

APLIKASI ANDROID DETEKSI SUPIR MENGANTUK PADA PENGENDARA MOBIL BERBASIS *FACE* *RECOGNITION*

Stefanus Wijaya, Nemuel Daniel Pah*, Rafina Destiarti Ainul

Fakultas Teknik Universitas Surabaya, Raya Kalirungkut, Surabaya 60293

*Corresponding author: nemuelpah@staff.ubaya.ac.id

Abstract—Humans have always developed technology to make life easier. One of the technological developments made is land transportation. In the development of land transportation is also accompanied by an increase in the accident rate of drivers due to drowsiness. This drowsiness detection system is made using an Android-based smartphone where this technology is almost used in various walks of life so that it can help to provide direct warnings to drivers who are drowsy. The system uses Haar-Cascade Classifier to detect faces and eyes and utilizes the training model to determine whether the person is sleepy or not. System development using Android Studio with Java and Python programming languages. Based on system testing on 9 different faces with 120 frame samples per face, with 40 cm in distance, and an angle of 20°, the accuracy result is 85% during the day with a light intensity of 725 lux.

Keywords: drowsiness detection, android, haar-cascade classifier

Abstrak— Manusia selalu mengembangkan teknologi untuk mempermudah hidupnya. Salah satu perkembangan teknologi yang dibuat adalah transportasi darat. Dalam perkembangan transportasi darat ini juga ditemani dengan bertambahnya tingkat kecelakaan para pengemudi akibat mengantuk. Sistem pendeteksi kantuk ini dibuat dengan menggunakan *smartphone* berbasis Android dimana teknologi ini hampir digunakan diberbagai lapisan masyarakat sehingga dapat membantu untuk memberikan peringatan secara langsung pada pengemudi yang sedang mengantuk. Sistem yang dibuat menggunakan *Haar-Cascade Classifier* untuk mendeteksi wajah dan juga mata dan memanfaatkan model hasil pelatihan untuk menentukan apakah orang tersebut sedang mengantuk atau tidak. Pembuatan sistem dengan menggunakan Android Studio dengan bahasa pemrograman Java dan Python. Berdasarkan pengujian sistem pada 9 wajah berbeda dengan 120 sampel frame per wajah, jarak 40 cm, dan sudut 20° mendapatkan hasil akurasi sebesar 85% pada siang hari dengan intensitas cahaya sebesar 725 lux.

Kata kunci: deteksi kantuk, android, haar-cascade classifier

Pendahuluan

Manusia selalu melakukan pengembangan teknologi untuk mempermudah kehidupannya. Salah satu teknologi yang dikembangkan adalah bidang transportasi. Perkembangan transportasi ini sangat membantu manusia untuk meminimalisir waktu yang digunakan untuk berpergian ke tempat yang jauh. Terlepas dari semua kemudahan yang diberi, seorang pengemudi selalu dituntut untuk tetap sadar dan fokus saat mengemudikan kendaraannya. Tidak sedikit pengemudi di jalanan yang mengabaikan peraturan utama dalam berkendara. Hal ini banyak disebabkan karena faktor kelelahan yang menyebabkan pengemudi mengantuk dan terjadi kecelakaan (databoks, 2018). Selain faktor kelelahan, hal ini terjadi karena saat perjalanan yang jauh dan sendirian dalam perjalanan membuat pengemudi juga sering mengantuk (Albadawi et al., 2022).

Oleh karena permasalahan tersebut sistem pendeteksi kantuk berbasis Android ini dibuat. Pembuatan sistem ini bertujuan untuk mendesain sistem yang dapat melakukan *face recognition*. *Face Recognition* adalah pengenalan wajah dengan menggunakan metode biometrik berdasarkan foto wajah (Encyclopedia.com, n.d.). Metode yang akan digunakan adalah *haar cascade*. Metode ini mempermudah pendeteksian garis dan juga sudut pada gambar atau memilih area dengan perubahan mendadak dalam intensitas piksel (Behera, 2020). Hasil desain akan diimplementasikan pada sistem Android. Hal ini dikarenakan *smartphone* berbasis Android mudah dibawa ke mana saja dan sudah banyak digunakan tidak terkecuali oleh pengendara kendaraan (Samat, 2022). Hasil Tugas Akhir ini diharapkan akan mengurangi angka kecelakaan yang diakibatkan rasa kantuk saat mengemudi.

Metode Penelitian

Langkah-langkah yang akan dilakukan dalam pembuatan tugas akhir ini adalah sebagai berikut:

1. Pengumpulan *Dataset*
Mengumpulkan *dataset* untuk melatih model *drowsiness detection*, pelatihan model pengenalan wajah menggunakan 9.511 gambar orang tidak mengantuk dan 2.214 gambar orang mengantuk (Ghoddoosian et al., 2019), (Petrellis et al., 2021). Untuk pelatihan model mata menggunakan 47.000 gambar mata tertutup dan 50.900 gambar mata terbuka (ASGHAR, 2022).
2. Pemrograman Awal
Pemrograman awal berisi pembuatan program untuk *training data* yang akan digunakan pada sistem, serta pembuatan program pada komputer agar bisa dicoba terlebih dahulu sebelum di aplikasikan ke *smartphone*. Pada tahap digunakan program Python. Hal ini dikarenakan Python merupakan bahasa pemrograman yang populer (Runtsch, 2022).
3. Pembuatan *User Interface*
Setelah program Python jadi, akan dilanjutkan ke tahap pembuatan *user interface* aplikasi Android agar memiliki tampilan yang menarik serta para pengguna bisa dengan mudah menggunakan aplikasinya.
4. Pemrograman Aplikasi Android
Pemrograman ini ditujukan agar aplikasi Android dapat bekerja dengan baik dan bisa mengeksekusi pemrograman yang sudah dilakukan pada Python.
5. Menghubungkan Python dengan Android
Penghubungan ini bertujuan agar aplikasi saat dijalankan dapat melakukan identifikasi wajah untuk menentukan apakah penggunanya sedang mengantuk atau tidak.
6. Pengujian Sistem
Melakukan pengujian *software* dengan menggunakan video kamera Android secara *real time*. Akan dilakukan pemrograman ulang atau *training* ulang jika diperlukan.
7. Penyusunan Laporan
Pada tahap ini, hasil penelitian akan dibuat laporan.

Hasil

Pengujian aplikasi berdasarkan beberapa permasalahan yang didapat pada saat mengendarai mobil yaitu:

1. Jarak antara kamera *smartphone* dengan wajah pengguna
2. Sudut antara kamera *smartphone* dengan wajah pengguna
3. Pencahayaan optimum yang dibutuhkan sistem.
4. Penggunaan aksesoris oleh pengguna
5. Hasil pembacaan jika menggunakan wajah orang yang berbeda
6. Adanya wajah penumpang yang masuk ke dalam *frame* kamera
7. Pengujian di dalam mobil
8. Suhu dan penggunaan baterai oleh aplikasi
9. Pengujian waktu pemrosesan per *frame*

Hasil pengujian meliputi akurasi, *recall*, dan presisi pada pengujian jarak akan ditampilkan pada Tabel 1, pengujian sudut akan ditampilkan pada Tabel 2, pengujian cahaya optimum akan ditampilkan pada Tabel 3, pengujian penggunaan aksesoris akan ditampilkan pada. Hasil pengujian dapat dilihat pada Tabel 4, pengujian perbedaan wajah akan ditampilkan pada Tabel 5, dan pengujian di dalam mobil akan ditampilkan pada Tabel 7.

Tabel 1

Data Pengukuran Akurasi, Recall, dan Presisi pada tiap Satuan Jarak

	Akurasi	Recall	Presisi
30 cm	98%	100%	97%
40 cm	98%	100%	97%
50 cm	83%	100%	80%

Tabel 2

Data Pengukuran Akurasi, Recall, dan Presisi pada tiap Satuan Sudut

	Akurasi	Recall	Presisi
10°	100%	100%	100%
20°	89%	85%	98%
30°	71%	100%	69%

Tabel 3

Data Pengukuran Akurasi, Recall, dan Presisi pada tiap Satuan Cahaya

	Akurasi	Recall	Presisi
Pagi (570 lux)	100%	100%	100%
Siang (730 lux)	89%	85%	98%
Sore (70 lux)	73%	100%	71%
Malam (0 lux)	67%	100%	67%

Tabel 4

Data Pengukuran Akurasi, Recall, dan Presisi pada tiap Penggunaan Aksesoris

	Akurasi	Recall	Presisi
Topi	71%	100%	69%
Kacamata	86%	100%	82%
Hijab	67%	100%	67%

Tabel 5

Data Pengukuran Akurasi, Recall, dan Presisi Beda Wajah

	Akurasi	Recall	Presisi
Orang ke-1	97%	99%	97%
Orang ke-2	100%	100%	100%
Orang ke-3	67%	55%	92%
Orang ke-4	98%	98%	100%
Orang ke-5	77%	69%	95%
Orang ke-6	89%	99%	86%
Orang ke-7	71%	100%	69%
Orang ke-8	68%	100%	68%
Orang ke-9	98%	99%	98%

Tabel 6

Data Pengujian Wajah Penumpang Masuk pada Frame Kamera

	Pengemudi	Penumpang
Wajah sejajar pengemudi (50%)	40 frame	0 frame
Wajah sejajar pengemudi (75%)	40 frame	0 frame
Wajah sejajar pengemudi (100%)	22 frame	18 frame
Wajah di belakang pengemudi (50%)	33 frame	7 frame
Wajah di belakang pengemudi (75%)	2 frame	38 frame
Wajah di belakang pengemudi (100%)	0 frame	40 frame

Tabel 7

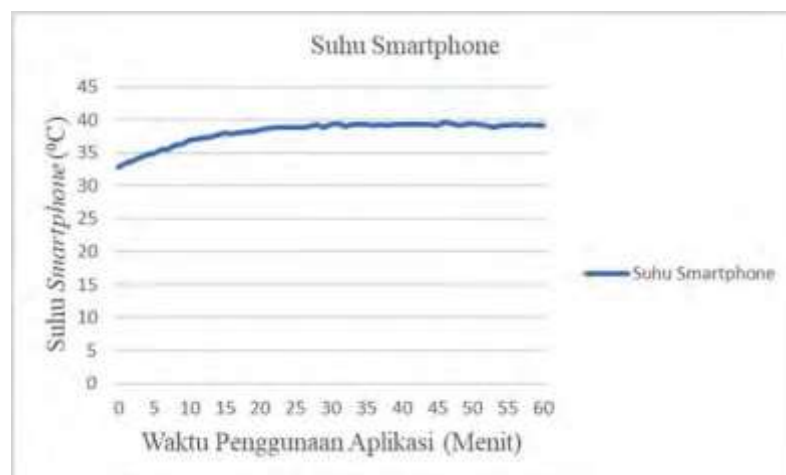
Data Pengukuran Akurasi, Recall, dan Presisi Pada Mobil

	Akurasi	Recall	Presisi
Siang	95%	100%	93%
Malam	67%	100%	67%

Pengujian suhu dan baterai akan ditampilkan pada Gambar 1 dan Gambar 2.



Gambar 1. Grafik penggunaan baterai saat menjalankan aplikasi.



Gambar 2. Grafik suhu HP saat menjalankan aplikasi.

Pengujian waktu pemrosesan sistem dengan menggunakan *facenet* ditunjukkan pada Tabel 8.

Tabel 8

Data Pengukuran Waktu Pemrosesan saat Menggunakan *Facenet* dan Tidak

Frame	Waktu Pemrosesan Tanpa <i>FaceNet</i> (s)	Waktu Pemrosesan Dengan <i>FaceNet</i> (s)
1	0,4	34,2
2	0,3	40,0
3	0,2	38,0
4	0,2	43,2
5	0,2	39,2
6	0,2	42,2
7	0,2	36,8
8	0,2	39,3
9	0,2	40,3
10	0,2	35,3

Diskusi

Dari hasil penelitian yang sudah dilakukan didapatkan bahwa jarak optimum adalah 30 cm. Namun untuk penggunaan pada mobil jarak 30 cm akan terlalu dekat dengan supir yang membuat tidak nyaman membuat jarak 40 cm sebagai jarak optimum dan ideal yang dapat digunakan saat berkendara. Pengujian berdasarkan sudut menggunakan referensi jarak 40 cm dan menghasilkan sudut optimum pada 10° . Pada pengujian sudut 30° banyak objek wajah yang tidak terdeteksi sehingga akurasi sistem menurun. Pada pengujian berikutnya akan digunakan sudut 20° sebagai referensi sudut, hal ini disebabkan sudut 10° akan menghalangi pandangan dari supir.

Pengujian pada perbedaan lux cahaya menggunakan jarak 40 cm dan sudut 20° sebagai referensi pengujian. Dari hasil pengujian didapatkan bahwa pengujian pada pagi hari merupakan kondisi cahaya optimum agar sistem dapat bekerja dengan baik. Pengukuran malam hari mendapatkan hasil terburuk karena sistem tidak dapat menangkap gambar apapun karena kondisi cahaya yang terlalu gelap. Penggunaan aksesoris seperti topi, kacamata, dan hijab akan berpengaruh dengan hasil klasifikasi yang dilakukan oleh sistem. Hal ini dikarenakan ada bagian dari wajah yang terhalang oleh aksesoris tersebut.

Masuknya wajah penumpang sejajar dengan supir bisa digunakan jika tidak masuk 100% pada frame kamera. Saat wajah penumpang masuk dari belakang supir lebih dari 50%, sistem tidak dapat digunakan karena wajah penumpang lebih sering digunakan daripada wajah supir. Pengujian dengan menggunakan perbedaan wajah dilakukan dengan 9 orang. Pada pengujian ini menghasilkan nilai akurasi rata – rata 85%. Pada pengujian dalam mobil mendapat akurasi yang tinggi pada 95% dengan jarak referensi 40 cm dan sudut 20° pada siang hari.

Penggunaan baterai oleh aplikasi sangat boros karena menghabiskan 35% setiap 1 jam penggunaan. Sedangkan suhu yang didapat cukup memuaskan karena tidak lebih dari 40°C . Penggunaan *facenet* tidak disarankan karena hasil pengolahan per *frame* oleh sistem memakan waktu kisaran 39 detik, sedangkan jika sistem tidak menggunakan *facenet* hanya memakan waktu kisaran 0,3 detik per *frame*-nya.

Kesimpulan

Dari hasil coba yang telah dilakukan pada sistem, didapatkan beberapa kesimpulan. Jarak dan sudut optimal sistem adalah 30 cm dan 10° antara wajah dengan kamera HP. Kondisi pencahayaan optimum ada pada pagi hari (570 lux) sedangkan pencahayaan minimum pada sore hari (70 lux). Penggunaan aksesoris mempengaruhi akurasi dari sistem, penggunaan hijab dan topi akan menutupi sebagian dari wajah sehingga akurasi menurun. Wajah penumpang yang masuk dari belakang supir lebih dari 75% lebih banyak digunakan daripada wajah supir sehingga sistem tidak dapat digunakan. Perbedaan kontras cahaya yang mengenai wajah akan mempengaruhi hasil klasifikasi sistem. Penggunaan baterai yang boros akibat penggunaan berat yang dilakukan secara terus – menerus. Suhu masih terjaga dibawah 40°C sehingga aman digunakan dalam jangka waktu yang lama. Adanya waktu yang diperlukan untuk memulai program dari Python yang tersemat pada aplikasi. *FaceNet* tidak dapat digunakan karena waktu pemrosesan yang sangat lama.

Pustaka Acuan

- Albadawi, Y., Takruri, M., & Awad, M. (2022). A Review of Recent Developments in Driver Drowsiness Detection Systems. *Sensors*, 22(5), 1–41. <https://doi.org/10.3390/s22052069>
- ASGHAR, M. H. (2022). *OACE - Open and Close Eyes Dataset*. Kaggle. <https://www.kaggle.com/datasets/muhammadhananasghar/oace-open-and-close-eyes-dataset>
- Behera, G. S. (2020). *Face Detection with Haar Cascade*. Towards Data Science. <https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08>
- Databoks. (2018). *Sepanjang 2017 Terjadi 98 Ribu Kali Kecelakaan Lalu Lintas*. Databoks. <https://databoks.katadata.co.id/datapublish/2018/09/21/sepanjang-2017-terjadi-98-ribu-kali-kecelakaan-lalu-lintas>

CALYPTRA VOI.12 (1) November 2023

- Encyclopedia.com. (n.d.). *Facial Recognition*. Retrieved June 22, 2022, from <https://www.encyclopedia.com/science/encyclopedias-almanacs-transcripts-and-maps/facial-recognition>
- Ghoddosian, R., Galib, M., & Athitsos, V. (2019). A realistic dataset and baseline temporal model for early drowsiness detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2019-June*, 178–187. <https://doi.org/10.1109/CVPRW.2019.00027>
- Petrellis, N., Zogas, S., Christakos, P., Mousoulitis, P., Keramidas, G., Voros, N., & Antonopoulos, C. (2021). Software Acceleration of the Deformable Shape Tracking Application: How to eliminate the Eigen Library Overhead. *ACM International Conference Proceeding Series, November*, 51–57. <https://doi.org/10.1145/3501774.3501782>
- Runtsch, R. (2022). *The Most Popular Computer Programming Languages of 2022*. Medium. <https://rruntsch.medium.com/most-popular-computer-programming-languages-in-2022-23e0f523391>
- Samat, S. (2022). *Living in a multi-device world with Android*. <https://blog.google/products/android/io22-multideviceworld/>