

## Rancang Bangun Sistem *Controlling Rpm* Pada *Main Engine* Berbasis *Arduino Uno* Guna Mencegah Terjadinya *Overspeed*

Reski Kurniawan , Agus Dwi Santoso , Upik Widyaningsih

Program Studi Diploma IV Teknologi Rekayasa Kelistrikan Kapal, Politeknik  
Pelayaran Surabaya

Alamat: Jl. Gunung Anyar Lor No.1, Gn. Anyar, Kec. Gn. Anyar, Surabaya, Jawa Timur  
60294

Korespondensi Penulis : [reskikurniawan2001@gmail.com](mailto:reskikurniawan2001@gmail.com)

### ABSTRACT

As an important means of sea transportation for the distribution of goods and human mobility between islands, ships require a reliable prime mover to avoid wear or damage, especially when overspeeding occurs which can result in fire. Therefore, a prototype was created that uses an IR Obstacle sensor to monitor the RPM speed of the rotating engine. By using Arduino Uno as a processor, this prototype is able to display the RPM value on the LCD and provide a signal in the form of a green light if the speed is below 4000 Rpm, as well as a red light and buzzer sound if overspeed occurs above 4000 Rpm, followed by the engine turning off after 15 seconds. Testing shows that this prototype has a high level of accuracy, with a difference between 0.08% to 2.73% and an average difference of 1.003%.

**Keywords:** RPM, overspeed, IR sensor, Arduino Uno.

### ABSTRAK

Sebagai alat transportasi laut yang penting untuk distribusi barang dan mobilitas manusia antarpulau, kapal memerlukan penggerak utama yang handal untuk menghindari keausan atau kerusakan, terutama saat terjadi overspeed yang dapat mengakibatkan kebakaran. Oleh karena itu, dibuatlah sebuah prototype yang menggunakan IR Obstacle sensor untuk memantau kecepatan RPM mesin berputar. Dengan menggunakan Arduino Uno sebagai prosesor, prototype ini mampu menampilkan nilai RPM pada LCD dan memberikan sinyal berupa lampu hijau jika kecepatan di bawah 4000 Rpm, serta lampu merah dan bunyi buzzer jika terjadi overspeed di atas 4000 Rpm, disusul dengan pemadaman mesin setelah 15 detik. Pengujian menunjukkan bahwa prototype ini memiliki tingkat keakuratan yang tinggi, dengan selisih antara 0,08% hingga 2,73% dan rata-rata selisih sebesar 1,003%.

**Kata kunci:** RPM, overspeed, sensor IR, Arduino Uno.

### PENDAHULUAN

Kapal laut menjadi transportasi penting dalam distribusi dan mobilitas, namun, mesin utama kapal sering mengalami kegagalan fungsi, termasuk dalam kasus *overspeed*. *Overspeed*, di mana kecepatan mesin utama melebihi batas yang ditetapkan, dapat menyebabkan kerusakan serius pada komponen internal. Untuk menghindari risiko ini, penggunaan alat pengaman seperti *overspeed travel* pada mesin utama diperlukan, tetapi terkadang hal ini tidak cukup. Oleh karena itu, diperlukan pengembangan *prototype* yang dapat memantau dan mengontrol kecepatan putaran mesin utama kapal. *Prototype* ini menggunakan sensor penghalang inframerah sebagai inputnya dan menghasilkan tampilan LCD yang menampilkan database putaran kecepatan mesin, memungkinkan pemantauan

konstan untuk mendeteksi *overspeed* dan mencegah kerusakan yang dapat membahayakan keselamatan.

Dengan menggunakan Arduino Uno sebagai CPU dan sensor IR *Speed* sebagai input, *prototype* ini memungkinkan para profesional untuk tetap waspada terhadap kondisi mesin, sehingga perawatan *preventif* atau perbaikan dapat dilakukan tepat waktu saat terjadi kelainan pada putaran mesin. Melalui pengembangan ini, diharapkan kapal-kapal dapat beroperasi dengan lebih aman dan efisien, serta mengurangi risiko kegagalan mesin yang dapat mengancam keselamatan dan keberlangsungan perjalanan laut.

## TINJAUAN PUSTAKA

### Sistem *Controlling* RPM

Sistem pengendalian RPM pada mesin adalah kunci untuk memantau dan mengatur putaran mesin, yang ditunjukkan dalam nilai RPM (*Revolution Per Minute*). Dengan menggunakan tachometer, baik dalam bentuk manual maupun digital, pengemudi dapat melihat peningkatan RPM yang sejalan dengan peningkatan kecepatan putaran mesin. Meskipun RPM dan kecepatan kendaraan tidak selalu proporsional, sensor RPM menjadi *prototype* penting yang memberikan batasan nilai energi untuk menentukan kondisi kerja mesin diesel, dengan putaran normal antara 1500 dan 1800 RPM. Sensor ini memberikan sinyal kepada aktuator untuk meningkatkan putaran RPM sesuai dengan beban yang diperlukan, sambil memastikan pasokan tegangan listrik di kapal memenuhi persyaratan untuk mencegah kerusakan pada sistem lainnya. ***Main Engine*** Mesin penggerak utama, atau yang sering disebut sebagai mesin induk, adalah perangkat vital dalam dunia kelautan yang menggerakkan kapal untuk melakukan pengangkutan muatan dari satu pelabuhan ke pelabuhan lainnya, baik itu manusia, cairan, komoditas padat, maupun gas. Mesin diesel, yang menggunakan susunan piston bolak-balik, mengubah energi panas dan tekanan dari mesin pembakaran internal menjadi energi mekanik melalui gerakan piston yang dikonversikan oleh poros engkol menjadi energi rotasi. Mesin utama, baik diesel maupun uap, adalah sumber utama tenaga yang menggerakkan kapal ke depan, dan merupakan komponen krusial dalam sistem penggerak kapal.

### Arduino Uno

**RANCANG BANGUN SISTEM CONTROLLING RPM PADA MAIN ENGINE  
BERBASIS ARDUINO UNO GUNA MENCEGAH TERJADINYA OVERSPEED**

Arduino Uno adalah *platform* mikrokontroler yang sumber terbuka dan mudah digunakan, didukung oleh *chip* ATmega328 sebagai basisnya. *Platform* ini dirancang untuk memfasilitasi pengembangan proyek interaktif dengan berbagai fitur yang mencakup osilator kristal 16 MHz, 6 input analog, 14 pin *input/output* digital (enam di antaranya dapat digunakan sebagai output PWM), *port* USB, konektor daya, *header* ICSP, dan tombol *reset*. Arduino Uno memiliki fleksibilitas untuk mendukung berbagai aplikasi, baik dengan baterai, konverter AC ke DC, atau koneksi USB ke komputer.



a) *Buzzer*

*Buzzer* adalah komponen listrik yang menghasilkan suara dengan mengubah getaran listrik. *Buzzer* beroperasi dengan prinsip yang serupa dengan pengeras suara, di mana arus listrik dialirkan melalui kumparan yang terhubung ke diafragma, yang kemudian bergerak maju mundur untuk menghasilkan suara.

b) *LCD (Liquid Crystal Display)*

LCD adalah jenis layar elektronik yang menggunakan teknologi logika CMOS untuk menampilkan data dalam bentuk karakter, huruf, atau grafik. Berbeda dengan layar yang menghasilkan cahaya sendiri, LCD memantulkan cahaya sekitarnya atau meneruskan cahaya dari lampu belakang untuk menampilkan informasi.

c) *Sensor IR Speed*

*Sensor IR Speed* adalah sensor yang digunakan untuk mengukur kecepatan putaran dengan memanfaatkan pantulan inframerah. Sensor ini menggunakan pemancar cahaya inframerah untuk mengirimkan sinyal, yang kemudian diterima oleh fototransistor setelah dipantulkan oleh objek atau penghalang, sehingga memungkinkan pengukuran kecepatan putaran.

d) *Dinamo*

Dinamo atau generator adalah alat listrik yang mengubah energi kinetik menjadi energi

listrik. Prinsip kerjanya mirip dengan generator, di mana putaran kumparan dalam medan magnet atau putaran magnet dalam kumparan menghasilkan aliran listrik.

e) *Relay*

*Relay* adalah perangkat elektronik yang menggunakan elektromagnet untuk mengoperasikan sekelompok kontak sakelar. Relay dapat mengubah arus listrik kecil menjadi arus listrik yang lebih besar dan berfungsi sebagai saklar yang dikendalikan oleh sinyal listrik.

f) *Tachometer*

*Tachometer* adalah alat yang digunakan untuk mengukur kecepatan putaran mesin dengan menghitung jumlah putaran dalam jangka waktu tertentu. Biasanya dilengkapi dengan panel yang menampilkan kecepatan putaran dalam satuan RPM (putaran per menit).

g) *Potensiometer*

*Potensiometer* adalah resistor tiga terminal yang menciptakan pembagi tegangan yang dapat disesuaikan. Potensiometer digunakan untuk mengatur peralatan listrik, seperti pengontrol volume pada amplifier atau sebagai pengontrol putaran pada dinamo dalam kasus ini.

### ***Overspeed***

*Overspeed*, yang terjadi ketika putaran mesin melampaui batas yang seharusnya, seringkali dipicu oleh masalah dalam sistem injeksi. Agus dan Eko (2015) menunjukkan bahwa peningkatan kecepatan berlebih disebabkan oleh jumlah bahan bakar yang berlebihan disemprotkan ke ruang bakar, mengakibatkan mesin berputar lebih cepat dari yang diperlukan dan menimbulkan getaran yang lebih tinggi. Dalam kondisi ideal, perhitungan teoritis jumlah bahan bakar yang harus disalurkan ke setiap silinder telah dilakukan oleh sistem pengapian pipa atau pompa nosel pada mesin diesel. Namun, dalam jangka waktu yang panjang, fenomena *overspeed* dapat merusak komponen *internal* mesin secara signifikan dan bahkan mengakibatkan risiko kebakaran yang serius. Langkah-langkah pencegahan yang dapat diambil untuk mencegah terjadinya

*overspeed* pada mesin induk kapal antara lain adalah melakukan pemeliharaan rutin dan terjadwal, memantau RPM mesin secara terus-menerus selama operasi, serta memastikan sistem kontrol mesin dan perangkat keamanan terpasang dengan baik dan berfungsi dengan benar. Selain itu, rumus RPM (*Revolution Per Minute*) digunakan untuk menghitung putaran

**RANCANG BANGUN SISTEM CONTROLLING RPM PADA MAIN ENGINE  
BERBASIS ARDUINO UNO GUNA MENCEGAH TERJADINYA OVERSPEED**

mesin dengan membagi 60 kali frekuensi putaran dengan jumlah kutub dalam satu putaran objek tersebut.

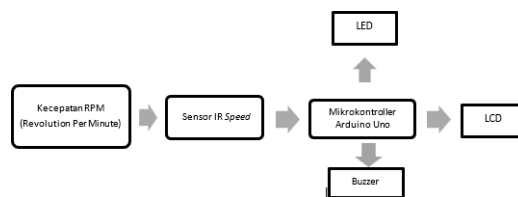
$$\text{RPM} = \frac{60 \times f}{p} \dots\dots\dots( 1 )$$

Keterangan:

- RPM adalah *Revolution Per Minute*.
- f adalah frekuensi putaran atau putaran per detik.
- p adalah jumlah kutub dalam satu putaran objek tersebut.

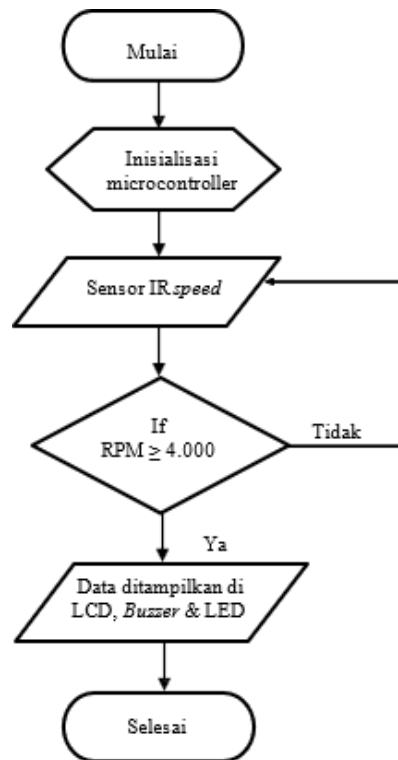
### **METODE PENELITIAN**

### **PERANCANGAN SISTEM**



Karya tulis ilmiah ini menggunakan metode penelitian *Research and Development (R&D)* dalam merancang sistem *controlling* RPM pada *main engine*. Sistem ini memanfaatkan IR *Obstacle* sensor sebagai *input*, Arduino Uno sebagai prosesor, dan menampilkan data RPM kecepatan putaran generator pada LCD saat terjadi *overspeed*. Diagram blok menunjukkan bagian-bagian utama sistem, seperti sensor IR, Arduino Uno, *buzzer*, LED, dan LCD. Rangkaian tersebut mencakup pemrograman mikrokontroler, sensor IR *speed*, dan LCD, dengan Arduino Uno sebagai pengontrol, sensor IR *speed* sebagai detektor kecepatan RPM, dan LCD sebagai tampilan data.

Alur kerja sistem ini memberikan panduan langkah-langkah yang sistematis dan jelas dalam pembuatan *prototype*, memudahkan pemahaman dan mengurangi risiko salah penafsiran informasi. Alur kerja tersebut tergambar dalam diagram pada gambar.



## PERANCANGAN *PROTOTYPE*

### Identifikasi Kebutuhan

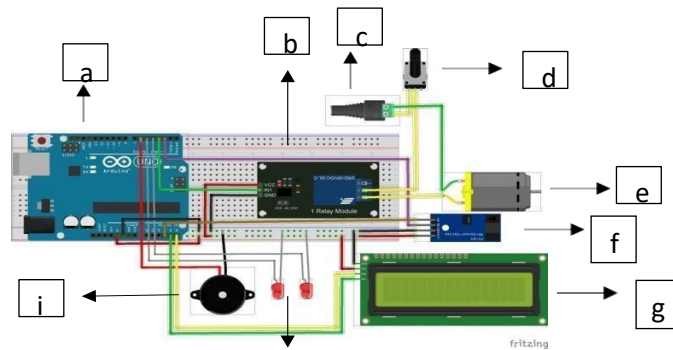
Berdasarkan desain sistem, maka kebutuhan dibagi menjadi dua yaitu kebutuhan *hardware* dan kebutuhan *Software*.

- 1) Kebutuhan *Hardware*:
  - a. Mikrokontroler    ArduinoUno
  - b. Sensor IR *Speed*
  - c. LCD
  - d. *Buzzer*
  - e. *Relay*
  - f. LED
  - g. Dinamo
  - h. Potensiometer
  - i. Tachometer
  - j. Adaptor

**RANCANG BANGUN SISTEM CONTROLLING RPM PADA MAIN ENGINE  
BERBASIS ARDUINO UNO GUNA MENCEGAH TERJADINYA OVERSPEED**

- k. Kabel *Jumper*
  - l. Kotak Akrilik
- 2) Kebutuhan *Software*:
- a. Software Arduino Uno
  - b. Rangkaian *elektronika*

Setelah mengetahui kebutuhan *prototype* penelitian melakukan rancangan *elektronika*. Rangkaian tersebut dapat dilihat pada gambar.



Pada gambar rancangan *elektronika*, komponen-komponen disusun sesuai dengan fungsinya:

- a. Arduino Uno berfungsi sebagai perangkat *open source* untuk merancang dan membuat perangkat elektronik dan *software*. *Relay* sebagai saklar utama ke komponen lainnya.
- b. *Socket Power* berfungsi sebagai penghantar listrik pada dinamo.
- c. Potensiometer digunakan untuk mengatur putaran pada dinamo.
- d. Dinamo berperan sebagai pengganti *main engine*.
- e. Sensor *IR Speed* digunakan untuk mendeteksi kecepatan RPM.
- f. LCD menampilkan data yang sudah dibaca oleh mikrokontroler.
- g. LED sebagai penanda sinyal, hijau menunjukkan kondisi normal dan merah menandakan *overspeed*.
- h. *Buzzer* berfungsi sebagai alarm yang akan berbunyi.

**Merancang Perangkat Lunak (*Software*)** Tahapan selanjutnya adalah merancang kode program menggunakan Arduino Uno. Program ini tidak memerlukan *chip programmer* karena sudah dilengkapi dengan *bootloader*. Komunikasi dengan komputer

menggunakan USB, memudahkan pengguna laptop yang tidak memiliki port serial atau RS323.

```

1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <LiquidCrystal_I2C.h>
4 #include <IRNonParasitic.h>
5 #include <IRremote.h>
6 #include <IRremoteESP8266.h>
7 #include <IRremote2.h>
8 #include <IRremote2ESP8266.h>
9 #include <IRremote2ESP8266.h>
10 #include <IRremote2ESP8266.h>
11 #include <IRremote2ESP8266.h>
12 #include <IRremote2ESP8266.h>
13 #include <IRremote2ESP8266.h>
14 #include <IRremote2ESP8266.h>
15 #include <IRremote2ESP8266.h>
16 #include <IRremote2ESP8266.h>
17 #include <IRremote2ESP8266.h>
18 #include <IRremote2ESP8266.h>
19 #include <IRremote2ESP8266.h>
20 #include <IRremote2ESP8266.h>
21 #include <IRremote2ESP8266.h>
22 #include <IRremote2ESP8266.h>
23 #include <IRremote2ESP8266.h>
24 #include <IRremote2ESP8266.h>
25 #include <IRremote2ESP8266.h>
26 #include <IRremote2ESP8266.h>
27 #include <IRremote2ESP8266.h>
28 #include <IRremote2ESP8266.h>
29 #include <IRremote2ESP8266.h>
30 #include <IRremote2ESP8266.h>
31 #include <IRremote2ESP8266.h>
32 #include <IRremote2ESP8266.h>
33 #include <IRremote2ESP8266.h>
34 #include <IRremote2ESP8266.h>
35 #include <IRremote2ESP8266.h>
36 #include <IRremote2ESP8266.h>
37 #include <IRremote2ESP8266.h>
38 #include <IRremote2ESP8266.h>
39 #include <IRremote2ESP8266.h>
40 #include <IRremote2ESP8266.h>
41 #include <IRremote2ESP8266.h>
42 #include <IRremote2ESP8266.h>
43 #include <IRremote2ESP8266.h>
44 #include <IRremote2ESP8266.h>
45 #include <IRremote2ESP8266.h>
46 #include <IRremote2ESP8266.h>
47 #include <IRremote2ESP8266.h>
48 #include <IRremote2ESP8266.h>
49 #include <IRremote2ESP8266.h>
50 #include <IRremote2ESP8266.h>
51 #include <IRremote2ESP8266.h>
52 #include <IRremote2ESP8266.h>
53 #include <IRremote2ESP8266.h>
54 #include <IRremote2ESP8266.h>
55 #include <IRremote2ESP8266.h>
56 #include <IRremote2ESP8266.h>
57 #include <IRremote2ESP8266.h>
58 #include <IRremote2ESP8266.h>
59 #include <IRremote2ESP8266.h>
60 #include <IRremote2ESP8266.h>
61 #include <IRremote2ESP8266.h>
62 #include <IRremote2ESP8266.h>
63 #include <IRremote2ESP8266.h>
64 #include <IRremote2ESP8266.h>
65 #include <IRremote2ESP8266.h>
66 #include <IRremote2ESP8266.h>
67 #include <IRremote2ESP8266.h>
68 #include <IRremote2ESP8266.h>
69 #include <IRremote2ESP8266.h>
70 #include <IRremote2ESP8266.h>
71 #include <IRremote2ESP8266.h>
72 #include <IRremote2ESP8266.h>
73 #include <IRremote2ESP8266.h>
74 #include <IRremote2ESP8266.h>
75 #include <IRremote2ESP8266.h>
76 #include <IRremote2ESP8266.h>
77 #include <IRremote2ESP8266.h>
78 #include <IRremote2ESP8266.h>
79 #include <IRremote2ESP8266.h>
80 #include <IRremote2ESP8266.h>
81 #include <IRremote2ESP8266.h>
82 #include <IRremote2ESP8266.h>
83 #include <IRremote2ESP8266.h>
84 #include <IRremote2ESP8266.h>
85 #include <IRremote2ESP8266.h>
86 #include <IRremote2ESP8266.h>
87 #include <IRremote2ESP8266.h>
88 #include <IRremote2ESP8266.h>
89 #include <IRremote2ESP8266.h>
90 #include <IRremote2ESP8266.h>
91 #include <IRremote2ESP8266.h>
92 #include <IRremote2ESP8266.h>
93 #include <IRremote2ESP8266.h>
94 #include <IRremote2ESP8266.h>
95 #include <IRremote2ESP8266.h>
96 #include <IRremote2ESP8266.h>
97 #include <IRremote2ESP8266.h>
98 #include <IRremote2ESP8266.h>
99 #include <IRremote2ESP8266.h>
100 #include <IRremote2ESP8266.h>

```

**Merancang Perangkat Keras (*Hardware*)** Tahapan perancangan perangkat keras bertujuan untuk menyusun alur kerja *prototype* secara sistematis, mempermudah proses perancangan.

## RENCANA PENGUJIAN

Rencana pengujian menguraikan alur kerja pengujian hasil dari rangkaian *prototype* yang telah dibuat. Pengujian dilakukan terhadap *prototype* yang mengendalikan kecepatan putaran mesin dengan menggunakan sensor IR *Speed*. Ketika terjadi *overspeed*, *buzzer* dan LED akan menyala untuk memberikan sinyal.

Pengambilan data penelitian akan dilakukan sebanyak 11 kali, di mana setiap pengujian akan mengevaluasi akurasi dan kinerja *prototype* sensor IR *Speed* pada RPM *main engine*. Data RPM yang ditampilkan di LCD akan menjadi referensi dalam pengujian ini.

## HASIL DAN PEMBAHASAN

Penelitian ini menyajikan hasil pembuatan sebuah *prototype* untuk mengontrol RPM pada mesin utama menggunakan sensor kecepatan IR berbasis Arduino Uno, dengan tujuan mencegah kecepatan yang terlalu tinggi (*overspeed*). Bab ini mencakup proses perakitan dan pengujian *prototype* tersebut, hingga diperoleh sebuah *prototype* yang dapat berfungsi dengan baik. Tujuan utama dari *prototype* ini adalah untuk menghasilkan respons yang diinginkan, di mana ketika kecepatan melebihi 4.000 rpm, *Buzzer* dan LED akan menyala, dan setelah 15 detik, mesin akan dimatikan untuk mengatasi *overspeed*.

### Pengujian Rangkaian

Pengujian rangkaian *prototype* controlling kecepatan RPM melibatkan komponen-komponen seperti sensor IR speed, microcontroller Arduino Uno, buzzer, relay, breadboard, dinamo, potensiometer, tachometer, LED, dan LCD. Rangkaian tersebut dirangkai untuk



**RANCANG BANGUN SISTEM CONTROLLING RPM PADA MAIN ENGINE  
BERBASIS ARDUINO UNO GUNA MENCEGAH TERJADINYA OVERSPEED**

memastikan bahwa prototype dapat berfungsi dengan baik. Setelah pengujian, hasilnya menunjukkan bahwa rangkaian tersebut dapat beroperasi dengan baik sesuai yang diharapkan.

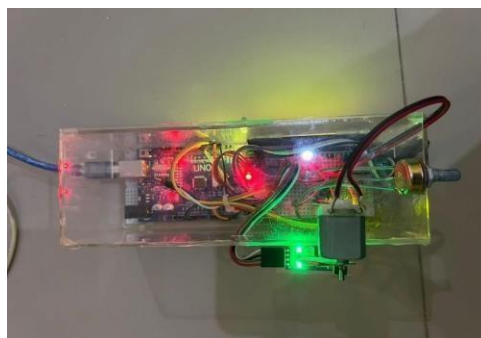
### **Pengujian Prototype**

Pada tahap pengujian *prototype*, dilakukan pengujian perangkat keras dan lunak. Pengujian perangkat keras melibatkan sensor IR *speed*, *relay*, *breadboard*, dinamo, potensiometer, tachometer, LED, dan LCD. Selanjutnya, pengujian perangkat lunak melibatkan Arduino Uno dan rangkaian *elektronika* secara keseluruhan. Pengujian sensor IR *speed* dilakukan dengan menggunakan dinamo sebagai pengganti mesin, dan hasilnya menunjukkan bahwa sensor dapat bekerja dengan baik dalam mendeteksi kecepatan RPM.

#### **1) Pengujian Sensor IR Speed**

Pada pengujian sensor IR *Speed*, sensor digunakan untuk mendeteksi kecepatan RPM pada mesin. Data yang dideteksi disimpan pada microcontroller untuk mengontrol *buzzer* dan LED sesuai pengaturan pada Arduino Uno. Pengujian dilakukan dengan menggunakan dinamo sebagai

pengganti mesin. Hasilnya menunjukkan bahwa sensor dapat bekerja dengan baik dalam mendeteksi kecepatan RPM pada dinamo.



#### **2) Pengujian LCD**

Setelah pengujian sensor berhasil, dilakukan pengujian LCD untuk memastikan bahwa LCD dapat menampilkan data yang dideteksi oleh sensor, seperti kecepatan RPM pada dinamo. Pengujian dilakukan beberapa kali untuk memastikan bahwa LCD berfungsi dengan baik. Hasilnya menunjukkan bahwa LCD berhasil menampilkan kecepatan RPM pada dinamo, dengan tampilan "0" saat dinamo dalam keadaan *off*.



### 3) Pengujian Perangkat *Input*

Pengujian perangkat input melibatkan pemasangan rangkaian perangkat keras ke Arduino Uno. Rangkaian tersebut mencakup sensor IR *speed* yang terhubung ke Arduino Uno untuk mentransfer data ke LCD 12C. Selain itu, rangkaian pada Arduino Uno juga meliputi *relay*, *potensiometer*, LED, *buzzer*, dan dinamo. Penjelasan tentang



koneksi pin-pin dapat dilihat di bawahini:

Tabel ARDUINO – LCD 12C

Arduino UNO	LCD 12C
5V	VCC
GND	GND
A4	SCL
A5	SDA

Tabel ARDUINO – RELAY

Arduino UNO	RELAY
5V	VCC
GND	GND
D3	INI

Tabel ARDUINO – SENSOR IR *SPEED*

Arduino UNO	SENSOR
5V	VCC
GND	GND
A3	AO
2	DO

Tabel ARDUINO – LED & BUZZER

Arduino UNO	Komponen
5V	VCC
GND	GND
D6	Buzzer
D5	Lampu Hijau
D4	Lampu Merah

Tabel POTENSIO METER – RELAY

Potensiometer	Relay
Analog	COM

Tabel RELAY – DINAMO

Dinamo	Relay
VCC	NO

Tabel DINAMO – DC JACK

Dinamo	DC Jack
GND	GND

**RANCANG BANGUN SISTEM CONTROLLING RPM PADA MAIN ENGINE  
BERBASIS ARDUINO UNO GUNA MENCEGAH TERJADINYA OVERSPEED**

**Penyajian Data**

Setelah menguji keseluruhan rangkaian, baik dari segi hardware maupun software, untuk memastikan bahwa setiap komponen bekerja dengan baik dan menghasilkan performa yang optimal, berikut adalah hasil dari pengujian ini:

No	Display		Indikator Lampu Led	Notifikasi
	LCD (Rpm)	Tachometer (Rpm)		Buzzer
<i>Off Speed</i>				
0	0	0	Led merah menyala	Buzzer nonaktif
<i>On Speed</i>				
1	2190 rpm	2217 rpm	Led hijau menyala	Buzzer nonaktif
2	1400 rpm	1406 rpm	Led hijau menyala	Buzzer nonaktif
3	2384 rpm	2328 rpm	Led hijau menyala	Buzzer nonaktif
4	3086 rpm	3095 rpm	Led hijau menyala	Buzzer nonaktif
5	3046 rpm	2965 rpm	Led hijau menyala	Buzzer nonaktif
6	3622 rpm	3640 rpm	Led hijau menyala	Buzzer nonaktif
7	2472 rpm	2505 rpm	Led hijau menyala	Buzzer nonaktif
8	1698 rpm	1684 rpm	Led hijau menyala	Buzzer nonaktif
9	2546 rpm	2544 rpm	Led hijau menyala	Buzzer nonaktif
10	2138 rpm	2133 rpm	Led hijau menyala	Buzzer nonaktif
<i>Over Speed</i>				
11	4202 rpm	4198 rpm	Led merah menyala	Buzzer aktif

Pengujian *prototype controlling RPM* menggunakan sensor IR *speed* dan dinamo sebagai pengganti *main engine*. Sistem ini mampu mengontrol kecepatan RPM pada dinamo, dimana sensor akan mengirim sinyal dan LED menyala ketika mendeteksi kecepatan. Pada tabel hasil pengujian, batas kecepatan RPM ditetapkan >4000 rpm. Ketika kecepatan berada di bawah 4000rpm, LED hijau menyala, tetapi jika melebihi 4000 rpm, terjadi *overspeed*, dengan LED merah dan *buzzer* menyala, dan mesin mati dalam 15 detik. Gambar menunjukkan pengambilan data pada *prototype*.



## Analisis Data

Setelah menyelesaikan uji coba rangkaian dan mencapai hasil optimal, analisis data diperlukan untuk memahami kemampuan prototype dalam mengontrol sistem kecepatan RPM pada mesin. Peneliti telah melalui berbagai tahap, mulai dari perancangan awal hingga pengujian prototype. Analisis data dari hasil pengujian digunakan untuk menilai tingkat keakuratan prototype. Untuk memverifikasi keakuratan, perbandingan dengan alat ukur lain dilakukan, dan tingkat akurasi dihitung dengan membandingkan data dari prototype dengan pengukuran manual. Persentase error dapat dihitung dengan membagi selisih pembacaan dengan nilai dari alat ukur lain, kemudian dikalikan dengan 100%.

$$\text{Error} = \frac{\text{Selisih Nilai Pembacaan}}{\text{Nilai Alat Ukur Digital}} \times 100 \% \dots\dots\dots (2)$$

Perbandingan sensor yang dibuat alat ukur dapat dilihat pada tabel berikut.

**PERBANDINGAN SENSOR DAN ALAT UKUR**

No.	Data Sensor & Alat Ukur		
	Data Sensor	Data Alat Ukur	Error
1.	2190 rpm	2217 rpm	1,23 %
2.	1400 rpm	1406 rpm	0,43 %
3.	2384 rpm	2328 rpm	2,40 %
4.	3086 rpm	3095 rpm	0,29 %
5.	3046 rpm	2965 rpm	2,73 %
6.	3622 rpm	3640 rpm	0,49 %
7.	2472 rpm	2505 rpm	1,32 %
8.	1698 rpm	1684 rpm	0,83 %
9.	2546 rpm	2544 rpm	0,08%
10.	2138 rpm	2133 rpm	0,23 %
Rata-Rata rpm			1,003 %
Data terbesar			2,73%
Data terkecil			0,08%

Tabel diatas memuat perbandingan antara data sensor dan alat ukur tachometer. Perbandingan ini bertujuan untuk mengevaluasi keakuratan *prototype*. Dari data tersebut, rata-rata *error*nya adalah 1,003%, menunjukkan ketidaksesuaian antara keluaran sensor dan alat ukur. Meskipun demikian, sensor dianggap berfungsi dengan baik karena perbedaan tersebut masih dalam batas toleransi.

## PENUTUP

### Kesimpulan

**RANCANG BANGUN SISTEM CONTROLLING RPM PADA MAIN ENGINE  
BERBASIS ARDUINO UNO GUNA MENCEGAH TERJADINYA OVERSPEED**

- 1) *Prototype controlling* kecepatan RPM pada *main engine* menggunakan Arduino Uno dan sensor IR *speed* berhasil dirancang untuk mencegah *overspeed*. Sensor ini terhubung dengan Arduino Uno untuk mendeteksi kecepatan RPM, yang ditandai dengan LED yang menyala.
- 2) Sistem *controlling* RPM yang dirancang berhasil bekerja dengan baik. Hasil pengambilan data menunjukkan tingkat keakuratan dalam mendeteksi kecepatan mesin, dengan selisih antara 0,08% hingga 2,73%, dan rata-rata selisih sebesar 1,003%. *Prototype* dapat mendeteksi kecepatan mesin di bawah batas maksimal (<4000 rpm) dengan LED hijau menyala, sementara jika kecepatan melebihi batas maksimal (>4000 rpm), LED merah dan *buzzer* akan aktif, dan mesin akan mati secara otomatis setelah 15 detik, sesuai dengan tujuan pencegahan *overspeed*.

### **Saran**

Hasil pengujian menunjukkan bahwa untuk meningkatkan akurasi pengendalian RPM mesin, digunakanlah sensor yang lebih presisi. Penelitian berikutnya disarankan untuk mengembangkan *prototype controlling* dengan sistem yang lebih praktis dan *modern*, yang sesuai dengan kebutuhan kapal.

### **DAFTAR PUSTAKA**

- Agus, E. (2015). \*Pengendalian overspeed pada mesin diesel\*. Semarang: Polines.
- Borg, W. R., & Gall, M. D. (1983). \*Educational research: An introduction\* (4th ed.). New York: Longman.
- Daryanto. (2004). \*Motor diesel\*. Jakarta: PT. Yrama Widya.
- Handoko. (2017, November). Sistem kendali perangkat elektronika monolitik berbasis Arduino Uno R3. \*Proceedings\*, 1–2.
- Hidayat, R., & Sapudin, B. S. (2018). Perancangan sistem keamanan rumah berbasis IoT dengan NodeMcu Esp8266 menggunakan sensor PIR HC-SR501 dan sensor smoke detector. \*Journal of Innovation and Applied Technology\*, 7(2), 139–148.
- Iskandariyanto, A., Brian, R., & Amrinsyah, R. (2020). Rancang bangun sistem monitoring tegangan, arus, dan frekuensi keluaran. \*Jurnal AMORI\*, 1, 3–8.
- Marta, M., Miftah, & Mintarso, C. J. (2010). Sistem monitoring putaran generator pada pembangkit listrik tenaga arus laut. \*Wave: Jurnal Ilmiah Teknologi Maritim\*, 3(1), 19–23.
- Pangestu, A. D., et al. (2019). Sistem monitoring beban listrik berbasis Arduino Nodemcu

Esp8266. *\*Journal of Electrical and Electronics Engineering\**, 4(1), 187–197.

Pramana, T., & Nababan, R. (2019). Perancangan perangkat penghitung jumlah penumpang pada kapal komersial menggunakan mikrokontroler. *\*Jurnal Sustainable Research and Industrial Application\**, 8(1), 18–29.

Saputro, B. (2021). *\*Best practices penelitian pengembangan (research & development) bidang manajemen pendidikan IPA\**. Academia Publication.

Widodo, E. D., Wardihani, S., Pramono, S., & Yulianto, T. (2018, April). Rancang bangun lampu duduk menggunakan LED dengan tiga level pencahayaan untuk mendukung industri kreatif kewirausahaan. *\*Proceedings\**.