

**SISTEM PEMANGGILAN ANTREAN OTOMATIS
PADA POLI PUSKESMAS BERBASIS
*INTERNET OF THINGS***

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma IV Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh:

Septriandi

NIM : 1052227

Abi Ari Sandy

NIM : 1052201

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2025**

LEMBAR PENGESAHAN

SISTEM PEMANGGILAN ANTREAN OTOMATIS PADA POLI PUSKESMAS BERBASIS *INTERNET OF THINGS*

Oleh:

Septriandi/1052227

Abi Ari Sandy/1052201

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma IV Politeknik Manufaktur Negeri Bangka Belitung

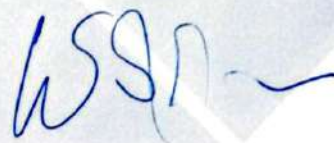
Menyetujui,

Pembimbing 1

Pembimbing 2



Zanu Saputra, M.Tr.T.



Yus Dwi Yanti, S.P., M.Si

Penguji 1

Penguji 2



Yudhi, M.T.



Valentin Vina Ratnapuri, S.Kel., M.Si

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan dibawah ini :

Nama Mahasiswa 1 : Septriandi NIM. 1052227

Nama Mahasiswa 2 : Abi Ari Sandy NIM. 1052201

Dengan Judul : Sistem Pemanggilan Antrean Otomatis pada Poli
Puskesmas

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata di kemudian hari kami melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 03 Juli 2025

Nama Mahasiswa

1. Septriandi

2. Abi Ari Sandy

Tanda Tangan

.....
.....

ABSTRAK

Di Puskesmas, layanan yang cepat dan efektif sangat penting untuk meningkatkan standar perawatan medis. Pasien sering mengeluh tentang sistem antrean manual karena waktu tunggu yang tidak jelas dan informasi yang tidak lengkap. Di poliklinik Puskesmas, penelitian ini bertujuan untuk mengembangkan dan menerapkan sistem pemanggilan antrean otomatis berbasis *Internet of Things* (IoT). Sistem ini secara otomatis memanggil pasien berdasarkan nomor antrean mereka menggunakan modul suara, tampilan digital, dan perangkat mikrokontroler. Petugas dapat lebih mudah mengawasi proses layanan dengan menggunakan *platform* berbasis web untuk mendapatkan dan memantau data antrean. Hasil pengujian menunjukkan bahwa sistem ini mampu mengurangi waktu tunggu pasien, meningkatkan efisiensi kinerja petugas, serta menciptakan suasana pelayanan yang lebih tertib dan informatif. Dengan penerapan sistem ini, diharapkan kualitas pelayanan di Puskesmas dapat meningkat secara signifikan.

Kata kunci : Antrean, IoT, Mikrokontroler, Sistem Suara, Otomatis.

ABSTRACT

In Community Health Centers, prompt and effective service is crucial to raising the standard of medical care. Patients frequently complain about manual queue systems because of unclear waiting times and incomplete information. In the Community Health Center polyclinic, this study intends to develop and deploy an Internet of Things (IoT)-based automated queue calling system. This system automatically calls patients based on their queue numbers using a voice module, digital display, and microcontroller device. Officers can more easily oversee the service process by using a web-based platform to obtain and monitor queue data. Testing results show that the system effectively reduces patient waiting time, improves staff performance efficiency, and creates a more orderly and informative service environment. The implementation of this system is expected to significantly enhance the quality of healthcare services at Puskesmas.

Keywords : Queue, IoT, Microcontroller, Voice System, Automatic.

KATA PENGANTAR

Assalamu'alaikum Warohmatullahi Wabarakatuh

Puji syukur kami panjatkan kepada ALLAH SWT. atas segala rahmat dan hidayah-Nya, sehingga kami dapat menyelesaikan karya tulis Proyek Akhir ini dengan baik. Proyek Akhir ini merupakan salah satu syarat yang harus dipenuhi untuk menyelesaikan program pendidikan Diploma IV di Politeknik Manufaktur Negeri Bangka Belitung.

Selama lima semester menempuh pendidikan, kami telah belajar banyak hal, dan melalui Proyek Akhir ini, kami mencoba menerapkan ilmu yang telah kami dapatkan, baik dari perkuliahan maupun pengalaman selama melaksanakan Program Kerja Lapangan. Proyek Akhir ini tidak hanya berupa pembuatan alat, tetapi juga penulisan makalah yang menjadi bukti perjalanan akademis kami.

Pada kesempatan ini, kami ingin mengucapkan terima kasih sebesar-besarnya kepada semua pihak yang telah membantu dan mendukung kami dalam menyelesaikan Proyek Akhir ini. Secara khusus, kami berterima kasih kepada:

1. Orang tua dari saudara Septriandi Yakni Bapak Endang Mulyajaya dan Ibu Asnah, serta orang tua dari saudara Abi Ari Sandy yakni Bapak Maruddin dan Ibu Yantina beserta kakak Cika Melia yang senantiasa memberikan doa dan motivasi yang luar biasa sehingga penelitian ini terselesaikan.
2. Bapak Zanu Saputra, M.Tr.T. selaku Pembimbing 1 dan Ibu Yus Dwi Yanti, S.P, M.Si selaku Pembimbing 2, yang telah meluangkan waktu, tenaga, dan pikiran untuk membimbing kami. Beliau memberikan banyak arahan, saran, dan solusi yang sangat membantu dalam penyelesaian Proyek Akhir ini.
3. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
4. Seluruh staf pengajar di Politeknik Manufaktur Negeri Bangka Belitung yang telah membagikan ilmu dan pengalaman selama kami menempuh pendidikan.
5. Rekan-rekan mahasiswa yang telah membantu dan mendukung selama proses pengerjaan Proyek Akhir ini.

6. Semua pihak yang telah memberikan bantuan, baik secara langsung maupun tidak langsung, yang tidak dapat kami sebutkan satu per satu.

Kami menyadari bahwa Proyek Akhir ini masih jauh dari sempurna karena kami adalah manusia biasa yang tidak luput dari kesalahan. Oleh karena itu, kami sangat mengharapkan saran, kritik, dan masukan yang membangun dari pembaca untuk perbaikan di masa mendatang. Semoga Proyek Akhir ini dapat bermanfaat dan menambah wawasan bagi semua pihak, khususnya rekan-rekan mahasiswa. Terima kasih, dan mohon maaf jika ada kekurangan dalam penulisan karya tulis ini. Wassalamu'alaikum Warohmatullahi Wabarakatuh.



Sungailiat, 21 Juli 2025

Penulis

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	ii
ABSTRAK.....	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan Proyek Akhir	2
1.4 Manfaat.....	2
BAB II DASAR TEORI	3
2.1 Antrean	3
2.2 <i>Internet of Things (IoT)</i>	4
2.3 Mikrokontroler ESP32 Sebagai Pengendali	4
2.4 <i>Thermal Printer Mini</i>	5
2.5 <i>Push Button</i>	6
2.6 DFPlayer Mini MP3	7
2.7 Micro SD-Card	8
2.8 <i>Liquid Crystal Display (LCD)</i>	9
2.9 Sistem Suara dalam Pemanggil Antrean	9
2.10 VS Code sebagai Pengembangan Sistem	10
2.11 Panel P10 dalam Tampilan Visual	11
BAB III METODE PELAKSANAAN	12

3.1	Desain Sistem	12
3.2	Metode Penelitian.....	12
3.2.1	Studi Kasus.....	12
3.2.2	Observasi Lapangan.....	13
3.2.3	Pengujian Alat.....	13
3.3	Arsitektur Sistem.....	13
3.4	Perancangan Alat.....	13
3.4.1	Blok Diagram Sistem.....	14
3.4.2	Rangkaian Perangkat Keras (Hardware)	14
3.5	Integrasi IoT	15
3.6	Pengujian	15
BAB IV	PEMBAHASAN	16
4.1	Deskripsi Alat Sistem Antrean Otomatis pada Puskemas.....	16
4.2	Diagram Blok Sistem Antrean Otomatis pada Puskemas	17
4.3	Prinsip Kerja Sistem Antrean Otomatis pada Puskemas.....	17
4.4	Perancangan Hardware Sistem Antrean Otomatis pada Puskemas.....	18
4.4.1	Perancangan Mekanik.....	18
4.4.2	Perancangan Elektrik.....	19
4.5	Pembuatan Hardware Sistem Antrean Otomatis pada Puskemas	20
4.5.1	Pembuatan Hardware Mekanik	20
4.5.2	Pembuatan Hardware Elektrik.....	22
4.6	Hasil Pengujian.....	22
4.6.1	Konfigurasi Web.....	22
4.6.2	Konektifitas Printer Thermal	23
4.6.3	Koversi Suara	24
4.6.3	Tampilan LCD	27
4.6.4	Tampilan Panel P10	28
BAB V	KESIMPULAN	35
5.1	Kesimpulan	35
5.2	Saran	36
DAFTAR PUTAKA		38

DAFTAR TABEL

Tabel	halaman
4.1. Hasil pengujian Web.....	23
4.2. Keberhasilan Printer	24
4.3. Suara Antrean pada jarak 5 meter	25
4.4. Suara Antrean pada jarak 10 meter	25
4.5. Suara Antrean pada jarak 15 meter	25
4.6. Keberhasilan Output LCD 4x20 I2C	27
4.7. Data untuk poli mata antrean 1	28
4.8. Data untuk poli mata antrean 2	29
4.9. Data untuk poli mata antrean 3	30
4.10. Data untuk poli gigi antrean 1	30
4.11. Data untuk poli gigi antrean 2	31
4.12. Data untuk poli gigi antrean 3	32
4.13. Data untuk poli anak antrean 1	32
4.14. Data untuk poli anak antrean 2	33
4.15. Data untuk poli anak antrean 3	34

DAFTAR GAMBAR

Gambar	halaman
2.1. <i>Internet of Things</i>	4
2.2. ESP 32	5
2.3. <i>Thermal Printer Mini</i>	6
2.4. <i>Push Button</i>	7
2.5. <i>Module DFPlayer Mini</i>	8
2.6. microSD Card	8
2.7. LCD I2C	9
2.8. Ilustrasi <i>Database</i>	9
2.9. Speaker	10
2.10. VS Code	10
2.11. Panel P10	11
3.1. <i>Flowchart</i> input dan output	13
4.1. Diagram Blok Sistem Antran Otomatis pada Puskemas	17
4.2. Tampak atas Desain box	19
4.3. Skematik Rangkaian Input	19
4.4. Skematik Rangkaian Output	20
4.5. Tampak atas Hardware Mekanik	20
4.6. Tampak bawah Hardware Mekanik	21
4.7. Tampak samping Hardware Mekanik	21
4.8. Hasil perakitan semua komponen	22
4.9. Tampilan Interface Web	22
4.11. Kertas Cetakan	23
4.12. Tampilan LCD 4x20	27
4.13. Tampilan poli mata untuk antrean 1	28
4.14. Tampilan poli mata untuk antrean 2	29
4.15. Tampilan poli mata untuk antrean 3	29

4.16. Tampilan poli gigi untuk antrean 1	30
4.17. Tampilan poli gigi untuk antrean 2	31
4.18. Tampilan poli gigi untuk antrean 3	31
4.19. Tampilan poli anak untuk antrean 1	32
4.20. Tampilan poli anak untuk antrean 2	33
4.21. Tampilan poli anak untuk antrean 3	33



DAFTAR LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup

Lampiran 2 : Kode Arduino

Lampiran 3 : Kode VSCode



BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pusat Kesehatan Masyarakat (Puskesmas) merupakan fasilitas layanan kesehatan tingkat pertama yang memiliki peran strategis dalam menjamin akses masyarakat terhadap pelayanan kesehatan dasar. Sebagai unit pelaksana teknis dari Dinas Kesehatan Kabupaten/Kota, Puskesmas bertanggung jawab menyelenggarakan upaya promotif, preventif, kuratif, dan rehabilitatif yang merata dan terjangkau. Namun, dalam praktiknya, berbagai permasalahan sering muncul, terutama terkait manajemen antrean yang belum optimal. Sistem antrean manual yang masih umum digunakan menyebabkan penumpukan pasien, ketidakteraturan giliran, serta keterlambatan dalam proses pelayanan, yang berdampak pada kepuasan pengguna layanan [1].

Antrean yang panjang dan tidak terkontrol berpotensi menimbulkan tekanan psikologis bagi pasien, khususnya kelompok rentan seperti lansia dan anak-anak. Dalam banyak kasus, pasien bahkan kehilangan giliran karena ketidaktahuan akan informasi antrean atau harus meninggalkan lokasi akibat waktu tunggu yang terlalu lama [2]. Hal ini menunjukkan perlunya modernisasi sistem antrean yang tidak hanya mempermudah proses pelayanan, tetapi juga memperkuat transparansi dan akuntabilitas layanan. Teknologi *Internet of Things* (IoT) menjadi solusi yang tepat untuk menjawab tantangan ini, karena mampu menghubungkan berbagai perangkat seperti tombol pemanggil, modul suara, dan tampilan digital dalam satu sistem terintegrasi berbasis jaringan [3].

Mikrokontroler seperti ESP32, yang telah dilengkapi Wi-Fi dan Bluetooth internal, sangat cocok digunakan untuk membangun sistem antrean otomatis karena mendukung komunikasi nirkabel dan efisiensi daya. Dengan dukungan modul seperti printer thermal mini, DFPlayer Mini, dan display LED P10, sistem ini dapat mencetak tiket, memutar suara pemanggilan, serta menampilkan nomor antrean

secara *real-time*. Sistem ini dirancang untuk mengurangi waktu tunggu pasien, mempermudah kerja petugas, dan menyajikan informasi layanan dengan lebih jelas dan terstruktur [4]. Maka dari itu, perancangan dan implementasi sistem pemanggilan antrean berbasis mikrokontroler pada Puskesmas menjadi upaya strategis untuk meningkatkan efektivitas, efisiensi, dan mutu pelayanan kesehatan publik di era *digital*.

1.2. Perumusan Masalah

Adapun Perumusan masalah dari proyek akhir ini yaitu :

1. Bagaimana sistem akan memanggil antrean pasien secara otomatis?
2. Bagaimana keefektifan sistem kepada petugas dan pasien?

1.3. Tujuan Proyek Akhir

Sesuai dengan uraian latar belakang, maka tujuan yang ingin dicapai dari penelitian ini yaitu:

1. Merancang dan mengimplementasikan sistem pemanggil antrean berbasis mikrokontroler pada Puskesmas.
2. Meningkatkan kualitas pelayanan kesehatan di Puskesmas dengan pemanfaatan teknologi IoT untuk meningkatkan efisiensi dan efektivitas pelayanan kesehatan.

1.4. Manfaat

Manfaat dari Proyek Akhir ini adalah:

1. Sistem ini mempercepat pemanggilan pasien dan mengoptimalkan alur pelayanan di poli Puskesmas dengan mengurangi waktu tunggu yang tidak efektif, karena pemanggilan dapat dipantau secara otomatis.
2. Dengan sistem yang terstruktur dan informasi yang *real-time*, pasien akan merasa lebih nyaman dan dilayani secara profesional.
3. Data antrean terekam secara otomatis, sehingga memudahkan pihak manajemen dalam menganalisis durasi pelayanan, jumlah pasien harian, dan kebutuhan sumber daya.

BAB II

DASAR TEORI

2.1. Antrean

Antrean merupakan elemen penting dalam manajemen pelayanan publik yang berfungsi sebagai mekanisme pengaturan giliran secara adil dan terstruktur. Dalam kehidupan masyarakat modern, antrean hadir hampir di setiap sektor, mulai dari transportasi, layanan administrasi, hingga pelayanan kesehatan. Menurut Wulandari dan Supriadi, antrean mencerminkan sistem sosial yang mengatur keterbatasan sumber daya agar tetap dapat diakses secara merata oleh masyarakat [5]. Keberadaan antrean menuntut disiplin dan penghormatan terhadap hak pengguna layanan lain, sehingga keberhasilan sistem antrean tidak hanya ditentukan oleh sistem teknis, tetapi juga oleh kesadaran kolektif masyarakat.

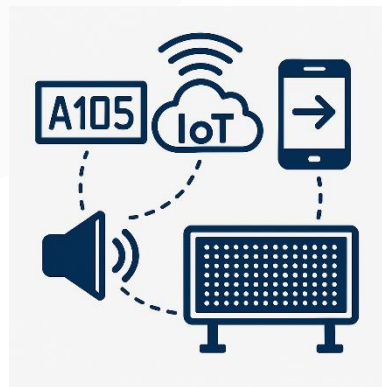
Namun, pengalaman mengantre yang buruk dapat menimbulkan ketidakpuasan pelanggan dan berpengaruh pada citra lembaga penyedia layanan. Perasaan frustrasi, bosan, atau tidak puas sering muncul ketika sistem antrean tidak berjalan dengan baik, seperti durasi tunggu yang terlalu lama atau tidak adanya kepastian giliran. Dalam studi yang dilakukan oleh Setiawan dan Halim, disebutkan bahwa sistem antrean manual rentan menimbulkan konflik sosial kecil serta menurunkan efisiensi pelayanan secara keseluruhan [6]. Hal ini menjadi lebih kompleks ketika terjadi pada sektor vital seperti layanan kesehatan, di mana waktu respons yang cepat sangat dibutuhkan oleh pasien.

Di fasilitas kesehatan seperti Puskesmas, antrean sering kali menjadi salah satu penyebab utama keterlambatan pelayanan dan rendahnya kepuasan pasien. Oleh karena itu, pengembangan sistem antrean otomatis berbasis teknologi menjadi kebutuhan yang mendesak. Penelitian oleh Pratama dan Lestari menunjukkan bahwa penerapan sistem antrean digital berbasis mikrokontroler dapat mengurangi waktu tunggu, meminimalisasi kesalahan pemanggilan, serta menciptakan lingkungan pelayanan yang lebih tertib dan transparan [7]. Otomatisasi sistem

antrean bukan hanya meningkatkan efisiensi operasional, tetapi juga memperkuat kepercayaan masyarakat terhadap institusi pelayanan publik.

2.2. *Internet of Things (IoT)*

Internet of Things (IoT) adalah teknologi yang memungkinkan perangkat fisik seperti sensor, mikrokontroler, dan modul komunikasi untuk saling terhubung melalui jaringan *internet* dan bertukar data secara otomatis. Dalam konteks sistem antrean otomatis, IoT berperan penting dalam mengintegrasikan berbagai komponen seperti tombol pemanggil, tampilan digital, dan modul suara agar dapat bekerja secara sinkron dan *real-time*. Dengan memanfaatkan konektivitas ini, proses pemanggilan antrean menjadi lebih efisien, terstruktur, dan mudah dipantau dari jarak jauh oleh petugas maupun pasien. Teknologi ini sangat relevan untuk meningkatkan kualitas pelayanan publik di fasilitas kesehatan seperti Puskesmas, karena mampu mengurangi waktu tunggu dan meminimalisir kesalahan manual dalam proses antrean [8].



Gambar 2.1. *Internet of Things*.

2.3. Mikrokontroler ESP32 Sebagai Pengendali

ESP32 adalah sebuah mikrokontroler berdaya rendah yang dilengkapi dengan Wi-Fi dan Bluetooth bawaan, dikembangkan oleh *Espressif Systems*, dan sering digunakan dalam berbagai proyek *Internet of Things (IoT)*. Chip ini memiliki prosesor *dual-core* Xtensa LX6 dengan kecepatan hingga 240 MHz, serta mendukung antarmuka komunikasi seperti UART, SPI, I2C, PWM, ADC, dan DAC, menjadikannya sangat fleksibel untuk berbagai kebutuhan sistem tertanam.

Dalam sistem antrean otomatis berbasis IoT, ESP32 berperan sebagai pusat pengendali yang mengatur komunikasi antara tombol input, tampilan LCD atau panel P10, modul suara DFPlayer Mini, serta koneksi ke antarmuka web atau aplikasi *mobile* [9]. Keunggulan lainnya adalah konsumsi daya yang rendah, fitur *deep sleep*, serta kapasitas memori dan kecepatan pemrosesan yang lebih tinggi dibandingkan generasi sebelumnya seperti ESP8266, menjadikannya sangat ideal untuk aplikasi yang memerlukan konektivitas *real-time* dan kontrol terintegrasi [9].



Gambar 2.2. ESP 32

Karena konektivitas *WiFi*-nya, yang memungkinkan transmisi data ke *platform cloud* seperti *Firebase*, mikrokontroler ini dipilih. Selain itu, ESP32 memiliki kemampuan pemrosesan yang cukup untuk menggabungkan perangkat *output* seperti LCD dan speaker serta sensor.

2.4. Thermal Printer Mini

Thermal printer mini adalah jenis printer kecil yang mencetak teks atau gambar menggunakan teknologi pemanasan langsung pada kertas termal, tanpa menggunakan tinta atau *toner*. Perangkat ini sangat cocok digunakan dalam sistem antrean otomatis, karena mampu mencetak nomor antrean secara cepat, sunyi, dan hemat energi. *Thermal printer* mini umumnya digunakan bersama mikrokontroler seperti Arduino atau ESP32, dengan koneksi serial UART atau USB TTL, dan mendukung lebar kertas standar 58 mm [10]. Dalam sistem antrean di Puskesmas, printer ini berfungsi mencetak tiket antrean pasien berisi informasi seperti nomor, waktu, dan tujuan layanan, yang kemudian digunakan sebagai identitas saat

pemanggilan berlangsung [11]. Keunggulannya mencakup ukuran yang ringkas, respons cepat, serta biaya operasional yang rendah karena tidak memerlukan tinta, menjadikannya solusi ideal untuk implementasi layanan publik berbasis IoT di ruang tunggu atau loket pendaftaran.



Gambar 2.3. *Thermal Printer Mini*

2.5. *Push Button*

Push button switch (saklar tombol tekan) adalah perangkat atau saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan (*unlock*), yaitu tidak mengunci. Sistem kerja *unlock* ini berarti tombol akan bekerja sebagai pemicu aliran arus listrik hanya saat ditekan, dan kembali ke kondisi semula (OFF) saat dilepas [12]. *Push button* umumnya digunakan sebagai input manual pada sistem elektronik berbasis mikrokontroler, termasuk dalam aplikasi seperti sistem antrean otomatis. Dalam sistem tersebut, *push button* berperan penting untuk memanggil antrean berikutnya secara *real-time*, di mana sinyal dari tombol akan dibaca oleh mikrokontroler seperti ESP32, lalu diteruskan ke modul suara, tampilan digital, atau *database* [13]. *Push button* tersedia dalam berbagai jenis, seperti *Normally Open* (NO) dan *Normally Closed* (NC), dan dapat dipasang dengan atau tanpa *pull-up* resistor, tergantung pada jenis logika yang digunakan dalam sistem.

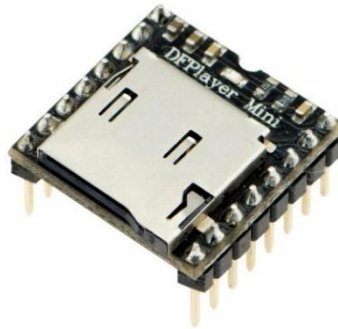


Gambar 2.4. *Push Button*

Push button switch hanya memiliki dua status sebagai perangkat penghubung atau pemutus yaitu ON dan OFF (1 dan 0). Karena semua perangkat listrik yang membutuhkan pasokan energi listrik pasti membutuhkan kondisi Hidup dan Mati, istilah "Hidup" dan "Mati" menjadi sangat penting.

2.6. DFPlayer Mini MP3

DFPlayer Mini adalah modul audio berukuran kecil dan berbiaya rendah yang dirancang untuk memutar *file* audio secara mandiri dari microSD card dalam format MP3, WAV, atau WMA. Modul ini sangat cocok digunakan dalam sistem berbasis mikrokontroler seperti Arduino atau ESP32 karena dapat langsung dikendalikan melalui komunikasi serial UART. Dalam sistem pemanggilan antrean otomatis, DFPlayer Mini berperan penting untuk memberikan pemanggilan suara otomatis berdasarkan nomor antrean yang disimpan sebagai file audio di microSD, sehingga pengguna tidak memerlukan komputer eksternal untuk pemutaran suara [14]. Modul ini mendukung koneksi langsung ke speaker aktif atau pasif, dan memiliki fitur *built-in amplifier* sederhana yang memungkinkan suara cukup keras untuk ruang publik seperti Puskesmas [15]. DFPlayer Mini sangat efisien dalam integrasi perangkat IoT karena hemat daya, mudah diprogram, serta memiliki dokumentasi dan komunitas pengguna yang luas di dunia *embedded system* [16].



Gambar 2.5. Module DFPlayer Mini

2.7. Micro SD-Card

MicroSD adalah kartu memori berukuran kecil yang digunakan sebagai media penyimpanan eksternal dalam berbagai perangkat elektronik, termasuk sistem berbasis mikrokontroler seperti Arduino dan ESP32. Dalam aplikasi sistem antrean otomatis, microSD berfungsi sebagai penyimpanan file audio (misalnya *file* .mp3 untuk pemanggilan nomor antrean) yang dibaca oleh modul seperti DFPlayer Mini [17]. Kartu microSD memiliki kapasitas penyimpanan mulai dari beberapa *megabyte* hingga puluhan *gigabyte*, serta mendukung format file FAT32 yang umum digunakan pada sistem tertanam. Namun, microSD tidak dapat dihubungkan langsung ke mikrokontroler; diperlukan modul antarmuka tambahan seperti microSD module atau DFPlayer Mini agar perangkat dapat membaca dan menulis data secara stabil.



Gambar 2.6. MicroSD Card

Dalam penelitian yang kami lakukan, kami menyimpan *file* audio dengan format mp3, sebagai contoh 0001.mp3, 0002.mp3, dan seterusnya sesuai dengan kebutuhan.

2.8. *Liquid Crystal Display (LCD)*



Gambar 2.7. LCD I2C

LCD 20x4 atau *Liquid Crystal Display* 20 karakter $\times$ 4 baris adalah jenis tampilan teks yang mampu menampilkan maksimal 80 karakter sekaligus, menjadikannya sangat ideal untuk aplikasi sistem informasi publik seperti antrean otomatis di Puskesmas. Dengan struktur 20 kolom dan 4 baris, LCD ini mampu menampilkan informasi yang lebih lengkap dan tidak terbatas seperti pada LCD 16x2, sehingga dapat menampilkan data seperti nomor antrean, nama pasien, poli tujuan, dan status layanan secara bersamaan [18]. Modul ini bekerja menggunakan *driver* HD44780 yang kompatibel dengan banyak mikrokontroler, dan umumnya dikombinasikan dengan modul I2C untuk menghemat jumlah pin yang digunakan pada papan kendali seperti Arduino atau ESP32 [19].

2.9. Sistem Suara dalam Pemanggil Antrean



Gambar 2.8. *Speaker*

Speaker merupakan perangkat *output* audio yang mengubah sinyal listrik menjadi gelombang suara melalui getaran membran. Dalam sistem pemanggilan antrean otomatis, *speaker* berfungsi untuk menyampaikan informasi suara seperti

pemanggilan nomor antrean, sehingga pasien dapat mengetahui giliran mereka secara akustik. Teknologi ini sangat penting untuk meningkatkan kenyamanan dan efektivitas pelayanan, terutama bagi pasien lansia atau mereka yang tidak memperhatikan tampilan visual seperti layar LCD. Sistem *speaker* yang digunakan biasanya terhubung dengan modul audio seperti DFPlayer Mini, yang membaca file suara dari microSD dan mengeluarkannya melalui *speaker* aktif atau *buzzer*. *Speaker* aktif sering dipilih karena memiliki penguat suara internal sehingga dapat menghasilkan suara yang lebih jelas dan keras tanpa memerlukan rangkaian tambahan [20]. Keandalan *speaker* dalam lingkungan pelayanan publik seperti Puskesmas menjadikannya komponen penting dalam sistem layanan berbasis *Internet of Things* (IoT) yang membutuhkan interaksi suara secara otomatis dan efisien [21].

2.10. VS Code sebagai Pengembangan Sistem



Gambar 2.9. VS Code

Visual Studio Code (VS Code) adalah sebuah editor kode sumber (source-code editor) lintas *platform* yang ringan namun sangat *powerful*, dikembangkan oleh Microsoft. VS Code mendukung berbagai bahasa pemrograman seperti JavaScript, Python, C++, dan C#, serta sangat populer dalam pengembangan aplikasi berbasis *Internet of Things* (IoT) karena kemampuannya diintegrasikan dengan *platform mikrokontroller* seperti Arduino dan ESP32 melalui ekstensi tambahan. Fitur-fitur unggulannya meliputi penyorotan sintaks (*syntax highlighting*), *auto-completion* (IntelliSense), *debugging* terintegrasi, manajemen proyek Git, serta *marketplace* ekstensi yang sangat kaya [22]. Dalam pengembangan sistem antrean otomatis berbasis ESP32, VS Code biasanya

digunakan bersama platformio atau Arduino *extension*, yang memudahkan pemrograman, kompilasi, dan unggah kode ke perangkat mikrokontroler [23]. Keunggulan VS Code yang ringan namun canggih membuatnya menjadi salah satu alat utama dalam pengembangan perangkat lunak modern, baik untuk kebutuhan *desktop*, *web*, maupun *embedded system*.

2.11. Panel P10 dalam Tampilan Visual



Gambar 2.10. Panel P10

Modul LED P10 adalah papan tampilan digital berbasis dot matrix yang terdiri dari susunan LED dengan pitch 10 mm, biasanya berukuran 32 kolom $\times$ 16 baris. Modul ini umum digunakan untuk menampilkan informasi secara *real-time* dan *visual*, seperti nomor antrean, informasi layanan, atau status pemanggilan dalam sistem antrean otomatis. Modul P10 mendukung pengendalian berbasis mikrokontroler seperti Arduino, ESP8266, atau ESP32, yang terhubung melalui port data (biasanya melalui koneksi serial *shift register* 74HC595 atau HUB75). Keunggulan modul ini meliputi keterbacaan tinggi, daya tahan yang baik, serta kemampuan menampilkan teks atau angka yang dapat diatur melalui pemrograman [24]. Dalam konteks pelayanan publik seperti Puskesmas, modul LED P10 memungkinkan informasi antrean ditampilkan dengan jelas dan menarik di ruang tunggu, sehingga meningkatkan keteraturan dan mengurangi kecemasan pasien yang menunggu giliran [25].

BAB III

METODE PELAKSANAAN

Puskesmas merupakan pusat kesehatan yang dimana menjadi titik bagi masyarakat untuk mendapatkan kemudahan akses pemeriksaan diri, namun ada beberapa keluhan akan kurangnya keefisiensi antrean yang diterapkan pada beberapa puskesmas terutama pada pedalaman. Berdasarkan *problem* inilah kami menggunakan metode yang terintegritas dengan akar *solvingnya*.

3.1. Desain Sistem

Perangkat IoT digunakan dalam arsitektur sistem ini untuk mengawasi dan mengendalikan antrean di pusat kesehatan. Jumlah poliklinik akan ditampilkan pada panel p10, dan sistem akan menggunakan tombol tekan untuk memanggil antrean baru setelah data pasien sebelumnya direkam dan dimasukkan ke dalam basis data. *Database* kemudian akan dipanggil melalui pengeras suara.

3.2. Metode Penelitian

Dalam penelitian ini, data yang peneliti kumpulkan untuk mendukung perancangan dan implementasi sistem pemanggil antrean otomatis berbasis *Internet of Things* (IoT) di poli puskesmas. Metode pengumpulan data yang peneliti gunakan meliputi:

3.2.1. Studi Kasus

Metode penelitian yang digunakan adalah metode studi kasus. Dimana metode ini tidak selalu menggunakan metode kuantitatif, hanya saja masih ada yang menggunakan metode kuantitatif. Pendekatan studi kasus merupakan prosedur penelitian menyeluruh yang berkonsentrasi pada satu aspek kasus yang dapat mengoptimalkan analisisnya. Biasanya terdapat beberapa kasus yang akan diteliti, akan lebih baik jika penelitian hanya berfokus pada masalah yang terjadi.

3.2.2. Observasi Lapangan

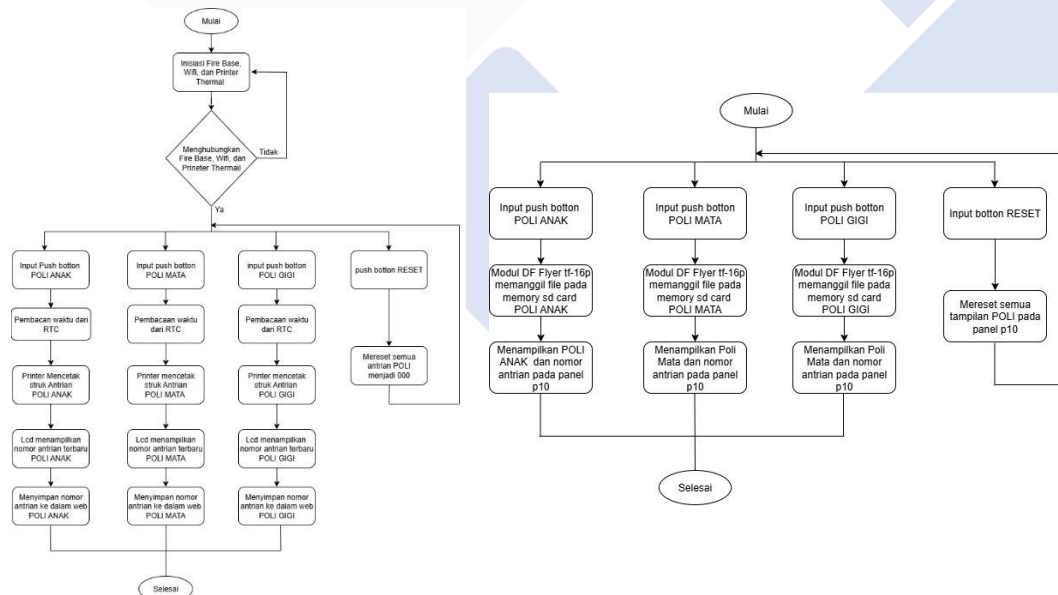
Observasi dilakukan secara langsung di lokasi Puskesmas untuk mengetahui alur pelayanan, sistem antrian yang berjalan, dan kendala yang dihadapi dalam mengelola antrian pasien. Hal ini membantu dalam merancang sistem yang sesuai dengan kebutuhan dan kondisi nyata di lapangan.

3.2.3. Pengujian Sistem

Pengujian bisa langsung dilakukan di Puskesmas setelah sistem dirancang. Data pengujian akan digunakan untuk menilai kepuasan pengguna, keandalan perangkat IoT, dan kinerja sistem.

3.3. Arsitektur Sistem

Ada banyak bagian yang saling terkait yang membentuk sistem ini. Mikrokontroler menerima data dari *pushbutton*, memprosesnya, dan menampilkan hasilnya di layar. Untuk memberi peringatan atau informasi lebih lanjut kepada pasien, sistem ini juga dapat dihubungkan ke aplikasi telepon pintar.



Gambar 3.1 Flowchart input dan output

3.4. Perancangan Alat

Perancangan alat pada sistem pemanggilan antrian otomatis ini dilakukan untuk mengintegrasikan perangkat keras (*hardware*) dan perangkat lunak

(software) berbasis *Internet of Things (IoT)*, sehingga mampu memanggil nomor antrean secara otomatis dan terintegrasi dengan sistem informasi puskesmas.

3.4.1. Blok Diagram Sistem

Perancangan dimulai dengan pembuatan blok diagram sistem untuk menggambarkan alur kerja antar komponen. Secara umum, sistem terdiri dari:

1. Input sebagai tombol pemanggil (*push button*) atau sistem kontrol dari *dashboard web*, yang berfungsi untuk memanggil antrean pasien secara otomatis. Jika terdapat beberapa poli yang berbeda seperti poli anak, gigi, dan mata maka sistem dapat memisahkan antrean berdasarkan kode unik masing-masing poli. Setiap tombol fisik atau kontrol pada dashboard web akan dikonfigurasi untuk mewakili satu poli tertentu, sehingga saat ditekan, sistem hanya memanggil antrean dari poli yang bersangkutan. Ini memastikan bahwa pasien dipanggil sesuai urutan dan lokasi layanan yang dituju, serta menghindari tumpang tindih antar-poli.
2. Proses Mikrokontroler (misalnya ESP32/NodeMCU) yang terhubung ke jaringan WiFi.
3. *Output Speaker* (untuk suara pemanggilan), *display LCD/LED* (untuk menampilkan nomor antrean), dan *web interface* sebagai panel kontrol.

3.4.2. Rangkaian Perangkat Keras (*Hardware*)

Komponen-komponen yang digunakan dalam perancangan alat meliputi:

1. Mikrokontroler (ESP32) adalah otak sistem yang mengendalikan proses pemanggilan dan koneksi *internet*.
2. *Speaker* Aktif digunakan Untuk mengeluarkan suara pemanggilan pasien secara otomatis (dapat menggunakan modul suara atau *Text-to-Speech*).
3. *LCD/LED Display* digunakan untuk menampilkan nomor antrean yang sedang dipanggil.
4. *Push Button* digunakan sebagai pemicu pemanggilan dari *database* yang tersimpan.

3.5. Integrasi IoT

Sistem ini akan terhubung ke *internet* sehingga dapat diakses secara *real-time*. Mikrokontroler akan berkomunikasi dengan *server* menggunakan protokol HTTP atau MQTT untuk sinkronisasi data antrean. Sistem juga mendukung pemanggilan otomatis berdasarkan antrean yang masuk dari aplikasi pendaftaran atau loket.

3.6. Pengujian Alat

Pengujian ini bertujuan untuk memastikan bahwa seluruh komponen sistem, baik perangkat keras maupun perangkat lunak, dapat berfungsi dengan baik sesuai dengan perancangan awal. Kegiatan pengujian dilakukan secara bertahap dan berulang selama proses pengembangan alat hingga alat benar-benar siap digunakan. Pengujian dimulai sejak tahap awal pembuatan, yaitu ketika komponen-komponen utama telah dirakit, hingga tahap akhir ketika sistem telah terintegrasi sepenuhnya dan dapat dijalankan secara otomatis. Metode pengujian yang digunakan mencakup pengujian fungsionalitas, kestabilan koneksi *internet*, responsivitas sistem terhadap input antrean, serta keakuratan output suara dan tampilan nomor antrean. Setiap proses diuji beberapa kali untuk memastikan tidak terjadi kesalahan, baik dalam membaca data dari *server*, serta menampilkan nomor antrean, maupun saat mengaktifkan suara pemanggilan.

BAB IV

PEMBAHASAN

Berdasarkan metode yang diuraikan pada bab sebelumnya, maka pada bab ini akan dijelaskan mengenai alat, proses pengerjaan, pengujian hingga hasil akhir pada proyek akhir bertajuk “Sistem Pemanggilan Antrean Otomatis pada Puskesmas berbasis *Internet of Things*”.

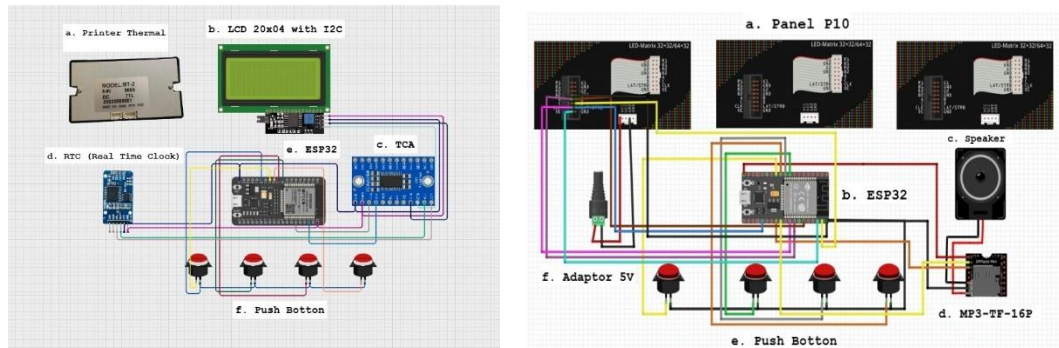
4.1. Deskripsi Alat Sistem Antrean Otomatis pada Puskesmas

Sistem antrean otomatis ini dikembangkan untuk memfasilitasi pasien dalam mengantre giliran konsultasi kesehatan, sekaligus bertujuan meningkatkan efisiensi dan kinerja perawat. Fungsi inti dari perangkat ini adalah menyediakan akses antrean yang lebih terstruktur dan terintegrasi, serta menyajikan informasi keluhan pasien secara aktual selama perawatan di fasilitas kesehatan.

Perangkat ini mengimplementasikan teknologi *Internet of Things* (IoT) untuk memungkinkan pemantauan melalui koneksi *internet*. Selain itu, alat ini mampu terhubung dengan *display* P10 menggunakan mikrokontroler ESP32 melalui komunikasi *Wi-Fi*. Integrasi kedua teknologi ini memungkinkan perangkat untuk memantau dan menyampaikan status keluhan pasien dengan data yang akurat.

Komponen utama yang digunakan meliputi ESP32, *printer thermal*, modul DFPlayer Mini, dan *display* P10, yang berfungsi sebagai sarana penerima keluhan pasien. Seluruh komponen sensor tersebut terhubung ke mikrokontroler ESP32 untuk mendeteksi masukan dari perangkat terhubung dan mengirimkan data ke modul DFPlayer Mini serta modul *display*.

4.2. Diagram Blok Sistem Antrean Otomatis pada Puskesmas



Gambar 4.1. Diagram Blok Sistem Antrean Otomatis pada Puskesmas

Berikut keterangan dari diagram blok di atas:

Blok Input

- a. *Thermal Printer*, sebagai *device* yang mencetak kertas antrean
- b. LCD 20x04 with I2C, sebagai visual nomor antrean terkini
- c. TCA, sebagai pengolah data untuk menganalisis dan memproses informasi
- d. RTC, sebagai untuk menyimpan dan menghitung waktu secara akurat
- e. ESP32, sebagai pusat mengumpulkan, memproses, dan mengirimkan semua data
- f. *Pushbutton*, sebagai pemicu

Blok Output

- a. *Display P10*, sebagai layar untuk menampilkan keluhan
- b. ESP32, sebagai pusat mengumpulkan, memproses, dan mengirimkan semua data
- c. *Speaker*, sebagai perangkat yang mengeluarkan data suara
- d. *Module df-player mini*, sebagai alat untuk mengeluarkan suara
- e. *Pushbutton*, sebagai pemicu
- f. *Adaptor 5V*, sebagai sumber untuk panel P10

4.3. Prinsip Kerja Sistem Antrean Otomatis pada Puskesmas

Sistem antrean otomatis ini bertujuan untuk mengoptimalkan efisiensi dan kenyamanan layanan di Puskesmas. Alur dimulai ketika pasien memperoleh nomor antrean melalui printer termal, dan mencetak nomor pada kertas termal. Nomor antrean tersebut kemudian divisualisasikan pada panel P10, mempermudah pasien dalam memantau urutan giliran mereka.

Selain representasi visual, sistem ini juga dilengkapi dengan modul DFPlayer Mini yang berfungsi memutar pemberitahuan suara nomor antrean, memastikan pasien yang berada di area lebih luas tetap dapat mendengar panggilan. Modul ini mengkonversi berkas audio digital menjadi sinyal analog yang dapat dialirkan melalui pengeras suara. Mikrokontroler ESP32 berfungsi sebagai pengendali utama seluruh sistem, menerima masukan dari tombol untuk menentukan nomor antrean dan mendistribusikan data ke printer termal, panel P10, serta modul DFPlayer Mini. Melalui integrasi perangkat keras dan perangkat lunak yang efektif, sistem ini menjamin kelancaran proses antrean, sehingga meningkatkan responsivitas dan efisiensi pelayanan kesehatan di Puskesmas.

4.4. Perancangan *Hardware* Sistem Antrean Otomatis pada Puskesmas

Prinsip Perancangan hardware untuk sistem antrean otomatis terdiri dari dua komponen utama yaitu, desain mekanik dan desain elektrik.

4.4.1. Perancangan Mekanik

Pada aspek perancangan mekanik sistem antrean otomatis, prioritas utama adalah konstruksi struktur fisik yang stabil dan efisien guna mengakomodasi seluruh komponen elektronik. Kotak perangkat dirancang dengan dimensi 18,5 cm x 14 cm x 6,5 cm dan ketebalan 3 mm, disesuaikan dengan ukuran serta bentuk PCB dan komponen lain, yang memfasilitasi perakitan dan pemeliharaan. Kotak ini diproduksi menggunakan teknologi cetak 3D dengan material plastik ABS, dipilih karena resistansinya terhadap suhu tinggi dan kelembaban. Desain kotak mengintegrasikan ventilasi yang memadai untuk mencegah akumulasi panas dan memastikan aliran udara optimal. Selain itu, kotak dilengkapi dengan slot akses untuk port USB, tombol poli, dan tombol *reset*, serta bukaan untuk pemasangan layar LCD. Melalui perancangan mekanik yang cermat, kotak perangkat ini tidak hanya berfungsi sebagai pelindung komponen elektronik, tetapi juga mendukung integrasi dan instalasi sistem antrean otomatis di lingkungan Puskesmas.

Secara keseluruhan, desain kotak perangkat telah dioptimalkan untuk menampung semua komponen yang relevan, menghasilkan konfigurasi yang teratur dan mendukung kinerja maksimal dari setiap elemen sistem. Maka, semua tempat

peletakan komponen telah diatur, seperti gambar di bawah ini:

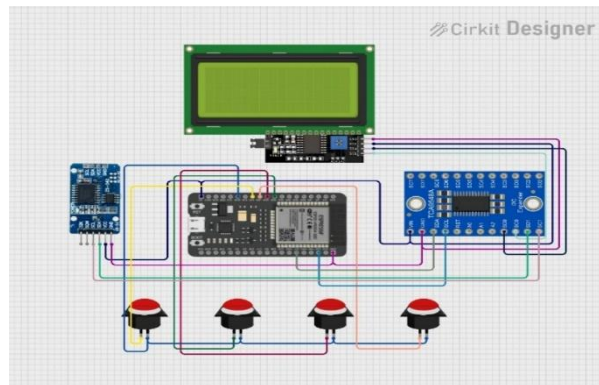


Gambar 4.2. Tampak atas Desain box

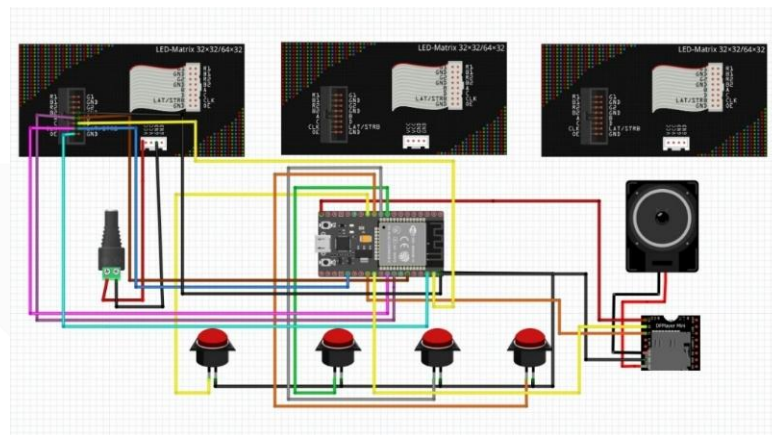
4.4.2. Perancangan Elektrik

Perancangan sistem antrean otomatis melibatkan penyusunan dan interkoneksi komponen elektronik untuk memastikan operasional pemantauan yang presisi. Tahap awal mencakup pembuatan diagram skematis rangkaian menggunakan perangkat lunak Cirkuit Designer guna memvisualisasikan serta merancang sirkuit dengan efisien.

Komponen-komponen vital, meliputi mikrokontroler Arduino Uno, modul DFPlayer Mini, panel P10, dan printer termal, diintegrasikan berdasarkan fungsinya masing-masing. Arduino Uno berperan sebagai unit kontrol pusat, memproses masukan pengguna dan mengarahkan sinyal ke modul lain. Modul DFPlayer Mini bertugas memutar notifikasi suara nomor antrean. Sementara itu, panel P10 berfungsi sebagai *display* visual untuk nomor antrean. Selanjutnya, printer termal mencetak nomor antrean pada kertas termal, menyediakan bukti fisik bagi pengguna. seperti yang terlihat di gambar 4.3 dan 44.



Gambar 4.3. Skematik Rangkaian Input



Gambar 4.4. Skematik Rangkaian Output

4.5. Pembuatan *Hardware* Sistem Antrean Otomatis pada Puskemas

Pembuatan *hardware* untuk sistem antrean otomatis terdiri dari dua komponen utama yaitu, desain mekanik dan desain elektrik.

4.5.1 Pembuatan *Hardware* Mekanik

Perakitan *hardware* mekanik merupakan tahap perakitan fisik dari semua komponen yang telah disiapkan. Semua komponen ditempatkan ke dalam *box* alat yang telah dirancang dan dicetak menggunakan *3D printing* sebelumnya. Setiap komponen memiliki tempat dan ruang yang sesuai. Selain itu, dipasangkan magnet di setiap sudut bagian *box* dan tutup *box* untuk memastikan *box* dapat ditutup dengan kuat tetapi juga mudah dibuka saat diperlukan.



Gambar 4.5. Tampak atas *Hardware Mekanik*



Gambar 4.6. Tampak bawah *Hardware Mekanik*



Gambar 4.7. Tampak samping *Hardware Mekanik*

4.5.2 Pembuatan *Hardware* Elektrik

Perakitan *hardware* elektrik merupakan tahap untuk memastikan semua komponen terhubung dengan benar dan siap beroperasi. Dalam proses ini, mikrokontroler ESP32, Module df-player mini dan *speaker* disambungkan satu sama lain. Setiap komponen tersebut dihubungkan menggunakan konektor yang berupa kabel *jumper*, seperti yang dapat dilihat pada gambar 4.8.



Gambar 4.8. Hasil perakitan semua komponen

4.6. Hasil Pengujian

Setelah alat proyek akhir ini selesai, terdapatlah hasil uji yang melibatkan proyek akhir yang berjudul “Sistem Pemanggilan Antrean Otomatis pada Poli Puskesmas Berbasis *Internet of Things*” ini. Berikut hasil dari uji yang kami lakukan:

4.6.1. Konfigurasi Web

Dalam tampilan *interface* Web kami hanya menampilkan *simple features*. Seperti pada gambar 4.9 :

<

Gambar 4.9. Tampilan *Interface Web*

Tabel 4.1. Hasil pengujian Web

No	Poli	Nomor	Status
1	anak	1	tersimpan
2	anak	2	tersimpan
3	anak	3	tersimpan
4	gigi	1	tersimpan
5	gigi	2	tersimpan
6	gigi	3	tersimpan
7	mata	1	tersimpan
8	mata	2	tersimpan
9	mata	3	tersimpan

Berdasarkan data pada tabel 4.1 dapat diketahui bahwa keberhasilan data dalam menyimpan sebagai bentuk *history* harian dari puskesmas tersebut. Dalam pembuatan webnya kita menggunakan *software VS Code* sebagai aplikasi *making interface web*. Dalam menghubungkan antara arduino dengan web kami menggunakan *software* ketiga yaitu *Firebase*.

4.6.2. Konektifitas Printer Thermal

Printer thermal dapat mencetak teks, *barcode*, dan gambar dengan kertas panas. Pada proyek ini printer dihubungkan ke arduino dengan trigger oleh *push button* untuk mencetak kertas dengan pilihan poli yang berbeda. Untuk tampilan kertas hasil cetakan dapat dilihat pada gambar 4.10 dibawah.



Gambar 4.10. Kertas Cetakan

Tabel 4.2. Keberhasilan Printer

No	Poli	Nomor	Output cetakan
1	mata	1	berhasil
2	mata	2	berhasil
3	gigi	1	berhasil
4	gigi	2	berhasil
5	anak	1	berhasil
6	anak	2	berhasil
7	mata	3	berhasil
8	mata	4	berhasil
9	gigi	3	berhasil
10	gigi	4	berhasil
11	anak	3	berhasil
12	anak	4	berhasil
13	mata	5	berhasil
14	mata	6	berhasil
15	gigi	5	berhasil

Dari tabel dapat dilihat bahwa printer berfungsi sebagaimana mestinya. Setiap poli dapat dicetak oleh Printer Thermal dengan baik tanpa ada kendala seperti *stuck* ataupun hal lainnya.

4.6.3. Konversi Output Suara

Module dfplayer mini akan membaca setiap format mp3 sesuai dengan waktu simpan, maksudnya setiap pemanggilan pembacaan format di arduino bukan berdasarkan penamaan file seperti 001.mp3 namun pemanggilan tergantung pada file mana yang terlebih dahulu tersimpan dimikroSD. Contoh jika file 002.mp3 tersimpan terlebih dahulu daripada 001.mp3 maka pada pembacaan 001 diarduino adalah file 002.mp3. Berikut adalah hasil pengambilan dari suara.

Tabel 4.3. Suara Antrean pada jarak 5 meter

No	Poli	Nomor	Output suara	Status
1	mata	1	"poli mata, antrian 1"	berhasil
2	mata	2	"poli mata, antrian 2"	berhasil
3	gigi	1	"poli gigi, antrian 1"	berhasil
4	gigi	2	"poli gigi, antrian 2"	berhasil
5	anak	1	"poli anak, antrian 1"	berhasil
6	anak	2	"poli anak, antrian 2"	berhasil

Pada percobaan pertama kami mencoba jarak 5 meter antara *speaker* dengan module dfplayer mini, dan mendapati bahwa tidak ada *error* pada setiap percobaan. Dengan suara yang jelas dan tegas namun memiliki *delay* 2 detik antara ketika menekan *pushbutton* dengan mulainya suara keluar.

Tabel 4.4. Suara Antrean pada jarak 10 meter

No	Poli	Nomor	Output suara	Status
1	mata	3	"poli mata, antrian 3"	berhasil
2	mata	4	"poli mata, antrian 4"	berhasil
3	gigi	3	"poli gigi, antrian 3"	berhasil
4	gigi	4	"poli gigi, antrian 4"	berhasil
5	anak	3	"poli anak, antrian 3"	berhasil
6	anak	4	"poli anak, antrian 4"	berhasil

Sama halnya dengan jarak 5 meter pada tabel 4.3, pada jarak 10 meter *output* suara juga berhasil keluar sebagaimana mestinya.

Tabel 4.5. Suara Antrean pada jarak 15 meter

No	Poli	Nomor	Output suara	Status
1	mata	5	"poli mata, antrian 5"	berhasil
2	mata	6	"poli mata, antrian 6"	berhasil
3	gigi	5	"poli gigi, antrian 5"	berhasil
4	gigi	6	"poli gigi, antrian 6"	berhasil
5	anak	5	"poli anak, antrian 5"	berhasil
6	anak	6	"poli anak, antrian 6"	berhasil

Pengujian dilakukan untuk mengevaluasi kinerja modul DFPlayer Mini dalam mengeluarkan output suara berbasis file audio .mp3 yang disimpan dalam microSD card. Dalam implementasi sistem antrean otomatis ini, DFPlayer Mini berfungsi sebagai media pemanggilan suara secara otomatis berdasarkan nomor antrean dan jenis poli yang dipilih. Namun perlu diperhatikan bahwa proses pemanggilan file tidak hanya bergantung pada penamaan file (misalnya 001.mp3, 002.mp3), melainkan juga pada urutan penyimpanan file di dalam microSD, karena DFPlayer Mini menggunakan indeks penyimpanan internal yang tidak selalu linier dengan penamaan file secara eksplisit. Oleh karena itu, disarankan menyimpan file audio secara berurutan dan tidak acak agar pemanggilan file melalui perintah kode program sesuai ekspektasi.

Pengujian dilakukan dengan tiga parameter jarak speaker terhadap sumber DFPlayer Mini, yaitu pada jarak 5 meter, 10 meter, dan 15 meter. Hasil uji coba menunjukkan bahwa pada jarak 5 meter, sistem mampu menghasilkan suara yang jernih, tegas, dan tanpa distorsi. Output audio untuk setiap poli (mata, gigi, anak) dan setiap nomor antrean berhasil dimainkan tanpa kendala. Delay sebesar ± 2 detik dari saat tombol ditekan hingga audio diputar teridentifikasi sebagai waktu proses normal dari mikrokontroler ke modul audio.

Selanjutnya, pada jarak 10 meter dan 15 meter, performa sistem tetap konsisten. Tidak terdapat penurunan kualitas suara maupun kesalahan pemutaran file. Semua file audio dipanggil sesuai urutan dan terdengar dengan jelas pada area pengujian. Keberhasilan sistem pada seluruh parameter ini menunjukkan bahwa modul DFPlayer Mini dan speaker yang digunakan mampu bekerja secara andal

hingga jarak 15 meter tanpa gangguan sinyal atau interferensi. Hal ini menunjukkan bahwa dalam skenario ruang tunggu Puskesmas yang luas, sistem tetap memberikan performa yang optimal.

Dengan demikian, dapat disimpulkan bahwa modul audio DFPlayer Mini merupakan solusi yang handal dan efisien untuk sistem pemanggilan antrean berbasis suara. Keberhasilannya pada pengujian ini mendukung implementasi sistem antrean otomatis yang tidak hanya informatif secara visual tetapi juga inklusif secara audio, terutama bagi pasien yang memiliki keterbatasan dalam mengakses informasi visual

4.6.4. Tampilan LCD

LCD 4x20 untuk memonitoring nomor antrean yang sudah diinput sebelumnya, dengan menggunakan pin SDA dan SCL sebagai komunikasi I2C (*Inter-Integrated Circuit*). LCD mampu menampilkan angka dan huruf dengan baik, juga dengan tambahan RTC (*Real-Time Clock*) mampu menampilkan waktu secara *real-time*. LCD dengan Printer juga saling terkoneksi melalui ESP32 sehingga data waktu saling terhubung. Berikut adalah contoh tampilan LCD.



Gambar 4.11. Tampilan LCD 4x20

Tabel 4.6. Keberhasilan *Output* LCD 4x20 I2C

No	Poli	Nomor	Counting	Jam
1	mata	1	berhasil	21.09.56
2	mata	2	berhasil	21.10.15
3	mata	3	berhasil	21.10.32
4	anak	1	berhasil	21.10.45
5	anak	2	berhasil	21.10.59
6	anak	3	berhasil	22.11.12
7	gigi	1	berhasil	21.11.31
8	gigi	2	berhasil	21.11.50
9	gigi	3	berhasil	21.12.05

LCD menerima sinyal data dari ESP32 setelah *push button* poli ditekan. Jadi jika *push button* poli mata ditekan maka LCD akan meng-*counting* antrean. Begitu juga dengan *push button* yang lain. Dan jika *push button reset* ditekan, maka LCD akan mereset semua count antrean.

4.6.5. Tampilan Panel P10

Panel P10 dikoneksikan dengan ESP32 yang juga terhubung dengan module dfplayer mini dan *push button* sebagai *trigger* setiap tindakan. Dengan adanya sinyal dari ESP32 akan memerintahkan panel P10 untuk meng-*counting* setiap tindakan antrean poli. Setelah alat proyek akhir ini selesai, terdapatlah hasil uji yang melibatkan proyek akhir yang berjudul “Sistem Pemanggil Antrean Otomatis ada Poli Puskesmas Berbasis *Internet of Things*” ini. Berikut adalah beberapa data yang kami ambil.



Gambar 4.12. Tampilan poli mata untuk nomor antrean1

Tabel 4.7. Data untuk poli mata antrean 1

Data	Skenario pengujian	Status	Keterangan
poli mata no antrian 1	koneksi arduino ke panel	berhasil	koneksi stabil
poli mata no antrian 1	keselarasan suara-visual	berhasil	selaras
poli mata no antrian 1	kecerahan panel	berhasil	tampilan jelas



Gambar 4.13. Tampilan poli mata untuk antrean 2

Tabel 4.8. Data untuk poli mata antrean 2

Data	Skenario pengujian	Status	Keterangan
poli mata no antrean 2	koneksi arduino ke panel	berhasil	koneksi stabil
poli mata no antrean 2	keselarasan suara-visual	berhasil	selaras
poli mata no antrean 2	kecerahan panel	berhasil	tampilan jelas



Gambar 4.14. Tampilan poli mata untuk antrean 3

Tabel 4.9. Data untuk poli mata antrean 3

Data	Skenario pengujian	Status	Keterangan
poli mata no antrean 3	koneksi arduino ke panel	berhasil	koneksi stabil
poli mata no antrean 3	keselarasan suara-visual	berhasil	selaras
poli mata no antrean 3	kecerahan panel	berhasil	tampilan jelas



Gambar 4.15. Tampilan poli gigi untuk nomor antrean1

Tabel 4.10. Data untuk poli gigi antrean 1

Data	Skenario pengujian	Status	Keterangan
poli gigi no antrean 1	koneksi arduino ke panel	berhasil	koneksi stabil
poli gigi no antrean 1	keselarasan suara-visual	berhasil	selaras
poli gigi no antrean 1	kecerahan panel	berhasil	tampilan jelas



Gambar 4.16. Tampilan poli gigi untuk antrean 2

Tabel 4.11. Data untuk poli gigi antrean 2

Data	Skenario pengujian	Status	Keterangan
poli gigi no antrean 2	koneksi arduino ke panel	berhasil	koneksi stabil
poli gigi no antrean 2	keselarasan suara-visual	berhasil	selaras
poli gigi no antrean 2	kecerahan panel	berhasil	tampilan jelas



Gambar 4.17. Tampilan poli gigi untuk antrean 3

Tabel 4.12. Data untuk poli gigi antrean 3

Data	Skenario pengujian	Status	Keterangan
poli gigi no antrean 3	koneksi arduino ke panel	berhasil	koneksi stabil
poli gigi no antrean 3	keselarasan suara-visual	berhasil	selaras
poli gigi no antrean 3	kecerahan panel	berhasil	tampilan jelas



Gambar 4.18. Tampilan poli mata untuk nomor antrean1

Tabel 4.13. Data untuk poli anak antrean 1

Data	Skenario pengujian	Status	Keterangan
poli anak no antrean 1	koneksi arduino ke panel	berhasil	koneksi stabil
poli anak no antrean 1	keselarasan suara-visual	berhasil	selaras
poli anak no antrean 1	kecerahan panel	berhasil	tampilan jelas



Gambar 4.19. Tampilan poli anak untuk antrean 2

Tabel 4.14. Data untuk poli anak antrean 2

Data	Skenario pengujian	Status	Keterangan
poli anak no antrean 2	koneksi arduino ke panel	berhasil	koneksi stabil
poli anak no antrean 2	keselarasan suara-visual	berhasil	selaras
poli anak no antrean 2	kecerahan panel	berhasil	tampilan jelas



Gambar 4.20. Tampilan poli anak untuk antrean 3

Tabel 4.15. Data untuk poli anak antrean 3

Data	Skenario pengujian	Status	Keterangan
poli anak no antrean 3	koneksi arduino ke panel	berhasil	koneksi stabil
poli anak no antrean 3	keselarasan suara-visual	berhasil	selaras
poli anak no antrean 3	kecerahan panel	berhasil	tampilan jelas

Analisa dari seluruh percobaan didapatkan bahwa panel P10 berfungsi dengan lancar, baik itu konektifitas atau tampilannya. Panel P10, dengan pixel pitch 10mm, menawarkan konektivitas yang sangat baik yang menjadi salah satu keunggulan utamanya, terutama untuk aplikasi layar besar. Desain modularnya memungkinkan panel-panel ini disambung dengan mudah menggunakan sistem penguncian cepat dan konektor daya serta data yang standar, seperti konektor GX16 atau bahkan IP65 untuk penggunaan luar ruangan. Konektivitas yang telah disesuaikan ini tidak hanya mempercepat proses instalasi dan pembongkaran, tetapi juga mempermudah pemeliharaan dan penggantian modul yang rusak. Fleksibilitas dalam koneksi serial atau paralel antar modul dan kabinet, yang diatur oleh kartu

penerima (*receiving card*) dan kartu pengirim (*sending card*), memastikan integritas sinyal dan distribusi daya yang efisien diseluruh permukaan layar. Sistem kontrol modern juga menyediakan kemampuan pemantauan koneksi secara *real-time*, sehingga masalah komunikasi dapat dideteksi dan diatasi dengan cepat.

Selain konektifitas yang andal, kecerahan panel P10 juga tergolong sangat baik, terutama untuk model yang dirancang untuk penggunaan *outdoor*. Mayoritas panel P10 *outdoor* memiliki tingkat kecerahan yang jauh di atas 5.000 nits, bahkan sering mencapai 6.500 nits atau lebih. Tingkat kecerahan yang tinggi ini krusial untuk memastikan visibilitas konten yang optimal bahkan di bawah terik matahari langsung, menjadikannya pilihan ideal untuk papan reklame digital di jalan raya, videotron stadion, atau layar informasi publik. Kecerahan yang baik ini tidak hanya sekedar angka, melainkan hasil dari penggunaan LED berkualitas tinggi, desain sirkuit yang efisien, dan manajemen termal yang efektif, yang semuanya berkontribusi pada *output* cahaya yang kuat dan konsisten sepanjang masa pakainya.

BAB V

KESIMPULAN

5.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem pemanggilan antrean otomatis pada poli Puskesmas berbasis *Internet of Things* (IoT) yang telah dilakukan, dapat ditarik beberapa kesimpulan utama sebagai jawaban atas perumusan masalah di Bab I:

1. Sistem mampu Memanggil Antrean Pasien secara Otomatis

Sistem ini dirancang dengan pendekatan modular dan memanfaatkan kemampuan mikrokontroler ESP32 yang mendukung komunikasi serial dan nirkabel. *Push button* berfungsi sebagai input digital untuk memicu siklus pemanggilan, di mana sinyal kemudian diproses oleh ESP32 untuk mengaktifkan DFPlayer Mini. Modul DFPlayer Mini kemudian memutar file audio dari microSD card sesuai urutan antrean yang telah ditentukan. *Output* audio disalurkan ke *speaker* aktif yang telah diuji hingga jarak 15 meter tanpa kehilangan kualitas suara yang signifikan, memungkinkan informasi panggilan didengar secara jelas di ruang tunggu. Selain itu, tampilan visual disinkronkan dengan sistem pemanggilan melalui panel LED P10 dan/atau LCD I2C, sehingga nomor antrean dan poli tujuan ditampilkan secara *real-time* dan konsisten. Integrasi semua komponen ini membuktikan bahwa sistem bekerja secara otomatis dan mandiri tanpa memerlukan intervensi manual dari operator, sehingga memenuhi prinsip smart system berbasis IoT.

2. Keefektifan Sistem terhadap Petugas dan Pasien

Adapun keefektifan sistem berdasarkan perspektif kedua pihak :

- a. Bagi Pasien: Sistem ini secara signifikan mengurangi waktu tunggu pasien dan meningkatkan kenyamanan karena informasi antrean yang jelas (visual dan audio). Pasien dapat menunggu di area yang lebih luas tanpa khawatir terlewat.
- b. Bagi Petugas: Sistem meningkatkan efisiensi petugas dengan mengotomatiskan proses pemanggilan antrean, meminimalkan kesalahan manual, dan menyediakan data antrean secara *real-time* melalui platform web. Hal ini

memungkinkan petugas untuk lebih fokus pada pelayanan inti dan manajemen Puskesmas dapat menganalisis data untuk optimasi layanan.

Secara keseluruhan, implementasi sistem pemanggilan antrean otomatis berbasis IoT ini terbukti meningkatkan kualitas pelayanan kesehatan di Puskesmas, menciptakan lingkungan yang lebih tertib, efisien, dan informatif bagi pasien maupun petugas.

5.2 Saran

Untuk pengembangan dan peningkatan sistem pemanggilan antrean otomatis di Puskesmas ini ke depannya, berikut beberapa saran yang bisa dipertimbangkan:

1. Pengembangan Fitur Lebih Lanjut pada Web

Perluasan fitur pada platform web dapat dilakukan, misalnya penambahan modul pelaporan statistik antrean harian/bulanan (jumlah pasien per poli, rata-rata waktu tunggu), fitur penjadwalan pasien, atau bahkan integrasi dengan sistem rekam medis elektronik Puskesmas.

2. Variasi Tampilan Informasi pada Panel P10

Selain nomor antrean, Panel P10 dapat dimanfaatkan untuk menampilkan informasi tambahan yang bermanfaat bagi pasien, seperti perkiraan waktu tunggu, nama poli yang sedang aktif, atau pengumuman kesehatan singkat.

3. Integrasi Notifikasi Pasien

Mengembangkan fitur notifikasi ke *smartphone* pasien (melalui aplikasi khusus atau SMS/WhatsApp *gateway*) dapat lebih meningkatkan kenyamanan, memungkinkan pasien memantau antrean dari luar area tunggu Puskesmas.

4. Optimasi Konsumsi Daya

Meskipun ESP32 dikenal hemat daya, untuk sistem yang beroperasi 24/7, optimasi lebih lanjut pada konsumsi daya atau penggunaan sumber energi alternatif (misalnya, tenaga surya mini) dapat dipertimbangkan, terutama untuk lokasi dengan pasokan listrik yang kurang stabil.

5. Peningkatan Robustness Sistem

Mengkaji dan meningkatkan ketahanan sistem terhadap gangguan jaringan atau *power failure*, misalnya dengan implementasi *backup power* (UPS) atau

mekanisme *offline mode* yang dapat menyimpan data sementara, akan sangat membantu menjaga keberlangsungan layanan.

6. Uji Coba Skala Lebih Besar

Melakukan uji coba di beberapa Puskesmas dengan karakteristik yang berbeda dapat memberikan data yang lebih komprehensif mengenai skalabilitas dan adaptabilitas sistem di berbagai kondisi.



Daftar Pustaka

- [1] R. Lestari and H. Aditya, "Evaluasi Sistem Antrian Manual di Pusat Kesehatan Masyarakat: Studi Kasus Puskesmas Tegalrejo," *Jurnal Manajemen Pelayanan Kesehatan*, vol. 8, no. 2, pp. 95–102, 2021.
- [2] S. Fatimah and A. Haryanto, "Dampak Waktu Tunggu Terhadap Kepuasan Pasien di Layanan Kesehatan Primer," *Jurnal Ilmu Kesehatan Masyarakat*, vol. 11, no. 3, pp. 211–218, 2022.
- [3] M. Syahputra and Y. Prasetya, "Implementasi Internet of Things (IoT) dalam Sistem Antrian Otomatis Berbasis Mikrokontroler," *J. Teknol. dan Sistem Komputer*, vol. 10, no. 1, pp. 56–63, 2022.
- [4] N. Rizki and H. Fauzan, "Rancang Bangun Sistem Antrian Otomatis Menggunakan ESP32, Display P10, dan Modul Audio," *J. Elektron. dan Sistem Informasi*, vol. 7, no. 2, pp. 120–127, 2023.
- [5] V. Wulandari and D. Supriadi, "Analisis Perilaku Masyarakat dalam Budaya Antrean Pelayanan Publik," *J. Ilmu Sos. dan Humaniora*, vol. 10, no. 2, pp. 101–109, 2021.
- [6] R. Setiawan and A. Halim, "Evaluasi Efektivitas Sistem Antrean Manual pada Layanan Umum," *J. Adm. Publik*, vol. 7, no. 1, pp. 55–63, 2020.
- [7] F. Pratama and I. Lestari, "Penerapan Sistem Antrean Digital Berbasis Mikrokontroler untuk Meningkatkan Layanan Kesehatan," *J. Teknol. dan Sistem Informasi*, vol. 4, no. 3, pp. 145–151, 2022.
- [8] Maulana RA, Widodo SP. Rancang Bangun Sistem Antrian Otomatis Berbasis IoT di Puskesmas. *Jurnal Teknologi Informasi dan Elektronika*.

2022;6(1):101–108.

- [9] Y. Ramadhan and A. Prasetyo, "Pemanfaatan Mikrokontroler ESP32 dalam Rancang Bangun Sistem Antrian Otomatis Berbasis IoT," *J. Elektron. dan Komput. Terapan*, vol. 8, no. 1, pp. 76–82, 2022.
- [10] A. Syahputra and F. Hadiyanto, "Rancang Bangun Mesin Antrian Otomatis Menggunakan Thermal Printer Mini dan Arduino," *J. Teknol. Komput. dan Aplikasinya*, vol. 8, no. 1, pp. 55–62, 2021.
- [11] R. Kurniawan and B. Setiawan, "Implementasi Printer Thermal Mini pada Sistem Pemanggilan Antrian Terintegrasi," *J. Sistem Informasi dan Komputerisasi*, vol. 6, no. 2, pp. 98–104, 2020.
- [12] R. Siregar and A. Hidayat, "Implementasi Push Button dalam Sistem Antrian Otomatis Berbasis Arduino," *J. Teknol. dan Sist. Komput.*, vol. 9, no. 2, pp. 150–157, 2021.
- [13] A. Hidayatullah and R. Maulana, "Rancang Bangun Sistem Pemanggilan Antrian Otomatis Berbasis Push Button dan Arduino," *J. Elektron. dan Sist. Inform.*, vol. 5, no. 2, pp. 89–94, 2021.
- [14] M. Alimin, M. Fauzi, and A. Nurfadillah, "Implementasi DFPlayer Mini dalam Sistem Pemutar Surah Al-Qur'an Otomatis," *J. Teknol. dan Riset*, vol. 12, no. 1, pp. 33–39, 2024.
- [15] Y. S. Putri and D. Hartanto, "Pemanfaatan microSD Card sebagai Penyimpan Audio dalam Sistem Antrian Otomatis," *J. Sist. Inform. dan Teknol. Inform. (JSITI)*, vol. 10, no. 1, pp. 55–61, 2022.
- [16] A. Fadillah and R. Suryadi, "Rancang Bangun Sistem Audio Pemanggil

Nomor Antrian Berbasis Arduino dan microSD," *J. Tek. Komput. UNIKOM*, vol. 7, no. 2, pp. 87–92, 2020.

- [17] R. Nurhadi and R. Anshori, "Penggunaan Kartu microSD pada Sistem Informasi Berbasis ESP32 untuk Penyimpanan Data Offline," *J. Teknol. dan Sist. Komput.*, vol. 9, no. 3, pp. 201–206, 2021.
- [18] N. Rachmawati and A. Suryawan, "Perancangan Display Informasi Antrian dengan LCD 20x4 Berbasis Mikrokontroler," *J. Teknol. dan Inform.*, vol. 5, no. 1, pp. 25–31, 2021.
- [19] S. Gunawan and R. Setiabudi, "Integrasi LCD I2C dan ESP32 dalam Sistem Informasi Digital," *J. Inform. dan Rekayasa Sist.*, vol. 6, no. 2, pp. 91–97, 2022.
- [20] E. Prakoso and D. Yuliani, "Rancang Bangun Sistem Pemanggilan Antrian Suara Menggunakan DFPlayer Mini dan Speaker Aktif," *J. Rancang Bangun Elektron.*, vol. 4, no. 2, pp. 66–72, 2021.
- [21] Z. Maulana and N. Hidayah, "Penerapan Teknologi Suara dalam Sistem Antrian Puskesmas Berbasis IoT," *J. Sist. Inform. Kesehatan*, vol. 9, no. 2, pp. 78–84, 2023.
- [22] A. Wahyudi and D. Salim, "Pemrograman ESP32 Menggunakan Platform VS Code dengan Ekstensi PlatformIO," *J. Teknol. Komput. dan Apl.*, vol. 9, no. 1, pp. 60–67, 2021.
- [23] T. Firmansyah and B. Aprianto, "Pengembangan Sistem Antrian Otomatis Berbasis ESP32 dengan VS Code dan Arduino IDE," *J. Riset Komput. dan Sist. Cerdas*, vol. 3, no. 2, pp. 99–106, 2022.

- [24] R. Nurdin and S. Hasanah, "Penggunaan Panel P10 dalam Sistem Antrian Digital pada Layanan Publik," *J. Teknol. Digit. dan Elektron.*, vol. 6, no. 1, pp. 74–80, 2021.
- [25] M. Satria and V. Lestari, "Perancangan Display LED P10 pada Sistem Informasi Antrian Terpadu," *J. Teknol. dan Inov. Sist. Inform.*, vol. 8, no. 3, pp. 112–118, 2020.



DAFTAR RIWAYAT HIDUP

1. Data diri

Nama : Septriandi
Tempat, Tanggal Lahir : Balunijuk, 11-09-2003
Alamat : Dusun 3 blok b, no. 78, Desa
Balunijuk
No. HP : 085758724911
Email : Septriandy030911@gmail.com
Jenis Kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

- | | |
|--|-----------------|
| 1. SD negeri 14 Balunijuk | 2009 - 2015 |
| 2. SMP Negeri 7 Pangkalpinang | 2015 - 2018 |
| 3. SMK Negeri 2 Pangkalpinang | 2018 - 2021 |
| 4. Politeknik Manufaktur Negeri
Bangka Belitung | 2022 - sekarang |

3. Pendidikan Non-Formal

1. Praktik Kerja Lapangan sebagai “Junior Enginer” di Swiss Bel-Hotel
2. Praktik Kerja Lapangan sebagai “Junior Enginer” di PT. HANABE

DAFTAR RIWAYAT HIDUP

1. Data diri

Nama : Abi Ari Sandy
Tempat, Tanggal Lahir : Mentok, 21-10-2004
Alamat : DUSUN VI PAIT JAYA
RT/RW : 002/000
No. HP : 082179557902
Email : abisandy914@gmail.com
Jenis Kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

- | | |
|--|-----------------|
| 1. SD negeri 16 Mentok | 2010 - 2016 |
| 2. SMP Negeri 3 Mentok | 2016 - 2019 |
| 3. SMA Negeri 1 Mentok | 2019 - 2022 |
| 4. Politeknik Manufaktur Negeri
Bangka Belitung | 2022 - sekarang |

3. Pendidikan Non-Formal

Praktik Kerja Lapangan sebagai “Junior Enginer” di PT. HANABE

KODE ARDUINO

1. BLOK INPUT

```
#include <WiFi.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "BluetoothSerial.h"
#include "RTCLib.h"
#include <Firebase_ESP_Client.h>
#include <time.h>

// ===== KONFIGURASI WIFI =====
const char* ssid = "Abi Yang Punya";
const char* password = "mimpiUSAHA";

// ===== FIREBASE =====
#define API_KEY "AIzaSyDNbZIP75VMgzEiUJOLij6_JlO1vJ16Ua0"
#define DATABASE_URL "https://projek-pengunjung-puskesmas-
default-rtdb.firebaseio.com/"

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
bool signupOK = false;

// ===== TCA9548A Multiplexer =====
#define TCAADDR 0x70
void tcaSelect(uint8_t channel) {
    if (channel > 7) return;
    Wire.beginTransmission(TCAADDR);
    Wire.write(1 << channel);
    Wire.endTransmission();
```

```
}
```

```
// ===== Inisialisasi =====
```

```
RTC_DS3231 rtc;
```

```
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

```
BluetoothSerial SerialBT;
```

```
uint8_t printerAddress[] = {0x66, 0x32, 0x4E, 0x9B, 0x4A, 0xE9};
```

```
// ===== Pin Tombol =====
```

```
const int buttonGigi = 14;
```

```
const int buttonAnak = 26;
```

```
const int buttonMata = 25;
```

```
const int buttonReset = 27;
```

```
// ===== Nomor Antrian =====
```

```
int antrianGigi = 0;
```

```
int antrianAnak = 0;
```

```
int antrianMata = 0;
```

```
unsigned long lastRTCUpdate = 0;
```

```
// ===== Fungsi Kalibrasi RTC dari NTP =====
```

```
void syncTimeFromNTP() {
```

```
    configTime(25200, 0, "pool.ntp.org"); // WIB (GMT+7)
```

```
    struct tm timeinfo;
```

```
    if (!getLocalTime(&timeinfo)) {
```

```
        Serial.println("✗ Gagal mendapatkan waktu NTP");
```

```
        return;
```

```
    }
```

```
    DateTime nowRTC(
```

```
        timeinfo.tm_year + 1900,
```

```

        timeinfo.tm_mon + 1,
        timeinfo.tm_mday,
        timeinfo.tm_hour,
        timeinfo.tm_min,
        timeinfo.tm_sec
    );
    rtc.adjust(nowRTC);
    Serial.println("☑ RTC disinkron dari NTP");
}

// ===== Fungsi Printer =====
void printCenter(String text) {
    SerialBT.write(0x1B); SerialBT.write(0x61); SerialBT.write(0x01);
    SerialBT.println(text);
}

void printBigText(String text) {
    SerialBT.write(0x1D); SerialBT.write(0x21); SerialBT.write(0x11);
    printCenter(text);
    SerialBT.write(0x1D); SerialBT.write(0x21); SerialBT.write(0x00);
}

// ===== Kirim Data ke Firebase dengan Retry =====
void kirimFirebase(String poli, int nomor, String waktu) {
    String path = "/antrian/" + poli;
    FirebaseJson json;
    json.set("poli", poli);
    json.set("nomor", nomor);
    json.set("waktu", waktu);

    if (Firebase.ready() && signupOK)

```

```

bool success = false;
int attempts = 0;
const int maxAttempts = 3;

while (!success && attempts < maxAttempts) {
    success = Firebase.RTDB.pushJSON(&fbdo, path.c_str(), &json);
    if (success) {
        Serial.println("☑ Data terkirim ke Firebase");
    } else {
        Serial.println("⚠️ Gagal kirim ke Firebase (percobaan " +
String(attempts + 1) + "): " + fbdo.errorReason());
        delay(1000);
    }
    attempts++;
}

if (!success) {
    Serial.println("❌ Gagal kirim ke Firebase setelah " +
String(maxAttempts) + " percobaan.");
}
} else {
    Serial.println("❌ Firebase tidak siap. Cek koneksi atau signup.");
}
}

// ===== Cetak Antrian dan Kirim =====
void cetakAntrian(String poli, int nomor) {
    char buffer[4];
    sprintf(buffer, "%03d", nomor);

```

```

tcaSelect(1);
DateTime now = rtc.now();
char waktu[25];
sprintf(waktu, "%02d:%02d:%02d %02d/%02d/%04d", now.hour(),
now.minute(), now.second(), now.day(), now.month(), now.year());

printCenter("");
printBigText("PUSKESMAS MENTOK");
printCenter("W5R9+5F8, Jl. Basuki Rahmat,");
printCenter("Sungai Daeng, Kec. Muntok");
printCenter("Kepulauan Bangka Belitung");
SerialBT.println("=====");
printBigText("POLI " + poli);
SerialBT.println("=====");
printBigText("NOMOR ANTRIAN");
printBigText(String(buffer));
SerialBT.println("=====");
printCenter("WAKTU CETAK:");
printCenter(String(waktu));
SerialBT.println("\n\n\n");

 kirimFirebase(poli, nomor, String(waktu));
tcaSelect(0);
}

// ===== LCD =====
void clearLine(int row) {
    lcd.setCursor(0, row);
    lcd.print(" ");
}

```

```

void updateLCDStatus() {
    char buffer[4];
    clearLine(1); sprintf(buffer, "%03d", antrianAnak); lcd.setCursor(2, 1);
    lcd.print("POLI ANAK : "); lcd.print(buffer);
    clearLine(2); sprintf(buffer, "%03d", antrianMata); lcd.setCursor(2, 2);
    lcd.print("POLI MATA : "); lcd.print(buffer);
    clearLine(3); sprintf(buffer, "%03d", antrianGigi); lcd.setCursor(2, 3);
    lcd.print("POLI GIGI : "); lcd.print(buffer);
}

```

```

// ===== SETUP =====

```

```

void setup() {
    Serial.begin(115200);
    Wire.begin();

    // LCD
    tcaSelect(0);
    lcd.init(); lcd.backlight(); lcd.clear();
    lcd.setCursor(0, 0); lcd.print("Menghubungkan WiFi");

    // WiFi
    WiFi.begin(ssid, password);
    int retries = 0;
    while (WiFi.status() != WL_CONNECTED && retries < 20) {
        delay(500);
        Serial.print(".");
        retries++;
    }

    if (WiFi.status() == WL_CONNECTED) {

```

```
Serial.println("\n☑ WiFi Connected: " + WiFi.SSID());  
lcd.setCursor(0, 1); lcd.print("WiFi: "); lcd.print(WiFi.SSID());  
} else {  
    Serial.println("\n✗ Gagal konek WiFi");  
    lcd.setCursor(0, 1); lcd.print("WiFi gagal. Reset.");  
    delay(3000);  
    ESP.restart();  
}
```

```
// Bluetooth  
SerialBT.begin("ESP32Master", true);  
SerialBT.setPin("0000", 4);  
SerialBT.connect(printerAddress);  
if (SerialBT.connected()) {  
    Serial.println("☑ Printer terhubung via Bluetooth");  
} else {  
    Serial.println("✗ Printer tidak terhubung!");  
}
```

```
// RTC  
tcaSelect(1);  
if (!rtc.begin()) {  
    Serial.println("✗ RTC Tidak Ditemukan");  
} else {  
    syncTimeFromNTP();  
}
```

```
// Tombol  
pinMode(buttonGigi, INPUT_PULLUP);  
pinMode(buttonAnak, INPUT_PULLUP);
```

```

pinMode(buttonMata, INPUT_PULLUP);
pinMode(buttonReset, INPUT_PULLUP);

// Firebase
config.api_key = API_KEY;
config.database_url = DATABASE_URL;
if (Firebase.signUp(&config, &auth, "", "")) {
    signupOK = true;
    Serial.println("☑ Firebase SignUp sukses");
} else {
    Serial.println("✗ Firebase SignUp gagal: " + fbdo.errorReason());
}

Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);

delay(1000);
tcaSelect(0);
updateLCDStatus();
}

// ===== LOOP =====
void loop() {
    if (millis() - lastRTCUpdate > 1000) {
        tcaSelect(1);
        DateTime now = rtc.now();
        char jam[9];
        sprintf(jam, "%02d:%02d:%02d", now.hour(), now.minute(),
now.second());

        tcaSelect(0);

```

```
clearLine(0);  
lcd.setCursor(3, 0); lcd.print("JAM: "); lcd.print(jam);  
lastRTCUpdate = millis();  
}
```

```
if (digitalRead(buttonGigi) == LOW) {  
    delay(200);  
    if (SerialBT.connected()) {  
        antrianGigi++;  
        cetakAntrian("GIGI", antrianGigi);  
        updateLCDStatus();  
    }  
    while (digitalRead(buttonGigi) == LOW);  
}
```

```
if (digitalRead(buttonAnak) == LOW) {  
    delay(200);  
    if (SerialBT.connected()) {  
        antrianAnak++;  
        cetakAntrian("ANAK", antrianAnak);  
        updateLCDStatus();  
    }  
    while (digitalRead(buttonAnak) == LOW);  
}
```

```
if (digitalRead(buttonMata) == LOW) {  
    delay(200);  
    if (SerialBT.connected()) {  
        antrianMata++;  
        cetakAntrian("MATA", antrianMata);  
        updateLCDStatus();  
    }  
}
```

```

    }
    while (digitalRead(buttonMata) == LOW);
}

if (digitalRead(buttonReset) == LOW) {
    delay(200);
    antrianGigi = 0; antrianAnak = 0; antrianMata = 0;
    tcaSelect(0);
    lcd.clear();
    lcd.setCursor(2, 1); lcd.print("== RESET SEMUA ==");

    if (SerialBT.connected()) {
        SerialBT.println("=====");
        printBigText("RESET SEMUA");
        printBigText("NOMOR ANTRIAN");
        SerialBT.println("=====\n\n\n");
    }

    delay(1500);
    updateLCDStatus();
    while (digitalRead(buttonReset) == LOW);
}
}

```

2. BLOK OUTPUT

```

#include <DMD32.h>

#include "fonts/SystemFont5x7.h"
#include <HardwareSerial.h>
#include <DFRobotDFPlayerMini.h>

// Panel P10 setup

```

```

#define DISPLAYS_ACROSS 3
#define DISPLAYS_DOWN 1
DMD dmd(DISPLAYS_ACROSS, DISPLAYS_DOWN);

// Tombol input
#define btnGigi 25
#define btnAnak 26
#define btnMata 27
#define btnReset 14

// Setup DFPlayer via UART2
HardwareSerial mySerial(2); // UART2: RX=16, TX=17
DFRobotDFPlayerMini myDFPlayer;

// Status tombol
bool lastStateGigi = false;
bool lastStateAnak = false;
bool lastStateMata = false;
bool lastStateReset = false;

// Nomor antrian dan suara per poli
int antrianGigi = 0;
int antrianMata = 0;
int antrianAnak = 0;

int gigiCounter = 1; // File 001.mp3 - 010.mp3
int mataCounter = 21; // File 011.mp3 - 020.mp3
int anakCounter = 41; // File 021.mp3 - 030.mp3

// Timer untuk panel P10
hw_timer_t* timer = NULL;

```

```

void IRAM_ATTR triggerScan() {
    dmd.scanDisplayBySPI();
}

void setup() {
    Serial.begin(115200);
    mySerial.begin(9600, SERIAL_8N1, 16, 17);

    pinMode(btnGigi, INPUT_PULLUP);
    pinMode(btnAnak, INPUT_PULLUP);
    pinMode(btnMata, INPUT_PULLUP);
    pinMode(btnReset, INPUT_PULLUP);

    Serial.println("Inisialisasi DFPlayer...");
    if (!myDFPlayer.begin(mySerial)) {
        Serial.println("Gagal konek ke DFPlayer!");
        while (true);
    }

    myDFPlayer.volume(25);
    Serial.println("DFPlayer siap.");

    // Setup timer P10
    uint8_t cpuClock = ESP.getCpuFreqMHz();
    timer = timerBegin(0, cpuClock, true);
    timerAttachInterrupt(timer, &triggerScan, true);
    timerAlarmWrite(timer, 100, true);
    timerAlarmEnable(timer);

    dmd.selectFont(SystemFont5x7);
    dmd.clearScreen(true);

```

```
    dmd.drawString(0,    0,    "PUSKESMAS    MENTOK",    18,  
    GRAPHICS_NORMAL);  
}
```

```
void loop() {  
    // Tombol GIGI  
    if (digitalRead(btnGigi) == LOW && !lastStateGigi) {  
        lastStateGigi = true;  
        antrianGigi++;  
        tampilkanAntrian("POLI GIGI", antrianGigi);  
        myDFPlayer.play(gigiCounter);  
        gigiCounter++;  
        if (gigiCounter > 20) gigiCounter = 1;  
        delay(300);  
    }  
    if (digitalRead(btnGigi) == HIGH) lastStateGigi = false;  
  
    // Tombol MATA  
    if (digitalRead(btnMata) == LOW && !lastStateMata) {  
        lastStateMata = true;  
        antrianMata++;  
        tampilkanAntrian("POLI MATA", antrianMata);  
        myDFPlayer.play(mataCounter);  
        mataCounter++;  
        if (mataCounter > 40) mataCounter = 21;  
        delay(300);  
    }  
    if (digitalRead(btnMata) == HIGH) lastStateMata = false;  
  
    // Tombol ANAK  
    if (digitalRead(btnAnak) == LOW && !lastStateAnak) {
```

```

lastStateAnak = true;
antrianAnak++;
tampilkanAntrian("POLI ANAK", antrianAnak);
myDFPlayer.play(anakCounter);
anakCounter++;
if (anakCounter > 60) anakCounter = 41;
delay(300);
}
if (digitalRead(btnAnak) == HIGH) lastStateAnak = false;

```

```

// Tombol RESET

```

```

if (digitalRead(btnReset) == LOW && !lastStateReset) {
    lastStateReset = true;

```

```

// Reset semua data

```

```

antrianGigi = 0;
antrianMata = 0;
antrianAnak = 0;
gigiCounter = 1;
mataCounter = 21;
anakCounter = 41;

```

```

Serial.println("🔧 RESET ditekan");

```

```

// Tampilkan countdown

```

```

for (int i = 3; i >= 1; i--) {
    dmd.clearScreen(true);
    char countdown[5];
    sprintf(countdown, "%d", i);
    int xPos = (DISPLAYS_ACROSS * 32 - strlen(countdown) * 6) / 2;

```

```

        dmd.drawString(xPos, 8, countdown, strlen(countdown),
    GRAPHICS_NORMAL);
        delay(800);
    }

    // Tampilkan "RESET BERHASIL"
    dmd.clearScreen(true);
    const char* resetDone = "RESET BERHASIL";
    int xCenter = (DISPLAYS_ACROSS * 32 - strlen(resetDone) * 6) / 2;
    dmd.drawString(xCenter, 8, resetDone, strlen(resetDone),
    GRAPHICS_NORMAL);
    delay(3000);

    // Kembali ke tampilan awal
    dmd.clearScreen(true);
    dmd.drawString(0, 0, "PUSKESMAS MENTOK", 18,
    GRAPHICS_NORMAL);
    }
    if (digitalRead(btnReset) == HIGH) lastStateReset = false;
}

void tampilkanAntrian(const char* label, int nomor) {
    char buffer[40];
    sprintf(buffer, "%s : %03d", label, nomor);
    dmd.clearScreen(true);
    dmd.drawString(0, 0, "PUSKESMAS MENTOK", 18,
    GRAPHICS_NORMAL);
    int xPos = (DISPLAYS_ACROSS * 32 - strlen(buffer) * 6) / 2;
    dmd.drawString(xPos, 8, buffer, strlen(buffer), GRAPHICS_NORMAL);
    Serial.println(buffer);
    delay(2000);}

```

KODE VSCODE

```
<!DOCTYPE html>

<html>

<head>

  <meta charset="UTF-8">

  <title>Data Antrian Puskesmas</title>

</style>

body {
  font-family: 'Segoe UI', sans-serif;
  background-color: #1e1e2f;
  color: #f0f0f0;
  padding: 20px;
}

h1 {
  color: #03dac6;
  text-align: center;
}

.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 20px;
  justify-items: center;
}

.poli-card {
  background: #2a2a3d;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0,0,0,0.4);
  width: 100%;
  max-width: 400px;
}
```

```
}  
h2 {  
    color: #03dac6;  
    margin-top: 0;  
    text-align: center;  
}  
label, select, button {  
    margin: 10px 5px 10px 0;  
}  
select, button {  
    padding: 5px 10px;  
    background-color: #03dac6;  
    border: none;  
    border-radius: 5px;  
    color: #000;  
    font-weight: bold;  
    cursor: pointer;  
}  
table {  
    width: 100%;  
    border-collapse: collapse;  
    margin-top: 10px;  
}  
th, td {  
    padding: 10px;  
    border: 1px solid #444;  
    text-align: left;  
}  
th {  
    background-color: #03dac6;  
    color: #000;
```

```

    }
  </style>
</head>
<body>
  <h1>Data Antrian Puskesmas</h1>
  <div class="container" id="poliContainer"></div>

  <script src="https://www.gstatic.com/firebasejs/9.23.0/firebase-app-
compat.js"></script>
  <script src="https://www.gstatic.com/firebasejs/9.23.0/firebase-database-
compat.js"></script>

  <script>
    const firebaseConfig = {
      apiKey: "AIzaSyD8pWmgousSiK3jtZ6BdDvVlgoQ-0CtHjs",
      authDomain: "penyimpanan-data-pengunjung.firebaseio.com",
      databaseURL: "https://penyimpanan-data-pengunjung-default-
rtbd.firebaseio.com",
      projectId: "penyimpanan-data-pengunjung",
      storageBucket: "penyimpanan-data-pengunjung.firebaseio.com",
      messagingSenderId: "778432181085",
      appId: "1:778432181085:web:e204d74db4520c89012eb4"
    };

    firebase.initializeApp(firebaseConfig);
    const db = firebase.database();
    const poliList = ['anak', 'gigi', 'mata'];

    function createPoliCard(poliName) {
      const container = document.getElementById('poliContainer');
      const card = document.createElement('div');

```

```

card.className = 'poli-card';
card.innerHTML = `
    <h2>Poli ${poliName.toUpperCase()}</h2>
    <label for="tgl-${poliName}">Tanggal:</label>
    <select id="tgl-${poliName}"><option value="">-- Semua Tanggal --
</option></select>
    <button      onclick="downloadJSON('${poliName}')">Download
JSON</button>
    <table>
    <thead>
        <tr><th>Nomor</th><th>Waktu</th></tr>
    </thead>
    <tbody id="tabel-${poliName}"></tbody>
    </table>
`;
container.appendChild(card);
}

const dataPerPoli = {};
const tanggalPerPoli = {};

function formatNomor(nomor) {
    const num = parseInt(nomor, 10);
    return num.toString().padStart(3, '0');
}

function ambilData(poli) {
    db.ref('antrian/' + poli).on('value', snapshot => {
        const data = snapshot.val();
        const tbody = document.getElementById(`tabel-${poli}`);
        const select = document.getElementById(`tgl-${poli}`);

```

```

tbody.innerHTML = "";
tanggalPerPoli[poli] = new Set();
dataPerPoli[poli] = [];

if (data) {
  for (const id in data) {
    const item = data[id];
    const tanggal = item.waktu?.split(" ")[1];
    if (tanggal) tanggalPerPoli[poli].add(tanggal);
    dataPerPoli[poli].push(item);
  }

  select.innerHTML = '<option value="">-- Semua Tanggal --</option>';
  Array.from(tanggalPerPoli[poli]).sort().forEach(tgl => {
    const opt = document.createElement("option");
    opt.value = tgl;
    opt.textContent = tgl;
    select.appendChild(opt);
  });

  tampilkanData(poli);
  select.addEventListener("change", () => tampilkanData(poli));
}

});
}

function tampilkanData(poli) {
  const tbody = document.getElementById(`tabel-${poli}`);
  const selectedTgl = document.getElementById(`tgl-${poli}`).value;
  tbody.innerHTML = "";

```

```

dataPerPoli[poli].forEach(item => {
  if (!selectedTgl || (item.waktu && item.waktu.includes(selectedTgl)))
  {
    const row = document.createElement("tr");
    const td1 = document.createElement("td");
    const td2 = document.createElement("td");
    td1.textContent = formatNomor(item.nomor);
    td2.textContent = item.waktu;
    row.appendChild(td1);
    row.appendChild(td2);
    tbody.appendChild(row);
  }
});
}

function downloadJSON(poli) {
  const selectedTgl = document.getElementById(`tgl-${poli}`).value;
  const filtered = dataPerPoli[poli].filter(item => !selectedTgl ||
(item.waktu && item.waktu.includes(selectedTgl)));
  const blob = new Blob([JSON.stringify(filtered, null, 2)], { type:
"application/json" });
  const url = URL.createObjectURL(blob);
  const link = document.createElement("a");
  link.setAttribute("href", url);
  link.setAttribute("download", `antrian-${poli}-${selectedTgl} ||
'semua'.json`);
  document.body.appendChild(link);
  link.click();
  document.body.removeChild(link);
}

```

```
    poliList.forEach(poli => {  
        createPoliCard(poli);  
        ambilData(poli);  
    });  
</script>  
</body>  
</html>
```

