

Aplikasi Pemilihan Model Rambut Pria Berdasarkan Bentuk Wajah Berbasis Android

Stefanus Kevin Joses¹, Henry Novianus Palit²

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

E-mail: stefanus.kj.15@gmail.com¹, hnpalit@petra.ac.id²

ABSTRAK

Saat ini, ketika seseorang pergi ke salon atau ke tempat untuk memotong rambut, seringkali tidak bisa menentukan model rambut. Apabila model rambut yang dipilih tidak sesuai biasanya membuat turunny nilai penampilan seseorang. Karena rambut merupakan salah satu faktor utama dalam nilai penampilan.

Untuk dapat menentukan model rambut yang sesuai, menurut pendapat dari [7] bentuk wajah merupakan faktor yang sangat penting dalam penentuan model rambut, ketidak samaan model rambut dan bentuk wajah akan menyebabkan rendahnya nilai penampilan seseorang. Maka dari itu penulis mengusulkan aplikasi pemilihan model rambut pria berdasarkan bentuk wajah berbasis android.

Hasil pelatihan aplikasi ini dalam menentukan bentuk wajah seseorang memiliki akurasi sebesar 90.952% dan dari 5 hairstylist yang melakukan pengujian terhadap aplikasi ini berkata bahwa aplikasi ini membantu pelanggan untuk menentukan model rambut yang sesuai dengan bentuk wajah mereka.

Kata Kunci: Aplikasi Mobile, Android, Backpropagation, OpenCV, EmguCV, Haar Cascade, Cascade Classifier.

ABSTRACT

Currently, when someone went to barber, they usually can't decide which hair models that they want. When the hairstyle that the hairstylist choose didn't match, it can make people have bad appearances. Because good hairstyle its one of the important part to make good appearances.

To decide hairstyle that looks good on someone, by the opinion of [7] face shape is the most important thing for choosing hairstyle that looks good on someone, if the hairstyle didn't match with their face shape, they will have a bad appearances. So by that factor the writer want to develop an android application for choosing a man's hairstyle based on his face shape.

The accuracy of this application training for determine a man's face shape development is 90.952% and the result from the survey to 5 hairstylist, they say that this application is really help people choose what kind of hairstyle that suit them.

Keywords: Mobile Application, Android, Backpropagation, OpenCV, EmguCV, Haar Cascade, Cascade Classifier..

1. PENDAHULUAN

Penampilan merupakan salah satu faktor utama seseorang untuk mendapatkan kesan pertama yang baik. Kesan ini diperoleh dari

penilaian seseorang terhadap orang lain saat pertemuan pertama yang dapat mempengaruhi sikap atau perilaku seseorang terhadap orang lain. Maka dari itu, penampilan sangat berpengaruh bagi kehidupan seseorang, yakni dalam berinteraksi dengan orang lain terutama dalam hal pekerjaan.

Penampilan yang baik dapat ditentukan dari cara berpakaian seseorang. Namun masih ada faktor lain yang tidak kalah penting dalam berpenampilan yang baik, yaitu model rambut. Seringkali orang pergi ke salon dengan tidak mengetahui model rambut apa yang cocok dengan dirinya. Akhirnya, mereka memilih untuk melihat contoh model rambut orang lain atau menyerahkan keputusan memilih model rambut kepada tukang potong. Terkadang hasil model rambut yang diperoleh cocok, tetapi banyak juga hasil model rambut yang tidak sesuai dengan ekspektasi karena faktor bentuk wajah dan bentuk kepala.

Seiring dengan perkembangan zaman, saat ini teknologi juga semakin berkembang pesat. Hal ini dapat dilihat pada salah satu produk teknologi yang banyak digunakan oleh masyarakat, yaitu *handphone*. Sebelumnya *handphone* hanya digunakan sebagai alat untuk berkomunikasi, akan tetapi saat ini kegunaan *handphone* semakin canggih dimana dapat digunakan untuk bertukar pesan bergambar hingga *entertainment*. Istilah untuk *handphone* tersebut sering kita kenal dengan sebutan "*smartphone*".

Saat ini, sebagian besar *smartphone* yang digunakan masyarakat berbasis Android. Menurut [3] Android merupakan sistem operasi *open source* yang berbasis Linux. Sistem operasi "*open source*" ini memungkinkan banyak pengembang *handphone* dapat mendesain sistem operasinya sendiri sehingga setiap *handphone* memiliki ciri khas yang berbeda [3]. Selain dapat mendesain sendiri kelebihan dari *open source* sendiri adalah lebih flexibel terhadap teknologi baru sehingga memungkinkan munculnya inovasi – inovasi yang baru dan pastinya lebih cepat dari pada sistem operasi yang tidak bersifat terbuka.

Guna mengatasi masalah yang telah dijelaskan sebelumnya, maka dibutuhkan sebuah aplikasi *mobile* yang memudahkan seseorang untuk menentukan model rambut. Aplikasi *mobile* yang diusulkan penulis menggunakan metode *Face Detection* yang dapat mengekstrak bagian wajah dan menempatkannya di sebuah gambar model rambut.

Menurut survei yang dilakukan oleh Raymond James, jumlah pengguna Android sebesar 49,3% dari total penjualan *handphone* yang ada di seluruh dunia sedangkan 46,3% merupakan pengguna iOS [2]. Survei tersebut menyatakan bahwa jumlah pengguna Android lebih besar dari pengguna iOS. Maka dari itu, penulis mengusulkan "Aplikasi Pemilihan Model Rambut Pria

Berdasarkan Wajah Berbasis Android” yang diharapkan dapat menjangkau dan membantu banyak *user*.

2. TINJAUAN PUSTAKA

2.1 Android Studio

Android Studio adalah IDE (*Integrated Development Enviroment*) pemograman *Android* resmi dari google. *Android Studio* merupakan aplikasi yang termasuk *user friendly*, selain dari desainnya fitur yang dimiliki *Android Studio* juga termasuk lengkap dan tidak hanya fitur *Android Studio* juga mempunyai *library* yang sudah ada dan siap dipakai [8].

Fitur yang dimiliki antara lain :

- **Instant Run**

Fitur *Instant Run* merupakan fitur yang memungkinkan compile program lebih cepat. Untuk pertama kali akan memakan waktu sedikit lebih lama tetapi untuk selanjutnya akan lebih cepat daripada biasanya dikarenakan *Android Studio* tidak membuat APK baru.

- **Code Editor Pintar**

Code Editor yang dimiliki oleh *Android Studio* tergolong pintar dikarenakan memiliki fitur *Auto Completion*. Fitur ini memunculkan rekomendasi mengenai *code* yang akan diketikkan. Selain itu *Android Studio* memiliki kemampuan analisa *code* yang mumpuni sehingga memungkinkan percepatan waktu pembuatan program sehingga membuat waktu sang pembuat lebih produktif.

- **Sistem Build yang Handal dan Fleksibel**

Pengguna *Android Studio* dimanjakan dengan kemudahan dalam melakukan *build* aplikasi *Android*, selain itu setelah program di compile untuk pertama kali maka file APK akan sudah secara otomatis terbentuk di dalam smartphone, hal ini membuat kemudahan untuk *run* selanjutnya dan tidak perlu repot untuk *build* aplikasi kembali. Selain itu aplikasi yang telah dibuat dapat diberi *signature* khusus dari sang pembuat.

- **Dapat Membuat Aplikasi untuk Semua Perangkat Android**

Tidak hanya untuk *Handphone* , *Android Studio* dapat digunakan untuk membuat aplikasi untuk *Smartwatch*, *Tablet*, *Android TV*, dan *Android Auto*.

2.2 EmguCV

EmguCV merupakan sebuah wadah *library* yang besar dan memperbolehkan memanggil fungsi dari OpenCV untuk *visual studio windows form application*. EmguCV memiliki peranan yang penting dalam aplikasi training dan aplikasi penamaan model rambut. Didalam emguCV juga terdapat sebuah *library* yaitu *Cascade Classifier* yang berfungsi sebagai pendeteksi wajah [5].

2.3 Backpropagation

Backpropagation merupakan *Multi-Layer Neural Network* yang memiliki 3 tahapan *FeedForward*, *Backpropagation*, dan *Weight update*. *FeedForward* merupakan tahapan dimana input dikalkulasikan dengan weight tiap lapisan untuk mendapatkan *output* . Setelah mendapatkan output, hasil dari output akan di kalkulasikan dengan *error function* untuk menentukan *weight gradients* dari tiap lapisan. Setelah mendapatkan *gradient*, setiap weight yang ada di *network* harus di perbaharui berdasarkan *gradient* [6]. Pada aplikasi ini , *Backpropagation* hanya akan

digunakan saat training penentuan bentuk wajah. Struktur dari *Backpropagation* pada aplikasi ini memiliki 4 (empat) *node* pada *input layer* dan 4 (empat) *node* pada *output layer*, sedangkan untuk jumlah *hidden layer* dan jumlah *hidden node* nanti akan di tentukan melalui percobaan *manual* [1].

Tata nama yang digunakan pada sub-bab kali ini adalah sebagai berikut :

- F_i = Total number of inputs neuron.
- Y_j = Output of the neuron in the hidden layer.
- Y_k = Output of the neuron in the output layer.
- X_i = Input.
- W_{ij} = Weight from input layer to hidden layer.
- X_{jk} = Input of the neuron in the hidden layer.
- W_{jk} = Weight from hidden layer to output layer.
- θ_j = Bias on hidden layer.
- θ_k = Bias on output layer.
- δ_k = Error gradient for output layer.
- δ_j = Error gradient for hidden layer.
- e_k = Error gradient fot output layer.
- α = Learning rate

Tahap pertama dalam *backpropagation* ini disebut dengan *Initialisation*. Tujuan dari tahapan ini adalah memberikan nilai *random* pada *weight* yang ada dengan *range* nilai:

$$\left(-\frac{2.4}{F_i}, +\frac{2.4}{F_i} \right) \quad (1)$$

Tahap kedua disebut dengan *activation* / aktifasi bertujuan untuk menghitung Y_j dan Y_k menggunakan *binary sigmoid function* yang mempunyai *range* (0,1). Rumus *binary sigmoid function* ini didefinisikan seperti ini :

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (2)$$

Tahapan yang terakhir adalah *weight training* dimana tahap ini digunakan untuk menghitung *error gradient* dari *output layer* dan *hidden layer*. Berikut langkah – langkah dari *backpropagation* adalah seperti ini:

Initialisation :

- *Random weight* dari rumus 2.1.

Activation :

- *Hitung output* dari hidden layer

$$Y_j = \text{sigmoid} \left[\sum_{i=1}^n X_i \times W_{ij} - \theta_j \right] \quad (3)$$

- *Hitung output* dari output layer

$$Y_k = \text{sigmoid} \left[\sum_{j=1}^m X_{jk} \times W_{jk} - \theta_k \right] \quad (4)$$

Weight training :

- *Hitung error gradient* yang ada di output layer

$$\delta_k = Y_k \times [1 - Y_k] \times e_k \quad (5)$$

dimana

$$e_k = Y_{d,k} - Y_k \quad (6)$$

- Hitung weight corrections dari output layer

$$\Delta w_{jk} = \alpha \times Y_j \times \delta_k \quad (7)$$

- Weight update dari output layer

$$w_{jk} = w_{jk} + \Delta w_{jk} \quad (8)$$

- Hitung error gradient yang ada di hidden layer

$$\delta_j = Y_j \times [1 - Y_j] \times \sum_{k=1}^l \delta_k \times w_{jk} \quad (10)$$

- Hitung weight corrections dari hidden layer

$$\Delta w_{ij} = \alpha \times X_i \times \delta_j \quad (11)$$

- Weight update dari hidden layer

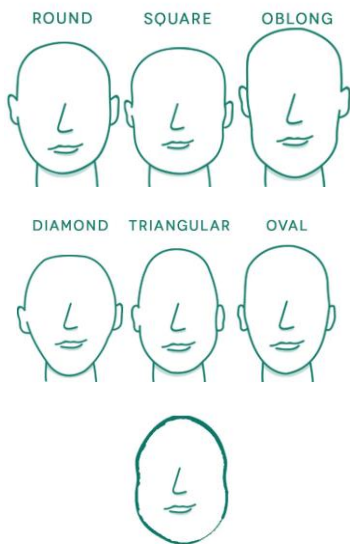
$$w_{ij} = w_{ij} + \Delta w_{ij} \quad (12)$$

2.4 Rumus Yang Akan Digunakan Untuk Menentukan Bentuk Wajah

Untuk dapat menentukan bentuk wajah seseorang, diperlukan 4 ukuran [7] yaitu :

- Dahi: Dihitung dari puncak lengkungan alis satu dan di tarik sampai ujung alis satunya.
- Tulang pipi: Dihitung dari pipi yang paling atas, dari tulang yang menonjol di bawah ujung luar mata
- Rahang: Dihitung dari ujung telinga satu sampai dengan ujung telinga lainnya.
- Panjang wajah: Dihitung dari titik tengah garis rambut di atas dahi hingga ujung dagu.

Pada Gambar 1 terdapat bentuk – bentuk wajah dan kriterianya :



Gambar 1 Template bentuk wajah [7]

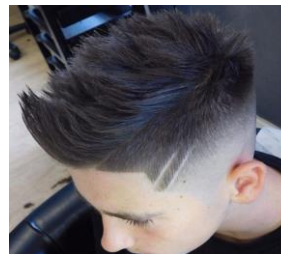
Keterangan [7]:

- Round : Tulang pipi dan panjang wajah memiliki ukuran yang sama. Dahi dan rahang memiliki ukuran yang sama juga tetapi dengan ukuran yang lebih kecil dari pada tulang pipi dan panjang wajah.

- Square : Semua ukuran sama.
- Oblong : Panjang muka paling besar daripada yang lainnya, dan untuk dahi, tulang pipi, rahang memiliki ukuran yang sama.
- Diamond: Panjang muka paling besar lalu untuk ukuran yang lainnya urut dari besar ke kecil, yaitu tulang pipi, dahi, dan paling kecil rahang.
- Triangular: Rahang memiliki ukuran terpanjang, yang kedua adalah tulang pipi, dan untuk dahi merupakan yang paling kecil. Untuk panjang muka bisa berapa saja.
- Oval: Panjang wajah lebih besar daripada panjang tulang pipi dan untuk dahi lebih panjang daripada rahang.
- Pear: Panjang tulang pipi paling besar, kedua rahang, dan yang paling kecil adalah dahi.

2.5 Model Rambut Yang Akan Digunakan

Model rambut yang akan digunakan dinamakan berdasarkan sumber dari website dan pengetahuan penulis, berikut beberapa model rambut yang disediakan dapat dilihat pada Gambar 2:



(a) Spiky



(b) Crew



(c) Pompadour



(d) Mohawk



(e) Curly

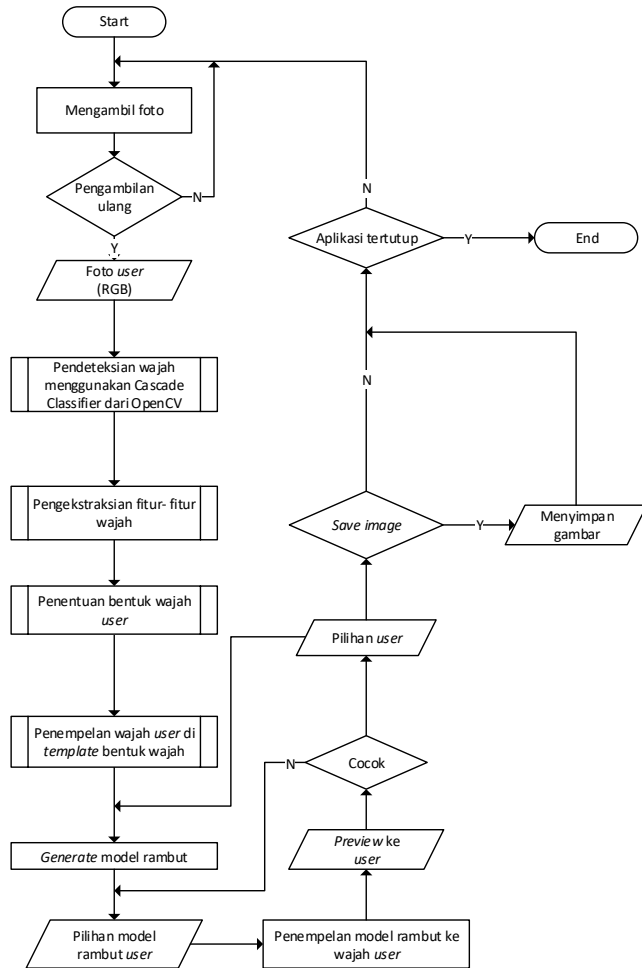
Gambar 2 Model rambut [4]

3. ANALISA DAN DESAIN SISTEM

3.1 Desain Flowchart

Dalam desain sistem ini akan dibahas tentang garis besar sistem kerja perangkat lunak, alur proses metode yang digunakan dalam mencapai tujuan sistem, dan desain tampilan perangkat lunak.

Sistem perangkat lunak yang digunakan untuk mendeteksi bentuk wajah *user* agar dapat menentukan rekomendasi model rambut, memiliki beberapa tahapan proses. Dimulai dari pendeteksian wajah pada *input* gambar yang berasal dari kamera, pendeteksian fitur-fitur penting pada wajah, pengekstraksian fitur penting pada wajah untuk di proses dalam *Feed Forward*, pemangkasan gambar wajah, dan pemilihan model rambut oleh *user* yang telah di rekomendasikan oleh sistem. Desain *flowchart* program utama dapat dilihat di Gambar 4.



Gambar 3 Desain *flowchart* Aplikasi Utama

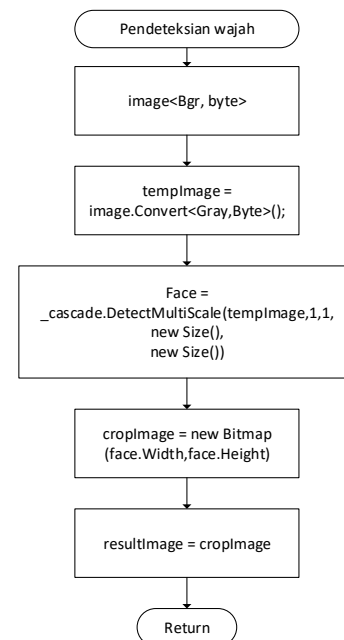
3.2 Flowchart Pendeteksian Wajah

Pendeteksian wajah pada aplikasi *training* ini menggunakan metode yang berbeda dengan di aplikasi pemilihan model rambut. Aplikasi *training* ini menggunakan *library Cascade Classifier* dari *emgucv* seperti yang sudah dijelaskan di Bab 2. Untuk gambaran pendeteksian wajah dapat dilihat pada Gambar 4

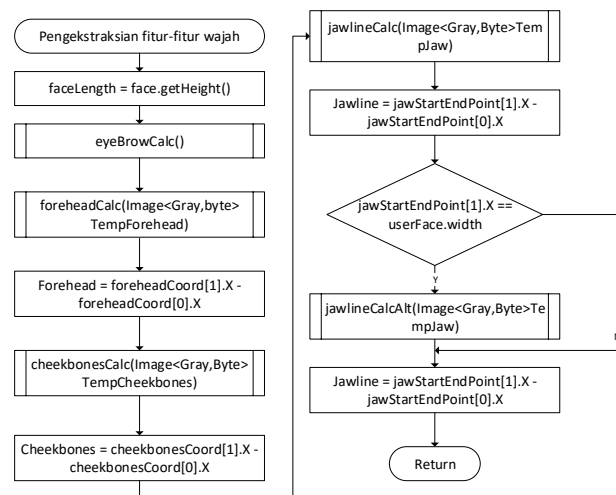
3.3 Flowchart Pengekstraksian Fitur Wajah

Pengekstraksian fitur – fitur wajah akan menggunakan rumus yang sama dengan aplikasi pemilihan model rambut. Sebelum dilakukan ekstraksi gambar terlebih dahulu dimasukkan kedalam fungsi *Canny Edge Detector*, yang memungkinkan untuk mendeteksi setiap garis yang ada pada wajah. *Library* yang digunakan untuk melakukan pengekstraksian adalah *Cascade Classifier* dari

OpenCV. Untuk gambaran dari pengekstraksian fitur wajah dapat dilihat pada Gambar 9.



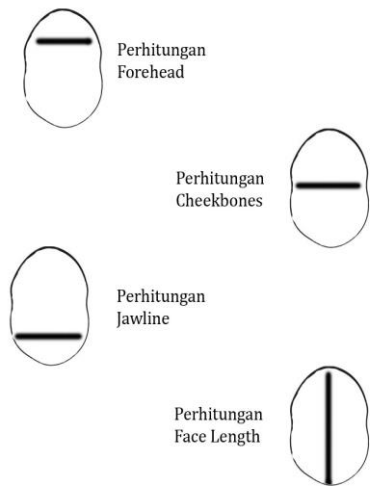
Gambar 4. Desain *flowchart* pendeteksian wajah



Gambar 5 Desain *flowchart* pengekstraksian fitur wajah

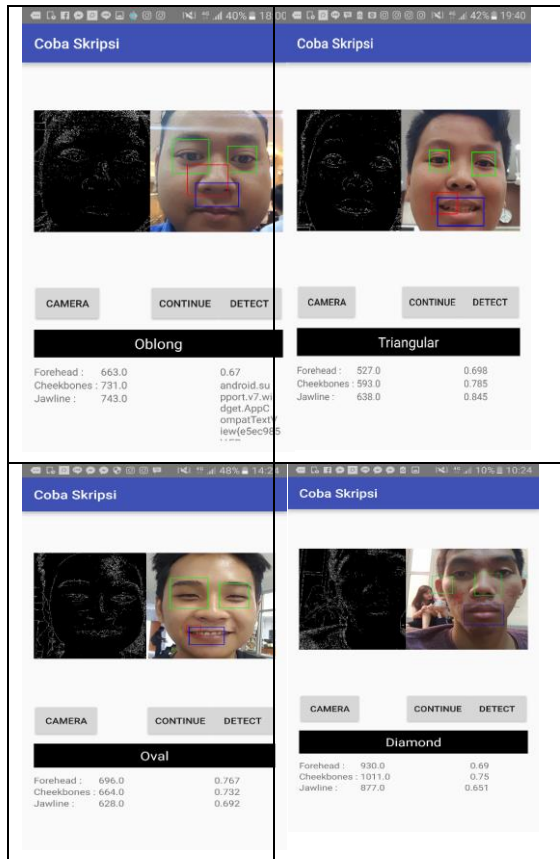
Keterangan:

- Semua nilai yang di dapat akan di bagi dengan lebar wajah, agar syarat awal dari tahapan *Activation* pada *backpropagation* dapat terpenuhi. Syarat ini dapat dilihat pada Rumus 3.
- Untuk perhitungan *faceLength*, tidak menggunakan rumus tertentu karena hasil *crop* area segi empat wajah tepat berada di ujung rambut dan bagian bawah dari dagu, sehingga untuk mendapatkan *faceLength* hanya perlu mendapatkan panjang dari bitmap wajah.
- Rumus untuk menghitung diadaptasi dari rumus yang telah di buat oleh Julia Philips. Untuk cara penghitungan dapat dilihat pada Gambar 6



Gambar 6 Perhitungan wajah

4. PENGUJIAN



Gambar 7 Hasil pendeteksian dan perhitungan wajah user

Tabel 1 Hasil pengujian optimal

Jarak (Cm)	Rotasi sumbu X	Rotasi sumbu Y	Hasil
20	0	0	Oblong
30	0	0	Oblong
40	0	0	Oblong
50	0	0	Oblong
60	0	0	Oblong
70	0	0	Round

Tabel 2 Hasil dari percobaan aplikasi

Jarak (Cm)	Rotasi sumbu X	Rotasi sumbu Y	Hasil
20	0	0	Oblong
20	0	5	Oval
20	0	10	Triangular
20	0	-5	Diamond
20	0	-10	Oval
20	5	0	Oblong
20	5	5	Triangular
20	5	10	Oblong
20	5	-5	Oval
20	5	-10	Square
20	10	0	Oval
20	10	5	Triangular
20	10	10	Diamond
20	10	-5	Triangular
20	10	-10	Round
20	-5	0	Oblong
20	-5	5	Triangular
20	-5	10	Pear
20	-5	-5	Diamond
20	-5	-10	Oblong
20	-10	0	Oval
20	-10	5	Round
20	-10	10	Square
20	-10	-5	Oval
20	-10	-10	Oval

Hasil aplikasi berupa aplikasi Android. Hasil pengujian dari aplikasi ini dapat dilihat pada Gambar 8.

Berikut disertakan juga beberapa hasil dari pengujian ketepatan pendeteksian bentuk wajah. Untuk hasil dari pengujian dapat dilihat pada Tabel 2.

Dari tabel diatas dapat disimpulkan bahwa jarak maksimal dan sudut paling optimal adalah 60cm dan 0 derajat atau dengan kata lain posisi *user* harus tegak lurus dengan posisi kamera. Tetapi pada Tabel 2 dapat dilihat dengan kemiringan – kemiringan tertentu hasil yang didapatkan bisa sama dengan jarak optimal, sebagai contoh pada jarak 40cm dan rotasi terhadap sumbu X sebesar -5 derajat dan rotasi terhadap sumbu Y sebesar -10 derajat hasil yang dihasilkan sangat berbeda jauh dengan hasil optimal yang didapatkan yaitu triangular. Sedangkan syarat untuk bentuk wajah triangular yaitu *jawline* memiliki ukuran yang paling besar dari *cheekbones* dan *forehead*, hal ini memang dimungkinkan dikarenakan ketika kepala *user* menunduk akan membuat *jawline* pengguna terlihat lebih besar dari aslinya, maka dari itu tingkat kemiringan sangat menentukan bentuk wajah. Selain itu faktor kemiringan juga menyebabkan cahaya yang mengenai wajah *user* berubah. Ketika sisi wajah yang lain lebih gelap dari yang lain maka sisi gelap ini dapat tidak dianggap sebagai *edge* oleh *canny* sehingga mengakibatkan penghitungan fitur – fitur wajah menjadi lebih pendek atau bahkan lebih panjang atau malah tidak bisa terdeteksi.

Hasil pengujian *backpropagation* dengan menggunakan 200000 iterasi dapat dilihat pada Tabel 3.

Tabel 3 Hasil pengujian *backpropagation*

Waktu	Jumlah <i>Hidden Node</i>	MSE	Akurasi Keseluruhan	Akurasi Non Training
53:11	14	19.329%	88.095%	46.74%
50:34	15	17.043%	87.619%	46.74%
53:00	16	19.500%	86.666%	52.48%
50:30	17	17.390%	84.761%	30.34%
50:30	18	13.577%	90.952%	36.9%

Berikut dilakukan survey terhadap lima hairstylist yang memiliki pengalaman minimal empat tahun dalam dunia hairstylist dan berada pada dua salon yang berbeda yaitu rudy hadisuwarno dan johnny andreaan. Berikut pertanyaan yang di ajukan kepada hairstylist dan untuk jawaban dari hairstylist dapat dilihat pada Tabel 4:

- Apakah rekomendasi model rambut sudah sesuai dengan bentuk wajah
- Apakah rekomendasi bentuk wajah sudah sesuai dengan bentuk wajah pengguna
- Apakah dengan adanya aplikasi ini membantu pelanggan untuk menentukan model rambut
- Saran terhadap pengembangan aplikasi ini

Tabel 4 Hasil survey dengan hairstylist

Nama	No 1	No 2	No 3	Saran
Dodi	Cocok	Cocok	Ya	Dibuat lebih rapi lagi untuk desainnya
Triana	Cocok	Cocok	Ya	Diperbaiki desain aplikasinya
Ani	Cocok	Cocok	Ya	Diperbanyak model rambutnya
Emma	Cocok	Cocok	Ya	Ditambahkan untuk pewarnaan rambut
Hamian	Cocok	Cocok	Ya	Tidak ada

Dari hasil beberapa percobaan diatas dapat disimpulkan bahwa lamanya proses *training* tidak berpengaruh terhadap hasil akurasi. Begitu juga dengan nilai *error* tidak selalu nilai *error* yang rendah mendapatkan hasil akurasi yang tinggi contohnya dapat dilihat pada bagian non *training*, untuk jumlah *hidden node* 15 dan 17. Untuk data non *training* didapatkan hasil yang rendah dikarenakan gambar yang digunakan merupakan sisa seleksi dari kurang lebih 1500 gambar dan hanya 210 yang dipakai untuk *training* dan gambar sisa 85 merupakan gambar yang memiliki *noise* dan *blur* yang cukup banyak. Maka dari itu diputuskan aplikasi penamaan dan penggolongan dan aplikasi pemilihan model rambut

menggunakan konfigurasi jumlah *hidden node* sebanyak 16, nilai dan syarat *learning rate* merupakan hasil dari pengamatan terhadap *backpropagation*, syarat tersebut merupakan diyakini titik jenuh *neural network* maka dari itu diperlukan perubahan *learning rate* agar kondisi *neural network* tidak menjadi jenuh.

5. KESIMPULAN

Berdasarkan sistem yang telah dikembangkan dan hasil pengujian yang dilakukan maka dapat diambil kesimpulan sebagai berikut:

- Hasil dari pendeteksian wajah dapat menghasilkan hasil yang berbeda – beda walaupun *user* sama. Hal ini bisa disebabkan oleh 4 sebab yaitu jarak antara wajah dan kamera, rotasi sumbu X wajah *user* terhadap kamera, rotasi sumbu Y *user* terhadap kamera, dan kondisi pencahayaan ruangan.
- Aplikasi ini sangat bergantung dengan *device* yang *user* gunakan dikarenakan akan berpengaruh terhadap kualitas gambar yang dihasilkan. Semakin baik kamera yang *user* pakai maka semakin banyak cahaya yang dapat ditangkap oleh kamera.
- Jumlah *hidden layer* dan *learning rate* pada *backpropagation* sangat mempengaruhi hasil akurasi dan *error* yang dihasilkan.
- Jumlah *hidden layer* yang digunakan untuk semua aplikasi berjumlah 18, dan untuk akurasi dengan data yang di *training* sebesar 90.952% dan untuk mse sebesar 13.577%.

6. REFERENSI

- [1] Agarwal, P., & Prakash, N. 2013. An Efficient Back Propagation Neural Network Based Face Recognition System Using Haar Wavelet Transform and PCA. *International Journal of Computer Science and Mobile Computing*, Vol. 2, No. 5, pp. 386-395.
- [2] Edwards, J. 2016. *Apple iOS vs Android Market Share*. URI=<http://www.businessinsider.co.id/apple-ios-v-android-market-share-2016-1/?r=UK&IR=T#Dpah9V4WeZ3rZxCj.97>
- [3] Goodwin, R. 2014. *What is Android? Google's Mobile OS Explained*. URI=<http://www.knowyourmobile.com/google/18749/what-android-googles-mobile-os-explained>
- [4] *Haircut Names For Men – Types of Haircuts*. n.d.. URI=<http://www.menshairstyletoday.com/haircut-names-for-men/>
- [5] Krishna, M. G., & Srinivasulu, A. 2012. Face Detection System On AdaBoost Algorithm Using Haar Classifiers. *International Journal of Modern Engineering Research (IJMER)*, Vol. 2, No. 5, pp. 3356-3560.
- [6] Mikolov, T.A. 2012. Statistical Language Models Based on Neural Networks. *FIT* (pp. 16-20). Czech Republic: University of Technology.
- [7] Philips, J. 2015. *How to Determine Face Shape Men*. URI=<https://www.birchbox.com/guide/article/how-to-determine-face-shape-men>
- [8] Tsani, E. F. 2016. *Kenapa Memilih Android Studio?* URI=<http://egi.degenius.id/kenapa-memilih-android-studio>