

## SISTEM PENGENALAN WAJAH DENGAN METODE LOCAL BINARY PATTERN HISTOGRAM PADA FIREBASE BERBASIS OPENCV

Fajar Setiawan dan Dewi Agushinta R.  
Universitas Gunadarma  
Jl. Margonda Raya No. 100, Depok, Jawa Barat 16424  
mlfauzi9@gmail.com, dewiar@staff.gunadarma.ac.id

### ABSTRAK

Pengenalan wajah merupakan satu teknologi biometrik yang memanfaatkan analisis dari pengolahan citra. Proses dari pengenalan wajah diantaranya adalah deteksi wajah dan identifikasi wajah. Kedua proses ini sebelumnya telah dikembangkan dalam bentuk proses tersendiri. Melalui sistem pengenalan wajah, identitas diri seseorang dapat diketahui dengan mudah hanya dengan memanfaatkan sistem kamera. Algoritma Local Binary Pattern Histogram (LBPH) merupakan satu metode yang digunakan untuk proses pengolahan fitur wajah seperti face detection dan face recognition. LBPH digunakan untuk pengolahan fitur wajah dan Haar Cascade untuk pengambilan frame wajah. Metode LBPH mengubah tekstur dari suatu citra wajah menjadi nilai biner, dan nilai tersebut mewakili bagian dari piksel-piksel suatu wajah yang membentuk sebuah lingkaran dan memiliki pusat sebagai acuan terhadap nilai-nilai tersebut. Jarak antar kerapatan dari nilai biner tersebut dinamakan neighbors. Database yang digunakan adalah database online dari firebase yang dinaungi oleh Google, sehingga dengan terciptanya dataset wajah secara online, pengoperasian pengenalan wajah dapat lebih efisien. OpenCV adalah modul pengolahan citra untuk membuat, mengubah, serta memodifikasi suatu citra digital agar bisa digunakan dalam banyak hal seperti pengolahan citra wajah di sini. Berdasarkan uji coba terhadap 62 citra wajah didapatkan persentase akurasi keseluruhan uji coba proses sistem pengenalan wajah ini sebesar 93.5%.

**Kata Kunci :** Biometrik, Deteksi Wajah, Firebase, Haar, LBPH

### PENDAHULUAN

Sistem pengenalan wajah mempunyai dua aplikasi utama yaitu verifikasi dan identifikasi. Verifikasi semata-mata mencocokkan data baru seseorang dengan data yang ada di basis data (*one to one*) dan umumnya menghasilkan dua keadaan yaitu *true* atau *false*. Sedangkan identifikasi merupakan proses mengenali seseorang dengan keputusan berdasarkan tingkat kedekatan atau kemiripan. Kemampuan mengukur karakteristik fisik atau perilaku yang dapat digunakan untuk memverifikasi atau mengidentifikasi seseorang disebut dengan biometrik. Sistem pengenalan wajah yang sifatnya *real-time* menjadi kebutuhan teknologi yang berorientasi ke masa depan. Teknologi di masa depan akan semakin cerdas, cepat, efisien dan praktis. Dalam perkembangan teknologi global, identifikasi merupakan bagian penting dalam terjaminnya kerahasiaan personal atau data. Tahap kerahasiaan ini akan terjamin dengan

memanfaatkan identifikasi wajah dalam pengaksesan dan pelayanannya [1].

Pengenalan wajah adalah salah satu teknik biometrik yang memungkinkan komputer atau mesin otentikasi untuk mengenali wajah manusia [2][3]. Salah satu aplikasi dari pengenalan wajah adalah identifikasi wajah secara *real-time* menggunakan kamera atau *webcam*, yaitu pengenalan wajah dari suatu citra yang terdiri dari banyak wajah dalam sebuah *dataset* wajah kemudian diaplikasikan keadaan nyata secara *real-time*. Untuk membedakan beberapa wajah manusia dalam suatu citra sangat sulit bagi sistem pengenalan wajah. Selain itu juga terdapat beberapa masalah pada sistem pengenalan wajah otomatis, yaitu ekspresi wajah, iluminasi atau pencahayaan dan jarak [4].

Tulisan ini tentang pembuatan sistem pengenalan wajah yang berbeda dengan mengimplementasikan metode *Haar Cascade* dan *Local Binary Pattern Histograms* (LBPH) yang digunakan di kehidupan nyata seperti purwarupa

identifikasi wajah seseorang yang pernah ada [5][6]. Metode *Haar Cascade* akan digunakan untuk mendeteksi wajah dan LBPH untuk mengekstraksi citra wajah.

Pemanfaatan metode *Haar Cascade*, selain mudah dalam pengaplikasian pada saat deteksi wajah, juga sudah disediakan oleh pengembang, sehingga mudah saat pemrosesan program dan tidak perlu dibuat dari awal [7].

Metode LBPH dapat mencari tekstur dari sebuah gambar kemudian mengambil setiap piksel dari gambar tersebut, sehingga kerapatan antar piksel dapat mengoptimalkan prediksi gambar saat dilakukan identifikasi wajah [8].

Sistem pengenalan wajah ini akan diukur kinerjanya dengan menggunakan kedua metode tersebut. Sistem dinyatakan berhasil jika wajah yang diambil dari *webcam* akan dikenali, ditunjukkan dengan jumlah keseluruhan citra teridentifikasi yang ditampilkan dari citra wajah berikut informasinya adalah sama. *Firebase* dari Google berbasis OpenCV digunakan sebagai dataset untuk pegujiannya [9][10].

#### METODE PENELITIAN

Beberapa langkah penelitian yang dilakukan untuk mendapatkan hasil adalah:

- Studi literatur: perumusan, pembelajaran, dan pengkajian mengenai sistem pengenalan wajah, konsep ekstraksi ciri, dan algoritma pengklasifikasian menggunakan berbagai referensi yang mendukung dalam menganalisis permasalahan yang ada
- Pencarian dan pengumpulan data: membuat suatu *database* yang terdiri dari banyak *image training* atau citra yang akan diujikan.
- Perancangan model sistem: melakukan desain sistem sesuai dengan kebutuhan sistem yang dibuat, hasilnya dituangkan dalam bentuk struktur diagram.
- Implementasi sistem: menerjemahkan hasil rancangan yang telah dilakukan sebelumnya menjadi sistem sebenarnya dengan menggunakan bahasa pemrograman Python.

- Pengujian dan analisis hasil: pengujian terhadap sistem yang telah dirancang dan dibentuk, kemudian dilakukan analisis terhadap sistem berdasarkan keluaran yang dihasilkan dari sistem tersebut. Luaran dari sistem ini dianalisis tingkat akurasi.

Sistem pengenalan wajah yang dibuat diterapkan pada *platform* berbasis *desktop*, dengan mengandalkan kamera *webcam* pengguna sebagai proses identifikasi wajah. Pada saat proses identifikasi wajah, tahapan dimulai dengan pengambilan sampel wajah baru, dan kemudian proses identifikasi wajah itu sendiri. Pada tahapan proses pengambilan sampel wajah, pengguna *admin* akan melakukan pengisian biodata terlebih dahulu terhadap kandidat. Kemudian data ini akan diunggah ke dalam *database google*, yaitu *firebase* sebagai fitur penyimpanan. Setelah data berhasil diunggah akan muncul *pop-up* dari sistem untuk melakukan pengambilan sampel wajah baru. Pada saat proses ini berjalan, sistem akan melakukan deteksi wajah menggunakan fitur dari *Haar Cascade frontal face* yang akan menangkap wajah kandidat dari arah tampak depan.

*Haar cascade* merupakan suatu file berbentuk xml yang digunakan untuk mendeteksi suatu objek, di dalamnya terdapat kumpulan data objek yang telah dilatih/ dikenali. Data ini akan digunakan untuk menguji suatu objek seperti objek wajah yang digunakan untuk menangkap wajah seseorang dari perangkat kamera.

Proses pengambilan sampel menangkap wajah sebanyak 30 kali, penangkapan wajah akan disimpan satu kali ketika sistem berhasil mendeteksi wajah seseorang. Ketika proses pengambilan wajah selesai, akan muncul *pop-up* untuk melakukan *training* data sampel wajah yang telah diambil sebelumnya untuk dapat dikenali sistem. Pada saat proses *training*, metode LBPH dijalankan pada tahap ini. Sistem komputer hanya mengenali kode berupa biner, sehingga data sampel wajah tersebut akan dikonversi oleh metode ini ke dalam bentuk biner. Tahap ini disebut dengan ekstraksi data wajah ke dalam biner oleh metode LBPH. Setelah proses *training*

selesai, tahapan identifikasi wajah dapat dilakukan.

Pengolahan data yang dilakukan merupakan pengolahan data citra yang akan diproses menjadi data citra yang dapat digunakan untuk pelatihan dan pengujian. Pengumpulan data yang dilakukan dengan mengambil citra wajah melalui kamera *streaming* secara *real-time*, kemudian citra wajah tersebut disimpan ke dalam *folder* dan diberi label untuk masing-masing data citra wajah tersebut. Objek penelitian yang digunakan adalah citra wajah laki-laki dan perempuan dengan rasio 17:13 sehingga total objek yang diujikan berkisar 900 data citra wajah. Objek penelitian lainnya untuk uji coba terhadap kemampuan sistem berupa 2 (dua) buah sketsa citra wajah digital berbentuk oval dengan kondisi, satu objek mendekati citra wajah seseorang dan satu objek sketsa wajah murni.

Perancangan model digunakan untuk mengatur *interface* atau tampilan sistem. *Proses Pendeteksian Wajah Dengan Haar Cascade*

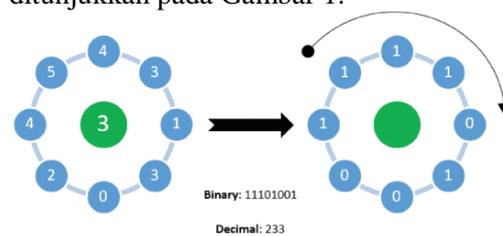
*Haar Cascade* adalah algoritma *machine learning* untuk pendeteksian objek yang digunakan untuk mengidentifikasi objek dalam gambar atau video dan berdasarkan pada konsep fitur yang diusulkan oleh Paul Viola dan Michael Jones [3][7]. *Haar Cascade* adalah pendekatan berbasis pembelajaran mesin di mana fungsi *cascade* dilatih dari banyak gambar positif dan negatif. Kemudian digunakan untuk mendeteksi objek dalam gambar lain. Algoritma ini memiliki empat tahap, yaitu seleksi fitur *Haar*, membuat *integral image* pelatihan Adaboost, dan klasifikasi *cascading* [8].

Proses awal *Haar Cascade* membutuhkan banyak sampel objek yang ingin dibentuk, terdiri dari positif *image* dan negatif *image*. Kemudian kedua data *image* tersebut dilatih sehingga menjadi bentuk file .xml. File inilah yang nantinya akan digunakan untuk mendeteksi suatu objek.

*Proses Ekstraksi Fitur Wajah dengan Metode LBPH*

Metode korelasi membutuhkan waktu komputasi yang besar dan jumlah

penyimpanan yang besar. Oleh karena itu, pengurangan fitur dan representasi wajah diperlukan dalam sistem pengenalan wajah. LBPH merupakan metode yang disukai dalam *computer vision*, *image processing*, dan *pattern recognition*, ini sesuai untuk ekstraksi fitur karena menggambarkan tekstur dan struktur gambar. Metode LBPH dapat diterapkan untuk mewakili gambar wajah dan mengurangi dimensi gambar, mengekstraksi fitur tekstur gambar dengan membagi gambar menjadi wilayah lokal dan mengekstraksi pola biner untuk setiap wilayah lokal. Operator LBP asli bekerja pada delapan tetangga piksel. Gambar akan dibagi menjadi daerah kecil yang disebut sel. Setiap piksel dalam sel dibandingkan dengan masing-masing dari delapan tetangganya. Nilai piksel tengah akan digunakan sebagai nilai ambang. Delapan tetangga-piksel akan ditetapkan ke satu jika nilainya sama dengan atau lebih besar dari piksel tengah, jika tidak, nilainya diatur ke nol [11]. Oleh karena itu, kode LBP untuk piksel tengah dihasilkan dengan menggabungkan kedelapan nilai piksel tetangga (satu atau nol) menjadi kode biner, yang dikonversi ke dimensi desimal 256 untuk kenyamanan sebagai tekstur deskriptor piksel tengahnya. Operator LBP ditunjukkan pada Gambar 1.



Gambar 1. Operator LBP [8]

Berikut formula matematika dari operator LBP [8]:

$$LBP(x) = \sum_{i=1}^8 s(G(x^i) - G(x)) 2^{i-1} \quad (1)$$

$$s(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (2)$$

Tahapan dalam metode LBPH dalam pembuatan sistem adalah :

- Mendefinisikan parameter (radius, tetangga, kisi x dan kisi y) menggunakan struktur parameter dari

paket LBPH. Fungsi *init* dipanggil untuk melewati struktur dengan parameter. Jika parameter tidak diatur, parameter akan diatur *default*.

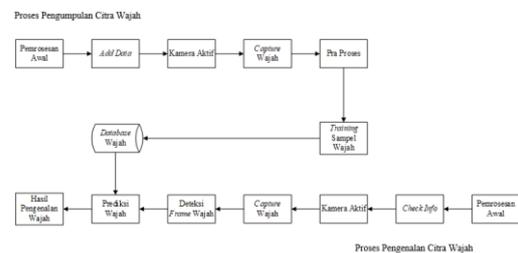
- Melatih metode dengan menjalankan fungsi *train* dari sebuah gambar dan sepotong label berdasarkan parameternya. Semua gambar harus memiliki ukuran yang sama. Label digunakan sebagai ID untuk gambar tersebut, ketika memiliki lebih dari satu gambar dengan tekstur/ subjek yang sama, label harus sama.
- Fungsi *train* pada awalnya akan memeriksa apakah semua gambar memiliki ukuran yang sama. Jika setidaknya satu gambar tidak memiliki ukuran yang sama, fungsi *train* akan mengembalikan kesalahan dan tidak akan dilatih.
- Kemudian, fungsi *train* akan menerapkan operasi LBP dasar dengan mengubah setiap piksel berdasarkan tetangganya menggunakan jari-jari *default* yang ditentukan oleh pengguna.
- Setelah pengambilan fitur wajah pada saat operasi LBP selesai, proses selanjutnya adalah mengekstraksi nilai *histogram* dari setiap piksel gambar yang sudah diambil berdasarkan jumlah *grid* (x dan y) yang dilewati oleh parameter. Setiap *histogram* (dari setiap kisi) hanya akan berisi 256 posisi (0~255) yang mewakili kemunculan setiap intensitas piksel. Setiap *histogram* perlu digabungkan untuk membuat *histogram* baru yang lebih besar. Misalnya *grid* yang dimiliki sebesar 8x8, maka 8x8x256 menjadi 16,384. Nilai tersebut adalah posisi yang akan terbentuk di *histogram* baru. *Histogram* terakhir mewakili karakteristik gambar asli.
- Gambar, label, dan *histogram* akan disimpan dalam struktur data untuk dapat membandingkan semua itu dengan gambar baru pada saat melakukan prediksi.
- Untuk memprediksi gambar baru, perlu memanggil fungsi *predict* lewat gambar sebagai parameter. Fungsi *predict* akan mengekstraksi *histogram* dari gambar

baru, membandingkannya dengan *histogram* yang disimpan dalam struktur data dan mengembalikan label dan jarak yang sesuai dengan *histogram* terdekat jika tidak ada kesalahan yang terjadi.

## HASIL DAN PEMBAHASAN

### Gambaran Umum Sistem Pengenalan Wajah

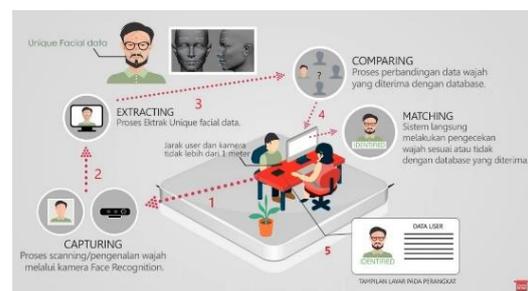
Gambaran umum bagaimana proses yang dilakukan oleh sistem pengenalan wajah, ditunjukkan pada gambar 2.



Gambar 2. Gambaran Umum Sistem

### Proses Pengenalan Wajah

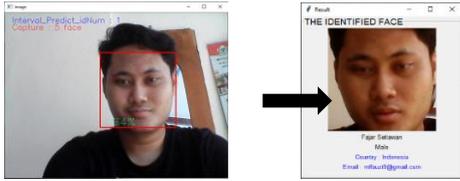
Gambar 3 merupakan tampilan ilustrasi proses pengenalan wajah. Tampilan ini menunjukkan bagaimana alur dari sistem pengenalan wajah berjalan, mulai dari pengambilan data citra wajah, pelatihan data citra wajah ke dalam bentuk biner menggunakan metode LBPH serta identifikasi secara langsung terhadap citra wajah menggunakan kamera *webcam* [6] [11].



Gambar 3. Ilustrasi Proses Pengenalan Wajah [12]

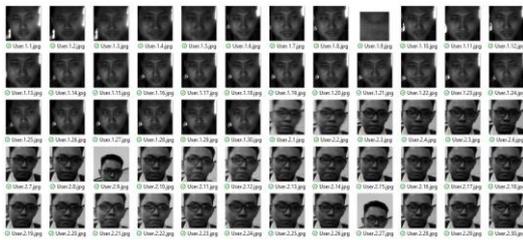
### Pengumpulan Dataset Wajah

Gambar 4 berikut merupakan salah satu tampilan hasil dari proses pengenalan wajah dengan *Haar*.



**Gambar 4.** Tampilan Hasil Proses Pengenalan Wajah

Proses pengenalan citra wajah berhasil ketika jumlah pengambilan gambar telah terpenuhi. Jumlah gambar yang diambil sebanyak 30 sampel yang kemudian disimpan dalam folder “dataset” seperti pada Gambar 5.



**Gambar 5.** Dataset Kumpulan Citra Wajah

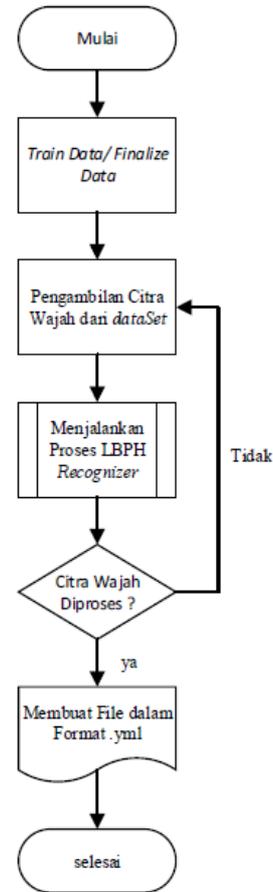
#### Training Dataset Wajah

Proses *training dataset* wajah merupakan bagian dalam pembuatan *database* wajah yang diambil dari citra wajah *user* yang berada pada folder “dataset”. Citra tersebut kemudian diolah menjadi data matriks dalam bentuk *bytearray* dan dari seluruh matriks wajah yang terkumpul dalam folder *dataset* tersebut disatukan menjadi sebuah file *.yml*. Langkah dalam diagram alir dari proses *training* wajah dapat dilihat di gambar 6.

#### Hasil Pengujian

Hasil pengujian dilakukan dengan menggunakan metode LBPH. Setelah seluruh citra diproses maka didapat hasil uji coba deteksi wajah. Uji coba dilakukan pada tiga jenis citra, yaitu uji coba terhadap (1) data citra wajah asli, (2) citra wajah digital yang berasal dari foto *smartphone*, dan (3) sketsa citra wajah berbentuk oval menggunakan dua citra uji yang berbeda dengan data latih.

Citra wajah asli merupakan pengambilan citra wajah yang diambil dari wajah seseorang secara langsung melalui kamera *webcam*.



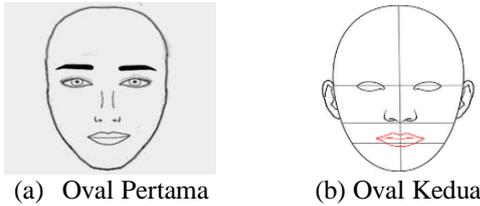
**Gambar 6.** Proses Training Wajah

Hasil pengujian deteksi citra wajah dari ketiga jenis citra dihitung persentase keberhasilan dengan rumus :

$$\text{Persentase keberhasilan} = \frac{\text{Jumlah citra wajah yang sesuai}}{\text{Jumlah uji coba citra wajah}} \times 100\% \quad (3)$$

Dari 30 citra wajah yang diujikan terhadap citra wajah asli pada uji coba (1) didapatkan hasil bahwa semua 30 citra tersebut sesuai. Akurasi persentase keberhasilan didapatkan sebesar 100%. Sedangkan dari uji coba (2) menunjukkan hasil sebanyak 27 citra yang sesuai dan 3 (tiga) citra yang tidak sesuai. Akurasi persentase keberhasilan didapatkan sebesar 90%. Uji coba (3) menggunakan sketsa wajah oval yang berbeda dengan data citra wajah yang berada di dalam *dataset* wajah, dapat dilihat pada Gambar 7. Hasil pengujian terhadap 2 (dua) citra wajah oval yang diujikan didapatkan hasil 1 (satu)

wajah yang sesuai dan 1 (satu) wajah yang tidak sesuai.



(a) Oval Pertama (b) Oval Kedua  
**Gambar 7. Sketsa Wajah Oval**

Dari jumlah seluruh citra yang diujikan menggunakan metode LBPH yaitu 62 citra dengan menggunakan tiga pengujian berbeda, tingkat akurasi dapat dihitung dengan menggunakan rumus :

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (4)$$

**Tabel 1. Nilai Prediksi**

		Nilai Sebenarnya	
		True	False
Nilai Prediksi	True	TP (True Positive) Correct Result	FP (False Positive) Unexpected Result
	False	FN (False Negative) Missing Result	TN (True Negative) Correct Absence of Result

- *True Positive* (TP): Wajah pengguna dikenali oleh sistem.
- *True Negative* (TN): Sistem mendeteksi selain wajah.
- *False Positive* (FP): Sistem salah mengenali wajah.
- *False Negative* (FN): Wajah pengguna tidak dikenali oleh sistem

**Tabel 2. Hasil Nilai Prediksi**

		Nilai Sebenarnya	
		True	False
Nilai Prediksi	True	58	4
	False	0	0

$$Accuracy = \frac{58 + 0}{58 + 0 + 4 + 0} = 93.5\%$$

Dari hasil pengujian yang telah dilakukan, maka didapatkan hasil keseluruhan. Perhitungan dari Tabel 1 dan Tabel 2 menunjukkan hasil cukup baik yaitu nilai akurasi sebesar 93.5% untuk hasil persentase seluruh jumlah citra yang diujikan dengan menggunakan metode LBPH.

## PENUTUP

Sistem pengenalan wajah berjalan cukup baik. Sistem mampu melakukan pendeteksian sekaligus pengenalan pada wajah pengguna berdasarkan *database* wajah yang ada. Sistem juga mampu melakukan pencocokkan yang cukup baik antara input wajah melalui kamera dengan *database* wajah yang berisi citra wajah pengguna (model). Pemanfaatan metode *Local Binary Pattern Histogram* (LBPH) ke dalam sistem bekerja cukup baik. Metode ini merupakan metode operator tekstur yang sederhana namun sangat efisien yang memberi label piksel pada suatu gambar dengan cara menetapkan lingkungan dari setiap piksel dan menganggap hasilnya sebagai *biner*.

Sistem pengenalan wajah ini dapat dikembangkan dengan menambahkan fasilitas *upload data user* untuk memudahkan pengumpulan *dataset* wajah oleh pengguna, sehingga pengumpulan wajah tidak bergantung pada data riil pada saat itu juga, namun dapat menambahkan dari data lampau yang sudah ada. Untuk masalah yang lebih kompleks seperti keamanan pada identitas di bandara dapat menggunakan metode yang lebih baik lagi seperti metode *Neural Network*, *Machine Learning*, atau lainnya.

## DAFTAR PUSTAKA

- [1] A. Fadlil dan S. Yeki, "Sistem Verifikasi Wajah Menggunakan Jaringan Saraf Tiruan Learning Vector Quantization", *Jurnal Informatika*, 4(2) 480–487, 2010.
- [2] L. K. P. Saputra, "Perbandingan Varian Metode Multiscale Retinex Untuk Peningkatan Akurasi Deteksi Wajah Adaboost HAAR-like", *Jurnal Teknik Informatika dan Sistem Informasi*, 2(1) 89–98, 2016.
- [3] M. D. Putro, T. B. Adji, dan B. Winduratna, "Sistem Deteksi Wajah dengan Menggunakan Metode Viola-Jones", *Science, Engineering and Technology*, 1–5, 2012.

- [4] D. Agushinta R., A. Suhendra, S. Madenda, and Suryadi, "Face Component Extraction Using Segmentation Method on Face Recognition System", *Journal of Emerging Trends in Computing and Information Sciences*, 2(2) 67-72, 2011.
- [5] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face Description with Local Binary Patterns: Application to Face Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12) 2037-2041, 2007.
- [6] M. A. Rahim, M. N. Hossain, and T. Wahid, "Face Recognition Using Local Binary Pattern (LBP)", *Global Journal of Computer Science and Technology Graphics & Vision*, 13 1-8. 2013.
- [7] P. Viola and M. Jones, "Rapid Object Detection Using A Boosted Cascade of Simple Features", *Proceeding IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.
- [8] M. A. Abuzneid and A. Mahmood, "Enhanced Human Face Recognition Using LBPH Descriptor, Multi-KNN, and Back-Propagation Neural Network", *IEEE Access*, 6 20641-20651, 2018.
- [9] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media, 2008.
- [10] A. Dash and B. N. Tripathy, *Prototype Drowsiness Detection System*. National Institute of Technology, Rourkela, 2012.
- [11] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Pattern", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) 971-987, 2002.
- [12] Keviantr. 2016. *Face Recognition System*, url: <https://www.youtube.com/watch?v=jC9OCXesYDs>, diakses pada tanggal 06 september 2019.