BAB 2
LANDASAN TEORI

2.1. Pembahasan Mengenai Tata Rias Wajah

Berikut ini adalah pembahasan mengenai tata rias wajah yang merupakan bagian dari sistem yang akan dibangun, yaitu sistem rekomendasi warna *foundation* untuk tata rias wajah menggunakan pengolahan citra.

2.1.1 Pengertian Tata Rias Wajah

Tata rias wajah adalah kegiatan mengubah penampilan dari bentuk asli sebenarnya dengan bantuan bahan dan alat kosmetik. Istilah *make up* lebih sering ditujukan kepada pengubahan bentuk wajah, meskipun sebenarnya seluruh tubuh bisa di hias. Fungsi tata rias adalah untuk menyempurnakan penampilan wajah, menggambarkan karakter tokoh, dan menambah aspek dramatik.

1. **Tata Rias Korektif**

   Merupakan suatu bentuk tata rias wajah yang bersifat menyempurnakan dan mengubah penampilan fisik yang dinilai kurang sempurna. Tata rias wajah korektif merupakan jenis tata rias wajah yang paling sering dilakukan oleh para wanita.

2. **Tata Rias Fantasi**

   Tata rias fantasi adalah suatu seni tata rias yang bertujuan untuk membentuk kesan wajah model menjadi wujud khayalan yang diangan-angankan, tetapi segera dikenali oleh yang melihatnya. Tata rias wajah fantasi data juga merupakan perwujudan khayalan seorang ahli kecantikan yang ingin melukiskan angan-angan berupa bunga atau hewan dengan merias wajah, melukis di badan, menata rambut, dan kelengkapannya.

3. Tata Rias Karakter

Tata rias yang mengubah penampilan wajah seseorang dalam hal umur, bangsa, sidat, dan ciri-ciri khusus yang melekat pada tokoh. Dari abad ke abad, dari tahun ke tahun wanita semakin peduli pada penampilan fisik. Era Victoria yang menjadi lambang kebangkitan industri dunia kosmetik dan tata rias wajah merupakan salah satu momen yang membuat manusia semakin akrab [3].

2.1.2 Skin Tone Guide

Skin tone guide telah diciptakan berdasarkan pengukuran secara sains dari ribuan *skin tones* dari berbagai tipe kulit manusia. Di formulasi khusus untuk mendekati representasi fisik warna kulit, pustaka ini adalah referensi visual lengkap dari *skin tones* untuk digunakan di pangsa pasar, dimana warna kulit menjadi kebutuhan. Hanya panduan ini yang tersedia secara global sebagai standar warna untuk mencocokkan *skin tones* [4].



Gambar 2.1 Skin Tone Guide

(*sumber : www.pantone.com)*

2.1.3 Pengertian Foundation

Alas bedak atau yang lebih dikenal dengan *foundation* saat ini sudah menjadi kosmetik wajib dalam riasan wajah. *Foundation* memegang peranan penting pada tahap riasan dasar. Ibarat melukis di atas kanvas, alas bedak merupakan polesan pertama sebelum polesan tata rias berikutnya. Tanpa polesan pertama, maka polesan selanjutnya tidak akan bisa menyatu dengan baik pada permukaan kanvas.

Demikian juga dengan alas bedak. Alas bedak merupakan dasar atau fondasi bagi keseluruhan tata rias wajah. Kulit wajah kita terkadang tidak mulus sempurna. Ada warna kulit yang tidak rata, ada flek, ada pula kerutan halus yang kadang membuat riasan tidak menyatu sempurna jika sebelumnya tidak diberi polesan alas bedak sebagai dasar.

Rangkaian alas bedak dasar jika diaplikasikan dengan benar akan membatu menyamarkan ketidaksempurnaan tersebut pada wajah. Sekaligus membantu bedak dan polesan riasan, selanjutnya menyatu alami dengan kulit, serta kulit tampak lebih mulus.

Namun, aplikasi alas bedak ini pada kulit harus dilakukan dengan cermat dan teliti, karena aplikasi alas bedak yang kurang tepat justu akan menonjolkan ketidaksempurnaan pada kulit.seperti kerutan halus, flek, dan warna kulit yang tidak homogen. Penggunaan alas bedak yang terlalu tebal juga tidak baik bagi riasan wajah. Karena, riasan akan tampak tidak alami, tebal, dan lebih mirip topeng [5].

2.1.3.1 Fungsi dan Manfaat Foundation

Seperti yang telah dijelaskan sebelumnya, alas bedak merupakan rangkaian riasan dasar yang sangat penting untuk hasil rias wajah yang maksimal.

2.1.3.1.1. Fungsi

Fungsi utama penggunaan alas bedak adalah untuk menyamarkan kekurangan atau ketidaksempurnaan pada wajah, seperti flek, bekas jerawat, kerutan halus, warna kulit yang tidak rata, dan sejenisnya.

2.1.3.1.2. Manfaat

1) Tata rias akan menempel dengan baik dan tahan lama paa wajah.
2) Wajah tampak halus dan mulus
3) Memberi kelembapan pada kulit
4) Menutupi warna-warna kulit wajah yang tidak merata

2.1.4 Pengenalan Bentuk Wajah

Wajah merupakan bagian depan dari kepala yang meliputi : daerah dari dahi hingga dagu, termasuk juga alis, mata, hidung, pipi, mulut, bibir, gigi, dan kulit. Wajah digunakan untuk menunjukkan sikap seseorang terhadap ekspresi wajah., penampilan, serta identitas dari seseorang dan tidak satu pun wajah yang sama atau serupa mutlak, bahkan pada manusia kembar sekalipun. Wajah sebagai objek dua dimensi digambarkan dengan berbagai macam iluminasi, pose, dan ekspresi wajah untuk diidentifikasi berdasarkan citra dua dimensi dari wajah tersebut. Oleh sebab itu dengan melihat wajah, manusia dapat mengenali atau mengidentifikasi seseorang dengan mudah. Manusia memiliki bentuk wajah yang berbeda-beda, yaitu :

1. Bentuk Wajah Panjang

Wajah panjang (kadang disebut “persegi panjang”) ukurannya lebih panjang dari pada lebar wajah. Wajah ini memiliki ukuran yang hampir sama pada pipi, dahi, dan rahang.

2. Bentuk Wajah Persegi

Bentuk wajah persegi kira-kira mempunyai pajang dan lebar yang sama, antara pipi dan panjangnya kira-kira 2,5 sampai 5 cm. Selain itu, ukuran pipi, dahi, dan rahang wajah persegi hampir sama, sisi wajah atas dan bawah hampir lurus. Rahang wajah persegi pasti mempunyai sudut yang jelas dan tajam pada bagiannya yang terlebar.

3. Bentuk Wajah Oval

Wajah oval adalah bentuk wajah proporsional mirip dengan bentuk telur terbalik. Lebih panjang dari pada lebarnya, dengan dahi. Lebih lebar dari pada rahang dan dagu yang halus membulat.

4. Bentuk Wajah Bulat

Wajah bulat seperti wajah persegi , kira-kira sama lebar dan panjangnya. Tetapi, wajah ini memiliki dahi yang relatif kecil dan garis rahang kecil yang melengkung. Bila selisih ukuran “pipi dan “panjang” sekitar 2,5 cm, ukuran dahi lebih kecil dari ukuran pipi, dan rahang hampir tidak mempunyai sudut tajam seperti wajah persegi.

2.1.5 Contour & Highlight

2.1.5.1 Contour

Contour adalah sebuah teknik *makeup* dimana kita dapat menyamarkan bagian wajah yang terlihat kurang sempurna.

2.1.5.2 Highlight

Highlight adalah teknik *makeup* yang fungsinya untuk menonjolkan bagian wajah agar telihat sempurna [6].

2.1.5.3 Tahap - Tahap Contour dan Highlight

1. pengguna perlu mempersiapkan wajah dengan dua shade warna. Oleskan *foundation* dengan satu warna lebih terang di bagian tengah wajah dan warna gelap di bagian pinggir wajah.
2. Baurkan kedua warna tersebut, menggunakan kuas untuk menciptakan ilusi kedalaman dan dimensi pada wajah.
3. Ratakan bagian *contour* dan beri *highlight* pada wajah. Tandai area wajah yang ingin ditambahkan *contour*, seperti bagian luar dahi, bagian sisi dagu, dan bagian luar hidung. Kemudian gunakan *highight* seperti di bagian tengah dahi, diatas tulang pipi, bagian tengah dagu, serta diantara hidung dan mulut.

4. Baurkan garis-garis *contour* dan *highlight* menggunakan kuas ataupun spons [6].



Gambar 2.2 Proses Contour dan Highlight

2.1.6 Image Sprite

Image Sprite adalah sekumpulan gambar berbeda yang disatukan dalam satu *file* gambar. Teknik ini dilakukan untuk mempercepat *loading* tanpa harus memanggil beberapa *file* gambar yang banyak. Cukup satu gambar dan kit tentukan lokasi gambar yang ingin ditampilkan [8].

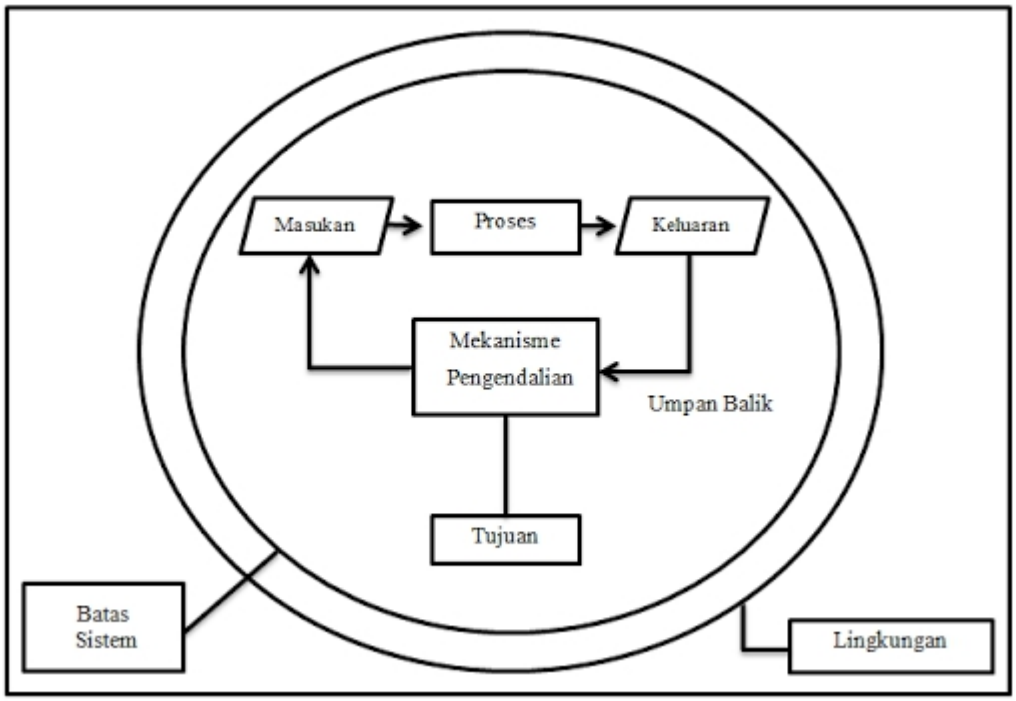
2.2 Landasan Teori

2.2.1 Konsep Dasar Sistem

Agar mampu melakukan analisis, yang sesuai dengan konsep-konsep yang berkaitan dengan sistem, sebaiknya perlu diingat kembali beberapa definisi tentang isi dari sistem. Isi tersebut meliputi data, informasi, pengetahuan dan kebijaksanaan.

2.2.2 Karakteristik Sistem

Karakteristik sistem adalah sistem yang memiliki komponen-komponen , batasan sistem, lingkungan sistem, penghubung sistem, masukan, keluaran, dan sasaran. Untuk detail nya berikut gambaran mengenai karakteristik sistem.



Gambar 2.3 Konsep Dasar Sistem

1. **Komponen (Component)**

Komponen merupakan bagian dari sistem interaksi , dimana keseluruhan komponen tersebut saling berinteraksi satu sama lain. Setiap komponen atau yang biasa disebut subsistem didalam sebuah sistem informasi memiliki sifat untuk menjalankan fungsi-fungsi tertentu didalam sebuah sistem informasi. Jadi, apabila komponen dari sistem informasi tidak dapat bekerja optimal, maka seluruh sistem informasi yang telah diimplementasikan tidak akan bekerja optimal.

2. Batasan Sistem (*Boundary)*

Karakteristik dari sebuah sistem informasi berikutnya adalah sebuah sistem informasi haruslah memiliki sebuah batasan sistem atau yang dikenal dengan istilah *boundary*. Batasan ini merupakan pembatas dari sebuah sistem informasi dengan sistem informasi lainnya, yang membuat sistem informasi tersebut menjadi satu buah kesatuan sistem informasi yang utuh, dan menunjukkan ruang lingkup yang dimilki oleh sistem informasi tersebut. Jadi, dengan adanya boundary ini, seuah sistem informasi tidak akan bekerja saling tumpang tindih satu sama lainnya, dan dapat berfungsi sesuai dengan tugas dan juga perannya masing-masing.

3. Lingkungan Luar Sistem (*Environment)*

Karakteristik dari sistem informasi berikutnya adalah memilki lingkungan luar dari sebuah sistem, atau yang disebut dengan environment. *Environment* merupakan keseluruhan sistem dan juga lingkungan yang berada di luar batasan atau *boundary* dari sebuah sistem informasi. Sebuah sistem akan disebut sebagai sistem informasi, apabila sistem tesebut memilki batasan atau *boundary*, dan juga memiliki lingkungan luar yang berbatasan langsung dengan sistem informasi tersebut.

4. Penghubung Sistem (*Interface*)

Interface atau antar muka merupakan karakteristik berikutnya yang harus dimilki oleh sebuah sistem informasi. Ya, suatu sistem akan dianggap sebagai sebuah sistem informasi yang dapat dioperasikan dengan baik dan juga optimal apabila sistem informasi tersebut memilki *interface* atau antar muka. *Interface* atau antarmuka ini merupakan media yang digunakan untuk dapat menghubungkan sebuah komponen atau subsistem yang terdapat pada sebuah sistem informasi. Hal ini mengacu pada karakteristik pertama pada sebuah sistem informasi, dimana sistem informasi memilki beberapa komponen dan juga subsistem yang menjadi dasar terbentuknya suatu keseluruhan sistem. Keseluruhan komponen dan juga subsitem tersebut di hubungkan dengan apa yang disebut dengan *interface*.Berarti,

sudah jelas terlihat, apabila suatu sistem informasi tidak memiliki interface, maka sistem tersebut tidak akan dapat berjalan dengan optimal.

5. Memiliki Masukan (input)

Karakteristik berikutnya dari sebuah sistem informasi adalah sistem input atau masukan. Input sistem atau sistem masukan ini meruapakan jenis energi yang digunakan untuk dimasukkan ke dalam suatu sistem. Masukan atau input ini terdiri dari dua jenis, yaitu: *Maintenance* merupakan input yang berhubungan dengan perawatan suatu sistem, dimana merupakan sebuah energi yang dimasukkan ke dalam sistem informasi, agar sistem informasi tersebut bisa berjalan dengan baik dan optimal.

6. Memiliki Keluaran (output)

Output atau keluaran merupakan karakteristik dari sistem informasi yang berikutnya. Output merupakan keluaran energi atau hasil yang diteruskan oleh input. Hasil atau output ini bisa berupa tampilnya data dan juga informasi yang muncul pada *display user*, yang berisi informasi. Dengan adanya output ini , maka setiap user yang menggunakan sistem informasi dapat mengakses dan juga memanfaatkan layanan informasi yang ditujukkan kepada dirinya, sehingga membuat sistem informasi dapat bekerja dengan optimal dan bermanfaat.

7. Memiliki Pengolah dan Pemrosesan Sistem

Karakteristik berikutnya yang harus dimilki oleh sistem informasi adalah sebuah pengolah data atau pemrosesan sistem. Pengolah data atau pemrosesan sistem ini merupakan komponen atau bagian di dalam sebuah sistem informasi yang memilki tugas utama untuk memproses input dari sebuah sistem informasi menadi keluaran atau output dari sebuah sistem informasi. Singkatnya, *processing system* ini membantu proses pengolahan data secara keseluruhan yang ada didalam sebuah sistem informasi, lalu mentransmisikan hasil dari pengolahan data tersebut menuju output yang dikeluarkan oleh sistem dan dapat diakses oleh user.

8. Memiliki Sasaran dan Tujuan Dari Sistem

Karakteristik terakhir merupakan karakteristik yang mungkin paling penting dari sebuah sistem informasi. Karakteristik tersebut adalah sasaran dari sistem. sasaran dari sistem merupakan analisis berupa siapa saja yang akan menggunakan sistem informasi ini. Tanpa adanya sasaran dari pembuatan sistem, maka sudah pasti sebuah sistem informasi tidak akan bisa bermanfaat dan juga berguna. Misalnya adalah, sebuah sistem informasi diimplementasikan untuk para auditor dan juga akuntan. Maka jenis dari sistem informasi yang akan diimplementasikan dan juga dikembangkan adalah jenis dari sistem informasi akuntasi, yang berisi data-data keuangan perusahaan atau suatu organisasi [9].

2.2.3 Klasifikasi Sistem

Klasifikasi sistem adalah suatu kesatuan antara satu komponen dengan komponen yang lain, karena itu tujuan sistem tersebut memiliki akhir yang berbeda untuk setiap kasusnya. Sehingga sistem dapat diklasifikasikan menjadi beberapa sistem, diantaranya:

1. **Sistem Abstrak** ***(Abstract System)*** adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sedangkan Sistem Fisik ***(Physics System)*** adalah sistem yang ada secara fisik.
2. **Sistem Alamiah** ***(Natural System)*** adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Sedangkan **Sistem Buatan Manusia** ***(Human System)*** melibatkan manusia dan mesin dalam dalam setiap prosesnya.
3. **Sistem Tertentu** ***(Deterministic System)*** adalah sistem yang beroperasi dengan tingkah laku yang dapat diprediksi. Sedangkan **Sistem Tak Tentu** ***(Probabilistic System)*** adalah sistem yang kondisi kedepannya tidak dapat diprediksi karena mengandung unsur probabilitas.
4. **Sistem Tertutup** ***(Closed System)*** adalah sistem yang tidak berhubungan dan tidak berpengaruh terhadap lingkungan luar. Sedangkan **Sistem Terbuka** ***(Open System)*** adalah sistem yang berhubungan dan berpengaruh terhadap lingkungan luar[9].

2.3 Pengolahan Citra

Definisi pengolahan citra adalah pengolahan suatu citra dengan menggunakan komputer secara khusus, untuk menghasilkan suatu citra yang lain. Sedangkan *computer vision* dapat didefinisikan setara dengan pengertian pengolahan citra yang dikaitkan dengan akuisisi citra, pemrosesan citra, klasifikasi, pengakuan, dan pencakupan keseluruhan, pengambilan keputusan yang diikuti dengan pengidentifikasian citra.

Ballard dan Brown (1982) mendefinisikan *computer vision* sebagai suatu kegiatan awal pengotomatisan dan pengintegrasian suatu pemrosesan dan representasi sebagai suatu persepsi visual dengan tahap-tahap tertentu.

Niblack(1986) memaparkan pengolahan citra sebagai "pengolahan citra dengan komputer", dan kemudian definisi singkat ini ditambah dengan "dan output atau hasil pengolahan citra tersebut juga berupa citra". Sementara "*computer vision"* melibatkan banyak teknik dari pengolahan citra dan juga lebih luas lagi, dalam wilayah suatu sistem yang lengkap, "mesin melihat".

Boyle dan Thomas(1988) memberi pengertian bahwa *computer vision* lebih dari hanya sekedar *image recognition*. Mereka juga menghadirkan operasi *low level processing* sebagai suatu algoritma pengolahan citra yang dapat kita sebut *purely* yang kemudian mengkategorikan citra tersebut didalam suatu *computer vision*.

Jensen (1986) memberi judul untuk bukunya *introductory Digital Image Processing – a remote sensing perspective,* dan memasukkan bagian-bagian substansial berdasarkan *image acquisition* dan *region classification*.

Gonzalez dan Wintz (1977) memasukkan bagian-bagian substansial (Jensen) berdasarkan deskripsi region dan deskripsi relasional, dan keseluruhan pendapat yang berkaitan dengan pengelompokkan bagian-bagian tersebut didalam wilayah *computer vision* [10].

2.3.1 Proses Umum Pelatihan Citra Wajah

5. Pendeteksian Wajah

Pendeteksian wajah adalah proses yang men-segmentasi area wajah dengan latar belakang dari suatu masukan citra. Proses ini bekerja dengan cara memeriksa citra yang diinputkan, jika memiliki citra wajah, maka akan dilakukan pemisana dengan cara memotog citra wajah dari latar belakang citra yang diinputkan. Jika masukan berbentuk video, proses yang dilakukan adalah proses *face tracking* . secara umum proses *face tracking* dan proses pendeteksian wajah memiliki fungsi yang sama. Perbedaanya terletak pada proses pendeteksiannya saja. Jika pada masukan berbentuk citra, sistem berjalan *offline* sehingga dapat menggunakan proses pendeteksian wajah, sedangkan pada masukan video, sistem berjalan secara *online* atau *real-time* yang membutuhkan pendeteksian secara langsung, maka proses yang digunakan adalah proses *face tracking*



Gambar 2.4 *Face Tracking*

6. Penyelarasan Wajah

Pada proses pendeteksian wajah, citra wajah yang didapatkan masih berupa perkiraan kasar atau masih memiliki kualitas yang cukup buruk seperti ukuran yang berbeda dengan ukuran normal, faktor pencahayaan yang kurang atau lebih,

kejelasan citra yang buruk dan sebagainya, maka perlu diadakannya proses penyelarasan wajah.

Proses penyelasaran wajah merupakan proses yang bertujuan untuk menormalisasi wajah dari citra wajah yang didapatkan dari proses pendeteksian wajah. Proses ini terdiri dari tahapan- tahapan sebagai berikut :

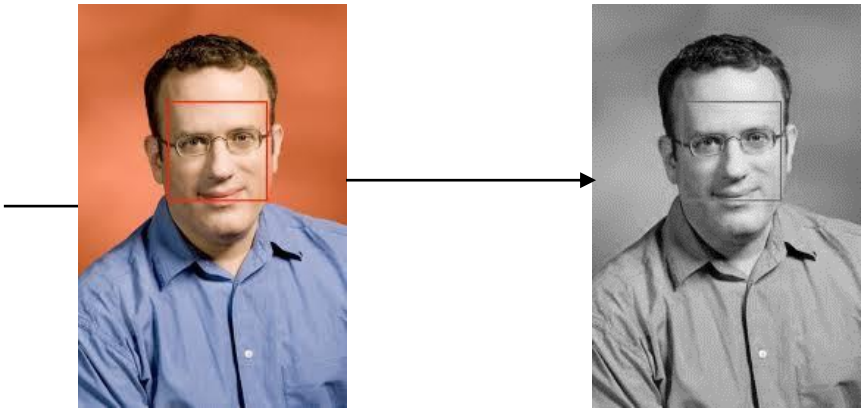
1. Grayscaling (tahap konversi warna RGB menjadi abu)
2. Cropping (tahap pemisana citra wajah dengan latar belakannya)
3. Resizing (tahap normalisasi dimensi citra)
4. Equalizing (tahap koreksi tingkat kecerahan citra)

1. Grayscaling citra

Grayscaling citra merupakan tahapan pertama dari proses penyelarasan. Pada tahap ini terjadi pengkonversian citra warna RGB menjadi citra berwarna abu. Citra warna RGB terdiri dari tiga parameter, yaitu merah (red), hijau (green), dan biru (blue). Jika citra warna RGB in diinputkan kedalam proses ekstraksi, maka proses tersebut akan sulit untuk dilakukan karena citra RGB terdiri dari tiga parameter, oleh karena itu diperlukan penyamaan parameter yaitu dengan cara grayscaling. Berikut adalah persamaan tahap grayscaling citra :

$$\mathbf{X = 0.299r + 0.587g + 0.114b}$$

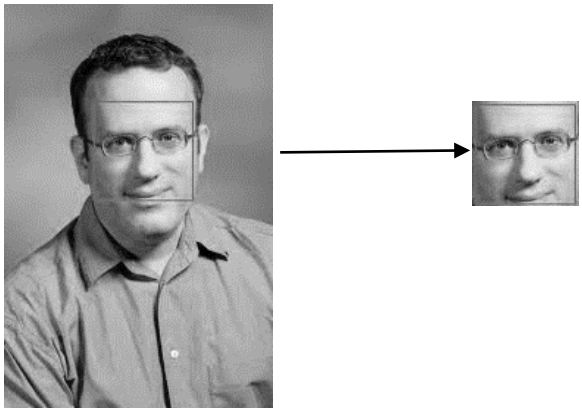
Dimana nilai RGB dengan rentang nilai pada integer 0-255 dan nilai X sebagai grayscale.



Gambar 2.5 Grayscaling Citra

2. Cropping citra

Pada tahapan ini terjadi pemotongan citra yang memisahkan citra wajah dengan citra masukannya, tujuannya untuk mengambil citra yang hanya diperlukan untuk proses ekstraksi, dalam hal ini adalah citra wajah dan membuang citra lain yang tidak diperlukan. Dimensi citra yang dipotong disesuaikan dengan dimensi dari proses segmentasi atau pengkotakan objek wajah yang dilakukan pada proses pendeteksian wajah.



Gambar 2.6 Cropping Citra

3. Resizing Citra

Pada tahap resizing citra, terjadi proses normalisasi dimensi citra wajah, yaitu proses pembesaran atau pengecilan dimensi citra wajah menjadi dimensi yang telah ditentukan. Tujuannya, untuk menyamakan dimensi wajah dari setiap citra yang diinputkan, sehingga pada prses ekstraksi citra nanti tidak akan ada perbedaan dimensi dari matriks citra wajah.

4. Equalizing Citra

Tahap ini adalah tahapan terakhir dari proses penyelarasan, yang tujuannya untuk memperjelas nilai histogram dari citra wajah hasil tahapan-tahapan sebelumnya. [11]



Gambar 2.7 Equalizing citra

2.3.2 Ektraksi Fitur Wajah

Feature Extraction adalah proses pengambilan ciri-ciri yang unik dari data yang akan diolah untuk memperkecil jumlah data mengambil informasi yang terpenting dari data yang diolah dan mempertinggi persisi pengolahan data. Pada proses ini menggunakan metode *Facial Landmark Detection.*

2.3.2.1 Facial Landmark Detection

Pendeteksian *facial landmarks* merupakan bagian dari *shape detection*. Memberikan masukan gambar, *shape predictor* mencoba melokalisasi bagian penting dari bentuk wajah.Dalam konteks *facial landmarks,* tujuannya adalah untuk mendeteksi bagian penting dari struktur wajah menggunakan metode deteksi bentuk.

Dalam mendeteksi *facial landmarks* terdapat dua proses, yaitu :

1. Langkah 1 : lokalisasi wajah pada gambar
2. Langkah 2 : mendeteksi struktur wajah pada ROI

Terdapat banyak variasi dari *facial landmark detectors,*tetapi seluruh metode intinya mencoba untuk melokalisasi label dari *face regions,* yaitu :

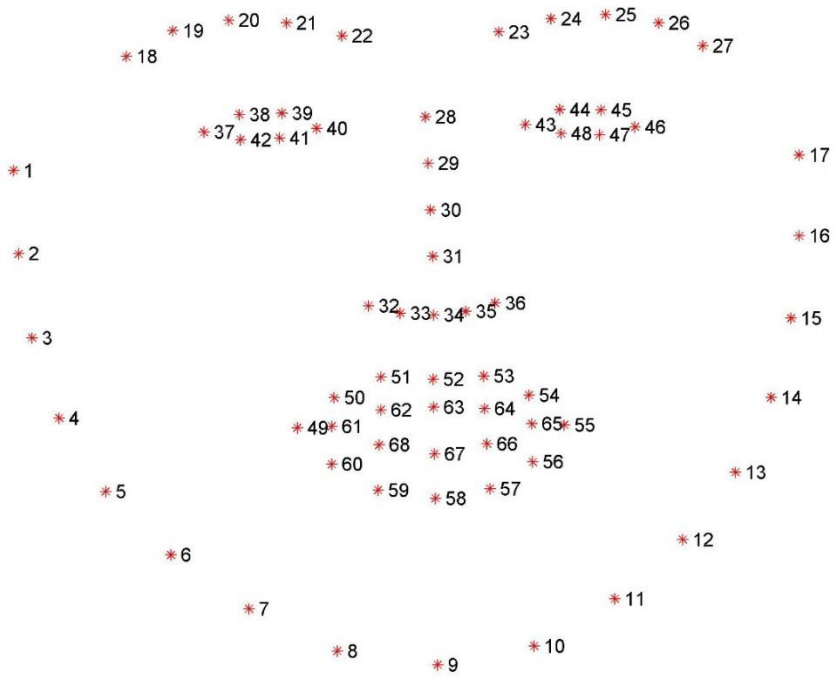
- Bibir
- Alis kanan
- Alis kiri
- Mata kanan
- Mata kiri
- Hidung
- Rahang

Facial landmark detector termasuk kedalam pustaka dlib sebagai sebuah implementasi dari "*One millisecond face alignment with an ensemble of regression trees*" jurnal yang dibuat oleh Kazemi dan Sullivan (2004). Metode ini dimulai menggunakan:

1. data latih berupa gambar. Gambar tersebut secara manual telah berlabel kordinat x dan y
2. sebelumnya, untuk lebih spesifik, hitung jarak antara piksel gambar masukan.

1. Memahami dlib untuk *facial landmark detector*

Data latih pada Pustaka dlib digunakan untuk menghitung estimasi lokasi 68 koordinat (x,y) yang memetakan struktur wajah. Berikut adalah visualisasi dari 68 koordinat pada wajah :



Gambar 2.8 Hasil Pencarian Fitur-Fitur Wajah

Angka-angka tersebut merupakan bagian dari 68 Ibug 300-w dataset yang mana dlib *facial landmark predictor* telah terlatih didalamnya.

2. Mendeteksi *facial landmarks* dengan dlib, OpenCV, dan Python

Untuk proses deteksi wajah fungsi yang digunakan pertama kali adalah **rect_to_bb** yang mana fungsi tersebut kependekan dari *rectangle to bounding box* . Fungsi ini menerima argumen tunggal yaitu **rect** yang diasumsikan sebagai *bounding box rectangle* dan diproduksi oleh dlib *detector (face detector)*. Objek **rect** termasuk pendeteksian dalam koordinat (x,y). Fungsi yang kedua digunakan adalah **shape_to_np.** Dlib *facial landmark detector* akan mengembalikan 68 koordinat (x,y) yang mengandung objek **shape** dalam *facial landmark regions*. Untuk mengubah objek ke NumPy array dapat menggunakan fungsi **shape_to np.** Fungsi selanjutnya dalam proses *facial landmark detection* adalah **face_utils** yang merupakan sub modul dari **imutils** [13].

2.3.3 Gaussian Blur

Salah satu metode yang digunakan untuk memberikan efek *blur* adalah dengan metode *Gaussian filter*. *Gaussian filter* adalah suatu *filter* dengan nilai pembobotan pada setiap piksel dipilih berdasarkan bentuk fungsi *Gaussian. Filter* ini sangat baik dan digunakan untuk menghilangkan *noise* yang bersifat sebaran normal, yang banyak dijumpai pada citra hasil proses digitalisasi menggunakan kamera. Hal ini terjadi karena merupakan fenomena alamiah akibat dari sifat pantulan cahaya dan kepekaan sensor cahaya pada kamera itu sendiri. *Gaussian filtering* termasuk dalam kelas *low-pass filters*. Jenis *filtering* ini merupakan salah satu jenis dari spatial *filtering* (suatu konsep modifikasi nilai piksel citra digital). [7]

Persamaan pada fungsi gaussian gambar 1 dimensi adalah sebagai berikut

$$:\boldsymbol{G(x)} = \frac{\mathbf{1}}{\sqrt{\mathbf{2\pi\sigma^2}}}\boldsymbol{e}\ \frac{\boldsymbol{-x^2}}{\mathbf{2\sigma^2}}$$

Sedangkan untuk gambar 2 dimensi, persamaan fungsi gaussian adalah sebagai berikut :

$$:\boldsymbol{G(x,y)} = \frac{\mathbf{1}}{\sqrt{\mathbf{2\pi\sigma^2}}}\boldsymbol{e}\ \frac{\boldsymbol{x^2+y^2}}{\mathbf{2\sigma^2}}$$

2.4 Perangkat Lunak Pendukung

2.4.1 Python

Python adalah bahasa pemrograman interpretatif yang dianggap mudah dipelajari serta berfokus pada keterabatasan kode. Dengan kata lain Python diklaim sebagai bahasa pemrograman yang memiliki kode-kode pemrograman yang sangat jelas, lengkap, dan mudah untuk dipahami.

Python secara umum berbentuk pemrograman berorientasi objek, pemrograman imperatif, dan pemrogaran fungsional

Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. [12]

2.4.1.1 Fitur Bahasa Pemrograman Python

Beberapa fitur dan kelebihan yang dimiliki Python adalah :

1. Memiliki koreksi kepustakaan yang banyak. Itu artinya, telah tersedia modul-modul "siap pakai" untuk berbagai keperluan.
2. Memiliki struktur bahasa yang jelas, sederhana, dan mudah dipelajari.
3. Berorientasi objek
4. Memiliki sistem pengelolaan memori otomatis *(garbage collection)* seperti halnya java.
5. Bersifat modular sehingga mudah dikembangkan dengan menciptakan modul-modul baru, baik dibangun dengan bahasa Python maupun C/C++ [12].

2.4.2 C# (C Sharp)

C# adalah salah satu bahasa pemrograman yang menggunakan framework .NET. seperti semua bahasa lainnya, C# memiliki aturan-aturan syntax dan kata-kata yang bisa digunakan untuk membuat aplikasi [13].

2.4.3 OpenCV

OpenCV (*Open Computer Vision)* adalah sebuah API *(Application Programming Interface)* yang dirilis dibawah lisensi BSD yang dapat dipakai oleh siapa saja karena bersifat *open source* diperuntukan untuk komersial dan akademis. OpenCV sendiri dirancang untuk efisiensi komputasi yang berfokus pada aplikasi secara *Real – time.* Dapat ditulis dengan berbagai bahasa pemograman seperti *C / C++ , Phyton, Java* dan *Php* yang telah dioptimalkan. Pada OpenCV mampu dijalankan pada berbagai jenis *Flatform* seperti Windows, Linux, Android dan Mac OS. Pada perpustakaan dapat memanfaatkan *Multi – core* yang diaktifkan dengan OpenCL dan CUDA yang sedang dikembangkan pada saat ini. OpenCV sendiri ditulis secara asli pada bahasa pemograman *C++* dan mempunyai template antarmuka yang bekerja dengan mulus dengan kontainer STL. OpenCV dibangun untuk media pembelajaran pada bidang ilmu visi komputer yang menyediakan

infrastruktur umum untuk mempercepat pada penggunaan persepsi mesin dalam produk komersial, memudahkan bisnis dalam memanfaatkan dan memodifikasi kode.

Pada OpenCV sudah ada 2500 algoritma yang sudah dioptimalkan yang mencangkup satu set lengkap visi komputer klasik dan algoritma pembelajaran pada mesin. Algoritma ini dapat digunakan untuk *Face Recognition, Face Detection, Face/Object Tracking* dan lain – lain [14].

2.5 Object Oriented Programming (OOP)

OOP adalah suatu cara mengorganisasi program dengan memodelkan objek-objek dunia nyata (seperti benda, sifat, sistem, dan lainnya) ke dalam sebuah bahasa pemrograman. Dalam bahasa pemrograman pemodelan OOP ini doaplikasikan dengan sebuah kelas, method, dann properti/variabel [15].

2.6 Unified Model Language (UML)

Unified Modelling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desai sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograma berorientasi objek (OO). Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang-orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat proses rancangan-bangu perangkat lunak efektif.

UML merupakan standar yang relatif terabuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorsium terbuka terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas, khususnya interoperibilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar-standar CORBA *(Common Object Request Broker Architecture).*

UML lahir dari penggabungan banyak bahasa pemodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. Sejak

kehadirannya pada tahun 1997, UML menghancurkan menara Babel tersebut menjadi sejarah [16].

2.6.1 Use Case

Use Case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan [17].

2.6.2 Activity Diagram

Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi, perbedaan prinsip antara diagram ini dan notasi diagram ini mendukung behavior parallel [17].

2.6.3 Class Diagram

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukkan properti dan operasi sebuah class dat batasan batasan yang terdapat dalam hubungan-hubungan objek tersebut. UML mengguakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah *class*.

2.6.4 Sequence Diagram

Sebuah *sequence diagram*, diagram, secara khusus, menjabarkan behavior sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini didalam *use case*.

2.7 Metode Pengujian Sistem

Metode pengujian sistem dilakukan untuk mengetahui efektifitas perangkat lunak yang digunakan. Metode pengujian sistem yang digunakan adalah dengan menggunakan pengujian *blackbox.*

1. Pengujian Blackbox

Terfokus pada unit program memenuhui kebutuhan *(requirement)* yang disebutkan dalam spesifikasi. Pada *black box testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati hasil dari unit itu sesuai dengan proses bisnis yang diinginkan [8].