
RANCANG BANGUN SMART AUDIO KAMPUS BERBASIS ESP32 DAN DFPLAYER DENGAN SISTEM PENJADWALAN OTOMATIS BERBASIS WEB

Muh Ahsan^{1*}, Muh. Ramli², Nur Inda³, Ika Handayani⁴

¹Informatika, Institut Teknologi dan Bisnis Muhammadiyah Polewali Mandar, Sulawesi Barat, Indonesia

²Informatika, Institut Teknologi dan Bisnis Muhammadiyah Polewali Mandar, Sulawesi Barat, Indonesia

³Informatika, Institut Teknologi dan Bisnis Muhammadiyah Polewali Mandar, Sulawesi Barat, Indonesia

⁴Informatika, Institut Teknologi dan Bisnis Muhammadiyah Polewali Mandar, Sulawesi Barat, Indonesia

¹ahsan@itbmpolman.ac.id,

²ramli@itbmpolman.ac.id,

³nurinda@itbmpolman.ac.id,

⁴ika.inf25@itbmpolman.ac.id

*Corresponding Author:

ahsan@itbmpolman.ac.id

(Received: 23-06-2026, Accepted: 28-06-2026, Published: 30-06-2026)

Abstrak

Sistem audio kampus yang dioperasikan secara manual sering menimbulkan keterlambatan pemutaran, ketidaksesuaian jadwal, dan ketergantungan terhadap operator, sehingga diperlukan sistem otomatis yang tepat waktu dan mudah dikelola. Penelitian ini bertujuan merancang dan membangun sistem Smart Audio Kampus berbasis ESP32 dan DFPlayer Mini dengan penjadwalan otomatis berbasis web. Metode Research and Development (R&D) diterapkan melalui tahapan identifikasi masalah, studi literatur, analisis kebutuhan, perancangan, implementasi perangkat keras dan lunak, pengujian Black Box Testing, serta analisis hasil. Sistem menggunakan ESP32 sebagai pusat kendali, DFPlayer Mini sebagai pemutar audio, Network Time Protocol (NTP) untuk sinkronisasi waktu, dan web server untuk pengaturan serta monitoring. Hasil penelitian menunjukkan sistem berhasil memutar audio otomatis sesuai jadwal yang diatur melalui antarmuka web, menampilkan waktu real-time, status DFPlayer Mini, audio sedang diputar, dan countdown menuju jadwal berikutnya. Sistem juga menyediakan mode operasi OFF, Senin–Jumat, dan Setiap Hari yang dapat disesuaikan dengan kebutuhan kampus. Seluruh fungsi utama sistem berjalan baik berdasarkan pengujian Black Box Testing. Dengan demikian, sistem ini menjadi solusi efektif dalam meningkatkan efisiensi pengelolaan audio kampus dan mendukung implementasi Smart Campus berbasis Internet of Things.

Kata Kunci: Smart Audio Kampus; ESP32; DFPlayer Mini; *Internet of Things*; Penjadwalan Otomatis

Abstract

Manually operated campus audio systems often cause delayed playback, scheduling inaccuracies, and dependency on operators, necessitating an automated system for timely and efficient audio management. This study aims to design and develop a Smart Campus Audio System based on ESP32 and DFPlayer Mini with a web-based automatic scheduling system. The Research and Development (R&D) method was employed, encompassing problem identification, literature review, requirements analysis, system design, hardware and software implementation, Black Box Testing, and result evaluation. The system was developed using an ESP32 microcontroller as the main controller, DFPlayer Mini as the audio playback module, Network Time Protocol (NTP) for time synchronization, and a web server for configuration and monitoring. The results demonstrate that the system successfully plays audio automatically according to predefined schedules configured through the web interface. The system displays real-time clock information, DFPlayer Mini status, currently playing audio, and countdown to upcoming schedules. Additionally, the system provides operational modes including OFF, Monday–Friday, and Everyday modes that can be adjusted based on campus requirements. Based on Black Box Testing, all

major system functions operated properly according to the design specifications. The proposed ESP32 and DFPlayer Mini-based Smart Campus Audio System with web-based automatic scheduling serves as an effective solution for improving campus audio management efficiency and supporting the implementation of an IoT-based Smart Campus environment.

Keywords: Smart Campus Audio; ESP32; DFPlayer Mini; Internet of Things; Automatic Scheduling

1. PENDAHULUAN

Perkembangan teknologi Internet of Things (IoT) telah mendorong transformasi digital di berbagai sektor, termasuk lingkungan perguruan tinggi dalam mewujudkan konsep *Smart Campus*. Salah satu fasilitas yang berperan penting dalam mendukung aktivitas kampus adalah sistem audio yang digunakan untuk menyampaikan informasi, pengumuman, pemutaran lagu kebangsaan, himne institusi, serta pemberitahuan waktu tertentu kepada sivitas akademika. Namun, pada umumnya sistem audio kampus masih dioperasikan secara manual oleh petugas menggunakan perangkat komputer atau pemutar audio konvensional. Metode ini memiliki kelemahan berupa ketergantungan terhadap operator, potensi keterlambatan pemutaran, risiko kesalahan manusia (*human error*), serta kurang efisien dalam pengelolaan jadwal berulang setiap hari, yang berdampak pada ketepatan penyampaian informasi dan efektivitas operasional sistem audio kampus.

Perkembangan mikrokontroler modern membuka peluang untuk mengatasi permasalahan tersebut melalui sistem audio otomatis berbasis IoT. ESP32 menjadi pilihan utama karena memiliki kemampuan komunikasi WiFi dan Bluetooth terintegrasi, konsumsi daya rendah, serta mendukung implementasi *web server* secara langsung [1]. Untuk pemutaran audio, DFPlayer Mini menawarkan solusi yang tepat sebagai modul pemutar MP3 yang mampu membaca file dari microSD dan dikendalikan melalui komunikasi serial UART, dengan ukuran kecil dan biaya implementasi relatif rendah [2]. Sinkronisasi waktu yang akurat untuk mendukung penjadwalan otomatis dapat diimplementasikan menggunakan *Network Time Protocol* (NTP), yang memungkinkan perangkat memperoleh informasi waktu melalui internet tanpa pengaturan manual [3].

Penelitian terdahulu telah menunjukkan berbagai implementasi sistem audio berbasis mikrokontroler. Mengembangkan sistem alarm otomatis berbasis jadwal menggunakan DFPlayer Mini yang mampu memutar informasi suara secara terprogram [4]. Membangun sistem bel sekolah berbasis IoT dengan integrasi ESP32 dan modul audio untuk meningkatkan efektivitas pengelolaan jadwal di lingkungan pendidikan [5]. Penelitian lain oleh [6] mengimplementasikan sistem pengumuman otomatis menggunakan ESP8266 dan DFPlayer Mini dengan pengendalian melalui aplikasi berbasis Android.

Namun demikian, sebagian besar penelitian yang telah dilakukan masih memiliki keterbatasan, antara lain: (1) belum menyediakan fitur monitoring perangkat secara *real-time* melalui antarmuka *web*, (2) tidak mengimplementasikan sinkronisasi waktu berbasis internet untuk memastikan akurasi penjadwalan, (3) belum memiliki sistem pengaturan jadwal yang fleksibel melalui *web server* internal, (4) tidak menyediakan informasi status pemutaran audio yang dapat dipantau langsung oleh pengguna, dan (5) belum mengakomodasi pengaturan mode operasi berdasarkan hari kerja secara terprogram. Kesenjangan penelitian (*research gap*) ini menunjukkan bahwa sistem audio kampus yang ada belum terintegrasi secara menyeluruh dalam hal otomatisasi penjadwalan, monitoring, dan pengendalian berbasis *web*.

Berdasarkan kesenjangan tersebut, penelitian ini mengusulkan pembangunan sistem *Smart Audio* Kampus berbasis ESP32 dan DFPlayer Mini dengan sistem penjadwalan otomatis berbasis *web*. Kebaruan (*novelty*) penelitian ini terletak pada integrasi fitur-fitur unggulan yang belum tergabung dalam satu sistem pada penelitian sebelumnya, yaitu: (a) sinkronisasi waktu menggunakan NTP untuk akurasi penjadwalan, (b) pengaturan jadwal melalui antarmuka *web browser* yang dapat diakses dari berbagai perangkat, (c) monitoring status DFPlayer secara *real-time* meliputi informasi audio yang sedang diputar, (d) *countdown* menuju jadwal berikutnya, serta (e) pengaturan mode operasi berdasarkan hari kerja (OFF, Senin–Jumat, dan Setiap Hari). Metode yang digunakan menggabungkan mikrokontroler ESP32 sebagai pusat kendali, DFPlayer Mini sebagai modul pemutar audio, dan protokol NTP untuk sinkronisasi waktu, dengan antarmuka *web server* yang berjalan langsung pada ESP32 sehingga tidak memerlukan perangkat tambahan untuk pengendalian.

Penelitian ini bertujuan untuk: (1) merancang dan membangun sistem *Smart Audio* Kampus berbasis ESP32 dan DFPlayer Mini, (2) mengimplementasikan sistem penjadwalan otomatis menggunakan sinkronisasi waktu berbasis NTP, (3) mengembangkan antarmuka berbasis *web* untuk pengaturan jadwal dan

pengendalian sistem audio, (4) mengimplementasikan fitur monitoring status perangkat secara *real-time* melalui *web browser*, dan (5) melakukan pengujian terhadap kinerja dan keandalan sistem yang dikembangkan. Dengan adanya sistem ini, pengelolaan audio kampus diharapkan dapat dilakukan secara lebih efektif, efisien, akurat, dan mendukung implementasi konsep *Smart Campus* berbasis IoT dengan biaya implementasi yang relatif rendah.

Tinjauan Pustaka

a) Internet of Things (IoT)

Internet of Things (IoT) merupakan konsep yang memungkinkan berbagai perangkat fisik terhubung melalui jaringan internet untuk melakukan pertukaran data secara otomatis tanpa memerlukan interaksi manusia secara langsung. Penelitian sebelumnya yang telah dilakukan oleh peneliti juga telah berhasil membuat tampilan aplikasi pengukuran tinggi dan berat badan ideal berbasis Internet of Things [7]. Teknologi Internet of Things memungkinkan berbagai perangkat terhubung melalui jaringan internet untuk melakukan monitoring dan pertukaran data secara real-time [8]. Implementasi Internet of Things tidak hanya diterapkan pada sistem monitoring lingkungan dan industri, tetapi juga telah banyak digunakan pada sistem keamanan dan pengelolaan fasilitas berbasis mikrokontroler ESP32 yang mampu melakukan monitoring serta pengiriman informasi secara real-time melalui jaringan internet [9]

b) Smart kampus

Smart Campus merupakan implementasi teknologi digital dan Internet of Things dalam lingkungan perguruan tinggi untuk meningkatkan efektivitas pengelolaan fasilitas, layanan akademik, dan aktivitas operasional kampus. Smart Campus memanfaatkan berbagai perangkat cerdas yang saling terhubung untuk mendukung proses monitoring, kontrol, dan pengambilan keputusan berbasis data.

c) Mikrokontroler ESP32

ESP32 merupakan mikrokontroler yang dikembangkan oleh Espressif Systems dan dirancang untuk mendukung berbagai aplikasi Internet of Things. ESP32 banyak digunakan dalam pengembangan sistem Internet of Things karena memiliki kemampuan komunikasi nirkabel yang terintegrasi serta mendukung implementasi sistem monitoring secara real-time. Pemanfaatan ESP32 pada sistem monitoring dan kontrol berbasis IoT menunjukkan bahwa mikrokontroler ini mampu memberikan kinerja yang baik dalam pengelolaan data dan komunikasi perangkat secara berkelanjutan [10]. ESP32 memiliki prosesor dual-core, modul WiFi dan Bluetooth terintegrasi, serta berbagai antarmuka komunikasi seperti UART, SPI, I2C, PWM, dan ADC. ESP32 tidak hanya digunakan sebagai pengendali perangkat IoT tetapi juga mampu mendukung aplikasi monitoring berbasis jaringan secara real-time [11]. ESP32 digunakan pada sistem monitoring, otomasi, smart home, smart campus, dan berbagai aplikasi IoT karena memiliki kemampuan komunikasi jaringan yang baik dengan biaya implementasi yang relatif rendah [12], dan juga ESP32 banyak digunakan dalam sistem monitoring dan otomasi karena mendukung konektivitas WiFi, web server, serta komunikasi data secara real-time [13]. ESP32 merupakan mikrokontroler yang memiliki modul WiFi dan Bluetooth terintegrasi sehingga sangat sesuai digunakan pada aplikasi Internet of Things [14].

d) DFPlayer Mini

DFPlayer Mini merupakan modul audio MP3 yang mampu memutar file suara secara langsung dari media penyimpanan microSD. Modul ini menggunakan komunikasi serial UART sehingga dapat dengan mudah diintegrasikan dengan berbagai jenis mikrokontroler. DFPlayer Mini banyak digunakan dalam sistem alarm otomatis, sistem informasi suara, media pembelajaran interaktif, dan perangkat audio berbasis mikrokontroler karena memiliki ukuran yang kecil, harga yang relatif murah, serta kemudahan integrasi dengan sistem tertanam [4]

e) Web Server pad ESP32

Web Server merupakan layanan yang memungkinkan perangkat memberikan informasi kepada pengguna melalui browser menggunakan protokol HTTP. ESP32 memiliki kemampuan untuk menjalankan web server internal sehingga tidak memerlukan server eksternal [15].

f) Network Time Protokol (NTP)

NTP (Network Time Protocol) adalah teknologi yang digunakan untuk penyebaran tanda waktu melalui jaringan internet, menggunakan standar komunikasi [6]. Penggunaan NTP memungkinkan perangkat memperoleh waktu yang selalu diperbarui secara otomatis sehingga tidak memerlukan penyesuaian waktu secara manual sebagaimana yang umum terjadi pada sistem berbasis RTC [3].

g) Preferences

Preferences merupakan library penyimpanan data permanen yang tersedia pada ESP32. Library ini memungkinkan data konfigurasi disimpan ke dalam flash memory sehingga tetap tersimpan meskipun perangkat dimatikan.

h) Sistem Penjadwalan Otomatis Audio

Penerapan sistem penjadwalan otomatis pada lingkungan pendidikan dapat meningkatkan efisiensi pengelolaan waktu dan mengurangi ketergantungan terhadap operator. Sistem yang mengintegrasikan ESP32, DFPlayer Mini, dan aplikasi monitoring mampu menjalankan pemutaran audio secara otomatis sesuai jadwal yang telah ditentukan serta memberikan fleksibilitas dalam pengelolaan sistem [5].

2. METODE PENELITIAN

2.1 Jenis Penelitian

Penelitian ini menggunakan metode *Research and Development* (R&D) atau penelitian dan pengembangan. Metode R&D dipilih karena penelitian ini tidak hanya bertujuan untuk menganalisis suatu fenomena, tetapi juga menghasilkan sebuah produk berupa sistem *Smart Audio* Kampus berbasis ESP32 dan DFPlayer Mini yang dapat digunakan untuk mengotomatisasi pemutaran audio sesuai jadwal yang telah ditentukan [1]. Metode ini meliputi proses analisis kebutuhan, perancangan sistem, pembangunan perangkat keras dan perangkat lunak, implementasi, serta pengujian sistem untuk memastikan bahwa sistem yang dikembangkan dapat berfungsi sesuai dengan tujuan penelitian.

2.2 Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan di lingkungan kampus Institut Teknologi dan Bisnis Muhammadiyah Polewali Mandar sebagai lokasi implementasi dan pengujian sistem *Smart Audio* Kampus. Dilaksanakan selama ± 3 bulan dari bulan April hingga Juni 2026. Rincian kegiatan penelitian disajikan pada Tabel 3.1. Tabel 3.1 Waktu Penelitian

No.	Kegiatan	Bulan
1	Studi Literatur	April 2026
2	Analisis Kebutuhan Sistem	April 2026
3	Perancangan Sistem	April 2026
4	Pembuatan Perangkat Keras	Mei 2026
5	Pengembangan Perangkat Lunak	Mei 2026
6	Pengujian Sistem	Juni 2026
7	Analisis Hasil	Juni 2026
8	Penyusunan Laporan	Juni 2026

2.3 Alat dan Bahan Penelitian

2.3.1 Alat Penelitian

Peralatan yang digunakan dalam penelitian ini ditunjukkan pada Tabel 3.2.

Tabel 3.2 Alat Penelitian

No	Alat	Fungsi
1	Laptop/PC	Pengembangan program dan pengujian sistem
2	Arduino IDE	
3	Smartphone	Pemrograman ESP32
4	Kabel USB	Pengujian web server
5	Multimeter	Upload program ke ESP32
6	Obeng	Pengukuran tegangan dan pengecekan rangkaian
		Pemasangan Komponen

2.3.2 Bahan Penelitian

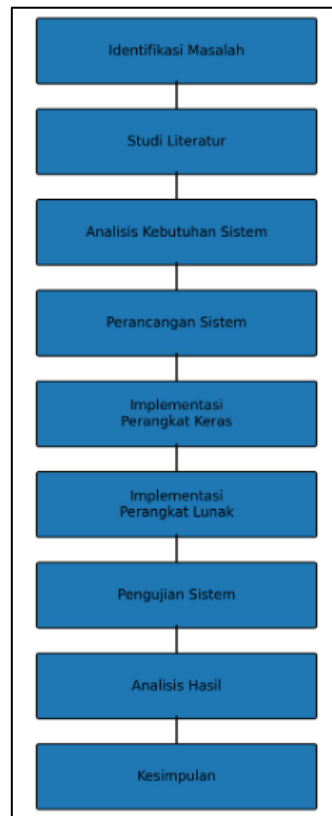
Bahan-bahan yang digunakan dalam penelitian ini disajikan pada Tabel 3.3.

Tabel 3.3 Bahan Penelitian

No	Bahan	Fungsi
1	ESP32 DevKit V1	Mikrokontroler utama
2	DFPlayer Mini	Modul pemutar audio
3	MicroSD Card	Penyimpanan file audio
4	Speaker Aktif	Output suara
5	Power Supply 5V	Sumber daya sistem
6	Kabel Jumper	Penghubung antar komponen

2.4 Tahapan Penelitian

Tahapan penelitian dilakukan secara sistematis mulai dari identifikasi masalah hingga pengujian sistem.



Gambar 3.1 Tahapan Penelitian

Sumber : Hasil Perancangan Peneliti (2026)

Tahapan penelitian dilaksanakan secara sistematis melalui sembilan langkah berurutan. Proses diawali dengan identifikasi masalah melalui observasi sistem audio kampus yang masih manual dan tidak terjadwal, kemudian dilanjutkan dengan studi literatur untuk mempelajari IoT, ESP32, DFPlayer Mini, NTP, dan penelitian terdahulu yang relevan. Setelah itu, dilakukan analisis kebutuhan untuk menentukan spesifikasi perangkat keras dan lunak, diikuti perancangan sistem yang mencakup desain arsitektur perangkat keras (konfigurasi pin ESP32-DFPlayer Mini), perangkat lunak (fungsi-fungsi pemrograman), dan antarmuka web dashboard. Tahap implementasi dibagi menjadi dua bagian, yaitu perakitan perangkat keras (menghubungkan ESP32, DFPlayer Mini, speaker, microSD, dan power supply) dan pengembangan perangkat lunak menggunakan Arduino IDE untuk membangun fitur koneksi WiFi, sinkronisasi NTP, web server, penjadwalan, monitoring, countdown, dan mode operasi. Selanjutnya, sistem diuji menggunakan metode Black Box Testing terhadap delapan skenario uji meliputi koneksi WiFi, akurasi NTP, pemutaran file audio, akses dashboard, penjadwalan otomatis, dan monitoring real-time. Hasil pengujian dianalisis untuk mengevaluasi keberhasilan sistem berdasarkan delapan indikator, mulai dari koneksi WiFi hingga stabilitas operasi. Tahap akhir adalah penarikan kesimpulan untuk menjawab rumusan masalah dan tujuan penelitian, serta penyusunan saran untuk pengembangan selanjutnya.

2.5 Indikator Keberhasilan Sistem

Sistem dinyatakan berhasil apabila memenuhi indikator keberhasilan sebagai berikut:

- 1) ESP32 terhubung ke jaringan WiFi.
- 2) Sinkronisasi NTP berhasil dilakukan dengan tingkat akurasi waktu yang memadai.
- 3) Jadwal dapat disimpan dan dibaca kembali dari memori *Preferences*.
- 4) DFPlayer Mini dapat memutar audio sesuai jadwal yang telah ditentukan.
- 5) *Dashboard* dapat diakses melalui *browser* pada berbagai perangkat.
- 6) Monitoring status perangkat berjalan normal dan menampilkan informasi yang akurat.
- 7) *Countdown* jadwal tampil dengan benar sesuai waktu yang tersisa.

- 8) Sistem mampu beroperasi secara stabil selama periode pengujian.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Penelitian

Penelitian ini menghasilkan sebuah sistem *Smart Audio* Kampus berbasis ESP32 dan DFPlayer Mini yang mampu melakukan pemutaran audio secara otomatis berdasarkan jadwal yang telah ditentukan melalui antarmuka berbasis *web*. Sistem yang dikembangkan dilengkapi dengan fitur sinkronisasi waktu menggunakan *Network Time Protocol* (NTP), pengaturan jadwal audio, monitoring status perangkat, *countdown* jadwal berikutnya, serta kontrol manual audio melalui *dashboard web*.

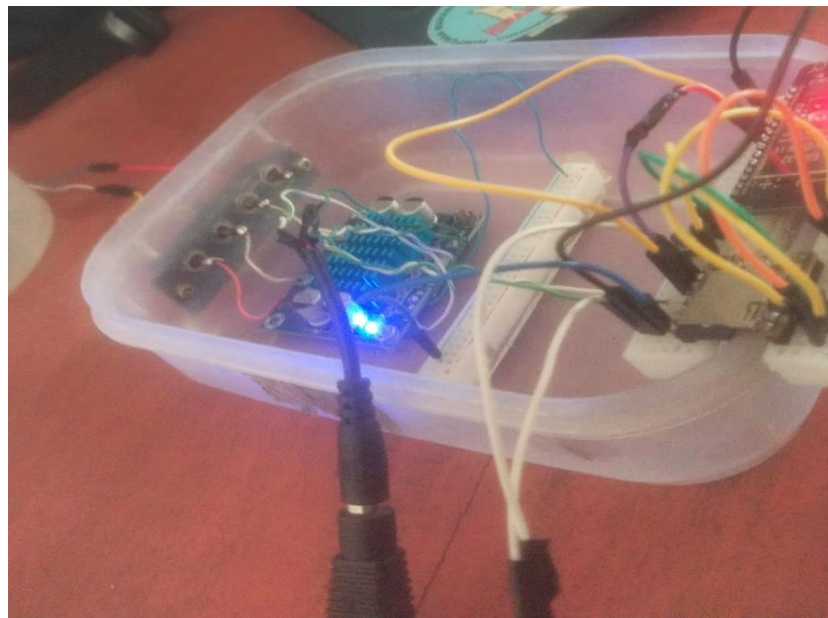
3.1.1 Implementasi Perangkat Keras

Implementasi perangkat keras dilakukan dengan menghubungkan ESP32 sebagai pusat kendali sistem dengan DFPlayer Mini sebagai modul pemutar audio. Komponen yang digunakan disajikan pada Tabel 3.1.

Tabel 3.1 Komponen yang Digunakan

No	Komponen	Fungsi
1	ESP32 DevKit V1	Pengendali utama sistem
2	DFPlayer Mini	Pemutar file audio
3	MicroSD Card	Penyimpanan file audio
4	Speaker	Output suara
5	Power Supply	Sumber daya sistem

Hasil rangkaian perangkat keras yang telah berhasil dirancang pada penelitian ini ditunjukkan pada Gambar 3.1.



Gambar 3.1 Implementasi Perangkat Keras
(Sumber: Hasil Perancangan Peneliti, 2026)

3.1.2 Implementasi Perangkat Lunak

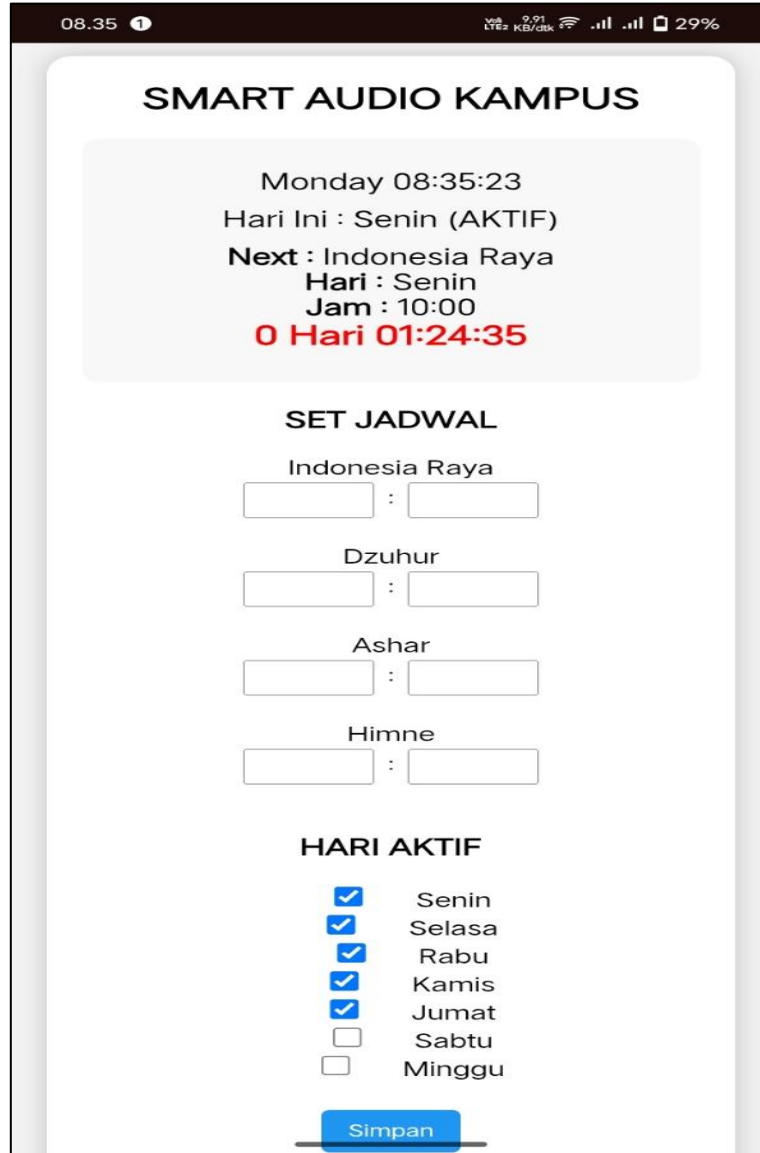
Perangkat lunak dikembangkan menggunakan bahasa pemrograman C++ pada Arduino IDE. Fitur utama yang berhasil diimplementasikan meliputi:

- 1) Koneksi WiFi untuk komunikasi data dan akses *web server*
- 2) Sinkronisasi waktu menggunakan NTP
- 3) *Web server* internal yang berjalan pada ESP32
- 4) Penyimpanan jadwal menggunakan *Preferences* (*Non-Volatile Storage*)
- 5) Monitoring status DFPlayer Mini

- 6) *Countdown* jadwal berikutnya
- 7) Mode operasi OFF, Senin–Jumat, dan Setiap Hari
- 8) Pemutaran audio otomatis berdasarkan jadwal
- 9) Pemutaran audio manual melalui *dashboard*

3.1.3 Implementasi Dashboard Web

Berikut adalah hasil *dashboard* yang berhasil dibuat pada penelitian ini.



Gambar 3.2 Tampilan Dashboard Web
(Sumber: Hasil Perancangan Peneliti, 2026)

Dashboard web merupakan antarmuka utama yang digunakan untuk mengelola dan memantau sistem. *Dashboard* ini dirancang menggunakan teknologi HTML dan CSS yang dijalankan langsung pada ESP32 melalui fitur *web server* internal sehingga dapat diakses menggunakan perangkat komputer maupun *smartphone* yang terhubung pada jaringan yang sama. Melalui *dashboard* ini, pengguna dapat melakukan konfigurasi sistem secara mudah tanpa memerlukan aplikasi tambahan.

Tampilan *dashboard* menampilkan informasi waktu secara *real-time* yang diperoleh dari sinkronisasi NTP. Selain itu, sistem juga menampilkan informasi hari aktif sistem, status operasional perangkat, serta jadwal audio berikutnya yang akan diputar. Pada bagian informasi jadwal berikutnya ditampilkan nama audio yang akan diputar, hari pelaksanaan, waktu pemutaran, dan *countdown* menuju jadwal tersebut. Fitur *countdown* ini memudahkan pengguna dalam memantau waktu pemutaran audio berikutnya secara langsung.

Dashboard juga menyediakan menu pengaturan jadwal audio yang terdiri atas jadwal pemutaran lagu Indonesia Raya, peringatan Dzuhur, peringatan Ashar, dan Himne Kampus. Pengguna dapat mengatur jam dan menit pemutaran masing-masing audio sesuai kebutuhan operasional kampus. Setelah pengaturan dilakukan, data jadwal akan disimpan ke dalam memori *Preferences* pada ESP32 sehingga tetap tersimpan meskipun perangkat dimatikan atau mengalami *restart*.

Selain pengaturan jadwal, *dashboard* dilengkapi dengan fitur pengaturan hari aktif yang memungkinkan pengguna menentukan hari operasional sistem. Pengguna dapat memilih hari Senin sampai Minggu melalui *checkbox* yang tersedia. Fitur ini memungkinkan sistem hanya menjalankan pemutaran audio pada hari-hari tertentu, misalnya hanya pada hari kerja (Senin–Jumat), sehingga sistem dapat menyesuaikan dengan aktivitas akademik kampus dan menghindari pemutaran audio pada hari libur.

3.2 Hasil Pengujian Sistem

Pengujian dilakukan menggunakan metode *Black Box Testing* untuk memastikan seluruh fungsi sistem berjalan sesuai dengan tujuan penelitian [1]. Pengujian mencakup koneksi WiFi, sinkronisasi NTP, pemutaran audio DFPlayer Mini, penjadwalan audio, dan fungsionalitas *dashboard web*.

3.2.1 Pengujian Koneksi WiFi

Tabel 3.2 Hasil Pengujian Koneksi WiFi

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	ESP32 terhubung ke WiFi	Mendapatkan alamat IP	Berhasil	Valid

Berdasarkan hasil pengujian pada Tabel 3.2, ESP32 berhasil terhubung ke jaringan WiFi dan memperoleh alamat IP sehingga *dashboard* dapat diakses melalui *browser*. Keberhasilan koneksi WiFi merupakan prasyarat utama bagi sistem untuk mengakses NTP dan menjalankan *web server*.

3.2.2 Pengujian Sinkronisasi Waktu NTP

Tabel 3.3 Hasil Pengujian Sinkronisasi NTP

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Sinkronisasi waktu	Waktu sesuai waktu aktual	Berhasil	Valid

Hasil pengujian pada Tabel 3.3 menunjukkan bahwa ESP32 berhasil melakukan sinkronisasi waktu dengan NTP Server sehingga sistem dapat menjalankan jadwal audio dengan tingkat akurasi yang baik. Akurasi waktu sangat penting untuk memastikan pemutaran audio terjadi tepat pada jadwal yang telah ditentukan.

3.2.3 Pengujian DFPlayer Mini

Tabel 3.4 Hasil Pengujian DFPlayer Mini

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Indonesia Raya	Audio diputar	Berhasil	Valid
2	Dzuhur	Audio diputar	Berhasil	Valid
3	Ashar	Audio diputar	Berhasil	Valid
4	Himne ITBM Polman	Audio diputar	Berhasil	Valid

Berdasarkan Tabel 3.4, DFPlayer Mini berhasil membaca *file* audio dari MicroSD Card dan memutarnya sesuai perintah yang diberikan oleh ESP32. Seluruh *file* audio yang diuji dapat diputar dengan kualitas suara yang baik melalui speaker.

3.2.4 Pengujian Penjadwalan Audio

Tabel 3.5 Hasil Pengujian Penjadwalan Audio

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	10:00	Indonesia Raya	Berhasil diputar	Valid
2	12:00	Dzuhur	Berhasil diputar	Valid

3	15:15	Ashar	Berhasil diputar	Valid
4	16:00	Himne ITBM Polman	Berhasil diputar	Valid

Hasil pengujian pada Tabel 3.5 menunjukkan bahwa sistem mampu menjalankan pemutaran audio secara otomatis sesuai jadwal yang telah ditentukan melalui *dashboard web*. Sistem membandingkan waktu aktual yang diperoleh dari NTP dengan jadwal yang tersimpan, dan ketika waktu sesuai, perintah pemutaran dikirimkan ke DFPlayer Mini.

3.2.5 Pengujian Dashboard Web

Tabel 3.6 Hasil Pengujian Dashboard Web

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Menampilkan waktu	Berjalan normal	Berhasil	Valid
2	Menampilkan status DFPlayer	Berjalan normal	Berhasil	Valid
3	Menampilkan countdown	Berjalan normal	Berhasil	Valid
4	Pemutaran manual	Berjalan normal	Berhasil	Valid

Seluruh fitur *dashboard web* berhasil berjalan sesuai dengan rancangan sistem seperti ditunjukkan pada Tabel 3.6. *Dashboard* dapat diakses melalui *browser* dan menampilkan informasi *real-time* tanpa perlu melakukan *refresh* halaman.

3.3 Pembahasan

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, sistem Smart Audio Kampus berbasis ESP32 dan DFPlayer Mini berhasil dikembangkan sesuai dengan tujuan penelitian. ESP32 mampu berfungsi sebagai pusat kendali yang mengintegrasikan seluruh komponen sistem, meliputi sinkronisasi waktu menggunakan NTP, penyimpanan konfigurasi menggunakan Preferences, web server internal, serta pengendalian DFPlayer Mini. Seluruh komponen perangkat keras dan perangkat lunak dapat bekerja secara terintegrasi untuk menghasilkan sistem yang stabil dan andal, sebagaimana dibuktikan melalui delapan skenario pengujian Black Box Testing yang seluruhnya berhasil dengan status valid.

Penggunaan NTP terbukti mampu meningkatkan akurasi waktu pemutaran audio dibandingkan penggunaan RTC (Real-Time Clock) yang memerlukan pengaturan manual [1]. Sistem memperoleh waktu dari server NTP melalui koneksi internet, sehingga tidak memerlukan penyesuaian waktu secara berkala oleh operator. Hal ini sejalan dengan penelitian Rahman dan Widodo (2023) yang menyatakan bahwa NTP memberikan tingkat akurasi waktu yang lebih baik dibandingkan metode manual [2]. Akurasi waktu ini sangat krusial karena sistem membandingkan waktu aktual dengan jadwal yang tersimpan, dan ketika waktu sesuai, perintah pemutaran dikirimkan ke DFPlayer Mini, sehingga menjamin ketepatan pemutaran audio pada jadwal yang telah ditentukan.

Dashboard web yang dikembangkan memberikan kemudahan bagi pengguna dalam melakukan konfigurasi jadwal dan monitoring sistem secara real-time. Pengguna dapat mengakses dashboard melalui perangkat apa pun yang terhubung ke jaringan yang sama tanpa memerlukan aplikasi tambahan, berbeda dengan penelitian sebelumnya yang menggunakan aplikasi berbasis Android yang memerlukan instalasi dan kompatibilitas perangkat tertentu [3]. Dashboard menampilkan informasi waktu real-time dari sinkronisasi NTP, status operasional perangkat, jadwal audio berikutnya, serta countdown menuju jadwal tersebut. Fitur penyimpanan jadwal menggunakan Preferences memastikan data tetap tersimpan meskipun perangkat dimatikan atau mengalami restart, sehingga pengguna tidak perlu melakukan konfigurasi ulang secara berkala.

Fitur countdown jadwal berikutnya, status DFPlayer Mini, dan mode operasi berbasis hari kerja menjadi nilai tambah yang tidak ditemukan secara lengkap pada penelitian-penelitian sebelumnya. Penelitian oleh Khairunnisa dan Hutasuhut (2023) hanya berfokus pada pemutaran audio berdasarkan jadwal tanpa menyediakan monitoring real-time [4], sementara penelitian oleh Prasetya dkk. (2025) mengembangkan sistem bel sekolah dengan integrasi IoT namun belum dilengkapi dengan fitur countdown dan mode operasi fleksibel [5]. Mode operasi OFF, Senin–Jumat, dan Setiap Hari memberikan fleksibilitas bagi pengelola kampus dalam menyesuaikan sistem dengan aktivitas akademik, di mana mode Senin–Jumat

memungkinkan sistem hanya beroperasi pada hari kerja sehingga menghindari pemutaran audio pada hari libur, yang sangat relevan untuk lingkungan pendidikan. Berdasarkan Tabel 4.7, penelitian ini memiliki keunggulan dibandingkan penelitian terdahulu dalam hal kelengkapan fitur, khususnya pada sinkronisasi waktu NTP, web dashboard, monitoring real-time, countdown, dan mode operasi berbasis hari kerja. Integrasi fitur-fitur tersebut dalam satu sistem menjadikan penelitian ini lebih komprehensif dalam mendukung implementasi Smart Campus berbasis IoT.

Sistem menunjukkan tingkat keandalan yang baik dengan seluruh fungsi utama berjalan sesuai yang diharapkan selama periode pengujian tanpa mengalami kegagalan atau kesalahan pada pemutaran audio. DFPlayer Mini berhasil membaca dan memutar seluruh file audio yang diuji, yaitu Indonesia Raya, Dzhur, Ashar, dan Himne ITBM Polman, dengan kualitas suara yang baik melalui speaker. Dengan demikian, seluruh tujuan penelitian yang meliputi perancangan sistem, implementasi NTP, pengembangan dashboard web, implementasi monitoring, dan pengujian kinerja sistem berhasil dicapai, sehingga sistem Smart Audio Kampus ini dapat menjadi solusi efektif dalam meningkatkan efisiensi pengelolaan audio kampus serta mendukung implementasi konsep Smart Campus berbasis Internet of Things.

3.4 Ketercapaian Tujuan Penelitian

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, seluruh tujuan penelitian berhasil dicapai seperti disajikan pada Tabel 3.7.

Tabel 3.7 Ketercapaian Tujuan Penelitian

No	Tujuan Penelitian	Keterangan
1	Merancang sistem Smart Audio Kampus	Tercapai
2	Mengimplementasikan NTP	Tercapai
3	Mengembangkan dashboard web	Tercapai
4	Mengimplementasikan monitoring sistem	Tercapai
5	Menguji kinerja sistem	Tercapai

4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan pada penelitian ini, dapat disimpulkan bahwa sistem Smart Audio Kampus berbasis ESP32 dan DFPlayer Mini dengan sistem penjadwalan otomatis berbasis web berhasil dikembangkan dan berfungsi sesuai dengan tujuan penelitian. Sistem mampu melakukan pemutaran audio secara otomatis sesuai jadwal yang telah ditentukan melalui antarmuka web, dengan ESP32 berfungsi sebagai pusat kendali dan DFPlayer Mini sebagai modul pemutar audio. Pemanfaatan Network Time Protocol (NTP) sebagai sumber waktu utama berhasil diimplementasikan sehingga sistem mampu menjaga ketepatan waktu pemutaran audio tanpa memerlukan modul Real Time Clock (RTC) tambahan, serta memudahkan proses sinkronisasi waktu secara otomatis melalui jaringan internet. Dashboard berbasis web yang dikembangkan berhasil digunakan untuk pengelolaan sistem secara real-time, meliputi pengaturan jadwal audio, pemilihan mode operasi, pemutaran audio manual, serta monitoring kondisi sistem tanpa memerlukan perangkat lunak tambahan. Fitur monitoring yang terdiri dari status koneksi WiFi, status DFPlayer Mini, informasi audio yang sedang diputar, dan countdown menuju jadwal berikutnya berhasil diimplementasikan sehingga memudahkan pengguna dalam memantau kondisi sistem secara langsung melalui browser. Sistem juga berhasil menerapkan mode operasi yang fleksibel, yaitu mode OFF, mode Senin–Jumat, dan mode Setiap Hari, yang memungkinkan sistem menyesuaikan jadwal pemutaran audio dengan kebutuhan operasional kampus. Berdasarkan hasil pengujian Black Box Testing, seluruh fungsi utama sistem yang meliputi koneksi WiFi, sinkronisasi waktu NTP, penyimpanan konfigurasi menggunakan Preferences, pemutaran audio otomatis, pemutaran audio manual, serta monitoring sistem melalui dashboard web berjalan sesuai dengan rancangan. Dengan demikian, sistem Smart Audio Kampus yang dikembangkan mampu meningkatkan efisiensi pengelolaan audio kampus karena mengurangi ketergantungan terhadap operator dan memberikan kemudahan dalam pengaturan jadwal secara terpusat melalui antarmuka web.

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat digunakan sebagai bahan pengembangan pada penelitian selanjutnya. Pertama, menambahkan fitur kalender akademik sehingga sistem dapat secara otomatis menyesuaikan jadwal pemutaran audio berdasarkan hari libur nasional, hari besar keagamaan, dan agenda akademik kampus. Kedua, mengintegrasikan sistem dengan database berbasis cloud sehingga pengelolaan jadwal dapat dilakukan dari lokasi yang berbeda melalui jaringan internet tanpa harus berada pada jaringan lokal kampus. Ketiga, menambahkan fitur manajemen audio yang memungkinkan pengguna mengunggah, menghapus, dan mengelola file audio secara langsung melalui dashboard web tanpa

perlu melepas kartu microSD. Keempat, mengembangkan sistem notifikasi berbasis Telegram, WhatsApp, atau email untuk memberikan informasi apabila terjadi gangguan pada sistem, seperti kegagalan koneksi WiFi, kegagalan sinkronisasi NTP, atau DFPlayer Mini yang tidak terdeteksi. Kelima, menambahkan dukungan multi-zona audio sehingga sistem dapat mengatur pemutaran audio pada beberapa area kampus secara terpisah sesuai kebutuhan. Keenam, mengintegrasikan sistem dengan perangkat Internet of Things (IoT) lainnya di lingkungan kampus sebagai bagian dari pengembangan konsep Smart Campus yang lebih luas. Ketujuh, menambahkan fitur pencadangan waktu menggunakan modul RTC sebagai sistem cadangan apabila koneksi internet tidak tersedia sehingga jadwal pemutaran audio tetap dapat berjalan dengan baik.

DAFTAR PUSTAKA

- [1] G. D. C. Bertoli, G. V. C. Fernandes, P. H. B. Monici, and C. H. De Araujo, "Design and implementation of intelligent packet filtering in IoT microcontroller-based devices," vol. 14, no. 8, pp. 1–11, 2021.
- [2] Y. Mirin, "Rancang Bangun Tempat Sampah otomatis Berbasis IoT Berbasis Gerakan, Suara dan Ketinggian Sampah," 2025.
- [3] A. L. Kulasekera, "A Systems-Engineered ESP32 DAQ Architecture and FAIR Data Workflow for Small-Scale Wind Turbine Performance Measurement in Tropical Environments," pp. 1–15, 2026.
- [4] I. Khairunnisa and A. Hutasuhut, "Prototype Smart Alarm Automated System Berbasis DFPlayer Mini untuk Mengefisiensikan Jadwal Waktu," vol. IX, no. 2, pp. 34–41, 2023.
- [5] A. Prasetya, Hardiansyah, Surojo, and P. Silalahi, "Sistem Bel dan Jadwal Pelajaran di Sekolah Berbasis IoT," 2025.
- [6] A. B. Setiawan, "Kajian Literatur Sinkronisasi Waktu Dengan Network Time Protocol Untuk Pemantauan Aktivitas Jaringan Telekomunikasi," vol. 5, no. 2, 2015, doi: 10.17933/jppi.2015.050200.
- [7] M. Ahsan, S. Sahibu, Nasrullah, M. Ramli, and A. A. Haidar, "BERAT BADAN IDEAL MENGGUNAKAN SENSOR Rancang Bangun Pengukuran Tinggi dan Berat Badan Ideal Menggunakan Sensor," vol. 16, no. 2, pp. 81–91, 2025.
- [8] L. Maulana *et al.*, "Implementasi Sistem Presensi Berbasis Iot (Internet Of Things) Menggunakan Platform Blynk Di Smk Cendikia Rancakalong," vol. 17, no. 1, pp. 14–29, 2024.
- [9] C. Supriadi, D. Setiawan, L. Adi, and S. Fegi, "INOVASI IOT UNTUK PENGELOLAAN DAN KEAMANAN RUANG ARSIP : IMPLEMENTASI ESP32 DENGAN SENSOR API DAN SUHU DHT11," vol. 1, no. 4, pp. 79–85, 2024.
- [10] A. Ziad and E. Darnila, "Development and Implementation of an ESP32 Microcontroller and Monitoring System for Smart Door Lock Using RFID Sensor for E-KTP ID and Fingerprint Based on the Internet of Things," vol. 00023, pp. 1–11, 2024.
- [11] A. Riwawan, Safaruddin, Ridwang, and Adriani, "Perancangan Prototype Sistem Deteksi Wajah untuk Keamanan Rumah menggunakan Smart Bell dengan ESP 32 Cam," vol. 2, no. 2, 2024.
- [12] V. B. Vales *et al.*, "Fine Time Measurement for the Internet of Things : A Practical Approach Using ESP32," vol. 9, no. 19, pp. 18305–18318, 2022.
- [13] Waryanto, Mursalim, T. Aprilia, and M. Khozin, "Implementasi Ganda Modul ESP32 untuk Kontrol Pintu Otomatis dengan Monitoring Realtime dan Autolock Terjadwal melalui Web dan Chatbot Dual Implementation of ESP32 Modules for Automated Door Control with Real-Time Monitoring and Scheduled Autolock through," vol. 7, no. 1, pp. 1–10, 2025.
- [14] M. Nizam, H. Yuana, F. T. Informasi, U. Islam, B. Blitar, and M. D. Switch, "MIKROKONTROLER ESP 32 SEBAGAI ALAT MONITORING PINTU BERBASIS WEB," vol. 6, no. 2, pp. 767–772, 2022.
- [15] A. Akbar, Zaenuddin, Z. Mutaqin, and L. D. Samsumar, "IoT-Based Smart Room Using Web Server-Based Esp32 Microcontroller Smart Room Berbasis IoT Menggunakan Microcontroller Esp32 Berbasis Web Server," vol. 1, no. 2, pp. 91–98, 2022.