

Kentongan Otomatis Berbasis Waktu Menggunakan RTC dan Mikrokontroler Arduino

Muhammad Adzkia Adi Syahputra ^{1*}, Repal Reta Ardiyanto ², Satrio Bondan Prayogo ³, Bagus Verian Prasetyo ⁴, Rudi Susanto ⁵

¹Teknik Informatika/Fakultas Ilmu
Komputer

Universitas Duta Bangsa

^{1*}240103240@mhs.udb.ac.id

²Teknik Informatika/Fakultas Ilmu
Komputer

Universitas Duta Bangsa

²240103146@mhs.udb.ac.id

³Teknik Informatika/Fakultas Ilmu
Komputer

Universitas Duta Bangsa

³240103226@mhs.udb.ac.id

⁴Teknik Informatika/Fakultas Ilmu
Komputer

Universitas Duta Bangsa

⁴240103209@mhs.udb.ac.id

⁵Fakultas Ilmu Komputer

Universitas Duta Bangsa

⁵rudi_susanto@udb.ac.id

Abstrak— Perangkat tradisional seperti kentongan telah lama digunakan dalam masyarakat sebagai alat komunikasi jarak jauh, terutama untuk pemberitahuan darurat atau penanda waktu. Seiring dengan perkembangan teknologi, efisiensi dan otomatisasi menjadi kebutuhan utama. Penelitian ini bertujuan untuk mengembangkan sistem kentongan otomatis yang bekerja berdasarkan waktu menggunakan mikrokontroler Arduino dan modul Real Time Clock (RTC). Metode yang digunakan dalam penelitian ini meliputi perancangan perangkat keras (hardware) dan pemrograman perangkat lunak (software) untuk mengatur waktu dan kendali aktuator servo. Proses implementasi dilakukan pada miniatur pos kamling sebagai simulasi kondisi nyata, dengan pengujian yang difokuskan pada sinkronisasi waktu RTC, tampilan LCD, dan respon servo. Hasil pengujian menunjukkan bahwa sistem mampu bekerja secara akurat dan konsisten, menjadikannya solusi efektif untuk kebutuhan peringatan rutin tanpa ketergantungan pada operator manusia.

Kata kunci— Kentongan otomatis, Arduino, RTC, Mikrokontroler, Sistem peringatan.

Abstract— Traditional tools such as the kentongan (wooden slit drum) have long been used in communities as a means of long-distance communication, especially for emergency alerts or time indicators. With technological advancement, efficiency and automation have become essential needs. This study aims to develop an automatic kentongan system that operates based on time using an Arduino microcontroller and Real Time Clock (RTC) module. The research method involves the design of both hardware and software to control time-based activation of a servo motor. The system was implemented on a miniature guard post as a real-world simulation, and testing focused on RTC time synchronization, LCD display output, and servo motor responses. The test results show that the system functions accurately and consistently, making it an effective solution for routine alert needs without reliance on human operators.

Keywords— Automatic kentongan, Arduino, RTC, Microcontroller, Alert system.

I. PENDAHULUAN

Kentongan merupakan salah satu warisan budaya tradisional yang masih digunakan dalam kehidupan masyarakat, terutama di wilayah pedesaan. Alat ini berfungsi sebagai media komunikasi jarak jauh yang digunakan untuk memberikan isyarat, seperti peringatan keamanan, penanda waktu ibadah, atau pengingat kegiatan masyarakat[1]. Namun, sistem kentongan konvensional yang masih dioperasikan secara manual dinilai kurang praktis

di era modern yang menuntut efisiensi dan otomatisasi[2].

Seiring berkembangnya teknologi, penggunaan mikrokontroler seperti Arduino memberikan peluang untuk mengotomatisasi berbagai fungsi tradisional. Arduino merupakan platform open-source yang populer karena kemudahannya dalam pemrograman dan dukungannya terhadap berbagai modul tambahan[3]. Salah satu modul penting dalam sistem berbasis waktu adalah Real Time Clock (RTC), seperti DS3231, yang mampu

menyimpan dan mempertahankan data waktu secara presisi meskipun perangkat utama dalam kondisi mati[4].

Penelitian sebelumnya oleh Zulfikri dkk[5]. telah mengimplementasikan modul RTC pada sistem pengontrol penerangan rumah berbasis IoT. Sistem tersebut memungkinkan lampu rumah menyala dan mati secara otomatis sesuai waktu yang telah ditentukan, dengan tujuan efisiensi energi di lingkungan rumah tangga. Namun, sistem tersebut berfokus pada kebutuhan individual dan menggunakan perangkat output berupa lampu.

Berbeda dari penelitian tersebut, penelitian ini merancang sistem kentongan otomatis berbasis waktu dengan output berupa motor servo yang menggerakkan alat kentongan fisik. Sistem ini ditujukan untuk kepentingan bersama di lingkungan masyarakat, seperti pengingat waktu ibadah, peringatan keamanan, atau jadwal kegiatan umum. Oleh karena itu, implementasinya lebih menekankan pada aspek sosial dan kebermanfaatannya kolektif.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem kentongan otomatis berbasis Arduino dan RTC, serta menguji efektivitas dan keandalannya dalam skenario dunia nyata.

II. METODOLOGI PENELITIAN

Metodologi yang digunakan dalam penelitian ini adalah metode rekayasa atau engineering design, yang mencakup proses perancangan, implementasi, dan pengujian sistem kentongan otomatis berbasis waktu[6]. Adapun tahapan-tahapan dalam penelitian ini dijelaskan sebagai berikut:

2.1 Perancangan Sistem

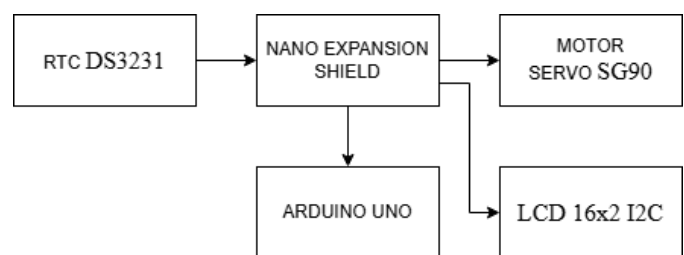
Perancangan sistem merupakan tahap awal dalam pengembangan alat kentongan otomatis berbasis waktu. Tujuannya adalah untuk menyusun arsitektur sistem secara menyeluruh, baik dari sisi fungsi logika maupun struktur fisik komponen. Sistem ini dirancang agar dapat mengaktifkan kentongan secara otomatis pada waktu-waktu tertentu yang telah diprogram, tanpa memerlukan intervensi manual.

Secara garis besar, sistem ini dibagi menjadi tiga bagian utama:

- 1) Unit kendali, yaitu mikrokontroler Arduino Uno yang mengatur semua proses kerja.
- 2) Unit waktu, yaitu modul RTC DS3231 yang memberikan informasi waktu aktual secara presisi.
- 3) Unit aktuator dan output, yaitu servo motor SG90 yang digunakan untuk memukul kentongan kayu dan LCD 16x2 I2C yang menampilkan waktu secara real-time.

2.1.1 Diagram Blok

Interaksi antar komponen digambarkan dalam diagram blok berikut.



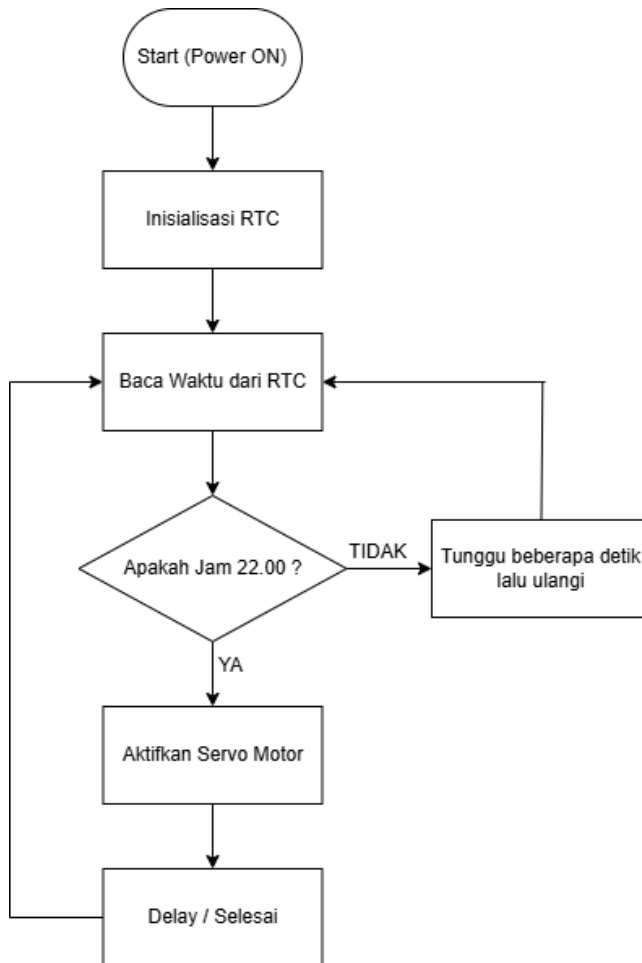
Gambar 1. Diagram blok sistem kentongan otomatis

Gambar 1 merupakan diagram blok sistem kentongan otomatis yang memperlihatkan bagaimana informasi waktu dari RTC dikirim ke Arduino melalui Nano Expansion Shield, kemudian diproses untuk mengaktifkan servo saat waktu yang diprogram tercapai. LCD digunakan untuk menampilkan informasi waktu kepada pengguna.

Arduino Uno berperan sebagai pusat kendali sistem, yang membaca data dari RTC dan menentukan apakah waktu yang sedang berjalan sesuai dengan waktu yang telah diprogram untuk membunyikan kentongan. Jika kondisi waktu terpenuhi, Arduino akan mengirim sinyal ke servo motor untuk melakukan gerakan pemukulan kentongan sesuai irama yang telah ditentukan. Pada saat yang sama, Arduino juga mengatur tampilan informasi yang muncul di LCD, seperti tanggal, waktu, suhu, dan status kentongan.

2.1.2 Flowchart

Untuk mempermudah pemahaman alur kerja sistem, digunakan diagram alir (flowchart) yang menggambarkan proses logika dari pengambilan waktu hingga eksekusi pemukulan kentongan.



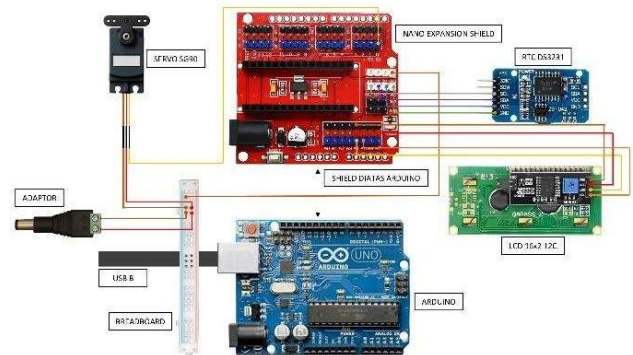
Gambar 2. Flowchart sistem kentongan otomatis

Gambar 2 merupakan flowchart sistem kentongan otomatis yang menjelaskan bahwa sistem membaca waktu dari RTC secara berkala, kemudian membandingkannya dengan waktu yang telah diprogram. Jika waktu cocok, maka servo akan diaktifkan untuk memukul kentongan, dan LCD akan menampilkan notifikasi. Setelah proses selesai, sistem kembali ke proses monitoring waktu.

2.1.3 Desain Pengkabelan Sistem

Setelah perancangan logika sistem dilakukan, langkah selanjutnya adalah menyusun desain pengkabelan untuk menghubungkan semua komponen dengan tepat dan efisien. Tujuan dari wiring diagram ini adalah memastikan semua koneksi antar komponen berjalan sesuai kebutuhan dan aman saat dioperasikan. Desain pengkabelan juga mempertimbangkan kestabilan suplai daya, keteraturan jalur kabel, serta kemudahan saat proses pemasangan dan

perawatan. Penempatan kabel yang rapi dan penggunaan konektor yang sesuai akan memudahkan proses pengujian serta mencegah terjadinya gangguan sambungan.



Gambar 3. Wiring diagram sistem kentongan otomatis

Gambar 3 merupakan wiring diagram sistem kentongan otomatis mendukung kestabilan sistem, mempermudah proses perakitan, serta memungkinkan pengujian dilakukan secara efisien sebelum sistem dipasang secara permanen.

III. HASIL DAN PEMBAHASAN

Hasil dari proses perancangan, perakitan, serta pengujian sistem kentongan otomatis berbasis waktu yang telah direalisasikan menggunakan mikrokontroler Arduino Uno. Perangkat ini memanfaatkan modul RTC DS3231 sebagai penentu waktu, motor servo SG90 sebagai aktuator pemukul kentongan, serta LCD 16x2 I2C sebagai penampil waktu secara real-time. Pembahasan dibagi menjadi beberapa subbagian, yaitu pembuatan hardware, pembuatan program, pengujian, dan evaluasi kinerja sistem.

3.1 Pembuatan Hardware

Tahap awal dalam pembangunan sistem kentongan otomatis adalah merancang dan merakit perangkat keras yang menjadi fondasi dari keseluruhan sistem. Arduino Uno digunakan sebagai pusat kendali utama yang mengatur semua proses kerja perangkat, dan dipasangkan dengan Nano Expansion Shield untuk mempermudah koneksi dan penataan jalur komunikasi antarkomponen.

Tahap merancang sistem kentongan otomatis dilakukan dengan menyusun konfigurasi

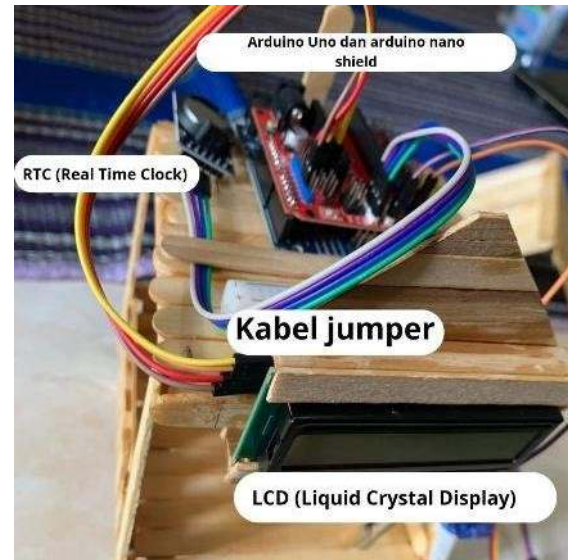
pengkabelan sebagai berikut. Modul RTC DS3231 dihubungkan ke pin SCL, SDA, VCC, dan GND pada Nano Expansion Shield, sedangkan LCD 16x2 I2C juga terhubung menggunakan jalur I2C yang sama ke shield tersebut. Rangkaian ini dapat dilihat pada Gambar 4, yang menunjukkan pemasangan Arduino Uno dengan Nano Shield, RTC, dan LCD 16x2 I2C.

Selanjutnya, servo motor SG90 dihubungkan ke pin D0 pada Nano Expansion Shield dan mendapatkan suplai daya dari breadboard. Breadboard tersebut menerima pasokan daya dari adaptor 5V 2A melalui DC barrel to terminal, yang berfungsi sebagai penghubung antara adaptor dengan jalur distribusi daya. Gambar 5 memperlihatkan konfigurasi perakitan breadboard, koneksi DC barrel, serta pemasangan servo ke shield secara keseluruhan.

Untuk mempermudah koneksi, Nano Expansion Shield digunakan sebagai media utama pemasangan kabel dari berbagai komponen dan ditempelkan di atas Arduino Uno menggunakan pin header, sehingga pengaturan pin menjadi lebih terstruktur. Breadboard berfungsi sebagai jalur distribusi tegangan dan koneksi yang fleksibel. Kabel jumper digunakan untuk menghubungkan antar pin dan komponen, memudahkan proses pengujian dan penyesuaian sistem. Sebagai bentuk implementasi fisik, sistem ini dipasang pada miniatur pos kamling, di mana kantong kayu berfungsi sebagai alat output utama yang dipukul oleh servo motor ketika waktu yang diprogram tercapai. Bentuk akhir dari perakitan sistem secara keseluruhan ditampilkan pada Gambar 6, yang menunjukkan implementasi pada miniatur pos ronda.

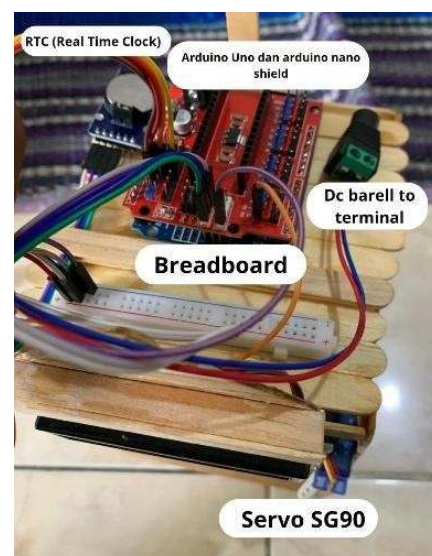
Tahap merakit dimulai dengan menyusun dan menghubungkan seluruh komponen utama. Arduino Uno, yang menjadi pusat kendali, dipasang Nano Expansion Shield untuk memudahkan koneksi ke komponen lainnya. Selanjutnya, modul RTC DS3231 dan LCD 16x2 I2C dipasang secara terpisah untuk memudahkan proses troubleshooting serta pengujian fungsional masing-masing modul. Gambar 4 menunjukkan proses pemasangan Arduino Uno

bersama Nano Shield yang terhubung dengan RTC dan LCD secara terpisah.



Gambar 4. Pemasangan Arduino Uno dengan Nano Shield, RTC DS3231, dan LCD 16x2 I2C

Setelah itu, servo motor SG90 dihubungkan ke pin D10 pada shield. Servo ini merupakan aktuator utama yang memukul kentongan. Untuk memastikan kestabilan catu daya, digunakan adaptor 5V 2A yang disambungkan melalui DC barrel to terminal, kemudian distribusi daya dialirkan melalui breadboard ke servo motor, sedangkan Arduino tetap mendapatkan pasokan daya dari port USB. Gambar 5, yang memperlihatkan hubungan antara breadboard, DC barrel, servo SG90, dan Nano Shield.



Gambar 5. Perakitan sistem breadboard, DC barrel, servo SG90, dan koneksi ke shield

Setelah semua komponen dirakit, sistem disusun ke dalam miniatur pos ronda yang dirancang secara khusus agar sesuai dengan skenario implementasi nyata. Pada tahap ini, seluruh koneksi dan posisi komponen diperiksa ulang agar terpasang secara aman, efisien, dan siap untuk dilakukan pengujian menyeluruh. Gambar 6 memperlihatkan hasil akhir perakitan seluruh sistem kentongan otomatis yang telah terpasang rapi di dalam miniatur pos ronda.



Gambar 6. Perakitan final sistem kentongan otomatis pada miniatur pos ronda

3.2 Pembuatan Program

Tahapan ini merupakan proses pengembangan perangkat lunak yang mengatur seluruh kerja sistem kentongan otomatis. Bahasa pemrograman yang digunakan adalah C/C++, dan seluruh proses pengkodean dilakukan pada Arduino IDE[7]. Program ini mengatur pembacaan waktu real-time dari modul RTC DS3231, menampilkannya ke layar LCD I2C 16x2, serta memicu motor servo SG90 untuk membunyikan kentongan otomatis pada waktu tertentu.

Struktur program dibagi menjadi tiga bagian utama: inisialisasi komponen di fungsi setup(), pembacaan waktu dan logika utama dalam fungsi loop(), serta eksekusi pemukulan kentongan pada fungsi kentongan(). Untuk menghindari pemacu berulang dalam menit yang sama, digunakan dua variabel logika yaitu sudahKentongan dan sedangKentongan. Selama proses pemukulan berlangsung, tampilan LCD akan menunjukkan pesan "Kentongan!" dan akan kembali

menampilkan waktu setelah semua irama selesai dimainkan. Setiap bagian program dirancang sesederhana mungkin agar mudah dipahami.

```

// RTC
#include "RTClib.h"
RTC_DS3231 rtc;
char daysOfTheWeek[7][12] = {"Ahad", "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu"};
int jam, menit, detik;
int tanggal, bulan, tahun;
String hari;
float suhu;

// LCD
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

// SERVO
#include <Servo.h>
Servo mekanik;

// FLAG
bool sudahKentongan = false;

void setup() {
  if (!rtc.begin()) {
    while (1) delay(10);
  }

  rtc.adjust(DateTime(2025, 6, 20, 21, 50, 50)); // SET MANUAL RTC

  lcd.begin();
  lcd.backlight();

  mekanik.attach(0); // pakai pin 0
  mekanik.write(0);
}

void loop() {
  // Ambil waktu dari RTC
  DateTime now = rtc.now();
  jam = now.hour();
  menit = now.minute();
  detik = now.second();
  tanggal = now.day();
  bulan = now.month();
  tahun = now.year();
  hari = daysOfTheWeek[now.dayOfTheWeek()];
  suhu = rtc.getTemperature();

  // Tampilkan waktu & suhu di LCD
  lcd.setCursor(0, 0);
  lcd.print(hari + ", " + String(tanggal) + "-" + String(bulan) + "-" + String(tahun) + " ");

  lcd.setCursor(0, 1);
  lcd.print(String(jam) + ":" + String(menit) + ":" + String(detik) + " ");
  lcd.print(suhu);
  lcd.print(" C ");

  // Cek waktu kentongan
  if ((jam == 22) && (menit == 0)) {
    if (!sudahKentongan) {
      kentongan(); // Jalankan kentongan
      sudahKentongan = true;
    }
  } else {
    sudahKentongan = false;
  }

  delay(500);
}

void kentongan() {
  // Tampilkan pesan di LCD
  lcd.setCursor(0, 1);
  lcd.print(" Kentongan! ");

  // Irama kentongan:
  // Tung
  mekanik.write(90); delay(100); mekanik.write(0); delay(500);
  // Tung
  mekanik.write(90); delay(100); mekanik.write(0); delay(500);
  // Tung panjang
  mekanik.write(90); delay(100); mekanik.write(0); delay(2000);
  // Tumpangtenggang
  for (int i = 0; i < 4; i++) {
    mekanik.write(90); delay(100); mekanik.write(0); delay(400);
  }
  delay(2000);
  // Tung
  mekanik.write(90); delay(100); mekanik.write(0); delay(500);
  // Tung
  mekanik.write(90); delay(100); mekanik.write(0); delay(500);

  // Ambil ulang waktu setelah kentongan selesai
  DateTime now = rtc.now();
  jam = now.hour();
  menit = now.minute();
  detik = now.second();
  tanggal = now.day();
  bulan = now.month();
  tahun = now.year();
  hari = daysOfTheWeek[now.dayOfTheWeek()];
  suhu = rtc.getTemperature();

  // Tampilkan kembali waktu & suhu
  lcd.setCursor(0, 0);
  lcd.print(hari + ", " + String(tanggal) + "-" + String(bulan) + "-" + String(tahun) + " ");

  lcd.setCursor(0, 1);
  lcd.print(String(jam) + ":" + String(menit) + ":" + String(detik) + " ");
  lcd.print(suhu);
  lcd.print(" C ");
}

```

Gambar 7. Potongan kode program sistem kentongan otomatis menggunakan Arduino IDE

Gambar 7 merupakan potongan kode program sistem kentongan otomatis menggunakan Arduino IDE yang dibuat modular dan efisien, sehingga mudah dikembangkan dan diuji. Dengan pendekatan ini, sistem tidak hanya mampu bekerja secara presisi sesuai waktu yang telah diprogram, tetapi juga memberikan umpan balik visual secara real-time kepada pengguna melalui tampilan LCD.

3.3 Pengujian dan Evaluasi

Pengujian kali ini mengambil jadwal ronda sebagai skenario nyata, sejalan dengan pemaparan Sanap (2025) bahwa RTC DS3231 cocok digunakan untuk aplikasi berbasis waktu seperti pengingat keamanan dan kegiatan rutin[8]. Hal ini merujuk pada tujuan awal sistem yang disebutkan dalam pendahuluan, yakni untuk pengingat waktu ibadah, peringatan keamanan, atau jadwal kegiatan umum. Jadwal ronda dipilih sebagai bentuk representatif kegiatan keamanan lingkungan yang membutuhkan penanda waktu otomatis.

Pengujian difokuskan pada tiga aspek penting, diantaranya adalah 1)Waktu pemicu yang telah diprogram dalam sistem (Arduino), 2)Waktu yang ditampilkan pada LCD berdasarkan data dari RTC DS3231, 3)Status aktifnya servo motor, yang menjadi indikator apakah kentongan berbunyi atau tidak.

Adapun arti dari Tabel 1 adalah sebagai berikut. Kolom Pengujian menunjukkan urutan seberapa pengujian dilakukan selama proses penelitian. Kolom Waktu Ditetapkan menampilkan waktu (dengan format jam:menit) yang telah ditentukan dalam kode program Arduino sebagai jadwal kentongan berbunyi. Kolom Waktu di LCD merupakan waktu yang tampil pada layar LCD berdasarkan pembacaan waktu secara real-time dari modul RTC, dengan format penulisan hari-bulan-tahun jam:menit:detik. Sementara itu, kolom Status Kentongan menunjukkan kondisi sistem, apakah kentongan dalam keadaan aktif (servo bergerak memukul) atau tidak aktif pada saat waktu tertentu berlangsung.

Tabel 1. Tabel hasil pengujian

Pengujian	Waktu Ditetapkan	Waktu di LCD	Status Kentongan
1	22:00	19-6-2025 21:1:52	Tidak berbunyi
2	22:00	19-6-2025 22:0:0	Berbunyi
3	22:00	19-6-2025 22:1:52	Tidak berbunyi
4	22:00	20-6-2025 1:0:28	Tidak berbunyi

Pada pengujian pertama, ketika waktu masih menunjukkan pukul 21:1:52, LCD menampilkan hari, tanggal, suhu, dan jam, namun servo tidak bergerak karena belum masuk waktu kentongan. Gambar 8 menunjukkan tampilan LCD pada kondisi ini, yaitu saat waktu masih lebih awal dari jadwal kentongan.



Gambar 8. Tampilan LCD pada pukul 21:00:10 sebelum waktu kentongan

Pada pengujian kedua, tepat saat waktu menunjukkan 22:00:00, LCD menampilkan teks “Kentongan!” dan servo bergerak memukul kentongan sesuai dengan irama yang telah diprogram. Tampilan LCD saat kentongan aktif dapat dilihat pada Gambar 9.



Gambar 9. Tampilan LCD menunjukkan teks "Kentongan!" dan servo aktif memukul

Pada pengujian ketiga, setelah kentongan selesai (pukul 22:1:52), LCD kembali menampilkan waktu normal, dan servo tidak melakukan gerakan lagi. Keadaan ini digambarkan pada Gambar 10.



Gambar 10. Tampilan LCD kembali normal setelah kentongan selesai

Pada pengujian keempat, ketika waktu menunjukkan 1:0:28, LCD tetap menampilkan waktu secara normal, dan servo tidak aktif, karena waktu tersebut bukan bagian dari jadwal kentongan. Kondisi sistem yang tetap dalam keadaan siaga tanpa aktivitas kentongan terlihat pada Gambar 11.



Gambar 11. Tampilan LCD pukul 1:0:28, sistem tetap normal tanpa aktivitas

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 1, sistem berhasil bekerja dengan baik dan sesuai skenario waktu yang telah ditentukan. Servo hanya bergerak sekali pada waktu yang diprogram, dan LCD memberikan tampilan informasi yang tepat pada setiap tahapnya. Kelebihan sistem ini meliputi, 1) Akurasi waktu yang tinggi berkat modul RTC DS3231, 2) Respons visual melalui LCD yang menampilkan status sistem, 3) Efisiensi daya dan kontrol gerakan servo yang tidak berulang.

Hasil dari implementasi sistem kentongan otomatis menunjukkan bahwa perangkat dapat bekerja sesuai dengan logika yang telah dirancang. Berdasarkan pengujian terhadap empat skenario waktu yang berbeda, sistem mampu mengeksekusi fungsi kentongan secara tepat waktu, ditandai dengan pergerakan servo dan tampilan informasi "Kentongan!" di layar LCD saat waktu kentongan tercapai. Hal ini menunjukkan bahwa integrasi antara modul RTC, Arduino, servo motor, dan LCD berjalan dengan sinkron dan stabil[9].

Pada pukul 22:0:0, sistem secara otomatis mengaktifkan servo untuk memukul kentongan dan menampilkan notifikasi di LCD, kemudian kembali ke tampilan waktu setelah irama selesai dimainkan. Respon waktu servo yang hanya aktif satu kali per siklus menit menunjukkan keberhasilan logika kontrol Boolean[10] yang

digunakan untuk menghindari eksekusi berulang dalam satu menit yang sama.

Sistem ini sangat memungkinkan untuk diterapkan dalam lingkungan masyarakat, seperti pos kamling atau mushola, sebagai pengingat waktu kegiatan rutin[11]. Keandalan sistem dalam bekerja secara otonom memperlihatkan keberhasilan rancangan baik dari sisi perangkat keras maupun perangkat lunaknya.

Untuk ke depannya, sistem dapat dikembangkan lebih lanjut dengan penambahan diantaranya, 1) Tombol manual atau remote untuk kentongan darurat, 2) Fitur pengaturan jadwal tanpa memodifikasi kode secara langsung, 3) Tampilan visual tambahan seperti LED atau buzzer sebagai notifikasi tambahan.

IV. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian, dapat disimpulkan bahwa sistem kentongan otomatis berbasis waktu menggunakan Arduino Uno, modul RTC DS3231, dan motor servo SG90 berhasil diimplementasikan dengan baik dan berfungsi sesuai yang diharapkan. Sistem mampu bekerja secara mandiri tanpa intervensi manusia dan menunjukkan kinerja yang stabil serta konsisten dalam mendeteksi waktu dan mengaktifkan kentongan pada jadwal yang telah ditentukan.

Pengujian menunjukkan bahwa sistem mampu mengenali waktu secara tepat, menampilkan informasi waktu dan suhu secara real-time melalui LCD, serta mengaktifkan servo dengan pola irama yang telah diprogram. Perangkat keras dan perangkat lunak bekerja secara terintegrasi dengan baik, menghasilkan sistem yang andal dan efisien.

Dengan demikian, sistem ini terbukti efektif sebagai solusi otomatisasi penanda waktu tradisional, dan layak untuk diterapkan di lingkungan masyarakat seperti pos ronda

atau mushola sebagai pengingat waktu kegiatan rutin.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan selama proses penelitian ini berlangsung. Ucapan terima kasih khusus disampaikan kepada dosen pembimbing yang telah memberikan arahan dan masukan yang berharga, serta kepada teman-teman yang selalu mendukung secara moral dan teknis.

REFERENSI

- [1] Fahri Zulfikar, "Sejarah Kentongan, Alat Komunikasi Tradisional Sejak Zaman Dahulu," *detikEdu*. Accessed: Jun. 27, 2025. [Online]. Available: <https://www.detik.com/edu/detikpedia/d-7077957/sejarah-kentongan-alat-komunikasi-tradisional-sejak-zaman-dahulu>
- [2] B. F. Effendi, D. Santoso, and F. D. Setiaji, "Perancangan Alat Kentongan Elektronik Menggunakan Mikrokontroler NodeMCU Berbasis Internet of Things (IoT)," *Univ. Kristen Satya Wacana, Salatiga*, pp. 1–11, 2017.
- [3] CuratePartners, "Exploring Arduino: The Open-Source Platform Empowering Innovation and Finding Specialized Talent for Your Projects," *ExploreArduino Blog*. Accessed: Jun. 27, 2025. [Online]. Available: <https://curatepartners.com/blogs/skills-tools-platforms/exploring-arduino-the-open-source-platform-empowering-innovation-and-finding-specialized-talent-for-your-projects>
- [4] I. Maxim Integrated Products, "DS3231 – Extremely Accurate I2C-Integrated RTC/TCXO/Crystal," 2015. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/ds3231.pdf>
- [5] Zulfikri, M. I. Sari, and F. Susanti, "Implementation of Current Sensor and RTC (Real Time Clock) at Home Lighting Control System Using Iot (Internet of Things).," *Appl. Sci.*, vol. 3, no. 3, p. 1762, 2017.
- [6] C. Dym, A. Agogino, O. Eris, D. Frey, and L. Leifer, "Engineering Design Thinking, Teaching, and Learning," *J. Eng. Educ. -Washington-*, vol. 94, pp. 103–120, Jan. 2005, doi: 10.1002/j.2168-9830.2005.tb00832.x.
- [7] Arduino, "What is Arduino?," *Arduino Documentation*. Accessed: Jun. 27, 2025. [Online]. Available: <https://docs.arduino.cc/learn/starting-guide/whats-arduino>
- [8] P. Sanap, "Design and Implementation of Real Time Clock using RTC DS3231 and Arduino Uno," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 13, pp. 1545–1551, Feb. 2025, doi: 10.22214/ijraset.2025.67162.
- [9] M. A. Mazidi, S. Naimi, and S. Naimi, "AVR Microcontroller and Embedded Systems The," 2011. [Online]. Available: http://www.microdigitaled.com/AVR/Articles/AVR_v1_online.pdf
- [10] J. Blum, *Exploring Arduino®*. 2019. doi: 10.1002/9781119405320.
- [11] Gesang Raharjo, Mochammad Djaoho, and Aris Sunawar, "Rancang Bangun Prototipe Alarm Pengingat Waktu Kuliah Berbasis Mikrokontroler Arduino Mega Di Ruang Unit Kegiatan Mahasiswa," *J. Electr. Vocat. Educ. Technol.*, vol. 4, no. 2, pp. 12–19, 2020, doi: 10.21009/jevot.0042.03