



**ANALISIS GERAK BIBIR DALAM PENGUCAPAN HURUF  
VOKAL BERDASARKAN *FACIAL LANDMARKS DETECTION*  
MENGUNAKAN METODE *REGRESSION TREES***

**TUGAS AKHIR**

**Program Studi  
S1 Teknik Komputer**

INSTITUT BISNIS  
DAN INFORMATIKA

**stikom**  
SURABAYA

**Oleh:**

**Abil Khosim**

**15410200044**

---

---

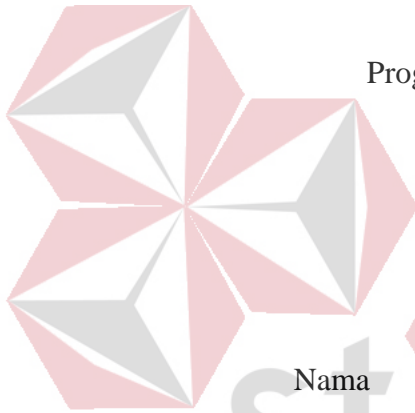
**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA  
2019**

**ANALISIS GERAK BIBIR DALAM PENGUCAPAN HURUF  
VOKAL BERDASARKAN *FACIAL LANDMARKS DETECTION*  
MENGUNAKAN METODE *REGRESSION TREES***

**TUGAS AKHIR**

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



**Disusun Oleh:**

Nama : Abil Khosim

NIM : 15.41020.0044

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

**2019**

*“Ketika mereka menikmati mimpi, kami memperjuangkan mimpi.”*

**Abil Khosim**

INSTITUT BISNIS  
DAN INFORMATIKA

**stikom**  
SURABAYA

**Dipersembahkan kepada Bapak, Ibu dan Keluarga saya yang selalu memotivasi dan memberikan doa yang terbaik kepada saya, dosen pembimbing 3 saya Andrea Hartoko, angkatan 15 Teknik Komputer yang selalu solid, Bapak/Ibu WB yang selalu mendampingi saya ketika diwarung, jajaran para mantan, dan pacar saya kelak di masa mendatang serta semua orang yang selalu membantu, mendukung dan memotivasi agar terus belajar dan berkarya lebih baik dari hari kemarin.**

**TUGAS AKHIR**  
**ANALISIS GERAK BIBIR DALAM PENGUCAPAN HURUF VOKAL**  
**BERDASARKAN *FACIAL LANDMARKS DETECTION* MENGGUNAKAN**  
**METODE *REGRESSION TREES***

Dipersiapkan dan disusun oleh

**Abil Khosim**

**NIM : 15410200044**

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

Pada : Juli 2019

**Susunan Dewan Penguji**

Pembimbing

**I. Dr. Susijanto Tri Rasmana, S.Kom., M.T.**

NIDN. 0727097302

**II. Heri Pratikno, M.T., MTCNA., MTCRE.**

NIDN. 0716117302

Pembahas

**I. Weny Indah Kusumawati, S.Kom., M.MT.**

NIDN. 0721047201

Tugas Akhir ini telah diterima sebagai salah satu persyaratan



untuk memperoleh gelar Sarjana  
FAKULTAS TEKNOLOGI  
DAN INFORMATIKA

**stikom**  
SURABAYA

**Dr. Jusak**

**Dekan Fakultas Teknologi dan Informatika**

**INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

**SURAT PERNYATAAN**  
**PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya:

Nama : Abil Khosim  
NIM : 15.41020.0044  
Program Studi : S1 Teknik Komputer  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Tugas Akhir  
Judul Karya : **ANALISIS GERAK BIBIR DALAM  
PENGUCAPAN HURUF VOKAL  
BERDASARKAN FACIAL LANDMARKS  
DETECTION MENGGUNAKAN METODE  
REGRESSION TREES**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, Juli 2019

Yang menyatakan  
  
Abil Khosim  
NIM: 15.41020.0044



## ABSTRAK

Komunikasi merupakan salah satu kunci pokok dalam bermasyarakat. Berbagai cara dan metode dapat digunakan agar bisa terjalin sebuah komunikasi. Salah satunya adalah dengan membaca ujaran, membaca ujaran merupakan suatu metode komunikasi secara verbal dalam memahami pembicaraan orang lain dengan melihat gerakan bibirnya. Dalam proses membaca ujaran diperlukan kunci untuk dapat mengenali kata yaitu huruf vokal. Biasanya membaca ujaran digunakan untuk berkomunikasi oleh penyandang tuna rungu. Hal ini dikarenakan kaum tuna rungu mempunyai keterbatasan dalam mendengar suara, sehingga memanfaatkan penglihatan untuk membaca gerakan bibir lawan bicara.

Berdasarkan hal tersebut maka dalam Tugas Akhir ini penulis membuat gagasan untuk menganalisis sistem pengolahan citra yang akan mengkonversi gerak bibir menjadi sebuah teks, dalam proses pengerjaan Tugas Akhir ini dibatasi hanya untuk huruf vokal saja. Perubahan gerak bibir menjadi teks ini dapat dilakukan melalui perekaman wajah secara *real-time* dengan proses pengolahan citra digital.

Sistem yang dibuat pada penelitian ini menunjukkan bahwa dapat mendeteksi pengucapan huruf vocal berdasarkan *facial landmarks detection* menggunakan metode *regression trees* dengan baik. Hasil pengujian, sistem dapat mendeteksi pengucapan huruf vokal A, I, dan U dengan akurasi sebesar 100%, huruf vokal E dengan akurasi sebesar 82% dan huruf vokal O dengan akurasi sebesar 68% dari 10 responden dengan 5 kali pengambilan data setiap pengucapan huruf vokal.

Kata Kunci: Huruf Vokal, Gerak Bibir, *Facial Landmarks Detection*, *Regression Trees*.

## KATA PENGANTAR

Atas berkat rahmat Tuhan Yang Maha Esa, maka penulis berhasil menyelesaikan dan menyusun naskah Tugas Akhir yang berjudul Analisis Gerak Bibir Dalam Pengucapan Huruf Vokal Berdasarkan *Facial Landmarks Detection* Menggunakan Metode *Regression Trees*. Tugas Akhir ini merupakan salah satu syarat yang harus ditempuh oleh setiap mahasiswa untuk meraih gelar kesarjanaan di Jurusan Teknik Komputer, Fakultas Teknologi dan Informatika, Institut Bisnis dan Informatika Stikom Surabaya.

Dalam mengerjakan Tugas Akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Budi Jatmiko, M.Pd., selaku Rektor Institut Bisnis dan Informatika Stikom Surabaya.
2. Bapak Dr. Jusak, selaku Dekan Fakultas Teknologi dan Informatika Institut Bisnis dan Informatika Stikom Surabaya.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Prodi S1 Teknik Komputer Institut Bisnis dan Informatika Stikom Surabaya.
4. Bapak Dr. Susijanto Tri Rasmana, S.Kom., M.T., selaku dosen pembimbing pertama yang telah memberikan arahan dan bimbingan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.



5. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku dosen pembimbing kedua yang telah memberikan arahan dan bimbingan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.
6. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku dosen pembahas yang telah memberikan arahan kepada penulis untuk menyelesaikan Tugas Akhir dengan baik.
7. Seluruh dosen pengajar Program Studi S1 Teknik Komputer Institut Bisnis dan Informatika Stikom Surabaya yang telah mendidik dan memberikan motivasi kepada penulis sehingga dapat menyelesaikan Tugas Akhir dengan baik.
8. Rekan-rekan mahasiswa S1 Teknik Komputer dan semua pihak yang telah memberikan dorongan dan bantuan baik secara langsung maupun secara tidak langsung.

Akhir kata, penulis berharap semoga segala sesuatu yang telah dihasilkan dalam pelaksanaan Tugas Akhir ini dapat bermanfaat.

Surabaya, Juli 2019

Penulis

## DAFTAR ISI

	Halaman
ABSTRAK .....	vii
DAFTAR ISI .....	x
DAFTAR GAMBAR .....	xiv
DAFTAR TABEL .....	xvii
DAFTAR LAMPIRAN .....	xviii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Sistematika Penulisan .....	3
BAB II LANDASAN TEORI .....	5
2.1 Huruf Vokal .....	5
2.2 <i>Histogram of Oriented Gradients</i> .....	7
2.3 <i>Facial Landmarks Detection</i> .....	7
2.4 <i>Regression Trees</i> .....	8
2.4.1 Langkah-Langkah Metode <i>Regression Trees</i> .....	10
2.4.2 Aspek Rasio Bibir .....	11
2.5 <i>Dlib C++ Library</i> .....	12
2.6 <i>Computer Vision</i> .....	12

2.7	<i>Open Computer Vision (OpenCV)</i> .....	13
2.8	<i>Python</i> .....	14
2.9	<i>Raspberry Pi</i> .....	15
2.9.1	<i>Raspberry Pi 3 Model B</i> .....	16
2.9.2	<i>GPIO Raspberry Pi 3 Model B</i> .....	17
2.10	<i>Pi Camera</i> .....	19
2.11	<i>Monitor</i> .....	19
BAB III METODOLOGI PENELITIAN.....		21
3.1	<i>Metode Penelitian</i> .....	21
3.2	<i>Perancangan Perangkat Keras</i> .....	22
3.3	<i>Perancangan Program</i> .....	24
3.3.1	<i>Flowchart Sistem Deteksi Gerak Bibir</i> .....	24
3.2.1	<i>Flowchart Metode Regression Trees</i> .....	25
3.4	<i>Deteksi Wajah</i> .....	31
3.5	<i>Deteksi Bibir</i> .....	32
3.6	<i>Deteksi Pengucapan Huruf Vokal</i> .....	34
3.6.1	<i>Pengambilan Nilai Setiap Node Berdasarkan Value Intensitas Piksel</i> ...	34
3.6.2	<i>Pengklasifikasian Range Node Untuk Setiap Huruf Vokal</i> .....	37
BAB IV HASIL DAN PEMBAHASAN .....		41
4.1	<i>Pengujian Raspberry Pi 3 Model B</i> .....	41
4.1.1	<i>Tujuan Pengujian Raspberry Pi 3 Model B</i> .....	41
4.1.2	<i>Peralatan Pengujian Raspberry Pi 3 Model B</i> .....	41
4.1.3	<i>Prosedur Pengujian Raspberry Pi 3 Model B</i> .....	42

4.1.4 Hasil Pengujian <i>Raspberry Pi 3 Model B</i> .....	43
4.2 Pengujian <i>Pi Camera</i> .....	44
4.2.1 Tujuan Pengujian <i>Pi Camera</i> .....	44
4.2.2 Peralatan Pengujian <i>Pi Camera</i> .....	44
4.2.3 Prosedur Pengujian <i>Pi Camera</i> .....	44
4.2.4 Hasil Pengujian <i>Pi Camera</i> .....	46
4.3 Pengujian Tingkat Akurasi Deteksi Bibir .....	47
4.3.1 Tujuan Pengujian Tingkat Akurasi Deteksi Bibir .....	47
4.3.2 Peralatan Pengujian Tingkat Akurasi Deteksi Bibir .....	47
4.3.3 Prosedur Pengujian Tingkat Akurasi Deteksi Bibir .....	48
4.3.4 Hasil Pengujian Tingkat Akurasi Deteksi Bibir .....	48
4.4 Pengujian Tingkat Akurasi Pengucapan Huruf Vokal .....	51
4.4.1 Tujuan Pengujian Tingkat Akurasi Pengucapan Huruf Vokal .....	51
4.4.2 Peralatan Pengujian Tingkat Akurasi Pengucapan Huruf Vokal ..	51
4.4.3 Prosedur Pengujian Tingkat Akurasi Pengucapan Huruf Vokal ...	51
4.4.4 Pengambilan Batas <i>Range</i> Untuk Uji Tingkat Akurasi Pengucapan Huruf Vokal Secara <i>Online (Realtime)</i> .....	53
4.4.5 Hasil Pengujian Tingkat Akurasi Pengucapan Huruf Vokal Secara <i>Online (realtime)</i> .....	54
4.4.6 Batas Ambang / Penyimpangan Pada Uji Deteksi Pengucapan Huruf Vokal .....	78
BAB V KESIMPULAN .....	85
5.1 Kesimpulan .....	85
5.2 Saran .....	85

DAFTAR PUSTAKA .....	87
BIODATA.....	101



## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Kontur Bentuk Bibir.....	6
Gambar 2.2 Penanda 68 Koordinat Bentuk Wajah .....	8
Gambar 2.3 Struktur <i>Regression Trees</i> .....	9
Gambar 2.4 <i>Node Facial Landmark</i> Pada Bibir.....	12
Gambar 25 Logo <i>Python</i> .....	15
Gambar 2.6 Logo <i>Raspberry Pi</i> .....	15
Gambar 2.7 <i>Raspberry Pi</i> 3 Model B.....	17
Gambar 2.8 <i>Raspberry Pi</i> GPIO Pin.....	18
Gambar 2.9 Fungsi GPIO <i>Raspberry Pi</i> 3 Model B .....	18
Gambar 2.10 <i>Pi Camera</i> .....	19
Gambar 2.11 <i>Monitor Raspberry Pi</i> .....	20
Gambar 3.1 Bagan Metodologi Penelitian.....	21
Gambar 3.2 Blok Diagram Sistem Deteksi Gerak Bibir .....	23
Gambar 3.3 <i>Flowchart</i> Sistem Deteksi Gerak Bibir .....	24
Gambar 3.4 <i>Flowchart Regression Trees</i> .....	26
Gambar 3.5 Data <i>Training</i> 1 .....	28
Gambar 3.6 Citra Gambar Pertama.....	29
Gambar 3.7 Citra Gambar Hasil .....	29
Gambar 3.8 Deteksi Wajah .....	31
Gambar 3.9 Deteksi Bibir .....	33

Gambar 3.10 Koordinat Bibir Pada 68 Penanda Koordinat Bentuk Wajah.....	35
Gambar 3.11 Nilai Intensitas Piksel Pada 68 Koordinat.....	36
Gambar 3.12 <i>Node</i> Aspek Rasio Bibir.....	38
Gambar 3.13 <i>Frame</i> Deteksi Wajah, Bibir Dan Aspek Rasio Bibir .....	40
Gambar 4.1 Pemberian Nama <i>File</i> Berekstensi *.py .....	42
Gambar 4.2 Contoh <i>Program</i> Sederhana Pertambahan Dua Bilangan Menggunakan <i>Python</i> .....	43
Gambar 4.3 Hasil <i>Running Program</i> Sederhana Pertambahan Dua Bilangan Menggunakan <i>Python</i> .....	43
Gambar 4.4 Pemasangan <i>Pi Camera</i> ke <i>Raspberry Pi</i> .....	45
Gambar 4.5 Contoh <i>Program</i> Sederhana Rekam Video Menggunakan <i>Pi Camera</i> ...	46
Gambar 4.6 Hasil <i>Running Program</i> Sederhana Rekam Video Menggunakan <i>Pi Camera</i> .....	47
Gambar 4.7 Deteksi Bibir .....	49
Gambar 4.8 Diagram Pengujian Tingkat Akurasi Deteksi Bibir .....	49
Gambar 4.9 Posisi Awal Uji Huruf Vokal .....	52
Gambar 4.10 Beberapa Responden Saat Melakukan Pengucapan Huruf A .....	54
Gambar 4.11 Diagram Pengujian Tingkat Akurasi Pengucapan Huruf Vokal A .....	56
Gambar 4.12 Beberapa Responden Saat Melakukan Pengucapan Huruf I.....	59
Gambar 4.13 Diagram Pengujian Tingkat Akurasi Pengucapan Huruf Vokal I.....	61
Gambar 4.14 Beberapa Responden Saat Melakukan Pengucapan Huruf U .....	64
Gambar 4.15 Diagram Pengujian Tingkat Akurasi Pengucapan Huruf Vokal U .....	66
Gambar 4.16 Beberapa Responden Saat Melakukan Pengucapan Huruf E.....	69
Gambar 4.17 Diagram Pengujian Tingkat Akurasi Pengucapan Huruf Vokal E.....	71

Gambar 4.18 Beberapa Responden Saat Melakukan Pengucapan Huruf O .....	74
Gambar 4.19 Diagram Pengujian Tingkat Akurasi Pengucapan Huruf Vokal O .....	76
Gambar 4.20 Pengucapan Huruf E Terbaca Huruf I Secara Bergantian (Ambigu)....	79
Gambar 4.21 Pengucapan Huruf O Terbaca Huruf U Secara Bergantian (Ambigu)..	80
Gambar 4.22 Pengucapan Huruf O Tidak Terdeteksi (Buruk) .....	82





## DAFTAR TABEL

	Halaman
Tabel 3.1 Perhitungan <i>Threshold With Dlib</i> .....	30
Tabel 3.2 Tabel Nilai Intensitas Piksel 20 <i>Node</i> Dari Satu Responden .....	37
Tabel 4.1 Tabel Pengujian Tingkat Akurasi Deteksi Bibir .....	50
Tabel 4.2 Klasifikasi <i>Range</i> Pengucapan Huruf Vokal .....	53
Tabel 4.3 Batas <i>Range</i> Pengucapan Huruf Vokal A .....	54
Tabel 4.4 Pengujian Tingkat Akurasi Pengucapan Huruf Vokal A .....	56
Tabel 4.5 Batas <i>Range</i> Pengucapan Huruf Vokal I .....	59
Tabel 4.6 Batas <i>Range</i> Pengucapan Huruf Vokal A .....	64
Tabel 4.7 Pengujian Tingkat Akurasi Pengucapan Huruf Vokal U .....	66
Tabel 4.8 Batas <i>Range</i> Pengucapan Huruf Vokal E .....	69
Tabel 4.9 Pengujian Tingkat Akurasi Pengucapan Huruf Vokal E .....	71
Tabel 4.10 Batas <i>Range</i> Pengucapan Huruf Vokal O .....	74
Tabel 4.11 Pengujian Tingkat Akurasi Pengucapan Huruf Vokal O .....	76
Tabel 4.12 Pengucapan Huruf Vokal O Ambigu .....	81
Tabel 4.13 Pengucapan Huruf Vokal O Buruk .....	83

## DAFTAR LAMPIRAN

	Halaman
LAMPIRAN 1. Hasil Uji Tingkat Sensitif Gerakan Bibir .....	88
LAMPIRAN 2. Hasil Uji Tingkat Akurasi Pengucapan Huruf Vokal.....	91
LAMPIRAN 3. Listing Program Sistem Deteksi Pengucapan Huruf Vokal .....	96



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Komunikasi merupakan salah satu kunci pokok dalam bermasyarakat. Kondisi fisik bukan lagi menjadi sebuah hambatan yang signifikan. Berbagai cara dan metode dapat digunakan agar bisa terjalin sebuah komunikasi. Salah satunya adalah dengan membaca ujaran, membaca ujaran merupakan suatu metode komunikasi secara verbal dalam memahami pembicaraan orang lain dengan melihat gerakan bibirnya. Dalam proses membaca ujaran diperlukan kunci untuk dapat mengenali kata yaitu huruf vokal. Biasanya membaca ujaran digunakan untuk berkomunikasi oleh penyandang tuna rungu. Hal ini dikarenakan kaum tuna rungu mempunyai keterbatasan dalam mendengar suara, sehingga memanfaatkan penglihatan untuk membaca gerakan bibir lawan bicara (Febriana, 2015).

Berdasarkan hal tersebut maka dalam Tugas Akhir ini penulis membuat gagasan untuk menganalisis sistem pengolahan citra yang akan mengkonversi gerak bibir menjadi sebuah teks, dalam proses pengerjaan Tugas Akhir ini dibatasi hanya untuk huruf vokal saja. Perubahan gerak bibir menjadi teks ini dapat dilakukan melalui perekaman wajah secara *real-time* dengan proses pengolahan citra digital.

Sampai saat ini sudah banyak penelitian yang mengarah pada permasalahan ini, khususnya deteksi gerakan bibir menggunakan pengolahan citra digital menjadi teks. Salah satu penelitian sebelumnya adalah deteksi gerak bibir dengan menggunakan

perangkat (*software*) Matlab 7.8.0. metode *Gray Level Co-occurrence* (GLCM) oleh Reski Novitriani dari Universitas Telkom (Novitriani, 2012).

Adapun penelitian lainnya deteksi gerak bibir menggunakan kurva Bezier dengan EmguCV menggunakan perangkat (*software*) Microsoft Visual Studio 2008 dan dengan bahasa pemrograman *Csharp* oleh Ratna Purwati Lakuary dari Universitas Gunadarma (Lakuary, 2015).

Berdasarkan permasalahan diatas maka penelitian selanjutnya akan dibuat sebuah analisis gerak bibir dalam pengucapan huruf vokal berbasis *facial landmarks detection* menggunakan metode *regression trees* dengan *Raspberry Pi*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka dirumuskan permasalahan:

1. Bagaimana proses deteksi bibir dengan berbasis *facial landmark detection* menggunakan metode *regression trees* ?
2. Bagaimana menganalisis fitur gerak bibir dalam pengucapan huruf vokal ?

## 1.3 Batasan Masalah

Untuk menghindari pembahasan yang lebih luas terkait dengan perancangan deteksi gerak bibir menggunakan kamera pada *Raspberry Pi*, maka penelitian ini ditentukan pada ruang lingkup sebagai berikut:

1. Perekaman menggunakan kamera *Raspberry Pi* dengan ukuran 5MP.
2. Bibir yang diteliti adalah bibir normal dan tidak cacat.

3. Menggunakan pengucapan yang normal (*logat*, ekspresi) sesuai dengan kaidah bahasa Indonesia.
4. Huruf yang diteliti hanya huruf vokal saja yakni a, i, u, e, o.
5. Kondisi pengambilan data di pencahayaan yang terang dengan menggunakan pencahayaan yang cukup tepat didepan objek.
6. Berjarak  $\pm 100$  cm dari kamera dan posisi wajah berada dalam *rectangle*.
7. Objek yang dideteksi adalah wajah dan mulut yang tidak terhalang apapun.
8. Kondisi awal dari deteksi adalah saat muncul *notification* “anda diam”.

#### 1.4 Tujuan

Berdasarkan rumusan masalah yang diuraikan diatas, maka tujuan penelitian ini adalah sebagai berikut:

1. Proses deteksi bibir berbasis *facial landmark detection* menggunakan metode *regression trees*.
2. Melakukan analisis fitur gerak bibir yang akan digunakan dalam pengucapan huruf vokal.

#### 1.5 Sistematika Penulisan

Pembahasan Tugas Akhir ini disusun menjadi 5 (lima) garis besar bab pembahasan, yaitu sebagai berikut:

## **BAB I        PENDAHULUAN**

Pada bab ini menyajikan pembahasan, mengenai latar belakang, rumusan masalah, batasan masalah, tujuan dari, dan sistematika penulisan.

## **BAB II        LANDASAN TEORI**

Pada bab ini disajikan teori penunjang dari permasalahan, diantaranya huruf vokal, *facial landmarks detection*, *regression trees*, *computer vision*, *OpenCV*, *Raspberry Pi*.

## **BAB III       METODE PENELITIAN**

Pada bab ini disajikan mengenai tahapan perancangan perangkat lunak deteksi gerak bibir. Dijelaskan proses pembuatan sub-sub program dan metode percobaan yang digunakan untuk menguji dan analisis gerak bibir.

## **BAB IV        HASIL DAN PEMBAHASAN**

Pada bab ini membahas tentang meliputi pengujian *raspberry pi 3* model B, pengujian *pi camera*, pengujian tingkat akurasi deteksi bibir, pengujian tingkat akurasi pengucapan setiap huruf vokal.

## **BAB V        PENUTUP**

Pada bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan serta saran untuk pengembangan penelitian selanjutnya.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Huruf Vokal**

Huruf vokal adalah bunyi ujaran akibat adanya udara yang keluar dari paru-paru tidak terkena hambatan atau halangan. Jumlah huruf vokal ada 5, yaitu a, i, u, e, dan o. Huruf vokal ini memiliki karakter, bunyi dan ciri masing masing baik dalam cara pengucapan dan perlakuannya (upi.edu, 2005).

a. **Vokal A**

- Dibunyikan dengan menjatuhkan rahang bawah sejauh mungkin, bukan dengan membuka ke samping.
- Gigi atas dan bawah tidak dilindungi atau ditutupi oleh kedua bibir atas dan bawah, bibir atas dan bawah harus kelihatan.
- Lidah diletakkan rata serta ujungnya menyentuh gigi bawah.

b. **Vokal I**

- Dengan tetap mempertahankan posisi rongga mulut dan tenggorokan pada saat mengucapkan vokal A dan E.
- Pada saat mengucapkan vokal I dikurangi sepertiga atau dua pertiga dari A dengan sedikit mendorong sudut bibir kesebelah dalam.

c. **Vokal U**

- Seperti pada posisi pengucapan vokal O.
- Mulut lebih dipersempit sedikit, dan bibir sedikit lebih didorong kedepan.

- Dapatkan vokal U yang sempurna, seperti pada kata busur dan sungguh.
- Bunyi tidak didominasi bunyi U yang utuh melainkan terkesan lebih boros dalam hal nafas.

d. Vokal E

- Sama dengan posisi pada saat mengucapkan A tetapi dengan mengurangi luasnya mulut sepertiga dari ucapan A.
- Kedudukan gigi atas dan bawah tetap dijaga.
- Rongga mulut dan tenggorokan tetap dipertahankan seperti pada vokal A.

e. Vokal O

- Berawal dari bentuk vokal A tetapi merubah bentuk bibir menjadi lonjong.
- Seperti corong, vokal O dapat dibunyikan dengan manis. Diupayakan sebulat atau selonjong mungkin.



Gambar 2. 1 Kontur Bentuk Bibir (upi.edu, 2005)

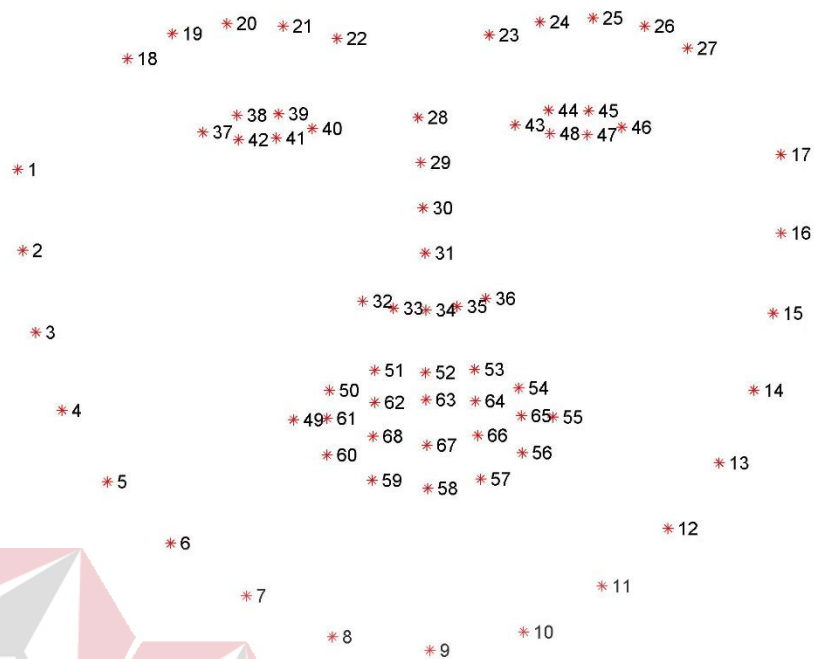


## 2.2 *Histogram of Oriented Gradients*

*Histogram of Oriented Gradients* (HOG) adalah salah satu metode yang digunakan dalam *image processing* untuk mendeteksi suatu objek pada citra gambar. Suatu objek dapat direpresentasikan dengan baik berdasarkan bentuk, untuk memperoleh informasi maka citra gambar akan dibagi menjadi beberapa *cell* dan setiap *cell*-nya akan dihitung sebagai *histogram of oriented gradients*. Setiap piksel dalam *cell* sangat berpengaruh saat melakukan *voting* bobot untuk membangun sebuah histogram yang berorientasi pada seluruh nilai gradien yang dihitung (Raut, 2018).

## 2.3 *Facial Landmarks Detection*

*Facial Landmarks Detection* adalah salah satu contoh keluaran terstruktur yang bertujuan untuk memprediksi bentuk geometri yang diperoleh dari sebuah data berupa citra wajah. *Facial Landmarks* adalah satu *set point* penting yang ada pada citra wajah manusia. Jumlah *landmark* bergantung pada *dataset* atau aplikasi yang digunakan (Raut, 2018). Salah satu model *facial landmarks* yaitu 68 *landmarks* ditunjukkan pada Gambar 2.2.



Gambar 2. 2 Penanda 68 Koordinat Bentuk Wajah (Raut, 2018)

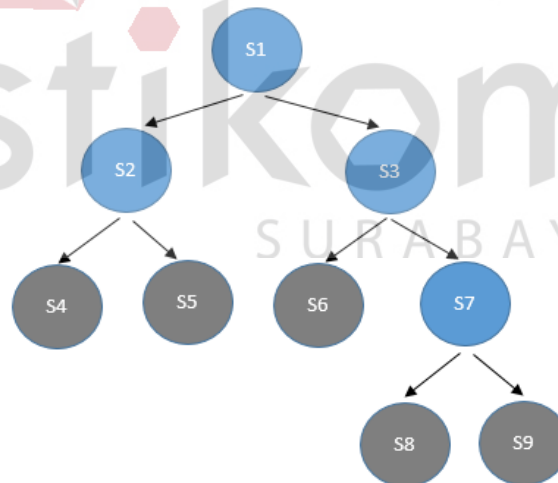
## 2.4 Regression Trees

*Classification and Regression Trees* (CART) adalah salah satu metode atau algoritma dari teknik pohon keputusan. CART adalah suatu metode statistik nonparametrik yang dapat menggambarkan hubungan antara variabel respon (variabel dependen) dengan satu atau lebih variabel prediktor (variabel independen). Menurut Breiman (1993), apabila variabel respon berbentuk kontinyu maka metode yang digunakan adalah metode regresi pohon (*regression trees*), sedangkan apabila variabel respon memiliki skala kategorik maka metode yang digunakan adalah metode klasifikasi pohon (*classification trees*). Variabel respon dalam penelitian ini berskala kontinu, sehingga metode yang akan digunakan adalah metode *regression trees*.

Metode *regression trees* dikenal sebagai metode percabangan rekursif biner, dimana kumpulan data yang disebut simpul selalu dicabang menjadi dua cabang yang disebut simpul anak. Menurut Breiman (1993), bagian utama dari *regression trees* adalah sebagai berikut:

1. Percabangan setiap simpul.
2. Penetapan simpul akhir.
3. Penentuan nilai dengan dugaan respon bagi setiap simpul akhir.

Pada Gambar 2.3 struktur *regression trees* memiliki satu simpul akar yang dinyatakan dengan S1 yang mengandung seluruh gugus data. Simpul dalam dilambangkan dengan S2, S3, S4, sedangkan simpul akhir dilambangkan dengan S4, S5, S6, S8, S9. Bentuk dari struktur *regression trees* ditunjukkan pada Gambar 2.3.



Gambar 2. 3 Struktur *Regression Trees*

### 2.4.1 Langkah-Langkah Metode *Regression Trees*

#### A. Aturan Percabangan

Menurut Breiman (1993), *regression trees* dibentuk melalui percabangan seluruh gugus data dimana didalamnya memiliki sederetan percabangan biner sehingga menghasilkan beberapa simpul akhir. Aturan untuk percabangan suatu simpul menjadi dua simpul anak sebagai berikut:

1. Setiap percabangan bergantung pada nilai yang berasal dari satu *variable* penjelas.
2. Untuk *variable* kontinyu  $X_j$ , percabangan yang digunakan adalah  $X_j \leq c$ , dimana  $c$  adalah median antara dua nilai pengamatan *variable*  $X_j$  secara berurutan. Jadi jika  $X_j$  memiliki nilai  $n$  yang berbeda, maka akan ada  $n-1$  percabangan.

#### B. Proses Percabangan

Menurut Breiman (1993), proses percabangan pada tiap simpul adalah dengan cara sebagai berikut:

1. Menentukan semua kemungkinan percabangan pada setiap *variable* penjelas.
2. Memilih percabangan terbaik dari masing-masing *variable* penjelas serta menentukan percabangan terbaik dari kumpulan percabangan terbaik tersebut.

### C. Aturan Penghentian

Setiap percabangan di cabang menjadi dua bagian, dua bagian tersebut disebut dengan simpul anak kiri dan simpul anak kanan. Proses percabangan diulang kembali terhadap dua simpul anak hingga tidak dapat dicabang. Breiman (1993) menyatakan bahwa proses rekursif berakhir jika banyak pengamatan pada simpul akhir  $\leq 5$ . Simpul yang tak bisa dicabang lagi disebut dengan simpul akhir.

### D. Aturan Penentuan Nilai Dugaan Respon

Pada *regression trees* jika suatu simpul merupakan simpul akhir, maka dugaan respon simpul akhir tersebut adalah rata-rata responnya. Dengan kata lain penentuan nilai dugaan respon bagi setiap simpul akhir adalah dengan menggunakan rata-rata responnya pada simpul terminal tersebut.

#### 2.4.2 Aspek Rasio Bibir

ARB merupakan aspek ratio bibir dimana digunakan untuk menentukan nilai piksel saat membuka dan menutupnya bibir. Penghitungan pada ARB menggunakan intensitas piksel pada titik *facial landmark* di seluruh bagian bibir. Nilai dari setiap *node* aspek rasio bibir diperoleh dari *datasheet* 300-W yang merupakan salah satu *datasheet facial landmarks detection* yang mempunyai 68 *node facial landmark*. Adapun bagian-bagian *node facial landmark* pada bibir ditujukan pada Gambar 2.4.



Gambar 2. 4 *Node Facial Landmark* Pada Bibir

## 2.5 *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*.

Beberapa algoritma *machine learning* yang terdapat pada *Dlib* antara lain:

1. *Reduced-Rank Methods for Large Scale Classification and Regression.*
2. *Relevance Vector Machines for Classification and Regression.*
3. *General Purpose Multiclass Classification Tools.*

## 2.6 *Computer Vision*

*Computer vision* adalah transformasi data dari kamera menjadi keputusan atau representasi baru. Semua transformasi tersebut dilakukan untuk mencapai beberapa tujuan tertentu *Computer vision* 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.

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

1. Proses penangkapan citra (*image acquisition*).
2. Proses pengolahan citra (*image processing*).
3. Analisis data citra (*image analysis*).
4. Proses pemahaman data citra (*image understanding*).

## 2.7 *Open Computer Vision (OpenCV)*

*OpenCV* adalah sebuah *library* yang berisikan banyak *function* pemrograman untuk teknologi *computer vision* secara *realtime* yang bersifat *open source* (bebas digunakan untuk hal yang bersifat akademis maupun komersial). *Library* ini ditulis dalam C dan C ++ dan berjalan di bawah Linux, Windows dan Mac OS X. *OpenCV* ditulis dalam C yang dioptimalkan dan dapat memanfaatkan prosesor *multicore*.

Contoh aplikasi dari *OpenCV* antara lain interaksi manusia dengan komputer, pengenalan wajah, pengenalan gerak, struktur dari gerakan, robotika (Helmirawan, 2012). Berikut fitur–fitur yang terdapat pada *OpenCV* antara lain:

1. Manipulasi data *image* (alokasi, rilis, duplikasi, pengaturan, konversi).
2. *Image* dan I/O video (masukan berbasis *file* dan kamera, keluaran *image/video file*).

3. Manipulasi matriks dan *vector* serta aljabar linier (produk, solusi, *eigenvalues*, SVD).
4. Beragam struktur data dinamis (daftar, baris, grafik).
5. Dasar pengolahan citra (filter, deteksi tepi, deteksi sudut, pengambilan sampel dan interpolasi, konversi warna, operasi morfologi, histogram).
6. Analisis struktur (komponen yang berhubungan, pengolahan kontur, transformasi jarak, variasi momen, *transformasi Hough*, perkiraan *polygonal*, menyesuaikan garis, *delaunay triangulation*).
7. Kalibrasi kamera (menemukan dan menelusuri pola kalibrasi, dasar estimasi matriks, estimasi homografi, responsensi *stereo*).
8. Analisis gerakan (*optical flow*, segmentasi gerakan, penelusuran).
9. Pengenalan objek (metode *eigen*, HMM).
10. *Graphical User Interface* Atau GUI (menampilkan *image/video*, penanganan *mouse* dan *keyboard*, *scroll-bars*).
11. Pelabelan *image* (garis, *polygon*, gambar teks).

## 2.8 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. 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.



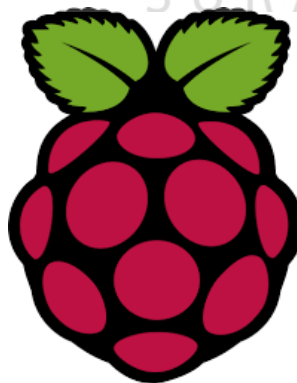


Gambar 2.5 Logo *Python*

(Sumber: <https://www.python.org/static/img/python-logo.png>)

## 2.9 *Raspberry Pi*

*Raspberry Pi* adalah komputer berukuran kartu kredit yang awalnya dirancang untuk pendidikan, terinspirasi oleh *Micro Micro* 1981. Tujuan Pencipta Eben Upton adalah menciptakan perangkat berbiaya rendah yang akan meningkatkan keterampilan pemrograman dan pemahaman perangkat keras di tingkat pra-universitas. Namun berkat ukurannya yang kecil dan harga yang dapat diakses, itu dengan cepat diadopsi oleh para pembuat tinker dan penggemar elektronik untuk proyek yang memerlukan lebih dari mikrokontroler dasar (seperti perangkat Arduino).



Gambar 2.6 Logo *Raspberry Pi*

(Sumber : <https://www.raspberrypi.org/app/uploads/2011/10/Raspi-PGB001.png>)

*Raspberry Pi* memiliki dua model yaitu model A dan model B. Perbedaan antara model A dan B terletak pada penggunaan *memory* dimana model A menggunakan *memory* sebesar 256 MB dan model B sebesar 512 MB. Adapun *ethernet port* yang terdapat pada model B namun tidak terdapat di model A. Desain *Raspberry Pi* didasarkan seputar SoC (*System-on-a-chip*) *Broadcom BCM2835*, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, *VideoCore IV GPU*, dan 256 *Megabyte* RAM (model B). Penyimpanan data menggunakan kartu SD untuk *booting* dan penyimpanan internal dalam jangka panjang. *Hardware Raspberry Pi* tidak memiliki *realtime clock*, sehingga *operating system* harus menggunakan *timer* jaringan *server*. Selain itu, juga dapat menggunakan fungsi *realtime* seperti DS1307 yang disambungkan dengan GPIO melalui *interface I2C (Inter-Integrated Circuit)*. Sistem operasi utama *Raspberry Pi* menggunakan Debian GNU/Linux dengan bahasa pemrograman *Python*. Pengembang *operating system* pada *Raspberry Pi* telah meluncurkan *operating system* yang dinamai Raspbian dimana mampu memaksimalkan perangkat *Raspberry Pi*.

### 2.9.1 *Raspberry Pi 3 Model B*

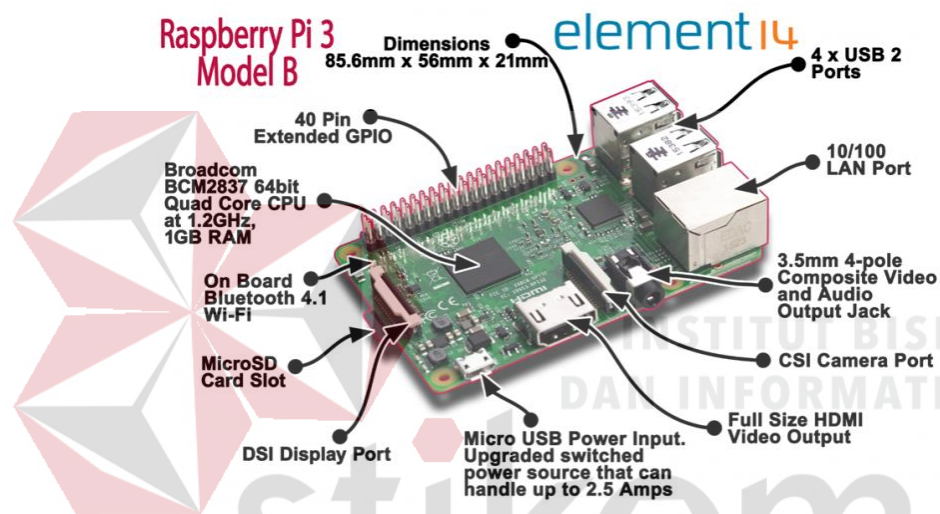
*Raspberry Pi 3 Model B* memiliki RAM 1GB dan grafis *Broadcom VideoCore IV* pada frekuensi *clock* yang lebih tinggi dari sebelumnya yang berjalan pada 250MHz. *Raspberry Pi 3 Model B* menggantikan *Raspberry Pi 2 Model B* pada bulan Februari 2016. Kelebihannya dibandingkan dengan *Raspberry Pi 2* adalah:

1. A 1.2GHz 64-bit *quad-core* ARMv8 CPU
2. 802.11n *Wireless LAN*

### 3. Bluetooth 4.1

### 4. Bluetooth Low Energy (BLE)

*Raspberry Pi 2 dan Raspberry Pi 3 memiliki 4 USB port, 40 pin GPIO, Full HDMI port, Port Ethernet, Combined 3.5mm audio jack and composite video, Camera interface (CSI), Display interface (DSI), slot kartu Micro SD, dan VideoCore IV 3D graphics core.*

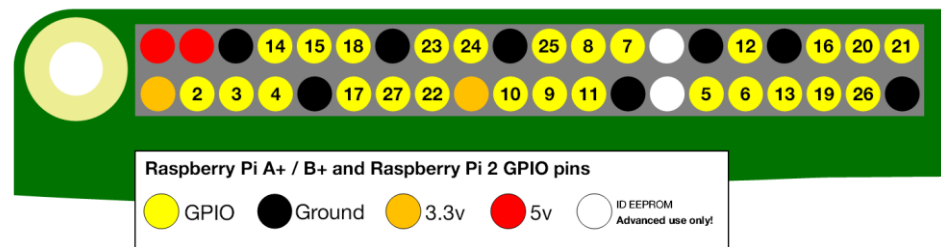


Gambar 2.7 Raspberry Pi 3 Model B

(Sumber : [https://www.element14.com/community/dtss-images/uploads/devtool/diagram/large/Raspberry\\_Pi\\_3\\_Model\\_B\\_with\\_1GB\\_of\\_RAM.png](https://www.element14.com/community/dtss-images/uploads/devtool/diagram/large/Raspberry_Pi_3_Model_B_with_1GB_of_RAM.png))

## 2.9.2 GPIO Raspberry Pi 3 Model B

GPIO merupakan sederet *pin* yang terdiri dari 40 *pin* dengan berbagai fungsi. Salah satu fitur dari *Raspberry Pi* adalah deretan GPIO (*input / output*) *pin* di sepanjang tepi atas pin board yang dimana dapat berfungsi sebagai *switch* untuk mengaktifkan dan menonaktifkan data *input* maupun *output*.



Gambar 2.8 Raspberry Pi GPIO Pin

(Sumber : <https://www.raspberrypi.org/documentation/usage/gpio/images/gpio-numbers-pi2.png>)

Berikut penjelasan mengenai fungsi masing-masing Pin GPIO pada Raspberry Pi 3 Model B.

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1, I <sup>2</sup> C)		DC Power 5v	04
05	GPIO03 (SCL1, I <sup>2</sup> C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)		(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Gambar 2.9 Fungsi GPIO Raspberry Pi 3 Model B

(Sumber : [www.element14.com/RaspberryPi](http://www.element14.com/RaspberryPi))

### 2.10 *Pi Camera*

*Pi Camera* adalah produk resmi dari *Raspberry Pi Foundation*, model 5MP asli dirilis pada tahun 2013, dan 8 MP Modul Kamera v2 dirilis pada tahun 2016. Pada sistem deteksi geraki bibir menggunakan *Pi Camera* yang berukuran 5MP. *Pi Camera* ini menggunakan sensor OmniVision OV5647 dengan sensor resolusi sebesar 2592 x 1944 *pixels*.



Gambar 2.10 *Pi Camera*

### 2.11 *Monitor*

*Monitor* digunakan sebagai *output* dari sistem ini apabila saat sistem ini berjalan memberikan keluaran dari bentuk bibir yang terdeteksi. *Output* ini berupa teks yang huruf vokal yang merupakan hasil dari keluaran pada raspberry pi.



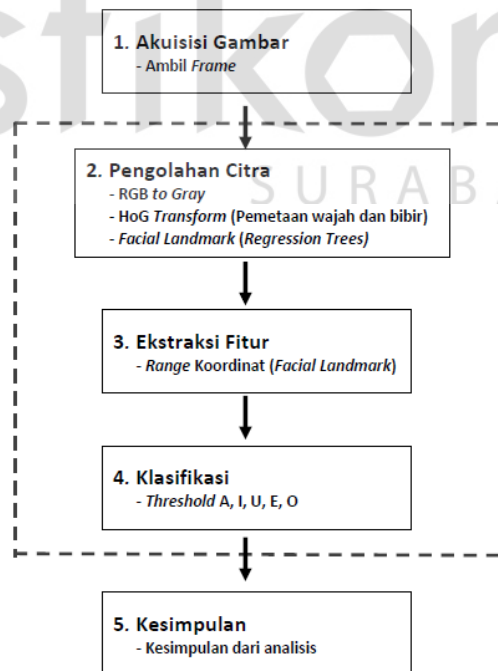
Gambar 2.11 *Monitor Raspberry Pi*

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Metode Penelitian

Metode penelitian yang digunakan pada Tugas Akhir ini melalui beberapa tahap. Dimana tahapan yang pertama kali dilakukan adalah *image acquisition*, yaitu pengambilan *frame* pada saat perekaman video secara *realtime*. Lalu *frame* tersebut yang merupakan citra RGB akan dirubah menjadi citra *grayscale* dan menggunakan metode HOG pada tahap *image processing* untuk memperkecil area komputasi. Setelah itu akan dilakukan *feature extraction* dengan ditambahkan 68 koordinat *facial landmark* serta menggunakan metode *regression trees* untuk memperoleh nilai *threshold* pada gerak bibir.



Gambar 3.1 Bagan Metodologi Penelitian

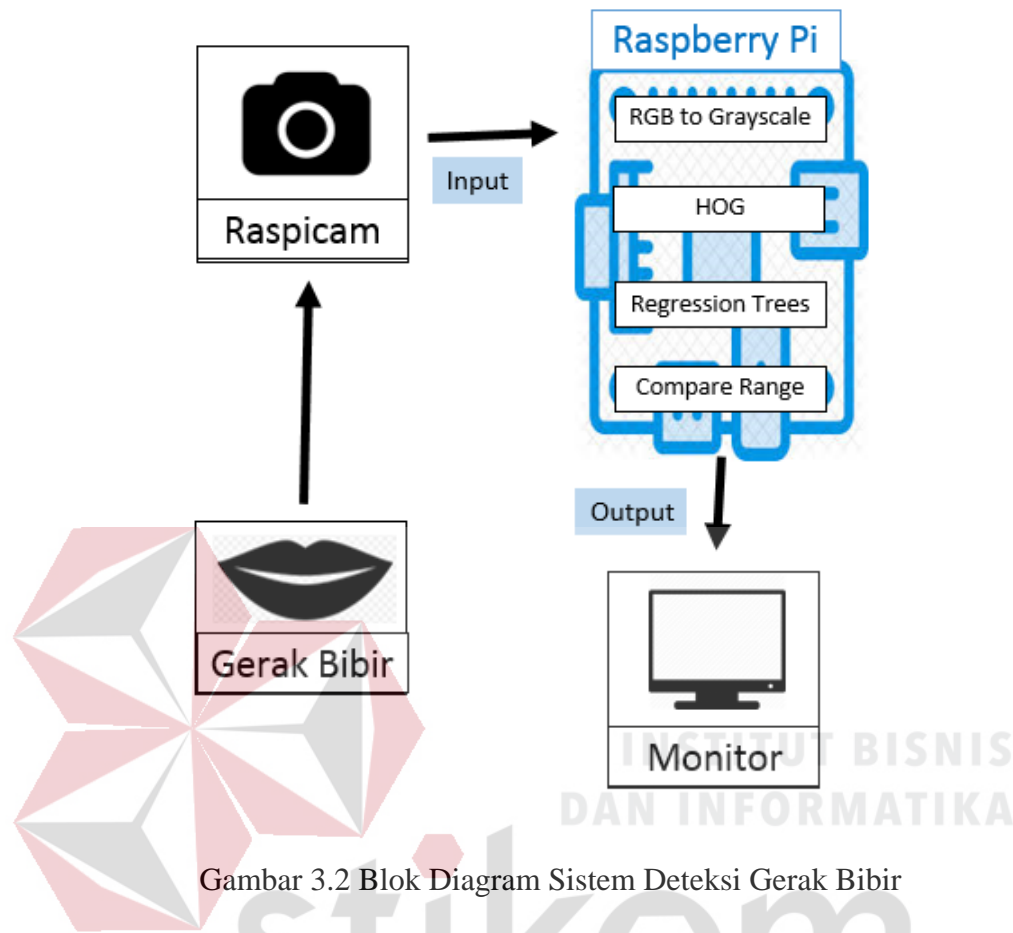
Tahap selanjutnya adalah *classification*, yaitu proses pengklasifikasian nilai *threshold* terhadap pengucapan setiap huruf vokal. Proses terakhir merupakan *conclusion* atau kesimpulan hasil akhir dari proses analisis yang dilakukan. Adapun pengujian kesesuaian data yang diambil oleh *Pi Camera* dan diproses oleh *Raspberry Pi* dilakukan dengan cara:

1. Menjalankan program sistem deteksi gerak bibir kemudian mengambil kesimpulan apakah program sudah sesuai dengan hasil yang diharapkan dan menganalisis perbedaan *value* pada setiap data.
2. Menjalankan program sistem deteksi gerak bibir kepada beberapa responden sebagai *sample* dan mengambil kesimpulan apakah data rata-rata yang diambil sesuai dengan hasil yang diharapkan serta besar selisih dari setiap *value* yang dihasilkan.

### 3.2 Perancangan Perangkat Keras

Perancangan program Tugas Akhir ini merupakan implementasi dari algoritma *regression trees* pada *facial landmarks detection* yang digunakan untuk mendeteksi bentuk bibir. Pendeteksi bentuk bibir tersebut dilakukan oleh *pi camera* yang kemudian data berupa citra gambar akan diproses oleh *raspberry pi* dan akan ditampilkan huruf vokal apa yang tengah diucapkan ke layar *monitor*.





Gambar 3.2 Blok Diagram Sistem Deteksi Gerak Bibir

Dari Gambar 3.2, dapat dijelaskan bahwa data citra gambar yang diperoleh dari kamera akan diproses oleh *Raspberry Pi* menggunakan *Python* dan *OpenCV* dengan metode *regression trees* untuk menentukan koordinat bibir. Setelah menemukan koordinat bibir, maka nilai koordinat bibir akan diolah menggunakan rumus aspek rasio bibir untuk menentukan hasil pergerakan bibir. Apabila hasil aspek rasio bibir yang diperoleh berada di salah satu *range threshold* yang telah ditentukan oleh penulis selama kurun waktu 1 detik, maka *monitor* akan menampilkan huruf vokal apa yang tengah diucapkan oleh responden.

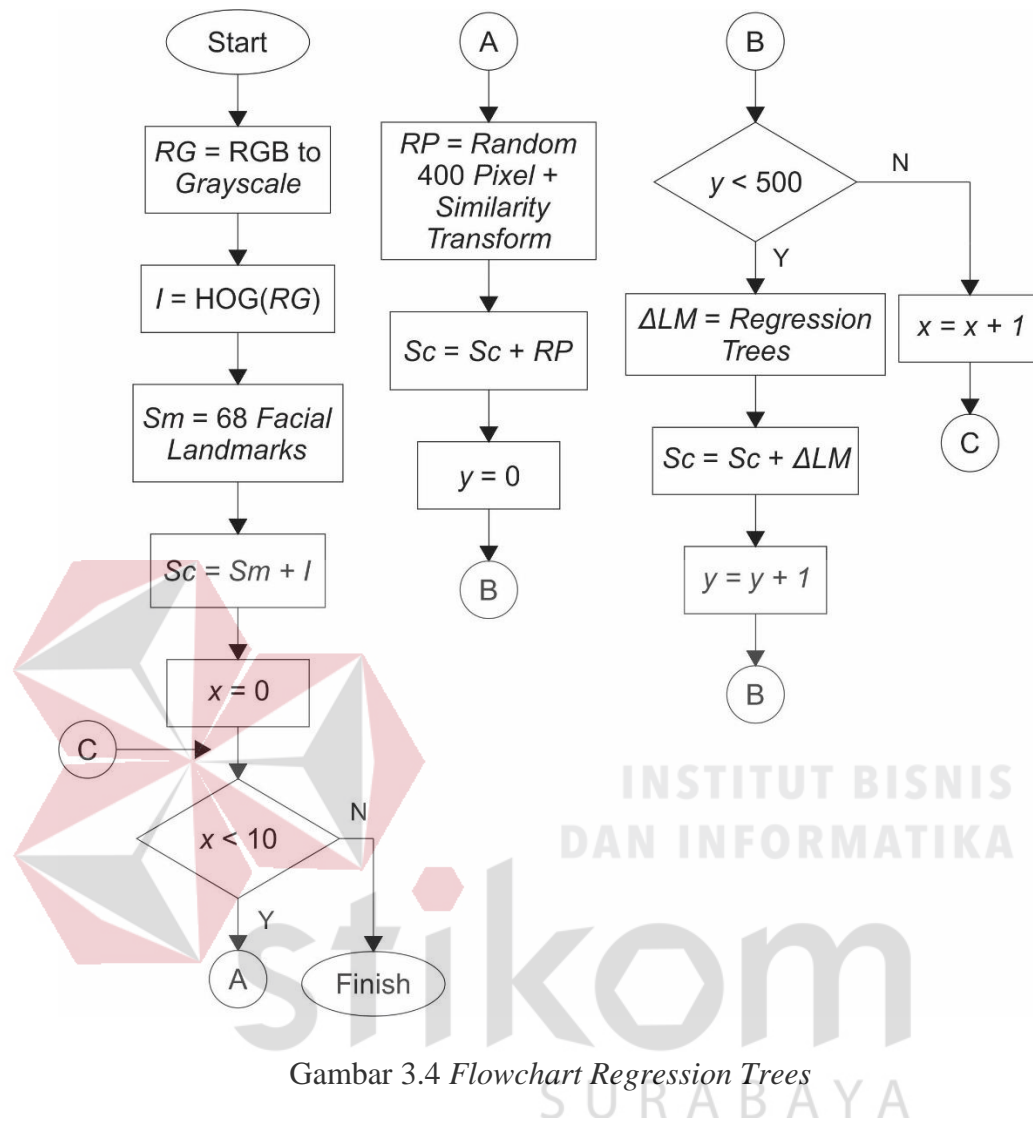


Dari citra gambar *grayscale* akan diolah menggunakan metode *Histogram of Oriented Gradients* untuk memperkecil proses komputasi selanjutnya hasil dari HOG merupakan area wajah dari responden. Batas koordinat tersebut kemudian akan digunakan untuk pembatas pemetaan 68 penanda koordinat bentuk wajah. Kemudian titik–titik koordinat tersebut akan digeser untuk pemetaan bentuk wajah menggunakan *looping regressor* selama beberapa kali. *Looping regressor* ini digunakan untuk memberikan penanda piksel pada citra wajah dengan menggunakan metode *regression trees*.

Kemudian sistem mengambil koordinat bibir pada 68 penanda koordinat bentuk wajah dimana bibir memiliki koordinat 49 hingga 60 untuk bibir bagian luar dan 61 hingga 68 untuk bibir bagian dalam. Nilai intensitas piksel pada koordinat (x, y) bibir akan diolah menggunakan *value range* dari aspek rasio bibir untuk menentukan pergerakan bibir.

### 3.2.1 Flowchart Metode Regression Trees

*Flowchart Regression Trees* mula–mula mengambil data citra gambar yang sudah dikonversi ke citra gambar *grayscale*. Kemudian citra gambar tersebut diolah menggunakan metode *histogram of oriented gradients* untuk memperkecil area komputasi.



Gambar 3.4 Flowchart Regression Trees

Area komputasi ini berbentuk persegi yang berada di area wajah responden. Pada area persegi ini akan dipetakan 68 penanda koordinat bentuk wajah yang akan digunakan untuk mendeteksi bagian-bagian yang ada pada wajah responden.

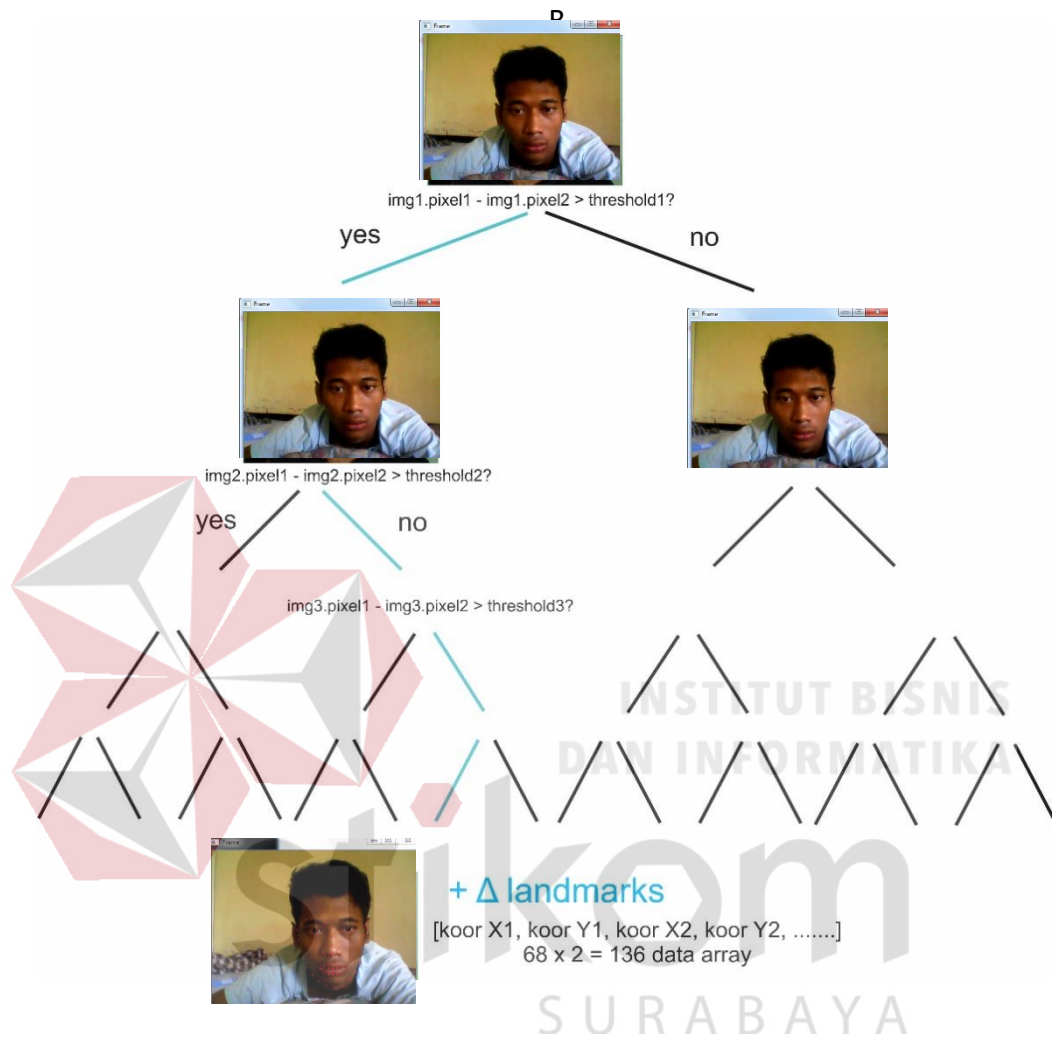
Kemudian dilakukan *looping* X sebanyak 10 kali untuk memberikan hasil regresi yang maksimal pada 68 penanda koordinat bentuk wajah. Pada setiap *looping* X, sistem akan melakukan penempatan 400 piksel secara acak menggunakan *similarity*

*transform* dari citra gambar  $S_m$  ke citra gambar  $S_c$ . Untuk *looping*  $X=1$  digunakan untuk *data training* yang pertama. Selanjutnya untuk *looping*  $X \geq 2$  hingga  $X \leq 10$  digunakan untuk *data training with dlib* untuk membandingkan beberapa data yang sudah ada pada *library dlib*.

Dari citra gambar  $S_c$ , sistem akan melakukan *looping*  $Y$  yang digunakan untuk membandingkan selisih perbedaan piksel dengan nilai *threshold* pada setiap percabangannya. Hasil dari perbandingan ini memiliki 16 kemungkinan yang akan dijadikan nilai dari selisih *landmark* untuk setiap *looping*  $Y$ . Kemudian citra gambar  $S_c$  akan ditambahkan dari nilai hasil selisih *landmark*.



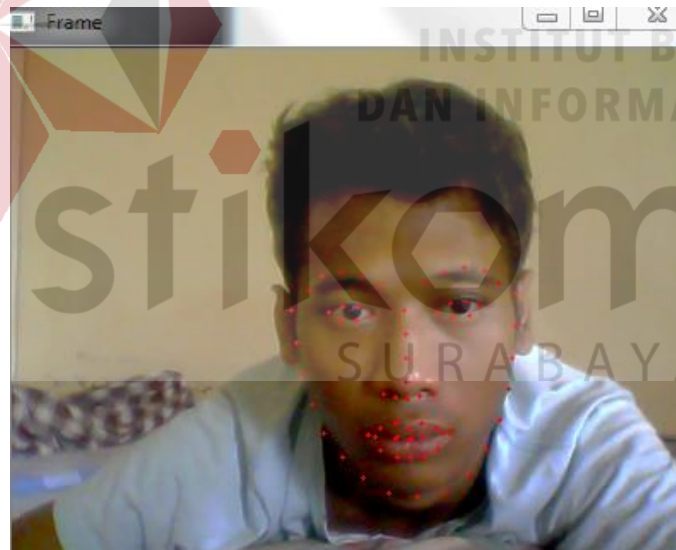
### A. Data Training 1



Gambar 3.5 Data Training 1



Gambar 3.6 Citra Gambar Pertama



Gambar 3.7 Citra Gambar Hasil

Dari Gambar 3.6, dapat dijelaskan bahwa *data training* ini merupakan implementasi dari metode *regression trees* dimana pengambilan keputusan menggunakan beberapa percabangan kondisi sehingga membentuk pohon keputusan.

Mula–mula pada citra gambar melakukan 400 *random pixel*. Kemudian dari 400 *pixel* tersebut, dilakukan *random* kembali untuk diambil 2 data *pixel* untuk diolah lalu dibandingkan dengan *random threshold* citra gambar. Pada sistem ini memungkinkan menghasilkan 16 hasil data  $\Delta$  *landmarks*. Hasil data tersebut berupa koordinat titik 68 *facial landmarks* yang terdiri dari x dan y, sehingga pada *array* data  $\Delta$  *landmarks* memiliki 136 data.

### B. Data Training With Dlib

*Data training with Dlib* secara konsep sama dengan *data training 1*, hanya saja *threshold* yang digunakan bukan dari hasil *random threshold* namun diambil dari maksimum korelasi perbandingan dengan *Datasheet* yang ada pada *Dlib*.

Tabel 3.1 Perhitungan *Threshold With Dlib*

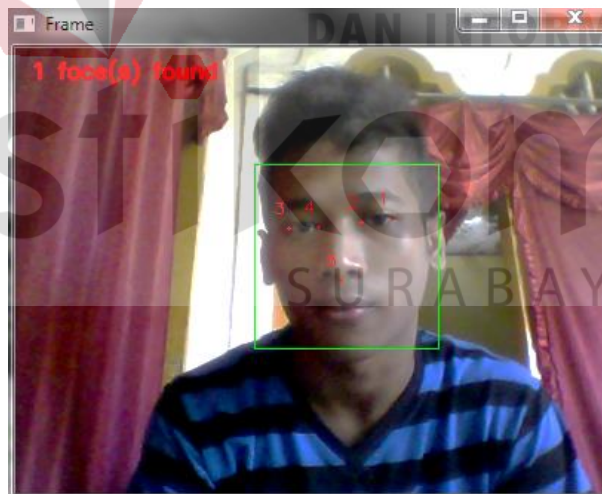
f1-f1	f1-f2	...	f400-f400	Ground Truth – Current Shape	rand(136)	d
p1,i1	p2,i1	...	p16000,i1	$\Delta 1$	$\Delta r$	d1
p1,i2	p2,i2	...	p16000,i2	$\Delta 2$	$\Delta r$	d2
...	...	...	...	...	...	...
p1,iN	p2,iN	...	p16000,iN	$\Delta N$	$\Delta r$	dN

Dari Tabel 3.1 perhitungan *threshold with dlib* diatas, dapat dijelaskan bahwa  $\Delta 1$  diambil dari operasi pengurangan citra gambar antara *ground truth* dengan *current shape*. Hasil tersebut kemudian akan dikalikan vektor dengan *random threshold* yang menghasilkan nilai d. Selanjutnya untuk data ke dua hingga data N, dilakukan dengan cara yang sama. Kemudian dilakukan operasi perhitungan maksimum korelasi dari nilai d.



### 3.4 Deteksi Wajah

Pada saat program dijalankan maka program akan mulai menjalankan kamera, yang dimana batas perekaman kamera meliputi objek serta lingkungan sekitar, sehingga batas komputasi yang dilakukan oleh program akan sangat besar. Oleh karena itu untuk memperkecil komputasi pada program yang berjalan maka perlu ditentukan sebuah batas perhitungan yang dimana berfungsi untuk memperkecil area perhitungan. Selain itu deteksi wajah ini juga berfungsi untuk melokalisasikan area wajah secara fleksibel sehingga bisa mendeteksi setiap inputan pada seluruh area *frame* dan menentukan apakah inputan yang masuk merupakan wajah atau bukan. Deteksi wajah tersebut bisa diamati pada Gambar 3.8.



Gambar 3. 8 Deteksi Wajah

Proses deteksi wajah ini dilakukan dengan menggunakan metode *Histogram of Oriented Gradients* (HOG) seperti yang telah dijelaskan pada sub bab sebelumnya. Adapun cuplikan program utama untuk melakukan proses deteksi wajah seperti diatas sebagai berikut:

```
# loop over the (x, y)-coordinates for the facial landmarks
# and draw each of them
for (i, (x, y)) in enumerate(shape):
    cv2.circle(frame, (x, y), 1, (0, 0, 255), -1)
    cv2.putText(frame, str(i + 1), (x - 10, y - 10),
                cv2.FONT_HERSHEY_SIMPLEX, 0.35, (0, 0, 255), 1)
```

### 3.5 Deteksi Bibir

Setelah wajah terdeteksi maka tahapan selanjutnya adalah deteksi bibir. Deteksi bibir merupakan kunci utama untuk melakukan komputasi dalam mendeteksi pengucapan huruf vokal nantinya. Deteksi bibir ini merupakan kelanjutan dari deteksi wajah, dimana hasil dari HOG merupakan area wajah dari responden. Batas koordinat tersebut kemudian akan digunakan untuk pembatas pemetaan 68 penanda koordinat bentuk wajah. Kemudian titik–titik koordinat tersebut akan digeser untuk pemetaan bentuk wajah menggunakan *looping regressor* selama beberapa kali. *Looping regressor* ini digunakan untuk memberikan penanda piksel pada citra wajah dengan menggunakan metode *regression trees*. Hasil deteksi bibir tersebut bisa diamati pada Gambar 3.9.



Gambar 3. 9 Deteksi Bibir

Kemudian sistem mengambil koordinat bibir pada 68 penanda koordinat bentuk wajah dimana bibir memiliki koordinat 49 hingga 60 untuk bibir bagian luar dan 61 hingga 68 untuk bibir bagian dalam. Adapun cuplikan program utama untuk melakukan proses deteksi bibir seperti diatas sebagai berikut:

```
# loop over the (x, y)-coordinates for the facial landmarks
# and draw them on the image
for (x, y) in shape [48:49]:
    cv2.circle(frame, (x, y), 1, (0, 0, 255), 1)
```

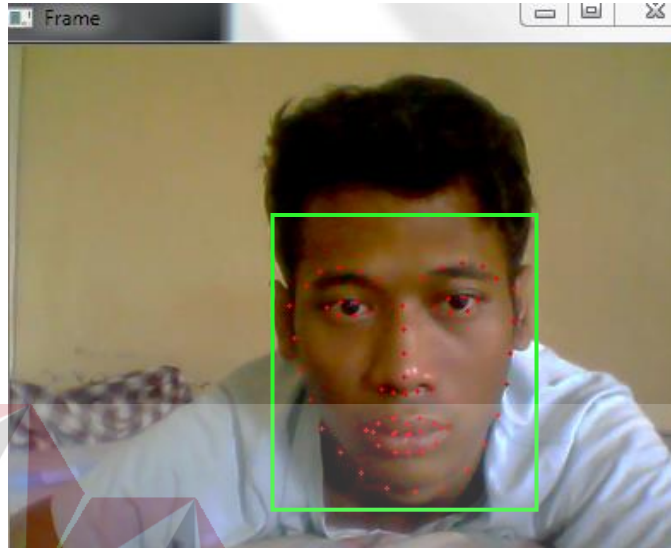
### 3.6 Deteksi Pengucapan Huruf Vokal

Dalam prosesnya untuk mendeteksi pengucapan setiap huruf vokal merupakan hal yang cukup kompleks. Untuk mendeteksi pengucapan huruf vokal belum tersedia sebuah *library* khusus maupun *function* yang bisa mendukung kinerja program untuk mendeteksi setiap pengucapan huruf vokal secara dinamis. Oleh karena itu dalam sistem deteksi pengucapan huruf vokal ini penulis menggunakan metode statis dalam menentukan setiap pengucapan huruf vokal. Dimana penentuan setiap nilai tersebut sesuai dengan intensitas piksel setiap *node* yang diperoleh dari *learning* program terhadap beberapa responden sebelumnya. Secara garis besar terdapat 2 tahapan yang perlu dilakukan dalam melakukan deteksi pengucapan huruf vokal yakni, pengambilan nilai setiap *node* berdasarkan *value* intensitas piksel dan pengklasifikasian *range node* untuk setiap huruf vokal.

#### 3.6.1 Pengambilan Nilai Setiap Node Berdasarkan Value Intensitas Piksel

Pengambilan nilai setiap *node* berawal saat sistem mengambil 1 frame dari kamera untuk diproses lebih lanjut. Proses selanjutnya adalah mengubah data citra gambar pada *frame* yang telah diambil dari RGB menjadi *grayscale*. Dari citra gambar *grayscale* akan diolah menggunakan metode *Histogram of Oriented Gradients* untuk memperkecil proses komputasi selanjutnya dimana hasil dari HOG merupakan area wajah dari responden. Batas koordinat tersebut kemudian akan digunakan untuk pembatas pemetaan 68 penanda koordinat bentuk wajah. Kemudian titik-titik koordinat tersebut akan digeser untuk pemetaan bentuk wajah menggunakan *looping regressor* selama beberapa kali. *Looping regressor* ini digunakan untuk memberikan

penanda piksel pada citra wajah dengan menggunakan metode *regression trees*. Titik-titik koordinat tersebut bisa diamati pada Gambar 3.10.



Gambar 3. 10 Koordinat Bibir Pada 68 Penanda Koordinat Bentuk Wajah

Kemudian sistem mengambil koordinat bibir pada 68 penanda koordinat bentuk wajah dimana bibir memiliki koordinat 49 hingga 60 untuk bibir bagian luar dan 61 hingga 68 untuk bibir bagian dalam. Nilai intensitas piksel pada koordinat (x, y) bibir inilah yang akan dimunculkan untuk dilakukan pengklasifikasian serta analisis terhadap nilai dan *range* setiap *node* pada saat pengucapan huruf vokal. Nilai dari 68 penanda koordinat dari *facial landmark* bila ditampilkan pada *command prompt* akan tampak seperti Gambar 3.11.



```

C:\ Windows Command Processor
<' [0] jarak euclidean 49 : ', 246.0894146443524>
<' [1] jarak euclidean 50 : ', 246.82990094394967>
<' [2] jarak euclidean 51 : ', 249.897979183506>
<' [3] jarak euclidean 52 : ', 255.69708641280997>
<' [4] jarak euclidean 53 : ', 258.7991499213241>
*****
<' [5] jarak euclidean 54 : ', 265.3073689138694>
<' [6] jarak euclidean 55 : ', 274.5268657162719>
<' [7] jarak euclidean 56 : ', 271.89703933658416>
<' [8] jarak euclidean 57 : ', 267.4210163767986>
<' [9] jarak euclidean 58 : ', 264.4919658515169>
*****
<' [10] jarak euclidean 59 : ', 259.50915205441214>
<' [11] jarak euclidean 60 : ', 253.86019774671254>
<' [12] jarak euclidean 61 : ', 248.1954874690513>
<' [13] jarak euclidean 62 : ', 253.97047072445253>
<' [14] jarak euclidean 63 : ', 258.3118270617898>
*****
<' [15] jarak euclidean 64 : ', 262.7565413077284>
<' [16] jarak euclidean 65 : ', 271.4940146669904>
<' [17] jarak euclidean 66 : ', 264.09278672466615>
<' [18] jarak euclidean 67 : ', 259.6709456215693>
<' [19] jarak euclidean 68 : ', 255.34486483969087>
*****
<' [0] jarak euclidean 49 : ', 246.0894146443524>
<' [1] jarak euclidean 50 : ', 247.5520147362974>
<' [2] jarak euclidean 51 : ', 250.63918289046507>
<' [3] jarak euclidean 52 : ', 255.69708641280997>
<' [4] jarak euclidean 53 : ', 258.7991499213241>
*****
<' [5] jarak euclidean 54 : ', 266.0695397823659>
<' [6] jarak euclidean 55 : ', 273.87953556262653>
<' [7] jarak euclidean 56 : ', 271.89703933658416>
<' [8] jarak euclidean 57 : ', 268.1510768205118>
<' [9] jarak euclidean 58 : ', 264.4919658515169>
*****
<' [10] jarak euclidean 59 : ', 259.50915205441214>
<' [11] jarak euclidean 60 : ', 253.86019774671254>
<' [12] jarak euclidean 61 : ', 248.1954874690513>
<' [13] jarak euclidean 62 : ', 254.65663156493687>
<' [14] jarak euclidean 63 : ', 259.04825805243314>
*****
<' [15] jarak euclidean 64 : ', 262.7565413077284>
<' [16] jarak euclidean 65 : ', 270.8394358286843>
<' [17] jarak euclidean 66 : ', 263.4236132164313>
<' [18] jarak euclidean 67 : ', 258.9903473104741>
<' [19] jarak euclidean 68 : ', 255.34486483969087>
*****

```

Gambar 3. 11 Nilai Intensitas Piksel Pada 68 Koordinat

### 3.6.2 Pengklasifikasian *Range Node* Untuk Setiap Huruf Vokal

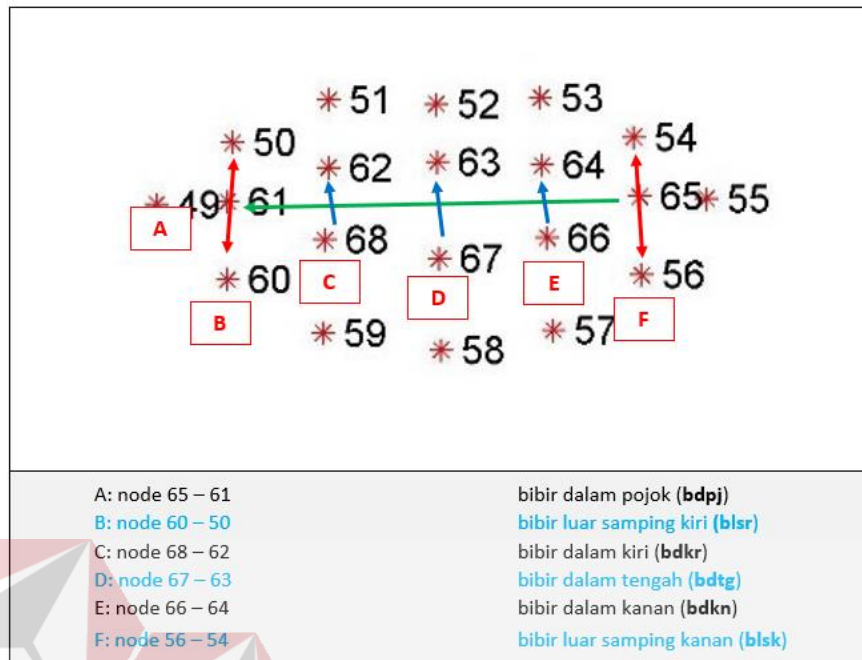
Nilai intensitas piksel pada 20 *node* di area bibir yang dimunculkan akan disimpan kedalam bentuk *file .xlsx* untuk dilakukan analisis guna menentukan *range* setiap huruf vokal. Nilai intensitas piksel pada 20 *node* di area bibir tersebut dapat diamati pada Tabel 3.2.

Tabel 3. 2 Tabel Nilai Intensitas Piksel 20 *Node* Dari Satu Responden

Orang ke -	Vokal	Data ke -	Value Node ke -																			
			49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68
abil 22th	A	1	216	215	222	230	236	248	262	263	257	252	244	232	219	226	232	238	258	251	245	237
		2	223	221	227	234	240	252	264	269	265	259	252	240	225	230	237	243	261	258	252	245
3		218	216	221	228	235	247	262	266	262	256	249	237	221	225	232	239	258	254	249	242	
4		218	215	220	228	234	247	261	267	264	257	250	237	222	224	231	238	257	255	250	242	
5		216	214	219	227	233	246	260	266	262	257	248	235	219	224	231	237	256	253	248	240	
I	1	214	218	228	236	243	255	267	267	261	254	246	233	219	231	239	247	264	254	248	239	
	2	216	220	227	236	242	253	267	267	262	254	247	234	220	232	239	246	263	253	247	239	
	3	216	219	227	235	241	252	266	266	260	253	245	232	219	231	238	245	262	252	245	238	
	4	216	219	228	236	242	252	264	265	260	253	245	232	219	231	238	245	261	253	246	238	
	5	215	215	221	230	236	248	262	263	257	251	242	231	219	226	234	240	258	248	243	235	
U	1	218	220	225	234	239	249	262	262	258	252	245	233	222	231	238	244	258	249	244	236	
	2	218	220	227	235	240	250	263	263	259	253	246	234	223	231	238	244	258	250	245	237	
	3	216	219	226	235	240	250	263	263	258	251	244	231	220	230	237	243	259	250	245	236	
	4	216	219	225	233	238	249	262	261	256	250	243	232	221	229	236	242	257	248	242	235	
	5	216	220	227	235	240	248	260	260	258	252	245	232	220	231	237	242	257	251	246	239	
E	1	211	212	219	227	234	245	257	262	257	251	242	229	214	224	231	238	253	248	242	235	
	2	212	211	217	225	230	242	255	260	256	250	242	230	216	222	229	235	252	247	242	234	
	3	210	210	216	223	229	241	254	258	253	247	241	229	214	221	227	233	250	245	238	232	
	4	211	210	216	224	230	241	253	258	254	247	240	228	214	221	228	234	250	244	238	231	
	5	210	210	216	224	230	242	255	259	254	248	241	228	214	221	228	234	251	246	240	231	
O	1	215	213	217	226	231	242	255	258	254	248	241	229	217	223	230	236	251	245	240	232	
	2	213	211	217	226	231	242	255	258	254	248	242	229	216	223	229	235	251	246	240	232	
	3	213	211	217	224	229	241	253	258	254	249	241	230	216	222	228	234	250	245	240	232	
	4	213	211	216	223	229	240	254	257	253	247	240	229	216	221	228	233	250	243	238	231	
	5	213	211	216	224	229	240	253	256	253	247	241	229	217	221	228	233	249	243	238	231	

Sistem mengambil koordinat bibir pada 68 penanda koordinat bentuk wajah dimana bibir memiliki koordinat 49 hingga 60 untuk bibir bagian luar dan 61 hingga 68 untuk bibir bagian dalam.





Gambar 3. 12 *Node* Aspek Rasio Bibir

Dalam menentukan *range* setiap *value* huruf vokal penulis hanya menggunakan beberapa *node* dari 20 *node* yang ada di area bibir. Hal ini berdasarkan analisis serta pengujian secara kontinyu. Penerapan sebagian *node* ini bertujuan untuk mengurangi komputasi yang tidak diperlukan serta menghindari bertabrakannya nilai *value* yang *linier* saat menggolongkan *range* setiap huruf vokal secara statis. *Node-node* tersebut yang akan dijadikan sebagai aspek rasio bibir dalam perhitungan pada program.

Berdasarkan aspek rasio bibir tersebut maka penulis merumuskan beberapa perhitungan sederhana yang diharapkan dapat mempermudah saat melakukan analisis statis *range* setiap huruf vokal yang dapat diamati pada Gambar 3.12.



Adapun perumusan dan penamaan *node* tersebut sebagai berikut:

1. Bibir dalam bagian ujung/pojok (bdpj)

Selisih *node* 65 dengan *node* 61

2. Bibir luar samping kiri (blsr)

Selisih *node* 60 dengan *node* 50

3. Bibir luar samping kanan (blsk)

Selisih *node* 56 dengan *node* 54

4. Bibir dalam bagian kiri (bdkr)

Selisih *node* 68 dengan *node* 62

5. Bibir dalam bagian tengah (bdtg)

Selisih *node* 67 dengan *node* 63

6. Bibir dalam bagian kanan (bdkn)

Selisih *node* 66 dengan *node* 64

Berdasarkan pengklasifikasian *node* diatas, maka di dalam program dibuat *function* khusus yang berfungsi untuk mempermudah pemanggilan setiap *array node* yang diperlukan dari *datasheet library* 68 koordinat *facial landmark*. *Function* dan *syntax* tersebut bisa dilihat seperti cuplikan program dibawah ini.

```
def node_a(varA, varB):
    return np.linalg.norm(varA - varB)

def arbA(lip):
    A = node_a(lip[19], lip[13])

    if int(A) > 0:
        text = "bdkr : {}".format(int(A))
        cv2.putText(frame, text, (10, 80), cv2.FONT_HERSHEY_SIMPLEX,
                    0.5, (255, 255, 0), 2)

    bdkr = A
    return bdkr
```

cuplikan program diatas menggunakan *function* untuk memanggil setiap *value node* 20 *facial landmark* yang tersimpan pada *array* dan mencari selisih nilai tersebut yang kemudian akan dimasukkan kedalam *variable* sebagai nilai dari salah satu set *variable* aspek rasio bibir, sehingga ketika program dijalankan maka akan menampilkan aspek rasio bibir kedalam layar *frame* yang dimana nilai tersebut diperoleh dari hasil pengurangan dari *node-node* yang sudah dideklarasikan sebagai objek analisis sebelumnya. Gambar 3.13 dibawah ini merupakan contoh *frame* yang telah dideklarasasi aspek rasio bibir.



Gambar 3. 13 *Frame* Deteksi Wajah, Bibir Dan Aspek Rasio Bibir

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Hasil pembahasan dan pengujian yang dilakukan oleh penulis akan dijelaskan pada bab ini. Tujuan dari bab ini adalah untuk mengetahui tingkat keberhasilan terhadap perancangan sistem yang telah diajukan dan dikerjakan. Tahapan pengujian yang akan dilakukan meliputi pengujian *raspberry pi 3* model B, pengujian *pi camera*, pengujian tingkat akurasi deteksi bibir, pengujian tingkat akurasi pengucapan setiap huruf vokal.

#### **4.1 Pengujian *Raspberry Pi 3* Model B**

##### **4.1.1 Tujuan Pengujian *Raspberry Pi 3* Model B**

Pengujian *Raspberry Pi 3* bertujuan untuk mengetahui apakah *Raspberry Pi 3* dapat berjalan dengan baik serta dapat mengeksekusi program yang menggunakan bahasa pemrograman *Python*.

##### **4.1.2 Peralatan Pengujian *Raspberry Pi 3* Model B**

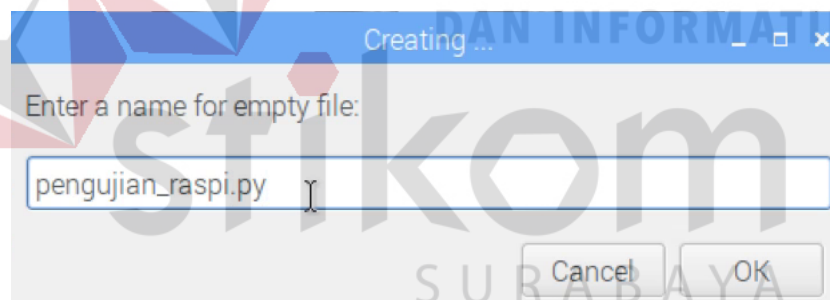
Dalam melakukan pengujian terhadap *Raspberry Pi 3* Model B dibutuhkan beberapa komponen pendukung agar kinerja dari *Raspberry Pi 3* Model B bisa lebih maksimal. Adapun komponen tersebut antara lain adalah:

1. *Raspberry Pi 3* Model B
2. Monitor
3. Mouse
4. Keyboard
5. *Charger Handphone (Output minimal 2A)*

#### 4.1.3 Prosedur Pengujian *Raspberry Pi 3 Model B*

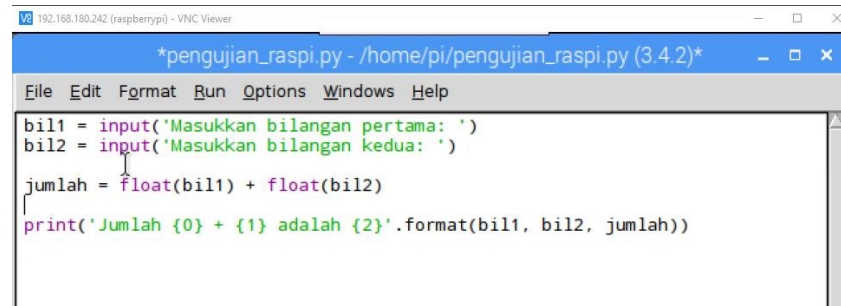
Dalam melakukan pengujian terhadap *Raspberry Pi 3 Model B* terdapat beberapa prosedur yang perlu dilakukan agar kinerja dari *Raspberry Pi 3 Model B* bisa lebih maksimal. Adapun prosedur tersebut antara lain adalah:

1. Menyambungkan kabel HDMI dari monitor ke *Raspberry Pi 3*, menyambungkan juga keyboard dan mouse ke *Raspberry Pi 3*.
2. Menyalakan *Raspberry Pi 3* dengan menyambungkan kabel *charger handphone*.
3. Membuka *file manager*.
4. Memilih *directory* yang diinginkan untuk menyimpan file *Python*. Kemudian mengeklik kanan, memilih *create new*, memilih *new file*.
5. Memberikan nama sesuai keinginan dengan ekstensi \*.py pada file tersebut.



Gambar 4.1 Pemberian Nama *File* Berekstensi \*.py

6. Membuat program sederhana untuk pertambahan dua bilangan. Contoh program sederhana dapat ditunjukkan pada Gambar 4.2.



```

192.168.180.242 (raspberrypi) - VNC Viewer

*pengujian_raspi.py - /home/pi/pengujian_raspi.py (3.4.2)*
File Edit Format Run Options Windows Help

bil1 = input('Masukkan bilangan pertama: ')
bil2 = input('Masukkan bilangan kedua: ')
jumlah = float(bil1) + float(bil2)
print('Jumlah {0} + {1} adalah {2}'.format(bil1, bil2, jumlah))

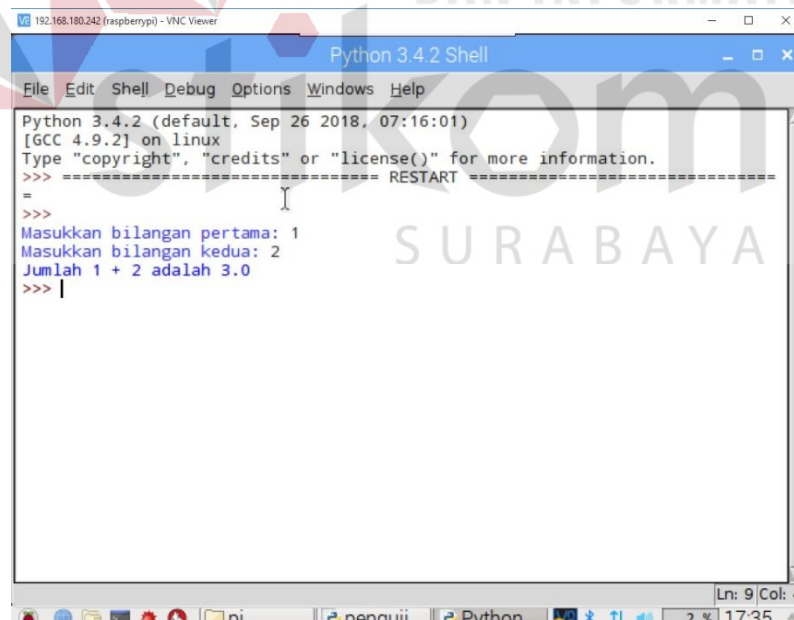
```

Gambar 4.2 Contoh *Program* Sederhana Pertambahan Dua Bilangan Menggunakan *Python*

7. Kemudian *run program* tersebut.

#### 4.1.4 Hasil Pengujian *Raspberry Pi 3 Model B*

Apabila *program* tidak ditemukan *error*, maka akan muncul *window* baru serta menampilkan *program* yang diinginkan. Contoh hasil *running program* dapat ditunjukkan pada Gambar 4.3.



```

192.168.180.242 (raspberrypi) - VNC Viewer

Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help

Python 3.4.2 (default, Sep 26 2018, 07:16:01)
[GCC 4.9.2] on linux
Type "copyright", "credits" or "license()" for more information.
>>> ----- RESTART -----
>>>
>>>
Masukkan bilangan pertama: 1
Masukkan bilangan kedua: 2
Jumlah 1 + 2 adalah 3.0
>>> |

```

Gambar 4.3 Hasil *Running Program* Sederhana Pertambahan Dua Bilangan Menggunakan *Python*

## 4.2 Pengujian *Pi Camera*

### 4.2.1 Tujuan Pengujian *Pi Camera*

Pengujian *Pi Camera* bertujuan untuk mengetahui apakah *Pi Camera* dapat berjalan dengan baik pada *Raspberry Pi 3* serta dapat merekam maupun memfoto suatu objek.

### 4.2.2 Peralatan Pengujian *Pi Camera*

Dalam melakukan pengujian terhadap *Pi Camera* dibutuhkan beberapa komponen pendukung agar kinerja dari *Pi Camera* bisa lebih maksimal. Adapun komponen tersebut antara lain adalah:

1. *Raspberry Pi 3 Model B*
2. *Pi Camera*
3. Monitor
4. Mouse
5. Keyboard
6. *Charger Handphone (Output minimal 2A)*

### 4.2.3 Prosedur Pengujian *Pi Camera*

Dalam melakukan pengujian terhadap *Pi Camera* dibutuhkan beberapa prosedur yang perlu dilakukan agar kinerja dari *Pi Camera* bisa lebih maksimal. Adapun prosedur tersebut antara lain adalah:

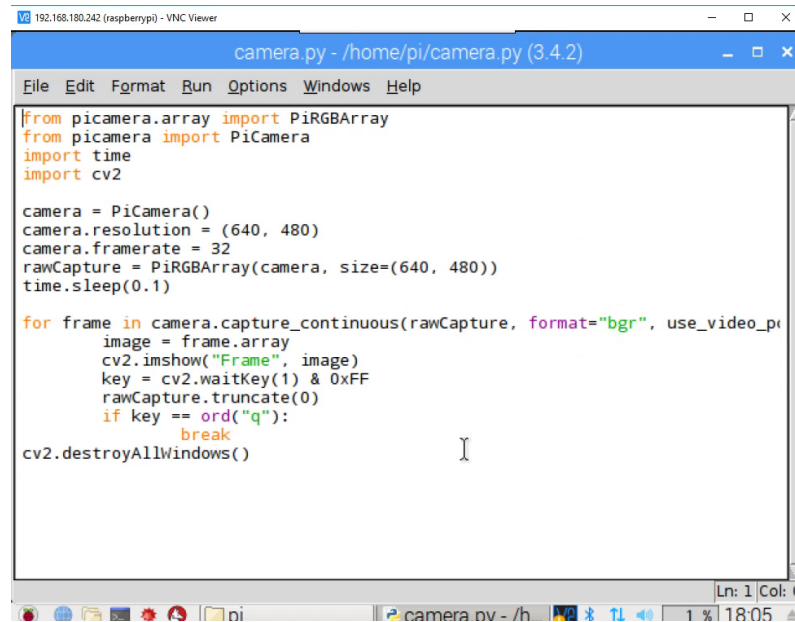
1. Menyambungkan kabel HDMI dari monitor ke *Raspberry Pi 3*, menyambungkan juga keyboard dan mouse ke *Raspberry Pi 3*.
2. Menyambungkan *Pi Camera* ke *Raspberry Pi 3*.



Gambar 4.4 Pemasangan *Pi Camera* ke *Raspberry Pi*

3. Menyalakan *Raspberry Pi 3* dengan menyambungkan kabel *charger handphone*.
4. Mengeklik *Raspberry Pi*, memilih *Preferences*, memilih *Raspberry Pi Configuration*.
5. Memilih menu *Interfaces*, kemudian mengubah ke *option Enable* pada pilihan *Camera*. Kemudian *Reboot Raspberry Pi*.
6. Membuka file manager.
7. Memilih *directory* yang diinginkan untuk menyimpan file *Python*. Kemudian mengeklik kanan, memilih *create new*, memilih *new file*.
8. Memberikan nama sesuai keinginan dengan ekstensi *\*.py* pada file tersebut.
9. Membuat program sederhana untuk merekam video. Contoh program sederhana dapat ditunjukkan pada Gambar 4.5.





```

camera.py - /home/pi/camera.py (3.4.2)
File Edit Format Run Options Windows Help

from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2

camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 32
rawCapture = PiRGBArray(camera, size=(640, 480))
time.sleep(0.1)

for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    image = frame.array
    cv2.imshow("Frame", image)
    key = cv2.waitKey(1) & 0xFF
    rawCapture.truncate(0)
    if key == ord("q"):
        break
cv2.destroyAllWindows()

```

Gambar 4.5 Contoh Program Sederhana Rekam Video Menggunakan Pi Camera

10. Kemudian *run program* tersebut.

#### 4.2.4 Hasil Pengujian Pi Camera

Apabila *program* tidak ditemukan *error*, maka akan muncul *window* baru serta menampilkan *program* yang diinginkan. Contoh hasil *running program* dapat ditunjukkan pada Gambar 4.6.





Gambar 4.6 Hasil *Running Program* Sederhana Rekam Video Menggunakan *Pi Camera*

### 4.3 Pengujian Tingkat Akurasi Deteksi Bibir

#### 4.3.1 Tujuan Pengujian Tingkat Akurasi Deteksi Bibir

Pada pengujian tingkat akurasi deteksi gerak bibir bertujuan untuk mengetahui apakah sistem deteksi bibir gerak ini dapat mendeteksi gerak bibir dalam beberapa kondisi secara akurat.

#### 4.3.2 Peralatan Pengujian Tingkat Akurasi Deteksi Bibir

Dalam melakukan pengujian terhadap tingkat akurasi deteksi bibir dibutuhkan beberapa komponen pendukung agar kinerja dari program bisa lebih maksimal. Adapun komponen tersebut antara lain adalah:

1. *Raspberry Pi* 3 Model B
2. Pi Camera
3. Monitor
4. Mouse

5. Keyboard
6. *Charger Handphone (Output minimal 2A)*

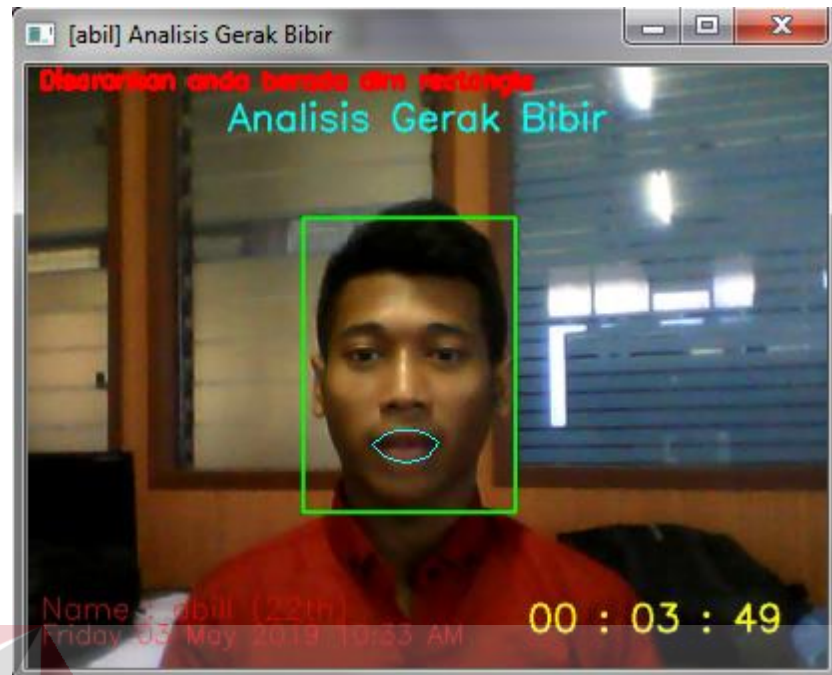
#### 4.3.3 Prosedur Pengujian Tingkat Akurasi Deteksi Bibir

Dalam melakukan pengujian terhadap tingkat akurasi deteksi bibir dibutuhkan beberapa prosedur yang perlu dilakukan agar kinerja dari program bisa lebih maksimal. Adapun prosedur tersebut antara lain adalah:

1. Menyambungkan kabel HDMI dari monitor ke *Raspberry Pi 3*, menyambungkan juga keyboard dan mouse ke *Raspberry Pi 3*.
2. Menyalakan *Raspberry Pi 3* dengan menyambungkan kabel *charger handphone*.
3. Membuka terminal (CLI) yang ada pada *Raspberry Pi 3*.
4. Menargetkan ke lokasi file berada kemudian *running* program tersebut.
5. Melakukan pergerakan bibir.
6. Melakukan langkah 4 hingga 5 ke beberapa orang.
7. Mengamati data tersebut.

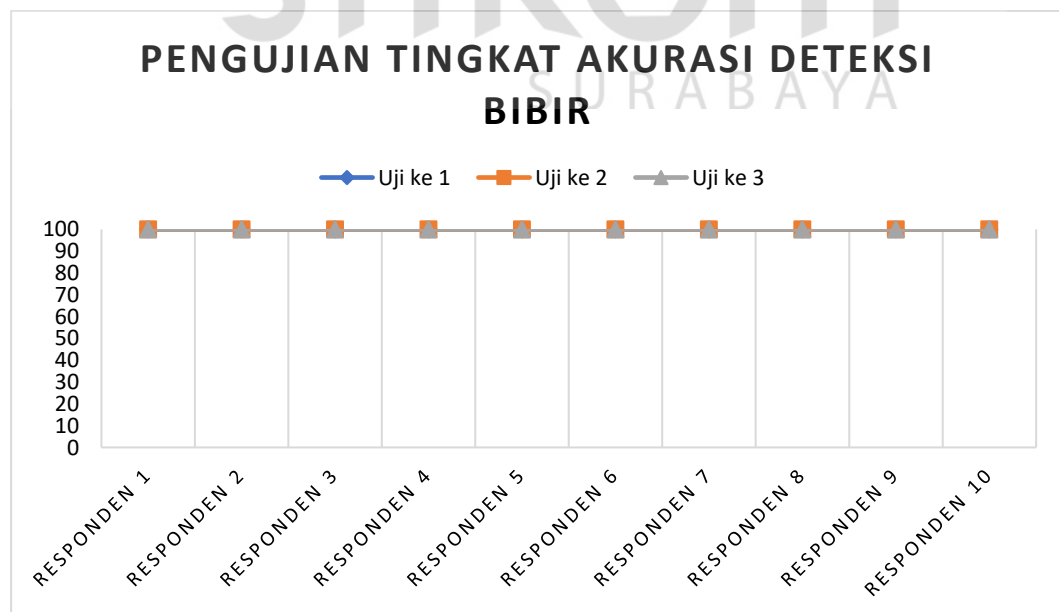
#### 4.3.4 Hasil Pengujian Tingkat Akurasi Deteksi Bibir

Apabila *program* tidak ditemukan *error*, maka akan muncul *window* baru serta menampilkan *program* yang diinginkan. Contoh hasil *running program* dapat ditunjukkan pada Gambar 4.7



Gambar 4. 7 Deteksi Bibir

Proses pengujian dilakukan dengan melakukan beberapa posisi oleh responden yakni bibir membuka dan menutup, melakukan pergerakan di area *frame*, dan memposisikan wajah dengan sudut kemiringan yang berbeda.



Gambar 4. 8 Diagram Pengujian Tingkat Akurasi Deteksi Bibir

Tabel 4. 1 Tabel Pengujian Tingkat Akurasi Deteksi Bibir

Responden ke -	Uji ke 1	Uji ke 2	Uji ke 3
1	Berhasil	Berhasil	Berhasil
2	Berhasil	Berhasil	Berhasil
3	Berhasil	Berhasil	Berhasil
4	Berhasil	Berhasil	Berhasil
4	Berhasil	Berhasil	Berhasil
6	Berhasil	Berhasil	Berhasil
7	Berhasil	Berhasil	Berhasil
8	Berhasil	Berhasil	Berhasil
9	Berhasil	Berhasil	Berhasil
10	Berhasil	Berhasil	Berhasil

Dari hasil diagram diatas, dapat dijelaskan bahwa dari 30 data *sample* uji coba terhadap 10 responden dapat mendeteksi gerak bibir dengan baik dan akurat, sehingga persentase data dapat dihitung sebagai berikut:

$$\begin{aligned}
 \text{Persentase data} &= \frac{(\text{banyak pengujian} \times \text{jumlah koresponden})}{\text{jumlah seluruh data}} \times 100\% \\
 &= \frac{(3 \times 10)}{30} \times 100\% \\
 &= \frac{30}{30} \times 100\% \\
 &= 100\%
 \end{aligned}$$

#### **4.4 Pengujian Tingkat Akurasi Pengucapan Huruf Vokal**

##### **4.4.1 Tujuan Pengujian Tingkat Akurasi Pengucapan Huruf Vokal**

Pada pengujian tingkat Akurasi pengucapan huruf vokal bertujuan untuk mengetahui apakah sistem dapat mendeteksi pengucapan huruf vokal, bibir mengucapkan huruf vokal maupun bibir diam. Pada pengujian ini, apabila bibir mengucapkan huruf vokal maka sistem akan menampilkan huruf vokal yang tengah diucapkan ke layar monitor.

##### **4.4.2 Peralatan Pengujian Tingkat Akurasi Pengucapan Huruf Vokal**

Dalam melakukan pengujian terhadap tingkat akurasi pengucapan huruf vokal dibutuhkan beberapa komponen pendukung agar kinerja dari *program* bisa lebih maksimal. Adapun komponen tersebut antara lain adalah:

1. *Raspberry Pi 3 Model B*
2. *Pi Camera*
3. *Monitor*
4. *Mouse*
5. *Keyboard*
6. *Charger Handphone (Output minimal 2A)*

##### **4.4.3 Prosedur Pengujian Tingkat Akurasi Pengucapan Huruf Vokal**

Dalam melakukan pengujian terhadap tingkat akurasi pengucapan huruf vokal dibutuhkan beberapa prosedur yang perlu dilakukan agar kinerja dari *program* bisa lebih maksimal. Adapun prosedur tersebut antara lain adalah:

1. Menyambungkan kabel HDMI dari monitor ke *Raspberry Pi 3*, menyambungkan juga keyboard dan mouse ke *Raspberry Pi 3*.
2. Menyalakan *Raspberry Pi 3* dengan menyambungkan kabel *charger handphone*.
3. Membuka terminal (CLI) yang ada pada *Raspberry Pi 3*.
4. Menargetkan ke lokasi file berada kemudian *running* program tersebut.
5. Berada pada posisi awal diam hingga muncul *notification* “Anda Diam”.

Seperti yang ditunjukkan pada Gambar 4.9



Gambar 4. 9 Posisi Awal Uji Huruf Vokal

6. Melakukan pengucapan huruf vokal.
7. Melakukan langkah 4 hingga 6 ke beberapa orang.
8. Mengamati data tersebut.

#### 4.4.4 Pengambilan Batas *Range* Untuk Uji Tingkat Akurasi Pengucapan Huruf Vokal Secara *Online (Realtime)*

Dalam menentukan pengucapan setiap huruf vokal maka diperlukan suatu batas *range* sebagai acuan untuk memprediksi setiap huruf vokal yang dilafalkan oleh responden. Batas *range* tersebut berfungsi untuk mengklasifikasikan setiap huruf vokal berdasarkan nilai intensitas piksel terhadap setiap koordinat *node facial landmark* yang terkait sesuai dengan aspek rasio bibir yang dipaparkan oleh Gambar 3.15.

Batas *range* diperoleh berdasarkan hasil analisis dari 10 responden dengan 5 kali pengambilan data pada setiap pengucapan huruf vokal terhadap masing-masing responden. Tabel analisis tersebut dapat diamati pada Tabel 3.1. Berdasarkan hasil analisis dan pengambilan data tersebut maka penulis menentukan beberapa batas nilai acuan pokok dalam mengklasifikasikan setiap huruf vokal.

Tabel 4. 2 Klasifikasi *Range* Pengucapan Huruf Vokal

Kondisi	Batas Range Node											
	bdpj		blsr		blpj		bdkr		bdtg		bdkn	
	min	max	min	max	min	max	min	max	min	max	min	max
Diam	26	28	10	11	25	33	0	2	0	3	0	2
A	25	28	15	19	30	35	7	12	8	13	7	12
I	33	40	12	14	39	42	3	8	4	8	3	8
U	20	24	9	11	25	33	0	2	0	2	0	2
E	28	32	15	17	34	38	3	7	4	8	3	8
O	20	23	11	14	26	29	2	6	3	6	2	6

Batas ini nantinya selain menjadi acuan juga akan menjadi sebuah nilai mutlak pada setiap pengujian-pengujian tingkat akurasi deteksi yang selanjutnya. Adapun batas *range* yang telah diklasifikasikan oleh penulis bisa dilihat pada Table 4.4 diatas.



#### 4.4.5 Hasil Pengujian Tingkat Akurasi Pengucapan Huruf Vokal Secara *Online (realtime)*

##### A. Hasil Pengujian Tingkat Akurasi Pengucapan Huruf Vokal A

Berdasarkan pengujian secara *online (realtime)* ke 10 responden yang mengucapkan huruf vokal A, maka didapatkan hasil seperti yang dipaparkan pada Gambar 4.10.



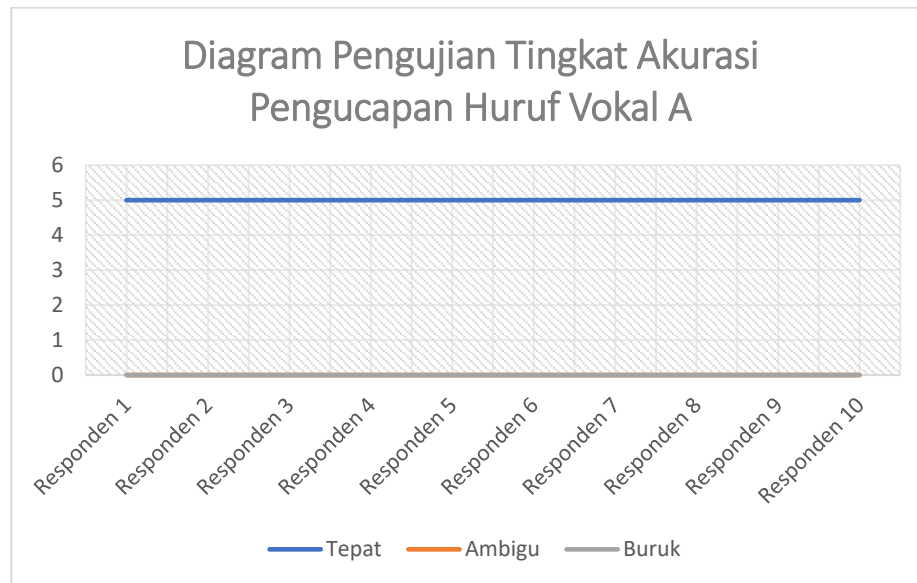
Gambar 4. 10 Beberapa Responden Saat Melakukan Pengucapan Huruf A

Tabel 4. 3 Batas *Range* Pengucapan Huruf Vokal A

Responden Ke -	Uji Ke -	Batas Range Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
1	1	27	17	31	10	11	10
	2	25	16	31	10	10	10
	3	25	16	31	10	10	9
	4	26	17	30	9	10	9
	5	25	16	31	8	9	8
2	1	25	15	31	8	9	8
	2	25	16	31	10	10	9
	3	26	17	30	9	10	9
	4	25	15	30	8	9	9
	5	28	18	32	11	11	10



Responden	Uji Ke	Batas Range Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
3	1	27	18	33	10	10	11
	2	24	17	30	9	9	9
	3	25	16	31	8	9	8
	4	26	17	32	11	11	11
	5	26	16	31	9	10	9
4	1	25	16	31	10	10	9
	2	25	15	30	8	9	9
	3	26	16	31	9	10	9
	4	28	18	34	10	11	10
	5	25	17	31	10	11	10
5	1	29	16	35	8	9	9
	2	29	17	35	8	9	8
	3	26	16	31	9	10	9
	4	28	18	34	10	11	10
	5	25	17	31	10	11	10
6	1	27	17	31	10	11	10
	2	25	16	31	10	10	10
	3	25	16	31	10	10	9
	4	26	17	30	9	10	9
	5	25	16	31	8	9	8
7	1	25	15	31	8	9	8
	2	25	16	31	10	10	9
	3	26	17	30	9	10	9
	4	25	15	30	8	9	9
	5	28	18	32	11	11	10
8	1	27	18	33	10	10	11
	2	24	17	30	9	9	9
	3	25	16	31	8	9	8
	4	26	17	32	11	11	11
	5	26	16	31	9	10	9
9	1	25	16	31	10	10	9
	2	25	15	30	8	9	9
	3	26	16	31	9	10	9
	4	28	18	34	10	11	10
	5	25	17	31	10	11	10
10	1	29	16	35	8	9	9
	2	29	17	35	8	9	8
	3	26	16	31	9	10	9
	4	28	18	34	10	11	10
	5	25	17	31	10	11	10



Gambar 4. 11 Diagram Pengujian Tingkat Akurasi Pengucapan Huruf Vokal A

Tabel 4.4 Pengujian Tingkat Akurasi Pengucapan Huruf Vokal A

Responden Ke -	Uji Ke -	Tepat	Ambigu	Buruk
1	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
2	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
3	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
4	1	v		
	2	v		
	3	v		
	4	v		
	5	v		

Responden Ke -	Uji Ke -	Tepat	Ambigu	Buruk
5	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
6	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
7	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
8	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
9	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
10	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
<b>Presentase Pengujian</b>		<b>100%</b>	<b>0%</b>	<b>0%</b>

Dari hasil diagram diatas, dapat dijelaskan bahwa dari 10 responden *sample* uji coba dapat mendeteksi pengucapan huruf vokal A secara akurat, sehingga persentase data dapat dihitung sebagai berikut:

1. Pengucapan vokal A tepat (%)
 
$$= \frac{\text{jumlah tepat}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{50}{50} \times 100\%$$

$$= 100 \%$$
2. Pengucapan vokal A ambigu (%)
 
$$= \frac{\text{jumlah ambigu}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{0}{50} \times 100\%$$

$$= 0 \%$$
3. Pengucapan vokal A buruk (%)
 
$$= \frac{\text{jumlah buruk}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{0}{50} \times 100\%$$

$$= 0 \%$$

#### **B. Hasil Pengujian Tingkat Akurasi Pengucapan Huruf Vokal I**

Berdasarkan pengujian secara *online (realtime)* ke 10 responden yang mengucapkan huruf vokal I, maka didapatkan hasil seperti yang dipaparkan pada Gambar 4.12.

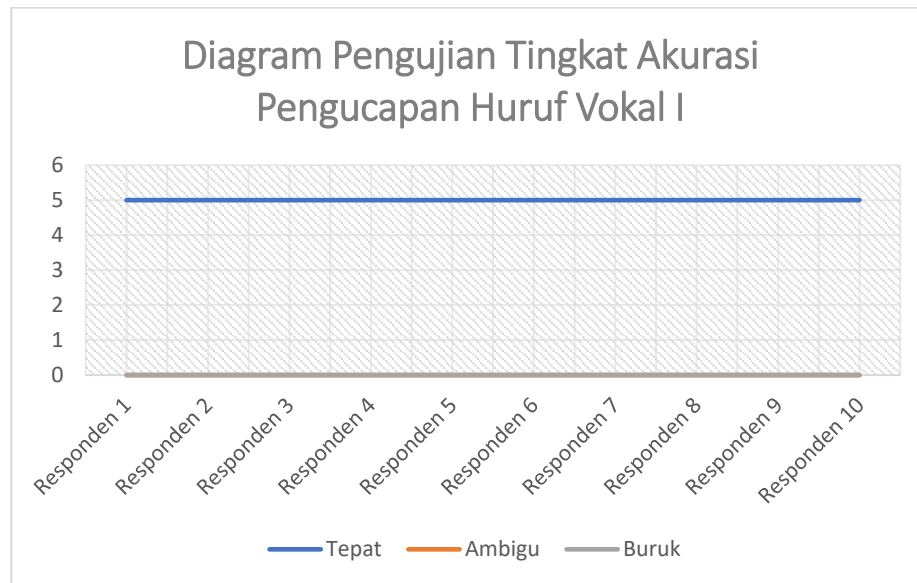


Gambar 4. 12 Beberapa Responden Saat Melakukan Pengucapan Huruf I

Tabel 4. 5 Batas Range Pengucapan Huruf Vokal I

Responden Ke -	Uji Ke -	Batas Range Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
1	1	35	13	41	5	5	6
	2	35	13	41	5	6	5
	3	34	13	40	5	5	5
	4	37	13	42	6	5	6
	5	36	13	41	5	6	6
2	1	34	14	40	5	5	5
	2	34	13	40	5	5	5
	3	37	13	42	6	5	6
	4	34	14	40	5	5	5
	5	35	14	41	6	5	5

Responden Ke -	Uji Ke -	Batas Range Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
3	1	36	13	42	5	5	6
	2	35	13	41	5	5	5
	3	33	13	39	5	5	5
	4	35	13	41	5	6	5
	5	33	13	39	4	5	4
4	1	35	13	41	5	6	5
	2	33	13	39	5	5	5
	3	36	13	41	5	6	6
	4	33	13	39	4	5	4
	5	37	13	42	6	5	6
5	1	36	13	41	5	6	6
	2	37	13	42	5	6	6
	3	33	13	39	4	5	4
	4	35	13	41	5	6	5
	5	34	13	40	5	6	5
6	1	35	13	41	5	5	6
	2	35	13	41	5	6	5
	3	34	13	40	5	5	5
	4	37	13	42	6	5	6
	5	36	13	41	5	6	6
7	1	34	14	40	5	5	5
	2	34	13	40	5	5	5
	3	37	13	42	6	5	6
	4	34	14	40	5	5	5
	5	35	14	41	6	5	5
8	1	36	13	42	5	5	6
	2	35	13	41	5	5	5
	3	33	13	39	5	5	5
	4	35	13	41	5	6	5
	5	33	13	39	4	5	4
9	1	35	13	41	5	6	5
	2	33	13	39	5	5	5
	3	36	13	41	5	6	6
	4	33	13	39	4	5	4
	5	37	13	42	6	5	6
10	1	36	13	41	5	6	6
	2	37	13	42	5	6	6
	3	33	13	39	4	5	4
	4	35	13	41	5	6	5
	5	34	13	40	5	6	5



Gambar 4.13 Diagram Pengujian Tingkat Akurasi Pengucapan Huruf Vokal I

Tabel 4. 6 Pengujian Tingkat Akurasi Pengucapan Huruf Vokal I

Responden Ke -	Uji Ke -	Tepat	Ambigu	Buruk
1	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
2	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
3	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
4	1	v		
	2	v		
	3	v		
	4	v		
	5	v		

Responden Ke -	Uji Ke -	Tepat	Ambigu	Buruk
5	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
6	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
7	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
8	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
9	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
10	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
<b>Presentase Pengujian</b>		<b>100%</b>	<b>0%</b>	<b>0%</b>



Dari hasil diagram diatas, dapat dijelaskan bahwa dari 10 responden *sample* uji coba dapat mendeteksi pengucapan huruf vokal E secara akurat, sehingga persentase data dapat dihitung sebagai berikut:

$$1. \text{ Pengucapan vokal I tepat (\%)} = \frac{\text{jumlah tepat}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{50}{50} \times 100\%$$

$$= 100 \%$$

$$2. \text{ Pengucapan vokal I ambigu (\%)} = \frac{\text{jumlah ambigu}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{0}{50} \times 100\%$$

$$= 0 \%$$

$$3. \text{ Pengucapan vokal I buruk (\%)} = \frac{\text{jumlah buruk}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{0}{50} \times 100\%$$

$$= 0 \%$$



INSTITUT BISNIS  
DAN INFORMATIKA  
**stikom**  
SURABAYA

### C. Hasil Pengujian Tingkat Akurasi Pengucapan Huruf Vokal U

Berdasarkan pengujian secara *online (realtime)* ke 10 responden yang mengucapkan huruf vokal U, maka didapatkan hasil seperti yang dipaparkan pada Gambar 4.14.

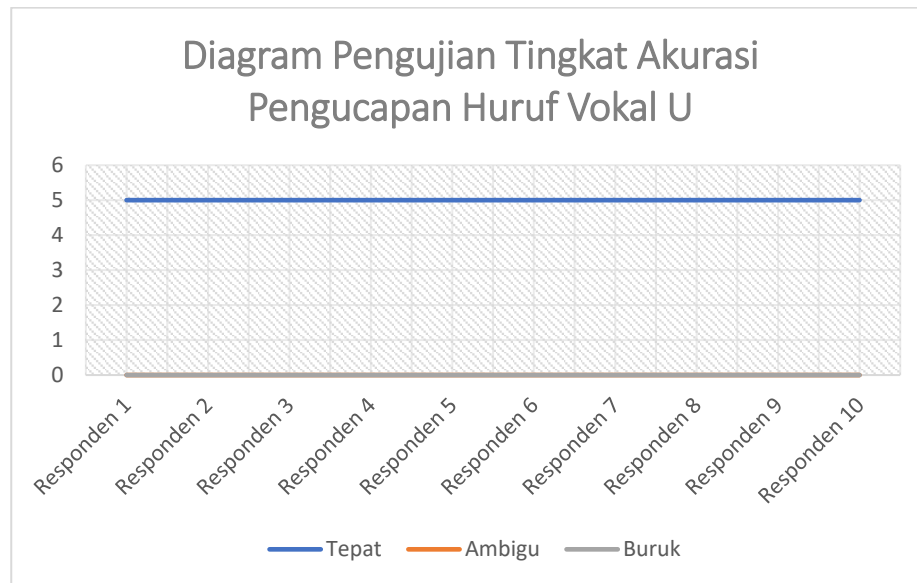


Gambar 4. 14 Beberapa Responden Saat Melakukan Pengucapan Huruf U

Tabel 4. 6 Batas *Range* Pengucapan Huruf Vokal A

Responden Ke -	Uji Ke -	Batas <i>Range</i> Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
1	1	21	11	27	2	2	2
	2	21	13	27	2	3	3
	3	21	11	27	1	2	1
	4	21	11	26	1	2	1
	5	20	10	26	1	1	2
2	1	21	11	26	2	2	1
	2	22	11	28	2	2	2
	3	20	10	25	1	2	2
	4	20	11	26	2	2	1
	5	21	11	27	1	2	2

Responden Ke -	Uji Ke -	Batas Range Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
3	1	21	11	26	2	2	1
	2	20	11	26	1	2	1
	3	21	11	27	1	2	1
	4	20	11	26	2	2	1
	5	22	12	28	3	3	4
4	1	22	12	28	2	2	2
	2	21	11	26	1	2	1
	3	27	11	33	1	2	1
	4	21	10	27	1	2	1
	5	21	12	26	2	2	2
5	1	27	11	33	2	1	1
	2	21	11	26	2	3	2
	3	21	12	27	2	2	2
	4	25	12	31	1	2	2
	5	22	12	27	1	2	1
6	1	21	11	27	1	2	2
	2	21	13	27	2	3	3
	3	21	12	27	1	1	1
	4	21	12	27	2	1	2
	5	20	10	26	1	1	2
7	1	21	11	27	2	2	2
	2	21	13	27	2	3	3
	3	21	11	27	1	2	1
	4	21	11	26	1	2	1
	5	20	10	26	1	1	2
8	1	21	11	26	2	2	1
	2	22	11	28	2	2	2
	3	20	10	25	1	2	2
	4	20	11	26	2	2	1
	5	21	11	27	1	2	2
9	1	21	11	26	2	2	1
	2	20	11	26	1	2	1
	3	21	11	27	1	2	1
	4	20	11	26	2	2	1
	5	22	12	28	3	3	4
10	1	22	12	28	2	2	2
	2	21	11	26	1	2	1
	3	27	11	33	1	2	1
	4	21	10	27	1	2	1
	5	21	12	26	2	2	2



Gambar 4.15 Diagram Pengujian Tingkat Akurasi Pengucapan Huruf Vokal U

Tabel 4. 7 Pengujian Tingkat Akurasi Pengucapan Huruf Vokal U

Responden Ke -	Uji Ke -	Tepat	Ambigu	Buruk
1	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
2	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
3	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
4	1	v		
	2	v		
	3	v		
	4	v		
	5	v		

Responden Ke -	Uji Ke -	Tepat	Ambigu	Buruk
5	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
6	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
7	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
8	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
9	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
10	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
<b>Presentase Pengujian</b>		<b>100%</b>	<b>0%</b>	<b>0%</b>

Dari hasil diagram diatas, dapat dijelaskan bahwa dari 10 responden *sample* uji coba dapat mendeteksi pengucapan huruf vokal U secara akurat.

Persentase data dapat dihitung sebagai berikut:

$$1. \text{ Pengucapan vokal U tepat (\%)} = \frac{\text{jumlah tepat}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{50}{50} \times 100\%$$

$$= 100 \%$$

$$2. \text{ Pengucapan vokal U ambigu (\%)} = \frac{\text{jumlah ambigu}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{0}{50} \times 100\%$$

$$= 0 \%$$

$$3. \text{ Pengucapan vokal U buruk (\%)} = \frac{\text{jumlah buruk}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{0}{50} \times 100\%$$

$$= 0 \%$$

#### **D. Hasil Pengujian Tingkat Akurasi Pengucapan Huruf Vokal E**

Berdasarkan pengujian secara *online (realtime)* ke 10 responden yang mengucapkan huruf vokal E, maka didapatkan hasil seperti yang dipaparkan pada Gambar 4.16.



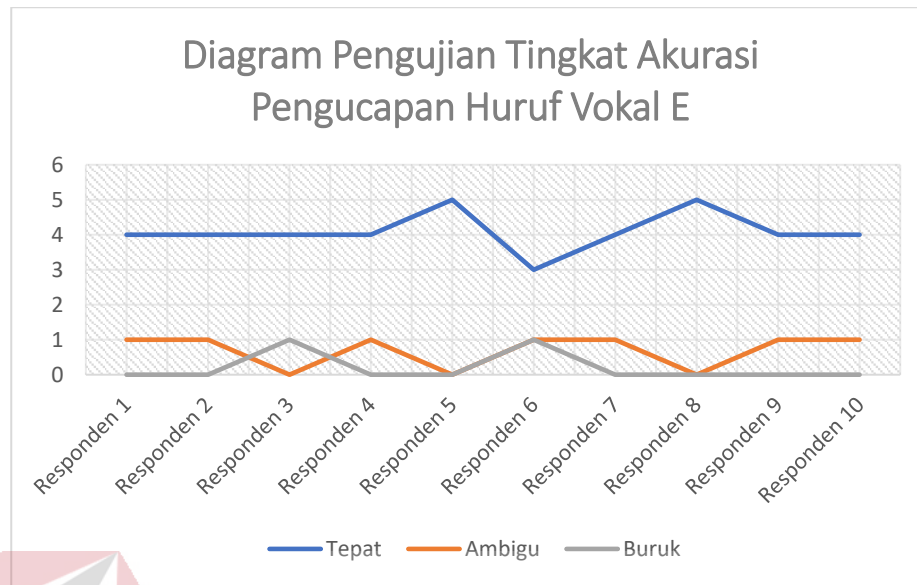
Gambar 4. 16 Beberapa Responden Saat Melakukan Pengucapan Huruf E

Tabel 4. 8 Batas *Range* Pengucapan Huruf Vokal E

Responden Ke -	Uji Ke -	Batas <i>Range</i> Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
1	1	31	14	37	5	5	5
	2	30	15	36	5	6	5
	3	28	10	34	0	1	1
	4	35	12	37	5	5	5
	5	29	14	35	4	4	4
2	1	30	15	35	5	5	5
	2	32	14	37	5	5	5
	3	30	14	36	5	6	5
	4	38	16	36	5	6	6
	5	30	15	35	5	5	5

Responden Ke -	Uji Ke -	Batas Range Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
3	1	30	14	36	5	6	5
	2	41	23	34	4	0	4
	3	30	15	36	5	6	5
	4	32	14	38	4	5	5
	5	32	14	37	4	4	4
4	1	26	15	38	5	6	5
	2	31	14	37	6	6	6
	3	32	14	37	5	6	6
	4	31	14	37	5	6	5
	5	31	15	37	5	6	5
5	1	31	14	37	5	5	5
	2	31	14	36	6	6	6
	3	31	14	37	5	6	5
	4	32	14	38	5	5	5
	5	31	14	37	6	6	6
6	1	32	13	37	5	6	5
	2	32	14	38	6	7	6
	3	43	18	44	9	9	8
	4	32	12	38	5	5	5
	5	32	14	37	5	5	5
7	1	31	14	37	5	5	5
	2	30	15	36	5	6	5
	3	28	10	34	0	1	1
	4	32	13	37	5	5	5
	5	29	11	35	4	4	4
8	1	30	15	35	5	5	5
	2	32	14	37	5	5	5
	3	30	14	36	5	6	5
	4	30	16	36	5	6	6
	5	30	15	35	5	5	5
9	1	30	14	36	5	6	5
	2	28	11	34	0	1	0
	3	30	15	36	5	6	5
	4	32	14	38	4	5	5
	5	33	14	37	4	4	4
10	1	28	15	38	7	9	7
	2	31	14	37	6	6	6
	3	32	14	37	5	6	6
	4	31	14	37	5	6	5
	5	31	15	37	5	6	5





Gambar 4.17 Diagram Pengujian Tingkat Akurasi Pengucapan Huruf Vokal E

Tabel 4. 9 Pengujian Tingkat Akurasi Pengucapan Huruf Vokal E

Responden Ke -	Uji Ke -	Tepat	Ambigu	Buruk
1	1	v		
	2	v		
	3	v		
	4		v	
	5	v		
2	1	v		
	2	v		
	3	v		
	4		v	
	5	v		
3	1	v		
	2			v
	3	v		
	4	v		
	5	v		
4	1		v	
	2	v		
	3	v		
	4	v		
	5	v		

Responden Ke -	Uji Ke -	Tepat	Ambigu	Buruk
5	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
6	1	v		
	2	v		
	3			v
	4		v	
	5	v		
7	1	v		
	2	v		
	3	v		
	4	v		
	5		v	
8	1	v		
	2	v		
	3	v		
	4	v		
	5	v		
9	1	v		
	2	v		
	3	v		
	4	v		
	5		v	
10	1	v		
	2	v		
	3		v	
	4	v		
	5	v		
<b>Presentase Pengujian</b>		<b>82%</b>	<b>14%</b>	<b>4%</b>

Dari hasil diagram diatas, dapat dijelaskan bahwa dari 10 responden *sample* uji coba ditemukan beberapa data yang tidak dapat mendeteksi pengucapan huruf vokal E secara akurat.

Persentase data dapat dihitung sebagai berikut:

$$1. \text{ Pengucapan vokal E tepat (\%)} = \frac{\text{jumlah tepat}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{41}{50} \times 100\%$$

$$= 82 \%$$

$$2. \text{ Pengucapan vokal E ambigu (\%)} = \frac{\text{jumlah ambigu}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{7}{50} \times 100\%$$

$$= 14 \%$$

$$3. \text{ Pengucapan vokal E buruk (\%)} = \frac{\text{jumlah buruk}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{2}{50} \times 100\%$$

$$= 4 \%$$



INSTITUT BISNIS  
DAN INFORMATIKA

stikom  
SURABAYA

### E. Hasil Pengujian Tingkat Akurasi Pengucapan Huruf Vokal O

Berdasarkan pengujian secara *online (realtime)* ke 10 responden yang mengucapkan huruf vokal O, maka didapatkan hasil seperti yang dipaparkan pada Gambar 4.18.

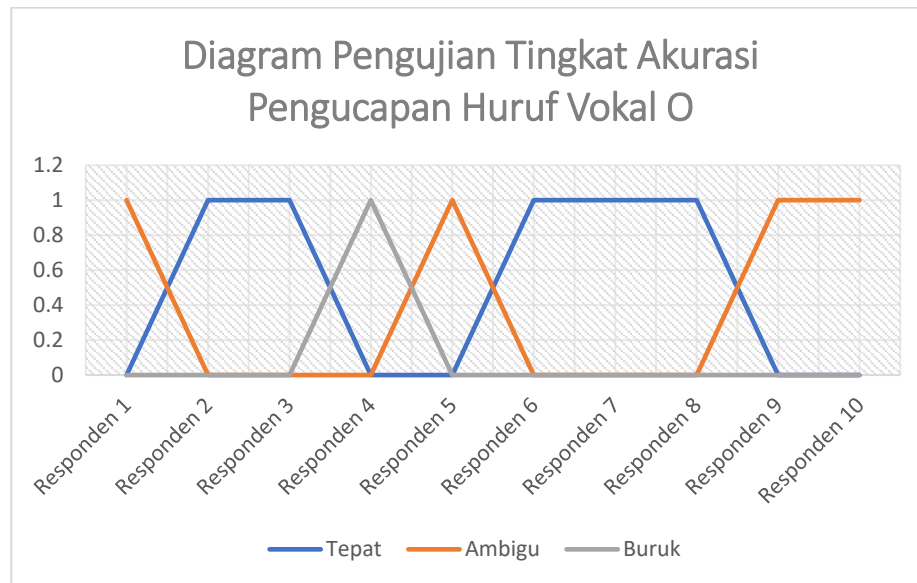


Gambar 4. 18 Beberapa Responden Saat Melakukan Pengucapan Huruf O

Tabel 4. 10 Batas *Range* Pengucapan Huruf Vokal O

Responden Ke -	Uji Ke -	Batas <i>Range</i> Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
1	1	22	13	29	5	5	5
	2	22	14	28	4	3	3
	3	24	13	28	3	2	4
	4	23	14	29	4	3	4
	5	27	15	29	4	3	4
2	1	22	14	28	4	4	3
	2	22	14	28	4	4	3
	3	23	14	29	3	4	4
	4	25	13	28	3	2	3
	5	21	12	27	4	3	4

Responden Ke -	Uji Ke -	Batas Range Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
3	1	22	13	28	3	4	4
	2	22	13	28	3	4	4
	3	20	10	26	3	3	3
	4	21	11	27	2	3	3
	5	23	14	29	3	4	4
4	1	22	13	28	4	4	4
	2	21	13	27	5	6	5
	3	22	14	28	4	5	4
	4	22	12	28	3	4	3
	5	24	14	28	4	5	4
5	1	22	14	28	5	5	5
	2	22	10	28	3	3	3
	3	22	14	28	4	5	4
	4	23	14	28	4	5	4
	5	25	13	29	5	5	5
6	1	22	14	28	4	5	4
	2	22	10	29	2	3	2
	3	22	11	28	3	3	2
	4	22	13	29	5	5	5
	5	21	13	27	5	6	5
7	1	22	13	29	5	5	5
	2	22	14	28	4	3	3
	3	22	15	28	4	4	4
	4	23	14	29	4	3	4
	5	23	14	29	4	3	4
8	1	22	14	28	4	4	3
	2	22	14	28	4	4	3
	3	24	14	29	1	2	2
	4	22	13	28	3	4	3
	5	21	12	27	4	3	4
9	1	22	13	28	3	4	4
	2	22	13	28	3	4	4
	3	24	12	26	2	1	3
	4	21	11	27	2	3	3
	5	23	14	29	3	4	4
10	1	24	20	30	4	0	4
	2	21	10	27	2	4	2
	3	22	12	28	4	5	4
	4	22	12	28	3	4	3
	5	22	14	28	4	5	4



Gambar 4. 19 Diagram Pengujian Tingkat Akurasi Pengucapan Huruf Vokal O

Tabel 4. 11 Pengujian Tingkat Akurasi Pengucapan Huruf Vokal O

Responden Ke -	Uji Ke -	Tepat	Ambigu	Buruk
1	1	v		
	2	v		
	3		v	
	4	v		
	5		v	
2	1	v		
	2	v		
	3		v	
	4		v	
	5	v		
3	1	v		
	2	v		
	3		v	
	4		v	
	5		v	
4	1	v		
	2	v		
	3	v		
	4	v		
	5		v	

Responden Ke -	Uji Ke -	Tepat	Ambigu	Buruk
5	1	v		
	2		v	
	3	v		
	4	v		
	5		v	
6	1	v		
	2		v	
	3	v		
	4	v		
	5	v		
7	1	v		
	2	v		
	3		v	
	4	v		
	5	v		
8	1	v		
	2	v		
	3		v	
	4	v		
	5	v		
9	1	v		
	2	v		
	3		v	
	4	v		
	5	v		
10	1			v
	2		v	
	3	v		
	4	v		
	5	v		
<b>Presentase Pengujian</b>		<b>68%</b>	<b>30%</b>	<b>2%</b>

Dari hasil diagram diatas, dapat dijelaskan bahwa dari 10 responden *sample* uji coba ditemukan beberapa data yang tidak dapat mendeteksi pengucapan huruf vokal O secara akurat.

Persentase data dapat dihitung sebagai berikut:

$$1. \text{ Pengucapan vokal O tepat (\%)} = \frac{\text{jumlah tepat}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{34}{50} \times 100\%$$

$$= 68 \%$$

$$2. \text{ Pengucapan vokal O ambigu (\%)} = \frac{\text{jumlah ambigu}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{15}{50} \times 100\%$$

$$= 30 \%$$

$$3. \text{ Pengucapan vokal O buruk (\%)} = \frac{\text{jumlah buruk}}{\text{jumlah seluruh data}} \times 100\%$$

$$= \frac{1}{50} \times 100\%$$

$$= 2 \%$$

#### 4.4.6 Batas Ambang / Penyimpangan Pada Uji Deteksi Pengucapan Huruf Vokal

Berdasarkan data hasil uji pengucapan setiap huruf vokal yang telah dilakukan terdapat perbedaan tingkat akurasi dalam mendeteksi setiap pengucapan huruf vokal. Perbedaan tersebut dapat diketahui dengan mengamati setiap tabel hasil uji pada tiap pelafalan. Dalam penerapannya terdapat 3 huruf vokal yang dapat terdeteksi dengan baik yaitu huruf A, U, dan E, sedangkan terdeteksi ambigu dan buruk pada pelafalan huruf I dan O.



Dari hasil analisis yang telah dilakukan diketahui bahwa perbedaan tingkat akurasi tersebut disebabkan karena adanya batas ambang atau penyimpangan *value* yang mana nilai dari *node* yang terdeteksi saat responden melakukan pelafalan berada diluar dari batas *range* aspek rasio bibir yang telah penulis tetapkan sebelumnya. Klasifikasi *range* pengucapan huruf vokal tersebut bisa dilihat pada Tabel 4.2. Selain penyimpangan terhadap batas *range* ARB yang telah ditentukan adanya kesalahan posisi awal responden yang tidak berada pada posisi awal “anda diam” seperti pada Gambar 4.9 juga menjadi faktor utama dalam akurasi dan kesesuaian deteksi tiap pengucapan huruf vokal.

#### A. Uji Deteksi Pengucapan Huruf Vokal Ambigu

Pengucapan huruf vokal yang ambigu dipicu karena adanya salah satu atau beberapa nilai *node* aspek rasio bibir yang berada diantara atau didalam batas *range* lebih dari satu huruf vokal. Sehingga sistem melakukan pembacaan data yang berhimpitan dan mendeteksi dua huruf yang berbeda secara bersamaan. Adapun pengucapan huruf vokal ambigu pada huruf I dan O tersebut bisa diamati pada Gambar 4.20 dan Gambar 4.21.



Gambar 4. 20 Pengucapan Huruf E Terbaca Huruf I Secara Bergantian (Ambigu)



Gambar 4. 21 Pengucapan Huruf O Terbaca Huruf U Secara Bergantian (Ambigu)

Saat pengucapan huruf vokal I nilai dari ARB bdpj dan blsr secara berurutan adalah 33 dan 14 sedangkan klasifikasi pengucapan huruf vokal E nilai ARB dari bdpj dan blsr secara berurutan adalah 28-32 dan 15-17. Dimana untuk klasifikasi pengucapan huruf I nilai ARB dari bdpj dan blsr secara berurutan adalah 33-40 dan 12-14. Dua nilai *node* ini menyimpang dari batas *range* yang telah ditentukan sebelumnya sehingga terjadi kebingungan program saat melakukan deteksi pengucapan huruf vokal. Begitupula dengan pengucapan huruf O yang terbaca ambigu terhadap pengucapan huruf U. Adapun nilai aspek rasio pengucapan huruf vokal ambigu tersebut dapat diamati pada Tabel 4.13 dengan kolom berwarna merah.

Tabel 4. 12 Pengucapan Huruf Vokal O Ambigu

Responden Ke -	Uji Ke -	Batas Range Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
1	1	22	13	29	5	5	5
	2	22	14	28	4	3	3
	3	22	13	28	4	4	4
	4	23	14	29	4	3	4
	5	23	14	29	4	3	4
2	1	22	14	28	4	4	3
	2	22	14	28	4	4	3
	3	23	14	29	3	4	4
	4	22	13	28	3	4	3
	5	21	12	27	4	3	4
3	1	22	13	28	3	4	4
	2	22	13	28	3	4	4
	3	20	10	26	3	3	3
	4	21	11	27	2	3	3
	5	23	14	29	3	4	4
4	1	22	13	28	4	4	4
	2	21	13	27	5	6	5
	3	22	14	28	4	5	4
	4	22	12	28	3	4	3
	5	24	14	28	4	5	4
5	1	22	14	28	5	5	5
	2	22	10	28	3	3	3
	3	22	14	28	4	5	4
	4	23	14	28	4	5	4
	5	25	13	29	5	5	5
6	1	22	14	28	4	5	4
	2	22	10	29	2	3	2
	3	22	11	28	3	3	2
	4	22	13	29	5	5	5
	5	21	13	27	5	6	5
7	1	22	13	29	5	5	5
	2	22	14	28	4	3	3
	3	22	15	28	4	4	4
	4	23	14	29	4	3	4
	5	23	14	29	4	3	4

Responden Ke -	Uji Ke -	Batas Range Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
8	1	22	14	28	4	4	3
	2	22	14	28	4	4	3
	3	24	14	29	1	2	2
	4	22	13	28	3	4	3
	5	21	12	27	4	3	4
9	1	22	13	28	3	4	4
	2	22	13	28	3	4	4
	3	24	12	26	2	1	3
	4	21	11	27	2	3	3
	5	23	14	29	3	4	4
10	1	24	20	30	4	0	4
	2	21	10	27	2	4	2
	3	22	12	28	4	5	4
	4	22	12	28	3	4	3
	5	22	14	28	4	5	4

### B. Uji Deteksi Pengucapan Huruf Vokal Buruk (Tidak Terdeteksi)

Pengucapan huruf vokal yang buruk dipicu karena adanya salah satu atau beberapa nilai *node* aspek rasio bibir yang berada diluar batas *range* dari semua huruf vokal, sehingga sistem tidak dapat mengklasifikasikan dan melakukan pembacaan data terhadap nilai setiap huruf vokal yang telah ditentukan oleh aspek rasio bibir. Adapun pengucapan huruf vokal yang buruk terdapat pada huruf O tersebut bisa diamati pada Gambar 4.22.



Gambar 4. 22 Pengucapan Huruf O Tidak Terdeteksi (Buruk)

Saat pengucapan huruf vokal O nilai dari ARB blsr dan blpj secara berurutan adalah 20 dan 45 sedangkan batas nilai minimum dan maksimum ARB untuk blsr dan blpj dari semua huruf vokal secara berurutan adalah 9-19 dan 25-42. Dimana nilai pada saat melakukan pengucapan huruf O tersebut berada diluar semua *range* ARB dari huruf vokal. Sehingga terjadi *missing value* saat program melakukan deteksi pengucapan huruf vokal. Adapun nilai aspek rasio pengucapan huruf vokal buruk tersebut dapat diamati pada Tabel 4.14 dengan kolom berwarna merah.

Tabel 4. 13 Pengucapan Huruf Vokal O Buruk

Responden Ke -	Uji Ke -	Batas <i>Range Node</i>					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
1	1	22	13	29	5	5	5
	2	22	14	28	4	3	3
	3	24	13	28	3	2	4
	4	23	14	29	4	3	4
	5	27	15	29	4	3	4
2	1	22	14	28	4	4	3
	2	22	14	28	4	4	3
	3	23	14	29	3	4	4
	4	25	13	28	3	2	3
	5	21	12	27	4	3	4

Responden Ke -	Uji Ke -	Batas Range Node					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
3	1	22	13	28	3	4	4
	2	22	13	28	3	4	4
	3	20	10	26	3	3	3
	4	21	11	27	2	3	3
	5	23	14	29	3	4	4
4	1	22	13	28	4	4	4
	2	21	13	27	5	6	5
	3	22	14	28	4	5	4
	4	22	12	28	3	4	3
	5	24	14	28	4	5	4
5	1	22	14	28	5	5	5
	2	22	10	28	3	3	3
	3	22	14	28	4	5	4
	4	23	14	28	4	5	4
	5	25	13	29	5	5	5
6	1	22	14	28	4	5	4
	2	22	10	29	2	3	2
	3	22	11	28	3	3	2
	4	22	13	29	5	5	5
	5	21	13	27	5	6	5
7	1	22	13	29	5	5	5
	2	22	14	28	4	3	3
	3	22	15	28	4	4	4
	4	23	14	29	4	3	4
	5	23	14	29	4	3	4
8	1	22	14	28	4	4	3
	2	22	14	28	4	4	3
	3	24	14	29	1	2	2
	4	22	13	28	3	4	3
	5	21	12	27	4	3	4
9	1	22	13	28	3	4	4
	2	22	13	28	3	4	4
	3	24	12	26	2	1	3
	4	21	11	27	2	3	3
	5	23	14	29	3	4	4
10	1	24	20	30	4	0	4
	2	21	10	27	2	4	2
	3	22	12	28	4	5	4
	4	22	12	28	3	4	3
	5	22	14	28	4	5	4

## BAB V

### KESIMPULAN

#### 5.1 Kesimpulan

Dari perancangan *program* hingga pengujian yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Metode *regression trees* memiliki akurasi yang tinggi untuk mendeteksi wajah maupun bagian spesifik wajah seperti bibir.
2. Posisi awal “Anda Diam” saat melakukan pengucapan huruf vokal menjadi faktor utama dalam dan kesesuaian deteksi tiap pengucapan huruf vokal.
3. Huruf A, I, U merupakan huruf yang terdeteksi paling baik dengan presentase akurasi 100% dari uji 10 responden.
4. Huruf E kadang-kadang terbaca ambigu terhadap huruf I dengan tingkat akurasi deteksi pelafalan huruf E sebesar 82% dari uji 10 responden.
5. Huruf O sering terbaca ambigu terhadap huruf U dan kadang-kadang tidak terdeteksi sehingga tingkat akurasi deteksi pelafalan huruf O paling rendah dibandingkan dengan huruf vokal yang lain yaitu sebesar 68% dari uji 10 responden.

#### 5.2 Saran

Dalam pengembangan selanjutnya dapat dilakukan pengklasifikasian secara dinamis atau secara otomatis dapat menentukan nilai dari setiap huruf vokal. Dalam *project* Tugas Akhir ini sistem masih berjalan secara statis, yang mana hasil setiap keluaran huruf vokal yang terdeteksi berasal dari analisis terhadap setiap

*node* dan merangkumnya menjadi *range* tertentu untuk menentukan huruf vokal apa yang tengah diucapkan oleh *user*.





## DAFTAR PUSTAKA

Belharbi, S. (2015). Input/Output Deep Architecture for Structured. *Arxis*, 17.

Breiman, L. (2017). Classification Algorithms and. 368.

Febriana, R. (2015). Analysis And Implementation Of Lip Movement Translator Application To Text Using. *e-Proceeding of Engineering*, 6.

Kazemi, V. (2014). One Millisecond Face Alignment with an Ensemble of Regression Trees. *KazemiCV*, 9.

Lakuary, R. P. (2015). Aplikasi Deteksi Gerak Bibir Menggunakan Kurva Bezier. *EmguCv*, 10.

Rajneesh. (2018). Real Time Drivers Drowsiness Detection and alert. 181.

Randa, A. F. (2016). Implementasi Metode Kombinasi Histogram Of. *Jurnal Teknik ITS*, 5.

Raut, N. (2018). Facial Emotion Recognition Using Machine Learning . *Masters Project*.

Samma, R. N. (2012). Simulasi Dan Analysis Sistem Deteksi Pengucapan Huruf Vokal Dengan Deteksi Gerak Bibir Berbasis Pengolahan Video Digital. *Karya Ilmiah - Skripsi (S1)*.

upi.edu. (2005). Fonologi Vokal.