
Perancangan Aplikasi Scan Wajah Menggunakan Algoritma Eigenface Sebagai Alat Bantu Pengawas Ujian Agen Asuransi Pada PT. Adisarana Wanaartha Life Cabang The District Medan

Yeni Riani Noviana Gulo. ^{#1}, Puji Sari Ramadhan^{#2}, Dr. Asyahri Hadi Nasyuha ^{#3}

^{#1} Program Studi Sistem Informasi, STMIK Triguna Dharma

^{#2,3} Program Studi Dosen Sistem Informasi, STMIK Triguna Dharma

Article Info

Article history:

Received Jun 12th, 201x

Revised Aug 20th, 201x

Accepted Aug 26th, 201x

Keyword:

Face Recognition

Eigenface

Wajah

ABSTRACT

Dalam mengenali wajah, manusia memiliki kemampuan yang luar biasa. Manusia mampu mengenali sampai ribuan wajah dan bisa mengingat dan mengenalinya walaupun setelah bertahun - tahun. Kemampuan dalam mengenali wajah yang dimplementasikan pada suatu alat atau sistem akan memberikan banyak manfaat pada kehidupan saat ini. Berbagai aplikasi dari alat dengan kemampuan seperti ini terbentang luas dari pencarian penjahat, kriminalitas, sistem akses keruangan, sampai interaksi manusia dengan komputer. Dalam tugas akhir ini akan dilaksanakan perancangan aplikasi scan wajah manusia dengan identifikasi wajah manusia sebagai media pengenalnya atau yang lebih dikenal sebagai face recogniton. Pengenalan wajah akan menggunakan kamera untuk menangkap wajah seseorang kemudian dibandingkan dengan wajah yang sebelumnya telah disimpan di dalam database tertentu. Secara garis besar proses dari pengenalan wajah ini adalah kamera webcam melakukan capture wajah. Kemudian didapatkan sebuah nilai R,G,B. Dengan menggunakan pemrosesan awal, dilakukan crop, konversi RGB ke Grayscale. Setelah dilakukan proses Grayscale, dilakukan tahap pengolahan wajah dengan menggunakan metode eigenface. Didalam metode eigenface ini terdapat beberapa tahapan inti yaitu: mengubah wajah menjadi matriks, menghitung rata-rata FlatVector, menentukan nilai eigenface dan melakukan proses identifikasi wajah dengan mencari nilai eigenface yang mendekati. Pengenalan wajah ini nantinya dapat dikembangkan untuk menjadi aplikasi absensi yang dapat diterapkan diperusahaan untuk mencegah manipulasi.

Kata Kunci: Face Recogniton, Eigenface, Wajah

Copyright © 2021 STMIK Triguna Dharma.

All rights reserved.

Corresponding Author:

Nama : Yeni Riani Noviana Gulo

Program Studi : Sistem Informasi

STMIK Triguna Dharma

Email : yeninovianag@gmail.com

1. PENDAHULUAN

Asuransi Jiwa Adisarana WanaArtha Life adalah sebuah perusahaan yang bergerak di bidang asuransi yang lebih dari 40 tahun telah melayani masyarakat Indonesia. Produk yang ditawarkan Asuransi Jiwa Adisarana WanaArtha Life beragam diantaranya asuransi jiwa tradisional dan produk deposito yang memiliki nilai investasi dan asuransi jiwa.

Kesuksesan suatu perusahaan asuransi tidak lepas dari kinerja seorang agen, sehingga perusahaan asuransi saling bersaing untuk merekrut agen - agen berkualitas begitu pula dengan Asuransi Jiwa Adisarana WanaArtha Life. Dalam proses perekrutan agen asuransi ada sebuah proses dimana agen harus mengikuti ujian yang disebut ujian AAJI (Asosiasi Asuransi Jiwa Indonesia) dimana sebelumnya agen – agen yang akan mengikuti ujian sudah di beri bekal dengan mengikuti training yang sudah di persiapkan oleh perusahaan – perusahaan asuransi. Ujian tersebut dilakukan agar agen mendapatkan sertifikat dan lisensi keagenan, lisensi ini seperti sebuah SIM kendaraan bermotor, dimana yang boleh mengendarai kendaraan bermotor harus lulus ujian SIM, demikian dengan AAJI adalah SIM nya seorang agen asuransi dan harus diperpanjang sesuai dengan masanya.

Namun dalam proses ujian sering kali pengawas ujian mendapatkan kesulitan ketika melakukan verifikasi data agen jika dilakukan dengan cara manual, terlebih disaat ada agen yang datang di penghujung waktu pengawas akan kesulitan melakukan verifikasi data karena pengawas juga harus mengejar waktu supaya agen tidak terlambat masuk ke ruang ujian dan tetap bisa mengikuti ujian AAJI.

Absensi yang menggunakan sistem *scan* wajah adalah sistem absensi yang dapat dilakukan dengan cara memadukan algoritma pengenalan wajah, yang mana algoritma tersebut nantinya akan dapat mengenali wajah seseorang kemudian dapat dipadukan dengan sistem absensi.

Dalam sistem absensi menggunakan pengenalan wajah terdapat beberapa metode yang digunakan. Misalnya metode *Haar Cascade Classifier* yang pernah digunakan oleh Dwiparaswati Implementasi *Face Recognition* secara *Real-time*. Dalam penelitian Dwiparaswati *Haar Cascade Classifier* memproses gambar dalam kotak - kotak, dimana dalam satu kotak terdapat beberapa pixel. Per kotak itu pun kemudian di-proses dan didapatkan perbedaan nilai (*threshold*) yang menandakan daerah gelap dan terang. Nilai – nilai inilah yang nantinya dijadikan dasar dalam *image processing* [1]. Namun dalam penelitian Dwiparaswati dengan berbagai dekomposisi *Haar Cascade Classifier* tingkat keberhasilan yang di dapat masih menunjukkan angka di bawah 80%.

Metode *Eigenface* adalah suatu algoritma pengenalan pola wajah yang berorientasi pada *Principle Component Analysis (PCA)*. Prinsip dasar dari sebuah pengenalan wajah adalah dengan melakukan *encode* kemudian dilakukan perbandingan dengan hasil *decode* yang sebelumnya telah dilakukan [2].

Metode Algoritma *Eigenface* dalam sistem pengenalan atau pendeteksi wajah yang termasuk dalam ranah pengolah citra adalah salah satu algoritma yang cukup populer dalam menyelesaikan masalah pengenalan wajah manusia, prinsip dasar pengenalan wajah adalah mengutip informasi unik wajah, kemudian di lakukan normalisasi wajah untuk mendapatkan nilai matriks pada wajah setelah itu dihitung nilai *eigenface* nya dan dibandingkan dengan hasil perhitungan nilai *eigenface* wajah yang akan di deteksi.

Metode *Eigenface* sudah digunakan di beberapa penelitian. Keberhasilan yang diperoleh dari penelitian pengenalan wajah dengan menggunakan metode *Eigenface* yang dilakukan oleh Wiryadinata sudah mencapai 85%

Python adalah salah satu bahasa pemrograman tingkat tinggi yang bersifat *interpreter, interactive, object - oriented*, dan dapat beroperasi hampir di semua *platform: Mac, Linux, dan Windows*. Python termasuk bahasa pemrograman yang mudah dipelajari karena sintaks yang jelas, dapat dikombinasikan dengan penggunaan modul - modul siap pakai, dan struktur data tingkat tinggi yang efisien [3].

Bahasa pemrograman *Python* adalah bahasa pemrograman yang mudah dibaca dan terstruktur, hal ini karena di gunakannya sistem indentasi. Yaitu memisahkan blok - blok program susunan indentasi. Jadi untuk memasukan sub - sub program dalam suatu blok, sub - sub program tersebut diletakkan satu atau lebih spasi dari kolom suatu blok program [4].

OpenCV (*Open Source Computer Vision Library*) adalah perpustakaan komputer *open source visi* dan perangkat lunak pembelajaran. OpenCV dibangun untuk menyediakan infrastruktur umum untuk aplikasi visi komputer dan untuk mempercepat penggunaan persepsi mesin dalam produk komersial [5].

2. METODE PENELITIAN

Metologi penelitian ialah tata cara yang disusun secara pasti, sistematis dan logis sebagai landasan untuk suatu kegiatan tertentu. Metode penelitian dilakukan agar dapat mendapatkan gambaran rancangan dari

penelitian yang meliputi antara lain: prosedur dan langkah - langkah yang harus ditempuh, waktu penelitian, sumber data, dan bagaimana data - data tersebut diperoleh dan selanjutnya diolah dan dianalisis.

Dalam pelaksanaan penelitian ini dilakukan dengan beberapa metode yaitu :

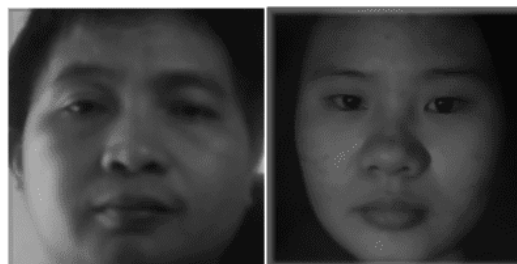
1. Observasi

Observasi ini dengan melakukan tinjauan secara langsung di PT. Adisarana Wanaartha Life Cabang The District Medan.

2. Wawancara

Dalam mengumpulkan data dan mendapatkan informasi dari PT. Adisarana wanaartha Life Cabang The District juga dilakukan pertemuan dan wawancara kepada pihak - pihak yang nantinya akan berhubungan dengan aplikasi yang akan dikembangkan. Pihak - pihak yang dimaksud adalah *Senior agency director* PT. Adisarana Wanaartha Life Cabang The District Bapak David Ng, yang telah meluangkan waktunya untuk melakukan Wawancara yang dilakukan pada tanggal 28 April 2021.

Data yang akan digunakan pada perancangan aplikasi ini adalah berupa data dua citra wajah yang di *capture* menggunakan *webcam* dengan ukuran 3×3 piksel maka *eigenvector* akan memiliki ukuran 2×9 ($2 \times 3 \times 3$). Dua buah citra wajah yang telah diubah menjadi matriks, kemudian diubah kedalam bentuk rataan *flatvector*. Data – data yang telah dikumpulkan kemudian akan dilakukan proses perhitungan untuk mencari nilai *eigenface*-nya.

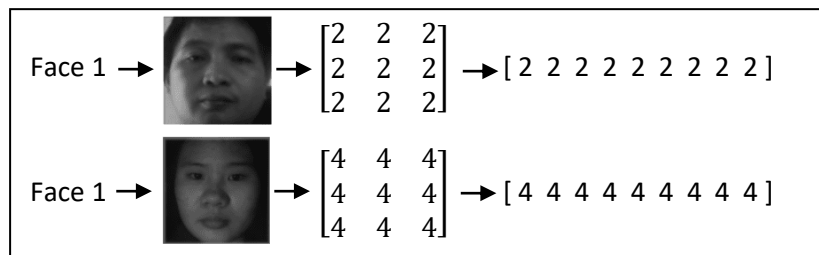


Gambar 3.3 Citra Wajah *Training*

Berikut ini adalah langkah – langkah yang dilakukan dalam perhitungan untuk mencari nilai *eigenface* nya :

1. Penyusunan Flat Vector

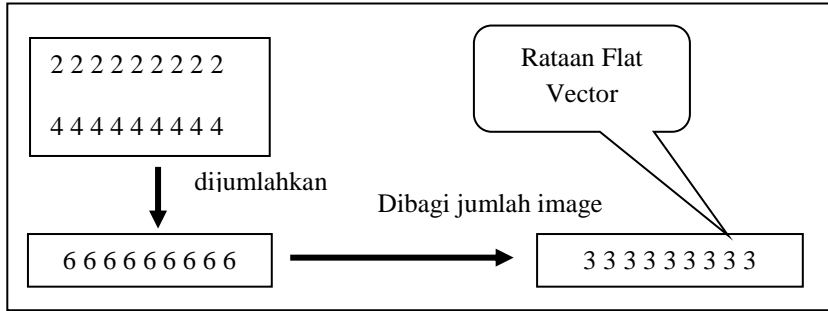
Langkah pertama adalah menyusun seluruh *training image* menjadi 1 matriks tunggal. Misalkan citra berukuran $H \times W$ piksel dan jumlahnya N buah, maka *flatvector* yang dihasilkan berdimensi $N \times (H \times W)$. Selanjutnya representasikan semua matriks *training* menjadi matriks dengan bentuk $N \times 1$ atau matriks linier.



Gambar 3.4 Penyusunan *Flatvector*

2. Penghitungan Rataan FlatVector

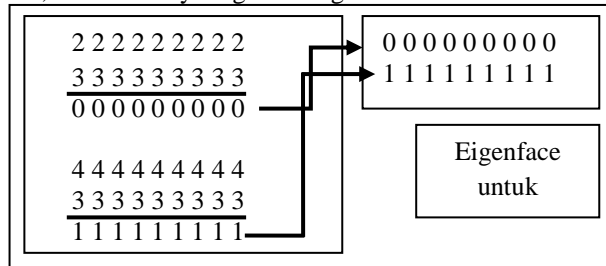
Setelah masing – masing *flatvector* citra wajah *training* didapatkan, jumlahkan seluruh *flatvector* sehingga memperoleh matriks berukuran $1 \times (H \times W)$. Setelah itu matriks tersebut dibagi dengan jumlah *image trainig* (N). Nilai *flatvector* citra akan digunakan untuk menghitung nilai *eigenface* citra wajah (*training image*).



Gambar 3.5 Penentuan Rataan Flatvector

3. Tentukan Nilai Eigenface

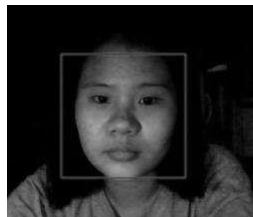
Untuk menghitung nilai *eigenface* nya dilakukan dengan menggunakan rata-rata *flatvector* citra yang telah didapatkan, dengan cara mengurangi baris - baris pada matriks *flatvector* dengan rata-rata *flatvector*. Jika didapatkan nilai di bawah nol, maka nilainya diganti dengan nol.



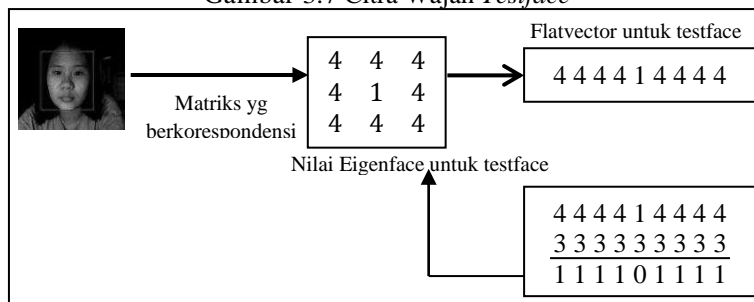
Gambar 3.6 Perhitungan Nilai Eigenface Untuk Training

4. Proses Identifikasi

Pada proses ini akan digunakan 1 *sample* citra wajah sebagai *testface* yang akan dihitung juga nilai *eigenface* nya untuk matriks *testface* dengan cara sebelumnya untuk menentukan nilai *eigenface* dan *flatvector* citranya.

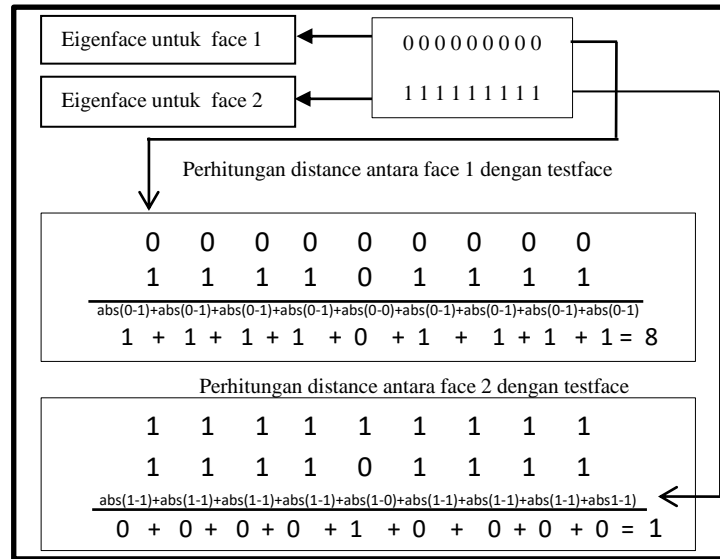


Gambar 3.7 Citra Wajah Testface



Gambar 3.8 Perhitungan Nilai Eigenface Untuk Testface

Dalam melakukan proses identifikasi dilakukan dengan menentukan jarak terpendek antara *eigenface test* dan *eigenvector training image*. Pertama tentukan nilai *absolut* dari pengurangan baris pada matriks *eigenface training image* dengan *eigenface* dari *testface*, kemudian jumlahkan elemen - elemen penyusun *vector* yang dihasilkan dari pengurangan tadi untuk menghasilkan jarak indeks, lakukan untuk semua baris kemudian, cari nilai yang terkecil.



Gambar 3.9 Proses Identifikasi Dengan *Input Image Testface*

Dari hasil perhitungan, diperoleh jarak citra *face 2* memiliki nilai yang terkecil yaitu satu, maka hasil identifikasi menyimpulkan bahwa *testface* lebih mirip dengan *face* dua dari pada *face* satu.

3. ANALISA DAN HASIL

Hasil tampilan antar muka adalah tahapan dimana sistem atau aplikasi siap untuk dioperasikan pada keadaan yang sebenarnya sesuai dari hasil analisis dan perancangan yang dilakukan, sehingga akan diketahui apakah sistem atau aplikasi yang dibangun dapat menghasilkan suatu tujuan yang dicapai, dan aplikasi *Scan Wajah* ini dilengkapi dengan tampilan yang bertujuan untuk memudahkan penggunaannya. Fungsi dari *interface* (antarmuka) ini adalah untuk memberikan *input* dan menampilkan *output* dari aplikasi. Pada aplikasi ini memiliki *interface* yang terdiri dari *Form login*, *Form Admin*, *Form Pengawas* dan *Form Laporan*.

3.1 Tampilan Form Login

Tampilan *Login* merupakan tampilan yang pertama kali muncul dan berfungsi untuk proses otorisasi *administrator* agar aplikasi terhindar dari *user-user* yang tidak bertanggung jawab sebelum masuk ke *Form Utama*.



Gambar 5.1 Tampilan *Form Login*

3.2 Tampilan Form Admin

Tampilan *Form Admin* merupakan tampilan yang muncul setelah admin *login* dengan *user id* dan *password* yang sudah di verifikasi dan benar.

Gambar 5.2 Tampilan *Form Admin*

Dalam *Form Admin* terdapat tombol – tombol yang memiliki masing – masing Fungsi diantaranya adalah :

1. Tombol Cari
Berfungsi untuk Mencari data Peserta Ujian yang sudah di daftar dengan memasukkan kata kunci pada kolom *keyword* lalu menekan tombol cari.
2. Tombol Hapus
Berfungsi untuk Menghapus data Peserta Ujian yang sudah di daftar dengan cara memilih salah satu data peserta yang ada pada tabel daftar peserta ujian kemudian menekan tombol hapus.
3. Tombol Tambah
Berfungsi untuk Menambah data Peserta Ujian dengan cara mengisi terlebih dulu *form* data yang ada pada tampilan utama admin kemudian menekan tombol tambah.
4. Tombol Training
Berfungsi untuk Melatih citra wajah dan disimpan di *database* sebagai citra wajah *training*.
5. Tombol Laporan
Berfungsi untuk menampilkan *form* laporan.

3.3 Tampilan *Form Pengawas*

Tampilan *Form Pengawas* merupakan tampilan yang muncul setelah pengawas *login* dengan *user id* dan *password* yang sudah di verifikasi dan benar.

Gambar 5.3 Tampilan *Form Pengawas*

Dalam *Form* Pengawas terdapat tombol – tombol yang memiliki masing – masing Fungsi diantaranya adalah :

1. Tombol Cari
Berfungsi untuk Mencari data Peserta Ujian yang sudah di daftar dengan memasukkan kata kunci pada kolom *keyword* lalu menekan tombol cari.
2. Tombol Deteksi
Berfungsi untuk deteksi wajah agen pada saat ujian AAJI untuk mengetahui apakah agen sudah terdaftar atau belum.

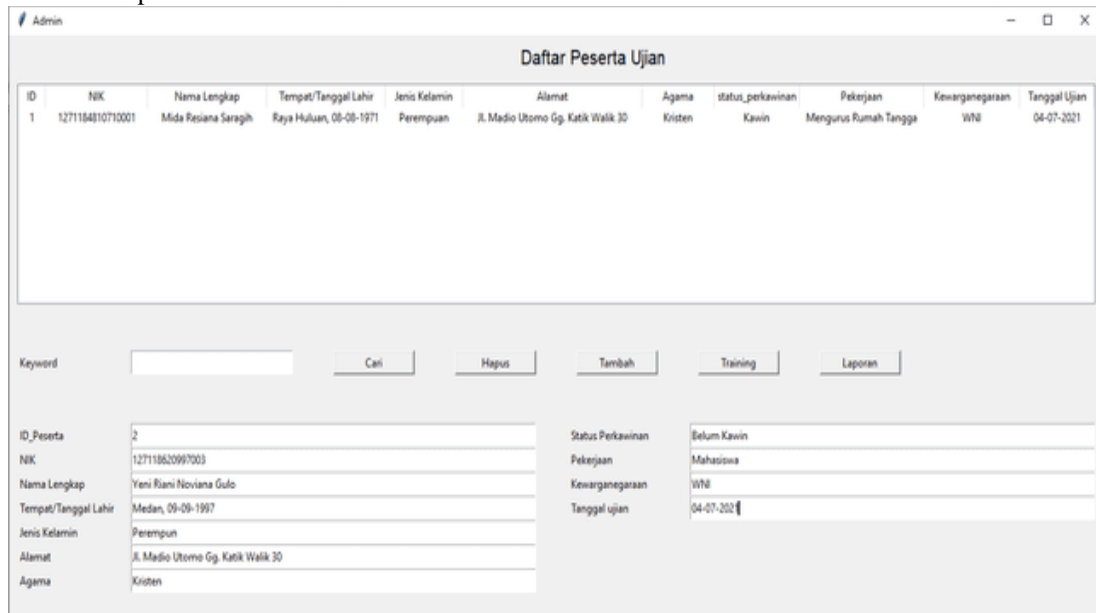
3.4 Pengujian

Pada bagian ini dilakukan pengujian dengan *sample* data baru atau adanya penambahan *record* data dari hasil pengolahan data sementara. Dalam memasukkan data peserta ujian, maka adapun hasil proses program *scan* wajah adalah sebagai berikut:

3.4.1 Pengujian Software

Pada uji coba *software* ini dikhususkan pengujian terhadap *form* Admin, *Form* Pengawas, Pengisian data peserta ujian, *scan* absensi, dan *form* laporan, karena pada *form - form* tersebut merupakan proses terpenting untuk aplikasi pengenalan wajah. Pada *form* admin ini akan menampilkan proses cari, hapus, tambah, *training*, dan laporan.

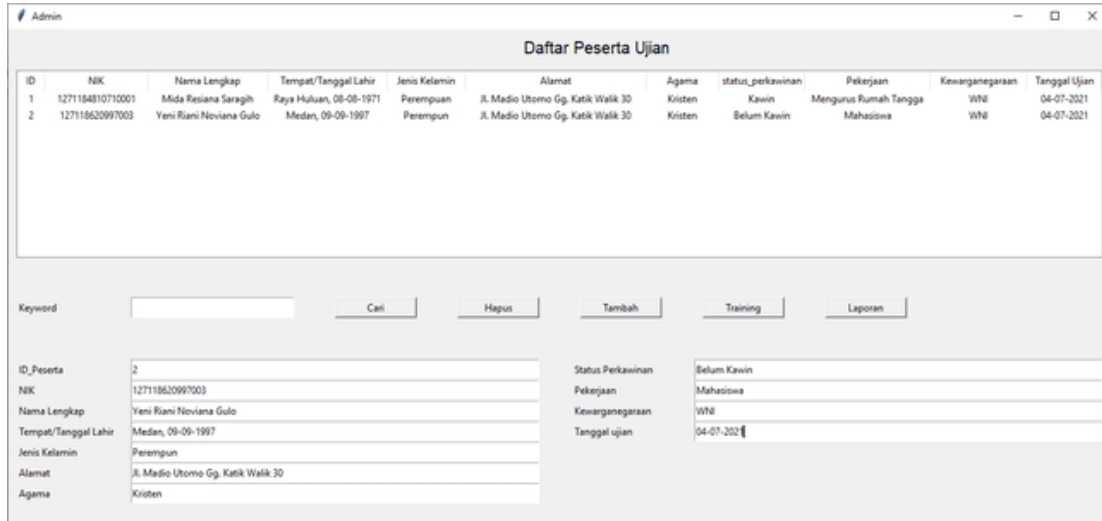
Proses pertama yang dilakukan adalah proses tambah, data peserta ujian akan diisi terlebih dahulu pada *form* admin selanjutnya admin menekan tombol tambah kemudian sistem akan mendeteksi ada tidaknya wajah yang di-*input* oleh *webcam* untuk mengambil gambar. Setelah wajah terdeteksi selanjutnya proses pengambilan gambar dilakukan sebanyak 50 kali, setelah itu data Peserta ujian dan citra wajah akan disimpan pada *database* peserta.



Gambar 5.4 Pengisian *Form* Data Peserta

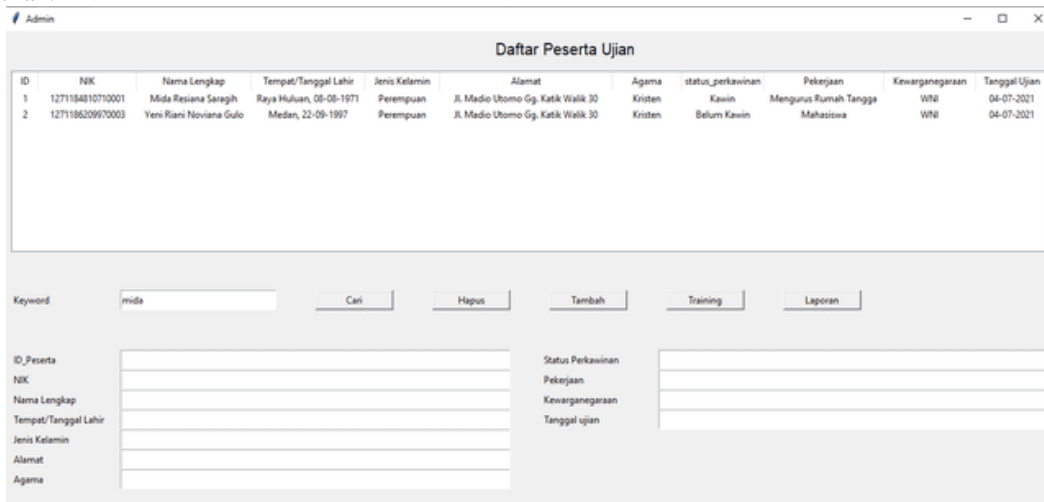


Gambar 5.5 Pengambilan Citra Wajah

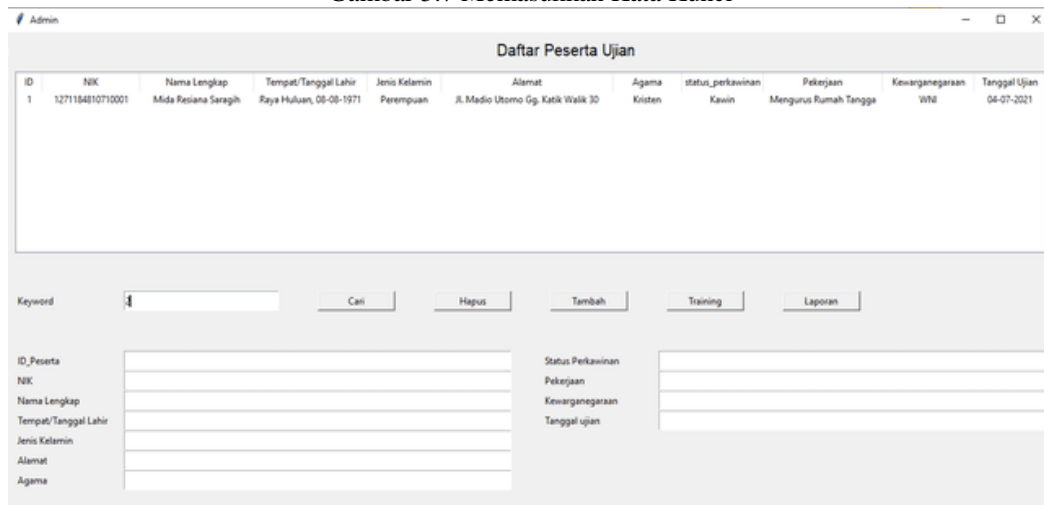


Gambar 5.6 Data Peserta Dan Citra Wajah Sudah Tersimpan Di Database

Proses kedua adalah proses cari, dimana admin melakukan pencarian dengan memasukkan kata kunci pada kolom *keyword* kemudian admin menekan tombol cari selanjutnya sistem akan melakukan proses pencarian.



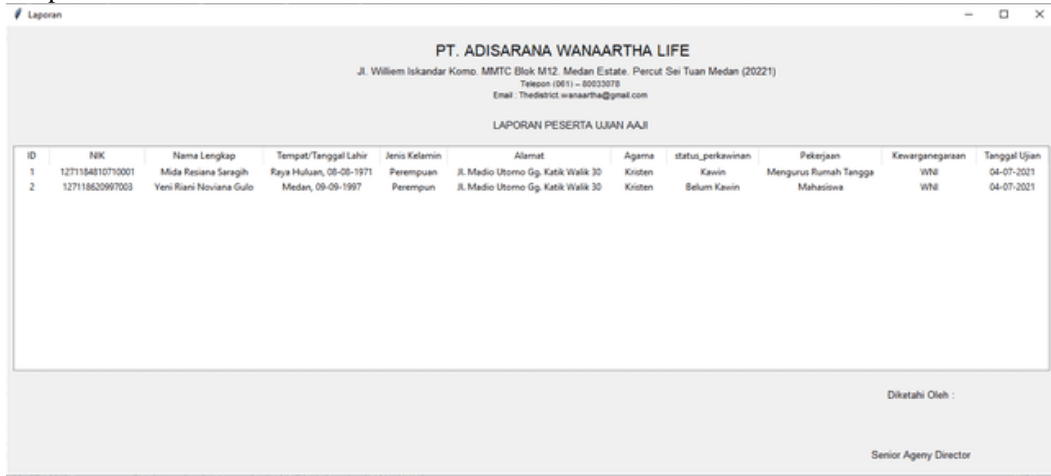
Gambar 5.7 Memasukkan Kata Kunci



Gambar 5.10 Proses Hapus Berhasil

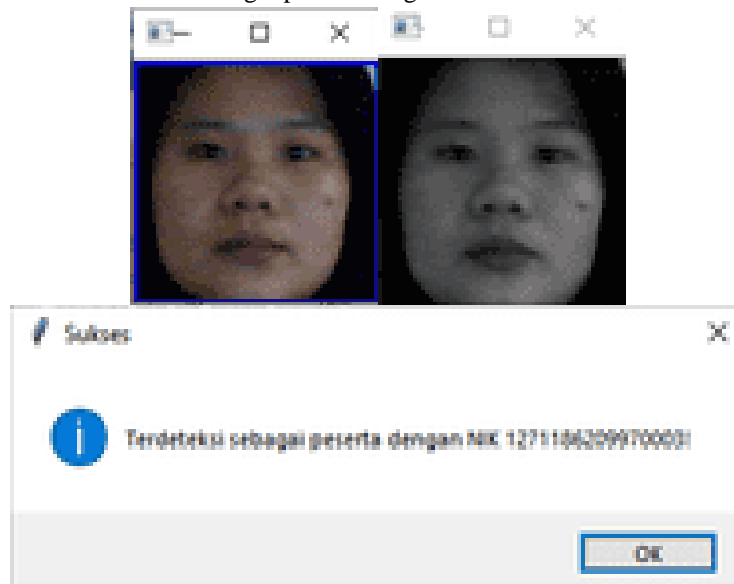
Proses lainnya ialah proses *training*, admin menekan tombol *training* selanjutnya sistem akan melakukan pelatihan terhadap citra wajah yang sudah disimpan kemudian citra wajah akan di simpan sebagai citra wajah *training* pada *database* peserta.

Proses berikutnya ialah proses laporan dimana laporan akan ditampilkan pada saat admin menekan tombol laporan.



Gambar 5.11 Laporan Berhasil Di Tampilkan

Proses deteksi wajah akan dilakukan pada *form* pengawas, pada proses ini akan menampilkan proses deteksi wajah yang tersimpan pada *database*, jika wajah yang di-input oleh *webcam* terdeteksi maka akan menampilkan pesan berisi Terdeteksi sebagai peserta dengan NIK !.



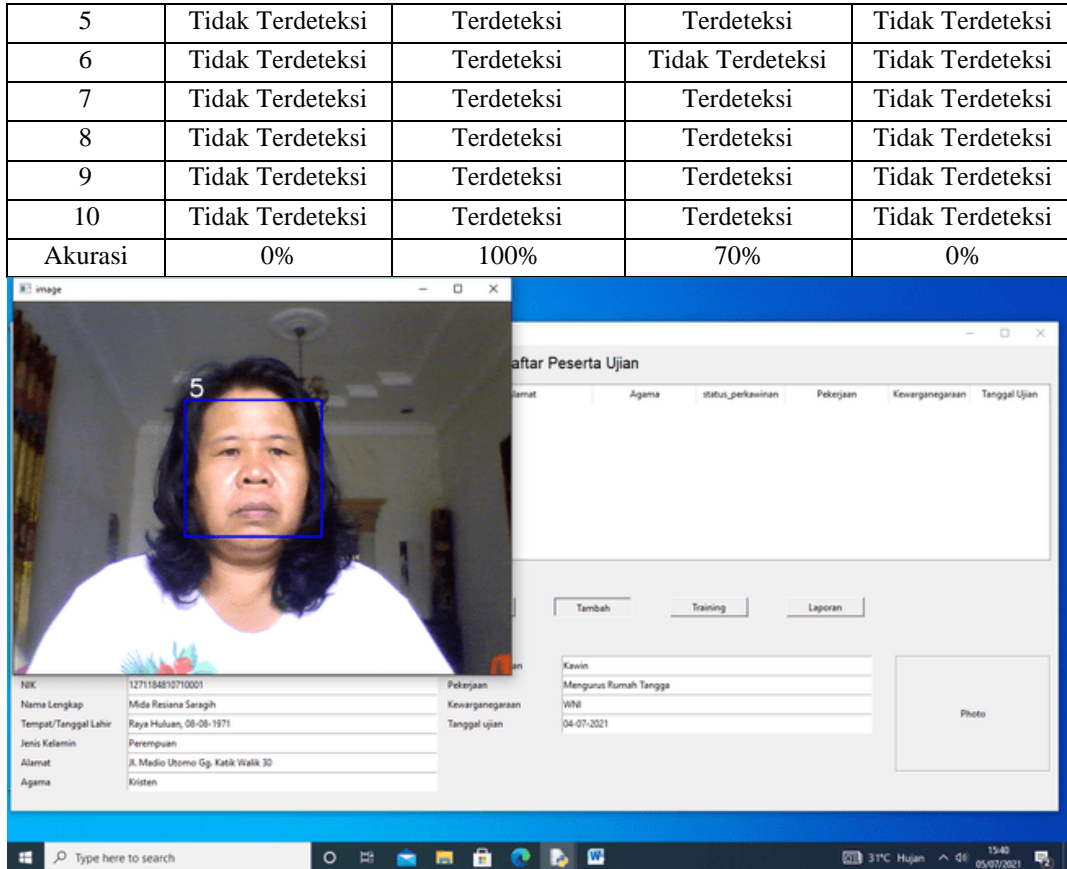
Gambar 5.12 Deteksi Wajah

3.4.2 Pengujian Deteksi Wajah

Pada pengujian ini program yang telah dibuat akan dilakukan pengujian pada kondisi cahaya pada objek citra wajah yang berbeda - beda. Pengujian ini bertujuan untuk menguji sejauh mana bentuk objek wajah dapat dideteksi.

Tabel 5.1 Pengujian Akurasi Deteksi Terhadap Cahaya.

| Uji ke- | Intesitas Cahaya pada citra wajah (Lux) | | | |
|---------|---|------------|------------------|------------------|
| | > 10 | 10 - 1000 | 1000 - 1500 | <1500 |
| 1 | Tidak Terdeteksi | Terdeteksi | Terdeteksi | Tidak Terdeteksi |
| 2 | Tidak Terdeteksi | Terdeteksi | Terdeteksi | Tidak Terdeteksi |
| 3 | Tidak Terdeteksi | Terdeteksi | Tidak Terdeteksi | Tidak Terdeteksi |
| 4 | Tidak Terdeteksi | Terdeteksi | Tidak Terdeteksi | Tidak Terdeteksi |



Gambar 5.13 Contoh Deteksi Dalam Kondisi Cahaya 800 Lux.

Dari hasil percobaan diatas dapat dilihat bahwa faktor pencahayaan sangat mempengaruhi deteksi objek wajah, dan didapatkan tingkat intensitas cahaya yang paling ideal adalah pada 10 – 1000 Lux, sehingga dalam hal ini program yang dijalankan harus berada pada kondisi pencahayaan 500 – 1000 Lux.

Pengujian yang dilakukan selanjutnya adalah pengujian akurasi deteksi terhadap jarak. Pengujian jarak ini dengan kondisi cahaya normal. Tujuan pengujian ini adalah mengetahui seberapa jauh program yang telah dibuat dapat mendeteksi wajah dengan variasi jarak yang diujicobakan.

Tabel 5.2 Pengujian Akurasi Deteksi Terhadap Jarak.

| Uji ke- | Jarak obyek (cm) | | | |
|---------|------------------|------------|------------------|------------------|
| | >30 | 30 - 200 | 200 - 270 | >270 |
| 1 | Tidak Terdeteksi | Terdeteksi | Terdeteksi | Tidak Terdeteksi |
| 2 | Tidak Terdeteksi | Terdeteksi | Terdeteksi | Tidak Terdeteksi |
| 3 | Tidak Terdeteksi | Terdeteksi | Tidak Terdeteksi | Tidak Terdeteksi |
| 4 | Tidak Terdeteksi | Terdeteksi | Tidak Terdeteksi | Tidak Terdeteksi |
| 5 | Tidak Terdeteksi | Terdeteksi | Terdeteksi | Tidak Terdeteksi |
| 6 | Tidak Terdeteksi | Terdeteksi | Tidak Terdeteksi | Tidak Terdeteksi |
| 7 | Tidak Terdeteksi | Terdeteksi | Terdeteksi | Tidak Terdeteksi |
| 8 | Tidak Terdeteksi | Terdeteksi | Terdeteksi | Tidak Terdeteksi |
| 9 | Tidak Terdeteksi | Terdeteksi | Terdeteksi | Tidak Terdeteksi |
| 10 | Tidak Terdeteksi | Terdeteksi | Terdeteksi | Tidak Terdeteksi |
| Akurasi | 0% | 100% | 70% | 0% |

Dari hasil percobaan diatas dapat dikatakan bahwa faktor jarak antara alat (*webcam*) dengan objek wajah mempengaruhi dalam pendeteksian. Hal ini dikarenakan *image data training* yang disimpan pada kondisi dan jarak tertentu. Untuk jarak efektif pendeteksian dilakukan pada jarak 30 – 270cm. Kecepatan

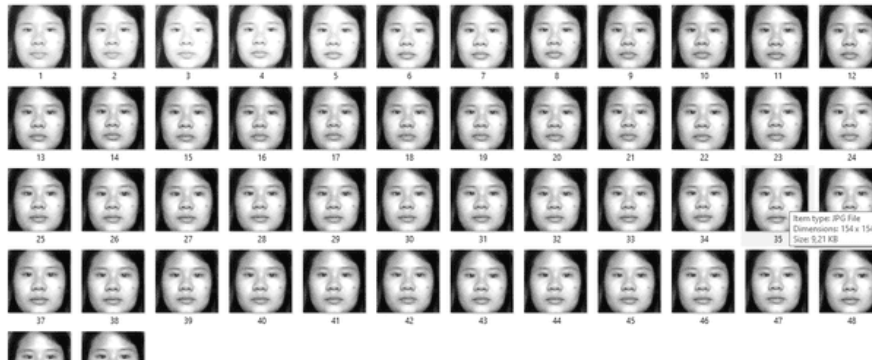
user saat pergerakan penangkapan objek wajah sangat berpengaruh terhadap baik buruknya deteksi sehingga perlu dilakukan terlebih dahulu pengujian agar menghasilkan kesimpulan kecepatan dalam pendeteksian objek wajah.

Tabel 5.3 Pengujian Terhadap Kecepatan Gerakan.

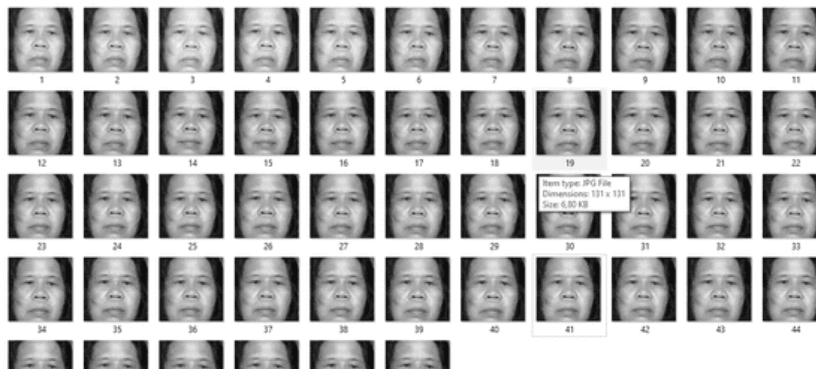
| Uji ke- | Diam | Sedang | Cepat |
|---------|------------|------------|------------------|
| 1 | Terdeteksi | Terdeteksi | Terdeteksi |
| 2 | Terdeteksi | Terdeteksi | Terdeteksi |
| 3 | Terdeteksi | Terdeteksi | Tidak Terdeteksi |
| 4 | Terdeteksi | Terdeteksi | Tidak Terdeteksi |
| 5 | Terdeteksi | Terdeteksi | Terdeteksi |
| 6 | Terdeteksi | Terdeteksi | Tidak Terdeteksi |
| 7 | Terdeteksi | Terdeteksi | Terdeteksi |
| 8 | Terdeteksi | Terdeteksi | Terdeteksi |
| 9 | Terdeteksi | Terdeteksi | Terdeteksi |
| 10 | Terdeteksi | Terdeteksi | Terdeteksi |
| Akurasi | 100% | 100% | 70% |

Dari tabel 5.3 Terlihat bahwa semakin cepat *user* mengerakan wajah, maka tingkat keberhasilan deteksi semakin kecil.

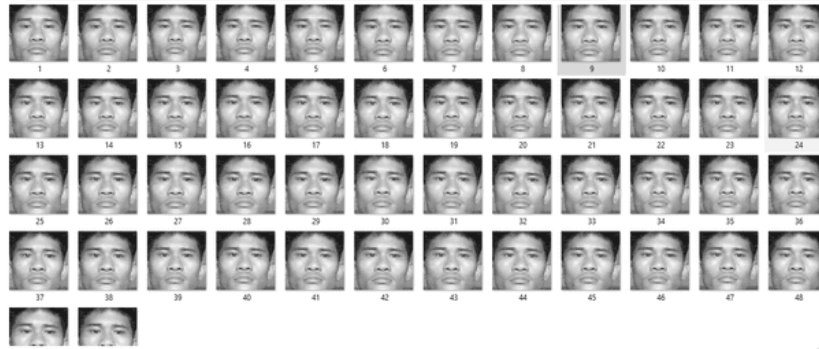
Juga dilakukan pengujian pengenalan deteksi wajah. Pengujian proses pengenalan wajah akan dilakukan dengan membandingkan beberapa gambar yang sudah tersimpan didalam *database*. Pada pengujian ini terdiri dari 3 wajah yang terdiri dari 50 sampel pada masing-masing wajah.



Gambar 5.14 50 Sample Wajah Yeni



Gambar 5.15 50 Sampel Wajah Mida



Gambar 5.16 50 Sampel Wajah Agus.

Pengenalan wajah dilakukan sebanyak 10 kali pada masing – masing wajah untuk mengetahui berapa % tingkat keakuratan pengenalan wajah

Tabel 5.4 Pengujian Pengenalan Deteksi Wajah Yeni

| Nama Sample Uji Wajah | Nama Sample Pengenalan Wajah |
|-----------------------|------------------------------|
| Yeni | Yeni |
| Yeni | Yeni |
| Yeni | Mida |
| Yeni | Yeni |
| Yeni | Yeni |
| Yeni | Yeni |
| Yeni | Mida |
| Yeni | Yeni |
| Yeni | Yeni |
| Yeni | Yeni |

Keakuratan = (Jumlah Hasil Benar / Jumlah Pengujian) x 100%

Keakuratan = (8/10) x 100% = 80%

Tabel 5.5 Pengujian Pengenalan Deteksi Wajah Mida

| Nama Sample Uji Wajah | Nama Sample Pengenalan Wajah |
|-----------------------|------------------------------|
| Mida | Mida |
| Mida | Mida |
| Mida | Mida |
| Mida | Mida |
| Mida | Yeni |
| Mida | Yeni |
| Mida | Mida |
| Mida | Mida |
| Mida | Mida |
| Mida | Mida |

Keakuratan = (Jumlah Hasil Benar / Jumlah Pengujian) x 100%

Keakuratan = $(8/10) \times 100\% = 80\%$

Tabel 5.6 Pengujian Pengenalan Deteksi Wajah Agus

| Nama Sample Uji Wajah | Nama Sample Pengenalan Wajah |
|-----------------------|------------------------------|
| Agus | Agus |
| Agus | Agus |
| Agus | Agus |
| Agus | Agus |
| Agus | Yeni |
| Agus | Agus |
| Agus | Agus |
| Agus | Agus |
| Agus | Agus |
| Agus | Agus |

Keakuratan = $(\text{Jumlah Hasil Benar} / \text{Jumlah Pengujian}) \times 100\%$

Keakuratan = $(9/10) \times 100\% = 90\%$

Pengujian pengenalan wajah ini dilakukan pada kondisi cahaya 500 - 1000 Lux, jarak 100 cm dan dengan kecepatan diam dan sedang. Dari tabel pengujian didapatkan hasil 80%, 80% dan 90% sehingga rata-rata keakuratan pendeteksian sample uji mencapai 83%.

4. KESIMPULAN

Dari hasil perancangan sampai dengan pengujian terhadap sistem aplikasi *scan* wajah yang dirancang dan dibangun maka dapat ditarik kesimpulan sebagai berikut:

1. Dengan menganalisa masalah tentang alat bantu *scan* wajah untuk membantu pengawas ujian PT. Adisrana Wanaartha Life Cabang The District Medan melakukan pendekatan riset dan wawancara untuk mengetahui apa kendala yang di hadapi saat ujian AAJI.
2. Telah berhasil diterapkan algoritma *eigenface* untuk mendeteksi wajah dengan bantuan *opencv*, sehingga dapat diimplementasikan ke dalam system dengan pencahayaan yang ideal yaitu 100-1500 Lux dan dengan jarak 100 cm serta dilakukan dalam keadaan diam dan menggunakan 50 sample wajah agar dapat mengenali wajah dengan baik.

UCAPAN TERIMA KASIH

Terima Kasih diucapkan kepada kedua orang tua serta keluarga yang selalu memberi motivasi, Doa dan dukungan moral maupun materi, serta pihak-pihak yang telah mendukung dalam proses pembuatan jurnal ini yang tidak dapat disebutkan satu persatu. Kiranya jurnal ini bisa memberi manfaat bagi pembaca dan dapat meningkatkan kualitas jurnal selanjutnya.

REFERENSI

- [1] W. Dwiparawati, S. Kom, and S. V. Hilmawan, "Implementasi Face Recognition secara Real-time dengan Metode Haar Cascade Classifier menggunakan OpenCV-Python," 2017.
- [2] E. Yanuarti and R. B. Isnanto, "Jurnal sisfokom," SISFOKOM, vol. 05, no. 0717, pp. 1–65, 2016.
- [3] I. N. Dedi Ary Prasetya, "Deteksi wajah metode viola jones pada opencv menggunakan pemrograman python," Simp. Nas. RAPI XI FT UMS, pp. 18–23, 2012.
- [4] H. Kurniawan *et al.*, "APLIKASI PENJAWAB PESAN SINGKAT AUTOMATIS DENGAN BAHASA PYTHON," 2011.
- [5] H. Muchtar and R. Apriadi, "Implementasi Pengenalan Wajah Pada Sistem Penguncian Rumah Dengan Metode Template Matching Menggunakan Open Source Computer Vision Library (Opencv)," *Resist. (elektRONika kEndali Telekomun. tenaga List. kOmpuTeR)*, vol. 2, no. 1, p. 39, 2019, doi: 10.24853/resistor.2.1.39-42.

Title of manuscript is short and clear, implies research results (First Author)

BIBLIOGRAFI PENULIS

| | |
|---|--|
|  | <p>Nama Lengkap : Yeni Riani Novina Gulo</p> <p>NIRM : 2017020927</p> <p>Tempat/Tgl.Lahir : Medan, 22 September 1997</p> <p>Jenis Kelamin : Perempuan</p> <p>Agama : Kristen</p> <p>Alamat : Jl. Bunga Cempaka Gg. Siompul No. 2 Ujung</p> <p>No/Hp : 0812 6524 4137</p> <p>Email : yeninovianag@gmail.com</p> <p>Program Keahlian : Pemmograman Berbasis Desktop</p> |
|  | <p>Nama Lengkap : Puji Sari Ramadhan, S.Kom., M.Kom</p> <p>NIDN : 0126039201</p> <p>Program Studi : Sistem Informasi</p> <p>Deskripsi : Dosen Tetap STMIK Triguna Dharma yang aktif mengajar dan fokus pada bidang keilmuan kecerdasan buatan dan data sains. Telah menulis 1 buku dibidang Ilmu komputer. Memiliki sebanyak 2 Hak Kekayaan Intelektual (HKI). Menjabat sebagai Ketua Program studi Sistem Informasi.</p> <p>Prestasi : Dosen Terbaik Tahun 2018, Pemenang PDP 2018 dan 2019.</p> |
|  | <p>Nama Lengkap : Dr. Asyahri Hadi Nasyuha, S.Kom., M.Kom</p> <p>NIDN : 0129048601</p> <p>Program Studi : Sistem Informasi</p> <p>Deskripsi : Dosen Tetap STMIK Triguna Dharma yang aktif mengajar dan fokus pada bidang keilmuan Komputer serta aktif dalam organisasi Cyber Programing Club. Telah menulis 3 (Tiga) buku dibidang Ilmu komputer. Memiliki sebanyak 2 (Dua) Hak Kekayaan Intelektual (HKI). Menjabat sebagai Ketua Lembaga Penjaminan Mutu.</p> <p>Prestasi : Finalis Lomba Aplikasi Mobile Kihajar 2018 BPMPK Kemendikbud Kategori Umum V-Lab, Hibah PDP 2020, Lulusan Terbaik S3 Fakultas Teknik Program Studi Pendidikan Teknologi Kejuruan, Universitas Negeri Padang.</p> |