Implementasi Program Presensi Mahasiswa dengan menggunakan Face Recognition

Richard Lawrence Thiosdor, Kartika Gunadi, Lily Puspa Dewi
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp.(031) - 2983455, Fax. (031) - 8417658
richardl.thiosdor@gmail.com, kgunadi@petra.ac.id, lily@petra.ac.id

ABSTRAK

Permasalahan penggunaan daftar absen fisik menyebabkan adanya ketidakvalid-an dimana mahasiswa tersebut melakukan "Penitipan Absen" yang dimana meminta mahasiswa lain untuk mewakili paraf absen. Permasalahan tersebut banyak ditemukan pada kegiatan kuliah.

Deteksi wajah mahasiswa dengan menggunakan library *Face Recognition* sebagai cara untuk validasi dalam proses pengambilan presensi. *Face Recognition* membutuhkan data wajah yang telah di *preprocess* dan menggunakan model *K-Nearest Neighbor (KNN)* atau *Support Vector Machine (SVM)* untuk memvalidasi wajah mahasiswa pada proses presensi.

Pengujian pada 15 sample gambar wajah dengan 40 jumlah *class* wajah memperolah akurasi rata-rata 99%. *Face Recognition* tidak dapat mendeteksi jika fitur wajah tertutup. Validasi Presensi mahasiswa berhasil dengan menggunakan *Face Recognition* untuk meminimalisir kecurangan dalam pengambilan presensi.

Kata Kunci: *Face Recognition*, Presensi, *KNN, SVM.*

ABSTRACT

The problem of using a physical attendance list causes a cheat where the student does "fake attendance" by asking another students to sign the attendance list on his/her behalf. This problems are often found in college activities.

Detection of student faces uses the Face Recognition library as a mean of validation in the attendance check process. Face recognition requires face images that have been preprocessed and uses the K-Nearest Neighbor model (KNN) or Support Vector Machine (SVM) to validate student faces in the attendance check process.

Testing on 15 sample face images with 40 total face classes yields an average accuracy of 99%. Face Recognition cannot detect faces if the facial features are obstructed. This validation of student attendance successfully uses Face Recognition to minimize cheating in taking attendance.

Keywords: *Face Recognition, Attendance Check, KNN, SVM*

1. PENDAHULUAN

Penelitian ini mengenai implementasi presensi mahasiswa dengan menggunakan *face recognition*. Metode yang digunakan untuk mendeteksi wajah mahasiswa dibagi 2, yaitu *K-Nearest Neighbor* dan *Support Vector machine*.

Presensi merupakan salah satu hal penting untuk mahasiswa belajar di universitas, dimana presensi digunakan sebagai bukti kehadiran mahasiswa dari mata kuliah yang diambil. Cara pengambilan absen pun ada banyak, tapi sebagian besar masih menggunakan daftar absen yang di cetak di kertas dan mahasiswa melakukan paraf untuk menandakan bahwa mahasiswa tersebut hadir dan mengikuti kelas yang diambil nya. Namun penggunaan daftar absen menyebabkan adanya ketidakvalid-an dimana mahasiswa tersebut melakukan "Penitipan Absen" dimana meminta teman yang dikenal untuk mewakili paraf absen dan tidak hadir dalam kegiatan perkuliahan serta merugikan mahasiswa tersebut.

Untuk mengatasi permasalahan tersebut pada skripsi ini akan dilakukan suatu perancangan sistem presensi dan pembuatan aplikasi yang dapat melakukan presensi mahasiswa dengan menggunakan sistem pengenalan wajah (*Facial Recognition System*) untuk mengganti paraf mahasiswa sehingga tidak ada nya proses ketidakvalid-an dalam pengambilan absen.

Sistem pengenalan wajah (*Facial Recognition System*) adalah teknologi yang dapat mengenali wajah seseorang didalam sebuah gambar *digital* atau *frame* video. Metode yang digunakan dalam sistem pengenalan wajah ada banyak, tapi secara garis besar bekerja dengan menyamakan fitur wajah dengan yang ada didalam *database*. Kelebihan dari penggunaan *face recognition* adalah tidak perlunya kontak dari bagian tubuh untuk melakukan absen, harganya tidak mahal, hasil nya dapat dipercaya, multifungsi, dan susah untuk dilakukan pemalsuan [4] yang dapat dilihat pada Tabel 1.

Tabel 1. Perbandingan antara *Face Recognition* dengan *Fingerprint*

Technology/characteristic	Face recognition	Fingerprint
Ergonomic aspect	Non-invasive	Invasive
Cost	Medium cost and Multi-purpose	Medium costa Single purpose
Reliability	Very high	High
Social acceptability	High	Very High
Accuracy	High	Very high
Spoofing (Creating a fake biometric)	Very difficult	Very easy
Behavior analysis	Possible (dynamic)	Impossible

2. LANDASAN TEORI

2.1 Presensi

Presensi adalah sebuah kegiatan pengambilan data yang digunakan untuk mencatat jumlah kehadiran dalam suatu kegiatan. Presensi tersebut kemudian dapat diproses untuk memberikan informasi untuk mengukur kesuksesan suatu kegiatan atau mencari informasi dari yang telah mengikuti kegiatan

tersebut. untuk penelitian ini Presensi ditujukan untuk mahasiswa dimana presensi tersebut digunakan untuk mencatat kehadiran mahasiswa yang datang maupun tidak datang.

Fungsi lain dari presensi ini adalah untuk mengetahui bagaimana efektif kelas yang sedang diikuti oleh mahasiswa jika banyak mahasiswa yang tidak hadir maka dapat disimpulkan bahwa kegitan mengajar tersebut tidak berhasil atau sebaliknya.

Pengambilan data presensi secara manual memiliki banyak kekurangan seperti data yang dimasukkan salah , data bisa hilang dan data bisa rusak. Kekurangan lainnya adalah pengambilan presensi yang lama sehingga mengurangi efisiensi dan efektivitas dalam pengambilan data.

2.2 Face Recognition

Face recognition adalah sebuah perangkat lunak yang bisa mengidentifikasi seseorang dengan menganalisa sebuah pola yang ada di kontur wajah orang tersebut. *Face Recognition* kebanyakan digunakan untuk bidang keamanan. Tapi setelah berkembangnya teknologi perangkat lunak ini sudah mulai digunakan untuk area lain selain keamanan seperti ATM, sistem kesehatan dan sistem kereta.

Ada banyak kelebihan menggunakan *Face Recognition* Berbeda dengan metode pengenalan lain, *Face recognition* tidak mengharuskan adanya kontak dari bagian tubuh. Wajah dapat ditangkap dari jauh dan dianalisa tanpa adanya interaksi dengan orang [5].

2.3 K-Nearest Neighbor (KNN)

Metode *K-Nearest Neighbor (KNN)* adalah sebuah metode yang populer digunakan untuk melakukan klasifikasi terhadap objek berdasarkan data yang jaraknya paling dekat dengan objek tersebut, Metode ini merupakan metode yang paling umum digunakan untuk estimasi dan prediksi karena mudah untuk di implementasi [6]

Keunggulan dari metode KNN ini adalah *relative* tidak terpengaruh dari *error* dari data dan juga dapat digunakan dengan kumpulan data dengan jumlah besar. Namun kekurangan metode ini adalah proses pelaksanan yang lambat jika volume data yang besar sehingga tidak praktis jika digunakan untuk melakukan prediksi secara cepat.

Tapi KNN masih berguna dalam memecahkan masalah yang memiliki solusi yang bergantung pada identifikasi objek yang hampir sama dengan satu yang lain [2].

2.4 *Support Vector Machine (SVM)*

Support Vector machine (SVM) merupakan salah satu metode dari *Machine Learning* yang sering digunakan untuk memprediksi suatu data dalam klasifikasi maupun regresi. Metode ini dikembangkan oleh Vladimir Vapnik dan merupakan metode yang melakukan pendekatan *supervised learning* [1].

Support Vector Machine adalah algoritma *Supervised Machine Learning* adalah algoritma yang sering dipakai untuk klasifikasi dibandingkan dengan *unsupervised learning* dikarenakan memiliki data training yang digunakan agar dapat mengklasifikasikan atau memprediksi data baru ke dalam suatu kelas atau label. Algoritma SVM berhasil digunakan untuk masalah klasifikasi yang berbeda [3].

Cara kerja *SVM* adalah mencari garis pemisah terbaik untuk memisahkan kedua *Class* yang berbeda. Garis pemisah ini disebut dengan *Hyperplane* yang berguna sebagai pemisah ruang *vector* menjadi dua *class* berbeda, untuk mendapatkan hasil optimal maka dicari jarak maksimum antara margin dari *data points* dari dua class.

Margin yang maksimal dapat memperkuat *future data points* yang bisa di klasifikasi dengan nilai *confidence* yang tinggi. SVM dapat menggunakan algoritma *kernel trick* dimana dapat berfungi untuk memudahkan komputasi dan efisien untuk transformasi data ke dimensi yang lebih tinggi.

3. DESAIN SISTEM

Bab ini akan membahas lebih lanjut mengenai desain sistem yang digunakan dalam pembuatan *Aplikasi Presensi* dengan menggunakan *Face Recognition*, arsitektur sistem, proses training, implementasi *Face Recognition*, pembuatan aplikasi presensi.

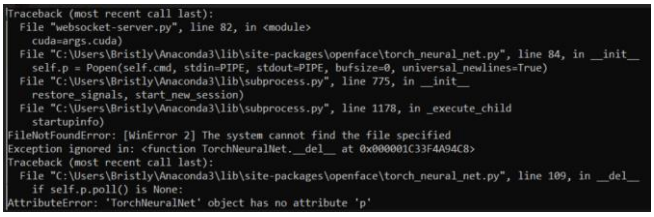
3.1 Analisa Sistem

Pada bagian ini akan menjelaskan mengenai Analisa yang dapat timbul dalam membuat penelitian ini. Pada bagian ini akan menjelaskan masalah tersebut dan bagaimana cara penulis dapat menyelesaikan masalah tersebut.

Untuk dapat mendeteksi wajah pada gambar digunakanlah library yang bernama Dlib. Digunakan Dlib *Facial landmark* untuk mendeteksi *keypoint* yang ada dalam gambar agar dapat mendeteksi yang mana yang dapat dibilang sebuah wajah, contohnya ujung mulut, ujung mata bagian luar atau bagian dalam, garis mulut.

Untuk membantu dan mempercepat proses deteksi wajah sebelum itu harus melakukan *preprocessing*. Dalam *preprocessing* ini akan dilakukan *resize* dan *cropping* dalam bentuk kotak pada gambar agar tidak memberatkan kapasitas memori computer.

Pada Gambar 1 dapat dilihat adanya *error* pada percobaan untuk implementasi *OpenFace Face Recognition* pada Operating sistem Windows jadi solusinya melakukan penggantian *library* yang bernama *face_recognition*.



Gambar 1. *Error* saat Penggunaan *OpenFace Face Recognition*

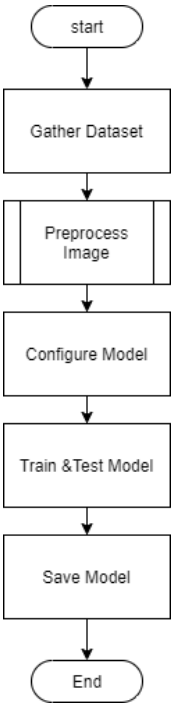
3.2 Desain Sistem aplikasi

Pada bagian ini akan menjelaskan tentang desain sistem program presensi wajah. Desain sistem dijelaskan dengan bantuan *Flowchart* yang terdiri dari desain sistem keseluruhan dan desain alur program. Desain sistem secara keseluruhan akan membahas alur program sedangkan desain alur program akan menjelaskan setiap bagian pada program.

3.2.1 Desain Training Classifier

Pada Gambar 2 merupakan *flowchart* dari proses *training classifier* wajah untuk model *KNN* dan *SVM*. Pertama dilakukan pengumpulan *dataset*, *dataset* yang digunakan adalah *sample dari LFW dataset*, *sample* termasuk 40 jenis wajah. Sample tersebut akan di *load* dan di *preprocess* untuk mengurangi *noise* yang dapat menurunkan akurasi dalam proses *train* dan *test*. Setelah *preprocess* selesai model akan dibuat dan diatur berdasarkan konfigurasi yang diinginkan, pada penelitian ini akan digunakan

konfigurasi parameter untuk *KNN* adalah *n_neighbor* dan *weights* dan model *SVM* akan digunakan *parameter kernel*. Setelah konfigurasi akan dijalankan ke proses *train test* untuk melihat akurasi yang didapatkan dari model dan model tersebut akan disimpan untuk digunakan pada pengenalan wajah mahasiswa pada aplikasi presensi.



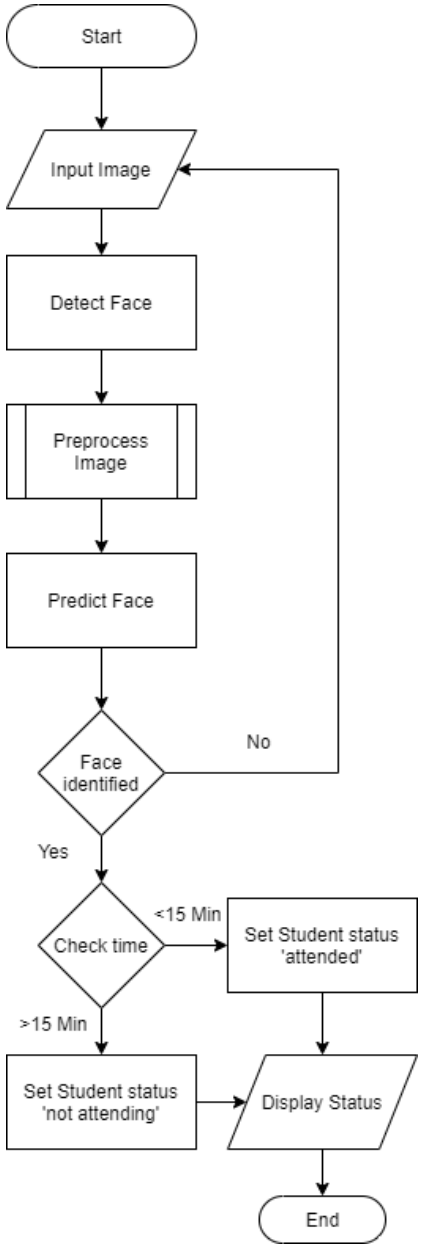
Gambar 2. Gambaran alur kerja *Classifier*

3.2.2 Desain Kerja Sistem Back end

Program *Server* bekerja dengan menerima input gambar yang diupload dari *front end* yang berupa hasil tangkapan foto wajah dari kamera. Gambar yang telah diinput akan kemudian dideteksi wajah dalam gambar untuk melakukan *preprocessing*. Pada proses *preprocessing* gambar akan di *crop* dengan ukuran (224x224) dengan titik tengah yang berpusat dengan wajah, hal ini dilakukan untuk mengurangi adanya *noise* dalam gambar yang dapat menurunkan akurasi dalam memprediksi wajah yang di input.

Setelah gambar di *preprocess*, *server* kemudian memprediksi wajah dalam gambar tersebut dan jika *output* dari prediksi benar maka alur selanjutnya akan mengecek waktu pengambilan gambar jika cocok dengan batas waktu yang ditentukan maka mahasiswa tersebut akan dinyatakan hadir dalam kegiatan perkuliahan.

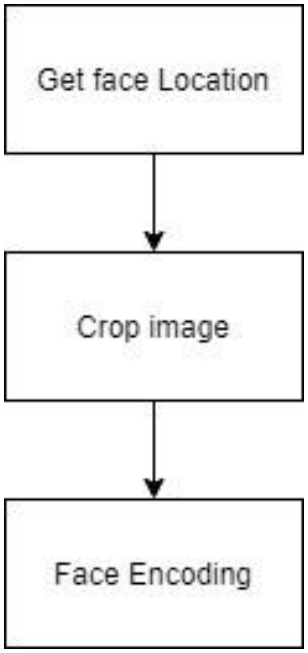
Jika prediksi wajah gagal maka akan dialihkan untuk melakukan pengambilan presensi lagi. Sistem tidak membedakan mahasiswa *valid* gagal dikenali dengan *impostor* saat melakukan presensi. Sistem hanya memprediksi akurasi wajah mahasiswa untuk mencapai *threshold* yang telah ditentukan agar dianggap valid. Alur kerja sistem dapat dilihat pada Gambar 3.



Gambar 3. Gambaran Sistem *Back end*

3.2.3 Flowchart Sub-Proses Preprocess Image

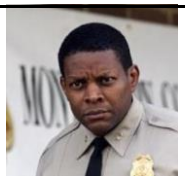
Pada Gambar 4 Awal alur proses akan dijalankan *function* untuk mendeteksi lokasi wajah dalam gambar, lokasi wajah akan didapatkan jika gambar tersebut memiliki wajah yang dimana fitur wajahnya (mata, mulut, alis,hidung) tampak jelas dan tidak tertutupi dengan objek lain atau wajah orang lain. Kemudian jika lokasi wajah ditemukan maka gambar tersebut akan dipotong untuk meningkatkan mengurangi *noise* dalam gambar. Setelah itu, gambar tersebut akan di *face encode*. *Face_encode* adalah proses mengkonversi gambar menjadi sebuah *128-dimensional vector space*.



Gambar 4. Flowchart *preprocess image*

Berikut adalah contoh gambar yang digunakan untuk proses *training* dan *validation*, *dataset* yang digunakan *adalah Labeled Faces in the Wild (LFW).*

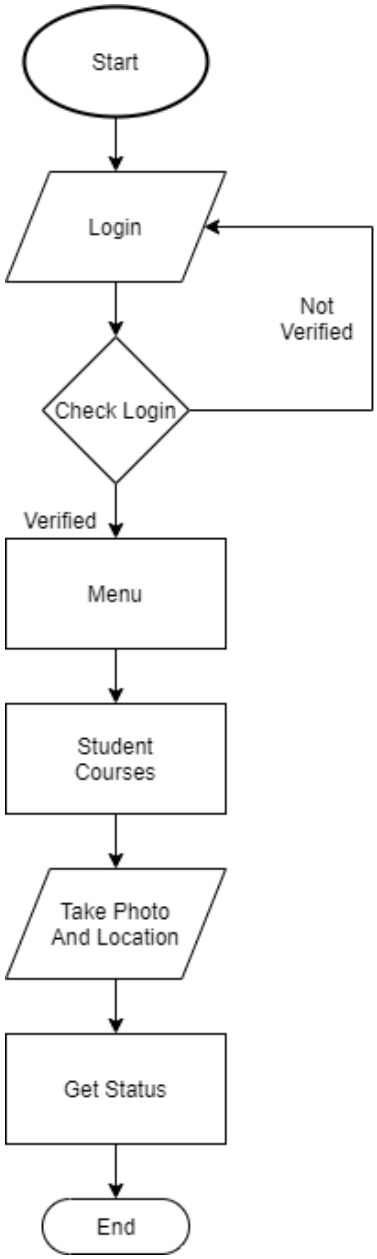
Tabel 2. Contoh *Sample* dan *Validation*

Sample	***Validation***
	

Pada Tabel 2 adalah Contoh *sample* wajah yang digunakan dalam *dataset* LFW. Pada Proses *Training* Model, Setiap *Class* (orang) terdiri dari 10 atau lebih foto sample wajah lalu *sample* foto tersebut akan di split untuk validasi hasil *training* dari model. Untuk t*raining* awal digunakan *dataset LFW* dikarenakan kurang nya *dataset* mahasiswa yang ingin dicoba pada proses *training*.

3.2.4 Desain Alur Aplikasi presensi

Pada Gambar 5 digambarkan alur presensi dalam aplikasi dimana aplikasi dimulai dengan page *login* dimana mahasiswa akan memasukkan NRP dan password untuk verifikasi masuk ke aplikasi. Lalu akan dibawa ke page *menu* dan mahasiswa akan memilih *menu* matakuliah untuk melakukan absensi setelah itu mahasiswa akan diminta untuk mengambil foto dari wajah untuk proses *face recognition*. Aplikasi akan mengirim foto dan lokasi mahasiswa untuk melakukan validasi dari presensi di *server*. Setelah selesai divalidasi mahasiswa akan mendapatkan *feedback* berupa status jika berhasil mahasiswa akan dialihkan kembali ke menu awal dan jika tidak mahasiswa akan diminta untuk melakukan proses pengambilan presensi.



Gambar 5. Gambaran Alur Aplikasi Presensi

3.3 Desain *User Interface*

Berikut adalah desain *user interface* untuk aplikasi presensi. Desain berikut merupakan gambaran kasar dari aplikasi presensi yang menggambarkan secara garis besar dan digunakan sebagai panduan untuk membuat aplikasi presensi yang sebenarnya.

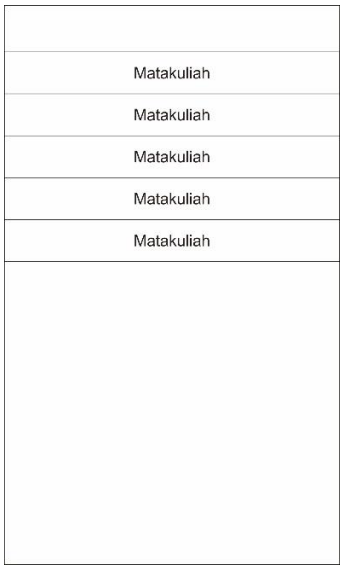
3.3.1 Desain Halaman Login

Berikut tampilan halaman *login* dari aplikasi presensi mahasiswa, tampilan terdiri dari input NRP, *Password* dan tombol *Login*. Jika Tombol *Login* ditekan maka aplikasi akan mengecek dengan data yang ada yang ada di *database* kemudian jika berhasil user akan masuk ke halaman matakuliah desain halaman dapat dilihat pada Gambar 6.

Gambar 6. Tampilan Desain Halaman *Login*

3.3.2 Desain Halaman Matakuliah

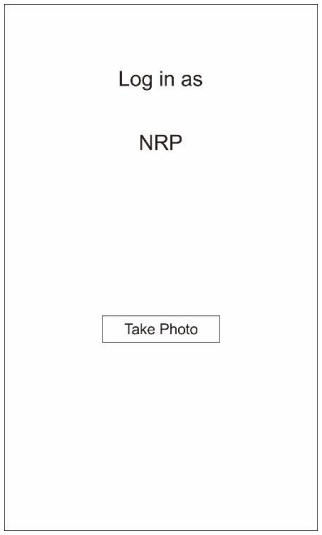
Pada Gambar 7 tersedia daftar dari matakuliah yang mahasiswa telah ikuti, kemudian mahasiswa dapat menekan daftar tersebut agar memilih kelas untuk melakukan proses pengambilan presensi.



Gambar 7. Tampilan Desain Halaman Matakuliah

3.3.3 Desain Halaman Presensi

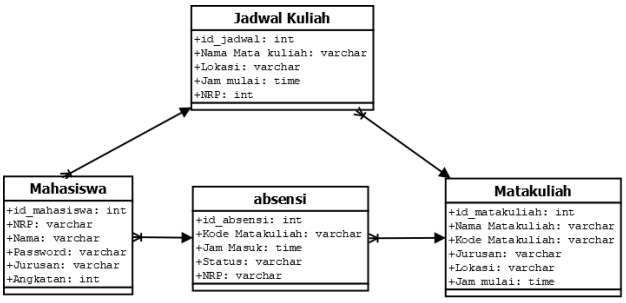
Pada Gambar 8 Mahasiswa melakukan proses presensi dengan tampilan yang sederhana tombol *take photo* akan mengarahkan mahasiswa untuk membuka kamera dan mahasiswa dapat mengambil foto, foto tersebut dikirim ke *server* dan di validasi.



Gambar 8. Tampilan Desain Halaman Presensi

3.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu model yang digunakan untuk Menyusun database agar dapat menggambarkan data tersebut yang memiliki relasi dengan *database* yang akan dibuat. Bentuk *database ERD* dapat dilhat pada Gambar 9.



Gambar 9. Gambaran Desain *Database*

4. IMPLEMENTASI SISTEM

Implementasi system dilakukan pada komputer dengan spesifikasi:

- RAM: 16GB, DDR3
- Memory: 1TB HDD
- CPU: Intel Core i7 6700HQ
- GPU: NVIDIA GeForce GTX 960M
- OS Windows 10 Pro

Instalasi Library yang diperlukan :

- opencv-python
- face-recognition
- numpy
- scikit-learn
- Flask

5. PENGUJIAN SISTEM

5.1 Pengujian Preprocessing sample gambar

Tabel 3. Urutan *Preprocessing sample* gambar

Sample Gambar	*Face locations*	*Encode (a vector of 128 values)*
		

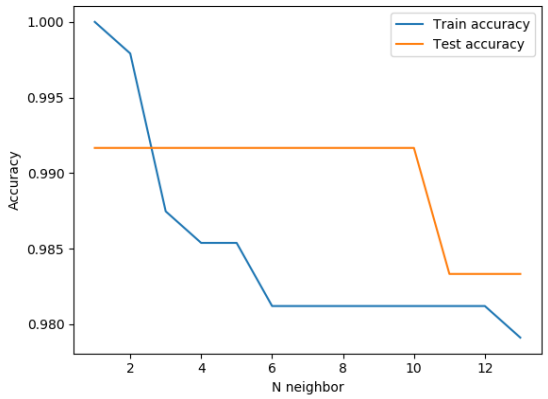
Pada Tabel 3 dapat dilihat tabel pengujian dimana pengujian teresbut dilakukan unutk mengetahui apakah aplikasi dapat mendeteksi fitur wajah dalam gambar yang telah diinput.

Berikut adalah contoh hasil dari *preprocessing* dari 40 class yang mesing masing class memiliki 10 lebih foto yang akan digunakan untuk proses training model, pada proses *preprocessing* foto tersebut akan dicari lokasi wajah cara penentuan lokasi wajah tersebut dengan mencari fitur wajah yang ada didalam foto. Algoritma program akan mencari dan menganalisa posisi *relative*, bentuk dan ukuran dari mata, mulut, hidung, tulang pipih, dan garis rahang.

5.2 Pengujian Menggunakan Model SVM & KNN

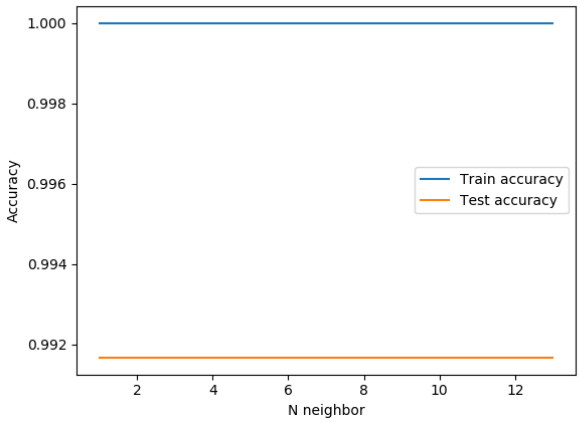
Pengujian dilakukan dengan jumlah class 41 dengan masing masing foto yang memiliki sekitar 15 foto dimana foto tersebut akan di *preprocess*, jika ada gambar tersebut tidak dideteksi ada wajah dalam gambar maka akan di abaikan, hal ini dilakukan untuk menambahkan akurasi dalam proses training.

5.2.1 Pengujian dengan menggunakan model KNN



Gambar 10. Grafik Pengujian Akurasi *KNN* Dengan *Parameter weights='uniform'*

Pada Gambar 10 Pengujian dilakukan dengan *parameter weights= 'uniform'* dan proses *training* dan *testing* mendapatkan *train accuracy* 0.997-0.987 pada *n-neighbor*= 2-3 dan *test accuracy* mencapai 0.991 pada semua *n-neighbor*.



Gambar 11. Grafik Pengujian Akurasi *KNN* Dengan *Parameter weights='distance'*

Pada Gambar 11 Pengujian dilakukan dengan *parameter weights= distance* dan proses *training* dan *testing* mendapatkan *train accuracy* 1.0 pada n-neighbor= 1-14 dan *test accuracy* mencapai 0.992 pada semua *n-neighbor*. Dilihat perbandingan antara *weights* dapat disimpulkan bahwa dengan menggunakan *parameter weights= 'distance'* hasil *train accuracy* mendapatkan hasil tertinggi yaitu 1 dan *test accuracy* mendapatkan nilai 0.992, sedangkan jika menggunakan *weights='uniform'* hasil *train accuracy* tertinggi sebesar 0.997 dan *test accuracy* tertinggi sebesar 0.991 dimana *n neighbor = 2*.

5.2.2 Pengujian menggunakan Model SVM

Tabel 4. Tabel Pengujian SVM dengan Parameter Kernel

Kernel	Training accuracy	Testing accuracy
Radial basis function	0.9791231732	0.9500000000
Polynomial	**0.9895615866**	**0.9666666667**
linear	0.9832985386	0.9666666667
Sigmoid	0.2421711899	0.1416666666

Pada Tabel 4 dapat disimpulkan bahwa *kernel polynomial* memiliki tingkat *accuracy* yang lebih tinggi dibandingkan dengan penggunaan *kernel* yang lain, pengujian ini membuktikan *kernel* ini lebih cocok digunakan untuk membangun model klasifikasi wajah mahasiswa.

5.3 Hasil Kuesioner

Untuk menjawab masalah untuk mencegah kecurangan 'penitipan absen' dilakukan pembagian kuesioner yang dibagi ke 10 mahasiswa untuk memberikan respon. Pertanyaan dapat dilihat pada Tabel 5.

Tabel 5. Tabel Pertanyaan Kuesioner

Pertanyaan	Penilaian		
	Ya	Tidak	lain
Apakah anda pernah melakukan 'titip absen'?	6	4	0
Apakah anda bisa melakukan 'titip absen' jika konsep absensi diganti menjadi konsep tersebut	6	3	1
Apakah dengan menggunakan presensi wajah kecurangan dapat dikurangi?	7	3	0

Tabel 5 merupakan hasil dari kuesioner yang terkumpulkan, 6 responden yang menjawab ya kalau konsep ini dapat mencegah kecurangan 'penitipan absen' dan 3 menjawab tidak. Dan lainnya jawabannya berupa cara 'penitipan absen' yang baru yaitu dengan membawa foto mahasiswa yang ingin diabsen.

6. KESIMPULAN

6.1 Kesimpulan

Berdasarkan hasil pengujian dapat disimpulkan beberapa hal sebagai berikut:

- Akurasi metode KNN memiliki akurasi yang jauh lebih tinggi dibandingkan dengan SVM
- Kecurangan 'titip absen' dari hasil kuesioner bahwa dengan menggunakan presensi wajah kecurangan tersebut dapat dikurangi.

6.2 Saran

Beberapa hal yang dijadikan saran dalam proses pengembangan selanjutnya antara lain:

- Dapat melakukan deteksi wajah banyak orang untuk mengurangi waktu yang diperlukan untuk mengambil presensi.
- Aplikasi Presensi dapat membedakan foto yang diambil langsung dengan foto yang ada di kertas atau *device* untuk mencegah kecurangan yang baru yang dapat dilakukan.

7. DAFTAR PUSTAKA

[1] Ahmad, I. et al. 2018. Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection. IEEE Access. 6, 33789–33795. DOI= https://doi.org/10.1109/access.2018.2841987

[2] Cunningham, P. & Delany, S. 2020. k-Nearest neighbour classifiers: 2nd Edition (with Python examples).

[3] Demidova, L., Klyueva, I., Sokolova, Y., Stepanov, N., & Tyart, N. 2017. Intellectual Approaches to Improvement of the Classification Decisions Quality on the Base of the SVM Classifier. Procedia Computer Science, 103, 222–230. DOI = https://doi.org/10.1016/j.procs.2017.01.070.

[4] Mohammed, K., Tolba, A. S., & Elmogy, M. 2018. Multimodal student attendance management system (MSAMS). Ain Shams Engineering Journal, 9(4), 2917–2929. DOI= https://doi.org/10.1016/j.asej.2018.08.002.

[5] Singh, S., & Prasad, S. V. A. V. 2018. Techniques and Challenges of Face Recognition: A Critical Review. Procedia Computer Science, 143, 536–543. DOI= https://doi.org/10.1016/j.procs.2018.10.427.

[6] Zhang, S. et al. 2018. Efficient kNN Classification With Different Numbers of Nearest Neighbors. IEEE Transactions on Neural Networks and Learning Systems. 29, 5, 1774–1785. DOI= https://doi.org/10.1109/tnnls.2017.2673241.