



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

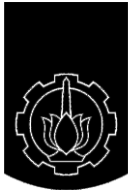
**TUGAS AKHIR - EC 184801**

## **VIRTUAL MAKE-UP BERBASIS VISUAL FACE LANDMARKING**

Dyah Retno Arumsari  
NRP 07211340000003

Dosen Pembimbing  
Dr. Eko Mulyanto Yuniarno, S.T., M.T.  
Dr. Supeno Mardi Susiki, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - EC 184801**

## **VIRTUAL MAKE-UP BERBASIS VISUAL FACE LANDMARKING**

Dyah Retno Arumsari  
NRP 07211340000003

Dosen Pembimbing  
Dr. Eko Mulyanto Yuniarno, S.T., M.T.  
Dr. Supeno Mardi Susiki, ST., MT.

DEPARTEMEN TEKNIK KOMPUTER  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020



## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Virtual Make-up Berbasis Visual Face Landmarking**” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, April 2020

Dyah Retno Arumsari  
NRP. 0721134000003



# LEMBAR PENGESAHAN

## Virtual Make-up Berbasis Visual Face Landmarking

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh: Dyah Retno Arumsari (NRP: 07211340000003)

Tanggal Ujian : 6 Juli 2020

Periode Wisuda : September 2020

Disetujui oleh:

Dr. Eko Mulyanto Yuniarno, ST., MT. (Pembimbing I)  
NIP: 19680601 199512 1 009

Dr. Supeno Mardi Susiki Nugroho, ST., MT. (Pembimbing II)  
NIP: 19700313 199512 1 001

Dr. I Ketut Eddy Purnama, ST., MT (Penguji I)  
NIP: 19690730 199512 1 001

Arief Kurniawan, ST., MT. (Penguji II)  
NIP: 19740907 200212 1 001

Ahmad Zaini, ST., M.Sc. (Penguji III)  
NIP: 19750419 200212 1 003



Mengetahui  
Kepala Departemen Teknik Komputer  
Dr. Supeno Mardi Susiki Nugroho, ST., MT.  
NIP: 19700313 199512 1 001

# ABSTRAK

Nama Mahasiswa : Dyah Retno Arumsari  
Judul Tugas Akhir : Virtual Berbasis Visual Face Landmarking  
Pembimbing : 1. Dr. Eko Mulyanto Yuniarno, ST., MT.  
2. Dr. Supeno Mardi Susiki, ST., MT.

Berkembangnya industri 4.0 diikuti juga dengan semakin maraknya jasa *e-commerce* terutama dibidang kecantikan. Hal ini memudahkan pelanggan untuk dapat berbelanja produk kecantikan dengan hanya menggunakan *gadget* atau *smartphone* pribadi tanpa harus meninggalkan rumah. Namun saat berbelanja secara online, pelanggan tidak dapat mencoba langsung produk yang akan dibeli, sehingga dibutuhkan sebuah layanan atau media untuk menjembatani kekurangan ini demi meningkatkan kepuasan pelanggan dan menurunkan tingkat ketidakcocokan antara kosmetik dan warna kulit pelanggan. Dengan adanya teknologi *Smart Mirror* yang dilengkapi kamera dan digabungkan dengan teknologi citra video menggunakan metode *Visual Face Landmarking* diharapkan dapat menghasilkan tampilan make-up secara virtual pada calon pembeli, sehingga dapat membantu untuk mengurangi ketidakcocokan produk kosmetik atau kecantikan. Metode *Visual Face Landmarking* berbasis Dlib pada OpenCV akan melakukan deteksi pada wajah yang telah ditangkap oleh webcam dan kemudian warna akan ditambahkan pada bagian-bagian wajah tertentu seperti mata, pipi dan bibir dalam bentuk make-up virtual. Kontrol pada cermin dilakukan melalui aplikasi android yang terhubung pada Raspberry Pi pada cermin.

*Kata Kunci: Smart Mirror, Visual Face Landmarking, Dlib, Raspberry Pi, Virtual Make-up.*

*Halaman ini sengaja dikosongkan*

# ABSTRACT

Name : Dyah Retno Arumsari  
Title : *Virtual Make-up Based on Visual Face Landmarking*  
Advisor : 1. Dr. Eko Mulyanto Yuniarno, ST., MT.  
2. Dr. Supeno Mardi Susiki, ST., MT.

With the development in 4.0 industry, also followed by vast e-commerce services especially in the beauty industry. This development makes it easier to shop beauty products with only using personal gadgets or smartphones from the comfort of one's home. But shopping online also comes with a few shortcomings, in this case potential buyers can not sample the products first before doing the transactions, so a services or media to solve this problem is highly sought after in order to minimize the beauty products incompatibility with the potential buyer's skin tones. With the smart mirror technology equipped with a camera paired with video image technology is expected to produce a virtual make-up look to potential buyers, in hopes of minimizing the incompatibility of the beauty products. Visual Face Landmarking method based on Dlib in OpenCV will detect important part of potential buyer's face by webcam to be then added some colors to the specific part of the face, such as eyes, cheeks and lips in the form of virtual make-up. Smart mirror will be controlled using Android application which then will be connected to Raspberry Pi inside the mirror.

*Keywords: Smart Mirror, Visual Face Landmarking, Dlib, Raspberry Pi, Virtual Make-up.*

*Halaman ini sengaja dikosongkan*

# KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT atas segala limpahan berkah dan hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul "Virtual Make-up pada Smart Mirror berbasis Visual Face Landmarking". Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer, FTEIC-ITS serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak, Adik, Saudara sepupu-sepupu, Bude, Bulek, Pakde, Paklek dan Mamang tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Dr. Supeno Mardi Susiki Nugroho, ST., MT. selaku Kepala Departemen Teknik Komputer, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.
3. Secara khusus penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Dr. Eko Mulyanto Yuniarno, ST., MT. dan Bapak Dr. Supeno Mardi Susiki Nugroho, ST., MT. atas bimbingannya selama pengerjaan tugas akhir ini.
4. Bapak-ibu dosen pengajar Departemen Teknik Komputer, atas bimbingan, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Teman-teman Tekkom 2013 yang masih berjuang bersama, dan teman-teman Tekkom 2014-2018 yang selalu menyemangati dan membantu saya.
6. Teman-teman Bilingual Class SMAN 1 Dumai yang selalu ada untuk mendengar keluh kesah dan menyemangati saya.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

*Halaman ini sengaja dikosongkan*

# DAFTAR ISI

<b>Abstrak</b>	<b>i</b>
<b>Abstract</b>	<b>i</b>
<b>KATA PENGANTAR</b>	<b>iii</b>
<b>DAFTAR ISI</b>	<b>v</b>
<b>DAFTAR GAMBAR</b>	<b>vii</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar belakang . . . . .	1
1.2 Permasalahan . . . . .	1
1.3 Tujuan . . . . .	1
1.4 Batasan masalah . . . . .	2
1.5 Sistematika Penulisan . . . . .	2
1.6 Relevansi . . . . .	3
<b>2 TINJAUAN PUSTAKA</b>	<b>5</b>
2.1 Citra Digital . . . . .	5
2.2 Jenis Citra . . . . .	6
2.2.1 Citra Biner (Monokrom) . . . . .	7
2.2.2 Citra Skala Keabuan <i>GreyScale</i> . . . . .	8
2.2.3 Citra Warna ( <i>True Color</i> ) . . . . .	9
2.3 Visi Komputer . . . . .	11
2.4 <i>Image Processing</i> . . . . .	12
2.5 Python . . . . .	12
2.6 OpenCV . . . . .	12
2.7 <i>Facial Landmark</i> . . . . .	13
2.8 Dlib C++ Library . . . . .	15
<b>3 DESAIN DAN IMPLEMENTASI SISTEM</b>	<b>17</b>
3.1 Desain Sistem . . . . .	17
3.2 Alur Kerja Sistem . . . . .	18
3.2.1 Persiapan Data Gambar . . . . .	18
3.2.2 Deteksi <i>Facial Landmark</i> . . . . .	19



3.2.3	Segmentasi Fitur Wajah . . . . .	21
3.2.4	Aplikasi Make-up . . . . .	22
<b>4</b>	<b>PENGUJIAN DAN ANALISA</b>	<b>27</b>
4.1	Aplikasi <i>Facial Landmark</i> . . . . .	27
4.2	Virtual Make-up menggunakan Gambar . . . . .	29
4.3	Virtual Make-up secara <i>Real Time</i> . . . . .	34
<b>5</b>	<b>PENUTUP</b>	<b>39</b>
5.1	Kesimpulan . . . . .	39
5.2	Saran . . . . .	40
	<b>DAFTAR PUSTAKA</b>	<b>41</b>
	<b>Biografi Penulis</b>	<b>43</b>

# DAFTAR GAMBAR

2.1	Koordinat Citra Digital . . . . .	6
2.2	Citra biner dan bentuk penyimpanannya dalam memori . . . . .	7
2.3	Citra Biner . . . . .	7
2.4	Citra Biner . . . . .	8
2.5	Citra Greyscale . . . . .	9
2.6	Palet warna kuning (255, 255, 0) . . . . .	9
2.7	Citra warna . . . . .	10
2.8	Citra warna . . . . .	10
2.9	Logo Python . . . . .	12
2.10	Contoh Facial Landmark . . . . .	14
2.11	Facial Landmark . . . . .	15
3.1	Flowchart Sistem . . . . .	17
3.2	Alur Proses Virtual Make-up . . . . .	18
3.3	Logitech 720p . . . . .	19
3.4	Facial Landmark dari depan . . . . .	20
3.5	Data Gambar Wajah . . . . .	21
3.6	Segmentasi Bagian Mata . . . . .	22
3.7	Alur Proses Virtual Make-up Lipstik . . . . .	23
3.8	Alur Proses Virtual Make-up Eyeliner . . . . .	24
3.9	Tanpa Make-up . . . . .	25
3.10	Setelah Make-up . . . . .	25
4.1	Facial Landmark Depan . . . . .	27
4.2	Facial Landmark Kanan . . . . .	28
4.3	Facial Landmark Kiri . . . . .	28
4.4	Kode untuk Virtual Make-up pada Gambar . . . . .	30
4.5	Kode untuk Virtual Make-up pada Gambar . . . . .	30
4.6	Sebelum Make-up . . . . .	31
4.7	Sesudah Make-up . . . . .	31
4.8	Sebelum Make-up . . . . .	31
4.9	Sesudah Make-up . . . . .	31
4.10	Sebelum Make-up . . . . .	32
4.11	Sesudah Make-up . . . . .	32
4.12	Sebelum Make-up . . . . .	32

4.13	Sesudah Make-up . . . . .	32
4.14	Sebelum Make-up . . . . .	33
4.15	Sesudah Make-up . . . . .	33
4.16	Sebelum Make-up . . . . .	33
4.17	Sesudah Make-up . . . . .	33
4.18	Sebelum Make-up . . . . .	34
4.19	Sesudah Make-up . . . . .	34
4.20	Kode untuk Virtual Make-up pada Video . . . . .	34
4.21	Kode untuk Virtual Make-up pada Video . . . . .	35
4.22	Sebelum Make-up . . . . .	36
4.23	Sebelum Make-up . . . . .	36
4.24	Sebelum Make-up . . . . .	37
4.25	Sesudah Make-up . . . . .	37
4.26	Sebelum Make-up . . . . .	37
4.27	Sesudah Make-up . . . . .	37
4.28	Sebelum Make-up . . . . .	37
4.29	Sesudah Make-up . . . . .	37
4.30	Contoh Warna . . . . .	38
4.31	Contoh Warna . . . . .	38

# BAB 1

## PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

### 1.1 Latar belakang

Berkembangnya industri 4.0 diikuti juga dengan semakin maraknya jasa e-commerce terutama dibidang kecantikan. Make-up e-commerce merupakan salah satu industri terbesar dan terlaris di dunia. Kemudahan dalam memilih produk-produk kecantikan melalui website atau aplikasi khusus e-commerce memberikan kemudahan bagi pelanggan terutama pelanggan Indonesia yang relatif konsumtif. Jika image processing digabung dengan e-commerce kecantikan akan membuat kegiatan berbelanja lebih efisien dan dapat menambah jumlah pilihan dari produk-produk tersebut.

Namun dengan banyaknya perbedaan skin tones dan undertones pada kulit, tidak semua warna kosmetik cocok digunakan oleh pelanggan. Salah satu kekurangan terbesar dari berbelanja secara online adalah pelanggan tidak dapat mencoba langsung produk yang akan dibeli. Sehingga berbelanja kosmetik memiliki resiko besar ketidakcocokan warna kosmetik pada warna kulit pelanggan. Dan mayoritas dari make-up e-commerce adalah tidak adanya return policy atau barang yang dibeli tidak bisa dikembalikan lagi karena alasan kebersihan dan lainnya.

### 1.2 Permasalahan

Diperlukannya sebuah teknologi yang memungkinkan pelanggan untuk mencoba warna kosmetik yang akan dibeli secara virtual demi menentukan kecocokan pada warna kulit.

### 1.3 Tujuan

Diharapkan adanya teknologi yang dapat membantu pembeli make-up untuk mencoba warna make-up yang diinginkan tanpa harus mencoba secara langsung.

## 1.4 Batasan masalah

Batasan masalah pada Tugas Akhir ini adalah :

1. Metode yang digunakan untuk mendeteksi wajah adalah Visual Face Landmarking.
2. Aplikasi virtual make-up dilakukan dengan 2 cara yaitu dengan menggunakan gambar dan juga secara *real-time* menggunakan webcam.

## 1.5 Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu :

1. BAB I Pendahuluan  
Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.
2. BAB II Dasar Teori  
Pada bab ini berisi tentang uraian secara sistematis teori-teori yang berhubungan dengan permasalahan yang dibahas pada pengerjaan tugas akhir ini. Teori-teori ini digunakan sebagai dasar dalam pengerjaan, yaitu informasi terkait pertumbuhan jumlah kendaraan, teori tentang *CCTV*, teori mengenai *deep learning*, dan teori-teori penunjang lainnya.
3. BAB III Perancangan Sistem dan Implementasi  
Bab ini berisi tentang penjelasan-penjelasan terkait sistem yang akan dibuat. Guna mendukung itu digunakanlah blok diagram atau *work flow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implementasi pada pelaksanaan tugas akhir.
4. BAB IV Pengujian dan Analisa  
Bab ini menjelaskan tentang pengujian yang dilakukan terhadap sistem hasil dari tugas akhir ini dan menganalisa sistem tersebut. Spesifikasi perangkat keras dan perangkat lunak yang digunakan juga disebutkan dalam bab ini. Sehingga ketika akan dikembangkan lebih jauh, spesifikasi perlengkapan-

nya bisa dipenuhi tanpa harus melakukan uji coba perangkat lunak maupun perangkat keras lagi.

## 5. BAB V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk mengembangkan lebih lanjut juga dituliskan pada bab ini.

## 1.6 Relevansi

Dalam tugas akhir ini, dilakukan penelitian pengolahan citra video berbasis *Visual Face Landmarking* pada OpenCV untuk penggunaan make-up secara virtual.

*Halaman ini sengaja dikosongkan*

## BAB 2

# TINJAUAN PUSTAKA

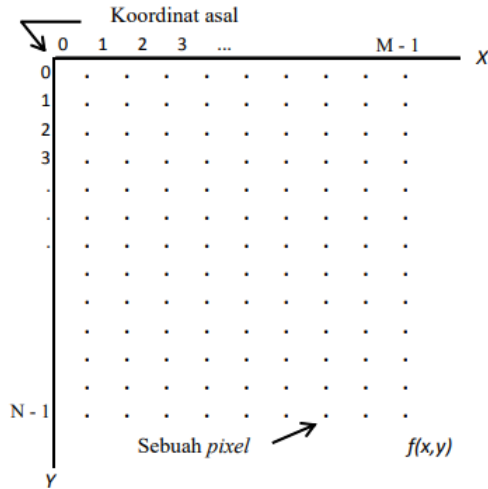
Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

### 2.1 Citra Digital

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik pengolahan citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam) [1]. Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer.

Secara matematis, citra merupakan fungsi kontinyu (*continue*) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Reperesentasi dari fungsi kontinyu menjadi nilai-nilai diskrit disebut digitalisasi citra. Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi  $f(x,y)$  yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel (*pixel = picture element*) atau elemen terkecil dari sebuah citra.





**Gambar 2.1:** Koordinat Citra Digital

Nilai pada suatu irisan antara baris dan kolom (pada posisi  $x,y$ ) disebut picture elements, image elements, pels, atau pixels. Istilah terakhir (pixel) paling sering digunakan pada citra digital. Pixel merupakan elemen terkecil dari sebuah citra. Pixel mempunyai 2 parameter yaitu koordinat dan intensitas atau warna.

Nilai yang terdapat di koordinat  $(x,y)$  adalah  $f(x,y)$ , yaitu besar intensitas atau warna dari pixel di titik itu. Berdasarkan bentuk matrik di atas, secara sistematis citra digital dapat ditulis sebagai fungsi intensitas  $f(x,y)$ , dimana harga  $x$  (baris) dan  $y$  (kolom) merupakan koordinat posisi dan  $f(x,y)$  adalah nilai fungsi pada setiap titik  $(x,y)$  yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari piksel di titik tersebut [2].

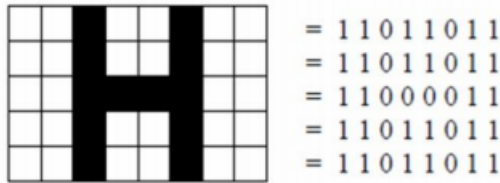
## 2.2 Jenis Citra

Nilai suatu pixel memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum untuk citra 8-bit jangkauannya adalah 0 – 255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra integer.

Berikut ini jenis-jenis citra berdasarkan nilai pixelnya [3].

### 2.2.1 Citra Biner (Monokrom)

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai pixel yaitu hitam dan putih. Citra biner juga disebut sebagai citra BW (*black and white*) atau citra monokrom. Pada citra biner setiap titik (pixel) bernilai 0 atau 1, masing-masing mempresentasikan warna tertentu. Contoh yang paling lazim, warna hitam bernilai 0 dan warna putih bernilai 1. Setiap titik (pixel) pada citra hanya membutuhkan media penyimpanan 1 bit, sehingga setiap byte dapat menampung informasi 8 titik (pixel).



**Gambar 2.2:** Citra biner dan bentuk penyimpanannya dalam memori

Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti segmentasi, pengambangan, ataupun morfologi.



**Gambar 2.3:** Citra Biner

## 2.2.2 Citra Skala Keabuan *GreyScale*

Citra grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian red, green dan blue memiliki warna yang sama, yaitu warna dari hitam, keabuan, dan putih. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Tingkatan keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih.



**Gambar 2.4:** Citra Biner

Citra *grayscale* memberi kemungkinan warna yang lebih banyak dari pada citra biner, karena ada nilai-nilai lain diantara nilai minimum (biasanya = 0) dan nilai maksimum. Banyaknya kemungkinan minimum dan nilai maksimumnya bergantung pada jumlah bit yang digunakan [4].

Citra skala keabuan mempunyai kemungkinan warna antara hitam (minimum) dan putih (maksimum). Contoh untuk skala keabuan 4 bit, maka jumlah kemungkinan nilainya adalah  $2^4 = 16$  (memiliki 16 warna), dan nilai maksimumnya adalah  $2^4 - 1 = 15$ , kemungkinan warna 0 (min) sampai 15 (maks). Sedangkan untuk skala keabuan 8 bit, maka jumlah kemungkinan nilainya adalah  $2^8 = 256$  (memiliki 256 warna), dan nilai maksimumnya  $2^8 - 1 = 255$ , kemungkinan warna 0 (min) sampai 255 (maks). Citra grayscale berikut memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan).



**Gambar 2.5:** Citra Greyscale

### 2.2.3 Citra Warna (*True Color*)

Pada citra warna, setiap titik mempunyai warna yang spesifik yang merupakan kombinasi dari 3 warna dasar, yaitu merah, hijau, dan biru. Format citra ini sering disebut sebagai citra RGB (*red-green-blue*). Setiap warna dasar mempunyai intensitas sendiri dengan nilai maksimum 255 (8 bit), dan warna minimum adalah putih. Red memiliki warna minimum putih dan warna maksimum merah. Green memiliki warna minimum putih dan warna maksimum hijau. Blue memiliki warna minimum putih dan warna maksimum biru. Misalnya warna kuning merupakan kombinasi warna merah dan hijau sehingga nilai RGB-nya adalah (255 255 0). Dengan demikian setiap titik (pixel) pada citra warna membutuhkan data 3 byte.



**Gambar 2.6:** Palet warna kuning (255, 255, 0)



Gambar 2.7: Citra warna

Jumlah kombinasi warna yang mungkin untuk citra format bmp ini adalah citra 24-bit, atau lebih dari 16 juta warna, dengan demikian bisa dianggap mencakup semua warna yang ada, inilah sebabnya format ini dinamakan *true color*.

orange			black	242 130 38	79 98 40	148 139 84	0 0 0
red	yellow	green		255 0 0	255 255 0	0 255 0	141 180 227
purple				150 0 150	255 255 255	0 204 255	151 72 7
pink		Blue		204 51 153	192 0 0	0 0 255	147 111 45

Citra warna

Representasi data digital

Gambar 2.8: Citra warna

## **Citra Warna (8 bit)**

Citra yang setiap pixel dari citra warna (8 bit) hanya diwakili oleh 8 bit dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna.

## **Citra Warna (10 bit)**

Pada citra 8 bit sebelumnya, untuk merah (R), hijau (G) dan biru (B), memiliki nilai 0 sampai 255. Sedangkan untuk citra 10-bit, pada komponen RGB nya memiliki nilai 0-1023. Ini berarti bahwa perkomponen RGB pada citra 10 bit adalah 4 kali sedetail 8 bit.

## **Citra Warna (16 bit)**

Citra warna 16 bit (biasanya disebut sebagai (highcolor) dengan setiap pixelnya diwakili dengan 2 byte memory (16 bit). Warna 16 bit memiliki 65.536 warna.

## **Citra Warna (24 bit)**

Setiap pixel dari citra warna 24 bit diwakili dengan 24 bit sehingga memiliki total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualisasikan seluruh warna yang dapat dilihat penglihatan manusia. Penglihatan manusia dipercaya hanya dapat membedakan hingga 10 juta warna saja. Setiap titik informasi pixel (RGB) disimpan ke dalam 1 byte data. Pada 8 bit pertama menyimpan nilai biru, kemudian diikuti dengan nilai hijau pada 8 bit kedua dan pada 8 bit terakhir merupakan warna merah.

## **2.3 Visi Komputer**

Visi Komputer adalah transformasi data dari kamera menjadi keputusan atau representasi baru. Semua transformasi tersebut dilakukan untuk mencapai beberapa tujuan tertentu. Visi komputer merupakan gabungan dari beberapa ilmu bidang seperti ilmu komputer, teknik elektro, matematika, fisiologi, dan ilmu kognitif serta kombinasi perangkat lunak dan perangkat keras pada komputer sehingga komputer dapat meniru penglihatan manusia.

Visi komputer memiliki banyak fungsi pendukung untuk mendeteksi kemampuan manusia dalam menangkap informasi dan data. Fungsi – fungsi pendukung tersebut antara lain:

- Proses penangkapan citra (*image acquisition*)

- Proses pengolahan citra (*image processing*)
- Analisa data citra (*image analysis*)
- Proses pemahaman data citra (*image understanding*)

## 2.4 *Image Processing*

Pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran berbentuk citra. Teknik pengolahan citra menggunakan komputer untuk mendigitasi pola bayangan dan warna dari gambar yang sudah tersedia. Informasi yang telah terdigitasi ini kemudian ditransfer ke layar dari monitor video. Pengolahan citra banyak digunakan dalam dunia fotogra misalnya mengubah intensitas cahaya sebuah foto, dunia perfilman misalnya animasi, dunia kedokteran misalnya untuk membuat analisa medis, dan dunia *game* [5].

## 2.5 Python

Python adalah salah satu bahasa pemrograman tingkat tinggi yang bersifat interpreter, interaktif, object-oriented dan dapat beroperasi di hampir semua platform, seperti keluarga UNIX, Mac, Windows, dan lainnya [6].



**Gambar 2.9:** Logo Python

Sebagai bahasa tingkat tinggi, Python termasuk salah satu bahasa pemrograman yang mudah untuk dipelajari karena sintaks yang jelas dan elegan, dikombinasikan dengan penggunaan module-module siap pakai dan struktur data tingkat tinggi yang efisien.

## 2.6 OpenCV

OpenCV (*Open Source Computer Vision*) merupakan *library open-source* yang memiliki algoritma-algoritma visi komputer. OpenCV memiliki struktur yang modular, yang berarti pada *package* ter-

dapat beberapa *library* statis [7]. Modul-modul yang tersedia adalah sebagai berikut:

- *Core functionality* (core) – modul padat yang menjelaskan struktur data dasar, termasuk Mat array multi-dimensi dan fungsi dasar yang digunakan oleh modul lainnya.
- *Image Processing* (imgproc) – sebuah modul *image processing* termasuk filter gambar linear dan non-liner, transformasi gambar geometri (resize, affine dan perspektif pembengkokan, mapping generic berbasis table), konversi jarak warna, histogram dan lainnya.
- *Video Analysis* (video) – sebuah modul analisis termasuk estimasi gerak, pengurangan background, dan algoritma tracking objek.
- *Camera Calibration and 3D Reconstruction* (calib3d) – algoritma multiple-view geometri dasar, kalibrasi kamera single dan stereo, estimasi pose objek, algoritma korespondensi stereo, dan elemen rekonstruksi 3D.
- *2D Features Framework* (features2d) – pendeteksian fitur penting, descriptor dan pencocokan descriptor.
- *Detection* (objdetect) – pendeteksian objek dan kejadian-kejadian dari kelas yang telah ditentukan sebelumnya (contohnya, wajah, mata, cangkir, manusia, mobil dan lainnya).
- *High-level GUI* (highgui) – interface yang mudah digunakan pada kemampuan UI yang sederhana.
- Video I/O (videoio) – interface yang mudah digunakan untuk mengambil video dan codec video.

## 2.7 Facial Landmark

Mendeteksi *facial landmark* merupakan salah satu bagian dari prediksi bentuk. Gambar atau video digunakan sebagai *input* (dan biasanya ROI yang memiliki spesifikasi objek yang ingin dideteksi), pendeteksi bentuk kemudian akan mencoba untuk melakukan alokasi poin-poin objek disepanjang bentuknya [8]. Mendeteksi *facial landmark* dibagi menjadi 2 langkah yaitu:

1. Melakukan alokasi wajah di gambar
2. Melakukan deteksi dari struktur utama wajah ROI.

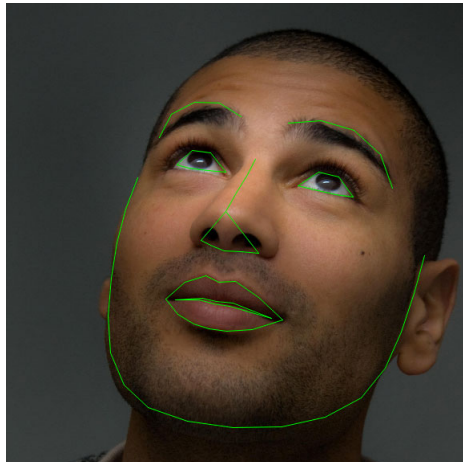
Langkah pertama dapat dilakukan dengan menggunakan *li-*



*brary* OpenCV yang bernama Haar cascades. Langkah pertama juga dapat dilakukan menggunakan algoritma berbasis *deep learning* khusus untuk alokasi wajah. Metode apapun yang digunakan tidak menjadi masalah karena hal yang terpenting untuk melakukan alokasi wajah adalah untuk mendapat *bounding box* wajah (contohnya koordinat-(x,y) wajah pada gambar).

Setelah bagian-bagian wajah ditemukan, dilanjutkan ke langkah kedua. Terdapat beberapa jenis pendeteksi *facial landmark*, tetapi inti dari semua metode yang ada adalah untuk menentukan bagian-bagian pada wajah sebagai berikut:

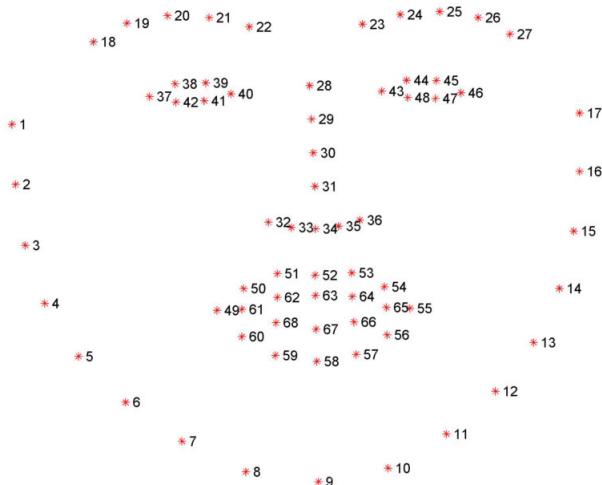
- Mulut
- Alis kanan
- Alis kiri
- Mata kanan
- Mata kiri
- hidung
- Rahang



**Gambar 2.10:** Contoh Facial Landmark

Pendeteksi *facial landmark* yang belum melakukan *training* didalam library dlib digunakan untuk melakukan estimasi dari 68

buah koordinat- (x, y) yang membentuk struktur wajah. Indeks dari 68 buah koordinat dapat divisualisasikan oleh gambar seperti berikut:



**Gambar 2.11:** Facial Landmark

Anotasi-anotasi tersebut merupakan bagian dari dataset pendeteksi *facial landmark* yang telah melakukan *training* [9].

## 2.8 Dlib C++ Library

Dlib adalah toolkit C++ modern yang berisi algoritma *machine learning* dan alat untuk menyelesaikan masalah menggunakan bahasa C++. Dlib ini digunakan baik di industri dan akademik dalam berbagai kondisi termasuk robotika, *embedded system*, ponsel, dan komputasi lainnya yang membutuhkan kinerja tinggi dan besar. Dlib termasuk *open source* sehingga dapat digunakan secara free [10]. Beberapa algoritma *machine learning* yang terdapat pada Dlib antara lain:

- *Conventional SMO Based Support Vector Machines for Classification and Regression*
- *Reduced-Rank Methods for Large Scale Classification and Regression*

- *Relevance Vector Machines for Classification and Regression*
- *General Purpose Multiclass Classification Tools*
- *Online Kernel RLS Regression Algorithm*
- *Online SVM Classification Algorithm*

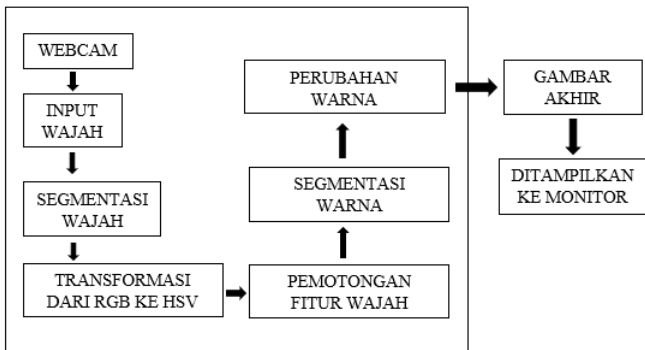
# BAB 3

## DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur dan kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan.

### 3.1 Desain Sistem

Tugas akhir ini merupakan salah satu bentuk dari pemanfaatan OpenCV untuk menampilkan virtual make-up.



Gambar 3.1: Flowchart Sistem

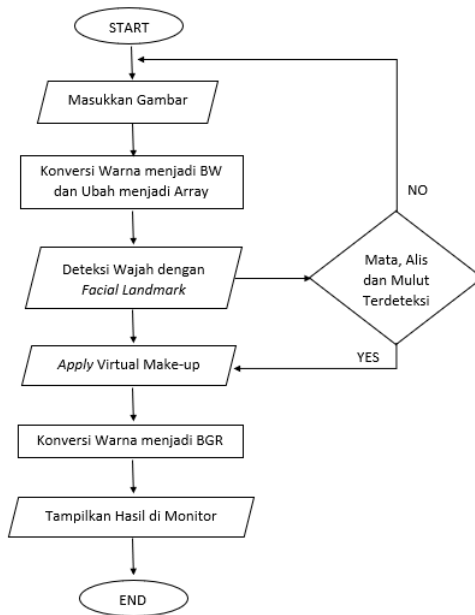
Proses dimulai dengan input wajah dari webcam. Segmentasi wajah kemudian dilakukan dengan melakukan pemotongan fitur-fitur wajah, seperti jarak mata, hidung, mulut dan alis. Setelah fitur-fitur wajah dikenali, warna dari gambar wajah akan ditransformasi dari RGB (*Red, Green, Blue*) menjadi HSV (*Hue, Saturation, Value*) untuk memudahkan pengenalan warna wajah dan fitur-fiturnya tanpa tergantung pada bayangan dan pencahayaan. Metode ini banyak digunakan pada pengenalan wajah dikarenakan struktur wajah yang beragam, tidak rata dan banyaknya bayangan

pada wajah yang membuat seakan-akan ada perbedaan warna pada wajah.

Segmentasi warna untuk virtual make-up dilakukan dengan menentukan nilai RGB dari warna make-up yang diinginkan. Perubahan warna pada fitur wajah yang dipilih menandakan virtual make-up telah berhasil diaplikasikan pada wajah.

### 3.2 Alur Kerja Sistem

Pengerjaan tugas akhir ini dibagi menjadi beberapa tahapan berdasarkan metodologi penelitian. Berikut merupakan alur proses dengan menggunakan masukkan gambar.



**Gambar 3.2:** Alur Proses Virtual Make-up

#### 3.2.1 Persiapan Data Gambar

Monitor yang dilengkapi oleh kamera beresolusi 720p akan melakukan deteksi wajah dengan cara memindai gambar dan mencari

area-area yang terdapat bagian wajah. Namun deteksi wajah dianggap kurang akurat jika wajah menggambarkan berbagai emosi yang membuat letak mata, mulut berubah dari tempat semula. Oleh karena itu untuk dapat menganalisa detil wajah dan agar dapat memperbesar bagian wajah yang diinginkan seperti mata, hidung dan mulut dibutuhkan titik-titik (*Facial Landmark*).



**Gambar 3.3:** Logitech 720p

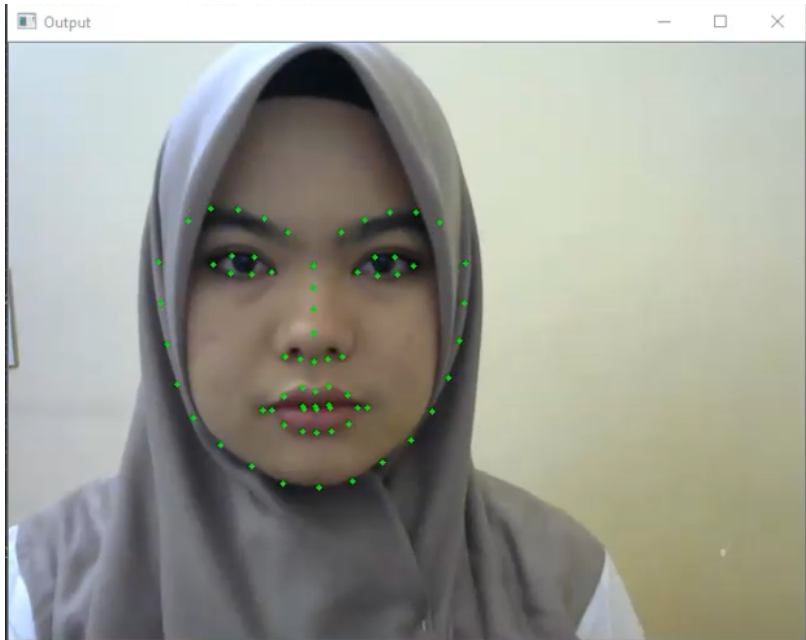
Kamera yang digunakan adalah Logitech dengan resolusi 720p.

SPESIFIKASI LOGITECH 720p	
Resolusi	720p
Frame Rate	30fps
Resolusi Sensor	3MP
Field of View	60°
Host Connection	USB Type-A USB 2.0
Tipe Fokus	Auto

**Tabel 3.1:** Spesifikasi Kamera Webcam

### 3.2.2 Deteksi *Facial Landmark*

Poin-poin *Facial Landmark* akan diletakkan pada gambar wajah yang telah dideteksi dengan menggunakan *library* pada Dlib. *Library* pada Dlib telah memiliki pendeteksi *facial landmark* yang terdiri atas 68 buah titik untuk mendeteksi rahang, dagu, alis, hidung, mata dan bibir.



**Gambar 3.4:** Facial Landmark dari depan

Titik-titik ini akan digunakan untuk menentukan *Region of Interest* (ROI) pada fitur wajah tertentu yang akan diberi warna baik didalam koordinat maupun diluar titik koordinat yang akan menjadi Virtual Make-up. Warna didalam koordinat seperti pada bagian bibir akan menjadi lipstick dan warna yang ada diluar, atau tepatnya, bagian atas koordinat pada mata akan menjadi *eyeliner* atau *eyeshadow Range Point* digunakan sebagai koordinat pada *Facial Landmark*.

Fitur Wajah	Range Point
Rahang kiri	0-7
Dagu	8
Rahang kanan	9-16
Alis kiri	17-21
Alis kanan	22-26
Batang hidung	27-30
Bawah hidung	31-35
Mata kiri	36-41
Mata kanan	42-47
Garis luar bibir	48-59
Garis dalam bibir	60-67

**Tabel 3.2:** Titik-titik pada Facial Landmark

### 3.2.3 Segmentasi Fitur Wajah

Segmentasi dilakukan dengan melakukan pemetaan mata, hidung, dan mulut dari gambar wajah. Hal ini dapat dilakukan dengan menggunakan *library* imutils. Setelah wajah terdeteksi, kemudian dilakukan *loop* di ROI fitur wajah secara terpisah.



**Gambar 3.5:** Data Gambar Wajah





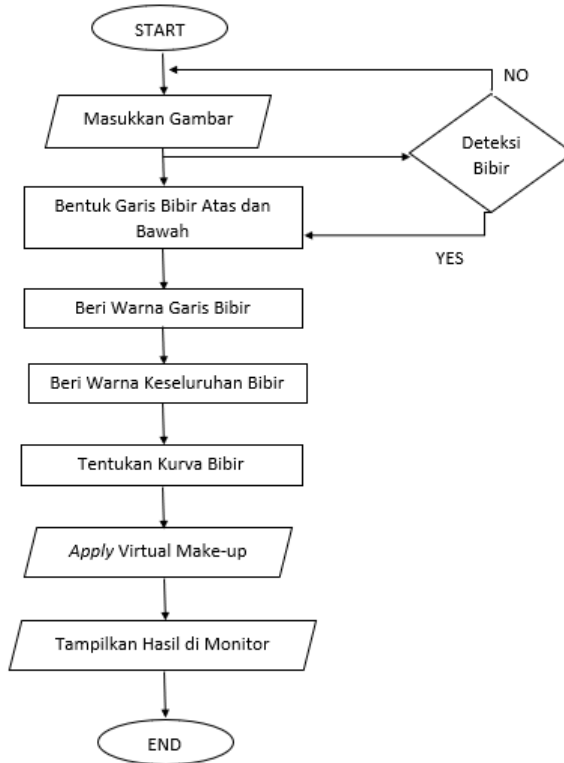
**Gambar 3.6:** Segmentasi Bagian Mata

Langkah-langkah deteksi ROI dimulai dengan loop gambar atau video secara live. Dengan menggunakan Haarcascade, dilakukan ekstrak komputasi wajah dalam bentuk kotak pada koordinat  $(x,y)$  yang berhubungan dengan bagian yang diinginkan, pada contoh kali ini bagian mata, dan kemudian menggunakan Numpy *array* untuk memotong dan ekstraksi.

Kemudian pilih warna untuk dapat di isi pada ROI.

### **3.2.4 Aplikasi Make-up**

Proses setelah *face landmarking* dan ekstraksi bagian wajah tertentu dapat dilakukan adalah penambahan virtual make-up berupa lipstik, dan *eyeliner*.



**Gambar 3.7:** Alur Proses Virtual Make-up Lipstik

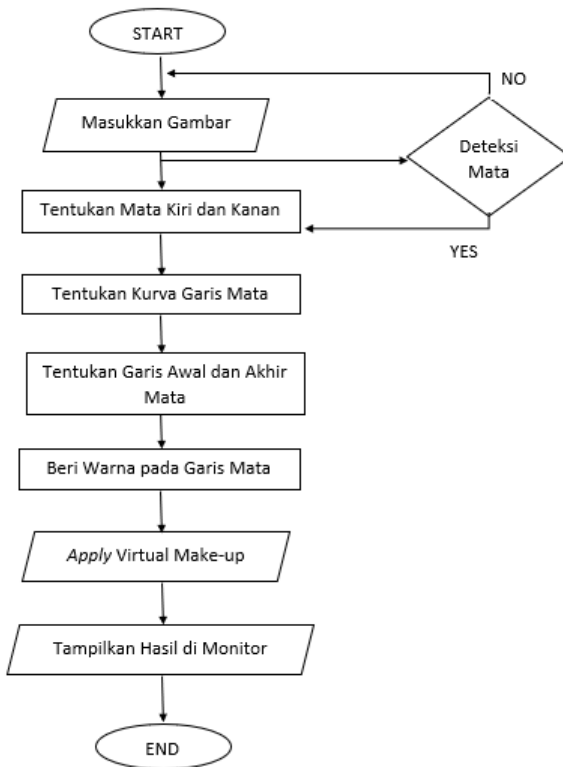
Alur proses pemakaian lipstik dimulai dengan memasukkan gambar yang akan ditambahkan virtual make-up. *Facial Landmarking* kemudian akan memulai proses pendeteksian bagian wajah. Pada proses penggunaan lipstik ini bagian wajah yang akan dideteksi dan diproses hanya bagian bibir. Jika bibir tidak dapat dideteksi atau tidak terlihat pada gambar maka proses akan kembali ke awal.

Kemudian kurvatur dari bentuk bibir akan dibentuk untuk menentukan bagian atas dan bawah. Penentuan ini berdasarkan 68 buah titik *facial landmarking* yaitu bagian bibir atas ada pada titik 49-55 dan 61-65, sedangkan bibir bawah dimulai dari titik 56-60 dan

66-68.

Warna pada garis bibir kemudian akan diaplikasikan pada gambar. Setelah warna garis bibir berhasil diaplikasikan, isi dari bibir kemudian juga diberi warna sama dengan warna garis bibir. Kurvatur bibir akan dipastikan penempatannya untuk memastikan warna telah diletakkan pada bibir secara tepat.

Virtual make-up akan berhasil saat semua langkah-langkah di atas telah dilewati. Hasil yang didapat adalah warna lipstick pada bibir dan hasil akan ditampilkan pada monitor.



**Gambar 3.8:** Alur Proses Virtual Make-up Eyeliner

Alur proses pada penggunaan *eyeliner* hampir sama dengan

proses penggunaan lipstik. Proses dimulai dengan memasukkan gambar. *Facial landmarking* kemudian akan melakukan deteksi wajah dan pada alur ini difokuskan untuk mendeteksi mata kiri dan kanan. Berdasarkan 68 titik *facial landmarking*, mata kiri dimulai dari titik 36-41 dan mata kanan dimulai dari titik 42-47.

Setelah garis mata diketahui, langkah selanjutnya adalah menambahkan warna pada garis mata bagian atas dengan warna yang diinginkan. Saat warna berhasil diletakkan pada mata, hasil virtual make-up akan ditampilkan di monitor dalam bentuk *eyeliner* berwarna.



**Gambar 3.9:** Tanpa Make-up      **Gambar 3.10:** Setelah Make-up

*Halaman ini sengaja dikosongkan*

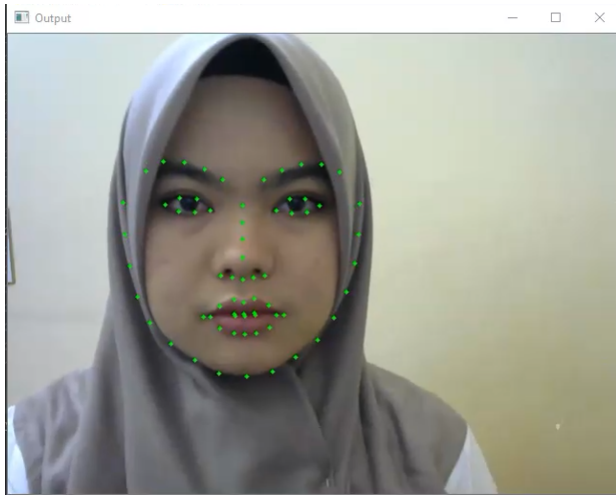
## BAB 4

# PENGUJIAN DAN ANALISA

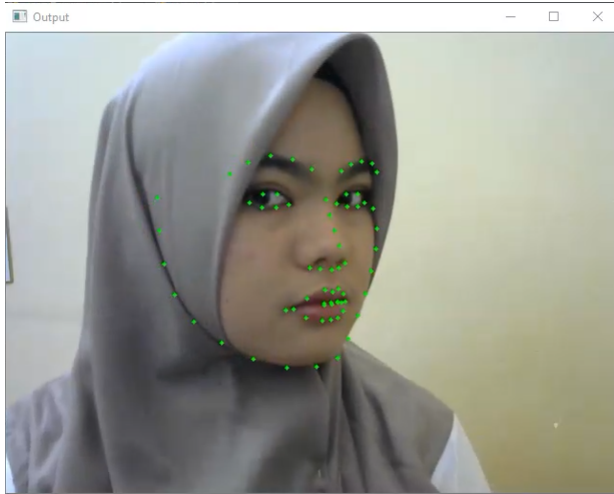
Pada bab ini dipaparkan hasil pengujian serta analisa dari desain sistem dan implementasi. Data yang digunakan dalam. Pengujian yang dilakukan di bagi menjadi beberapa bagian yaitu sebagai berikut:

### 4.1 Aplikasi *Facial Landmark*

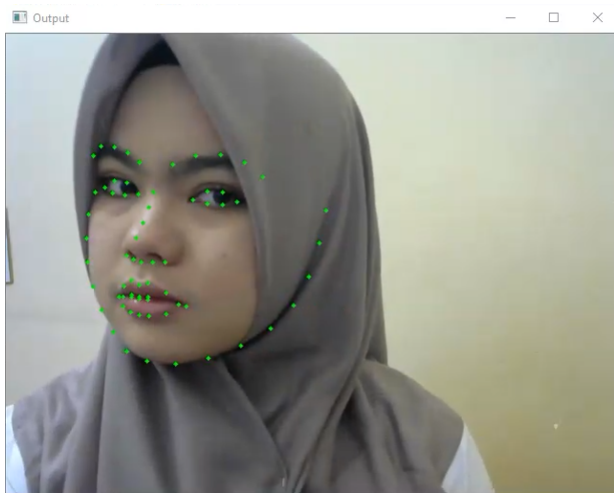
Pada pengujian titik-titik *facial landmark* didapatkan hasil sebagai berikut:



**Gambar 4.1:** Facial Landmark Depan



**Gambar 4.2:** Facial Landmark Kanan



**Gambar 4.3:** Facial Landmark Kiri

Berdasarkan titik-titik *facial landmark*, dilakukan juga pengujian banyaknya titik yang tampak pada derajat wajah tertentu.

Contohnya jika wajah diputar ke kiri atau ke kanan.

Perputaran Wajah	Banyak Titik
10°	68 titik
20°	68 titik
30°	68 titik
40°	68 titik
50°	68 titik
60°	68 titik
70°	68 titik
80°	0 titik
90°	0 titik

**Tabel 4.1:** Titik-titik pada Facial Landmark

Pada percobaan perputaran wajah dapat disimpulkan bahwa titik-titik tetap akan menghitung dan dapat mendeteksi semua titik yaitu 68 buah kecuali saat posisi wajah berpaling 90°. Posisi wajah yang hanya terlihat bagian samping sepenuhnya tidak akan terdeteksi oleh *facial landmark*. *Facial landmark* hanya dapat digunakan untuk mendeteksi wajah selama rahang kiri dan rahang kanan terlihat oleh kamera. Jika bagian mata atau bibir tidak terlihat, *facial landmark* masih bisa memberi asumsi terhadap posisinya. Tetapi jika salah satu rahang tidak terlihat *facial landmark* tidak dapat memberi asumsi sehingga seluruh titik diwajah juga hilang.

Peletakkan titik dan pengenalan wajah dengan metode *facial landmarking* termasuk akurat dan masih bisa dideteksi asalkan posisi wajah tidak berpaling 90°, dimana hanya terlihat profil samping wajah.

## 4.2 Virtual Make-up menggunakan Gambar

Percobaan pertama dilakukan dengan menggunakan *input* gambar. Proses dimulai dengan menggunakan *library* pada OpenCV untuk membaca dan melakukan penempatan titik-titik facial landmark. Pada percobaan ini digunakan *library* *dlib shape predictor* dan *frontal face detector*.



```

PREDICTOR_PATH = "shape_predictor_68_face_landmarks.dat"
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(PREDICTOR_PATH)

image = cv2.imread('image2.png')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
image = np.array(image)
faces = detector(image, 1)

```

**Gambar 4.4:** Kode untuk Virtual Make-up pada Gambar

Langkah selanjutnya adalah membaca *file* gambar dan untuk memudahkan pemberian warna, format *default* BGR diubah menjadi RGB untuk menghindari kebingungan saat pengisian warna. Kemudian gambar diubah menjadi *array* dengan menggunakan numpy dan wajah akan dideteksi menggunakan *frontal face detector*.

```

for face in faces:
    landmarks = predictor(image, face)
    landmarks=face_utils.shape_to_np(landmarks)
    a=ApplyMakeup(landmarks)
    image=a.apply_lipstick(image, 185, 88, 97)#rgb
    image=a.apply_liner(image)

```

**Gambar 4.5:** Kode untuk Virtual Make-up pada Gambar

Kemudian *shape predictor* digunakan untuk menentukan *facial landmarking* pada gambar dalam bentuk *array*. *Apply lipstick* dan *apply make-up* merupakan *class* tersendiri yang telah diatur algoritma nya untuk menampilkan lipstik dan *eyeliner* pada gambar wajah. Warna lipstik dan eyeliner dapat ditentukan dengan memasukkan nilai RGB yang diinginkan]. Kemudian nilai RGB tersebut akan diletakkan pada bibir dan mata pada wajah di gambar.

Hasil dari virtual make-up pada wajah dapat di *blend* lumayan sempurna dan akurat sesuai dengan titik-titik poin dari *Face Landmarking* dengan syarat posisi wajah menunjukkan kedua rahang.



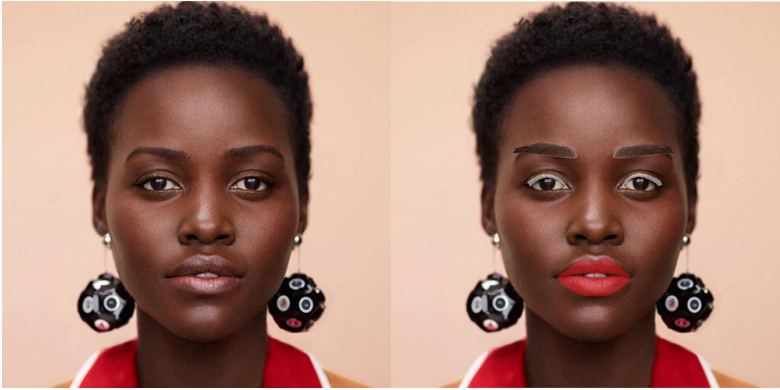
**Gambar 4.6:** Sebelum Make-up

**Gambar 4.7:** Sesudah Make-up



**Gambar 4.8:** Sebelum Make-up

**Gambar 4.9:** Sesudah Make-up



**Gambar 4.10:** Sebelum Make-up    **Gambar 4.11:** Sesudah Make-up



**Gambar 4.12:** Sebelum Make-up    **Gambar 4.13:** Sesudah Make-up



**Gambar 4.14:** Sebelum Make-up   **Gambar 4.15:** Sesudah Make-up

Jika gambar wajah tampak samping digunakan, dimana posisi kedua rahang tidak terlihat, *facial landmark* tidak dapat menentukan fitur wajah secara akurat, dan penempatan virtual make-up juga tidak sesuai dengan posisi yang seharusnya.



**Gambar 4.16:** Sebelum Make-up   **Gambar 4.17:** Sesudah Make-up



**Gambar 4.18:** Sebelum Make-up    **Gambar 4.19:** Sesudah Make-up

### 4.3 Virtual Make-up secara *Real Time*

Metode virtual make-up secara *real time* ini memiliki konsep program yang sama dengan metode menggunakan gambar, namun karena virtual make-up diaplikasikan secara *real time* sehingga membutuhkan proses yang lebih berat dan hal ini mengakibatkan terjadinya *lagging* pada video yang cukup signifikan. Namun diluar kekurangan tersebut, warna dan aplikasi dapat digunakan secara akurat pada titik-titik wajah.

```
print("[INFO] starting video stream...")
video_capture = WebcamVideoStream(src=1).start()

while True:
    frame = video_capture.read()
    original=frame
    faces = detector(frame, 1)
```

**Gambar 4.20:** Kode untuk Virtual Make-up pada Video

Proses dimulai dengan melakukan *capture* melalui *webcam*. Ke-

mudian gambar dimasukkan kedalam variabel *faces* yang akan menggunakan *library* pada dlib *frontal face detector* untuk mendeteksi wajah pada video.

```
for face in faces:
    landmarks = predictor(frame, face)
    landmarks = face_utils.shape_to_np(landmarks)

    a=ApplyMakeup(landmarks)
    frame=a.apply_lipstick(frame, 19, 14, 99)#bgr
    frame=a.apply_liner(frame)

cv2.imshow('Video', frame)
cv2.imshow('Original',original)
```

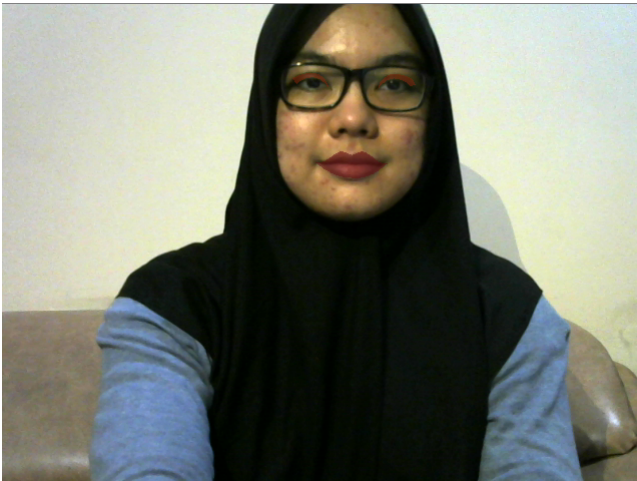
**Gambar 4.21:** Kode untuk Virtual Make-up pada Video

Langkah selanjutnya adalah menggunakan *shape predictor* pada wajah di video dan dalam bentuk *array* untuk memudahkan pengenalan. *Class* untuk melakukan aplikasi make-up kemudian diberikan pada gambar, yaitu pemberian dalam bentuk lipstik dan *eyeliner* yang warnanya dapat diberikan sesuai nilai RGB katalog make-up yang diinginkan.





**Gambar 4.22:** Sebelum Make-up



**Gambar 4.23:** Sebelum Make-up

Penggunaan kacamata tidak berpengaruh pada ketepatan aplikasi virtual make-up dengan syarat fitur wajah yang lain tetap ter-

lihat jelas dan dapat dikenali oleh *facial landmarking*.



**Gambar 4.24:** Sebelum Make-up   **Gambar 4.25:** Sesudah Make-up



**Gambar 4.26:** Sebelum Make-up   **Gambar 4.27:** Sesudah Make-up



**Gambar 4.28:** Sebelum Make-up   **Gambar 4.29:** Sesudah Make-up





**Gambar 4.30:** Contoh Warna

**Gambar 4.31:** Contoh Warna

Dengan menggunakan color picker pada aplikasi seperti Corel Draw dan Photoshop, akan didapatkan nilai RGB dari masing-masing warna lipstik yang kemudian akan di masukkan kedalam program dan kemudian diaplikasikan pada wajah secara *real time* maupun hanya lewat gambar.

# BAB 5

## PENUTUP

### 5.1 Kesimpulan

Dalam penelitian ini, telah diimplementasikan Virtual Make-up secara *real time* dan melalui gambar menggunakan *Visual Face Landmarking*, didapatkan kesimpulan sebagai berikut:

1. Untuk mendapatkan hasil video dengan kualitas yang memuaskan pada virtual Make-up secara *real time* harus menggunakan komputer dengan GPU yang cukup kuat dan tinggi dikarenakan beratnya beban pada proses program yang digunakan.
2. Untuk pengaplikasian *blush on* yaitu warna kemerahan pada pipi dibutuhkan teknologi *machine learning* tingkat tinggi dikarenakan pada metode *Visual Face Landmarking* hanya menggunakan 68 titik poin pada wajah yang tidak termasuk bagian pipi sehingga sulit untuk dilakukan pengisian warna pada ROI tersebut.
3. 68 buah titik yang merupakan dasar dari proses *facial landmarking* dapat digunakan secara akurat pada wajah di gambar maupun secara *real time* dengan syarat wajah harus berada pada derajat dibawah  $80^\circ$  dimana kedua buah rahang dapat terlihat. Jika posisi wajah tidak sesuai dengan syarat diatas maka tingkat keakuratan *facial landmarking* menurun sehingga hasil make-up virtual pada wajah tidak sesuai dengan tempat seharusnya atau pada *worst case scenario* make-up virtual tidak terlihat atau tidak dapat diaplikasikan sama sekali pada wajah.
4. Penggunaan aksesoris pada wajah seperti kacamata atau hijab tidak menurunkan tingkat keakuratan virtual make-up berbasis (*facial landmarking*) dengan syarat yaitu posisi wajah sesuai dan fitur wajah tidak tertutupi.

5. Aplikasi make-up menggunakan metode *filter* seperti pada sosial media relatif ringan karena memiliki metode hanya meletakkan gambar yang sudah tersedia pada fitur wajah tertentu, sedangkan virtual make-up berbasis *facial landmarking* memiliki metode dimana warna secara manual diaplikasikan pada fitur wajah seperti bibir dan mata.

## 5.2 Saran

Demi pengembangan lebih lanjut mengenai tugas akhir ini, disarankan beberapa langkah lanjutan sebagai berikut:

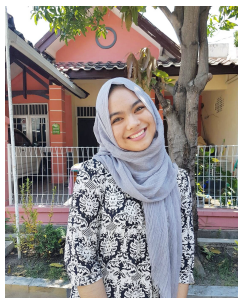
1. Sebaiknya menggunakan komputer dengan GPU yang tinggi sehingga hasil virtual make-up secara *real time* dapat lebih lancar dan menghindari *lagging* yang berlebihan
2. Dengan menggunakan komputer yang memiliki spesifikasi lebih bagus, maka make-up yang lebih lengkap seperti penggunaan *blush on* dan *countour* pada seluruh wajah dapat ditambahkan pada make-up dengan menggunakan metode *machine learning* yang memiliki titik atau poin fitur pengenalan wajah diatas 68 buah titik *facial landmarking*.

## DAFTAR PUSTAKA

- [1] R. K. dan Alan Novi Tompunu, "Pengolahan citra digital untuk mendeteksi obyek menggunakan pengolahan warna model normalisasi rgb," Seminar Nasional Teknologi Informasi dan Komunikasi Terapan, 2011. (Dikutip pada halaman 5).
- [2] E. Sutoyo, T. dan Mulyanto, "Teori pengolahan citra digital," Yogyakarta, Penerbit ANDI, 2009. (Dikutip pada halaman 6).
- [3] P. Darma, "Citra digital dan ekstraksi fitur," Yogyakarta: Graha Ilmu, 2010. (Dikutip pada halaman 7).
- [4] R. C. Gonzales, "*Digital Image Processing*," New Jersey: Prentice-Hall, Inc., 2002. (Dikutip pada halaman 8).
- [5] B. d. K. F. Achmad, "Teknik pengolahan citra digital menggunakan delphi," Yogyakarta, Penerbit ARDI, 2005. No citations.
- [6] <https://www.python.org/doc/essays/blurblurb/>. terakhir diakses tanggal 4 Agustus 2020 9.48 PM. (Dikutip pada halaman 12).
- [7] <https://docs.opencv.org/2.4/modules/core/doc/core.html>. terakhir diakses tanggal 4 Agustus 2020 9.40 PM. (Dikutip pada halaman 13).
- [8] N. L. Ulla Delfana Rosiani, Rosa Andrie Asmara, "Penerapan facial landmark point untuk klasifikasi jenis kelamin berdasarkan citra wajah," JIP (Jurnal Informatika Polinema), 2020. (Dikutip pada halaman 13).
- [9] S. Kumar, "*Face Recognition using OpenFace*." <https://medium.com/analytics-vidhya/face-recognition-using-openface-92f02045ca2a>. terakhir diakses tanggal 4 Agustus 2020 8:30 PM. (Dikutip pada halaman 15).
- [10] L. R.-M. d. A. L. Jair Cervantes, Farid Garcia-Lamont, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," Neurocomputing, 2019. (Dikutip pada halaman 15).

*Halaman ini sengaja dikosongkan*

## BIOGRAFI PENULIS



Dyah Retno Arumsari, lahir pada tanggal 22 Agustus 1995 di Dumai, Riau. Penulis merupakan anak pertama dari 3 bersaudara dari Bapak Yanus Supantoro dan Ibu Endang Sriwahyuni. Saat ini penulis berdomisili di Tangerang Selatan, Banten. Pada tahun 2007 penulis menyelesaikan pendidikan SD di SD 1 YKPP Bukit Datuk di Dumai, provinsi Riau. Tahun 2010 penulis lulus dari SMP Negeri 2 Dumai. Untuk Pendidikan menengah keatas, penulis melanjutkan studi di SMA Negeri 1 Dumai dan lulus di tahun 2013. Penulis sedang menjalani studi di Teknik Komputer, Fakultas Teknologi Elektro dan Informatika Cerdas di Institut Teknologi Sepuluh Nopember Surabaya. Selama perkuliahan, penulis aktif sebagai anggota Media Informasi di International Office ITS 2015/2016 dan UKM CLICK ITS. Untuk kritik, saran atau pertanyaan mengenai tugas akhir ini, dapat menghubungi penulis di [eno.arumsari@gmail.com](mailto:eno.arumsari@gmail.com)

*Halaman ini sengaja dikosongkan*