | -141 | -63 | 81 |
| 59 | 69 | -103 | -80 | 55 |
| 77.4 | 79.4 | -144.6 | -87.6 | 75.4 |
| 58.2 | 54.2 | -78.8 | -92.8 | 59.2 |
| 27.4 | 72.4 | -99.6 | -33.6 | 33.4 |
| 31.2 | 80.2 | -69.8 | -75.8 | 34.2 |
| 70.8 | 70.8 | -140.2 | -75.2 | 73.8 |
| 18 | 23 | -37 | -33 | 29 |
| 42.2 | 48.2 | -54.8 | -71.8 | 36.2 |
| 18.2 | 43.2 | -88.8 | -11.8 | 39.2 |
| -6.4 | 2.6 | -14.4 | 2.6 | 15.6 |
| 15 | 18 | -5 | -32 | 4 |

1. Mencari nilai kovarian

Mencari nilai kovarian dengan persamaan.

**L = Aʹ × A**

Tabel 5. 5 Nilai kovarian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MATRIK KOVARIAN | | | | |
| L1 | L2 | L3 | L4 | L5 |
| 28381.88 | 29867.3 | -52264 | -34760 | 28774.9 |
| 29867.28 | 37763.7 | -59395 | -39475 | 31239.3 |
| -52263.72 | -59395 | 104547 | 62609.1 | -55497 |
| -34760.32 | -39475 | 62609.1 | 46598.5 | -34972 |
| 28774.88 | 31239.3 | -55497 | -34972 | 30454.9 |

1. Mencari *Eigenvalue* dan mencari *Eigenvector* (v)

*Eigenvalue* = Nilai ciri yang terdapat pada data *training* .

Tabel 5. 6 Nilai EigenValue

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MATRIK EIGENVALUE | | | | |
| D1 | D2 | D3 | D4 | D5 |
| 236829 | 0 | 0 | 0 | 0 |
| 0 | 7047.49 | 0 | 0 | 0 |
| 0 | 0 | 3470.21 | 0 | 0 |
| 0 | 0 | 0 | 398.944347 | 0 |
| 0 | 0 | 0 | 0 | 4.3E-12 |

*Eigenvector* = Nilai eigenvalue yang lebih besar dari 0

Tabel 5. 7 Nilai EigenVector

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MATRIK EIGENVECTOR | | | | |
| V1 | V2 | V3 | V4 | V5 |
| 1 | 1.22683E-76 | 4.95436E-77 | 9.46341E-77 | 2.17686E-78 |
| -3.70939E-77 | 1 | 7.5167E-72 | 2.48044E-72 | 1.71961E-72 |
| -8.3368E-77 | -1.56823E-70 | 1 | 3.69513E-71 | 1.91442E-71 |
| -2.61622E-77 | -2.71669E-71 | -1.18755E-71 | 1 | 3.36839E-69 |
| -1.04365E-76 | -4.64876E-71 | -2.66883E-71 | -4.56709E-69 | 1 |

1. Mencari *Eigenfeature*

***Eigenfeatur*e= A x *Eigenvector***

Tabel 5. 8 Nilai EigenFeature

|  |  |  |  |
| --- | --- | --- | --- |
| MATRIK EIGENFEATURE | | | |
| EigenF1 | EigenF2 | EigenF3 | EigenF4 |
| 78 | 45 | -141 | -63 |
| 59 | 69 | -103 | -80 |
| 77.4 | 79.4 | -144.6 | -87.6 |
| 58.2 | 54.2 | -78.8 | -92.8 |
| 27.4 | 72.4 | -99.6 | -33.6 |
| 31.2 | 80.2 | -69.8 | -75.8 |
| 70.8 | 70.8 | -140.2 | -75.2 |
| 18 | 23 | -37 | -33 |
| 42.2 | 48.2 | -54.8 | -71.8 |
| 18.2 | 43.2 | -88.8 | -11.8 |
| -6.4 | 2.6 | -14.4 | 2.6 |
| 15 | 18 | -5 | -32 |

1. Mencari nilai PCA *training*

***PCA training* = *Eigenfeature*ʹ x A**

Keterangan :

A = Matriks Normalisasi

Tabel 5. 9 Hasil PCA training

|  |  |  |  |
| --- | --- | --- | --- |
| PCA TRAINING | | | |
| 28381.88 | 29867.28 | -52263.7 | -34760.3 |
| 29867.28 | 37763.68 | -59395.3 | -39474.9 |
| -52263.72 | -59395.3 | 104546.7 | 62609.08 |
| -34760.32 | -39474.92 | 62609.08 | 46598.48 |

**5.2.2 Klasifikasi Jenis Kelamin**

Dari ekstraksi fitur diatas, maka akan dilakukan klasifikasi dengan *metode Naive Bayes.* [*Naive Bayes Classifier*](https://en.wikipedia.org/wiki/Naive_Bayes_classifier) merupakan salah satu metode *machine learning* yang memanfaatkan perhitungan probabilitas dan statistik. Berikut ini nilai pembanding dari citra data tes yang diambil dari rata-rata pada masing-masing fitur di data *training* :

Tabel 5. 10 Tabel rata-rata fitur training

|  |  |  |
| --- | --- | --- |
| Rata-Rata Fitur *Training* | | |
| Fitur | Laki-Laki | Perempuan |
| Mata Kiri | 140,91 | 150,829 |
| Mata Kanan | 138,621 | 155,667 |
| Hidung | 138,81 | 160,401 |
| Mulut | 129,493 | 148,512 |

Dari hasil rata-rata dari data *training* maka dilakukan perhitungan dengan *Naive Bayes.* Selanjutnya dilakukan perhitungan untuk pencocokan nilai untuk menghasilkan keluaran jenis kelamin yaitu laki-laki dan perempuan. Untuk hasil klasifikasiya adalah jika nilai perempuan lebih kecil dari nilai laki-laki maka citra wajah tersebut adalah perempuan. Berikut ini adalah alur penerapan klasifikasi dengan *Naive Bayes* dengan persamaan berikut :

..................................................................................... (5.1)

Penjelasan uraian rumus sesuai dengan sistem :

: nilai probabilitas hipotesis yang akan dicari classnya terhadap data latih

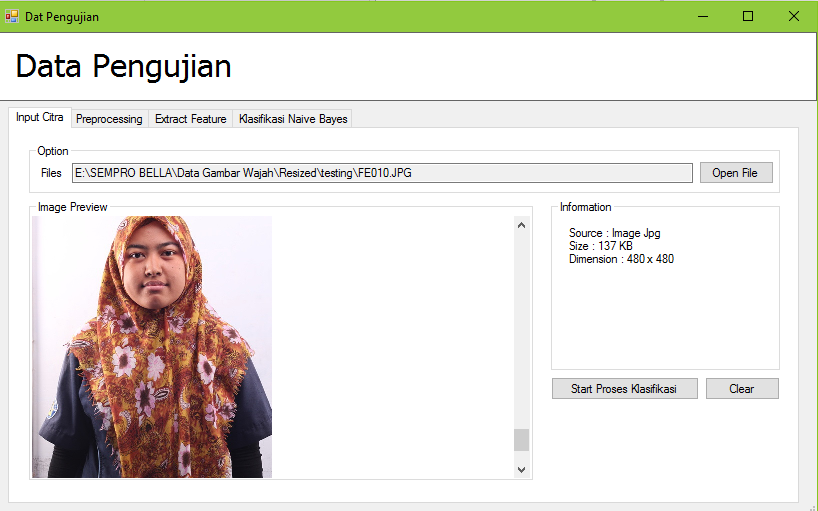
: nilai probabilitas hipotesis terhadap data latih

: nilai probabilitas hipotesis

: nilai probabilitas data uji

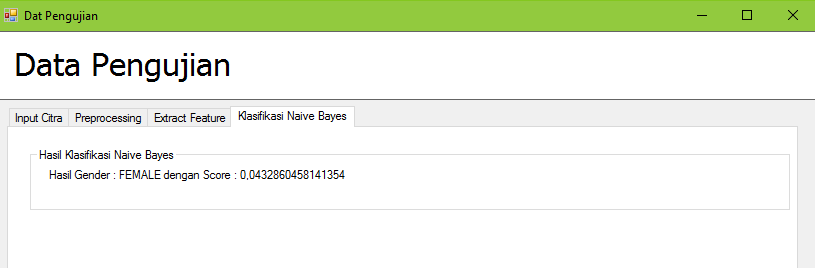
Dari hasil persamaan diatas, maka diimplementasikan pada aplikasi sebagai acuan untuk klasifikasi jenis kelamin . Sistem akan mengecek setiap nilai fitur dari data tes sampai nilai tersebut memiliki hasil kecocokan probabilitas.

Berikut adalah contoh hasil yang didapatkan pada citra uji data perempuan dan laki-laki dengan metode *Naive Bayes* :



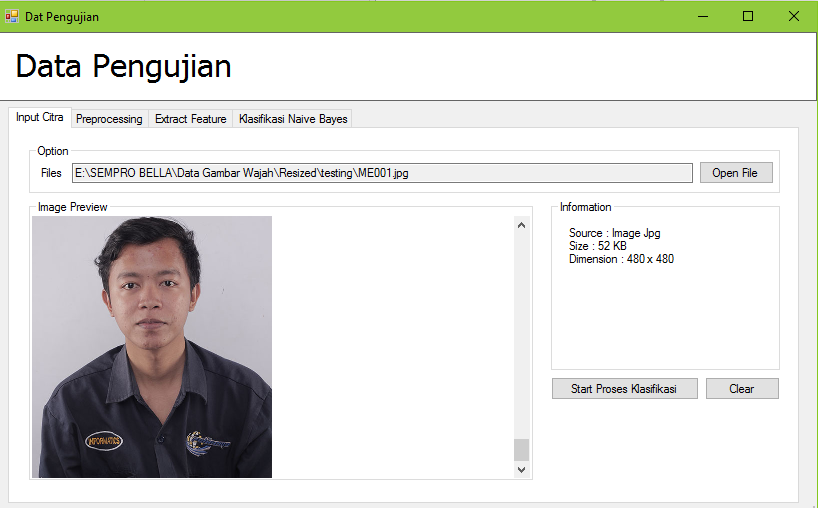
Gambar 5. 22 Citra Uji 1

Dari inputan diatas maka diperoleh hasil klasifikasi berupa :



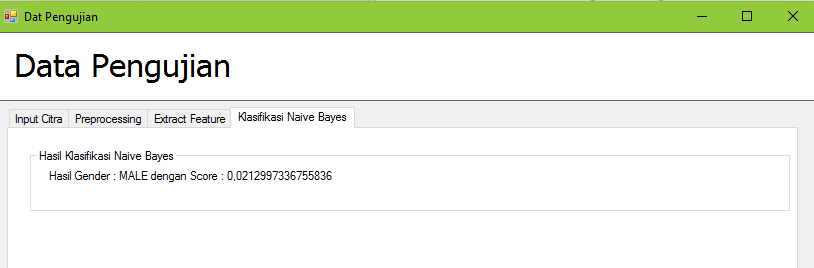
Gambar 5. 23 Hasil klasifikasi citra uji 1

Hasil yang didapatkan yaitu perempuan dengan nilai probabilitas 0,043 yang artinya kecocokan masing-masing fitur tersebut masuk kedalam kelompok perempuan.



Gambar 5. 24 Citra uji 2

Dari inputan data diatas maka diperoleh hasil klasifikasi berupa :



Gambar 5. 25 Hasil klasifikasi citra uji 2

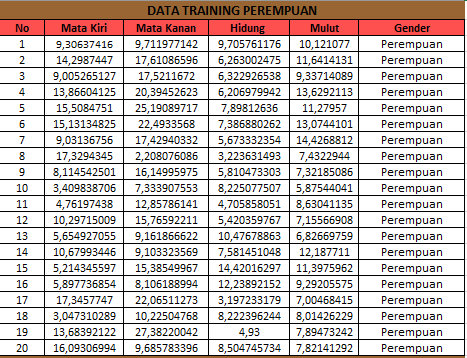
Hasil yang didapatkan adalah laki-laki dengan nilai probabilitas 0,021 yang artinya kecocokan masing-masing fitur tersebut masuk kedalam kelompok laki-laki.

Dari sistem tersebut maka dapat disimpulkan bahwa Naive bayes memiliki 2 parameter diantaranya adalah :

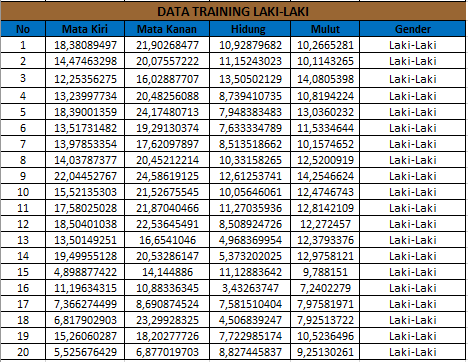
1. *Naive Bayes* melakukan pencocokan tiap-tiap fitur dengan fitur data yang di uji
2. Setelah fitur ditemukan maka dilakukan klasifikasi fitur dengan *class* pada data *training*

Tahapan penerapan metode *Naive Bayes* diantaranya sebagai berikut :

1. Tahap 1 🡪 Menghitung jumlah *class* yang ada

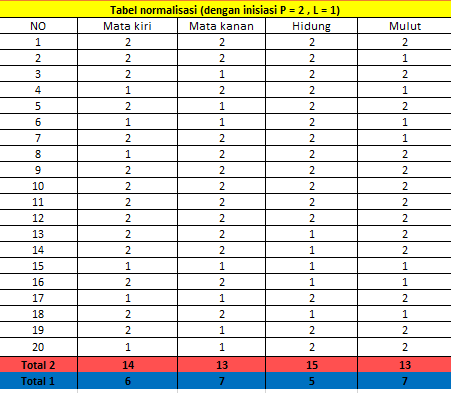


Gambar 5. 26 Perhitungan excel training perempuan



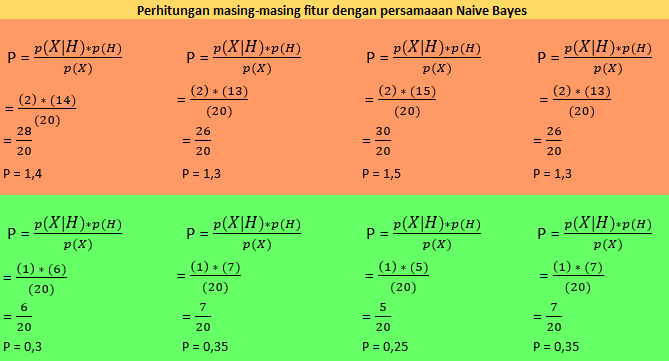
Gambar 5. 27 Perhitungan excel training laki-laki

1. Tahap 2 🡪 Menghitung jumlah kasus yang sama dengan *class* yang sama



Gambar 5. 28 Normalisasi nilai

1. Tahap 3 🡪 Mengkalikan semua hasil *variable*



Gambar 5. 29 Perhitungan nilai keempat fitur

1. Tahap 4 🡪 Membandingkan hasil nilai probabilitas yang tertinggi dari data jenis kelamin yang ada

Total Nilai probabilitas Perempuan



Total Nilai Probabilitas Laki-Laki

Gambar 5. 30 Nilai probabilitas dan hasil klasifikasi

Jadi dari kedua hasil tersebut , maka hasil klasifikasi menggunakan metode *Naive Bayes* mendapatkan nilai probabilitas tertinggi pada jenis kelamin perempuan dengan total *score* 1,375 .

# **BAB VI. PENGUJIAN DAN PEMBAHASAN**

## **Pengujian Proses**

Setelah melakukan proses *training* data, untuk melihat apakah aplikasi ini telah berhasil seperti yang diinginkan akan dilakukan pengujian. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan aplikasi tersebut.

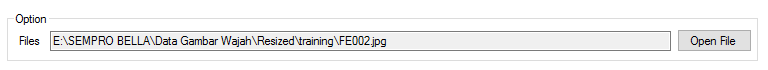
Pengujian dilakukan berdasarkan spesifikasi sistem dan pengujian performansi. Pengujian spesifikasi sistem yang dilakukan meliputi pengujian kesesuaian proses, pengujian kesesuaian data dan pengujian tingkat akurasi citra. Pengujian performansi dilakukan dengan serangkaian percobaan-percobaan dalam kondisi tertentu yang dapat mempengaruhi hasil dari aplikasi ini.

* + 1. **Pengujian performansi**

Dalam proses pengujian performansi ini diantaranya meliputi pengujian pendeteksian input wajah, pengujian deteksi wajah serta fitur-fitur wajah , pengujian ekstraksi fitur wajah serta pengujian tingkat akurasi klasifikasi.

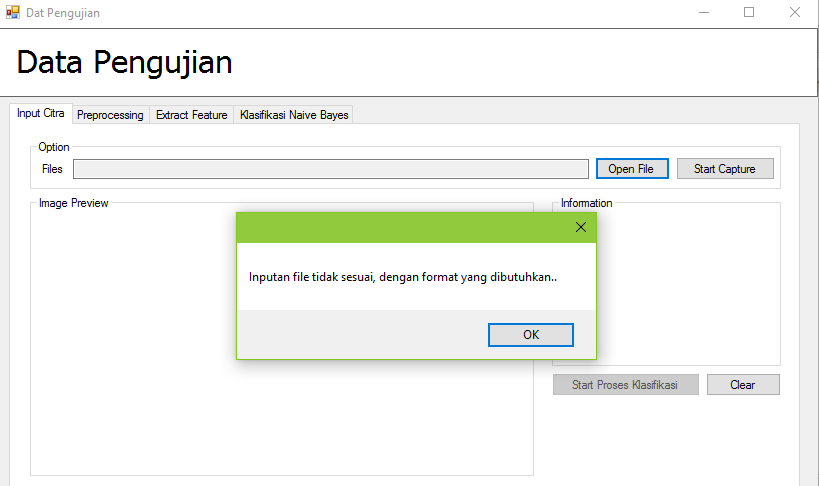
1. Pengujian *input image from drive*.

Pada proses *input image* ini terdapat 1 *button* yang yaitu *open file* *(from drive)* seperti pada gambar berikut :



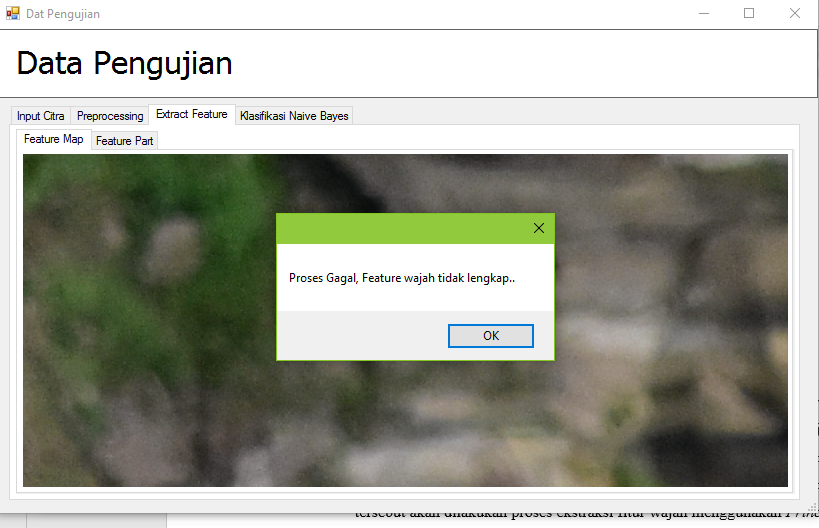
Gambar 6. 1 Button pengambilan gambar

Jika file yang dipilih tidak sesuai maka akan keluar *text box* seperti gambar berikut :



Gambar 6. 2 Pesan error inputan citra

Berikut adalah pesan proses gagal apabila input tidak sesuai atau sistem tidak menemukan fitur pada citra wajah



Gambar 6. 3 Pesan fitur tidak terdeteksi

1. Pengujian dan analisa ketepatan deteksi wajah serta fitur-fitur wajah.

Selanjutnya adalah pengujian deteksi wajah serta fitur-fitur dari wajah diantaranya adalah mata hidung dan mulut. Pada aplikasi ini klasifikasi jenis kelamin pada citra wajah dibutuhkan pengambilan fitur-fitur wajah untuk diambil nilai dari masing-masing fitur wajah. Maka berdasarkan nilai-nilai tersebut akan dilakukan proses ekstraksi fitur wajah menggunakan *Principal Component Analysis*. Berikut adalah tabel pedeteksian fitur wajah terhadap beberapa percobaan dengan jumlah data *training* yang berbeda:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Jumlah Data Training | | | | | | | | | | | | | | |
| 20 | | | | | 30 | | | | | 40 | | | | |
| Wajah | Mata Kiri | Mata Kanan | Hidung | Mulut | Wajah | Mata Kiri | Mata Kanan | Hidung | Mulut | Wajah | Mata Kiri | Mata Kanan | Hidung | Mulut |
| 1 |  | √ | √ | √ | √ |  | √ | √ | √ | √ |  | √ | √ | √ | √ |
| 2 |  | √ | √ | √ | √ |  | √ | √ | √ | √ |  | √ | √ | √ | √ |
| 3 |  | √ | √ | √ | √ |  | √ | √ | √ | √ |  | √ | √ | √ | √ |
| 4 |  | √ | √ | √ | √ |  | √ | √ | √ | √ |  | √ | √ | √ | √ |
| 5 |  | √ | √ | √ | √ |  | √ | √ | √ | √ |  | √ | √ | √ | √ |
| Tingkat  Akurasi | 100 % | | | | | 100 % | | | | | 100 % | | | | |

Tabel 6. 1 Pengujian deteksi wajah

Keterangan :

Tanda “√” menunjukkan fitur wajah terdeteksi **tepat** oleh sistem

Tanda “√” menunjukkan fitur tidak terdeteksi **tidak tepat** oleh sistem

1. Pengujian dan analisa tingkat akurasi untuk klasifikasi jenis kelamin .

Pengujian klasifikasi jenis kelamin berdasarkan citra wajah dilakukan dengan metode *Naive Bayes* yang bertujuan untuk mencocokkan nilai *input image* yang menghasilkan sebuah jenis kelamin artinya apakah laki-laki atau perempuan. Kriteria pengujian adalah inputan citra wajah yang akan dilakukan ekstraksi fitur yang menghasilkan keluaran jenis kelamin. Berikut ini tabel hasil akurasi citra wajah berdasarkan ketepatan fitur-fitur wajah terhadap beberapa percobaan dengan jumlah data *training* yang berbeda:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Jumlah Data *Training* | | | | | | | | | | | |
| 20 | | | | 30 | | | | 40 | | | |
| Hasil | | Status | | Hasil | | Status | | Hasil | | Status | |
| *Female* | *Male* | *True* | *False* | *Female* | *Male* | *True* | *False* | *Female* | *Male* | *True* | *False* |
| 1 | **-** | √ | **-** | √ | **-** | √ | **-** | √ | **-** | √ | **-** | √ |
| 2 | √ | **-** | √ | **-** | √ | **-** | √ | **-** | √ | **-** | √ | **-** |
| 3 | **-** | √ | **-** | √ | **-** | √ | **-** | √ | **-** | √ | **-** | √ |
| 4 | **-** | √ | **-** | √ | **-** | √ | **-** | √ | **-** | √ | **-** | √ |
| 5 | √ | **-** | √ | **-** | **-** | √ | **-** | √ | √ | **-** | √ | **-** |
| 6 | √ | **-** | **-** | √ | √ | **-** | **-** | √ | **-** | √ | √ | **-** |
| 7 | √ | **-** | **-** | √ | **-** | √ | √ | **-** | **-** | √ | √ | **-** |
| 8 | √ | **-** | **-** | √ | **-** | √ | √ | **-** | **-** | √ | √ | **-** |
| 9 | **-** | √ | √ | **-** | **-** | √ | √ | **-** | **-** | √ | √ | **-** |
| 10 | √ | **-** | **-** | √ | **-** | √ | √ | **-** | **-** | √ | √ | **-** |
| 11 | **-** | √ | √ | **-** | **-** | √ | √ | **-** | **-** | √ | √ | **-** |
| 12 | **-** | √ | √ | **-** | **-** | √ | √ | **-** | **-** | √ | √ | **-** |
| 13 | **-** | √ | √ | **-** | √ | **-** | **-** | √ | **-** | √ | √ | **-** |
| 14 | √ | **-** | **-** | √ | **-** | √ | √ | **-** | **-** | √ | √ | **-** |
| 15 | √ | **-** | **-** | √ | **-** | √ | √ | **-** | **-** | √ | √ | **-** |
| 16 | **-** | √ | √ | **-** | √ | **-** | **-** | √ | **-** | √ | √ | **-** |
| 17 | **-** | √ | √ | **-** | **-** | √ | √ | **-** | **-** | √ | √ | **-** |
| 18 | **-** | √ | √ | **-** | √ | **-** | **-** | √ | **-** | √ | √ | **-** |
| 19 | **-** | √ | √ | **-** | **-** | √ | √ | **-** | **-** | √ | √ | **-** |
| 20 | √ | **-** | **-** | √ | **-** | √ | √ | **-** | **-** | √ | √ | **-** |
| 21 | - | √ | √ | **-** | **-** | √ | √ | **-** | **-** | √ | √ | **-** |
| Akurasi | 40 % | | | | 60 % | | | | 80 % | | | |

Tabel 6. 2 Pengujian tingkat akurasi klasifikasi

Keterangan :

Tanda “√” menunjukkan hasil testing **sesuai** dengan data training

Tanda “√” menunjukkan hasil testing **tidak sesuai** dengan data training

* + 1. **Pengujian Blackbox**

Untuk tahap pengujian sistem menggunakan metode *blackbox*. Metode ini memungkinkan adanya pengembangan untuk melatih seluruh fungsi pada sistem. Metode ini digunakan untuk menemukan kesalahan pada saat aplikasi berjalan. Berikut *blackbox* dari pengujian sistem :

Tabel 6. 3 Pengujian Blackbox

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Proses** | **Hasil** | **Keterangan** |
| ***Form Training*** | Melakukan *input image from drive* | Menampilkan objek yang di tangkap pada *Image Box.* Jika terdapat objek wajah maka disimpan | Sukses |
| Melakukan *preprocessing* yaitu *resize* dan *grayscale* | Menampilkan hasil resize ukuran serta hasil *grayscale* pada *image* | Sukses |
| Melakukan deteksi wajah serta *cropping* fitur | Menampilkan hasil deteksi wajah serta *cropping* fitur | Sukses |
| Melakukan Ekstraksi fitur dengan metode *(Principal Component Analysis) PCA* serta memilih *gender* | Menampilkan tabel nilai PCA serta rata-rata pada setiap masing-masing fitur wajah | Sukses |
| Melakukan penyimpanan data *training* | Pemberitahuan data berhasil disimpan | Sukses |
| ***Form Testing*** | Melakukan *input image from drive* | Menampilkan objek yang di tangkap pada *Image Box.* Jika terdapat objek wajah maka disimpan | Sukses |
| Melakukan proses klasifikasi jenis kelamin | Menampilkan hasil *preprocessing ,* deteksi wajah, ekstraksi fitur serta klasifikasi otomatis dari data *training* jadi dapat menampilkan hasil jenis kelamin serta nilai probabilitas | Sukses |

Pada tabel dapat dilihat hasil pengujian sistem menggunakan *blackbox*. Berdasrkan dari hasil pengujian sistem menggunakan *blackbox,* maka dapat ditarik kesimpulan bahwa aplikasi klasifikasi jenis kelamin pada citra wajah menggunakan metode *Naive Bayes* berjalan sesuai harapan.

## **Analisa Hasil Penelitian**

Berdasarkan jumlah data latih yang ada yaitu sebanyak 40 data dengan rincian 20 data perempuan dan 20 data laki-laki, maka dilakukan beberapa pengujian dengan jumlah data uji sebanyak 21 data yang diantaranya terdapat pengujian untuk ketepatan deteksi wajah dan fitur dengan hasil akurasi sebesar 100 % ,artinya 40 data latih menunjukkan bahwa fitur wajah terdeteksi tepat oleh sistem. Pengujian selanjutnya yaitu pengujian untuk akurasi klasifikasi jenis kelamin dengan menggunakan 20 data latih yang menghasilkan akurasi sebesar 40 % , lalu ditambahkan menjadi 30 data dengan hasil akurasi 60 % , dan ditambahkan data latih terakhir menjadi 40 data yang menghasilkan akurasi pengujian sebesar 80 %.

Sedangkan untuk analisa pengujian deteksi wajah menggunakan kamera *handphone* dan berbagai macam gaya atau ekspresi dapat disimpulkan bahwa sistem akan mengeksekusi ketika fitur telah didapatkan yaitu mata kiri, mata kanan,hidung, dan mulut. Namun hasil pengujian tersebut menghasilkan beberapa fitur wajah tidak terdeteksi ,jadi untuk hasil klasifikasi jenis kelamin menjadi kurang akurat. Oleh sebab itu diperlukan data set yang sempurna sehingga tingkat akurasi klasifikasi jenis kelamin dengan metode Naive Bayes tidak akan menurun.

Jadi, dari beberapa pengujian yang telah berhasil dilakukan maka dapat diambil sebuah analisa yaitu semakin banyak data *training* atau latih maka semakin bagus pula hasil yang didapatkan pada pengujian serta hasil dari tingkat akurasi pada klasifikasi jenis kelamin dengan Metode *Naive Bayes* .

# **BAB VII. KESIMPULAN**

## **Kesimpulan**

Dari hasil penelitian yang telah dilakukan, maka dapat disimpulkan :

1. Proses deteksi wajah menggunakan metode *Haar Cascade Clasifier* tergolong baik. Hal ini dapat dilihat dari hasil ketepatan pendeteksian pada pengujian dengan hasil sebesar 100 % .
2. Tingkat ketepatan deteksi fitur wajah sangat berpengaruh pada klasifikasi jenis kelamin.
3. Proses ekstraksi menggunakan metode *Principal Component Analysis* sangat berpengaruh pada fitur yang diklasifikasikan.
4. Nilai ekstraksi fitur tergantung pada kualitas gambar
5. Proses klasifikasi dengan metode *Naive Bayes* hasilnya cukup akurat untuk mengklasifikasikan jenis kelamin berdasarkan nilai probabilitas yang didapatkan pada setiap *gender* laki-laki dan perempuan dengan hasil akurasi 80 %.

## **Saran**

Berdasarkan penelitian ini, ada beberapa hal yang disarankan yaitu :

1. Untuk ekstraksi fitur dapat ditambahkan jumlah *data set*, karena semakin banyak data yang dimiliki maka tingkat keakuratannya semakin tinggi.
2. Pada proses training dapat ditambahkan fungsi multiple image select
3. Pada proses klasifikasi dapat digunakan dengan metode *jaringan syaraf tiruan* dan *SVM* supaya didapatkan akurasi yang lebih baik serta dikembangkan menjadi aplikasi berbasis *mobile.*

# **DAFTAR PUSTAKA**

[1] Choirina, Priska.2015. Deteksi jenis berdasarkan citra wajah jarak jauh dengan metode Haar Cascade Classifier Studi Kasus : Politeknik Negeri Malang

[2] Suwandari, Rizky.2015. Pengenalan Ekspresi Wajah Berdasarkan Ekstrasi Fitur menggunakan metode *Principal Component Analysis* Studi Kasus : Politeknik Negeri Malang

[3] Rosiani, Ulla D, dkk.2016. Jurnal pendeteksian jenis kelamin pada gambarwajah 2D dengan ekstrasi fitur berbasis Geometri. Jurnal Jurusan Teknologi Informasi Politeknik Negeri Malang

[4] Kholistianingsih, 2012, Identifikasi *Wajah Menggunakan Principal Component Analysis Dengan Penambahan Fitur-Fitur Geometris*. Purwokerto: Universitas wijayakusuma.

[5] Pramana, Aditya Eka. 2011. *Perangkat Lunak Untuk Mengendalikan Pointer Dengan Mata Mengunakan Metode Haar Cascade Dan Eye Tracking*. Malang: Universitas Brawijaya.

[6] Modak, Santanu, dkk.2014. A Comparative study of Classifiers’ Performance for Gender Classification. India : University of Burdwan

[7] Algoritma Principal Component Analysis (PCA), 2013 [Online] Tersedia: http://accord-framework.net/docs/html/T\_Accord\_Statistics\_Analysis\_Principal ComponentAnalysis.html

# **LAMPIRAN-LAMPIRAN**

Lampiran 1. Pengujian deteksi wajah berdasarkan data training menggunakan algoritma Haar Cascade Classifier

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Wajah** | | **Mata Kiri** | | **Mata Kanan** | | **Hidung** | | **Mulut** | | **No** | | **Wajah** | | **Mata Kiri** | | **Mata Kanan** | | **Hidung** | | **Mulut** | |
| **1** |  | | **√** | | **√** | | **√** | | **√** | | **12** | |  | | **√** | | **√** | | **√** | | **√** | |
| **2** |  | | **√** | | **√** | | **√** | | **√** | | **13** | |  | | **√** | | **√** | | **√** | | **√** | |
| **3** |  | | **√** | | **√** | | **√** | | **√** | | **14** | |  | | **√** | | **√** | | **√** | | **√** | |
| **4** | |  | | **√** | | **√** | | **√** | | **√** | | **15** | |  | | **√** | | **√** | | **√** | | **√** | |
| **5** | |  | | **√** | | **√** | | **√** | | **√** | | **16** | |  | | **√** | | **√** | | **√** | | **√** | |
| **6** | |  | | **√** | | **√** | | **√** | | **√** | | **17** | |  | | **√** | | **√** | | **√** | | **√** | |
| **7** | |  | | **√** | | **√** | | **√** | | **√** | | **18** | |  | | **√** | | **√** | | **√** | | **√** | |
| **8** | |  | | **√** | | **√** | | **√** | | **√** | | **19** | |  | | **√** | | **√** | | **√** | | **√** | |
| **9** | |  | | **√** | | **√** | | **√** | | **√** | | **20** | |  | | **√** | | **√** | | **√** | | **√** | |
| **10** | |  | | **√** | | **√** | | **√** | | **√** | | **21** | |  | | **√** | | **√** | | **√** | | **√** | |
| **11** | |  | | **√** | | **√** | | **√** | | **√** | |

Lampiran 2 . Pengujian klasifikasi berdasarkan data training menggunakan metode Naive Bayes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Wajah ke - | Fitur Wajah | | | | Jenis Kelamin  ( Naive Bayes ) | Jenis Kelamin  ( Benar ) |
| Mata kiri | Mata kanan | Hidung | Mulut |
| 1 | **√** | **√** | **√** | **√** | Perempuan | Perempuan |
| 2 | **√** | **√** | **√** | **√** | Perempuan | Perempuan |
| 3 | **√** | **√** | **√** | **√** | Laki-Laki | Perempuan |
| 4 | **√** | **√** | **√** | **√** | Laki-Laki | Perempuan |
| 5 | **√** | **√** | **√** | **√** | Laki-Laki | Perempuan |
| 6 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 7 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 8 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 9 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 10 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 11 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 12 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 13 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 14 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 15 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 16 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 17 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 18 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 19 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 20 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |
| 21 | **√** | **√** | **√** | **√** | Laki-Laki | Laki-Laki |

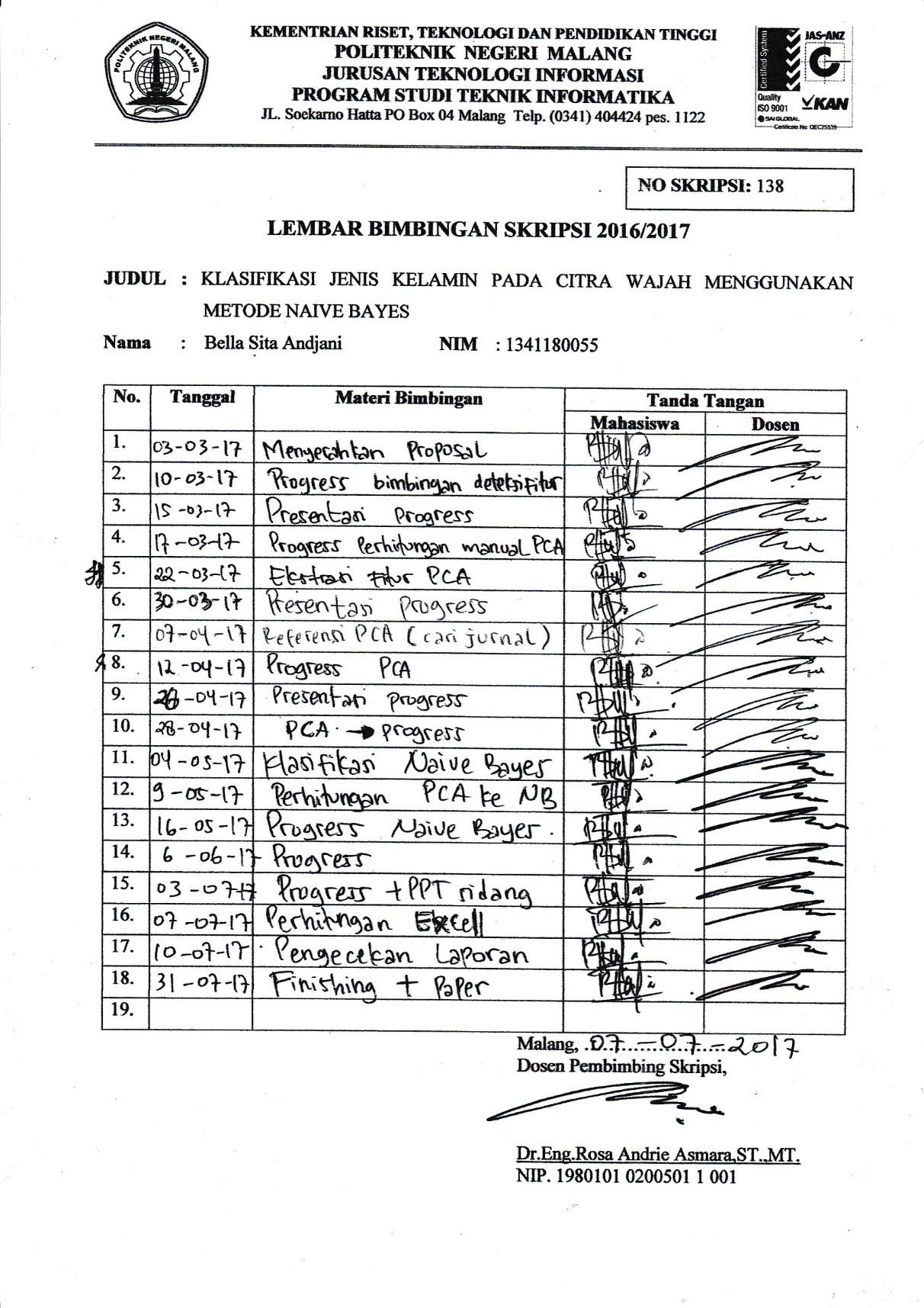
Lampiran 3. Pengujian deteksi wajah berdasarkan data training menggunakan kamera handphone

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Wajah** | **Mata Kiri** | **Mata Kanan** | **Hidung** | **Mulut** | **Hasil Klasifikasi** |
| **1** |  | **√** | **√** | **√** | **√** | **Laki-Laki** |
| **2** |  | **√** | **√** | **√** | **√** | **Perempuan** |
| **3** |  | **√** | **√** | **√** | **√** | **Laki-laki** |
| **4** |  | **√** | **√** | **√** | **√** | **Perempuan** |
| **5** |  | **√** | **√** | **√** | **√** | **Laki-laki** |
| **6** |  | **√** | **√** | **√** | **√** | **Perempuan** |
| **7** |  | **√** | **√** | **√** | **√** | **Perempuan** |
| **8** |  | **√** | **√** | **√** | **√** | **Perempuan** |
| **9** |  | **√** | **√** | **√** | **√** | **Laki-laki** |
| **10** |  | **√** | **√** | **√** | **√** | **Perempuan** |

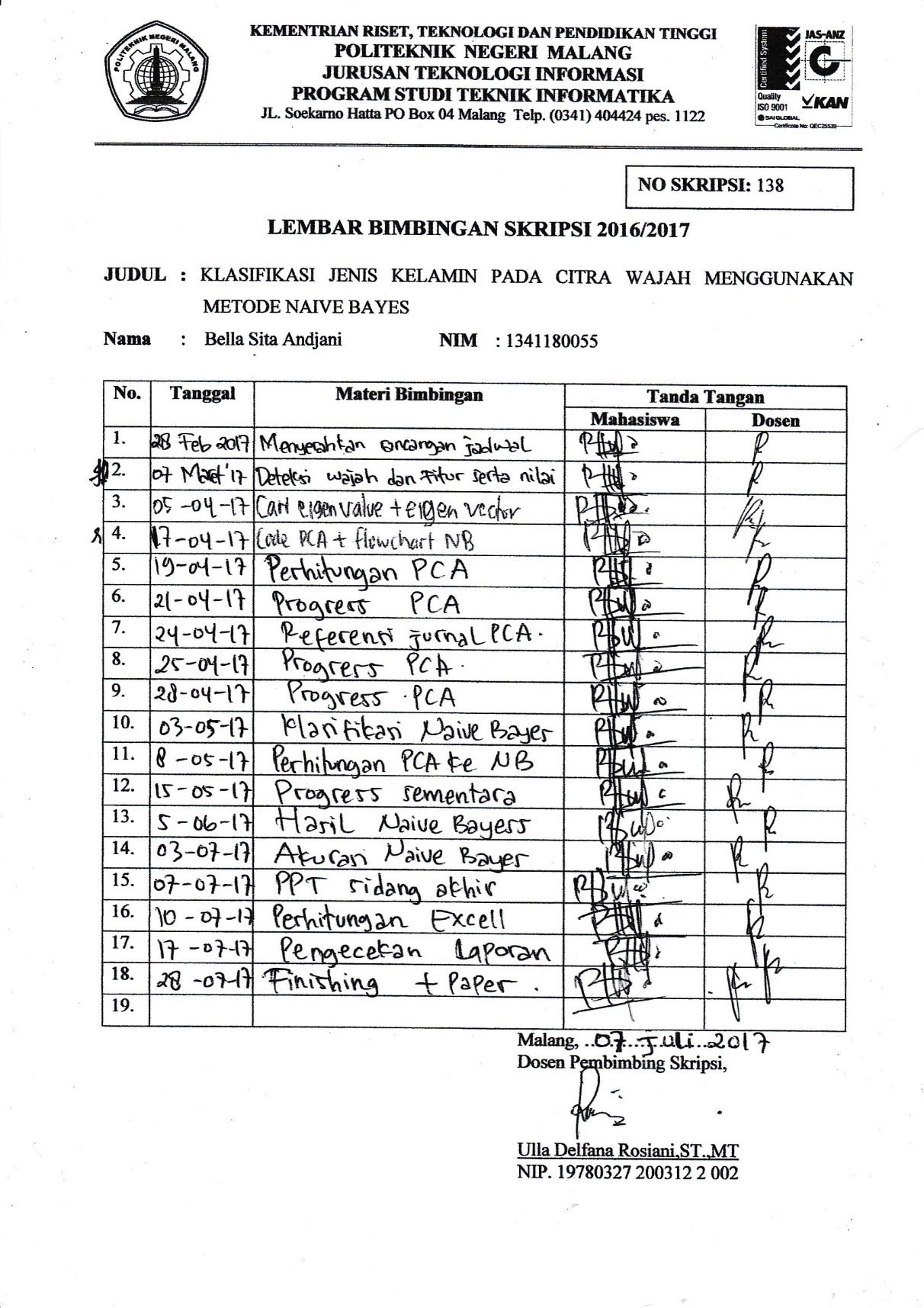
Lampiran 4. Pengujian deteksi wajah berdasarkan data training dengan berbagai gaya ekspresi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Wajah** | **Mata Kiri** | **Mata Kanan** | **Hidung** | **Mulut** | **Hasil Klasifikasi** |
| **1** |  | **√** | **√** | **√** | **√** | **Perempuan** |
| **2** |  | **√** | **√** | **√** | **√** | **Perempuan** |
| **3** |  | **√** | **√** | **√** | **√** | **Perempuan** |
| **4** |  | **√** | **√** | **√** | **√** | **Perempuan** |
| **5** |  | **-** | **√** | **-** | **√** | **-** |
| **6** |  | **√** | **√** | **√** | **√** | **Laki-laki** |
| **7** |  | **-** | **-** | **-** | **√** | **-** |
| **8** |  | **√** | **√** | **√** | **√** | **Laki-laki** |
| **9** |  | **√** | **√** | **√** | **√** | **Laki-laki** |
| **10** |  | **√** | **√** | **-** | **-** | **Laki-laki** |

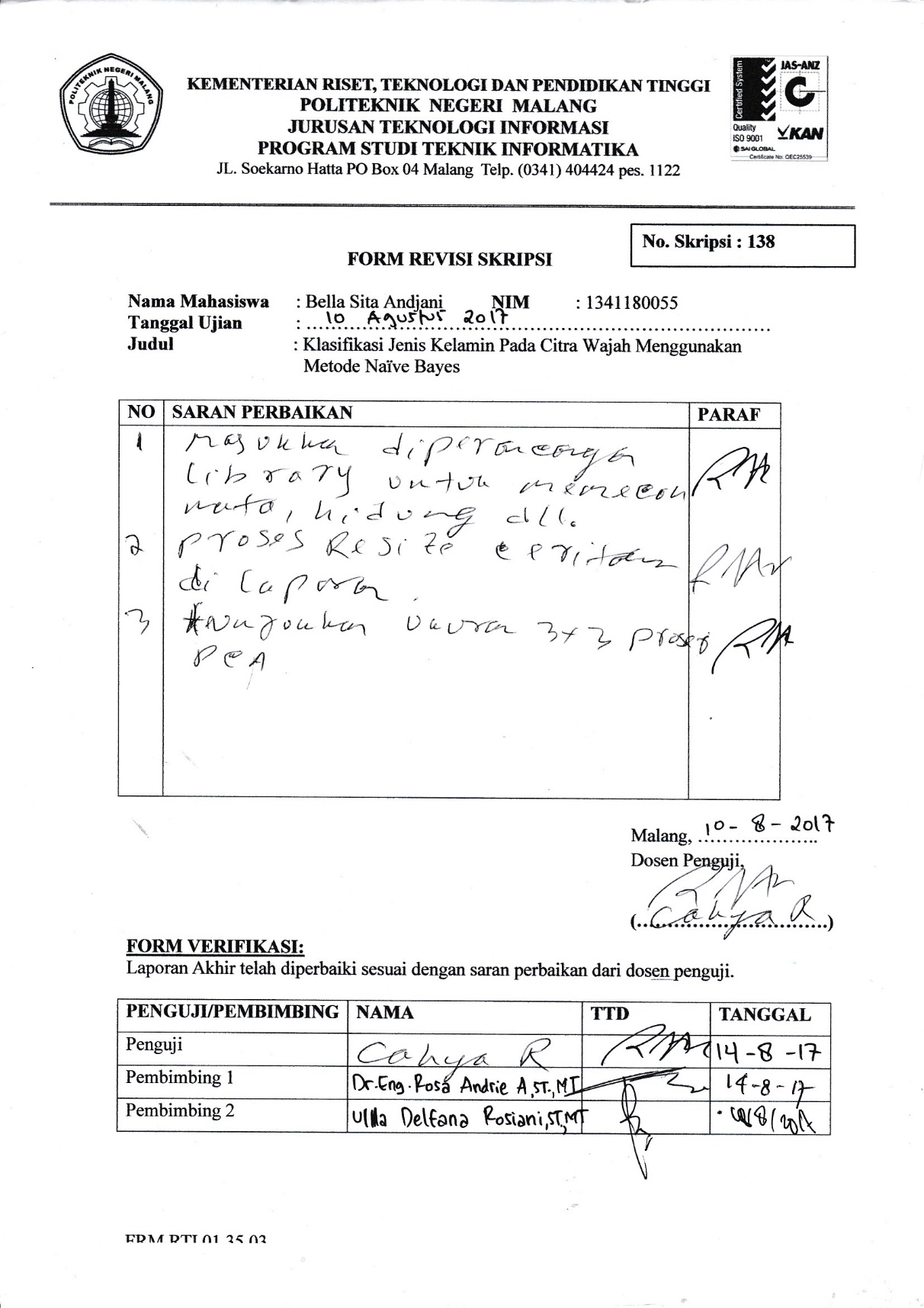
Lampiran 5. Lembar bimbingan skripsi ( Pembimbing 1 )



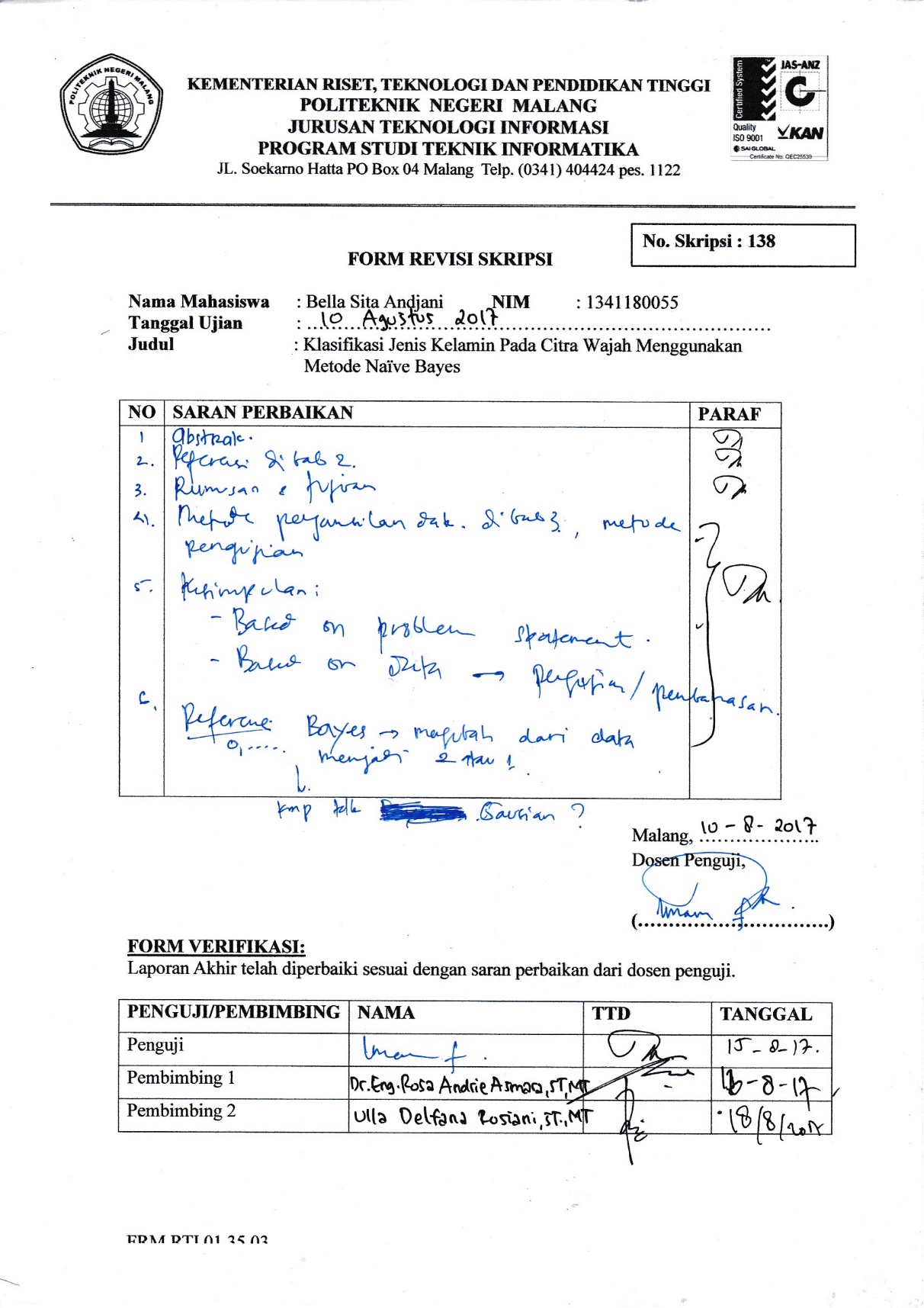
Lampiran 6. Lembar bimbingan skripsi ( Pembimbing 2 )



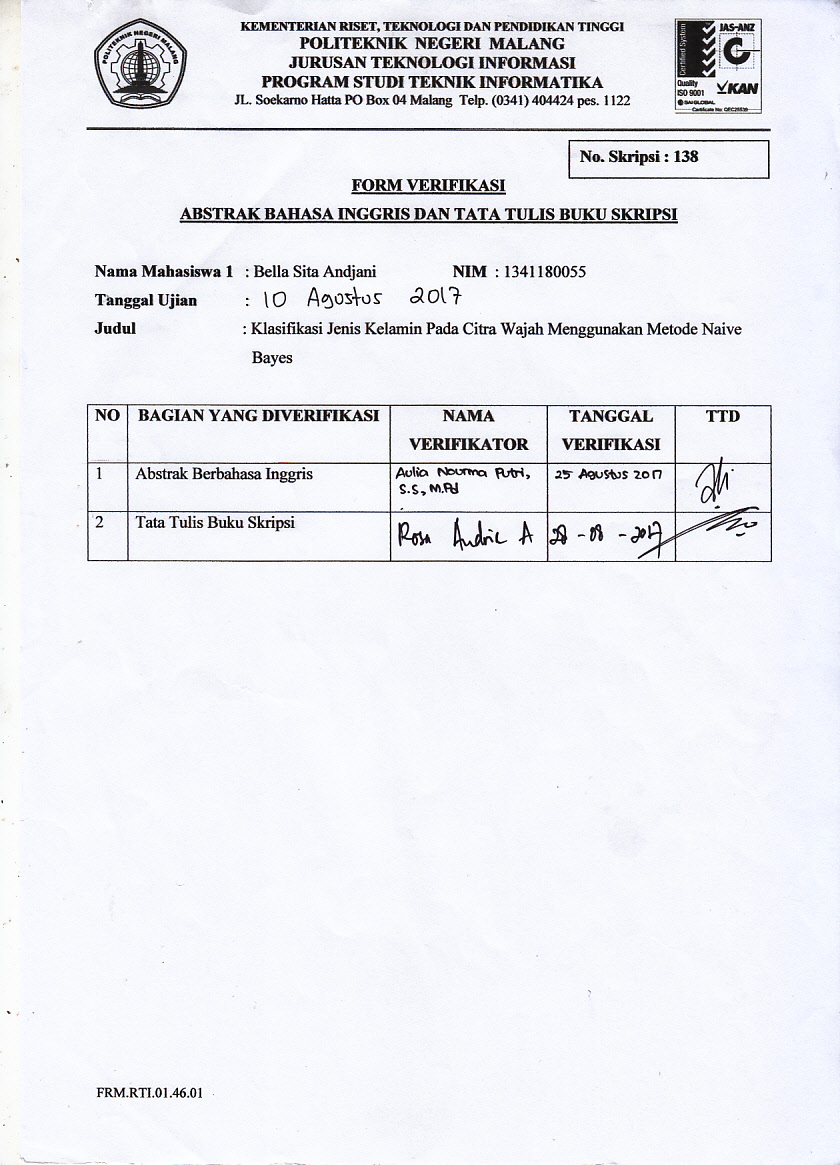
Lampiran 7. Lembar revisi skripsi ( Penguji 1 )



Lampiran 8. Lembar revisi skripsi ( Penguji 2 )



Lampiran 9. Lembar verifikasi skripsi

****

Lampiran 10. Profil Penulis



|  |  |  |
| --- | --- | --- |
| Nama Lengkap | : | Bella Sita Andjani |
| Tempat Tanggal Lahir | : | Sidoarjo, 16 Juni 1994 |
| Jenis Kelamin | : | Perempuan |
| Alamat | : | Desa Kalitengah Selatan RT 4 / RW 3 Tanggulangin Sidoarjo |
| No. Telepon | : | 081703535899 |
| Email | : | [bellasitaandjani@gmail.com](mailto:bellasitaandjani@gmail.com) |

**Riwayat Pendidikan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama** | **Tempat** | **Tahun** | **Pendidikan** |
| SDN Kalitengah 1 Sidoarjo | Sidoarjo,  Jawa Timur | 2000 – 2006 | - |
| SMP Negeri 2 Candi | Sidoarjo,  Jawa Timur | 2006 – 2009 | - |
| SMK Telekomunikasi Darul ‘Ulum Jombang | Jombang, Jawa Timur | 2009 – 2012 | Multimedia |
| Institute Teknologi Sepuluh Nopember Surabaya | Surabaya, Jawa Timur | 2012 – 2013 | Diploma I Teknik Informatika |
| Politeknik Negeri Malang | Malang, Jawa Timur | 2013 - 2017 | D4 Teknik Informatika |