



Rancang Bangun Pendekripsi Kantuk Pada Pengendara Roda Empat Berbasis NodeMCU ESP8266

Dina Fajriah¹, Ari Syaripudin^{2*}

^{1,2}Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang, Tangerang Selatan, Indonesia

Email: ¹dinafajriah@gmail.com, ^{2*}dosen00671@unpam.ac.id

(* : coresponding author)

Abstrak – Jumlah kecelakaan yang terjadi di dunia mengalami peningkatan. Populasi yang meningkat serta adanya jumlah kendaraan yang meningkat membuat parahnya kecelakaan terjadi hampir setiap hari. Salah satu penyebab terjadinya kecelakaan lalu lintas adalah mengalami rasa kantuk atau bisa disebut *microsleep*. Untuk menghindari dan meminimalisir kasus kecelakaan lalu lintas pada pengemudi roda empat, maka dibutuhkan sistem dengan sebuah alat untuk memberi peringatan pada pengemudi roda empat saat mengalami rasa kantuk atau *microsleep* secara *real time* melalui kedipan mata pengemudi. Sistem ini berupa komponen yang terdiri dari mikrokontroler NodeMCU, *sensor infrared* yang disematkan pada sebuah kacamata, dan *buzzer* sebagai alarm. Sistem ini bekerja dengan mendekripsi kelopak pengemudi saat kondisi kelopak mata tertutup, *sensor infrared* akan memberi sinyal berupa lampu LED yang menyala, kemudian memberi sinyal ke *buzzer* yang mengeluarkan suara untuk menyadarkan pengendara yang tertidur. Teknik pengujian ini menggunakan uji *blackbox* dan *whitebox*, dengan melakukan semua kemungkinan yang terjadi dan dilakukan secara berulang-ulang dan mengungkap kesalahan implementasi seperti manajemen kunci yang buruk dengan menganalisis cara kerja internal dan struktur perangkat lunak. Pengujian ini dilakukan dengan menggabungkan semua komponen, yakni, NodeMCU, *sensor infrared* FC-51, sinyal *buzzer*, sehingga didapatkan nilai persentase sebesar 86,6% keberhasilan dalam mendekripsi kantuk pada pengendara.

Kata Kunci: Kantuk; NodeMCU ESP8266; Sensor Inframerah; Pengendara; Pengemudi

Abstract – *The number of accidents that occur in the world has increased. The increasing population and the increasing number of vehicles mean that serious accidents occur almost every day. One of the causes of traffic accidents is experiencing drowsiness or what is known as microsleep. To avoid and minimize traffic accident cases for four-wheeled drivers, a system with a tool is needed to warn four-wheeled drivers when they experience drowsiness or microsleep in real time through the driver's blinking. This system is a component consisting of a NodeMCU microcontroller, an infrared sensor embedded in a pair of glasses, and a buzzer as an alarm. This system works by detecting the driver's eyelids when the eyelids are closed, the infrared sensor will give a signal in the form of an LED light that is on, then gives a signal to the buzzer which makes a sound to wake up the sleeping driver. This testing technique uses blackbox and whitebox tests, by doing all the possibilities that occur and is done repeatedly and uncovering implementation errors such as bad key management by analyzing the internal workings and structure of the software. This test was carried out by combining all the components, namely, NodeMCU, FC-51 infrared sensor, buzzer, so that a percentage value of 86.6% was obtained in detecting drowsiness in motorists.*

Keywords: Drowsiness; NodeMCU ESP8266; Infrared Sensors; Rider, Driver

1. PENDAHULUAN

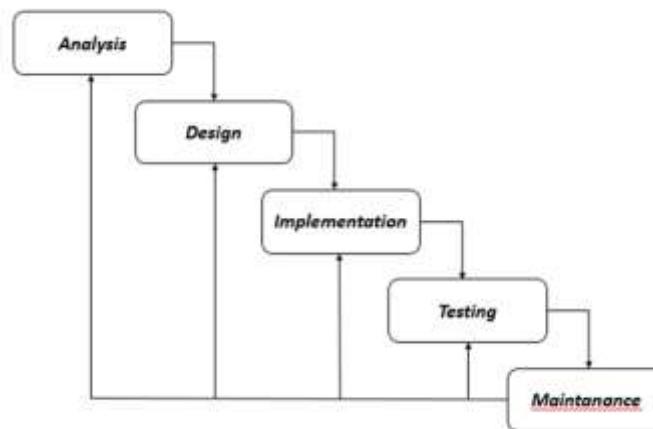
Kecelakaan lalu lintas merupakan kejadian yang tidak disengaja dan tidak dapat dihindari oleh manusia, juga bisa terjadi kapan saja. Saat ini, jumlah kecelakaan yang terjadi di dunia mengalami peningkatan. Populasi yang meningkat serta adanya jumlah kendaraan yang meningkat membuat parahnya kecelakaan terjadi hampir setiap hari. Sebagian besar, negara-negara berkembang menjadi sasaran kecelakaan di jalan dari hari ke hari. Kasus kecelakaan di Indonesia pun juga mengalami peningkatan. Dari data Korlantas Polri yang sudah dipublikasikan oleh Kementerian Perhubungan, pada tahun 2021 angka kecelakaan lalu lintas di Indonesia mencapai 103.645 kasus. Kasus kecelakaan lalu lintas di Indonesia cenderung fluktuatif. Tahun 2019 menjadi kasus tertinggi kecelakaan, yaitu mencapai 116.411 kasus. Salah satu penyebab terjadinya kecelakaan lalu lintas adalah mengalami rasa kantuk atau bisa disebut *microsleep*. Menurut jurnal Cyberarea Vol. 2 No. 5, *Microsleep* adalah keadaan tidur panjang berkisar antara 1 sampai 30 detik, dimana orang yang mengalami kondisi ini gagal merespon sensor motorik dan menjadi tidak sadarkan diri. Untuk menghindari dan meminimalisir kasus kecelakaan lalu lintas pada pengendara

roda empat, maka dibutuhkan sebuah alat untuk memberi peringatan pada pengemudi roda empat saat mengalami rasa kantuk atau *microsleep* secara *real time* melalui kedipan mata pengemudi.

2. METODE

2.1 Metode Waterfall

Di bawah ini adalah tahapan dari metode *waterfall*, yaitu :



Gambar 1. Metode *Waterfall*

a. *Analysis*

Di tahap ini perlu dilakukan analisis kebutuhan sistem dan pemaparan kebutuhan yang harus dipenuhi oleh sistem dan alat yang akan dibangun. Pemodelan proses bisnis menggunakan *Entity Relationship Diagram* (ERD).

b. *Design*

Pada tahap ini merupakan tahap perancangan sistem dan alat seperti, perancangan basis data, perancangan sistem, perancangan perangkat keras (alat pendekripsi).

c. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan di uji untuk fungsionalitas yang disebut sebagai *unit testing*.

d. *Testing*

Tahap ini merupakan tahap pengujian sistem yang telah dibuat. Metode pengujian yang digunakan adalah pengujian blackbox untuk melihat kesalahan fungsi program dan melakukan *debugging*, pengujian whitebox untuk mengungkap kesalahan implemenasi, dan pengujian alat pendekripsi kantuk.

e. *Maintenance*

Di tahap ini merupakan tahap pemeliharaan sistem dan alat berupa proses perbaikan kesalahan jika sewaktu-waktu ditemukan kesalahan dan program yang sudah di rancang tidak berjalan setelah digunakan.

2.2. Alur Kerja Sistem

Gambar 2. Alur Kerja Sistem menunjukkan pengguna akan memulai input sistem pendekripsi dengan melakukan registrasi akun / *login* akun. Ketika sudah masuk kedalam sistem, pengguna membuat perjalanan yang terdiri dari tempat awal, tempat tujuan, estimasi perjalanan, serta jarak perjalanan. Kemudian sistem akan memulai perjalanan dan pengguna menggunakan alat pendekripsi

kantuk. Jika pengguna terdeteksi kantuk, *buzzer* akan menyala memberi peringatan kepada pengguna dan data akan masuk ke riwayat kantuk. Jika pengguna tidak terdeteksi kantuk, *buzzer* tidak akan menyala untuk peringatan dan data tidak masuk ke riwayat kantuk. Kondisi ini akan terus berlanjut hingga pengguna menyelesaikan perjalannya.



Gambar 2. Alur Kerja Sistem

3. ANALISA DAN PEMBAHASAN

Pada bagian ini berisi hasil dari kegiatan penelitian yang sudah dilakukan

3.1 Analisa dan Perancangan

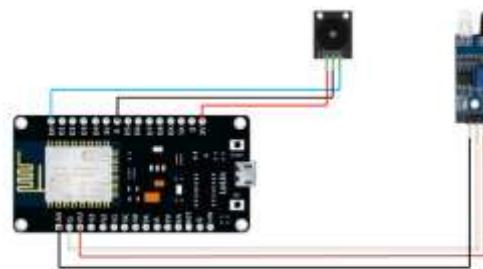
a. Gambaran Umum

Gambaran umum dari alat pendeteksi kantuk ini, diantaranya :

1. Alat terhubung dengan modul NodeMCU melalui kabel *jumper*.
2. Alat akan terkoneksi ke *server* NodeMCU melalui *wifi module* dari kacamata pengguna.
3. Alat akan mendeteksi mata (objek) untuk menentukan pengguna mengantuk atau tidak mengantuk.

b. Perancangan Perangkat Keras

Perangkat keras akan disusun dengan beberapa komponen yang terhubung dengan komponen lainnya di perancangan alat ini. Di bawah ini skematis dari perangkat keras.

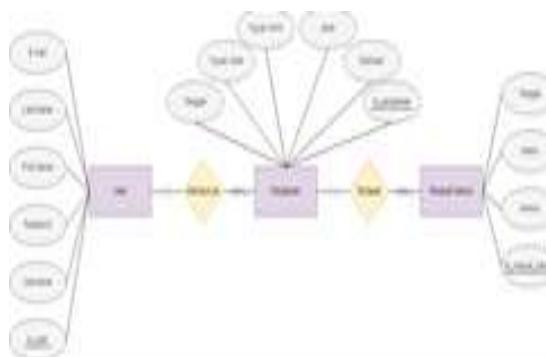


Gambar 3. Skematik Perangkat Keras

Terdapat 3 komponen utama pada perancangan perangkat keras ini yang dapat membuat sistem bekerja dengan baik dan saling terhubung satu dengan yang lainnya. Komponen ini terdiri dari, mikrokontroler NodeMCU, *Sensor Infrared* (Ir) FC-51, dan *Buzzer Low Active* 3 pin.

c. Perancangan Perangkat Lunak

Pada tahap perancangan ini, dilakukan perancangan database system menggunakan *Entity Relationship Diagram* (ERD). ERD dipergunakan untuk menginterpretasikan, menentukan dan menyimpan kebutuhan-kebutuhan untuk sistem pemrosesan database. Dalam pembuatan sistem ini, juga perlu menggunakan ERD untuk mengetahui hubungan setiap entitas dan atribut. ERD dapat ditunjukkan pada Gambar 4.

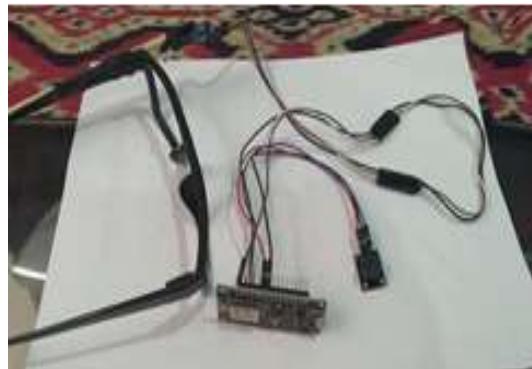


Gambar 4. ERD

1. Relasi antara *User* dan *Perjalanan* adalah *One to Many*, yang artinya satu *User* dapat memiliki banyak data *Perjalanan*.
2. Relasi antara *Perjalanan* dan *Riwayat Kantuk* adalah *One to Many One to Many*, yang artinya satu *Perjalanan* dapat memiliki banyak *Riwayat Kantuk*.

d. Implementasi Sistem

Setelah membuat rancangan perangkat lunak, selanjutnya adalah membuat *prototype* sistem, sebuah kacamata yang dihubungkan dengan seluruh komponen, dengan sumpit yang disematkan pada gagang kacamata menggunakan solasi, sumpit ini berfungsi sebagai pegangan untuk menyematkan *sensor infrared* dan mengatur posisi jarak dari *sensor infrared* terhadap objek yang akan di deteksi. Dapat dilihat Gambar 5 dibawah ini.



Gambar 5. Prototype Kacamata Pendeksi Kantuk



Gambar 6. Penggunaan Kacamata Pendeksi

3.2 Pengujian

a. Uji Blackbox

Uji *blackbox* adalah pengujian berdasarkan fungsi dari program. Tujuan pengujian ini untuk menemukan kesalahan fungsi pada program. Pengujian dilakukan dengan melakukan semua kemungkinan yang terjadi dan dilakukan secara berulang-ulang. Jika dalam pengujian tersebut masih ditemukan kesalahan, maka akan dilakukan pencarian dan perbaikan atau debugging untuk memperbaiki kesalahan yang sudah terjadi. Setelah selesai melakukan perbaikan, maka akan mengulang pengujian. Pengujian dan perbaikan ini akan dilakukan terus menerus sampai didapatkan hasil yang terbaik. Hasil uji blackbox dapat dilihat pada tabel dibawah ini :

Tabel 1. Uji Blackbox

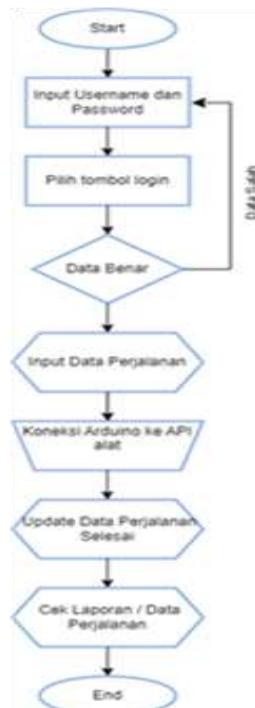
No.	Aktivitas Pengujian	Cara Pengujian	Hasil Yang Diharapkan	Hasil Uji
1	<i>Sign Up User</i>	Memasukkan <i>username</i> , <i>password</i> , <i>confirm password</i> , <i>first name</i> , <i>last name</i> , dan <i>email</i>	Aplikasi akan menambahkan data <i>user</i> ke dalam sistem	Diterima
2	<i>Sign In User</i>	Memasukkan <i>username</i> dan <i>password</i>	Aplikasi akan menampilkan halaman utama sistem	Diterima

3	Input Data Perjalanan	Memasukkan tanggal, tempat awal, tujuan akhir, jarak (km), dan estimasi (jam), serta klik Mulai	Aplikasi akan menambahkan data perjalanan pada sistem dan sistem akan berjalan	Diterima
4	Update Status Perjalanan menjadi Selesai	Klik Berakhir Perjalanan	Aplikasi akan menambahkan data perjalanan ke dalam riwayat perjalanan pada sistem	Diterima
5	Source Arduino	Alat akan membaca nilai sensor	Alat dapat membaca nilai dari sensor	Diterima
6	API alat	Data dari alat dan sensor masuk ke database	Data berhasil masuk ke <i>database</i>	Diterima
7	<i>Sign Out User</i>	Klik Keluar	<i>User</i> keluar dari sistem	Diterima

b. Uji *Whitebox*

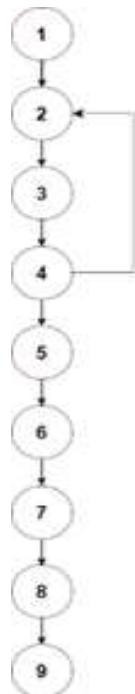
Pengujian *whitebox* adalah metode desain kasus uji yang menggunakan struktur kontrol desain prosedural untuk mendapatkan kasus uji. Pengujian *whitebox* dapat mengungkap kesalahan implementasi seperti manajemen kunci yang buruk dengan menganalisis cara kerja internal dan struktur perangkat lunak. Pengujian kotak putih berlaku pada tingkat integrasi, unit, dan sistem dari proses pengujian perangkat lunak. Dalam pengujian kotak putih, penguji perlu melihat ke dalam kode sumber dan mencari tahu unit kode mana yang berperilaku tidak pantas. (Ehmer, dalam Ehmer 2012).

Di bawah ini merupakan tahapan-tahapan pengujian dimulai dari pembuatan *flowchart*, *flowgraph*, perhitungan kompleksitas *cyclomatic*, perhitungan jalur independen, serta uji kasus.



Gambar 7. Flowchart login

Setelah membuat *flowchart*, maka dapat dibuat *flowgraph* seperti di bawah ini :



Gambar 8. *Flowgraph Login*

Setelah membuat *flowgraph*, maka dapat ditentukan :

Jumlah Node : 9

Jumlah Edge : 9

Tentukan kompleksitas *cyclomatic* :

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 9 - 9 + 2 \\ &= 2 \end{aligned}$$

Maka jumlah jalur yang akan di uji adalah 2 jalur.

Jalur 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9

Jalur 2 = 1 – 2 – 3 – 4 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9

Tabel 2. Deskripsi Jalur 1 *Login*

Path	1
Jalur Independen	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9
Skenario	<ol style="list-style-type: none">1. Mulai2. Masukkan <i>username</i> dan <i>password</i>3. Pilih <i>login</i>4. Validasi data benar



-
5. *Input* data perjalanan
 6. Menghubungkan Arduino ke API alat
 7. *Update* data perjalanan selesai
 8. Memeriksa laporan / data perjalanan
 9. Selesai

Hasil Pengujian	Berhasil
------------------------	----------

Tabel 3. Deskripsi Jalur 2 *Login*

Path	2
Jalur Independen	1 – 2 – 3 – 4 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9
Skenario	<ol style="list-style-type: none">1. Mulai2. Masukkan <i>username</i> dan <i>password</i>3. Pilih <i>login</i>4. Validasi data salah5. Kembali ke halaman utama, memasukkan ulang <i>username</i> dan <i>password</i>6. Pilih <i>login</i>7. Validasi data benar8. <i>Input</i> data perjalanan9. Menghubungkan Arduino ke API alat10. <i>Update</i> data perjalanan selesai11. Memeriksa laporan / data perjalanan12. Selesai

Hasil Pengujian	Berhasil
------------------------	----------

c. Pengujian *Sensor Infrared* FC-51

Pada pengujian ini dilakukan dengan menggunakan kacamata yang sudah disematkan dengan *sensor infrared* dan terdapat 2 skema yaitu posisi kelopak mata terbuka dan tertutup. Kemudian, melakukan percobaan dan didapatkan data seperti tabel dibawah ini.

Tabel 4. Pengujian *Sensor Infrared* FC-51

Percobaan ke	Posisi Kelopak Mata	Nilai Deteksi Sensor Ir	Persentase Keberhasilan Deteksi
1	Terbuka	<i>High</i>	100%
	Tertutup	<i>Low</i>	
2	Terbuka	<i>High</i>	100%
	Tertutup	<i>Low</i>	
3	Terbuka	<i>High</i>	100%



	Tertutup	<i>Low</i>	
4	Terbuka	<i>High</i>	100%
	Tertutup	<i>Low</i>	
5	Terbuka	<i>High</i>	100%
	Tertutup	<i>Low</i>	
6	Terbuka	<i>High</i>	100%
	Tertutup	<i>Low</i>	
7	Terbuka	<i>High</i>	100%
	Tertutup	<i>Low</i>	
8	Terbuka	<i>High</i>	100%
	Tertutup	<i>Low</i>	
9	Terbuka	<i>High</i>	100%
	Tertutup	<i>Low</i>	
10	Terbuka	<i>High</i>	100%
	Tertutup	<i>Low</i>	
11	Terbuka	<i>High</i>	100%
	Tertutup	<i>Low</i>	
TOTAL PERSENTASE KEBERHASILAN			100%

Hasil persentase akurasi keberhasilan *sensor infrared* dalam mendeteksi kelopak mata dapat dihitung menggunakan cara :

$$\text{Persentase} = \frac{\text{Persentase Keberhasilan}}{\text{Total percobaan}}$$

Setelah melakukan perhitungan diatas, maka dapat hasil nilai persentase keberhasilan deteksi *sensor infrared* sebesar 100%.

d. Pengujian *Module Buzzer*

Pengujian ini dilakukan sebanyak jumlah pengujian pada *sensor infrared*, yakni 11 kali, dengan nilai awal *buzzer* adalah *HIGH* yang artinya *buzzer* menyala. Berikut hasil pengujian *buzzer*.

Tabel 5. Pengujian *Buzzer*

No.	Nilai <i>Buzzer</i>	Hasil <i>Buzzer</i>	Ket
1	<i>HIGH</i>	Menyala	Sukses
2	<i>HIGH</i>	Menyala	Sukses
3	<i>HIGH</i>	Menyala	Sukses
4	<i>HIGH</i>	Menyala	Sukses
5	<i>HIGH</i>	Menyala	Sukses
6	<i>HIGH</i>	Menyala	Sukses
7	<i>HIGH</i>	Menyala	Sukses



8	HIGH	Menyala	Sukses
9	HIGH	Menyala	Sukses
10	HIGH	Menyala	Sukses
11	HIGH	Menyala	Sukses

Tabel diatas menunjukkan bahwa dari 11 percobaan dengan mengirim perintah ke *buzzer* untuk mendapat nilai *HIGH* yang semestinya menyala dan menimbulkan suara, maka persentase keberhasilan yang dihasilkan *buzzer* adalah 100%.

e. Pengujian Seluruh Sistem

Pengujian ini dilakukan dengan menggabungkan semua komponen menjadi satu sistem, yaitu sebagai sistem pendekripsi kantuk dengan *sensor infrared* FC-51 mendekripsi kelopak mata pengendara dan mengirimkan sinyal berupa alarm *buzzer* kepada pengendara. Pengujian dilakukan sebanyak 15 kali dengan dibagi 3 cara, yaitu posisi kelopak mata terbuka, berkedip, dan tertutup. Berikut hasil pengujian dapat dilihat pada tabel di bawah ini :

Tabel 6. Hasil Pengujian Seluruh Sistem

Posisi Kelopak Mata	Deteksi Kantuk	Hasil Buzzer	Ket
Terbuka	(X)	Tidak Menyala	Berhasil
	(X)	Tidak Menyala	Berhasil
	(X)	Tidak Menyala	Berhasil
	(X)	Tidak Menyala	Berhasil
	(X)	Tidak Menyala	Berhasil
Berkedip	✓	Menyala	Gagal
	✓	Menyala	Gagal
	(X)	Tidak Menyala	Berhasil
	(X)	Tidak Menyala	Berhasil
	(X)	Tidak Menyala	Berhasil
Tertutup	✓	Menyala	Berhasil
	✓	Menyala	Berhasil

Setelah melakukan 15 kali percobaan dengan 3 cara yang berbeda, sistem mampu menjalankan fungsinya sebanyak 13 kali. Sehingga didapatkan hasil persentase keberhasilan sistem keseluruhan dalam mendekripsi kantuk serta mengirimkan sinyal *buzzer* sebesar 86,6%.

4. KESIMPULAN

Setelah melakukan seluruh pengujian pada sistem dengan NodeMCU sebagai mikrokontroler, maka dapat disimpulkan bahwa sistem mampu mendekripsi kantuk pada pengendara



dengan memposisikan kelopak mata agar terdeteksi oleh *sensor infrared* FC-51 dan mengirimkan sinyal berupa alarm *buzzer* ke pengendara. Nilai keberhasilan pengujian sistem ini sebesar 86,6%.

Penelitian ini dapat dikatakan bahwa alat dan sistem yang dibuat pada awal penelitian berjalan sesuai harapan. Namun, peneliti menyarankan untuk pengembangan selanjutnya yaitu pendektsian kantuk ini menambahkan *pulse sensor*, agar nilai akurasi pendektsian kantuk ini tinggi.

REFERENCES

- Sharma, Shivani & Sebastian, Shoney (2019). IoT Based Car Accident Detection and Notification Algorithm for General Road Accident. 9 (5). 4020-4026.
- Syofian, Andi & Yultrisna (2019). Helm Untuk Memberitahu Kondisi Fisik Pengendara Sepeda Motor Saat Mengemudi Berbasis Mikrokontroler. 21 (1). 1-7.
- Adika, Ibnu Adib, Suradji, Djarot & Tsani, Muhammad Rifqi (2020). Sistem Peringatan Durasi Mengemudi Berbasis Internet of Things Guna Keselamatan Berlalu Lintas Menggunakan Arduino. 786-795.
- Suraya, Hoppy, Ziad, Ibnu, & Suroso (2021). Rancang Bangun Alat Pendektsi Kantuk Pada Mobil Berbasis IoT Menggunakan Raspberry Pi dan Kamera. 20 (3). 385-391.
- Indru Moorjani, Arjun & Suryo Putranto, Leksmono (2021). Hubungan Antara Rawan Bosan dan Kemudahaan Tertidur dengan Microsleep Saat Mengemudi. 4 (3). 729-736.
- databoks.(2022, 24 Maret). Angka Kecelakaan Lalu Lintas di Indonesia Meningkat di 2021,Tertinggi dari Kecelakaan Motor. Diakses pada 24 Mei 2022, dari <https://databoks.katadata.co.id/datapublish/2022/03/24/angka-kecelakaan-lalu-lintas-di-indonesia-meningkat-di-2021-tertinggi-dari-kecelakaan-motor>
- Maiza, Muhammad Irviner Fidellio, Budi, Agung Setia & Syauqy, Dahnial (2022). Implementasi Low Power pada Sistem Notifikasi Kantuk pada Pengemudi menggunakan Finite-State Machine berbasis Arduino. 6 (7). 3077-3084.
- Indobot (2022, 20 Februari). Datasheet NodeMCU ESP8266 Lengkap dengan Pin dan Cara Akses. Diakses pada 22 Desember 2022, dari <https://indobot.co.id/blog/datasheet-nodemcu-esp8266-lengkap-dengan-pin-dan-cara-akses/>