

Rancang Bangun dan Pengembangan Sistem Penjadwalan Sholat Digital Berbasis Internet of Things

Emil Naf'an^{a,1*}, Nilda Tri Putri^{b,2}

^a Universitas Putra Indonesia YPTK Padang, Padang, Indonesia

^b Program Studi Program Profesi Insinyur Universitas Andalas, Padang, Indonesia

¹ emilnafan@upitypk.ac.id; ² nildatp@eng.unand.ac.id;

* Penulis Korespondensi

Abstract

Penelitian ini membahas rancang bangun dan pengembangan sistem penjadwalan sholat digital berbasis *Internet of Things* (IoT) untuk meningkatkan keandalan dan fleksibilitas sistem yang telah digunakan sebelumnya. Sistem lama yang dibangun menggunakan Arduino Mega2560, RTC DS1307 dan panel 7-segment dengan *driver* ULN2003 memiliki keterbatasan, yaitu tidak tersedianya sinkronisasi waktu otomatis serta pengaturan jeda iqomah yang masih dilakukan secara manual. Dalam penelitian ini, pengembangan dilakukan dengan menambahkan modul ESP32 yang terhubung ke jaringan Wi-Fi untuk melakukan sinkronisasi waktu secara otomatis melalui server *Network Time Protocol* (NTP). Modul ini juga menyediakan antarmuka berbasis web yang dapat diakses melalui *smartphone*, sehingga pengaturan jadwal dan jeda iqomah menjadi lebih fleksibel. Komunikasi data antara ESP32 dan Arduino Mega2560 dilakukan melalui protokol *Universal Asynchronous Receiver Transmitter* (UART) yang terbukti stabil selama pengujian. Hasil pengujian menunjukkan bahwa sistem mampu menjaga deviasi waktu di bawah satu detik dan pengaturan jadwal melalui antarmuka web dapat dilakukan dalam waktu kurang dari dua detik. Sistem ini dinilai lebih andal, efisien, dan sesuai dengan kebutuhan operasional masjid modern, serta mampu menggantikan sistem lama yang kurang fleksibel dalam pengelolaan waktu ibadah.

Article History

Diterima 10 Juni 2025

Diperbaiki 12 Juli 2025

Diterbitkan 25 Oktober 2025

Keywords

Jadwal Sholat Digital

Internet of Things

ESP-32

Arduino

Sinkronisasi Waktu

Jeda Iqomah



This is an open-access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license

1. Pendahuluan

Sistem penjadwalan sholat digital merupakan perangkat penting dalam mendukung pelaksanaan ibadah di masjid, khususnya dalam memastikan ketepatan waktu sholat secara konsisten sesuai lokasi geografis [1] [2] [3] [4] [5] [6] [7] [8]. Di Masjid Rahmatan Lil 'Alamin, Universitas Putra Indonesia YPTK Padang, telah terpasang dua unit sistem jadwal sholat digital yang dikembangkan secara bertahap pada tahun 2016 [9] dan 2019 [10]. Kedua sistem tersebut memberikan manfaat signifikan bagi jamaah, terutama dalam menampilkan waktu ibadah secara visual dan *real time*.

Seiring dengan perkembangan infrastruktur kampus dan kebutuhan ruang ibadah yang lebih luas, muncul kebutuhan akan sistem yang lebih modern dan fleksibel. Pada tahun 2019, dilakukan pembangunan masjid baru dengan lokasi yang lebih strategis, sehingga pengurus yayasan meminta pengembangan sistem jadwal sholat digital baru yang dilengkapi fitur tambahan seperti waktu syuruq dan tampilan *running text* berukuran besar untuk mendukung ruang ibadah yang lebih besar. Meskipun secara fungsional sistem lama berjalan baik, terdapat beberapa kendala teknis dalam pengelolaan waktu.

Permasalahan utama yang ditemukan adalah proses sinkronisasi waktu global yang masih dilakukan secara manual dengan menekan tombol fisik pada perangkat secara berkala, sehingga rentan terhadap deviasi waktu yang dapat menimbulkan ketidaksesuaian jadwal. Selain itu, pengujian

menunjukkan adanya perbedaan akurasi antara dua unit alat yang terpasang di ruangan yang sama, yang dapat menyebabkan kebingungan di kalangan jamaah saat perbedaan waktu adzan terjadi.

Untuk mengatasi permasalahan tersebut, dalam penelitian ini dilakukan rancang bangun dan pengembangan sistem penjadwalan sholat digital berbasis *Internet of Things* (IoT) [11][12]. Sistem yang dirancang ini mengintegrasikan modul ESP32 yang terhubung ke jaringan Wi-Fi dan dapat memperoleh waktu global secara otomatis melalui server *Network Time Protocol* (NTP) [13]. Selain itu, sistem ini juga menyediakan antarmuka web yang dapat diakses melalui *smartphone* untuk kemudahan pengaturan jadwal sholat dan jeda iqomah.

Dalam sistem yang dikembangkan, komunikasi antara ESP32 dan Arduino Mega2560 [14] dilakukan melalui protokol *Universal Asynchronous Receiver Transmitter* UART untuk memastikan sinkronisasi data secara *real time* dan stabil. Pendekatan ini tidak hanya meningkatkan akurasi waktu, tetapi juga efisiensi pengelolaan sistem dan mengurangi ketergantungan pada intervensi manual. Dengan demikian, inovasi ini diharapkan mampu memenuhi kebutuhan operasional masjid modern yang mengutamakan presisi, kemudahan, dan keandalan sistem penjadwalan sholat digital.

2. Metode

Penelitian ini bertujuan untuk merancang dan mengembangkan suatu sistem berupa alat Jadwal Sholat Digital berbasis *Internet of Things* (IoT). Metode yang dilakukan dalam penelitian ini yaitu : Studi Sistem Lama, Perancangan Sistem, Implementasi Sistem dan Pengujian Sistem.

2.1. Studi Sistem Lama

Sistem jadwal sholat digital yang dikembangkan pada tahun 2016 menggunakan Arduino Mega2560 sebagai kontroler utama yang bertanggung jawab mengontrol tampilan waktu. Untuk mengendalikan modul 7-segment *common anoda*, sistem ini memanfaatkan kombinasi IC 74HC595 sebagai *shift register* dan ULN2003 sebagai *driver*, yang berfungsi untuk memperkuat arus ke masing-masing segmen LED. Sistem ini hanya mampu menampilkan waktu sholat harian secara statis berdasarkan data yang dimasukkan secara manual tanpa adanya pembaruan otomatis dari jaringan internet. Selain itu, pengaturan jeda iqomah dilakukan secara manual melalui tombol lokal yang terpasang di perangkat [15], sehingga memerlukan keterlibatan fisik dari pengurus masjid setiap kali ingin melakukan perubahan. Pada sistem ini, waktu syuruq belum ditampilkan, sehingga informasi tersebut tidak tersedia bagi jamaah yang ingin menunda sholat subuh hingga waktu syuruq berlalu. Sistem ini secara keseluruhan belum mendukung kendali jarak jauh dan belum terintegrasi dengan waktu global, sehingga akurasi waktu tergantung pada penyetelan manual.

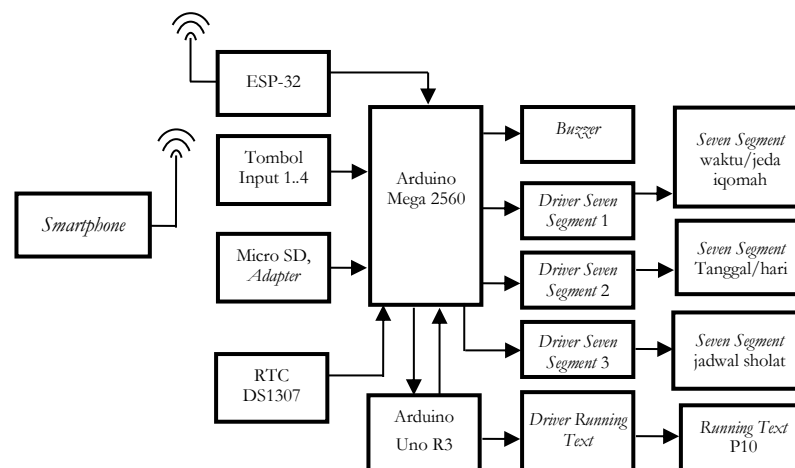
Pada pengembangan sistem tahun 2019, perangkat keras utama masih menggunakan Arduino Mega2560 sebagai pemegang kendali dari keseluruhan rangkaian. Pembaruan penting dari versi sebelumnya adalah penambahan fitur waktu syuruq, sehingga sistem mampu memberikan informasi lebih lengkap terkait jadwal sholat. Meski demikian, sistem ini belum terhubung ke internet dan tidak mendukung sinkronisasi waktu global secara otomatis, sehingga jam sistem masih bergantung pada penyesuaian manual yang rentan terhadap kesalahan waktu. Namun, terdapat kemajuan dalam hal antarmuka pengguna, yaitu dengan adanya fitur pengaturan tampilan *running text* melalui *smartphone*, yang memungkinkan pengurus masjid menambahkan pengumuman atau informasi lain yang ditampilkan di layar. Kendati begitu, fitur pengaturan jeda iqomah dan sinkronisasi waktu global melalui *smartphone* belum tersedia, sehingga fleksibilitas sistem masih terbatas. Dengan demikian dapat dikatakan bahwa sistem ini belum sepenuhnya mengadopsi teknologi *Internet of Things* (IoT).

2.2. Perancangan Sistem

Dalam melakukan perancangan sistem terdapat dua hal yang perlu dilakukan yaitu perancangan baik dari segi perangkat keras (*hardware*) maupun perangkat lunak (*software*). Perancangan tersebut didasari atas ketentuan antara lain :

- Sistem dirancang agar dapat menampilkan informasi waktu secara lengkap, termasuk jam, menit, dan detik serta waktu tunda jeda iqomah pada *seven segment* waktu, berdasarkan data yang diperoleh dari modul *Real Time Clock* RTC DS1307 [16].
- Selain itu, sistem juga harus mampu secara bergantian menampilkan informasi tanggal, bulan, tahun, dan nama hari pada *seven segment* yang sama dengan tampilan jeda iqomah.
- Ketika waktu sholat tiba, sistem harus secara otomatis mengaktifkan *buzzer* sebagai penanda bagi jamaah bahwa waktu sholat telah masuk.
- Sistem juga harus dapat menampilkan hitungan mundur untuk waktu jeda iqomah, sehingga jamaah memiliki gambaran waktu yang tersedia untuk melaksanakan sholat sunat sebelum dimulainya sholat fardhu.
- Sistem harus mampu menampilkan pesan yang dikirim dari *smartphone* pada *running text*, sehingga memungkinkan penyampaian informasi tambahan secara fleksibel kepada jamaah.
- Sistem harus mampu melakukan sinkronisasi waktu secara otomatis dengan waktu global melalui koneksi internet, sehingga akurasi waktu tetap terjaga.
- Sistem harus mendukung pengaturan waktu jeda iqomah secara langsung melalui *smartphone*, untuk memudahkan penyesuaian tanpa harus mengakses perangkat keras secara langsung.

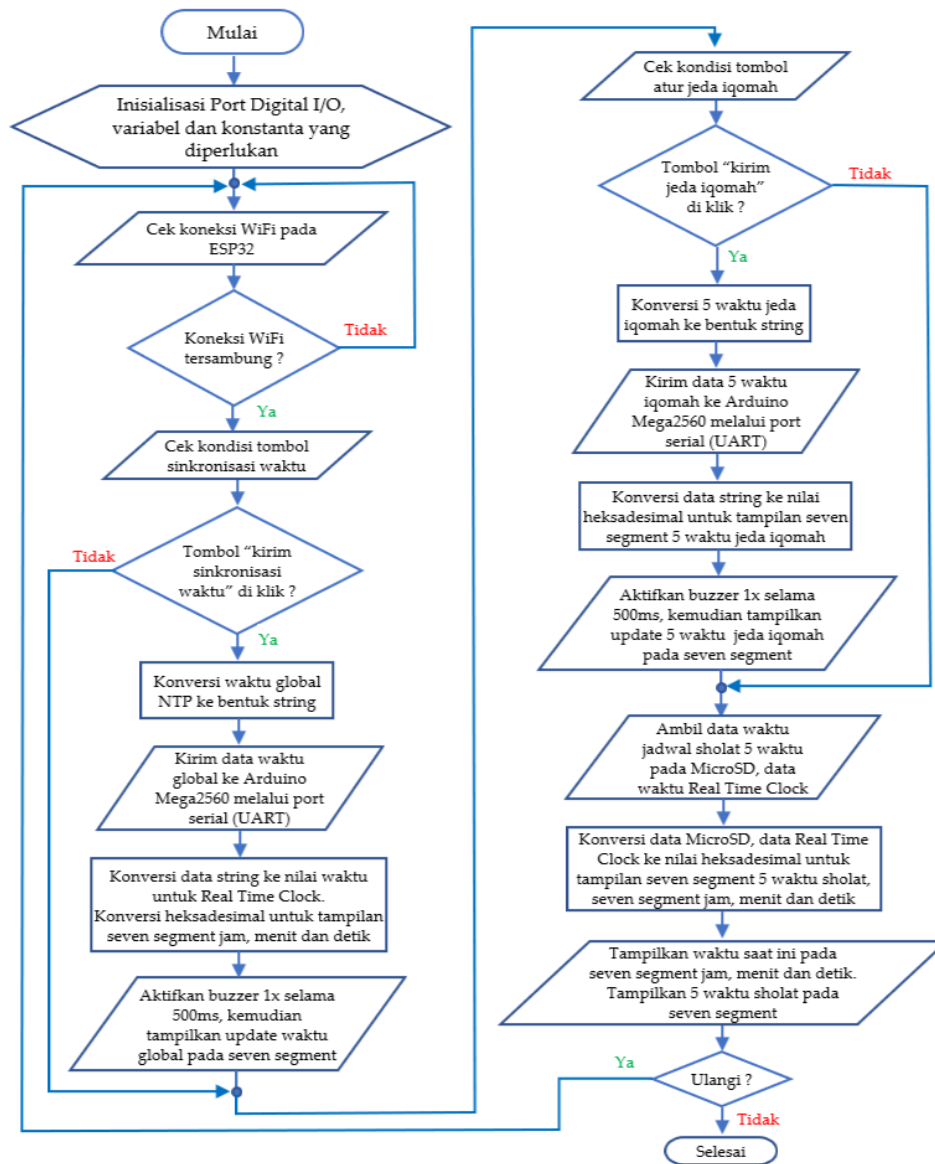
Adapun blok diagram dari sistem yang dirancang dapat dilihat pada gambar 1.



Gambar 1. Blok Diagram Jadwal Sholat Digital

Pada blok diagram gambar 1 tersebut terlihat bahwa sistem yang dirancang terdiri dari tiga modul kontroler, yaitu Arduino Mega2560, Arduino Uno R3, dan ESP32. Arduino Mega2560 berperan sebagai pengendali utama sistem [17] [18], sedangkan Arduino Uno R3 berfungsi untuk menerima data dari Mega2560 dalam bentuk pesan teks yang akan ditampilkan pada *running text* tipe P10. Arduino Uno R3 memproses data tersebut berdasarkan program yang tertanam di dalamnya, kemudian mengirimkannya ke *driver* P10 untuk ditampilkan pada panel *dot matrix* P10. Sistem dilengkapi dengan empat tombol input yang digunakan dalam proses pengaturan jadwal sholat, yaitu tombol *setting*, *pindah_posisi*, *tambah_nilai*, dan *simpan*. Data jadwal sholat disimpan dalam modul MicroSD. Pembaruan data waktu sholat dilakukan setiap pergantian hari pada pukul 00:00:01 dan ditampilkan pada *seven segment* khusus jadwal sholat. Untuk menjaga akurasi waktu meskipun catu daya dimatikan, digunakan modul *Real Time Clock* DS1307 yang dapat beroperasi hingga tahun 2099. Waktu ini kemudian diolah oleh Arduino Mega2560 dan ditampilkan pada *seven segment* dalam format jam, menit, dan detik. Sebagai indikator masuknya waktu sholat digunakan *buzzer*, yang berbunyi selama 500 milidetik sebanyak tiga kali berturut-turut. Sistem mendapatkan sumber daya dari jaringan listrik PLN 220VAC. Selain itu, sistem ini juga menggunakan modul ESP32 [19] yang berfungsi menerima instruksi dari *smartphone*, seperti sinkronisasi waktu global dan

pengaturan jeda waktu iqomah. Komunikasi antara ESP32 [20] dan Arduino Mega2560 dilakukan melalui *port serial* (UART) [21] [22]. Pemilihan ESP32 disebabkan beberapa kelebihanannya dibandingkan pendahulunya, ESP8266 [23], terutama dalam bidang *Internet of Things* (IoT) [24] [25]. Secara keseluruhan, sistem dikendalikan oleh Arduino Mega2560 dengan dukungan Arduino Uno R3 dan ESP32, berdasarkan data dan logika program yang tertanam pada masing-masing modul.



Gambar 2. Flowchart Sistem Jadwal Sholat Digital Berbasis *Internet of Things* (IoT)

Gambar 2 menjelaskan *flowchart* sistem dari proses sinkronisasi waktu, pengaturan jeda iqomah melalui *smartphone*, hingga pengendalian tampilan jam, menit dan detik serta jadwal sholat 5 waktu pada *seven segment*, pengaktifan *buzzer* oleh Arduino Mega2560.

2.3. Implementasi Sistem

Pada tahap ini dilakukan implementasi dari sistem yang telah dirancang. Implementasi sistem dilakukan dengan membangun perangkat (alat) yang telah dirancang sebelumnya baik dari segi perangkat keras (*hardware*) maupun perangkat lunak (*software*). Dari segi perangkat keras (*hardware*), perangkat (alat) dibangun mulai dari pembuatan rangkaian elektroniknya. Untuk bagian luar alat dibuat menggunakan panel dekoratif kayu yang dilapisi dengan resin. Panel kayu ini dipilih

agar tampilannya lebih bagus dan estetik. Gambar 3 menunjukkan proses pembuatan komponen elektronik. Gambar 4 menunjukkan proses pemasangan pada panel dekoratif kayu yang sudah dilapisi resin. Sedangkan pada gambar 5 menunjukkan proses pemasangan peralatan pada masjid Rahmatan Lil 'Alamin.



Gambar 3. Proses pembuatan komponen elektronik



Gambar 4. Proses pemasangan komponen pada panel dekoratif kayu



Gambar 5. Proses pemasangan peralatan di Masjid

2.4. Pengujian Sistem

Pada tahap ini dilakukan pengujian terhadap sistem yang telah dirancang. Mekanisme pengujian sistem sebagai berikut :

- a. *Unit Testing*
Pengujian masing-masing unit komponen program untuk meyakinkan bahwa sudah beroperasi secara benar.
- b. *Module Testing*
Pengujian terhadap koleksi unit-unit komponen yang saling berhubungan.
- c. *Subsystem Testing*
Pengujian terhadap koleksi modul-modul yang membentuk suatu *subsystem* (aplikasi).
- d. *System Testing*
Pengujian terhadap integrasi subsystem, yaitu keterhubungan antar *subsystem*.

3. Hasil dan Pembahasan

Setelah peralatan dibuat, selanjutnya peralatan diuji untuk mengetahui sejauh mana kinerja dari peralatan yang dirancang. Pengujian dilakukan pada beberapa rangkaian dari blok diagram antara lain ; pengujian rangkaian tombol, *seven segment*, *buzzer*. Disamping itu pengujian juga dilakukan pada sinkronisasi waktu global dan pengujian pengiriman data jeda iqomah. Adapun gambar keseluruhan Jadwal Sholat Digital tersebut dapat dilihat pada gambar 6 di berikut ini.



Gambar 6. Tampilan Fisik Jadwal Sholat Digital

3.1. Pengujian Rangkaian Tombol

Pengujian dilakukan dengan cara mengamati nilai tegangan pada saat tombol dalam kondisi ditekan maupun tidak ditekan. Hasil pengukuran ditampilkan pada Tabel 1.

Tabel 1. Pengujian Tombol Input 1.4

No.	Kondisi Tombol	Tegangan Output (VDC)
1	Ditekan	0.00 V
2	Tidak Ditekan	4.99 V

Berdasarkan data pengujian, diketahui bahwa ketika tombol ditekan, tegangan output sebesar 0,00 VDC, yang dalam sistem digital diartikan sebagai logika '0' (*low*). Hal ini terjadi karena jalur output tombol terhubung langsung ke ground. Sebaliknya, saat tombol tidak ditekan, tegangan output tercatat sebesar 4,99 VDC. Kondisi ini disebabkan oleh koneksi output ke sumber tegangan melalui resistor sebesar 4,7 K Ω , yang dalam sistem digital dikategorikan sebagai logika '1' (*high*). Dengan demikian, dapat disimpulkan bahwa rangkaian tombol input 1

hingga 4 berfungsi dengan baik dan sesuai untuk dibaca sebagai input digital oleh modul Arduino.

3.2. Pengujian *Seven Segment* Waktu (jam, menit, detik)

Pengujian dilakukan dengan mengamati secara langsung kinerja *seven segment* dalam menampilkan angka 0 hingga 9 pada format waktu (jam, menit, dan detik). Hasil pengujian menunjukkan bahwa *seven segment* mampu menampilkan seluruh angka dari 0 sampai 9 dengan baik.



Gambar 7. Pengujian Tampilan *Seven Segment* Waktu (jam, menit, detik)

Sebagian hasil dari proses pengujian ditampilkan pada Gambar 7. Berdasarkan hasil tersebut, dapat disimpulkan bahwa sistem *seven segment* dalam menampilkan waktu (jam, menit, dan detik) bekerja secara optimal.

3.3. Pengujian *Seven Segment* Tanggal / Hari

Pengujian dilakukan dengan mengamati secara langsung tampilan tanggal dan hari pada *seven segment*, lalu membandingkannya dengan data tanggal yang ditampilkan pada laptop (PC). Hasil dari pengujian ini ditampilkan pada Gambar 8.



Gambar 8. Pengujian Tampilan *Seven Segment* Tanggal / Hari

Dalam gambar 8 tersebut terlihat bahwa tampilan hari dan tanggal muncul secara bergantian dengan jeda waktu sekitar 3 detik. Interval waktu ini dipilih berdasarkan estimasi rata-rata waktu yang dibutuhkan seseorang untuk membaca informasi hari dan tanggal, yaitu sekitar ± 3 detik. Setelah dilakukan perbandingan antara tampilan pada *seven segment* dan pengaturan waktu pada *laptop*, hasil yang diperoleh sama, yaitu hari Kamis bertepatan dengan tanggal 05 Juni 2025. Berdasarkan hasil tersebut, dapat disimpulkan bahwa rangkaian *seven segment* untuk penunjuk waktu (jam, menit, dan detik) berfungsi dengan baik.

3.4. Pengujian *Seven Segment* Jadwal Sholat

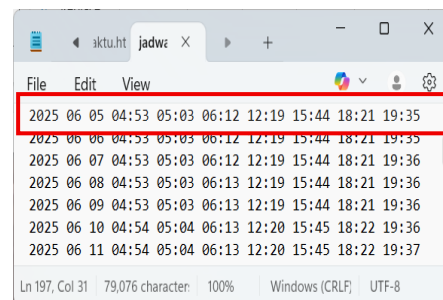
Pengujian dilakukan dengan cara mengamati secara langsung tampilan waktu sholat pada *seven segment*, lalu membandingkannya dengan data jadwal sholat yang tersimpan di dalam MicroSD. Hasil dari pengujian ini ditunjukkan pada Gambar 9.



(a)



(b)



(c)

(a) Tampilan Jadwal Sholat Pada Alat

(b) Tampilan Jadwal Sholat Pada Kalender

(c) Tampilan Jadwal Sholat Pada *MicroSD*

Gambar 9. Pengujian Tampilan *Seven Segment* Jadwal Sholat

Dari gambar 9 tersebut, terlihat bahwa waktu sholat untuk tanggal 05 Juni 2025 yang ditampilkan oleh alat sesuai dengan jadwal pada kalender maupun data di *MicroSD*. Rincian nilai tersebut dapat dilihat pada Tabel 2.

Tabel 2. Pengujian Data Jadwal Sholat

No.	Nama	Alat	Kalender	MicroSD	Hasil
1	Imsak	04:53	04:53	04:53	Sama
2	Shubuh	05:03	05:03	05:03	Sama
3	Syuruq	06:12	06:12	06:12	Sama
4	Dzuhur	12:19	12:19	12:19	Sama
5	Ashar	15:44	15:44	15:44	Sama
6	Maghrib	19:21	19:21	19:21	Sama
7	Isya	19:35	19:35	19:35	Sama

Dengan demikian dari hasil yang diperoleh dapat dikatakan bahwa rangkaian *seven segment* jadwal sholat dapat bekerja dengan baik.

3.5. Pengujian Rangkaian *Buzzer*

Pengujian dilakukan dengan memberikan sinyal tegangan berupa logika '0' dan '1' pada input rangkaian *buzzer*, kemudian mengamati bunyi yang dihasilkan. Hasil dari pengujian ini disajikan dalam Tabel 3.

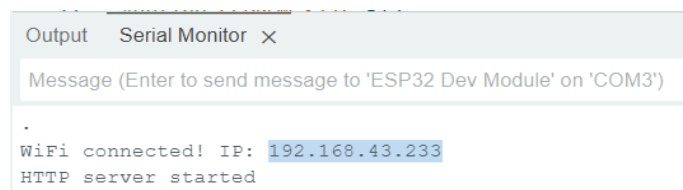
Tabel 3. Pengujian *Buzzer*

No.	Logika Input	Kondisi <i>Buzzer</i>
1	'0'	Tidak aktif
2	'1'	Aktif / Mengeluarkan Suara / bunyi yang bisa didengar dalam radius ± 30 meter.

Hasil pengujian menunjukkan bahwa saat diberikan logika '0', *buzzer* berada dalam kondisi nonaktif (tidak menghasilkan suara). Sebaliknya, ketika logika '1' diberikan, *buzzer* aktif dan menghasilkan bunyi yang dapat terdengar hingga jarak sekitar ± 30 meter. Berdasarkan pengujian ini, dapat disimpulkan bahwa rangkaian *buzzer* berfungsi dengan baik dan memenuhi kriteria untuk digunakan sebagai penanda waktu masuk sholat.

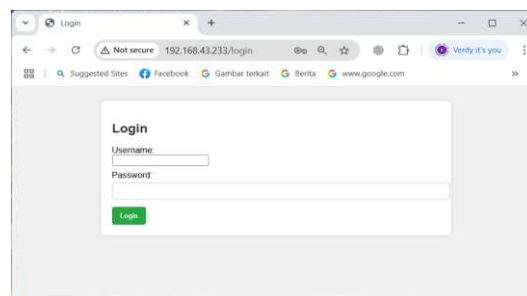
3.6. Pengujian Sinkronisasi Waktu Global

Pengujian ini bisa dilakukan menggunakan *browser* pada *smartphone* maupun *browser* pada *laptop*. Sistem akan membaca data waktu yang berasal dari server *Network Time Protocol* (NTP) yang terdapat pada *smartphone* maupun *laptop*. Sebelum pengiriman data waktu tersebut, pastikan modul ESP32 sudah terhubung ke jaringan Wi-Fi. Jika sudah terhubung, selanjutnya lakukan pengaturan sinkronisasi waktu, admin melakukan login pada *web browser* dengan memasukkan alamat IP : <http://192.168.43.233>. Alamat ini diperoleh dari tampilan pada Serial Monitor Arduino IDE, seperti ditunjukkan pada gambar 10 berikut ini.



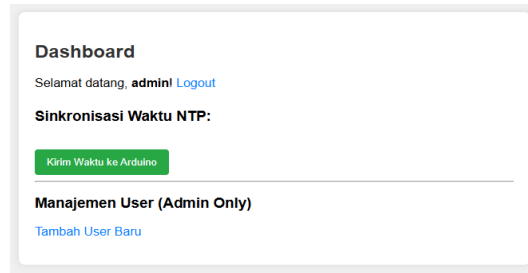
Gambar 10. Tampilan Alamat IP Akses ESP32

Selanjutnya lakukan login sebagai Admin dan masukkan *password*nya, seperti tampilan pada gambar 11.



Gambar 11. Tampilan Login ke Sistem

Setelah berhasil login, Selanjutnya akan tampil *dashboard* untuk pengaturan jeda iqomah dan sinkronisasi waktu. Untuk melakukan sinkronisasi waktu *Network Time Protocol* (NTP) klik tombol **Kirim Waktu ke Arduino**, seperti ditunjukkan pada gambar 12.



Gambar 12. Tampilan *Dashboard* Pengaturan Sinkronisasi Waktu NTP

Gambar 13 menunjukkan tampilan hasil pengaturan sinkronisasi waktu *Network Time Protocol* (NTP) pada alat. Tampilan pada *seven segment* waktu (jam, menit, dan detik) menunjukkan angka : 11:39:45. Masing-masing alat jadwal sholat menunjukkan nilai yang sama. Ini menunjukkan bahwa pengaturan sinkronisasi waktu NTP dapat bekerja dengan baik.

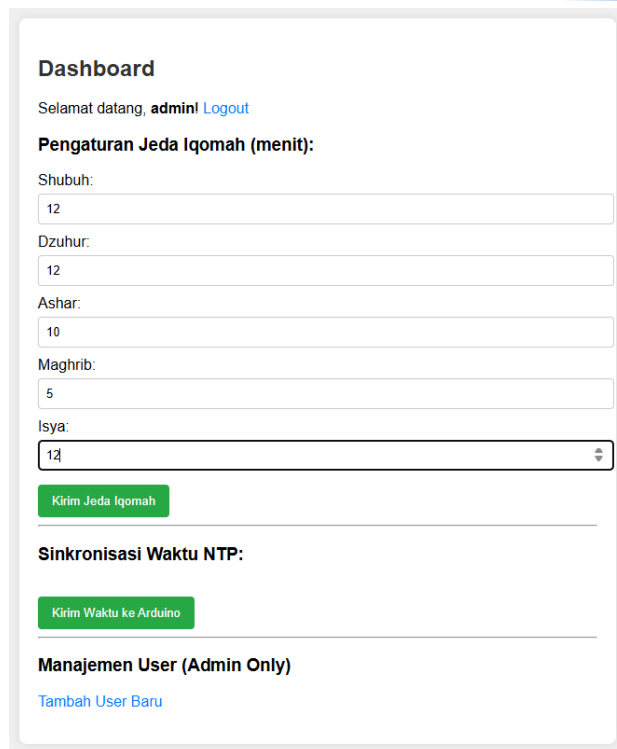


Gambar 13. Tampilan Hasil Pengaturan Sinkronisasi Waktu NTP Pada Alat

3.7. Pengujian Waktu Jeda Iqomah

Pengujian pengaturan waktu jeda iqomah dilakukan dengan mengirimkan data waktu jeda iqomah ke Arduino Mega2560 melalui ESP32. Pengaturan jeda iqomah dilakukan secara bersamaan untuk kesemua waktu sholat, yaitu waktu Shubuh, Dzuhur, Ashar, Maghrib dan Isya. Perubahan nilai waktu jeda iqomah sangat mudah dilakukan, cukup dengan menekan tombol panah atas maupun bawah disisi kanan *dashboard*, seperti ditunjukkan pada gambar 14.

Setelah tombol **Kirim Jeda Iqomah** diklik, maka nilai waktu Shubuh, Dzuhur, Ashar, Maghrib dan Isya akan dikirimkan ke Arduino Mega2560. Setelah dilakukan pengujian, maka diperoleh hasil bahwa nilai jeda iqomah yang dikirimkan ke Arduino Mega2560 sama dengan nilai yang ditampilkan oleh alat Jadwal sholat Digital. Dengan demikian pengaturan waktu jeda iqomah melalui ESP32 yang dikirim pada *smartphone* dalam bekerja dengan baik.



Dashboard

Selamat datang, **admin!** [Logout](#)

Pengaturan Jeda Iqomah (menit):

Shubuh:

Dzuhur:

Ashar:

Maghrib:

Isya:

[Kirim Jeda Iqomah](#)

Sinkronisasi Waktu NTP:

[Kirim Waktu ke Arduino](#)

Manajemen User (Admin Only)

[Tambah User Baru](#)

Gambar 14. Tampilan *Dashboard* Pengaturan Sinkronisasi Waktu Jeda Iqomah

4. Kesimpulan

Hasil penelitian dan pengujian menunjukkan bahwa sistem penjadwalan sholat digital yang dikembangkan untuk Masjid Rahmatan Lil 'Alamin (RLA) Universitas Putra Indonesia YPTK Padang berhasil meningkatkan akurasi waktu melalui integrasi modul ESP32 yang terhubung ke server NTP untuk sinkronisasi otomatis. Dukungan konektivitas Wi-Fi memungkinkan penyesuaian jadwal secara fleksibel melalui *smartphone*, sehingga mempermudah pengelolaan operasional harian. Pengujian terhadap fitur pengiriman data jeda iqomah dari *smartphone* ke ESP32 juga memberikan hasil yang memuaskan. Data dapat diterima dengan cepat dan akurat, lalu diteruskan ke Arduino Mega2560 melalui komunikasi UART tanpa adanya keterlambatan signifikan. Rata-rata waktu respons sistem tercatat kurang dari satu detik, membuktikan bahwa komunikasi antar modul berlangsung secara stabil dan andal. Selama pengujian, sistem tetap berfungsi secara lokal meskipun terjadi gangguan jaringan. Informasi waktu tetap ditampilkan oleh RTC internal, dan fitur sinkronisasi waktu otomatis akan aktif kembali saat jaringan tersedia. Hal ini menunjukkan bahwa sistem memiliki ketahanan terhadap kondisi jaringan yang tidak stabil, menjadikannya lebih andal untuk digunakan dalam jangka panjang di lingkungan masjid. Hingga laporan ini disusun, sistem jadwal sholat digital yang dirancang telah digunakan secara aktif di Masjid Rahmatan Lil 'Alamin dan beroperasi dengan lancar selama lebih dari 5 tahun, yaitu sejak tahun 2019 hingga tahun 2025. Temuan ini menegaskan bahwa pendekatan berbasis *Internet of Things* (IoT) yang diterapkan mampu memberikan solusi praktis, efisien, dan berkelanjutan untuk pengelolaan jadwal ibadah secara digital, serta memiliki potensi besar untuk diterapkan di masjid-masjid lain di masa mendatang.

Deklarasi

Kontribusi Penulis. Semua penulis berkontribusi secara bersama-sama dengan kontributor utama dalam artikel ini. Semua penulis membaca dan menyetujui versi akhir dari artikel yang diajukan.

Pernyataan Pendanaan. Tidak ada penulis yang menerima dana atau hibah dari lembaga atau badan pendanaan untuk penelitian ini.

Konflik Kepentingan. Penulis menyatakan tidak ada konflik kepentingan.

Informasi Tambahan. Tidak ada informasi tambahan dalam artikel ini.

Daftar Pustaka

- [1] Y. H. Kanoi, S. Abdussamad, and S. W. Dali, "Perancangan Jam Digital Waktu Sholat Menggunakan Arduino Uno," *Jambura J. Electr. Electron. Eng.*, vol. 1, no. 2, pp. 32–39, Oct. 2019.
- [2] W. Helma, H. Alam, J. W. Syafrawali, and R. . Bangun, "Rancang Bangun Running Text Led Display Jadwal Waktu Sholat Berbasis Arduino Uno Sebagai Media Informasi," *J. Electr. Technol.*, vol. 5, no. 2, pp. 2502–3624, 2020.
- [3] S. Amri, W. M. Faizal, A. Azizul, P. Almubarak, and N. Azima, "Implementasi Jadwal Shalat Digital Dengan Menggunakan Running Text Di Mushalla Kampus Politeknik Negeri Bengkulu," *Tanjak J. Pengabd. Kpd. Masy.*, vol. 5, no. 1, Jun. 2024.
- [4] I. H. Kurniawan, L. Hayat, and A. Fauzan, "Implementasi Teknologi Jadwal Waktu Sholat dan Media Informasi Digital Berbasis Mikrokontroler di Wilayah Pimpinan Ranting Muhammadiyah Klahang, Kecamatan Sokaraja, Kabupaten Banyumas," *J. Pengabd. Tek. dan Sains*, vol. 5, no. 1, p. 25, Jan. 2025.
- [5] S. Rosad, A. Yudhana, and A. Fadlil, "Jadwal Sholat Digital Menggunakan Metode Ephemeris Berdasarkan Titik Koordinat Smartphone," *IT J. Res. Dev.*, vol. 3, no. 2, pp. 30–43, Jan. 2019.
- [6] S. Sarifudin, M. Manshur, and A. Tirtana, "Penggunaan Komunikasi Bluetooth Pada Smartphone Android Untuk Pengiriman Data Pada Jam Digital Berbasis Arduino," *J. ELTIKOM*, vol. 1, no. 2, pp. 102–112, Jan. 2018.
- [7] A. W. Putra, R. Nuryanto, and A. Tafrikhatin, "Fitur Peningkat Kegiatan Masjid Dengan Kontrol Wi-Fi Berbasis ESP-32 Pada Jam Digital," *J. Pendidik. Tambusai*, vol. 5, no. 3, pp. 6177–6187, 2021.
- [8] M. Anton and M. H. Basri, "Perancangan+Jam+Istisana+Otomatis+Menggunakan+Running+Text+dan+Speaker+Sebagai+Alat+Bantu+Waktu+Sholat+Di+Masjid+Nurul+Hidayah+Al-Taqwa," vol. 5, no. 2, pp. 43–48, 2020.
- [9] E. Naf'an, "Akurasi Sistem Penjadwalan Sholat Digital Menggunakan Arduino Sebagai Pengendali," *J. Sistim Inf. dan Teknol.*, vol. 1, no. 4, pp. 77–84, 2019.
- [10] Emil Naf'an, "PENGEMBANGAN SISTEM PENJADWALAN SHOLAT DIGITAL DENGAN PENGIRIMAN PESAN PADA RUNNING TEXT MELALUI KOMUNIKASI BLUETOOTH," *J. Teknoif Tek. Inform. Inst. Teknol. Padang*, vol. 9, no. 1, pp. 47–54, Apr. 2021.
- [11] O. Barybin, E. Zaitseva, and V. Brazhnyi, "Testing the Security ESP32 Internet of Things Devices," in *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, 2019, pp. 143–146.
- [12] M. Anusha, P. B. Kumar, V. Akhil, M. Gouthami, M. . Chinnaiah, and S. Shaik, "Internet of Things (IOT) based energy monitoring with ESP 32 and using Thingspeak," in *2024 10th International Conference on Communication and Signal Processing (ICCSPP)*, 2024, pp. 1383–1387.
- [13] T.-C. Hou, L.-H. Liu, Y.-K. Lan, Y.-T. Chen, and Y.-S. Chu, "An Improved Network Time Protocol for Industrial Internet of Things," *Sensors*, vol. 22, no. 13, p. 5021, Jul. 2022.
- [14] T.Mangayarkarasi, Rajakumar.P, Sivabalan.M, and N. Gokul.R, "Arduino Based Smart Metering System with Smart 2560," in *2024 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, 2024, pp. 1–4.
- [15] L. A. S. I. Akbar, M. S. Iqbal, D. F. Budiman, A. S. Rachman, G. W. Wiriasto, and . S., "PEMANFAATAN RUNNING TEXT SEBAGAI ALAT BANTU INFORMASI WAKTU SHOLAT DI MASJID YAYASAN DARUL HIKMAH LOMBOK TENGAH," *J. Bakti Nusa*, vol. 2, no. 1, pp. 9–14, Feb. 2021.
- [16] I. H. Kurniawan, L. Hayat, and D. N. K. Hardani, "Rancang Bangun Teknologi Penampil dan Peningkat Waktu Sholat Digital Di Lingkungan Pimpinan Cabang Muhammadiyah Baturaden," *J. Pengabd. Tek. dan Sains*, vol. 2, no. 01, Feb. 2022.
- [17] A. Drymonitis, "Introduction to Arduino," 2024, pp. 67–134.
- [18] Arduino, "ARDUINO MEGA 2560 REV3," 2019.

-
- [19] U. N. Malang, "Pengembangan Modul Pembelajaran Antarmuka dan Komunikasi Data Menggunakan Protokol MQTT dan ESP32," vol. 24, no. 3, pp. 392–407, 2024.
- [20] F. A. Aryatama and S. Samsugi, "Sistem Keamanan Kendaraan Bermotor Dengan ESP32 Menggunakan Kontrol Android," *SMATIKA J.*, vol. 14, no. 01, pp. 167–181, Jul. 2024.
- [21] R. Sonwane, A. Deshmukh, and S. Choudhary, "UART (Universal Asynchronous Receiver Transmitter) for Serial Data Communication: Design and Implementation on FPGA Platform," in *2023 2nd International Conference on Futuristic Technologies (INCOFT)*, 2023, pp. 1–4.
- [22] R. B. Chithra, S. A R, N. E. Mujassim, M. Gupta, and P. K. N, "Design and Implementation of UART With Effective Serial Communication," in *2024 Asia Pacific Conference on Innovation in Technology (APCIT)*, 2024, pp. 1–5.
- [23] Z. Didi and I. El Azami, "IoT, Comparative Study Between the Use of Arduino Uno, Esp32, and Raspberry pi in Greenhouses," 2022, pp. 718–726.
- [24] M. Sarosa *et al.*, "Air Cleaning System Based On The Internet Of Things (IoT)," in *2023 International Conference on Electrical and Information Technology (IEIT)*, 2023, pp. 367–371.
- [25] P. Bellini, P. Nesi, and G. Pantaleo, "IoT-Enabled Smart Cities: A Review of Concepts, Frameworks and Key Technologies," *Appl. Sci.*, vol. 12, no. 3, p. 1607, Feb. 2022.