



# DESIGN OF PLESSEY AW S II RADAR SPEED MONITORING SYSTEM BASED ON INTERNET OF THINGS

Agung Setiyawan, Mardiana Irawaty  
Institut Teknologi Dirgantara Adisutjipto  
agung527652@gmail.com

**Abstrak.** *The Indonesian Air Force (TNI AU) has the main task of maintaining state sovereignty, especially in the airspace which operates Radar as one of the Main Tools and Armament Systems (Alutsista) owned by the Indonesian Air Force. Radar is used to detect/detect military aircraft traffic, monitor the movements of domestic aircraft or foreign aircraft that enter the radar coverage of the Sovereignty of the Republic of Indonesia (NKRI). The rotation of the radar antenna can be affected by high wind speeds which can result in a drag and thrust on the rotation of the radar antenna, adjusting the direction of the wind. But at this time, the Indonesian Air Force does not yet have an internet-based antenna rotation speed monitoring system, so a system is needed to determine the rotational speed of a radar antenna that can be accessed remotely using internet media. The methods used in designing an internet of things based antenna speed rotation monitoring system include literature studies, hardware and software system design, selection of tools and materials according to their specifications and functions and testing and analysis to determine their capabilities. The results obtained from writing this final project are expected to be useful for the Indonesian Air Force, especially for the Radar Unit (Satrad). This design can be used as a prototype monitoring system for rotating speed per minute on a Radar antenna that can be accessed remotely. Apart from that, it can also reduce the number of personnel in assignment, so that it can divert personnel to other tasks that are more needed by the unit.*

*Keywords: Monitoring System, Radar Antenna Loop, Arduino, Internet of Things.*

## PENDAHULUAN

Tentara Nasional Indonesia Angkatan Udara (TNI AU) sesuai dengan Undang Undang Republik Indonesia Nomor 34 Tahun 2004 Pasal 10 memiliki tugas antara lain adalah menegakkan hukum dan menjaga keamanan di wilayah udara yurisdiksi nasional sesuai dengan ketentuan hukum nasional dan hukum internasional yang telah diratifikasi serta melaksanakan tugas Tentara Nasional Indonesia (TNI) dalam pembangunan dan pengembangan kekuatan matra udara.

TNI Angkatan Udara (TNI AU) memiliki tugas pokok menjaga kedaulatan negara khususnya di wilayah udara. Dengan kondisi geografis wilayah Indonesia yang sangat luas dan potensi konflik yang rawan terjadi dengan pihak luar, maka dibutuhkan suatu Sistem Pertahanan Udara Nasional (Sishanudnas) yang memadai, yang dapat menjamin keamanan wilayah udara nasional.

*Radio Detecting and Ranging* (Radar) merupakan salah satu Alat Utama dan Sistem Persenjataan (Alutsista) yang dimiliki oleh TNI AU. Pada saat ini Satuan Radar (Satrad) TNI AU berjumlah 19 dan tersebar di berbagai wilayah Indonesia. Satuan Radar akan selalu melakukan



kegiatan operasi maupun pemeliharaan yang harus dilakukan secara optimal dan selalu berkoordinasi dengan beberapa satuan antara lain Depo Pemeliharaan 50 (Depohar 50), dan Komando Operasi Udara Nasional (Koopsudnas).

Dalam dunia militer, Radar digunakan untuk mengetahui/mendeteksi lalu lintas pesawat militer, memantau pergerakan pesawat dalam negeri atau pesawat asing yang masuk ke Radar coverage wilayah Negara Kedaulatan Republik Indonesia (NKRI) dan mengamati daerah perbatasan terhadap ancaman pesawat asing. Radar militer tidak hanya berfungsi untuk memantau pesawat militer saja, tetapi juga membantu dalam lalu lintas pesawat sipil.

Pada pengoperasiannya, Radar berputar sebesar  $360^\circ$  yang mana secara umum berputar sebanyak 6 putaran per menit (rpm) menggunakan motor AC 3 *Phase*. Perputaran antena Radar dapat terpengaruh oleh kecepatan angin yang tinggi dapat berakibat suatu drag (hambatan) dan thrust (dorongan) pada putaran antena Radar, menyesuaikan arah datangnya angin. Kecepatan angin yang bertiup dapat berubah-ubah dalam setiap saat. Penahan yang diakibatkan oleh angin dapat berpengaruh pada putaran antena yang seharusnya tetap konstan 6 rpm, oleh karena itu diperlukan suatu sistem monitoring kecepatan putaran antena.

Kecepatan angin yang melebihi ambang toleransi juga dapat berpengaruh pada sistem mekanika Radar apabila antena tetap dipaksakan untuk berputar. Untuk mengetahui kecepatan putaran yang dapat diakses jarak jauh tersebut maka diperlukan suatu sistem yang dapat memonitor kecepatan putaran antena Radar berbasis *internet of things* (IoT) sehingga apabila melebihi ambang batas angin yang diperbolehkan, secara otomatis motor akan mematikan putaran antena Radar.

## METODE

### Proses Penelitian

Pada pelaksanaan penelitian, digunakan metode yang bertujuan untuk menyelesaikan masalah dalam penelitian. Adapun metode yang akan dilakukan dalam penelitian ini adalah sebagai berikut:

#### 1. Studi Literatur

Mengumpulkan dan mempelajari berbagai bahan literatur terkait dengan *Mikrokontroler* Arduino, driver motor L298N, motor DC, LCD 2x16, sensor kecepatan, tata cara pemrograman, serta melakukan berbagai pencarian referensi dan sumber-sumber relevan lain sebagai acuan dalam pembuatan Tugas Akhir.

#### 2. Perancangan Sistem

Membuat perancangan sistem yang meliputi desain hardware dan perancangan *software* dengan menggunakan *Mikrokontroler* Arduino, driver motor L298N, motor DC, LCD 2x16, sensor kecepatan serta melakukan berbagai pencarian referensi dan sumber-sumber relevan lain sebagai acuan dalam pembuatan Tugas Akhir.

#### 3. Alat dan bahan

Menuliskan komponen yang dibutuhkan pada penyusunan tugas akhir dan penelitian beserta spesifikasi komponen.

#### 4. Pengujian dan Analisis

Melakukan pengujian sistem secara keseluruhan baik perangkat keras maupun perangkat lunak dengan cara mengoperasikan dan mengkomunikasikan perangkat lunak di masing-masing bagian, kemudian menjalankan perangkat keras sesuai dengan perintah perangkat lunak dan data dari sensor.



## Perancangan Sistem

Perancangan sistem yang melibatkan Arduino mencakup perancangan *hardware* dan *software*. Berikut adalah langkah-langkah umum untuk perancangan sistem proteksi dan *monitoring revolution per minute* antena Radar berbasis arduino guna meningkatkan *safety awarness* alutsista TNI AU:

1. Perancangan *Hardware*
2. Perancangan *Software*

## Perancangan *Hardware*

Perancangan *hardware* merupakan proses merancang dan mengembangkan komponen fisik berupa mikrokontroler Arduino, sensor kecepatan, potensiometer, relay, motor DC dan driver motor L298N atau perangkat elektronik lainnya. Dalam perancangan *hardware* membutuhkan pemilihan komponen, desain sirkuit, pengukuran, simulasi dan pengujian untuk memastikan bahwa *hardware* dapat beroperasi sesuai dengan kebutuhan dan spesifikasi yang ditentukan.

## Perancangan *Software*

Perancangan perangkat lunak digunakan untuk mengintegrasikan seluruh perangkat keras dan menjadi pusat kendali dari alat. Mikrokontroler perlu di program terlebih dahulu agar dapat bekerja sesuai dengan kebutuhan. Program yang dibuat bertujuan mempermudah pengguna dalam menggunakan alat. Perangkat lunak dirancang menggunakan *software* Arduino IDE. Aplikasi yang dibutuhkan adalah aplikasi Arduino IDE (*Integrated Development Environment*) karena aplikasi ini mudah dalam membuat fungsi-fungsi logika dasar dan mudah di mengerti karena menggunakan Bahasa C, selain *software* Arduino IDE untuk memasukkan program ke dalam Arduino, dibutuhkan driver USB, dan Arduino *board* nya sendiri agar program yang telah dibuat dapat berjalan.

## HASIL DAN PEMBAHASAN

### Pengukuran Tegangan

Pengukuran tegangan dilakukan terdiri dari pengukuran tegangan AC dan tegangan DC. Pengukuran tegangan ini bertujuan untuk mengukur tenaga/energi yang digunakan untuk menyuplai seluruh sistem yang ada agar dapat berjalan dengan baik. Pengukuran dimulai dengan pengukuran pada *output* sumber tegangan 220 V AC menggunakan multimeter. Dari pengukuran yang dilakukan, didapatkan hasil bahwa *output* tegangan 220 V AC terukur sebesar 219 V AC (selisih kurang dari 1%), maka dapat dinyatakan bahwa tegangan bekerja dengan baik. Selanjutnya pengujian tegangan DC yang dilakukan untuk mengetahui tegangan yang digunakan sebagai catu daya utama pada Arduino komponen lainnya. Tegangan masukan dari catu daya utama sebesar 12 V DC dan diturunkan menjadi sekitar 5 V DC sebagai tenaga kerja Arduino dan komponen lain. Hasil pengujian ditunjukkan pada Tabel di bawah ini.

Tabel Pengujian Tegangan DC

No	Vin (DC Supply) (Volt)	Vin (Terukur di mul- timeter)	Vout (Pengukuran) (Volt)	Vout (tampil LCD) (Volt)	Error (%)
1	12,5	12,51	5,12	5,14	0,39

2	12,4	12,43	5,10	5,16	1,18
3	12,3	12,32	5,10	5,11	0,20
4	12,2	12,22	5,06	5,09	0,59
5	12,1	12,13	5,02	5,12	1,99
6	12	12,01	5,01	5,14	2,59
7	11,9	11,90	5,01	5,12	2,20
8	11,8	11,79	5,05	5,16	2,18
9	11,7	11,70	5,08	5,15	1,38
10	11,6	11,62	5,10	5,14	0,78
11	11,5	11,51	5,06	5,12	1,19
Rata-rata <i>error</i> (%)					1,33

Dari tabel hasil pengujian menunjukkan bahwa masih terjadi selisih nilai tegangan antara perhitungan dengan hasil pengukuran. Namun selisih perbedaan tegangan hasil pengukuran cenderung lebih tinggi dibandingkan dengan hasil perhitungan. Rata-rata tingkat kesalahan dari pengujian rangkaian pembagi tegangan yaitu kurang dari sama dengan 1,33%.

$$Error (\%) = \frac{Selisih\ Nilai\ Pembacaan}{Nilai\ Pengukuran} \times 100\%$$

Berdasarkan rumus di atas, hasil perhitungan yang didapat adalah sebagai berikut:

$$Error (\%) = \frac{5,14 - 5,12}{5,12} \times 100\%$$

$$Error (\%) = \frac{0,02}{5,12} \times 100\%$$

$$Error (\%) = 0,39 \%$$

### Pengukuran Potensiometer *Wirewound*

Pengukuran potensiometer untuk menguji ADC (*Analog-to-Digital Converter*) pada Arduino, berikut adalah langkah-langkah yang dapat dilakukan adalah sebagai berikut:

1. Hubungkan kaki tengah potensiometer ke pin analog pada Arduino (pin A0) Arduino.
2. Hubungkan kaki luar potensiometer ke tegangan referensi (5 Volt DC pada Arduino).
3. Hubungkan kaki terakhir dari potensiometer ke *ground* (Gnd) pada Arduino.
4. Pembuatan program Arduino untuk membaca nilai analog dari pin yang digunakan (A0) dan mencetak nilainya pada *Serial Monitor*.

```
void setup() {
  Serial.begin(9600); // Inisialisasi komunikasi serial
}
void loop() {
  int sensorValue = analogRead(A0); // Baca nilai analog dari pin A0
  Serial.println(sensorValue); // Cetak nilai pada Serial Monitor
  delay(1000); // Tunda sebentar sebelum membaca nilai berikutnya
}
```

Hasil yang didapatkan menyesuaikan dengan posisi potensiometer dan tegangan referensi yang digunakan. ADC pada Arduino memiliki rentang nilai digital 0 hingga 1023, yang sesuai dengan rentang tegangan referensi yang digunakan. Pada rangkaian ini menggunakan potensiometer 5k Ohm dengan tegangan referensi 5V, ADC akan mengubah tegangan analog dari potensiometer menjadi nilai digital antara 0 hingga 1023. Jika posisi potensiometer berada di tengahnya, maka kemungkinan nilai ADC sekitar 512 (setengah dari 1023). Ketika memutar potensiometer ke arah nol (putaran searah jarum jam), nilai yang dibaca oleh ADC akan mendekati 0. Sebaliknya, ketika Anda memutar potensiometer ke arah maksimum (putaran berlawanan jarum jam), nilai yang dibaca akan mendekati 1023. Tabel di bawah ini menggambarkan potensiometer 5k Ohm dengan tegangan referensi 5V dan ADC Arduino dengan rentang nilai 0-1023. Hasil pengujian nilai *resistance* dengan nilai ADC ditunjukkan pada Tabel di bawah ini.

Tabel Pengujian nilai *resistance* dengan nilai ADC

No	<i>Resistance</i> (Ohm)	Nilai ADC
1	0	0
2	1k	205
3	2k	410
4	3k	615
5	4k	820
6	5k	1022

### Pengujian LCD I2C

Pengujian LCD dilakukan dengan memprogram Arduino dengan sederhana menggunakan *software* Arduino IDE. Sebelumnya LCD sudah dihubungkan dengan I2C terlebih dahulu. Arduino dihubungkan dengan I2C untuk mengakses LCD, modul I2C digunakan untuk memberikan efisiensi pada penggunaan pin Arduino. Pengujian LCD dilakukan dengan mengatur kontras dari LCD dengan mengatur potensio dari I2C sampai karakter dari LCD dapat tampil dengan jelas. Berikut adalah tampilan pengujian LCD 16x2 dengan kode program menampilkan kata "Monitoring RPM Antena" ditunjukkan pada Gambar di bawah ini.



Pengujian LCD 16x2

Pada pengujian rangkaian penampil menggunakan LCD 16x2 dengan I2C yang berfungsi untuk menampilkan informasi pengukuran pada LCD. Pada rangkaian ini, pin yang digunakan adalah pin GND, VCC, SDA dan SCL. Hubungan rangkaian LCD pada sistem kecepatan putaran antena Radar sehingga dari hasil pengujian LCD yang digunakan sudah bekerja dengan baik.

## Pengujian Sensor Kecepatan

Pengukuran kecepatan dan untuk menguji sensor kecepatan menggunakan piringan *Rotary Encoder* dengan Arduino. Adapun langkah-langkah yang dilakukan adalah sebagai berikut:

1. Hubungkan pin A pada *Rotary Encoder* ke pin digital pada Arduino. Pin A yang digunakan sebagai input utama (pin A ke pin 2 pada Arduino).
2. Hubungkan pin B pada *Rotary Encoder* ke pin digital pada Arduino. Pin B yang digunakan sebagai input pembantu (pin B ke pin 3 pada Arduino.)
3. Hubungkan pin GND pada *Rotary Encoder* ke *ground* (GND) pada Arduino.
4. Hubungkan pin VCC pada *Rotary Encoder* ke tegangan referensi yang sesuai. Misalnya, jika *Rotary Encoder* bekerja dengan tegangan 5V, hubungkan pin VCC ke 5V pada Arduino.
5. Pasang *library Rotary Encoder* pada Arduino IDE. Anda dapat melakukannya dengan membuka menu "Sketch" -> "Include Library" -> "Manage Libraries" dan mencari "*Rotary Encoder*". Kemudian, pilih *library* yang tepat dan klik "Install".
6. Buat program Arduino untuk membaca sinyal dari *Rotary Encoder* dan mencetak kecepatan yang terdeteksi pada *Serial Monitor*.

Berikut adalah contoh program Arduino menggunakan *library Rotary Encoder* untuk menguji sensor kecepatan dengan piringan *Rotary Encoder*:

```
#include <RotaryEncoder.h>
const int pinA = 2;
const int pinB = 3;
RotaryEncoder Encoder(pinA, pinB);
void setup() {
  Serial.begin(9600); // Inisialisasi komunikasi serial
  Encoder.begin();
}
void loop() {
  Encoder.tick(); // Membaca sinyal Rotary Encoder
  if (Encoder.isRight()) {
    Serial.println("Putaran searah jarum jam");
  } else if (Encoder.isLeft()) {
    Serial.println("Putaran berlawanan jarum jam");
  }
  // Menghitung kecepatan (misalnya, putaran per menit)
  float speed = Encoder.getPulsesPerSecond() * 60.0;
  Serial.print("Kecepatan: ");
  Serial.print(speed);
  Serial.println(" RPM");
  delay(100); // Tunda sebentar sebelum membaca sinyal berikutnya
}
```

Setelah *upload* atau mengunggah program ini ke dalam *board* Arduino, maka langkah selanjutnya membuka *Serial Monitor* dengan *baud rate* 9600 di IDE Arduino. *Serial monitor* Arduino memuat informasi tentang arah putaran (searah atau berlawanan jarum jam) dan kecepatan (dalam putaran per menit) yang terdeteksi oleh *Rotary Encoder*. Berikut Tabel yang merupakan hasil pengukuran kecepatan dan korelasinya dengan tegangan output yang dihasilkan.

Tabel Hasil pengukuran kecepatan putaran antenna

No.	Tegangan Analog (Volt)	Kecepatan (Rpm)
1	0	0
2	0,5	0
3	1	0
4	1,5	0
5	2	0
6	2,5	2,8
7	3	4,2
8	3,5	5,4
9	4	6,1
10	4,5	6,5
11	4,9	7,1

Pengukuran kecepatan pada *Rotary Encoder* dan konversi tegangan menjadi kecepatan akan bervariasi tergantung pada perangkat *Rotary Encoder* yang digunakan dan implementasi program Arduino. Spesifikasi *Rotary Encoder* yang digunakan serta penyesuaian dalam program untuk mendapatkan tabel yang akurat dan sesuai dengan pengukuran. Kecepatan putaran dapat dihitung dengan membagi jumlah pulsa dengan interval waktu yang telah ditentukan. Interval waktu ini biasanya diukur dalam detik atau milidetik sehingga dapat melaksanakan perhitungan kecepatan dalam putaran per menit (Rpm) atau putaran per detik (Hz).

### Pengujian Relay

Pengujian *relay* ini bertujuan untuk mengetahui apakah koil *relay* dan kontak *relay* berfungsi atau tidak. Adapun langkah-langkah yang dilaksanakan pada saat pengujian *relay* adalah sebagai berikut:

1. Hubungkan pin IN pada relay ke pin digital pada Arduino (pin IN ke pin 9 pada Arduino).
2. Hubungkan pin VCC pada *relay* ke tegangan referensi yang sesuai (pin VCC ke 5V pada Arduino).
3. Hubungkan pin GND pada relay ke *ground* (GND) pada Arduino.
4. Buat program Arduino untuk mengendalikan *relay*.

```
const int relayPin = 9;

void setup() {
  pinMode(relayPin, OUTPUT);
}

void loop() {
  digitalWrite(relayPin, HIGH); // Menghidupkan relay
  delay(1000); // Tunda selama 1 detik

  digitalWrite(relayPin, LOW); // Mematikan relay
  delay(1000); // Tunda selama 1 detik
}
```

Setelah mengunggah program ini ke Arduino, maka *relay* berubah antara keadaan aktif (ON) dan non-aktif (OFF) setiap detik. Berikut Tabel berisi hasil pengujian *relay*.

Tabel hasil pengujian *relay*

No	Kondisi <i>Relay</i>	Kontrol Pin	Kontak	Hasil
1	Non-aktif	<i>LOW</i>	NO Open	Tidak Terhubung
2	Aktif	<i>HIGH</i>	NO Tertutup	Terhubung
3	Non-aktif	<i>LOW</i>	NC Tertutup	Terhubung
4	Aktif	<i>HIGH</i>	NC Open	Tidak Terhubung

Keterangan:

- Kondisi *Relay*: Menunjukkan apakah *relay* dalam keadaan aktif (diaktifkan) atau non-aktif (tidak diaktifkan).
- Kontrol Pin: Nilai yang diberikan pada pin kontrol (IN) untuk mengendalikan *relay*. Misalnya, HIGH (tinggi) atau LOW (rendah).
- Kontak: Status kontak *relay*, NO terbuka (tidak terhubung) atau tertutup (terhubung), atau NC tertutup (terhubung) atau terbuka (tidak terhubung).
- Hasil: Hasil yang diharapkan dari pengujian, apakah ada aliran listrik (terhubung) atau tidak ada aliran (tidak terhubung) tergantung pada kondisi *relay* dan kontak sakelar.

Pengujian ini dilakukan dengan cara memasukkan perintah untuk menyalakan *relay* sehingga dengan begitu dapat terlihat apakah *relay* yang digunakan berfungsi dengan baik atau tidak. Batas nilai yang diberikan untuk kecepatan motor di atas limitasi atau di bawah limitasi. Perintah yang diberikan yaitu apabila kecepatan motor lebih besar atau sama dengan limitasi dan kecepatan motor lebih kecil atau sama dengan limitasi maka *relay* akan aktif/ON. Sedangkan apabila kecepatan motor berada di dalam limitasi maka *relay* akan OFF/tidak aktif.

Tabel Hasil Pengujian *Relay* dengan Kecepatan Limitasi

No.	Kecepatan (Rpm)	Kondisi <i>Relay</i>	Keterangan
1	4	OFF	Baik
2	4,5	OFF	Baik
3	5	OFF	Baik
4	5.5	OFF	Baik
5	6	OFF	Baik
6	6.5	OFF	Baik
7	7	ON	Baik
8	7.5	ON	Baik
9	8	ON	Baik
10	8.5	ON	Baik

Data yang dihasilkan pada Tabel di atas menunjukkan bahwa *relay* bekerja dengan baik dimana saat kecepatan di luar limitasi maka *relay* dapat bekerja untuk menonaktifkan *power* motor.

### Pengujian Sistem Monitoring

Pengujian sistem secara keseluruhan dilakukan dengan cara pengujian pembacaan sensor kecepatan dan memberikan input nilai kecepatan lebih dari limitasi sistem. Seluruh subsistem yang telah diuji secara terintegrasi dengan tujuan memastikan seluruh subsistem bekerja sesuai dengan parameter yang telah ditentukan.

Pengujian pertama dilakukan dengan memberikan power pada *driver motor* sehingga motor akan berputar dan sensor kecepatan akan membaca hasil putaran. Selanjutnya mikrokontroler akan mengolah data tersebut dan menampilkan pada *liquid cristal display* (LCD). Berikut Tabel merupakan hasil pengujian pembacaan sistem secara keseluruhan.

Tabel Hasil Pengujian Sistem Secara Keseluruhan.

No.	Analog	Voltage	Rpm	Cut Off	Keterangan
1	614	3,0	4,0	-	<i>In tolerance</i>
2	655	3,2	4,5	-	<i>In tolerance</i>
3	737	3,6	5,1	-	<i>In tolerance</i>
4	778	3,8	5,6	-	<i>In tolerance</i>
5	819	4,0	6,0	-	<i>In tolerance</i>
6	860	4,2	6,3	√	<i>Out of tolerance</i>
7	901	4,4	6,7	√	<i>Out of tolerance</i>
8	983	4,8	7,0	√	<i>Out of tolerance</i>
9	1022	5,0	7,2	√	<i>Out of tolerance</i>

Berdasarkan hasil pengujian di atas, seluruh sistem bekerja normal sesuai dengan fungsinya. Sehingga dapat disimpulkan bahwa sistem sudah mencapai tujuan dengan pemenuhan seluruh parameter subsistem.

### Pengujian ESP32

Pengujian ESP32 adalah proses verifikasi dan evaluasi kualitas serta kinerja jaringan ESP32. Tujuan pengujian ini adalah untuk memastikan bahwa jaringan ESP32 berfungsi dengan baik, memenuhi standar yang ditetapkan, dan dapat mendukung transfer data yang stabil dan efisien. Perangkat yang terhubung ke jaringan ESP32 dapat saling terhubung dan berkomunikasi dengan baik. Ini melibatkan memeriksa kabel, konektor, dan *switch* serta *Internet Protocol Address* (IP Address) yang digunakan dalam jaringan. Adapun langkah-langkah pengujian komunikasi ESP32 adalah sebagai berikut:

1. Buka *Command Prompt* pada komputer.
2. Di *Command Prompt*, ketikkan perintah "ipconfig". Perintah ini akan menampilkan informasi tentang koneksi jaringan pada komputer dan menampilkan informasi seperti "IPv4 Address", "Subnet Mask", dan "Default Gateway" yang terkait dengan koneksi ESP32.
3. *Setting* alamat IP Arduino *board* berada dalam subnet yang sama dengan komputer. Jika alamat IP Arduino *board* tidak sesuai, dapat diubah dengan mengedit program Arduino dan memberikan alamat IP yang sesuai menggunakan metode seperti *ESP 32.begin()*. Setelah mengubah program Arduino, *unplug* dan *replug* Arduino *board* untuk menerapkan perubahan.

4. Setelah alamat IP Arduino *board* dan komputer berada dalam *subnet* yang sama, dan pengujian koneksi ESP32 terhubung ke server *database* menggunakan alamat IP yang benar dan *port* yang sesuai.

Adapun hasil dari pengujian ESP32 yang telah dilakukan seperti pada Tabel di bawah ini adalah sebagai berikut:

Tabel Hasil Pengujian ESP32

No	Jenis Pengujian	Parameter yang diukur	Hasil	Keterangan
1	Konektivitas	Ketersediaan jaringan	Berhasil	perangkat terhubung
2	<i>Latensi</i>	Waktu Tiba Paket Data	3 ms	Rendah dan responsif
3	<i>Jitter</i>	Variasi Waktu Tiba Paket	2 ms	Stabil dan minimal
4	<i>Throughput</i>	Kecepatan Transfer Data	95 Mbps	Sesuai dengan ekspektasi

Keterangan:

1. Jenis Pengujian: Jenis pengujian yang dilakukan, seperti konektivitas, *throughput*, *latensi*, *jitter*, dan lainnya.
2. Parameter yang Diukur: Parameter atau metrik yang diukur dalam pengujian tersebut, seperti kecepatan, waktu tiba paket data, dll.
3. Hasil Pengujian: Hasil yang diperoleh dari pengujian, seperti nilai kecepatan, waktu tiba, dan sebagainya.
4. Keterangan: Catatan atau keterangan tambahan tentang hasil pengujian atau hal-hal yang perlu diperhatikan dari pengujian tersebut.

Dalam tabel ini, dapat terlihat hasil dari setiap pengujian yang telah dilakukan, jaringan ESP32 beroperasi sesuai harapan dan memenuhi kriteria kualitas yang diinginkan. Dalam tabel tersebut dapat diperoleh informasi yang mudah dipahami dan dapat digunakan untuk evaluasi performa jaringan serta perbaikan atau penyesuaian yang mungkin diperlukan jika ada parameter yang tidak memenuhi standar yang diinginkan.

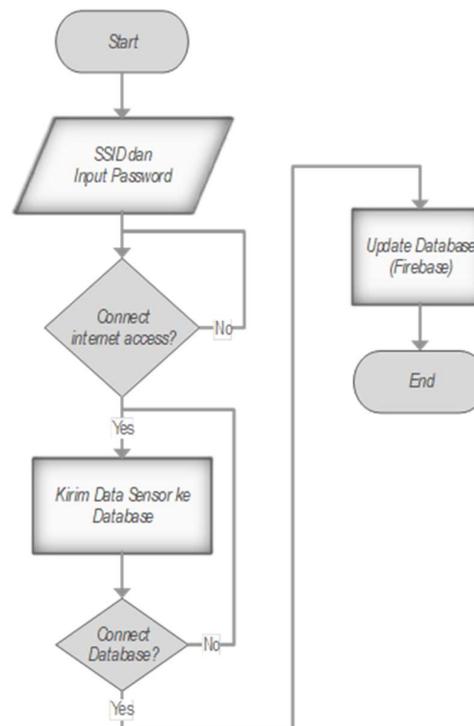
### Pengujian Software

Pengujian *software* yang dikembangkan untuk perancangan sensor kecepatan menggunakan papan Arduino. Pengujian bertujuan untuk memverifikasi dan memvalidasi fungsionalitas *software* serta memastikan bahwa sensor kecepatan berfungsi sesuai dengan spesifikasi yang ditetapkan. Selain itu, pengujian ini bertujuan untuk menguji batas kehandalan, *error* pada sistem, dan performa sistem. Adapun pengujian *software* mencakup pengujian unit, pengujian pengolahan data, pengujian integrasi, dan pengujian fungsional.

Perancangan sistem yang digunakan pada sistem *monitoring* kecepatan putaran antenna secara *real-time* yaitu dengan memanfaatkan teknologi berbasis *Internet of things*. Sesuai dengan teori *Internet of things* (IoT) menurut Yudho Yudha Yudhanto adalah sebuah suatu konsep dimana konektivitas internet dapat bertukar informasi satu sama lainnya dengan benda-benda yang ada disekelilingnya. *Internet of things* dalam sistem informasi manajemen *safety* dan *emergency* yang merupakan pengembangan teknologi dengan mengintegrasikan sensor-sensor input, mikrokontroler dan *software* dalam berkomunikasi, menghubungkan, bertukar data sensor input yang berupa

kecepatan. Adapun unsur-unsur yang digunakan dalam *internet of things* (IoT) meliputi sensor, kecerdasan buatan (mikrokontroler ESP32), koneksi jaringan (wi-fi) internet dan perangkat pendukung yang dapat meningkatkan ketepatan, kecepatan dan performa dari IoT. Keberadaan IoT dapat membantu dalam memonitor dan mengelola data kecepatan putaran secara *real-time*. Sebelum menuliskan program pada ESP32 terlebih dahulu meng-*install driver* ESP32 agar mikrokontroler ESP32 dapat dikenal dan *syntax* program yang ditulis bisa masuk ke dalam mikrokontroler ESP32. Selanjutnya menginstall *library* dari *firebase* menuliskannya pada Arduino IDE, mendeklarasikan alamat *database* dan juga kode autentifikasinya serta koneksi dengan *database server* melalui jaringan internet. Adapun *flowchart* koneksi ESP32 dengan *web server (firebase)* dapat dilihat pada Gambar di bawah ini.

- a. *Flowchart* Koneksi ESP32 dengan *Software Firebase* melalui Jaringan wi-fi untuk mengakses internet.



*Flowchart* Koneksi ESP32 dengan *Web Server (Firebase)*.

- b. Program pengiriman data sensor input ke dalam *database*. *Database* atau basis data merupakan kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. Dalam pembuatan *database* diperlukan *mysql* yang menghubungkan *script Personal Home Pages (PHP)* menggunakan perintah *query*. Selain itu, juga diperlukan *Phpmyadmin* yang digunakan untuk mengelola *database mysql* dengan *graphic user interface (GUI)* yang memudahkan dari antarmuka pengelolaan asli *mysql* yang berbasis teks. Berikut merupakan program singkat tentang pengiriman data sensor input ke dalam *database (firebase)*.



```
#include <Firebase_ESP_Client.h>
//Provide the token generation process info.
#include "addons/TokenHelper.h"
//Provide the RTDB payload printing info and other helper functions.
#include "addons/RTDBHelper.h"
// Insert your network credentials
#define WIFI_SSID "Sathar 22"
#define WIFI_PASSWORD "12345678"
// Insert Firebase project API Key
#define API_KEY "AlzaSyAaCnrWskGjHIPPWmXc217WDr3jxcrvnrA"
// Insert RTDB URL define the RTDB URL */
#define DATABASE_URL "https://rpmRadar-default-rtdb.asia-southeast1.firebaseio.com/"
#define USER_EMAIL "user@rpmRadar.com"
#define USER_PASSWORD "rpmRadar"
void setup()
{
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  /* Assign the api key (required) */
  config.api_key = API_KEY;

  /* Assign the RTDB URL (required) */
  config.database_url = DATABASE_URL;
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;

  /* Assign the callback function for the long running token generation task */
  config.token_status_callback = tokenStatusCallback; //see
  addons/TokenHelper.h

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
  if (Firebase.ready() && (millis() - tmLast > 3000)){
    sensor.update();
    if (Firebase.RTDB.setFloat(&fbdo, "room/kecepatan",
    publishData.temp))}
```

```

digitalWrite(PIN_LED, HIGH);
delay(100);
digitalWrite(PIN_LED, LOW);
delay(100);
Serial.println("PASSED");
Serial.println("PATH: " + fbdo.dataPath());
Serial.println("TYPE: " + fbdo.dataType());
}
else {
Serial.println("FAILED");
Serial.println("REASON: " + fbdo.errorReason());
}

```

- c. Program *website* yang digunakan pada perancangan sistem monitoring kecepatan.

```

DatabaseReference refkecepatan =
    FirebaseDatabase.instance.ref("room/kecepatan");
double datakecepatan = 0;
void updatekecepatan(double data) {
    datakecepatan = data;
    print("temp: ${datakecepatan}");
}
}

```

- d. Pengujian koneksi *web server* sebagai sistem *Internet of Things* (IoT). Tabel berikut ini merupakan hasil pengujian koneksi pengiriman data input ke *database* yang telah dilakukan.

Tabel Pengujian Koneksi pengiriman Data Input ke *Database*

No.	Data	Penerimaan	Keterangan
1	Kecepatan	Terdeteksi	Terkirim
2	Kecepatan limit	Terdeteksi	Terkirim

### Pengujian Unit

Pengujian unit dilakukan untuk memastikan bahwa setiap fungsi dan modul dalam *software* beroperasi dengan benar. Pada perancangan sensor kecepatan ini, beberapa fungsi yang diuji meliputi:

1. Pengujian Konfigurasi Sensor. Pengujian ini memastikan bahwa sensor kecepatan telah dikonfigurasi dengan benar pada perangkat keras Arduino. Ini melibatkan memastikan pin *input/output* yang benar telah ditentukan, dan sensor diinisialisasi dengan pengaturan yang tepat.
2. Pengujian Pembacaan Sensor. Pengujian ini dilakukan untuk memverifikasi apakah fungsi pembacaan sensor berfungsi dengan benar. Data *dummy* sebagai simultan dapat digunakan untuk mensimulasikan sinyal sensor, dan hasil pembacaan yang diperoleh dibandingkan dengan nilai yang diharapkan.



3. Verifikasi Koneksi Fisik. Pastikan bahwa sensor kecepatan terhubung secara fisik dengan papan Arduino dengan benar. Periksa penggunaan pin *input/output* yang sesuai pada Arduino dan pastikan kabel atau konektor yang digunakan terpasang dengan baik.
4. Pemeriksaan Ketersediaan *Library*. Memastikan bahwa *library* tersebut telah di-*instal* dengan benar dalam lingkungan pengembangan Arduino IDE. Verifikasi versi *library* dan kompatibilitasnya dengan papan Arduino yang digunakan.
5. Inisialisasi Sensor. Tulis kode program yang bertanggung jawab untuk menginisialisasi sensor kecepatan, LCD dan motor DC. Pada tahap ini, pastikan bahwa pin *input/output* yang digunakan oleh sensor telah ditentukan dengan benar dalam kode.
6. Pengujian Komunikasi. Uji komunikasi antara Arduino dengan sensor kecepatan, LCD dan motor DC. Memastikan bahwa data dapat dikirim dan diterima dengan benar antara Arduino dan sensor serta tampil pada LCD.
7. Pengujian Konfigurasi Sensor. Verifikasi pengaturan dan konfigurasi sensor kecepatan yang diperlukan. Ini termasuk pengaturan parameter seperti *baud rate*, mode pengukuran, filter sinyal, atau parameter lain yang relevan.
8. Uji Fungsionalitas Dasar. Setelah konfigurasi sensor selesai, lakukan pengujian fungsionalitas dasar untuk memastikan bahwa sensor kecepatan bekerja dengan benar.

### Pengujian Pengolahan Data

Fungsi pengolahan data yang digunakan untuk menghitung jumlah putaran roda atau mengonversi data sensor menjadi kecepatan, pengujian ini memastikan bahwa fungsi-fungsi tersebut beroperasi dengan benar. Input data yang diketahui dan nilai yang diharapkan digunakan untuk memverifikasi keakuratan pengolahan data. Pada perancangan sensor kecepatan ini, beberapa pengujian pengolahan data meliputi:

1. Fungsi Baca Sensor(). Pengujian dilakukan untuk memverifikasi fungsi ini dapat membaca data dari sensor kecepatan dengan akurasi yang memadai. Sejumlah *input dummy* diberikan untuk mensimulasikan pembacaan sensor, dan *output*-nya dibandingkan dengan nilai yang diharapkan.
2. Fungsi Hitung Kecepatan(). Pengujian dilakukan untuk memverifikasi fungsi ini dapat menghitung kecepatan berdasarkan data yang diterima dari sensor. Beberapa kombinasi nilai sensor diberikan sebagai *input*, dan *output*-nya diverifikasi dengan hasil yang diharapkan.
3. Pengujian Konversi Sinyal. Konversi sinyal dari sensor kecepatan ke nilai kecepatan yang sesuai telah dilakukan dengan benar.
4. Pengujian Perhitungan Kecepatan. Pengujian ini bertujuan untuk memastikan bahwa perhitungan kecepatan berdasarkan data sensor dilakukan dengan akurat.
5. Pengujian Tampilan pada LCD. Modul LCD untuk menampilkan kecepatan kendaraan, lakukan pengujian tampilan pada LCD. Verifikasi nilai kecepatan yang dihasilkan dari pengolahan data ditampilkan dengan benar pada LCD. Pengujian terhadap tampilan angka kecepatan dan pengukuran sesuai dengan yang diharapkan, dan memastikan bahwa tampilan aktual pada LCD sesuai dengan nilai yang dihasilkan.
6. Pengujian Pembaruan Tampilan. Pengujian nilai kecepatan dapat *update*/pembaruan tampilan pada LCD. Perubahan nilai kecepatan secara bertahap dan mengamati tampilan pada LCD dapat meng-*update* tampilan kecepatan secara tepat waktu.
7. Pengujian Konsistensi Tampilan. Pengujian untuk memastikan konsistensi tampilan pada LCD dengan memberikan variasi nilai kecepatan sebagai input dan mengamati tampilan pada LCD tetap konsisten dan tidak terjadi lonjakan atau fluktuasi yang tidak semestinya. Pada tahap ini berfungsi untuk memastikan bahwa tampilan pada LCD mencerminkan kecepatan yang dihasilkan secara stabil dan akurat.

Pengujian Integrasi Sensor dan LCD. Sensor kecepatan dan modul LCD dapat bekerjasama dalam sistem yang kompleks, integrasi yang baik antara keduanya. Pengujian ini berfungsi untuk melaksanakan verifikasi nilai kecepatan yang dihasilkan oleh sensor dapat ditransmisikan dan ditampilkan dengan benar pada LCD.

### **Pengujian Integrasi**

Pengujian integrasi dilakukan untuk memastikan bahwa komponen *software* berinteraksi dengan baik satu sama lain. Pada perancangan sensor kecepatan ini, pengujian integrasi melibatkan pengujian komunikasi antara papan Arduino dan sensor kecepatan serta LCD. Beberapa pengujian yang dilakukan meliputi:

1. Komunikasi Serial. Pengujian dilakukan untuk memverifikasi papan Arduino dapat mengirim dan menerima data secara benar melalui komunikasi serial dengan sensor kecepatan seperti komunikasi serial atau komunikasi melalui bus I2C. Data *dummy* dikirim dari sensor kecepatan ke Arduino, dan Arduino kemudian mengirimkan data balasan. Keakuratan komunikasi dan keandalan data dievaluasi.
2. Pengujian Sinkronisasi. Mensinkronkan perangkat keras lain yang terlibat dalam sistem seperti modul potensiometer dan motor DC. Pengujian ini berfungsi untuk memastikan bahwa komunikasi dan sinkronisasi antara Arduino dan perangkat keras tambahan berjalan dengan baik. Fungsi-fungsi yang berkaitan dengan interaksi antara perangkat keras diuji untuk memverifikasi kinerjanya.
3. Pengujian Kesesuaian Pin. Pengujian kesesuaian terhadap pengaturan pin *input/output* pada Arduino sesuai dengan kebutuhan komponen tambahan, seperti sensor kecepatan, modul LCD dan *driver motor* serta potensiometer. Hal ini bertujuan untuk memastikan kesesuaian pin yang digunakan untuk komunikasi dan pengendalian, serta pastikan tidak terjadi bentrok penggunaan pin yang dapat mengganggu kinerja sistem.
4. Pengujian Kesesuaian Protokol. Protokol komunikasi I2C memerlukan kesesuaian pengaturan protokol tersebut antara Arduino dan LCD seperti pengaturan *baud rate*, alamat perangkat, atau pengaturan protokol lainnya telah dikonfigurasi dengan benar untuk memastikan komunikasi yang sukses antara komponen-komponen tersebut.
5. Pengujian Kinerja Keseluruhan. Pengujian kinerja keseluruhan sistem sensor kecepatan dengan komponen-komponen yang terintegrasi untuk mengetahui komunikasi dan interaksi antara komponen berjalan dengan baik dalam kondisi nyata. Evaluasi respons sistem, waktu pembaruan tampilan, dan konsistensi kinerja keseluruhan.

### **Pengujian Fungsional**

Pengujian fungsional dilakukan untuk memverifikasi terhadap perangkat lunak sensor kecepatan memenuhi persyaratan fungsional yang telah ditetapkan. Pada pengujian ini, perangkat keras yang digunakan papan Arduino, sensor kecepatan, dan modul LCD. Metode pengujian yang digunakan meliputi pengujian unit, pengujian integrasi, dan pengujian fungsional. Hasil dari pengujian tersebut akan digunakan untuk mengevaluasi performa dan keandalan *software* dalam mendeteksi kecepatan kendaraan dengan akurat dan memberikan peringatan yang sesuai. Beberapa pengujian fungsional yang dilakukan pada perancangan sensor kecepatan ini meliputi:

1. Pengujian Penghitungan Kecepatan. Pengujian dilakukan untuk memastikan bahwa sensor kecepatan dan *software* secara akurat menghitung kecepatan putaran yang dihasilkan.
2. Pengujian Pendeteksian Kecepatan Maksimum. Pengujian dilakukan untuk memverifikasi bahwa perangkat lunak mampu mendeteksi dan memberikan peringatan saat putaran melampaui batas kecepatan maksimum yang ditentukan. Putaran antenna



- dengan kecepatan yang bervariasi digunakan sebagai *input*, dan respons perangkat lunak dievaluasi.
3. Pengujian Deteksi Kecepatan. Pengujian ini bertujuan untuk memverifikasi sensor kecepatan dan *software* mampu mendeteksi dan mengukur kecepatan putaran dengan akurat.
  4. Pengujian Peringatan Kecepatan. Pengujian ini dimulai apabila sistem sensor kecepatan melampaui batas kecepatan tertentu. Pengujian ini memastikan bahwa peringatan diberikan secara tepat waktu dan sesuai.
  5. Pengujian Stabilitas. Pengujian ini bertujuan untuk memastikan bahwa *software* sensor kecepatan stabil dan dapat mengatasi situasi yang mungkin tidak terduga, seperti fluktuasi sinyal sensor atau gangguan lingkungan. Pengujian dilakukan dengan memberikan input yang beragam dan memantau kinerja *software* serta responsnya.
  6. Pengujian Keandalan. Pengujian ini mengevaluasi keandalan *software* sensor kecepatan dalam jangka waktu yang lebih lama. *Software* dijalankan dalam kondisi nyata dan dipantau untuk melihat apakah ada masalah yang muncul seiring berjalannya waktu atau penggunaan yang berulang.

## PENUTUP

### Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, untuk mewujudkan sistem monitoring kecepatan putaran antena Radar dapat dilaksanakan melalui strategi dan upaya yang telah disusun untuk mengatasi pokok-pokok persoalan, sehingga implementasi sistem monitoring dapat terlaksana dengan baik dan mampu mendukung tugas TNI Angkatan Udara. Adapun kesimpulan yang dapat dibuat sebagai jawaban dari pokok-pokok persoalan adalah sebagai berikut:

1. Dengan adanya sistem proteksi dan monitoring revolution per minute antena radar, mampu meningkatkan safety awareness Satuan Radar TNI Angkatan Udara dengan melakukan shutdown secara otomatis apabila terjadi putaran antena yang diluar limitasi.
2. Berdasarkan data pengujian dan analisa yang telah dilakukan, sensor *Rotary Encoder* dapat digunakan untuk mengetahui kecepatan putaran antena Radar, sehingga kecepatan dapat dipantau dan tampil di liquid cristal display (LCD).
3. Implementasi sistem proteksi dan monitoring kecepatan antena yang terdiri atas beberapa subsistem seperti input, proses dan output yang berfungsi untuk mengetahui kecepatan dan sebagai sistem proteksi pada mekanik gearbox antena Radar. Apabila kecepatan melebihi batas toleransi (limitasi) maka putaran antena akan reduce speed hingga berhenti.

### Saran

Dalam mewujudkan sistem monitoring kecepatan putaran antena Radar berbasis *internet of things* (IoT), maka diajukan beberapa saran sebagai berikut:

1. Dibutuhkan peran dari Dislitbangau yang dalam hal ini sebagai Lembaga penelitian dan pengembangan terkait alutsista TNI AU. Diharapkan dengan adanya peran Dislitbangau menjadikan sistem yang dibuat menjadi lebih sempurna dan optimal sesuai dengan standar militer yang telah ditetapkan oleh TNI AU. Dalam hal ini penerapan sistem IoT dengan menggunakan aplikasi berbasis *android* sehingga memudahkan personel mengakses data menggunakan *smartphone*.



2. Kerja sama dengan Dislambangjaau terkait pemenuhan parameter-parameter standar *safety* yang sesuai dengan lingkungan kerja TNI AU khususnya Satuan Radar terkait dengan pengoperasian alat tersebut.
3. Koordinasi dengan Puslaik TNI AU selaku *stakeholder* dalam penerbitan dan pengesahan sertifikasi suatu alat pendukung alutsista TNI AU, yang mana berguna untuk legalitas alat tersebut di dalam pengoperasian pada Alutsista TNI AU.
4. Kerja sama dengan Disinfohtaau untuk memberikan dukungan *database* guna menyimpan data kecepatan putaran antenna secara ter-*enkripsi*.

## REFERENSI

- Arduino, 14 Februari 2023. [Online]. Available: <https://www.arduino.cc/en/Main/Software> .
- Arif Rakhman Suharso, A., Muhamad Fauzi, R., Dwi Kurniawan, A., Renaldo, A., & Hartono. (2016). PERANCANGAN SISTEM RADAR PENDETEKSI OBJEK MENGGUNAKAN SENSOR ULTRASONIK BERBASIS ARDUINO. *Jurnal Rekayasa Teknologi Nusa Putra*, 3(1), 20-26.
- B. Adrian, "Menunggu Realisasi Pengadaan Radar, KSAU Tinjau Satrad Jayapura yang Sudah Selesai Dibangun," 27 Agustus 2022. [Online]. Available: <https://mylesat.com/2022/08/27/menunggu-realisasi-pengadaan-radar-ksau-tinjau-satrad-jayapura-yang-sudah-selesai-dibangun/>.
- Diarsyah Amarullah, Mochammad Djaohar, Massus Subekti, "PENGATURAN KECEPATAN MOTOR DC SERI BERBASIS ARDUINO UNO," *Journal of Electrical and Vocational Education and Technology*, pp. Vol.4, No.2, ISSN 2548-9178, 2019.
- G. W. Stimson, *Introduction to Airborne Radar* (2nd ed.), United Amerika Serikat: SciTech Publishing., 2013.
- J. & H. L. I. Toomay, *Radar: Principles, Technology, Applications.*, United Amerika Serikat: SciTech Publishing., 2012.
- J. A. S. Mark A. Richard, *Prinsiples of Modern Radar*, United States of America: SciTech Publishing, Edison, NJ, 2010.
- J. Utomo, *Rancang Bangun Pengendali Dan Monitoring*, Lampung: Fakultas Teknik Universitas Lampung, 2016.
- K. S. & J. Hylén, "Getting Started with Arduino IDE 2," 4 April 2023. [Online]. Available: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2>.
- L. SLM, "Lab Sheet Instrumentasi," Fakultas Teknik Universitas Negeri Yogyakarta, Yogyakarta, 2010.
- Lestari, *Optimasi Kontrol PID Pneumatik Control Valve Pada High Pressure Heater PLTU Pelabuhan Ratu Berbasis DCS MAXDNA*, Cimahi: UNJANI, 2016.
- M. Anggara, *Sistem Kendali PID Untuk Kecepatan Motor DC Berbasis Arduino Untuk Penguat Mosfet Dab Feedback Motor DC*, Bandung: POLITEKNIK NEGERI BANDUNG, 2017.
- M. F. Arif, "Sistem Kontrol Kecepatan Motor DC D-6759 Berbasis Arduino Mega 2560," *Makalah Seminar Universitas Brawijaya*, Malang, 2015.
- M. Margolis, *Arduino Cookbook*, United States of Amerika: O'reilly Media, Inc., 2011.
- Mabesau, *Doktrin TNI Angkatan Udara Swa Bhuwana Paksa*, Jakarta: Mabesau, 2019.
- Mabesau, *Surat Keputusan Kepala Staf TNI Angkatan Udara Nomor Kep/4/III/1999*, Jakarta: Mabes TNI Angkatan Udara, 1999.
- Muhammad Iqbal Rangkuti, Ardianto Pranata, Ishak Ishak, *Sistem Radar Untuk Monitoring Objek Sekitar Berbasis Internet Of Things (IOT)*



- R. A. Fatoni, Sistem Navigasi Robot Beroda Berbasis Teknik Kendali PID Menggunakan Mikro-kontroller Arduino Mega2560, Cimahi: UNJANI, 2017.
- Radi Birdayansyah, Noer Sudjarwanto, Osea Zebua, "Pengendalian Kecepatan Motor DC Menggunakan Perintah Suara Berbasis Mikrokontroler Arduino," Jurnal Rekayasa dan Teknologi Elektro, pp. Volume 9, No. 2, Mei 2015, 2015.
- Setneg, Undang-Undang Nomor 34 Tahun 2004 tentang Tentara Nasional Indonesia, Jakarta: Sekretariat Negara Republik Indonesia, 2004.
- Theta D Innarwaty Putri, Winarno Sugeng, Fuadi Ramdani, "Sistem Pembayaran Elektronik Pada Transportasi Angkutan Kota Menggunakan *Rotary Encoder*," MIND (Multimedia Artificial Intelligent Networking *Database*) Journal, pp. Vol 6 No. 1 Halaman 1-15, 2021.
- Tjahja Odianto, Bambang Suprijono, Winda Andrianta Widya Natasari, Perancangan Pengendali Kecepatan Motor Arus Searah 1 HP 220 Volt Dengan Metode PID Berbasis Mikrokontrol-ler, Surabaya: Universitas ITATS, 2015.
- Y. P. Haniah, "Pengenal Teknologi Radar Untuk Pemetaan Spasial di Kawasan Tropis," TEKNIK – Vol. 32 No.2 Tahun 2011, ISSN 0852-1697, p. Hal. 115, 2011.
- Y. Y. Yudhanto, Pengantar Teknologi Internet Of Things (IoT), Surakarta: UNS, 2020.