PERANCANGAN SISTEM MONITORING WEBSITE DAN IMPLEMENTASI OPERASIONAL SIARAN STASIUN RADIO KOMUNITAS D3 TEKNOLOGI TELEKOMUNIKASI

1st Arrijal Habib
D3 Teknologi Telekomunikasi Telkom University
Bandung, Indonesia
arrijalhabib@student.telkomuniversity.ac.id

2nd Denny Darlis, S.Si., M. T. D3 Teknologi Telekomunikasi Telkom University
Bandung, Indonesia
dennydarlis@telkomuniversity.ac.id

3rd Dr. Sugondo Hadiyoso, A. Md, S.T M.T. D3 Teknologi Telekomunikasi Telkom University
Bandung, Indonesia
sugondo@telkomuniversity.ac.id

***Abstrak* — Perkembangan teknologi penyiaran memberikan peluang bagi radio komunitas untuk memanfaatkan sistem monitoring siaran yang praktis dan hemat biaya, khususnya pada stasiun yang menyiarkan melalui radio internet dan FM RDS secara bersamaan. Penelitian ini mengembangkan sistem monitoring berbasis web yang mampu menampilkan informasi siaran secara real-time dari aplikasi Mixxx, dengan tujuan memudahkan pengelola stasiun melakukan pemantauan tanpa perlu mengakses aplikasi penyiar secara langsung. Sistem ini dirancang menggunakan Raspberry Pi sebagai server, memanfaatkan framework Flask untuk pembuatan website, serta HTML, CSS, dan JavaScript untuk antarmuka yang sederhana dan mudah dipahami. Data yang disajikan mencakup status aplikasi Mixxx, judul lagu, nama penyanyi, timeline siaran, dan status koneksi siaran, yang diambil dari metadata berformat JSON. Hasil pengujian menunjukkan bahwa sistem mampu menampilkan informasi secara akurat dan real-time, serta mendukung fitur penambahan lagu langsung dari website, layanan request lagu dari pendengar, dan tampilan tiga lagu teratas yang paling sering diputar. Implementasi fitur ini berjalan lancar dengan pembaruan data yang konsisten dan waktu respon singkat, sehingga pengalaman monitoring menjadi lebih interaktif, efisien, dan mendukung transformasi menuju penyiaran digital yang lebih modern.**

***Kata kunci*— Raspberry Pi, Mixxx, Monitoring Siaran, Website, Flask, Radio Internet, FM RDS, Radio Komunitas**

I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah membawa perubahan yang signifikan di dunia, termasuk komunitas radio yang kini perlu beradaptasi agar tetap relevan. Salah satu isu utama yang dihadapi komunitas radio adalah keterbatasan jangkauan dan frekuensi pemancar. Hal ini mendorong munculnya ide untuk memungkinkan pengguna radio mendengarkan siaran favorit mereka di mana saja dan kapan saja. Oleh karena itu, dilakukan penelitan perancangan sistem monitoring website berbasis raspberry pi dengan biaya yang rendah dan fleksibiltas tinggi, sehingga bisa disiarkan dimanapun dan kapanpun. Raspberry Pi bertindak sebagai pemancar yang mengirimkan data yang sudah dihubungkan ke internet ataupun melalui FM RDS, kemudian data yang dikirimkan akan diencode menggunakan beberapa Software untuk menampilkan output yang diinginkan.[1]

Pada penelitian kali ini Raspberry Pi berperan sebagai perangkat utama yang menjalankan fungsi pemancar siaran, baik melalui koneksi internet maupun secara lokal dengan dukungan teknologi FM RDS. Selain itu raspberry pi berfungsi sebagai server tempat merancang website monitoring yang akan dibuat. Dalam penelitian kali ini, penulis merancang sebuah website monitoring yang dirancang khusus untuk memudahkan operator dalam memantau dan mengatur jalannya siaran. Website tersebut menampilkan berbagai informasi penting terkait siaran, seperti status aplikasi Mixxx, nama penyiar, judul lagu, nama penyanyi, tema siaran, timeline siaran, serta jadwal siaran. Dengan adanya sistem monitoring ini, pengelolaan siaran menjadi lebih terstruktur dan dapat dilakukan secara jarak jauh, sehingga mendukung efisiensi dan stabilitas dalam proses penyiaran.

Penggunaan raspberry Pi yang mudah dan fleksibel dapat menjadi solusi untuk pengiriman audio ke perangkat yang diterima. Dengan memanfaatkan kemampuan Raspberry Pi, kita dapat memproses sinyal radio Internet dan menyiarkannya secara langsung, memberikan akses mudah kepada para pendengar serta bisa melakukan siaran dimanapun dan kapanpun tanpa adanya batasan.[2] Sistem ini tidak hanya memungkinkan siaran dilakukan secara fleksibel melalui jaringan internet maupun pemancar FM RDS, tetapi juga memberikan kemudahan bagi operator untuk memantau dan mengelola siaran secara real-time melalui website monitoring yang terintegrasi. Dengan biaya yang relatif rendah dan kemudahan implementasi, hasil penelitian ini diharapkan dapat mendorong digitalisasi siaran di komunitas D3 Teknologi Telekomunikasi di masa mendatang.

II. KAJIAN TEORI

A. Raspberry Pi 4

Raspberry Pi 4 adalah komputer mini berukuran kecil namun bertenaga setara desktop. Perangkat ini memiliki RAM, modul radio, port USB, PMIC untuk distribusi daya, port Ethernet, chip SoC sebagai otak pemrosesan, serta GPIO untuk menghubungkan komponen elektronik lain, sehingga cocok untuk eksperimen dan pengembangan IoT. Dengan sistem operasi berbasis Linux yang fleksibel dan Python sebagai bahasa pemrograman utama, Raspberry Pi 4 mampu mendukung berbagai aplikasi, seperti server lokal, automasi, sistem monitoring, hingga penyiaran radio berbasis internet [5].

B. Sistem Operasi dan Antarmuka Grafis pada Raspberry Pi

Raspberry Pi umumnya menggunakan sistem operasi berbasis Linux yang dikenal sebagai Raspberry Pi OS atau sebelumnya disebut Raspbian. Sisstem ini merupakan hasil adaptasi dari distribusi Debian yang dioptimalkan khususuntuk arsitektur prosesor ARM, sehingga mampu memberikan performa yang lebihbaik terutama dalam pemrosesan data numerik dan floating point.

Raspberry Pi OS dilengkapi dengan lebih dari 35.000 paket perangkat lunak bawaan, sehingga memudahkan pengguna dalam instalasi berbagai aplikasi tambahan. Selain Raspberry Pi OS, tersedia juga pilihan sistem operasi alternatif seperti Android, Arch Linux ARM, Fedora, Chromium OS, dan RISC OS. Namun, Raspberry Pi OS tetap menjadi pilihan utama karena dukungan komunitas yang kuat dan kompatibilitasnya yang luas . Raspberry Pi OS menyediakan tampilan grafis berbasis desktop environment yang ringan seperti LXDE atau PIXEL. Antarmuka ini memungkinkan pengguna untuk mengoperasikan perangkat secara visual tanpa harus mengandalkan perintah baris (command line), sehingga lebih mudah digunakan oleh operator dari kalangan non teknis.[6] Dalam Tugas Akhir ini, antarmuka grafis dimanfaatkan untuk server Icecast, mengelola layanan FM RDS, dan merancang website monitoring siaran berbasis Flask. Kehadiran antarmuka desktop ini menjadikan Raspberry Pi sebagai sistem yang ramah pengguna dan fleksibel untuk kebutuhan penyiaran komunitas secara menyeluruh.

C. Mixxx

Mixxx merupakan perangkat lunak open-source yang dirancang untuk kebutuhan pertunjukan musik secara langsung, khususnya untuk DJ. Aplikasi ini menawarkan berbagai fitur seperti sinkronisasi tempo, perpindahan lagu otomatis (crossfading), dan tampilan visual audio. Dalam penelitian oleh Andersen yang dipresentasikan pada NIME 2003, Mixxx dikembangkan sebagai alternatif digital dari perangkat DJ konvensional, dengan fokus pada kemudahan penggunaan dan interaksi langsung antara pengguna dengan sistem audio digital[8].

D. Icecast

Icecast adalah server gratis untuk siaran langsung melalui internet. Icecast digunakan untuk membuat radio online atau radio internet. Icecast memiliki fitur- fitur yang sangat mudah ditambahkan dan mendukung standar terbuka untuk interaksi, sehingga memudahkan pendengar untuk mengaksesnya. Icecast mendukung streaming audio simultan. Yang artinya setiap aliran memiliki "mount point" yang dapat diakses pengguna melalui URL jaringan.[10]

E. Flask

Flask merupakan salah satu framework web berbasis bahasa pemrograman Python yang termasuk dalam kategori microframework. Dalam proyek sistem monitoring siaran radio komunitas ini, Flask dimanfaatkan sebagai kerangka utama dalam membangun aplikasi web yang menampilkan status siaran dari Mixxx melalui server Icecast. Dengan menggunakan Flask, pengembangan website menjadi lebih terstruktur dan efisien, karena kerangka kerja ini memungkinkan pengaturan alur logika aplikasi dan tampilan antarmuka secara terpisah namun terintegrasi.

Sebagai microframework, Flask tidak menyertakan berbagai pustaka tambahan secara default, seperti pengelolaan basis data atau validasi form. Namun, Flask 9 tetap fleksibel karena mendukung integrasi dengan berbagai ekstensi pihak ketiga yang dapat memperluas fungsionalitas sesuai kebutuhan pengembang. Hal ini menjadi keuntungan tersendiri dalam proyek ini, karena hanya fitur yang dibutuhkan saja yang diaktifkan, sehingga aplikasi tetap ringan dan cocok dijalankan di perangkat seperti Raspberry Pi yang memiliki sumber daya terbatas. Meskipun disebut sebagai microframework, Flask tetap mendukung pengembangan aplikasi skala menengah hingga besar. Pendekatan modularnya membuat sistem website monitoring yang dikembangkan menjadi lebih mudah untuk dikustomisasi dan dikembangkan lebih lanjut, baik dalam penambahan fitur status siaran, nama penyiar, maupun integrasi tampilan grafis menggunakan HTML dan CSS.[12]

F. Website Monitoring

Website merupakan media berbasis halaman-halaman web yang saling terhubung, dikembangkan oleh individu, kelompok, maupun institusi untuk menyajikan informasi kepada pengguna secara interaktif dan menarik.[13] Dalam konteks sistem siaran radio komunitas ini, website dirancang sebagai sarana monitoring siaran yang menampilkan data penting secara real-time, seperti status aplikasi Mixxx, nama penyiar, tema siaran, serta metadata lagu yang sedang diputar.

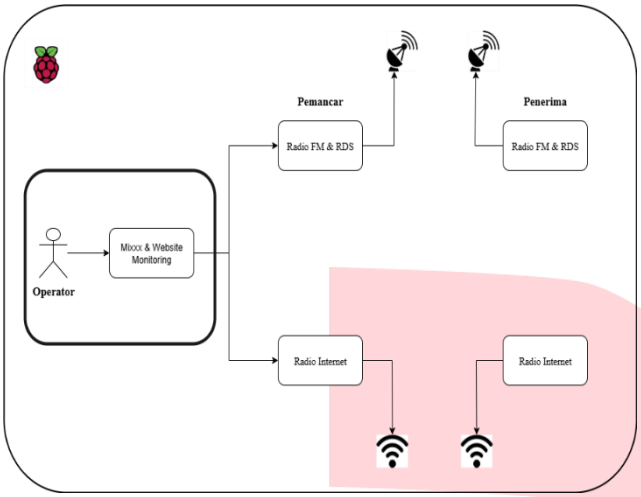
Pembuatan website ini menggunakan framework Flask yang berbasis Python sebagai pengelola sisi backend. Flask dipilih karena ringan dan fleksibel, sehingga cocok untuk dijalankan pada perangkat seperti Raspberry Pi. Informasi siaran diperoleh dari server Icecast dalam format JSON, kemudian diproses dan disajikan melalui halaman web. Di sisi antarmuka pengguna (frontend), digunakan HTML Mengatur serta mendesain tampilan isi halaman website, [14]serta CSS untuk mempercantik tampilan visual. Dengan pendekatan ini, website monitoring mampu memberikan kemudahan akses informasi bagi operator siaran dan penyiar melalui browser, baik secara lokal maupun jaringan luas.

II. METODE

Bab ini membahas metode yang digunakan dalam perancangan dan implementasi sistem manajemen siaran radio komunitas berbasis web. Sistem ini dirancang untuk mengintegrasikan Mixxx sebagai apliaksi untuk menginput konten siaran, Icecast sebagai server siaran, serta website monitoring. Proses komunikasi dilakukan melalui jaringan internet, di mana audio yang diputar di Mixxx dikirim ke server Icecast dan kemudian dapat diakses pendengar melalui URL streaming. Selain itu, metadata siaran seperti judul lagu, status Mixxx, jadwal siaran, timeline siaran,tiga lagu teratas, tambah dan hapus lagu serta status koneksi ditampilkan secara real-time pada website monitoring untuk memudahkan operator dalam mengelola dan memantau siaran.

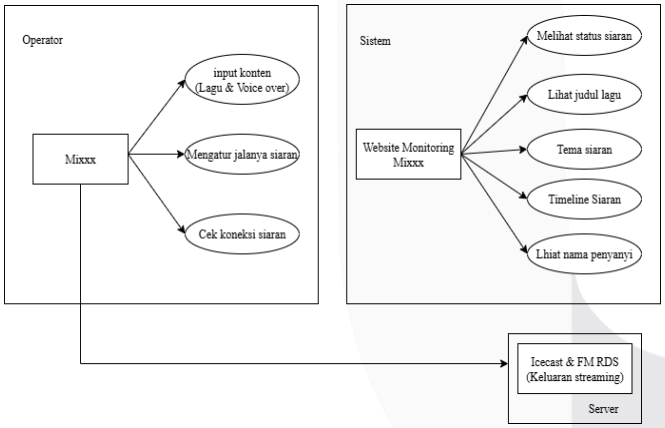
A. Blok Diagram Sistem

Pada Gambar 3.1 mengilustrasikan sistem manajemen siaran radio komunitas berbasis Raspberry Pi yang terdiri dari: Mixxx dan website monitoring sebagai media utama dalam pembuatan konten, pemancar, dan penerima radio internet dan FM RDS. Setiap komponen memiliki perannya masing-masing dalam mendukung pengelolaan dan distribusi siaran radio baik melalui internet maupun FM RDS.



GAMBAR 1
Blok Diagram Sistem

Bagian yang telah ditandai merupakan fokus utama pada proyek saya yaitu perancangan Sistem monitoring website radio komuniats D3 Teknologi Telekomunikasi. Penjelasan lebih detail terkait bagian ini akan disampaikan pada *Use Case Diagram* yang menunjukkan fitur dan alur kerja aplikasi website secara spesifik.

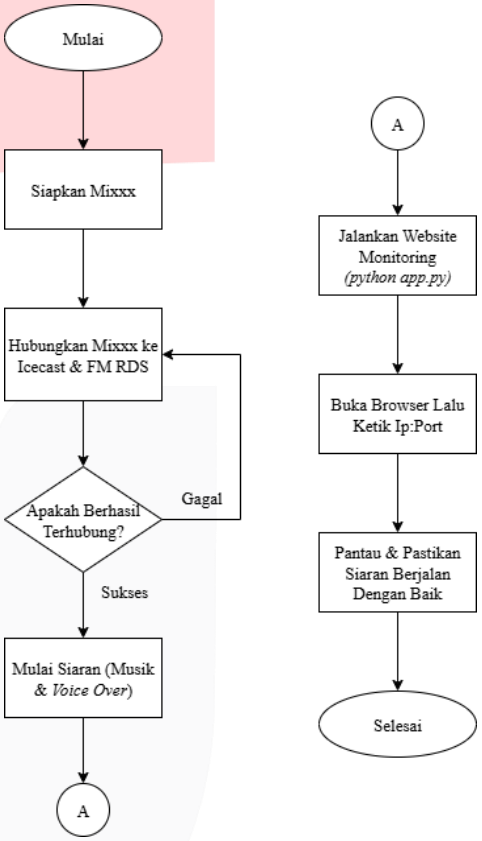


GAMBAR 2
Use Case Diagram

Use Use case diagram menggambarkan sistem siaran radio komunitas dengan Mixxx sebagai pusat kendali siaran yang berperan mengelola input konten (lagu dan voice over), mengatur jalannya siaran, serta memastikan koneksi ke Icecast dan FM RDS berjalan baik. Pada tugas akhir ini, Mixxx digunakan untuk dua skenario, yaitu input suara langsung dan playlist musik otomatis. Website monitoring yang dibuat operator berfungsi menampilkan metadata siaran secara real-time, diambil dari Icecast yang tersambung ke Mixxx melalui Flask Python, kemudian ditampilkan pada website berupa status siaran, judul lagu, nama penyanyi, tema siaran, dan jadwal atau timeline. Dengan adanya website monitoring, operator lebih mudah memantau siaran tanpa harus mengakses langsung sistem utama, sehingga sistem ini tidak hanya mendukung kelancaran siaran, tetapi juga meningkatkan efisiensi pemantauan, dokumentasi, dan pengelolaan siaran radio komunitas secara keseluruhan.

B. Tahapan Perancangan

Bagian ini menjelaskan terkait tahapan dalam merancang dan mengimplentasikan sistem pemancar radio internet. Sistem ini dibuat untuk 13 mendukung operasional stasiun radio komunitas yang akan diimplemetasikan sebagai station radio untuk kegiatan di D3 Teknologi Telekomunikasi. Adapun tahapann perancangannya dapat dilihat pada flowchart berikut :



GAMBAR 3
Use Case Diagram

Tahap pertama dalam perancangan sistem monitoring siaran radio komunitas ini adalah memastikan bahwa aplikasi Mixxx sudah terinstall dan siap untuk digunakan. Aplikasi Mixxx merupakan sebuah aplikasi siaran yang digunakan sebagai pemutar audio sekaligus pengelola siaran radio digital. Setelah dipastikan 14 aplikasi mixxx sudah siap digunakan, langkah selanjutnya adalah menghubungkan mixxx dengan Icecast. Proses koneksi ini dilakukan melalui menu Preferences > Live Broadcasting pada aplikasi Mixxx. Pada bagian ini, operator perlu mengisi parameter koneksi seperti Type, Mountpoint, Host, Port, Username, dan Password, yang semuanya harus disesuaikan dengan konfigurasi server Icecast sebelumnya pada raspberry pi. Aapabila mixxx sudah terkoneksi dengan icecast, langkah selanjutnya adalah mencentang live broadcasting pada mixxx, tunggu beberapa saat hingga muncul notifikasi connected. Apabila sudah muncul notifikasi, maka icecast dan mixxx sudah tersambung dan kita sudah dapat melakaukan Siaran. Untuk siaran berbasis FM, digunakan pemancar FM RDS, Raspberry Pi juga berfungsi sebagai pengelola file audio yang akan dipancarkan. Lagu yang diputar telah disesuaikan dengan

urutan lagu yang ada pada mixxx

Apabila konfigurasi dasar telah dilakukan, langkah berikutnya adalah menjalankan aplikasi Mixxx sebagai pusat pengendali siaran. Pada tahap ini, terdapat dua skenario siaran yang dapat dijalankan oleh operator, yang pertama playlist musik otomatis dan yang kedua yaitu siaran dengan menggunakan voice over, yaitu input suara secara langung melalui mikrofon yang sudah tersambung ke mixxx. Pada skenario pertama yaitu playlist musik otomatis, operator telah menyusun playlist lagu yang akan diputar secara otomatis oleh Mixxx. Playlist ini dapat diatur sebelumnya berdasarkan tema siaran saat itu. Ini sangat bermanfaat ketika operator tidak dapat terus-menerus mengendalikan siaran secara manual, sehingga sistem tetap dapat berjalan secara otomatis dan konsisten. Dalam mode ini, metadata seperti judul lagu dan nama penyanyi tetap dikirimkan ke server Icecast untuk ditampilkan pada website monitoring. Sementara pada skenario kedua, yaitu siaran menggunakan voice over, operator hanya bertugas sebagai konfigurasi awal, selanjutnya itu tanggung jawab penyiar dalam kelancaran siaran tanpa pantauan operator dan risiko kesalahan nya kecil. Skenario ini lebih fleksibel dan cocok untuk siaran interaktif atau program yang bersifat tematik, seperti talkshow atau sesi diskusi. Dalam mode ini, operator biasanya menggunakan mikrofon eksternal dan mixer audio tambahan untuk menghasilkan kualitas siaran yang lebih profesional. Selama aplikasi Mixxx berjalan, aliran audio dari kedua skenario tersebut dikirimkan secara bersamaan ke dua jalur, ke server Icecast untuk disiarkan melalui internet, dan ke pemancar FM RDS untuk disiarkan melalui gelombang radio lokal. Raspberry Pi berperan sebagai penghubung utama yang mengelola aliran audio ke kedua arah tersebut. Dengan sistem ini, jangkauan siaran menjadi lebih luas dan fleksibel karena mencakup pendengar dari dua media sekaligus, yaitu internet streaming dan radio FM analog.

Untuk memudahkan pemantauan status siaran oleh operator, dirancanglah sebuah website monitoring sederhana yang menampilkan informasi seperti status siaran, tema siaran, judul lagu yang diputar, nama penyanyi serta timeline lagu yang akan diputar. Website ini dibuat menggunakan framework Flask dan dikembangkan 15 langsung di Raspberry Pi. Untuk menjalankannya, operator membuka terminal pada Raspberry Pi dan masuk ke direktori proyek monitoring dengan perintah cd ~/monitoring-mixxx. Selanjutnya, menjalankan aplikasi Flask menggunakan perintah python app.py. Setelah aplikasi Flask aktif, pengguna dapat membuka browser dan mengetikkan alamat IP dari Raspberry Pi beserta port yang telah dikonfigurasikan untuk mengakses tampilan website monitoring. Tampilan ini dirancang agar sederhana namun informatif, sehingga memudahkan operator dalam memantau apakah Mixxx sedang berjalan, atau ada kendala. Fungsi dari sistem monitoring ini sangat krusial, terutama ketika operator harus melakukan multitasking. Dengan adanya website ini, operator dapat lebih mudah mendeteksi apabila terjadi gangguan pada proses siaran, seperti metadata tidak terkirim atau Mixxx dalam keadaan off. Website monitoring ini juga mendukung proses troubleshooting karena informasi yang ditampilkan bersifat langsung dan real-time. Dengan alur ini, sistem yang dirancang mampu mengintegrasikan proses siaran digital dan analog, serta menyediakan sistem monitoring real-time yang ringan dan dapat diakses dari jaringan lokal. Pendekatan ini tidak hanya mendukung efisiensi operasional, tetapi juga memperkuat kontrol kualitas terhadap proses siaran.

C. Instalasi dan Konfigurasi Perangkat

- Raspberry Pi

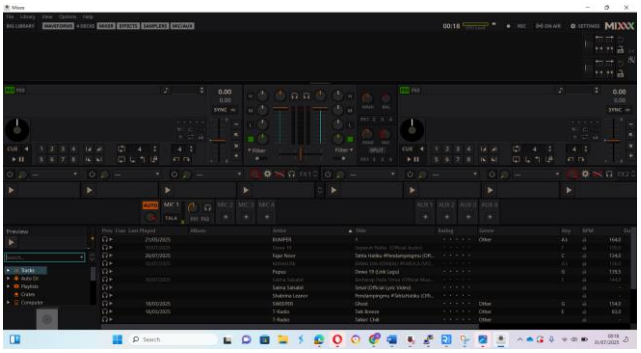
Perangkat utama yang digunakan pada tugas akhir kali ini adalah Raspberry Pi 4 Model B dengan kapasitas RAM 4 GB, kartu microSD 32 GB sebagai media penyimpanan. Sistem operasi yang digunakan adalah Raspberry Pi OS (64-bit), karena memiliki tingkat kompatibilitas tinggi dengan perangkat keras, mendukung berbagai driver yang dibutuhkan, dan stabil dalam menjalankan aplikasi Mixxx maupun server streaming Icecast. Proses instalasi dilakukan dengan men-flash citra Raspberry Pi OS ke microSD menggunakan aplikasi Raspberry Pi Imager. Setelah sistem operasi terpasang, dilakukan konfigurasi awal seperti pengaturan hostname, zona waktu, koneksi jaringan Wi-Fi. Konfigurasi jaringan diatur agar Raspberry Pi dapat terhubung ke jaringan lokal, sehingga mempermudah integrasi dengan laptop penyiar, server streaming, dan website monitoring. Selain itu, Raspberry Pi juga diatur agar dapat diakses secara jarak jauh melalui SSH (Secure Shell) untuk akses berbasis terminal, dan VNC (Virtual Network Computing) untuk akses tampilan desktop secara visual. Pengaturan ini memungkinkan operator melakukan pengelolaan, pemantauan, dan pemeliharaan sistem tanpa harus menghubungkan perangkat monitor, keyboard, dan mouse secara langsung ke Raspberry Pi.

GAMBAR 4
OS Raspberry Pi

Perangkat Raspberry Pi 4 Model B digunakan sebagai pusat pengendali sistem karena memiliki performa memadai, konsumsi daya rendah, dan ukuran yang minimalis.

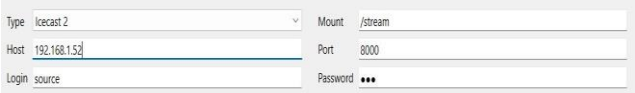
- Mixxx

Instalasi dan konfigurasi Mixxx dilakukan untuk mendukung pengelolaan siaran audio secara langsung (real-time). Mixxx merupakan perangkat lunak DJ mixing yang mampu mengatur, memutar, dan mengelola konten audio, serta terintegrasi dengan server streaming seperti Icecast dan FM RDS, sehingga sesuai digunakan pada sistem siaran radio berbasis Raspberry Pi. Proses instalasi dimulai dengan mengunduh installer dari situs resmi https://mixxx.org/download/ dan memilih versi terbaru yang sesuai dengan sistem operasi Windows 64-bit. Setelah instalasi selesai, Mixxx dibuka dan lagu dapat dimasukkan ke deck menggunakan metode drag and drop dari folder penyimpanan setelah sebelumnya diunduh dalam format mp3.



GAMBAR 5
Tampilan Mixxx

Agar dapat menyiarkan audio secara langsung, Mixxx dikonfigurasi untuk terhubung ke server *streaming* melalui menu *Preferences* pada bagian *Live Broadcasting*, dengan mengisi parameter *Host*, *Port*, *Mountpoint*, *Username*, dan *Password* sesuai pengaturan pada Icecast. Setelah koneksi berhasil, opsi Live Broadcasting diaktifkan hingga muncul notifikasi *Connected*. File audio kemudian dapat diputar, dan pendengar dapat mengakses siaran melalui alamat IP dan port yang telah dikonfigurasikan pada browser.



GAMBAR 6
Konfigurasi Mixxx

Mixxx memiliki beragam fitur yang dapat digunakan untuk mendukung proses siaran radio secara efektif. Dalam tugas akhir ini, pemilihan fitur dilakukan berdasarkan kebutuhan operasional siaran. Adapun fitur-fitur pada Mixxx yang digunakan dalam tugas akhir ini adalah sebagai berikut.

TABEL 1
Fitur Mixxx

Fitur Mixxx Yang Digunakan	Keterangan
Live Broadcasting	Mengirimkan audio siaran secara langsung ke server streaming seperti Icecast di Raspberry Pi
AutoDJ	Memutar lagu secara otomatis
Cue Points	Menentukan titik awal atau bagian tertentu pada lagu
Sound Effect	Menambahkan efek suara
Michrophone Input	Menghubungkan mikrofon untuk siaran
Multiple Decks	Memutar dan mengatur lebih dari satu lagu sekaligus
Recording	Merekam seluruh siaran dalam bentuk file audio
Library Management	Mengelola koleksi lagu secara terorganisir

- Website Monitoring

Website monitoring Mixxx berbasis Flask di Raspberry Pi dirancang untuk memudahkan pemantauan siaran radio secara real-time. Melalui website ini, operator dapat memantau status Mixxx, judul lagu, nama penyanyi, tema, dan timeline lagu yang sedang berjalan. Proses implementasi dimulai dengan instalasi Flask melalui terminal Raspberry Pi menggunakan perintah pip install flask. Perintah ini berfungsi mengunduh dan memasang seluruh pustaka yang dibutuhkan hingga proses selesai tanpa error. Tahap berikutnya adalah masuk ke direktori proyek ~/monitoring-mixxx yang berisi file utama app.py, folder templates untuk halaman HTML, serta folder static untuk file pendukung seperti CSS, JavaScript, dan gambar. File app.py berfungsi mengatur routing, mengambil metadata dari server Icecast, serta memperbarui data secara real-time. Pengeditan file dilakukan melalui nano app.py pada terminal raspberry pi.

Untuk mengatur tampilan halaman web, folder templates dibuka menggunakan perintah cd templates, lalu file index.html diedit menggunakan nano index.html. Pada file ini operator dapat menambahkan elemen-elemen seperti status Mixxx, judul lagu, nama penyiar, dan elemen visual lain seperti tabel atau grafik dengan menggunakan JavaScript. Setelah pengeditan selesai, file disimpan dengan mengetik ctr + o dan keluar dengan mengetik ctrl +x. Setelah melakukan pengeditan pada di app.py dan nano index.html. Jalankan website dengan mengetik python app.py pada terminal, tunggu sampe muncul proses running pada terminal yang artinya server Flask aktif.

pi@raspberrypi:~/monitoring-mixxx $ python app.py

** Serving Flask app "app" (lazy loading)*

**Environment: production*

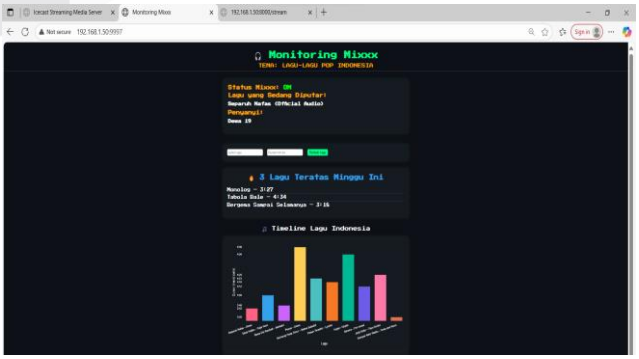
WARNING: This is a development server. Do not use it in a production deployment.

Use a production WSGI server instead.

** Debug mode: off*

**Running on http://0.0.0.0:5999/ (Press CTRL+C to quit)*

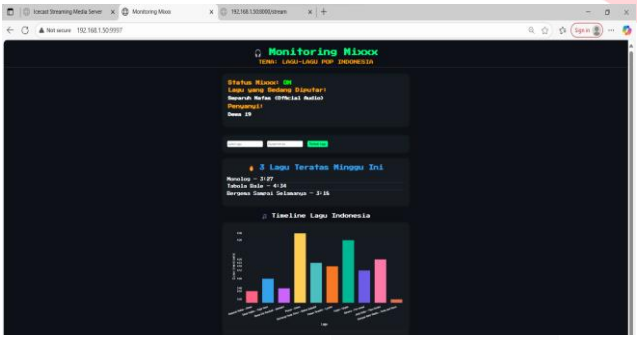
Setelah muncul keterangan Running on http://0.0.0.0:5999/ pada terminal, langkah berikutnya adalah membuka browser yang berada pada jaringan yang sama, lalu memasukkan alamat IP Raspberry Pi beserta port yang telah diatur (misalnya http://192.168.1.58:5999). Jika halaman berhasil dimuat, website monitoring dapat langsung digunakan untuk memantau status siaran secara real time.



GAMBAR 6
Pengujian Ip dan Port Website Monitoring

D. Desain Antarmuka Website Monitoringan

Desain antarmuka website monitoring dibuat untuk memudahkan operator maupun pengguna dalam memantau siaran radio berbasis Mixxx dan Icecast secara real-time. Tampilan menggunakan tema gelap (dark theme) agar nyaman digunakan dalam durasi lama, serta dirancang responsif sehingga dapat diakses melalui berbagai perangkat seperti laptop maupun ponsel. Informasi yang disajikan meliputi status Mixxx, judul lagu yang sedang diputar, nama penyanyi, tema siaran, serta timeline lagu yang divisualisasikan dalam bentuk grafik batang. Perancangan antarmuka dilakukan langsung menggunakan kombinasi HTML, CSS, dan JavaScript dengan dukungan pustaka Chart.js, tanpa melalui pembuatan mockup di Figma atau sejenisnya. Meski demikian, prinsip desain tetap mengutamakan kemudahan navigasi, keterbacaan, dan penataan elemen yang rapi.



GAMBAR 7
Desain Website Monitoring

Tabel 2
Fitur Website Monitoring

Elemen	Keterangan
Tema	Menampilkan topik atau genre lagu yang sedang diputar pada siaran.
Status Mixxx	Menampilkan status aplikasi Mixxx, apakah ON (terhubung) atau OFF (tidak terhubung).
Metadata Lagu	Menampilkan judul lagu dan nama penyanyi yang sedang diputar.
Timeline Siaran	Menampilkan durasi beberapa lagu dalam satu segmen siaran
Auto Refresh	memperbarui informasi di halaman secara otomatis secara berkala setiap 10 detik
3 Lagu Teratas Pekan Ini	Menampilkan daftar tiga lagu dengan jumlah pemutaran terbanyak selama periode satu minggu terakhir.
Tambah Lagu	Fitur yang memungkinkan pengguna mengajukan lagu baru untuk dimasukkan

IV. HASIL DAN PEMBAHASAN

A. Pengujian Operasional Siaran

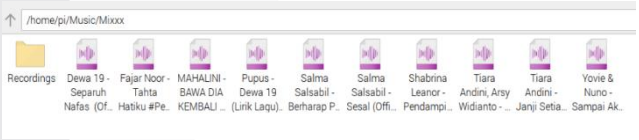
Pada tahap ini dilakukan pengujian operasional siaran berdasarkan dua skenario yang telah dirancang untuk memastikan sistem berjalan dengan baik dan informasi siaran tampil real-time pada website monitoring. Skenario pertama adalah pemutaran playlist musik otomatis menggunakan fitur Auto DJ di Mixxx, di mana lagu diputar berurutan tanpa campur tangan operator. Skenario kedua adalah input suara langsung melalui mikrofon (voice over), sehingga penyiar dapat berbicara dan didengar secara live oleh pendengar pada perangkat mereka.

- Pengujian Skenario Pertama (Playlist Msuik Otomatis)

Pada skenario pertama, pengujian dilakukan dengan mengaktifkan fitur Auto DJ pada Mixxx untuk memutar playlist secara otomatis tanpa intervensi operator. Tujuan pengujian ini adalah mengevaluasi kemampuan sistem siaran berjalan otomatis, sekaligus memverifikasi kestabilan pengiriman metadata lagu ke Icecast dan penampilannya di website monitoring. Selama siaran, website diharapkan menampilkan status Mixxx "ON", judul lagu, nama penyanyi, serta daftar lagu selanjutnya melalui timeline secara real-time, dengan auto-refresh tiap 10 detik. Hasil pengujian menunjukkan siaran berjalan lancar sesuai skenario dan seluruh informasi tampil di website tanpa keterlambatan maupun kesalahan data.

GAMBAR 8
Playlist Lagu di Mixxx

Pada sisi FM RDS, siaran lagu dilakukan melalui pemutaran audio yang berasal dari aplikasi Mixxx yang telah terhubung dengan sumber file di folder /home/pi/Music/Mixxx.



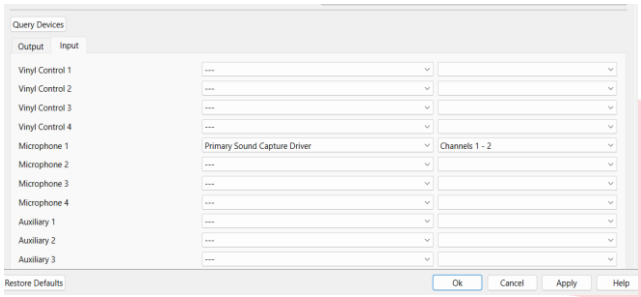
GAMBAR 9
Playlist Lagu di Folder Mixxx

- Pengujian Skenario Kedua (*Voice Over)*

Pada skenario kedua, siaran berasal dari suara langsung penyiar (voice over) untuk menyampaikan informasi, podcast, atau pengumuman secara real-time. Sebelum berbicara, operator harus mengaktifkan fitur “TALK” pada Mixxx agar mikrofon menyala, kemudian penyiar dapat berbicara melalui mikrofon yang tersambung ke laptop. Dengan demikian, Mixxx tidak hanya berfungsi memutar lagu, tetapi juga mendukung siaran langsung suara yang diteruskan ke layanan streaming sehingga pendengar di kedua platform dapat menerima siaran secara bersamaan.

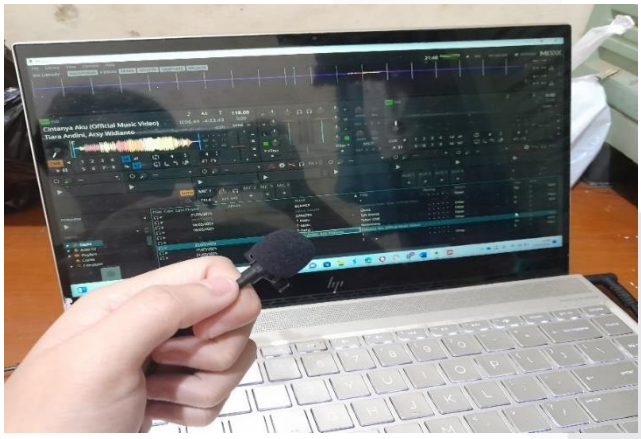
GAMBAR 10
Mikrofon Pada Mixxx

Setelah tombol TALK diaktifkan, colokkan mikrofon yang sudah disiapkan ke laptop baik melalui port audio maupun koneksi USB. Pastikan perangkat lunak sistem operasi telah mengenali mikrofon tersebut dan pengaturan input audio di Mixxx sudah diarahkan ke perangkat yang sesuai melalui menu Preferences → Sound Hardware.



GAMBAR 11
Pengaturan Sound Pada Mixxx

Setelah koneksi berhasil diverifikasi, penyiar dapat langsung mulai berbicara menggunakan mikrofon tersebut. Pada kondisi ini, Mixxx akan secara otomatis menyesuaikan tingkat audio dengan menghentikan atau menurunkan volume musik yang sedang diputar (*ducking*), sehingga suara penyiar menjadi prioritas utama. Mekanisme ini bertujuan untuk memastikan bahwa pesan atau informasi yang disampaikan terdengar dengan jelas oleh pendengar tanpa gangguan dari latar musik, baik pada platform streaming maupun website monitoring.



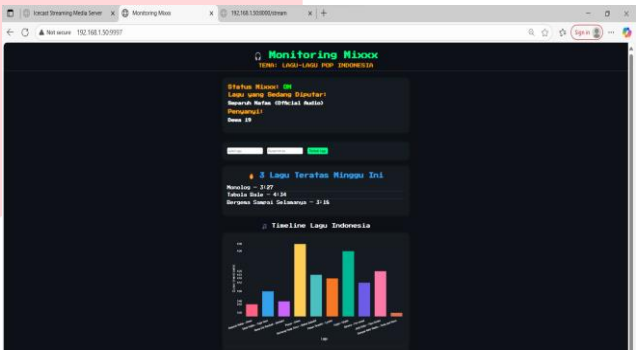
GAMBAR 12
Pengujian Input Suara

Pengujian dikatakan berhasil apabila suara yang diinputkan keluar pada website icecast. Selain itu, suara penyiar harus terdengar stabil tanpa adanya distorsi, noise berlebihan, atau hilangnya koneksi streaming selama proses siaran.

.

B. Pengujian Website Monitoring

Pengujian website monitoring dilakukan untuk memastikan antarmuka web mampu menampilkan informasi siaran secara real-time, akurat, dan dapat diakses melalui jaringan. Pengujian mencakup akses website, pemutakhiran data otomatis, dan kesesuaian metadata. Hasil pengujian menunjukkan halaman website dapat diakses melalui LAN tanpa kendala, responsif, serta menampilkan status Mixxx "ON" dan metadata lagu (judul, penyanyi, tema siaran) sesuai dengan kondisi siaran di aplikasi. Fitur auto-refresh setiap 10 detik berjalan optimal, di mana setiap pergantian lagu pada Mixxx langsung tercermin di website tanpa perlu refresh manual. Metadata yang ditampilkan juga konsisten dengan data di Mixxx, tanpa keterlambatan atau kesalahan informasi. Selain itu, fitur tambahan seperti timeline siaran dan daftar lagu populer mingguan tampil dengan baik. Integrasi antara Mixxx, Icecast, dan Flask terbukti berjalan lancar, sehingga website monitoring dapat diandalkan operator maupun pendengar untuk memantau siaran secara real-time.



GAMBAR 13
Aksesan Ip dan Port Website Monitoring

C. Pengujian Parameter Siaran

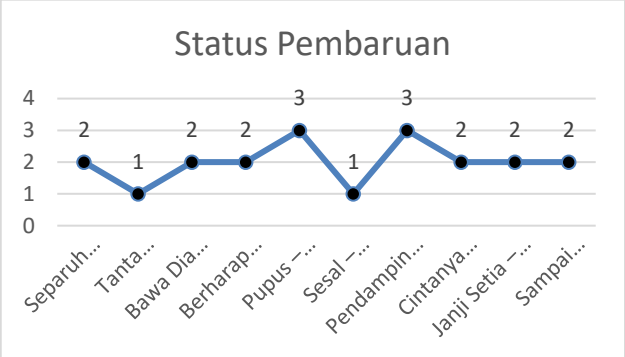
Pengujian parameter siaran dilakukan untuk memastikan sistem dapat menampilkan informasi melalui website monitoring dengan cepat, tepat, dan stabil. Pengujian meliputi waktu pengiriman metadata siaran dari Mixxx ke website, kestabilan pembaruan metadata selama siaran, serta kapasitas website dalam mengakomodasi akses pengguna secara bersamaan. Hasil pengujian dianalisis untuk menilai performa sistem dan menjadi acuan perbaikan ke depannya.

- Waktu Pengiriman Metadata Siaran dari Mixxx ke Website

Pengujian ini bertujuan mengetahui waktu yang dibutuhkan agar metadata siaran dari Mixxx muncul pada halaman website monitoring. Metadata mencakup judul lagu, nama penyanyi, dan keterangan program yang penting untuk memberi konteks siaran kepada pendengar maupun pihak pemantau. Kecepatan pengiriman metadata berpengaruh pada kemutakhiran informasi, sehingga keterlambatan dapat menurunkan kualitas pengalaman pengguna. Metode pengujian dilakukan dengan mengganti lagu di Mixxx, mencatat waktu perubahan metadata di aplikasi, lalu membandingkannya dengan waktu munculnya metadata yang sama di website. Selisih waktu dihitung sebagai durasi pengiriman, dan pengujian diulang beberapa kali untuk memperoleh hasil yang konsisten. Dari sini dapat diketahui sejauh mana sistem mampu memperbarui informasi siaran dengan cepat, sinkron, dan akurat dengan

kondisi siaran langsung.

TABEL 3
Grafik Status Pembaruan Metadata Siaran dari Mixxx ke Website



- Pengujian Kestabilan Pembaruan Metadata

Pengujian ini bertujuan memastikan metadata yang tampil di website monitoring selalu benar, akurat, dan diperbarui konsisten selama siaran. Metadata berisi judul lagu, nama penyanyi, atau keterangan program yang sedang berjalan. Kestabilan pembaruan menjadi penting karena keterlambatan, kesalahan, atau hilangnya informasi dapat menurunkan kualitas pengalaman pengguna dan kepercayaan terhadap sistem. Proses pengujian dilakukan dengan menjalankan siaran selama ±45 menit. Setiap pergantian lagu atau perubahan informasi dicatat waktunya, lalu dibandingkan dengan waktu munculnya metadata baru di website. Penguji juga mencatat jika terjadi jeda pembaruan, metadata kosong, atau informasi tidak sesuai. Hasil pengamatan dianalisis untuk mengetahui penyebabnya, apakah dari keterlambatan sistem, gangguan koneksi, atau masalah server. Dari sini dapat dinilai sejauh mana sistem mampu menjaga sinkronisasi metadata dalam satu segmen siaran serta menjadi bahan evaluasi perbaikan performa.

TABEL 4
Tabel Stabilitas Pembaruan Metadata Siaran per Interval Waktu

Waktu (Menit)	Kestabilan Website
1-10	Stabil
11-20	Stabil
21-30	Stabil
31-40	Stabil
41-45	Stabil

- Pengujian Kapasitas Akses Website Monitoring

Pengujian ini dilakukan untuk mengetahui sejauh mana website monitoring mampu menangani akses dari beberapa pengguna secara bersamaan tanpa penurunan kinerja. Dalam praktiknya, website dapat diakses operator maupun pendengar pada waktu yang sama sehingga perlu dipastikan tetap stabil. Pengujian dilakukan dengan menggunakan beberapa perangkat seperti laptop, komputer, dan ponsel yang terhubung ke jaringan sama dan mengakses website secara bersamaan. Selama pengujian diamati kecepatan pemuatan halaman, konsistensi metadata di tiap perangkat, serta kemungkinan gangguan seperti tampilan tersendat, pembaruan terhenti, atau keterlambatan muat ulang. Hasil pengujian memberikan gambaran batas kapasitas optimal website, menjadi dasar untuk menilai apakah sistem sudah cukup handal atau memerlukan peningkatan melaui optimasi kode, konfigurasi server, maupun penambahan sumber daya perangkat keras, sehingga dapat dijadikan acuan pengembangan di masa mendatang.

TABEL 5
Tabel Hasil Pengujian Kapasitas Akses Website Monitoring secara Bersamaan Pada Beberapa Perangkat

Perangkat	Sistem Operasi	Browser	Keterangan
Laptop	Windows 10	Chrome, Edge	Stabil
Android	Android 10	Chrome	Stabil
Iphone	Ios 16	Safari	Stabil

C. Analisis Hasil Pengujian

Berdasarkan hasil uji coba, sistem siaran radio yang dibangun dengan Mixxx, Icecast, FM RDS, dan website monitoring mampu bekerja sesuai tujuan, baik saat memutar playlist lagu maupun menggunakan fitur voice over, dengan kualitas suara tetap jelas meski terdapat jeda beberapa detik. Website monitoring berhasil menampilkan informasi siaran secara real-time mencakup judul lagu, nama penyanyi, status Mixxx, tema siaran, tiga lagu teratas, fitur tambah lagu, serta timeline lagu dalam bentuk grafik batang yang memudahkan operator memantau durasi dan urutan. Pembaruan metadata berjalan stabil dengan rata-rata waktu 1-3 detik, tetap responsif di berbagai perangkat, dan mampu menampung hingga 3 operator melalui LAN tanpa penurunan kinerja. Selama satu segmen siaran ±45 menit, koneksi Mixxx, Icecast, dan FM RDS tetap stabil, serta data yang ditampilkan konsisten di semua perangkat. Secara keseluruhan, sistem ini memenuhi tujuan dari sisi kecepatan, ketepatan, dan kestabilan, meski pengembangan lanjutan tetap dapat dilakukan, seperti menambah fitur kontrol siaran langsung dari website, mempercepat interval pembaruan metadata, serta menguji skala jaringan lebih luas.

V. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem siaran radio komunitas berbasis Raspberry Pi, dapat disimpulkan bahwa sistem yang dikembangkan berhasil mengintegrasikan Raspberry Pi sebagai server utama, Mixxx sebagai aplikasi penyiaran, serta website monitoring untuk menampilkan informasi siaran secara real-time. Pengujian menunjukkan sistem berjalan stabil, di mana Raspberry Pi mampu menjalankan proses siaran tanpa kendala, Mixxx berhasil menginput konten beserta metadata siaran, dan siaran dapat tersalurkan melalui radio internet serta FM RDS. Website monitoring juga dapat menampilkan status Mixxx, nama penyanyi, judul lagu, tema siaran, serta timeline dalam satu segmen untuk mempermudah operator memantau jalannya siaran. Secara keseluruhan, sistem ini membuktikan bahwa pendekatan berbasis open-source dengan perangkat hemat daya seperti Raspberry Pi sangat cocok untuk keperluan siaran komunitas, pendidikan, maupun skala kecil, serta memiliki potensi pengembangan lebih lanjut sesuai kebutuhan dan perkembangan teknologi.

REFERENSI

[1] A. Mustofa, S. Kusmaryanto, R. Ambarwati, and B. Dwika Nanda, “Transmitter Gelombang Frekuensi Radio (RF) FM Berbasis Raspberry Pi,” Jurnal

FORTECH, vol. 3, no. 2, hlm. 85–92, Jan 2023, doi: 10.56795/fortech.v3i2.105.

[2] S. Pella and H. F. J. Lami, “Pengembangan sistem kontrol otomatis menggunakan sensor pada rumah pintar,” J. Media Elektro, vol. 16, no. 1, hlm. 15–22, Mar 2022, 10.35508/jme.v16i1.4786. Universitas Nusa Cendana, Kupang, doi: 10.35508/jme.v16i1.4786.

[3] T. M. Diansyah and E. Ilyanda, “Rancangan media penyimpanan berbasis mikrokontroler menggunakan Raspberry Pi sebagai mini server portabel,” Prosiding Seminar Nasional Teknologi Informasidan

[4] T. R. M. Fitrah, N. Yudha, and Roslidar, “Perancangan sistem pengontrolan dan monitoring suhu berbasis IoT menggunakan Telegram,” J. Komput. Inform. Teknol. dan Elektro, vol. 7, no. 1, hlm. 7–14, 2021, doi: 10.21428/72955.

[5] A. Triyono, “Implementasi Raspberry Pi untuk aplikasi signate board jadwal kuliah berbasis web dengan sistem operasi Linux,” Tugas Akhir, STMIK Amikom Yogyakarta, 2018

[6] WillyC6, “Internet Radio/ Web Radio With Raspberry Pi 3 (headless) : 8 Steps - Instructables,” Instructables. Diakses: 9 Februari 2025. [Daring]. Tersedia Pada: https://www.instructables.com/Internet-Radio-Web-Radio-With-Raspberry-Pi 3-headl/

[7] Mixxx Development Team, “Introduction to Mixxx,” Mixxx 2.3 User Manual. [Daring]. Tersedia tps://manual.mixxx.org/2.3/en/chapters/introduction.ht

mlT. “Mixxx: Towards Novel DJ Interfaces.” 9 Pada: H. Andersen,

[8] A. A. Prasetyo, “Mari mengenal Icecast,” Andika Adi Prasetyo Blog, Mar 2017. Diakses: Februari 2025. [Daring]. Tersedia Pada: https://andikaadiprasetyo.blogspot.com/2017/03/mari-mengenal-icecast.html

[9] Rachmawati, Munawar, and S. Tari, “Analisis quality of service (QoS) radio streaming menggunakan Icecast pada Wi-Fi 802.11n,” Seminar Nasional Teknologi Informasi dan Komunikasi, Mar 2023.

[10] A. Lawrence, “Apa itu Flask? Pengertian, kelebihan, dan kegunaannya,” Makinrajin.com, 2023. [Daring]. https://makinrajin.com/blog/flask-adalah/

[11] I. Rahadian, “Penggunaan Flask untuk Pemula”, Modul Pelatihan Internal (Pelatihan Python Dasar), Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, Bandung, Indonesia, hlm. 1–4, Mar. 2022.

[12] I. Rochmawati, “Analisis user interface situs web Iwearup.com,” Universitas Bina Sarana Informatika, 2019. [Daring]. Tersedia: http://www.iwearup.com. Diakses: 8 Agustus 2025.

[13] S. Mariko, “Aplikasi website berbasis HTML dan JavaScript untuk menyelesaikan fungsi integral pada mata kuliah kalkulus,” J. Inov. Teknol. Pendidik., vol. 6, no. 1, hlm. 80–91, 2019, doi: 10.21831/jitp.v6i1.22280.

.