

Sistem Presensi Mahasiswa Menggunakan *Face Recognition* Dengan Metode *Facenet* Pada Android

Evelyn, Rudy Adipranata, Kartika Gunadi

Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236

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

E-mail: nyleve.bukaj2@gmail.com, rudya@petra.ac.id, kgunadi@petra.ac.id

ABSTRAK

Sistem presensi kelas online di UK Petra saat ini menggunakan *QR code* untuk melakukan presensi. Sistem *QR code* ini memiliki beberapa kekurangan yaitu mudah dikelabui oleh mahasiswa. Untuk melakukan pengelabuan pada sistem *QR code* ini mahasiswa hanya perlu menitipkan akunnya kepada temannya agar bisa melakukan absensi. Sistem ini terkadang memiliki kendala untuk dieksploitasi oleh mahasiswa dengan menitipkan akun mereka kepada temannya untuk melakukan absen. Untuk menghindari mahasiswa yang menitipkan akunnya, presensi menggunakan *face recognition* merupakan salah satu cara untuk menghindarinya.

Sistem presensi mahasiswa menggunakan berbasis Android dengan menggunakan *Facenet* sebagai model untuk sistem pengenalan wajah. Untuk mengukur hasil dari pengenalan wajah menggunakan dan membandingkan nilai hasil dari rumus *L2Norm* dan *Cosine Similarity* dengan *threshold* berbeda. Hasil dari pengenalan wajah menunjukkan bahwa rumus *Cosine Similarity* paling optimal ketika menggunakan *threshold* 0.5f dengan *score* terbaik 0.5104218 dengan *score* akurasi 0.77162087. Sedangkan rumus *L2Norm* paling optimal ketika menggunakan *threshold* 8.0f dengan *score* terbaik 5.8973804 dan *score* akurasi 5.8973804.

Kata Kunci: Sistem presensi, pengenalan wajah, *Facenet*, *L2Norm*, *Cosine Similarity*, *Cloud Firestore*

ABSTRACT

As of today, the student attendance system in the University of Petra uses a QR code system to do their daily attendance. This QR code system has some flaws that are often exploited by the students. Students usually exploit this system by asking their peers to login into their accounts and take the attendance for them by scanning the shared QR code. Implementing face recognition can be one of the means of prevention.

This attendance system is an Android based system using Facenet model for the facial recognition system. Formulas L2Norm and Cosine Similarity is used for comparison means for the results of the face recognition system. Results show that cosine similarity is most optimal when using the 0.5f threshold with the score of 0.5104218 and accuracy with the score of 0.77162087. Meanwhile, L2Norm results show that it is most optimal when using the 8.0f threshold with the score of 5.8973804 and accuracy with the score of 5.8973804.

Keywords: attendance system, facial recognition, *Facenet*, *L2Norm*, *Cosine Similarity*, *Cloud Firestore*.

1. PENDAHULUAN

Sistem presensi kelas online di UK Petra saat ini menggunakan *QR code* untuk melakukan presensi. Sistem *QR code* ini memiliki beberapa kekurangan yaitu mudah dikelabui oleh mahasiswa. Untuk melakukan pengelabuan pada sistem *QR code* ini mahasiswa

hanya perlu menitipkan akunnya kepada temannya agar bisa melakukan absensi. Sistem ini terkadang memiliki kendala untuk dieksploitasi oleh mahasiswa dengan menitipkan akun mereka kepada temannya untuk melakukan absen.

Untuk menghindari mahasiswa yang menitipkan akunnya, presensi menggunakan *face recognition* merupakan salah satu cara untuk menghindarinya. Sistem *face recognition* ini tidak hanya bermanfaat untuk menghindari kecurangan pada absensi kelas namun, juga dapat diimplementasikan untuk sistem presensi pada acara-acara perkuliahan seperti acara HIMA yang membutuhkan presensi peserta untuk merekap kehadirannya. Sistem akan melakukan penyimpanan wajah mahasiswa sehingga jika bukan pemilik asli akunnya tidak bisa melakukan presensi. *Face recognition* memiliki banyak metode seperti *eigenface*, *facenet*, dan beberapa lainnya. *Facenet* merupakan metode baru yang dikembangkan oleh Google, yang menggunakan algoritma *convolutional neural network* (CNN). *Facenet* merupakan pengembangan dari *Tensorflow* yang merupakan *library machine learning* yang dimiliki oleh Google. *Tensorflow* bisa digunakan diberbagai platform sehingga cukup fleksibel. Sementara *Opencv* adalah framework lain yang digunakan untuk *face recognition* dengan menggunakan algoritma *eigenface*. *Eigenface* adalah algoritma *face recognition* yang ditemukan pada tahun 2008 dan dikembangkan lagi oleh *opencv*. Oleh karena itu *opencv* sudah agak tertinggal teknologinya dibandingkan *tensorflow* sehingga akurasi yang dihasilkan *facenet* lebih tinggi dari *opencv* karena terdapat *machine learning* pada metode *facenet*. [8]

Untuk menghindari masalah pada jaringan internet yang sering digunakan mahasiswa untuk melakukan presensi terlambat, penyimpanan data akan menggunakan *firebase*. *firebase* memiliki sistem untuk melakukan pengiriman data menunggu koneksi internet sudah lancar. Sistem presensi akan meminta *permission* untuk mengakses waktu pada perangkat untuk melakukan pengecekan pada perangkat, apakah sudah menggunakan settingan waktu *auto* atau tidak. Pada waktu pengiriman data jika terjadi kendala internet waktu sudah tersimpan terlebih dahulu, sehingga tidak bisa dikelabui dengan alasan internet terganggu. Bagian *firebase* yang digunakan adalah *Cloud Firestore* yang sudah memiliki sistem tersebut. Data wajah user akan disimpan di dalam *firebase* untuk melakukan pengecekan saat pengiriman.

Fitur *firebase* lainnya yang digunakan untuk melakukan *face recognition* adalah *MLKit* untuk melakukan *face detection* pada perangkat. *MLKit* merupakan seperangkat *library* untuk melakukan berbagai hal menggunakan *machine learning*. Untuk melakukan *face recognition* dibutuhkan *MLKit* untuk melakukan *face detection* dan *Facenet* untuk melakukan *face recognition*. Untuk mendapatkan data dari kamera dibutuhkan *library CameraX* yang berfungsi untuk mengolah hasil dari kamera *handphone*. Hasil dari *CameraX* dapat digunakan pada fungsi didalam *MLKit*.

2. LANDASAN TEORI

2.1 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Sebelum dibeli oleh Google, Android dikembangkan dan dimiliki oleh Android Inc., tetapi masih dibantu dana oleh google dalam pengembangannya. Sistem operasi ini dirilis resmi tahun 2007. Bersamaan dengan *Open Handset Alliance*, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. [9]

2.2 Kotlin

Kotlin adalah bahasa pengembangan dari Java yang sejak 2016 menjadi bahasa utama untuk Android studio. Di antara perkembangan perangkat lunak Java yang penting dalam beberapa tahun terakhir adalah munculnya bahasa alternatif untuk JVM seperti Groovy, Jython, dan JRuby — dan, baru-baru ini, munculnya Scala. Bahasa baru yang diketik secara statis, Kotlin, dinamai menurut nama sebuah pulau Rusia di lepas pantai St. Petersburg, tempat tinggal Andrey Breslav dari Kotlin dan tim Kotlin, baru-baru ini mendapat perhatian. Sebuah gagasan dari perusahaan pengembang perangkat lunak Ceko yang sangat dipuji, JetBrains, pembuat Java IDE IntelliJ IDEA, Kotlin dinobatkan sebagai Bahasa Bulan Ini dalam edisi Januari 2012 Jurnal Dr. Dobb.[7]

2.3 Android Studio

Android Studio adalah salah satu *integrated development environment* (IDE) resmi untuk android yang dikembangkan oleh Google dan dibangun diatas IntelliJ. [3] Android studio dikhususkan untuk pengembangan aplikasi android dan menyediakan alat-alat yang telah terintegrasi yang dapat memudahkan untuk pengembangan aplikasi seperti Git, dan Gradle.

2.4 Tensorflow

TensorFlow adalah *library machine learning* yang dapat digunakan untuk berbagai hal. TensorFlow merupakan *library opensource* yang dikembangkan oleh Google. TensorFlow memiliki anak bagian yang ditujukan untuk IoT dan *mobile* bernama TensorFlow Lite. TensorFlow Lite dapat digunakan untuk *Image classification, Object Detection, Pose Estimation, dan Segmentation*. [1]

2.5 CameraX

CameraX adalah *support library* Jetpack, yang dibuat untuk memudahkan Anda mengembangkan aplikasi kamera. CameraX memiliki antarmuka API yang konsisten dan mudah digunakan yang dapat digunakan di sebagian besar perangkat Android, dengan kompatibilitas hingga Android 5.0 (API level 21).

Meskipun memanfaatkan kapabilitas camera2, CameraX menggunakan pendekatan berbasis kasus penggunaan yang lebih sederhana, yang memperhatikan siklus proses. CameraX juga menyelesaikan masalah kompatibilitas perangkat secara otomatis sehingga tidak perlu menyertakan kode khusus perangkat dalam codebase. Fitur-fitur ini mengurangi jumlah kode yang perlu ditulis saat menambahkan kapabilitas kamera ke aplikasi. [6]

2.6 Cloud Firestore

Cloud Firestore adalah *database* yang fleksibel dan skalabel untuk pengembangan seluler, *web*, dan *server* di Firebase dan *Google Cloud Platform*. Seperti *Firestore Realtime Database, Cloud Firestore* membuat *data* Anda tetap terhubung di aplikasi klien

melalui *listener realtime* dan menawarkan dukungan secara *offline* untuk seluler dan *web*. Dengan begitu, Anda dapat membuat aplikasi yang responsif dan mampu bekerja tanpa harus bergantung pada latensi jaringan atau koneksi Internet. Cloud Firestore juga menawarkan integrasi yang lancar dengan produk Firebase dan *Google Cloud Platform* lainnya, termasuk *Cloud Functions*. [4]

2.7 MLKit

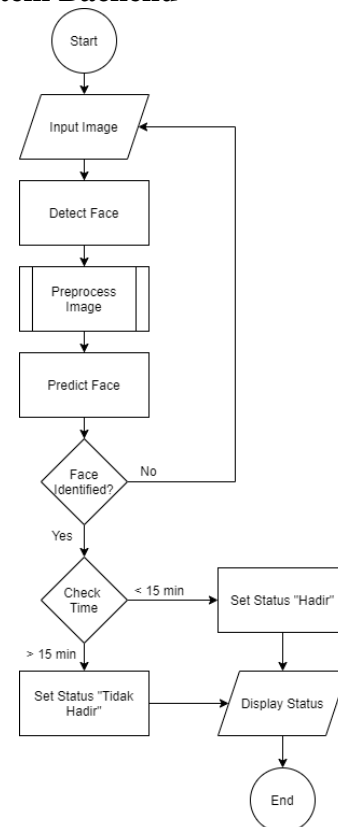
MLKit adalah SDK seluler yang menghadirkan keahlian *machine learning* Google untuk aplikasi Android dan iOS dalam paket yang handal dan mudah digunakan. Baik untuk pemula maupun yang sudah berpengalaman dalam menggunakan *machine learning*, MLKit juga dapat diimplementasikan pada fungsi yang diperlukan hanya dengan beberapa baris kode. Tidak perlu pengetahuan mendalam tentang jaringan *neural* atau pengoptimalan *model* untuk memulai. Di sisi lain, MLKit menyediakan API yang mudah digunakan dan dapat membantu Anda menggunakan model TensorFlow Lite kustom di aplikasi seluler. [5]

2.8 Facenet

FaceNet adalah sebuah *neural network* yang memetakan wajah seseorang menjadi *Euclidean space* (kumpulan dari geometrical point) yang dimana nantinya *geometrical point* tersebut menentukan nilai untuk mengukur tingkat kemiripan wajah. [4] Sehingga, jika wajah tersebut semakin mirip akan membuat nilainya semakin kecil dan kebalikannya jika wajah tersebut semakin berbeda akan menghasilkan nilai yang semakin besar [2]. Dimana rumus yang digunakan untuk Facenet adalah rumus L2Norm dan *Cosine Similarity*.

3. DESAIN SISTEM

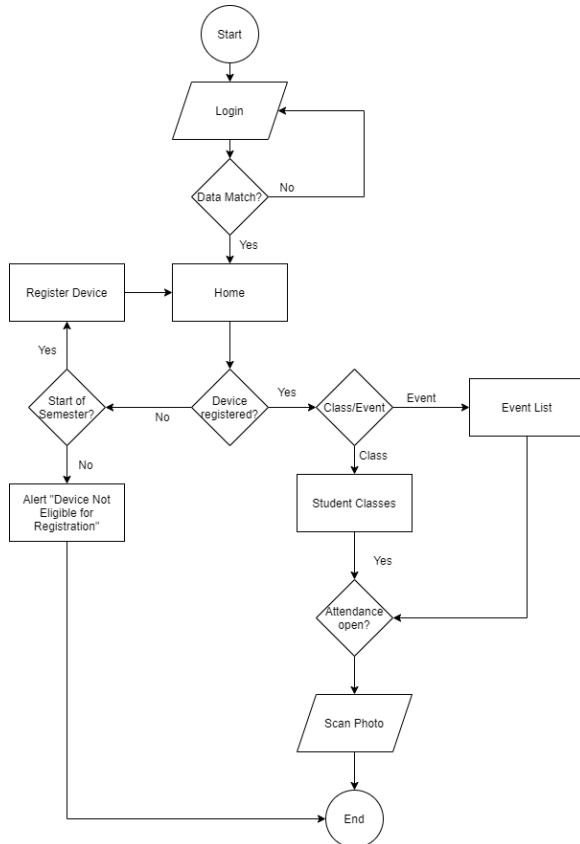
3.1 Sistem Backend



Gambar 1. Flowchart design kerja sistem Back-end

Pada Gambar 1, process pertama yang dilakukan adalah menerima input gambar yang diambil saat scanning wajah dari kamera. Gambar yang telah diinput ini kemudian akan di preprocess untuk *cropping* dan *resizing*. Setelah tahap preprocessing, akan terjadi prediksi wajah dalam gambar tersebut, jika *output* dari prediksi benar maka selanjutnya akan ada pengecekan waktu pengambilan gambar atau *scan* wajah mahasiswa. Apabila *scan* dilakukan sebelum batas waktu yang ditentukan maka status mahasiswa akan dinyatakan hadir dalam kegiatan perkuliahan. Jika prediksi gagal, maka proses *scanning* atau pengambilan gambar akan dilakukan kembali.

3.2 Alur Presensi



Gambar 2. Flowchart Alur Presensi Front-end

Pada Gambar 2 aplikasi akan dimulai dari page login dimana mahasiswa akan menginput NRP dan password untuk verifikasi masuk kedalam aplikasi. Setelah berhasil login, akan tampil page Home, disini akan ada proses pengecekan tanggal, jika waktu tepat pada awal semester, mahasiswa akan diminta untuk me registrasinya perangkatnya. Jika tidak mahasiswa akan mendapatkan pilihan untuk presensi kelas atau presensi kepanitiaan sebagai peserta.

Apabila mahasiswa memilih untuk melakukan presensi kelas, mereka akan di *direct* ke halaman pilihan kelas, dimana mereka dapat memilih kelas untuk melakukan presensi. Setelah memilih kelas, mahasiswa akan melakukan scanning wajah. Jika hasil scan wajah valid dan waktu presensi tidak lebih dari *time range* yang diberikan, maka mahasiswa akan tercatat “Hadir”.

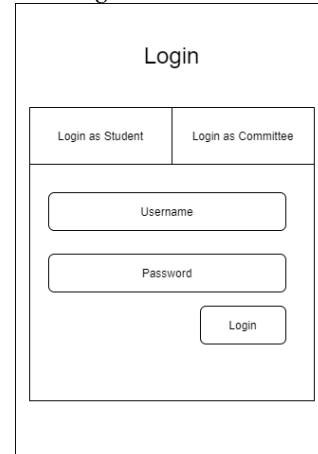
Apabila peserta memilih untuk melakukan presensi acara, mereka akan di *direct* ke halaman pilihan acara yang menge-list daftar acara yang mereka ikuti. Setelah memilih acara, peserta akan

melakukan scanning wajah. Jika hasil scan wajah valid dan waktu presensi tidak lebih dari *time range* yang diberikan, maka mahasiswa akan tercatat “Hadir”.

3.3 Desain User Interface

Desain *user interface* menunjukkan desain *layout* dari aplikasi absensi. Platform yang digunakan untuk desain *user interface* berbasis Android.

3.3.1 Halaman Login



Gambar 3. User Login

Pada Gambar 3 adalah desain tampilan halaman login dari aplikasi, tampilan terdiri dari input username atau NRP, password, tombol login dan pilihan login sebagai siswa atau sebagai panitia. Jika tombol login ditekan, maka aplikasi akan mengecek dengan data yang ada di *database*. Jika data yang di inputkan benar, akan di *direct* ke halaman *Home* aplikasi.

3.3.2 Halaman List Mata Kuliah dan Acara

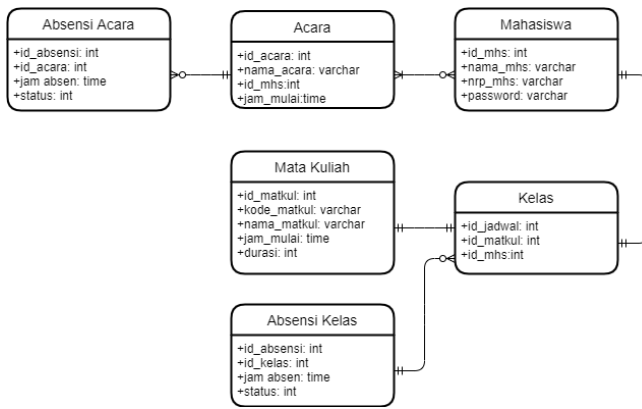
	LOGOUT		LOGOUT
CLASS 1		EVENT 1	
CLASS 2		EVENT 2	
CLASS 3		EVENT 3	
CLASS 4		EVENT 4	

Gambar 4. Halaman List Mata Kuliah dan Acara

Pada Gambar 4 adalah halaman untuk men-display semua kelas dan acara kepanitiaan yang diikuti oleh mahasiswa dengan menggunakan *Recycler View*.

3.4 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah suatu model yang digunakan untuk penyusunan database agar dapat menggambarkan relasi antar data yang akan dibuat. Desain ERD until aplikasi presensi dapat dilihat pada Gambar 5.



Gambar 5. ERD Model database sistem absensi

3.5 Desain Database

Desain database menunjukkan kolom-kolom tabel untuk penyimpanan data dalam database.

3.5.1 Tabel Mahasiswa

Tabel 1 merupakan daftar atribut yang ada di tabel mahasiswa. Tabel mahasiswa digunakan sebagai master data mahasiswa.

Tabel 1. Mahasiswa

Nama Field	PK	FK	Tipe Data	Length	Keterangan
id_mhs	*		Integer		Primary Key dalam tabel mahasiswa
nama_mhs			Varchar	50	Nama mahasiswa
nrp_mhs			Varchar	10	NRP Mahasiswa
password			Varchar	32	Password dari akun mahasiswa

3.5.2 Tabel Kelas

Tabel 2 merupakan daftar atribut yang ada di tabel kelas. Tabel Kelas merupakan tabel gabungan dari tabel mahasiswa dan tabel mata kuliah.

Tabel 2. Kelas

Nama Field	PK	FK	Tipe Data	Length	Keterangan
id_kelas	*		Integer		Primary Key dalam tabel kelas
id_matkul		*	Integer		Foreign Key dari Tabel Mata Kuliah
id_mhs		*	Integer		Foreign Key dari Tabel Mahasiswa

3.5.3 Tabel Mata Kuliah

Tabel 3 merupakan daftar atribut yang ada di tabel mata kuliah. Tabel ini digunakan sebagai master data mata kuliah.

Tabel 3. Mata Kuliah

Nama Field	PK	FK	Tipe Data	Length	Keterangan
id_matkul	*		Integer		Primary Key dalam tabel mata kuliah
kode_matkul			Varchar	10	Kode Mata Kuliah
nama_matkul			Varchar	32	Nama Mata kuliah
jam_mula			Time		Waktu perkuliahan mata kuliah dimulai
Durasi			Int		Durasi mata kuliah diajar per pertemuan

3.5.4 Tabel Absensi Kelas

Tabel 4 merupakan daftar atribut yang ada di tabel absensi kelas. Tabel absensi kelas merupakan tabel penyimpanan kehadiran mahasiswa per pertemuan kelas.

Tabel 4. Absensi Kelas

Nama Field	PK	FK	Tipe Data	Length	Keterangan
id_absensi	*		Integer		Primary Key dalam tabel absensi kelas
id_kelas		*	Integer		Foreign Key dari Tabel Kelas
jam_absen			Time		Waktu mahasiswa melakukan absensi
Status			Integer		Status kehadiran mahasiswa. Jika status = 0, Alpa. Jika status = 1, hadir.

3.5.5 Tabel Acara

Tabel 5 merupakan daftar atribut yang ada di tabel Acara. Tabel acara digunakan sebagai master data acara.

Tabel 5. Tabel Acara

Nama Field	PK	FK	Tipe Data	Length	Keterangan
id_acara	*		Integer		Primary Key dalam tabel acara

nama_acara			Varchar	32	Nama acara
id_mhs		*	Integer		Foreign Key dari Tabel Mahasiswa
jam_mulai			Time		Waktu acara dimulai

3.5.6 Tabel Absensi Acara

Tabel 6 merupakan daftar atribut yang ada di tabel absensi acara. Tabel absensi acara merupakan tabel penyimpanan kehadiran mahasiswa dalam sebuah acara.

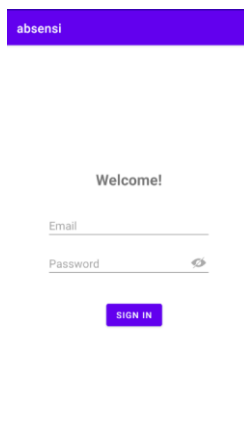
Tabel 6. Absensi Acara

Nama Field	PK	FK	Tipe Data	Length	Keterangan
id_absensi	*		Integer		Primary Key dalam tabel absensi acara
id_acara		*	Integer		Foreign Key dari Tabel Acara
jam_absen			Time		Waktu mahasiswa melakukan absensi
Status			Integer		Status kehadiran mahasiswa. Jika status = 0, Alpa. Jika status = 1, hadir.

4. PENGUJIAN SISTEM

4.1 Pengujian Login

Pada halaman login dilakukan pengujian untuk login kedalam aplikasi. Pengujian dilakukan dengan cara memasukkan email dan password yang sebelumnya sudah pernah didaftarkan. Halaman login dapat dilihat pada Gambar 6.



Gambar 6. Pengujian Login

4.2 Pengujian Homepage

Pada halaman home, mahasiswa dapat melihat semua kelas yang diikuti pada aplikasi yang dapat dilakukan absensi. Halaman home dapat dilihat pada Gambar 7.

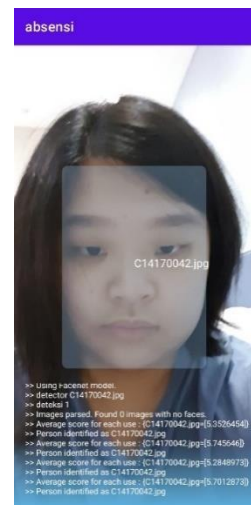


SIGN OUT

Gambar 7. Home Page

4.3 Pengujian Kamera dan Bounding Box

Ketika mahasiswa memilih kelas untuk absensi, mahasiswa akan di redirect pada halaman kamera untuk scan wajah. Wajah mahasiswa akan muncul bounding box seperti yang ada pada Gambar 8.



Gambar 8. Kamera dan bounding box

4.4 Proses Pengujian

Pada pengujian ini, terdapat 100 dataset yang digunakan untuk training, terdiri dari 5 foto dari tiap orang. Proses training dataset untuk face recognition menggunakan pretrained model dengan rumus L2Norm dan cosine similarity dengan beberapa threshold berbeda. Hasil training merupakan hasil dari perbandingan antara wajah yang ada di dataset dan wajah yang di scan.







4.4.1 Hasil L2Norm

Untuk hasil rumus L2Norm, dapat dilihat pada Tabel 7 bahwa score yang lebih kecil daripada threshold yang sudah ditentukan merupakan hasil yang benar.







Untuk menentukan rumus dan threshold yang lebih optimal untuk dipakai, pengujian dilakukan dengan membandingkan pengukuran L2Norm dengan cara membedakan hasil akhir score wajah yang di

scan dengan wajah dari orang yang sama di dataset. Hasil dapat dilihat pada Tabel 8.

Tabel 7. Tabel Training L2Norm dengan wajah berbeda

Test	Data 1	Data 2	Data 3	Data 4	Data 5
					
7.0f	9.814593	9.720658	6.203924	9.8109865	9.080256
8.0f	10.675922	11.114479	5.316419	10.055315	9.948887
9.0f	9.399267	9.166858	6.21867	9.0559025	9.047331

Tabel 8. Tabel Training Score L2Norm dengan wajah sama







Test	Data 1	Data 2	Data 3	Data 4	Data 5
					
7.0f	7.054462	6.463128	7.392383	8.092092	6.570368
8.0f	6.4861917	5.8973804	7.152704	8.2018175	6.5178304
9.0f	5.4499264	6.1280284	6.034551	7.422044	5.3603487

4.4.2 Hasil Cosine Similarity







Untuk hasil rumus *Cosine Similarity*, dapat dilihat pada Tabel 9 bahwa *score* yang lebih besar daripada *threshold* yang sudah ditentukan merupakan hasil yang benar.

Untuk menentukan rumus dan *threshold* yang lebih optimal untuk dipakai, pengujian dilakukan dengan membandingkan pengukuran *cosine similarity* dengan cara membedakan hasil akhir *score* wajah yang di *scan* dengan wajah dari orang yang sama di dataset. Hasil dapat dilihat pada Tabel 10

Tabel 9. Tabel Training Cosine Similarity dengan wajah berbeda

Test	Data 1	Data 2	Data 3	Data 4	Data 5
					
7.0f	9.814593	9.720658	6.203924	9.8109865	9.080256
8.0f	10.675922	11.114479	5.316419	10.055315	9.948887
9.0f	9.399267	9.166858	6.21867	9.0559025	9.047331

Tabel 10. Tabel Training Score Cosine Similarity dengan wajah sama

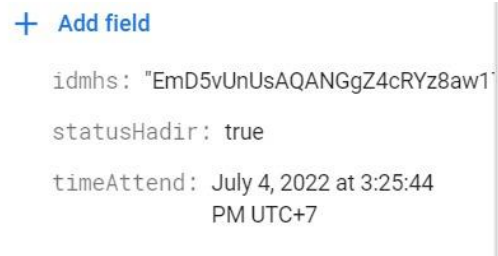
Test	Data 1	Data 2	Data 3	Data 4	Data 5
					
7.0f	7.054462	6.463128	7.392383	8.092092	6.570368
8.0f	6.4861917	5.8973804	7.152704	8.2018175	6.5178304
9.0f	5.4499264	6.1280284	6.034551	7.422044	5.3603487

4.5 Pengujian Status Kehadiran

Ketika mahasiswa berhasil absen, status kehadiran pada database akan di-*update* menjadi *true*, seperti pada Gambar 10 dan *pop-up* pada Gambar 9 akan muncul untuk memberitahukan mahasiswa berhasil dilakukan.

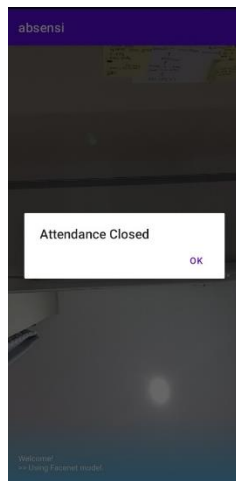


Gambar 9. Pengujian Status Kehadiran



Gambar 10. Update Status pada Database

Apabila absensi sudah tidak lagi dibuka dan/atau waktu sudah melewati batas absensi, maka akan muncul *pop-up* pada untuk memberitahukan bahwa kelas/acara sudah tidak menerima absensi lagi. Seperti yang ada pada Gambar 11.



Gambar 11. Pop-up Warning absensi tutup

4.6 Pengujian Rekap Absensi

Rekap absensi dapat dilihat pada Gambar 12. Rekap absensi berfungsi untuk mengecek status kehadiran mahasiswa dalam kelas untuk setiap pertemuan.

absensi	
Pertemuan ke - 1 ERD 15 Jun 2022	Alpa
Pertemuan ke - 2 Mapping 4 Jul 2022	Hadir

Gambar 12. Rekap Absensi

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil implementasi yang telah dilakukan maka dapat ditarik kesimpulan sebagai berikut:

- Absensi mahasiswa berhasil diambil dengan face recognition dengan mengintegrasikan struktur database yang sudah diberikan oleh Universitas Kristen Petra.
- Dari hasil analisa untuk pengenalan wajah, rumus L2Norm paling akurat ketika menggunakan *threshold* 8.0f dengan score 5.316419. Rumus *cosine similarity*

paling akurat ketika menggunakan *threshold* 0.5f dengan score 0.77162087 untuk mengenali wajah.

- Dari hasil analisa yang dilakukan, rumus L2Norm paling optimal ketika menggunakan *threshold* 8.0f dengan score terbaik 5.8973804. Rumus *cosine similarity* paling optimal ketika menggunakan *threshold* 0.5f dengan score terbaik 0.5104218.

5.2 Saran

Berdasarkan hasil Aplikasi absensi kelas berbasis Android Studio, terdapat beberapa saran:

- Program dapat dikembangkan agar lebih lengkap, seperti menambahkan laporan kehadiran mahasiswa dan peserta pada aplikasi.
- *Face recognition* dapat dikembangkan dengan menambahkan dataset muka mahasiswa agar dapat lebih akurat lagi.

6. DAFTAR REFERENSI

- [1] Asy'Ari, M. Z. (2020, November 13). *Apa itu tensorflow? 3 Hal Penting Untuk Dipahami*. Auftechnique. <https://auftechnique.com/apa-itu-tensorflow>
- [2] Bah, S. M., & Ming, F. (2020). An improved face recognition algorithm and its application in attendance management system. *An Improved Face Recognition Algorithm and Its Application in Attendance Management System*, 5, 100014. <https://doi.org/10.1016/j.array.2019.100014>
- [3] Craig, C., & Gerber, A. (2015). *Learn Android Studio : Build Android Apps Quickly and Effectively*. New York: Apress.
- [4] Google. (2019). *Cloud Firestore | Firebase Documentation*. Firebase. <https://firebase.google.com/docs/firestore>
- [5] Google. (2020). *ML Kit for Firebase*. Firebase. <https://firebase.google.com/docs/ml-kit>
- [6] Google. (2020). *Ringkasan CameraX*. Android Developers. <https://developer.android.com/training/camerax?hl=id>
- [7] Heiss, J. J. (2015, April). *The Advent of Kotlin: A Conversation with JetBrains' Andrey Breslav*. Diambil kembali dari Java: <https://www.oracle.com/technical-resources/articles/java/breslav.html>
- [8] Putra, W. S. E. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS*, 5(1). <https://doi.org/10.12962/j23373539.v5i1.15696>
- [9] Sunaryono, D., Siswanto, J., & Anggoro, R. (2019). An android based course attendance system using face recognition. *Journal of King Saud University*.