

---

---

## RANCANGAN PENGGERAK MOBIL *BUGGY* OTOMATIS BERBASIS MIKROKONTROLER DI POLITEKNIK PENERBANGAN MEDAN

Hezkiel Omega Putra Pinem<sup>1</sup>, Ayub Wimatra<sup>2</sup>, Sartika M. Sibarani<sup>3</sup>,  
Catra Indra Cahyadi<sup>4</sup>

Politeknik Penerbangan Medan, Medan

e-mail: <sup>1</sup>hezkielpinen@gmail.com, <sup>2</sup>ayub83wimatra@gmail.com,  
<sup>3</sup>sibmagdalena@gmail.com, <sup>4</sup>catraindracahyadi@gmail.com

**Abstract:** *This research aims to design an automatic buggy vehicle driver system based on the ESP32 microcontroller at Medan Aviation Polytechnic. The system is designed to enhance the operational efficiency of campus vehicles, enabling them to operate autonomously without requiring a direct driver. The ESP32 microcontroller serves as the central controller, regulating the DC motor for steering, pneumatic motors for gas and brake pedals, and a motor driver to control the current to the motors. The system's performance testing showed that the power required for the pneumatic motors (gas and brake) is 36 watts, and for the DC motor steering, it is 72 watts. Turning tests at a 90° angle and braking tests, where the braking distance was reduced by 10 cm after calibration, indicated that the system operates as expected. These tests confirmed the system's responsiveness and accuracy in automatic operation. Technical challenges in testing, such as adjusting sensor sensitivity and calibrating the pneumatic system, were addressed through repeated trials. This research contributes to the development of vehicle automation in educational settings and offers potential for further advancement, such as integrating LiDAR and GPS sensors for improved navigation accuracy.*

**Keywords:** *Automatic Buggy Vehicle Driver, Microcontroller, ESP32, Control System, Vehicle Automation*

**Abstrak:** Penelitian ini bertujuan merancang sistem penggerak mobil *buggy* otomatis berbasis mikrokontroler ESP32 di Politeknik Penerbangan Medan. Sistem ini dirancang untuk meningkatkan efisiensi operasional kendaraan kampus yang dapat bergerak secara otonom tanpa memerlukan pengemudi langsung. Mikrokontroler ESP32 digunakan sebagai pusat kendali, mengatur motor DC untuk setir, motor pneumatik untuk pedal gas dan rem, serta motor *driver* untuk mengatur arus motor. Pengujian kinerja sistem menunjukkan daya yang dibutuhkan untuk motor pneumatik gas dan rem sebesar 36 watt, serta 72 watt untuk motor DC penggerak setir. Pengujian belokan pada sudut 90° dan pengujian pengereman yang dipersingkat 10 cm setelah kalibrasi menunjukkan kinerja yang sesuai harapan. Pengujian ini mengonfirmasi responsivitas dan akurasi sistem dalam pengoperasian otomatis. Tantangan teknis dalam pengujian, seperti pengaturan sensitivitas sensor dan kalibrasi sistem pneumatik, berhasil diatasi melalui uji coba berulang. Penelitian ini memberikan kontribusi pada pengembangan otomasi kendaraan di lingkungan pendidikan dan berpotensi untuk pengembangan lebih lanjut, seperti penggunaan sensor LiDAR dan GPS untuk meningkatkan akurasi navigasi.

**Kata kunci:** Penggerak Mobil *Buggy* Otomatis, Mikrokontroler, ESP32, Sistem Kendali, Otomasi Kendaraan

## PENDAHULUAN

Perkembangan teknologi transportasi modern saat ini tidak hanya berfokus pada peningkatan performa kendaraan, tetapi juga pada efisiensi energi dan keberlanjutan lingkungan. Mobil listrik menjadi salah satu solusi yang terus dikembangkan karena mampu menekan emisi karbon dan mengurangi ketergantungan pada bahan bakar fosil (Songli et al., 2024). Sejalan dengan itu, kemajuan dalam otomasi dan kecerdasan buatan mendorong lahirnya kendaraan otonom yang dapat beroperasi dengan minim intervensi manusia, serta meningkatkan kenyamanan dan keselamatan berkendara (Latipah et al., 2025). Integrasi antara kendaraan listrik dan sistem kendali otomatis dipandang sebagai langkah strategis menuju mobilitas berkelanjutan di masa depan (Setiyo et al., 2023).

Di Indonesia, implementasi teknologi otomasi pada kendaraan masih menghadapi tantangan, mulai dari biaya produksi yang tinggi, kesiapan infrastruktur, hingga regulasi yang ketat. Namun, dalam skala yang lebih kecil, teknologi ini dapat mulai diuji coba di lingkungan pendidikan sebagai sarana penelitian sekaligus solusi peningkatan efisiensi operasional (Afriandy Maulana et al., 2023).

Politeknik Penerbangan Medan sebagai perguruan tinggi kedinasan di bawah Kementerian Perhubungan memiliki berbagai sarana pendukung pendidikan, salah satunya kendaraan operasional berupa mobil *buggy*. Selama ini, pengoperasian mobil *buggy* masih bersifat manual sehingga memerlukan pengemudi (Aziz et al., 2020).

Hal ini terkadang kurang efisien, terutama ketika kendaraan dibutuhkan secara mendadak atau saat tidak tersedia pengemudi. Sebagai contoh, untuk menjemput tamu di area kampus, petugas harus berjalan ke tempat parkir lalu membawa kendaraan, yang dapat memakan waktu hingga 20 menit.

Penelitian terdahulu telah mengaplikasikan otomasi dan IoT pada berbagai sistem non-kendaraan, misalnya untuk sistem penyiraman otomatis menggunakan IoT untuk tanaman anggrek hitam Papua, yang memperlihatkan bahwa sensor, aktuator, dan kendali otomatis dapat memberikan efisiensi operasional dan respons yang baik (Wimatra et al., 2024). Selain itu, penelitian (Wimatra et al., 2019) juga membahas kontrol otomatis dan perintah kendali remote yang mendekati konteks otomasi kendaraan, meskipun belum langsung diterapkan pada kendaraan operasional kampus atau mobil *buggy*. Dari penelitian-penelitian tersebut terlihat bahwa meskipun teknologi kendali otomatis sudah cukup matang di domain lain, terdapat gap dalam penerapan di kendaraan operasional kampus, khususnya dalam sistem kendali motor/car movement secara otomatis dikombinasikan dengan aplikasi pemanggilan kendaraan (on-demand).

Dengan adanya sistem kendali otomatis, mobil *buggy* dapat digerakkan secara mandiri melalui perintah aplikasi, sehingga proses penjemputan kendaraan lebih cepat, efisien, dan responsif. Selain mendukung kegiatan operasional, penelitian ini juga selaras dengan upaya Politeknik Penerbangan Medan dalam mengintegrasikan konsep Revolusi Industri 4.0 ke dalam dunia pendidikan (Fadlan Siregar & Cahyadi, 2021).

Dalam tugas akhir ini, fokus utama diarahkan pada perancangan sistem kendali motor sebagai inti dari sistem otomasi. Sistem kendali motor akan diatur agar dapat merespons instruksi mikrokontroler dengan presisi, sehingga mobil *buggy* mampu bergerak sesuai arah dan kecepatan yang ditentukan secara otomatis.

## METODE

Metode penelitian yang digunakan dalam penelitian ini adalah *Research and Development* (R&D), yaitu metode yang

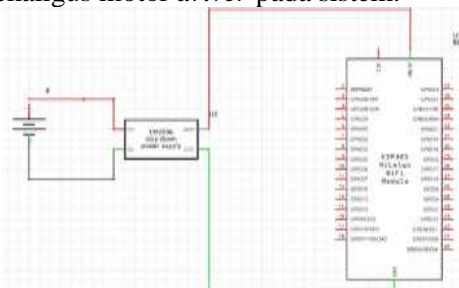
bertujuan untuk menghasilkan suatu produk sekaligus menguji efektivitasnya (Suyatmo et al., 2020). Menurut (Sugiyono, 2020), penelitian pengembangan digunakan untuk menciptakan atau menyempurnakan produk agar mampu memberikan manfaat sesuai dengan tujuan yang telah ditetapkan. Dalam penelitian ini, produk yang dikembangkan berupa sistem penggerak mobil *buggy* otomatis berbasis mikrokontroler. Proses pengembangan produk mengacu pada model ADDIE (*Analysis, Design, Development, Implementation, Evaluation*) karena model ini bersifat sistematis, fleksibel, dan sesuai untuk pengembangan produk teknologi (Mariam, 2019).

## HASIL DAN PEMBAHASAN

### Instalasi Perangkat Keras

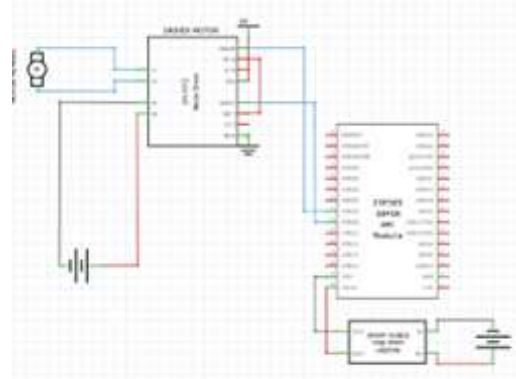
Perancangan sistem penggerak mobil *buggy* otomatis dengan ESP32, *power supply step-down* LM2596, motor *driver* module BTS7960, pneumatic linier, dan motor DC 24 volt. Komponen utama seperti ESP32 dan motor *driver* ditempatkan dalam sebuah box.

Selanjutnya dilakukan proses integrasi, dimulai dari menghubungkan modul *step-down* LM2596 dengan ESP32, di mana pin VOUT+ LM2596 disambungkan ke pin VIN 5V ESP32 sebagai sumber tegangan utama, sedangkan pin VOUT- dihubungkan ke pin GND ESP32 untuk jalur ground. Konfigurasi ini berfungsi sebagai suplai tegangan 5V yang dibutuhkan ESP32 sekaligus motor *driver* pada sistem.



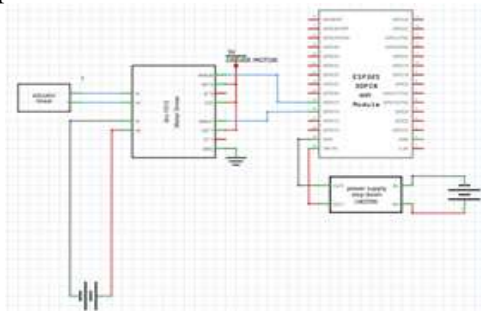
Gambar 1 Konfigurasi ESP32 dengan LM2596

Rangkaian pada Gambar 1 menggunakan baterai sebagai sumber daya yang diturunkan menjadi 5V oleh modul LM2596 untuk mensuplai ESP32S dan *driver* motor L298N. ESP32S mengatur arah dan kecepatan motor melalui pin GPIO18, GPIO19, dan GPIO23 dengan sinyal PWM, yang kemudian diteruskan *driver* motor ke motor kemudi. Semua ground dari komponen dihubungkan bersama agar rangkaian stabil.



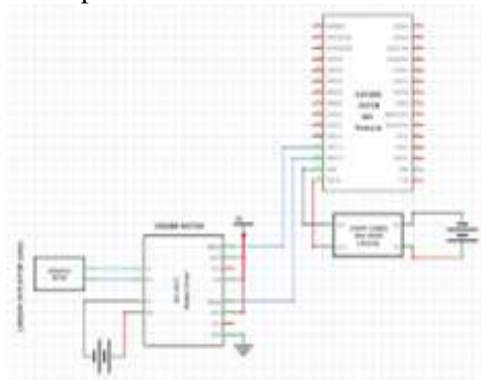
Gambar 2 Konfigurasi ESP32 dengan Motor Driver dan Motor DC

Rangkaian yang ada pada Gambar 2 ini menghubungkan aktuator linear, motor *driver* L298N, ESP32, dan modul LM2596. Sumber daya utama masuk ke LM2596 untuk diturunkan menjadi 5V yang mensuplai ESP32 dan motor *driver*. ESP32 mengontrol arah aktuator linear melalui pin IN1 dan IN2, sementara motor *driver* menyalurkan arus ke aktuator melalui OUT1 dan OUT2. Semua *ground* disatukan agar tegangan stabil, dan aktuator linear pada sistem pneumatik digunakan untuk menekan pedal rem secara otomatis.



Gambar 3. Konfigurasi Pneumatic Linear dengan Motor Driver dan ESP32

Rangkaian ini terdiri dari aktuator linear yang dikendalikan motor *driver* L298N dengan ESP32 sebagai pengatur utama dan LM2596 sebagai sumber tegangan 5V seperti pada Gambar 3. GPIO22 dan GPIO23 pada ESP32 terhubung ke pin IN1 dan IN2 *driver* motor untuk mengatur arah gerak aktuator, sementara OUT1 dan OUT2 *driver* motor terhubung ke aktuator linear agar dapat bergerak maju atau mundur. LM2596 menyalurkan tegangan dari sumber utama ke VIN ESP32 dan pin 12V *driver* motor, dengan semua ground disatukan agar stabil. Pada rangkaian ini, aktuator linear difungsikan untuk menekan pedal gas secara otomatis melalui mekanisme pneumatik yang dapat dilihat pada Gambar 4 dibawah ini.



**Gambar 4 Konfigurasi Pneumatic dengan Motor Driver dan ESP32**

### Instalasi Perangkat Lunak

Pada proses instalasi perangkat lunak, Arduino IDE berperan sebagai penyusun instruksi yang nantinya digunakan untuk mengatur modul-modul yang terhubung ke Arduino IDE. Modul memproses satu persatu perintah sesuai dengan yang telah dibuat di Arduino IDE dan diunggah ke ESP32. Setelah itu, program dapat diunggah ke ESP32 dengan menghubungkan kabel USB dari laptop ke board. Tahap awal instalasi perangkat lunak adalah menginstal library yang dibutuhkan melalui menu *Sketch* → *Include Library* → *Manage Library*. *Library* yang digunakan antara lain WiFi.h, HTTPClient.h, ArduinoJson.h, HardwareSerial.h, dan TFmini. Setelah itu, *library* dapat dipanggil pada sketch

agar seluruh modul dapat berfungsi sesuai kebutuhan sistem seperti Gambar 5 dibawah ini.



**Gambar 5 Tampilan Library yang sudah di Include**

Pada Gambar 6, bagian setup() dalam program mengatur perangkat keras yang digunakan dengan mendeklarasikan pin output untuk mengontrol motor (stir, rem, dan gas) menggunakan pinMode(). Selain itu, ESP32 dihubungkan ke jaringan Wi-Fi serta komunikasi serial diinisialisasi untuk kebutuhan debugging dan komunikasi antar perangkat.



**Gambar 6 Tampilan Program Pengaturan Perangkat Keras**

Dapat dilihat pada Gambar 7 dalam loop(), ESP32 secara berkala melakukan HTTP GET request ke server untuk mengambil data aksi yang harus dijalankan. Jika perintah yang diterima berupa "maju", "steady", atau "rem", maka ESP32 akan mengontrol motor sesuai instruksi yang diterima.



**Gambar 7 Proses Pengambilan Data dan Eksekusi**

Jika perintah "maju", "steady", atau "rem" diterima, ESP32 akan mengaktifkan pin yang terhubung dengan motor untuk menjalankan aksi sesuai instruksi. Fungsi maju(), berhenti(), dan rem() digunakan untuk mengatur output pin dalam menggerakkan aktuator (motor). Tampilan kontrol akuator dapat dilihat pada Gambar 8 dibawah ini.

```
void loop() {  
  unsigned long ambilget1 = millis();  
  if(ambilget1 - ambilget >=1000){  
    getAksi();  
    ambilget = millis();  
  }  
  
  // while(BacaSensor.available()>0){  
  //   // data ="";  
  //   char c = BacaSensor.read();  
  //   data += c;  
  //   delay(10);  
  //   Serial.println(data);  
  // }  
  // data.trim();  
  if(aktifkan == true){  
    while(BacaSensor.available()>0){  
      // data ="";  
      char c = BacaSensor.read();  
      data += c;  
      delay(10);  
      Serial.println(data);  
    }  
  }  
}
```

**Gambar 8 Kontrol Akuator berdasarkan Perintah**

Pada Gambar 9 program juga mengirimkan status ke server setelah aksi dijalankan. Hal ini dilakukan melalui fungsi kirimDataKosong(), yang mengirimkan HTTP GET request untuk memberi tahu server bahwa perintah telah selesai diproses.

```
}  
// if(count_aktuator <1){  
//   maju();  
// }  
if(count_aktuator == 4){  
  state = false;  
  data = "10";  
}  
  
if(state == true){  
  maju();  
}  
  
} else if(data == "steady"){  
  unsigned long a1 = millis();  
  Serial.println("berhenti");  
  if(a1 - a >=50){  
    count_aktuator++;  
    a = millis();  
  }  
  tarik_maju();  
  BacaSensor.print("berhenti");  
  kirimDataKosong();  
}
```

**Gambar 9 Pengiriman Data ke Server**

Pada program ini juga menggunakan parsing JSON untuk membaca data dari server. Data status yang dikirim dalam format JSON diproses oleh ESP32 untuk menentukan aksi yang harus dijalankan. Dapat dilihat pada Gambar 10 dibawah ini, saat program dijalankan.

```
void getAksi() {  
  if (WiFi.status() == WL_CONNECTED) {  
    HTTPClient http;  
    String url = serverName; // ganti dengan endpoint Anda  
    http.begin(url);  
    // ...  
    StaticJsonDocument<256> doc;  
    DeserializationError error = deserializeJson(doc, payload);  
    if (!error) {  
      const char* aksi = doc["data"]["aksi"];  
      String conv = String(aksi);  
      Serial.println(aksi);  
      if(conv == "parkir"){  
        aktifkan = true;  
        BacaSensor.print("ok");  
      }  
    }  
  }  
}
```

**Gambar 10 Parsing JSON dan Pengambilan Aksi dari Server**

Dalam program ini, koneksi Wi-Fi dan HTTP dilengkapi dengan mekanisme penanganan *error* agar perangkat tetap dapat berfungsi dengan baik meskipun terjadi gangguan jaringan. Jika koneksi gagal, pesan *error* akan ditampilkan pada Gambar 11 serial monitor.

```
if (httpResponseCode > 0) {  
  String response = http.getString();  
  Serial.println("Response:");  
  Serial.println(response);  
} else {  
  Serial.print("Error on sending GET: ");  
  Serial.println(httpResponseCode);  
}
```

**Gambar 11 Koneksi Wi-Fi dan HTTP**

### Pengujian Alat

Dalam penelitian ini, sistem penggerak mobil *buggy* otomatis berbasis mikrokontroler ESP32 telah dirancang dan diuji dengan menggunakan berbagai komponen, seperti motor DC, motor pneumatik, dan sensor. Berikut ini adalah hasil dan pembahasan terkait pengujian

yang dilakukan.

**Tabel 1 Hasil Pengujian Motor dan Sensor**

Komponen	Fungsi	Hubungan dengan sensor	Peran dan penggerak
ESP32 (Mikrokontroler)	Pusat kendali	Mengelola Input dari sensor dan mengirimkan perintah	Mengatur pergerakan motor dan aktuator pneumatic
Sensor Garis ( <i>Line Follower</i> )	Mendeteksi jalur yang harus diikuti	Mengirimkan data jalur ke ESP32	Menyesuaikan pergerakan kendaraan sesuai jalur
Motor Pneumatik (Gas/Rem)	Penggerak pedal gas dan rem	Dijalankan sesuai perintah dari ESP32	Menekan pedal gas atau rem otomatis
Motor DC (Setir)	Penggerak arah setir	Menerima perintah dari ESP32	Mengarahkan setir ke kiri/kanan

Pada Tabel 1 yang dapat kita lihat di atas, dalam pengujian sistem penggerak mobil *buggy* otomatis, daya yang dibutuhkan oleh masing-masing komponen telah diukur untuk memastikan bahwa setiap elemen dalam sistem berfungsi dengan efisien dan sesuai dengan spesifikasi yang ditentukan. Berdasarkan hasil pengujian, daya yang dibutuhkan oleh komponen utama seperti motor DC untuk penggerak setir dan

motor pneumatik untuk pedal gas dan rem.

Rumus untuk menghitung daya (Watt) dalam sistem listrik adalah:

$$P = V \times I$$

Keterangan:

P = Daya (Watt)

V = Tegangan (Volt)

I = Arus (Ampere)

**Tabel 2 Perhitungan Penggunaan Daya**

Komponen	Tegangan (Vdc)	Arus (Ampere)	Daya (Watt)	Keterangan
Pneumatik (Gas)	12V	3A	36 watt	Menggerakkan pedal gas secara otomatis
Pneumatik (Rem)	12V	3A	36 watt	Menggerakkan pedal rem otomatis
Motor DC (Penggerak Setir)	24V	3A	72 watt	Menggerakkan setir sesuai perintah

Pada pengujian sistem penggerak mobil *buggy* otomatis, didapatkan hasil uji seperti pada

Tabel 2 menunjukkan bahwa waktu yang diperlukan untuk menempuh jarak 3 meter adalah 10 detik, dengan penerapan pengereman yang efektif pada jarak 20 cm. Data ini menunjukkan efisiensi waktu

yang dapat dicapai oleh sistem otomatis dibandingkan dengan metode manual, yang sebelumnya memerlukan waktu lebih lama.

Selain itu, dilakukan perbandingan antara hasil uji sebelum dan sesudah pengaturan jarak pengereman. Sebelum pengaturan, jarak pengereman yang

tercatat adalah 30 cm seperti pada Tabel 3, sementara setelah pengaturan sistem, jarak pengereman berhasil dipersingkat menjadi

20 cm. Analisis perbandingan ini menunjukkan peningkatan dalam efisiensi pengereman dan responsivitas sistem.

**Tabel 3 Pengujian**

Jarak (meter)	Waktu (detik)	Jarak pengereman sebelum	Jarak pengereman sesudah
2 M	10	30	20



**Gambar 12 Grafik Penggunaan Rem Sebelum dan Sesudah**

Secara keseluruhan, dapat dilihat pada Gambar 12 grafik hasil pengujian daya menunjukkan bahwa setiap komponen dalam sistem penggerak mobil *buggy* otomatis menggunakan daya sesuai dengan kebutuhan operasionalnya. Sistem ini dirancang agar komponen-komponen seperti motor DC dan motor pneumatik dapat bekerja secara efisien, tanpa pemborosan daya, yang pada akhirnya mendukung kelancaran dan ketahanan sistem secara keseluruhan.

## SIMPULAN

Berdasarkan hasil rancangan penggerak mobil *buggy* otomatis berbasis mikrokontroler di Politeknik Penerbangan Medan, dapat disimpulkan bahwa perancangan sistem ini memerlukan perencanaan yang terstruktur, di mana mikrokontroler berperan sebagai pusat kendali untuk memproses perintah dan mengatur pergerakan motor penggerak

secara otomatis. Proses tersebut melibatkan pemrograman logika kontrol agar mobil dapat bergerak sesuai instruksi, baik secara manual maupun otonom dengan menggunakan mikrokontroler ESP32S serta sensor *line follower* sebagai pendeteksi jalur. Sistem ini memanfaatkan beberapa komponen utama, antara lain mikrokontroler ESP32S, motor DC 16 volt sebagai penggerak setir, dua aktuator pneumatik untuk menggerakkan pedal gas dan rem, motor *driver* sebagai penguat arus, serta *power supply step-down* sebagai sumber daya. Tantangan teknis yang dihadapi meliputi proses kalibrasi pada sistem rem dan gas agar mobil dapat bergerak halus tanpa hentakan atau kelambatan, serta penyetelan sensor garis agar mampu membaca jalur secara akurat. Permasalahan ini dapat diatasi melalui proses trial and error secara berulang hingga diperoleh pengaturan yang sesuai baik pada respons motor maupun sensitivitas sensor terhadap jalur.

## DAFTAR PUSTAKA

- Afriandy Maulana, N., Adeliya Wahyu Ningrum, D., & Zeinudin, M. (2023). *Regulasi Hukum Mengenai Keselamatan dan Pertanggungjawaban Kendali Otomatis Mobil Listrik di Indonesia*. 384–391.
- Aziz, M., Marcellino, Y., Agnita Rizki, I., Anwar Ikhwanuddin, S., & Welman Simatupang, J. (2020). *Studi Analisis Perkembangan Teknologi Dan Dukungan Pemerintah Indonesia Terkait Mobil Listrik*. 22(1).
- Fadlan Siregar, M., & Cahyadi, C. I. (2021). Implementation of Automatic Car Cleaning System with Microcontroller System Atmega 8. *Article in Budapest International Research and Critics Institute (BIRCI-Journal) Humanities and Social Sciences*. <https://doi.org/10.33258/birci.v4i3.2548>

- Latipah, D., Subhiyanto, S., Adityarini, E., & Mardiansah, M. A. (2025). Integrasi Kecerdasan Buatan Dalam Industri Otomotif: Strategi, Tantangan dan Arah Transformasi Digital. *Go Infotech: Jurnal Ilmiah STMIK AUB*, 31(1), 197–204. <https://doi.org/10.36309/goi.v31i1.373>
- Mariam, N. (2019). *The Development of an ADDIE Based Instructional Model for ELT in Early Childhood Education*. <https://www.researchgate.net/publication/344377742>
- Setiyo, S., Cahyadi, C. I., Saputra, W., Sudjoko, R. I., & Faizah, F. (2023). Prototipe Pengaturan Sistem Kontrol Otomatis Fuel Treatment Tangki Bulanan Genset pada Power Station Bandar Udara. *Jurnal Kumparan Fisika*, 6(1), 55–64. <https://doi.org/10.33369/jkf.6.1.55-64>
- Songli, Y., Rachman, R. M., Denny, Arrang, A. T., Tangaran, B., & Gusty, S. (2024). *Mobilitas Berkelanjutan: Mobil Listrik di Era Society 5.0*. Arsy Media.
- Sugiyono. (2020). *Metodologi Penelitian Kuantitatif, Kualitatif, Research and Development*.
- Suyatmo, S., Cahyadi, C. I., Syafriwel, S., Khair, R., & Idris, I. (2020). Rancang Bangun Prototype Robot Pengantar Barang Cargo Berbasis Arduino Mega Dengan IOT. *Jurnal Sistem Komputer Dan Informatika (JSON)*, 1(3), 215. <https://doi.org/10.30865/json.v1i3.2186>
- Wimatra, A., Palpialy, Jemi. V, Sulistianingsih, I., Akbar, A., & Nasution, D. (2024). *Internet of Things on Automatic Watering Systems for Papuan Black Orchids*.
- Wimatra, A., Prayitno, H., & Nasution, D. (2019). Enhanced of House Security System Based Pir Sensor and Microcontroller Based. *Journal of Physics: Conference Series*, 1361(1). <https://doi.org/10.1088/1742-6596/1361/1/012048>