

Deep Learning Untuk Mendeteksi Kelelahan Pekerja

Dadang Sujana¹, Sutresna Juhara², Monita Rahayu³, Mahmudin⁴, Maulana Fany Kurnia Pratama⁵

Universitas Islam Syekh-Yusuf Jl. Maulana Yusuf No.10 Tangerang Banten 15118, Indonesia^{1,2,3,4,5}
dadangsujana@unis.ac.id¹, sjuhara@unis.ac.id², monita@unis.ac.id³, mahmudin@unis.ac.id⁴

Abstrak— Salah satu komponen penyebab kecelakaan kerja adalah human error. Kecelakaan kerja dapat terjadi ketika pekerja mengalami kelelahan baik secara fisik maupun mental. Indikator kelelahan yang digunakan pada penelitian ini adalah rasa kantuk. Mengantuk saat melakukan pekerjaan dapat menyebabkan menurunnya konsentrasi dalam bekerja yang tentunya akan berpengaruh pada produktivitas kerja dan dapat berakibat menimbulkan kecelakaan kerja. Tujuan penelitian ini adalah meminimalisir kecelakaan yang terjadi pada lingkungan kerja terutama yang disebabkan oleh rasa kantuk sehingga bisa mencapai zero accident dengan melibatkan perkembangan teknologi. Metode yang digunakan adalah membuat rancangan sistem dengan bahasa python, algoritma CNN dan implementasi sistem menggunakan Raspberry Pi yang bisa mendeteksi mata bagi para pekerja atau karyawan untuk mengetahui rasa kantuk yang dapat dilihat dari seberapa sering menutup mata yang terdeteksi oleh kamera kemudian sistem akan bekerja dan mengirimkan sinyal alarm sebagai pengingat bagi pekerja untuk tetap fokus dalam bekerja sehingga tidak menghambat kelancaran produksi. Hasilnya adalah alat pendeteksi kelelahan (kantuk) dengan tahap scan wajah baik secara langsung menggunakan kamera maupun menggunakan foto, shape predictor, eye landmark yang sudah diuji coba menggunakan 15 sampel pekerja dengan tingkat akurasi alat sebesar 93%.

Kata Kunci — alat pendeteksi, algoritma, kantuk, kelelahan

Abstract— One of the components that cause work accidents is human error. Work accidents can occur when workers experience fatigue both physically and mentally. The fatigue indicator used in this study is drowsiness. Drowsiness while doing work can cause a decrease in concentration at work which of course will affect work productivity and can result in work accidents. The purpose of this study is to minimize accidents that occur in the work environment, especially those caused by drowsiness so that it can achieve zero accidents by involving technological developments. The method used is to design a system with python language, CNN algorithm and system implementation using Raspberry Pi which can detect eyes for workers or employees to determine sleepiness which can be seen from how often closing the eyes detected by the camera then the system will work and send alarm signal as a reminder for workers to stay focused on work so as not to hamper the smooth production. The result is a fatigue detector (drowsiness) with a facial scan stage either directly using the camera or using photos, shape predictors, eye landmarks that have been tested using 15 samples of workers with an accuracy rate of 93%.

Keywords — detection device, algorithm, drowsiness, fatigue

I. PENDAHULUAN

Organisasi Buruh Internasional atau *International Labour Organization* (ILO) melaporkan bahwa setiap 15 detik, seorang pekerja meninggal akibat kecelakaan yang berhubungan dengan pekerjaan atau penyakit di seluruh dunia. Setiap 15 detik, sebanyak 153 pekerja mengalami kecelakaan yang berhubungan dengan pekerjaan. Di lokasi konstruksi di seluruh

dunia, sedikitnya sebanyak 60.000 kecelakaan fatal terjadi setiap tahun, yang mana berarti terjadi satu kecelakaan fatal setiap 10 menitnya (Construction, 2021).

Salah satu penyebab kecelakaan kerja adalah *human error*. *Human error* terjadi karena pekerja mengalami kelelahan baik kelelahan fisiologis maupun psikologis. Kelelahan terjadi menyebabkan rasa kantuk yang pada akhirnya

akan menyebabkan menurunnya konsentrasi dalam bekerja (Amirullah et al., 2019).

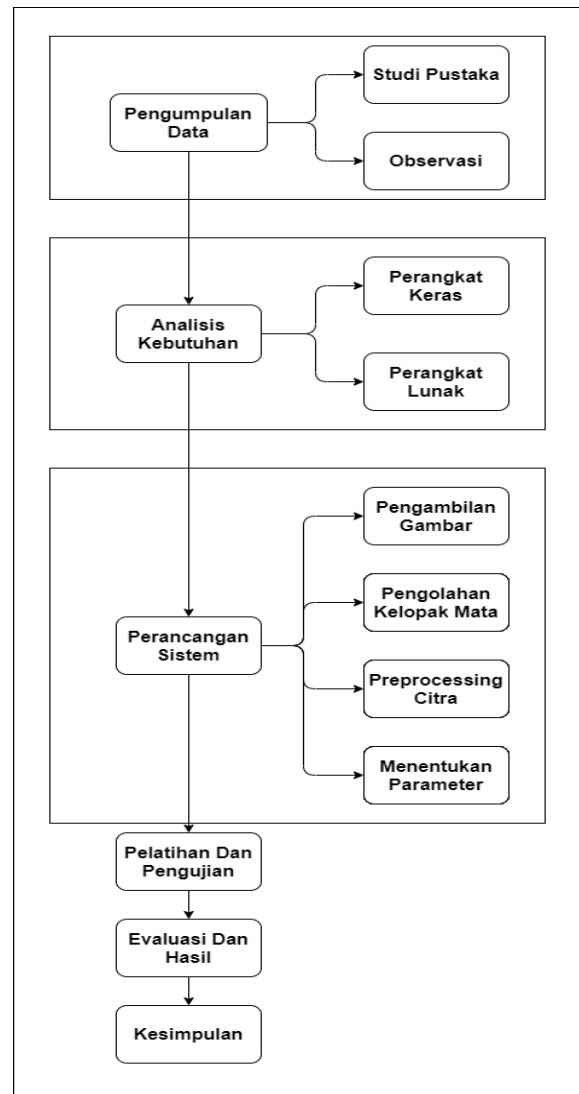
Untuk mengurangi kecelakaan kerja akibat rasa kantuk tersebut, perlu adanya sistem yang dapat mendeteksi rasa kantuk pada pekerja. Target dari sistem ini adalah *zero accident* dari kecelakaan yang disebabkan rasa kantuk. Sistem yang dibuat akan mendeteksi rasa kantuk dari seberapa sering mata menutup yang terdeteksi kamera kemudian sistem akan mengirimkan sinyal *alarm* sebagai pengingat bagi pekerja. Sistem dibuat agar pekerja tetap fokus dan tidak menghambat kelancaran produksi (Efendi et al., 2020).

Deteksi kantuk dapat dilakukan dengan sistem *computer vision* secara otomatis dengan menggunakan kamera *web real-time* yang nanti akan di hubungkan dengan *alarm* deteksi yang berbunyi apabila terlihat mengalami kantuk dan pengujian dapat didasarkan pada gaya wajah dan usia (Zein, 2018).

Identifikasi kelelahan dengan model pengenalan mata kantuk dapat digunakan untuk objek yang memiliki kecepatan dan ketepatan yang tinggi dalam hal ini akurasi alat yang dibuat harus tinggi (Imanuddin et al., 2019). Untuk itu perlu dilakukan pendekatan *Artificial Intelligence* (AI) berbentuk *learning, reasoning, serta self-correction* yang didalamnya terdapat proses penalaran berpikir dan tindakan (Ratnawati, D and Liliana, D and Regasari, R and Muflikhah, 2016). Dalam AI diperlukan perangkat lunak yang akan dijadikan sebagai otak dari alat yang dibuat (Franceschetti, 2018).

II. METODE PENELITIAN

Dalam metode penelitian dimulai dari pengumpulan data, proses analisis kebutuhan dan perancangan sistem untuk mengetahui tahapan implementasi dalam perancangan, sehingga dapat membentuk suatu kerangka dari proses perancangan sistem. Beberapa langkah dilakukan untuk menentukan kebutuhan sistem yang akan dibuat dan untuk mengatasi masalah penggunaan algoritma *convolutional neural network* (CNN) dalam mengenali kelopak mata. Berikut alur penelitian:



Gambar 1. Alur Penelitian

Setelah pengumpulan data dilakukan selanjutnya dilakukan analisis kebutuhan perangkat keras dan perangkat lunak. Kebutuhan perangkat keras dapat dilihat pada tabel 1

Tabel 1. Kebutuhan Perangkat Keras

No	Nama Perangkat	Keterangan
1	Raspberry Pi3B	Perangkat ini digunakan untuk perangkat utama dalam pembuatan program dan algoritma dari sistem deteksi kelopak mata.
2	Webcam	Perangkat ini digunakan sebagai alat untuk menangkap gambar pengemudi.
3	Buzzer	Alat ini di gunakan sebagai alarm bagi pengemudi yang terdeteksi mengantuk.
4	LCD TFT 1.8 inch 160x128	Layar untuk menampilkan keterangan verifikasi

No	Nama Perangkat	Keterangan
		pengenalan wajah berupa dikenal atau tidak dikenal.
5	Adaptor	Sebagai arus listrik untuk menjalankan <i>raspberry</i> .

Untuk perangkat keras menggunakan *raspberry pi* yang berfungsi melakukan *learning to program* yang memberikan pelajaran untuk melakukan percobaan dengan komputer yang didalamnya terdapat *project platform* (Irsan et al., 2019). Dan *Raspberry Pi* memiliki sistem operasi berbasis linux tidak seperti komputer biasa yang kita gunakan (Royhan, 2020). *Raspberry pi* terdapat beberapa jenis dengan karakteristik dan fungsi yang beragam (Cicolani, 2021).

Untuk kebutuhan perangkat lunak dapat dilihat pada tabel 2.

Tabel 2. Kebutuhan Perangkat Lunak

No	Nama Perangkat	Keterangan
1	Software VNC	Sebagai platform Bahasa pemrograman <i>python</i> dalam pembuatan sistem.
2	<i>Library OpenCV</i>	Digunakan sebagai pengolahan <i>image</i> serta video.

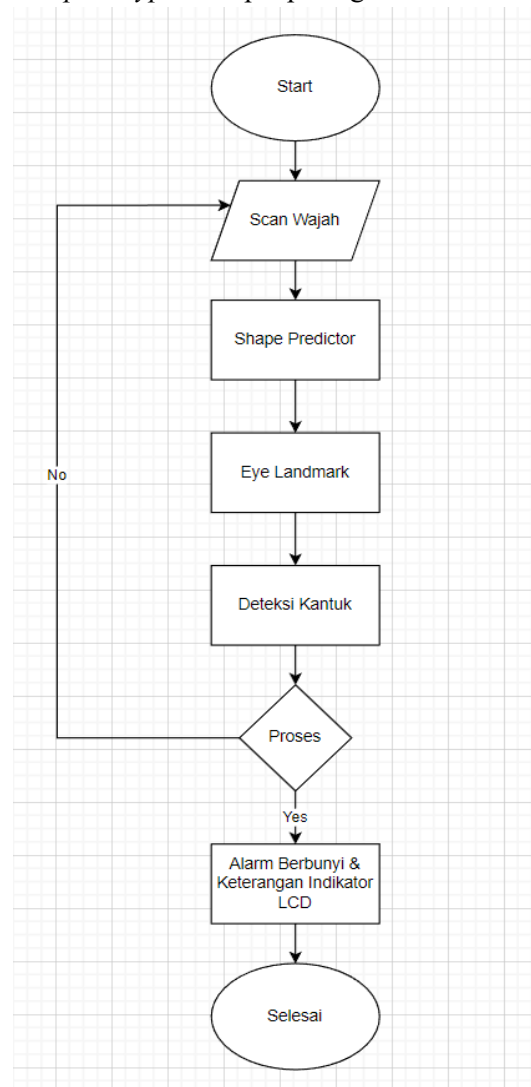
Untuk perangkat lunak menggunakan *Virtual Network Computing* (VNC) yang merupakan aplikasi *remote-control* yang membolehkan buat mengendalikan pc lain lewat koneksi VNC. VNC *Client* serta *Server* butuh terdapatnya internet yang menghubungkan antara kedua fitur ini dalam satu jaringan komunikasi yang sama (Khan et al., 2018).

Selain itu juga dibutuhkan *Library Open CV* adalah salah satunya menyediakan kemudahan dalam infrastruktur *computer vision* yang membantu untuk membangun aplikasi yang cukup canggih dan cepat (Zufar & Setiyono, 2016)

Perancangan Sistem

Sebelum dilakukan pembuatan sistem, diperlukan sebuah perancangan sistem untuk menjadi gambaran seperti apa alur sistem yang akan berjalan nanti, dari sebuah rancangan ini akan terlihat pada bagian mana algoritma *Convolutional Neural Network* (CNN) yang akan diterapkan pada pembuatan sistem. Pada

pengumpulan data telah dijelaskan data yang akan diolah dimana data yang sudah di kumpulkan yaitu berupa foto wajah yang nanti akan digunakan sebagai dataset. Kemudian dataset yang sudah ada tersebut akan diproses pada *Image Processing* dengan menggunakan algoritma *Convolutional Neural Network* (CNN), setelah selesai pemrosesan maka akan menghasilkan output atau keluaran yang berupa hasil identifikasi kelopak mata. Rancangan sistem *prototype* terdapat pada gambar 2.

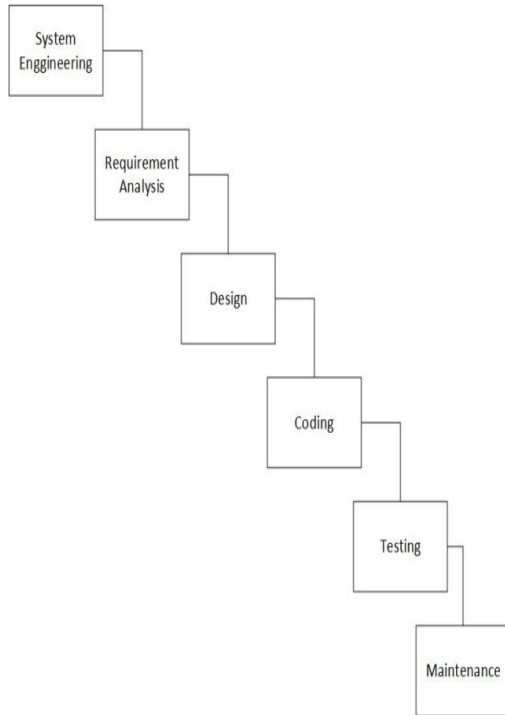


Gambar 2. Rancangan Sistem *Prototype*

III. HASIL DAN PEMBAHASAN Metode Pengembangan Sistem

Prototype pengembangan sistem didasari oleh konsep model kerja, dimana pengembang bisa berhubungan sepanjang proses pembangunan sistem. Proses ini terstruktur serta mempunyai

tahapan-tahapan wajib seperti yang terlihat pada gambar 3.



Gambar 3. Model *Prototype*

Perancangan Perangkat Lunak

Perancangan perangkat lunak sistem berlangsung pada program Python di Raspberry Pi, yang diakses melalui aplikasi VNC Viewer untuk me-remote desktop Raspberry Pi. Sistem dibentuk dengan menggunakan aplikasi Python versi 3.9.2, OpenCV versi 4.9.5 dan opencv-contrib, dlib beserta Numpy sebagai modul yang mendukung untuk sistem komputasi ilmiah.

Face detector menggunakan Dlib dan Open CV

Proses awal tahapan pembuatan sistem ini adalah *face detector* atau mendeteksi wajah sebagai objek dasar. Mendeteksi wajah dalam sistem ini menggunakan detektor objek HOG + Linear SVM pada dlib yang telah dilatih sebelumnya khusus untuk tugas deteksi wajah. Hasil akhir dalam deteksi wajah ini kita mendapatkan kotak pembatas wajah (yaitu, koordinat (x, y) dari wajah).

Pustaka dlib menyediakan dua fungsi yang dapat digunakan untuk deteksi wajah:

1. HOG + SVM Linier: dlib.
get_frontal_face_detector ()
2. MMOD : dlib.
cnn_face_detection_model_v1 (modelPath)

Pada sistem ini menggunakan detektor objek HOG + Linear SVM, dimana kita memanggil function get_frontal_face_detector (). Kemudian objek akan melalui proses *resize* untuk merubah ukuran gambar menjadi 450 x 450 dan objek akan di konversi BGR to RGB

Proses *Shape Predictor*

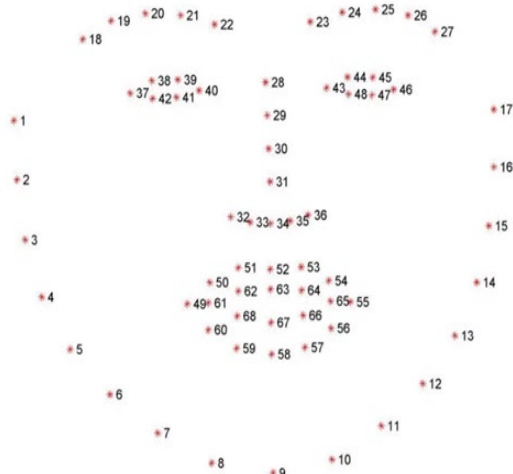
Prediktor bentuk digunakan untuk memprediksi kunci (x, y) koordinat dari bentuk yang diberikan. Pustaka prediktor bentuk yang paling umum dan terkenal adalah prediktor landmark wajah menggunakan pustaka dlib yang digunakan untuk melokalisasi struktur wajah individu, antara lain seperti mata, alis, hidung, bibir/mulut, dan garis rahang.

Untuk dapat memperkirakan prediktor bentuk menggunakan serangkaian algoritma antara lain :

1. Memeriksa serangkaian intensitas piksel input yang jarang (yaitu, "fitur" ke model input)
2. Melewati fitur ke dalam *Ensemble of Regression Trees (ERT)*
3. Menyempurnakan lokasi yang diprediksi untuk meningkatkan akurasi melalui serangkaian *regression*

Kumpulan data iBug (*Intelligent behaviour Understanding Group*) 300-W

Untuk melatih prediktor bentuk menggunakan dlib yang dibutuhkan dalam penelitian ini yaitu anotasi dari titik koordinat mata, akan menggunakan set data iBUG 300-W tetapi dengan sedikit perubahan. Tujuan iBUG-300W adalah untuk melatih prediktor bentuk yang mampu melokalisasi setiap struktur wajah individu, termasuk mata, alis, hidung, mulut, dan garis rahang. Dataset itu sendiri terdiri dari 68 pasang nilai integer pada total 7.764 gambar, nilai-nilai ini adalah (x, y) yaitu koordinat dari struktur wajah. Koordinat struktur wajah terdapat pada gambar 4.



Gambar 4. Koordinat Struktur Wajah

Berdasarkan gambar 4, dapat memperoleh peta koordinat mana yang menjadi struktur wajah, antara lain :

1. Mulut dapat diakses melalui poin [49, 68].
2. Alis kanan melalui titik [18, 22].
3. Alis kiri melalui poin [23, 27].
4. Mata kanan menggunakan [37, 42].
5. Mata kiri dengan [43, 48].
6. Hidung menggunakan [28 , 36].
7. Dan rahang melalui [1, 17].

Data tersebut yang dijadikan sebagai dataset untuk prediktor bentuk wajah yang terdapat pada *shape_predictor_68_face_landmarks.dat*.

Penerapan *Eye Landmark*

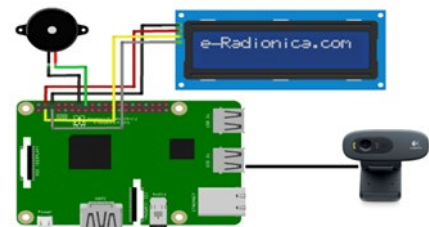
Setelah memiliki *shape predictor*, pada tahap ini akan dilakukan penerapan yang dimana akan menggunakan pustaka *dlib* untuk prediktor khusus bagian mata dan menggunakan sumber video dari usb kamera pada raspberry. *Eye landmark* ini yang menjadi prediktor awal klasifikasi kelelahan yang teridentifikasi dari rasa kantuk berdasarkan kerapatan mata pada wajah.

Sistem Kerja Alat

Raspberry Pi untuk mengimplementasikan sistem yang sudah dibuat, dengan menggunakan modul usb kamera, modul lcd 16x2, buzzer, serta perangkat lunak *vnc viewer* pada laptop sebagai monitor untuk menampilkan hasil output.

Raspberry pi merupakan selaku otak dari sistem ataupun selaku pusatnya, *usb* kamera

disini memakai kamera yang *support* dengan *raspberry pi* yang berperan selaku masukan suatu objek yang hendak dideteksi setelah itu hendak diproses oleh *mikroprocessor* sehingga hendak menciptakan keluaran ataupun *output* hasil dari deteksi kantuk pada *monitor*, *buzzer* serta *lcd 16x2* yang hendak berjalan cocok dengan keadaan.



Gambar 5. Rangkaian Alat

Rangkaian alat yang dibuat, dimana *raspberry pi* diatas tersedia pin *digital input* dan *output* sebagai penghubung *buzzer* dan *lcd 16x2*. Sedangkan *usb* kamera menggunakan port *usb* sebagai penghubung antara kamera dan *raspberry*.

Pengujian Alat

Setelah semua rangkaian telah dibuat dan sudah memasukan program ke dalam *raspberry pi* selanjutnya adalah pengujian alat untuk mengetahui sejauh mana alat ini dapat bekerja. Hasil pengujian *eye landmark* terdapat pada tabel 3.

Tabel 3. Pengujian *Eye Landmark* berdasarkan Jarak

No	Sampel	Jarak	Hasil
1	A	25 cm	Teridentifikasi normal
		30 cm	Teridentifikasi normal
		40 cm	Teridentifikasi normal
		50 cm	Teridentifikasi normal
		60 cm	Teridentifikasi normal
		25 cm	Teridentifikasi kantuk (terjadi)

No	Sampel	Jarak	Hasil
2	B		kelelahan)
		30 cm	Teridentifikasi kantung
		40 cm	Teridentifikasi kantung
		50 cm	Teridentifikasi kantung
		60 cm	Teridentifikasi kantung

Dari tabel 3, hasil uji coba terhadap klasifikasi kelelahan saat bekerja yang diindikasikan dengan rasa kantuk dengan berbagai ukuran jarak tertentu, dengan *true* label sebagai pengenalan yang harus ditampilkan pada monitor dimana *true label* ini merupakan kondisi apakah mata dalam kondisi normal atau kantuk, Adapun beberapa hasil uji coba terhadap A dengan jarak 25cm sampai 60cm teridentifikasi sebagai kondisi normal dan B dengan jarak 25cm sampai 60cm teridentifikasi sebagai kondisi kantuk. Pengambilan testing ini dilakukan dengan kondisi kurang cahaya pada malam hari dan pada siang hari yang cukup cahaya.

Selanjutnya dilakukan pengujian terhadap posisi kepala dan penggunaan aksesoris dapat dilihat pada tabel 4.

Tabel 4. Pengujian *Eye Landmark* berdasarkan Kemiringan dan Penggunaan Aksesoris

No	Kondisi	Hasil
1	Kemiringan Kanan 30 ⁰	Terdeteksi Kantuk
2	Kemiringan Kiri 30 ⁰	Tidak terdeteksi
3	Penggunaan Aksesoris kacamata	Tidak terdeteksi

Untuk pengujian kemiringan kepala baik menghadap kanan maupun kiri dalam ruangan pada jarak 30 cm terdeteksi sangat baik, tetapi untuk pengujian penggunaan aksesoris berupa kacamata tidak dapat terdeteksi pada alat.

Selanjutnya adalah pengujian menggunakan foto atau gambar 2 dimensi yang dapat dilihat pada tabel 5.

Tabel 5. Pengujian *Eye Landmark* Menggunakan Foto

No	Sampel	Hasil
1	A	Terdeteksi Kantuk
2	B	Deteksi Normal

Pengujian menggunakan foto dari handphone menunjukkan bahwa alat dapat mengidentifikasi.

$$Accuracy = \frac{(10) + (3 - 1) + 2}{15} \times 100\%$$

Dari 15 kali uji sampel dapat dihitung tingkat akurasi sebagai berikut:

$$Accuracy = \frac{14}{15} \times 100\% = 93\%$$

Accuracy pada algoritma *Histogram Oriented Gradient + Support Vector machine*

IV. KESIMPULAN

1. Pengembangan *prototype* dengan melakukan perancangan perangkat lunak sistem dengan program *python* versi 3.9.2 di *Raspberry Pi*, yang diakses melalui aplikasi *VNC Viewer* untuk *me-remote desktop Raspberry Pi*
2. Nilai akurasi dari testing sebesar 93%, maka dari itu dengan penerapan algoritma telah mencapai hasil yang cukup baik dan akurat.
3. Dalam pengujian dengan kemiringan 30 derajat dan aksesoris seperti kacamata dan sejenisnya yang dilakukan di dalam ruangan hasilnya adalah sebagai berikut:
 - a. Jarak 30cm kemiringan ke kanan = Terdeteksi
 - b. Jarak 30cm kemiringan ke kiri = Terdeteksi
 - c. Jarak 30cm menggunakan aksesoris = Tidak Terdeteksi

Dari hasil diatas menunjukkan bahwa untuk mendeteksi dalam keadaan kemiringan kepala sudah cukup akurat dan belum dapat mendeteksi menggunakan aksesoris.

DAFTAR PUSTAKA

- [1] Amirullah, M., Kusuma, H., & Tasripan, T. (2019). Sistem Peringatan Dini Menggunakan Deteksi Kemiringan Kepala pada Pengemudi Kendaraan Bermotor yang Mengantuk. *Jurnal Teknik ITS*, 7(2). <https://doi.org/10.12962/j23373539.v7i2.31011>.
- [2] Cicolani, J. (2021). Beginning Robotics with Raspberry Pi and Arduino. In *Beginning Robotics with Raspberry Pi and Arduino*. Apress. <https://doi.org/10.1007/978-1-4842-6891-9>
- [3] Efendi, Y., Putri, A. N., Rahmaddeni, & Imardi, S. (2020). Prototype Alarm Deteksi Mata Kantuk Menggunakan. *Journal Of Information System And Informatics Engineering*, 4(2), 77–83.
- [4] Franceschetti, D. R. (2018). *Principles of Robotics & Artificial Intelligence*. Grey House
- [5] Imanuddin, I., Alhadi, F., Oktafian, R., & Ihsan, A. (2019). Deteksi Mata Mengantuk pada Pengemudi Mobil Menggunakan Metode Viola Jones. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 18(2), 321–329. <https://doi.org/10.30812/matrik.v18i2.389>
- [6] Irsan, M., Stephen, E. B., & Supriyono, A. (2019). Rancang Bangun Monitoring Ruang CCTV Server Menggunakan Raspberry Pi Di Sekolah Pelita Harapan Lippo Village. II(2), 23–29.
- [7] Khan, S., Rahmani, H., Shah, S. A. A., & Bennamoun, M. (2018). A Guide to Convolutional Neural Networks for Computer Vision. In *Synthesis Lectures on Computer Vision* (Vol. 8, Issue 1). <https://doi.org/10.2200/s00822ed1v01y201712cov015>
- [8] Ratnawati, D and Liliana, D and Regasari, R and Muflikhah, L. (2017). Modul Bahan Ajar Kecerdasan Buatan. In *Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya*.
- [9] Royhan, M. (2020). Perancangan Peringatan dan Monitoring di Boiler Generator PLTU Terintegrasi Dengan Whatsapp , Berbasis Arduino dan Raspberry. *Jurnal Teknik Informatika Unis*, 8(1), 79–89.
- [10] Zein, A. (2018). Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON. *Sainstech: Jurnal Penelitian Dan Pengkajian Sains Dan Teknologi*, 28(2), 22–26. <https://doi.org/10.37277/stch.v28i2.238>
- [11] Zufar, M., & Setiyono, B. (2016). Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-Time. *Jurnal Sains Dan Seni ITS*, 5(2), 128862. <https://doi.org/10.12962/j23373520.v5i2.18854>.