

Rancang Bangun Alat Jemur Kerupuk dengan Mikrokontroler Berbasis IoT

Budi Sunarto¹, Achmad Anwari², Hanisa Azizah Lutfiah³

^{1,2,3}Program Studi Teknik Elektro, Sekolah Tinggi Teknologi Texmaco, Indonesia
Email : bdsunarto84@gmail.com, hanisaazizahlutfiah63@gmail.com

Received 19 Februari 2025 | Revised 4 Maret 2025 | Accepted 18 Maret 2025

ABSTRAK

Sistem jemuran kerupuk otomatis berbasis *NodeMCU ESP8266* dengan menggunakan sensor *DHT 11* dan *LDR* dirancang untuk mengoptimalkan proses pengeringan dengan kontrol otomatis. Sistem ini juga dilengkapi dengan notifikasi berbasis telegram untuk memberikan informasi *real time* kepada pengguna dengan parameter waktu, suhu, dan intensitas cahaya. Hasil pengujian menunjukkan bahwa sistem mampu menggerakkan rak jemuran secara otomatis berdasarkan kondisi rak jemuran akan keluar otomatis pada saat suhu $>30^{\circ}\text{C}$ dan nilai *LDR* <100 (cahaya terang) pada pukul 08:00, dan rak penjemuran masuk saat suhu $<30^{\circ}\text{C}$ atau nilai *LDR* >100 (cahaya redup) pada pukul 16:00. Selain itu, sistem mampu mendeteksi kondisi hujan dengan parameter suhu $<30^{\circ}\text{C}$ dan nilai *LDR* <100 di antara pukul 08:00 hingga 16:00, meskipun kondisi tersebut tidak terjadi selama pengujian. *Limit switch* digunakan untuk memastikan pergerakan rak jemuran agar berhenti di posisi yang telah ditentukan, dan pengujian respons motor menunjukkan bahwa motor tidak langsung berhenti saat *limit switch* aktif karena adanya *delay* pengaruh kualitas jaringan, tetapi tetap sesuai dengan logika sistem.

Kata kunci: *NodeMCU ESP8266*, sensor *ldr*, sensor *dht 11*, *internet of things*, *Telegram bot*

ABSTRACT

The automatic cracker drying system based on NodeMCU ESP8266 using DHT 11 and LDR sensors is designed to optimize the drying process with automatic control. This system is also equipped with telegram-based notifications to provide real-time information to users with time, temperature, and light intensity parameters. The test results show that the system is able to move the drying rack automatically based on the condition of the drying rack will automatically come out when the temperature is $>30^{\circ}\text{C}$ and the LDR value is <100 (bright light) at 08:00, and the drying rack enters when the temperature is $<30^{\circ}\text{C}$ or the LDR value is >100 (dim light) at 16:00. In addition, the system is able to detect rain conditions with temperature parameters $<30^{\circ}\text{C}$ and LDR values <100 between 08:00 and 16:00, although these conditions did not occur during the test. Limit switches are used to ensure that the movement of the drying rack stops at a predetermined position, and motor response testing shows that the motor does not stop immediately when the limit switch is active due to a delay due to the influence of network quality, but remains in accordance with the system logic.

Keywords: *NodeMCU ESP8266*, *ldr sensor*, *dht 11 sensor*, *internet of things*, *Telegram bot*.

1. PENDAHULUAN

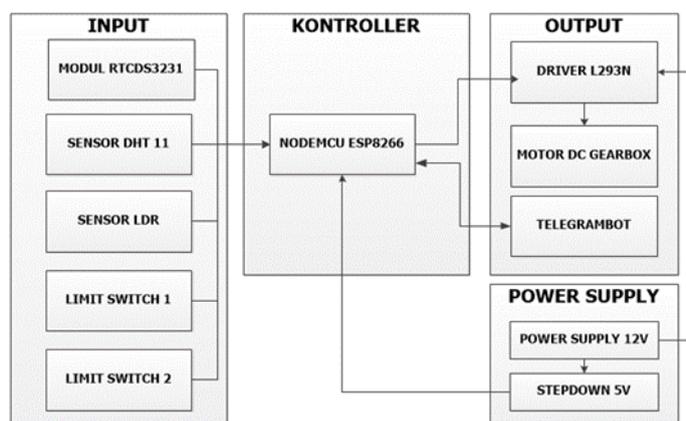
Otomatisasi telah menjadi bagian penting dalam kehidupan sehari-hari, termasuk dalam proses pengeringan kerupuk ro, makanan tradisional Indonesia. Sistem otomatis ini dirancang menggunakan sensor *LDR* (*Light Dependent Resistor*), sensor DHT 11, dan modul *Real Time Clock* (*RTC*) yang dikendalikan oleh NodeMCU 8266.

Sensor *LDR* mendeteksi intensitas cahaya, sensor DHT 11 mengukur suhu, dan modul RTC mengatur waktu. Sistem ini berfungsi untuk mengatur proses penjemuran kerupuk ro secara otomatis. Misalnya, jika waktu menunjukkan lebih dari pukul 08:00 dengan cahaya terang dan suhu di atas 30°C, kerupuk akan dijemur. Sebaliknya, jika waktu lebih dari pukul 16:00 dengan cahaya redup dan suhu di bawah 30°C, kerupuk akan otomatis dimasukkan ke dalam rak untuk melindunginya dari cuaca buruk.

Integrasi dengan aplikasi Telegram memungkinkan pengguna untuk memantau dan mengontrol proses pengeringan kerupuk ro dari jarak jauh, sehingga meningkatkan efisiensi dan melindungi kerupuk dari kerusakan akibat cuaca.

2. METODE

Perancangan Sistem alat ini meliputi melakukan studi pustaka untuk mengetahui spesifikasi dari alat yang akan digunakan, mendesain mekanik, elektrik, dan perangkat lunak. Lalu membuat flowchart dan blog diagram sistem dari alat yang akan dirancang, dan menguji alat secara keseluruhan.

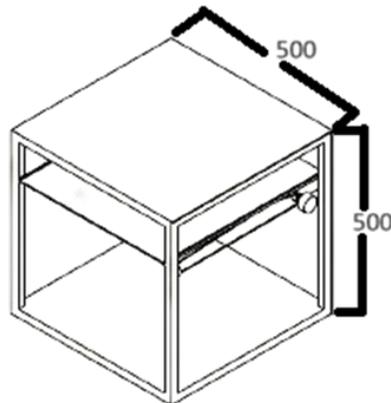


Gambar 2.1 Blog Diagram sistem alat

Gambar 2.1 menunjukkan blok diagram alat yang dirancang untuk penjemuran kerupuk. Pada bagian input, terdapat beberapa komponen: modul RTC DS3231 untuk menyediakan waktu akurat, sensor DHT 11 untuk mengukur suhu lingkungan, dan sensor *LDR* untuk mendeteksi intensitas cahaya. Limit switch digunakan untuk mendeteksi posisi jemuran, memastikan motor berhenti saat jemuran mencapai posisi yang diinginkan. NodeMCU ESP8266 berfungsi sebagai pusat kontrol sistem, berkomunikasi dengan bot Telegram melalui jaringan nirkabel, memungkinkan pengguna untuk memantau dan mengendalikan sistem dari jarak jauh. Pada bagian output, *driver* motor L298N digunakan untuk mengendalikan motor DC gearbox yang menggerakkan jemuran. Bot Telegram terintegrasi dengan mikrokontroler untuk memberikan notifikasi kepada pengguna dan menerima perintah, sehingga pengguna dapat mendapatkan informasi status sensor secara real-time dan mengirim instruksi manual. Sumber tegangan menggunakan *power supply* 12V/3A untuk motor DC, sementara komponen stepdown DC menurunkan tegangan dari 12V menjadi 5V untuk NodeMCU.

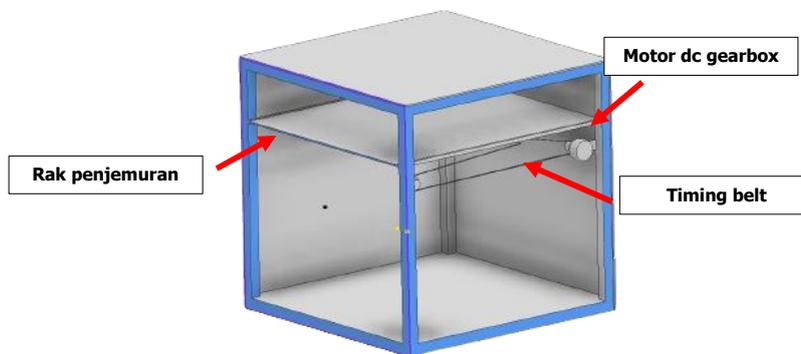
2.1 Desain Mekanik

Pada gambar 2.2 diperlihatkan desain mekanik rancangan Jemuran kerupuk ro otomatis menggunakan sensor *LDR* dan sensor suhu berbasis NodeMCU 8266 dengan monitoring telegram.

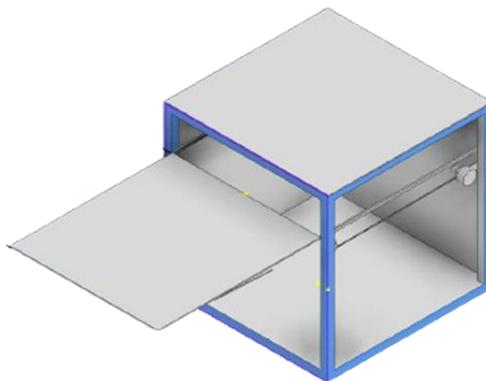


Gambar 2.2 Desain Mekanik

Alat yang akan dirancang memiliki bentuk kubus dengan dimensi, panjang 50 cm, tinggi 50 cm dan lebar 50 cm, bahan rangka menggunakan besi *hollow* ukuran 2 cm x 2 cm. Mekanisme dari alat yang akan dirancang merupakan sebuah sistem pengeringan otomatis dengan *drawer slide* yang digerakkan oleh *motor dc gearbox* untuk menggerakkan papan pengering seperti diperlihatkan pada gambar 2.3 dan 2.4.



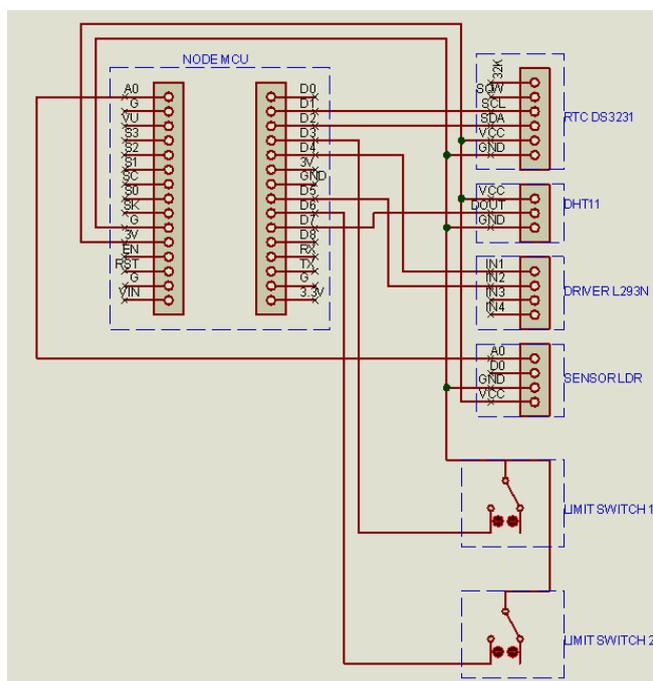
Gambar 2.3 Desain Mekanik tempat penyimpanan masuk



Gambar 2.4 Desain Mekanik tempat penyimpanan keluar

2.2 Desain Elektrik

Berikut rancangan skematik pengkabelan mikrokontroler Arduino dengan perangkat input dan output.



Gambar 2.5 Rancangan skematik pengkabelan mikrokontroler

2.3 Desain Perangkat Lunak

Flowchart perangkat lunak dari sistem jemuran kerupuk ro otomatis berbasis NodeMCU ESP8266 dimulai dengan proses inialisasi perangkat input dan output. NodeMCU mengonfigurasi perangkat input seperti rtc, sensor dht, sensor *LDR*, dan motor. Setelah itu, NodeMCU memastikan koneksi ke internet untuk fungsi optimal termasuk pengendalian jarak jauh dan pengiriman notifikasi. Pengguna dapat memilih mode operasi, manual atau otomatis, untuk mengendalikan motor pada rak jemuran sesuai instruksi.



Gambar 2.6 Flowchart program alat

3. HASIL DAN PEMBAHASAN

3.1 Hasil Perancangan Mekanik

Rangka alat jemuran kerupuk otomatis terbuat dari besi hollow 2 cm x 2 cm, berukuran 50 cm x 50 cm x 50 cm. Alat ini menggunakan motor DC gearbox 12v yang dikendalikan NodeMCU 8266 dan dilengkapi modul RTC ds3231, sensor *LDR*, dan sensor suhu dht 11. Ada 2 limit switch untuk mendeteksi akhir gerakan rak jemuran.



Gambar 3.1 Hasil perancangan mekanik rak jemuran bergerak masuk



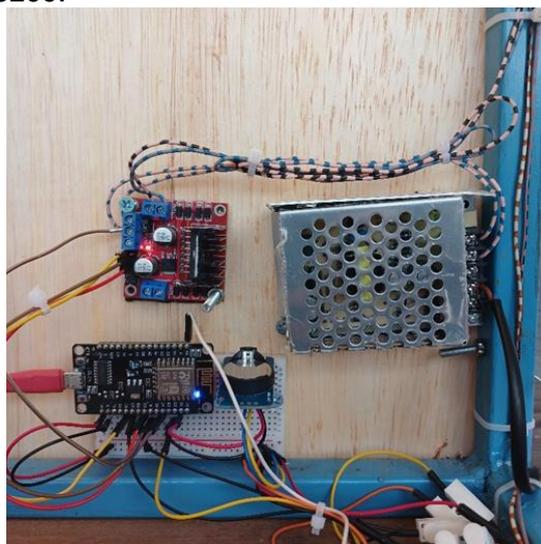
Gambar 3.2 Hasil perancangan mekanik rak jemuran keluar



Gambar 3.3 Mekanisme timing belt menggunakan motor dc

3.2 Hasil Perancangan Elektrik

Gambar 3.4 merupakan tata letak komponen-komponen elektrik yang terpasang pada papan, rangkaian ini berfungsi sebagai pusat kendali dari sistem jemuran kerupuk otomatis berbasis NodeMCU ESP8266.



Gambar 3.4 Hasil perancangan elektrik

Terdapat komponen yang terpasang pada breadboard seperti NodeMCU ESP8266, modul RTC ds3231, *driver* motor L298N, dan *power supply*. NodeMCU ESP8266 merupakan pusat kendali dari sistem keseluruhan, menerima input dari sensor dan mengontrol pergerakan output berupa motor serta mengirimkan notifikasi ke Telegram. Terdapat komponen rtc DS3231 menyediakan waktu yang akurat, memungkinkan sistem untuk menjalankan logika penjemuran berdasarkan waktu yang ditentukan. Selain itu juga terdapat *Driver* motor L298N, yang digunakan untuk mengontrol motor DC yang menggerakkan jemuran keluar dan masuk. *Driver* ini menerima sinyal dari NodeMCU untuk menentukan arah dan kecepatan motor. Motor tersebut memungkinkan jemuran bergerak otomatis berdasarkan perintah yang diberikan oleh mikrokontroler. Komponen lainnya adalah *power supply* yang berfungsi menyediakan tegangan yang dibutuhkan untuk mengoperasikan seluruh sistem, termasuk NodeMCU dan motor, untuk kabel dari masing-masing komponen terhubung disusun dengan rapi, untuk memudahkan proses perawatan.

3.3 Hasil Perancangan Software

1. Pemrograman Nodemcu esp8266

```
void handleManualControl(String message) {
  if (manualControl) {
    if (message == "/keluar") {
      if (!jemuranKeluar && digitalRead(LIMIT_SWITCH_OUT)) {
        moveMotor(true, 70);
        jemuranKeluar = true;
        bot.sendMessage(chatID, "Proses penjemuran manual dimulai"); }
    } else if (message == "/masuk") {
      if (jemuranKeluar && digitalRead(LIMIT_SWITCH_IN)) {
        moveMotor(false, 70);
        jemuranKeluar = false;
        bot.sendMessage(chatID, "Proses penjemuran manual selesai");
      }
    } else if (message == "/stop") {
      stopMotor();
      bot.sendMessage(chatID, "Motor dihentikan"); } } }
```

Fungsi *handleManualControl(String message)* mengontrol mode manual pada sistem jemuran otomatis melalui perintah *Telegram*. Fungsi ini menerima pesan dan melakukan tindakan sesuai isi pesan. Jika pengguna mengirim /keluar, sistem memeriksa dua kondisi: posisi jemuran harus keluar dan *limit switch* luar tidak aktif. Jika terpenuhi, motor bergerak keluar dengan fungsi *moveMotor(true, 70)* dan mengirim pesan ke pengguna. Jika pengguna mengirim /masuk, sistem memeriksa apakah jemuran keluar dan limit switch dalam tidak aktif. Jika ya, motor bergerak masuk dengan *moveMotor(false, 70)* dan mengirim pesan. Perintah /stop menghentikan motor dengan *stopMotor()* dan mengirim pesan. Jika pengguna mengirim /auto, sistem keluar dari mode manual dan mengatur variabel *manualControl* menjadi *false*, mengaktifkan mode otomatis berdasarkan kondisi yang telah diprogram.

```
void autoControlLogic() {
  float temperature = dht.readTemperature();
  int ldrValue = analogRead(A0);
```

```

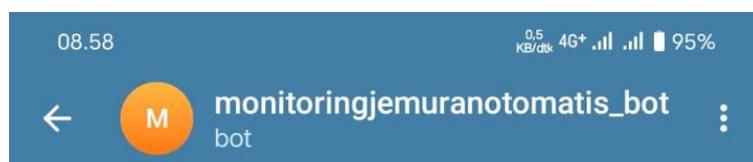
DateTime now = rtc.now();
if (!digitalRead(LIMIT_SWITCH_IN) || !digitalRead(LIMIT_SWITCH_OUT)) {
  stopMotor();
  return; // Hentikan logika otomatis jika limit switch aktif
}
if (now.hour() >= 8 && ldrValue < 100 && temperature > 30 && !jemuranKeluar) {
  if (digitalRead(LIMIT_SWITCH_OUT)) {
    moveMotor(true, 70);
    jemuranKeluar = true;
    bot.sendMessage(chatID, "Jemuran otomatis dikeluarkan karena cuaca cerah dan panas.");
  }
}
// Logika untuk jam 16 sore ke atas, kondisi sudah sore
else if (now.hour() >= 16 && ldrValue > 100 && temperature < 30 && jemuranKeluar) {
  if (digitalRead(LIMIT_SWITCH_IN)) { // Pastikan limit switch IN tidak aktif
    moveMotor(false, 70);
    jemuranKeluar = false;
    bot.sendMessage(chatID, "Jemuran otomatis dimasukkan karena kondisi sudah sore."); } }
else if (now.hour() >= 8 && now.hour() < 16 && ldrValue > 100 && temperature < 30 && jemuranKeluar) {
  if (digitalRead(LIMIT_SWITCH_IN)) { // Pastikan limit switch IN tidak aktif
    moveMotor(false, 70);
    jemuranKeluar = false;
    bot.sendMessage(chatID, "Jemuran otomatis dimasukkan karena kondisi mendadak gelap atau dingin (antisipasi hujan)."); } } }

```

Program ini mengontrol pergerakan jemuran otomatis berdasarkan waktu, suhu, dan intensitas cahaya. Jemuran keluar saat kondisi ideal, dan masuk jika kondisi berubah atau ada potensi hujan.

2. Telegram Bot

Software Telegram bot yang telah dibuat memungkinkan pengguna untuk mengontrol proses penjemuran secara manual melalui perintah yang dikirimkan melalui *telegram*. *Bot* ini memanfaatkan *NodeMCU ESP8266* untuk membaca data dari sensor suhu, kelembapan, dan intensitas cahaya, serta mengirimkan status dan notifikasi melalui *telegram*. Dengan integrasi ini, pengguna dapat dengan mudah memulai, menghentikan, atau mengatur kembali proses penjemuran kerupuk sesuai kebutuhan secara *real time*.



Gambar 3.5 Akun Telegram bot

3.4 Pengujian Penjemuran Berdasarkan Parameter Waktu, Suhu, dan Intensitas Cahaya

Pada tabel 3.1 diperlihatkan hasil pengujian suhu yang dilakukan pada tanggal 1 Januari 2025 mulai pukul 06.00 WIB sampai dengan 17.00 WIB berdasarkan data pengujian yang

ditampilkan dalam tabel tersebut, analisis dilakukan untuk menentukan waktu optimal proses penjemuran otomatis serta kondisi saat rak jemuran masuk atau keluar.

Tabel 3.1 Pengujian Kondisi Penjemuran

NO	WAKTU	SUHU <i>DHT11</i>	NILAI <i>LDR</i>
1	6:00	25	109
2	7:00	28.9	60
3	8:00	43.9	48
4	9:00	38	55
5	10:00	44.8	43
6	11:00	47.6	37
7	12:00	41.1	42
8	13:00	36.3	80
9	14:00	34.7	86
10	15:00	32.3	86
11	16:00	32.1	105
12	17:00	30	112

Gambar 3.6 menunjukkan hasil pengujian suhu dan intensitas cahaya (nilai *LDR*) yang diukur secara *real time* pada tanggal 1 Januari 2025. Data diambil setiap jam, mulai pukul 06:00 hingga pukul 17:00, berikut grafik dari hasil data tabel 3.1.



Gambar 3.1 Grafik hasil pengujian perubahan suhu dan nilai *LDR* berdasarkan waktu

Pada pukul 06:00, suhu berada di 25°C, dengan nilai *LDR* sebesar 109, yang menunjukkan intensitas cahaya masih rendah karena matahari baru terbit. Seiring waktu, suhu mulai meningkat, dan nilai *LDR* menurun karena intensitas cahaya bertambah. Pada pukul 07:00, suhu mencapai 28,9°C, sementara nilai *LDR* menurun ke 60, menunjukkan cahaya matahari yang mulai cukup terang. Memasuki pukul 08:00, suhu melonjak signifikan menjadi 43,9°C, dan nilai *LDR* menurun lebih jauh ke 48. Ini menunjukkan kondisi yang sangat ideal untuk memulai proses penjemuran, dengan suhu yang panas dan intensitas cahaya yang tinggi. Kondisi ideal ini berlanjut hingga pukul 12:00, dengan suhu berkisar antara 41,1°C hingga 47,6°C, dan nilai *LDR* tetap di bawah 50.

Pada pukul 13:00, suhu mulai menurun menjadi 36,3°C, dan nilai *LDR* meningkat ke 80, menandakan intensitas cahaya mulai berkurang. Penurunan suhu berlanjut hingga pukul 16:00, di mana suhu berada di 29,8°C, dan nilai *LDR* meningkat menjadi 105, menandakan

kondisi cahaya yang semakin redup. Pada pukul 17:00, suhu semakin menurun ke 29,5°C, dan nilai *LDR* mencapai 112, menunjukkan bahwa matahari hampir terbenam dan intensitas cahaya sangat rendah.

Dari data ini, dapat disimpulkan bahwa waktu ideal untuk memulai proses penjemuran adalah setelah pukul 08:00, ketika suhu lebih dari 30°C dan nilai *LDR* kurang dari 100. Proses penjemuran sebaiknya dihentikan setelah pukul 16:00, ketika suhu turun di bawah 30°C dan nilai *LDR* lebih dari 100. Untukantisipasi hujan, jika suhu turun di bawah 30°C dan nilai *LDR* juga kurang dari 100 antara pukul 08:00–16:00, rak jemuran harus segera dimasukkan.

3.5 Pengujian Respon Motor dan Limit Switch

Pada pengujian ini, dilakukan uji coba terhadap respon *limit switch* terhadap pergerakan motor dalam berbagai mode operasi, baik otomatis maupun manual. Tujuan dari pengujian ini adalah untuk memastikan bahwa *limit switch* memberikan respon yang sesuai untuk menghentikan pergerakan motor dc baik saat rak penyimpanan bergerak keluar maupun masuk. Pada sistem ini, terdapat dua *limit switch* yaitu *limit switch* 1 berfungsi untuk mendeteksi posisi rak saat berada di dalam dan *limit switch* 2 berfungsi untuk mendeteksi posisi rak saat berada di luar.

Tabel 3.1 Pengujian respon *limit switch*

No	Pengujian	<i>Limit switch</i> 1	<i>Limit switch</i> 2	Keterangan
1	Mode otomatis rak penyimpanan keluar	Tidak terkena	terkena	Berhenti dengan <i>delay</i>
2	Mode otomatis rak penyimpanan masuk	Terkena	Tidak terkena	Berhenti dengan <i>delay</i>
3	Mode manual rak penyimpanan keluar	Tidak terkena	terkena	Berhenti dengan <i>delay</i>
4	Mode manual rak penyimpanan masuk	Terkena	Tidak terkena	Berhenti dengan <i>delay</i>

Dari hasil pengujian tabel 3.3 yaitu pengujian respon *limit switch* terhadap kendali motor dc. Dalam setiap pengujian, motor tidak langsung berhenti begitu *limit switch* terkena, melainkan terdapat *delay* beberapa detik sebelum motor berhenti sepenuhnya. *Delay* ini disebabkan oleh jaringan yang memengaruhi pengolahan data pada sistem berbasis NodeMCU.

3.6 Pengujian Mode Otomatis

Pengujian mode otomatis dilakukan untuk memastikan sistem dapat bekerja secara mandiri sesuai dengan parameter yang telah ditentukan. Pada mode ini, sistem akan memantau data dari sensor suhu (*DHT11*) dan sensor cahaya (*LDR*) secara *real-time* untuk menentukan kapan rak penyimpanan harus keluar atau masuk. Pengujian ini juga mencakup respon sistem terhadap kondisi lingkungan, seperti nilai suhu, intensitas cahaya, dan waktu yang telah diatur. Pengujian dilakukan dalam rentang waktu tertentu untuk mendapatkan data parameter lingkungan serta memastikan sistem merespon sesuai logika yang telah diprogramkan.

Berdasarkan hasil tabel 3.3 pengujian mode otomatis, sistem menunjukkan respon yang sesuai dengan logika pemrograman yang telah ditentukan.

Tabel 3.2 Pengujian mode otomatis

No	Waktu	Suhu (°C)	Nilai LDR	Aksi Sistem	Status Motor	Keterangan
1	08:00	32.1	45	Rak keluar	ON	Sesuai parameter
2	12:00	40.5	35	Rak tetap keluar	OFF	Tidak ada perubahan
3	16:00	29.5	105	Rak masuk	ON	Sesuai parameter
4	13:00	36	85	Rak tetap keluar	OFF	Tidak ada perubahan
5	10:00	41.2	40	Rak tetap keluar	OFF	Tidak ada perubahan
6	9:00	35	60	Rak tetap keluar	OFF	Tidak ada perubahan
7	7:00	28	110	Rak tetap masuk	OFF	Suhu dan LDR tidak sesuai
8	17:00	27.5	115	Rak tetap masuk	OFF	Sesuai parameter

Berikut analisa dari hasil pengujian pada pukul 08:00, suhu mencapai 32.1°C dan nilai *LDR* berada di angka 45. Kondisi ini memenuhi parameter sistem sehingga motor dc aktif menggerakkan rak penjemuran keluar. Pada pukul 10:00, di mana suhu mencapai 41.2°C dan nilai *LDR* 40. Sistem mempertahankan posisi rak tetap keluar karena lingkungan sesuai dengan parameter. Pada pukul 16:00, suhu turun menjadi 29.5°C dan nilai *LDR* mencapai 105. Sistem mendeteksi perubahan parameter lingkungan mengaktifkan motor dc untuk menggerakkan rak penjemuran masuk. Pada pukul 17:00 nilai suhu 27.5°C) dan nilai *LDR* yang tinggi 115 rak penjemuran tetap berada di posisi masuk.

Sistem berhasil menentukan kondisi rak keluar dan masuk berdasarkan parameter suhu dan nilai *LDR*. Pada rentang waktu antara pukul 08:00 hingga 16:00, rak penyimpanan keluar jika suhu lebih dari 30°C dan nilai *LDR* kurang dari 100. Setelah pukul 16:00, rak masuk jika suhu di bawah 30°C atau nilai *LDR* lebih dari 100. Pengujian ini menunjukkan bahwa sistem mode otomatis telah bekerja sesuai dengan logika yang diharapkan.

3.7 Pengujian Kekuatan Jaringan Terhadap Waktu Pergerakan Mode Manual

Pengujian ini bertujuan untuk mengukur pengaruh kecepatan jaringan (Mbps) terhadap waktu *respons* sistem ketika mengirim instruksi manual melalui *Telegram* hingga motor bergerak sesuai instruksi. Pengujian dilakukan dengan variasi kecepatan jaringan internet.

Tabel 3.3 Pengujian Kekuatan Jaringan terhadap Waktu Pergerakan Mode Manual

No	Perintah	Pwm motor	Kecepatan Jaringan Unduh (Mbps)	Kecepatan Jaringan Unggah (Mbps)	Provider	Waktu Respons (detik)	Keteranga
1	/keluar	70	53.24	17.39	PT Telkom	17	Motor bergerak keluar
2	/masuk	70	53.24	17.39	PT Telkom	17	Motor bergerak masuk

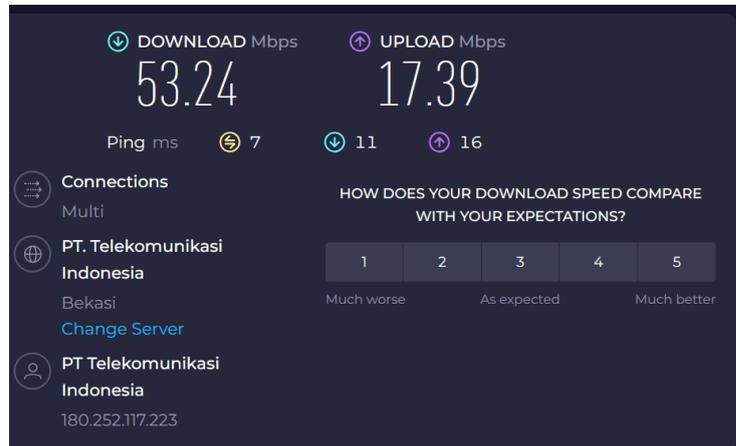
Rancang Bangun Alat Jemur Kerupuk Dengan Mikrokontroler Berbasis IOT

No	Perintah	Pwm motor	Kecepatan Jaringan Unduh (Mbps)	Kecepatan Jaringan Unggah (Mbps)	Provider	Waktu Respons (detik)	Keterangan
3	/keluar	70	53.24	17.39	PT Telkom	16	Motor bergerak keluar
4	/masuk	70	53.24	17.39	PT Telkom	17	Motor bergerak masuk
5	/keluar	70	53.24	17.39	PT Telkom	17	Motor bergerak keluar
6	/masuk	70	53.24	17.39	PT Telkom	17	Motor bergerak masuk
7	/keluar	70	23.08	25.30	XL	24	Motor bergerak keluar
8	/masuk	70	23.08	25.30	XL	21	Motor bergerak masuk
9	/keluar	70	23.08	25.30	XL	22	Motor bergerak keluar
10	/masuk	70	23.08	25.30	XL	21	Motor bergerak masuk
11	/keluar	70	23.08	25.30	XL	21	Motor bergerak keluar
12	/masuk	70	23.08	25.30	XL	23	Motor bergerak masuk

Berdasarkan hasil pengujian pada tabel 3.4 yaitu pengujian untuk mengukur waktu respons sistem dalam menjalankan perintah manual /keluar dan /masuk pergerakan motor menggunakan dua *provider* jaringan berbeda, yaitu PT Telkom dan XL Axiata. Hasil menunjukkan adanya perbedaan waktu respons yang dipengaruhi oleh kecepatan jaringan, khususnya kecepatan unduh. Pada pengujian dengan PT Telkom, kecepatan unduh rata-rata sebesar 53.24 Mbps, dan kecepatan unggah 17.39 Mbps. Dengan jaringan yang stabil, waktu *respons* sistem tercatat konsisten pada 16 hingga 17 detik untuk setiap perintah, baik untuk pergerakan motor keluar maupun masuk. Stabilitas ini mencerminkan bahwa kecepatan unduh yang tinggi memberikan kemampuan sistem untuk menerima perintah dengan cepat tanpa gangguan. Sedangkan pengujian dengan XL menunjukkan kecepatan unduh yang lebih rendah, yaitu 23.08 Mbps, meskipun kecepatan unggah lebih tinggi, yaitu 25.30 Mbps. Waktu respons pada jaringan ini berkisar antara 21 hingga 24 detik, yang lebih lambat dibandingkan PT Telkom. Hal ini menunjukkan bahwa kecepatan unduh yang lebih rendah dapat memengaruhi keterlambatan penerimaan perintah, meskipun kecepatan unggah cukup tinggi.

Dari hasil ini, dapat dianalisa bahwa kecepatan unduh memiliki pengaruh yang lebih signifikan terhadap waktu respon dibandingkan kecepatan unggah. Hal ini disebabkan oleh proses

penerimaan perintah dari pengguna yang lebih banyak bergantung pada kecepatan unduh. *Provider* dengan kecepatan unduh tinggi, seperti PT Telkom, memberikan hasil yang lebih baik dalam hal waktu respon.



Gambar 3.2 Speed test jaringan PT Telekomunikasi Indonesia



Gambar 3.3 Speed test jaringan XL

Gambar 3.7 dan gambar 3.8 merupakan hasil pengukuran kecepatan jaringan dilakukan menggunakan aplikasi *Speed test*. Pengukuran dilakukan untuk memastikan kecepatan unduh dan unggah dari jaringan yang digunakan selama pengujian. Penggunaan aplikasi *Speedtest* memastikan keakuratan data kecepatan jaringan sehingga dapat melihat hubungan antara kecepatan jaringan dan waktu respon motor terhadap perintah *telegram*.

3.8 Pengujian Kinerja Motor Berdasarkan Kondisi Beban

Hasil pengujian tabel 3.5 yaitu pengujian untuk mengukur waktu respon motor DC dalam kondisi beban yang bervariasi, mulai dari 0 gram hingga 1600 gram, menggunakan nilai pwm sebesar 70. Respon waktu dihitung sejak motor mulai bergerak, yaitu ketika limit switch pertama dilepas, hingga mencapai limit switch kedua. Proses ini dilakukan untuk dua arah pergerakan, yaitu keluar dan masuk menggunakan mode manual.

Waktu respons motor pada setiap kondisi beban tercatat konsisten, yaitu 13 detik untuk pergerakan keluar maupun masuk. Tidak adanya perubahan signifikan dalam waktu respon menunjukkan bahwa motor mampu menangani peningkatan beban hingga 1600 gram tanpa

mengalami penurunan performa. Secara keseluruhan, pengujian ini dapat menggerakkan beban pada rak jemuran dengan variasi beban hingga 1600 gram, tanpa memengaruhi waktu respon.

Table 3.5 Pengujian kinerja motor berdasarkan kondisi beban

No	Kondisi Beban	Pwm motor	Beban (gram)	Waktu Keluar (detik)	Waktu Masuk (detik)
1	Beban 1	70	0	13	13
2	Beban 2	70	200	13	13
3	Beban 3	70	400	13	13
4	Beban 4	70	600	13	13
5	Beban 5	70	800	13	13
6	Beban 6	70	1000	13	13
7	Beban 7	70	1200	13	13
8	Beban 8	70	1400	13	13
9	Beban 9	70	1600	13	13

4. KESIMPULAN

Sistem jemuran otomatis berbasis NodeMCU ESP8266 yang dirancang dalam penelitian ini berhasil mengoptimalkan proses pengeringan kerupuk dengan kontrol otomatis yang efisien. Dengan menggunakan sensor DHT 11 untuk mengukur suhu dan sensor *LDR* untuk mendeteksi intensitas cahaya, sistem dapat secara otomatis menggerakkan rak jemuran berdasarkan kondisi lingkungan. Hasil pengujian menunjukkan bahwa rak jemuran akan keluar secara otomatis pada suhu di atas 30°C dan nilai *LDR* di bawah 100 pada pukul 08:00, serta akan masuk pada suhu di bawah 30°C atau nilai *LDR* di atas 100 pada pukul 16:00.

Sistem ini juga dilengkapi dengan notifikasi berbasis *Telegram* yang memungkinkan pengguna untuk memantau dan mengontrol proses penjemuran dari jarak jauh, memberikan informasi *real-time* mengenai status suhu, waktu, dan intensitas cahaya. Pengujian terhadap *limit switch* menunjukkan bahwa sistem dapat menghentikan pergerakan rak jemuran dengan akurat, meskipun terdapat *delay* yang disebabkan oleh kualitas jaringan.

Secara keseluruhan, sistem ini menunjukkan kinerja yang baik dalam mengatur proses penjemuran kerupuk secara otomatis, serta mampu mendeteksi kondisi cuaca yang tidak menguntungkan, seperti potensi hujan. Dengan demikian, alat ini tidak hanya meningkatkan efisiensi dalam proses pengeringan, tetapi juga melindungi kerupuk dari kerusakan akibat cuaca buruk. Penelitian ini memberikan kontribusi signifikan dalam penerapan teknologi *Internet of Things (IoT)* dalam industri pengolahan makanan tradisional.

5. DAFTAR RUJUKAN

- [1] R. Suandi, "Penjemur Kerupuk Otomatis Menggunakan Sistem Kecerdasan Buatan Arduino Uno Dengan Menerapkan Metode Fuzzy Logic Controller (FLC)," *JURIKOM (Jurnal Riset Komputer)*, vol. 7, no. 2, 2020, doi: 10.30865/jurikom.v7i2.2099.
- [2] A. Dhanial, A. A. Nugroho, and M. Rifan, "Prototipe Sistem Kendali Otomatis Atap Jemuran Berbasis Internet of Things," *Autocracy: Jurnal Otomasi, Kendali, dan Aplikasi*

Industri, vol. 7, no. 1, 2021, doi: 10.21009/autocracy.071.1.

- [3] I. K. Wijayanti, Nurchim, and J. Maulindar, "PERANCANGAN SMART HOME JEMURAN OTOMATIS BERBASIS INTERNET OF THINGS," *INFOTECH journal*, vol. 9, no. 1, 2023, doi: 10.31949/infotech.v9i1.5344.
- [4] A. Satriadi, Wahyudi, and Y. Christiyono, "Perancangan Home Automation Berbasis *NodeMCU*," *Transient*, vol. 8, no. 1, 2019.
- [5] S. Hadi, R. P. M. D. Labib, and P. D. Widayaka, "Perbandingan Akurasi Pengukuran Sensor LM35 dan Sensor *DHT 11* untuk Monitoring Suhu Berbasis Internet of Things," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 6, no. 3, 2022, doi: 10.30998/string.v6i3.11534.
- [6] E. A. Prastyo, "Motor *Driver* Motor L298N," *edukasiaelektronika*.
- [7] M. Z. Sasongko and S. Sucipto, "Desain Prototype IoT Menggunakan Bot Telegram Berbasis Text Recognition," *RESEARCH: Journal of Computer, Information System & Technology Management*, vol. 4, no. 1, 2021, doi: 10.25273/research.v4i1.7420.
- [8] S. Pradana, "Rtc Ds3231," *2019*, vol. 12, no. 2, 2018