

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

Abil Khosim¹⁾ Susijanto Tri Rasmana²⁾ Heri Pratikno³⁾

Program Studi/Jurusan Sistem Komputer
Institut Bisnis dan Informatika Stikom Surabaya
Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email: 1)abilae86@gmail.com, 2)susyanto@stikom.edu, 3)heri@stikom.edu

Abstrak: Komunikasi merupakan salah satu kunci pokok dalam bermasyarakat. Berbagai cara dan metode dapat digunakan agar bisa terjalin sebuah komunikasi. Salah satunya adalah metode komunikasi secara verbal. Komunikasi secara verbal ialah memahami suatu pembicaraan dari orang lain dengan berfokus melihat gerakan bibir dari lawan bicara. Huruf vokal merupakan kunci terpenting dalam sebuah komunikasi verbal. Penyandang tuna rungu biasanya menggunakan metode dengan komunikasi secara verbal ini dalam berinteraksi. Membaca dan memahami gerakan bibir dari lawan bicara merupakan salah satu pemanfaatan indra penglihatan kaum tuna rungu untuk menutupi keterbatasan dalam menerima suara (Febriana, 2015). Berdasarkan hal tersebut maka dalam Tugas Akhir ini penulis membuat gagasan untuk menganalisis sistem pengolahan citra yang 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 *realtime* dengan proses pengolahan citra digital. Dalam penelitian ini dirancang sebuah sistem untuk mendeteksi pengucapan huruf vokal berdasarkan deteksi *facial landmarks* menggunakan metode *regression trees* dengan baik. Berdasarkan hasil uji, sistem mampu 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, Deteksi Bibir, Deteksi *Facial Landmarks*, *Regression Trees*.

PENDAHULUAN

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 metode komunikasi secara verbal. Komunikasi secara verbal ialah memahami suatu pembicaraan dari orang lain dengan berfokus melihat gerakan bibir dari lawan bicara. Huruf vokal merupakan kunci terpenting dalam sebuah komunikasi verbal. Penyandang tuna rungu biasanya menggunakan metode dengan komunikasi secara verbal ini dalam berinteraksi. Membaca dan memahami gerakan bibir dari lawan bicara merupakan salah satu pemanfaatan indra penglihatan kaum tuna rungu untuk menutupi

keterbatasan dalam menerima suara (Febriana, 2015).

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).

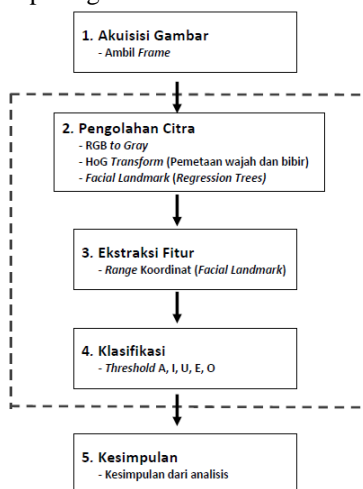
Berdasarkan hal tersebut maka dalam Tugas Akhir ini penulis membuat gagasan untuk menganalisis sistem pengolahan citra yang mengkonversi gerak bibir menjadi sebuah teks, dalam proses pengerjaan Tugas Akhir ini dibatasi hanya untuk huruf vokal saja. Perubahan gerak bibir menjadi teks ini dapat dilakukan melalui perekaman wajah secara *realtime* 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 ([openlibrary_telkomuniversity](https://openlibrary.telkomuniversity.ac.id/), 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 pada penelitian selanjutnya dibuat sebuah analisis gerak bibir dalam pengucapan huruf vokal berbasis *facial landmarks detection* menggunakan metode *regression trees* dengan *Raspberry Pi*.

METODE PENELITIAN

Metode yang diimplementasikan pada penelitian 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 dirubah menjadi citra *grayscale* dan menggunakan metode HOG pada tahap *image processing* untuk memperkecil area komputasi. Setelah itu dilakukan *feature extraction* dengan ditambahkan 68 koordinat *facial landmark* serta menggunakan metode *regression trees* untuk memperoleh nilai *threshold* pada gerak bibir.



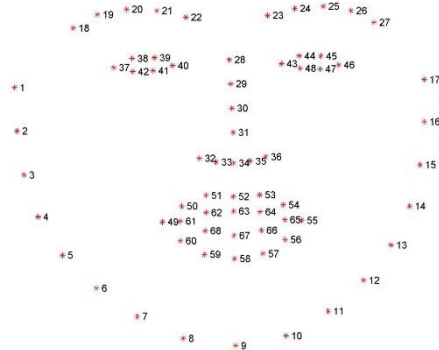
Gambar 1. Bagan metodologi penelitian

Histogram of Oriented Gradients

Untuk mendeteksi sebuah citra gambar digunakan metode *Histogram of Oriented Gradient*. Untuk mendapatkan sebuah hasil, citra gambar dibagi dalam banyak sel dan nantinya setiap sel tersebut dianggap sebagai bentuk dari *histogram of oriented gradients* (Randa, 2016).

Deteksi Facial Landmarks

Citra wajah merupakan sebuah data dari hasil prediksi bentuk geometri yang merupakan tujuan awal dari deteksi *facial landmarks*. *datasheet* yang digunakan sangat mempengaruhi jumlah titik yang terdapat pada *landmark*. Dalam sistem pendeteksi gerak bibir ini menggunakan 68 penanda *landmarks* wajah (Raut, 2018).

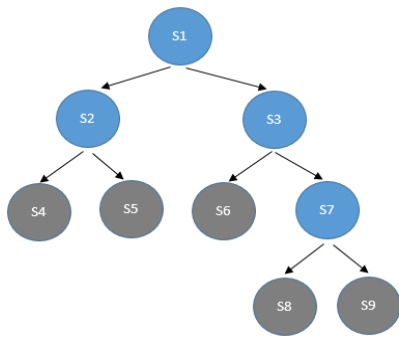


Gambar 2. Penanda koordinat 68 Landmark wajah (Raut, 2018)

Regression Trees

Teknik pohon keputusan mempunyai beberapa metode, salah satunya adalah *Classification and Regression Trees*. Yang mana bentuk kontinyu merupakan variabel respon dari *regression trees*, sedangkan bentuk skala kategorik merupakan variabel respon dari *classification trees* (Breiman, 2017).

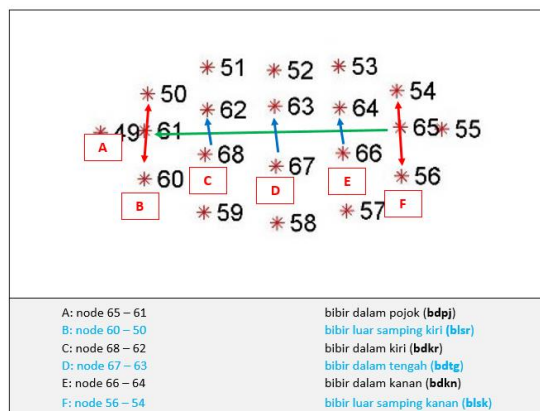
Metode percabangan atau rekursif *binner* yang merupakan kumpulan data/simpul yang dicabangkan menjadi dua bagian (anak data) dan menghasilkan data yang tidak bisa bercabang lagi (data akhir) disebut dengan *regression trees* (Breiman, 2017).



Gambar 3. Struktur pohon regresi

Aspek Rasio Bibir

Aspek rasio bibir berfungsi dalam menentukan nilai piksel saat bibir melakukan pelafalan. Penghitungan pada aspek rasio bibir menggunakan intensitas piksel pada titik *facial landmark* di seluruh bagian bibir. Adapun bagian-bagian tersebut ditunjukkan pada gambar 4.

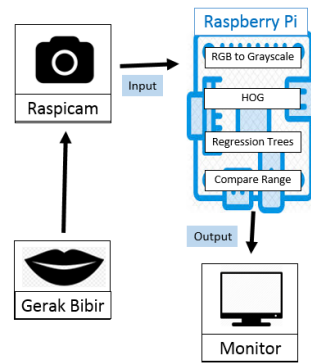


Gambar 4. Node aspek rasio bibir

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*.

Perancangan Perangkat Keras

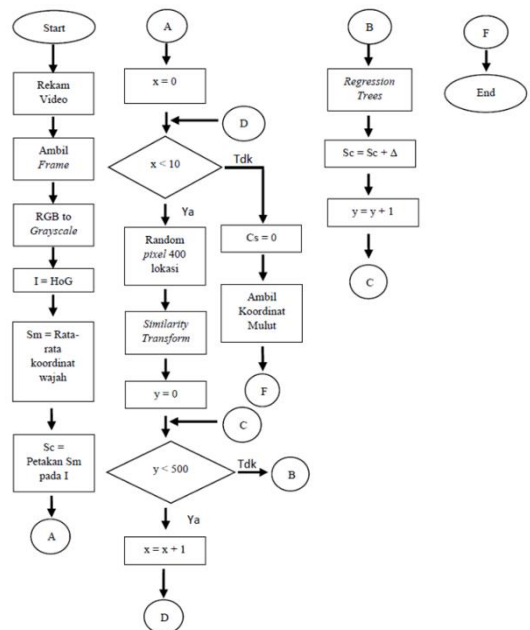
Perancangan program Tugas Akhir ini merupakan implementasi dari algoritma *regression trees* pada *facial landmarks detection* yang digunakan untuk mendeteksi bentuk bibir. Pendeteksi bentuk bibir tersebut dilakukan oleh *pi camera* yang kemudian data berupa citra gambar diproses oleh *raspberry pi*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



Gambar 5. Perancangan perangkat keras

Berdasarkan gambar 5 diatas dapat dipahami alur dari perancangan, yaitu data gerak bibir yang didapatkan dari *Pi Camera* langsung diproses oleh *Python* dan *OpenCV* menggunakan *Raspberry Pi* dengan menentukan koordinat bibir menggunakan metode pohon regresi. Setelah mendapatkan koordinat dari bibir, maka *value* koordinat dari bibir diolah dengan beracuan pada rumus perhitungan aspek rasio bibir untuk menemukan *value* pergerakan bibir. Bila hasil aspek rasio bibir yang dihasilkan berada di salah satu *range value* yang telah ditetapkan oleh penulis, maka *monitor* menampilkan huruf vokal apa yang tengah diucapkan oleh responden

Perancangan Sistem Program Flowchart Pendeteksi Gerak bibir



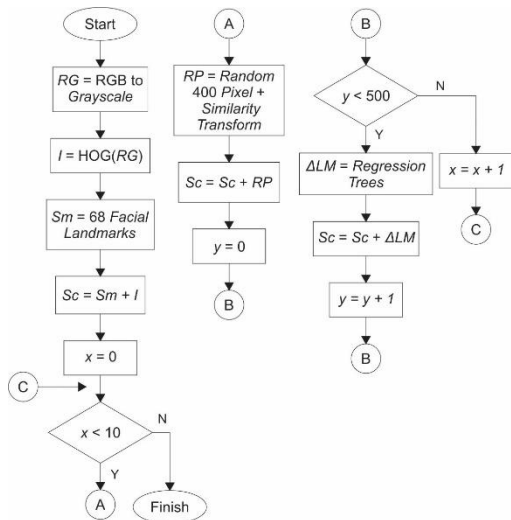
Gambar 5. Flowchart pendeteksi gerak bibir

Berdasarkan gambar 7 bisa dijelaskan alur dari sistem pendeteksi gerak bibir bekerja. Pertama kali sistem mengambil 1 *frame* untuk diproses, yang mana *frame* diambil dari kamera. Tahap selanjutnya ialah merubah data dari citra gambar yang terdapat dalam *frame* menjadi *grayscale* dari bentuk citra awal RGB.

Setelah citra gambar dirubah menjadi *grayscale* maka dilakukan metode *Histogram of Oriented Gradients* yang bertujuan memperkecil area dan proses perhitungan. Tahap berikutnya hasil *Histogram of Oriented Gradients* yang merupakan area wajah dari responden. Batas area pada koordinat yang dihasilkan dijadikan sebagai pembatas dalam memetakan penanda 68 koordinat pada wajah. Selanjutnya titik-titik dari koordinat digeser guna memetakan bentuk wajah dengan menggunakan metode *looping regressor* hingga beberapa kali. *Looping regressor* berfungsi untuk memberikan sebuah penanda piksel terhadap citra wajah dengan beracuan pada metode pohon regresi.

Setelah itu sistem mengambil koordinat bibir pada 68 *node* koordinat pada wajah yang mana 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 diolah menggunakan *value range* dari aspek rasio bibir untuk menentukan pergerakan bibir.

Flowchart Metode Pohon Regresi



Gambar 6. Flowchart pohon regresi

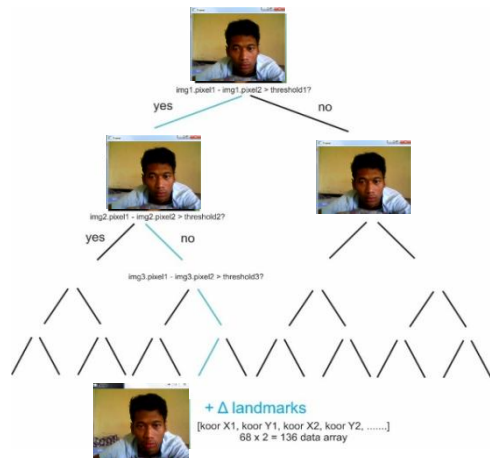
Flowchart Pohon Regresi dimulai dengan merubah citra gambar yang telah diambil kedalam bentuk *grayscale*. Setelah itu untuk memperkecil area komputasi maka digunakan metode *histogram*

of oriented gradients. Akan terdapat sebuah area persegi disepanjang batas citra wajah. Didalam area ini 68 koordinat akan mulai mendeteksi bagian - bagian yang merupakan citra dari wajah.

Setelah itu program *looping X* sebanyak 10 kali untuk untuk mendapatkan 68 koordinat pada wajah secara maksimal. Setiap melakukan *looping X*, menempatkan 400 piksel secara *random* menggunakan *similarity transform* yang merupakan citra gambar dari *Sm* menuju citra gambar *Sc*. Sedangkan *data training* yang pertama menggunakan *looping X=1*. Sedangkan *looping X* >=2 sampai *X* <=10 berfungsi sebagai *data training* dengan *dlib* guna dibandingkan dengan data-data yang telah tersedia dalam *library dlib*.

Berdasarkan citra dari gambar *Sc*, maka sistem menjalankan *looping Y* untuk mencari selisih dari perbedaan piksel terhadap nilai *threshold* di setiap hasil dari percabangannya. Dari *looping Y* menghasilkan 16 kemungkinan yang merupakan hasil dari perbandingan dan menjadi sebuah nilai dari selisih yang dihasilkan oleh *landmark*. Lalu selanjutnya nilai hasil dari selisih *landmark* ditambahkan dengan citra gambar *Sc*.

Data Pelatihan 1



Gambar 7. Data pelatihan 1



Gambar 8. Citra pertama



Gambar 9. Citra hasil

Pada gambar 9, bisa diketahui bahwa data pelatihan ini adalah penerapan dari metode pohon regresi dimana dalam pengambilan keputusan berdasarkan beberapa kondisi percabangan hingga membentuk sebuah pohon regresi / keputusan. Awal mula citra gambar melakukan 400 kali *random pixel*. Kemudian diambil 2 data *pixel* dari 400 *pixel* tersebut melalui proses *random* kembali untuk diolah dan dibandingkan dengan citra gambar dari *random threshold*. Dalam sistem ini menghasilkan kemungkinan 16 hasil data berupa Δ *landmarks*. Dimana data yang dihasilkan berupa koordinat 68 titik *facial landmarks* yang mana terdiri atas x dan y, sehingga membuat data Δ *landmarks* pada array mempunyai 136 data.

Data Pelatihan dengan Dlib

Data pelatihan dengan *Dlib* sama dengan data pelatihan 1 bila dilihat secara konsep, hanya berbeda *threshold* yang dipakai bukan dihasilkan oleh *random threshold* tetapi didapatkan dari perbandingan maksimum korelasi dengan *Datasheet* yang terdapat didalam *Dlib*.

Tabel 1. Perhitungan *Threshold* dengan *Dlib*

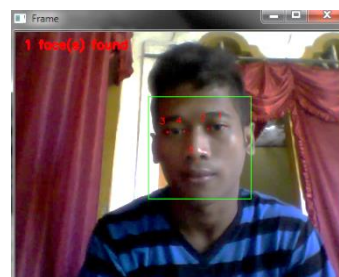
fr1- fr1	fr1- fr2	...	fr400- fr400	Gt- Sc	rand (136)	dt
q1, i1	q2, i1	...	q16000, i1	$\Delta 1$	Δr	dt1
q1, i2	q2, i2	...	q16000, i2	$\Delta 2$	Δr	dt2
...

fr1- fr1	fr1- fr2	...	fr400- fr400	Gt- Sc	rand (136)	dt
q1, in	q2, in	...	q16000, in	Δn	Δr	dt _n

Berdasarkan tabel 1 diatas dapat diamati bahwa $\Delta 1$ diperoleh dari hasil pengurangan citra gambar *ground truth* terhadap *current shape*. Selanjutnya dilakukan perkalian antara hasil selisihan dengan vektor dan *random threshold* sehingga menghasilkan nilai dt. Setelah itu dilakukan cara yang sama untuk memperoleh data yang kedua hingga data ke N. Setelah itu berdasarkan nilai dt dilakukan operasi perhitungan maksimum (Rajneesh, 2018).

Deteksi Wajah

Pada saat program dijalankan maka program mulai menjalankan kamera, yang dimana batas perekaman kamera meliputi objek serta lingkungan sekitar, sehingga batas komputasi yang dilakukan oleh program 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 mengfiksasi 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 10.



Gambar 10. Deteksi wajah

Proses deteksi wajah ini dilakukan menggunakan sistem *Histogram of Oriented Gradients* (HOG) seperti yang telah dijelaskan pada subbab sebelumnya.

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 dipakai sebagai batas pemetaan 68 penanda koordinat pada bentuk wajah. Selanjutnya menggunakan *looping regressor* titik-titik dari koordinat dilakukan pemetaan dengan mengeser setiap koordinat tersebut. *Looping regressor* ini berfungsi untuk memberikan sebuah penanda piksel di area citra wajah dengan mengimplementasikan metode *regression trees*. Hasil deteksi bibir tersebut bisa diamati pada gambar 11.



Gambar 11. Deteksi bibir

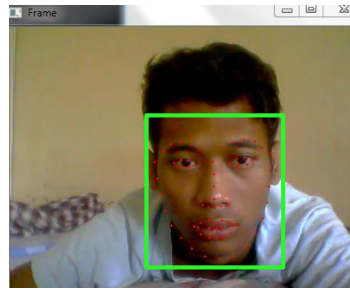
Deteksi Pengucapan Huruf Vokal

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.

Pengambilan Nilai Setiap Node Berdasarkan Value Intensitas Piksel

Pengambilan nilai setiap *node* berawal saat sistem mengambil sebuah frame dari *pi camera* untuk dilakukan proses lebih lanjut. Proses selanjutnya ialah merubah data pada citra gambar di *frame* yang sudah diambil dari citra RGB menjadi citra *grayscale*. Dari citra gambar *grayscale* tersebut diproses lagi menggunakan metode *Histogram of Oriented Gradients* guna memperkecil proses dari komputasi selanjutnya

dimana hasil dari HOG merupakan area wajah dari responden. Batas koordinat tersebut kemudian digunakan untuk pembatas pemetaan 68 penanda koordinat bentuk wajah. Kemudian titik-titik pada koordinat tersebut digeser secara berkala untuk pemetaan bentuk wajah menggunakan *looping regressor* dalam kurun 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 12.



Gambar 12. Koordinat 68 Node

Kemudian sistem mengambil setiap koordinat 68 penanda bibir pada 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 dimunculkan untuk dilakukan pengklasifikasian serta analisis terhadap nilai dan *range* setiap *node* pada saat pengucapan huruf vokal.

HASIL DAN PEMBAHASAN

Hasil Uji Tingkat Akurasi Pendeteksi Bibir

Untuk mengetahui apakah program pada sistem deteksi gerak bibir ini bisa mendeteksi gerak bibir dalam berbagai kondisi secara tepat dan akurat. Apabila *program* tidak ditemukan *error*, maka muncul *window* baru serta menampilkan *program* yang diinginkan. Contoh hasil *running program* dapat ditunjukkan pada gambar 13.



Gambar 13. Deteksi bibir

Proses pengujian dilakukan dengan melakukan beberapa posisi oleh responden yakni

bibir membuka dan menutup, melakukan pergerakan di area *frame*, dan mengucapkan beberapa huruf vokal.

Tabel 2. Uji tingkat akurasi pendeteksi bibir

Responden Ke-	Uji Ke-1	Uji Ke- 2	Uji Ke- 3
1	Sukses	Sukses	Sukses
2	Sukses	Sukses	Sukses
3	Sukses	Sukses	Sukses
4	Sukses	Sukses	Sukses
5	Sukses	Sukses	Sukses
6	Sukses	Sukses	Sukses
7	Sukses	Sukses	Sukses
8	Sukses	Sukses	Sukses
9	Sukses	Sukses	Sukses
10	Sukses	Sukses	Sukses

Berdasarkan hasil tabel 2, bisa disimpulkan bahwa dari 10 responden dengan 30 data *sample* uji coba dapat mendeteksi gerak bibir dengan baik dan akurat dengan presentase akurasi 100%.

Uji Tingkat Akurasi Pengucapan Huruf Vokal

Untuk mengamati apakah sistem bisa mendeteksi pengucapan huruf vokal, bibir mengucapkan huruf vokal maupun bibir diam. Dalam pengujian ini, ketika bibir mengucapkan huruf vokal maka sistem menampilkan huruf vokal yang tengah diucapkan ke layar monitor.

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 4.

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

Tabel 3. Klasifikasi *Range* Pengucapan Huruf Vokal

Kondisi	Batas Range Node											
	bdpj		blsr		bjpj		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

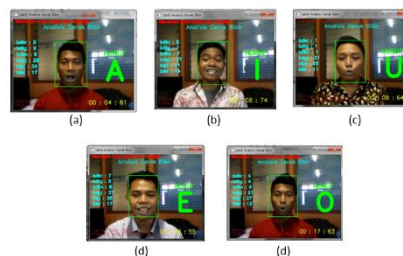
Batas ini nantinya selain menjadi acuan juga menjadi sebuah nilai mutlak pada setiap pengujian-pengujian tingkat akurasi deteksi yang selanjutnya. Selain itu pada saat melakukan pengujian berada pada posisi awal diam hingga muncul *notification* “Anda Diam” merupakan kunci utama untuk mendapatkan nilai deteksi huruf vokal dengan akurasi yang tinggi. Posisi tersebut bisa diamati pada gambar 14.



Gambar 14. Posisi awal uji huruf vokal

Hasil Pengujian Tingkat Akurasi Pengucapan Huruf Vokal A, I, U, E, dan O

Berdasarkan pengujian secara *online* (*realtime*) ke 10 responden yang mengucapkan huruf vokal A, maka didapatkan hasil seperti yang dipaparkan pada Gambar 4.



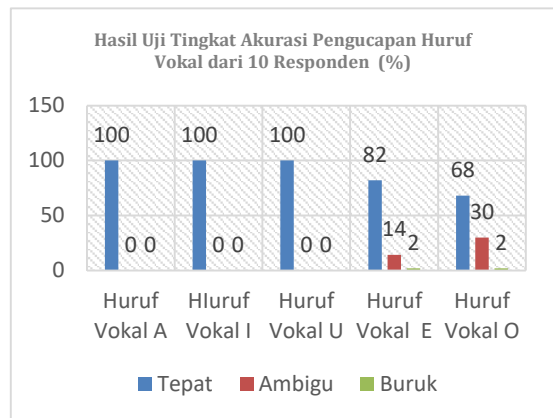
Gambar 15. Responden saat melakukan pengujian (a) Huruf A (b) Huruf I (c) Huruf U (d) Huruf E (e) huruf O

Tabel 4. *Range Value* rata-rata responden saat pengucapan huruf vokal A, I, U, E, dan O

Koresponden Ke-	Uji Ke-	Batas Range Node Rata-rata					
		bdpj	blsr	blsk	bdkr	bdtg	bdkn
1	A	26	16	31	9	10	9
	I	35	13	41	5	5	6
	U	21	11	27	1	2	2
	E	30	13	36	4	4	4
	O	22	14	29	4	4	4
2	A	26	16	31	9	10	9
	I	35	14	41	5	5	5
	U	21	11	26	2	2	2
	E	30	15	36	5	5	5
	O	22	13	28	4	4	3
3	A	26	17	31	9	10	10
	I	34	13	40	5	5	5
	U	21	11	27	2	2	2
	E	30	14	36	4	4	4
	O	22	13	28	3	4	4
4	A	26	16	31	9	10	9
	I	35	13	40	5	5	5
	U	22	11	28	1	2	1
	E	31	14	37	5	6	5
	O	22	13	28	4	5	4
5	A	26	16	31	9	10	9
	I	35	13	41	5	5	6
	U	21	11	27	1	2	2
	E	30	13	36	4	4	4
	O	22	14	29	4	4	4
6	A	26	16	31	9	10	9
	I	35	14	41	5	5	5
	U	21	11	26	2	2	2
	E	30	15	36	5	5	5
	O	22	13	28	4	4	3
7	A	26	16	31	9	10	9
	I	35	13	40	5	5	5
	U	22	11	28	1	2	1
	E	31	14	37	5	6	5
	O	22	13	28	4	5	4
8	A	26	16	31	9	10	9
	I	35	13	40	5	5	5
	U	22	11	28	1	2	1
	E	31	14	37	5	6	5
	O	22	13	28	4	5	4
9	A	26	16	31	9	10	9
	I	35	13	41	5	5	6
	U	21	11	27	1	2	2
	E	30	13	36	4	4	4
	O	22	14	29	4	4	4
10	A	26	16	31	9	10	9
	I	35	13	41	5	5	6
	U	21	11	27	1	2	2
	E	30	13	36	4	4	4
	O	22	14	29	4	4	4

Range value pada tabel 4 merupakan hasil dari perhitungan 50 data yang diambil nilai rata-ratanya pada setiap pengucapan huruf vokal.

Berdasarkan hasil analisis dari 10 responden dengan 5 kali pengambilan data pada setiap pengucapan huruf vokal terhadap masing-masing responden. Maka diperoleh sebuah data tingkat akurasi pengucapan setiap huruf vokal yang mana dapat diamati pada grafik gambar 16.



Gambar 16. Diagram pengujian tingkat akurasi pengucapan huruf vokal

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, I, dan U, sedangkan terdeteksi ambigu dan buruk pada pelafalan huruf E 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 3. Selain penyimpangan terhadap batas *range* aspek rasio bibir yang telah ditentukan adanya kesalahan posisi awal responden yang tidak berada pada posisi awal “anda diam” seperti pada gambar 14 juga menjadi faktor utama dalam akurasi dan kesesuaian deteksi tiap pengucapan huruf vokal.

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. Selain itu juga sistem memunculkan dua huruf yang berbeda tersebut secara bergantian (*blink*) sehingga rancu. Adapun pengucapan huruf vokal ambigu pada huruf I dan O

tersebut bisa diamati seperti gambar 17 dan gambar 18.



Gambar 17. Pengucapan huruf E terbaca sebagai huruf I secara bergantian (Ambigu)



Gambar 18. Pengucapan huruf O terbaca sebagai huruf U secara bergantian (Ambigu)

Saat pengucapan huruf vokal I nilai dari ARB bdpj dan blsr secara berurutan ialah 33 dan 14 sedangkan klasifikasi pengucapan huruf vokal E nilai ARB dari bdpj dan blsr secara berurutan ialah 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 ditetapkan sebelumnya sehingga terjadi kebingungan program saat melakukan deteksi pengucapan huruf vokal. Begitupula dengan pengucapan huruf O yang terbaca ambigu terhadap pengucapan huruf U.

KESIMPULAN

Berdasarkan perancangan sistem dan hasil dari pengujian terhadap program yang telah diujikan, maka didapatkan sebuah kesimpulan 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, U, E merupakan huruf yang terdeteksi paling baik dengan presentase akurasi 100% dari uji 10 responden.
4. Huruf I terkadang terbaca ambigu terhadap huruf E dengan tingkat akurasi deteksi pelafalan huruf I sebesar 82% dari uji 10 responden.
5. Huruf O sering terbaca ambigu terhadap huruf U dan terkadang tidak terdeteksi sehingga tingkat akurasi deteksi pelafalan huruf O paling rendah dibandingkan dengan huruf vokal yang lain yaitu sebesar 68% dari uji 10 responden.

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

- 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 Dengan Kombinasi Dari Histogram Of. *Jurnal Teknik ITS*, 5.
- Raut, N. (2018). Facial Emotion Recognition Using Machine Learning . *Masters Project*. upi.edu. (2005). Fonologi Vokal.