

Sistem Registrasi Dan Identifikasi Wajah Untuk Akses Fasilitas Universitas Kristen Petra Dengan Kombinasi Facenet Dan Hierarchical Navigable Small Worlds

August Berlin Tungka

Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Siwalankerto 121 – 131 Surabaya 60236

+62-31-298-3452

c14180003@john.petra.ac.id

ABSTRAK

Kartu Tanda Mahasiswa digunakan untuk menggunakan fasilitas dan acara Universitas Kristen Petra. Masalah yang dapat timbul dengan kartu-kartu tersebut adalah penyalahgunaan oleh oknum-oknum yang tidak bertanggungjawab. Karena itu, Universitas memerlukan sistem identifikasi wajah sebagai alternatif pengganti. Tujuan penelitian ini adalah untuk Menyusun sistem identifikasi wajah yang cepat, akurat, dan mudah digunakan.

Metode yang dipakai untuk menyelesaikan masalah diatas adalah kombinasi metode *Facenet* dan *Hierarchical Navigable Small Worlds* (HNSW). *Facenet* digunakan dalam pengolahan data wajah sebagai vektor 128 dimensi yang akan digunakan untuk *searching*. HNSW merupakan metode pencarian *k Nearest Neighbors* yang digunakan dalam melakukan pencarian dalam Jumlah yang besar.

Dengan menggunakan metode ini, sistem melakukan identifikasi wajah dalam waktu rata-rata 1 detik.

Kata Kunci: Deteksi wajah, Pengenalan wajah, *Facenet*, *Hierarchical Navigable Small worlds*, HNSW

ABSTRACT

The Identity Cards of students are used to access the facilities and participate in events of Petra Christian University. The problem which arises from these cards is the misuse by the irresponsible group. For this, the university needs the face identification system as the alternative. This thesis is meant to build a fast, accurate, and easy to use face identification system.

The methods used to solve the problem is a combination of Facenet and Hierarchical Navigable Small Worlds (HNSW). Facenet is used to process the face into a 128-dimension vector which will be used for searching. HNSW is a k Nearest Neighbor search method which is used in a large-scale search.

Using the method, the system takes an average time of 1 second to identify faces.

Keywords: *Face detection, face recognition, Facenet, Hierarchical Navigable Small worlds, HNSW*

1. PENDAHULUAN

Universitas Kristen Petra menyediakan banyak fasilitas yang dapat digunakan mahasiswa, dosen, dan staf untuk mendukung kegiatan belajar dan mengajar. Kartu Tanda Mahasiswa / Kartu Identitas Dosen diperlukan untuk mengakses beberapa fasilitas. Kartu-kartu tersebut dapat hilang atau digunakan oleh orang lain untuk

mengakses fasilitas-fasilitas tertentu sehingga dapat mengganggu keamanan.

Untuk menyelesaikan masalah tersebut, Universitas ingin menggunakan alternatif lain dalam akses fasilitas dengan pembuatan sistem yang hanya memerlukan wajah untuk menggantikan Kartu Tanda Mahasiswa. Wajah lebih mudah digunakan untuk mengenali seseorang daripada sidik jari dan iris [1]. Selain itu, identifikasi wajah tidak membutuhkan peralatan khusus dan aktivitas subjek yang lebih sedikit daripada sidik jari dan mata.

Terdapat beberapa aplikasi identifikasi wajah yang telah dirilis dan ramai digunakan. *Churchix* dapat melakukan identifikasi wajah secara real-time, namun menyimpan data dalam bentuk foto mentah. *Presensiku* menggunakan identifikasi wajah dari foto *selfie* atau *wefie* untuk melakukan presensi di lokasi tertentu dan izin. Namun, *Presensiku* menyimpan data foto wajah untuk presensi dan izin secara mentah. *Bri.co.id* menyediakan layanan pembukaan rekening *online* di mana pengguna melakukan verifikasi dengan rekaman *video* dan foto *selfie* dengan KTP. Foto wajah yang disimpan secara mentah dapat mengancam privasi pengguna jika disalahgunakan.

Masalah yang diangkat dalam penelitian ini adalah kebutuhan Universitas Kristen Petra akan pembuatan sistem identifikasi wajah yang efektif untuk akses fasilitas. Sebuah *database* wajah disiapkan untuk menyimpan wajah mahasiswa, dosen, dan tenaga kependidikan. Untuk menjaga privasi dan agar tidak membebani server, data wajah disimpan sebagai *feature descriptor*. Sistem ini siap diuji coba setelah *database* disiapkan. Wajah dari subjek diproses menggunakan *Facenet* kemudian dicocokkan dengan *database* menggunakan *Hierarchical Navigable Small Worlds*. Jika ada kecocokan dengan *database*, subjek diperbolehkan mengakses fasilitas.

Penelitian ini diharapkan dapat menghasilkan sistem yang dapat membantu menciptakan lingkungan universitas yang aman.

2. PEMBAHASAN

2.1 Face Recognition

Identifikasi wajah (*Face Recognition*) adalah salah satu topik penelitian dalam *Computer Vision* dan *Deep Learning* [4]. Wajah merupakan fitur biometrik yang umum dan familiar untuk mengidentifikasi seseorang dan tidak membutuhkan peralatan khusus atau bantuan langsung dari subjek [1]. Perbedaan lebih lanjut ditunjukkan pada Tabel 1.

Tabel 1. Perbedaan Identifikasi Wajah dan Sidik Jari [1]

Karakteristik	Identifikasi Wajah	Sidik Jari
Aspek Ergonomis	<i>Non-invasive</i>	<i>Invasive</i>
Harga	<i>Medium cost</i> dan <i>multi-purpose</i>	<i>Medium cost</i> dan <i>single purpose</i>
Reliabilitas	Sangat tinggi	tinggi
Penerimaan sosial	Tinggi	Sangat tinggi
Akurasi	Tinggi	Sangat tinggi
Kemudahan Spoofing (Membuat biometric palsu)	Sangat sulit	Sangat mudah
Analisis perilaku	Dapat dilakukan (dinamis)	Tidak dapat dilakukan

Identifikasi wajah dilakukan dalam tiga langkah [5]. Wajah dideteksi dari gambar atau *video* menggunakan *Face Detector* untuk memperoleh *bounding box* letak wajah. Wajah dapat dideteksi dengan ekspresi, intensitas, dan skala yang bervariasi. Selanjutnya, fitur-fitur wajah (mata, mulut, dan hidung) digunakan untuk mengatur letak wajah sesuai koordinat-koordinat yang telah diatur. Ini dilakukan untuk mengurangi efek dari rotasi dan ukuran wajah dalam *image*. Wajah tersebut kemudian diproses menjadi *feature descriptor* yang dibandingkan menggunakan *similarity score* yang dihitung melalui metode tertentu. Pasangan wajah dengan *similarity score* yang memenuhi batas tertentu dianggap memiliki identitas yang sama.

2.2 Facenet

Facenet dikembangkan oleh Schroff, Kalenichenko & Philbin [6] untuk identifikasi wajah. Facenet menggunakan *Deep Convolutional Network* untuk memproses wajah langsung menjadi vektor 128 dimensi. Wajah tidak diberikan *alignment* 2D maupun 3D sebelum *feature extraction*.

Training dilakukan menggunakan *triplet loss*. *Triplet* terdiri atas dua wajah dengan identitas yang sama dan satu dari identitas yang berbeda.

Pada tahap awal pengembangan, Facenet menggunakan dua tipe *neural network* - konvensional dan *Inception network*. Dari hasil pengujian, *Inception network* memiliki tingkat akurasi yang lebih tinggi daripada *Neural Network* Konvensional. *Inception Resnet v1* digunakan dalam tahap pengembangan selanjutnya oleh David Sandberg.

2.3 Nearest Neighbor Search

Face Recognition dengan *Deep Learning* memproses wajah menjadi *high-dimensional vectors* dengan ukuran mulai dari 100 dimensi [6]. Mencari Exact Nearest Neighbor merupakan problem dengan computation cost yang sangat tinggi karena *Curse of Dimensionality* [2]. Waktu dan *computing space* yang digunakan berskala linier dengan jumlah data dan eksponensial terhadap *dimensionality* dari dataset. Hal ini menyebabkan metode *Exact Nearest Neighbors search* tidak efisien untuk *high-dimensional dataset* dengan jumlah data yang besar. Hampir semua *Exact*

Method lebih buruk daripada *Brute Force Method* untuk dataset dengan *dimensionality* yang tinggi [7].

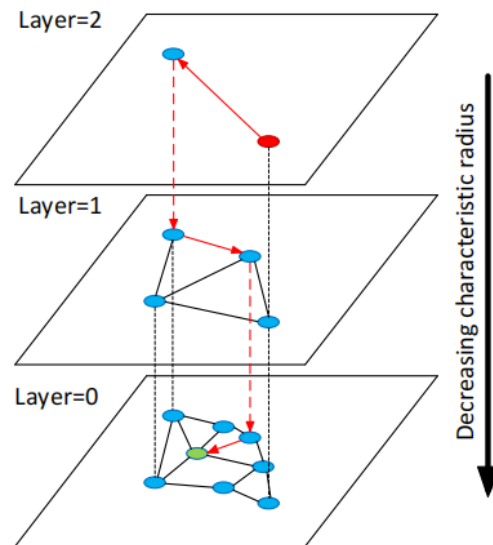
Untuk menangani *dataset* berjenis ini, penelitian algoritma searching berfokus pada *Approximate Nearest Neighbors Search* [2]. *Approximate Nearest Neighbors Search* melakukan *searching* dengan hasil berupa perkiraan *Nearest Neighbors* yang sebenarnya. Hasil akan disaring berdasarkan jarak *Nearest Neighbor* hasil dengan *Nearest Neighbor* yang sebenarnya.

2.4 Hierarchical Navigable Small Worlds

Hierarchical Navigable Small Worlds (*Hierarchical NSW*, *HNSW*) merupakan salah satu metode Nearest Neighbor Search yang digunakan untuk mencari dari data berskala besar. Metode ini dikembangkan oleh Yu. A. Malkov dan D. A. Yashunin [3] dengan pendekatan Proximity Graph.

Hierarchical NSW menghubungkan data-data (elemen-elemen) menggunakan *link*. *Link* tersebut kemudian dipisahkan ke beberapa *layer* di dalam *multilayer graph* berdasarkan jarak antar data. Jumlah *link* per elemen di semua *layer* dapat diatur dengan nilai konstan untuk meningkatkan skalabilitas.

Searching dimulai dari *layer* teratas dengan jarak antar elemen yang panjang. Searching dilakukan secara *greedy* hingga elemen terdekat dengan elemen yang dicari (*local minimum*) ditemukan di *layer* tersebut. Searching kemudian dimulai ulang dari elemen tersebut di *layer* berikutnya. Proses ini diulang terus menerus hingga *layer* terbawah. Proses ini ditunjukkan dalam Gambar 1.



Gambar 1. Skema Pencarian dalam Hierarchical Navigable Small Worlds

3. DESAIN SISTEM

3.1 Analisis Kebutuhan

Kebutuhan yang akan dipenuhi dalam penelitian ini adalah kebutuhan akan sistem registrasi dan identifikasi wajah yang praktis dan cepat. Untuk membuat sistem yang cepat dan praktis, aplikasi pendaftaran wajah dikembangkan untuk *handphone* agar dapat digunakan banyak orang.

Untuk mengatasi kemungkinan perubahan wajah pengguna, data wajah dapat diperbarui sesuai kebutuhan. Untuk memastikan agar

data wajah selalu diperbarui, aplikasi pendaftaran wajah hanya menerima input kamera. Karena ada kemungkinan perbedaan yang signifikan antara wajah di dalam foto dengan wajah pengguna saat ini, aplikasi tidak menerima input berupa foto galeri.

3.2 Identifikasi Wajah menggunakan Facenet

Untuk Menyusun sistem identifikasi wajah yang akurat, komponen pengenalan wajah yang digunakan adalah Facenet. Implementasi Facenet yang digunakan adalah implementasi oleh David Sandberg yang dijelaskan dalam Subbab 2.2.

3.2.1 Arsitektur Jaringan

Dalam implementasi oleh David Sandberg, arsitektur jaringan yang digunakan adalah Inception Resnet v1 dengan *input* berukuran 160 x 160 x 3. *Output* yang dihasilkan merupakan vektor yang berukuran 128 dimensi.

3.2.2 Pretraining

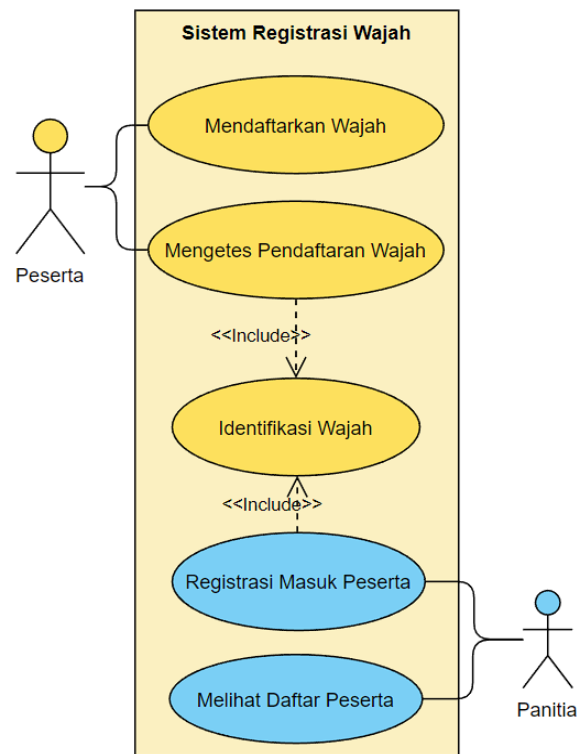
Pengenalan wajah akan dilakukan terhadap mahasiswa dalam jumlah yang besar (10.000 orang). Karena keterbatasan *dataset* wajah mahasiswa yang ada, *pretraining* dengan *dataset* yang besar dilakukan untuk meningkatkan akurasi. *Pretraining* dilakukan menggunakan *dataset* VGGFace2 dan *softmax loss function*. *Training* dilakukan sebanyak 275.000 langkah selama 30 jam menggunakan GPU Nvidia Titan X Pascal. *Framework* yang digunakan untuk *pretraining* adalah Tensorflow r1.7, CUDA 8.0, dan CuDNN 6.0.

3.3 Pencarian Menggunakan Hierarchical Navigable Small Worlds

Untuk melakukan pencarian dari data yang berjumlah besar (10.000 orang) dalam waktu yang singkat, sistem menggunakan metode *Hierarchical NSW* yang dijelaskan dalam Subbab 2.4. Implementasi *Hierarchical NSW* yang digunakan adalah *framework* ElasticSearch dengan fungsi *k-nearest Neighbor* (kNN) *search*. Elasticsearch juga digunakan sebagai *framework* penyimpanan vektor wajah.

3.4 Use Case Diagram

Use case diagram menjelaskan gambaran fitur yang dapat digunakan oleh masing-masing pihak pada saat menggunakan sistem yang telah dibuat. Sistem digunakan oleh peserta acara untuk mendaftarkan wajah dan mengecek pendaftaran wajah. Panitia acara menggunakan sistem untuk mengecek daftar peserta dan melakukan registrasi masuk. Pengetesan pendaftaran wajah dan registrasi masuk peserta menggunakan fungsi identifikasi wajah. *Use Case Diagram* ditunjukkan dalam Gambar 2.



Gambar 2. Use Case Diagram dari Sistem

3.5 Database

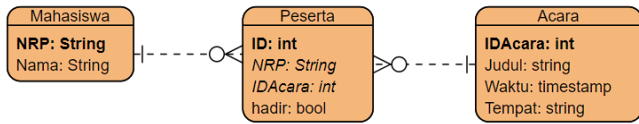
Database pendaftaran terdiri atas dua database: *database* acara dan *database* wajah.

Database wajah terdiri atas tabel wajah yang menyimpan data NRP, nama, dan vektor wajah. Tabel 2 menjelaskan tabel wajah.

Tabel 2. Tabel Database Wajah

Nama atribut	Tipe	Atribut	Keterangan
ID	String	Primary key	ID data wajah
NRP/NIP	keyword		ID pemilik data wajah
name	keyword		Nama pemilik data wajah
vector	Dense_vector 128		Feature descriptor wajah

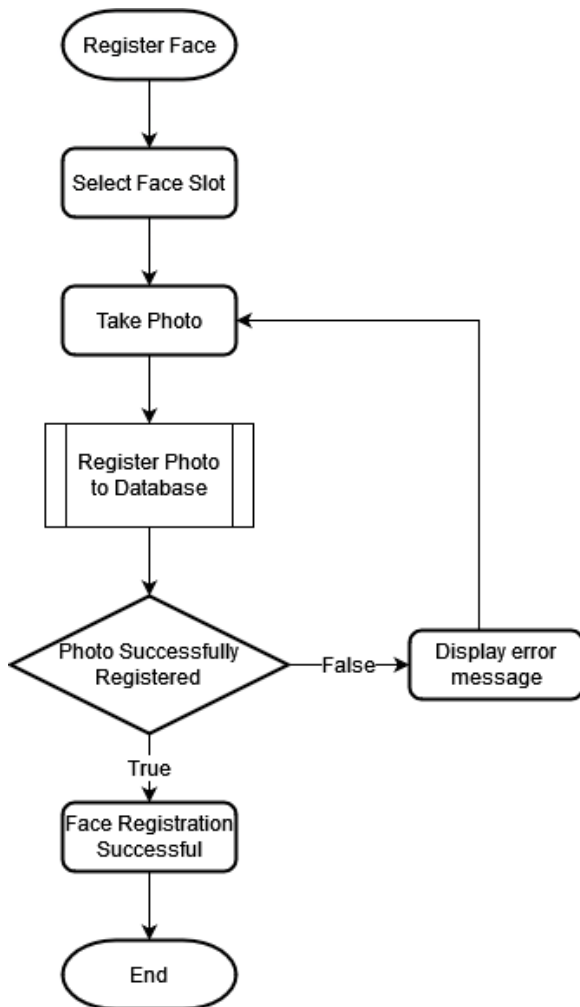
Database acara terdiri atas tabel mahasiswa, tabel acara, dan tabel peserta acara. Tabel mahasiswa menyimpan informasi nama dan NRP mahasiswa. Tabel acara menyimpan nama, waktu, dan tempat pelaksanaan acara. Tabel peserta menyimpan data acara yang didaftar oleh mahasiswa dan apakah mahasiswa yang bersangkutan hadir dalam acara tersebut. *Entity Relationship Diagram* dari *Database* acara dapat dilihat dalam Gambar 3.



Gambar 3. Entity Relationship Diagram dari Database Acara

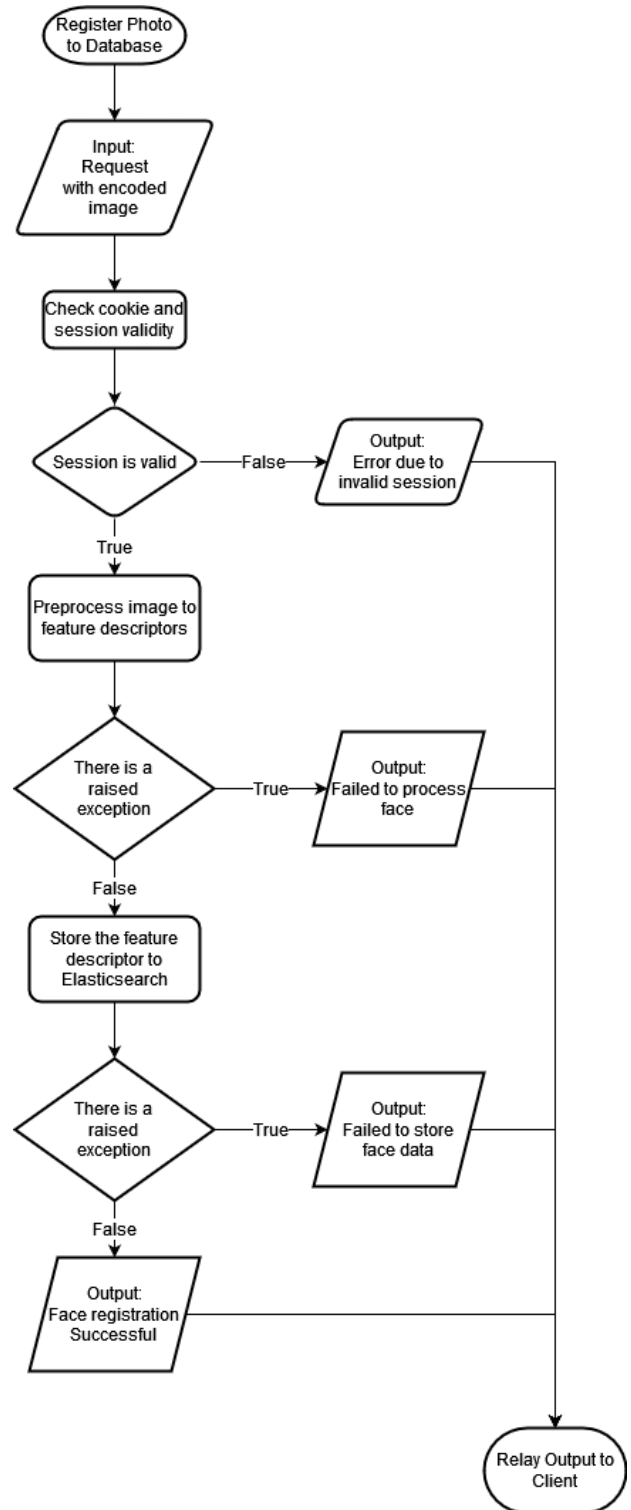
3.6 Alur Pendaftaran Wajah

Sistem akan meminta pengguna untuk memilih slot wajah yang akan didaftarkan. Setelah memilih slot wajah, pengguna akan diminta untuk mengambil foto wajah. Foto tersebut ditunjukkan agar pengguna dapat memilih untuk mengambil ulang foto atau melanjutkan pendaftaran. Jika pengguna memilih untuk melanjutkan pendaftaran, pengguna akan diminta untuk mengisi data identitas sebelum mengirim foto. Foto dikirim ke *database* lalu diproses seperti yang ditunjukkan dalam Gambar 5, lalu aplikasi akan menunggu hasil pendaftaran. Setelah hasil diterima, pengguna dapat memilih untuk mengulangi pendaftaran atau kembali ke menu utama. Jika pendaftaran foto berhasil, pengguna dapat melihat pesan bahwa wajah berhasil disimpan. Jika pendaftaran foto gagal, pesan bahwa penyimpanan wajah gagal dan alasannya akan ditunjukkan. *Flowchart* dari proses ini ditunjukkan pada Gambar 4.



Gambar 4. Flowchart Registrasi Wajah (Front-end)

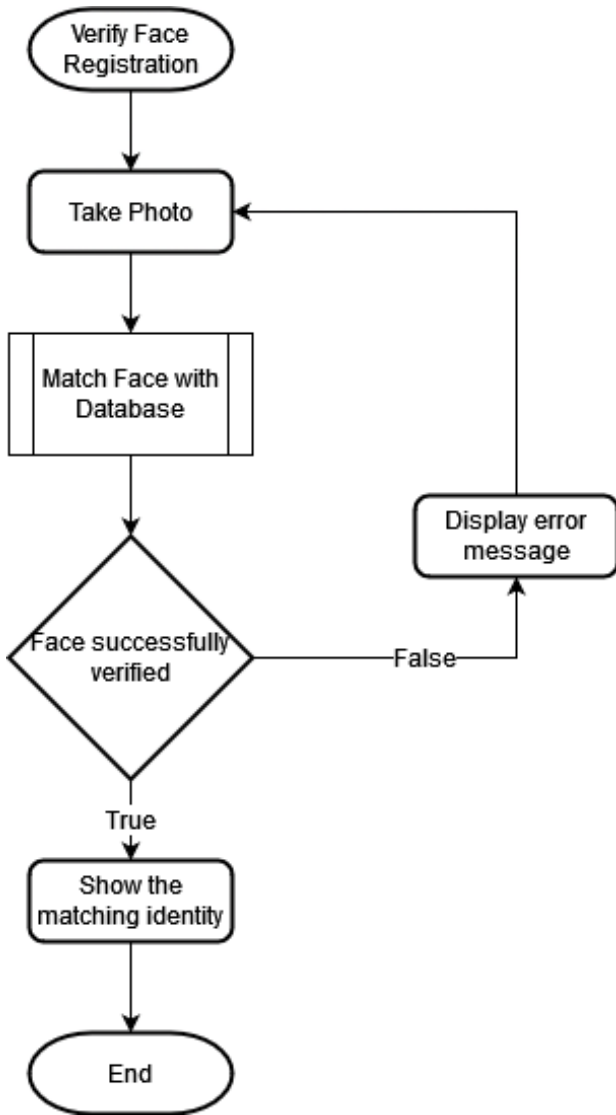
Foto wajah diambil lalu ditunjukkan untuk memberi pengguna pilihan untuk mengambil ulang foto atau mendaftarkan wajah. Wajah yang didaftarkan kemudian diproses lalu disimpan di dalam *server*. Hasil atau kegagalan pendaftaran kemudian ditunjukkan kepada pengguna. *Flowchart* dari proses ini ditunjukkan pada Gambar 5.



Gambar 5. Flowchart Registrasi Wajah (Back End)

3.7 Alur Identifikasi Wajah

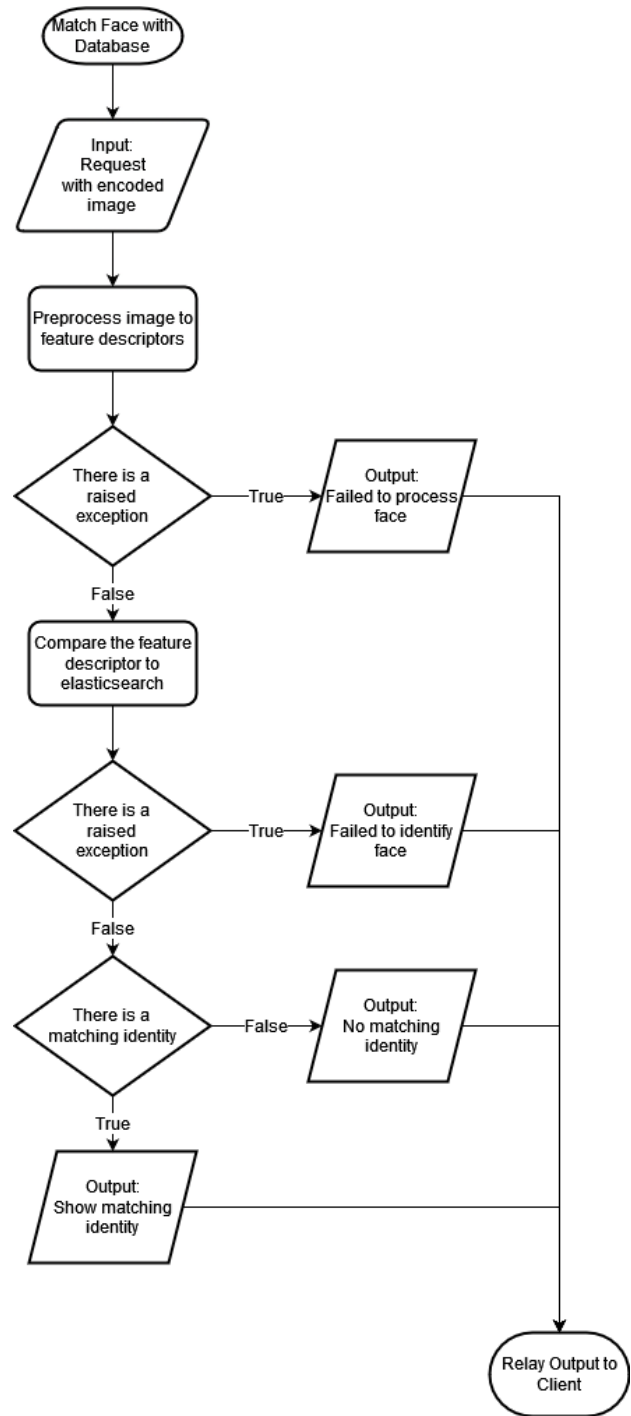
Dalam fungsi pengecekan pendaftaran wajah atau pendaftaran masuk peserta, identifikasi wajah akan dilakukan. Pengguna akan diminta untuk mengambil foto menggunakan kamera. Hasil foto kemudian dicocokkan dengan *database* seperti yang ditunjukkan oleh Gambar 7. Hasil identifikasi lalu dikirimkan kepada aplikasi dan ditunjukkan kepada pengguna. Jika verifikasi wajah gagal, pesan bahwa wajah gagal dikenali ditunjukkan kepada pengguna. Jika identifikasi wajah berhasil, identitas yang cocok dengan wajah yang difoto akan ditunjukkan kepada pengguna. Flowchart dari proses ini ditunjukkan dalam Gambar 6.



Gambar 6. Flowchart Identifikasi Wajah (Front End)

Di dalam server, data wajah diproses menjadi vektor *feature descriptor*. Jika proses gagal, pesan kegagalan pemrosesan wajah akan disampaikan ke *client*. Jika proses berhasil, *feature descriptor* kemudian dicocokkan di dalam *database* ElasticSearch. Jika identifikasi gagal, pesan kegagalan penyimpanan disampaikan ke *client*. Jika identifikasi berhasil, hasil identifikasi akan diproses. Jika ada hasil identifikasi, identitas yang cocok akan ditunjukkan

kepada *client*. Jika tidak ada kecocokan, pesan bahwa tidak ada kecocokan akan ditunjukkan kepada *client*. Flowchart dari proses ini ditunjukkan dalam Gambar 7.



Gambar 7. Flowchart Identifikasi Wajah (Back-end)

4. HASIL PENGUJIAN

Pengujian dilakukan dalam dua tahap: Pengujian akurasi identifikasi dan lama identifikasi. Faktor-faktor lainnya yang mempengaruhi hasil dijelaskan dalam Subbab 4.3.

4.1 Pengujian Akurasi Identifikasi

Pengguna mengambil foto wajah yang menghadap kamera. Setelah foto wajah diambil, pengguna kemudian mengisi data nama, nomor identitas, dan slot. Foto wajah dan data identitas didaftarkan ke server ketika pengguna menggunakan tombol “Daftarkan Wajah”. Setelah mendaftarkan wajah, pengguna mengambil foto wajah di mana mereka menghadap kamera (0 derajat). Kemudian, wajah pengguna yang menghadap ke arah 30, 45, 60, dan 90 derajat dari kamera akan dicocokkan dengan database. Untuk setiap sudut dan wajah subyek, pengujian identifikasi dilakukan sekali.

Hasil pengujian akan diberi label *true* jika wajah subyek terdaftar dan hasil identifikasi yang muncul sesuai dengan wajah subyek. Hasil pengujian juga diberi label *true* jika subyek tidak terdaftar dan hasil identifikasi menyatakan bahwa tidak ada kecocokan. Hasil pengujian akan diberi label *false* jika wajah subyek terdaftar namun hasil identifikasi yang muncul tidak sesuai dengan wajah subyek atau tidak ada kecocokan.

Hasil pengujian ditunjukkan pada Tabel 3.

Tabel 3. Hasil Pengujian Akurasi Identifikasi

Nama (NRP)	0°	30°	45°	60°	90°
angelo brian rafaél (c14200062)	True	True	False	False	False
Michael Halim (c14190119)	True	True	True	True	False
Peter Yudhistira (c14190067)	False	False	True	False	False
Denzel Daniel D'Assante Tangsaputra (C14200160)	True	True	True	False	False
michael christian wijaya (c14200087)	True	True	True	True	False
Marcellino Julian Gozal (c14200043)	True	True	False	False	False
Glenn Steven Santoso (c14200010)	True	False	False	False	False
Eric Saputra Lays (c14180048)	True	False	False	False	False
Jimmy Marcelino (c14200177)	False	False	False	False	False
Kalvin Allan (c14200072)	True	False	False	False	false
August Berlin Tungka (c14180003)	True	False	False	False	False
Alvin Andrianto (c14210202)	False	False	True	True	False

Dari hasil pengujian di atas, 9 dari 12 identifikasi dengan wajah menghadap kamera menghasilkan *true*. Pada sudut 45 dan 30

derajat, tingkat akurasi menurun menjadi 41,2% (5 dari 12). Pada sudut 60 derajat, tingkat akurasi menurun menjadi 25%. Aplikasi tidak dapat melakukan identifikasi yang akurat pada sudut 90 derajat. Dari hasil percobaan ini, sudut pengambilan foto wajah berpengaruh terhadap hasil identifikasi.

4.2 Pengujian Lama Identifikasi

Pengujian waktu dimulai ketika anggota panitia mulai menunjukkan layar kamera. Peserta acara kemudian mengambil foto wajah mereka lalu menekan tombol untuk identifikasi. Waktu awal disimpan di dalam variabel *start* sebelum pesan “memverifikasi wajah” ditunjukkan.

Pengujian waktu selesai ketika hasil identifikasi muncul di aplikasi. Waktu akhir disimpan di dalam variabel *end* setelah hasil identifikasi ditunjukkan. Selisih antara variabel *start* dan *end* dihitung lalu dikonversi dalam satuan mikrodetik lalu dibagi dengan angka satu juta. Konversi dalam satuan detik menghasilkan angka bulat. Pembagian hasil waktu dengan angka satu juta dilakukan dengan dasar bahwa satu detik setara dengan satu juta mikrodetik. Hasil pembagian selisih waktu kemudian ditunjukkan di dalam *log* perangkat. *Log* perangkat kemudian difilter untuk memperoleh waktu hasil perhitungan.

Hasil Pengujian ditunjukkan dalam Tabel 4.

Tabel 4. Hasil waktu Identifikasi

Pengujian ke-	Waktu (detik)
1	4.976546
2	0.751893
3	0.758061
4	0.857452
5	0.79405
6	1.060784
7	1.031164

Identifikasi yang dilakukan pertama kali setelah sistem mulai berjalan membutuhkan waktu yang lebih lama untuk memulai fungsi pengolahan foto wajah dan searching. Setelah identifikasi pertama, identifikasi memerlukan waktu sekitar satu detik. Lama identifikasi dapat dipengaruhi oleh keadaan jaringan dan performa server.

4.3 Pembahasan Faktor-faktor Tambahan

Akurasi identifikasi dipengaruhi oleh beberapa faktor. Hasil pengujian akurasi menunjukkan bahwa sudut pengambilan foto wajah berpengaruh terhadap hasil identifikasi. Dalam pengujian di mana penulis melakukan identifikasi dengan tingkat cahaya yang berbeda, wajah penulis gagal diidentifikasi. Hal ini menunjukkan bahwa tingkat cahaya juga mempengaruhi hasil identifikasi.

5. KESIMPULAN

Berdasarkan dari hasil penelitian dan pembahasan, maka kesimpulan dalam penelitian ini adalah sebagai berikut:

1. Sistem dapat menangani identifikasi wajah namun tingkat akurasi masih di bawah 80%.
2. Sistem dapat melakukan identifikasi wajah dalam waktu 1 detik, namun dipengaruhi oleh jaringan dan performa *server*.

6. SARAN

Berdasarkan kesimpulan yang ada, saran yang dikemukakan penulis untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Melakukan pengujian lebih dalam penggunaan baterai.
2. Melakukan pengujian waktu yang diperlukan untuk terjadi *error* yang mengharuskan pengguna melakukan *restart* aplikasi.
3. Melakukan perbandingan waktu registrasi masuk menggunakan *handphone* dan jaringan seluler.
4. Melakukan pengembangan lebih lanjut untuk meningkatkan akurasi identifikasi wajah.

7. DAFTAR PUSTAKA

- [1] Adjabi, I., Ouahabi, A., Benzaoui, A., & Taleb-Ahmed, A. (2020). Past, Present, and Future of Face Recognition: A Review. *Electronics*, 9 (8), 1188. DOI: 10.3390/electronics9081188.
- [2] Aumüller, M., Bernhardsson, E., & Faithfull, A. (2020). ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Information Systems*, 87, 101374. DOI: 10.1016/j.is.2019.02.006.
- [3] Malkov, Y. A. & Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42 (4), 824-836. DOI: 10.1109/TPAMI.2018.2889473.
- [4] Mei, W & Weihong, D. (2021). Deep face recognition: A survey. *Neurocomputing*, 429, 215-244. DOI: 10.1016/j.neucom.2020.10.081.
- [5] Ranjan, R. et al. (2018). Deep learning for understanding Faces: machines may be just as good, or better, than humans. *IEEE Signal Processing Magazine*, 35 (1), 66-83. DOI: 10.1109/MSP.2017.2764116.
- [6] Schroff, F., Kalenichenko, D., Philbin, J. (2015). *FaceNet: A unified embedding for face recognition and clustering*. Paper presented at Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA (pp. 815–823). DOI: 10.1109/CVPR.2015.7298682.
- [7] Wen, L. et al. (2019). Approximate nearest neighbor search on high dimensional data — experiments, analyses, and improvement. *IEEE Transactions on Knowledge and Data Engineering*, 32 (8), 1475 - 1488. DOI: 10.1109/TKDE.2019.2909204.